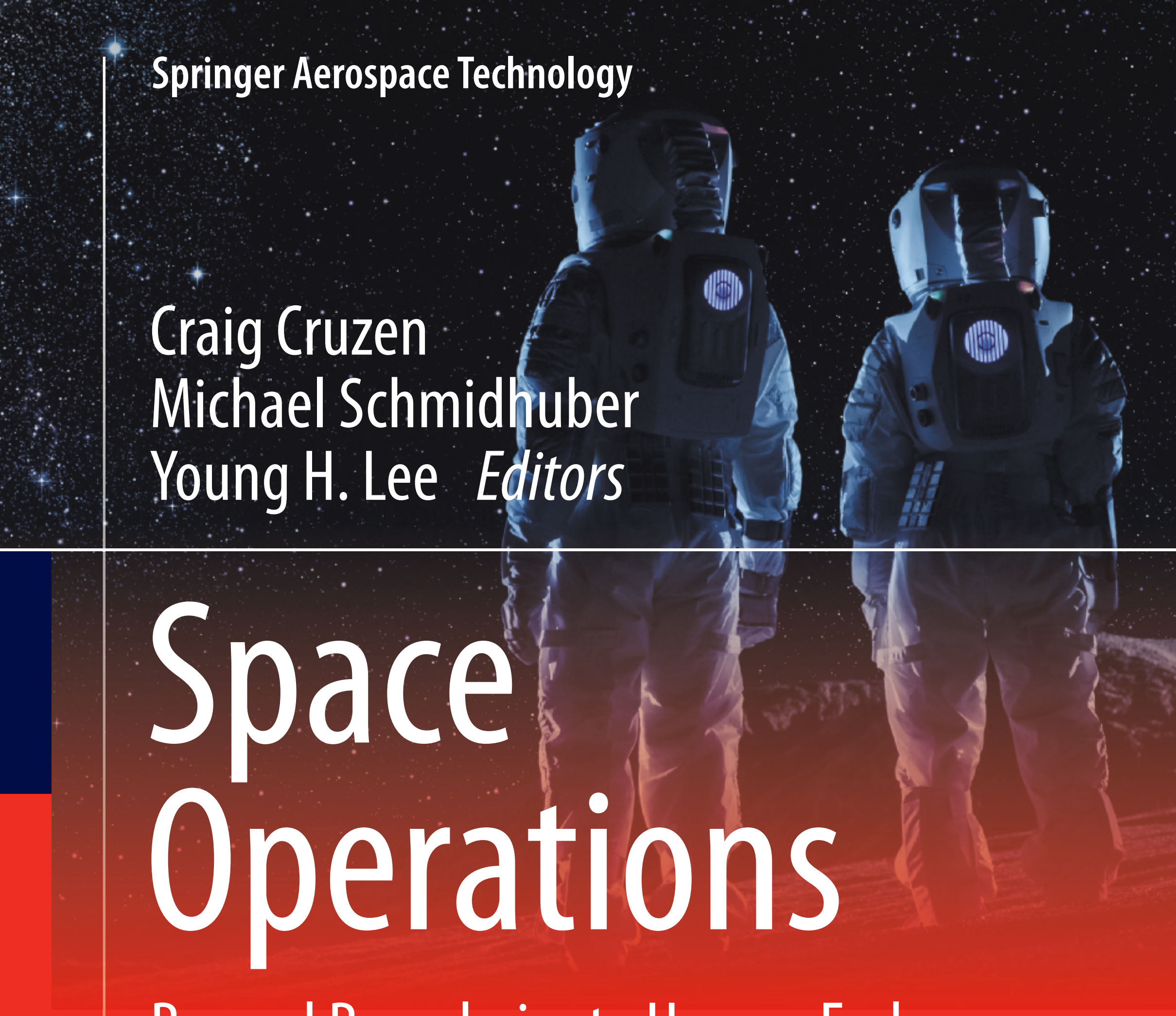
Springer Aerospace Technology



Craig Cruzen
Michael Schmidhuber
Young H. Lee *Editors*

Space Operations

Beyond Boundaries to Human Endeavours

Springer

Springer Aerospace Technology

The series explores the technology and the science related to the aircraft and spacecraft including concept, design, assembly, control and maintenance. The topics cover aircraft, missiles, space vehicles, aircraft engines and propulsion units. The volumes of the series present the fundamentals, the applications and the advances in all the fields related to aerospace engineering, including:

- structural analysis,
- aerodynamics,
- aeroelasticity,
- aeroacoustics,
- flight mechanics and dynamics
- orbital maneuvers,
- avionics,
- systems design,
- materials technology,
- launch technology,
- payload and satellite technology,
- space industry, medicine and biology.

The series' scope includes monographs, professional books, advanced textbooks, as well as selected contributions from specialized conferences and workshops.

The volumes of the series are single-blind peer-reviewed.

To submit a proposal or request further information, please contact:

Mr. Pierpaolo Riva at pierpaolo.riva@springer.com (Europe and Americas)
Mr. Mengchu Huang at mengchu.huang@springer.com (China)

The series is indexed in Scopus and Compendex

More information about this series at https://link.springer.com/bookseries/8613

Craig Cruzen · Michael Schmidhuber ·
Young H. Lee
Editors

Space Operations

Beyond Boundaries to Human Endeavours

Springer

Editors

Craig Cruzen
NASA Marshall Space Flight Center
Huntsville, AL, USA

Young H. Lee
NASA Jet Propulsion Laboratory,
California Institute of Technology
Pasadena, CA, USA

Michael Schmidhuber
German Space Operations Center
German Aerospace Center (DLR)
Weßling, Germany

ISSN 1869-1730 ISSN 1869-1749 (electronic)
Springer Aerospace Technology
ISBN 978-3-030-94627-2 ISBN 978-3-030-94628-9 (eBook)
https://doi.org/10.1007/978-3-030-94628-9

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword

During the 2019 fall meeting of the International Committee on Technical Interchange for Space Mission and Ground Data Systems, also known as SpaceOps, the technical program committee compiled the program for the SpaceOps 2020 conference in Cape Town, South Africa. We were "ready to move" to the presentation of an exciting conference, the first time in Africa. Little did we know that the world was about to change unprecedentedly. By February 2020, the South African Organizing Committee knew that they would require some form of contingency. Understandably, in March 2020, we made the decision to postpone SpaceOps 2020 because of the COVID-19 pandemic. At this stage, we still hoped that the pandemic would have resolved itself within a year and we could host an in-person conference. By June 2020, it was clear we were in for the long haul and we needed to decide to either cancel or host a virtual event. After various difficult discussions, we decided that the South African National Space Agency (SANSA) would host a virtual event, as we believed that the community would want to get together in some form to share their work and engage in an exchange of ideas.

The technical program committee decided on new criteria for eligibility to the conference because of the special circumstances. We gave those who already submitted papers the opportunity to re-submit their work either as is, or with updates to the postponed conference. Then, we also accepted new submissions for the virtual edition of the conference. The virtual format of the conference required authors to pre-record video presentations that were placed in a repository for on-demand viewing. In the "live" program, the papers represented in this book were presented in a "live" manner with authors available for questions and answers.

With the virtual conference being free, the papers got exposure to a far larger (be it sometimes non-technical) audience. Using the opportunity to expose a new audience to space operations, the plenary program featured some of the more popular themes in space operations: human spaceflight, space situational awareness, commercialization of space and specifically for this conference, a plenary on space operations during the pandemic. The technical program also featured papers related to the special circumstances of this period.

The topics for SpaceOps 2021 saw an expansion from the previous conference, with papers organized in the following topics presented:

- Mission Design and Management
- Operations Concepts
- Flight Execution
- Ground Systems Engineering
- Data Management
- Planning and Scheduling
- Guidance, Navigation, and Control
- Communications Architectures and Networks
- Human Spaceflights and Operations
- Cross Support, Interoperability, and Standards
- Human Factors, Training and Knowledge Transfer
- Space Transportation System
- Artificial Intelligence for Space Operations
- Cybersecurity for Space Operations
- Safety and Sustainability of Space Operations
- Beyond Boundaries in Human Endeavour

Apart from the conference specific topic, we added three new topics: artificial intelligence for space operations, cybersecurity and safety and sustainability. These showed some of the new emphasis in space operations, with the artificial intelligence topic growing even between the postponed conference and the virtual event! In all, 263 papers in the 16 topics were presented from 25 countries with new nations participating, likely because of the virtual nature and the free entrance. The free registration allowed for a much larger attendance, with 2,730 registrations and 1,935 actual participants. Let us hope the new exposure will bode well for the future SpaceOps conferences.

While the Virtual Edition exposed the work of the global SpaceOps community to a far broader audience, we missed the personal interaction and the opportunities for exchange of ideas not just in the official program but also in the margins of the conference and the bitterly missed social events. When we return to in-person conferences in 2023 (now odd year conferences), we will certainly have a new appreciation for in-person interaction, but hopefully we have learnt from the pandemic and improved our operations concepts and environment to be even more resilient and efficient when we do space operations.

This book presents a selection of the best papers from the SpaceOps 2021 conference. It should be noted that the conference was filled with technical work of exceptional quality. With a wide range of topics, there will be something for every enthusiast of space operations.

In 2018, when the theme of the 2020 conference "Beyond border in Human Endeavor" was decided on, we were living in a world where populism reared its head among the leaders of nations and progressively the nations shut themselves off from one another. We wanted to show that the international space operations community recognized that most borders are artificial in their nature and that successful missions require cooperation across national boundaries but also across the public and private sectors and industry and academia. The advent of the pandemic further emphasized this tighter national control, but thankfully the tools were available to overcome these restrictions and continue cooperation.

Of course, the technical program would not have been possible were it not for the authors, participants, the topic chairs, and the collection of volunteers who reviewed the papers for the program and ensured that we had the required submissions on time and with the highest level of quality. My sincere thanks to all who contributed to the success of the conference and the publication of this collection of manuscripts, a valuable keepsake for the authors and contributors to SpaceOps 2021 virtual edition.

Eugene Avenant
South Africa National Space Agency (SANSA)
SpaceOps 2021 Virtual Edition Technical Program
Committee Chair
Pretoria, South Africa

Preface

The SpaceOps organization was founded in 1990 to foster technical discussions on all aspects of space mission operations and ground data systems among space agencies, academic institutions, space operators and industry. The organization aims to facilitate and encourage the exchange of managerial and technical information via periodic symposia concerning spacecraft, ground systems and mission operations. Other formal and informal meetings, workshops and publication of managerial and technical information are also significant objectives.

Traditional SpaceOps conferences have been organized on a biennial basis and are hosted by a selected participating space agency. Conference features include technical sessions, plenary sessions, poster presentations, social and networking events, industry exhibition and sponsorship opportunities. The sixteenth symposium was scheduled to be hosted by the South African National Space Agency (SANSA) and held in Cape Town, South Africa, in May of 2020 unfortunately was postponed due the worldwide pandemic. Ultimately SpaceOps 2021 was held as a virtual/online event May 3-5, 2021. Its theme was **Beyond Boundaries to Human Endeavours**.

Following a precedent set at the 2006 conference, the organizers of SpaceOps 2021 decided to publish a book of “best” papers reflecting representative subjects presented at the conference. The SpaceOps conference topic chairs reviewed and selected papers for this book. The topic chairs and technical organizers included:

Eugene Avenant (SANSA)	Young Lee (NASA/JPL)	Alice Bowman (JHU/APL)
David Welch (LASP)	Sabrina Eberle (DLR)	Hubert Fraysse (CNESS)
Keyur Patel (NASA/JPL)	Francois Jocteur-Monrozier (CNES)	Hamid Salim (MYSA)
Zeina Mounzer (TPZ)	Suzanne Dodd (NASA/JPL)	Martin Wickler (DLR)
Vladimir Nazarov (IKI)	Shinichi Nakamura (JAXA)	Fabio D’Amico (ASI)

(continued)

(continued)

Brian Giovannoni (NASA/JPL)	Klaus-Juergen Schulz (ESA)	Thomas Müller (DLR)
Cesare Capararo (Altecspace)	Gian-Paolo Calzolari (ESA)	Christophe Belzile (CSA)
Sean Burns (EUMETSAT)	Michael Schmidhuber (DLR)	Julio Monreal (ESA)
Craig Cruzen (NASA/MSFC)	Gérard Galet (CNES)	Saeed Hussain AL Mansoori (MBRSC)
Thierry Levoir (CNES)	Kevin Marston (EUMETSAT)	Andrew Monham (EUMETSAT)
Alexi Glover (ESA)	Gladys Magagula (SANSA)	Harry Shaw (NASA/GSFC)

The selected papers were examined to assess the technical accuracy and completeness of the information. Then, they were edited for clarity, logical organization and emphasis of importance to space operations.

The editors wish to express special gratitude to the conference topic chairs, the organizers, the SpaceOps Executive Committee, and the SpaceOps Communications, Outreach, and Publications Group; all were instrumental in the development and publication of this book.

Finally, and most importantly, the editors would like to acknowledge the authors with their contributions to this publication. Without their hard work and diligence, this esteemed compilation of conference best papers would not have been possible.

Huntsville, USA
Weßling, Germany
Pasadena, USA
November 2021

Craig Cruzen
Michael Schmidhuber
Young H. Lee

Contents

Mission Management

Ground Systems and Software

Flight Operations

Mission Management

An International Standard Procedure for Managing Spacecraft Emergency Cross Support (SECS)

Lucy Santana, LaNetra Tate, Jean-Marc Soula, Tsutomu Shigeta, Hirokazu Hoshino, Fabio D'Amico, Sangil Ahn, Nikki Desch, Wendy Evans, Peter Willburger, John Reynolds, Catherine Barclay, Jean-Michel. Roquebert, Heather Stewart, and Thomas Beck

Abstract In September 2019, the Interagency Operations Advisory Group (IOAG) Spacecraft Emergency Cross Support (SECS) Working Group presented the SECS Standard Operating Processes and Procedures (SOP) to the annual IOAG conference (IOAG-23). The SOP presents a harmonised approach for emergency recovery support entailing processes and services that reduce response times related to critical emergency situations. The implementation of these services will be achieved by:

- Encouraging member agencies to follow the guidelines outlined in the SOP
- Encouraging member agencies to establish arrangements that enable execution of SECS

L. Santana · T. Beck
European Space Agency (ESA), Paris, France

L. Tate · N. Desch · W. Evans · C. Barclay · H. Stewart
National Aeronautics and Space Administration (NASA), Washington, D.C, USA

J.-M. Soula · J.-Michel. Roquebert
Centre National d'Études Spatiales (CNES), Paris, France

T. Shigeta · H. Hoshino
Japan Aerospace Exploration Agency (JAXA), Chofu, Japan

F. D'Amico
Agenzia Spaziale Italiana (ASI), Rome, Italy

S. Ahn
Korea Aerospace Research Institute (KARI), Daejeon, South Korea

P. Willburger
Deutsches Zentrum Fuer Luft- Und Raumfahrt (DLR), Cologne, Germany

J. Reynolds (✉)
HessenTech GmbH, Griesheim, Germany
e-mail: john.reynolds@esa.int

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_1

The SOP provides guidance to agency Service Users, i.e. any agency mission, current or future, that may require additional support beyond their routine and contingency support, in order to recover from an Emergency Condition that threatens the life of the spacecraft. Initially, support is limited to IOAG member Agencies; however, the support, as defined, has the potential to expand the "service user" and "service provider" base. In addition, the IOAG is surveying interest from Commercial Service Providers for participation. The SECS SOP defines three specific categories of standard support that can be made available by service providers.

- Committed Support
- Acknowledgement Support
- Non-Registered Support

The SOP describes the "recovery" services that service providers may perform, covering a wide variety of contingency situations, including:

- Downlink or uplink engineering services for diagnostics (no real-time telemetry or telecommand transfer)
- Tracking data delivery and/or processing
- Full telemetry, tracking and command (TT&C) Services

These services require the support of various branches of a service provider's infrastructure, namely:

- Flight dynamics
- Ground stations
- Data communications
- Asset scheduling

The SOP deals with items of particular interest to mission managers, including what constitutes a spacecraft emergency, radio frequency (RF) licensing, points of contact, and the SECS asset database. As a "Proof of Concept", various demonstration exercises were performed utilising stations from multiple agencies tracking spacecraft which, although not actually in emergency, require preparation activities in line with a contingency acquisition. Completion of this SOP is a major milestone for this working group. The document focuses on emergency support for robotic missions. The working group plans to expand its scope to encompass emergency support for human spaceflight missions. More information on the IOAG can be found on the following website: https://www.ioag.org. The SOP can be located on the IOAG by following the "Documents" link then the "Public" link, or can be found directly on the website: https://www.ioag.org/Public%20Documents/IOAG%20Spacecraft%20Emergency%20Cross%20Support%20SOP.pdf.

Abbreviations

AOS	Acquisition of Signal
CCSDS	Consultative Committee for Space Data Systems
DSN	Deep Space Network
EbNo	Energy per Bit to Noise ratio
EsNo	Energy per Symbol to Noise Ratio
ESTRACK	European Space Tracking Network
FCT	Flight Control Team
GEO	Geostationary Orbit
IOAG	Interagency Operations Advisory Group
ITU	International Telecommunication Union
JPL	Jet Propulsion Laboratory
LEO	Low Earth Orbit
LEOP	Launch and Early Orbit Phase
MOCC	Mission Operations Control Centre
ODM	Orbit Data Message
OLP	Open Loop
RF	Radio Frequency
SECS	Spacecraft Emergency Cross Support
SGICD	Space to Ground Interface Control Document
SLE	Space Link Extension
SOP	Standard Operating Procedures
TC	Telecommand
TLE	Two Line Elements
TLM	Telemetry
TT&C	Telemetry, Tracking and Command

1 Introduction

The Spacecraft Emergency Cross Support Working Group (SECWG) SOP defines the grounds for declaring a spacecraft emergency as the following:

- Spacecraft emergency mode is the anomalous state of the spacecraft in which its persistence will cause the spacecraft's loss entirely or losing spacecraft's essential facilities (payload excluded).
- For human spaceflight missions, any of the above conditions or any external or internal conditions that could negatively affect the health and safety of the crew members.

Neither a ground segment failure by itself, nor loss of science or payload data, is considered a direct cause for declaring a spacecraft emergency.

1.1 Current IOAG Membership

IOAG Members
Agenzia Spaziale Italiana (ASI)
Centre National d'Études Spatiales (CNES)
Canadian Space Agency (CSA)
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
European Space Agency (ESA)
Japan Aerospace Exploration Agency (JAXA)
National Aeronautics and Space Administration (NASA)
United Kingdom Space Agency (UKSA)
IOAG Observers
Australian Space Agency (ASA)
Chinese National Space Administration (CNSA)
Indian Space Research Organisation (ISRO)
Korea Aerospace Research Institute (KARI)
Roscosmos State Corporation for Space Activities (ROSCOSMOS)
South African National Space Agency (SANSA)
United Arab Emirates Space Agency (UAESA)

1.2 Consequences of a Lost Mission

The unplanned end of a mission results in a myriad of detrimental consequences.

- Losing significant financial investment in the spacecraft's development and its related infrastructure such as control systems and ground segment.
- The loss of data to the science community.
- Potential danger to other missions, if the spacecraft is positioned in an orbital trajectory that is also occupied by other spacecraft.
- Spacecraft operations often require a relatively large team of 1st, 2nd and 3rd line support personnel. Loss of a spacecraft can have serious impacts on the morale of individual personnel, the team and the community at large.

1.3 Current User Community

The Standard Operating Processes and Procedures (SOP) initially applies to IOAG member agencies, although the use cases might apply to non IOAG agencies in the future. The SOP could be considered as a catalogue of services and assets for any mission manager whether the project is in its planning stage or, if already in orbit, the project has identified a significant shortcoming in its operational strategy.

2 The Spacecraft Emergency Cross Support Working Group (SECSWG)

The IOAG was chartered to create a working group dedicated to the provision and standardisation of recovery cross support to spacecraft in emergency conditions, i.e. SECSWG. The Terms of Reference for the Working Group are as follows:

- Encouraging member agencies to follow the guidelines outlined in a SECS SOP when preparing for and coordinating cross support for a spacecraft emergency
- Encouraging member agencies to establish arrangements that enable the execution SECS SOP.

The current participating members in the Working Group are a subset of the IOAG membership, (i.e. ASI, CNES, DLR, ESA, JAXA, KARI, NASA). A major milestone for the Working Group was the presentation of the SOP to the IOAG for adoption and issue with the member agencies.

3 Support Scenarios

The amount of coordination and preparation for a service provider and a service user directly affects the available response time to support an emergency. Early coordination can significantly reduce the response time during an emergency; however, preparing a ground station for an emergency support that may never need incurs costs both to the user and the provider. Reflecting this, the SOP describes three types of service support scenarios that vary in the amount of preparation and investment prior to an emergency and the associated response times that can be expected.

3.1 Committed Support

A service user has contacted a service provider and, through established agreements, the provider has agreed that some of its assets can be used in the SECS process guaranteeing functioning pre-validated TT&C services. The service user has previously identified the assets that it considers appropriate to the recovery of the spacecraft, for example:

- The selected ground stations are tailored to support the spacecraft acquisition downlink and/or uplink

- RF licenses and International Telecommunication Union (ITU) filing have been confirmed
- The configuration has been validated for telemetry (TLM) recovery and/or command transmission
- End to end communications infrastructure has been validated for data transfer
- The service provider will periodically test the configuration and ground communications

3.2 Acknowledged Support

A service user has contacted a service provider, and the service provider agreed that some of its assets could be potentially used in the SECS process. Such identified assets were considered appropriate to the recovery of the spacecraft, i.e.

- The selected ground stations are tailored to provide the Service User with Engineering/Diagnostic services at a minimum.
- Standard TT&C services can be provided depending on the level of support readiness such as availability of ground communications lines, ITU filing and RF Licenses.
- The ground station configuration may not have been pre-validated and periodic testing will not be performed.

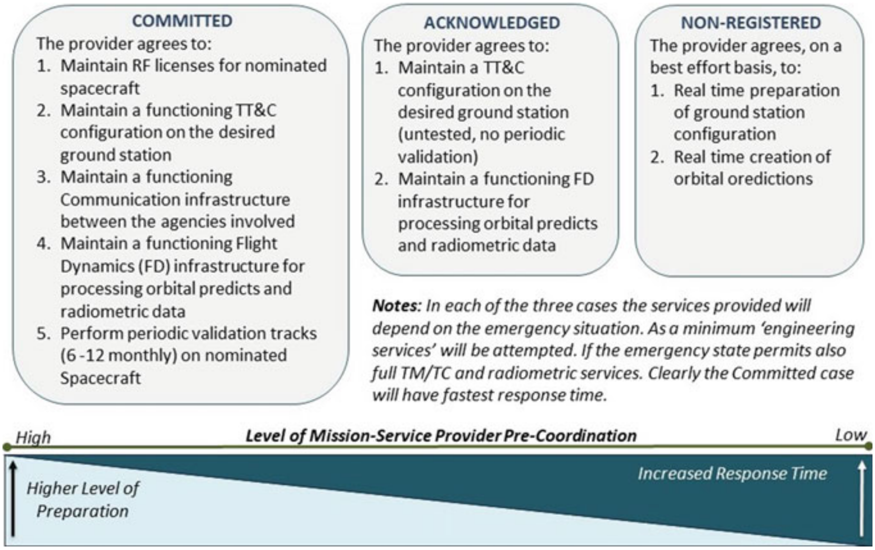
Effectively this means that a service user and service provider may negotiate the level of support readiness to an affordable level so that further standard TT&C services are available for emergency support.

3.3 Non-Registered Support

A service user that has not coordinated SECS services with a service provider prior to an emergency, i.e. ground segment and infrastructure, is not immediately available. Although workable, this approach dramatically increases the response time for emergency support.

3.4 Support Overview

The support scenarios can be thus summarised:



COMMITTED

The provider agrees to:

1. Maintain RF licenses for nominated spacecraft
2. Maintain a functioning TT&C configuration on the desired ground station
3. Maintain a functioning Communication infrastructure between the agencies involved
4. Maintain a functioning Flight Dynamics (FD) infrastructure for processing orbital predicts and radiometric data
5. Perform periodic validation tracks (6 -12 monthly) on nominated Spacecraft

ACKNOWLEDGED

The provider agrees to:

1. Maintain a TT&C configuration on the desired ground station (untested, no periodic validation)
2. Maintain a functioning FD infrastructure for processing orbital predicts and radiometric data

NON-REGISTERED

The provider agrees, on a best effort basis, to:

1. Real time preparation of ground station configuration
2. Real time creation of orbital oredictions

Notes: *In each of the three cases the services provided will depend on the emergency situation. As a minimum 'engineering services' will be attempted. If the emergency state permits also full TM/TC and radiometric services. Clearly the Committed case will have fastest response time.*

High
Level of Mission-Service Provider Pre-Coordination
Low
Higher Level of Preparation
Increased Response Time

4 Categorisation of Services that Comprise Recovery Operations

While the general processes outlined in the SOP could apply to most emergency cases, they are generally for service users that comply with the applicable Consultative Committee for Space Data Systems (CCSDS) core standards as specified in the IOAG Service Catalogue#1 [1]. Non CCSDS compliant Users may be limited to engineering services, if available.

The SOP foresees the provision of four types of service, namely:

- Flight Dynamics Services
- Engineering Services
- Standard TT&C Services
- Network Services (Data Transfer)

4.1 Flight Dynamics Services

Committed and acknowledged support scenarios are expected to have established functioning infrastructure for transferring and processing of the spacecraft trajectory prior to the occurrence of an emergency. Thus, trajectory predictions can be created by the provider's flight dynamics and provided to the supporting station to initiate program track on the spacecraft. In the event of no pre-existing infrastructure, the service user is required to provide the service provider with the latest and/or expected trajectory predictions for the spacecraft in the CCSDS Orbit Data Message (ODM) format, recommended standard CCSDS 502.0-B-2 [2]. Any radiometric data collected by the supporting ground station will be delivered to the service user for processing and orbit determination. Alternatively, the service provider flight dynamics may be requested to provide orbit diagnostics and orbit determination. As mentioned previously all data transfer must conform to the CCSDS ODM Format.

4.2 Engineering Services

If a spacecraft is in an emergency condition, some functionality is lost due either to a system failure on the platform or non-nominal trajectory, which may prevent the provision of standard TT&C services. In such cases, the service provider can provide engineering services to determine the status, attitude, or orbit of the spacecraft to assist in the recovery from the anomalous condition.

4.2.1 Downlink Engineering Search Services

The Spacecraft Search Service category applies to cases in which the spacecraft trajectory is non-nominal, thus preventing ground stations from acquiring the downlink. These cases typically occur after launch, erroneous injection, or after a trajectory correction manoeuvre in which the on-board thrusters did not perform as predicted. The search techniques used to locate the spacecraft may include:

- Use of an antenna with wider beam-width, i.e., acquisition aid with smaller aperture but with link budget limitations
- Antenna scanning, predefined search pattern, e.g. conical scan
- Along track search, applying time offsets to antenna predicts
- Multiple trajectories, flight dynamics provide predictions corresponding to fixed error cases, e.g. ±3 sigma against a nominal case.

If the search is successful, the first outcome of the service is confirmation that the spacecraft transmitter is “on”. The antenna should then continue to track the spacecraft and collect passive measurements, e.g. antenna angles when in auto-track and raw 1-Way doppler, to allow computation of a new trajectory.

4.2.2 Downlink Engineering Signal Analysis Services

This category applies to cases in which the spacecraft downlink signal is non-nominal, e.g. the ground station cannot lock, demodulate or decode the received signal. Assuming the spacecraft transmitter is not functioning correctly, these services attempt to analyse the signal and provide the service user with useful information, e.g. spectrum analyser display, automatic gain control (AGC), energy per symbol to noise ratio (Es/No), energy per bit to noise ratio (Eb/No) recording etc. This data may assist the service user in preparing the recovery operations that are required. The diagnostic techniques that could be applied comprise:

- *Spectral Analysis:* The supporting ground station captures, records and displays (real time online) the spectrum of the received signal, thus confirming frequency, noise and modulation scheme.
- *Level Analysis:* Plotting the received signal level may help determine the attitude and signal-to-noise levels. Displaying level fluctuations could also give indications of the attitude and the spin rate of the spacecraft.
- *Lock Indications:* The supporting ground station can confirm, carrier lock, subcarrier lock, symbol lock, TLM Decoder lock and so on to determine up to what point the signal is processed correctly. Frame error counters in the decoders could be provided against the number of good frames received.
- *Open Loop Recording:* The supporting station records the digitised signal using an open loop recording system potentially allowing reconstruction of the TLM Stream.

4.2.3 Uplink Engineering Services

Uplink services apply to cases in which the downlink signal from the spacecraft is acquired, which is proof of life and proof of trajectory. Failure to acquire the downlink does not preclude this service; however, up-linking in the blind significantly increases the difficulty and complexity of recovery operations. The uplink engineering service attempts to increase the probability of the on-board receiver locking on to the uplink carrier utilising the following techniques:

- *Acquisition Sweep Range and Rate Adjustment:* The supporting station uses a sweep range wider than the nominal value and/or uses a sweep rate slower than nominal value to increase the probability that the receiver locks onto the uplink carrier. Another potential mode would be to sweep constantly or ramp the uplink frequency.
- *Acquisition Sequence Adjustment:* The supporting ground station uses an acquisition sequence longer than the nominal value to increase the probability that the on-board symbol synchroniser achieves bit lock onto the acquisition preamble.

4.2.4 Local Radiation Services

In situations of imminent danger to the spacecraft, the service user may request the service provider to radiate to the spacecraft despite having no communication link to the mission operations control centre (MOCC). This service can facilitate the spacecraft condition, i.e. if the spacecraft is coherent, the on-board lock can be confirmed by monitoring the downlink frequency tracking the uplink sweep.

4.2.5 Terminal Uplink Beam-Width Expansion

When the spacecraft trajectory has an extremely high uncertainty, radiation of an uplink signal from a standard aperture configuration may not be capable of acquiring the spacecraft during an emergency. This type of contingency event typically occurs when the LEOP of the spacecraft injection is flawed.

To expand the beam-width (coverage) of an uplink signal a smaller antenna, e.g. horn antenna, is fixed to the tracking antenna and connected to the station transmitter. This approach is primarily used on low earth orbiting (LEO) missions. The expanded Beam-width corresponds to a drastic limitation in uplink power; therefore. the link budget determines whether this technique can be effective.

4.3 Standard Services

The Standard Services available for SECS include the core services specified in the IOAG Service Catalog #1 [1], provided that the pre-conditions stated below are satisfied.

4.3.1 Return Data Delivery

Any core Return Data Delivery services specified in the IOAG Service Catalog#1 can be used as a standard SECS service provided that:

- The supporting ground station receives, demodulates and decodes telemetry on downlink correctly
- The ground link between station and MOCC is established using space link extension (SLE).

4.3.2 Forward Data Delivery

Any core Forward Data Delivery Service specified in the IOAG Service Catalof#1 can be used as a standard SECS service provided that:

- The supporting station already receives, demodulates and decodes TLM on the downlink correctly.
- The spacecraft receives the uplink signal correctly and executes the received Telecommands correctly.
- The ground link between the MOCC and the supporting station is established using SLE.

As mentioned earlier Forward services can potentially be provided without downlink acquisition, i.e. commanding in the blind.

4.3.3 Radiometric Services

Any Radiometric Service specified in the IAOG Services Catalog #1 (except for the Delta Differential One-Way Ranging (DDOR) service can be provided. This assumes that both the downlink and/or the uplink have been acquired correctly.

- 1-way doppler measurements, i.e. the downlink only has been acquired.
- 1-way doppler measurements, i.e. both the Downlink and Uplink has been acquired however the spacecraft on-board transponder is in non-coherent mode.
- 2-way doppler, i.e. both the downlink and uplink has been acquired and the on-board transponder is in coherent mode.
- Ranging, i.e. both the downlink and uplink has been acquired and that the on-board ranging transponder is compatible with the available ranging techniques available at the supporting station.

4.4 Network Services

Missions that have established a committed support agreement with a provider will have accepted the cost of having a permanent network infrastructure in-situ and can expect to receive telemetry as soon as the spacecraft is acquired. This also applies to the transmission of telecommands once the uplink has been acquired.

Missions that have an acknowledged support agreement may not be willing pay for permanent communications infrastructure. However they should have at least pre-agreed plans designed for the creation of an infrastructure with relatively low lead time such as Internet virtual private network (VPN) or perhaps the implementation of a “Bent Pipe” configuration should both User and Provider already have links established with a third party.

Unregistered missions that have no existing agreements will obviously have to accept that the lead time for the creation of links will be longer and depend on the network expertise of both parties a and the hardware that is available at both sites. If the creation of a VPN is not viable, it could be possible to purchase the communications services of a commercial “cloud” service to establish links on a temporarily. This may, however, raise security issues.

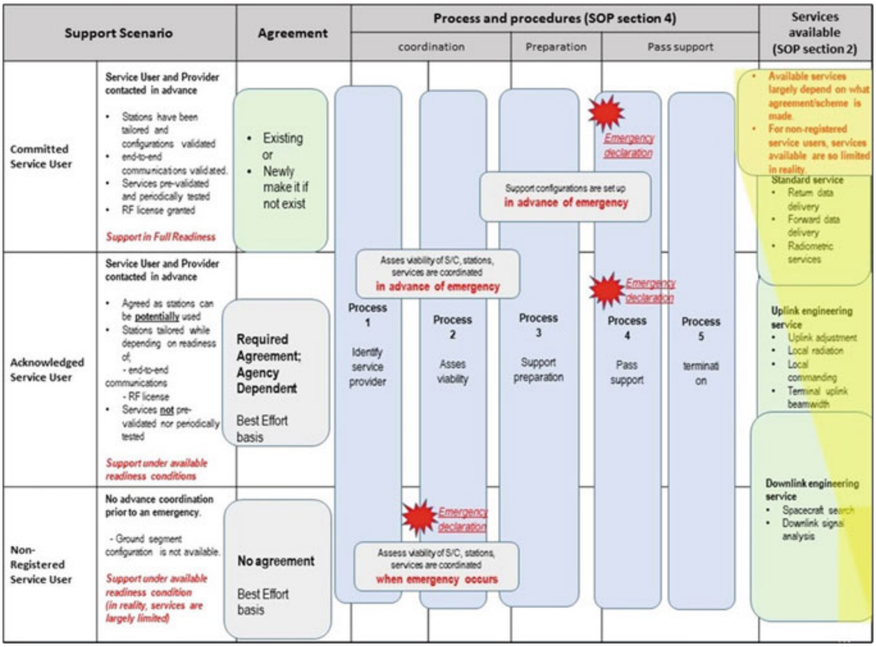
Baseline for Data transfer between a station and a Control Centre is the use of Space Link Extension protocols as defined by the CCSDS Standards(Space Link Extension (SLE) Multiple Blue Books encompassing possible Services [3].

5 Standard Operating Procedures

The targeted readership profile of the SOP is, fundamentally, for a mission manager and their team approaching launch and are designing the ground segment to cover the operational life of a the spacecraft taking into account not only routine activities but also potential emergency situations. Additionally it is useful for flight control teams that have identified a potential weakness in the current ground segment and may decide that it should be augmented with additional resources to deal with an emergency.

It also gives guidance to missions that are already in emergency and require emergency cross support to recover a spacecraft that is in imminent danger of being lost. The processes involved can be summarised as follows

1. Identify service provider
2. Asses viability of service provider for SECS
3. Support preparation
4. Pass support
5. Support termination.


Support Scenario
Agreement
Process and procedures (SOP section 4)
coordination
Preparation
Pass support
Services available (SOP section 2)
Committed Service User
Service User and Provider contacted in advance
• Stations have been tailored and configurations validated
• end-to-end communications validated.
• Services pre-validated and periodically tested
• RF license granted
Support in Full Readiness
• Existing or
• Newly make it if not exist
Acknowledged Service User
Service User and Provider contacted in advance
• Agreed as stations can be potentially used
• Stations tailored while depending on readiness of;
- end-to-end communications
- RF license
• Services not pre-validated nor periodically tested
Support under available readiness conditions
Required Agreement; Agency Dependent
Best Effort basis
Non-Registered Service User
No advance coordination prior to an emergency.
- Ground segment configuration is not available.
Support under available readiness condition (in reality, services are largely limited)
No agreement
Best Effort basis
Process 1
Identify service provider
Process 2
Asses viability
Process 3
Support preparation
Process 4
Pass support
Process 5
terminati on
Asses viability of S/C, stations, services are coordinated
in advance of emergency
Support configurations are set up
in advance of emergency
Assess viability of S/C, stations, services are coordinated
when emergency occurs
Emergency declaration
Emergency declaration
Emergency declaration
• Available services largely depend on what agreement/scheme is made.
• For non-registered service users, services available are so limited in reality.
Standard service
• Return data delivery
• Forward data delivery
• Radiometric services
Uplink engineering service
• Uplink adjustment
• Local radiation
• Local commanding
• Terminal uplink beamwidth
Downlink engineering service
• Spacecraft search
• Downlink signal analysis

5.1 SECS Asset List

Whatever the scenario, the SOP provides a list of assets that allows the mission to select resources that best fit their needs. The IOAG member agencies have identified a subset of candidate communication assets, i.e. ground stations which may be made available to provide SECS services.

Selection of a SECS asset (s) by a given mission is dictated by numerous factors such as compatibility, visibility and performance, e.g.

1. Site location (ensures required geometric coverage).
2. Typical station usage, e.g. LEO, GEO, Lagrange, Deep Space
3. Available spectral bands such as near earth S-/X-Band (range LEO to GEO), near earth S-/X-Band (range GEO to 2 × 10^6 km). or deep space S-/X-Band (range >2 × 10^6 km)
4. Station specification such as figure of merit (G/T), equivalent isotropically radiated power (EIRP) and so on.

The asset information is provided in the SOP, however it is also planned to have the service user performance specifications of each ground station using a web accessible asset characteristic database.

The current asset Table is shown in Appendix 1.

5.2 *Points of Contact*

Each participating agency has nominated a list of contacts to coordinate and execute emergency cross support

5.2.1 Initial Point of Contact (IPoC)

The IPoC is the first person the service user contacts to coordinate SECS, regardless of which support scenario that is required. In general the IPoC is at the managerial level and corresponds to the function responsible for providing SECS. The IPoC coordinates any legal and administrative steps to prepare for SECS and oversee the technical preparations.

5.2.2 Operational Point of Contact (IPoC)

The IPOC provides the service user with the OPoC information. The OPoC is the real time interface during an emergency support. All support scenarios require that the OPoC is contacted to plan and execute all recovery operations.

A spacecraft emergency declaration is normally issued by the mission operations manager. This declaration cannot be triggered autonomously by mission operations staff “on console”. Each service provider is responsible for validating requests for support per their respective internal agency procedures. Initial contact with the OPoC triggers the service provider’s internal processes, e.g.

1. Authentication of emergency declaration
2. Commitment of resources required

3. Preparation of ground segment
4. Mitigation of operational impact on routine mission operations.

5.3 Information Exchange

The spacecraft specifications are critical for the preparations of the ground segment selected by the user and a template has been prepared to give guidance to the user regarding what is required. Typically this information would be found in the mission space to ground interface control document (SGICD). The template is shown in Appendix 2.

5.3.1 Preparatory Activities

Clearly for both committed and acknowledged scenarios the service providers and service users execute the activities prior to any emergency. The non-registered scenario requires that they be executed at the time of the service request on a best effort basis. Support preparation is coordinated between the user, the OPoC and the ground station personnel. Table 1 lists the preparatory activities.

5.3.2 Service Provider Information Exchange During Emergency Support

The following assumes that the interaction at the management level, via the IPoC, is complete, i.e. the service provider agrees to make the requested assets available for emergency support. The actions are also dependant on the categories of the required SECS services.

- Service provider confirms the receipt of the latest orbital predicts and report on the computed tracking times and ephemeris for each contingency pass which can be provided.
- Service provider asses and mitigates any scheduling conflicts concerning the use of the requested asset
- Service provider provides confirmation of acquisition of signal (AOS), i.e. proof of life
- Service provider provides an orbit diagnostic in the ODM format, if applicable.
- Service provider transfers TLM frames, if applicable.
- Service user confirms transmission of commands, if applicable.
- Service provider provides radiometric data, if applicable.

Table 1 Spacecraft emergency support preparation

Step	Action description	Committed	Acknowledged	Non-registered
1	Service user: provide the spacecraft specifications which contains the service user configuration data to the service provider	Required	Required	✓
2	Service provider: configure ground stations for the pre-selected specific services	Required	Required	✓
3	Service provider: obtain RF license	Required	May be required	–
4	Both: determine ground communication line routing path between service user MOCC and service provider, including security aspects	Required	May be required	–
5	Both: test and validate ground communication line routing path	Required	✓	–
6	Conduct periodic end to end validation and testing (6–12 months)	Required	–	–
7	Both: agree upon parameters and techniques for engineering services	✓	✓	✓
8	Both: establish and maintain functioning FD infrastructure	Required	May be required	–
9	Service user: provide spacecraft trajectory file (s)	✓	✓	✓
10	Both: establish operations concept for offline data transfer	Required	May be required	✓
11	Both: exchange contact information	Required	Required	✓
12	Service provider: provide instruction regarding next steps to the OPoC	Required	Required	✓

Required = Completed prior to Emergency, ✓ = Completed at the time of Emergency

5.3.3 Service User Information Exchange During Emergency Support

The service user provides a pre-pass voice briefing to ensure that all parties are aware of the objectives and any limitations for the upcoming pass. They also confirm that the current ODM or two line element (TLE) is the latest version.

After starting the track and acquiring the spacecraft, the flight control team (FCT) provides updates to the provider in real time throughout the service. The FCT confirms the receipt of good TLM Frames, then summarises the health of the spacecraft and condition of the operational transponder. The FCT should provide advance notice of any recovery operations that could cause a loss of signal (LOS), a change in frequency or a change in TLM or TC Rates.

5.3.4 Termination of Support

When the services user declares the end of the spacecraft emergency and recovery of the mission, the service user will provide a debriefing message describing the contingency and the effectiveness of the recovery operations. The service provider will produce a report on the assets and services that participated in the recovery and specific outputs of the scheduling and statistics systems.

6 Proof of Concept

6.1 Committed Scenario

On February 10th, 2020, ESA launched its solar orbiter mission (SOLO) to study the Sun. Routine Science operations will be conducted solely from the ESA 35 m Deep Space Network comprising of Cebreros (Spain), New Norcia (Australia) and Malargue (Argentina). In the event of a critical Spacecraft Contingency and by extension a spacecraft emergency, the ESA/NASA Cross Support agreement can trigger SECS from the Jet Propulsion Laboratory (JPL) deep space network (DSN), in particular, support from the 70 m antennae and their high power amplifiers. As previously stated the following have been implemented and will be regularly validated:

1. Points of Contact
2. Flight Dynamics Infrastructure
3. End to End Data Communications Infrastructure
4. Station Configuration
5. Periodic Validation Test Plan
6. RF Licensing and ITU Filing
7. Voice Communication
8. SLE Configuration (Return All Frames (RAF) & Command Link Transmission Unit (CLTU)
9. Scheduling Interfaces.

6.2 Acknowledged Scenario

Between JAXA and CNES, the agreement for spacecraft tracking cross support has been concluded and this agreement can trigger the SECS from both agencies. For the purpose of exercise and demonstration of emergency support capabilities, the Downlink Engineering Search and Signal Analysis Services stated in Sects. 4.2.1 and 4.2.2 were performed under the simulated acknowledged scenario. In this exercise, CNES declared a loss of on-orbit CNES satellite and requested JAXA to search for the satellite by providing the nominal orbital information which was intentionally time-offset for this exercise. JAXA configured its stations at short notice and initiated search tracking with the antenna scanning and the predefined search pattern for the first tracking pass and then applied the time offsets to the antenna predicts for the second tracking pass. Figures 1 and 2 of the Spectrum Analyser display and the

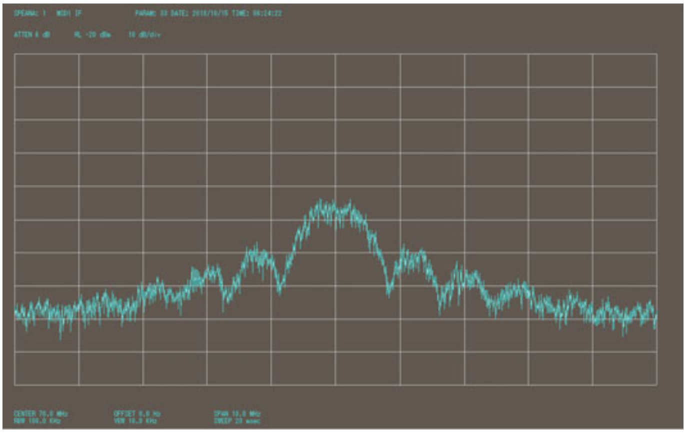
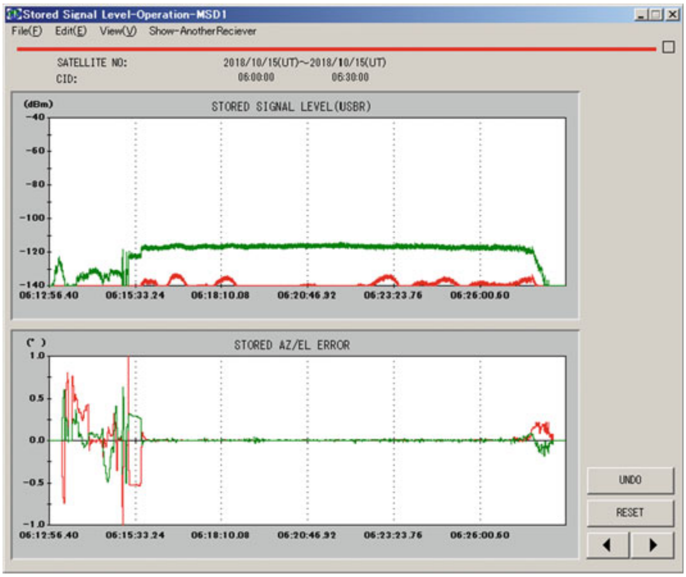


Fig. 1 Satellite spectrum



SATELLITE NO: 2018/10/15(UT)~2018/10/15(UT)
CID: 06:00:00 06:30:00

Fig. 2 AGC Level (RHC/LHC) and antenna pointing Az/El error profile

AGC level were provided to CNES to verify the correct downlink signal of the CNES satellite. With this exercise, both agencies demonstrated the usefulness of the SECS SOP.

6.3 Non Registered Scenario

Every ESA ground station is furnished with a subset of Spacecraft IDs for test and simulation purposes, namely NETSAT. For the purposes of this exercise Cebreros and NETSAT were selected. The target for acquisition was the HAYABUSA II (ISAS/JAXA) spacecraft. The HAYABUSA II SGICD was used for the spacecraft specifications. The orbital predicts which were already available for the Malargue Station were translated to NETSAT.

The NETSAT configuration tables for the Downlink, Uplink and Radiometric subsystems, e.g. Frequency Plans, Doppler Predictions, Demodulators, Decoders and TLM Recording etc., were created. This was time consuming and the lead time would typically be a minimum of 1 man day. The exercise was considered to be closed loop only, i.e. an open loop configuration was not created.

The station mimic was made available on the Web streamer providing a real time display of the station operations, if it were required by the MOCC.

The goal was to acquire the spacecraft and record station performance parameters. A communications infrastructure between Cebreros and the HYB2 MOCC was not available, therefore the TLM frames were recorded on the Cebreros SLE Servers which could be retrieved manually by FTP and transfer to the User MOCC offline.

The spacecraft was successfully acquired at 512 sps. The downlink signal strength was marginal and there were many BAD frames, however some were flagged as GOOD and could have been processed by the MOCC if necessary. The recorded frames were retrieved to the European space operations centre (ESOC) but were not forwarded to the MOCC.

The following diagram provides a snapshot of the Cebreros station acquiring HYB2 downlink. The charts in the bottom left corner display the Carrier Levels on Receiver 1 and 2 (Figs. 3 and 4).

Although the uplink chain was configured, the X-Band transmitter was not switched "on" since there was no emergency and Cebreros did not possess an RF license.

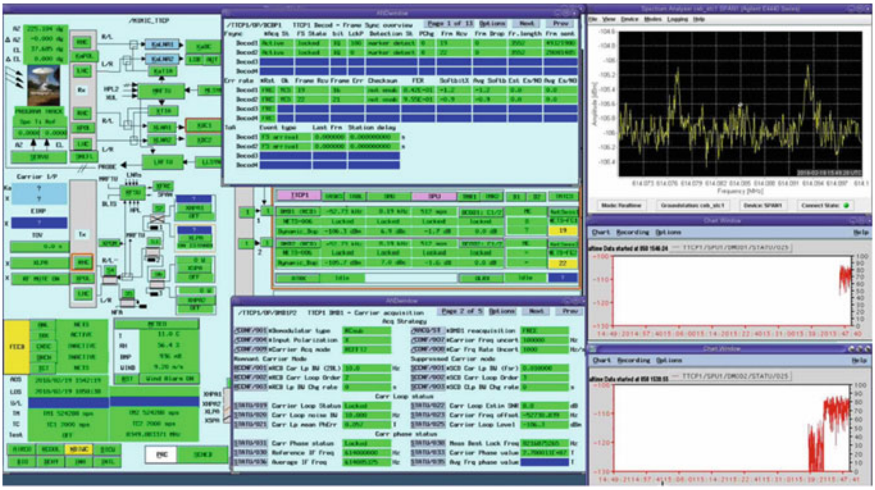
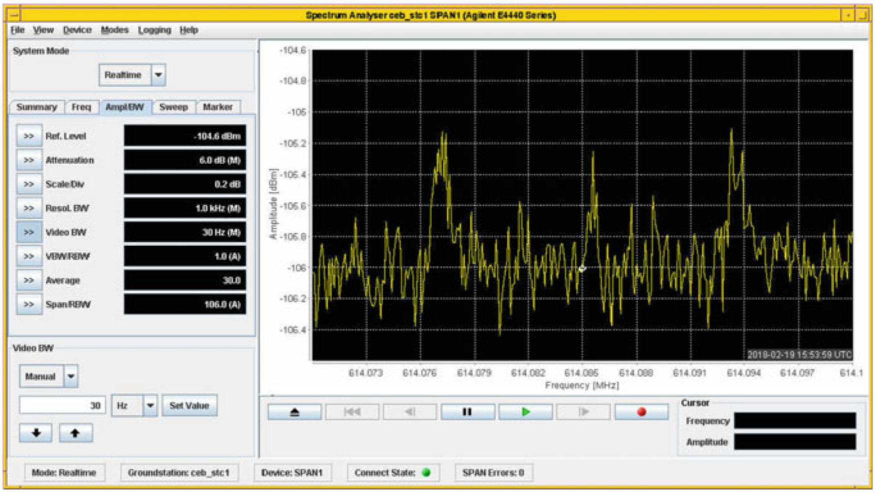


Fig. 3 Snapshot of Cebreros contingency acquisition of test target Hayabusa-2 downlink



Spectrum Analyser ceb_stc1 SPAN1 (Agilent E4440 Series)

System Mode: Realtime

Ref. Level	-104.6 dBm
Attenuation	6.0 dB (M)
Scale/Div	0.2 dB
Resol. BW	1.0 kHz (M)
Video BW	30 Hz (M)
VBW/RBW	1.0 (A)
Average	30.0
Span/RBW	106.0 (A)

Video BW: Manual

30 Hz Set Value

Mode: Realtime | Groundstation: ceb_stc1 | Device: SPAN1 | Connect State: ● | SPAN Errors: 0

Fig. 4 Hayabusa-2 downlink spectrum (proof of concept acquisition)

7 Addition and Validation of New Terminals to the Asset List

With the completion of the refurbishment of the "Goonhilly-6 (GHY6)" 32 m in Cornwall, southwest England, the station has undergone validation testing for support of a deep space mission (ESA Mars Express) and a high earth orbit (ESA Integral). A full spectrum of support activities were exercised, i.e. telemetry, telecommand and tracking (TT&C) services. The testing was performed in both Xray Band (X-Band) and Sierra band (S-Band) The Communications infrastructure was validated using the SLE. It is planned to validate the station for Lagrange point (ESA GAIA) and Lunar (ISRO GAGANYAAN) orbits second half of 2021.

When the second issue of the SOP is released, GHY6 will be added to the asset list under the auspices of the United Kingdom Space Agency (UKSA) which is a member of the IOAG.

8 Discussion

The SOP cannot predict the nature of every spacecraft emergency, it contains however a list of potential situations, involving both signal processing problems, trajectory problems, or worst case both in parallel.

Recovery of a spacecraft will very often require transmission of an Uplink signal from the provider. Committed or Acknowledged support assumes that RF Licensing and ITU filing has been discussed and finalised if considered necessary.

RF Licensing is an extremely sensitive issue and conforming to ITU regulations is considered mandatory. The asset table in Appendix 1 indicates which countries categorically refuse to radiate without a license. The table also lists assets that would radiate should the spacecraft be in imminent danger of loss of mission.

In the event of a non-registered user approaching a service provider for assistance in recovery operations an uplink may well be requested. Paragraph 4.9 of the ITU regulations states the following:

No provision of these regulations prevents the use by a station in distress, or by a station providing assistance to it, of any means of Radio communication at its disposal to attract attention, make known the condition and location of the station in distress and obtain or provide assistance.

This means no ITU regulation prohibits a ground station from providing support to a spacecraft or an astronaut in a life threatening situation!

9 Conclusions

The fruits of the work of the standards groups is evident by the ability to leverage those standards with a spacecraft emergency where rapid call up of support is needed with a high confidence that the interface will be compatible. The team has worked to develop a standard operating procedure that leverages those standards, and defines a set of terms and processes that enable coordination of support as rapidly as possible. The value of some minimum levels of coordination cannot be understated as it is recognized that time is likely of the essence during a spacecraft emergency and pre-coordination for authorization, as is defined under the committed and acknowledged scenario's, will likely yield the best opportunity for saving the mission.

Future work will continue to refine the SOP and help to broaden the mission set to human space flight missions and also engage commercial providers.

Acknowledgements The support and guidance of the IOAG leadership, in particular, Michael Schmidt is acknowledged in encouraging the progress and benefits of this effort.

Appendix 1 (Current SECS Assets Table)

Agency	Size	Location	Antenna ID	Typical Station usage[1]	S-band		X-band		Conditions to take into account for providing support		Remarks
	(M)				Max EIRP (dBm)	G/T (dBK)	Max EIRP (dBm)	G/T (dBK)	Available Support services[2] (specify if any constraint)	Is Uplink RF license required for Emergency support? [3]	
ASI	10	Malindi, KEN		LEO	98.0	21.3	-	-	IOAG SC#1 core services (TLM, TC, RNG)	An S-band full coverage annual license is available	It supports Arianespace launches from Kourou CSG (A5, VEGA) and SpaceX launches. SLE compliant
	13	Malindi, KEN		LEO	99.0	21.3	-	-	IOAG SC#1 core services (TLM, TC, RNG)	An S-band full coverage annual license is available	It supports Arianespace launches from Kourou CSG (A5, VEGA) and SpaceX launches. SLE compliant
CNES	11	Kourou, GUF		LEO, G	101.0	22.5	-	35.5	All	-	
	10	Kerguelen Islands, FRA		LEO, G	101.0	21.5	-		All	-	
	11	Aussaguel, FRA		LEO, G	101.0	22.5	-	35.5	All	-	
	6.4	Aussaguel, FRA		LEO	85.0	17.0	-		All	-	
	11	Hartebeesthoek, ZAF		LEO, G	101.0	22.5	-	35.5	All	-	
	13	Kiruna, SWE		LEO, G	98.0	22.5	-	34.0	All	-	
	13	Inuvik, NT, CAN		LEO, G	98.0	22.5	-	34.0	All	Required	
DLR	15	Weilheim, DEU		LEO, G, L, H	108.0	26.7	-	-	All	-	
	15	Weilheim, DEU		LEO, G, L, H	109.0	27.8	-	-	All	-	
	30	Weilheim, DEU		H, D	-	-	-	44.0	All	-	also L-Band DL
	7.3	Neustrelitz, DEU		LEO	-	17.0	-	31.0	All	-	
	7.3	Neustrelitz, DEU		LEO	90.0	17.0	-	31.0	All	-	
	7.3	Neustrelitz, DEU		LEO	90.0	17.0	-	31.0	All	-	

(continued)

(continued)

Agency	Size	Location	Antenna ID	Typical Station usage[1]	S-band		X-band		Conditions to take into account for providing support		Remarks
	(M)				Max EIRP (dBm)	G/T (dBK)	Max EIRP (dBm)	G/T (dBK)	Available Support services[2] (specify if any constraint)	Is Uplink RF license required for Emergency support? [3]	
	11.5	Neustrelitz, DEU		LEO	93.0	22.0	-	36.0	All	-	also Ka DL
	9	O'Higgins, ANT		LEO	92.0	19.5	-	32.0	All	-	
	13	Inuvik, NT, CAN		LEO	100.0	22.4	-	35.9	All	Required	
ESA	4.5	New Norcia, AUS	NNO2	LEO	-	-	101.5	28.0	All	-	Includes 0.75 m Acquisition Aid
	15	Kiruna, SWE	KIR1	LEO, G, L, H	102.9	29.5	-	39.2	All	-	
	13	Kiruna, SWE	KIR2	LEO, G, L, H	98.3	22.8	-	36.5	All	-	
	15	Kourou, GUF	KRU1	LEO, G, L, H	105.0	29.4	114.0	42.0	All	-	
	35	Cebreros, ESP	CEB1	G, L, H, D	-	-	139.0	52.4	All	Required	also K & Ka DL
	35	New Norcia, AUS	NNO1	G, L, H, D	128.0	39.2	138.7	51.2	All	Required for S-band[4]	
	35	Malargue, ARG	MLG1	G, L, H, D	-	-	139.5	52.1	All	-	also K & Ka DL and Ka UL
KARI	13	Daejeon, KOR		LEO	88.0	23.0	-	36.0	Engineering service only	Required	SLE is not operational yet
	9	Daejeon, KOR		LEO, G	85.0	19.0	-		Engineering service only	Required	SLE is not operational yet
	7.3	Daejeon, KOR		LEO	83.0	19.5	-	34.0	Engineering service only	Required	SLE is not operational yet
	11	Jeju, KOR		LEO	83.0	20.0			Engineering service only	Required	SLE is not operational yet
NASA	4.7	Wallops Island, VA, USA	LEO-T	LEO	89.2	17.0	-		All	-	
	11.3	Wallops Island, VA, USA	WG1	LEO,G,L,H	94.6	23.6	-	34.5	All	-	
	11.3	Fairbanks, AK, USA	AS1	LEO,G,L,H	94.6	22.0	-	35.2	All	-	

(continued)

(continued)

Agency	Size	Location	Antenna ID	Typical Station usage[1]	S-band		X-band		Conditions to take into account for providing support		Remarks
	(M)				Max EIRP (dBm)	G/T (dBK)	Max EIRP (dBm)	G/T (dBK)	Available Support services[2] (specify if any constraint)	Is Uplink RF license required for Emergency support? [3]	
	9.1	Fairbanks, AK, USA	AS2	LEO,G,L,H	89.0	21.2	-	36.2	All	-	
	11	Fairbanks, AK, USA	AS3	LEO,G,L,H	95.7	22.9	-	35.2	All	-	
	18.3	White Sands, NM, USA	WS1	LEO,G,L,H	102.0	29.6	-	-	All	-	
	11.3	Kennedy Space Center, FL, USA	KUS	LEO,G,L,H	87.0	17.2	-	-	All	-	
	11.3	Ponce De Leon, FL, USA	PDL	LEO,G,L,H	87.0	17.2	-	-	All	-	
	70	Canberra, AUS	43	D,L,H	135.6 (2110 -2118 MHz) 127.4 (2090–2091 MHz)”	49.8	145.8	61.5	All	-	In all cases for all stations the JPL Frequency Spectrum Mgr will follow up post support for license if needed. (2090–2091 MHz) Emergency only
	70	Goldstone, CA, USA	14	D,L,H	135.6 (2110 -2118 MHz) 127.4 (2090–2091 MHz)”	49.8	145.8	61.5	All	-	(2090–2091 MHz) Emergency only
	70	Madrid, ESP	63	D,L,H	135.6 (2110 -2118 MHz) 127.4 (2090–2091 MHz)”	49.8	145.8	61.5	All	-	(2090–2091 MHz) Emergency only (2110 -2118 MHz) Emergency only
	34	Canberra, AUS	34	D,L,H,G	128.7	40.8	139.5	54.2	All	-	LEOP support for Geosynchs S or X Band
	34	Canberra, AUS	36	D,L,H,G	108.8	40.8	139.5	54.2	All	-	LEOP support for Geosynchs S or X Band
	34	Canberra, AUS	35	D,L,H,G			139.5	54.2	All	-	LEOP support for Geosynchs X Band only

(continued)

(continued)

Agency	Size	Location	Antenna ID	Typical Station usage[1]	S-band		X-band		Conditions to take into account for providing support		Remarks
	(M)				Max EIRP (dBm)	G/T (dBK)	Max EIRP (dBm)	G/T (dBK)	Available Support services[2] (specify if any constraint)	Is Uplink RF license required for Emergency support? [3]	
	34	Goldstone, CA, USA	24	D,L,H,G	128.7	40.8	139.5	54.2	All	-	LEOP support for Geosynchs S or X Band
	34	Goldstone, CA, USA	25	D,L,H,G			139.5	54.2	All	-	LEOP support for Geosynchs X Band only
	34	Goldstone, CA, USA	26	D,L,H,G	108.8	40.8	139.5	54.2	All	-	LEOP support for Geosynchs S or X Band
	34	Madrid, ESP	54	D,L,H,G	128.7	40.8	139.5	54.2	All	-	LEOP support for Geosynchs S or X Band
	34	Madrid, ESP	55	D,L,H,G			139.5	54.2	All	-	LEOP support for Geosynchs X Band only
	34	Madrid, ESP	65	D,L,H,G	108.8	39.4	139.5	53.2	All	-	LEOP support for Geosynchs S or X Band
JAXA	10	Katsuura, JPN		LEO, G	101.0	22.5	-	-	All (see note1)	Required	**Note1:** For Radiometric Services, the Pseudo-Noise [PN] Ranging Systems is available only from DS stations but not conform to the CCSDS standard. JAXA LEO stations have presently no plan to implement the PN ranging system. Therefore, only satellite transparent mode is available in the regenerative mode and JAXA's internal format is used for this service **Note2:** "Engineering service only" stations are not connected to SLE
	20	Katsuura4, JPN		LEO, G, L	97.7	27.7	-	39.0	All (see note2)	Required	
	10	Masuda, JPN		LEO, G	101.0	22.5	-	-	All (see note1)	Required	
	10	Okinawa, JPN		LEO, G	101.0	22.5	-	-	All (see note1)	Required	
	18	Okinawa2, JPN		LEO, G	104.8	25.5	-	-	All (see note1)	Required	
	10	Mingenue, AUS		LEO, G	101.0	22.5	-	-	All (see note1)	Required	
	10	Santiago, CHL		LEO, G	101.0	22.5	-	-	All (see note1)	-	

(continued)

(continued)

Agency	Size	Location	Antenna ID	Typical Station usage[1]	S-band		X-band		Conditions to take into account for providing support		Remarks
	(M)				Max EIRP (dBm)	G/T (dBK)	Max EIRP (dBm)	G/T (dBK)	Available Support services[2] (specify if any constraint)	Is Uplink RF license required for Emergency support? [3]	
	10	Kiruna, SWE		LEO, G	101.0	22.5	-	-	All (see note1)	*(being confirmed)*	
	10	Maspalomas, ES		LEO, G	101.0	22.5	-	-	All (see note1)	-	
	20	Uchinoura, JPN		LEO, L, H	110.0	32.8	-	43.0	Engineering service only	N/A	
	34	Uchinoura, JPN		LEO, L, H, D	115.0	38.4	138.7	50.0	All (see note1)	Required	
	11	Uchinoura, JPN		LEO	103.6	23.8	-	-	Engineering service only	N/A	
	64	Usuda, JPN		L, H, D	-	40.0	143.0	49.5	All (see note1)	Required	

[1] Typical Station usage

This section shows the type of satellites most-typically operated by the ground station. Service User may refer to this section to ensure the suitability of assets in terms of frequency band and/or antenna drive speed when identifying which antenna(s) can be used.

Example:

LEO = Low Earth Orbit; G = Geostationary; L = Lunar or its vicinity including Lagrange Point;
H = High Earth Orbit including Highly Elliptical Orbit; D = Deep Space

[2] Available support services (specify constraints, if any)

Availability of the SOP services. Specify constraints, if applicable.

Example:

All = All Services may potentially be available; however, availability of specific services need to be coordinated with Service Provider.;
Engineering services only (because no real-time interface, such as SLE function, is available at the ground station)
Specify constraints, if any (if any of Core SC#1 functions is not available for standard TT&C services)

[3] Is RF uplink license required for emergency support?

Specify if RF license must be obtained prior to providing uplink services.

Example:

Required: = formal RF license process is required for this station

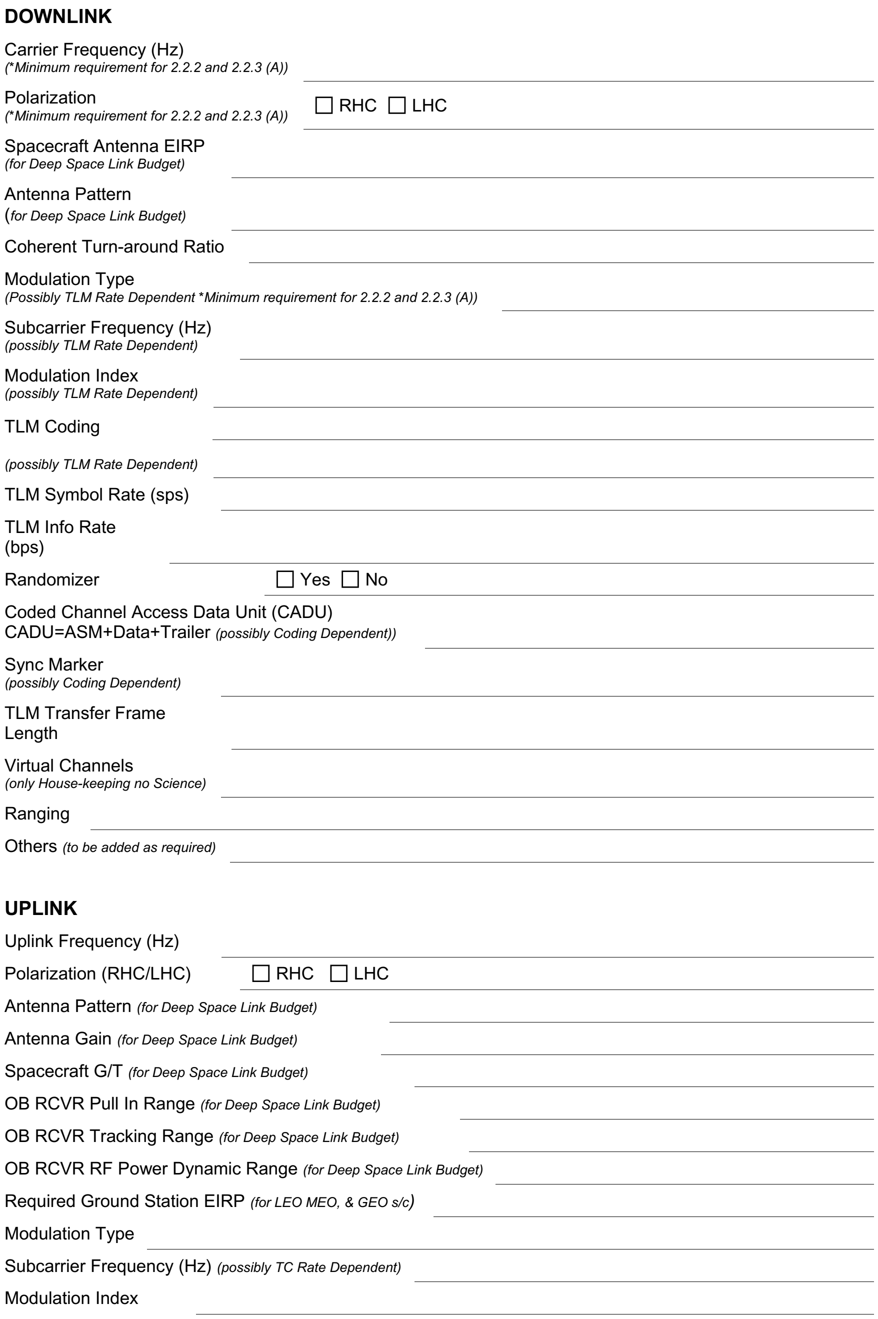
"-" = Ground station does not require formal RF license for emergency support; however, real-time coordination with relevant administrative bodies may be required. Ground stations do require RF uplink license for uplink services during non-emergency situations, such as periodic testing.

Antenna ID: CCSDS station ID

[4] Full allocation not available

N/A: Not applicable

Appendix 2 (Spacecraft Specification Template)



DOWNLINK

Carrier Frequency (Hz)
*(*Minimum requirement for 2.2.2 and 2.2.3 (A))*

Polarization
*(*Minimum requirement for 2.2.2 and 2.2.3 (A))* □ RHC □ LHC

Spacecraft Antenna EIRP
(for Deep Space Link Budget)

Antenna Pattern
(*for Deep Space Link Budget)*

Coherent Turn-around Ratio

Modulation Type
*(Possibly TLM Rate Dependent *Minimum requirement for 2.2.2 and 2.2.3 (A))*

Subcarrier Frequency (Hz)
(possibly TLM Rate Dependent)

Modulation Index
(possibly TLM Rate Dependent)

TLM Coding

(possibly TLM Rate Dependent)

TLM Symbol Rate (sps)

TLM Info Rate
(bps)

Randomizer □ Yes □ No

Coded Channel Access Data Unit (CADU)
CADU=ASM+Data+Trailer *(possibly Coding Dependent))*

Sync Marker
(possibly Coding Dependent)

TLM Transfer Frame
Length

Virtual Channels
(only House-keeping no Science)

Ranging

Others *(to be added as required)*

UPLINK

Uplink Frequency (Hz)

Polarization (RHC/LHC) □ RHC □ LHC

Antenna Pattern *(for Deep Space Link Budget)*

Antenna Gain *(for Deep Space Link Budget)*

Spacecraft G/T *(for Deep Space Link Budget)*

OB RCVR Pull In Range *(for Deep Space Link Budget)*

OB RCVR Tracking Range *(for Deep Space Link Budget)*

OB RCVR RF Power Dynamic Range *(for Deep Space Link Budget)*

Required Ground Station EIRP *(for LEO MEO, & GEO s/c)*

Modulation Type

Subcarrier Frequency (Hz) *(possibly TC Rate Dependent)*

Modulation Index

TC Coding

TC Rate

CLTU min length (Octets)

CLTU max length (Octets)

TC Protocol (PLOP1/PLOP2) □ PLOP1 □ PLOP2

TC Format Standard

TC Pseudo Randomizer

Idle Pattern Length

Uplink Sweep Profile □ Sweep range and speed □ For Deep Space □ Wide Band
□ Intermediate Band □ Narrow Band

Others *(to be added as required)*

RANGING TYPE

Ranging Major Tone Frequency *(or OB BW)*

Modulation Type

TX Tone Modulation Index (Uplink)

RX Tone Modulation Index (Downlink)

Standard RNG Code Lengths *(or OB BW)*

Ranging Channel Equivalent Noise Bandwidth

On board transit time

GROUND IMPLEMENTATION

Communications

SLE Services

Service Instances

Voice

References

1. IOAG Service Catalogue #1, Issue 2, Revision 2 16/09/2020
2. CCSDS Orbit Data Messages 502.0-B-2 Nov 2009
3. CCSDS Standards: Space Link Extension (SLE) Multiple Blue Books encompassing possible Services

Regulatory Constraints and Operational Best Practices to Ensure the Safety of Non-routine Space Operations

Lorenzo Arona, Martyn Fogg, Mark Loveday, Brit Wengenmayr, and Ini Ituen

Abstract Avanti is a British operator of geostationary communications spacecraft. From its inception in 2002, the company evolved into a significant regional operator (Fogg et al. [1]), which continues to be at the forefront of technological advancement in the industry. Nimbleness to relocate its satellites in order to respond to shifting business needs is part of Avanti's success story. Satellite relocations are non-routine operations that—by their own nature—raise more risks than the routine activities. Avanti has developed a framework to assess the operational and regulatory risks associated with these operations and mitigate them through careful planning and meticulous execution. In our experience, regulatory requirements often translate into operational constraints, and operational risks often require regulatory support for an effective mitigation. For this reason, safe non-routine operations require transparency, clear communications, and a cooperative mindset among all teams. Similarly, communications to third parties is crucial to ensure the safety of non-routine operations. Disseminating operational products to fellow operators, watchdogs, and service providers allows Avanti to mitigate both the risk of close approach and the risk of radio-frequency interference. Avanti happily shares its experience in order to foster collaboration among players, promote industry best practices, and contribute to the safety of the space environment.

L. Arona (✉) · M. Fogg · M. Loveday · I. Ituen
Space Systems Planning and Operations, Avanti Communications, Cobham House, 20 Black Friars Lane, London EC4V 6EB, UK
e-mail: lorenzo.arona@avantiplc.com

B. Wengenmayr
Regulatory Department, Avanti Communications, Cobham House, 20 Black Friars Lane, London EC4V 6EB, UK

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_2

1 Non-Routine Space Operations

Non-routine satellite operations, like launch and early orbit phase (LEOP) or decommissioning, result from a need to change the satellite mission and are, by their very nature, transitional. Avanti gathered extensive non-routine operations expertise by frequently relocating its satellites in response to business needs, as depicted in Fig. 1.

This paper will focus on the particular example of relocation of geostationary satellites, but the core principles herein presented will be applicable to many other scenarios. In what follows, we will refer to the satellite to be relocated as *ownship*.

The operational planning, scheduling and execution for non-routine activities is subject to national and international regulatory oversight related to collision avoidance, space debris, safe operation in outer space, interference management and national licensing. In Avanti's experience, the two aspects are strictly interconnected and feed into each other. For example, a regulatory deadline can pose constraints on the date of the spacecraft's arrival to a target orbital position. Operations at the target orbital position need ground stations able to support the spacecraft telemetry, tracking and commanding (TT&C) frequencies. These ground stations often need to be tested and calibrated prior to the start of the spacecraft relocation and that, in turn, puts a deadline on the regulatory team to acquire the needed national licenses in time.

The Avanti approach to performing non-routine operations is to start with a risk assessment, then burn down the risks throughout the operations preparation phase and execution phase. The main operational and regulatory risks that have been identified are:

- Risk of close approach with debris or active satellite;
- Risk of thruster over- or under-performances;
- Risk of interruption to the capability to send commands and retrieve telemetry;
- Risk of poor quality of orbit determination;
- Risk of radio-frequency interference with other active satellites;
- Risk of degradation to the spacecraft power input;
- Risk of non-compliance with international treaties and national law.

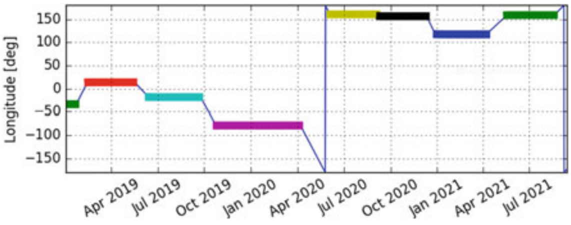


Fig. 1 Longitude evolution of one of Avanti satellites—each relocation is a non-routine operation

2 Regulatory Aspects

Operation in Outer Space is regulated by the United Nations Treaty on Peaceful Use of Outer Space (UNCPUOS). Under this treaty, the countries that supervise the operator are responsible for national space activities, whether carried out by governmental or non-governmental entities. Many countries have put in place rules to deal with liability for damage caused by their space objects and safe operation in space as well as avoidance of space debris.

These international rules for spectrum associated with orbital resources and their rational, efficient and economical use are further regulated under the International Telecommunication Union (ITU) Constitution and Convention [2], and the Radio Regulations (RR) [3]. These contain the fundamental principles and the specific regulations governing the following major elements:

- Frequency spectrum, its allocation and assignment;
- Rights and obligations of ITU member administrations in obtaining access to the spectrum and orbit resources;
- International recognition of these rights.

When a United Kingdom (UK) based company like Avanti intends (a) launching or procuring the launch of a space object, (b) operating a space object, or (c) performing any activity in outer space, it must currently obtain the required licensing from the UK's national space regulator, the Civil Aviation Authority (CAA–the national regulator changed from the UK Space Agency to the CAA in summer 2021). The regulatory agencies, licensing frameworks and information required for obtaining and maintaining licensed vary widely around the globe.

The information required by the CAA is specific to any planned non-routine mission. In the case of a relocation, the national space regulator requires the satellite target orbital position, the names of the associated network filings, and an indication of the administration responsible for these (referred to as Notifying Administrations). It also requires two sets of operationally relevant information, which would govern station-keeping and TT&C communications at the target orbital position, detailed in Sects. 2.1 and 2.2. Other information the national regulator may require are proof of the ownship's third party liability insurance, as well as verification of ownship's current technical status before relocation start in form of a satellite system performance health check.

2.1 Station-Keeping

The ITU Regulations governing station-keeping [4] state that a satellite should maintain their positions within ± 0.1° of longitude of their nominal positions irrespective of the cause of variation, unless this causes an unacceptable level of interference to other satellites.

For this reason, in order to obtain the approval in the operator's home country by the relevant regulatory body to relocate ownship, the analysis of the target orbital position, which will be described in Sect. 3.1, must be carried out.

2.2 TT&C Frequency Assignment and Satellite Network Filings

In practice, the national regulatory authority responsible for frequency management will generally check that TT&C frequency assignments are available to the gateway operator and that the corresponding satellite network filings are in place for the satellite operator. In the case of Avanti, the relevant UK authority is the Office of Communications (Ofcom). During the Outer Space Act approval process to the non-routine mission, the CAA liaises with Ofcom to ensure that the use of spectrum is in accordance with the international regulatory framework and supports the safe operation of the spacecraft at all times.

The TT&C frequency assignments used by the relocating satellite are required solely for the UK CAA approval process and can be provided in different ways, unless the TT&C antenna for the relocation is located in a country where the application for the earth station license also requires a TT&C filing name.

This approval process can be quite time-consuming. Hence, the choice of the TT&C infrastructure used to communicate with ownship throughout its non-routine mission (which will be described in Sect. 3.4) must be made as soon as possible into the planning stage, and extensive communications with multiple operators are required.

This article does not address the national licensing for the ground stations and associated spectrum for TT&C in the countries where they are located. For missions that cross several continents, there could be several such stations, which need to be licensed in distinct countries. Dealing with multiple regulators, and often highly divergent approaches, adds complexity and time that equally needs to be factored in. This type of license may be applied for by a third-party in case the temporary use of the TT&C station is leased. In that case, close coordination with the operational and regulatory teams of the supplier is required.

The regulatory approval process can be time-consuming and often lies on the critical path to plan the non-routine operations. Refer to Sect. 4 for some consideration on the timing of both regulatory and operational activities.

3 Operational Aspects

3.1 Analysis of the Target Orbital Position

Once the target orbital position is known, the first step to mitigate both the risk of close approach between ownship and active satellites and the risk of radio-frequency interference (RFI) is to run an analysis of the region around the target orbital position. The most obvious way is to rely on the public two-line elements (TLE) catalogue available for download at Space Track [5]. Another valuable source is the Satellite Database available for download at the Union of Concerned Scientists website [6]: this has the advantage of including much more data about each satellite, but it is updated only a few times per year on a best effort basis. The more costly option to perform the target slot analysis is to rely on commercial providers of operational services, which can use their own sensors to obtain first-hand information about the satellites operating in the neighbourhood (whether they are listed in the public catalogue or not).

If the target orbital position is already inhabited by one or more satellites, the actual longitude slot targeted for station-keeping can differ from it. When possible, it is good practice to leave a no-fly-zone between the two spacecrafts. For example, if both ownship and the neighbouring satellite are controlled in station-keeping boxes ± 0.1° wide, then it would be better for the centres of the two boxes to lie at least 0.22° away.

Another aspect to consider is the TT&C and payload frequencies of neighbouring satellites. While some public sources have partial information about the frequencies used by satellites' beacons and transponders, for more complete information the operator of the neighbour satellite will have to be approached, for example, through a FLYBY process (see Sect. 3.9). Once accurate information on the neighbouring satellites' frequencies is known, it is useful to calculate the off-axis radiation of the TT&C and gateway antennas serving ownship (see also Sect. 3.4). Once this is known, it is possible to calculate the minimum longitude distance that would not cause RFI to the neighbouring satellites. The minimal longitude distance between ownship and the neighbouring satellites, which guarantees compliance with the ITU Regulations cited in Sect. 2.1, will be the larger between the safety-of-flight distance and the distance that guarantees no Adjacent Satellite Interference (ASI).

3.2 Mission Planning

Once the details of the neighbourhood around the target orbital position are known, and once the deadline to arrive at destination has been set, then the design of the relocation can start. A relocation comprises three phases:

- A **drift-start** phase, in which ownship performs orbit correction manoeuvres to induce a longitude drift rate;
- A **free-drift** phase, in which no orbit correction manoeuvres are performed, and the spacecraft drifts towards its target orbital position;
- A **drift-stop** phase, in which the spacecraft performs orbit correction manoeuvres to virtually nullify the longitude drift rate and enter the target orbital position;

The drift-start and—especially—the drift-stop phases are considered critical operations, as they are executed at geostationary altitude, where other active spacecraft operate. We will consider each phase in further detail.

3.2.1 Drift-Start

The main challenge during drift-start is to avoid close approach with co-located spacecraft. When possible, it is worth considering a temporary biasing of the co-located spacecraft inclination and eccentricity to increase separation during this phase. It is also worth considering running a Monte Carlo simulation to study the effect of manoeuvre under- or over-performance over the inter-satellite distance in the co-located cluster. Operational ephemerides, including the best estimate of the drift-start manoeuvres, should be distributed to neighbour operators (see also Sect. 3.3).

It is technically possible to plan all the manoeuvres in advance, and update the operational ephemerides only after drift-start operations are concluded. Avanti uses a more cautious approach. During drift-start and drift-stop operations, Avanti performs orbit determination (OD) by using both RF sensors and optical sensors.

Under the right light conditions, this guarantees a faster convergence to a low-covariance solution. The result of this OD process is fed into the planning of the next manoeuvre, usually performed about 12 h after the previous one. This approach (which Avanti uses also during drift-stop—see 3.2.3) has the advantage of mitigating the risks of close approach and limiting the re-scheduling of operations, but has the drawback of requiring out-of-hours support from all the operational teams.

3.2.2 Free-Drift

There are two major challenges associated with free-drift phases:

1. The increase in orbital inclination;
2. The evolution of the orbital eccentricity;

The inclination will grow at its natural pace, but it is usually not advisable to perform inclination manoeuvres during the free-drift, as the parasitic components of the ΔV can alter both the orbital eccentricity and the arrival date. In Avanti experience, a different approach is more advantageous. Once the duration of the free-drift has been decided, the total growth of inclination throughout the drift can be estimated. If ownship needs to be inclination-controlled, then the inclination can be biased by the right amount before the start of the relocation. The drawback of this option is that it might not be feasible if ownship is originally operating within a cluster of co-located spacecraft. Note also that for some spacecraft inclination, control must be suspended during eclipse season, and this needs to be considered in the overall relocation planning.

The evolution of the eccentricity can potentially be a problem for long, slow relocations. The IADC Space Debris Mitigation Guidelines [7] recommend avoiding penetration of the zone's geostationary equatorial orbit (GEO) ± 40 km during relocations. GEO is the geostationary altitude at 35,786 km and the zone GEO ± 40 km is where operational satellites are station-keeping (sometimes referred to as geostationary belt). If the transfer orbit during free-drift would be perfectly circular, this would translate into a minimum longitude drift rate of approximately 0.5°/day. Interestingly, this means that there is a safety-of-flight upper limit to the duration of each relocation. A non-null orbital eccentricity, though, causes oscillations in the spacecraft altitude. As the eccentricity evolves uncontrolled during the free-drift phase, there are only two options to avoid penetrating the station-keeping zone. The first is to increase the longitude drift rate. The second is to bias the eccentricity prior to starting the relocation, in the same way as it can be done for the inclination. Once again, this option might not be available if other spacecraft operate in the departure slot in co-location with ownship.

3.2.3 Drift-Stop (Primary and Backup)

The risk of thruster under- or over-performance is particularly severe at drift-stop, as it might cause both close approach with neighbouring satellites and overshoot of the target orbital position.

At Avanti, the main mitigation is the design of both a nominal and a backup drift-stop plan. First, the drift-stop manoeuvres are planned as late as possible, usually within 36 h of ownship arrival at the target orbital position. This plan is the backup plan. In a second step, Avanti designs a different drift-stop plan in which the first manoeuvre is anticipated by 24 h. This causes a slight delay in the time of arrival but hardly any change to the total ΔV. By doing this, Avanti creates a nominal drift-stop plan that is very robust. If the first drift-stop manoeuvre aborts for any reason, the SOC has 24 h to investigate the anomaly, recover it, and schedule the backup drift-stop plan.

An example of primary and backup drift-stop plan is provided in Table 1, Figs. 2, and 3.

Note that any anomaly that results in overshooting the target orbital position would also need TT&C support to be recovered. For this reason, it is good practice to choose TT&C sites in such a way that the target orbital position is not at the edge of the visibility arc. Note also that ASI would need to be coordinated even in case of anomaly, so it is good practice to gather information about the frequencies of the

Table 1 Sample primary and backup drift-stop manoeuvre plans

	Backup drift-stop plan	Primary drift-stop plan
Manoeuvre #1	03/06/2020 12:24	02/06/2020 12:37
Manoeuvre #2	04/06/2020 00:24	04/06/2020 12:37
Manoeuvre #3	04/06/2020 12:24	05/06/2020 00:53

Note that while the first manoeuvre is anticipated approximately 24 h in the primary plan with respect to the backup plan, the other manoeuvres are delayed. This is because the drift rate after the execution of the first manoeuvre is lower in the primary plan than in the backup

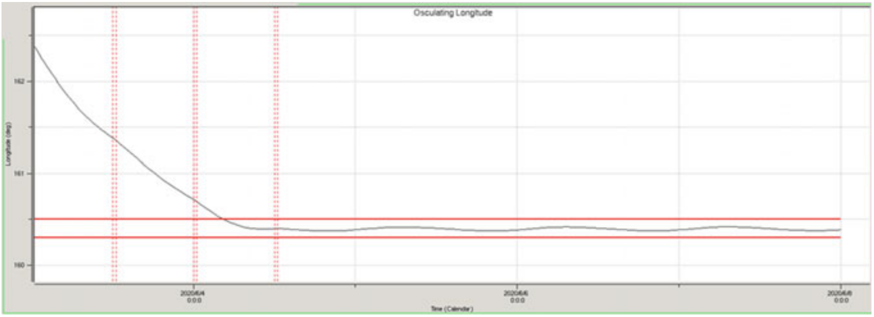


Fig. 2 Longitude evolution through a sample backup drift-stop plan, with manoeuvres spaced only 12 h apart

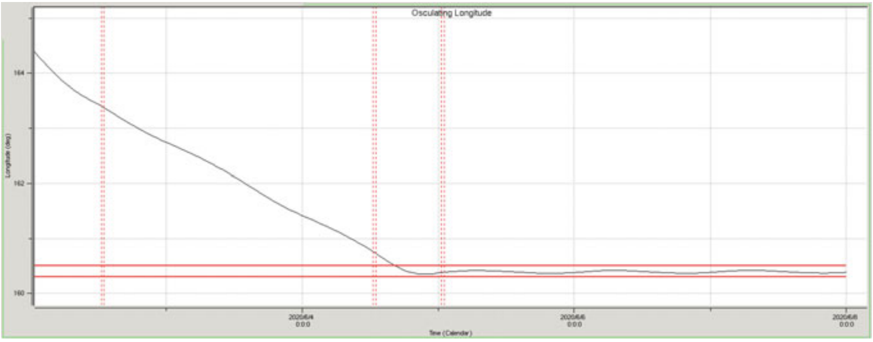


Fig. 3 Longitude evolution through a sample primary drift-stop plan, which in this case allows 48 h between the first and the second manoeuvre

satellites operating beyond the target orbital position—this concept will be expanded in Sect. 3.9.

If instead the first drift-stop manoeuvre delivers acceptable performance, the extra time before the next manoeuvre in the nominal drift-stop plan can be used to plan and execute a tiny touch-up manoeuvre to fine-tune the orbit.

3.3 Operational Ephemerides Distribution

Once the nominal drift-stop plan has been completed, the resulting orbital ephemerides can be distributed to all the conjunction assessment service providers. Avanti relies on multiple resources, including the 18th Space Control Squadron (18th SPCS), the Space Data Association (SDA), the Space Surveillance and Tracking service of the European Union (EUSST), the Royal Air Force Commercial Integration Cell (RAF CIC), and commercial services. Further information on the interfaces between Avanti and these entities are provided in Sect. 3.10.

For those operators that have signed a SSA Sharing Agreement with United States Strategic Command (USSTRATCOM), it is good practice to raise an Orbit Data Request (ODR) about one month prior to drift-start [8]. This will not only give access to the Advanced Services of the 18th but also mitigate the risk of close approach with an object outside the public catalogue. An ODR should be raised no later than 30 days before drift-start [8].

If it is unavoidable to cross the station-keeping box of another satellite during drift-start or drift-stop, for example, due to limitations of the ownship thruster system, it is good practice to exchange ephemerides. For SDA members, ephemerides-vs-ephemerides conjunction assessment is done automatically [9]. In all the other cases, it has to be arranged with the other operator. For this purpose, it is good practice for each satellite operator to include their own contact details in their Space Track account [5].

3.4 Identification of Suitable TT&C Resources and Ranging Requirements

Once the mission-planning phase is completed, suitable TT&C antennas must be chosen. To mitigate the risk of interruption to the capability to send commands and retrieve telemetry, Avanti asks two antennas for the drift-start and drift-stop phases, in order to ensure dual-station ranging and redundancy during critical operations. One single tracking antenna is considered adequate for the free-drift phase. As mentioned in Sect. 3.2.3, antennas used to support drift-stop operations should have adequate visibility of both sides of the target orbital position.

In case of long drifts that cannot be covered by one single antenna, the ground station network should be chosen in such a way to guarantee a suitable overlap of the visibility intervals of the antennas doing the handover. In order to ensure enough time for antenna testing and calibration, Avanti prefers overlap of about ten days—how much this translates into longitude depends on the longitude drift rate of the specific relocation.

In order to minimize the risk of poor orbit determination performances, the antennas' longitude should differ from the target orbital position by at least 10°, and the geodetic coordinates of each pair of antennas should differ by at least 10°. Mono-pulse antennas are preferable to Step-track.

In Avanti experience, it is advisable to increase the frequency of ranging measurement as critical operations loom. While in station-keeping and free-drift phases, one ranging per station per hour is sufficient, between 12 h prior to the first manoeuvre and 12 h after the last, the frequency is increased to one measurement every 15 min.

Once suitable TT&C infrastructure has been identified, the process to ensure full compliance with the Radio Regulations, as described in Sect. 2.2, can start. Note that a change to the TT&C infrastructure might potentially require a minor change to the choice of the station-keeping box at destination, depending on the off-axis radiation of the chosen TT&C site. Regulatory activities would run in parallel with all the operational activities needed to connect, test, and calibrate the TT&C resources. These are described in Sect. 3.5.

3.5 Networking and Baseband

Relocating a spacecraft is easier when the operator already has access to a proprietary network of TT&C resources that provide uninterrupted visibility of the geostationary orbit. Avanti is a regional operator, hence, it does not have this luxury, and third-party TT&C resources must be linked to the operational network prior to the start of the relocation. In particular, the changes to a mission profile require a modification to the way the TT&C resources are managed by the Avanti Mission Control System (MCS). In Avanti's case, this conflicted with the original MCS requirements, which originally assumed that each satellite would operate at the same orbital position throughout its operational lifetime. This is reflected in how the Avanti MCS manages the configuration of TT&C resources, where the data is held in static configuration variables that are read during the initialisation of the software.

3.5.1 Redundancy

Configuration of the TT&C network needs to support a mission critical service, especially during critical spacecraft operations. In Avanti's case, we have opted for a minimum of two independent network paths, from our geographically diverse operations sites to the third-party TT&C suppliers. This solution is sketched in Fig. 4. Our TT&C network also needs to be agile and cost effective, which means Avanti has selected to use an Internet Protocol Security (IPSec) tunnel rather than dedicated E1/T1 lines.

In Avanti's case, the required data transfer rate for housekeeping telemetry is very low. Thus, two diverse dual redundant IPSec tunnels across the public internet (or a cloud provider's network) to our third-party TT&C antennas are able to balance our network requirements of confidentiality, integrity, availability, and flexibility.

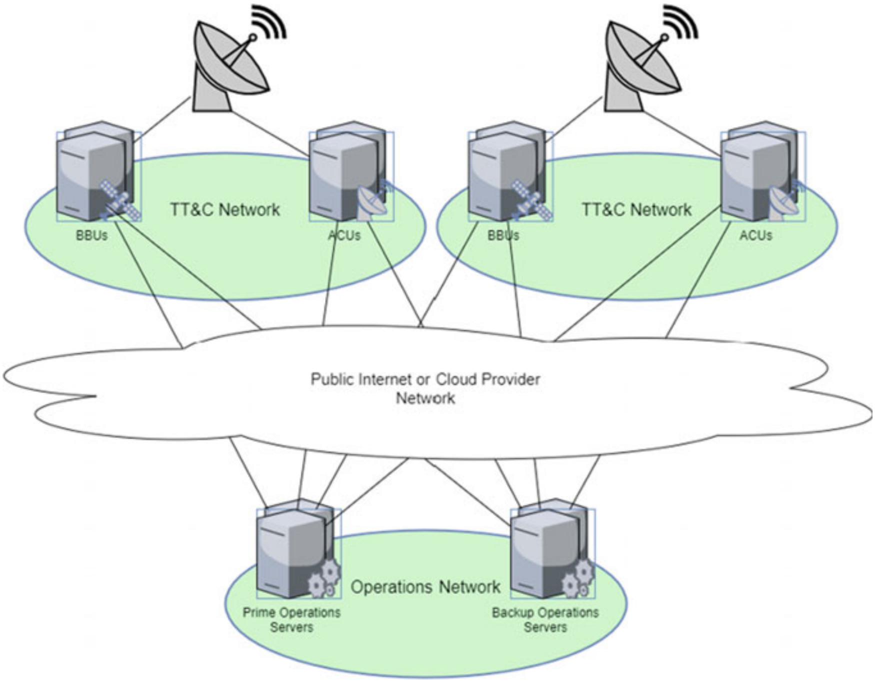


Fig. 4 TT&C Network Overview

3.5.2 Cortex Linking

Configuration of the baseband units with a third-party supplier required a different process than how Avanti managed the configuration of its baseband units. For a GEO mission, the baseband unit would be configured at mission start and the configuration captured at the end of system acceptance testing. Thus, the configuration file of a baseband unit would contain both generic and specific configuration ground and spacecraft parameters. Sharing this configuration file with a third-party TT&C supplier is neither practical nor desirable, as it contains sensitive information not needed by a third-party. Even with a Non-Disclose Agreement (NDA) in place with a third-party, it is good practice to apply a "need to know" policy.

The option Avanti chose was to extract the baseband parameters that the third-party supplier needs and capture them in an American Standard Code for Information Interchange (ASCII) file that complies with the INI format. This provided a highly structured human readable file that can be shared with third-parties, who could then integrate into their configuration process as they saw fit.

The INI format is also compatible with the Python configparser module (which is available on the operating system of Avanti's MCS), thus enabling the use of certified RPMs from a trusted source. This removes the need to perform additional threat and vulnerability checks, which can be time-consuming and costly.

It also opens the possibility to use the MCS automation scripting language, Satellite Procedure Execution Language & Library (SPELL), which is built on Python, to interface with the baseband application programming interface (API) to check the deployed configuration matches the system configuration model.

For the TT&C management, Avanti chose to create a new tool where an engineer could capture all the static parameters related to the TT&C resource required for the relocation and on-station phase on the mission profile. This tool can generate the TT&C configuration files required by the MCS and any other software application for any specific phase on the mission. This tool contains a database of the configuration of all TT&C resources that we have been used to date. Thus, if a TT&C resource is used again at a later phase of the mission profile, Avanti can be sure that the MCS is configured the same as the previous time, thereby ensuring that the telemetry archives are consistent throughout the life of the mission. This is important for any telemetry that is stored as raw, where calibration is applied during retrieval.

3.5.3 TT&C Testing (Ownship not Necessarily in Line-Of-Sight)

The link to third-party TT&C resources needs to be tested prior to operational use. Housekeeping telemetry tends to be a low transmission rate. Receiving a 512 kbit transfer frame every 2 s is quite normal for a spacecraft in geostationary orbit. Thus, bandwidth demands on the network are secondary to the need for a reliable link between the MCS data centres and third-party TT&C resources.

To profile a network link to a third-party TT&C resource, which is usually an IPSec tunnel across the public internet or via a cloud provider, the network tools that are available on the operating system of the MCS should be used. The only option open to Avanti, to avoid an install of untrusted software, is to use the ping command. Where the payload size is increased to match the Maximum Transmission Unit (MTU) of the network and the interval set to match the TM transfer frame rate of ownship.

During the testing phase, the ping statistics should be captured every hour to build a picture of the reliability of the network link. It is good practice to allow the duration of the testing to be 7 to 10 days. From the collected data, it is possible to infer the probability of experiencing a telemetry drop, the time of day at which that drop could happen, and for how long. It can also be used to determine if the IPSec end point timers are aligned.

3.6 TT&C Provisioning

After baseband connectivity with the operational network has been established, and assuming ownship is within visibility, the new TT&C resource needs to undergo a comprehensive testing phase. In case ownship departing orbital location is not within visibility of the new resource, this testing phase must be conducted during the free-drift phase (see Sect. 3.2.2), which adds an element of time pressure on it.

3.6.1 TT&C Testing (Ownship in Line-Of-Sight)

A live TM/TC testing session should be conducted prior to switching over to a newly acquired TT&C site, when ownship is within the visibility of the TT&C site. The aim of TC testing is to ensure the desired on-board receiver power level is achieved and different combinations of uplink frequencies and polarisations are tested and recorded for future reference. For the TM testing, the aim is to achieve stable TM status on the ground control system with reasonable downlink Eb/No on the cortex.

3.6.2 TT&C Calibration

For more accurate orbit determination using a newly acquired TT&C site ranging data, it is good practice to account for the delays introduced into ranging data by ground equipment. To measure these delays, a long-loop test, or station calibration, is carried out. This typically entails de-pointing the antenna from the spacecraft and introducing a Test Loop Translator (TLT) into the RF loop, to take the ranging delay measurement for different combinations of the uplink frequency, polarisation, and downlink frequency.

The result of the long-loop test can be inputted to the Flight Dynamics (FD) system, which will discount this value from the two-way light-time read for every ranging measurement.

To improve further the orbit determination results with the new TT&C resource, in Avanti's experience, there are three options:

1. Schedule a period without orbital correction manoeuvres or wheel off-loadings—ideally 48 h or more—fully within the visibility of the new TT&C resource, then collect ranging, azimuth, and elevation measurements on at least an hourly basis. If the geometry is not too poor, this should allow for successful orbit determination and estimation of the remaining ranging, azimuth, and elevation biases.
2. Schedule a period of dual-station ranging, in which the satellite is fully within the visibility of both the new TT&C resource and of a station whose ranging delay is well known and stable. In the FD software, allow the uncertainty on the ranging delay to be high for the former and very little for the latter station. This should allow the orbit determination to converge to a reasonable estimate for the new station ranging delay.
3. Request support from commercial third-parties, which can assist in calibrating the station delay by relying on a proprietary network of well-calibrated optical sensors.

It is worth pointing out that the ITU itself recommends performing station calibration to reduce orbit determination error [4]. Hence, good orbit determination is a regulatory requirement.

3.7 Distribution of Flight Dynamics Products

Once the Mission Planning phase (Sect. 3.2) has been completed and all TT&C resources are fully integrated within the operational software (as long as regulatory approval has been received), the actual relocation can be scheduled and all operational products can be distributed to the intended partners. These products include:

- Manoeuvre plans, to allow the scheduling of on-site support to critical operations even out of nominal office hours;
- Operational ephemerides, to be loaded to conjunction assessment portals and distributed to partners and fellow operators (see also Sect. 3.3);
- Associated TLEs, Intelsat 11 Parameters files, and Azimuth, Elevation & Ranging (AER) data for all ground stations, to be used in the event of a ground station anomaly to assist in the re-acquisition of ownship, or to keep the station in command-track if absolutely necessary;
- Eclipses, sensor blindings, visibility events, and longitude predictions, to aid scheduling and allow de-conflicting operations;

These mission products then drive the operations planning described in the next sections. Operations may be time dependent, or longitude dependent, so having the mapping between the two allows greater planning accuracy.

3.8 Operations Planning

The execution of a relocation requires careful management of satellite resources to ensure robust connectivity, a stable thermal environment, and power management. The operations involved also need to be de-conflicted from the ‘nominal’ operations that would be performed on the spacecraft, so spacecraft relocation causes a large amount of careful planning to ensure that the correct resources are in place and the drift goes smoothly and concludes safely. This is generally coordinated via a Sequence of Events (SoE), which provides a sequential visualisation of all the events that are involved in the whole relocation timeline from start longitude/date to end longitude/date. For each longitude slot within the relocation path, the Sequence of Events itemises:

- **Operational Commanding Activities:** This includes drift–start and drift-stop manoeuvres, ranging campaigns, solar array re-peaking, and so on.

- **TT&C Management**: It is vital that stable TT&C links are maintained during the relocation, particularly for the drift-start and drift-stop manoeuvres. As the spacecraft orbital location changes, several TT&C ground resources at diverse sites may be required to maintain visibility. Redundancy must also be provided, in networking and in ground station resources and, at most times, hourly dual station ranging is required to enable stable orbit determination. During the drift, switching operations and ranging operations are carried out and in cases where third party services are used to provide TT&C support, this may require coordination with the third party out of hours. It is therefore important to plan the TT&C resources as much as possible before the start of the relocation and the SoE is used to document the prime site in use, the redundancy options available along with antenna coverage and tracking configurations.
- **Hinder Management:** During relocations, ownship will pass above or below other spacecraft operating within the geostationary belt and in order to avoid interference issues, the passes must be tracked and planned for. The dates and times of expected passes are recorded in the SoE. Hinder management is discussed further in Sect. 3.9.
- **TCR Configuration**: In conjunction with the TT&C provision and hinder management, it is often necessary to reconfigure spacecraft telemetry, command and ranging (TCR) subsystems. The SoE therefore documents the active onboard Telemetry Transmitter for telemetry downlink and the active Command Receiver that is being selected for uplink, to comply with frequency hinder requirements from other operators. If there are no hinder requests for a particular orbital location, the default TCR configuration is selected.
- **Spacecraft Platform Bias**: For situations where ground stations are at the fringes of antenna coverage, or when using narrow coverage antennas, attitude must be more carefully controlled with respect to ground station, so it is common to adjust platform bias throughout the drift. Progressive pitch and roll operations are often carried out. Note that platform bias needs to be considered when planning orbital correction manoeuvres in the drift-start and drift-stop phases.
- **Eclipse Operations**: To ensure the power subsystem is prepared for the eclipse season and correctly configured following it, balancing, charging/discharging, and configuration operations are carried out. These operations cannot be executed simultaneously with the drift-start or drift-stop manoeuvres, so have to be de-conflicted. The timings of the eclipses themselves will vary according to the mission profile, and are provided by FD (see Sect. 3.7). The same applies to any other geometry-dependant event, such as sensor blindings. Note that some spacecraft cannot execute out-of-plane manoeuvres during eclipse season, and this potentially needs to be taken into account when pre-biasing the inclination, as mentioned throughout Sect. 3.2.
- **Solar Array Re-peak**: As the drift progresses and therefore the orbital location changes, the incident sun angle on the solar arrays' changes. In absence of re-peaking, the cumulative off-pointing angle during the drift will be the difference between the current longitude and the start longitude. For spacecraft that do not

have auto-tracking solar arrays, or have chosen to have it disabled, it is necessary to monitor the array power and to periodically re-peak the solar array angle.

3.9 RFI Coordination (FLYBY) and Hinder Management

One reason for which the SoE is useful is that it summarizes commanding activity over time and longitude, which can be used as an input in the process to manage ASI. During all satellite operations, and particularly during relocations, regulatory guidelines and responsible good practice mean that radio-frequency emissions should be carefully controlled to avoid interference with other spacecraft. Avanti makes every effort to coordinate drift operations via FLYBY email notifications distributed to other operators.

Avanti's approach is to despatch FLYBY notifications regarding an up-coming relocation well before the drift start to allow ample time for frequency coordination. Where analysis shows a potential for ASI, coordination may be started several months in advance, and for all other operators, this is done at least 30 days prior to drift start. These messages include information about ownship catalogue number, initial longitude, drift direction, TT&C frequencies (payload frequencies are not included as all payload is off during relocations), and TT&C sites. Figure 5 shows an overview of the FLYBY coordination process.

In order to mitigate the risk of ASI in case of overshoot of the target orbital position (for example, because of an in-orbit anomaly during drift-stop operations), some operators may choose to communicate in the FLYBY notice a relocation arc slightly wider than the expected one. In this case, it is good practice to privately communicate what the actual plan is to the operators of satellites in the neighbourhood of the target orbital position, in order to coordinate operations if necessary.

Most operators would usually provide response even when there are no ASI concerns, but some would raise hinder requests. Different requests are handled in different ways:

- **Single Uplink or Downlink hinder request**: If another spacecraft operator requests a hinder of only one of the telemetry (TM) or telecommand (TC)

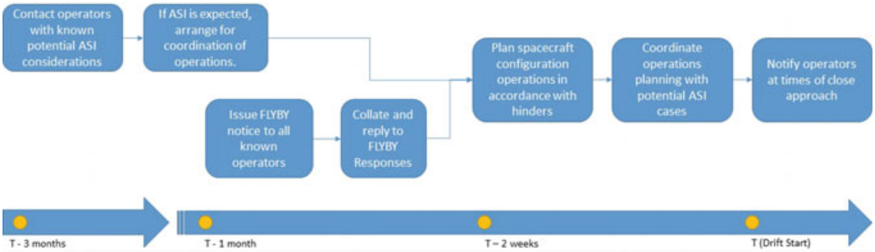

Contact operators with known potential ASI considerations
If ASI is expected, arrange for coordination of operations.
Issue FLYBY notice to all known operators
Collate and reply to FLYBY Responses
Plan spacecraft configuration operations in accordance with hinders
Coordinate operations planning with potential ASI cases
Notify operators at times of close approach
T - 3 months
T - 1 month
T – 2 weeks
T (Drift Start)

Fig. 5 FLYBY coordination process

frequencies, the alternate one(s) are still available during the flyby. For single TM hinders, an on-board TCR reconfiguration will be required to switch to the alternate Telemetry Transmitter (TTX). For single TC hinders, ground station reconfiguration will be required to select the alternate Command Receiver TC frequency.

- **Dual Uplink Hinder request**: If a dual TC hinder is received from another spacecraft operator, this implies all command carriers will need to be lowered within usually ± 1° of the affected spacecraft's location. In most cases, this kind of hinder request is usually met easily, as no commanding is usually required during free-drift, except for contingency situations. For the rare cases of planned operations during the free-drift, or when the affected spacecraft is in close proximity with the drift–start / drift-stop longitudes, the affected spacecraft operator will usually have to indicate which of the ownship TC frequencies will pose minimal ASI inconveniences. Furthermore, ranging measurement and other non-critical commands can be scheduled less frequently throughout the hinder period to further mitigate interference.
- **Dual Downlink Hinder request**: If a dual TM hinder request is received, the affected spacecraft operator will usually indicate which of the ownship TM frequencies will pose minimal ASI inconveniences. At least, one of the TM frequencies needs to be active to provide the ownship health status.

Once all hinders have been agreed with the requesting parties, these can be inserted into the Sequence of Events.

3.10 Conjunction Assessment and Collision Avoidance

The risk of close approach with space debris during any operation is common in the current space environment. Drift-start operations can be optimized and rescheduled to avoid any close approach, and potential impacts to the drift-stop planning can be mitigated by altering the longitude drift rate accordingly. Other phases of the relocation can be more difficult to handle, though.

For each conjunction, Avanti usually makes a preliminary assessment of the collision risk by comparing the radial miss-distance at the Time of Close Approach (TCA) to the sum of the radial components of the covariance matrices.

In Avanti's experience, during free-drift, when ownship is not manoeuvring, the predictions provided by the 18th SPCS are very reliable and consistent. This means that sub-kilometre radial miss-distances can be acceptable. Should the analysis highlight that a collision avoidance manoeuvre is necessary, then Avanti would prefer the execution of an East or West manoeuvre to increase the radial component of the miss distance. The choice of the manoeuvre direction would depend on the geometry at TCA or, more specifically, on the sign of the radial component of the miss distance.

Unfortunately, any in-plane manoeuvre during free-drift would alter the drift-stop operations planning. There are several reasons for which this can be undesirable, for example, because of regulatory deadlines, or because of resource allocation and scheduling conflicts, or again because the new orbital eccentricity would cause a violation of the IADC guidelines. For these reasons, after TCA, it might be necessary to perform another manoeuvre, in the direction opposite to the collision avoidance manoeuvre, to re-establish the agreed timeline.

A close approach during drift-stop operations is the most operationally challenging. If the event is reported sufficiently in advance, depending on the other operational and regulatory constraints mentioned above, it might still be worth to anticipate one of the drift-stop manoeuvres to delay ownship arrival up until after TCA. Should that not be possible, then it would fall upon the FD team to design a set of drift-stop manoeuvres that guarantee sufficient separation. It can happen that the debris is drifting very slowly through the target orbital position. In that case, an option could be to analyse the orbital inclination and eccentricity of the debris, and design a set of drift-stop manoeuvre that include ownship and the debris into a co-located cluster, using an eccentricity/inclination separation strategy.

Collision avoidance can happen only when timely and reliable information is available. Several institutional, not-for-profit, and commercial entities provide conjunction assessment services for operators.

Most of these services are two-tier systems that first perform an initial assessment using the data of the 18th Space Control Squadron, then refine the solution by both tasking additional observation from available sensors, and using operational ephemerides provided by the operator (see Sect. 3.3).

As mentioned before, Avanti relies on the 18th SPCS, the EUSST, the SDA, and the UKSpOC CIC. On top of distributing operational ephemerides to all these players, Avanti informs both the 18th and the UKSpOC of its long-term plans by raising an Orbit Data Request.

When the 18th sends a CDM to Avanti, this gets automatically distributed to partners as well. The Space Data Association re-runs the conjunction assessment using operational ephemerides from its members, while the EUSST performs an independent analysis of high-interest events by tasking its own sensors. The UKSpOC instead assists in the coordination between civilian and military satellite, which are usually not included in the public catalogue on which CDMs are based. High-level schematics of the interfaces among Avanti and its partners in assessing conjunctions are provided in Fig. 6.

3.10.1 Change to Concept of Operations in Response to COVID-19

While non-routine operations are intrinsically riskier than routine ones, the Coronavirus Disease 2019 (COVID-19) pandemic in 2020/21 posed additional unexpected challenges. This section will elaborate on how Avanti faced this crisis.

The pressure to implement the social distancing guidelines of the World Health Organization and national government triggered a digital transformation roadmap with the timescales of hours and days rather than months or years. In the case of Avanti, implementing social distancing guidelines meant that virtual private network

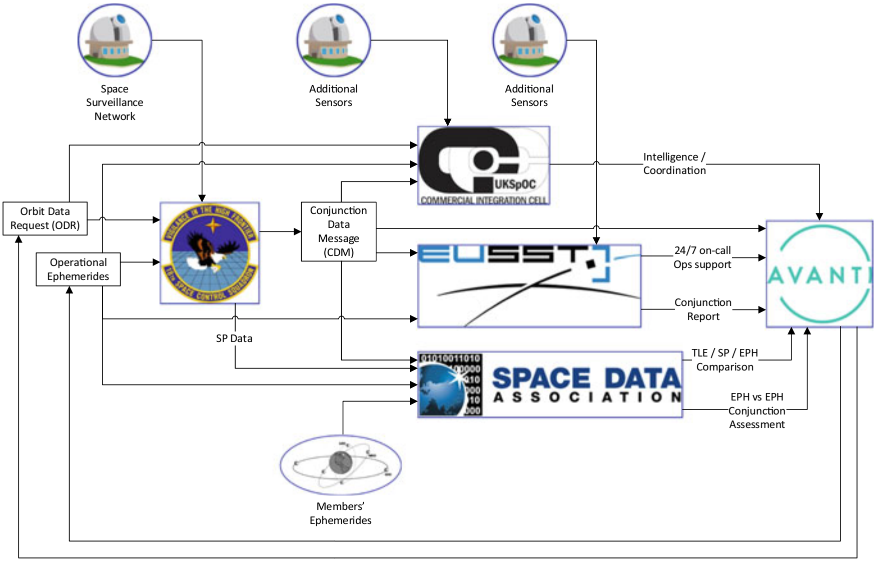


Fig. 6 Schematics of interfaces for conjunction assessment

(VPN) access points—previously only used by the ground systems team for command line administration activities—were needed to support the other spacecraft operations teams.

The choices that Avanti made were driven by exploiting the existing design of the Satellite Operations Module (SOM) network rather than deploying new technologies. This was to minimise the disruption of the controller beyond what was caused by the pandemic. For Avanti, the key to understand how to change the concept of operations was to put the controller at the centre of the model, by looking at what the controller needs from the system, and vice versa.

The changes introduced to the SOM were different for each of the four teams of the Spacecraft Operations Centre:

- **Spacecraft Controllers Team (SCT)**: It was clear from the beginning that the SCT could not work away from the office, as the controller's needs from the system are nearly impossible to replicate at home. Thus, what the concept of operations needed to do was to change to remove all face-to-face interaction between members of SCT.
- **Flight Dynamics (FD) Team**: For the members of the FD team, the needs of the systems were to allow them to access their individual workstations in the SOM from outside the office. Thus, what the system needed to do was to perform identity authentication and only allow access to their own workstation.
- **Spacecraft Operations Engineering (SOE) Team**: The SOE team has two distinct needs from the system. One is to follow the spacecraft operations alongside the SCT, the other is to run procedure verification and training sessions. In the

former case, what the system needs to do was to perform identity authentication and to allow 'view only' access to the MCS. In the latter case, the FD access model applies.

- **Ground Systems (GS) Team**: No changes to the SOM were required.

3.10.2 Staffing and Resource Allocation

The operations necessary to carry out an orbital relocation are in addition to those required to operate a geostationary satellite, and to those required to operate the rest of the fleet. Therefore, it is usually necessary to provide additional staffing support for relocations.

In the case of Avanti, the SCT already provides 24/7 control and monitoring of the fleet and can comfortably manage the additional relocation operations. However, to account for the rare case where the Prime Satellite Control Centre becomes unexpectedly unavailable, for example, during a fire evacuation, the backup SCC must also be staffed by members of the SCT during critical operations.

Since the drift-start and drift-stop manoeuvres are particularly critical, additional support is also needed from other teams. The FD team provides cover during the manoeuvre file generation and delivery. A portion of this typically occurs out-of-hours, as orbit determination from each manoeuvre feeds into the planning of the next, as mentioned in Sect. 3.2.1. The SOE team also provides on-site support to the manoeuvres to ensure that engineering expertise is readily available and to address any issues as soon as possible. During the free-drift phase, the SOE team supports any other ad hoc activities as necessary, such as TCR reconfigurations, solar array re-peaking and platform biasing. Similarly, the GS team supports to ensure quick resolution of any system issues. Where third-party TT&C resources are used, provider on-site support is requested.

In order to provide further mitigation against any potential disruption during the critical drift/stop manoeuvre phases, an operational change freeze is implemented across all operational departments to ensure stable TT&C links and SOM inter-site connectivity are maintained. During this time, no changes, maintenance or testing are carried out if it is deemed to present a risk.

4 Timeline

In the previous sections, we covered the regulatory and operational activities that need to be performed to mitigate the risks associated with non-routine operations. The timeline of these activities is governed by several factors, including the guidelines of the national regulatory authority and resource availability. For many activities, though, there are no established best practices on when to initiate them.

In the case of Avanti, the timeline is mainly governed by the CAA guideline to allow a minimum of six months for the processing of an application (see Sect. 2). This

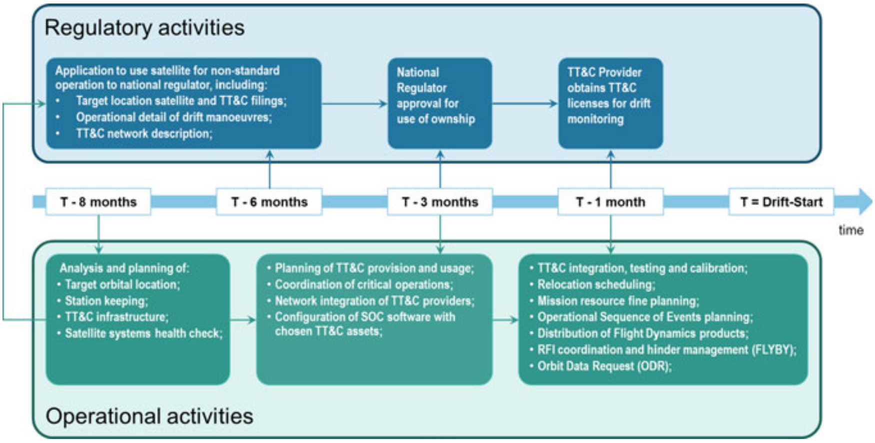

Regulatory activities
Application to use satellite for non-standard operation to national regulator, including:
• Target location satellite and TT&C filings;
• Operational detail of drift manoeuvres;
• TT&C network description;
National Regulator approval for use of ownship
TT&C Provider obtains TT&C licenses for drift monitoring
T - 8 months
T - 6 months
T - 3 months
T - 1 month
T = Drift-Start
time
Analysis and planning of:
• Target orbital location;
• Station keeping;
• TT&C infrastructure;
• Satellite systems health check;
• Planning of TT&C provision and usage;
• Coordination of critical operations;
• Network integration of TT&C providers;
• Configuration of SOC software with chosen TT&C assets;
• TT&C integration, testing and calibration;
• Relocation scheduling;
• Mission resource fine planning;
• Operational Sequence of Events planning;
• Distribution of Flight Dynamics products;
• RFI coordination and hinder management (FLYBY);
• Orbit Data Request (ODR);
Operational activities

Fig. 7 Illustrative timeline of regulatory and operational activities

essentially requires that a number of activities, like the analysis of the target orbital position (Sect. 3.1) and the identification of suitable TT&C resources (Sect. 3.4), are performed up to eight months prior to drift-start to mitigate regulatory risks.

In Avanti experience, the work to integrate third-party TT&C resources into the operational network should start no later than three months prior to drift-start with resource planning, cortex linking, and network testing (Sect. 3.5). Coordination activities like raising an ODR, distributing FD products (Sect. 3.7), or initiating a FLYBY process (Sect. 3.9) should start no later than one month before drift-start. These will feed into a preliminary Sequence of Events, which will be then updated with hinder management activities as request from other operators in response of the FLYBY notice become available (see again Fig. 5 for details).

If all goes as planned, regulatory and operational preparatory activities come to completion about one month prior to drift-start, when the TT&C antennas are properly calibrated (Sect. 3.6) and the TT&C provider obtains the proper licenses to support the non-routine operation with the chosen sites.

A high-level illustrative timeline of regulatory and operational activities is provided in Fig. 7.

5 Conclusions

From its inception in 2002, Avanti evolved into a significant regional operator (Fogg et al. [1]), which continues to be at the forefront of technological advancement in the industry. Nimbleness to relocate its satellites in order to respond to shifting business needs is part of Avanti's success story. In this paper, Avanti has summarized how operational best practices for safe non-routine operations flow from international treaties and national laws, and has shared its experience in implementing these best practices during the preparation and execution of spacecraft relocations.

Avanti's experience highlights how a close collaboration between the regulatory and operations teams is necessary to properly mitigate all the risks associated with non-routine operations, and how constraints on one side would translate on requirements on the other. Avanti is committed to safe operations in space and shares the lessons learnt from its experience to contribute to the preservation of the space environment.

Appendix

Acronyms and Abbreviations

18th SPCS	18th Space Control Squadron
AER	Azimuth, Elevation, and Ranging
ASCII	American Standard Code for Information Interchange
ASI	Adjacent Satellite Interference
CAA	Civil Aviation Authority
CDM	Conjunction Data Message
CIC	Commercial Integration Cell
COVID-19	Coronavirus Disease 2019
Eb/No	Energy per Bit to Noise Power Spectral Density Ratio
EUSST	Space Surveillance and Tracking Service of the European Union
FD	Flight Dynamics
GEO	Geostationary
GS	Ground Systems
IADc	Inter-Agency Space Debris Coordination Committee
IPSec	Internet Protocol Security
ITU	international Telecommunications Union
LEOP	Launch and Early Orbit Phase
MCS	Mission Control System
MTU	Maximum Transmission Unit
NDA	Non-Disclosure Agreement
OD	Orbit Determination
ODR	Orbit Data Reques
Ofcom	Office of Communications
'Ownship'	Satellite to be relocated
RFI	Radio-Frequency Interference
RPM	RedHat Package Manager
SCT	Spacecraft Controllers Team
SDA	Space Data Association
SPELL	Satellite Procedure Execution Language & Library
SoE	Sequence of Events

SOC	Spacecraft Operations Centre
SoE	Sequence of Events
SOE	Spacecraft Operations Engineering
SOM	Satellite Operations Module
SSA	Space Situational Awareness
SSN	Space Surveillance Network
TC	Telecommand
TCA	Time of Close Approach
TCR	Telemetry, Command and Ranging (spacecraft subsystem)
TM	Telemetry
TLE	Two-Line Elements
TLT	Test Loop Translator
TTX	Telemetry Transmitter
TT&C	Telemetry, Tracking, and Commanding
UCS	Union of Concerned Scientists
UNCPUOS	United Nations Treaty on Peaceful Use of Outer Space
UKSpOC	United Kingdom Space Operations Centre
USSTRATCOM	US Strategic Command
VPN	Virtual Private Network

References

1. Fogg M, Arona L, Loveday M, (2018) Spacecraft operations: defining the avanti way. In: SpaceOps conference, Marseille, France. https://arc.aiaa.org/doi/pdfplus/10.2514/6.2018-2444
2. ITU Constitution and Convention. https://www.itu.int/en/history/Pages/ConstitutionAndConvention.aspx
3. ITU Radio Regulations. https://www.itu.int/pub/R-REG
4. Recommendation ITU-R S.484-3. https://www.itu.int/dms_pubrec/itu-r/rec/s/R-REC-S.484-3-199203-I!!PDF-E.pdf
5. Space Track. https://www.space-track.org/
6. UCS Satellite Database. https://www.ucsusa.org/resources/satellite-database
7. Support to IADC Guidelines Rev5.7 (2019). https://www.iadc-home.org/documents_public
8. 18th Space Control Squadron, Spaceflight Safety Handbook for Satellite Operators.pdf. https://www.space-track.org/documents/Spaceflight_Safety_Handbook_for_Operators.pdf
9. Space Data Association. http://www.space-data.org/sda/

Parker Solar Probe Pre-launch Mission Operations Orbit-In-The-Life Mission Simulation

Kimberly J. Ord

Abstract Launched on a Delta IV-Heavy rocket from Cape Canaveral on August 12, 2018, NASA's Parker Solar Probe (PSP) will travel closer to the Sun than any other spacecraft. PSP is designed to complete 24 solar encounters over its seven-year mission. During the 24 orbits, PSP gradually shrinks its orbit around the Sun, coming as close as 3.83 million miles using a series of seven Venus flybys. The spacecraft will explore the inner region of the heliosphere and perform in-situ and remote sensing observations of the magnetic field, plasma, and accelerated particles. The spacecraft was designed and built by the Johns Hopkins University Applied Physics Laboratory (APL) in Laurel, Maryland, where mission operations are currently conducted. This paper describes the development and execution of the mission operations "Orbit-in-the-Life" and early operations mission simulations performed using the spacecraft during thermal vacuum testing. The author will first discuss selection and organization of activities and events to be tested and effects of the spacecraft engineering team's desire to use mission operations tests to also perform spacecraft performance requirement testing. Next, the author will discuss selecting the orbit to be performed during the "Orbit-in-the-Life" test and the challenges for condensing a one-hundred-and-twenty-day orbit into a ten-day test. Then, preparing for the test, including executing an accelerated version of the orbit planning process with the four instrument teams and pre-testing the tests using the hardware-in-the-loop simulator will be described. Last, executing the test will be discussed including dealing with anomalies and lessons learned.

Keywords Testing · Simulations · Planning

K. J. Ord (✉)
Space Exploration Sector, Johns Hopkins University Applied Physics Laboratory, 11100 Johns Hopkins Road, Laurel, MD 20723, USA
e-mail: Kimberly.ord@jhuapl.edu

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_3

Nomenclature

AU	Astronomical Unit is the distance from the Sun to the Earth. One AU equals 149,597,870,691 m.
Rs	Solar Radius is the distance from the center of the Sun to its surface, 696,000 kilometers

1 Introduction

The objective of this paper is to discuss the development and execution of the Parker Solar Probe (PSP) Mission Simulation #3 performed during thermal vacuum testing (TVAC). The test was broken into two parts. The first part included tests from multiple orbits during thermal balance testing. The second part was an accelerated orbit-in-the-life test for Orbit 5 during thermal cycle testing. The paper will begin with an overview of the spacecraft, instruments, and mission. Then, it will summarize Mission Operations spacecraft testing in general and during thermal vacuum testing in particular. Next, the detailed discussion of Mission Simulation #3 testing will begin, including: activity selection, orbit selection, test preparation and coordination, and test execution. Finally, anomalies and lessons learned will be described.

2 Mission and Spacecraft Overview

2.1 Mission Overview

PSP was launched on 12 August 2018 on board a Delta IV-Heavy rocket from Cape Canaveral, Florida. The spacecraft is in a highly elliptical heliocentric orbit (See Fig. 1). The prime science period for each orbit is when the spacecraft is less than 0.25 astronomical units (AU) from the Sun, which is indicated by the white dashed circle in Fig. 1. This period is referred to as solar encounter. At the time of the SpaceOps 2021 conference, PSP will just be completing its 8th solar encounter, which was from 24 April–4 May 2021.

PSP is designed to complete 24 solar encounters and will use seven Venus gravity assist flybys to decrease the orbit period from 162 to 88 days and the perihelion from 35.7 Rs to 9.86 Rs. The fourth Venus gravity assist flyby was completed on 20 February 2021. The orbits are referred to by their orbit number, starting with #1 being the first after launch.

At a top level, orbits in the PSP trajectory are similar. The central feature of each is the solar encounter period, centered roughly ±6 days around perihelion. During this time, the spacecraft is primarily devoted to science measurement campaigns.

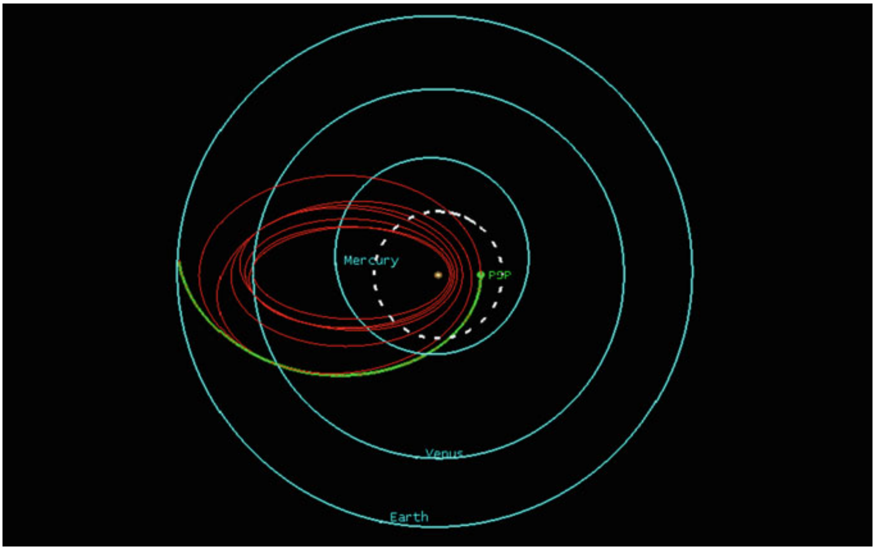


Fig. 1 PSP mission trajectory

Communication with the ground is limited, so science data and housekeeping data are stored on a solid-state recorder (SSR). The portion of orbits outside 0.25 AU are the cruise/downlink segments, where science data downlink occurs as communications are available and operations such as trajectory correction maneuvers (TCMs) take place.

2.1.1 Orbit Attitude Regime Overview

PSP operates in four attitude regimes based on the solar distance. The first is Aphelion. In this regime, the spacecraft + Z axis is off pointed from the sun by 45° in order to allow sun on the radiators to warm the cooling system. Aphelion attitude is required when the spacecraft is >0.79 AU from the sun. The second attitude regime is Aphelion-Variable. It is flown when the spacecraft is between 0.7 AU–0.79 AU from the sun. In the Aphelion-Variable regime, the spacecraft + Z axis can be pointed between 0°–45° from the sun and the off pointing is used mostly for communications. The third attitude regime is Umbra. Umbra is used between 0.25 AU–0.7 AU. In the Umbra regime the spacecraft + Z axis is pointed at the sun. However, rolls around the sunline are allowed. The last attitude regime is Encounter. This attitude is used when the spacecraft is in a solar encounter at < 0.25 AU from the sun. In this attitude, the spacecraft + Z axis is pointed at the Sun and the + X axis is in the velocity vector. Aphelion and Encounter attitudes are shown in Fig. 2.

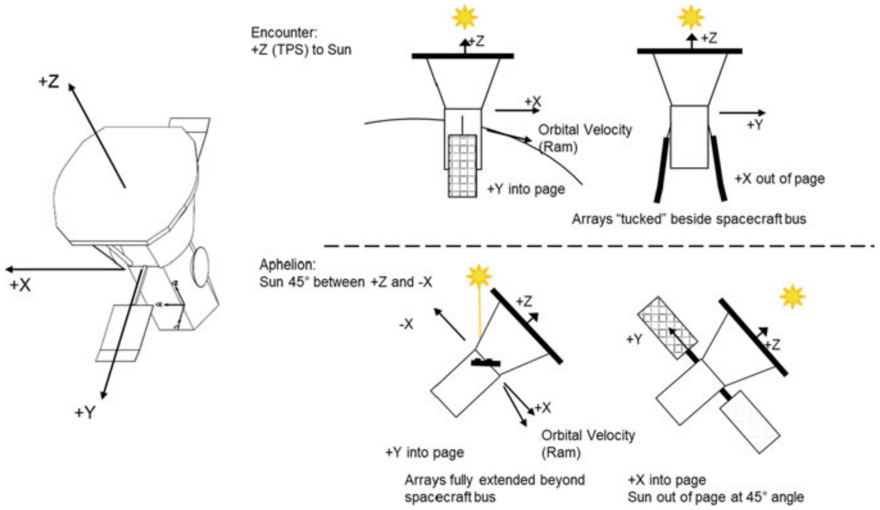


Fig. 2 Encounter and aphelion attitude regimes

2.2 *Spacecraft Overview*

A photograph of PSP is shown in Fig. 3. Starting at the top, the thermal protection system (TPS) shields most of the spacecraft and the instruments from the Sun during encounter. Mounted on a truss directly beneath the TPS are the four cooling system primary radiators (CSPR). The primary spacecraft structural bus is beneath the CSPRs. Most of the instruments and spacecraft components are mounted either on the interior or exterior of the primary spacecraft structural bus.

2.2.1 Solar Arrays

The solar array shown in Fig. 3 is in the stowed position for launch. There are two articulated solar arrays. Each includes a boom extension arm, a primary section, and a secondary section. Each solar array has two independent axes of rotation. The main axis of rotation is referred to as the "flap" angle. Flap angle rotation is about the spacecraft X-axis. Each solar array can be positioned between 0°–88° with 0° being fully extended, so the wing is perpendicular to the spacecraft's primary bus. (See Fig. 4) The purpose of the flap angle is to allow the arrays to be moved from full sun when at 0° to full shade behind the TPS.

The other axis of rotation is the "feather" angle. The feather angle is rotation about the solar array attachment axis. Each solar array can feather ± 85°. 0° feather is when the normal of the active array surface is parallel with the spacecraft + Z axis and +90° is in the ram (+X) direction. The feather angle is used to maintain

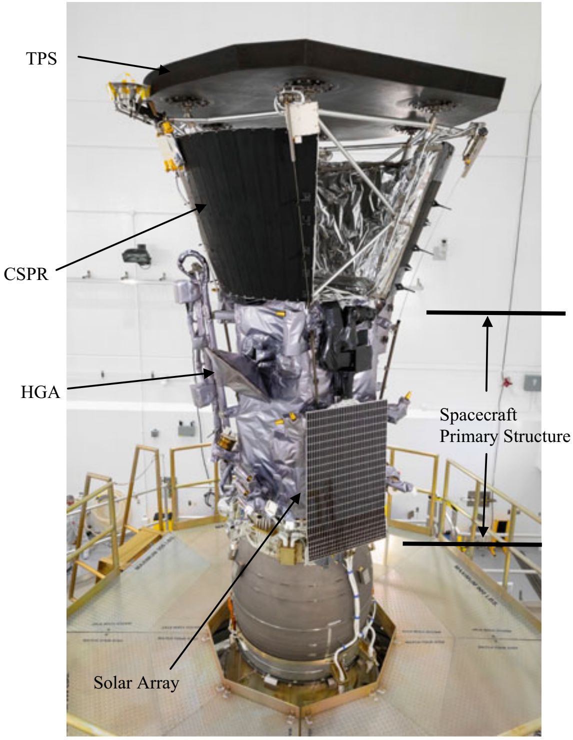


Fig. 3 PSP Installed on the 3rd Stage

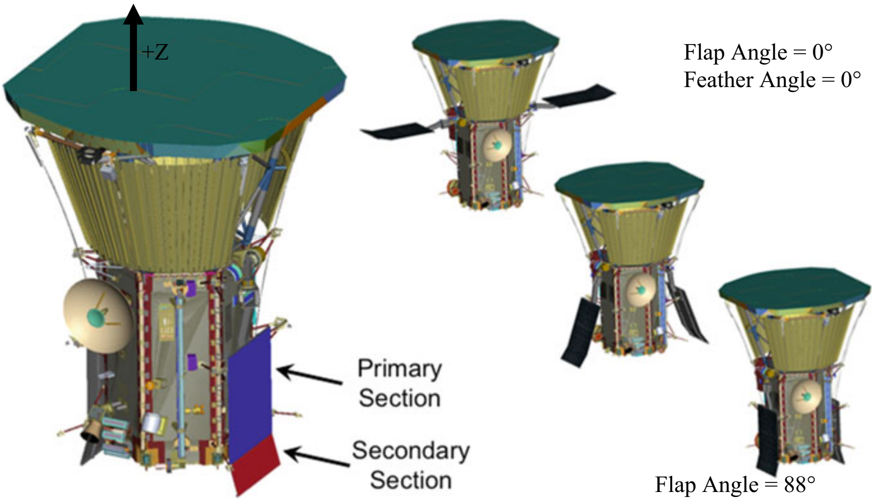


Fig. 4 Solar arrays

sun pointing during spacecraft slews when farther from the sun. The feather angle is restricted to ≤2° if the flap angle is >0.5°.

2.2.2 Solar Array Cooling System

The solar array cooling system (SACS) pumps water through heat exchangers on the back of the two solar arrays and the CSPRs. The SACS has primary and redundant pumps. It uses software control switching between operating at high speed versus low speed based on the temperature of the water.

2.2.3 Telecommunications

The telecommunications subsystem provides the radio frequency (RF) uplink, downlink and radiometric tracking data services with Earth, through the Deep Space Network (DSN). The telecommunications subsystem is a dual-band communications system. Uplink and low data-rate downlink use the X-band deep space allocation, and high data-rate downlink uses the Ka-band deep-space allocation. Besides data downlink, both the X-band and Ka-band downlinks provide 2-way Doppler, ranging, and delta-differential one-way ranging (D-DOR) services. Because of power constraints, only one downlink (X-band or Ka-band) is active during a contact.

The spacecraft has four X-band antennas, two low-gain antennas (LGAs) and two fanbeam antennas, and one high-gain antenna (HGA) used for Ka-band. The X-band antennas are used for uplink and low data-rate downlink. The low downlink rates range from 10 bps to 100,000 bps just after launch. The HGA is used for high data-rate downlink. High rate downlink ranges from 50 kbps–555 kbps. The HGA has one axis of motion and can move ±45° from the –X axis in the XZ plane.

The system can also transmit four beacon tones to indicate spacecraft health. Because of the close proximity to the Sun, beacon tone A is transmitted during DSN contacts in encounter to indicate the spacecraft is functioning nominally in operational mode Level 3. Beacon tones B and C are used to indicate the spacecraft has demoted to Operational Mode Level 2, which tone is transmitted depends on the anomaly that caused the demotion. Beacon tone D is used if the spacecraft has demoted to safe mode–standby. The spacecraft nominal and safe modes are discussed in Sect. 2.2.4.

2.2.4 Flight Software

The spacecraft has three single board computers (SBCs). Two of the SBCs are always powered and run in a primary and hot spare configuration. The third SBC is the backup spare. Currently, it is nominally powered, but may be turned off if required. The flight software (FSW) comprises command and data handling (C&DH), guidance, navigation, and control (G&C), and solar array pointing control (SAOPS). The same set of FSW is used for all SBCs with execution limited for certain processes on the hot and backup spares.

PSP has three operational modes and three safe modes. The operational modes are: Level 1, Level 2 and Level 3. Level 3 is the nominal fully operational mode. Level 2 is partially degraded operations. Full science can be performed at Level 2. However, the spacecraft time-tag command sequence is suspended and the telecommunications system is configured for constant LGA beacon. Operations are degraded further in Level 1. Level 1 is a temporary operational level to charge the battery. In this mode, the instruments are not powered, the time-tagged command sequence is suspended, and the telecommunications system is off.

The spacecraft will automatically transition down from Level 3 to Level 2 if there is less than full G&C control, a time-tagged command or time-tagged macro has an execution failure, or the prime SBC demotes. It will transition from Level 3 directly to Level 1 if the battery state of charge (SOC) is insufficient for the telecommunications and payload operations. Similarly, it will demote from Level 2 to Level 1 if the battery SOC is insufficient for telecommunications and payload operations. Conversely, the spacecraft will automatically promote from Level 1 to Level 2 once the battery SOC is high enough to support telecommunications and payload operations. However, it requires a ground command besides being in full G&C control and having a good battery SOC to transition from Level 2 to Level 3.

The safe modes are: safe mode solar array, safe mode earth acquisition, and safe mode standby. During safe modes the instruments are not powered, mission default pointing is used, and any G&C control mode can be used.

Safe mode solar array is a transient mode that responds to critical power or thermal faults by commanding the solar arrays to a safe flap angle based on current solar distance. It responds to solar array over temperature, solar array cooling system under temperature, umbra violation, solar array excessive flux, aphelion attitude violation, and low battery SOC.

Safe mode earth acquisition demotions are in response to command loss timer (CLT) expiration. The CLT is routinely set by mission operations during contacts. Safe mode earth acquisition rotates through radio and antenna configurations in an attempt to regain contact with Earth.

Safe mode standby occurs because of SBC over cycling. If autonomy detects three SBC rotations have occurred within a short amount of time, it will transition the spacecraft to safe mode standby.

Fault detection, isolation and recovery software, referred to as autonomy, also runs on the SBCs. The same autonomy software is on all three SBCs but it is only active on the prime.

2.3 Instrument Overview

PSP has four instrument suites. Figures 5 and 6 show the locations of the instruments on the spacecraft.

Fig. 5 Ram facing view

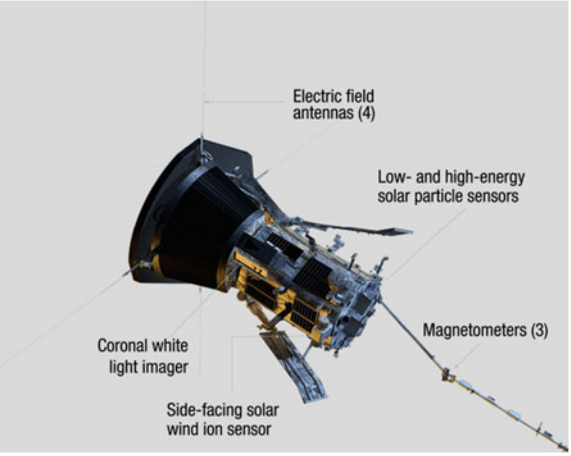
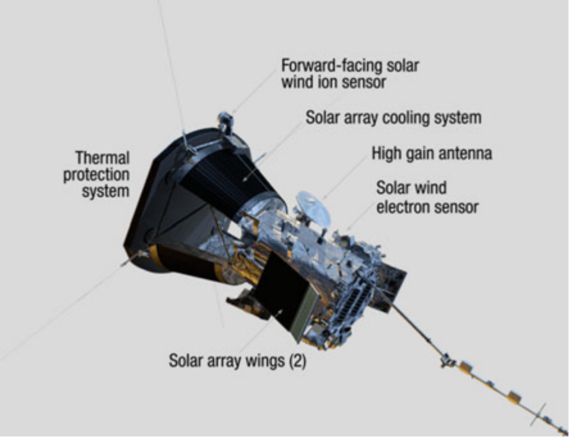

Electric field antennas (4)
Low- and high-energy solar particle sensors
Magnetometers (3)
Coronal white light imager
Side-facing solar wind ion sensor

Fig. 6 Anti-Ram facing view


Forward-facing solar wind ion sensor
Solar array cooling system
Thermal protection system
High gain antenna
Solar wind electron sensor
Solar array wings (2)

2.3.1 Fields Experiment (FIELDS)

The FIELDS experiment directly measures DC/Low frequency electric fields; DC/Low frequency magnetic fields, plasma waveforms, spectra and cross-spectra; spacecraft floating potential; solar and interplanetary radio emissions; and quasi-thermal noise spectrum.

FIELDS has five electric field antennas (V1, V2, V3, V4 and V5) with preamplifier electronics, two fluxgate magnetometers, one tri-axial search-coil magnetometer and a main electronics package. The FIELDS V1-V4 antennas are attached to the truss just below the TPS. The V5 antenna, fluxgate magnetometers, and search-coil magnetometer are on the magnetometer (MAG) boom. (See Figs. 5 and 6).

FIELDS has internal memory where science data is stored for transfer to the spacecraft SSR later. This is performed using a higher rate of transfer than the normal data transfer and must be configured on the spacecraft prior to the transfer beginning.

2.3.2 Integrated Science Investigation of the Sun (IS⊙IS)

IS⊙IS observes energetic electrons, protons and heavy ions that are accelerated to high energies. IS⊙IS comprises two parts—Energetic Particle Instrument—Low Energy (EPI-Lo) and Energetic Particle Instrument—High Energy (EPI-Hi). EPI-Lo has eight wedge sensors, each having ten apertures. EPI-Hi has two low-energy telescopes and one high-energy telescope. IS⊙IS is located on the exterior of the + X side of the spacecraft. (See Fig. 5) EPI-Hi and EPI-Lo are controlled independently and each has their own command and telemetry interface.

2.3.3 Solar Wind Electrons Alphas and Protons (SWEAP)

SWEAP measures the thermal velocity distribution functions of electrons, alphas (fully ionized helium) and protons (ionized hydrogen) in the solar wind. The distribution function is used to determine velocity (speed & direction), density, and temperature of the solar wind. The SWEAP suite comprises the Solar Probe Cup, the Solar Probe Analyzers and the Solar Wind Electronics Module. (See Figs. 5 and 6).

SWEAP has internal memory where science data is stored for transfer to the spacecraft SSR later. This is performed using a higher rate of transfer than the normal data transfer and must be configured on the spacecraft prior to the transfer beginning.

2.3.4 Wide-Field Imager for Solar PRobe (WISPR)

WISPR images the inner coronal plasma by measuring the photospheric light scattered from the free electrons in the plasma. WISPR comprises two parts; the WISPR Instrument Data Processing Unit and the WISPR instrument module. The WISPR instrument module is on the outside of the + X (RAM) side of the spacecraft and comprises an inner and outer telescope. The fields-of-view (FOVs) of the inner and outer telescopes are optimized to avoid scattered and radiated visible light from the FIELDS antennas. The combination of the two FOVs permits the observation of the plasma from about 2.5 Rs (at perihelion) to just beyond the position of the spacecraft. The instrument module is on the RAM side of the spacecraft so it will image the plasma that the spacecraft is going to fly through.

WISPR also has internal memory where images may be stored for transfer to the spacecraft SSR. However, unlike FIELDS and SWEAP, the transfer occurs using its nominal data transfer rate and no spacecraft configuration is required.

2.3.5 Instrument Science Data SSR Allocation

Science data is recorded either directly on the spacecraft SSR or on internal instrument recorders then transferred to the spacecraft SSR. During the planning cycle for each orbit, the science team determines how much memory on the SSR each team will be able to use. This is referred to as their recorder allocation. The allocations are stored in a table on-board the spacecraft and to keep from filling the SSR, the spacecraft flight software will stop recording data for an instrument on the SSR once they reach their allocation.

3 Mission Operations Spacecraft Testing

3.1 Spacecraft Testing Overview

Four formal Mission Simulations were performed as milestones for phases of integration and test (I&T) of the spacecraft. The mission simulations are required to test:

- All primary spacecraft modes and mode transitions
- Launch and separation
- Deployments
- Typical science acquisition sequences followed by spacecraft-to-ground station contact operations or SSR playback, including data processing by the ground processing facilities.
- High-risk maneuvers
- One-time command sequences. Examples of this would be instrument commissioning or special maneuvers.
- Contingency recovery procedures.

All mission operations testing was performed from the Mission Operations Center (MOC) at APL. The same ground system, telemetry displays, and command scripts used for flight were used for testing.

Mission Simulation #1 was a five-day test performed after the integrating the core of the spacecraft subsystems while the spacecraft was at APL. It focused on spacecraft activities scheduled during early operations during commissioning. It included first contact scripts, deployments, G&C commissioning activities, trajectory correction maneuver #1 (TCM-1), telecommunication checkouts, and routine spacecraft operations.

Mission Simulation #2 was a five-day test performed after instrument integration while the spacecraft was at APL. Each instrument suite had a single dedicated day to perform testing. Day five of the test focused on multi-instrument activities.

Mission Simulation #3 (MSIM 3), which is the focus of this paper, was performed during TVAC testing while the spacecraft was at the Goddard Spaceflight Center

(GSFC). It was split into two parts. The first was a “combination orbit” comprising activities from three different orbits. The combination orbit was performed during the thermal balance portion of the TVAC. The second part was the “Orbit-in-the-Life” test, which was performed during the thermal cycle portion of TVAC.

Mission Simulation #4 was a five-day test performed at the launch site. It was used to re-test early operations and critical activities for both the spacecraft and the instruments.

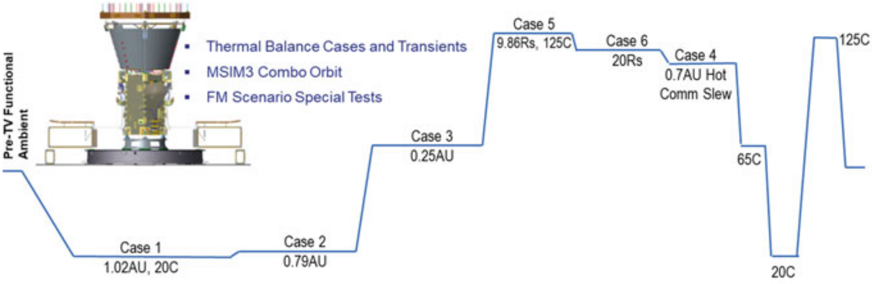
3.2 Thermal Vacuum Testing Overview

PSP TVAC was conducted at GSFC in January, February, and March 2018. It was conducted in two parts–thermal balance and thermal cycle–with a break in between to reconfigure the thermal vacuum chamber and the spacecraft.

3.2.1 Thermal Balance

Thermal balance was a 12-day test conducted from 28 January–8 February, 2018. The purpose of the thermal balance test was to hold the spacecraft at various temperatures expected to occur during flight and perform selected activities at those temperatures. Figure 7 shows the temperature profile.

To optimize test time, the team decided to use mission operations procedures and tests to validate spacecraft requirements. The mission operations Mission Simulation



	Case 1		Case 2		Case 3	Case 5	Case 6	Case 4	FM Scenarios	
	PC3 1.02AU Cold Aphelion No Inst X-band on		PC16 0.79AU Cold Umbra No Inst B Htrs/ A Htrs/ A&B Htrs X-band off		PC22 0.25AU Cold Science Inst on Xband on	PC24 9.86Rs Hot Science TPS Sim Off/On Inst on Xband off	PC23 20Rs Hot Science TPS Sim On Inst on Xband on	PC21 0.7AU Hot Comm Slew Kaband on	5 UnderTemp Encounter 8 UnderTemp Aphelion	6 OverTemp
Day: 1	2	3	4	5	6	7, 8	9	10	11	12

Fig. 7 Thermal balance temperature profile

#3 activities were performed throughout the entire thermal balance test and timed to occur at the appropriate temperatures.

3.2.2 Thermal Cycle

Thermal cycle was a 30-day test in February and March 2018. The mission operations testing during thermal cycle took place over nine consecutive days from 24 February–4 March, 2018. Figure 8 shows the thermal cycle temperature profile. The period for the mission operations Mission Simulation #3 is highlighted in green in Fig. 8.

3.3 Mission Simulation #3 Overview

The overall objective of Mission Simulation #3 was to test early operation and commissioning activities, nominal orbit-in-the-life activities, and selected contingencies. The activities performed include:

<u>**Early Operations & Commissioning**</u>:

- Launch
- Post-separation sequence & first contact
    - The post-separation sequence ran automatically on the spacecraft when separation from the launch vehicle was detected. It detumbled the spacecraft to gain

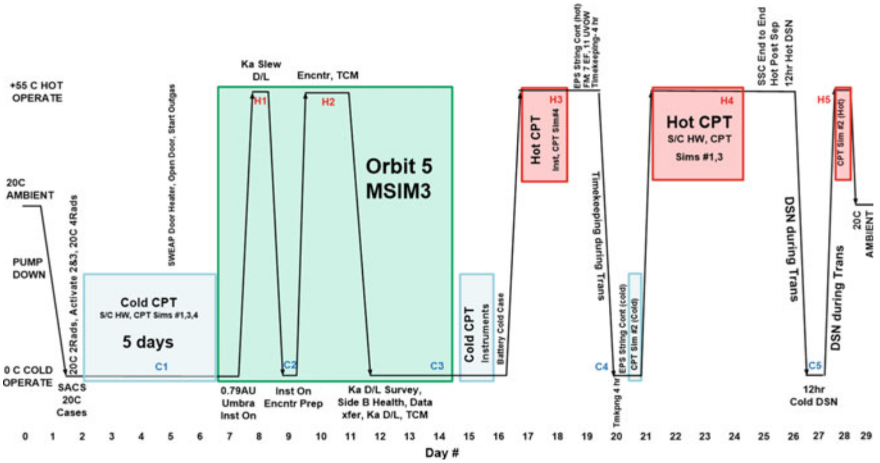


Fig. 8 Thermal Cycling Temperature Profile

  attitude control, activated the telecommunication system and began transmitting telemetry, deployed the solar arrays and activated the SACS system and began flowing water through CSPRs 1&4.

- HGA deployment
- FIELDS E-Field antennas clamshell deployment
  - This released the 4 electric field antennas from the mechanisms that held them in place for launch.
- TCM-1 (Launch correction maneuver)
- FIELDS E-Field Antennas hinge deployments
  - This deployed the 4 electric field antennas from the launch position near the spacecraft body to the flight positions beyond the TPS shown in Figs. 5 and 6.
- SWEAP SPAN A door heater on & door deployment
- SWEAP transient slew
  - The SWEAP transient slew moved the spacecraft off of the normal spacecraft pointing to allow SWEAP to perform calibrations. It had to be performed at > 0.935 AU for thermal reasons.
- SACS Radiators 2 & 3 activation
  - This flowed water through CSPRs 2&3.
- WISPR door open

Nominal Orbit Activities:

- A Venus flyby with an eclipse
- Attitude slew for aphelion to umbra pointing
- Attitude slew for umbra to aphelion pointing
- Power on instruments
- Prepare for an encounter & enter encounter
- Perform the encounter science
- Perform a nominal beacon contact during an encounter
  - Because of being so close to the Sun, during encounter, the spacecraft transmits beacon tone A to show spacecraft health instead of telemetry.
- Exit an encounter
- Perform nominal X-band contacts in aphelion, umbra, and variable pointing at various downlink & uplink rates
- Perform nominal Ka-band contacts at various downlink rates
- Perform Ka-band rate stepping
- Nominal TCMs (that are not TCM-1)
- Uplink nominal spacecraft command loads
- Have instruments perform real-time and stored file commanding
- Downlink instrument survey data

  - The instrument teams use the survey data to select what science data to downlink during orbits when there is not enough Ka-Band Science downlink to downlink all the data.

- Perform FIELDS and SWEAP high-speed data transfers
- Downlink science
- Perform a redundant avionics side health check.
  - This powers on and performs a communications check with redundant avionics.

Contingencies:

- Operational mode Level 2 beacon contact during encounter
  - When the spacecraft demotes to Operational mode Level 2, the time-tag command load is suspended, so it would not automatically reconfigure the spacecraft for the scheduled beacon contact. So, mission operations autonomy was developed to replace the time tags to reconfigure the spacecraft for scheduled beacon contacts and transmit beacon tone B or C that indicated the spacecraft had demoted to operational mode Level 2. The times of the scheduled beacon contacts are stored on-board the spacecraft as variables (referred to as storage variables) prior to each encounter.
- Perform an encounter exit in operational mode level 2
- Recover to operational mode level 3 from operational mode level 2
- Perform a negative initial acquisition
- Recover from safe mode earth acquisition to operational mode level 3
- Recover from safe mode standby to operational mode level 3
- Perform an SBC rotation to cause a change in attitude

4 Orbit Selection

The goal in selecting activities for the orbit-in-the-life test was to cover all the nominal spacecraft operations and all the nominal instrument operations that involved coordination with the spacecraft and other instrument teams. The idea was to perform the activities in order around a single orbit, beginning at aphelion and ending at the following aphelion. We started by compiling a list of all the activities required to satisfy both spacecraft requirement testing and mission operations testing. Then the activities on the list were mapped to various orbits, creating a matrix of which activities occurred in each orbit.

Orbit 5 was chosen for the orbit-in-the life test. It was chosen because it contained all the activities and attitude regimes desired to be tested except operations at 9.86 Rs, which was already planned to be tested during the combination orbit during thermal balance.

In addition, Orbits 21 and 22 were selected for the non-early operations/Orbit 1 activities for the combination orbit during thermal balance. This was because of spacecraft thermal testing requirements to simulate the 9.86 Rs region and the thermal testing for the Venus Flyby with an eclipse.

5 Test Preparation and Coordination

Once the orbits were selected, the actual test planning began. The goal of the mission simulation was to execute the planning, building, testing, and executing the activities as realistically as possible.

5.1 Test Blocks

The first step in the process was to obtain representative DSN contact schedules for the orbits selected. Initial DSN contact requirements were delivered to the DSN scheduler defining the number, duration and types of DSN contacts desired during the phases of the orbit or for special events. Once the representative schedules for Orbits 5, 21, and 22 were received from the DSN scheduler, specific DSN contacts were selected to be performed during the tests. The DSN contacts were chosen to provide a mixture of downlink and uplink rates using both the Ka-band and X-band RF systems during all the attitude regimes. Beacon contacts during the solar encounter were also chosen. Table 1 shows the DSN contacts chosen for Orbit 5 with their uplink and downlink rates.

Once the DSN contacts were chosen, the contacts and other test activities were divided into test blocks that covered a continuous period and could be executed individually and out of order if required. The blocks were named based on what type of activities they contained. E blocks contained early operations commissioning activities and were performed during thermal balance. O blocks contained orbit-in-the-life activities in Orbit 5 and were performed during thermal cycling. G blocks contained generic activities from Orbit 21 & 22 and were performed during thermal balance. A blocks contained planned anomaly response and recovery tests and were performed during thermal balance. Tables 2 and 3 summarize the orbit, spacecraft solar distance, downlink rate, and uplink rate for each test block (Table 4).

Table 5 in Sect. 8.1 contains a detailed description of each test block, any planned deviations from how the activity would be performed in flight, and the test results.

Figure 9 shows a graphic of the O test blocks overlaid on Orbit 5 and if the tests were during hot or cold portions of the thermal cycle. Blue rectangles are DSN contacts, the ones with blue borders are Ka-band contacts. Yellow rectangles are instrument activities. Green rectangles are solar distance-based activities that would be performed out of contact. Purple triangles are TCMs. Pink stars are ephemeris loads. Activities with red Xs are activities in Orbit 5 that were not performed during the orbit-in-the-life test.

Table 1 Orbit 5 contacts

Activity	BOT	DSN Station	Duration	Uplink rate, bps	Downlink rate, bps
X-band Aphelion Contact	2020 108 11:05:00	DSS 54	4:05	31.25	10
X-band Aphelion Variable Contact	2020 115 11:10:00	DSS 54	4:05	31.25	10
Ka-Band Track	2020 121 12:00:00	DSS 55	4:20	31.25	52,083.33
X-band Encounter Prep	2020 147 12:00:00	DSS 55/DSS 26	8:05	31.25	10
X-Band Encounter Beacon	2020 158 16:00:00	DSS 25	4:00	NA	Beacon
X-Band Post Encounter	2020 166 14:15:00	DSS 24	3:30	500	160
X-Band TCM 13 Prep	2020 167 06:25:00	DSS 54	3:40	500	160
X-Band TCM 13	2020 170 16:00:00	DSS24	7:40	500	320
Ka-Band Downlink Science Survey Data	2020 181 16:00:00	DSS 25	4:00	500	555,555
Ka-Band Side B Health Check	2020 182 12:00:00	DSS 25	10:00	500	263,157
X-Band TCM 14 /Ka-Band Post TCM 14	2020 183 16:00:00	DSS 25/DSS 34	8:30	500	320 (X) 500,000 (Ka)
Ka Band Science Downlink	2020 191 12:00:00	DSS 25	10:00	500	333,333
X-Band CMDMD	2020 193 04:25:00	DSS 54	7:25	2000	320

5.2 *Test Guidance and Control (G&C) Simulation Setup*

During TVAC, the spacecraft was connected to a simulator to provide feedback to the spacecraft systems. A scenario setup had to be created for each test block that simulated correct beginning spacecraft attitude, solar distance, and spacecraft time.

Table 2 Thermal balance combination orbit test block summary

Block	Name	Orbit	Solar distance (AU)	Downlink rate	Uplink rate
E1	Post Separation & First Contact	1	1.017↓	X-band: 100 kbps	2 kbps
E2	TCM 1	1	1.017↓	X-band: 10 kbps	2 kbps
E3	Instrument Commissioning Deployments	1	0.946↓	X-band: 10 kbps	2 kbps
E4	SACS Radiators 2 & 3 Activation	1	0.89↓	X-band: 100 kbps	2 kbps
G0	Slew from Aphelion to Umbra Pointing at 0.79 AU	1	0.79↓	X-band: 100 kbps	2 kbps
G1	Venus Flyby with Eclipse	21	0.728↓	X-band: 100 kbps	2 kbps
G2-1	Pre-Encounter Prep	22	0.3↓	X-band: 100 kbps	2 kbps
G2-2	Encounter Entry	22	0.25↓	X-band: 100 kbps	2 kbps
G3	Encounter Exit	22	0.703↑	Ka-band: 30,120 bps Ka-band: 37,313 bps Ka-band: 555,555 bps	31.25 bps
A1	Operational Mode Level 2 Beacon Contact	22	0.2↓	X-band: Beacon	7.8125 bps
A2	Encounter Exit in Operational Mode Level 2	22	0.25↑	X-band: 100 kbps	2 kbps

↓Spacecraft moving towards the Sun ↑Spacecraft moving away from the Sun

The spacecraft solar arrays and the HGA could move during the orbit-in-the-life test. As part of the setup for the test blocks, the HGA and solar arrays were pre-positioned to the proper starting point.

6 Orbit-in-the-Life Orbit Planning

An accelerated version of the orbit planning process was performed for Orbit 5 as part of the orbit-in-the-life test. The actual orbit planning process begins six months prior to the initial aphelion of the orbit to allow time for the science team to coordinate activities amongst themselves and any external organizations. Since no actual science was being collected during TVAC testing, coordination between the PSP science teams themselves, and between the PSP science teams and external teams was not needed so the schedule could be compressed. The nominal schedule also includes

Table 3 Thermal cycle orbit-in-the-life (orbit 5) test block summary

Block	Name	Solar distance (AU)	Downlink rate	Uplink rate
O1	Aphelion X-band Contact	0.82↓	10 bps	31.25 bps
O2	Aphelion Variable X-band Contact	0.78↓	10 bps	31.25 bps
O3	Ka-band Contact	0.72↓	52 kbps	31.25 bps
O4	Incoming Orbital Events	0.7↓ & 0.53↓	100 kbps	2 kbps
O5	Pre-Encounter Prep	0.29↓	10 bps	31.25 bps
O6 pt 1	Encounter Entry & 5 h of Encounter Science	0.25↓	100 kbps	2 kbps
O6 pt 2	Nominal Beacon Contact	0.19↓	Beacon	7.8125 bps
O6 pt 3	Encounter Exit	0.25↑	100 kbps	2 kbps
O7	Post Encounter X-Band Contact	0.39↑	160 bps	500 bps
O7a	TCM 13 Prep	0.39↑	160 bps	500 bps
O8	TCM 13	0.48↑	320 bps	500 bps
O10	Ka-band Contact Science Survey Data Downlink	0.647↑	555,555 bps	500 bps
O11	Redundant Side Health Check	0.66↑	320 bps	500 bps
O12	TCM 14	0.672↑	X-band: 320 bps Ka-band: 500,000 bps	500 bps
O13a	SWEAP High Speed Data Transfer	0.75↑	X-band: 100 kbps	2 kbps
O13b	FIELDS High Speed Data Transfer	o.75↑	X-band: 100 kbps	2 kbps
O14	Ka-band Contact Science Data Downlink	0.75↑	Ka-band: 333,333 bps Ka-band: 555,555 bps	500 bps
O15	Commanded Momentum Dump	0.7699↑	X-band: 320	2 kbps
O16	0.79 AU slew from Umbra to Aphelion	0.79↑	X:band: 100 kbps	2 kbps

↓Spacecraft moving towards the Sun ↑Spacecraft moving away from the Sun

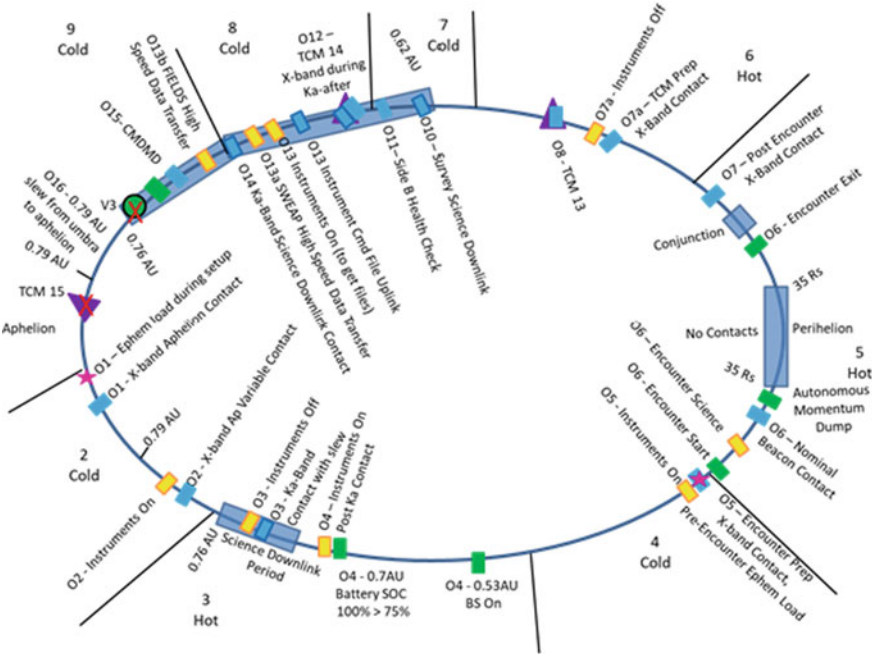


Fig. 9 Orbit 5 graphical plan

time for several iterations of the planning process to be performed. For Mission Simulation #3 preparation, only one loop through the process was performed.

The accelerated orbit planning process started three months prior to Mission Simulation #3. This exercised the tools used by both the mission operations and science operations teams. All data products for the process were created. At the end of the process the spacecraft and instrument command files, time tag command loads, and command timelines to be used during the mission simulation were created.

7 Pre-Test Verification

All test blocks were tested using either or both a hardware-in-the-loop simulator or with the spacecraft prior to the mission simulation. The tests were used to verify the time-tag command loads, command scripts, the G&C simulation setup scenarios, and to practice the timing of performing some activities. If a verification test failed, analysis was performed to determine the reason for the failure. If possible, the issue was corrected, and the block was re-tested. In one case, since the test procedure could not be corrected prior to the mission simulation, the test block was removed from the orbit-in-the-life test and performed during a standalone test later.

Positioning the HGA and solar arrays during the orbit-in-the-life test was also practiced on the spacecraft. This was a time-consuming process that involved manually stepping the solar array flaps, solar array feathers, and the HGA to the starting point. This could take up to an hour and a half, depending on how far the appendages had to move.

It was discovered during the preparation testing, that the EPI-HI instrument could not ride through an abrupt time jump created by moving from one test block to the next and would have to be powered off. Because of the length of time the instrument would be off, its survival heaters had to be powered on. It was also required to disable certain autonomy rules during the transition to prevent undesired rule firing. A transition sequence to perform between test blocks was developed. The time jump sequence between test blocks was practiced on the spacecraft twice prior to execution of the actual mission simulation during TVAC.

8 Test Execution and Results

During Mission Simulation #3, the spacecraft was in the thermal vacuum chamber at GSFC. The I&T team was also located at GSFC. The mission operations team was located in the MOC at APL. The instrument teams were located in a Science Operations Center (SOC) also at APL. During execution of Mission Simulation #3, the spacecraft command and telemetry link was transferred from the I&T team at GSFC to the Mission Operations team in the MOC. Testing was performed during two shifts, from approximately 7:00–23:00 each day. Overnight, the command and telemetry link were transferred back to the I&T team. The spacecraft remained continuously powered for the entire duration of the thermal balance and thermal cycle tests. The overnight configuration of the spacecraft varied depending upon the test configuration requirements. During thermal balance, the spacecraft had to remain in the required configuration for the thermal case being tested. During thermal cycle, the telecommunications system was reconfigured to playback the data recorded on the SSR during the day. Control of the spacecraft was also transferred back to the I&T team when transitioning to a new test block while the new G&C simulation was setup and positioning of the solar arrays or HGA was performed.

Since one of the goals of the test was to perform the activities in an as flight like way as possible, during the orbit-in-the-life test, time tags were only loaded during simulated DSN contacts. In addition, most activities being performed outside of a DSN contact were performed with only limited safety monitoring by the I&T team. The mission operations team and science operations team could not monitor telemetry during these out-of-contact periods mimicking what would be seen inflight.

One standard deviation from flight was that no thrusters were actually fired during testing. The thrusters were safed prior to the tests.

The test block execution sequence is shown in Table 4. It summarizes who on the team performed each step and where they were located. Steps 1–5 were repeated for each test block, then step 6 was performed at the end of testing each day.

Table 4 Test block execution sequence

Step Number	Step	Performed By	Location
1	Setup Spacecraft & Simulation	I&T Team and G&C Team	GSFC
2	Move HGA and Solar Arrays to their starting positions (Orbit-in-the-life-test only)	G&C Team	GSFC
3	Hand spacecraft control from I&T to Mops	I&T Team and Mops Team	GSFC and APL
4	Execute Test Block	Mops Team & Instrument Teams	APL
5	Hand spacecraft control back to I&T	I&T Team and Mops Team	GFSC and APL
6	Place spacecraft in a safe configuration for overnight	G&C Team	GSFC

8.1 Test Block Results

Table 5 describes the purpose, planned deviations from flight, and the outcome for each test block.

Table 5 Test block results

Thermal Balance Combination Orbit Blocks	
Test Block:	**E1: Post Separation & First Contact**
Purpose:	Perform a flight like post separation sequence, MOPS first contact activities, and mission day 2 deployments. The mission day 2 deployments included HGA launch lock release, MAG boom hinge releases, and FIELDS E-Field antennas 1–4 clamshell releases
Planned Deviations:	The magnetometer boom was not deployed due to hardware safety concerns while in the thermal vacuum chamber
Result:	The first contact and deployments were performed as expected. There were no anomalies
Test Block:	**E2: TCM-1**
Purpose:	Execute TCM-1. TCM-1 was the launch correction maneuver and was planned to be performed 7 days after launch
Planned Deviations:	No thrusters were fired
Result:	The G&C software aborted the maneuver at 38% complete. An investigation determined the maneuver aborted due to an issue with the flight software. A new version of flight software was loaded to the spacecraft during the break between the thermal balance and thermal cycling portions of the TVAC
Test Block:	**E3: Instrument Commissioning Deployments**

(continued)

Table 5 (continued)

Thermal Balance Combination Orbit Blocks	
Test Block:	**E1: Post Separation & First Contact**
Purpose:	Perform instrument commissioning deployments including the FIELDS E-Field antennas 1–4 hinge deployment, SWEAP SPAN A door open, and WISPR door open
Planned Deviations:	The WISPR door open was not performed. Even though the activities were scheduled for mission day 30, the G&C simulation was setup for mission day 34. This was planned so only one G&C simulation setup would need to be performed
Result:	The deployments were performed as expected
Test Block:	**E4: SACS Radiators 2 & 3 Activation**
Purpose:	Release water into SACS radiators 2 & 3. During the post separation sequence, water was only released into radiators 1 & 4. Once the spacecraft reached 0.9AU, the water had to be released into the other two radiators. This activity was performed using mission operations autonomy rules and macros
Planned Deviations:	The activation was performed at 100 kbps downlink / 2 kbps uplink instead of the expected flight rates of 10 bps downlink / 7.8125 bps uplink. This was done so the activation could be monitored in real-time and aborted if required
Result:	The activation was performed nominally
Test Block:	**G0: Spacecraft Slew from Aphelion to Umbra Pointing at 0.79 AU**
Purpose:	Practice performing a spacecraft slew from the aphelion to umbra pointing regimes at 0.79AU
Planned Deviations:	It was performed at 100 kbps downlink / 2 kbps uplink instead of out of contact. This was done so the activity could be monitored in real-time during the test
Result:	The slew was performed nominally
Test Block:	**G1: Venus Flyby with Eclipse**
Purpose:	Test the pre and post Venus eclipse G&C software and MOPs sequences for eclipse. The thermal conditions and G&C simulation for this test was setup for the Venus flyby in Orbit 21. However the mission operations command sequence for the activities around the eclipse were timed for the longer 11 min duration of the eclipse in Orbit 5
Planned Deviations:	It was performed at 100 kbps downlink / 2 kbps uplink instead of the nominal lower rates. This was done so the activity could be monitored in real-time during the test
Result:	The Loss of Cooling autonomy rule fired post eclipse and switched the SACS system from Pump A to Pump B. During the simulation of the eclipse, the water temperature cooled and the pump speed changed from high speed to low speed. During the speed change, autonomy received the lowered dP values before it received the lowered measured speed values, so the dP thresholds used were for high speed and dP and this triggered the rule. The autonomy rule was updated to allow a 3-s grace period during a speed change before triggering the pump swap

(continued)

Table 5 (continued)

Thermal Balance Combination Orbit Blocks	
Test Block:	**E1: Post Separation & First Contact**
Test Block:	**G2-1: Pre-Encounter Preparation - Orbit 22**
Purpose:	Perform all activities to prepare for solar encounter including: powering on all instruments, load the encounter time-tag sequence, load the storage variables for faulted beacon contacts during encounter, set the command loss timer, and individual instrument pre-encounter commanding
Planned Deviations:	It was performed at 100 kbps downlink / 2 kbps uplink instead of the nominal lower rates. This was done so the activity could be monitored in real-time during the test. The time-tags were only loaded for encounter entry and not all of encounter and encounter exit. This was done because of other testing being performed between the encounter entry preparation and the rest of the thermal balance encounter test blocks
Result:	All the activities were completed nominally. However, after the EPI-HI instrument was powered on, it was noticed they were filling their record allocation on the SSR very quickly. The team was notified and they did real-time commanding to lower their data production rates
Test Block:	**G2-2: Encounter Entry—Orbit 22**
Purpose:	Perform the nominal encounter entry time tag sequence and verify nominal autonomy rules fire as expected
Planned Deviations:	It was performed at 100 kbps downlink / 2 kbps uplink instead of the nominal out of contact. This was done so the activity could be monitored in real-time during the test
Result:	All the activities were completed nominally
Test Block:	**A1: Operational Mode Level 2 Beacon Contact**
Purpose:	Test the MOPS autonomy rules and macros that perform beacon contacts during encounter when the spacecraft is in operational mode level 2. The normal operating mode of the spacecraft is operational mode level 3. When the spacecraft transitions to operational mode level 2 the time tag sequence is suspended. Normally the time tag sequence would configure the RF system for the beacon contact
Planned Deviations:	The demotion to operational mode level 2 was triggered deliberately
Result:	The MOPS autonomy rule triggered as expected and configured the spacecraft for a beacon contact at the proper time. However, beacon tone A was transmitted instead of beacon tone B. The cause of the unexpected beacon tone B was traced to a MOPS macro missing a command to reassert the beacon tone after reconfiguring the radio. The macro was updated
Test Block:	**A2: Encounter Exit—Orbit 22**
Purpose:	Exit encounter in operational mode level 2 then recover back to operational mode level 3 to test the autonomy response at encounter exit and the mission operations procedure to recover from operational mode level 2 to level 3
Planned Deviations:	It was performed at 100 kbps downlink/2 kbps uplink instead of out of contact. This was done so the activity could be monitored in real-time during the test

(continued)

Table 5 (continued)

Thermal Balance Combination Orbit Blocks	
Test Block:	**E1: Post Separation & First Contact**
Result:	The autonomy rules fired as expect at encounter exit, which powered off the instruments and reconfigured the RF system to transmit beacon tone B. However, during the recovery to operational mode Level 3, the spacecraft performed a slew to the last commanded attitude. It was determined that an attitude command to the current attitude, needed to be sent prior to sending the promotion command, so the last commanded attitude matches the current attitude and the spacecraft will not slew. The mission operations recovery procedure was updated
Test Block:	**G3: Ka-band Downlink—Orbit 22**
Purpose:	Perform a Ka-band DSN contact and playback the data on the SSR
Planned Deviations:	The contact continued overnight and was terminated via MOPS real-time command instead of time tags
Result:	The spacecraft slewed to the Ka-band downlink attitudes, RF setup and playback of the SSR data was completed nominally
Thermal Cycle Orbit-in-the-Life Blocks	
Test Block:	**O1: Aphelion X-band Contact**
Purpose:	Perform a nominal X-band contact in the aphelion attitude regime. Sun Distance: 0.82 AU, Uplink Rate: 31.25 bps, Downlink Rate: 10 bps, Instruments Off
Planned Deviations:	None
Result:	The time-tag sequence configured the spacecraft as expected for the contact. The contact plan was performed nominally. However, SWEAP uplinked a relatively large command file during block O1 which resulted in some other instrument files not being uplinked during the allotted DSN contact time. Basically, the simulated contact ended before all instrument command files were transmitted. Due to this issue, a new data product was added that listed the uplink time and expected uplink data volume for each DSN contact. In addition, instrument commanding is now scheduled in the timeline, so the instrument teams are aware of when they are commanding and do not conflict with each other
Test Block:	**O2: Aphelion Variable X-band Contact**
Purpose:	Power on the Instruments via time tags, perform a nominal X-band contact during the aphelion variable solar distance range. Sun Distance: 0.78 AU, Uplink Rate: 31.25 bps, Downlink Rate: 10 bps, Instruments On
Planned Deviations:	None
Result:	The time-tag sequence powered on the instruments and configured the spacecraft as expected for the contact. The contact plan was performed nominally
Test Block:	**O3: Ka-band Contact**
Purpose:	Power off the instruments via time-tags, perform a nominal Ka-band contact, and power the instruments back on via time-tags. Sun Distance: 0.72 AU, Uplink Rate: 31.25 bps, Downlink Rate: 52 K bps, Instruments Off

(continued)

Table 5 (continued)

Thermal Balance Combination Orbit Blocks	
Test Block:	**E1: Post Separation & First Contact**
Planned Deviations:	None
Result:	The time tag sequence powered off the instruments and configured the spacecraft as expected for the contact. However, commanding during the contact while SSR playback was enabled was extremely slow. It was determined that when there is a large difference in uplink and downlink rates, the timers for the file handshaking should be paused so that the transactions do not block/slow down commanding. A new flight constraint was added and the contact setup script that set the timers was updated to pause them in this case. In addition, when WISPR was powered on after the contact, the WISPR Stale Telemetry autonomy rule fired and powered WISPR back off. It was determined the SpaceWire link to WISPR must be disabled around WIPSR Power On and Off. The mission operations procedures and macros used to power WISPR on and off were updated
Test Block:	**O4: Incoming (decreasing spacecraft solar distance) Orbital Events**
Purpose:	Slew from the aphelion attitude regime to the umbra attitude regime at 0.7AU. Then power on the backup spare processor at 0.53 AU
Planned Deviations:	On orbit, the battery SOC will normally discharge from 100 to 75% automatically over a few days prior to crossing 0.7 AU. During the test, the discharge was commanded. The activities were performed at 100 kbps downlink and 2 kbps uplink rates instead of being out of contact
Result:	Both activities were performed nominally
Test Block:	**O5: Pre-Encounter Preparation**
Purpose:	Perform activities to prepare for encounter including: an X-band contact, loading encounter time-tags, loading the operational mode level 2 storage variables, setting the CLT, loading the ephemeris, dumping the data summary table, and pre-encounter instrument commanding. Sun Distance: 0.29 AU, Uplink Rate: 31.25 bps, Downlink Rate: 10 bps, Instruments On
Planned Deviations:	On orbit, the activities are performed over multiple contacts
Result:	All activities were completed nominally
Test Block:	**O6 Part 1: Encounter Entry and 5 h of Encounter Science—Orbit 5**
Purpose:	Perform the encounter entry time-tag sequence including: power on the redundant star tracker, power on the group A & B catbed heaters, load the G&C red dump parameters, slew to the encounter attitude, open the instrument encounter attitude file, power off the SWEAP SPC survival heater, load the instrument encounter data recording changes, and reassert both sun sensor electronics on. Then the instruments performed five hours of science data collection. Also trigger the PSE SCE Sweep algorithm
Planned Deviations:	The activities were performed at 100 kbps downlink and 2 kbps uplink instead of the nominal out of contact

(continued)

Table 5 (continued)

Thermal Balance Combination Orbit Blocks	
Test Block:	**E1: Post Separation & First Contact**
Result:	All activities were performed nominally. FIELDS performed some real time commanding due to an error in their uplinked command file
Test Block:	**O6 Part 2: Nominal Beacon Contact—Orbit 5**
Purpose:	Perform a nominal beacon contact. Sun Distance: 0.19 AU, Instruments On
Planned Deviations:	On orbit, a beacon contact is 4 h with either 1 h of a beacon signal and 3 h of carrier wave or 4 h of beacon. During the test 1 h of beacon and 1 h of carrier wave was performed
Result:	The beacon contact performed as expected and confirmed beacon tone A and carrier wave at correct times
Test Block:	**O6 Part 3: Nominal Encounter Exit—Orbit 5**
Purpose:	Perform a nominal encounter exit time-tag sequence including: Power off the redundant star tracker, power off the catbed heaters, load the instrument cruise data recording changes, power on the SWEAP SPC survival heater, load the G&C red dump parameters, close the instrument encounter attitude file
Planned Deviations:	The activities were performed at 100 kbps downlink and 2 kbps uplink instead of the nominal out of contact
Result:	All activities performed nominally
Test Block:	**O7: Post Encounter X-Band Contact—Orbit 5**
Purpose:	Perform a nominal X-band contact shortly after encounter exit. Sun Distance: 0.39 AU, Uplink Rate: 500 bps, Downlink Rate: 160 bps Instruments On
Planned Deviations:	None
Result:	The time-tag sequence configured the spacecraft as expected for the contact. The contact plan was performed nominally
Test Block:	**O7a: TCM 13 Preparation**
Purpose:	Perform an X-band contact to prepare for TCM-13. Sun Distance: 0.39 AU, Uplink Rate: 500 bps, Downlink Rate: 160 bps, Instruments On
Planned Deviations:	None
Result:	The time-tag sequence configured the spacecraft as expected for the contact. The contact plan was performed nominally
Test Block:	**O8: TCM 13**
Purpose:	Execute TCM-13 during an X-band contact. Sun Distance: 0.48 AU, Uplink Rate: 500 bps, Downlink Rate: 320 bps, Instruments Off
Planned Deviations:	No thrusters were fired
Result:	All the instruments were powered off nominally via time-tags. The RF transition to/from carrier wave was completed successfully. The slews to/from burn attitude completed successfully. The maneuver completed successfully. This maneuver was the first maneuver performed with the software updated made after the failure of TCM-1 in Test Block E2
Test Block:	**O10: Ka-band contact for Science Survey Data Downlink**

(continued)

Table 5 (continued)

Thermal Balance Combination Orbit Blocks	
Test Block:	**E1: Post Separation & First Contact**
Purpose:	Perform a nominal Ka-band contact to downlink the science survey data. Sun Distance: 0.647 AU, Uplink Rate: 500 bps, Downlink Rate: 555,555 bps, Instruments Off
Planned Deviations:	None
Result:	The time-tag sequence configured the spacecraft as expected for the contact. However, due to lower than expected science data recorded to the SSR and overnight playback, playback completed quickly. Therefore, all files on the SSR were downlinked not just the science survey data
Test Block:	**O11: Redundant Side Health Check**
Purpose:	Perform a redundant side health check. This check powers on the backup avionics and performs a communications check
Planned Deviations:	None
Result:	This block was not performed during Mission Simulation #3 because of issues found with testing the procedure on the hardware-in-the-loop simulator. It was performed later after TVAC nominally
Test Block:	**O12: TCM-14**
Purpose:	Execute TCM-14 during an X-band contact followed by a Ka-band contact to downlink data. Sun Distance: 0.672 AU, X-band: Uplink Rate: 500 bps, Downlink Rate X-band: 320 bps, Ka-band: 500,000 bps, Instruments Off
Planned Deviations:	None
Result:	The time-tag RF transition to/from carrier wave was completed successfully. The slew to/from burn attitude completed successfully. The maneuver completed successfully. The RF transition to Ka-Band completed successfully
Test Block:	**O13a: SWEAP High-Speed Data Transfer**
Purpose:	Perform a SWEAP high-speed data transfer
Planned Deviations:	The activities were performed at 100 kbps downlink and 2 kbps uplink instead of the nominal out of contact
Result:	The SWEAP power on, high-speed data transfer start, high-speed data transfer stop, and power off time-tags all completed nominally. However, many FSW event error messages were seen during the high-speed data transfer. These error messages were an indicator that not all of the desired SWEAP science data was being moved to the spacecraft SSR during the high-speed transfer activity. The issue was traced to a bug in the SWEAP FSW, which was corrected. The high-speed data transfer was retested during Mission Simulation #4
Test Block:	**O13b: FIELDS High Speed Data Transfer**
Purpose:	Perform a FIELDS high-speed data transfer, then power all of the other instruments back on
Planned Deviations:	The activities were performed at 100 kbps downlink and 2 kbps uplink instead of the nominal out of contact

(continued)

Table 5 (continued)

Thermal Balance Combination Orbit Blocks	
Test Block:	**E1: Post Separation & First Contact**
Result:	The FIELDS power on, high-speed data transfer start, and high-speed data transfer stop all completed nominally. The other instrument power on time tags completed nominally. However, SWEAP requested not to be powered on due to chamber pressure issues. Their time tags were cleared prior to execution
Test Block:	**O14: Ka-band Contact for Science data downlink**
Purpose:	Perform a nominal Ka-band contact to downlink the science data collected and perform a Ka-band rate step during the contact. Sun Distance: 0.75 AU, Uplink Rate: 500 bps, Downlink Rate: 333,333 bps rate step to 555,555 bps, Instruments Off
Planned Deviations:	None
Result:	The time-tag sequence configured the spacecraft as expected for the contact. The rate step to 555,555 bps time-tags completed nominally. Due to lower than expected science data recorded to the SSR, playback completed very quickly
Test Block:	**O15: Commanded Momentum Dump**
Purpose:	Execute a commanded momentum dump during an X-band contact. Sun distance: 0.7699 AU, Uplink Rate: 2000 bps, Downlink Rate: 320 bps, Instruments On (except SWEAP)
Planned Deviations:	None
Result:	The commanded momentum dump was completed successfully
Test Block:	**O16: Slew from Umbra to Aphelion at 0.79 Au**
Purpose:	Slew from the umbra attitude regime to the aphelion attitude regime at 0.79 AU. Then power off the instruments
Planned Deviations:	The activities were performed at 100 kbps downlink and 2 kbps uplink instead of the nominal out of contact
Result:	The slew from Umbra to Aphelion completed nominally. However, the Detect WIPSR Stale Aliveness autonomy rule fired when WISPR was powered off. It was determined when the updates were made to add the SpaceWire commands due to the issue during Block O3, the commands were added in the wrong place. The WISPR power off procedures were updated

9 Lessons Learned

Mission Simulation #3 went extremely well. However, there were several lessons learned that could be beneficial to other missions.

First, coordinate and communicate more than you think is necessary and then do more. Mission Simulation #3 was highly choreographed because the mission operations tests were being used to verify spacecraft requirements, the nature of the test environment, and the nature of the spacecraft and mission. Coordination had to be done between the Mission Operations team and the I&T team, the Mission Operations team and the spacecraft engineering team, the I&T team and the spacecraft engineering team, the instrument teams and Mission Operations team, and among the instrument teams. Even with many pre-test planning meetings, detailed test procedures, and well-tested scripts, miscommunications still happened. For example, early during the orbit-in-the-life test, an incident occurred that caused all the time-tag commands loaded to the spacecraft to execute. During the orbit-in-the-life test, the HGA and solar arrays were put in a safe position by the G&C team at the end of testing each day as no active testing was being performed overnight and a new test block would be started in the morning. Part of each test block during the orbit-in-the-life test was to load the on-board time-tag sequences for the next test block. During the first overnight, the Mission Operations team received a phone call from the I&T team saying that all the loaded time-tags had expired and fired at once. It was quickly determined that the time tags had expired while the G&C team was running their script to park the appendages. They were using a generic script that they had developed for spacecraft subsystem testing that used a position in Orbit 22, which also used a time in Orbit 22 in the simulation setup. Since the orbit-in-the-life test was being performed for Orbit 5, as soon as they set the spacecraft time to Orbit 22, four years in the future, all the time tags fired. The orbit being simulated was discussed in planning meetings for months. That time-tags were being loaded and would remain over night was also discussed many times, and yet, the miscommunication that the spacecraft time could not be changed overnight still occurred.

Second, practice team interactions. The interactions among the various teams went much smoother during Mission Simulation #3 because sequences where the simulation and spacecraft setup and control were handed off among the teams were practiced prior to the actual mission simulation during thermal vacuum testing. Because of the practice, team members knew what to expect and approximately how long the various steps would take. Having realistic time estimates for setup and handovers also helped layout the test sequences.

Third, test all procedures and test scripts as realistically as possible prior to the actual mission simulation. Running all the test blocks on the hardware-in-the-loop simulator allowed us to find most of the problems with the test scripts and procedures prior to running them during the mission simulation. This not only allowed the test to run smoother, it made it easier to troubleshoot problems.

Fourth, sometimes operational concepts need to be adjusted. On PSP, for instrument commanding, the ground system is set up in a bent pipe model. This means that during a command contact, the command connection is opened by the Mission Operations team and the instrument teams command their own instruments directly. The command files for each instrument go into a queue. Then, they are uplinked in a round robin manner based on the order the command files are received by the MOC ground system. This process worked successfully on several previous missions. However, those missions had much higher uplink data rates than PSP. After performing Test Block O1 at an uplink data rate 31.25 bps where one large instrument command file took the entire test duration to uplink, essentially blocking all the other instruments from commanding, the team realized this operational concept needed to be adjusted.

During flight, the instrument teams now schedule their commanding so other teams are aware of when commanding is being done. Also, they normally will reserve an entire uplink contact to do their commanding. In the cases where more than one instrument team needs to command during the same contact, they go one at a time instead of simultaneously.

Fifth, hardware-in-the-loop simulators are not the spacecraft. Even though Test Block E2 for TCM-1 had been tested on the hardware-in-the-loop simulator, the maneuver failed during testing with the spacecraft. Because of the nature of the flight software bug, it could only be found during spacecraft testing.

Finally, performing testing in as realistic a manner as possible is worthwhile. Several of the issues discussed in the previous lessons could not have been found during individual subsystem testing. It took the combination of testing the spacecraft using realistic uplink and downlink rates, real operational procedures, and placing the spacecraft into as realistic a simulation as possible.

10 Conclusion

In conclusion, Mission Simulation #3 performed during thermal vacuum testing went very well. This was because of pre-test preparations performed by the team, including practicing team interactions and executing test scripts on the hardware-in-the-loop simulator prior to the test. Issues with flight software and operational products were found both during pre-test preparation and test execution, which were resolved and re-tested. Several lessons were learned which can apply to future tests.

In addition, Orbit 5 was completed successfully in flight from 3 April–2 August 2020. The success of Orbit 5 in particular and other orbits in general in flight was and is greatly because of the team being well practiced and ready and the pre-flight work done during Mission Simulation #3.

Acknowledgements The author would like to acknowledge all the members of the Parker Solar Probe mission operations team, the I&T team, the spacecraft subsystems and instrument teams who worked to make Mission Simulation #3 a success.

Appendix

Acronyms/Abbreviations

APL	The Johns Hopkins University Applied Physics Laboratory
CLT	Command Loss Timer
CSPR	Cooling System Primary Radiator
DSN	Deep Space Network
EPI-Hi	Energetic Particle Instrument–High

EPI-Lo	Energetic Particle Instrument–Low
FIELDS	Fields Experiment
FSW	Flight Software
GNC	Guidance, Navigation & Control
HGA	High Gain Antenna
ISOIS	Integrated Science Investigation of the Sun
I&T	Integration & Test
LGA	Low-gain antenna
MAG	magnetometer
MOC	Mission Operations Center
PSP	Parker Solar Probe
RF	Radio Frequency
SACS	Solar Array Cooling System
SBC	Single Board Computer
SOC	Science Operations Center
SOC	State of Charge
SSR	Solid State Recorder
SWEAP	Solar Wind Electrons Alphas and Protons
TCM	Trajectory Correction Maneuver
TPS	Thermal Protection System
TVAC	Thermal Vacuum Testing
WISPR	Wide-field Imager for Solar PRobe

Evolution of the Canadian Radarsat Satellites

Christophe Belzile, Christian Carrié, Nimita Wadhwa, Brian Lawrence, Neil Gibb, and Peter Allan

Abstract Since 1995, Canadians and international users have benefited from the high-resolution imagery captured by three generations of RADARSAT satellites: RADARSAT-1, RADARSAT-2 and the RADARSAT Constellation Mission (RCM). RADARSAT imagery is used for coastal water surveillance, ice formation in shipping lanes, disaster zone observation and ecosystem monitoring. The technology incorporated into these satellites has progressed over the past two decades. Significant technological advancements were made between RADARSAT-1 and its successors: RADARSAT-2 and RCM. The constellation of three RCM satellites has significant differences over its predecessors: RADARSAT-1 and RADARSAT-2. With three equally spaced RCM spacecraft on the same orbital plane, imagery of Canada and large maritime areas of the coasts can now be acquired daily instead of every three days. The orbit of the three RCM satellites is controlled within a 120-m tube in order to provide Coherent Change Detection (CCD) and Differential Interferometric SAR capabilities. The length of the SAR antenna has been reduced from 15 m to 7.5 m and the size of the solar panels and batteries have been reduced significantly because of evolving battery technology and more efficient solar cells. Another significant difference is RCM satellites include Automatic Identification System (AIS) receivers in

C. Belzile (✉) · C. Carrié · N. Wadhwa · B. Lawrence
Department of Space Utilization, Canadian Space Agency, 6767 route de l'Aéroport, St-Hubert, Québec J3Y 8Y9, Canada
e-mail: Christophe.belzile@canada.ca

N. Wadhwa
e-mail: Nimita.wadhwa@canada.ca

B. Lawrence
e-mail: Brian.lawrence@canada.ca

N. Gibb
Macdonald Dettwiler and Associates, 13800 Commerce Pkwy, Richmond, British-Columbia V6V 2J3, Canada
e-mail: neil.gibb@mdacorporation.com

P. Allan
Macdonald Dettwiler and Associates, 21025 Trans-Canada Hwy, Ste-Anne-de-Bellevue, Québec 9X 3R2, Canada
e-mail: peter.allan@mdacorporation.com

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_4

order to track ship traffic in maritime regions, where the AIS-generated data can be correlated with the SAR images to identify ships of interest. Ground control of all three generations of RADARSAT satellites is based at the Saint-Hubert (SHUB) mission control facility at the Canadian Space Agency (CSA) headquarters at Saint-Hubert, Quebec, Canada. RCM operations are performed at a newly constructed Primary Control Facility (PCF) that includes image order handling, mission planning, flight dynamics, satellite control, image product generation and image quality subsystems. Independent S-band and X-band ground stations have been available for RADARSAT-1 and RADARSAT-2; however, RCM Canadian Ground Stations have been upgraded for simultaneous S-band and X-band communication. In order to receive and process AIS data and augment SAR imagery in Canada's maritime regions, two additional ground terminals have been established by the Canadian Department of National Defence. This paper will compare and contrast the three generations of RADARSAT satellites and provide overviews of their designs. It will also summarize the evolution of the Ground Segment, and how technological advancements and lessons learned have impacted the concept of operations for this new fleet.

Keywords RADARSAT · Synthetic Aperture Radar · Evolution · CANADA

1 Evolution of Radarsat

In 2020, 1,283 satellites were launched, which stands as the highest number of satellite launches in a year as compared to all the previous. Therefore, it becomes more and more important to justify spending money and sending satellites up which are filling up the Low Earth Orbit (LEO). Earth Observation (EO) satellites with their polar orbit enable us to observe all the Earth and collect data from a vantage point that doesn't compromise national sovereignty. Humans have an enormous impact on the Earth. Earth observations are indispensable to monitor and manage the negative impacts of humanity. Since 1995, Canadians and international users have benefitted from the high-resolution synthetic aperture radar (SAR) imagery captured by three generations of RADARSAT satellites: RADARSAT-1, RADARSAT-2 and the RADARSAT Constellation Mission (RCM). RADARSAT-1's operational life was over three times its expected lifespan of five years. Launched in 1995, it kept functioning until May 2013. RADARSAT-2 was launched in 2007 and is still operational and has exceeded its lifespan of seven years by a factor of two. RADARSAT imagery is used for coastal water surveillance, ice formation in shipping lanes, disaster zone observation and ecosystem monitoring. This paper will cover the evolution of three generations of RADARSAT satellites. The evolution includes the satellites themselves and how the payload evolved (i.e., SAR). Many upgrades to the ground stations were made and whole dedicated ground segment were required (Fig. 1).

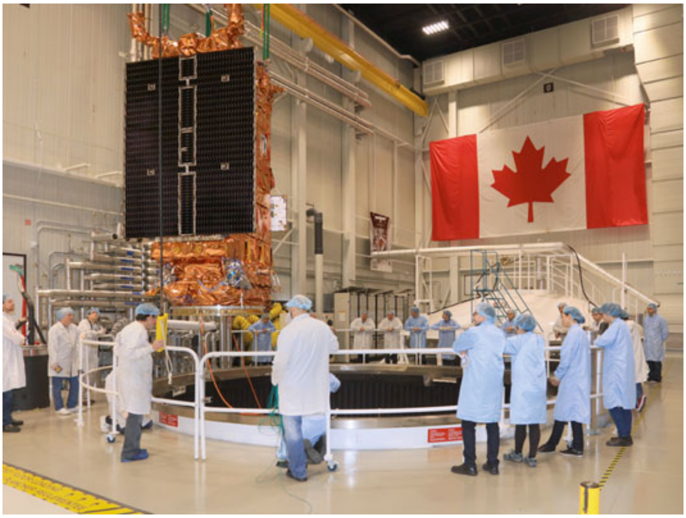


Fig. 1 RCM satellite being lowered in temperature vacuum chamber

1.1 Operational Model Transition

The operations model changed from one generation of RADARSAT to another. The operations model evolved from a fully government owned and operated satellite (RADARSAT-1) to RADARSAT-2. To mitigate cost escalation risks, the Canadian government decided RADARSAT-2 to be owned and operated by private industry with Canada having a set amount of free images. The Canadian Government decided they would revert to being the owner and operators of the most recent iteration of RADARSAT satellites (i.e., RCM). Currently, we have the operations group with civil servants and contractors. We're moving to a model of contractors only operations group with civil servants embedded in key areas within the group. The satellites will remain under government control and management.

1.2 Synthetic Aperture Radar (SAR) Technology Evolution

The technology incorporated into these satellites has progressed over the past two decades. Significant technological advancements were made between RADARSAT-1 and its successors: RADARSAT-2 and RCM. The SAR instrument evolved from transmitting from a high-power amplifier through a slotted waveguide passive antenna, with phase control in the cross-track plane using high-power phase shifters, and reception via a single receiver to a fully active antenna using transmit and receive modules (TRM), distributed across the antenna array.

Table 1 RADARSAT BUS manufacturer

Mission	Manufacturer	Model
RADARSAT-1	Ball Aerospace	BCP-4000
RADARSAT-2	Thales-Alenia	Prima Bus
RCM	Magellan Aerospace	MAC-200 SmallSAT bus

The generations of the transmit pulse waveforms progressed from the three different bandwidths of RADARSAT-1 to a wide variety of pulses with various durations and bandwidths. The evolution of the pulse characteristics resulted from the received signal filtering that allowed more choice in the selection of the pulse waveforms. The single polarization of RADARSAT-1 (HH) evolved to the full polarimetric capability of RADARSAT-2 and further enhanced in RCM by the addition of compact polarimetry (Table 1).

1.3 BUS Development Throughout RADARSAT Program

The first generation of RADARSAT used immense solar arrays to generate power and now with RCM the size of the solar array is considerably smaller (see Table 2) because the SAR antenna is smaller and the power requirements are less. In addition, the efficiency of solar arrays has increased (Table 3).

Table 2 RADARSAT Spacecraft Comparison

	RADARSAT 1	RADARSAT 2	RCM
Mass of Spacecraft	2750 kg	2200 kg	1480 kg
SAR antenna Length	15 m	15 m	6.75 m
Solar Panel Dimensions (LxH)	1.3 m × 2.2 m (1 panel —composed of 2 arrays of 5 panels)	1.8 m × 3.7 m (1 panel —composed of 2 array of 3 panels)	1.7 m × 2.2 m
Solar Array Power Generation	2.5 KW	3.1 KW	0.875 KW
Orbit	800 km, 98.6º inclination Dawn-Dusk	798 km, 98.6º inclination Dawn-Dusk	600 km, 97.7º inclination Dawn-Dusk

Table 3 RADARSAT ground station network

Sat/GS	SHUB	SASK	GATN	KRN	ICAN	PASS
RADARSAT-1	S	S	X			
RADARSAT-2	S	S	X		S/X	S/X
RCM	S	S	S/X	S/X	S/X	S/X

The remaining component in the power system are the batteries. In RCM, there are two Lithium-Ion batteries. Each battery is composed of 352 Cells and rated at 66 A·h. This differs greatly from RADARSAT-2, which is a single NiH2 battery of 23 cells rated at 97 A·h. For RADARSAT-1, three 48 A·h batteries provided energy storage. On-board storage of SAR image data improved from magnetic tape reels on RADARSAT-1 to solid-state, digital mass memory units. Security was also enhanced by introducing encryption of S-band and X-band data (RADARSAT-2 and RCM).

For RCM, on-board AIS was added for the enhanced ship detection capabilities. The AIS system allows both on-board message detection as well as ground processed. The AIS service can be combined with the SAR images creating high-value products for the National Defence.

2 Ground Segment Evolution

The RCM ground segment was designed and integrated for the RCM mission based on some RADARSAT-2 algorithms, but the technology was upgraded to the latest state-of-the-art software configurations.

The real-time control system for RADARSAT-1 and RADARSAT-2 were essentially driven by the bus manufacturer (Spaceraft Operations Console (SOC) for RADARSAT-1 and Satellite Control and Operations System 2000 (SCOS-2000) for RADARSAT-2). The multi-mission environment in the CSA's satellite operations infrastructure required harmonization of the satellite control across its missions and, therefore, a commercial off-the-shelf (COTS) tool was selected. The ITOS from Hammers Company has been used for all CSA missions after RADARSAT-1 and its latest version Galaxy (or ITOS 8) was selected for RCM.

Antennae were built to support the extra passes that would result from RCM. These new antennae reside at Gatineau (GATN), Inuvik (ICAN) and Prince Albert (PASS) ground stations. These are known as S-Band and X-Band Ground Terminal (SXGT). Another ground station was added at a northern latitude outside Canada to reduce image delivery time, related to X-band downlink, from five hours to 30 min. The station is in Kiruna, Sweden and called the Northern Ground Terminal (NGT).[1]

For RADARSAT-2 we have 13–14 passes a day, for RCM this number jumped to 34–36, mostly because there's three satellites, therefore, we needed a rapid method to confirm ground stations availability. This is where the Antenna Reservation System (ARS) comes into play. It enables us to get ground station availability within five minutes versus many hours that was required previously for RADARSAT-2. It also, in turn, allows us to do Fast Tasking. Fast Tasking is where an approved client (GoC or partner) can ask with four hours' notice a specific scene and obtain it within10-30 min after acquisition. The delay depends on the urgency of the request and the priority of the client. For Global and Canadian disaster management application,

[1] Reference: Patrick Irvin, CSA TT&C Engineer.

RCM provides 2-h data latency from downlink to data delivery, and for ecosystem monitoring applications, 24-h data latency from downlink to data delivery.

A Transfer Agent (TA) was added to the Data Archiving Software (DAS) at SXGT ground stations. The purpose is to downlink priority images before other images. It also directs where the image is directed to either St-Hubert (SHUB), a CSA location, or Polar Epsilon 2 (PE2), a DND location in Ottawa. From SHUB, the images are sent to the Canadian Data Processing Facility (CDPF) in Gatineau. The images destined for PE2 are overlayed with AIS data.

In order to minimize reimaging in the same area many times in different imaging modes, CSA developed standard coverage. Standard coverage are SAR image acquisition plans that are primarily designed for the operational needs of the Canadian government. These comprise pre-defined and pre-planned image acquisition based on common parameters (e.g., imaging modes and AOIs). The design is such that they're intended to offer consistent and predictable SAR based on long-lead planning.

2.1 Flight Dynamics Optimization

The Flight Dynamics (FD) system were custom-built and adapted to each RADARSAT missions.

For RADARSAT 1, it was specifically built software that ran on the now defunct Virtual Address eXtension (VAX) servers. The operating system was Virtual Memory System (VMS). Everything was done manually and used command line even though it had a graphical user interface (GUI). This was virtualized on a Windows machine when the hardware and software were no longer supported by the original manufacturer.

The FD system for RADARSAT-2 is composed of a custom-built software interfacing with COTS software packages. The original system at launch, termed ODMP (orbit determination and manoeuvre planning), was improved on after several years in operation with the addition of EDOT (Enhanced Definitive Orbit Tool). EDOT offers an improvement in orbit determination accuracy and automation. EDOT runs automatically and does not need operator intervention during nominal operations. The FD system for RCM used EDOT as a base for orbit determination as it too interfaces with System Tool Kit (STK) and Orbit Determination Tool Kit (ODTK) from Analytical Graphics Inc. (AGI), and runs automatically. A significant improvement with the FD system for RCM is automatic manoeuvre generation capabilities. The orbit control strategy for RCM is more ambitious than previous missions as it maintains a reference orbit tube to within 120 m at all times throughout the orbit. This reference tube has a 12-day repeat ground-track so with a three-satellite constellation, there is a four-day revisit time to almost any location on the Earth. Two types of manoeuvres, drag make up (in-track and cross-track component) and inclination, are planned autonomously by the FD system. This is a significant improvement because manoeuvres for earlier RADARSAT mission took several hours.

The FD System for RCM generates and distributes the determined orbits, predicted orbits, orbit offsets, Tube statuses and eclipse reports automatically. These products are distributed to the planning software. The orbit offset is used to determine the predicted orbits which contain the two line element (TLE) Tube excursions that can occur when the satellite ground track constraints are exceeded and when the inclination is not within the given constraints. The tube status refers to how much time the spacecraft is out of the tube. This file will be empty if a spacecraft is in the tube 100% for the three-day period the tube status covers.

3 Radarsat Constellation Mission (Rcm)

4 RCM SAR Sub-System

The RCM payload is a continuation of the evolution from RADARSAT-2 introducing a system that supported up to 16 beam ScanSAR imaging and more capable multiple polarisation options compared with RADARSAT-2. The RCM imaging modes are shown in Fig. 2 with the main characteristics of the payload given in Table 4. The RCM payload uses a small antenna at a lower orbit altitude with the capability of transmitting vertical, horizontal or circular polarisation, receiving either linear or dual polarisation. The polarisation can be switched in bursts in strip map imaging or alternating every beam cycle in ScanSAR, making the RCM payload more versatile in terms of polarimetry than RADARSAT-2.

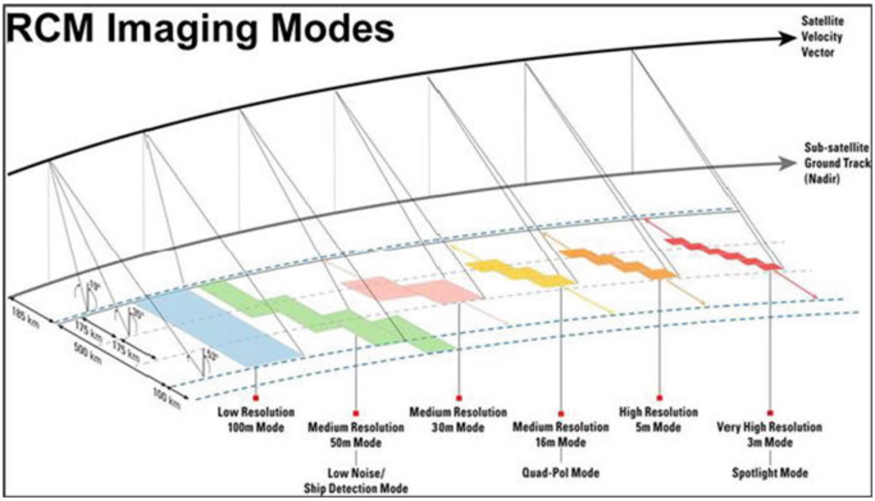

RCM Imaging Modes
Satellite Velocity Vector
Sub-satellite Ground Track (Nadir)
185 km
175 km
175 km
500 km
100 km
19°
35°
53°
Low Resolution 100m Mode
Medium Resolution 50m Mode
Low Noise/ Ship Detection Mode
Medium Resolution 30m Mode
Medium Resolution 16m Mode
Quad-Pol Mode
High Resolution 5m Mode
Very High Resolution 3m Mode
Spotlight Mode

Fig. 2 RCM imaging modes

Table 4 RCM Payload Characteristics

Parameter	Characteristics
Center Frequency	5.405 GHz
Polarisation	VV, or HH, or Quad pol or compact pol or burst VV-HH
Polarisation Isolation	Better than 30 dB
Antenna Size	6.75 m × 1.37 m
Resolution	Variable from 3 to 100 m
NESZ	−20 to −30 depending on beam mode and incidence angle

The payload in the RCM context includes all data storage and downlink functions to provide a more integrated system (this is actually a reversion back to the RADARSAT-1 configuration). To support the maritime surveillance mission, the RCM payload incorporates an AIS receiver with the capability of collecting AIS transmissions and combining the received data with a SAR image or collecting the information independent of the radar.

The basic functions of the major elements are:

The antenna provides the radio frequency (RF) interface to free space. It contains the final stage RF amplification to generate the required pulse transmissions and includes the first stage of receive amplification. The antenna has the capability of forming several different beam shapes, using a set of pre-defined aperture excitations. The antenna provides the capability of transmitting vertical, horizontal or circular polarisations and simultaneously receiving vertical and horizontal polarisations. The antenna is reset on every pulse and, hence, allows alternating vertical and horizontal transmissions to support quad-polarisation imaging.

The Payload Control Unit (PLCU) acts as the operational coordinator for the radar operations, programming the various hardware elements of the radar to perform the required imaging scenario. The PLCU responds to a set of image requests to manage the state of the other payload equipment.

The Power Distribution Unit (PDU) provides the power switching functions for the operational heaters and Tile Controller Units (TCU) in the SAR Antenna.

The Central Electronics (CE) is the main radar transponder. It generates the required modulated pulses to drive the antenna, and collects and digitizes the vertically and horizontally polarised return echoes from the ground. The digital representation of the echo is formatted and sent to the Solid-State Mass Memory (MMU).

MMU act as a temporary storage for the formatted science data. Data is provided by the Central Electronics Subsystem (CES), and subsequently sent to the X-Band transmitter, via the external encryption unit in the bus, for transmission to the receiving ground station.

X-Band transmitter is used to transmit the science data to the ground (science data in the context of the RCM payload is SAR image data, AIS data, payload ancillary data and Bus ancillary data). The data is provided from the MMU, via the encryption unit, in a parallel bit stream. The transmitter takes the parallel data and uses it to modulate a X-band carrier, using an 8-PSK type modulation. This modulated RF is then sent to a dedicated antenna for transmission to the ground.

The Automatic Identification System (AIS) subsystem comprises a receiver and two sets of simple monopole antennas. The AIS receiver is used to record AIS transmissions from all ships within the field of view of an orthogonal pair (two mono-pole antennas oriented at 90° to each other) of AIS antennas. This data is either combined with a concurrent SAR image data set or is stored separately.

5 RCM Payload Operations

From an operations point of view, there are some challenges to overcome. The first was to minimize the command volume needed to cover up to 24 h of autonomous operation, and the second was to design the Payload to allow units to be powered off when not needed to conserve power (the platform provided the required orbit average and peak power to support imaging, but could not support a full standby mode for the payload).

The RCM payload acts autonomously, with limited support from the Bus. The payload operation and control is provided by the PLCU which provides the commands for operating the units, handles fault detection, isolation, and recovery (FDIR) and also provides the radar timing function.

Payload operation is based on the concept of activities. To acquire SAR image data, AIS data or to downlink data to the ground, the user loads the appropriate activity request to the payload. The PLCU then initializes the required units and implements the activity at the requested time. During periods between activity execution, units that will not be needed in the near term are powered off. This power management and unit utilization management is performed by the PLCU independently.

This also means that the PLCU schedules the execution of all activities based on the activity requests. As part of that scheduling, the PLCU also performs some limited conflict resolution between activity types.

Activity requests are used to:

- Define the image mode, start time of the image (transmission of the first pulse), and the duration of the image.
- Define the start of a data downlink from the stored data in the payload on the X-band link. This can be on either of the downlink channels, and may request the data from the current real¼–time collection, or data from a previously stored image.
- Collect data from the AIS receiver. This may be done in parallel with an image activity or as a separate activity with no imaging.

Activity requests are self-contained definitions and are in the form of a single command load. Sequencing of activities is only needed when overlapping image and AIS activities when AIS data is to be merged with the image data. In this case, errors associated with the command sequence will cause the PLCU to reject one of the activities or it will result in loss of data.

Because the payload units are not always powered on or active, telemetry collection from the payload by the bus is implemented using a request and only if data is available, then telemetry is supplied. Further reduction of the overall telemetry volume can be achieved by reducing the collection rate or the transmission rate to the ground by the command and data handling subsystem in the Bus.

6 RCM AIS Payload and Application

The idea behind the addition of AIS receivers is to enhance the maritime surveillance capability of the mission. The AIS payload will receive ships' messages in a wider swath than the accessible swath of the SAR. The AIS payload does not exist on RADARSAT-1 and RADARSAT-2 and is unique to RCM.

As noted in the above sections, the received AIS data can be combined with a SAR image or can be collected information independent of the radar. The payload schedules activities for AIS data collections according to requests received from the user. Each request identifies what is to be performed, when it is due to start and, for most types, how long it is to last. This is the only method available to the user for executing any form of AIS acquisition.

Any AIS activity is defined using the general AIS activity that defines the start time and duration of the activity, with a time value to set the method of data acquisition as buffered within the AIS receiver or transferred to the MMU via the CE. The payload software uses these to determine when to enable power to the CE and MMU and when to open and close the data file in the MMU. An AIS activity can be executed at the same time as an image activity, provided that the AIS activity does not cause any conflict in storing data in the MMU. AIS only data can be collected independent of any imaging activity as well.

Four AIS antennas are located one on each of the four corners of the SAR antenna, which is a rectangular planar array electrical aperture of 6.74 by 1.410 m which faces the Earth during normal operations. The AIS receiver is based on Direct RF Sampling (DRFS) technology using a single RT3PE3000L Flash FPGA to implement most of the functionality.

Each receiver (there are two–primary and redundant, sharing a single housing but otherwise independent) comprises a pair of RF amplifiers (split into two boards), a digital processor board housed in the upper tray housing, and an Electronic Power Conditioning (EPC) unit housed in the lower tray housing. The EPC comprises a Power Control and Conditioning Unit (PCCU), DC/DC converter and regulator board. The primary and redundant sides are designed to be operated in a cold-redundant configuration, i.e., only one side powered at a time. The receiver is turned on and off by a discrete telecommand via the PLCU interface. The PLCU interface responds to row-column relay drive pulses. One row-column pair is used to command the receiver on, the other pairs command the receiver off. The DC power interface and PLCU telecommand interface are both Galvanically isolated from the chassis and equipment ground system.

The operational TM/TC on the PLCU interface is provided by a dual-redundant CAN bus, 6-byte Telecommands control the mode of operation (e.g., raw spectrum or on-board processing for each channel). Telemetry requests enable the receiver to report health and status of the unit.

The unit is designed to operate over the temperature range from -10C to 50C. The receiver is designed to operate with 100% duty cycle and with a service life of 7.33 years in LEO.

7 RCM Image Calibration and Quality

Many image quality and calibration techniques were introduced on RADARSAT-1 and built on during the following RADARSAT-2 and RCM missions. Continuous improvement of ScanSAR processing, optimization of existing beam patterns and addition of new operating modes are a few of the techniques used to maintain excellent image quality. This section summarizes the changes in routine image quality techniques used across RADARSAT-1, RADARSAT-2 and RCM, as shown in Table 5.

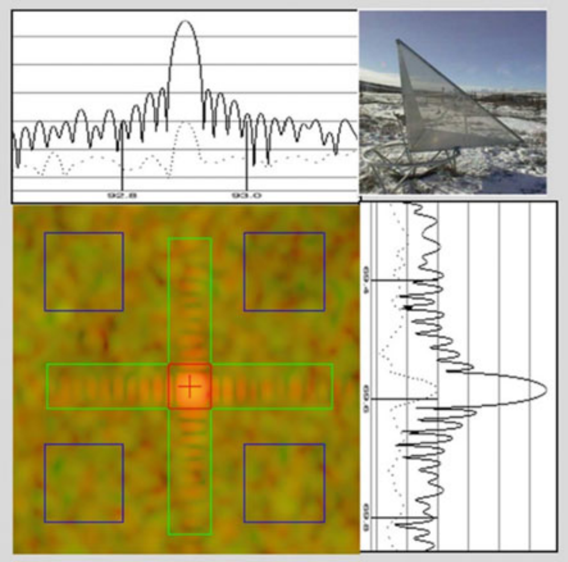
Impulse Response Measurements

Starting with RADARSAT-1, passive and active point targets have been used to measure impulse response functions using sites across Canada. Since 2011, additional sites have been used in order to expand our capability to characterize the impulse responses for the growing number of operational modes on RADARSAT-2. These are located in Switzerland, Argentina, Japan, California, Alaska and Sweden. This also

Table 5 Image Quality Monitoring techniques

		RS1	RS2	RCM
Impulse Response Measurements	IRW	□	□	□
	PSLR	□	□	□
	ISLS	□	□	□
	Geolocation	□	□	□
Noise Levels	Noise	□	□	□
	NESZ	□	□	□
	SDNR	□	□	□
Radiometric Accuracy and Stability		□	□	□
Polarimetric Accuracy and Stability		*NA*	□	□
Beam pointing Accuracy and Stability		*NA*	□	□
Non-Imaging Calibration		*NA*	□	□

Fig. 3 RADARSAT-1 corner reflector and antenna dish point target measurements



helps reduce the error caused by pointing issues, weather effects (particularly snow accumulation), contamination by radar returns from the surroundings, and factors that modulate radar returns, particularly tracking mechanisms of antenna dishes (Fig. 3). Many of these sites were used during RCM commissioning and the following routine operations.

Noise Levels

System noise levels have been measured using calibration beam modes with the antenna operating in receive only (no transmitted pulse). These measurements are typically taken over the south pacific doldrums.

Interference

Significant interference from ground-based radar sources and wireless networks is rare. Frequently, image artifacts are seen on RADARSAT-2 due to mutual C-Band interference with Sentinel-1a (launched 2014) and Sentinel-1b (launched 2016), as shown in Fig. 4. As the ground tracks of all spacecraft are well known, the crossing points are predicted and monitored to minimize impacts to image quality. To date, no significant interference between RADARSAT-2 and RCM has been observed.

Radiometric Accuracy and Stability

Since the RADARSAT-1 mission, radiometric accuracy and stability have been monitored using acquisitions from over the Amazon rain forest where the backscatter properties are well known (Fig. 5). The extracted elevation patterns are compared against

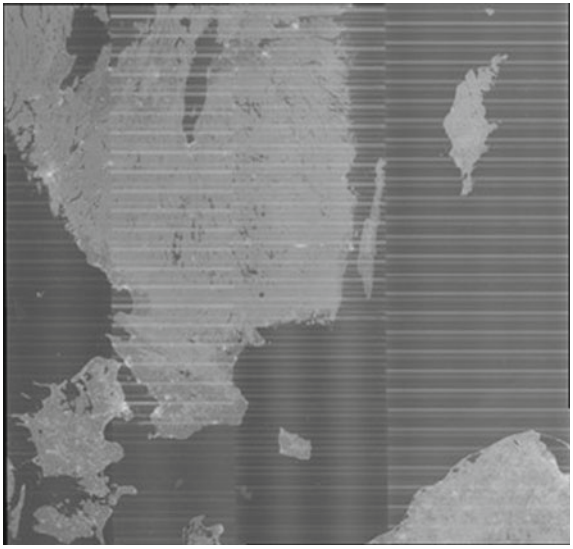


Fig. 4 Mutual radar interference between Sentinel-1a as seen from RADARSAT-2

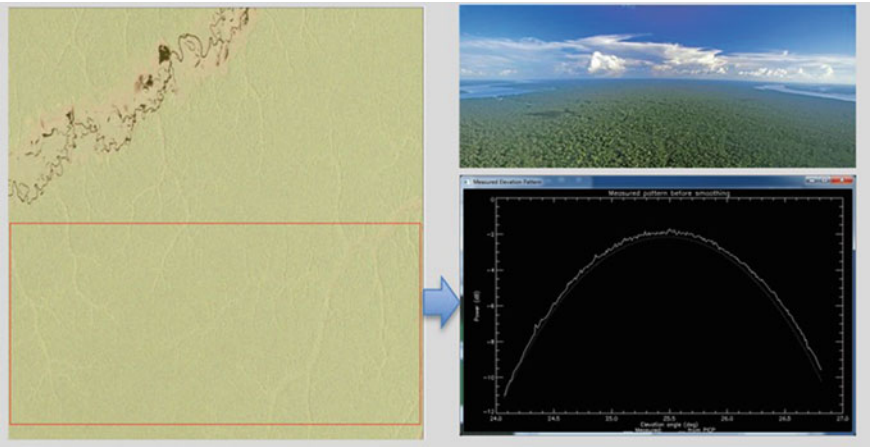


Fig. 5 Amazon rainforest distributed target measurements

reference patterns to determine any radiometric deviations such as seasonal variations and imaging mode differences [incidence angles, polarization (RADARSAT-2, RCM), pass direction and look direction (RADARSAT-1, RADARSAT-2)].

Because of the failure of the on-board recorder operations of RADARSAT-1 in 2008, it was no longer possible to store images acquired over the Amazon for later downlink over a Canadian network station. Routine radiometric measurements were acquired over the Canadian Boreal Forest belt and were successfully used for the rest of the mission.

Polarimetric Accuracy and Stability

The technological advancements of RADARSAT-2 and RCM required additional monitoring techniques. The new ability to image in dual and cross pol modes on RADARSAT-2 and compact pol on RCM added the need to monitor polarimetric accuracy and stability parameters. Inter-channel registration and the imbalance and crosstalk between vertical and horizontal polarized channels on transmit and receive are monitored by analysis of Quad-Pol (fully polarimetric) mode Single-Look Complex image products acquired over the Amazon rain forest.

Beam pointing Accuracy and Stability

During the RADARSAT-1 mission, a process was developed to monitor beam boresight angles. From the start of the RADARSAT-2 mission, beam pointing accuracy has been systematically monitored in all three axes (roll, pitch and yaw) through the analysis of image products and their metadata.

Monitoring of beam pointing errors in both elevation and azimuth show that the RADARSAT-2 Attitude Operations Control System (AOCS) upgrade performed in December 2018 (after losing two spacecraft attitude gyros) continues to perform well.

TRM health (Non-Imaging Calibration)

With the utilization of Transmit Receive Modules (TRMs) on RADARSAT-2 and RCM routine, TRM health monitoring tests were introduced using a suite of diagnostic beam modes. These tests consist of operating the TRMs individually and as rows/columns to assess the amplitude and phase performance. The tests routinely performed over the Siberian boreal forests and the Pacific Doldrums for both RADARSAT-2 and RCM.

8 RCM Mission Operations and Status

As of November 2020, the RCM constellation reached a full-year of nominal routine operations. The health of the constellation is nominal in that the three spacecraft are fully operational and delivering data to the users with the required performance metrics (availability, volume, quality and latency). The Flight Operations Team (FOT) has managed approximately 95 system anomalies since launch (Space segment and Ground Segment combined). The majority are minor and non-recurring anomalies. Constellation system availability has been above the required 90% requirement.

Figure 6 shows the total weekly SAR on-time trend since March 2020. We observe a general trend in usage towards maximum capacity target of 4200 min of SAR on-time per week.

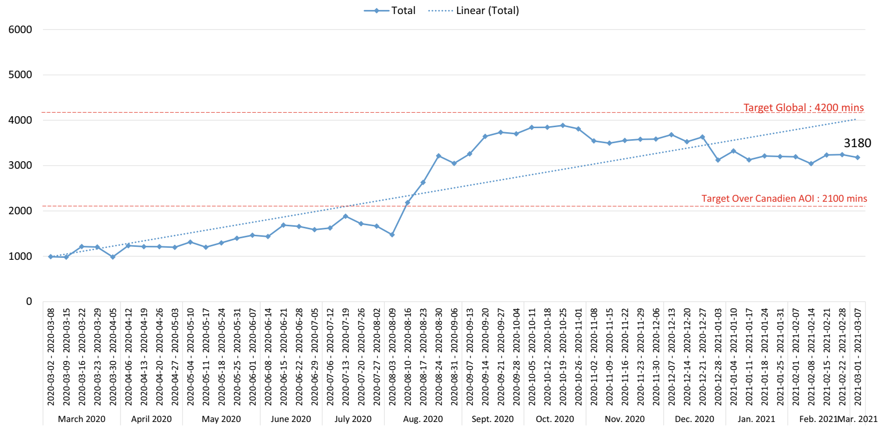


Fig. 6 Total weekly SAR On-time evolution

Payload anomalies with significant outage impacts have been investigated in priority and for which ground system changes have been designed and deployed. For a few anomalies, a payload flight software is being designed and should be uploaded in coming months.

The Magellan Bus platform is operating nominally on all RCM spacecraft and has caused no system outage to date.

Two Collision Avoidance (COLA) manoeuvres were executed nominally. The first was on November 25th where a drag make up burn (has both an in-track and cross-track component) was performed, but the probability of conjunction with the unknown object was deemed too high so a prograde burn was immediately performed to ensure the miss distance was acceptable. The second was a possible conjunction between RCM-2 and Centauri-1 on January 2nd 2021 and an avoidance manoeuvre was performed that served two purposes: the prograde manoeuvre was aimed at augmenting the miss distance and maintaining the RCM Ground Track.

The RCM Ground Segment was a new-generation integrated design with some level of heritage on some components such as the Planning Systems and the real-time control system (Galaxy from the Hammers company). During the first-year, the reality of flight operations has triggered various system and procedural issues that the development phase had not foreseen in the pre-launch verifications. The number of issues was significant and aggravated by the coronavirus disease of 2019 (COVID-19) situation, but the FOT supported by the Ground segment sustaining team could converge and operationalize the ground segments at the end of phase E1.

In order to better predict and foresee failure, CSA has built a tool to see trends and possibly over-stressing of the payload. The Central Electronics (CE) unit of the RCM payload is the main radar transponder that generates modulated pulses to drive the antenna and digitizes received echoes from the ground. The CE powers on when the satellite is executing a SAR image activity and/or acquiring AIS data, and is powered off otherwise. The number and duration of these on/off cycles cause thermal fatigue damage to the components within the CE.

CSA is in the process of designing a software tool, referred to as the CE TF (Thermal Fatigue) Tool, that estimates thermal fatigue damage of critical CE components to ensure the payload activity schedules are not over-stressing the unit. Although the tool has been designed and developed by CSA, its algorithm is based on thermal fatigue damage assessments and component qualification testing completed by RCM prime contractor, Macdonald, Dettwiler & Associates (MDA), during the development of RCM. The tool uses spacecraft telemetry to determine the number and duration of CE cycles along with the delta temperature change of the CE as a result of each cycle. Applying the Modified Coffin-Manson equation to the spacecraft telemetry in conjunction with MDA TF analysis and qualification, data allows for a method of estimating the thermal fatigue damage experience by the CE components during RCM operations. The tool uses this approach to calculate the damage incurred by components up to the present time, referred to as the past TF damage, as well as to estimate the damage that will be incurred from the present to the end of the mission, referred to as future TF damage. The future TF damage requires certain assumptions to be made concerning the CE temperature towards the end of the mission. The total TF damage that is estimated to be incurred by a component at the end of the mission is the sum of the past TF damage and the future TF damage. The damage variable D used by the tool to represent TF damage gives the fraction of a component TF life varying from 0% for an undamaged component up to 100% for a fully damaged component leading to its functional or physical catastrophic failure. The tool converts this total thermal fatigue damage to a design safety factor, where the safety factor is simply the inverse of the damage.

Although the tool is still under development, it is envisioned that RCM operations engineers, who will verify the outputs to ensure that the safety factors of the CE components over the mission life satisfy design requirement thresholds, will run the tool every 12 days. If a safety factor violates a threshold, an investigation will need to be conducted to determine the best path forward (i.e., update the CE TF Tool model, or adjust the RCM payload Duty Cycle). It is also envisioned that engineers will periodically perform a detailed review of the CE TF Tool outputs and the pertinent spacecraft telemetry in order to determine whether the algorithm and/or model parameters need to be updated as the mission progresses.

The COVID impacts on RCM operations started in March 2020 and has been since a source of continuous adaptations. Given the RCM system is operated from a highly restricted zone, teleworking options were reduced and hence a delicate balance between health and safety, operations and security had to be established in a rapid and agile way. Rapid decision making was required at the operational level in coordination with the private sector partners and the Canadian Space Agency's Health and Safety COVID committees. As a result, there have been no COVID cases within the Satellite Operations Team and the RCM operations continued nominally with no service degradation.

9 Future of the Radarsat Project

The RADARSAT program has been on-going for 25 years. We expect it to be even more relevant and necessary for the future mostly with climate change as it has its direct effect on weather and increased frequency of natural disasters. Naturally, all the technology enhancement that occurred within 25 years (i.e., batteries, solar panels and electronics) made their way into the RADARSAT satellites, in addition, many features were added to enhance their capabilities, such as AIS. In order to increase the capacity, instead of relying on a large single satellite, a constellation of three satellites was selected.

In the future, managing our resources and natural disasters using space assets will be a requirement and RCM will enable us to do just that with a re-visit cycle of four days of almost anywhere on earth. The capability of imaging anywhere in Canada within 24 h will be especially essential. Climate change is an even greater concern for Canadians because the augmentation in temperature near the poles is much higher than for the rest of the globe. 20% of Canada's landmass is in the Arctic, therefore, this is a top priority item for Canada. Another disconcerting concern in the Arctic is that the permafrost is melting. This is alarming because of the amount of carbon that will be released in the atmosphere. This is something that we can monitor with greater ease from space. Other areas of concern are water located on mountains (i.e., glaciers) are disappearing. We can monitor these phenomena and gather data on how quickly the glaciers are receding using RCM. Canada has the longest total coastline in the world with 243,042 km (includes mainland coast and the coasts of offshore islands). Therefore, the rising level of the oceans and all the pollutants sent into them are of a significant concern to Canadians and the international community. This can be observed from the spatial vantage point. Deforestation is also an important problem, as it releases more carbon dioxide in the air than vehicles, especially in the rainforest, but this is equally a problem in Canada. All this is important because of a great quote from Ms. Maria Fernanda Espinosa Garcès, President of the United Nations General Assembly: "If you can't measure it, you can't manage it". It really summarizes climate change and how EO is essential in its monitoring.

One of the possible enhancements for the next generation of RADARSAT satellites would be to be able to image the Antarctic as RCM cannot do this because of the orbit and the positioning the C-band radar on the satellite. This was not a priority for Canada, as our country is near the Arctic. RADARSAT-1 could image the arctic, and it was one was one of its greatest feat. Albeit, it required a complex series of manoeuvres, but was possible. The reason it's important is since 2017, we've seen Icebergs detach from Antarctica, and it is important to monitor and measure this phenomenon. Canada needs to be and stay at the forefront with other space faring nations for Earth Observation, therefore, we're currently working on the next generation of SAR satellites.

Appendix

Acronyms/Abbreviations

AIS	Automatic Identification System
AOI	Area of Interest
ARS	Antenna Reservation System
CCD	Coherent Change Detection
CE	Central Electronics
CES	Central Electronics Subsystem
CSA	Canadian Space Agency
DINSAR	Differential Interferometric SAR
EDOT	Enhanced Definitive Orbit Tool
EO	Earth Observation
FD	Flight Dynamics
FDIR	Failure Detection, Isolation and Recovery
GoC	Government of Canada
IQ	SImage Quality Subsystem
ITO	SIntegrated Test and Operations System
LEO	Low Earth Orbit
MMU	Mass Memory Unit
PSK	Phase-shift keying
PDU	Power Distribution Unit
PLCU	Payload Control Unit
ODMP	Orbit Determination and Manoeuvres Planning
RCM	Radarsat Constellation Mission
RF	Radio Frequency
SAR	Synthetic Aperture Radar
MCC	SHUBST-Hubert
MET	Mission Elapsed Time
TA	Transfer Agent
TF	Thermal Fatigue
TLE	Two Line Element
TRM	Transmit Receive Modules
VAX	Virtual Address eXtension
VMS	Virtual Memory System
STK	Satellite Tool Kit

Psyche Mission's End-to-End Information System Architecture: Advantages, Challenges, and Operability

Richa Sirohi, Robert R. Moore, Lloyd R. Deforrest, Marla S. Thornton, Kristina L. Larson, Daniel D. Wenkert, and Greg J. Kazz

Abstract The Psyche Mission is a mission to the asteroid "(16) Psyche", featuring three science instruments and gravity science. (16) Psyche, located in the asteroid belt between Mars and Jupiter, will be the first potential metal world—instead of rock or ice—visited by the National Aeronautics and Space Administration (NASA). This Jet Propulsion Laboratory (JPL) managed mission will explore Psyche for 21 months after an earliest launch date of August 2022 and a 3.5 year cruise. In this paper, the End-to-End Information System's (EEIS) concept, architecture, and the Consultative Committee for Space Data Systems (CCSDS) standards implementation of Psyche are studied and analyzed for fulfilment of mission requirements and for satisfaction of operational constraints. The EEIS is a virtual system comprising distributed data system functions through the subsystems. The system is defined by how the flight, mission, and launch systems work together to enable Psyche's data flows (uplink, downlink, spacecraft, and ground), as well as validate, account for, process, distribute, and store Psyche's data. This data includes spacecraft commands, spacecraft health, and instrument science data. EEIS Engineers are specifically responsible for the

R. Sirohi (✉) · R. R. Moore · L. R. Deforrest · M. S. Thornton · K. L. Larson · D. D. Wenkert · G. J. Kazz
NASA Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr, Pasadena, CA 91109, USA
e-mail: richa.sirohi@jpl.nasa.gov

R. R. Moore
e-mail: robert.r.moore@jpl.nasa.gov

L. R. Deforrest
e-mail: lloyd.r.deforrest@jpl.nasa.gov

M. S. Thornton
e-mail: marla.s.thornton@jpl.nasa.gov

K. L. Larson
e-mail: kristina.l.larson@jpl.nasa.gov

D. D. Wenkert
e-mail: daniel.wenkert@jpl.nasa.gov

G. J. Kazz
e-mail: greg.j.kazz@jpl.nasa.gov

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_5

concept formulation of the information system, its design architecture, as well as implementing the CCSDS standards in the flight to ground interface from a high-level project system engineering perspective. This includes evaluating the mission's operational constraints and requirements, as well as inherited mission infrastructure. Criteria for evaluating mission information include the quantity, quality, latency, and continuity (QQCL) of the data. In this paper, the Psyche EEIS will be evaluated in relation to these four criteria. Thus, this paper focuses on: the driving EEIS design requirements based on program, project, science, and operability requirements; the views and analysis of the EEIS conceptual design; the CCSDS standards implementation; and the EEIS layered architecture comprising its data flows, flight assets, mission operations system (MOS), ground/science data systems (SDS), and multi-mission services. The possible advantages and limitations of the Psyche EEIS architecture and suggestions for future space missions are also discussed.

Keywords Psyche · CCSDS · EEIS · Communications · Architecture · Standards · Operability

Acronyms/Abbreviations

AMMOS	Advanced Multi-Mission Operations Systems
AMPCS	AMMOS Mission Data Processing and Control System
AOS	Advanced Orbiting Systems
API	Application Programming Interface
APID	Application Identifier
APL	Applied Physics Laboratory
ASU	Arizona State University
ATLO	Assembly, Test, and Launch Operations
AVS	Avionics
BPS	Bits per Second
C&DH	Command & Data Handling
CCSDS	Consultative Committee for Space Data Systems
CFDP	CCSDS File Delivery Protocol
CLTU	Command Link Transmission Unit
CMD	Command
CRC	Cyclic Redundancy Check
D/L	Downlink
DCD	Data Capture and Delivery Subsystem
DEA	Digital Electronics Assembly
DMTK	Data Management Toolkit
DP	Data Product
DSCC	Deep Space Communications Complex
DSN	Deep Space Network
DSOC	Deep Space Optical Communications

DSOpC	JPL Deep Space Operations Center
DSS	Deep Space Station
DTT	Downlink Tracking & Telemetry
DTU	Denmark Technical University
ECC	Emergency Control Center (DSN)
ECR	Engineering Change Request
EEIS	End-to-End Information System
EH&A	Engineering, Housekeeping, & Accountability
EOF	End-of-File
ESA	European Space Agency
EVR	Event Report
FCPL	Flight Core Product Line
FCTLU	Forward CLTU
FEI	File Exchange Server
FP	Fault Protection
FS	Flight System
FSTB	Flight System Testbed
FSW	Flight Software
GDS	Ground Data System
GIF	Ground Interface Facility
GLR	Ground Laser Receiver for DSOC
GLT	Ground Laser Transmitter for DSOC
GNC	Guidance, Navigation, and Control
GRNS	Gamma-Ray & Neutron Spectrometer
GSE	Ground Support Equipment
HGA	High Gain Antenna
INS	Instrument
IP	Internet Protocol
ISRO	Indian Space Research Organization
JAXA	Japanese Aerospace Exploration Agency
JHU	Johns' Hopkins University
JPL	Jet Propulsion Laboratory
KASI	Korea Astronomy and Space Science institute
LGA	Low Gain Antenna
LV	Launch Vehicle
MAG	Magnetometer
MDNav	Mission Design and Navigation
MDS	Monitor Data Server
MGSS	Multi-Mission Ground Systems and Services
MIT	Massachusetts Institute of Technology
MOS	Mission Operations System
MS	Mission System
MSA	Mission Support Area
MSL	Mars Science Laboratory
NASA	National Aeronautics and Space Administration

NVM	Non-volatile memory
OpNav	Optical Navigation
PB	Product Builder
PCE	Psyche Compute Element
PDA	Product Distribution Assembly
PDS	Planetary Data System
PDU	Product Data Unit
PI	Principal Investigator
PLD	Payload
PLOP	Physical Layer Operations Procedure
POCC	Project Operations Control Center
QQCL	Quantity, Quality, Latency, and Continuity
RAF	Return All Frames
RCF	Return Channel Frames
S/C	Spacecraft
SCET	Spacecraft Event Time
SCID	Spacecraft Identifier
SCLK	Spacecraft Clock
SDC	Science Data Center
SDS	Science Data System
SDST	Small Deep Space Transponder
SEM	Stationary Electronics Module
SEP	Solar Electric Propulsion
SFG	Special Function Gateway
SLE	Space Link Extension
SMAP	Soil Moisture Active Passive
SPK	SPICE Kernel
TCP	Transmission Control Protocol
TIF	Telemetry Interface
TLM	Telemetry
TZ	T-Zero
U/L	Uplink
V&V	Verification & Validation
VC	Virtual Channel
VCID	Virtual Channel Identifier
WSTS	Workstation Test Set

1 Introduction

1.1 Psyche Project & Mission Objectives

The Psyche Mission is a journey to a metal world, a large asteroid known as (16) Psyche. One of the largest in the solar system, (16) Psyche is a potential M-type asteroid and remnant nickel–iron core of a once rocky body [1]. The asteroid orbits the Sun between 2.5–3.3 astronomical unit (AU). After an earliest launch date in August 2022 aboard a SpaceX Falcon Heavy launch vehicle (LV), and a 3.5 year cruise with a Mars gravity assist, the flight system (FS) will explore the metal world for 21 months. Figure 1 below depicts the mission timeline for this Discovery class mission [2]. Over four science orbits, the Psyche mission will conduct observations using four instruments—a magnetometer, gamma ray and neutron spectrometers,

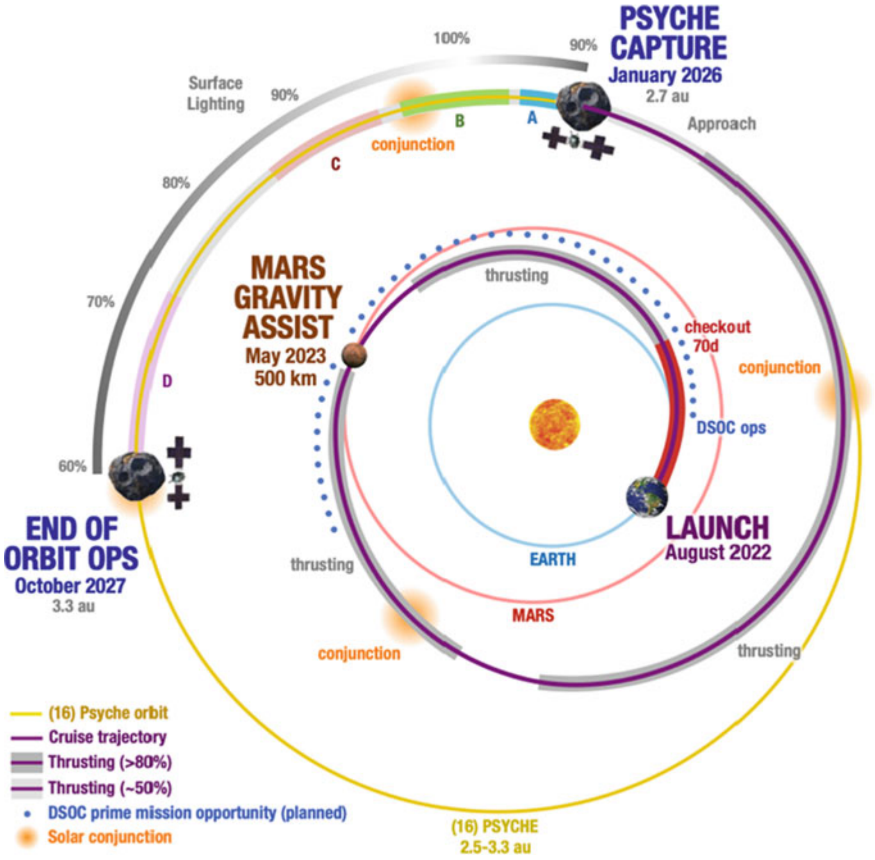


Fig. 1 Psyche mission overview

redundant multispectral imagers, and gravity science (using a X-band high-gain antenna [HGA] and three low-gain antennas [LGA]).

The Psyche project is organized into a Science System, Flight System (includes Spacecraft [S/C] and Payload [PLD]), a Mission System (MS)—including the MOS, Ground Data System (GDS), Mission Design and Navigation (MDNav), and the Science Data Center (SDC) at Arizona State University (ASU)). Overall, the project comprises multiple organizations including: NASA-JPL provides the project management, elements of the power, avionics (AVS), and telecommunication subsystems, integration and testing, and mission operations; ASU provides the Principal Investigator (PI: Dr. Lindy Elkins-Tanton) and Deputy PI (Dr. Jim Bell), the multispectral imager and the science data center; Maxar provides the Solar Electric Propulsion (SEP) chassis; Johns' Hopkins University-Applied Physics Laboratory (JHU-APL) provides the Gamma-Ray and Neutron Spectrometer (GRNS); Massachusetts Institute of Technology (MIT) provides gravity science and magnetometer science; Denmark Technical University (DTU) provides the magnetometer; Malin Space Systems is the Imager instrument vendor; NASA and SpaceX provide the Launch Services and the Falcon Heavy Launch Vehicle; and NASA-JPL provides the Deep Space Optical Communications (DSOC) technology demonstration [3, 4]. DSOC is a demo and not part of the core mission objectives, but will prove the use of optical communications from deep space spacecraft. The Psyche spacecraft also carries a 2.0 m fixed Cassegrain high-gain antenna and three low-gain antennas for nearly full sky coverage.

If (16) Psyche is the exposed metal core of a once rocky body, whose exterior has been stripped away by collisions and impacts, then it presents a rare and unique instance of a planetary core that can be observed. The following are the Psyche Project's five science objectives:

1. Determine whether Psyche is a core, or if it is primordial un-melted material.
2. Determine the relative ages of Psyche's surface.
3. Determine whether small metal bodies incorporate the same light elements into the metal phase as are expected in the Earth's high-pressure core.
4. Determine whether Psyche was formed under conditions more oxidizing or more reducing than Earth's core.
5. Characterize Psyche's topography.

1.2 End-to-End Information System Engineering

An End-to-End Information System (EEIS) is the set of functions that are distributed across the flight system, launch vehicle, mission system and science system, that interoperate cooperatively to collect, transport, store, translate, integrate, and manage mission (e.g., science, engineering, radio metric, command, ancillary) information. These functions are performed cooperatively by flight and ground elements to achieve mission objectives. Thus, the EEIS is a cross-cutting function of project systems engineering.

Data Quality, Quantity, Continuity, and Latency (QQCL) offers EEIS engineering with four prime metrics for both specifying requirements and evaluating the performance of the data accountability of the project's EEIS [5]. This paper will evaluate the Psyche EEIS in terms of its QQCL metrics described below. Table 1 includes a few key Psyche requirements in relation to each EEIS metric.

Quality: Metric speaks to how "good" or "bad" the telemetry is in terms of errors introduced into the telemetry once it is generated by the source (instrument or engineering subsystem) on board the spacecraft. This metric is specified in terms of a transfer frame error rate.

Quantity: Metric speaks to the volume of data telemetered by the spacecraft to the destination(s). High-level science objectives, along with the overall data loss characteristics of the EEIS, determine the total percentage of data expected to be received by the ground segment. This metric is specified in Tbits, Gbits, or Mbits depending on the EEIS performance characteristics.

Continuity: Metric speaks to the tolerance of the EEIS to the size and frequency of losses, i.e., data gaps in the telemetry. When the error correcting capability of the code is exceeded in the channel (i.e., telemetry point), these transfer frames become undecodable and result in erasures, i.e., gaps in the decoded telemetry stream. Data applications on the ground may be sensitive to gaps in their data streams, making it difficult to reconstruct their data products (DPs) or to do engineering trend analysis. The EEIS Engineer specifies data continuity requirements as part of the End-to-End accountability design.

Latency: Speaks to the amount of time a user must wait before receipt of the data. Depending upon the needs of the MOS and PIs, the EEIS Engineer specifies data latency requirements such that the expectations of the end users of the EEIS will be designed into the system and tested. For example, bandwidth from the ground stations for the project is limited. Data latency requirements are needed to ensure that data will be received by the end users based upon negotiated pre-agreements.

Table 1 Key QQCL requirements on Psyche

Metric	Key Psyche Requirements
Quality	• Return at least 95% of science and ancillary data
Quantity	• MAG, GRNS, INS HK data at constant 3 kbps through mission • SC Housekeeping (HK) downlink ~1 kbps • Onboard storage of 1024 Gb at beginning of life and 614 Gb (check 611,230) for end of life • Psyche shall return at least 806 Gb of science data
Continuity	• Downlink frame gap rate < 1E-4 between flight system transmitter and ground station receiver • Uplink frame error rate (FER) < 1E-4 as measured between the ground station transmitter and the output of the Flight System command decoder
Latency	• MS capable of delivering DSOC FLT Telemetry to DSOC MOS with latency of 2 min

1.3 Heritage

Psyche's Mission System draws heavily from the Dawn mission [6]. The concept of operations as an electric propulsion mission, which conducts non-discovery-responsive systematic mapping of our target, is derived from Dawn. The mission design and navigation processes and tools trace their heritage directly to Dawn as well. The GDS leverages modern Multi-mission Ground Systems and Services (MGSS) hardware and software architectures used by Mars Science Lab (MSL) [7], Soil Moisture Active Passive (SMAP) [8], InSight [9], Mars2020 Perseverance Rover [10], and Europa Clipper [11], and also incorporates the new Flight Core Product Line (FCPL) Flight Software (FSW) [12]. Psyche's core science data processing and distribution are conducted at the highly experienced ASU SDC. Maxar's experience in checking out and conducting initial operations of its systems is infused via its side-by-side participation in Initial Checkout; Maxar personnel, thereafter, have access to telemetry and are on call for one work year of support throughout the rest of the mission for diagnosis and anomaly resolution.

The Psyche FSW architecture adheres to the JPL FCPL architecture. The adoption of FCPL is part of an institutional need for a strategic and cohesive approach to FSW development but also presents a challenge for Psyche as one of the first projects to adapt and implement FCPL. The Psyche FSW architecture is a Space and Time Partitioned architecture since that is the FCPL baseline.

Psyche's Flight Ground Interface Control Document (FGICD) and tailored CCSDS File Delivery Protocol (CFDP) Class 1 design are based on SMAP's architecture. Psyche's GDS is utilizing SMAP's Advanced Multi-Mission Operations Systems (AMMOS) Mission Data Processing and Control System (AMPCS) Product Builder [13], which is a limited implementation of Class 1 CFDP, instead of the standard CFDP engine, which implements CFDP class 1 and 2. The use of SMAP's AMPCS Product Builder saves the Psyche project both time and money by using existing tools. The AMPCS Product Builder behaves similarly to CFDP Class 1 and is used to verify data product checksums and file sizes, thus, verifying whether the complete product was received on the ground. Psyche also retains the command structure as well as the Engineering, Housekeeping, & Accountability (EH&A) and Event Report (EVR) telemetry packet structure of previous JPL missions to fully leverage available AMPCS tools. Note, EVRs are text-based status messages and a type of telemetry.

1.4 Key Architectural Drivers

Key drivers of the EEIS architecture are based on characteristics of the Psyche mission that drive its implementation and design as well as new technologies, processes, software, operational constraints, etc. that must be accommodated in Psyche. Some of these drivers are described below.

Discovery Class Mission: Psyche is a Class B, Discovery mission which commands several NASA requirements to be placed on the mission in order to meet Class B flight projects standards. As a Discovery Class Mission, Psyche must advance knowledge and exploration in our Solar System, add scientific data and other products to the Planetary Data System (PDS) archive for all scientists to access, announce scientific progress and results in peer-reviewed literature and media, expand the pool of well-qualified PIs and Project Managers for implementation, and implement technology advancements proven in related programs.

Dual-String Command & Data Handling (C&DH): This impacts the amount of telemetry Psyche will produce. The Psyche Compute Element (PCE) pair is a dual redundant system in the avionics that implements the C&DH processing functions onboard the spacecraft. Each PCE processes a "string" of data. The "prime" communicates with the ground and can read data from the other "online" string. In the event of a fatal error in the "prime" string, the "online" string would become the "prime" string. The prime PCE can also telemeter data concurrently from redundant hardware on the single-fault tolerant spacecraft.

Space Link Extension (SLE): The Deep Space Network (DSN) uses CCSDS SLE to communicate with non-JPL mission GDSs. Prior to the Psyche mission, JPL missions were excepted from using SLE due to ready access to JPL internal forward and return services [14]. Starting with Psyche and with every new JPL mission moving forward, JPL missions must use CCSDS SLE forward and return clients to communicate with the DSN. The reason is mostly to adopt international standards (i.e., CCSDS) used by all non-JPL operated missions, not only by organizations such as the European Space Agency (ESA), Indian Space Research Organization (ISRO), Japanese Aerospace Exploration Agency (JAXA), Korea Astronomy and Space Science Institute (KASI), but also missions operated by other NASA Centers (i.e., Goddard Space Flight Center). The DSN's support of the DSN Telemetry Service using the CCSDS SLE protocols enables Psyche to contribute to the multi-mission reuse of multi-mission ground data system infrastructure software at JPL. Since Psyche is the first JPL mission to use SLE to communicate with the DSN, Psyche GDS has been the pathfinder for adaptation testing of multi-mission SLE client software.

Flight Core Product Line (FCPL): Psyche is required to adopt and customize the JPL FCPL, a time and space partitioned flight software, which will fly for the first time on Psyche. This may impact the implementation of ground software and operability. While the project has prioritized managing this implementation risk, and the Verification & Validation (V&V) campaigns will test functionality of the FSW, there will be inevitable features of the software that will only be discovered in flight. Also, Psyche will be using the FCPL Sequence Module developed by the Europa Clipper mission. The Sequence module will provide sequencing needs required by Psyche along with additional features that were not planned for use—some of which are even undesirable—on Psyche. WorkStation Test Set (WSTS) is a simulation tool that combines both flight and ground software at the same workstation. WSTS workstations will be utilized by the Psyche team to test FSW prior to running tests and commands on the Flight System Testbed (FSTB). Typically, WSTS runs faster than real-time while system testbeds run at real-time. However, the process of making WSTS compatible with the FCPL resulted in WSTS only being able to run at real-time for Psyche. This places a burden on testers to test commands and develop procedures efficiently and forces them to make tested sequences less flight-like to meet the allotted test times.

Deep Space Optical Communications (DSOC): Psyche is accommodating DSOC, an optical communications technology demonstration provided by NASA. DSOC operations largely take place early in Psyche's cruise phase, which will demand on the mission system teams. Additionally, because of the accuracy required for the DSOC ground station knowledge of Psyche's location, the navigation team will be supporting a faster ephemeris update schedule than what would otherwise be required. Also, DSOC must reserve time at the Palomar ground telescope significantly before their opportunities, which forces the early Psyche mission events schedule to depend on the launch date.

Multi-Organization Built Spacecraft: The Psyche mission is a multi-organization project. The result of having multiple institutions and parties contribute is that interfaces become more difficult to manage and engineering responsibilities that were traditionally done within JPL's purview have become distributed. While this distributed system of project engineering has its advantages, it also poses challenges as it's a first for a JPL mission of this size. In Psyche's case, it is new to have flight system built by over one organization. Typically, the flight system is completely handled by a vendor or completely by JPL (with use of sub-contractors, of course). However, on Psyche, the Flight System is a hybrid build. Maxar builds the SEP Chassis which includes the Spacecraft structure, high-gain antenna, sensors and actuators for guidance, navigation and control (GNC), electric propulsion subsystem, thermal subsystem, part of the AVS subsystem, and a majority of the power subsystem, while JPL integrates a majority of the telecom subsystem, half of the AVS subsystem, and the instruments, prior to conducting system testing at JPL.

2 The End-to-End Information System

2.1 EEIS Data Flow

The data flow in Fig. 2 depicts how and what type of data is shared through the Psyche systems. The diagram is a high-level view and does not illustrate data structure or all nodes in the data transfers. The purpose is to convey the functionality and high-level architecture of the EEIS.

As illustrated in Fig. 2, the mission system handles commanding and telemetry and interfaces with the SDS, which distributes data to the instrument teams, science data center, and planetary data system. DSOC, a technology demonstration, provides

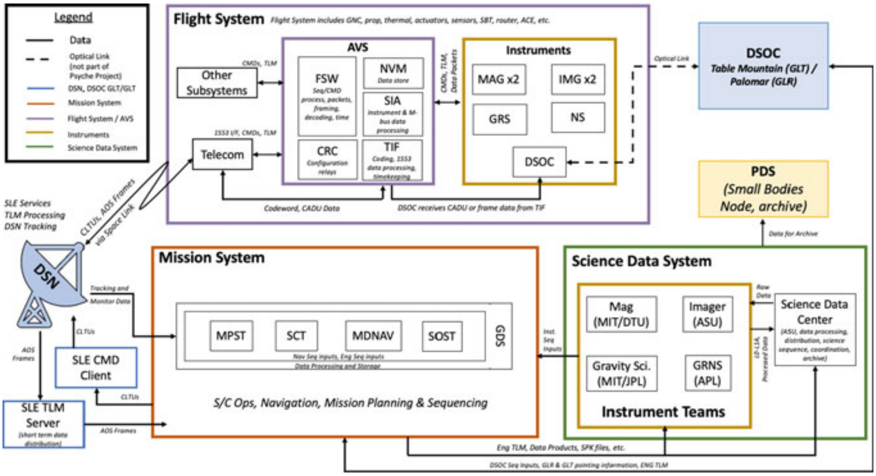

Legend
Data
Optical Link (not part of Psyche Project)
DSN, DSOC GLT/GLT
Mission System
Flight System / AVS
Instruments
Science Data System
Flight System
Flight System includes GNC, prop, thermal, actuators, sensors, SBT, router, ACE, etc.
AVS
Other Subsystems
CMDs, TLM
FSW
Seq/CMD process, packets, framing, decoding, time
NVM
Data store
SIA
Instrument & M-bus data processing
CRC
Configuration relays
TIF
Coding, 1553 data processing, timekeeping
1553 I/F, CMDs, TLM
Telecom
Codeword, CADU Data
DSOC receives CADU or frame data from TIF
CMDs, TLM, Data Packets
Instruments
MAG x2
IMG x2
GRS
NS
DSOC
Optical Link
DSOC
Table Mountain (GLT) / Palomar (GLR)
PDS
(Small Bodies Node, archive)
Data for Archive
SLE Services
TLM Processing
DSN Tracking
CLTUs, AOS Frames via Space Link
DSN
Tracking and Monitor Data
CLTUs
AOS Frames
SLE CMD Client
CLTUs
SLE TLM Server
(short term data distribution)
AOS Frames
Mission System
MPST
SCT
MDNAV
SOST
GDS
Nav Seq inputs, Eng Seq inputs
Data Processing and Storage
S/C Ops, Navigation, Mission Planning & Sequencing
Inst. Seq Inputs
Science Data System
Mag (MIT/DTU)
Imager (ASU)
Gravity Sci. (MIT/JPL)
GRNS (APL)
Instrument Teams
Raw Data
L0-L1A, Processed Data
Science Data Center
(ASU, data processing, distribution, science sequence, coordination, archive)
Eng TLM, Data Products, SPK files, etc.
DSOC Seq Inputs, GLR & GLT pointing information, ENG TLM

Fig. 2 Overall EEIS data flow

Psyche with an additional communication pathway to send data directly to the Palomar receiver and receive commands from the Table Mountain transmitter.

2.2 Payload

The Psyche payload includes two imagers for science and optical navigation, a magnetometer with two sensor heads, a Gamma-Ray and Neutron Spectrometer (GRNS) comprising separate GRS and NS units for measurements of the elemental composition of the first meter of (16) Psyche's surface, and DSOC which is a technology demonstration. DSOC will communicate at ranges up to 2.7 AU from Earth. Not every downlink opportunity will utilize this optical communication downlink capability. Gravity science will be achieved using the DSN's two-way X-band Doppler and ranging measurements. The DSN transmits an X-Band uplink (7.2 GHz) signal to the spacecraft receiver where the signal is turned around and the amplified coherent-downlink (8.4 GHz) is transmitted back to the DSN station.

2.3 Flight System

Telecommunications: The Psyche telecommunications system is designed to return science data at 150 kbps at up to 4.0 AU from the Earth, support navigation during cruise and asteroid operations, and support gravity science at the asteroid via low-gain antennas. The flight system carries a 2.0 m fixed Cassegrain high-gain antenna and three low gain antennas with nearly full sky coverage.

C&DH: The primary processing unit for the Psyche spacecraft is the PCE. The two PCEs serve as a pair of redundant computer hosts for JPL FSW. Each PCE runs flight software on a RAD750 processor and provides all radio uplink and downlink interfaces. It receives commands from the telecom subsystem, formats and forwards them to the spacecraft hardware for execution, and it collects and stores engineering and science data in non-volatile flash memory. The PCE formats and forwards file-based telemetry to the telecom system for downlink. The Psyche flight software is a variant of the JPL FCPL software. The rest of the Psyche C&DH subsystem is provided by Maxar. The Maxar subsystem provides command and telemetry interfaces to all the subsystems. It supports both a MIL-STD-1553 bus for communications with the radios, DSOC, star trackers, PDAs, and inertial measurement units, as well as a set of RS-485 buses for communications with Attitude Control Electronics (ACE), thermal control systems, pyros, solar array drives, power management and power distribution units, SEP systems, and the Smart Battery Tray (SBT). All command and telemetry paths are cross-strapped and redundant. There are two main types of data: real-time engineering and recorded data products (only recorded data is in non-volatile memory, NVM). For data that controls the spacecraft, there are system parameters (NPM), shared records (REU SRAM), and ground commands which include uplinked files, hardware, and immediate commands.

2.4 Mission System

The Mission System comprises several functions and subsystems, including Mission System Engineering, the SDC, the Mission Operation System (MOS), and the Ground Data System (GDS). These then interact with the Science Team, Planetary Data System (PDS), Launch Services, and the ground stations (DSN, DSOC MOS, Palomar/Table Mountain). The responsibilities and data shared between all these subsystems are described below in Fig. 3.

Psyche leverages NASA's AMMOS tools and services as the backbone of the GDS. AMMOS provides uplink and downlink processes including telemetry processing, storage, distribution, and display; tracking and navigation processing, command processing, sequencing, science planning, file storage, and ancillary data (e.g., SPICE kernels) sufficient to meet project needs for development, testing, and operations. Psyche also leverages the AMMOS AMPCS to process and analyze spacecraft and payload telemetry data. Figure 4 below illustrates the GDS architecture on Psyche and the data flow among them.

The GDS software development is managed by JPL and all ground software is categorized by mission criticality and follows JPL institutional guidelines. Complete software functionality testing is performed at the program and subsystem level. After

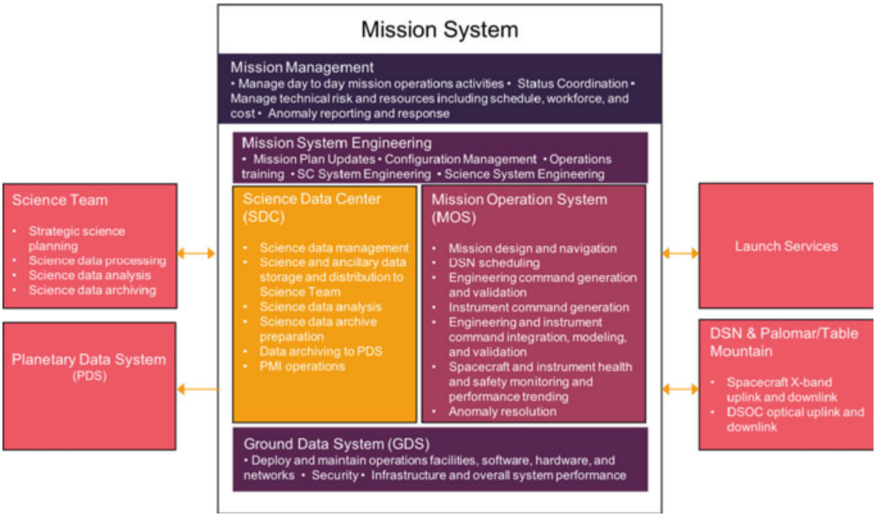

Mission System
Mission Management
• Manage day to day mission operations activities • Status Coordination • Manage technical risk and resources including schedule, workforce, and cost • Anomaly reporting and response
Mission System Engineering
• Mission Plan Updates • Configuration Management • Operations training • SC System Engineering • Science System Engineering
Science Team
• Strategic science planning
• Science data processing
• Science data analysis
• Science data archiving
Planetary Data System (PDS)
Science Data Center (SDC)
• Science data management
• Science and ancillary data storage and distribution to Science Team
• Science data analysis
• Science data archive preparation
• Data archiving to PDS
• PMI operations
Mission Operation System (MOS)
• Mission design and navigation
• DSN scheduling
• Engineering command generation and validation
• Instrument command generation
• Engineering and instrument command integration, modeling, and validation
• Spacecraft and instrument health and safety monitoring and performance trending
• Anomaly resolution
Launch Services
DSN & Palomar/Table Mountain
• Spacecraft X-band uplink and downlink
• DSOC optical uplink and downlink
Ground Data System (GDS)
• Deploy and maintain operations facilities, software, hardware, and networks • Security • Infrastructure and overall system performance

Fig. 3 Psyche phase E mission system functional diagram

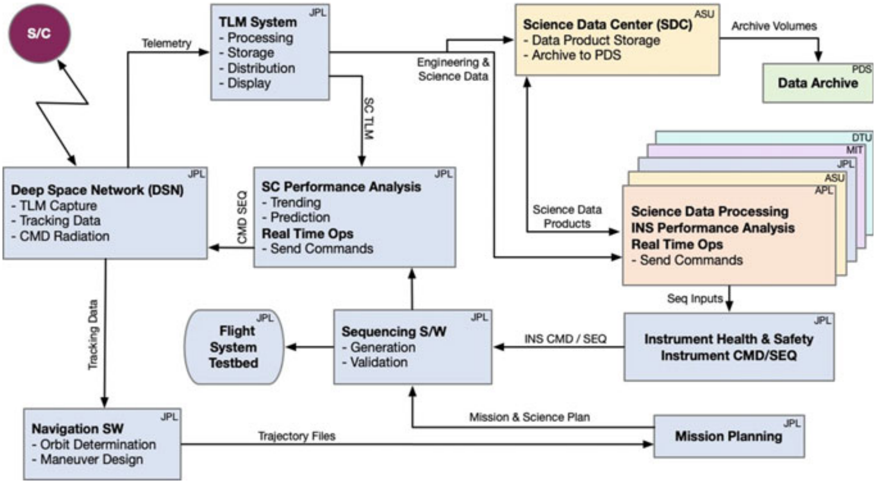


Fig. 4 GDS functions and data flow

GDS integration, sufficient software testing is performed to demonstrate system functionality. The GDS provides workstations and virtual machines to test flight software which each contain the WSTS environment, simulation support equipment (SSE) software (including environment simulators), and the telemetry and command system of GDS. The GDS also provides capabilities to the Psyche AVS testbeds and flight system testbed (FSTB). The testbeds require a simple emulator for the DSN, obtained from the Net Acquire Corporation.

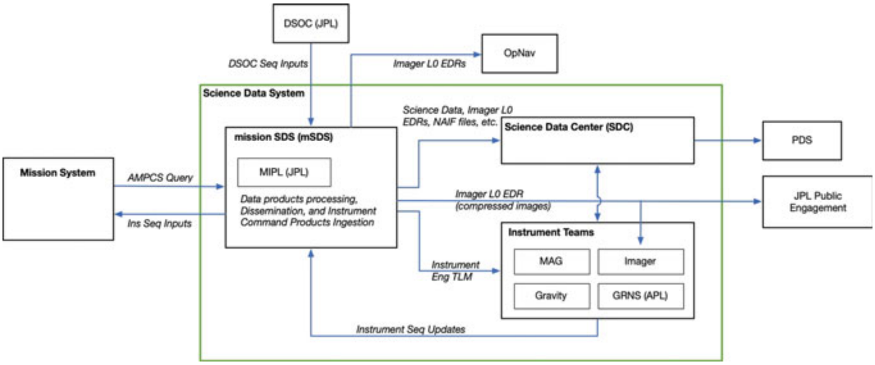


Fig. 5 Science data system

The **Science Data System** is illustrated in Fig. 5. The Psyche GDS at JPL provides four types of data for the science team:

(1) Science data packets from the instruments group into files;
(2) Image files from the Imager instrument;
(3) Radio science data (from the DSN); and
(4) Channelized engineering/housekeeping telemetry from the instruments and other spacecraft systems. Note, channelized telemetry is a telemetry point.

All of these data are acquired by the SDC at ASU for distribution to the instrument teams and the Psyche science team in general. Each instrument team and the gravity science team acquire all necessary input data from the SDC and process these into PDS4 Raw, Calibrated, and Derived products. These are then returned to the SDC for review, validation, and analysis by the general Psyche science team. After internal review, these data are packaged by the SDC into PDS4 bundles and collections and submitted to the PDS for external review and archiving. The wider community accesses all Psyche science data products from the PDS (specifically the Small Bodies Node) [15] (Fig. 6).

Table 2 shows the size of the Raw, Calibrated, and lower-level Derived data products Psyche anticipates archiving from the cruise and approach phases and from orbit at (16) Psyche (in GB). The total size of these products from cruise and approach phases is 1.2 TB and from orbital operations is 1.3 TB.

2.5 *Deep Space Network*

The Psyche mission will use the DSN. The DSN includes several facilities that support Psyche including the Goldstone, Canberra, and Madrid complexes with their 34 and 70 m antennas, the Deep Space Operations Center (DSOpC) at JPL, as well

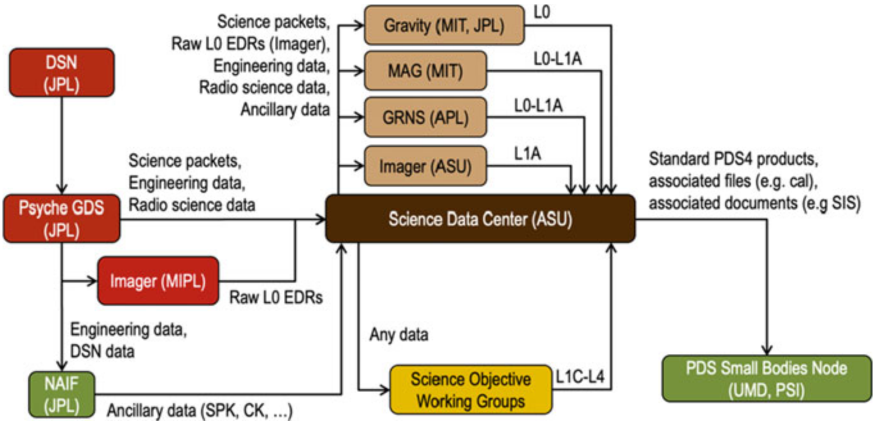

DSN (JPL)
Science packets, Raw L0 EDRs (Imager), Engineering data, Radio science data, Ancillary data
Gravity (MIT, JPL)
MAG (MIT)
GRNS (APL)
Imager (ASU)
L0
L0-L1A
L0-L1A
L1A
Science packets, Engineering data, Radio science data
Psyche GDS (JPL)
Science Data Center (ASU)
Standard PDS4 products, associated files (e.g. cal), associated documents (e.g SIS)
Imager (MIPL)
Raw L0 EDRs
Engineering data, DSN data
Any data
NAIF (JPL)
Ancillary data (SPK, CK, ...)
Science Objective Working Groups
L1C-L4
PDS Small Bodies Node (UMD, PSI)

Fig. 6 SDC data flow diagram

Table 2 Standard science data products (GB) archived in PDS

Instrument	Data product	Cruise/approach	Orbit
Magnetometer	Raw (L0) magnetic field components	404.04	165.77
	Calibrated (L1A) magnetic field time series	448.93	184.2
	Resampled (L1B) field—s/c signature removed	246.92	101.31
	Derived (L2) field in J2000, etc. coordinates	57.02	303.92
Imager	Raw (L0) binary images	6.5	156.3
	Calibrated (L1A) radiance images	13.1	312.6
GRNS	Raw (L0) time series of gamma ray counts	10.6	22.1
	Calibrated (L1A) time series of gamma ray flux	10.6	22.1
	Raw (L0) time series of raw NS counts	2.2	1.1
	Calibrated (L1A) time series of neutron flux	2.2	1.1
Gravity science	Raw (L0) radio metric tracking data	12.5	30.5
	Raw (L0) media calibration file	0.003	0.014
	Raw (L0) spacecraft mass history file	0.001	0.001
	Raw (L0) spacecraft small-forces file	0.005	0.012

as the DTF-21, CTT-22, and MIL-71 test facilities. The MOS is responsible for scheduling with the DSN.

The DSN 70 m antenna will be requested for safe-mode recovery. DSN 34 m arrays are an option for safe mode support as there will be a time when there are only 2 complexes with available 70 m dishes. In the event of a Psyche spacecraft emergency, the DSN will immediately coordinate release of DSN resources to support the emergency. Figure 7 illustrates how commands are sent and telemetry and monitor data are received from the DSN.

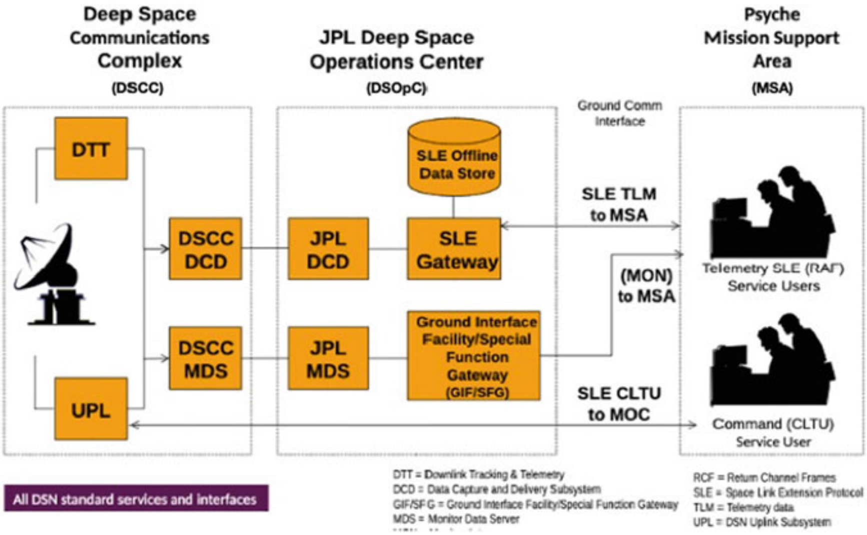

Deep Space Communications Complex (DSCC)
JPL Deep Space Operations Center (DSOpC)
Psyche Mission Support Area (MSA)
Ground Comm Interface
DTT
DSCC DCD
DSCC MDS
UPL
JPL DCD
JPL MDS
SLE Offline Data Store
SLE Gateway
Ground Interface Facility/Special Function Gateway (GIF/SFG)
SLE TLM to MSA
(MON) to MSA
SLE CLTU to MOC
Telemetry SLE (RAF) Service Users
Command (CLTU) Service User
All DSN standard services and interfaces
DTT = Downlink Tracking & Telemetry
DCD = Data Capture and Delivery Subsystem
GIF/SFG = Ground Interface Facility/Special Function Gateway
MDS = Monitor Data Server
RCF = Return Channel Frames
SLE = Space Link Extension Protocol
TLM = Telemetry data
UPL = DSN Uplink Subsystem

Fig. 7 DSN Psyche command (CMD), telemetry (TLM), and monitor data flow (physical view)

2.6 Launch System

The Launch Phase EEIS activities ensure that the flight and ground communication systems are ready to support the launch window (baselined for August 2022 at the earliest on a SpaceX Falcon Heavy LV) and post-separation activities. The launch phase begins with the final spacecraft power-on in the launch countdown and ends when two-way communication has been established following array deployment and safe mode entry.

The EEIS team is concerned with ensuring communications are established between the spacecraft and ground after separation. The first ground contact is expected through the 34 m DSN tracking station at Canberra to confirm the health of the flight system. Once the DSN acquires the downlink signal, the operations team will send a no-op command to test the uplink path and monitor telemetry to verify the command was received.

3 EEIS Architecture & CCSDS Implementation

3.1 EEIS Architecture

Figure 8 illustrates the EEIS architecture from a functional viewpoint [16] and illustrates the types of data shared at each interface within and between the flight and ground segments. The flight segment is depicted on the left separated by payload

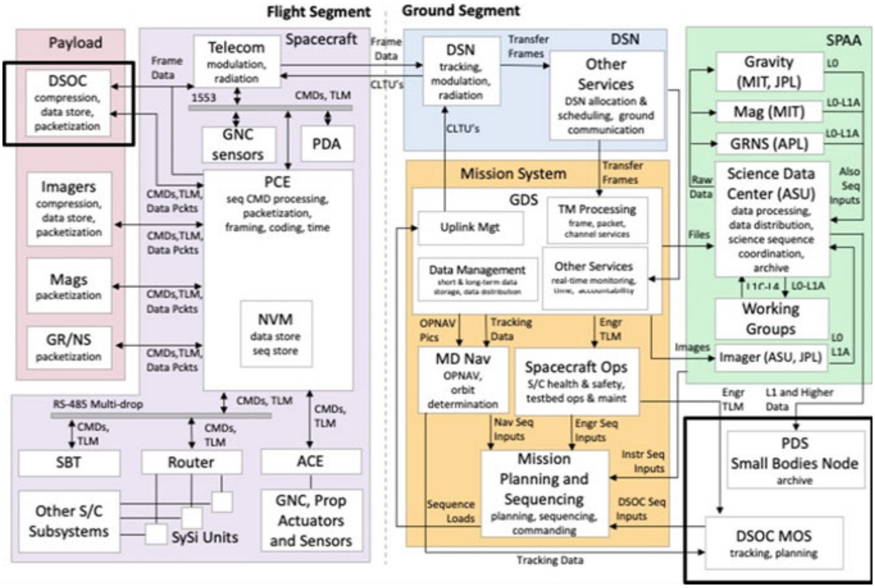


Fig. 8 Psyche EEIS functional viewpoint

and flight systems, while the ground segment is on the right, including the DSN, Mission System, SDS, and PDS. On the flight system, data interfaces between the PCE and subsystems are depicted, as well as the communication flow from flight to the DSN/ground. The communication path from DSOC to DSOC MOS is omitted from this diagram, as it is not the principal method of communication. For more information on the Transfer Frames sent to the ground and processed by the GDS, see the protocol stacks in Sect. 3.2.

3.2 Protocol Stack

The Psyche mission end-to-end protocol stack in Fig. 9 illustrates which communications protocols [16, 17] span system elements and the hierarchical nature of the protocol layers throughout the EEIS. All data link on Psyche uses a different combination of protocols including a subset of: payload and other flight system data formats, data bus protocols, space link and CFDP protocols, ground network protocols and use of SLE protocols, and data monitoring in the application layer.

Figure 10 illustrates the protocol stack used for return link communication in greater detail, focusing on how data flows and is processed from C&DH through the DSN ground station. Figure 11 illustrates how data then flows through the ground segment: DSN, GDS, and to the SDC/Instrument Teams. The data flows from left to right during downlink and is processed from bottom to top with the red steps being

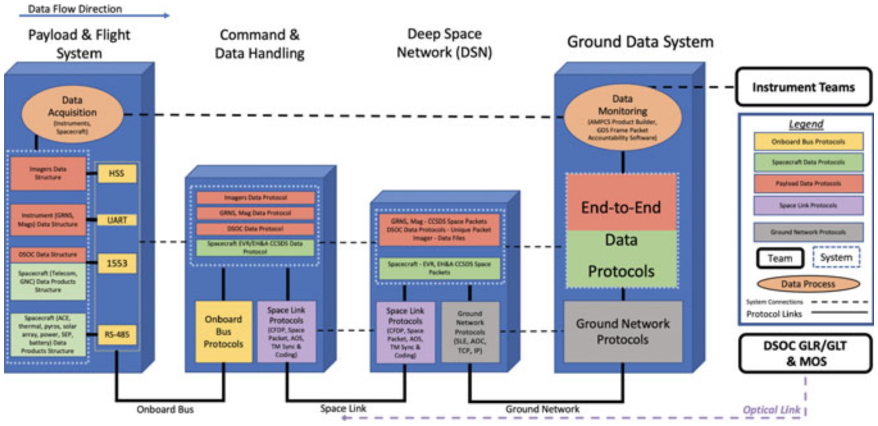
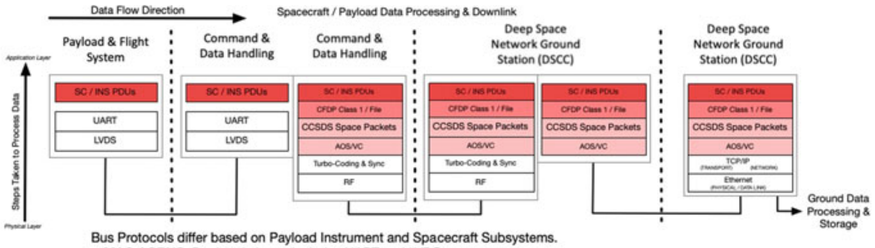
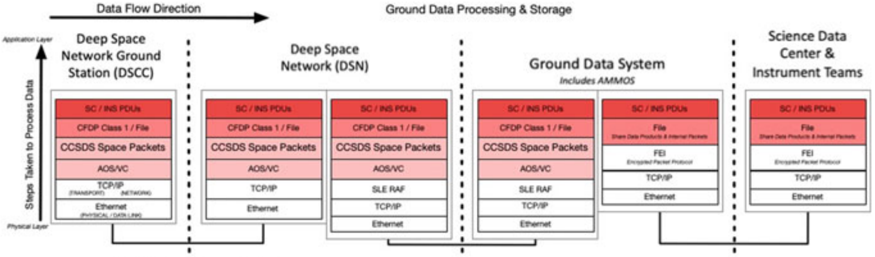

Data Flow Direction
Payload & Flight System
Command & Data Handling
Deep Space Network (DSN)
Ground Data System
Instrument Teams
Data Acquisition (Instruments, Spacecraft)
Imagers Data Structure
HSS
Instrument (GRNS, Mags) Data Structure
UART
DSOC Data Structure
1553
Spacecraft (Telecom, GNC) Data Products Structure
Spacecraft (ACE, thermal, pyros, solar array, power, SEP, battery) Data Products Structure
RS-485
Imagers Data Protocol
GRNS, Mag Data Protocol
DSOC Data Protocol
Spacecraft EVR/EH&A CCSDS Data Protocol
Onboard Bus Protocols
Space Link Protocols (CFDP, Space Packet, AOS, TM Sync & Coding)
GRNS, Mag - CCSDS Space Packets
DSOC Data Protocols - Unique Packet
Imager - Data Files
Spacecraft - EVR, EH&A CCSDS Space Packets
Space Link Protocols (CFDP, Space Packet, AOS, TM Sync & Coding)
Ground Network Protocols (SLE, AOC, TCP, IP)
Data Monitoring (AMPCS Product Builder, GDS Frame Packet Accountability Software)
End-to-End Data Protocols
Ground Network Protocols
Legend
Onboard Bus Protocols
Spacecraft Data Protocols
Payload Data Protocols
Space Link Protocols
Ground Network Protocols
Team
System
Data Process
System Connections
Protocol Links
DSOC GLR/GLT & MOS
Onboard Bus
Space Link
Ground Network
Optical Link

Fig. 9 General Psyche EEIS communications view (downlink)



Bus Protocols differ based on Payload Instrument and Spacecraft Subsystems.
- **MAG / GRNS:** Depicted above, includes UART and LVDS
- **Spacecraft** (GNC Sensors, Power, Thermal, SEP, etc.): RS-485
- **Spacecraft** (Telecom, GNC): 1553
- **DSOC:** 1553
- **Imagers:** HSS

Fig. 10 Flight system to ground segment communication protocols



SC / INS PDUs are defined as:
- **MAG/GRNS:** INS CCSDS Space Packets
- **Spacecraft:** CCSDS Space Packets
- **DSOC:** 1553 Bus Packets
- **Imagers:** N/A, stack only applicable up through Data Files

Fig. 11 Ground segment communication protocols

largely end-to-end protocols from the Payload to the SDC (the “redder” the layer, the more end-to-end the protocol is) [18–21]. For context, the Advanced Orbiting System (AOS)/Virtual Channel (VC) box is an example of the data link layer.

3.3 Uplink

Uplink data is received by the spacecraft via one of two radio interfaces (ports) or a T-Zero (TZ) wired interface for launch vehicle or ground support equipment (GSE) commanding. There are two X-Band SDSTs (Small Deep Space Transponders) connected to the two radio interfaces, serving as primary and redundant transceivers. They are cross-strapped to the PCE via their respective Telemetry Interface (TIF) uplink cards. The TZ interface is dedicated per PCE and functionally identical to that of an SDST, and can be used to receive an uplink data stream during testing and up until launch. The uplink communications protocols which interface with these spacecraft uplink interfaces are illustrated in Fig. 12.

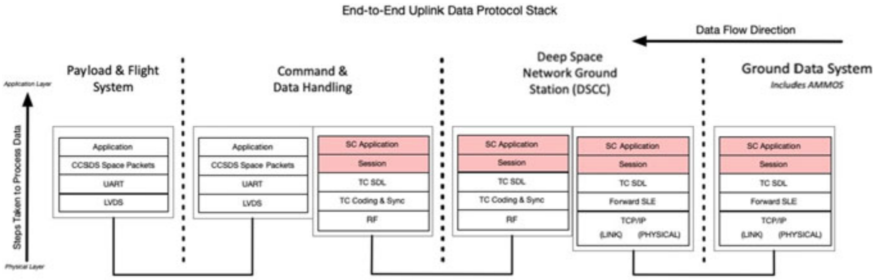
In the application layer, commanding is accomplished through the following types of uplink products.

Hardware Commands: Uplinked to the S/C for immediate execution upon receipt and validation. These commands fit within a single code word and do not require FSW for execution.

Immediate Commands: Uplinked to the spacecraft for immediate execution upon receipt and validation by the FSW.

File Loads: File loads are data uplinked to the spacecraft for the purpose of transferring data encapsulated in files to the FSW. A single File Load transfers exactly one data file to the FSW. Large file transfers require concatenating several file loads after they are uplinked.

Sequences: Sequences are a special type of file load. The commands within a sequence are dispatched in a specific order with specific timing. When sequences are uplinked, stored on-board, and executed immediately after validation with relative time-tags, the sequences are called load-and-go sequences.



Note: Protocols used between the C&DH and Payload & FS units differ by data bus type and unit. The above stack reflects the Mag and GRNS interfaces to the C&DH.

Fig. 12 End-to-end uplink data protocol stack

3.4 SLE Forward Command Link Transmission Unit (FCLTU) Service

The DSN provides a capability to radiate mission-supplied command data as a stream to a spacecraft. The SLE FCLTU service uses the Transmission Control Protocol (TCP)/Internet Protocol (IP) communications protocols to ensure reliable and error-free transmission of command data from the Project Operations Control Center (POCC) to the DSN Deep Space Station (DSS).

For FCLTU, the Psyche MOS submits CLTUs, in SLE Protocol Data Units (PDUs) [22–24]. MOS interacts with the DSN to: establish an association using TCP/IP; send annotated CLTUs to the DSN; obtain notifications and reports regarding status, configuration, and performance of the SLE; and temporarily stop and later re-start the sending of CLTUs. Production of the Forward CLTU service by the DSN entails processing the CLTUs transferred from the MOS through the necessary transformations to modulate the radio frequency carrier channel, providing uplink communications with the spacecraft. The DSN transmits the CLTUs in the order submitted from the MOS. The DSN may perform checks to determine if the CLTU complies with applicable constraints set by service management. However, the DSN does not interpret or modify the contents of a CLTU. The DSN does transmit idle sequence between CLTUs as needed per PLOP-2 (Physical Layer Operations Procedure).

The FCLTU service provides buffers for the CLTUs to maintain radiation of a steady stream of CLTUs in a ground communications environment of variable latency. FCLTU also supports emergency commanding of the Psyche spacecraft during periods when the communication lines between Psyche Ops and the DSS are down. This capability is pre-negotiated and solely used for placing the spacecraft in safe-mode.

3.5 Uplink Data Accountability

Psyche provides the following accountability features:

Accountability for uplink of commands is provided by EH&A counters and EVR messages.

Accountability for TC Transfer Frames is provided by an EH&A counter and EVR messages:

1. EH&A channel that records the number of TC frames that have failed their validity checks
2. EVRs are generated for codeblocks with uncorrectable errors
3. EVRs are generated when a frame fails validity checks, including valid FECF value, valid frame length, valid frame header bits, valid VCID, valid frame sequence number, and valid SCID.

For Hardware, Immediate, and Sequenced Commands, Psyche provides the following uplink accountability mechanisms in both real-time and recorded telemetry:

1. EH&A channels that count how many Immediate and Sequenced Commands have failed or been dispatched
2. EH&A channels for the last Hardware and Immediate Commands executed, listed by Opcode
3. EH&A channels that count Immediate, Hardware, and Sequenced Commands
4. EVRs are generated for Hardware Command execution success or error

For file loads, Psyche provides the following uplink accountability mechanisms:

1. FSW has the capability to perform a file cyclic redundancy check (CRC) check against the CRC stored with the file metadata
2. EVRs are generated reporting the success of file CRC checks
3. EVRs are generated when an uplinked file fails validity checks, including valid file type, valid file name length, valid file size
4. EH&A channel that counts successful file loads
5. A listing of all the files on the file system by file name prefix can be requested by the ground.

Psyche's decision not to implement a reliable file transfer protocol on the uplink stems from (1) limited resources for implementation and testing of a new protocol to JPL, (2) significant margin on the uplink telecom interface, and (3) concerns about delays in uplink operations at (16) Psyche because of the round-trip light time to the probe.

3.6 Data Management

In flight, channelized TLM is generated by FSW, and science data is generated by each instrument and stored on the NVM card. Real time telemetry data is continuously pushed to the telecommunications system, regardless of whether a communications pass is in progress, and is also stored. The flight system provides sufficient memory storage for 36 days of operations without requiring deletion of any data on board. Files in the file system are never deleted, except by ground command. It is a ground responsibility to monitor the telemetry indicating the available free space and delete files from the file system after confirmation of ground receipt so that the file system never fills up. The resolution and storage of both engineering and science telemetry are configurable by the ground. The FSW does not compress engineering or science data, but the Imagers and the Magnetometer do compress their data prior to sending it to FSW.

During downlink, the FSW creates transfer frames in priority order. The highest priority of telemetry is real-time EH&A telemetry, followed by real-time EVR telemetry, followed by data products. Recorded EH&A and EVR data are also placed in data products for later transmission to the ground. Each data product has a downlink priority associated with its Application ID (APID) at design time, which is stored as metadata in the data product catalog that tracks the data stored in flash memory. The data product catalog stores, for each product, its APID, size, downlink priority, the creation time for the product, whether the product has been sent to the ground, and transmission information on the data product parts. Within a priority level, the oldest unsent data products are downlinked first. The FSW repeatedly transmits telemetry, indicating the number of unsent data products and the data volume of recorded data products yet to be sent. Recorded data priorities can be set by ground command for individual data products or for all products within a given time range, but only for already-created data products.

Retransmission of files is implemented simply by commanding that a data product be marked as not having been sent, and allowing the downlink data management software to perform normally. Retransmission of individual parts of a data product may also be requested by the ground. The data product catalog, which maintains the state of all data products on-board the spacecraft, is stored in volatile memory. In the event of a reset or PCE side swap, the data product catalog will be reconstructed anew from file system metadata, with product priorities in their default state, and science products marked as sent. This ensures that spacecraft telemetry is transmitted to the ground as soon as possible after a reset. Science products' transmission status can subsequently be modified as part of the recovery.

3.7 On-Board Data Accountability

The only closed-loop commanding that takes place on the Flight System is to/from the instruments. For commands that result in Spacecraft intercom bus messages, the FSW considers "command completion" to be equivalent to command dispatch from the command or sequencing module to other FSW elements. The system does issue certain critical commands twice on intercom data buses, and the system does telemeter incomplete/failed data bus transactions. However, except for instrument commands, FSW commanding proceeds in an open-loop fashion when a fault occurs in command distribution. The Flight System relies on system fault protection (FP) to catch any state mis-comparison.

DSOC, the magnetometer, and GRNS are instruments which generally stream data to the PCE. If the Prime PCE cannot receive data because of a fault, most instrument data will be lost. Accountability of individual instrument packets is done through the packet sequence count field and timestamps within data products. The Imager is slightly different because it stores images for transfer to the NVM card. FSW keeps a record of all images on the Digital Electronics Assembly (DEA) storage and autonomously handles their transfer to the PCE.

When the file system reaches its maximum throughput, the rate that ground operators can command transmission of lossless data products may drop. FSW utilizes a buffer to transfer data to the file system, so if the buffer is being used to drain the most recently requested data, then a subsequent command to create a lossless product will fail because that resource is currently in use. Psyche does not anticipate any data losses by this effect since the Psyche Mission System has modeled data production requests to avoid losses as part of its sequence modelling process. Should a PCE swap occur, the new Prime PCE can list files and transfer files from the now online PCE.

3.8 Downlink

All data on the spacecraft is eventually packetized in CCSDS space packets. Each type of packet or product has a unique APID, which is a part of the CCSDS format for Space Packets.

The various data types are processed and queued for downlink differently. Figure 13 illustrates a simplified version of the process by which the data from on-board sources are selected, prioritized, and packetized before being sent to the telecommunications subsystem for transmission to Earth. For a more detailed look at the downlink data flow, see Fig. 14.

The two primary methods of downlink include real-time downlink and recorded data playback. In real-time downlink, the S/C continuously generates data in real-time that manifests as a live feed of EVRs and channelized telemetry (i.e., EH&A). Based on their APID, real-time EVRs and real-time EH&A are fed to packet buffers that ingest the data for downlink. The real time packet buffers are ring buffers, which can fill up and lose data if they are not drained as fast as they are filled. If data is overwritten, the oldest data in the ring buffer is overwritten. Therefore, the ground must configure EH&A selection criteria such that the generation is less than the ground-configured transmission rate. In the scenario where telemetry is generated faster than it can be transmitted, a counter (one per buffer) is incremented for each packet lost.

For recorded data playback, the process is only active when the ground commands it. The playback software begins a cyclic process of selecting the next data product to be downlinked, packetizing it, and then sending the resulting packets to the data products packet buffer. Data product selection is based on priority. Each APID has a built-in priority. Metadata about data products are stored in a data product catalog,

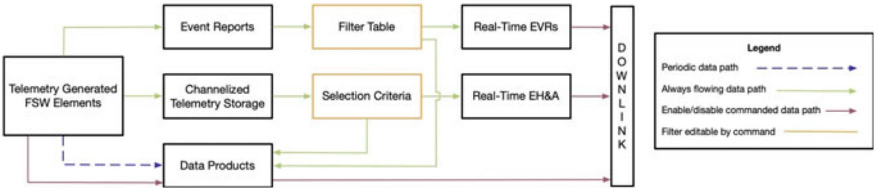


Fig. 13 High-level downlink management diagram

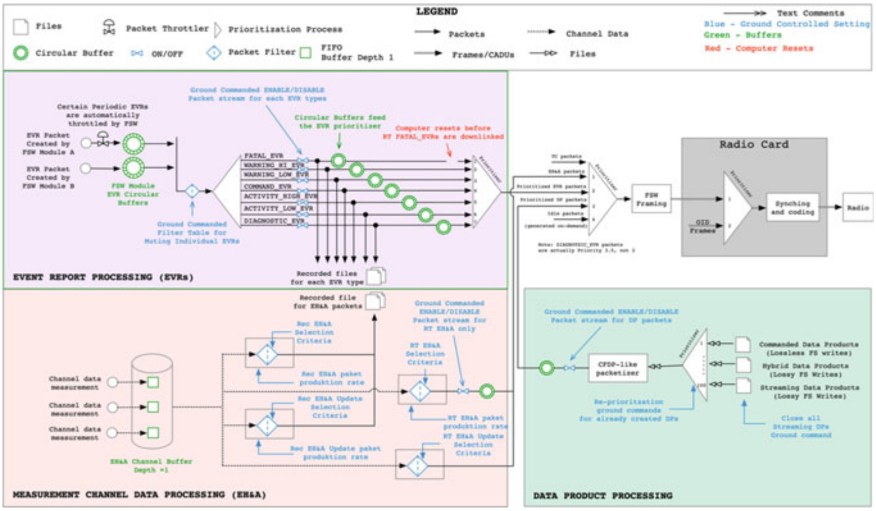

LEGEND
Files
Packet Throttler
Prioritization Process
Circular Buffer
ON/OFF
Packet Filter
FIFO Buffer Depth 1
Packets
Frames/CADUs
Channel Data
Files
Text Comments
Blue - Ground Controlled Setting
Green - Buffers
Red - Computer Resets
Ground Commanded ENABLE/DISABLE Packet stream for each EVR types
Certain Periodic EVRs are automatically throttled by FSW
EVR Packet Created by FSW Module A
EVR Packet Created by FSW Module B
FSW Module EVR Circular Buffers
Circular Buffers feed the EVR prioritizer
Computer resets before RT FATAL_EVRs are downlinked
FATAL_EVR
WARNING_HI_EVR
WARNING_LOW_EVR
COMMAND_EVR
ACTIVITY_HIGH_EVR
ACTIVITY_LOW_EVR
DIAGNOSTIC_EVR
Prioritizer
Ground Commanded Filter Table for Muting Individual EVRs
Recorded files for each EVR type
EVENT REPORT PROCESSING (EVRs)
TC packets
EH&A packets
Prioritized EVR packets
Prioritized DP packets
Idle packets (generated on-demand)
Prioritizer
Note: DIAGNOSTIC_EVR packets are actually Priority 3.5, not 3
FSW Framing
Radio Card
Prioritizer
OID Frames
Synching and coding
Radio
Recorded file for EH&A packets
Ground Commanded ENABLE/DISABLE Packet stream for RT EH&A only
Rec EH&A Selection Criteria
RT EH&A Selection Criteria
Rec EH&A paket produktion rate
Channel data measurement
Channel data measurement
Channel data measurement
Rec EH&A Update Selection Criteria
RT EH&A paket produktion rate
RT EH&A Update Selection Criteria
Rec EH&A Update paket produktion rate
EH&A Channel Buffer Depth =1
MEASUREMENT CHANNEL DATA PROCESSING (EH&A)
Ground Commanded ENABLE/DISABLE Packet stream for DP packets
CFDP-like packetizer
Prioritizer
Commanded Data Products (Lossless FS writes)
Hybrid Data Products (Lossy FS Writes)
Streaming Data Products (Lossy FS Writes)
Re-prioritization ground commands for already created DPs
Close all Streaming DPs Ground command
DATA PRODUCT PROCESSING

Fig. 14 Detailed downlink management diagram

which tracks, for each product, the APID to which it is assigned, its downlink priority, the creation time of the product, and whether or not the product has been sent to the ground. Within a priority level, the oldest unsent products are downlinked first. The FSW pushes telemetry, indicating the number of unsent data products and the number of bytes remaining to be sent.

There are four ways in which the ground can control the downlink priority of data products: (1) by command, set the priority of an already-created single data product, (2) by command, all existing products with a given APID and time range can be reprioritized, (3) by command, all existing products for all APIDs in a specified time range can be reprioritized, and (4) commands that cause the FSW to generate a data product have an argument to specify the priority of the product created.

Once selected, the data from the file is read from the file system. The data from the file is first wrapped in CFDP PDUs, then in CCSDS Space Packet headers. No additional error detection is undertaken in either step. The format of the Product Data Units (PDUs) is borrowed from the CFDP specification. The completed packets are then placed into the appropriate packet buffer, the data product is marked as having been sent into the data products catalog, and the process begins again for the next product.

If there is a file system error reading the file, file system fault protection may be invoked, and depending on the nature of the error and the response, the operation either succeeds with a warning (e.g., bit errors encountered), and playback continues, or system fault protection is invoked and playback is terminated by the system. Errors indicative of either file system or data catalog corruption (e.g., the requested file does not exist) will cause the playback FSW to send an EVR.

While no integrity checking is done on data read from the file system, the S/C does provide a means to correlate errors in data at rest as detected by the NAND hardware with downlinked data products. A file checksum is included in the EOF PDU for error control use by the ground. Additionally, the contents of certain data products contain checksums or similar data integrity verification mechanisms. For example, Space Packets produced by the instruments include error control fields which can be verified by the end users after being extracted from downlinked data products. This is the primary method of end-to-end error detection.

3.9 Retransmission

The ground can flag data products for retransmission at the file level and the data product "part" level. For Psyche, data product part means CFDP DPDU. One command allows for marking a single data product for retransmission by specifying the data product name, data product part range start, and data product part range end. A second command allows for marking multiple data products for retransmission via time and/or APID filters. This second command will mark the entire file for retransmission. The mechanism in FSW for marking files for retransmission is editing the data product catalog, which exists in volatile memory. In the event of a reset of the PCE or side swap to the redundant PCE, ground-commanded changes to data product priorities will be lost, as will indications of which products have been transmitted. The data product catalog will be rebuilt with default data product priorities and all "critical" products (specified in APID declaration in the APID dictionary) will be marked as not sent. All other products (non-critical) are marked as sent. Therefore, the ground will need to alter the data product state in the data product catalog via the retransmit commands to see non-critical products following a reset. If priorities other than defaults are desired, a priority command will need to be issued from the ground to alter the data product priorities.

Retransmits are required if there is an error in the checksum or a data product part is missing. Retransmits are automatically given the highest priority for a given APID because they are the oldest in unsent data. This method of retransmission is based upon a successful heritage design on the Dawn mission. Real-time EH&A and EVR packets are prioritized in the downlink stream over the data product packets, but the MOS can choose selection criteria in a way that enables sharing of the bandwidth. This requires adjusting the real-time EH&A channel selection criteria by file load or command and/or lowering the rate of EH&A real-time packet production to make the EH&A packets small enough and infrequent enough to share the bandwidth. Properly performing this balancing act may require estimating the average number of telemetry channels the S/C will autonomously select to include in this packet.

3.10 SLE Return All Frames (RAF) Service

The DSN provides a capability to acquire and capture telemetry frames from a spacecraft. The selection is made through a setting in the downlink spacecraft configuration file. The DSN delivers telemetry frames that it acquires from spacecraft to the Psyche MOS by the SLE RAF service [25–28]. The interface for telemetry delivery is between the DSN and a Project Operations Control Centers (POCC). The SLE RAF service features two delivery modes. For the online delivery mode, telemetry delivery to Psyche mission system proceeds in parallel with the acquisition of telemetry from the spacecraft. For the offline delivery mode, telemetry acquired from the spacecraft during a tracking pass is recorded and only delivered after the pass has ended.

The Psyche MOS interacts with the DSN to (1) establish an association between the MOS and DSN using TCP/IP, (2) receive annotated telemetry frames from the DSN, (3) obtain notifications and reports regarding status, configuration, and performance of the service, and (4) temporarily stop and later re-start the receipt of telemetry frames and to release an association. Production of the RAF service by the DSN entails processing the telemetry frames as received through the space-to-ground link and through the necessary transformation into the SLE Protocol. The DSN transmits the Telemetry Frames in the order received from the space link. The DSN may perform checks to determine if the Telemetry Frames comply with applicable constraints set by service management.

Data may be temporarily buffered until a latency limit, set by service management, is reached and subsequently discarded if it cannot be delivered within that latency. This buffering may occur during short durations where communications are flow controlled between the DSN and Psyche MOS. Data may also be buffered effectively indefinitely during periods when the space to ground data rates exceed ground communications bandwidth for distributing the data between the DSN and Mission System.

3.11 Downlink Data Accountability

All engineering and science data on the spacecraft, other than real-time EH&A and EVR data, is stored in data products created and managed by FSW. The Psyche S/C provides a data product catalog which is available to the ground. Requesting the data product catalog creates a data product that lists the current set of DPs available for downlink, along with the following metadata that will allow ground operations to query the data content: APID, size, time of creation, downlink priority, sent/unsent status, data product part range. The ground may request retransmission of an entire data product or part(s) of data product by command as well as re-prioritize already created data products. Only the ground has the capability to delete on-board data products, which is nominally done only after the downlinked data product passes error checks.

The downlinked data is accounted for in several places on the ground. The GDS tracks received frames against the SLE Inventory Report published by the DSN, as well as packets, data product parts, and complete data products that have passed their error control checks. The spacecraft team uses a subset of this information to inform retransmission activities.

GDS publishes downlinked instrument data products, processed EH&A, SCLK-SCET (Spacecraft Clock—Spacecraft Event Time) correlation files, ground-generated SPICE Kernel (SPK) files, and many more ancillary data file types to a File Exchange Interface (FEI) server, which is queried by the science data center. The FEI application programming interface allows the science data center to transfer and account for all data the GDS has published over a two-week period. Data is archived at the JPL GDS in case of loss at the SDC. Accountability among the SDC, instrument teams, and science working groups is provided by the SDC portal software.

3.12 Tools & Multi-Mission Services

The Psyche GDS is heavily based on software components from the AMMOS [28], which is a suite of ground system software funded by NASA Headquarters and managed by the Multi-Mission Ground Systems and Services (MGSS) organization within JPL. AMMOS software components are available, at no cost, to all NASA funded missions. Selected AMMOS software components are downloaded from the MGSS configuration management system by Psyche GDS personnel and then "adapted" for integration into the Psyche GDS. Since AMMOS software components were developed for a wide variety of mission types and complexities, adaptation of AMMOS components typically involves configuring each component for the Psyche GDS environment and FGICD specifications. For the Psyche GDS, some specialized AMMOS component tailoring has been required, which is accomplished through MGSS Engineering Change Requests (ECRs).

The AMMOS functions selected for inclusion into the Psyche GDS include Telemetry Processing, Command Transmission, Planning and Sequencing, Mission Design and Navigation, Instrument Data Processing and Archiving, GDS Delivery and Deployment, Configuration Management and Security.

Besides the AMMOS suite of multi-mission software components used by the Psyche GDS, the Psyche Mission System has implemented a set of tools and services that are managed entirely within the Psyche mission. These tools and services are either obtained from other missions and adapted for Psyche Mission System use, or have been developed and tested entirely within the Psyche mission in order to meet Psyche mission requirements. Examples of these Psyche adapted or developed tools and services include the Data Management Toolkit (DMTK), which provides data product accountability and the ACELog, which provides a real-time, automated log of radiated commands.

3.13 Operability

Certain aspects of the EEIS design lend themselves favorably to the mission's ease of operations, like the fact that parameters used by FSW can be updated easily and often by a single command. This reduces the need for frequent FSW code updates or "patches", which require extensive testing programs and long uplink times.

In addition, the features of the sequencing FSW are both enabling and necessary for this mission. As a low-thrust mission, Psyche requires the ability to control onboard events over long periods of time (up to about four weeks) and in a highly predictable, deterministic manner. In addition, capabilities such as looping, branching, onboard telemetry checking, and logic enable the ground to sequence more complex sequences for payloads without requiring FSW to code those behaviors.

Another capability that will increase data return robustness and efficiency is partial data product retransmission and the ability to reprioritize data products from their hard-coded prioritization. This allows the operations team to efficiently utilize limited bandwidth for targeted retransmissions and control when recorded data is downlinked. This is especially useful in tight timelines like safe mode recovery or Optical Navigation (OpNav) imaging. The large onboard data storage additionally decreases the need for timely data analysis, retransmission, and onboard deletion and allows those activities to be more flexibly placed in the sequence planning process.

There are other aspects of the design that pose challenges for Psyche's unique operations. One example of this is that the smallest patch that can be done to FSW code is a full partition of code rather through updates to minor portions of code, thus even minor changes like updating the default value of a parameter in FSW code require the update and uplink of a large portion of FSW code.

Another limitation is that channelized telemetry is recorded at a single rate, which is limited to 1 Hz. Because of several channels that are included in this packet, it will likely be produced much slower than 1 Hz throughout the mission in order to stay within the data return budget. Various channels may be needed faster than this packet will be produced (perhaps even faster than 1 Hz) in order to support things such as calibrations or science ancillary data needs. Those channels may be included redundantly in other data products in order to record them at the necessary frequency.

Last, although the EH&A architecture allows for very nuanced content in the real-time EH&A packet, Psyche has an incredibly low safe-mode downlink rate of 10 bps and, thus, requires very efficient and predictable usage of the bandwidth. Initial safe-mode diagnosis downlink tracks will often not be longer than about four hours because of availability of DSN assets and to ensure that the operations team gets enough channels to assess spacecraft status. The safe-mode real-time packet will need to be very deterministic in the number of channels that are included. In these low downlink rates scenarios, the flexibility in the EH&A downlink configuration cannot be utilized.

4 Challenges & Advantages

4.1 Technical Challenges

The Psyche EEIS faced several technical challenges in its implementation because of interface constraints, project decisions, new technologies, and design implications.

Including DSOC, for example, added unique interfaces between the DSOC Project and the Flight System and Mission System, which required tight coordination with the DSOC Project for end-to-end development and testing. To simplify accommodation of the DSOC instrument, interfaces between the DSOC instrument and the Psyche GDS have been minimized. The Psyche GDS does not rely on DSOC obtained telemetry in any way. Additionally, the later design and implementation of the MAG DTU instrument interface was caused by the original magnetometer instrument being de-manifested.

The hybrid Flight System design splits the avionics design between JPL and Maxar, which impacts integration—especially for the flight harness and bus interfaces.

The new time and space partitioned FSW (FCPL) architecture also introduced challenges. The FSW has to budget processing resources and buffer sizes based on expected maximum data volumes. Late changes to partition resource allocation require regression testing. Latency requirements are inherently more difficult to meet with a time partitioned architecture. This space partitioning allows partial, limited FSW modification instead of a complete software reload for every modification. With a required FSW update, the entire FSW will have to be replaced.

Each instrument has unique data interfaces since each instrument has a unique data link layer (and often physical layer). Because each instrument has a unique data interface, there is a greater variety of requirements that must be closed by the instrument and spacecraft teams. Thus, waivers or exceptions will be utilized on Psyche more often than on a mission with a standard packet interface to instruments.

Also, new GDS SLE interfaces to the DSN require early testing of the SLE service with GDS before the end-to-end tests involving the Flight System. Luckily, Psyche will benefit from sharing this SLE implementation with the Europa Clipper mission.

4.2 EEIS Design Limitations

While the Psyche EEIS design focuses on maximizing operability, it also has some architectural weaknesses. This is partially due to constraints from necessary design decisions and inherited architecture.

The Psyche mission is downlink limited and managing engineering data production in order to maximize science return will be necessary. Psyche selected an X-band telecom system since it was the simplest implementation with a sufficient data rate for science downlink, had sufficient performance capability for gravity studies, required less telecom hardware than a Ka-band system, and was fully compliant with SFCG 23-1. The X-band system does not leave a generous margin for engineering data downlink at maximum Earth-Probe range.

The engineering telemetry architecture for a deep space orbiter needs to be predictable, efficiently utilize bandwidth, and be highly configurable. This could be achieved with an architecture of fixed-size, fixed-offset packets with limited overhead and fine configurability of channel sampling frequencies per APID. However, the inherited JPL EH&A architecture differs. The EH&A packet architecture makes it difficult to predict the exact content and size of EH&A real-time packets during nominal operations since the S/C is autonomously select some of the channels. Intended as a boon for operability, the autonomous EH&A capability adds some complexity to downlink-constrained missions. Also, the single timestamp applied to EH&A data means greater uncertainty in data timestamping and lack of atomic measurement reads. Furthermore, many channels on the spacecraft have duplicates for redundant hardware (A or B) and for the string of measuring hardware (A or B). The EH&A architecture does not allow for filtering the downlink to only including the prime hardware channel measured through the prime measurement hardware, even though the S/C is aware of primness states. The science instrument data collection scheme does not have these weaknesses.

The limitations on the bandwidth available for engineering data downlink include safe mode, which in the worst case is limited to carrier-only operations when only a single 34 m Beam Waveguide antenna is available. Arrayed 34 m or a single 70 m dish allow for data downlink. This adds two operational complexities to the mission. First, the MOS must distinguish between thrust verification carrier-only downlink operations and safe-mode carrier-only downlink operations based on time the carrier is detected since the mission does not employ separate sub-carriers or tones to indicate spacecraft state. Second, the mission has pre-selected the 395 channels to be included in each safe mode EH&A packet. This all but guarantees that diagnostic data products, lower on the downlink priority list, cannot be downlinked until the S/C is recovered enough to achieve a higher downlink data rate.

Psyche has tight OpNav data quality and latency requirements. During Approach, OpNav images must be returned and processed very quickly so that the MDNav team can incorporate the results into the thrust plan. Lost OpNav data has more severe mission consequences than losing mapping data since it is critical for navigation into the various orbits.

Focusing on uplink, EVRs are issued for command dispatch from certain FSW modules, but there is no closed-loop commanding on board for verifying commanded state changes. While fault protection is designed to detect and correct anomalous states, this happens at a higher-level than some commanded operations. Due to the mapping nature of the mission, Psyche has no end-to-end accountability system for mapping downlinked telemetry and science data to commanded actions or tasks. Timestamps on telemetry suffice for accountability purposes.

4.3 EEIS Architectural Advantages

Psyche's use of a single SDC for the interface between the instrument and gravity science teams and the Psyche GDS introduces several simplifications that make the system more robust. The instrument and gravity science teams all pull data from the same SDC, ensuring that they will use the same data as each other as inputs to any review, analysis, or processing. These teams do not need to generate the PDS4 standard archive structures (e.g., data product bundles) on their own. The SDC has the expertise to do so and will for all teams. This architecture enables the entire Psyche science team to have rapid access to all mission science data from the raw packets all the way through high-level derived data products for review and analysis.

The Avionics architecture's use of space and time partitioned software (i.e., FCPL) has its challenges as a new FSW implementation but allows increases the robustness of the architecture. Time partitioning improves the determinism of FSW behavior by fixing the execution schedule of software modules. The architecture also provides increased robustness, as a failure in one FSW partition, such as hung code or an overrunning task, is detected as a time fault and can be addressed through fault protection. Additionally, the FSW architecture allows for autonomous inter-string communications and state and parameter updates to enable, when necessary, quick transition from prime to the backup string.

The Psyche EEIS, in general, provides a lot of flexibility to operators. The EH&A channelized telemetry, both in real time and recorded products, is highly configurable. In addition, the sequencing flight software allows for sophisticated and telemetry-based commanding if necessary. Operators will be able to adapt to a wide range of in-flight challenges due to this flexibility in the software.

The Psyche mission design includes on-board data storage for up to 36 days (must look up correct number). This large amount of storage lessens the risk of missing one or more DSN passes or frequent on-board data product deletions (or something like this).

The Psyche GDS design includes a sub-MSA venue called the MSA overflow venue. Critical, local flight operators, including those that command the spacecraft, use MSA venue resources. Remote mission personnel use Overflow venue resources. Remote mission personnel include JPL based spacecraft team members that are not resident in the MSA and remotely located science team members. The overflow venue can accommodate large and varying numbers of GDS users without affecting or influencing the MSA venue's resources.

5 Conclusion

The Psyche Mission's EEIS architecture and CCSDS implementation is a well-designed architecture that leverages numerous advantageous CCSDS standards and heritage from the past JPL missions.

The Psyche EEIS architecture is not only a well-engineered and operable system for the Psyche mission but provides strategic value to JPL. It is advantageous that Psyche implements CCSDS SLE RAF, tailored Class 1 CFDP instead of using fully proprietary solutions because of multi-mission reuse of these capabilities. These proven capabilities imply less testing, less cost to project and less implementation risk on future missions. Psyche, unlike Mars and Lunar missions, can't claim any cross-support advantages between similar missions or human spaceflight missions for implementing these standards. Psyche's leverage of previous mission heritage is cost effective and reduces risk while minimizing testing needs. Using core flight software shared with the Europa Clipper mission demonstrates multi-mission reuse of flight software across multiple platforms.

In addition, Psyche is enabling the DSOC technology demonstration for optical communications, which is a first for JPL and a major advance for space communication [29, 30]. In effect, DSOC is testing out these standards for deep space communications for CCSDS agencies, including NASA. The promise of optical communications it to provide ten times the data rate that is possible with RF communications, or similar size, weight, and power. DSOC itself is a technology demo, not an operational system, so it is not the final culmination of that effort, but it will be capable of transmitting at 266.67 Mbps at the closer range. Psyche will also implement other CCSDS standards for the first time at JPL, such as SLE RAF, which will provide a path for multi-mission reuse and ease of interoperability with other CCSDS member space agencies.

Overall, the Psyche EEIS performs well against QQCL metrics, enables new technologies and standards, and addresses key operability needs.

Acknowledgements The research was carried out as part of the Psyche Mission at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

References

1. Elkins-Tanton LT, Asphaug E, Bell JF, Bercovici H, Bills B, Binzel R, et al (2020) Observations, meteorites, and models: a pre-flight assessment of the composition and formation of (16) Psyche. J Geophys Res Planets 125:e2019JE006296. https://doi.org/10.1029/2019JE006296
2. Clark B, Faulconer C, Gamber T (2006) Formulation of discovery-class mission concepts. IEEE Aerosp Electron Syst Mag 21(4):27–33
3. Biswas A (2019) NASA's deep space optical communications—an update. In: Laser congress 2019 (ASSL, LAC, LS&C), OSA Technical Digest (Optical Society of America, 2019), paper LTh1B.1
4. Biswas A, Srinivasan M, Piazzolla S, Hoppe D (2018) Deep space optical communications. In: Proc. SPIE 10524. Free-space laser communication and atmospheric propagation XXX, 105240U, 15 February 2018. https://doi.org/10.1117/12.2296426
5. Pham T (2006) A perspective on DSN system performance analysis. In: SpaceOps 2006 conference, AIAA 2006-5536. https://doi.org/10.2514/6.2006-5536
6. Fieseler PD, Taylor J, Klemm RW (2020) Dawn spacecraft performance: resource utilization and environmental effects during an 11-year mission. J Spacecr Rocket 57(1):147–159

7. Cabane M (2009) Mars Science Laboratory (MSL) and the future missions to Mars. Proc Int Astron Union 5(H15):710–711. https://doi.org/10.1017/S1743921310011117
8. Entekhabi D et al (2010) The soil moisture active passive (SMAP) mission. Proc IEEE 98(5):704–716. https://doi.org/10.1109/JPROC.2010.2043918
9. Banerdt WB, Smrekar SE, Banfield D et al (2020) Initial results from the InSight mission on Mars. Nat Geosci 13:183–189. https://doi.org/10.1038/s41561-020-0544-y
10. Williford KH, Farley KA, Stack KM, Allwood AC, Beaty D, Beegle LW, Bhartia R, Brown AJ, de la Torre Juarez M, Hamran S-E, Hecht MH, Hurowitz JA, Rodriguez-Manfredi JA, Maurice S, Milkovich S, Wiens RC (2018) Chapter 11—the NASA Mars 2020 rover mission and the search for extraterrestrial life. In: Cabrol NA, Grin EA (eds) From habitability to life on mars. Elsevier, Amsterdam, pp 275–308. ISBN 9780128099353
11. Phillips CB, Pappalardo RT (2014) Europa clipper mission concept: exploring Jupiter's ocean moon. Eos Trans AGU 95(20):165
12. Shi Y (2020) Software reliability in space applications-facts, trends and challenges. In: Reliability and maintainability symposium 2020
13. Ko A, et al. (2010) The evolvable advanced multi-mission operations system (AMMOS): making systems interoperable. In: SpaceOps 2010 conference delivering on the dream hosted by NASA Marshall space flight center and organized by AIAA
14. M. Sørensen E, et al (2002) SLE is going operational. In: SpaceOps 2002 conference
15. Planetary Data System. https://pds.nasa.gov/home/about/
16. Book M (2008) Reference architecture for space data systems. Recommended practice, CCSDS 311.0-M-1
17. Shames P, Skipper J (2006) Toward a framework for modeling space systems architectures. In: Proceedings of the ninth international conference on space operations (Rome, Italy). SpaceOps
18. Blue Book (2015) AOS space data link protocol. CCSDS 732.0-B-3
19. Silver Book (2003) Space packet protocol. CCSDS 133.0-B-1
20. Blue Book (2007) CCSDS file delivery protocol (CFDP). CCSDS 727.0-B-4
21. Blue Book (2008) Space link extension—internet protocol for transfer services. CCSDS 913.1-B-1
22. Red Book (2000) Space link extension—DSN forward CLTU service specification. CCSDS 912.1-R-1.99h
23. Silver Book (2004) Space link extension—forward CLTU service specification. CCSDS 911.1.-B-2-S
24. Blue Book (2010) Space link extension—forward CLTU service specification. CCSDS 912.1-B-3
25. Red Book (1999) Space link extension—DSN return all frames service specification. CCSDS 911.1-R-1.7
26. Silver Book (2004) Space link extension—return all frames service specification. CCSDS 911.1-B-2-S
27. Blue Book (2010) Space link extension—return all frames service specification. CCSDS 911.1-B-3
28. Advanced Multi-Mission Operations System. https://ammos.nasa.gov
29. Blue Book (2019) Optical communications physical layer. CCSDS 141.0-B-1
30. Blue Book (2019) Optical communications coding and synchronization. CCSDS 142.0-B-1

PLUMMRS: Towards Safe Multi-robot Task Planning and Execution

Ana Huamán Quispe, Stephen Hart, and Seth Gee

Abstract As space exploration moves farther from Earth, robotics will play an ever-increasing role in NASA's upcoming missions. Farther scenarios present new challenges, such as the absence or a very limited presence of human supervision for long periods of time. Spacecraft will have to be more dependent on automation and robotic agents to perform maintenance and repair tasks. Rather than having robots work side by side with humans, *robots will have to work side by side with each other*. In response to these needs, this chapter presents PLUMMRS (A Collection of Plan Ledgers and Unified Maps for Multi-Robot Safety), a software framework that facilitates sharing of environmental and internal state information to enable safe, efficient navigation and manipulation tasks by heterogeneous robot teams working in a shared workspace. The goal of PLUMMRS is to provide simple APIs for existing single-agent planning and execution systems to leverage. This will allow previously individualistic robots to be used safely in multi-agent contexts. This chapter describes PLUMMRS architecture and a prototype implementation, showcasing its utility in scenarios such as the ISS with a small team of robots involving 2 Astrobees and a Robonaut, performing both colocated and collaborative tasks.

Keywords Robotics operations · Autonomy · Multi-agent systems · Shared representation

A. Huamán Quispe (✉) · S. Hart · S. Gee
TRACLabs, Inc, 16969N Texas Ave. Suite #300. Webster, San Antonio, TX 77598, USA
e-mail: ana@traclabs.com

S. Hart
e-mail: swhart@traclabs.com

S. Gee
e-mail: seth@traclabs.com

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_6

1 Introduction

As space exploration missions move farther from Earth for longer periods of time, they are becoming more dependent on automation and robotics than ever before. The Gateway is a perfect example of this, where intra-vehicular robots (IVR) will perform maintenance and monitoring tasks with limited remote human supervision [1]. This will require multiple, heterogeneous robots to share a workspace, using their complementary skills to perform complex tasks that could otherwise not be achieved. As missions become more complex, autonomous multi-robot teams will be needed for more than just spacecraft maintenance. With habitat construction, exploration of complex environments (e.g., swarms delving into lava tubes [8]), and large-scale regolith mining, all will require teams of specialized robotic systems.

One major hurdle to robust multi-robot operations in space is the same hurdle faced by multi-robot applications on Earth—from autonomous vehicles to mobile warehouse robots, co-located robotic "individuals" do not adequately share state information. State information can comprise a variety of data, including: environmental knowledge (like a map of object locations), current position and velocity in a map, expected position in the future, what task is currently being performed, what resources are needed to complete the current task, where shared resources (like a tool) may have been left after a previous task, and even data about if/when/why either low-level or high-level plans failed. Sharing some amount of state information is not only a necessity for tasks that require cooperation between multiple robots, but is critical even for groups of independent physical agents that must share the same workspace—consider a day of rush hour traffic if all traffic signals (which control the state of intersections) in the U.S. suddenly stopped working.

A large logistics company might have the luxury of purchasing all the same make and model robots so that perception data, navigation planning, task planning, and safety monitoring can be all handled via a proprietary centralized processing system. This might allow robots to make plans that avoid going down the same aisle that another robot is in, or might let two bin picking robots pick out of the same bin at the same time. In a sense, the multi-robot system is treated as a single robotic entity planning from a single unified model of the world. However, a more likely scenario for the future of most manufacturing, large-scale inspection, security, and emergency applications is that a variety of different robot makes and models will be deployed and will need to work safely and efficiently alongside each other and humans. Each of these robots will have its own internal model of the world, and its own desires and expectations to fulfill the high-level tasks it is programmed to perform. To date, there is no easy way for such robots to share this information with each other.

Space exploration requires even more diversity in the population of robots. Each new mission mandates new hardware, different processing and communications constraints, and improved software algorithms, so almost every new robotic platform will be highly unique. Even within NASA, individual robot projects like Robonaut 2 (R2) [2], Valkyrie, Astrobee [3], and Viper do not share a common codebase that facilitates straightforward sharing of rich internal state information. Furthermore, even individual instances of the R2 and Astrobee platforms are not necessarily equipped with the robust software needed to adequately share state information between robots in the same work area. This problem will only get worse as other international space agencies and commercial space companies begin to deploy more robotic systems.

In direct response to NASA's need for perception systems for interior environment monitoring, modeling, and navigation and for operational subsystems that increase robot autonomy, our presented work proposes to create a collection of software processes that will facilitate sharing of environmental and internal state information to enable safe, efficient navigation and manipulation tasks by heterogeneous robot teams working in a shared workspace. We call this software collection PLUMMRS (A Collection of Plan Ledgers and Unified Maps for Multi-Robot Safety). The goal of PLUMMRS is to provide simple APIs for existing single-agent planning and execution systems to leverage. This will allow existing individualistic robots to be used safely in multi-agent contexts. That is, PLUMMRS can be used by any individual robot in a group of robots to contribute to, and benefit from, a unified model of not only geometric and semantic perception data but also of expected and currently executing motion- and task-level plan data. PLUMMRS is *not* a planning framework, as each robot is expected to have its own "black-box" motion and task-planning capabilities; however, PLUMMRS can be used by both motion- and task-planning & control systems as (1) an oracle of shared knowledge, (2) a safety monitor based on the shared knowledge representation, and (3) an arbiter (and eventually a full scheduler) that attempts to loosely coordinate the short-term and long-term desires proposed by all individual robots, which are trying to independently complete their tasks while sharing geometric space and physical resources.

2 Background

The proposed PLUMMRS framework does not depend on any specific robot control architecture, however for all the presented proof-of-concept demonstrations we used our CRAFTSMAN framework to program the tasks. In this section, we briefly discuss this tool.

2.1 CRAFTSMAN

Over the past 6 years, TRACLabs researchers have been developing and deploying the CRAFTSMAN framework, which comprises a suite of components that work together in order to provide advanced mobile manipulation capabilities to robot applications. CRAFTSMAN is designed to be robot agnostic and was developed to provide an easy-to-configure and easy-to-use tool suite for both expert and non-expert developers [4]. It provides advanced kinematics, obstacle-free finger/tool-tip planning, and motion-generation algorithms for both configuration and Cartesian spaces. It also provides a graphical task description language called the *Affordance Template* (AT) framework [5] that uses the underlying motion algorithms to greatly simplify the design and use of task-oriented behaviors. The current software implementation uses libraries from the *Robot Operating System (ROS)* [7] ecosystem, including the inter-process messaging and 3D visualization tools, and also provides integration with standard and custom navigation packages for systems that have mobile capabilities.

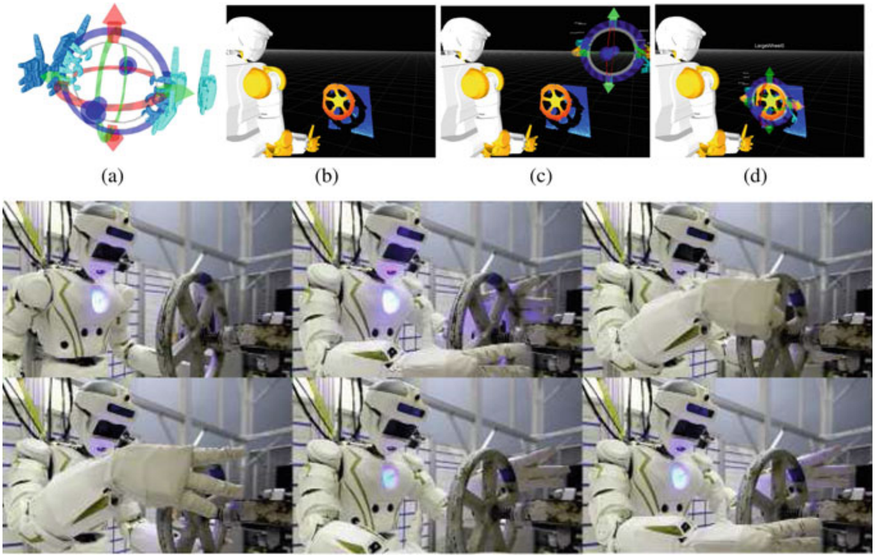


Fig. 1 Wheel-turning affordance template used with the NASA Valkyrie humanoid. **a** A two-handed wheel-turning template. The virtual wheel object is shown along with ordered end effector waypoints for both of the robot's hands in light blue. Note that each waypoint also specifies a hand pose (e.g., open versus closed). A 6-DOF gimbal around the object can be used by an operator to adjust the object's position. **b–d** Views of the 3D environment, where a human operator adds the wheel-turning AT and manually registers it to the robot's 3D sensor data using the available interactive controls. (bottom) The Valkyrie robot moving its end effectors through the AT waypoints and successfully turning the wheel

Affordance templates (ATs) allow a programmer to specify sequences of end effector waypoints represented in the coordinate systems of the environmental objects that a robot interacts with. Virtual overlays of both the objects and the task-specific end effector waypoints can be visualized in a 3D interactive environment, along with an avatar of the robot and the robot's sensory data (Fig. 1). Waypoints can be defined using any reference frame in the robot's workspace, but are usually defined with respect to objects. Waypoints include conditioning information like the type of navigation to use or the style of arm motion to prefer (e.g., straight-line versus joint motion). Waypoints can also specify attached objects, so that grasped objects will be accounted for automatically when checking collisions during planning. Recent extensions to the affordance template specification allow stance locations to be defined regarding the virtual AT objects [6] such that a mobile robot “knows” where to move to in order to accomplish the manipulation task.

To date, TRACLabs has deployed the CRAFTSMAN suite on various proof-of-concept flexible manufacturing cells, the 5-armed Robomantis platform (for Motiv Space Systems), NASA Valkyrie humanoid (for NASA Johnson Space Center), and various International Space Station robot simulations (for NASA JSC/NASA Ames). A closely-related predecessor to CRAFTSMAN was used with the Boston Dynamic Atlas humanoid in TRACLabs’ DARPA-funded entry into the DARPA Robotics Challenge.

3 PLUMMRS

The proposed innovation is summarized by the high-level diagram in Fig. 2. There are three proposed elements in PLUMMRS that will allow it to (1) unify perceptual information, spatial knowledge, and state information (`Unified Map`), (2) track

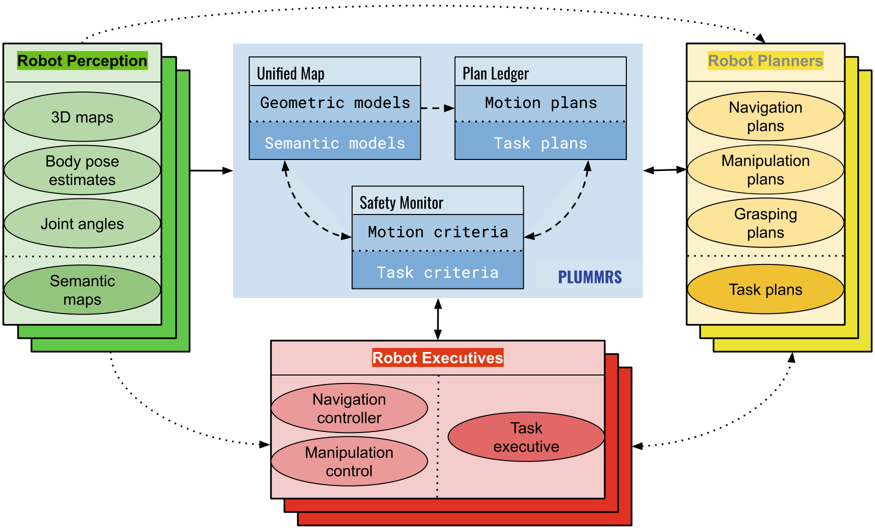


Fig. 2 The initial high-level diagram of PLUMMRS for multi-robot safety. It is assumed that a single individual robot has its own internal implementation of the traditional **`Sense`**, **`Plan`**, **`Act`** (or **`Observe`**-**`Orient`**- **`Decide`**-**`Act`**) loop. PLUMMRS provides a collection of processes (Unified Maps, Plan Ledgers, and Safety Monitors) that enhances the Sense, Plan, Act loop with simple-to-integrate API calls. PLUMMRS allows these individual robots to share perception models and state information, while also validating individual robot plans with respect to other queued and executing plans for safety. PLUMMRS is expected to have components for handling both geometric volume information for motion planning and semantic/symbolic information for task planning

proposed and currently executing motion trajectories along with task-level symbolic plans that require shared resources (`Plan Ledger`), and (3) monitor state, validate safety, and broadcast concerns at a fast loop rate (`Safety Monitor`).

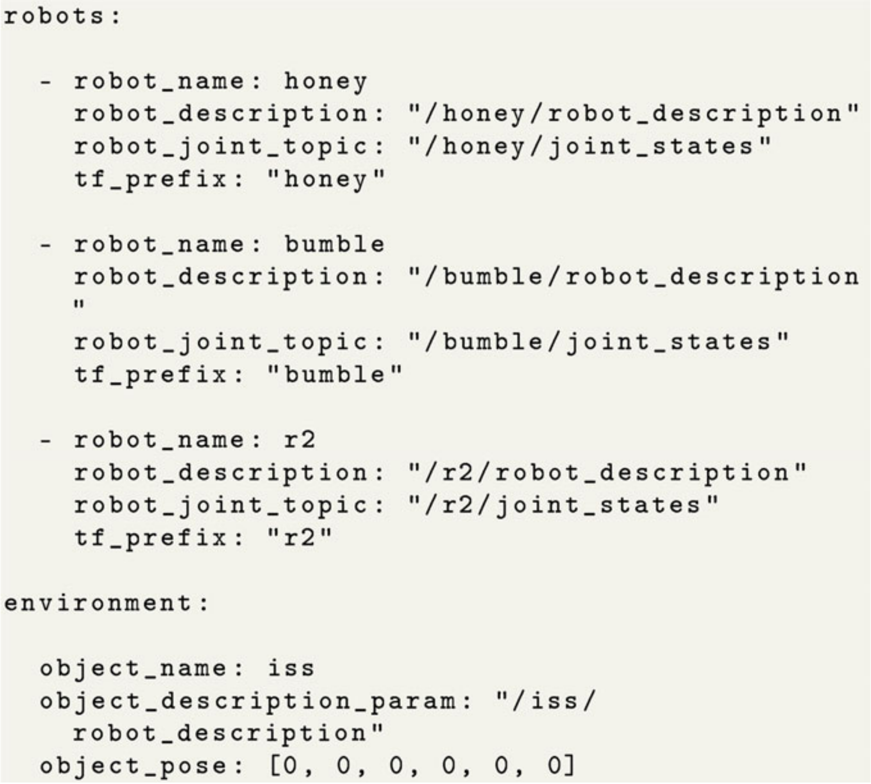
3.1 Unified Map

Whether operating inside a home, a factory, or a spacecraft, robots need a model of the environment in which they operate. Depending on the application, different types of information will be needed. For mobile robot navigation tasks, spatial information, such as an occupancy map, might be sufficient for success. For manipulation tasks, object data (e.g., location in the map, 3D shape data, and mass properties) will be required for the robot to effectively reason about and manipulate its environment. For collaborative robot teams (e.g., for maintenance tasks, such as detecting and fixing leaks), a shared, integrated scene representation is necessary for effective collaboration. Even for non-collaborating robots, sharing environmental knowledge improves planning and control by providing all robots with more complete knowledge and the most up-to-date information available—Astrobee may delay an inspection task in Tranquility Node 3 if R2 reports Unity Node 1 is occupied by crew members performing another task.

This module was implemented to have the following characteristics:

– It should receive robot state information from each independent agent. For instance, if there are N robots, the Unified Representation module will be subscribed to N joint_states topics.
– It should support the specification of static world knowledge, such as the description of an environment in which the robot operates (e.g., the ISS module).
– It should support dynamic addition/deletion of robots. We intend for PLUMMRS to be flexible and not need to restart the processes every time a new robot is turned on or off.
– It should support basic geometric representation of objects in the environment, such as primitive, meshes, and octotrees (output of standard RGBD sensors).
– It should periodically publish a single topic with all the condensed information received from multiple sources into a single ROS message, which could then be used for any planning/control node to inform its decisions.

Our prototype implementation of the Unified Map module consists of a ROS class, aptly named UnifiedRepresentation, which runs with a default frequency 30 Hz. The class receives as initial arguments a desired frequency, and optionally, a .yaml with some basic information of the robots that will be monitored by the node. Figure 3 shows an example of one such configuration file for the scene that involves two Astrobees and the R2. As it can be seen, the configuration file contains simple blocks for each robot, indicating the sources to which the Unified Representation node will have to subscribe to get the robot's updated state information. Note that this parameter is optional, since our node also allows the user to use a service to add robots during runtime. The only mandatory parameter that must be set at the start of operations is the **reference frame**; that is the fixed frame which will be used as the base frame to localize all the entities in the scene (robots and objects). The fixed frame is normally a robot-independent frame (unless the robot itself is fixed), such as **world**.



```
robots:

  - robot_name: honey
    robot_description: "/honey/robot_description"
    robot_joint_topic: "/honey/joint_states"
    tf_prefix: "honey"

  - robot_name: bumble
    robot_description: "/bumble/robot_description
    "
    robot_joint_topic: "/bumble/joint_states"
    tf_prefix: "bumble"

  - robot_name: r2
    robot_description: "/r2/robot_description"
    robot_joint_topic: "/r2/joint_states"
    tf_prefix: "r2"

environment:

  object_name: iss
  object_description_param: "/iss/
    robot_description"
  object_pose: [0, 0, 0, 0, 0, 0]
```

Fig. 3 Example configuration file for the 02-Astrobee-01-R2 scenario

Internally, the Unified Representation class contains an assortment of derived classes of the base class **Entity**, such as **RobotEntity** and **EnvironmentEntity**, the latter which is mainly a container for a vector of **ObjectEntity** instances. These Entity classes provide:

- **An update function:** At every time step, this function is called and updates the frame information of the specific element represented. For a RobotEntity, this means updating the joint state and robot's pose (if mobile) regarding the reference frame. If the element is an object, the update function might (1) do nothing if the object is fixed, or (2) it would listen to a transform and update its pose if the object is mobile, or (3) update the object's geometry data if the object represents sensor information (e.g., an octotree).

- **A ROS message:** Each of these classes transforms their updated data into ROS message form, which is then assembled by the UnifiedRepresentation node and published as a single message.

The Unified Representation node publishes the compiled information into a topic named **world_state**, which outputs messages of type **plummrs_msgs/WorldState**. The current structure of this message is shown in Fig. 4.

```
std_msgs/Header header
string reference_frame
plummrs_msgs/RobotState[] robots
plummrs_msgs/EnvironmentState environment
```

Fig. 4 plummrs_msgs/WorldState.msg, which is published in the **world_state** topic from the Unified Representation node

As it is shown, the world state contains the current state of the robots plus environmental information. The message **plummrs_msgs/RobotState** (Fig. 5) stores the robot's current location, its joint state and any manipulated object the robot is handling. As for the environment itself, it is represented as an array of objects (Fig. 6), each of which is described as a combination of their intrinsic geometric information and their pose regarding the reference frame (Fig. 7). The object geometry message supports objects that are fixed or movable (Fig. 8), in the latter case, the parameter **parent** allows for an object to have different parent frames during runtime, which is useful if an object is being manipulated for over one robot at different points in time.

```
plummrs_msgs/Robot robot
geometry_msgs/PoseStamped robot_pose
sensor_msgs/JointState joint_state

plummrs_msgs/ObjectState[] attached_objects
```

Fig. 5 plummrs_msgs/RobotState.msg

```
plummrs_msgs/ObjectState[] objects
```

Fig. 6 plummrs_msgs/EnvironmentState.msg

```
plummrs_msgs/Object object
geometry_msgs/PoseStamped pose
```

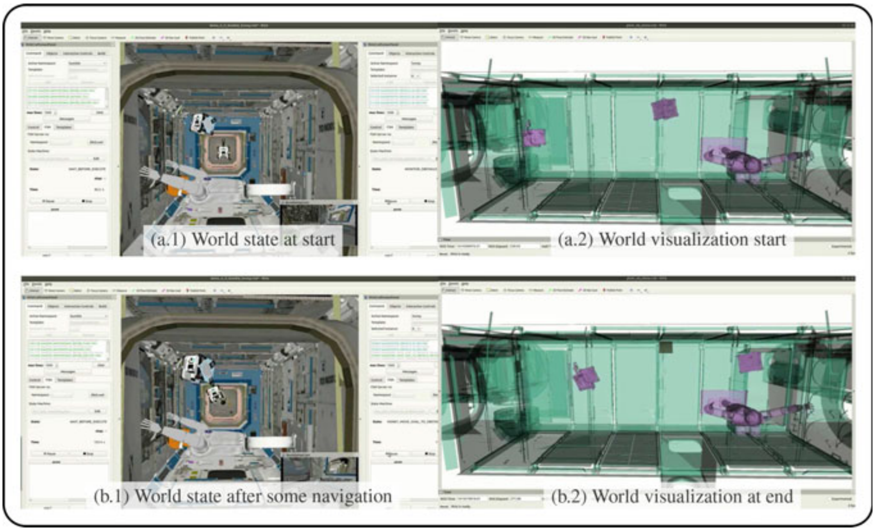
Fig. 7 plummrs_msgs/ObjectState.msg

```
string name
plummrs_msgs/ObjectGeometry geometry
string parent_frame
bool movable
```

Fig. 8 plummrs_msgs/Object.msg

3.1.1 Visualizing the World State

In order to verify that the **world_state** topic is publishing an accurate snapshot of the multi-robot scene, we created a Rviz Display plugin to visualize the **plummrs_msgs/WorldState** messages. Figures 9 and 10 show our plugin in action, publishing all the information from the Unified Representation node into a single topic, which is visualized in our custom display. Notice that this display is showing the specific robots being simulated besides the static ISS model and sensor data.



(a.1) World state at start

(a.2) World visualization start

(b.1) World state after some navigation

(b.2) World visualization at end

Fig. 9 Visualization of our world_state message. Images at the left are the "real" scenario. Images at the right (purple and green semi-transparent display) are the visualization of the messages being published in the **world_state** message, which condense all the individual pieces of information of the system into a single source

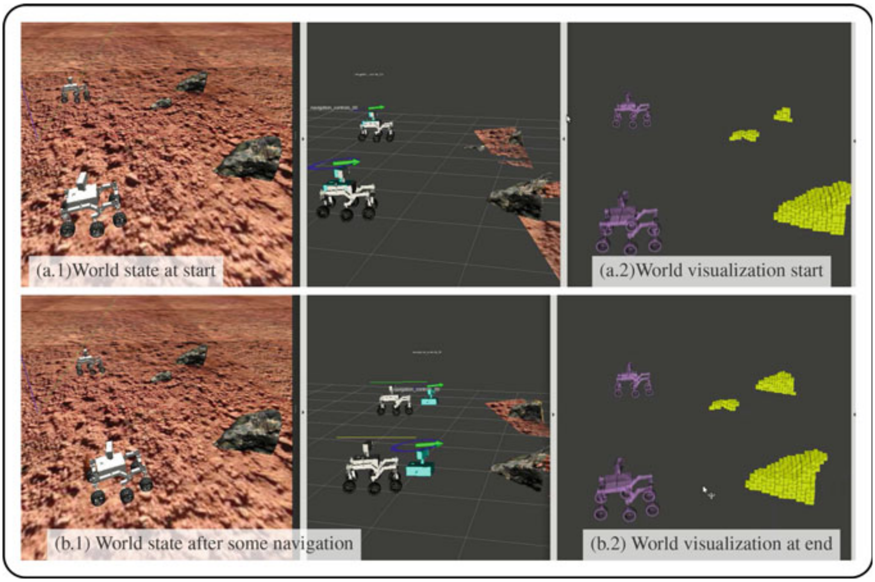

(a.1)World state at start
(a.2)World visualization start
(b.1) World state after some navigation
(b.2) World visualization at end

Fig. 10 Visualization of our world_state message for 2-rover setup. The purpose of this evaluation scenario is to test if the UnifiedRepresentation could handle the management of sensor data information (octotrees). Comparing the upper and lower rows, it can be observed that the octotree data varies as the rovers move. The UnifiedRepresentation updates this information accordingly (yellow semi-transparent display at the right). Video available at https://youtu.be/g7GVnViLlfU

3.2 Plan Ledger

The Unified Map element provides current and past knowledge about obstacles, objects, and their relations to an environmental frame-of-reference, along with robots' positions in the environment. While useful, information about the current state of the world is not sufficient for the level of safety needed for teams of robots working in close-proximity. In PLUMMRS, information regarding expected future state is handled by the *Plan Ledger* element. The Plan Ledger exists to ensure that each robot's proposed plans, even in the scenario where no robot planners utilize the Unified Map information, are validated for safety against all other executing (or soon-to-be executed) robot plans.

There are legacy safety systems that rely purely on current state information (and possible estimates of future state based on extrapolations from the recent past) to warn each robot when unsafe events are imminent. This approach might lead to deadlock, backtracking, and possibly even physical collisions. It is safer and more efficient for the team as a whole if robots develop plans with the knowledge of what other robots expect to be doing soon [8]. For example, if two robots need to occupy the same space or use the same tool to accomplish different tasks, then one robot should yield to the other and perhaps re-schedule the task at a later time.

3.2.1 Multi-Robot Collision Detection

Our first step in the development of the Plan Ledger was to create an enhanced Collision Detection module (named Plummrs Collision Detection) that supports multi-robot collision queries. We accomplish this by making our Plummrs Collision Detection subscribe to the **world_state** topic published by the Unified Representation node, and using this information to populate the Collision Detection's internal state. For simplicity, this internal state was designed to have a structure similar to the UnifiedMap module, namely, to consist of a set of derived classes of a **BaseCollision** class, similar to the **BaseEntity** class. These derived classes are **RobotCollision**, **EnvironmentCollision** and **ObjectCollision**. These classes provide two main functions:

- **update:** Every time a new message arrives on the world state topic, this function uses its data to update the FCL Collision Geometry objects that constitute the specific element represented. With a robot, this means updating all its links' poses using the Forward Kinematics of the robot's urdf, plus attaching/detaching new objects being held if applicable. With an object, the update function simply updates the pose of the object regarding the reference frame. If the object is a sensor object, such as an Octotree, the FCL Collision Geometry must be deleted and re-created every time the sensor data changes.
- **getSnapshot:** This function allows an application to query for the current state of every Collision instance, represented by a vector of FCL Collision Geometry objects.

If a new robot/object is added/deleted in the Unified Map, the Plummrs Collision Detection module will accordingly add/delete an instance of the corresponding Collision derived class automatically by comparing the latest message with its current internal state.

Figure 11 shows some results obtained using our Collision Detection module. As a reminder, all the robots are running independently and are not aware of each other. In Fig. 11a the robots start all in a collision-free configuration. In Fig. 11b, Honey moved towards Robonaut and it is colliding with its torso. The collision detection module indicates this by terminal messages as well as by publishing the geometries of the colliding pairs of links in red. Figure 11c shows Honey after moving away from Robonaut, collision-free again. Figure 11d shows Bumble moving towards Honey and now both of them colliding, as shown by the red markers.

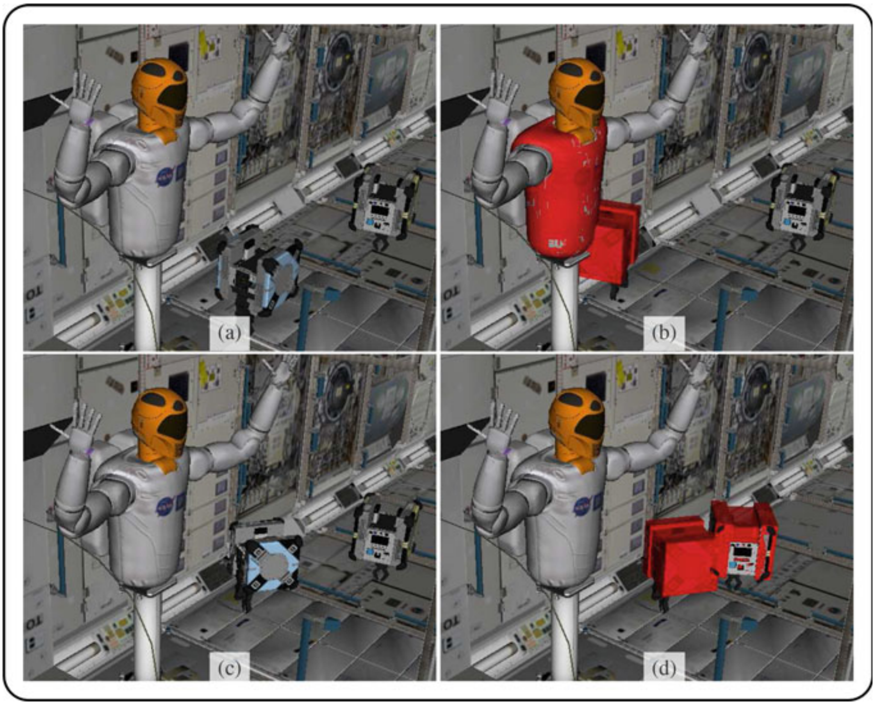


Fig. 11 Collisions detected between robots running independently. Our Collision Detection module receives information of the current World State as published from the Unified Representation node and it checks for collisions as updated robot states arrive. Video available at https://youtu.be/4s32sGW6oW4

3.2.2 Validation Services for Robot Plans

The Plan Ledger has as main purpose to offer a way for each robot to verify **before execution** that an intended motion plan is collision-free with respect to the rest of the environment. We implemented this functionality by enabling the Plan Ledger to advertise 2 services:

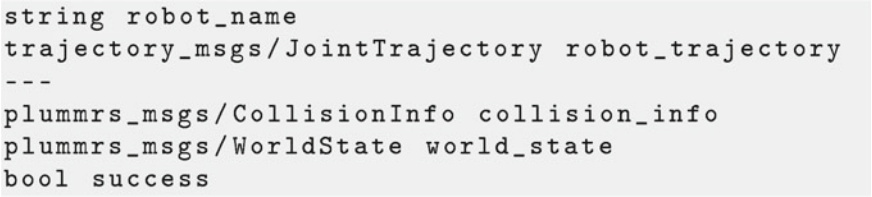
- **validate_robot_trajectory:** Allows a robot planner to send a request with the robot name and a joint trajectory corresponding to a limb motion (e.g., Robonaut's arm). If the robot has navigation capabilities, it is assumed that the robot base will not move while executing the limb motion.
- **validate_robot_navigation:** Allows a (mobile) robot planner to send a request with the robot name and a navigation path. If the robot has limbs, it assumes they do not move during navigation.

Both service messages are depicted in Figs. 12 and 13. As can be seen, besides returning whether the submitted plans are collision-free (success = true), the Plan

Ledger also provides feedback information to the robot planners on what caused the plan to be considered invalid, if that were the case:

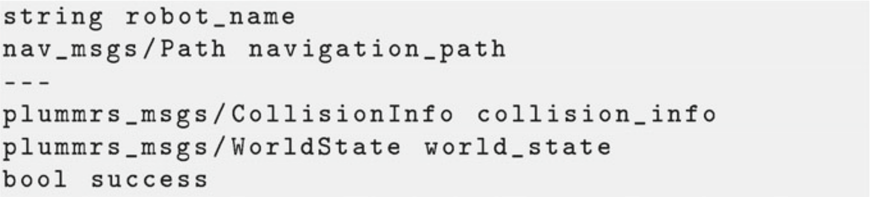
- **collision_info:** Returns the pair of geometries that produced a collision between the querying robot and the environment (e.g., R2’s left palm and Honey’s gripper left distal link).
- **world_state:** Returns the whole world state at the exact point in time at which the plan validation request was sent.

Note that the valid state of a plan is determined simply by checking each independent trajectory point against the current world state, which is the equivalent to say that we just compare the sweeping volume of the submitted path against the other collision objects. No time constraint is considered in this prototype for simplicity, meaning that we assume the path is going to be executed immediately.



```
string robot_name
trajectory_msgs/JointTrajectory robot_trajectory
---
plummrs_msgs/CollisionInfo collision_info
plummrs_msgs/WorldState world_state
bool success
```

Fig. 12 Validate_Robot_Trajectory.srv



```
string robot_name
nav_msgs/Path navigation_path
---
plummrs_msgs/CollisionInfo collision_info
plummrs_msgs/WorldState world_state
bool success
```

Fig. 13 Validate_Robot_Navigation.srv

There are multiple ways in which a planner can use the feedback information of the Plan Ledger. Upon receiving a false validation result, a robot planner could choose to: (1) Wait for a short period of time and then submit the same plan, and repeat the process till the plan is deemed valid; (2) Use the collision_info data and plan around the collision point by adding a nearby additional waypoint that locally modifies the path just enough to avoid the source of collision, and (3) Use the full world state information as input to a more general collision-free planner, such as RRT, to circumvent the source of collision.

We experimented with those 3 options by modifying existing CRAFTSMAN planners and using the feedback information as described. Figure 14 shows an example of using the Plan Ledger world state feedback information to replan around the Robonaut arm, which moves such that its forearm is in the way of Astrobee's inspection path. The final demonstrations show examples of such planner being used to avoid collisions between the Astrobees and the Astrobee and Robonaut.

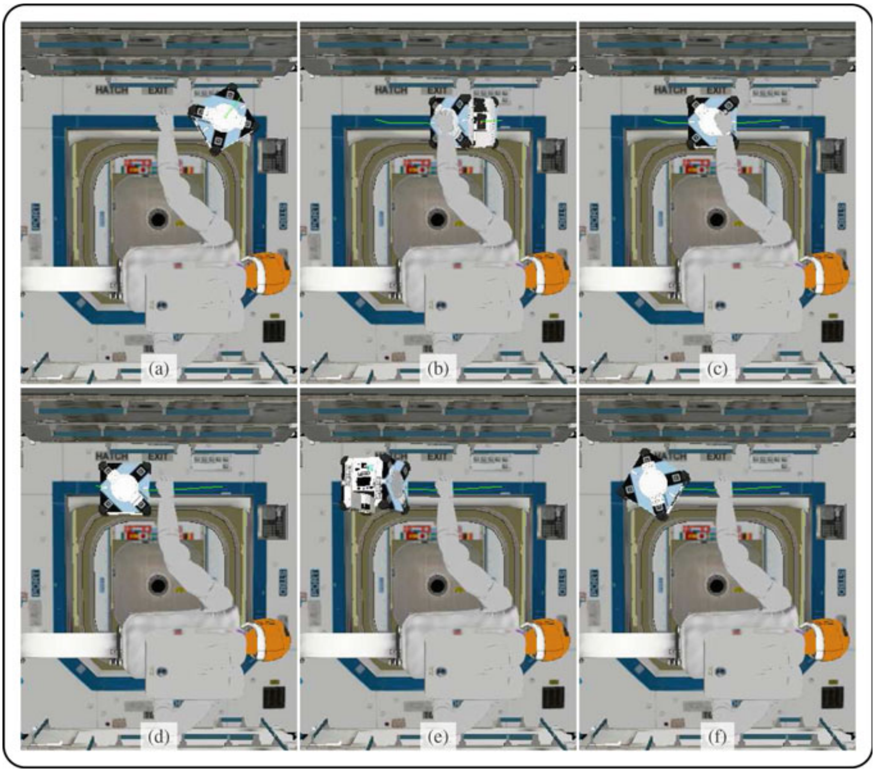


Fig. 14 Astrobee using Plan Ledger feedback information to replan around Robonaut arm and avoid collision (video available in https://youtu.be/nXJWiaRKCN8)

3.3 Safety Monitor

While the Unified Map element can provide a single-agent robot planner current and historical data about the shared team's environment and each robot's state in the past and present, the Plan Ledger helps planning systems coordinate in the future. However, even the best-laid plans of robots can go awry. The *Safety Monitor* element is meant to reinforce and improve upon any execution-time safety performed by individual robots. Specifically, the Safety Monitor element is meant to (1) quickly process all incoming sensor, state, and plan information, (2) estimate the progress of all currently executing plans, (3) check for any potential issues with the remaining execution of plans, (4) determine if any plans need to be changed, and (5) broadcast the status and any plan changes to all team robots.

Plan execution issues can include: potential collisions based on new sensor data or new information in the Unified Map, a robot not adequately following its plan (which is an issue both for that robot and for nearby robots), or plan preemption by some higher-priority event, like a human moving too close to a robot. In the simplest implementation, any relevant change in the world state (Unified Map, immediate change in state data, or a change in the Plan Ledger from a higher-priority plan) that would cause an executing plan to not pass the Plan Ledger validity test would simply cause that plan to be invalidated, and the robot executing that plan would be expected to safely Halt. More sophisticated implementations might simply slow down robots when potentially unsafe situations are predicted far in the future, waiting to completely invalidate a plan until danger becomes unavoidable with continued execution.

While the Unified Map element will probably need to process data at sensor rates of up to 60 frames per second, and the Plan Ledger will be expected to handle at least 1–2 plan requests a second, a Safety Monitor, even in the most basic form, will probably need to process incoming data at least 100 Hz. The Safety Monitor will need to perform plan validation tasks that are similar to how the Plan Ledger performs validation, but with a much faster deadline. Thus, the developed approach must streamline computation as much as possible. A less conservative, tiered approach to plan validation might be used, or perhaps a time-window approach could be used—only validating plans for the next t seconds into the future instead of validating all plans to completion on every cycle.

Our Safety Monitor prototype consists of a single ROS node that kept a vector of instances of the class **BasePlan**. These classes represented the robots' plans that were currently being executed. The Safety Monitor had 3 main sources to obtain/share information of the safety status of the global multi-robot system:

- **Unified Representation:** The Safety Monitor is subscribed to this topic to obtain the latest information of the environment current state.
- **safety_monitor_nav_paths:** The Safety Monitor is subscribed to this topic. Any robot's executor can publish to this topic with 2 pieces of information: (1) A robot name, and (2) A navigation path that is being executed. Upon receipt, the Safety Monitor stores this information in a **BasePlan** (and deletes old plan if any exists). If a threshold period of time passes and no new path information is received for a robot, the BasePlan is automatically deleted, as it is assumed the path finished being executed.
- **safety_monitor_info:** The Safety Monitor constantly checks for collisions between the **BasePlans** stored and the environment as well as between the **BasePlans** themselves. If a collision appears to happen at some point in the evaluated plan, that information is published in the **safety_monitor_info** topic. A reasonable safety implementation (which was done for the demos) is to enhance a robot's executor by adding a subscription to this topic, and if a notification of possible

collision is received, a preempt call is requested, possibly followed by a replanning request. This latter step should be performed by a higher-level task management module – in our case, a Finite State Machine.

4 Demonstration Tasks

In this section, we describe 3 tasks involving caretaking and maintenance of a spacecraft being performed by 2 robots working together. General technical details common to the 3 demonstrations are:

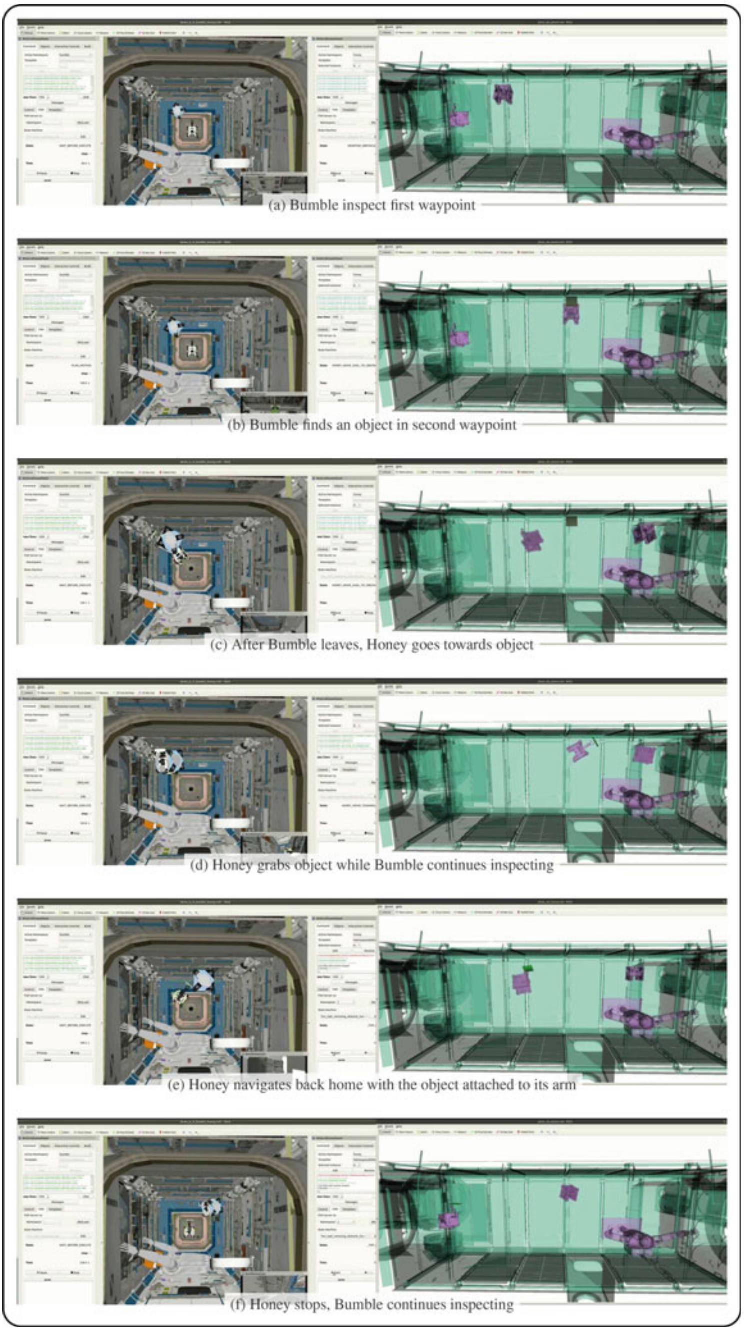
- Simulations were performed in Rviz.
- Models of the ISS and Astrobees were used as found in the NASA's github Astrobee repository.
- All tasks were encoded using Finite State Machines.
- The only operator input during the whole run happens at the beginning of each video, when the user clicks (a single time) the Start button on the Finite State Machine GUI. All the operations that are seen in the videos are fully commanded/sequenced by the FSMs, using the information from the UnifiedRepresentation and PlanLedger.

4.1 Demonstration 1: Visual Inspection and Obstacle Removal Task Performed by a 2-Astrobee Team

Video of run: https://youtu.be/sDGP5-8WJoQ (speed: 3x)
How it works: This demonstration consists of having 2 Astrobees (Honey and Bumble), each of them running their own Finite State Machine. Their individual operation can be described:

- Bumble performs an inspection task, consisting of flying to 6 inspection points and taking pictures at each of these locations. If an anomaly is observed, such as an unknown object being detected, Bumble proceeds to send the object's geometry and pose information to the Unified Representation node via a service call (`attach_object_to_env`), making the data accessible by any of the other robots.
- Honey performs a monitoring task: While Bumble is inspecting, Honey periodically checks the world state messages published by the Unified Representation process. If a new object is added in the environment (presumably by Bumble), Honey uses the shared object's pose information to navigate towards the object, pick it up with its arm, and bring it back to a default Home pose.

Use of PLUMMRS:


(a) Bumble inspect first waypoint

(b) Bumble finds an object in second waypoint

(c) After Bumble leaves, Honey goes towards object

(d) Honey grabs object while Bumble continues inspecting

(e) Honey navigates back home with the object attached to its arm

(f) Honey stops, Bumble continues inspecting

Fig. 15 Demo 1: Bumble Astrobee finds an alien object during inspection and Honey removes it. Task is possible by sharing information using the UnifiedRepresentation, and safe motions are planned and executed using the information and feedback from Robot Arbiter (video available in https://youtu.be/sDGP5-8WJoQ)

- Honey obtains the object's shared information from the Unified Representation topic.
- Honey immediately attempts to plan a navigation path to reach the unknown object. As can be seen by comparing image Fig. 15b and c, this Astrobee only starts to move towards the object once Bumble is away from the object. This is not scripted. Rather, Honey's planner sends a `validate_navigation_path` request to the PlanLedger, which returns a false result since the other Astrobee is blocking the way. Honey's planner keeps resending the same plan until it is deemed valid, which happens when Bumble is far enough.
- Once Honey reaches the unknown object, it notifies the UnifiedRepresentation by making 2 service calls: (a) Detach object from environment, and (b) Attach object to Honey. As seen in Fig. 15d and e, the UnifiedRepresentation module accurately tracks the object as being appended to Bumble's arm.

4.2 Demonstration 2: Handover Task Between an Astrobee and Robonaut

Video of run: https://youtu.be/lv_HNaVcExc (speed: 2x)
How it works: This demonstration consists of having one Astrobee (Honey) transport a toolbox for Robonaut to pick up. All the activities in the video are managed by a single sequential Finite State Machine that alternates between states operating on Honey or on Robonaut. The task can be further described in detail:

- Honey is requested to approach Robonaut while holding a toolbox.
- Honey sends a service request (`attach_object_to_robot`) to the Unified Representation module, referencing the toolbox object.
- Honey plans a navigation path that avoids collision with Bumble, which is on standby (Fig. 16b–d).
- Honey reaches the vicinity of Robonaut. The latter consults with the Unified Representation module and obtains the location of the toolbox. It extends its arm to grasp it and then puts the toolbox away.

Use of PLUMMRS:

- The location of the toolbox is shared between robots by means of the UnifiedRepresentation module.
- When Honey plans a navigation path towards Robonaut, its initial path is a straight line between its start pose and the goal pose in Robonaut's vicinity. This path is modified by Honey's planner after being submitted to the PlanLedger for validation. The Arbiter returns a negative valid result and feedback regarding the source of collision (Bumble). The planner uses this information to plan around Bumble.
- Collision checking between the toolbox being carried by the Astrobee and the environment is performed by the Safety Monitor all through Honey and Robonaut motion, as they each carry the object.

- When Robonaut plans a reach motion towards the toolbox brought by Honey, it also replans its initial path in order to avoid collision with the Astrobee. This is done—in a similar manner to Astrobee's case—by using the Robot Arbiter feedback information. Both Astrobee and Robonaut use an RRT-based planner to replan around obstacles.

Observed limitations of the prototyped PLUMMRS:

- When passing objects between robots, the CollisionDetection module briefly indicates that the object is colliding with the robot that initially carried the object. The Safety Monitor alerts all robots of this event by publishing it on the safety topic, which might produce preempting calls that stop the motion of the participating robots. Distinctions between object transfers and actual collisions with the environment should be possible to indicate (this is not currently the case).

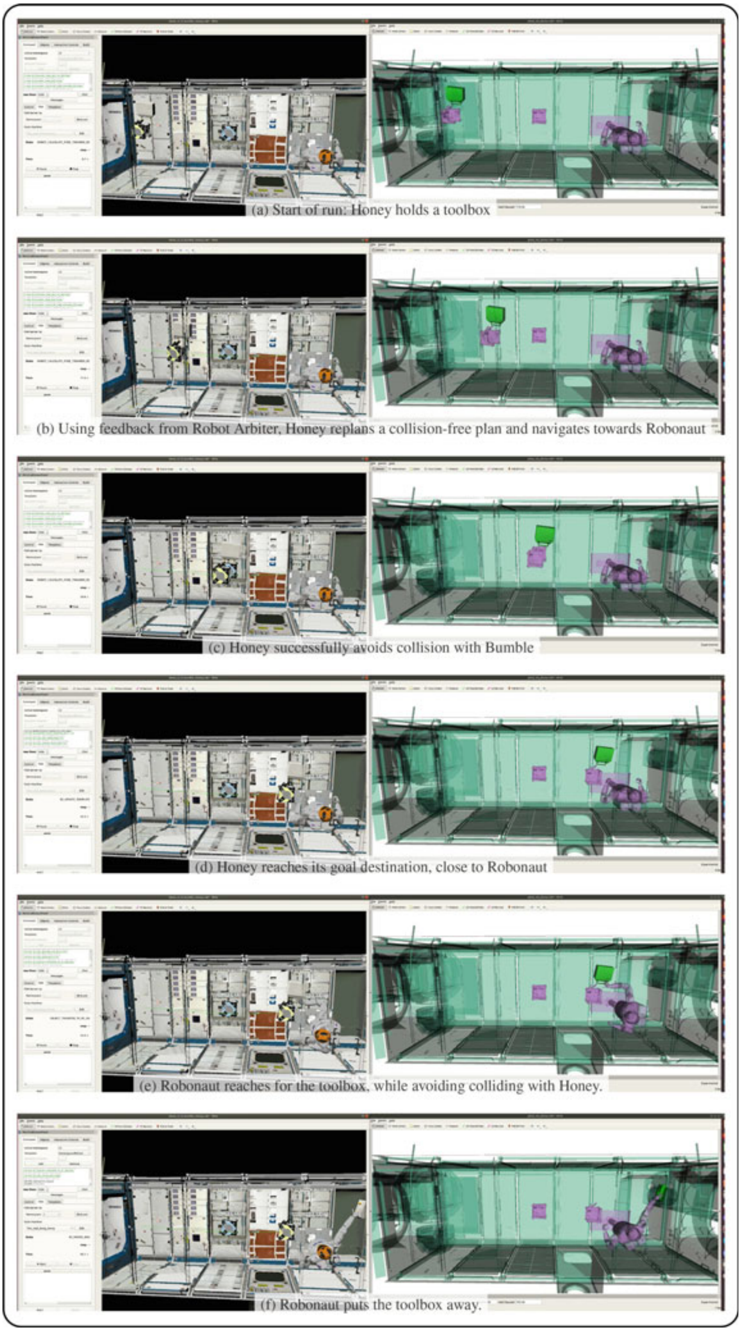
4.3 Demonstration 3: Astrobee and Robonaut Working Together to Localize and Pick up a Wrench

Video of run: https://youtu.be/ieJKT1b06R0 (speed: 2x)
How it works: This demonstration comprises Bumble and Robonaut working together to pick up a wrench. All the activities in the video are managed by a single sequential Finite State Machine that alternates between states operating on Bumble or on Robonaut. The task can be further described in detail as follows:

- Robonaut requires a wrench, which is within its workspace but not within its visual field.
- Bumble navigates towards an inspection point in Robonaut's workspace. It constantly monitors its camera stream, searching to localize a known model of the wrench (camera views are shown at upper right corner in right-side of snapshots depicted in Fig. 17).
- Wrench is found (Fig. 17c) and Robonaut reaches for it (Fig. 17d).
- Bumble navigates back to its home pose, whereas Robonaut brings the wrench closer to its body.

Use of PLUMMRS:

- Location of the wrench is shared by Bumble to the Unified Representation and then it is used by the Robonaut.
- Wrench geometry and pose are tracked by the Unified Representation and used for collision checks by both the Arbiter and Safety Monitor.
- Robonaut plans collision-free paths to reach the wrench such that it avoids hitting Bumble. As in the aforementioned demos, it does so by using the feedback information from the Robot Arbiter to replan its original plans.



(a) Start of run: Honey holds a toolbox

(b) Using feedback from Robot Arbiter, Honey replans a collision-free plan and navigates towards Robonaut

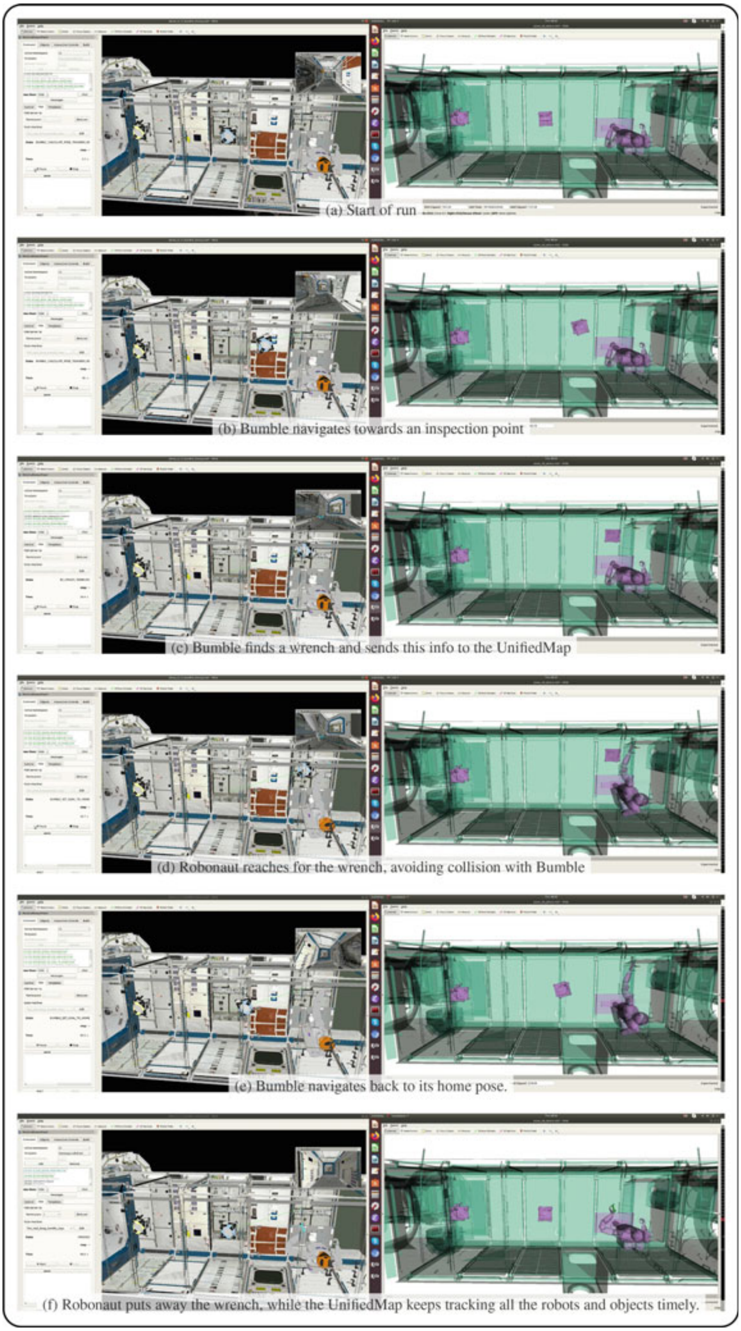
(c) Honey successfully avoids collision with Bumble

(d) Honey reaches its goal destination, close to Robonaut

(e) Robonaut reaches for the toolbox, while avoiding colliding with Honey.

(f) Robonaut puts the toolbox away.

Fig. 16 Demo 2: Honey Astrobee brings a toolbox to Robonaut, managing to avoid Bumble during navigation. Task is possible by sharing information using the UnifiedMap, and safe motions are planned and executed using the information and feedback from Robot Arbiter (video available in https://youtu.be/lv_HNaVcExc)



(a) Start of run

(b) Bumble navigates towards an inspection point

(c) Bumble finds a wrench and sends this info to the UnifiedMap

(d) Robonaut reaches for the wrench, avoiding collision with Bumble

(e) Bumble navigates back to its home pose.

(f) Robonaut puts away the wrench, while the UnifiedMap keeps tracking all the robots and objects timely.

Fig. 17 Demo 3: Bumble Astrobee navigates to an inspection point and finds a wrench, which is then grasped by Robonaut. Task is possible by sharing information of the newfound object using the UnifiedMap, and safe motions are planned and executed using the information and feedback from Robot Arbiter (video available in https://youtu.be/ieJKT1b06R0)

5 Conclusion

This chapter presented our proposed PLUMMRS framework consisting of three modules that allow a team of heterogeneous robots to share information in order to accomplish tasks safely. These modules were tested with different tasks in simulation (inspection and handoff), using two different robotic platforms (Astrobee and R2). Under the current state-of-the-art, these tasks would require an operator overseeing the operation, stopping one or more robot if an unsafe situation arises and restarting a task command. The experimental results of our prototyped framework suggest that the proposed PLUMMRS modules have real utility in multi-robot heterogeneous applications, and that (1) the need for a framework that allows multi-robot teams to share information and monitors safe motions exists, (2) such framework is a requirement for a higher level of automation, such as the one that will be needed for robots in space applications.

Acknowledgements This work was supported by NASA contract number 80NSSC20C0611 "A Collection of Plan Ledgers and Unified Maps for Multi-Robot Safety". The authors would like to thank José Benavides of the Ames Research Center, who provided feedback on the work described in this paper. The authors would also like to thank Brian Coltin and Marina Gouveia for providing assistance and guidance with the Astrobee simulation in Gazebo.

References

1. Adamek C (2019) Gateway system requirements. Tech. Rep. DSG-RQMT-001, NASA Johnson Space Center
2. Badger J, Gooding D, Ensley K, Hambuchen K, Thackston A (2016) ROS in space: a case study on robonaut, 2nd edn. Springer International Publishing
3. Barlow J, Smith E, Smith T, Bualat M, Fong T, Provencher C, Sanchez H (2016)Astrobee: a new platform for free-flying robotics on the international space station. In: Proceedings of the international symposium on artificial intelligence, robotics and automation in space (i-SAIRAS)
4. Beeson P, Hart S, Gee S (2016) Cartesian motion planning & task programming with CRAFTSMAN. In: RSS 2016 workshop on task and motion planning
5. Hart S, Dinh P, Hambuchen K (2015) The affordance template ros package for robot task programming. In: 2015 IEEE international conference on robotics and automation (ICRA). IEEE, pp 6227–6234
6. James J, Weng Y, Hart S, Beeson P, Burridge R (2015) Prophetic goal-space planning for human-in-the-loop mobile manipulation. In: Proceedings of the IEEE RAS humanoids conference
7. Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R, Ng AY (2009) Ros: an open-source robot operating system. In: ICRA workshop on open source software, vol 3. Kobe, Japan, 5p
8. Vaquero T, Troesch M, Chien S (2018) An approach for autonomous multi-rover collaboration for mars cave exploration: preliminary results. In: International symposium on artificial intelligence, robotics, and automation in space (i-SAIRAS)

Automated Software for Crewed Spacecraft—Bridging the Gap from Sci Fi to Reality

Robert C. Dempsey, Edward A. Van Cise, Michael L. Lammers, and Richard S. Jones

Abstract With a voice command or a few taps on the console, the spacecraft pivots on a dime at high velocity and gently docks to an orbiting space platform. This is the image most people have of the complex software computations and integrated hardware performance necessary for a spacecraft to successfully perform an automated launch, rendezvous, and docking. Today's reality is that while computer operations are advancing rapidly, science fiction over-simplifies and over-sells current capabilities. This paper discusses the integration of spacecraft computer automation into the operation of one of the United States' new Commercial Crew vehicles—the Boeing CST-100 *Starliner*. Lessons learned by the Boeing Mission Operations team, a unique private–public partnership with NASA, from conceptual design through real-time operation of the first test flight will be discussed along with evolution of the system to prepare for the second uncrewed test flight. Focus will center on how operations have learned to use the automated software to their advantage while also knowing how to adjust the automation in response to spacecraft anomalies. One goal of advanced spacecraft automation is the ability to reduce both the crew workload and the ground control footprint while at the same time increasing spacecraft and mission flexibility. Historically, crewed spacecraft required many operators on the ground to use a plethora of tools to compute nominal and contingency mission trajectories. Moving those sophisticated software tools to being onboard the vehicle can reduce the need for such complex ground support. Given that today's spacecraft software is not yet as capable or as flexible in all circumstances as the computers depicted in movies, there is usually a trade-off between software automation cost and the flexibility of that software resulting in compromises between what is performed on the

R. C. Dempsey (✉) · E. A. Van Cise · M. L. Lammers · R. S. Jones
Flight Directors, NASA Lyndon B. Johnson Space Center, Houston, TX 77058, USA
e-mail: robert.c.dempsey@nasa.gov

E. A. Van Cise
e-mail: edward.a.vancise@nasa.gov

M. L. Lammers
e-mail: michael.l.lammers@nasa.gov

R. S. Jones
e-mail: richard.s.jones@nasa.gov

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_7

spacecraft and what is left to onboard crew or ground control. An additional challenge discussed in this paper is the added complexity when the system is still evolving in a developmental program. For missions that go beyond the Moon, software that autonomously controls nearly every aspect of a crewed mission will become a necessity, given the long-time delays between the spacecraft and Earth's ground control teams. The lessons learned by Boeing and its Mission Operations team, through the design and implementation of *Starliner's* hardware and software automation, will be able to inform future public and private spacecraft design. As the technologies and capabilities evolve, incorporating lessons learned in successful low Earth orbit commercial crew vehicle missions, spacecraft designs will continue to improve and be able to better enable safe execution of human missions to the Moon and beyond.

Keywords Spacecraft automation · Human spaceflight · Spacecraft safety · Software configuration management

1 Introduction

Spacecraft in movies or books often easily perform amazing feats of computation, such as calculating real-time launch profiles, rendezvous trajectories and landings on any type of surface. As is often the case, the reality of today's spacecraft operations is far different. As on-board, radiation hardened computer process capacity continues to expand and mission requirements grow ever more challenging, automated operation of spacecraft is growing rapidly. All of today's spacecraft use some level of automated software, but it has largely been relegated to only performing functions associated with Fault/Failure Detection, Isolation and Recovery (FDIR). Routinely used for activity or observation scheduling, automation is usually a short-term plan of events built regularly during the mission by ground operators (e.g., [1]). Even in the most well-known examples of extraterrestrial probes, true automation is often related to specific mission phases (e.g., entry and landing) with humans still performing day-to-day exploration activities (e.g., [2]). Allowing software to control every aspect of spacecraft operations, including computing a rendezvous trajectory, performing proximity operations and docking with other crewed spacecraft, automated undocking both in nominal and off-nominal circumstances, and performing automated re-entry and landing is relatively new in the realm of human spaceflight. Automation in crewed spaceflight has evolved considerably from the 1960s, where computers aided the human operators to having the capability of performing every function during an entire mission. While Artificial Intelligence, where the onboard software evaluates its environment, assesses its mission plan and then self-determines the next action, offers significant advancement in automation, its use to date on spacecraft is limited [3]. The Russian "Soyuz" spacecraft, generally regarded as a mostly automated rendezvous vehicle, still requires tracking and ground uplink of target parameters and ground verification of rendezvous burns [4].

This paper does not discuss details of specific algorithmic approaches such as the proximity operations ones discussed by [5] but focuses on the development, integration, and operation of an automated system for a crewed vehicle, in particular, the *Starliner*. Below, we describe how the flight control team shaped the vehicle's automation and learned how to interface with such an equipped spacecraft. Although focused primarily on the Boeing CST-100 *Starliner* crewed spacecraft, this paper also relies on extensive experience with the International Space Station (ISS). *Starliner's* automated rendezvous and proximity operations were demonstrated by the Defense Advanced Research Projects Agency and NASA Orbital Express Project ([6–8]).

Boeing conducted an uncrewed Orbital Flight Test (OFT) in December 2019 to prepare for a Crewed Flight Test (CFT), then planned to occur in 2020. Designed to test all aspects of the hardware and software, the mission profile included launch on the United Launch Alliance (ULA) Atlas V, rendezvous with the ISS about 25 hours later, and return to the Earth using parachutes and landing on airbags at the White Sands Space Harbor in New Mexico. As originally envisioned, all of this mission was to be performed via autonomous software, with Mission Control largely playing an oversight role with minimal, periodic instructional uplinks (See also [9]). However, a timing error in the autonomous system caused a key orbital insertion burn following Atlas V separation to not take place as expected. Because of extensive training and preparation by Mission Control for various scenarios where the automation may not be able to function as originally planned, the control team in Houston was able to uplink a “manual” burn, placing *Starliner* in a stable orbit. Unfortunately, too much propellant was used to allow a rendezvous with the ISS. Boeing then elected to refly the uncrewed test, designated Orbital Flight Test-2 (OFT-2) before the CFT. This software error and its impact on the automated software, as well as the ways in which the ground team was able to intervene, is discussed below.

2 Unique Partnerships

Beginning in 2004 with the Commercial Space Launch Amendments Act, the United States began significantly expanding the role of private companies in many NASA programs. Perhaps the greatest success to date has been NASA's Commercial Orbital Transportation Services (COTS) and Commercial Resupply Services (CRS) programs, where Space Exploration Technologies Corporation's (SpaceX) cargo *Dragon* and Northrup Grumman's *Cygnus* routinely conduct cargo resupply missions to the ISS. Through a combined government investment of $5.9 billion for 31 flights to both companies on the CRS-1 contract, NASA spurred a revitalization of the American commercial launch industry [10]. Additional flights were purchased in the follow-up contract, CRS-2. Low cost and frequent launch capability resulting from the innovations and competition of these companies has benefitted not only NASA but other government satellite operators and commercial launch customers as well.

With these legislative changes helping to create a path, NASA's Commercial Crew Program (CCP) was created in 2010 to facilitate the development of a U.S. commercial crew space transportation capability to and from the ISS which could ultimately lead to the availability of commercial human spaceflight services for government and commercial customers. The CCP would also help facilitate the U.S. Based on the successful COTS program, NASA supported two development phases, Commercial Crew Development (CCDev) 1 and 2, where multiple private companies would develop key technologies for a crewed spacecraft. A key requirement was that NASA was not providing sole source funding but was partnering with companies who were expected to invest some of their own capital into the program. Instead of the final vehicle being U.S. property run by a contractor, the integrated system would be owned by the company and essentially leased to NASA. This unique relationship also meant that the private company retained maximum intellectual property rights while NASA would be granted limited rights use. NASA down selected to three companies – SpaceX, Boeing, and Sierra Nevada Corporation (now called Sierra Space)—in August 2012 to further develop fully integrated space transportation systems. On 16 September 2014, NASA selected SpaceX's *Dragon* and Boeing's CST-100 *Starliner* to demonstrate key capabilities, including ascent abort, an uncrewed space flight and a crewed test flight. In March 2019, SpaceX launched their uncrewed test vehicle followed by a crewed flight in May 2020. SpaceX began regular crewed flights in November 2020.

Adding yet another layer between public and private partnerships, Boeing opted to partner with the US Government for their *Starliner* operations by contracting with NASA's Mission Control teams in the Flight Operations Directorate (FOD) at the Johnson Space Center to create its own Mission Operations (MO) flight controllers. One reason Boeing selected FOD as its operations team was because NASA already had extensive infrastructure and processes in place to operate a crewed vehicle. Using a Reimbursable Space Act Agreement (RSAA) to effectively subcontract NASA civil servants, Boeing inherited FOD's methods for developing flight rules [11], procedures, crew training and extensive background in operating crewed vehicles. Crew and flight control operator training were also already in place.

An interesting and unanticipated benefit of this partnership with FOD was that Boeing had quick and easy access to a workforce that they could add or subtract quickly as need and budget allowed. For example, MO was leveraged by Boeing to help significantly in the design and development of the vehicle and software, something not anticipated when the RSAA was signed. The MO team even provided testing of some aspects of the integrated automated software. This level of involvement in design and development was something that the FOD team had never previously done, which expanded the FOD experience base and also allowed Boeing to quickly add "contractors" without having to put a separate provider on contract. This synergy also allowed the MO team to reduce their training since a significant amount of experience would be gained from direct participation in the spacecraft design.

This design and development involvement raised some interesting challenges within NASA and FOD. While any bidding company could have entered into an RSAA with NASA, only Boeing elected to do so. This added a significant challenge to NASA to ensure that proprietary data was not leaked from other companies to Boeing or vice versa.

Leveraging lessons learned through years of human spaceflight successes and failures, the FOD has developed a successful, methodical approach to human spaceflight operations. This approach has been relied upon in the development of space programs to help engineers and program managers locate potential weak areas of requirements or design, and then provide options for resolving the issues. With *Starliner* and the CCP, FOD was going to have to learn something new–spacecraft automation. Experience had taught MO that even with the best-designed hardware or software, failures can come in completely unanticipated ways.

With the intended automation of the spacecraft, the RSAA for the MO team provided a relatively small sized flight control team compared to the number of people utilized for previous spaceflight programs. This smaller team size would appeal to a commercial company as it would be fewer human resources that needed to be covered. It also appealed to FOD because it meant that the broader FOD workforce could remain engaged in other spaceflight programs, such as the ISS and beyond Earth exploration programs. To be even more appealing to Boeing's need to be budget conscious, the MO team was architected such that its size would reduce in number of people after the initial test flights. FOD personnel could also be trained to operate multiple spacecraft. There did not need to be as many people assigned 100% to Boeing MO because more flight controllers could be cross trained to operate *Starliner*, the ISS, Orion, or some other spacecraft, moving between vehicles as missions came close to launching.

One area of unanticipated consequence dealt with the oversight role that NASA provides to ensure that requirements were being met. With CST-100, MO was tasked with providing some of Boeing's data to NASA for this verification work. As part of NASA requirements review, that data was reviewed and assessed by non-MO members of the same NASA FOD organization. A healthy tension developed, given that different components of the same FOD organization that were providing and assessing the same data.

With 60 years of human spaceflight experience, including operations that resulted in the loss of 3 crews, FOD was a bit conservative in its approach. This would sometimes create minor disagreement with Boeing when MO appeared to "gold plate" the design or operations when Boeing felt what they had adequately met requirements. However, the personnel assigned to the *Starliner* program could operate with an open mind, and both sides would give and take, bringing the best of both viewpoints together. Other elements in NASA, within FOD and the CCP, would ensure that such debates did not compromise crew safety.

3 Development

In 2011, NASA outlined the requirements for its commercial crewed vehicles, stating at the outset that the integrated vehicle critical systems shall be autonomous [12].

At the same time, the crew would have the ability to override automation during all phases of flight. While this duality of requirements would lead to a robust and safe design, it also led to several challenges because of having to develop a system that could seamlessly transition between completely autonomous and fully manually controlled. Also, NASA initially required that all such crew actions would be performed by a single pilot, with all other astronauts serving only as passengers. NASA later decided to train a co-pilot to assist in all operations based on lessons learned from the fields of both Cockpit Resource Management and Spaceflight Resource Management [13–16]. The CCP requirement documents stated that fault tolerance for the control of catastrophic hazards would be based on analysis of hazards, failure modes, and associated risks. NASA defines a catastrophic hazard as "The <END ITEM > shall be designed such that no combination of two failures, or two operator errors, or one of each can cause a disabling or fatal personnel injury, or loss of the [spacecraft]" [17]. In general, dual fault tolerance or single fault tolerance with dissimilar redundancy for these controls was required. As Boeing designed the *Starliner*, crewed action during safety critical activities (e.g., ascent abort and docking) were sometimes utilized to meet NASA's safety fault tolerance requirements. By design, the crew would nominally not take any action during the entire flight. An exception to this rule was the release of the parachutes after landing, which resulted from a NASA concern that software could prematurely jettison the parachutes before touchdown was achieved.

In some contingency cases, the crew was the preferred response to a failure, most notably the execution of an ascent abort because of an issue with the *Starliner* spacecraft, such as a cabin leak. In this scenario, no spacecraft failure or automated response was deemed appropriate for initiating the complex, and potentially risky, abort sequence to separate the *Starliner* from the rocket and perform an emergency landing in the water and the crew would be in the best position to decide to abort. Instead, the ground would either direct the crew to abort or the capsule commander would do so based on situational awareness. The health of the launch vehicle is separately monitored by the Emergency Detection System (EDS) onboard the Atlas V and the EDS will take actions to initiate spacecraft aborts in time critical conditions where human response time would be inadequate, such as a catastrophic failure of the launch vehicle [18].

Early in the development of *Starliner*, Boeing selected MO to operate the vehicle, allowing the team that would ultimately control the spacecraft to participate in the vehicle's design. Because of the spacecraft being highly automated, the MO team was sized to be significantly smaller than previous NASA programs. For example, the ISS program started out with dozens of "front room" (i.e., the main controllers visible in video of the Mission Control Center, MCC) and "back room" (i.e., additional support controllers in other areas of MCC) that over the years were reduced and merged as the vehicle matured and reached "assembly complete" in 2011. In contrast, MO was envisioned to comprise about a dozen front room only operators with no additional back room support. Several consoles would only be staffed during specific phases (e.g., rendezvous) while the rest would have operators working two 13-h shifts. Some positions usually utilized in NASA missions, such as crew surgeon or the person who communicated with the astronauts, would be essentially government services supplied by NASA.

Starliner, by design, is a crewed spacecraft. Per the contract with NASA, Boeing was required to perform an uncrewed test flight prior to putting astronauts on the vehicle. This was the intent of the OFT. This presented a significant challenge to the development of the OFT mission. Per design, the crew was a "leg" in the fault tolerance. This meant that Boeing was either going to have to add additional hardware (e.g., an additional flight computer to provide additional redundancy), additional software (i.e., to provide an independent processing method), or require the ground to respond or perform additional testing and analysis. Since the OFT was a single mission, Boeing generally elected to not add additional hardware or software. In general, the ground was an adequate method for responding to contingencies, however, this did add additional requirements on a communication link with the small ground team and was only viable in cases where there was enough time-to-effect to allow the ground time to identify an issue and respond. Software changes were made for a few cases where the ground could not be expected to respond in time, such as performing a rendezvous abort if near the ISS and a single computer were to fail. Thus, for the OFT mission without a human crew onboard, oversight and management of several safety hazards in close proximity to the ISS took on the approach of utilizing a "three-legged stool" (see Fig. 1).

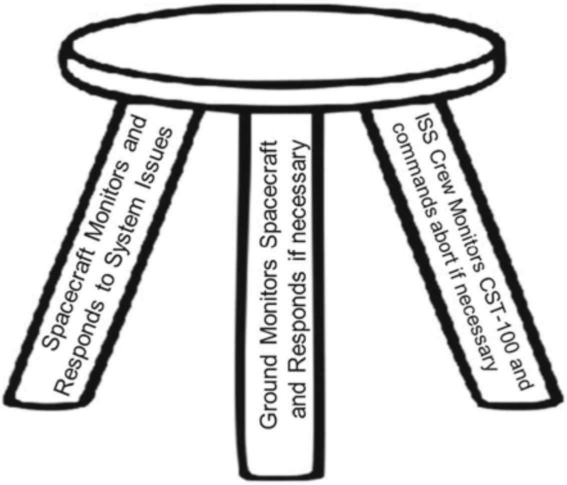

Spacecraft Monitors and Responds to System Issues
Ground Monitors Spacecraft and Responds if necessary
ISS Crew Monitors CST-100 and commands abort if necessary

Fig. 1 The OFT Three-Legged Stool utilized the onboard flight software as the primary means of detecting and responding to hardware or software faults. Without a crew onboard, the ground team would have also monitored the spacecraft and commanded the spacecraft to take recovery actions if it failed to do so on its own. During rendezvous with ISS, the ISS crew would have also monitored telemetry from the CST-100 and would command an abort if the spacecraft violated predefined limits

Generally, software performs stochastic or repeated operations on the vehicle, such as activating a fan or cooling pump to run with a specific value of revolutions per minute. Most spacecraft utilize a process of FDIR to identify failures and automatically perform an operation to either recover the function or at least disable the defective part/function. For example, if the above fan or cooling pump experiences an overcurrent condition, the software will detect the threshold violation, safe the offending unit (usually by deactivating it) and then activating an alternate fan or pump to maintain cooling. Such FDIR is not part of the discussion in this paper. *Starliner* employs more sophisticated automation of operations by creating a set of sequencers that control every aspect of operations from launch until post landing vehicle power down. Two sequencers controlled transitional and pointing operations and a third managed all general operations. All sequencers operated independently of each other but were designed to work together (for example, each sequencer had its own commands to conduct a maneuver or burn in a coordinated fashion). During the highly dynamic phase of atmospheric ascent, the spacecraft software generally is in a monitor mode only, conducting minor activities while the Atlas V controlled the launch vehicle and ascent trajectory.

Spacecraft actions were performed based on various trigger conditions such as range from the ISS, time since a previous action, altitude and so on. These sequencers executed the same commands that the ground control team could execute. In fact, it is the reliance on the Mission Elapsed Time (MET) trigger that caused the shortened OFT mission (cf. Sect. 5).

Mission Operations was involved in the design and testing of the software and sequencers. Using their many years of experience with crewed vehicles, the MO involvement was important in ensuring that the software not only met NASA and Boeing requirements but provided operability by the ground and the crew. Operability comprises two key areas: flexibility and usability. Through the years, the FOD flight controllers have learned that flexibility is often critical to mission success. Often, situations arise that were not foreseen during the design or development of the spacecraft or mission. These could range from deficiencies in designs not meeting specifications or, as in the most famous case of *Apollo 13*, in reaction to an unanticipated failure. In the case of *Starliner*, NASA requirements were still evolving as the CST-100 Critical Design Review was completing.

Usability has several aspects. Clearly, the crew and ground interfaces must be simple to use and intuitive. In terms of automation, usability is of limited import if the interface is properly designed. An example of where the interplay becomes significant is how easy it can be for the ground to perform an operation. For *Starliner,* some operations require exiting the automated sequencers, performing some action, and resuming the self-directed, autonomous control. This complex transition between ground control and autonomous control, in turn, limits such actions to when the vehicle is in a benign state (e.g., coasting) as opposed to dynamic (e.g., performing a rendezvous burn).

A key characteristic of the sequencers is the ability to modify their behavior. At any time during the mission, an entirely new flight plan could be uplinked to the spacecraft, though this was not actually a practical option. The reason for this is that to make a change to the plan would require careful development, including agreement among stakeholders the change was appropriate, followed by simulation and validation testing on a ground-based software test rig to ensure its safe operation in both nominal and numerous potential off-nominal conditions. This process would normally take many months. While theoretically feasible during flight, in practice, this was not considered anything other than an extreme contingency case due to the short (1 to 3 day) flight from launch to ISS docking and the risk of potential mistakes when changing the flight plan rapidly. Instead, the ground relied on behavior modification capabilities. For example, the ground was able to selectively inhibit an instruction. This might be needed if the concept of operations changes or to work around a hardware or software deficiency found late in development.

Another operator command allows the ground the ability to force the autonomous sequence to perform an instruction and move on to the next event, even if a logic condition is not met. This command was mainly a contingency case where the sequencer would otherwise hang up for some unexpected reason, such as if sensors failed to detect a condition the crew or ground knew was met. MO used this command in its toolbox to provide flexibility without changing software. For example, since the OFT was uncrewed NASA required the vehicle to perform certain demonstrations of safety measures (e.g., precise attitude control, station keeping, etc.) prior to entering the Keep Out Sphere (KOS) of the ISS [19]. The sequencer would force the *Starliner* to hold outside the KOS and when the ground teams were confident everything was working correctly, the MO team would issue a command to advance the sequencer to allow the vehicle to proceed closer.

During development, the MO team created several canned operational sequences. These included planned operations such as Far-field rendezvous, proximity operations, and the ability to fly a 360-degree loop around the ISS for visual inspection. MO also generated a few contingency sequences, such as rendezvous abort to make an emergency landing or stopping a rendezvous and going to a safe parking orbit, performing an unplanned translation maneuver, and so on. The software also had the ability to jump from one sequence into another, i.e., a software GoTo command. This allowed for quick transitions between the nominal timeline and, for instance, an emergency landing in case of a fire or an unexpected rapid depressurization. The ground, crew or automated software could make these jumps. When NASA-required demonstrations were added to the mission plan, it was easiest to GoTo the appropriate OFT-only sequence rather than rewrite the nominal sequence. For example, the abort sequence demonstration would be performed at a convenient time in the far field phase of the mission, rather than trying to insert a condition such as a system failure near the ISS. At the appropriate time, the MO team commanded *Starliner* to "GoTo" the abort sequence and when the sequence was complete, command it to resume the nominal mission sequence. Another lesson learned from the OFT mission was that when making such jumps, all other activities and sequencers had to be examined. Even though other sequences acting in parallel were ostensibly not correlated, spaceflight sometimes found unanticipated relationships that could potentially cause problems. Until Artificial Intelligence becomes de rigueur, automated systems will always have the challenge that they rely on the developers and operators to envision every possible scenario before execution.

A common design requirement for software is the ability to make changes. Of course, at any time the code itself can be changed but this is usually a time intensive, laborious process which involves careful testing and review. Normally such changes are made only if the software is completely not meeting a requirement. More adaptability is provided via a parameter file where thresholds, initialization values or even a list of commands can be updated in real-time. *Starliner* and ISS benefit significantly by this ability. Usually, the process to validate these files is easier and faster to perform, though, of course, with flexibility comes complexity. While the change of a given value might be easy to make, for example, changing a 0 to a 1, the process of validating the change might take a significant amount of time. The ISS can generally make these updates within 24-h (a capability which is used very sparingly and cautiously) while the dynamic nature of the CST-100 mission required updates to be available in four hours in some situations. Besides having multiple stakeholders verify the right parameter was changed to the correct value, the modified software might need to run in a simulator for several hours to verify that it performs as expected in all possible usage cases, nominal and off-nominal.

Sometimes, even this update flexibility was not fast enough for the operations team. For example, consider the communications link which, for *Starliner,* was primarily the NASA Tracking and Data Relay Satellite System (TDRSS). With many users, the TDRSS requires a great deal of integration in scheduling specific satellites, which typically takes several weeks. If a launch slips at the last minute or a mission timeline changes suddenly, either due to late priority change or to respond to a real-time contingency, adjusting the communications schedule can take much effort. When the schedule changes, the heavily utilized TDRSS time may not even be available at a critical time of need (e.g., docking) requiring negotiation with the NASA TDRSS network director. Satellite time might become available minutes before the needed time. Thus, the communication link schedule might need to be updated within hours or minutes of an event. Since the MO team performed the actual scheduling of the TDRSS, it made sense to provide them the ability to build and uplink the required file. The communications file, in this case, was a file that informed the spacecraft of all TDRSS events that had been scheduled so that the spacecraft knew which of its antennas to activate at specific times to ensure a solid communication link. Therefore, tools that were certified to build a validated TDRS schedule file were developed and implemented in the operations environment in Mission Control instead of in the offline flight software engineering environment.

A key decision early in the *Starliner* program was to have the operations team "own" the key input to the sequencers, a flat file that is essentially the English readable form of what the sequencers would run, not unlike ground timelines used in previous programs. This Sequence Command Input File (SCIF) allowed the operators to shape the mission plan as development progressed. For example, about a year before flight, MO conducted simulations with the full flight control team using the draft file. These simulations allowed the flight controllers to see how the integrated hardware and software performed. In this manner, early versions of Concepts of

Operations (CONOPS), flight rules ([11]), and procedures could be table-topped and evaluated. As experience was gained and mission requirements evolved, the input file was modified. This file is treated very similarly to the flight software. Therefore, its design and any changes must be carefully evaluated, impacted (e.g., there is cost to make an update) and tested or retested. This, in turn, limits the amount of flexibility that can be incorporated. Since the CST-100 program is a new one with a significant amount of discovery, there was a constant balance required to making updates as the team gained experience working with the automated system while preventing a perpetual series of updates and potential delays.

Significant evolution in the CONOPS for *Starliner* occurred with regard to calculating the orbital maneuvering burns. From the initial orbital insertion (OI) burn up through docking, *Starliner's* onboard software calculates a multiple sequence burn targeting plan that requires no interaction from the ground. For this automated vehicle, the ground was mainly supposed to support with contingency responses. Initially, there were limited options to adjust the burn plan. For example, if *Starliner's* trajectory were predicted to intercept orbital debris, an abort sequence would be commanded from the ground which took the vehicle out in front of but below ISS in a safe "parking orbit." From that orbit, the ground can recommence the rendezvous targeting a docking the next day. As the ground team began training this technique and became more familiar with both the limitations and the flexibilities of the sequences, it was realized alternate approaches could be utilized.

By leveraging the integrated training environment, the ground team subsequently developed several CONOPS and procedures to perform different debris avoidance maneuvers. If the effect to the overall rendezvous was minor, one option was to just put the vehicle into a coasting mode until the debris conjunction was safely in the past. Automation would then be resumed, and the onboard system would calculate an updated rendezvous trajectory and continue with the mission. Another option was for the ground to calculate a specific burn vector via a manual delta velocity (MDV) burn, where the ground calculates a burn vector and uplinks the values to a special sequence. This option, however, used more ground resources and leveraged heavily off the extensive experience of the MO team, which by design was small (see [9]). Thus, through leveraging of the training environment that included flight software and vehicle system models coupled with a ground control team that collectively leveraged nearly 60 years of human spaceflight successes and failures, a diverse set of capabilities were established for a wide variety of potential uses.

An additional challenge was presented by the NASA-required demonstrations of the rendezvous sensors. Although *Starliner's* sensors were based on the Vision-based Software for Track, Attitude, and Ranging (Vis-STAR) system developed for Orbital Express [6], NASA required careful validation prior to completely relying on them for a docking with the ISS. Redundant hardware on *Starliner* provides fault tolerance to system failure, while robust software algorithm with extensive testing and simulation, provide confidence in the overall safety of the system. Even then, the crew provides a cross-check that the system was operating correctly. For example, if the navigation system indicated the target, in this case the ISS, was in the center of the field of view but the crew saw instead the vehicle off to one side, the crew would possibly take an action to correct the vehicle's trajectory. On the OFT mission, there was no crew to perform this vetting. Adding onetime only software changes was not required since tools on the ground could compensate. Instead, the flight controllers on the ground combined raw measurements of where the spacecraft sensors measured the position of the target along with *Starliner's* trajectory observed from the Global Positioning System (GPS) run through independent navigation software to independently verify that the onboard system was correctly and safely heading to the ISS.

Note that even with an automated spacecraft, there are cases where significant calculation on the ground is still required. A good example is the flight dynamics process. Unlike NASA's Space Shuttle which had a multiple minute launch window on ISS missions, the Atlas V used for *Starliner* required an instantaneous launch window because of the design and certification of the launch abort system. Calculating the launch through docking timeline is relatively straightforward until considerations such as timing the arrival time to the ISS so that the crew was available to perform abort demonstrations and allowing for trajectory dispersions of the launch vehicle were added to mix. Variables such as weather at the launch site, docking port availability at the ISS, and launch scrub turn around requirements for the vehicles and cargo cause the need to calculate many trajectories. Early in the RSAA, this task was outsourced to the MO flight dynamics team, which strained the small nature of the team.

4 Training

Traditionally, NASA human spaceflight flight control team training encompassed individual flight control team positions learning about how the various systems, subsystems, and components within their assigned area of responsibility functioned (see [20]). These systems can range from life support to communications, guidance, navigation, and control, flight dynamics, power systems, mechanical systems, rendezvous operations, the flight plan, and so on. Training included the actual functionality as well as how the various piece parts integrated together, not just within that system, but with the spacecraft as a whole. The flight controller needs to understand how their portion of the spacecraft affects, influences, and achieves the various mission objectives. The knowledge needs to scope not only the planned mission, but also any myriad of failure situations. The flight controllers are always learning and assessing from the perspective of crew safety, vehicle safety, and mission success—in that priority order. For human spaceflight missions, the flight controllers must also learn how the onboard crews interact with their systems, how they are affected by it, and how the crew can be utilized to help the system in various failure cases. Because of the development nature and complexity of every crewed space system, NASA flight controllers are typically deeply involved in discovering and solving operational and system engineering problems unanticipated by the hardware and software design teams.

One of the most useful activities for the operations team was to conduct simulations using the Training Simulation Integration Laboratory (TSIL). Ostensibly, the TSIL environment is designed to provide a flight like real time simulation to train the crew in the cockpit and ground operators in the MCC. Since the TSIL ran flight software in a mostly plug-and-play mode, on multiple sessions, with an easy to operate user interface, it also provided an excellent opportunity to stress the software in a fully integrated fashion and in smaller research sessions. Although many of the hardware systems in TSIL utilize low fidelity models, the facility provides invaluable opportunity to see how the system performed. While this opportunity for insight could be made about many integrated space systems, it is especially critical for highly automated crewed vehicles.

As automated spacecraft come into reality, flight controller training has had to change and adapt to account for and include this automation. In previous spacecraft, the ground team directly controlled nearly everything that the spacecraft was doing, either because the ground directly commanded functions to occur as is the case of the ISS or because the ground worked with the onboard crew to take specific actions (e.g., the Space Shuttle).

The ISS now uses an automation engine, known as Timeliner [21], on core systems in a simple implementation that monitors for specific failure conditions and takes specific actions. A more extensive use is in place on non-critical payload systems. Stepping up to the use of, and reliance on, this simple software set took several years with guarded step-by-step progression. For CST-100, the move to a fully automated spacecraft transpired quickly at the very outset of the design and the flight controllers had to learn not only what automation meant and looked like but then also how to augment their training to incorporate and ultimately exploit the automation to ensure crew safety, vehicle safety, and mission success.

By the time the OFT flew, the flight controllers had developed a significant depth of knowledge of all the automated processes that the spacecraft could utilize. Each system discipline then moved on to understand how their system could be affected by the course of automation's sequencing, especially if the sequencing deviated from the original planning. In failure cases, the flight controllers must have solid knowledge and foresight of how a failure may cause the automation to stop functioning properly because it is waiting on a trigger condition from hardware that it not operating (and would thus continue to sit and wait unless told by the ground or onboard crew to continue).

As preparations for *Starliner's* OFT progressed, the flight controller training revealed that there was an additional need in the mission control room. There needed to be a position that specifically was trained on the overall functionality and integration of the automation itself. This new assignment fell to the Flight Activities Officer (FAO) position, the position that traditionally built ground and crew timelines on previous missions. The FAO team built expertise and console tools to watch over the progression of the automation. They also developed the ability and knowledge to determine where the automation may have complications or stop working in the future due to failure conditions in the spacecraft. A key finding from much of this training was that in the face of contingency, the operations team might have to inhibit steps in the sequencers and perform functions manually. This might occur because a failure might cause a trigger to get "stuck" or the automation could potentially power off the last unit in a redundant system.

A significant finding in the first years of training for *Starliner* operations was developing the techniques to integrate knowledge of the automation's functionality into the overall cadence of operating the mission. This training also led the flight control and training teams developing the ability to stress, poke, and prod on the automation in order to find both its strengths and weaknesses. Weaknesses were passed back to the engineering teams to determine if improvements or changes should be made. Strengths were documented and leveraged so that the flight controllers could develop techniques to utilize those strengths in a variety of ways. These additional operational uses were not initially planned when the software was developed but showed (in training and during the OFT mission itself) to be extremely useful in helping the ground teams work through mission problems not initially foreseen.

This training merged well with MO's role in software development mentioned earlier. Since MO owned the SCIF, there was a great deal of synergy between training and its development. Often, the operations team would take lessons learned from a simulation and make significant modifications to the SCIF. However, every update had to be carefully reviewed, its impacts to the flight software development schedule evaluated and testing plan updated.

While this approach was a significant advantage for ground and crew training, software development, and actual mission execution, a disadvantage was that MO's role in performing these tasks was not clearly understood or anticipated in the early stages of development of the *Starliner* program when work was scoped, and manpower and other resources were allocated. Thus, MO was continually assessing the relative priorities of the SCIF or other software changes (and the associated testing and verification) against the cost in time, personnel, and other resources that were already targeted to be used for other aspects of the program. See also [9].

Although the focus was primarily on the OFT mission, training simulations with the CFT crew were also conducted in parallel. Besides providing some early training to the crew, this provided two significant benefits: allowing lessons learned from crew interaction to be folded into the SCIF early on and it ensured the OFT mission reflected a planned crewed timeline as much as possible. The closer OFT was to the CFT mission, the more realistic the OFT test flight would be and fewer changes would have to be made between flights. However, since some software development (e.g., crew displays) were not scheduled until after OFT, there were limits in this approach.

Additionally, as a completely software-based simulator, trades are made by simulator developers on the level of complexity to build into the simulation models of hardware components. Software running in a power controller, for example, can be emulated in a relatively straightforward manner. Software running in an inertial measurement unit, however, may approach that of the vehicle flight computer itself in complexity. Being able to properly explore interactions between automation on the flight computers and the hardware the automation is controlling often depends on the fidelity of the hardware simulations.

This presents two challenges the training teams must always consider. One is that the software, which includes the various automation sequences and files, in the simulator may be lagging the software planned for use in flight and the teams must ensure they understand the differences between simulator and flight software. The second is that configuration management of the various training loads (which include the flight software as well as all the various model parameters) must be maintained so that the training teams, flight control teams, and crews always understand how closely their training configuration matches what will actually fly on a given mission.

A critical lesson learned during training was that automation does not always behave as expected, especially when faced with failures. As mentioned above, the ground operators were involved in the software's design from nearly the beginning of the program. With MO having extensive, recent experience with the Space Shuttle and the ISS, sequences were developed based on lessons learned in these programs. Of course, there are significant differences between these vehicles and *Starliner*. For example, the Space Shuttle could take as many as three days to land after undocking from the ISS to touchdown, enjoyed numerous landing sites around the globe and with its wing structure had a significant amount of cross range allowing "steerability" from many orbits to a landing opportunity offset over a thousand kilometers from its orbit ground track. On the other hand, *Starliner* touches down only four to five hours after undocking, has a much smaller entry cross range, and has only a few landing sites concentrated in the Southwest continental United States.

In one simulation of the undocking to landing phase, due to a failure, one step in the main sequencer was inhibited since it would not occur correctly. Unrealized at the time, the step inhibited was applied to a trigger tied to a specific event, with subsequent commands set to follow as soon as that (now inhibited trigger) was satisfied. Therefore, when the desired step was inhibited, the subsequent commands executed right away, unexpectedly, until the sequencer hit the next unmet trigger condition because the trigger that normally forced the automation to wait had been inhibited. This forced the flight control team to manually back out of some entry configuration that had been entered prematurely. See Fig. 2. While it seems intuitively obvious, when there are a large number of sequences with various permutations in the way they can be ordered, it may not be obvious how changing the sequencer impacts the overall flow. Other similar issues arose from use of the GoTo command. It took several full-team integrated simulations to identify these pitfalls and develop techniques to identify them in advance.

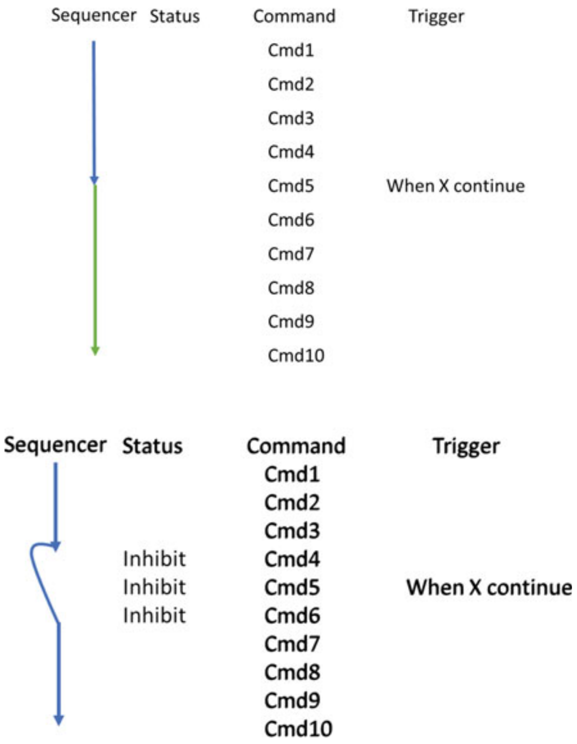


Fig. 2 Normally, commands (Cmd) would execute one after another as shown in the top (blue line). At Cmd5, the sequencer would stop until the trigger condition X occurs. Once X occurred, the sequence would continue (green line). However, as shown in the bottom graphic, when several commands were quickly inhibited in the simulation, the sequencer did not pause and, as should have been expected, skipped over the inhibited commands without pause

5 Execution—the Orbital Flight Test

Starliner's uncrewed OFT lifted off from the Kennedy Space Center on 20 December, 2019 at 11:36 UTC. In general, the automation worked as expected; however, a flight software error in the calculation of the MET caused a series of events to occur that forced the operations team to make some real time changes. While most of the mission objectives were achieved, a key one–docking with the ISS–could not be performed.

The MET clock begins counting up from zero at liftoff, maintaining a running time for the entire mission. Normally, it is just another way to track time during the mission when other measures such as UTC become cumbersome. With *Starliner,* the only place MET is directly used during the mission is to perform the Orbital Insertion (OI) burn. A significant part of the spacecraft's mass is the Launch Abort System and the propellant it uses to remove the Crew Module from the rocket with catastrophic failures. Once beyond the need for an ascent abort and after separation from the Atlas V, the Orbital Maneuvering and Abort Control (OMAC) engines are used to perform the OI burn, putting the *Starliner* into a sustainable orbit.

To simplify the trajectory calculations, the OI burn was designed to occur at a fixed MET. Multiple events in different sequencers are designed to perform orbital insertion. First, the guidance software must calculate where in the orbit the Atlas V has deposited the spacecraft (which for the OFT was exactly as targeted) and what translational burn maneuver is needed to put the *Starliner* into orbit. The burn calculation for the OI maneuver also sets up all the subsequent burns all the way to docking. The translational and attitude pointing sequencers ensure that the spacecraft is oriented in the correct thrust vector alignment mode.

After launch vehicle separation, the controlling sequencer configured for orbital operations and set the OI time of ignition to the planned time. Unrealized at the time to the Boeing team, a software error actually initialized the MET clock, not at liftoff but approximately 11 hours prior to launch. Thus, the burn was calculated to be in the past, as was the maneuver to thrust alignment.

The ground team executed a contingency plan, uplinking the MDV command designating alternate burn target parameters. Performing a manual burn that remains within the capabilities of the onboard automation requires several commands from the ground to get the various sequencers synchronized. Because of complications with commanding caused by intermittent satellite communication with *Starliner* caused by various sources of interference, the MO team was unable to completely synchronize the translation and pointing sequencers before the time of ignition of the contingency MDV sequence. Therefore, the commanded burn initially started off performing a coarse multi-axis thruster burn while maintaining a fine attitude, keeping its solar arrays tightly pointed at the sun. The ground team's attempt to improve the *Starliner's* burn orientation to better align the intended thrust direction with the *Starliner's* OMAC engines was not completely successful due to the difficulty of getting the commands onboard at the right time because of the same issues with the communication link. Both OMAC and reaction control system engines continued to fire in a non-optimal vehicle attitude, expending extra propellant that was normally allocated to dock with the ISS. Once the vehicle had achieved a safe orbital altitude, the ground team intervened for a last time to disengage the manual maneuver and prevent further inefficient consumption of the remaining propellant. Approximately one orbital revolution later, the ground commanded another burn, successfully raising the *Starliner's* perigee by a small amount to prove that the *Starliner's* translational maneuver capability had been fully recovered. See Fig. 3.

Fig. 3 (Left) From Left to right flight controllers Joe Jones, Ramon Gonzalez and Carson Sparks discussing how to perform the OI burn. (Right) Flight Directors Mike Lammers (left) and Richard Jones (right) look out over the control room as the team uplinked the MDV maneuver

At this point, *Starliner* was in a stable orbit. However, the higher than expected propellant usage made getting to the ISS and back to the ground just barely out of reach, and the rendezvous was terminated. The mission was replanned with the decision to target the landing opportunity that was available for two days into the mission. Competition between the coarse and fine attitude control also caused the thrusters to fire significantly more than intended cause some hardware degradation. With the mission duration significantly reduced, the flight control team attempted to complete as many flight objectives as possible.

One such objective involved demonstrating the ability for the vehicle to perform a fully autonomous abort near the ISS. The objective was to test the capability far away from ISS to develop confidence for safe execution, if needed, near the space station. The abort sequence would perform a series of predefined burns. By design of the test occurring in the far field phase of the trajectory, the spacecraft systems were not configured for full redundancy, as would be the case close to the ISS. During performance of this test, one of the thrusters performed erratically due to excessive wear and tear induced trying to complete the initial orbit insertion burns. Although the spacecraft likely would have safely retreated from the ISS, the total expected delta velocity fell slightly short of the target velocity.

When the radio link for the ship-to-ship (S/S) radio between *Starliner* and ISS performed much better than expected, locking up at 650 km versus the expected 10 km, an objective of opportunity was attempted by trying to send a command from the ISS to the CST-100 to turn on its navigation lights. This demonstration is a key capability that is tested before entering the proximity of the ISS to ensure that, if required, the ISS crew could issue an abort command. Although the command was transmitted and clearly received, it was not accepted and executed.

After *Starliner* made it to orbit following the MET anomaly, Boeing, with NASA help, quickly reviewed other critical phases of flight for potential problems. This review resulted in the team discovering and correcting a software issue that would have manifested during the *Starliner's* crew and service module separation sequence. Because of some incorrect channelization between the computers and thrusters, the autonomous disposal sequence would not have executed per analysis. Uncorrected, there was a potential that the modules would collide during the separation sequence. Identifying the error was one thing, but updated software loads had to be built and tested to ensure it functioned well when run inside all of *Starliner's* automation. The operations team then had to carefully update the flight software shortly before execution.

On December 22, 2019, the MO team engaged the entry automation sequence to bring *Starliner* home during its 33rd orbit of the Earth. *Starliner* proceeded to target and execute a fully autonomous deorbit burn and atmospheric re-entry and parachute landing at White Sands Space Harbor in New Mexico. The only human intervention performed was a planned operator command to separate the parachutes following landing (a function normally performed by the crew when onboard).

6 Post-Flight Software Improvements and Orbital Flight Test-2

Since several significant software problems occurred during the OFT mission, an Independent Review Team (IRT) with Boeing and NASA personnel was formed to assess all aspects of the flight software. Three major anomalies were the focus of their review: the MET clock, the Service Module disposal burn, and the communications link. The IRT recommended 61 changes to the flight software process and 19 changes to the communication system, all of which were implemented by Boeing. Many of the recommendations addressed processes including using higher fidelity hardware in software testing and performing more End-to-End type tests. Insufficient interface testing was listed as the root cause with the incorrectly set MET. Improved testing could verify the software was properly fixed, but it showed that significant vulnerabilities to the automated sequences were still present.

In parallel, the MO team looked at ways to improve robustness of the sequences. The pointers and other methods used to maintain synchronization between various sequencers were assessed and improved. Efforts were also made to make the automation more robust to loss of the communication link with Mission Control. Examination of the autonomous abort test sequence that was performed indicated room for two improvements. One was realizing that if the system performance is degraded, one burn may not be complete before automation would start the next burn. This would be possible because each burn had a prescribed delta velocity and the software would extend the burn duration to meet that velocity value. With under-performing thrusters, the delta velocity might not be met before the timer expired for the next burn to initiate, resulting in failure to accomplish the previous burn. Of course, in the situation where an abort might be needed the vehicle would be configured for full redundancy, meaning multiple failures would first need to occur to need the benefit of this change. The second improvement was the addition of an attitude maneuver at the beginning of the test sequence to ensure as much of the thrust was in the direction opposite of the ISS (in this reference frame, the negative velocity vector direction). Figure 4 shows the improvements.

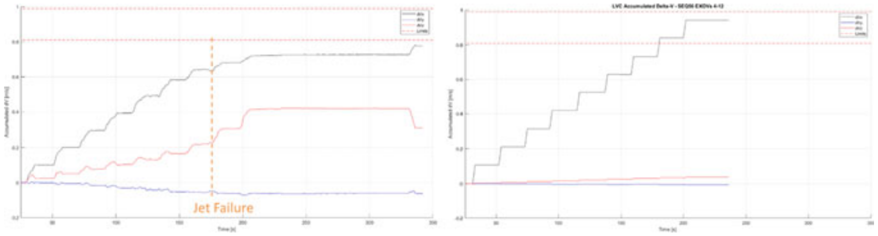


Fig. 4 (Left) Performance of the simulated abort burn during the OFT mission with the delta velocity in each direction indicated by the different colored lines. The black line is the axis of the preferred thrust direction. The orange failure indicates where an erratic thruster sensor led to degraded performance. (Right) modified sequence to be used on OT-2, which allows pauses in between burns and added an attitude maneuver to ensure the thrust would predominantly be in the direction away from ISS

The issue with the S/S command failing was fairly straightforward. Since the radio link was only intended to work in close proximity to the space station, commands were blocked from executing at large ranges. This again illustrates the challenges of trying to move activities around in a highly automated system. With no safety implications, this range restriction was removed from the software for OFT-2 and subsequent missions.

On April 6, 2020, Boeing announced they would re-fly the uncrewed test, designated OFT-2, rather than proceed directly to the CFT. This allowed the changes from the IRT to be implemented before the next flight. There was another benefit from the delay in the CFT: more simulations for the MO team. To keep costs low and because the vehicle is highly automated, only a few integrated simulations had been planned. The extra time allowed for more training opportunities, which were also needed to exercise the post-IRT modified software.

The OFT-2 mission was scheduled to launch in August 2021. However, a hardware problem with some Service Module thruster valves caused a rollback of the spacecraft and delay in the mission to another time.

7 Summary

Although automated spacecraft operations allow safe operations for the crew, it is critical that flexibility and appropriate interfaces with the ground team be incorporated in the design from the start.

With the *Starliner*, which is designed to ferry crews between Earth and ISS, reliability is crucial to ensure the safety of the crew under extreme conditions including returning deconditioned astronauts, who have been in space for six-months or more.

In addition, there is a significant risk to both *Starliner* and the ISS (the ISS crew and the multibillion-dollar advanced laboratory) should a collision between the two vehicles occur. The risks associated with ensuring crew and vehicle safety mandate a robust verification and validation process of the *Starliner*, including its automated software.

A key realization when determining the extent to automate a spacecraft is to discern when it is most beneficial to use the full capabilities of the automation software and when it is best to bypass the software and rely on the human operator (either the onboard crew or ground-based control team). A key factor in this decision-making process includes time criticality of actions, and the impact of responses. Computing and calculating an orbital adjustment burn requires a quick calculation, lending itself to computer automation to perform the task. However, an incorrect burn computation could have catastrophic repercussions if it changes the trajectory to put it on an intercept path to the ISS. Further adding to the consideration is the extent to which flight crew must have the ability to seamlessly manually intervene on actions the automation is taking. The incorrect orbit insertion and subsequent flight control override to achieve orbit provided a reminder that, especially in a first flight situation, where human intervention can be critical.

Although rendezvous was not performed on the OFT, the development of the ground tools to independently verify the navigation system shows that, in certain circumstances, a robust ground control team is still required with an automated vehicle under nominal situations. Otherwise, further software or hardware may be required in critical situations like an uncrewed vehicle docking with the ISS.

Although the clock issue on the OI burn for OFT was a relatively straightforward software error, its impact on the automated system was significant. Each sequencer performed as expected given the erroneous software inputs, but the integrated outcome of the sequencers being out of sync had not anticipated an error, such as the burn time being in the past. Sync points or cross checks, used elsewhere in the software, were not in place here but will be in place for the next flight.

The experiences of the OFT mission demonstrated the importance and usefulness of having the operations teams closely involved in the design, development, and testing of the CST-100. In theory, Boeing could have brought the MO team late to the development phase of the program. Boeing's investment in the RSAA with NASA was well served by being able to leverage the MO team's decades of prior spaceflight operations experience. Not only did this provide operational insight into the design, which increased mission flexibility, it also provided a mechanism for integrating vehicle systems and information across the various Boeing engineering and design groups. Given that MO team would fly the integrated spacecraft, MO team's knowledge of various system developments and challenges helped inform operations and design changes to other elements of the spacecraft.

One goal of advanced spacecraft automation is the ability to reduce both the crew workload and the ground control footprint. Development of *Starliner,* to include its first test flight, has shown that significant progress has been made in this area. However, additional requirements, such as programmatically required demonstration tests, unanticipated aspects of the automated software, and flexibility in preparing for multiple launch windows can limit these improvements initially. As flights continue, with the CFT and then regular crewed missions to the ISS, continued improvements and enhancements will occur as more experience is gained.

For missions that go beyond the Moon, software that autonomously controls nearly every aspect of a crewed mission will become a necessity given the large time delays between the spacecraft and Earth's ground control teams. The lessons learned by Boeing and its MO team, through the design and implementation of *Starliner's* hardware and software automation, will be able to continue to inform future public and private spacecraft design. As the technologies and capabilities evolve, incorporating lessons learned in successful low Earth orbit commercial crew vehicle missions, spacecraft designs will continue to improve and be able to better enable safe execution of human missions to the Moon and beyond. The fully autonomous spacecraft depicted in science fiction is not quite here yet, but swift progress is certainly being made towards it.

Acknowledgements The authors wish to thank Chris Ferguson, Steve Gauvain, Pooja Jesrani, Ray Bigonesse and Rosie for their help in creating this paper.

Appendix

Acronyms/Abbreviations

CCP	Commercial Crew Program
CCDev	Commercial Crew Development
CONOPS	Concept of Operations
UTC	Coordinated Universal Time
CFT	Crewed Flight Test
EDS	Emergency Detection System
FDIR	Fault/Failure Detection, Isolation, and Recovery
FAO	Flight Activities Officer
FOD	Flight Operations Directorate
GPS	Global Positioning System
KOS	Keep Out Sphere
IRT	Independent Review Team
ISS	International Space Station
OFT	Orbital Flight Test
OFT-2	Orbital Flight Test-2
OI	Orbital Insertion
OMA	COrbital Maneuvering and Abort Control
MDV	Manual Delta Velocity
MCC	Mission Control Center
MET	Mission Elapsed Time
MO	Mission Operations

RSAA	Reimbursable Space Act Agreement
SCIF	Sequence Command Input File
TDRSS	Tracking Data and Relay Satellite System
TSIL	Training Simulation Integration Laboratory
ULA	United Launch Alliance
Vis-STAR	Vision-based Software for Track, Attitude, and Ranging

References

1. Fukunaga AS, Rabideau G, Chien S, David Y (1997) ASPEN: a framework for automated planning and scheduling of spacecraft control and operations, In: Proceedings of the international symposium on AI, robotics and automation in space (i-SAIRAS). Tokyo, Japan
2. Ferri P, Sørensen EM (1998) Automated mission operations for Rosetta. In: Proceeding of the fifth international symposium on space mission operations and ground data system: spaceops, vol 98. 10.1.1.547.2575&rep=rep1&type=pdf
3. Straub J et al (2013) Application of collaborative autonomous control and the open prototype for educational NanoSats framework to enable orbital capabilities for developing nations. In: Proceedings of the 64th international astronautical congress
4. Murtazin R, Petrov N (2012) Acta Astronautica 7:77–82
5. Schulte PZ, Spencer D (2016) Acta Astronautica 118:168–186
6. Leinz MR, Chen CT, Beaven MW (2008) Orbital express autonomous rendezvous and capture sensor system (ARCSS) flight test results. In: International society for optics and photonics SPIE defense and security symposium, vol 6958(8)
7. Mulder T (2008) Orbital express autonomous rendezvous and capture flight operations, part 2 of 2: AR&C exercise 4, 5, and end-of-life. In: AIAA/AAS astrodynamics specialist conference and exhibit
8. Mulder TA (2008) Orbital express autonomous rendezvous and capture flight operations, part 2 of 2: AR&C exercise 4, 5, and end-of-life. In: AIAA/AAS astrodynamics specialist conference and exhibit
9. Dempsey RC, Van Cise E, Lammers M, Jones R (2021) Operating a crewed spacecraft in the age of commercial space using private/government partnership. SpaceOps16,x1231
10. NASA Office of Inspector General, Office of Audits (2018) 26 April 2018 IG-18-016 (A-17-013-00) https://oig.nasa.gov/docs/IG-18-016.pdf#page=3
11. Herd A, Dempsey D (2013) Flight rules: purpose and use. In: Sgobba T, Allahdadi FA, Rongier I, Wilde PD (eds) Safety design for space operations. The International Association for the Advancement of Space Safety, Butterworth-Heinemann
12. CCT-REQ-1130, NASA ISS Crew transportation and services requirements document
13. Kanki BG, Helmreich RL, Anca JM (2010) Science direct. Academic Press/Elsevier, Amsterdam. Internet resource, http://www.sciencedirect.com/science/book/9780123749468
14. Rogers D (2002) NASA's space flight resource management program: a successful human performance error management program. In: AIAA SpaceOps 2002 conference. AIAA , Houston, Texas, 9–12 Oct 2002. https://doi.org/10.2514/6.2002-T4-12
15. O'Keefe W (2008) Space flight resource management training for international space station flight controllers. In: AIAA space 2008 conference & exposition. AIAA, San Diego, California, 9–11 Sept 2008
16. Baldwin E (2008) Integrating space flight resource management skills into technical lessons for international space station flight controller training. In: Proceedings of the 3rd annual conference of the international association for the advancement of space safety. Rome
17. SSP 50021 Safety Requirements Document

18. Holguin M, Herbella G, Mingee R (2010) Commercial crew launch emergency detection system the key technology for human rating EELV. In: Proceedings of the AIAA space 2010 conference & exposition. Anaheim, California. AIAA 2010-8670, https://arc.aiaa.org/doi/pdf/https://doi.org/10.2514/6.2010-8670
19. Koons DS, Schreiber C (2010) Risk mitigation approach to commercial resupply to the international space station
20. The International Space Station-Operating an Outpost in the New Frontier (2018) Gov. Printing Office. Dempsey R (ed) https://go.usa.gov/xQbvH
21. Brown R, Braunstein E, Brunet R, Grace R, Vu T, Zimpfer D, Dwyer W (2002) Timeliner: automating procedures on the ISS. https://ntrs.nasa.gov/search.jsp?R=20100036766, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100036766.pdf

Designing a Console for Future Space Operations

Alexander Gerald Seidel

Abstract The launch of SpaceX's Crew Dragon Demo-2 Capsule generated a wave of new technology and inspiration for innovation within the space industry, especially in function and design. One major difference observed by those that watched the launch was the prominent use of a touch screen console. The question this research addresses: are touch screen consoles superior to those that use mechanical switches and controls? With implementing new technology, concerns have been made about both the safety of the new design and its efficiency. Through the utilization of journal articles and public information released by space agencies, this research focuses in survey of current and past console design through an industrial engineering lens. As there are many factors to consider that are atypical to traditional touch screen when considered for use in a spacecraft including g-forces, vibrations, redundancy requirements and user traditionally wearing multi-layer protective equipment, many aspects of the usefulness were evaluated. Additionally, assumptions about current technological advancements like autonomous navigation are considered being a part of the ultimate spacecraft system. Recommendations on how the console design should be conducted in the future are included in this paper. Ultimately, it is determined that a hybrid solution would be the best path for a dual-fault tolerant system. This would allow for the infusion of technology, enabling a more diverse space traveler, while prioritizing safety in this path of exploration and operations.

Keywords Console design · Touch screens consoles · Mechanical consoles · NASA · Commercial spaceflight

1 Introduction

As we are witnessing in real-time, the space race has transformed from government-driven origination to commercial partnerships that infuse new ideas for both design and function to meet the requirements of human spaceflight. This research was

A. G. Seidel (✉)
Industrial Engineering B.S, The Pennsylvania State University, State College, PA 16801, USA

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_8

conducted to analyze the ergonomic evolution of spacecraft for humans traveling into space, to the International Space Station, our moon, and beyond.

A shift has already begun who is going into space. As humanity looks to the future, we are veering away from the top 1% of fighter pilots that made up the original astronaut corps to the commercialization of spaceflight for the 99% percentile of human beings. Later this year, the first all civilian astronaut crew is planned to launch [1].

Since the United States space program began in the 1950s, the consoles within spacecraft have remained the same. While the layout and size of the console have changed, the primary mechanisms of the display and controls have remained constant. Dating back to the first airplanes, a majority of flight vehicles have featured mechanical systems in the cockpit as knobs and dials. In the space industry, mechanical systems have been the primary and only source of control.

A focus on the console and related cockpit design was an important topic to research as the use of prominent touch screens during the SpaceX Crew Dragon Demo-2 (DM-2) launch on May 30, 2020. It sparked a large volume of commentary from observers around the world as they watched the first launch of Americans from American soil since the last voyage of the shuttle program. With this recent launch, which featured a touch screen console, SpaceX has revolutionized the future of spaceflight and has opened doors for the innovation of future space.

travel. Touch screens are an infusion of modern technology and are one factor that must be considered as we look to redesign and reengineer spacecraft that will be used for future generations.

There are many considerations for the move from heritage technology to newer platforms, and some questions and concerns will be addressed in this research. Is it safe? Is it efficient? Is it reliable? In a field such as space travel, one that deals with high costs, both regarding monetary costs and human life, it is important that innovations are thoroughly investigated and analyzed. The question remains: are touch screens better suited than mechanical systems for the space capsules of the future?

2 Background

2.1 Designing for a Larger Variety of Minds and Body Types

As the United States space program began in the 1950s, the astronaut candidates fit a very specific mold. The original candidates selected were primarily military pilots at the top of their class [2]. This group of men were chosen for their fast-operational skills, multitasking abilities, and psychological strength. Dr. T. Keith Glennan, NASA's first Administrator, said the evaluations "told our medical consultants and scientists of their superb adaptability to their coming flight" [3]. These astronauts possessed skills that had been practiced for hundreds of hours through the prior experience of flying military aircraft.

The current goals of human explorations differ greatly from the goals of the past. Missions in the 1960s were centered on getting humanity to the moon, answering President John F. Kennedy's call to action in his famous Rice University speech. Today, space has evolved from a purely military and scientific endeavor into a place for commercialization and expansion. This shift in purpose for human spaceflight is crucial to understand as the next generation of spacecraft is being engineered.

With the goals of SpaceX, Blue Origin, Virgin Galactic, and other private companies to commercialize space, it is time to change the focus from designing for the 1% and transition to designing for future space explorers. This means that space travel should be designed for people of varying body types, cultures, and education. With console design, designs must be easy to learn and operate, while still maintaining safety standards.

Figures 1 and 2 depict the inside of spaceflight capsules and illustrate the perspective of the console from the pilot's position. The DM-2 console is less cluttered, more visually appealing considering modern standard and is well interfaced compared to the Apollo 11 console. The DM-2 console is more visually appealing through the proper use of "white space", the ability to not be cluttered, and color [6]. While a pilot may respond that the Apollo 11 console is easier to navigate since they have

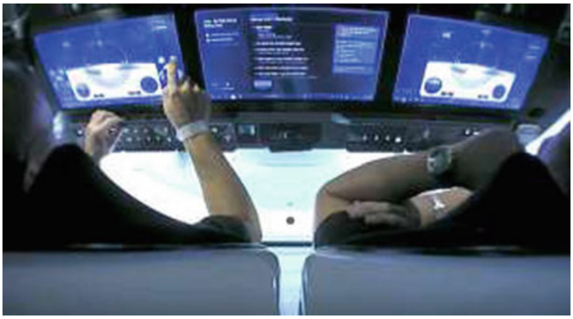


Fig. 1 SpaceX DM-2 console [4]

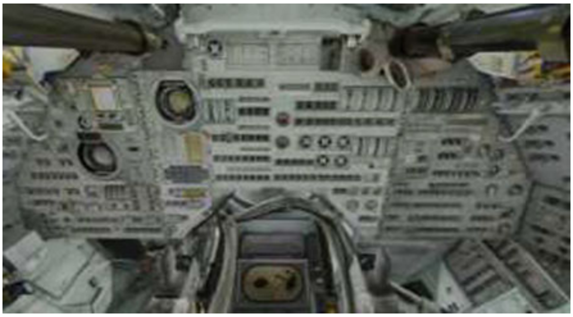


Fig. 2 NASA Apollo 11 capsule [5]

trained on similar designs for hundreds of hours, the 99.8% of the population, who are not pilots, would agree that the Demo-2 console is more user-friendly [7].

As the United States moves to put humans back on the Moon and eventually on Mars, it is important that all passengers of the spacecraft can navigate the controls. In 2020, touch screen displays are part of everyday lives in cultures around the world, from smartphones to way-finding signs in buildings. Touch screens have become more comfortable to use than the original mechanical systems as it utilizes better user interface design using iconography, colors, and white space. This design can also be changed easier through a simple update or planned toggle. By implementing a touch screen console, a wider variety of candidates will be selected to become astronauts.

Through the use of this new console integration versus a mechanical one, more people will be able to operate a spacecraft because of the transferable skills that are learned in our daily lives.

3 Technology Landscape and Research Opportunities

3.1 Added Viewing of the Experience is Marketable

A touch screen console can be condensed to a smaller size than its mechanical counterpart. This can be achieved by having menus where not all the controls are displayed at once and the pilot can navigate to controls that are only necessary at that point in the flight. The compact size of a touch screen console allows for other features to be added to the cabin of a capsule. With the space industry moving towards commercialization, this can allow more space for additional passengers, luxury items (items that are non-mission critical and would not be onboard a purely scientific mission), and more window space. A smaller console creates an opportunity to enhance the experience of the space traveler.

Furthermore, adding window space is not a requirement as space is being commercialized, however, windows will make spaceflight more marketable. The recently unveiled Virgin Galactic spacecraft design, as seen in Fig. 3, showcases a large amount of window space for each passenger. It would be expected if someone was flying for space tourism, they would want to see what is happening outside the spaceship. An additional aspect of the touch screen design is that the screen can become an acting window by adding cameras to the outside of the spacecraft if adding windows is not possible. Users could check camera views from a touch screen console. This advantage is not possible with knobs and dials or a system without a display screen.

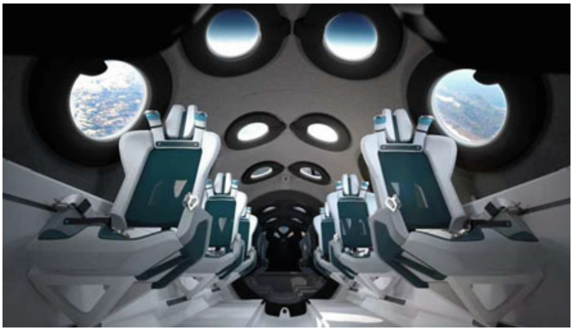


Fig. 3 Virgin galactic spaceship cabin [8]

3.2 Condensation Concerns from Past Experience

During the Apollo 13 mission, the buildup of condensation was present on the console. The condensation created a safety risk as water and electricity can create sparks and fire. While any electronic system is going to fail if an abundance of water is introduced, one positive of the touch screen is that it can be easily wiped off compared to a mechanical console. A simple cloth can clean any liquid off the screen with one simple swipe. A flat screen is much easier to maintain than small, hard to clean areas of a mechanical console. This is analogous to cleaning off a mechanical keyboard with hard-to-reach areas between the keys compared to cleaning a smartphone screen off, which can be done in a matter of seconds. The danger with the mechanical console is that the tolerances between the switches and interface of the console are not zero. For example, the ball in socket type of joysticks will always have manufacturing tolerances that are nonzero. Unlike a mechanical console that possesses small openings for water to seep through, a touch screen console is an enclosed system that can be made airtight.

3.3 Bridging International Differences

Being able to display instructions and tasks in the passengers' first language can be crucial in safety critical situations. On a mechanical console, words are printed onto the buttons and knobs to identify their functions. On a touch screen, however, translation can easily occur in a few microseconds. The touch screen could be used as a reference manual so that directions are not needed from mission control. When we look at designing future space travel, it is important to think on a global scale with systems that can easily toggle to the languages of any passenger.

3.4 Flashing Signals

At the start of research, it was hypothesized that creating a special color scheme could enhance functionality. The belief was that buttons could be changed to specific colors to represent certain inherent meanings. On a global scale, it is universally accepted that red means "stop", "error", or "bad" and green symbolizes "go" or "good". A study in 1999 on the use of color in an Air Traffic Control (ATC) display contradicted this theory [9]. The ATC displays were monochrome, lacking color, at this time. It was believed that by adding color coding to certain functions, ATC could work more efficiently. The results of the study concluded that controllers started to rely too heavily on the color coding and it only decreased their performance levels. However, while data on the addition of color was being collected, the study also experimented by add blinking lights onto the displays. Blinking lights were shown to have a positive impact on the performance of the controllers.

In this proposal for touch screen consoles, I recommend that important functions of the display blink or flash when they are to be pressed or alerting the user. While color coding may not be a major driver to help efficiency, especially in an alert or emergency situation, adding blinking lights will allow users to quickly understand the proper functions of the spacecraft. Blinking technology, which is currently used on mechanical consoles, should be continued to use on future touch screen consoles, especially in commercialized space flight. With users who are less familiar with all of the navigations of flight, blinking indicators can help astronauts run through procedures in the correct order. Blinking lights can highlight when and where to be pressing throughout the entire flight.

Furthermore, many restaurants, many of which are fast-food restaurants, use blinking lights to display urgency. Within the kitchen, monitors display ticket times and information on how long a guest has been waiting for their order. While color coding is used, green for less than 2 min, yellow for less than 10, and red for anything over 10, color does not give a quick representation of the information. When asking coworkers who worked alongside me in a restaurant that used this type of system, they felt that color did not quickly help give information. However, once a ticket time hit 10 min and turned red, that ticket on the display began to blink. This blinking displayed urgency and quickly drew an eye on that item. Through these interviews and that of the ATC study, blinking lights should be a primary method of displaying information on any console of the future.

3.5 Overcoming Precision and Touch Accuracy

One concern of a touch screen console is the lack of control that an astronaut will have over the display panel. Two major concerns come with use of a touch screen console: the first being a lack of precision on the screen during turbulence and the second regards accidentally hitting the screen causing a bad command, especially during turbulence.

A problem with a flat screen is that it is easy for a user's finger to drag across the screen without accuracy. This described swiping motion could be a command of its own. Compared to rigid switches that extrude from the console, the touch screen display introduces a new set of possible errors on the command screen, which creates concern for the level of safety control that the user has. For example, during a period of high turbulence, a mechanical system is better for controlling the spacecraft because of the haptic feedback that the switches provide. A touch screen requires more precision than a mechanical switch due to the rigidness of a mechanical switch.

Because of this reasoning, it would be in the best interest if there were still a few mechanical buttons located around the touch screen portion of the console. These buttons would be used for emergency critical functions only. In the DM-2 design, there were a few mechanical buttons under the touch screen as seen at the bottom of Fig. 4. These buttons provided the astronauts with certain safety maneuvers in the event of an emergency. While touch screen technology may not require these fault-tolerant systems in the future, with our current technology, these buttons are crucial.

Furthermore, there are a few possibilities to combat the lack of feedback that touching a piece of glass gives you. Implementing "force/pressure touch" technology solves this issue. Current smartphones such as the most recent iPhones use this technology. While the display does not move, sensor pressures are able to vibrate, giving the user the feedback with a click. This was originally invented by Apple for its Mac touchpads and has since been introduced into larger scale technology. As seen in Fig. 5, there are force sensors under that surface that create a sense of touch. At present, this technology has size limitations, but with more research, could make its way onto the next iteration of touch screen console designs.

Accidentally touching the display of the console could end in a catastrophic result if touch screen commands are pushed at the wrong time. While this is possible with

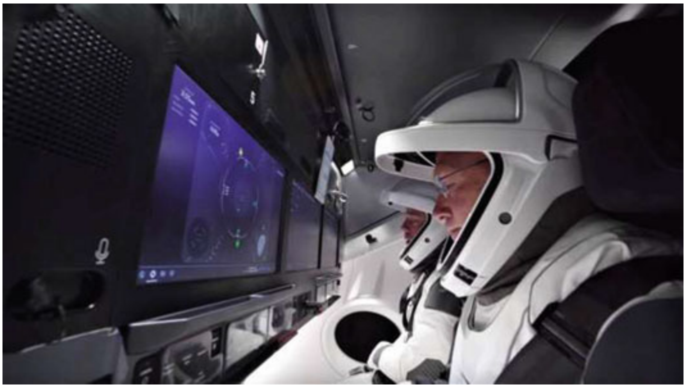


Fig. 4 Mechanical buttons located under touch screen on DM-2 console [10]

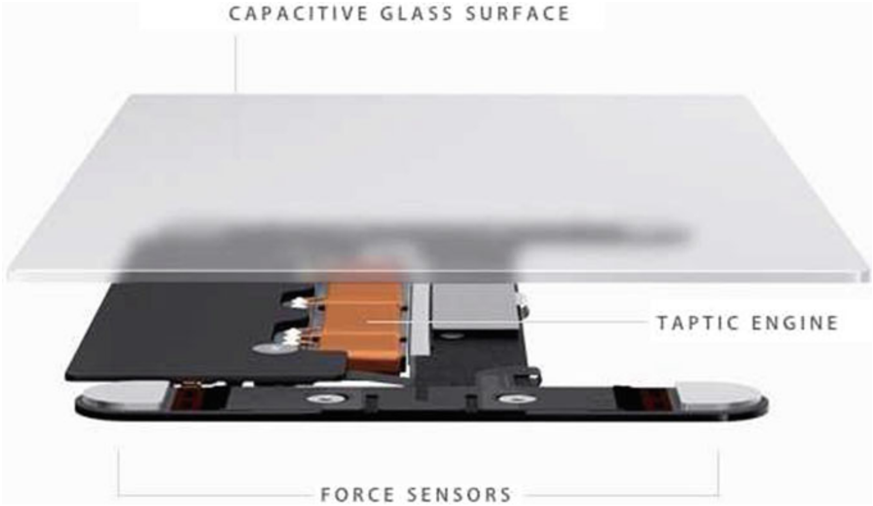


Fig. 5 MacBook force touch trackpad [11]

mechanical switches, certain safety precautions can be implemented such as the implementing button coverings.

Adding preventative measures for accidental touches can be solved with trivial programming. By adding a switch to the side of the touch screen display, the astronaut can control whether the touch screen is able to be interacted with. In the event of turbulence, the astronaut would flip the touch screen to a display only mode. The screen would act merely as a monitor and would not respond to touch input until the switch was returned to touch screen mode. This would give the astronaut full control of the touch screen and would prevent any accidental clicks.

3.6 Touch Screen Capabilities with Multi Touch Technology

While a touch screen may take up less space in the cockpit, this does not mean that functions of the mechanical cockpit are completely lost. In many ways, functions are enhanced. One additional function of a touch screen is through the use of multi touch technology. When a user holds down on a certain button or key and uses slightly more force, an extra menu can pop up. For example, on the iPhone, if an app is held down, an array of functions pop up relating to the app. This function is similar to right clicking on a Windows processing system. This would allow an astronaut to access a menu of options, which would increase overall efficiency and safety.

3.7 Touch Screen Capabilities with Spacesuit Gloves

While the capabilities of a touch screen are wide, challenges with their primary interface, the gloves, remain a major obstacle to this technology's implementation. Original spacesuits could not to use a touch screen console as the gloves did not have the technological capabilities built into them. Recent innovations such as farkle finger technology have made touch screens accessible without having direct skin contact. Farkle finger technology utilizes electrically conductive materials on the glove that allows for screens to be pressed. This technology has been commonly used by motorcycle riders who wear gloves so they can also use their smartphone touch screens. This technology has proven reliable through its users' reviews, but it may not be ready for spaceflight.

SpaceX has implemented touch screen capabilities within their DM-2 spacesuits, and they proved to be successful on their first few missions. The material used within the gloves is conducive to using a touch screen. As seen in Fig. 6, these gloves wrap closely to the astronaut's hand and adding the touch screen capability poses no safety risk to the astronaut. The touch screen capable gloves are no larger than gloves used within non-touch screen missions as seen by comparing Figs. 6 and 7.

These touch screen gloves were most recently used by the SpaceX Inspiration4 crew. This flight was the world's first civilian mission to orbit the Earth. The current design of the gloves could fit four unique body types and hand sizes. This mission featured a similar three touch screen display as seen on the DM-2 mission. During the pre-launch sequence, SpaceX streamed views from within the cabin. During this period the astronauts were touching the screens and interacting with them with no issue or hesitation. From what the audience could see on the stream, motions included the pinpoint touching of buttons and drag features. The gloves appeared to offer the same range of motion that a normal finger would possess. With every mission, we further learn that these gloves are applicable for trained astronauts and civilians alike, a piece of knowledge that is crucial as we head towards an age of increased space tourism.

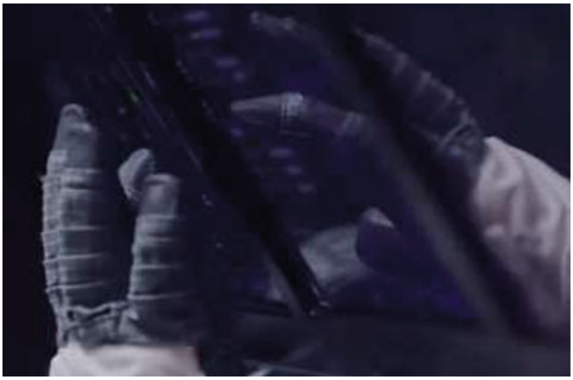


Fig. 6 SpaceX touch screen capable gloves [12]

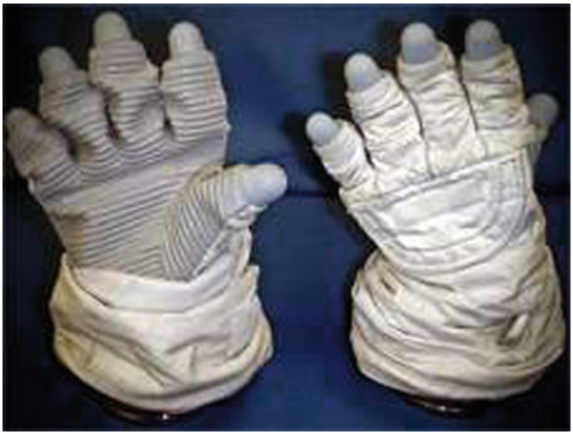


Fig. 7 NASA EVA gloves [13]

4 Recommendations

Based on this research, a hybrid console, a console with primarily touch screen control and a few mechanical buttons, should be the direction for the near-term implementation of human space vehicles. The DM-2 console is the baseline for future consoles and proved successful hybrid implementation with today's technology.

The hybrid console will be easier for a larger variety of passengers to use through the advanced UI of a touch screen. Furthermore, compacting the console will allow for more window space, however, if a larger touch screen is required, cameras from the outside of the spacecraft can input a feed directly to the monitors. This design also allows for cultural and international bridges to be formed as directions and manuals can easily be translated. While currently used on mechanical consoles as well, flashing lights should play an important part in the UI design of touch screen consoles.

To overcome the dangers of accidental touches, a switch should be added to the side of the touch screen console that controls whether the touch screen is able to receive input. This will make the touch screen two step fault tolerant and will eliminate any accidental touches, especially during times of heavy turbulence. Using technology such as force touch and multi-touch will also allow passengers to have more functionality while using the console. These functions, which only can be used on a touch screen compared to a mechanical switch, allow astronauts to quickly navigate through tasks and processes.

Overall, this paper proposes the idea, that instead of reengineering the mechanical consoles of the past, touch screen consoles should be innovated by using the DM-2 as a starting point. As seen by the DM-2 console, a touch screen console creates a higher level of functionality while maintaining the same safety standards as a mechanical console. Not only does it pose benefits to current astronauts, but as we move to the future in an age of space commercialization, space "tourists" will also benefit greatly from a touch screen console. The Inspiration4 mission is an example of new age astronauts utilizing touch screen technology that they are familiar with. Four civilians could navigate their way around the globe using a few buttons, touch screen capable gloves, and a touch screen console.

5 Conclusion

Mechanical systems in the consoles of aircraft were used before the space program even began. Buttons and switches have been used since the start of any type of flight, ranging back to the early 1900s. With years of experience and proven success of using a mechanical console, it is challenging to consider innovations on a heritage design. For nearly 60 years, NASA has established the success of the mechanical console, however, it is time to use innovative and tested designed to support near and deep space exploration. Through this research on implementing touch screen consoles in spacecraft, it is in humanity's best interest to continue down the path that SpaceX proved successful on DM-2.

While the utilization of a touch screen console exclusively is not in the best interest of current astronauts because of safety concerns, creating a baseline for touch screen consoles will provide scientific data that will allow for efficient space travel in the future. Creating a hybrid console, one that focuses primarily on touch screens while still having mechanical switches for emergencies, is crucial in the development of spaceflight. Instead of investing in the past, it is time to invest in our future. With the future of human spaceflight trending towards commercialization, touch screens will allow companies to reach a larger population. Touch screen consoles allow for more ergonomic design choices within the cockpit relating to the user interface and space allocation for controls in addition to other features like windows. Challenges do exist with converting to touch screens such as glove requirements and the possibility of condensation buildup. However, with the correct implementation of systems that support the touch screen console, these issues do not negate the positives of the newer design.

In conclusion, a hybrid design of touch screen and mechanical systems should be the direction we head in the console design industry. Touch screens are more customizable and can account for cultural differences that a mechanical console cannot handle. By using flashing lights on the console, astronauts can easily understand what is going on within the screen in front of them. As we move forward in our exploration of space, it is important that we engineer for the future while also maintaining safety standards and two fault-tolerant systems.

In conclusion, the implementing of touch screens in the capsules of spacecraft is a critical technology. Touch screens will allow astronauts to efficiently and safely navigate through the reaches of space beyond Earth. As humanity expands its reaches across the galaxy, touch screen consoles will play a major factor in the successes and feats that we will accomplish.

Acknowledgements The writer of this paper would like to acknowledge Tara Dulaney of the NASA Goddard Space Flight Center in Maryland, for being a mentor and inspiration in the creation of this paper. Additional thanks to The Bernard M. Gordon Learning Factory at The Pennsylvania State University, including the PSIMES department for connecting the writer and mentor of this paper. Furthermore, the writer would like to thank Dr. Ling Rothrock for continued support through this process. Lastly, the writer would like to thank all the family and friends who showed support during this paper's creation including Jennifer Seidel, Stephen Seidel, Katherine Seidel, and Suhail Al Sharabati.

References

1. Whitemore G (2021) When will we have civilian space travel? Forbes 11 Feb 2021
2. National Research Council (2011) Preparing for the high frontier: the role and training of nasa astronauts in the post-space shuttle era. The National Academis Press, Washington, DC
3. NASA (2020) This month in NASA history: NASA introduces the first astronauts, 10 Apr 2019. https://appel.nasa.gov/2019/04/10/this-month-in-nasa-history-nasa-introduces-the-first-astronauts/. Accessed 10 Aug 2020
4. O'Kane S (2020) Watch NASA astronauts fly SpaceX's crew dragon using touchscreens. The Verge 30 May2020. https://www.theverge.com/2020/5/30/21275753/nasa-spacex-astronauts-fly-crew- dragon-touchscreen-controls. Accessed 27 July 2020
5. Kluger J (2019) What was the Apollo 11 spacecraft like? Climb Inside in 3D. Time 28 June 2019. https://time.com/4419804/apollo-11-virtual-reality-3d/. Accessed 15 July 2020
6. Coates S (2014) White space: an overlooked element of design. Western Ketucky University
7. Federal Aviation Administration (2020) Civil airmen statistics, 22 Apr 2020. https://www.faa.gov/data_research/aviation_data_statistics/civil_airmen_statistics/
8. Cao S (2020) Richard branson is 'Ready to Blast Off'. In: Newly unveiled virgin galactic spaceship, TheObserver, 28 July 2020. https://observer.com/2020/07/virgin-galactic-spaceship-cabin-design-reveal/. Accessed 31 July 2020
9. Office of Aviation Research, Federal Aviation Administration (1999) Guidelines for the use of color in ATC displays. Department of Transportation, Washington, DC
10. Groh J (2019) SpaceX's crewed dragon launch debut likely to slip into 2020 as NASA pursues "realistic" dates, Teslarati, 3 August 2019. https://www.teslarati.com/spacex-nasa-crewed-dragon-launch-debut-delays/. Accessed 12 July 2020
11. Profis S, Elliott M (2020) How the MacBook's force touch trackpad works. CNET, 27 Oct 2016. https://www.cnet.com/how-to/what-is-force-touch-macbook/. Accessed 14 July 2020
12. Charboneau T (2020) All about circuits, 1 June 2020. https://www.allaboutcircuits.com/news/inside-look-at-new-spacex-spacesuits/. Accessed 15 Apr 2021
13. NASA (2008) Learn about spaceuits, 13 Nov 2008. https://www.nasa.gov/audience/foreducators/spacesuits/home/clickable_suit_nf.html. Accessed 15 July 2020

Space Operations Fuelling Space Awareness and Science Education in South Africa—Supporting STEM Education in the Knowledge Economy

Daniel Abednigo Matsapola

Abstract The Ten-Year Innovation Plan (2008–2018) of the Department of Science and Innovation (DSI) identified space science and technology as one of the five grand challenges for South Africa to become a key contributor to global space science and technology, with a National Space Agency, a growing satellite industry, and a range of innovations in space sciences, earth observation, communications, navigation, and engineering. The South African National Space Agency (SANSA) was established in 2008 and opened its doors to serve humanity on April 1, 2011. SANSA absorbed the Council for Scientific and Industrial Research (CSIR) Satellite Applications Centre (SAC) and the National Research Foundation (NRF) Hermanus Magnetic Observatory (HMO). Two SANSA divisions, Space Operations, and Earth Observation (EO), were created from CSIR SAC at Hartebeeshoek (HBK) and Pretoria, respectively. The SANSA 2015–2020 Strategy identified strategic goal 3 to develop national human capacity and ensure transformation of the South African space sector. SANSA's EO established the Science Advancement Services (SAS) unit in January 2013 in order to develop and implement the National Space Awareness Programme (NSAP) to stimulate youth interest in science. The DSI established the Science Engagement Strategy (SES) in January 2015 to develop the national science engagement programme. The SANSA NSAP is aligned to the DSI SES objectives. This paper reflects on the national achievements of the SANSA EO SAS NSAP spanning the period of 2013–2014 to 2019–2020. The paper reflects on the impact of the DSI SES on the SANSA EO SAS NSAP activities, the achievement of national footprint during National Science Week 2019, and the importance of developing youth entrepreneurial mindset in science, technology, engineering and mathematics (STEM) education. Last, this paper showcases the alternative approach for sustainable space awareness in the 52 districts of South Africa and the Human Capital Development Pipeline Model that forms the connecting bridge between learners in the school system and the professionals in the space industry.

D. A. Matsapola (✉)
Science Engagement Manager, Earth Observation Division, Enterprise Building, Gauteng Province, South African National Space Agency, Mark Shuttleworth Street, The Innovation Hub, Pretoria 0087, South Africa
e-mail: dmatsapola@sansa.org.za

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_9

Keywords Space · Awareness · Science · Engagement · Youth · Empowerment

1 Introduction

South Africa transitioned to democratic government on April 27, 1994 with a rich history of participation in space that dates to the beginning of the space race between Soviet Union and the United States of America (USA) in the 1950s. The CSIR Naval Telecommunications Research Laboratory agreed to operate and maintain a Minitrack station in Esselen Park, Johannesburg, on behalf of the American Naval Research Laboratory in 1958 [1]. The Johannesburg Satellite Tracking and Data Acquisition Network station (Joburg STADAN) operations were moved from Esselen Park to Hartebeeshoek in 1960, becoming one of the busiest network stations in the Goddard Space Flight Centre satellite Telemetry, Tracking and Command (TT&C) network [1].

South Africa was under apartheid rule from 1948 to 1993, during which space activities were carried out by the privileged white minority at locations strategically placed out of sight and inaccessible to the majority Africans. The National Aeronautics and Space Administration (NASA), the USA space agency, was established in 1958 and operated the Hartebeeshoek (known as HBK to the international space community) facility during 1960–1975 to support its early space missions. The HBK operations included the reception of satellite images from Landsat 1, the first satellite of the USA's Landsat Program launched on July 23, 1972.

The CSIR took over the operations at HBK in 1975 after NASA left South Africa [1] and evolved the facility into the CSIR Satellite Applications Centre (SAC) that was manned by majority white staff by 1993. HBK received the first Meteosat image in 1977 into its growing archive of satellite imagery. South Africa initiated its first space programme in the 1980s, with the objective to develop an Earth Observation satellite, a launcher and all the necessary facilities to support these activities [1]. The CSIR SAC celebrated 50 years (1960–2010) in operation at HBK in 2010 and published the SAC book to mark this great milestone achievement.

The satellite assembly, integration and testing (AIT) facilities were constructed at Houwteq in Grabouw in the Western Cape Province, 70 km southeast of Cape Town while the launch facility was established at Arniston, on the Indian Ocean coastline, 110 km southeast of the Houwteq AIT facility [1]. The launch facility, at the Denel Overberg Test Range, successfully launched a rocket into space in the late 1980s, and considerable investments were made in the South African industry to support the first space programme which was discontinued in 1994 before any satellite was launched [1].

The transition from the apartheid era to the new democratic South Africa, since 1994, led to the creation by the democratic government of the Department of Arts, Culture, Science and Technology (DACST) that developed the 1996 White Paper on Science & Technology which stated that Science and Technology (S&T) are essential components of the government's strategy for creating the South Africa of the future.

It envisioned a South Africa that uses S&T to become economically competitive on a global scale and provide essential services, infrastructure, and effective health care for all South Africans [2].

The Department of Communications funded in 1998 the establishment of the Institute for Satellite and Software Applications (ISSA) at the Houwteq satellite AIT facilities in Grabouw, Western Cape, to offer post-graduate qualifications in information technology, satellite and software engineering in partnership with local and foreign universities. This catalytic human capital development program ran for 8 years and by 2005 it had produced 500 post-graduates [1], of which 155 came through Stellenbosch University [https://m.youtube.com/watch?v=8dtU8OrzoMc], who built South Africa's first satellite, Sunsat, that was launched by NASA at the Vandenberg Air Force Base into space on February 23, 1999.

On August 1, 2002, DACST was divided into the Department of Arts and Culture, and the Department of Science and Technology (www.dst.gov.za) [3]. The 2002 National Research and Development Strategy (NRDS) projected extensive investment by the Department of Science and Technology (DST) in science promotion, focused specifically on the need to "make science attractive, accessible and relevant through media, public engagement and promotional programmes", to attract learners into science and technology through large public science programmes and to massify the number of public understanding and engagement activities in order to increase the number of matriculants achieving university entrance in Mathematics and Science.

The NRDS assumes that the systematic and focused provision of adequate information about science and technology would serve to interest more people (especially learners) in science and improve public appreciation of science and technology. These functions have been accommodated in the South African Agency for Science and Technology Advancement (SAASTA) of the National Research Foundation (NRF). In May 2018, the 6th Administration merged the Department of Higher Education and Training with the Department of Science and Technology, leading to the renaming of DST to the Department of Science and Innovation (DSI). In 2019, the NRF Act was amended to provide NRF SAASTA the mandate to operate as the national coordinator of the Science Engagement Strategy in South Africa.

SANSA was created to promote the use of space and strengthen cooperation in space-related activities while fostering research in space science, advancing scientific engineering through developing human capital, and supporting industrial development in space technologies [see http://www.sansa.org.za]. On April 1, 2011, SANSA opened its doors to serve humanity through its five divisions namely, SANSA Corporate Office (Pretoria), SANSA Space Science (Hermanus), SANSA Space Engineering (Pretoria), SANSA Space Operations (Hartebeeshoek) and SANSA EO (Pretoria). SANSA EO moved from HBK in 2013 to The Innovation Hub in Pretoria. Science Advancement and Human Capital Development were cross-cutting activities embedded in all the SANSA divisions.

2 National Space Awareness Programme at SANSA EO

The Science Advancement Services (SAS) unit was established in January 2013 at SANSA EO division with core functions being to develop and implement the national space awareness programme in South Africa reaching all nine provinces annually, advocacy for awareness and increased use of EO by the youth and public, support science, technology, engineering, mathematics and innovation (STEMI) education in the national curriculum, and to profile and promote careers in space science and technology nationally. The SANSA EO SAS used existing platforms to deliver its core functions and had a target of directly engaging with 2000 school learners during its first financial year 2013 2014 in operation.

During the 2013–2014 financial year, the activities of the SAS unit were measured by the number of learners reached through direct and specific engagement, using the available platforms and the science advancement manager reported quarterly against the set targets that are published in the SANSA Annual Performance Plan 2013–2014. The SAS unit worked with the SANSA EO researchers, technologists, and scientists from other units to engage with school learners. The key performance indicator changed during the 2014–2015 financial year to the number of youths directly engaged by SANSA. In January 2015, Cabinet approved the DSI Science Engagement Strategy that identified 11 target publics to guide all science engagement services towards building a society that is scientifically literate.

A. SANSA Science Engagement Platforms

SANSA EO has been using well-established science engagement platforms since January 2013 to create an awareness of EO benefits and increase its innovative use by the target publics.

1. Science Festivals in 8 Out of 9 Provinces

Science festivals draw mass participation, are public events comprising science, engineering and technology (SET) activities that celebrate science in a festive, fun-filled, and exciting way. The oldest science festival in South Africa, Sasol Scifest, was launched in 1997 in Grahamstown as the first of its kind [4] and there are new ones being organized to broaden the reach of disadvantaged communities annually. Science festivals are generally week-long activities organized at a particular location where exhibitors gather to engage with target audiences. SANSA exhibits and conducts space awareness workshops at the DSI-supported science festivals, distributed in eight provinces of South Africa, except for the Western Cape Province, where none exists currently. The science festivals are aligned with the national school calendar from January to December, whereas SANSA implements its national space awareness programme through the April-March government financial year cycle.

2. National Science Week (2013–2019)

National Science Week (NSW) is an annual celebration of science, engineering, and technology, attracting thousands of learners and members of the public to workshops, science shows and lectures, which are held at universities, schools, science centres and public facilities countrywide, led by the Department of Science and Innovation. It is run in all nine provinces simultaneously at multiple sites per province. SANSA SAS unit has consistently participated in NSW since 2013 and progressively built the capacity to implement space awareness workshops in all nine provinces simultaneously in the first week of August 2019, with the project team of 18 staff from SANSA EO (15), SANSA Space Science (1) and the Department of Science and Innovation (2).

3. **Direct School Visits by SANSA**

The SAS unit has established the Network of Science Educators in multiple provinces to support STEM education and routinely visits school premises to conduct space awareness workshops and profile careers in space science and technology using space scientists, researchers, and engineers as role models. SANSA also supports the DSI in carrying out its national agreement with the Department of Basic Education (DoBE) that is implemented at the 9 Provincial Education Departments. The top priority of the DoBE is to improve the learners' performance in mathematics and physical science subjects at the primary and secondary levels.

In 2006, the Department of Basic Education introduced Geographic Information System (GIS) as part of the grade 10 geography syllabus for the first time. It has, meanwhile, been extended to grades 11 and 12 (Scheepers, 2009). SANSA EO division developed the school edition of the Fundisa Disk, the human capital development resource that was originally (2008) targeted at tertiary institutions in South Africa as part of the "Data Democracy for Developing Countries" legacy project of South Africa to enrich the learning of remote sensing and GIS. The Fundisa Disk School Edition is targeting geography educators teaching grades 10, 11, and 12.

The SAS unit works with geography educators to promote the teaching of GIS and remote sensing in schools as the foundation for developing future geo-informatics professionals. The Southern African Geography Teachers Association (SAGTA) is the representative body of over 2000 geography educators that the SAS unit works with to promote the increased awareness and use of EO by the school-based youth. SAGTA members routinely invite SANSA to engage directly with their geography learners in the classroom and SANSA has developed practical learning resources with hands-on activities that the learners enjoy.

The SAS unit has partnered with the South African Radio Astronomy Observatory (SARAO) to train natural science educators in the Tshwane South District of the Gauteng Education Department (GED) since October 2018, with the support of NRF SAASTA. The GED educators' workshop was held at SANSA Space Operations. SANSA and SARAO have also delivered the "Planet Earth and Beyond" educator training in the Sekhukhune Education District in March 2019, leveraging other SANSA national partnerships. The Tshwane South District is the top performing education district in South Africa and Gauteng Province in terms of Matric results during the 2019 and 2020 calendar years.

4. **World Space Week During 04–10 October Annually**

SANSA works with the South African space industry annually during World Space Week and beyond to profile the role played by various industry players in the space value chain and routinely hosts public groups at its satellite ground station at the SANSA Space Operations facility in Hartebeeshoek, Gauteng Province. The outreach model has varied during 2013 to date from the NRF SAASTA-coordinated weeklong activities targeting one province and culminating in the open day on the Saturday where all exhibitors gather in one venue to exhibit both the upstream and downstream space applications, to the entity-driven dispersed model where different space organizations carry out their own space awareness activities at selected sites independently. The World Space Week Association identifies a new theme for each year to guide the global celebration of space in enriching life on Earth.

5. **Guided Public Tours of SANSA Facilities**

South Africa has wonderful space facilities from its over 60 years of space heritage, most of which are in the Western Cape and Gauteng Provinces. The SAS unit routinely conducts guided tours of the prestigious SANSA Space Operations facility in Hartebeeshoek that attracts local and international visitors. The standard package takes two hours every Thursday since 2019 and is very popular with school groups led by geography and natural science educators, as well as annual industry tours by university students. Guided tours were conducted daily upon arrangement on Mondays to Fridays during 09h00 to 16h00 prior to 2019. Parents bring family groups and conference organizers also use the facility for conference ice breaker sessions as well as conference closing day excursions.

6. **Role Modelling and Career Exhibitions**

SANSA EO scientists and researchers avail themselves to engage with grades 8–11 learners who are studying natural science, mathematics, physical science, and life sciences at disadvantaged schools through the NRF SAASTA national Role Modelling campaigns. There are many other career exhibition platforms with a national footprint that SANSA uses to promote careers in space science with the aim to increase the uptake of mathematics and physical science at secondary school level.

The SAS unit routinely hosts job shadowing candidates from local schools at SANSA facilities as part of the National Space Awareness Programme and facilitates their exposure to specialist space fields within the organization. Strong partnerships are forged with the Network of Non-profit Organizations (NPOs) that promote STEMI enrichment programs. The database of visitors to SANSA facilities benefits from unique platforms such as the Space Tour 2019 that brought the former NASA astronaut, Dr. Don Thomas to the University of Pretoria in September 2019 and the public lecture given by former NASA Administrator, Major General Charles F. Bolden, at the CSIR International Convention Centre in November 2018 targeting all the 11 target publics.

3 Recent Initiatives

1. SANSA Science Clubs Model

Science Clubs are a unique platform for science engagement with the 11 target publics identified in the DSI SES, especially learners and educators in schools nationwide. NRF SAASTA has the mandate to coordinate Science Clubs in South Africa and all the SANSA EO Science Clubs are registered with NRF SAASTA. The DSI has signed an agreement with the National Department of Basic Education (NDoBE) to support STEM education in schools. This DSI-NDoBE agreement is being implemented at the provincial level with the Provincial Departments of Education (PDBE) that are the custodians of schools education. The main objective of the Science Clubs project is to improve learners' attitude towards mathematics and science subjects.

The DSI and the Limpopo Department of Education (LDOE) run the pilot project for School-Level Science Engagement Initiatives since 2020, with the purpose to create a culture of science in schools. The DSI-LDOE School Level Science Engagement Initiatives Operational Plan was developed for the 2020 school calendar year, with four broad objectives. Objective 1 is to improve learners' attitude to mathematics and science subjects and the initiatives under this objective are Science Clubs

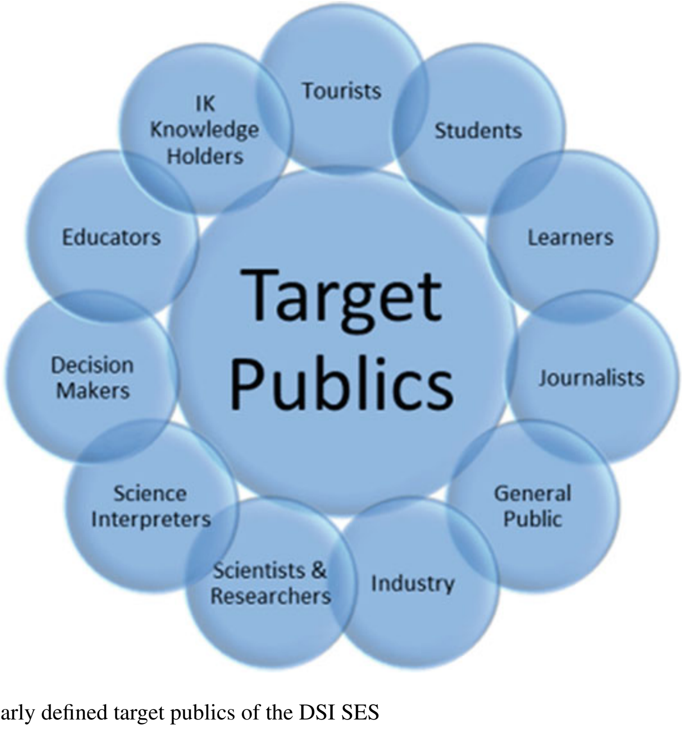


Fig. 1 Clearly defined target publics of the DSI SES

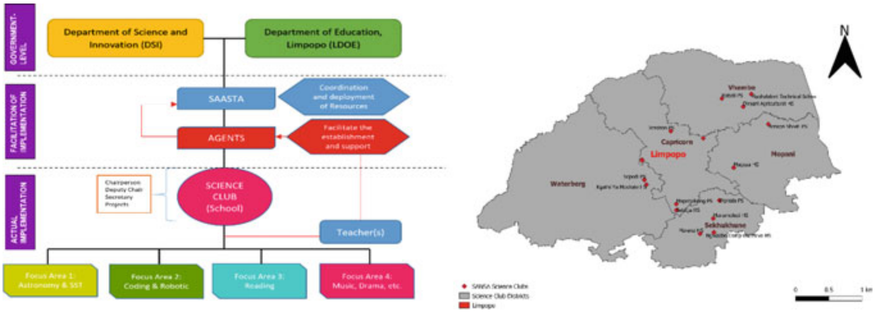

GOVERNMENT-LEVEL
FACILITATION OF IMPLEMENTATION
ACTUAL IMPLEMENTATION
Department of Science and Innovation (DSI)
Department of Education, Limpopo (LDOE)
SAASTA
Coordination and deployment of Resources
AGENTS
Facilitate the establishment and support
Chairperson
Deputy Chair
Secretary
Projects
SCIENCE CLUB (School)
Teacher(s)
Focus Area 1: Astronomy & SST
Focus Area 2: Coding & Robotic
Focus Area 3: Reading
Focus Area 4: Music, Drama, etc.
N
Vhembe
Capricorn
Limpopo
Mopani
Waterberg
Sekhukhune
SANSA Science Clubs
Science Club Districts
Limpopo
0
0.5
1 km

Fig. 2 DSI-LDOE SANSA Science Clubs to be established

and Mass Participation programs. SANSA was one of more than 20 science organizations that were invited to facilitate the establishment and support of Science Clubs in selected Limpopo Province primary and secondary schools (Fig. 1).

The LDOE identified the 107 schools targeted for Science Clubs establishment in all the 10 education districts of Limpopo Province and SANSA was allocated 16 primary and secondary schools (Fig. 2 map) to establish Science Clubs at by March 31, 2020, a target date that was disrupted by the coronavirus global pandemic.

SANSA EO division received grant funding in August 2020 to promote the priority research area of Space Science and Technology (SST) in the Eastern Cape, Mpumalanga, Gauteng, and Limpopo Provinces by March 31, 2021. SANSA EO used its working partnership with the Capricorn District Municipality (CDM) to promote the use of EO in disaster management at 16 schools that are adopted by CDM in their Disaster Risk Reduction School Competition. SANSA EO first attended and exhibited at the District Disaster Management School Competition held in Polokwane on February 12, 2020, celebrated under the theme “Reduce Disaster Damage to Critical Infrastructure and Disruption of Basic Services” during which five high schools from different local municipalities within the Capricorn District participated in the essay competition. SANSA and CDM have established 16 Science Clubs in the Capricorn District during March 15–19, 2021.

2. The Impact of the DSI Science Engagement Strategy on the SANSA NSAP

The DSI Science Engagement Strategy was published in January 2015 and has provided a systematic, rather than project-based, framework that leverages the National System of Innovation for national science engagement, targeting clearly defined segments of society. It promotes greater co-operation in all the three spheres of government (national, provincial, and local), and involves multiple role players who contribute their knowledge and expertise towards addressing national priorities. It has provided SANSA with the opportunity to pilot the Science Clubs Model in the Limpopo Province and the context for scaling this model in other provinces and districts nationwide.

Clear roles have been assigned to the various role players, with the DSI driving the national science engagement agenda, NRF SAASTA playing the national coordination role of all science engagements and SANSA carrying out its mandate through the facilitator role for the promotion of the priority research area of space science and technology. The democratic government's 6th Administration advocates for the District Development Model as the new way of work in government aimed at distributing the economic development across South Africa's 52 districts rather than concentrating it in the major economic centers like Johannesburg, Cape Town, and Durban. This provides SANSA with the unique opportunity to leverage national and global partnerships to increase space awareness.

South Africa is progressively becoming a key contributor to global space science and technology and is co-chair of the Group on Earth Observation (GEO), a global partnership of governments and organizations that envision "a future wherein decisions and actions for the benefit of mankind are informed by coordinated, comprehensive and sustained Earth Observations". South African organizations have been participating in the global space forums post-1994, with the CSIR as an Associate Member of the Committee on Earth Observation Satellites (CEOS) since 1998 and SANSA as a full CEOS member since 2010. These international efforts and initiatives advocate for increased awareness and use of EO in sustainable development, with initiatives targeting developing countries.

The DSI Science Engagement Strategy and its Implementation Plan 2017, along with the District Development Model of South Africa's 6th Administration, provide the framework for linking local space awareness initiatives with continental and global projects such as the Sendai Framework for Disaster Risk Reduction, the Paris Agreement for Climate Change, and the United Nations 2030 Agenda for Sustainable Development. SANSA EO leverages these global partnerships to benefit African youth and to redress the imbalances of the past through the comprehensive National Space Awareness Programme primarily aimed at contributing to the critical mass of skills for transforming the national space sector through long-term space awareness and educational interventions.

3. The SANSA 2020–2025 Strategy opportunities

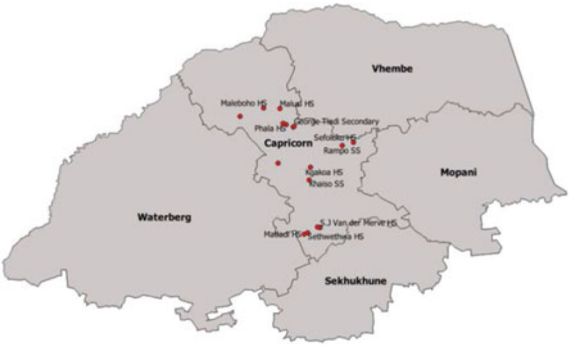
SANSA has a vision to develop "an integrated national space capability that responds to socioeconomic challenges in Africa by 2030" and its mission is "to provide leadership in unlocking the potential of space for the advancement and benefit of humanity". In 2006, the department of education introduced GIS as part of the grade 10 geography syllabus for the first time. It has, meanwhile, been extended to grades 11 and 12 (Scheepers, 2009). SANSA will use the Science Clubs Model to empower geography educators in rural areas to embrace digital platforms in delivering practical GIS and remote sensing classroom activities. SANSA already works with the Southern African Geography Teachers Association (SAGTA), a non-profit organization of geography educators who expressed huge interest in space science to feature more prominently in the geography curriculum.

SANSA is also playing an important role as Implementing Partner in the Digital Earth Africa (DEA) project that leverages cloud platforms to develop continental scale EO products and services that can also be used at a local level. SANSA will work with local partners in capacity building like universities and research councils to use the DEA products and services to deliver increased awareness by African youth of their environment through such Science Clubs activities as “Map My Village”, “My Digital Village” and “My Digital District”. The Science Clubs’ activities will be aligned with the three EO global priorities of disaster management, climate change and sustainable development goals advocated by GEO and CEOS.

SANSA will expand its support to promote the priority research area of space science and technology in the National Network of Science Centers coordinated at the Southern African Association of Science and Technology Centers (SAASTEC), especially the Science Centers based at universities, to leverage the existing space capacity in the various university faculties. To meaningfully transform the South African space sector, space awareness efforts must begin at the primary school level and take a long-term, integrated, multi-stakeholder investment approach. Universities continue to play the key role in the space knowledge transfer and research, with 5 of the 6 South African satellites developed by Stellenbosch University and the Cape Peninsula University of Technology (CPUT). Pilot studies promoting the downstream space applications will be carried out by SANSA in two of South Africa’s poorest provinces of Limpopo and the Eastern Cape during the 2021–2025 strategic planning horizon.

The National Space Awareness Programme has the goal to establish and support at least 180 registered Science Clubs with the National Network of Partners in government and the private sector by the March 31, 2025. SANSA will leverage its national and global partnerships developed during its first 10 years in operation (2011–2021) to develop sustainable space awareness models that can be scaled into the African continent, with an initial focus on the Southern African Development Community (SADC) that South Africa is a member of. The activities of the Space Clubs allow SANSA to link to continental and global initiatives driven at GEO and CEOS with local activities to position Africa to realize the Youth Demographic Dividend alluded to in Agenda 2063–The Africa We Want (Fig. 3).

Fig. 3 SANSA CDM science clubs established on March 2021



4 Results and Discussion

South Africa has celebrated 22 years as an African spacefaring nation on February 23, 2021, following the successful launch of the 64 kg Sunsat developed by post-graduate students at Stellenbosch University. SANSA celebrated 10 years in operation on the April 1, 2021, the period during which four South African satellites, ZACUBE-01, ZA-Aerosat, N-Sight and ZACUBE-02 were launched into the low earth orbit. Satellite naming competitions for the ZACUBE-01 and ZACUBE-02 government-sponsored missions were carried out nationally in schools, with the high school learner from Matatiele in the Alfred Nzo District of the Eastern Cape Province winning by naming ZACUBE-01 Tshepisosat. Tshepiso means hope in Sesotho language. The naming of ZACUBE-02 could not be concluded due to conflict with existing names. SANSA worked with NRF SAASTA to promote the satellite naming competition in schools based largely in the rural areas.

The SANSA 2020–2025 Strategy provided the framework for progressively transforming the South African space sector to represent the national demography. The 2019–2020 results for the National Space Awareness Programme show that over 25,000 youth were directly engaged in all the 9 provinces of South Africa, mapped using GIS at the district level, with 36 of the 52 districts registering space awareness activities (see Fig. 4). The youth engagement target set by the SANSA 2015 – 2020

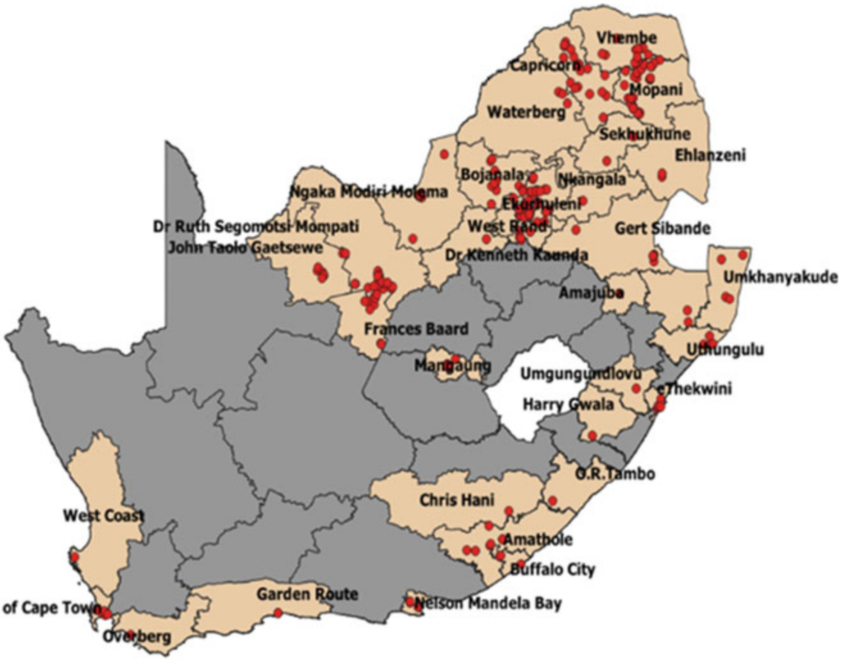


Fig. 4 SANSA earth observation youth engagement reach in 2019 2020

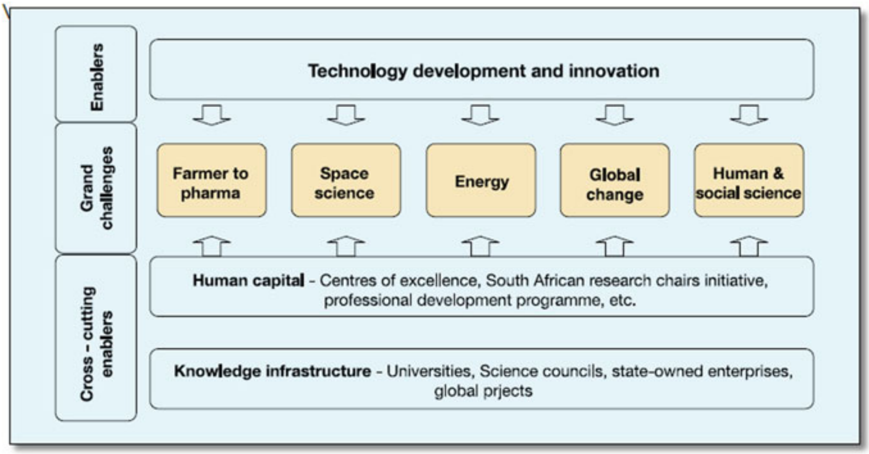

Enablers
Technology development and innovation
Grand challenges
Farmer to pharma
Space science
Energy
Global change
Human & social science
Cross - cutting enablers
Human capital - Centres of excellence, South African research chairs initiative, professional development programme, etc.
Knowledge infrastructure - Universities, Science councils, state-owned enterprises, global prjects

Fig. 5 Grand challenges and enablers of the ten-year innovation plan

Strategy was exceeded and more South African youths in rural parts of the country are engaging with space science and technology through SANSA and its growing network of partners.

The approval of the DSI Science Engagement Strategy by Cabinet in January 2015 provided the systematic and enabling environment for SANSA EO SAS unit to leverage resources from the National System of Innovation to break into national scale science engagement. The announcement of the District Development Model by President Ramaphosa in 2019 provided SANSA and the space industry with the greatest opportunity since 1994 to showcase the value of space science and technology in the national development agenda, measured at the district level. The SANSA 2020 – 2025 Strategy guides the development of successful space awareness models in South Africa that can be scaled up to the African continent level, starting in the SADC region (Fig. 5).

5 Conclusion and Recommendations

The National Space Awareness Programme at SANSA EO division provides the bridge between learners in the school system through its national portfolio of School-Level science engagement and the diversity of space professionals. South Africa takes a long-term view on its journey of migration from the resource-based to the knowledge-driven economic model. SANSA EO implements the Integrated Human Capital Development Pipeline Model, coordinated nationally by the SAS unit.

The Ten-Year Innovation Plan (TYIP) identified space science and technology as one of five grand challenges designed to stimulate multi-disciplinary thinking and to challenge South Africa's researchers to answer existing questions, create new disciplines and develop new technologies. The TYIP proceeds from government's broad socioeconomic mandate, particularly the need to accelerate and sustain economic growth. To close the gap between South Africa and those countries identified as knowledge-driven economies, the National System of Innovation must urgently confront South Africa's failure to commercialize the results of scientific research and our inadequate production of knowledge workers capable of building a globally competitive economy.

Selling education to Africa's 420 million youth aged between 10 and 35 in the era of 30.5% unemployment rate in South Africa (March 2021) requires growing support for STEM education that leverages local and global partnerships. The high unemployment rate is an indicator of an oversupply of job seekers rather than job creators who create or extract value from nature. Through the SANSA EO Science Clubs national project, aimed at improving learners' attitude to mathematics and science subjects, and linking with the human & social science grand challenge, SANSA will promote awareness of economics as a social science and showcase space applications' linkages with the development of economic insights. The entrepreneurial mindset will be inculcated through the SANSA EO Science Clubs and careers in space science and technology will progressively feature space entrepreneurs from small, medium and micro enterprises (SMME) and large established South African space companies.

The capacity of the South African space industry to absorb space graduates from universities is very low, and this has been the case since the advent of democracy in 1994. This requires a rethink on how entrepreneurship and business education in South Africa is delivered. The SANSA Space Clubs present an opportunity to stimulate the entrepreneurial mindset in South African youth, considering the high unemployment rate, the job losses registered during the coronavirus global pandemic, and the need for entrepreneurs and investors by governments all over the world to achieve sustainable economic recovery and growth. SANSA EO will support the creation of the youth wing of the 2019-established South African space industry body ZASpaceInc (https://zaspaceinc.org) to promote the space industry's inclusion of youth challenges in its industry development strategies.

References

1. Magagula SG, Witten J (2018) Space education and awareness in South Africa-programmes, initiatives, achievements, challenges, and issues. In: 2018 SpaceOps conference, Marseille, France, 28 May–01 June 2018 (Reference to a conference/congress paper)
2. https://www.gov.za/sites/default/files/gcis_document/201409/sciencetechnologywhitepaper.pdf. Accessed 13 Apr 2021
3. esat.sun.ac.za. Accessed 10 Apr 2021
4. Wilmot B (2021) Sasol SciFest: promoting a "culture" of science in South Africa in a festive way: museums, science and education. Journals.co.za. Accessed 13 Apr 2021

Development and Simulation of a South African Satellite Camera on a Satellite Testbench for Capacity Building in Space Operations, Training and Research

Brendon Maongera, Kai Leidig, René Laufer, Peter Martinez, Andy B. Armitage, Per Danielsson, and Jens Eickhoff

Abstract More and more countries are operating their own national satellites. In recent years, states like Peru, Angola, Bangladesh and Qatar have launched their very first satellite missions. Many of these states aim for multiple socio-economic benefits: the mission tasks typically cover disaster management, agricultural monitoring, water management, fisheries and national security. With their own ground stations and a mission control system, the corresponding national institution can fully control such satellites after launch. A problem that emerging space nations often face is that of knowledge transfer to their respective local industry and academia, especially in the area of onboard computers, onboard software and operations. In 2016, the University of Stuttgart and the University of Cape Town (UCT) decided to strengthen and deepen the cooperation in research and education in the field of space sciences and technologies between both institutions. This partnership, supported by additional industrial sponsors Cobham Gaisler, Sweden and Terma B.V Netherlands, and Airbus, provided UCT with a fully representative satellite simulation, a digital twin of the "Flying Laptop" mission. This comprises the simulator, the flight software and a real mission control system—namely the Satellite Control and Operation System CCS5. The CCS5 runs together with the spacecraft simulation software on a single high-performance workstation. The third application is the flight software development environment. Multiple monitors allow users to simultaneously display

B. Maongera (✉) · R. Laufer · P. Martinez
Department of Electrical Engineering, University of Cape Town, Menzies Building, Upper Campus, Library Road, Rondebosch Cape Town 7701, South Africa
e-mail: MNGBRE003@myuct.ac.za

K. Leidig · J. Eickhoff (✉)
Institute of Space Systems, University of Stuttgart, Pfaffenwaldring 29, 70569 Stuttgart, Germany
e-mail: eickhoff@irs.uni-stuttgart.de

J. Eickhoff
Airbus Defence and Space GmbH, 88090 Friedrichshafen, Germany

A. B. Armitage
Terma A.S, Schuttersveld 9, 2316 XG Leiden, The Netherlands

P. Danielsson
Cobham Gaisler AB, Kungsgatan 12, 411 19 Göteborg, Sweden

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_10

the different parts of the simulation as it is running: The Airbus simulator SimTG (Simulator Third Generation) itself, the hardware emulator for the onboard software called TSIM, and a visualization of the satellite in orbit. This equipment enables now satellite software development and operator training at UCT. For UCT, this is a significant step in academic excellence, research and development (R&D) and innovation and capacity building in the field of space sciences and technology. Students who gained experience at Stuttgart University and further insights from Airbus Defense and Space in Fridrichafen will perform the training in Cape Town. This ensures that the personnel in Cape Town will be fully trained in the system's operation. In addition, phone support from experienced Airbus engineers is available. At the time of writing this abstract, the system is being established. The team at UCT will simulate a flight scenario which covers vegetation monitoring using optical payload instruments and a detailed explanation on how to model a satellite and its subsystems and the effects of the simulator on local capacity growth. This will be an impactful demonstration of South African academic competence in mission operations and spacecraft modelling.

Keywords Training and knowledge transfer · Simulation and operations · Educational application of space operation

Acronyms/Abbreviations

ACS	Attitude control system
CCS5	Satellite Control and Operation System CCS5
CCSDS	Consultative Committee for Space Data Systems
ESA	European Space Agency
PC	Personal Computer
PUS	Packet Utilization Standard
SANSA	South African National Space Agency
S/C	Spacecraft
SimTG	Simulator 3rd Generation (Airbus satellite simulation toolkit

1 Introduction

In recent years, many emerging space agencies of various countries around the world are formed. States like Egypt, Angola, Nigeria, Qatar, Brazil have launched or about to launch their first satellites. Many of these nations invest in space technology to help improve their socio-economic benefits. They use satellites to help with disaster management, national security, vegetation monitoring (agricultural land, impact of urban spatial development and grasslands).

The problem these states are confronting is that the knowledge of space technology is not being transferred to their local industry and academia. Most of these states use existing mature space nations to build and control the operations of their satellites with limited knowledge transferred back to local engineers or scientists.

If emerging space nations have their own antenna and ground station, they can operate their own satellites after launch. Usually, emerging space nations are limited by financial constraints of their budgets to develop launch sites, own launch capabilities and other main ground infrastructure to propel satellites into space, etc. But they are able to transfer knowledge on the following particular systems of a satellite: the onboard computer, onboard software and operations of a satellite.

In particular, to South Africa as an emerging space nation, UCT has partnered with the University of Stuttgart to establish a satellite simulation testbench of the "Flying Laptop" mission, and with their industrial partner Airbus Defence and Space. The "Flying Laptop" is a 120 kg microsatellite with dimensions of $60 \times 70 \times 80\,\text{cm}^3$. This academic satellite has been developed in cooperation with Airbus Defence and Space and was developed following a paradigm to initially simulate the satellite and after design has been consolidated (including flight software and operations concept), to subsequently replace simulated equipment by real flight hardware in the loop – until the satellite is complete. This approach is described in detail in [1].

1.1 The "Flying Laptop" Satellite

The primary objective of the "Flying Laptop" is to perform Earth observations using the multispectral cameras and maritime tracking of ships using the automated identification system (AIS). It was developed at the University of Stuttgart in collaboration with Airbus Defence and Space. Ph. D, Masters and senior undergraduate students developed and built the satellite, all providing contributions to the satellite or research related to the satellite. They launched it in 2017 and is operational since then (Fig. 1).

The "Flying Laptop" satellite comprises the following subsystems:

- Payload
- ACS
- Telemetry, tracking and command
- Data handling
- Thermal control system
- Power supply system
- Mechanisms (deployable solar panels).

The spacecraft (S/C) is operated applying industrial command and control standards such as the Consultative Committee for Space Data Systems (CCSDS) telecommunication channel and the European Space Agency (ESA) packet utilization standard. Its computer is based on a radiation hard LEON3FT chip and internally is based on a SpaceWire architecture using the remote memory access protocol (RMAP) for SpaceWire protocols [2, 3]. The "Flying Laptop" has six different control modes:

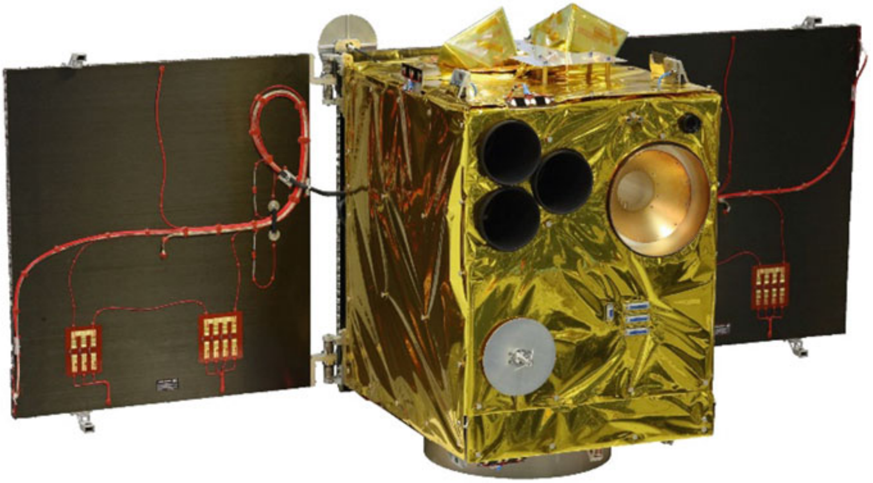


Fig. 1 The “Flying Laptop” satellite fully assembled in 2017 before Shipment to Launch © IRS, Uni. Stuttgart

detumbling mode, safe mode, idle mode, inertial pointing mode, nadir pointing mode, and target pointing mode.

The detumbling mode is to slow the rate of rotation of the satellite. The safe mode is used when there is a failure on the satellite. The idle mode is used when the payload is not in operation, but the satellite is doing routine operations. The inertial, nadir and target pointing modes are used when the payload is in operation.

1.2 The UCT Simulator Testbench

The SimTG simulator from Airbus Defence and Space models the S/C, space environment and the S/C dynamics. It runs on a PC with Linux operating system and in more advanced hybrid configurations; it is able to run the spacecraft simulation in real time. The satellite and its subsystems are simulated by software models of each hardware component. The modelling is performed in C+ + language with a user interface that resembles MATLAB. Telecommand and Telemetry packet definitions about the simulated models are stored in the Mission Information Base (MIB) of the connected Mission Control System (MCS) which itself is based on CCS5 from Terma A.S., Netherlands (Fig. 2).

The simulator infrastructure consists of:

- The Flight Software development environment, including Flight Software in source code.
- The mission control system CCS5 which is a standard for ESA and its industry partners.

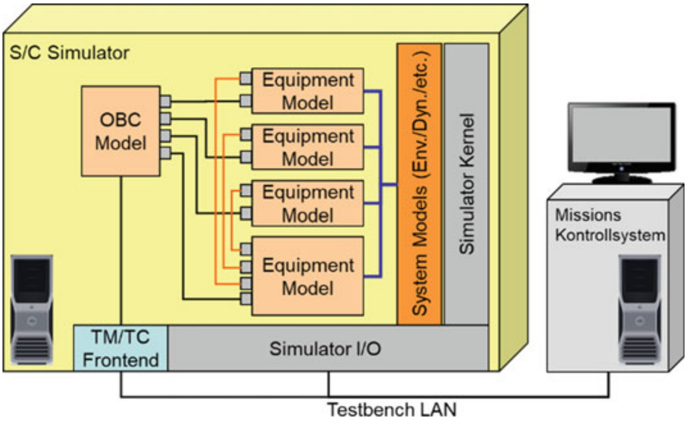


Fig. 2 Simulator testbench with fully simulated onboard computer © J. Eickhoff, from [1]

- The real-time S/C simulator with the satellite equipment models in source code.
- 3D visualization based on the Celestia software with diverse satellite geometry models.

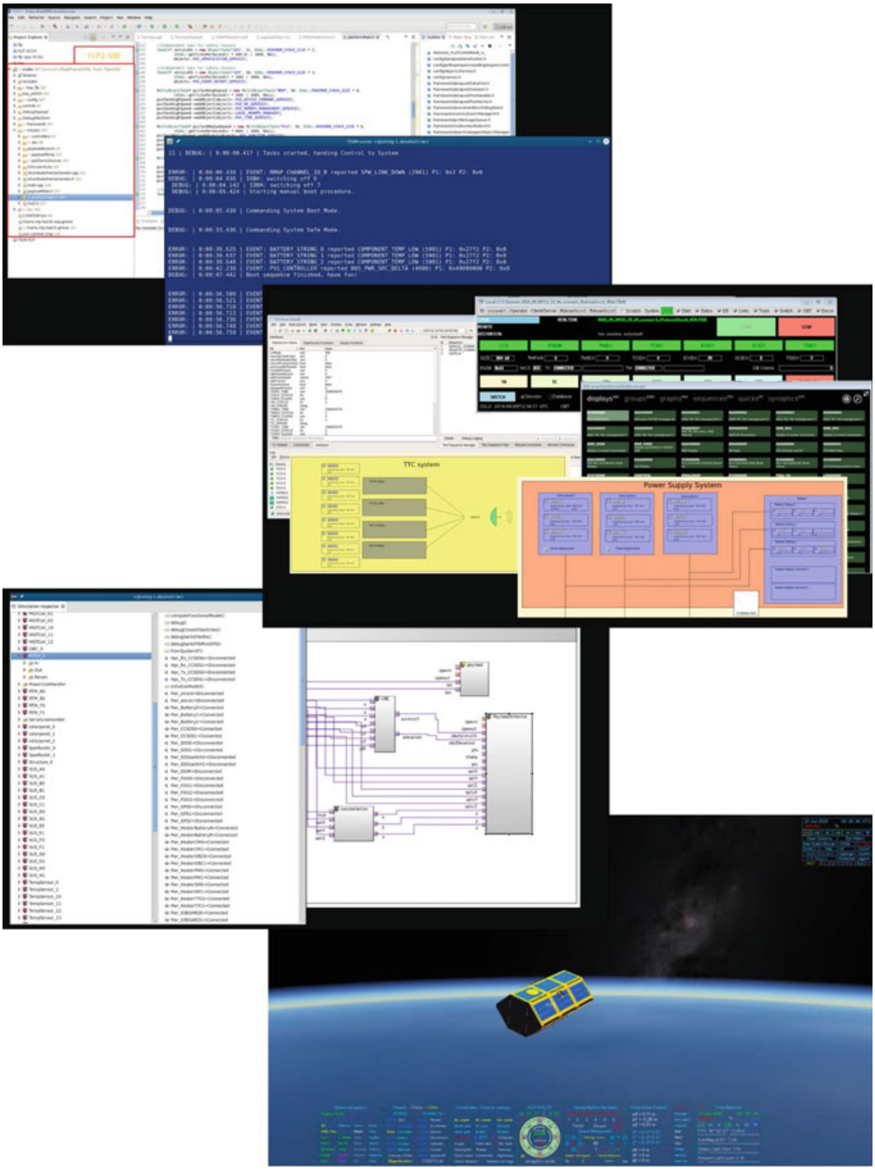
Figure 3 summarizes the main machine interface for each of the elements. In the current sponsored setup, all functions run on a single PC for cost reasons.

Such a fully simulated "digital twin" of a satellite can be used for flight software development and for development of satellite operations and failure management concepts. The simulator in this testbench simulates the onboard data handling, a simplified radio-transmission subsystem (without radio frequency (RF) chain), power management and a simplified thermal control subsystem. The payloads are rudimentary models only, since the simulator shall later be modified to model camera equipment from local South African suppliers (in example, SCS Space).

A flight operation can be commanded on the simulator from launch phase, to in-orbit operation and to End-of-Life disposal. This paper will demonstrate commanding of the first passover scenario to the simulated "Flying Laptop" and will demonstrate capability building in satellite operations and use of Flight Software and a professional MCS. The selected flight scenario targets for vegetation monitoring, which is South Africa's primary objectives of satellite research.

2 Vegetation Monitoring in South Africa

The South African Mission Advisory Committee's (EO-MAC) main objectives are to use remote sensing for vegetation monitoring in South Africa. This objective has been transferred to South Africa's next Earth observation satellite, the EO-SAT1. Other South African satellites that have contributed to vegetation monitoring in the past are SUN-SAT1 and SumbandilaSat.



```
11 | DEBUG: | 0:00:00.417 | Tasks started, handing Control to System

ERROR: | 0:00:00.430 | EVENT: RMAP_CHANNEL_10_R reported SPW_LINK_DOWN (2901) P1: 0x2 P2: 0x0
DEBUG: | 0:00:04.036 | IOBH: switching off 6
 DEBUG: | 0:00:04.142 | IOBH: switching off 7
 DEBUG: | 0:00:05.424 | Starting manual boot procedure.

DEBUG: | 0:00:05.430 | Commanding System Boot Mode.

DEBUG: | 0:00:33.436 | Commanding System Safe Mode.

ERROR: | 0:00:39.625 | EVENT: BATTERY_STRING_0 reported COMPONENT_TEMP_LOW (5901) P1: 0x27f2 P2: 0x0
ERROR: | 0:00:39.637 | EVENT: BATTERY_STRING_1 reported COMPONENT_TEMP_LOW (5901) P1: 0x27f2 P2: 0x0
ERROR: | 0:00:39.648 | EVENT: BATTERY_STRING_2 reported COMPONENT_TEMP_LOW (5901) P1: 0x27f2 P2: 0x0
ERROR: | 0:00:42.230 | EVENT: PSS_CONTROLLER reported BUS_PWR_SRC_DELTA (4500) P1: 0x49000000 P2: 0x0
DEBUG: | 0:00:47.442 | Boot sequence finished, have fun!

ERROR: | 0:00:56.509 | EVENT
ERROR: | 0:00:56.521 | EVENT
ERROR: | 0:00:56.710 | EVENT
ERROR: | 0:00:56.723 | EVENT
ERROR: | 0:00:56.736 | EVENT
ERROR: | 0:00:56.748 | EVENT
ERROR: | 0:00:56.759 | EVENT
```

Fig. 3 Simulator testbench screenshots © UCT, IRS Uni Stuttgart and Airbus

Vegetation properties, such as quality, species, biomass, etc., can be achieved by using multispectral and hyperspectral sensors. Vegetation includes crop, forest and rangeland monitoring. The current biomes in South Africa are: grassland, savanna, succulent Karoo, Noma Karoo, forest, fynbos, desert and thicket [4].

South Africa's main goal for vegetation monitoring is to do vegetation species discrimination, standing vegetation, biomass estimation, vegetation health or quality and vegetation cover mapping [4]. In addition, monitoring vegetation to cover change which relates to both vegetation quality and species mapping is of interest.

The study of vegetation monitoring allows researchers to see the effects of climate change in areas because of droughts or rainfall, farm animals eating grasslands, farmlands growing and reducing wildlife and the effect of urban development on surrounding grasslands.

South Africa National Space Agency (SANSA) plans to create a nationally co-ordinated programme to increase remote sensing skills and continue investment in high resolution space-borne imaging systems and advanced image processing capability. With optical camera systems on satellites the size of EO-SAT1 or the "Flying Laptop", vegetation monitoring with substantial resolution and in multiband spectral spread can be achieved.

For the first simulated scenario using the testbench—which was experimentally commanded to the real "Flying Laptop" by the University of Stuttgart - the panoramic camera was used which has a wide field of view and almost natural colour spectrum.

3 Flight Scenarios

The presented flight scenario mimics vegetation monitoring over the Cape Town area using one satellite. The defined operations sequence is commanded first to the UCT simulator and provides satellite simulation of the flight and the step-by-step commanding. The same flight passes were then commanded by Kai Leidig at the Institute of Space Systems (IRS), University of Stuttgart, to the real "Flying Laptop" satellite which generated imagery of the Cape Town area.

3.1 Simulated Scenario

The elaborated mission scenario comprised two observations in two consecutive orbits passing approximately over the Cape Peninsula. To analyse realistic ground tracks, the Two-Line Elements (TLEs) of the real "Flying Laptop" were introduced into an orbit analysis tool and the simulation propagated the virtual satellite along the tracks accordingly.

In Fig. 4, the 2D ground track of the flight scenario for Cape Town is shown as well as a 3D orbit for when the satellite passes over Cape Town, captures images and then sends the data down when it passes over the IRS ground station.

The set-up of the pure flight scenario comprises the simulation initialization, loading of flight software, specification of a consistent orbit propagation start condition of the satellite with all its parameters. Besides that, a simplistic commanding of

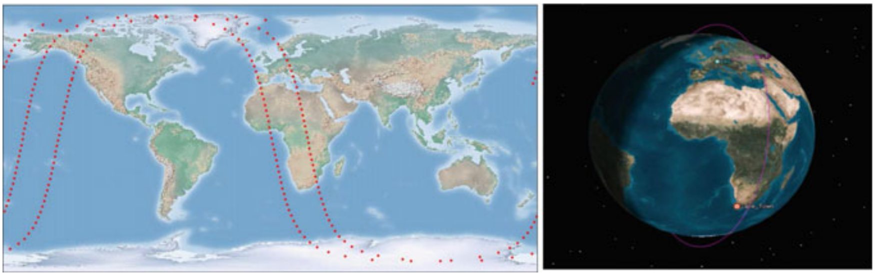


Fig. 4 Mission planning in 2D with IRS and target visibilities and sun/eclipse. © Airbus DS

a virtual vegetation monitoring camera was added to gain a first-hand experience on commanding.

Table 1 shows a breakdown of the mission plan procedure for the satellite during its orbit. The table only covers one of the consecutive paths. The satellite was in an idle mode in orbit, before it reached its target point. From the table, it can be seen that the satellite first changes from an idle (sun-pointing rotating) mode to an inertial

Table1 Procedure outline for Cape Town flight scenario

Step Number	System/Subsystem	State Transition	Satellite Mode	ACS Mode
0	System		Idle	Idle
1	System	Activate nominal operations mode	Idle	Idle
2	Payload	Stand-by mode	Nominal operations	
3	System/ACS/Payload		Nominal operations	Inertial Pointing
4	System/ACS/Payload		Nominal operations	Nadir Pointing
5	System/ACS/Payload		Nominal operations	Target Pointing
6	System/ACS/Payload	Image acquisition mode	Nominal operations	Target Pointing
7	System/ACS/Payload		Nominal operations	Nadir Pointing
8	System/ACS/Payload		Nominal operations	Inertial Pointing
9	System/ACS/Payload	Payload Idle mode		
10	System/ACS	Activate system Idle mode	Idle	Idle
11	Payload	Off mode		
12	System		Idle	Idle

mode and then next to a nadir pointing to get the camera view “down to Earth”. Then, the satellite is commanded to point to the target (Cape Town), rotating to left looking in the pass leading East of the Cape and right looking in the pass transiting West of the Cape. Since the simulated satellite does not yet have a realistic payload modelling, the level of detail ends at this stage.

3.2 Scenario Commanded to “Flying Laptop” Satellite

The vegetation flight scenario on the simulator was now commanded to the real “Flying Laptop” satellite by the University of Stuttgart. The satellite performed two ascending passes, one from the East and another from the West, with the focal point on the Cape Peninsula. For simplicity of operations, the satellite’s panoramic camera was used instead of the multispectral one. Both cameras anyway would not be realistic for a professional vegetation monitoring as e.g., performed by the European Union’s Copernicus Sentinel-2 fleet.

The “Flying Laptop” satellite captured 30 images of the Cape Peninsula in both passes. The picture shown was taken during the overpass that occurred on the 18th of February 2020. The pass number 15440 is shown in Fig. 5 when the satellite was targeting the Cape Peninsula from the East. Figure 5 shows the field-of-view and the satellite ground track—both from the IRS mission planning tool. Figure 6 shows the captured image from the satellite of the Cape Peninsula—unfortunately with several reflecting clouds.

The pass number 15442 is shown in Fig. 7 when the satellite was targeting the Cape Peninsula from the West. Shown in Fig. 7 is the panoramic camera’s field-of-view and satellite ground tracking. Figure 8 shows the captured of the Cape Peninsula and St. Helena Bay—again with a lot of clouds since this was the consecutive orbit 90 min later.

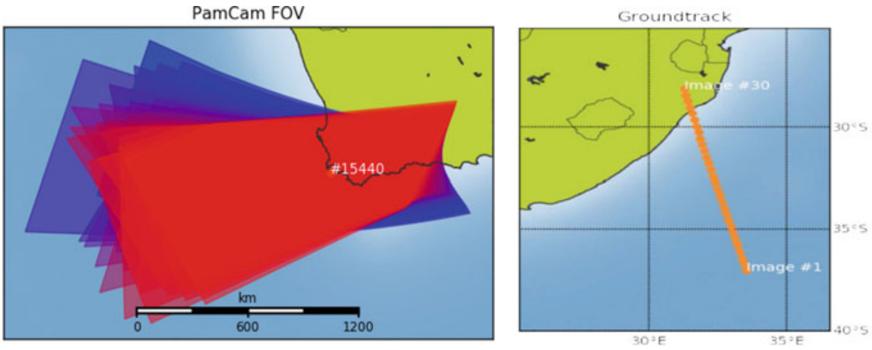


Fig. 5 Panoramic camera field of view for passover one and ground track

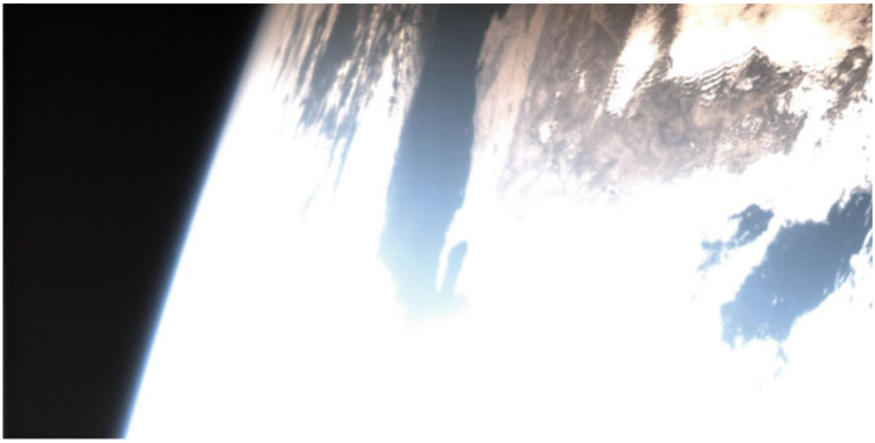


Fig. 6 The Cape Peninsula—© IRS, Uni Stuttgart

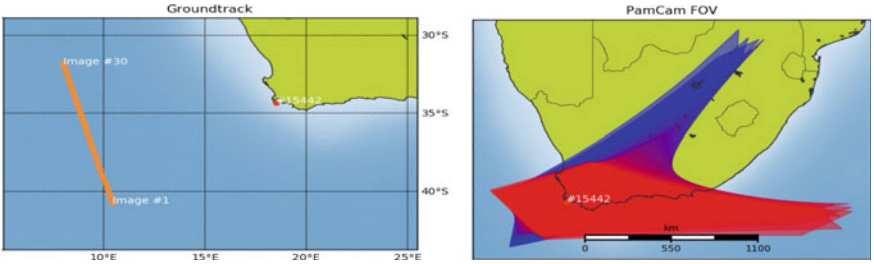


Fig. 7 Panoramic camera field of view and satellite ground track

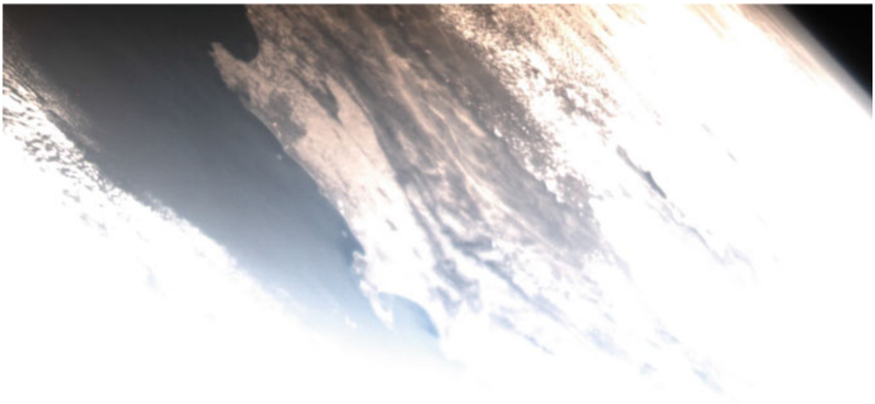


Fig. 8 Cape peninsula and St. Helena Bay—© IRS, Uni Stuttgart

4 Simulated Gecko Imager

The South African developed Gecko imager is a small satellite camera payload developed by SCS space. The internal components and basic operation of the camera were simulated in the UCT Satellite testbench. The camera was integrated to the simulated "Flying Laptop" and flight scenarios were conducted. The flight scenarios were passover flights over the Cape Peninsula. The satellite was located at an altitude of 700 km. To generate the equivalent pictures that the Gecko would capture when in full operation in the space environment, the StaticMap [5] program is used to generate the required images.

4.1 Development Process

The main operations of the Gecko Imager were simulated, along with all major components of the Gecko - i.e., electronics and optics assembly. Once development of the simulated Gecko was complete, it was connected to other satellite subsystems, and at each level of integration, tests were conducted to confirm operation. Commands were sent to the "Flying Laptop" satellite through the CCS5 to confirm the Gecko can be operated from the ground station.

4.2 Gecko Imager Flight Scenario

The flight scenario simulated on the testbench was to capture images of the Cape Peninsula from the East, West and Central areas of the Peninsula. In Fig. 9, the satellite flight scenario occurred and captured an image over the central Cape Peninsula.

In Fig. 10, the satellite flight scenario occurred to the east of the Cape Peninsula. The camera captured False Bay and parts of the Cape Peninsula, which is in the bottom left corner.

In Fig. 11, the satellite flight scenario occurred to the West of the Cape Peninsula. The camera captured parts of the Cape Peninsula and Cape Town.

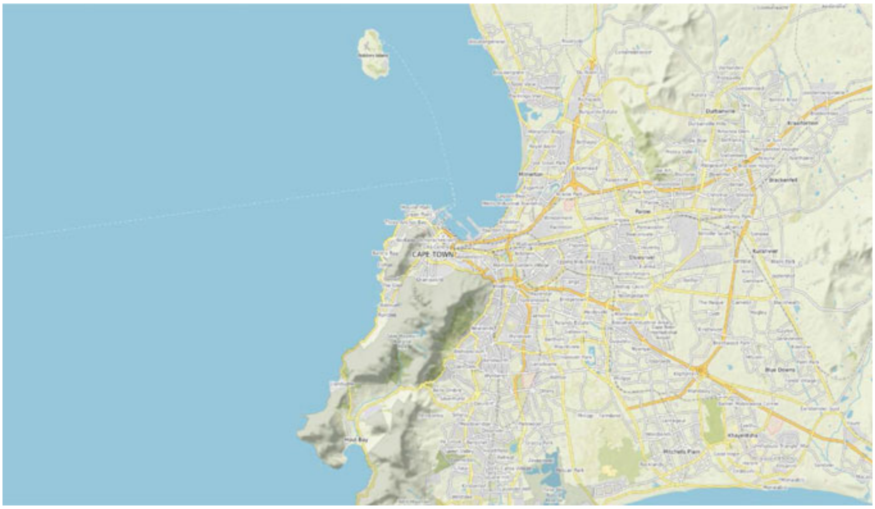


Fig. 9 Cape peninsula from the simulated Gecko imager

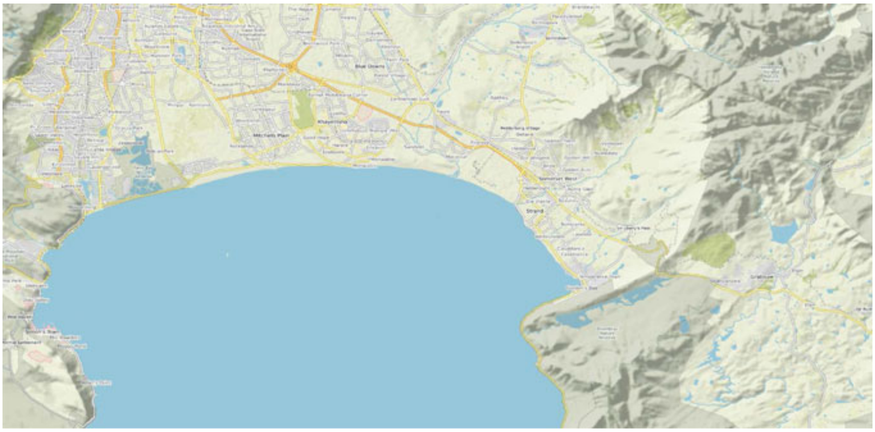


Fig. 10 Cape peninsula from the east captured by the simulated Gecko imager

5 Better Coverage with Multi-Satellite Constellation

In Fig. 12, there is a good quality image of the Cape Peninsula, achieved through a satellite pass leading directly over the Cape Peninsula and during clear sky conditions. The photo is an older image from the "Flying Laptop" from 2018, taken with the same panoramic camera.

To achieve a shorter revisit cycle to reach the target as above in realistic time intervals, often enough, preferably in nadir pointing mode and to track plant growth

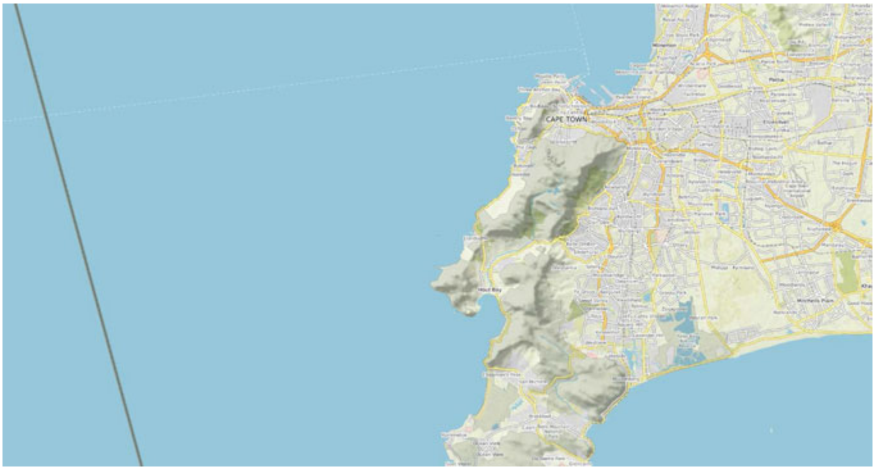


Fig. 11 Cape Peninsula from the west captured by the simulated Gecko Imager

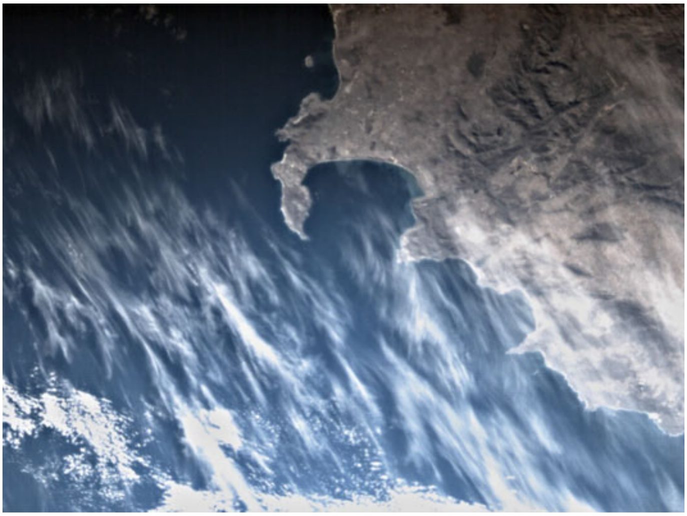


Fig. 12 Good quality picture of Cape Peninsula through Nadir looking Passover—© IRS, Uni Stuttgart

on a weekly basis, usually a fleet of 2–4 or more satellites is useful. This implies the capability to operate a fleet of satellites in constellation or formation. For this purpose, the IRS, University of Stuttgart, has developed the Multi-Mission Operations Toolkit (see [6]). For testing this with realistic scenarios, up to four simulated satellite instances of the same type as now available for UCT are being commanded. Figure 13 shows a block diagram of the internals of the toolkit and Fig. 14 provides an impression of the user interface.

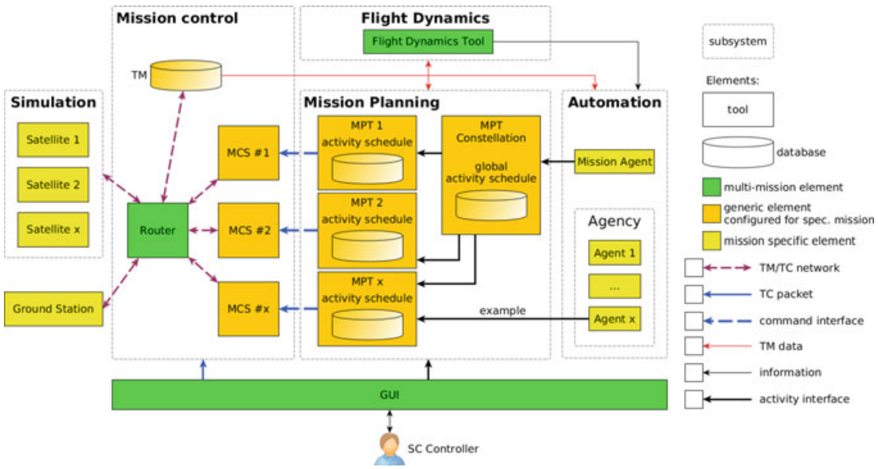


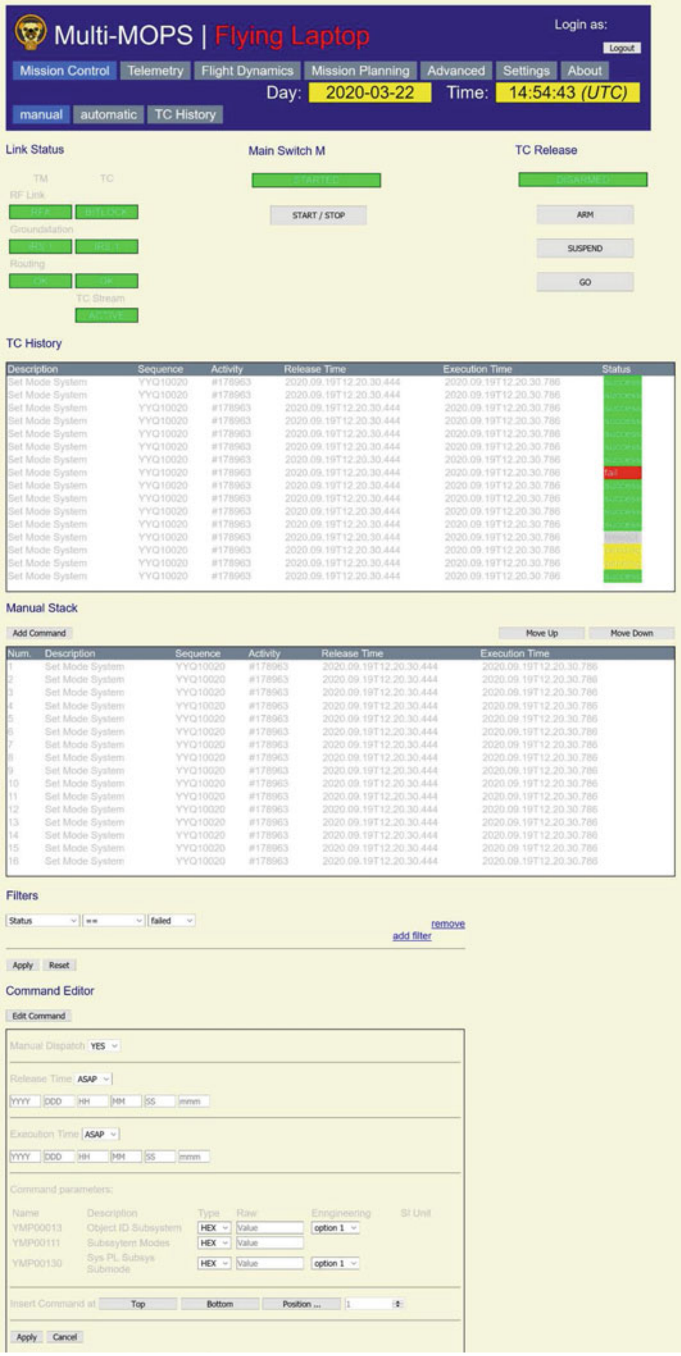
Fig. 13 Multi-Mission operations system architecture © K. Leidig, IRS, Uni Stuttgart [6]

6 Effects on Local Capacity Growth

6.1 *Lectures*

During the last years, UCT has gained several guest lecturers who contributed significantly to the knowledge in satellite design, onboard computers and onboard software as well as in satellite operations, simulation and verification—see Fig. 15.

- The lectures for satellite onboard computers focus on radiation hard and commercial space microprocessors, on data buses, command decoding and telemetry encoding and all key elements that make-up the onboard computers.
- The onboard software lectures focus on the software that runs on onboard computers, payload computers and data processing, satellite memory, the complex AOCS subsystem, and also the power control and distribution unit. Students learn about onboard software static and dynamic software architecture.
- The lectures for satellite operations allow students to understand the satellite's operational concept design. This includes the S/C operational phase concept and the satellite authentication concept, to just list a few. Other topics are S/C onboard autonomy, failure recovery operations, mission operations infrastructure, and training.
- The satellite simulation lectures teach students simulation-based approaches used in design verification and infrastructures for spacecraft, as a well as the different setups of verification infrastructure that can be applied.
- The satellite verification and testing lectures teach students on how to efficiently verify onboard software and hardware equipment for the satellite and how to



Multi-MOPS | Flying Laptop

Login as:

Logout

Mission Control | Telemetry | Flight Dynamics | Mission Planning | Advanced | Settings | About

Day: 2020-03-22 Time: 14:54:43 *(UTC)*

manual | automatic | TC History

Link Status

	TM	TC
RF Link	RFA	BITLOCK
Groundstation	IRS 1	IRS 1
Routing	OK	OK
TC Stream		ACTIVE

Main Switch M

STARTED

START / STOP

TC Release

DISARMED

ARM

SUSPEND

GO

TC History

Description	Sequence	Activity	Release Time	Execution Time	Status
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	fail
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	[illegible]
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	pending
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	pending
Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786	success

Manual Stack

Add Command | Move Up | Move Down

Num.	Description	Sequence	Activity	Release Time	Execution Time
1	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
2	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
3	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
4	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
5	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
6	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
7	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
8	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
9	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
10	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
11	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
12	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
13	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
14	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
15	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786
16	Set Mode System	YYQ10020	#178963	2020.09.19T12.20.30.444	2020.09.19T12.20.30.786

Filters

Status | == | failed

remove

add filter

Apply | Reset

Command Editor

Edit Command

Manual Dispatch: YES

Release Time: ASAP

YYYY | DDD | HH | MM | SS | mmm

Execution Time: ASAP

YYYY | DDD | HH | MM | SS | mmm

Command parameters:

Name	Description	Type	Raw	Enngineering	SI Unit
YMP00013	Object ID Subsystem	HEX	Value	option 1	
YMP00111	Subsaytem Modes	HEX	Value		
YMP00130	Sys PL Subsys Submode	HEX	Value	option 1	

Insert Command at: Top | Bottom | Position ... | 1

Apply | Cancel

Fig. 14 Multi-Mission operations system—User Interface © K.Leidig, IRS, Uni Stuttgart [6]

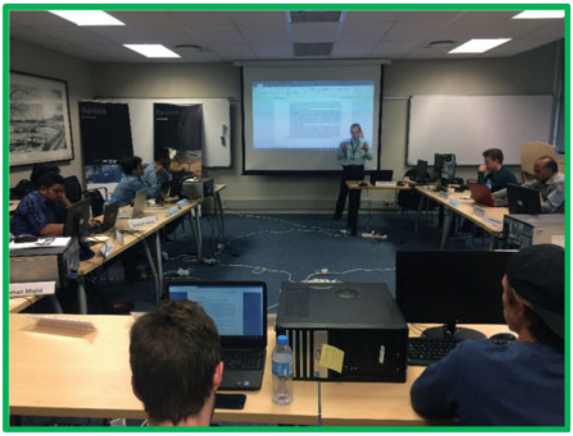


Fig. 15 Students learning about space operations in a classroom © UCT, SpaceLab

create a functional test program for the onboard software and environmental test of the hardware.

The now available simulator testbench extends significantly the possibilities for the students since it allows for hands-on experience in these fields using a virtual satellite with a full featured real onboard software. This can be used for hands-on trainings and workshops. And can be the nucleus for deriving from the existing design a future South African satellite mission. It opens the door for Master and PhD thesis projects in satellite software, design, verification, and simulation.

6.2 Spacecraft Modelling

Spacecraft modelling involves modelling all aspects of the spacecraft from the subsystems of the satellite to the communication protocols, power consumption and electrical interfaces between all systems and subsystems.

As students get used to working on the simulator, they learn to model all aspects of a satellite in depth as done in (local) industry. This reduces the gap between local academia and industry and improves the coordination between these two spheres.

Students learn to model and test the satellite subsystems in MATLAB/Simulink and transfer the models from Simulink to the simulator. Important models to simulate on a satellite would be the AOCS, the thermal control and the power control subsystems. All these models and their control algorithms can be verified and tested on the simulator testbench.

6.3 Satellite Operations

Satellite operations involve specific phases throughout the lifetime of a satellite such as.

- The Launch and Early Orbit Phase
- The Commissioning Phase
- The Routine Operations Phase
- And finally, the Disposal Phase known as the De-Orbit Phase.

The simulator testbench will help the students as well as South Africa to grow in those four keys areas of satellite operation while allowing students to experience all the phases of satellite operations and train them in the mission operation execution.

The simulator will also provide students with experience being part of a flight control team comprising: the flight director position, various subsystem specialists like command operator, planner, flight dynamics, ground data systems.

This will contribute to reduce gaps between local industry and local academia. In future, students would be well equipped with industry standards, hands-on experience and operational knowledge. The experience gained would apply to any spacecraft, especially LEO and GEO missions, which are the most commonly used. For South Africa, LEO spacecraft operation would be most beneficial, to increase remote sensing applications.

The UCT testbench surely will attract South African satellite operations industry.

6.4 Handling a Real Satellite Flight Software

Students gain experience in handling the CCS5 generic mission control system. The mission control system allows students to learn how to create command and telemetry definitions as well as synoptic displays for different space missions. It also offers to learn how to monitor and control over one satellite as satellite operators. This is an invaluable skill that can be utilized in industry and further reduces gaps between academia and industry.

When students leave for industry, they will be able to contribute to the development of own satellite flight software for local entities, helping to boost South Africa’s impact in the global space market. This will significantly increase the level of knowledge and expertise in commercial companies and governmental organisations. Local partners will have the opportunity for training students to use their satellite flight software, allowing more valuable time to be used for development. Great commercial opportunities in cooperation with Airbus Defence and Space are opened up here, since Airbus Defence and Space uses the same flight software architecture in their “Flexible LEO Platform” (FLP2) for small and medium satellites. Industry partners also will be able to write software drivers for their local payload sensor products for the FLP2 platform to offer them to a global community.

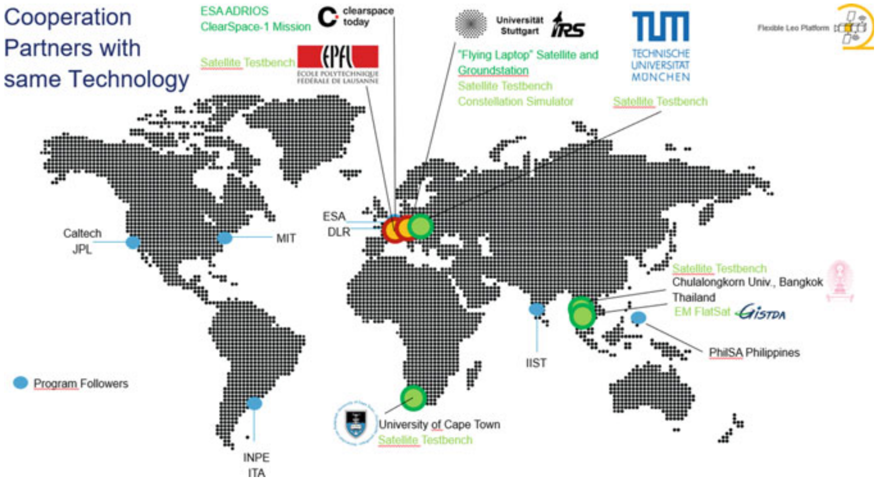


Fig. 16 Partners cooperating on missions based on the "Flexible LEO Platform" (FLP2) © Airbus Defence and Space

6.5 *Enabling Space Research*

With this testbench based on the FLP technology, and software and simulation infrastructure, UCT and its South African collaborators become partner of a global community of users. South African institutions like UCT and local industry can partner with other nations already investing in the newer "Flexible LEO Platform" (FLP2) technology from Airbus Defence and Space and with their partners for future missions, like Thailand and the Philippines as shown in Fig. 16.

The UCT testbench will also allow students to perform more hardware in the loop testing and this will further allow UCT to be a centre for South African satellite development greatly improving South Africa's remote sensing technologies (e.g., cameras, sensors, data processing algorithms). Figure 17 shows the types of applications South Africa could develop for its satellites.

7 Conclusion

The simulator testbench is the essential step forward to enable UCT and its students to gain practice in the so far lectured technology fields.

- Onboard Computers,
- Onboard Software
- Operations Scenarios,
- Simulation and Verification

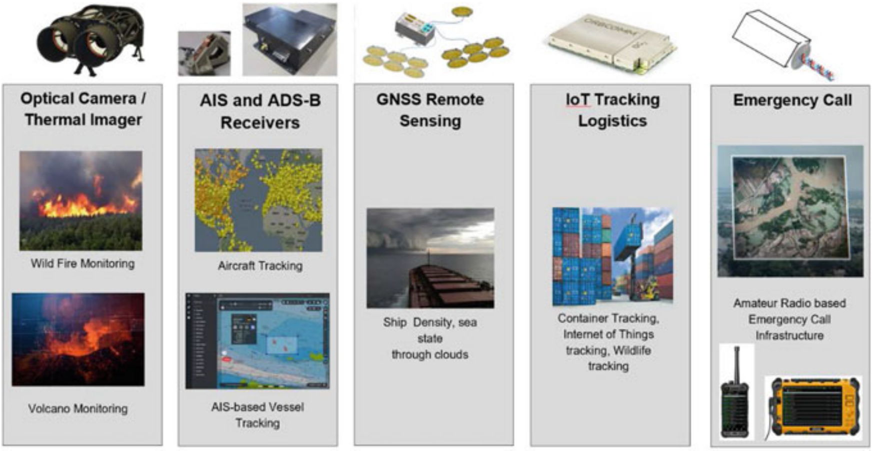

Optical Camera / Thermal Imager
Wild Fire Monitoring
Volcano Monitoring
AIS and ADS-B Receivers
Aircraft Tracking
AIS-based Vessel Tracking
GNSS Remote Sensing
Ship Density, sea state through clouds
IoT Tracking Logistics
Container Tracking, Internet of Things tracking, Wildlife tracking
Emergency Call
Amateur Radio based Emergency Call Infrastructure

Fig. 17 Multi-Mission operations system architecture © K. Leidig, IRS, Uni Stuttgart [6]

The simulated satellite with its LEON3 processor and SpaceWire architecture represents latest technology as used by global space players —e.g. at Airbus Defence and Space. With further funding in example from SANSA or local industry the simulator testbench setup at the University of Cape Town has the potential to be upgraded to the Airbus "Flexible LEO Platform" design and could resemble the testbench currently installed at the Chulalongkorn University in Thailand [7]. This testbench brings latest technology hands-on expertise to South Africa.

Acknowledgements The UCT SpaceLab team and its students thank all supporters of this initiative and especially the in-kind sponsors namely

- University of Stuttgart, Simulator Models and tailored Flight Software
- Terma A.S CCS5 license
- Cobham Gaisler AB, TSIM CPU Emulator and last but not least the team at
- Airbus Defence and Space, Friedrichshafen, Germany - SimTG License, initiative, overall organization and export clearance/shipment of the setup to UCT SpaceLab.

References

1. Eickhoff J (2009) Simulating spacecraft systems
2. Eickhoff J (2016) The FLP microsatellite platform: flight operations manual
3. Eickhoff J (2012) Onboard computers, onboard software and satellite operations: an introduction
4. Mutanga O, Dube T, Ahmed F (2016) Progress in remote sensing: vegetation monitoring in South Africa. S Afr Geogr J 98(3):461–471
5. Lingg C. StaticMap. https://github.com/komoot/staticmap

6. Leidig K, Gaißer S, Mohr U, Schweigert R, Wenzel S, Klinkner S, Eickhoff J (2020) Multi-mission operations system supporting satellite constellations, SpaceOps-2020,4,14,x281. In: 16th international conference on space operation, Cape Town, South Africa, 3–5 May 2021
7. Eickhoff J (2020) FLP2 satellite testbench in Thailand, 25 Feb 2020. https://www.linkedin.com/pulse/flp2-satellite-testbench-thailand-jens-eickhoff/ Accessed 09 Mar 20

Ground Systems and Software

From Theory to Practice: Operational Implementation of Telemetry Outlier Detection at EUMETSAT

P. L. Losco, A. De Vincenzis, J. Pergoli, and R. Dyer

Abstract In order to better support the Telemetry monitoring of both current and future missions at EUMETSAT, flight control teams have developed semi-supervised Outlier Detection algorithms which compliment traditional monitoring techniques such as Out of Limit Monitoring. This Outlier Detection has now been in use for several years. Through the process of developing and utilising the Outlier Detection algorithms, it has been found that the main challenges are not related to the choice, tuning or development of the algorithm. Instead, they are related to the more practical aspects surrounding the presentation of results, choice of parameters to be monitored, filtering of results and the labelling of Nominal Data. Moreover, these practical problems are largely algorithm independent and so once an adequate algorithm was developed, it was realised that chasing a better algorithm before these practical had been addressed would simply lead to a delay in the use of this technique. This paper discusses the Outlier Detection algorithm and its development, but the main focus is on the lessons learned from this process which led to the identification of practical problems and solutions to them. In particular, the Outlier Detection results are presented in a way which allows users to understand why outliers have been flagged, which minimises the "Black Box" effect. This, however, implies limiting the number of dimensions used when performing Outlier Detection. Another issue observed was floods of outliers being reported during non-nominal operations or anomalies. Not only does this mask real issues, but being since Outlier Detection is a relatively new technique, it also serves to erode trust in the process. This has

P. L. Losco (✉)
LSE Space GmbH, Gilching, Germany
e-mail: pio.losco@external.eumetsat.int

A. De Vincenzis
Exostaff GmbH, Bickenbach, Germany
e-mail: alberto.devincenzis@external.eumetsat.int

J. Pergoli
CLC Space GmbH, Alsbach-Hähnlein, Germany
e-mail: jonathan.pergoli@external.eumetsat.int

R. Dyer
EUMETSAT, Darmstadt, Germany
e-mail: richard.dyer@eumetsat.int

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_11

mainly been resolved by a conditional Outlier Detection method which has been developed by categorising parameters hierarchically as Influencers and Followers, and also by careful selection of parameters. Finally, the main sticking point for any semi-supervised Outlier Detection techniques is keeping up to date with the labelling of Nominal Data. Again, this is partially addressed through careful selection of parameters in order to avoid having too many datasets to maintain, however it is the one area where development is still on-going within EUMETSAT, and so this paper describes the future plans to facilitate the labelling of Nominal Data.

Acronyms/Abbreviations

CHART	Component Health Analysis & Reporting Tool
EPS	EUMETSAT Polar System
FCT	Flight Control Team
FDIR	Fault Detection Isolation and Recovery
GEO	Geostationary Orbit
GERB	Geostationary Earth Radiation Budget Experiment
HKTM	Housekeeping Telemetry
IOU	Instrument Optical Unit
LEO	Low Earth Orbit
MetoOp	Meteorological Operational Satellite
MHS	Microwave Humidity Sounder
MSG	Meteosat Second Generation
NRT	Near Real-Time
OOL	Out-Of-Limit
OTD	Outlier Threshold Distance
SEVIRI	Spinning Enhanced Visible and InfraRed Imager

1 Introduction

Spacecraft operators are increasingly facing the problem of how to efficiently and effectively monitor in-orbit Spacecraft. With the next generations of satellites (or constellations of satellites) being characterised by an increasing number of on-board observables, the manual inspection of all parameters is not feasible. Automated monitoring using traditional Out of Limit (OOL) monitoring also has many limitations [2].

The FCT at EUMETSAT is therefore been evolving the satellite telemetry monitoring concept [3], in particular following the trend of developing a semi-supervised learning algorithm for outlier detection [1, 2, 4–6]. Outlier detection is able to identify unusual behaviour that may be indicative of problems that are developing, without needing telemetry to go outside the defined ground or on-board limits. Various statistics for each parameter are automatically calculated over an appropriate period (e.g. one orbit for Earth Orbiting Spacecraft), and statistics from the latest period are compared against statistics from a manually selected nominal period. In the case of EUMETSAT, a comparison is based on the distances between data-points in an n-dimensional space, where n is the number of statistics used. The algorithm is used in an operational-offline Near Real Time (NRT) context, which means that engineers have access to the results offline and use it to trigger investigations into unusual behaviour.

Despite the algorithm's relative simplicity, it was found that after several iterations of tuning, its performance when comparing a test-dataset to a manually selected Nominal-dataset was perfectly adequate for any parameter on any mission. At the same time, practical barriers were preventing its wider adoption in a real operational context. In particular, work was clearly necessary in the following areas:

- Deciding which parameters to select for monitoring;
- Avoiding false alarms/floods of alarms by implementing conditional monitoring;
- Finding an easy and intuitive way for engineers to maintain the nominal dataset.

Spending effort on formal performance assessment and optimisation of the algorithm while these practical problems remained was judged to be an ineffective use of resources. This is especially true since the practical problems were clearly algorithm independent—i.e. they could be solved now and would still be applicable even if the algorithm were substituted at a later date.

The Sect. 2 is dedicated to the overview of the Outlier Detection algorithm developed at EUMETSAT and its integration into CHART framework, the existing offline monitoring facility [7].

Then, the three aforementioned problems–whose discussion represent the main objective of this paper- are respectively analysed in Sects. 3, 4 and 5. Each section aims to show how each single topic was identified within the operational context, how it was characterised and how it was/will be tackled. Where possible, practical examples from both LEO and GEO missions are used in order to support the discussion. Specifically, a description of the so-called *followers-influencers* logic to group parameters in order to reduce the number of false positive is presented in Sect. 3. Section 4 contains a strategy to choose the relevant parameters. Section 5 describes the current approach for engineers to maintain the tool in use. Section 6 concludes this work and suggests some future development of both the current algorithm and its application at EUMETSAT.

2 Overview of the Algorithm, Its Development and Initial Exploitation

2.1 Algorithm Description

The Outlier Detection algorithm was developed based on a k Nearest Neighbours (k-NN) approach [5], with a Nominal and Test dataset being defined by periodic parameter statistics in a 4-D Space of Min, Mean, Max and Standard Deviation. Typically, Orbital Statistics are used as these are pre-computed and stored in the back-end database of the Telemetry Archive, CHART. The concept is illustrated in Fig. 1. The timeseries plot in 1a) is showing the evolution of a parameter, with the Nominal Data Set covering nominal behaviour in light blue and two Test Data Sets highlighted in grey. The Nominal and Test Dataset(s) are then visualised in 1b) using three of the possible six 2-D projections, with the Nominal Dataset being the black points as shown in the two plots at the bottom, and the Test Dataset being the Green, Yellow, Orange and Red data points, according to their outlierness.[1]

The first step in the process is data normalisation. This is necessary to ensure all statistics carry equal weighting. The Nominal Dataset is scaled such that the values of each statistic fall in the range 0–1. Each statistic in the Test Dataset is then scaled by the same factor respectively.

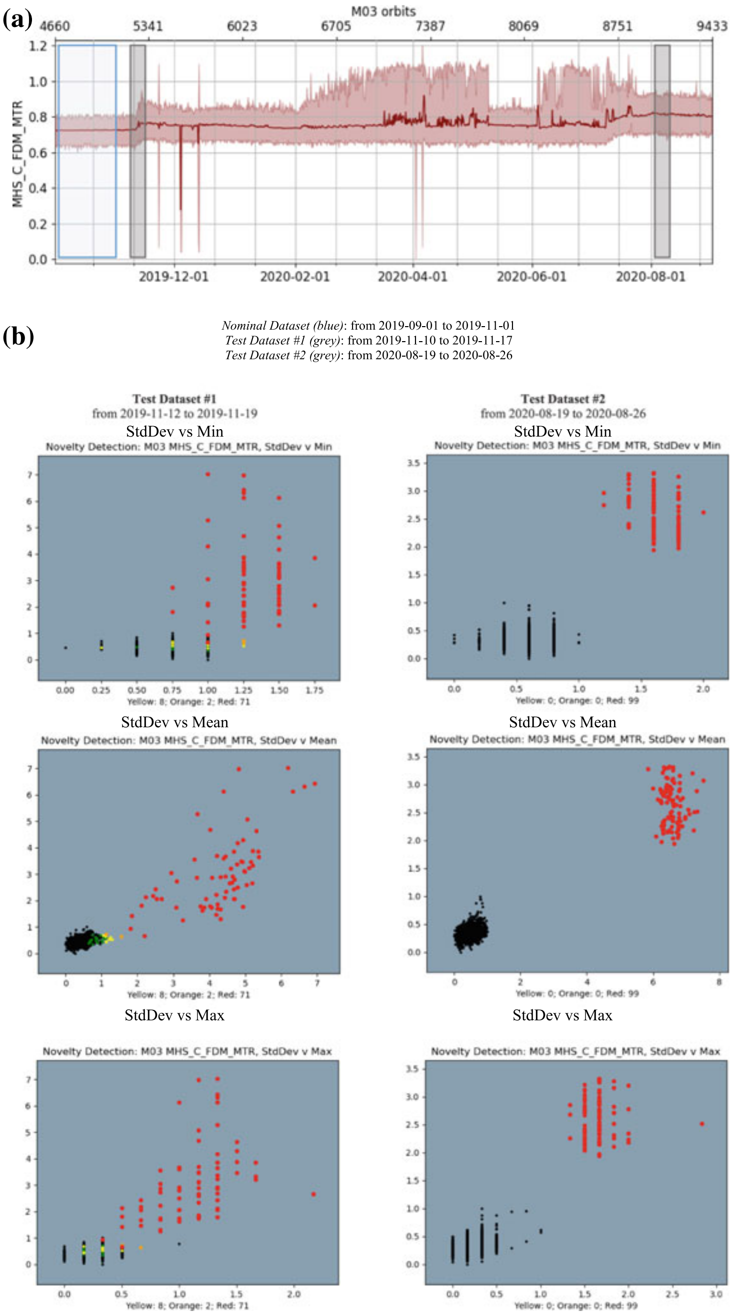
Each point in the Test Dataset is assessed by comparing the Euclidean distance in 4D space to its nearest neighbour in the Nominal Dataset against an Outlier Threshold Distance (OTD). The OTD is a representative measure of the local distances between points in the Nominal Dataset. To determine a suitable OTD, the crux is to prevent a few very tightly packed points in the local Nominal Dataset having too much influence without being overly reliant on the most isolated points. To achieve this, we identify our Test Data point's 20 nearest neighbours in the Nominal Dataset, and for each one of those we find the mean distance to their 20 nearest neighbours. This gives an array of 20 values, with the OTD being the mean value of the 10 largest.

This approach was still found to be overly sensitive when the local Nominal Data points were closely packed. Tuning the algorithm in terms of the number of neighbours assessed or cut-off thresholds could resolve this, however this made the algorithm too insensitive when the Nominal Data points were sparse. Figure 2 shows for example how a Test Data point in a hole in the Nominal Data may be marked as an outlier, while simultaneously a clear outlier near some sparse points in the Nominal Data would be marked as an inlier.

This problem was simply resolved by dynamically stretching OTD when below a certain value and shrinking it when above, according to the following equation.

$$\mathrm{OTD} = \mathrm{OTD}\left(\frac{\alpha}{\mathrm{OTD}}\right)^{\beta} \tag{1}$$

[1] Note that in this case, the anomaly was detected several months before an OOL would have been triggered.



Nominal Dataset (blue): from 2019-09-01 to 2019-11-01
Test Dataset #1 (grey): from 2019-11-10 to 2019-11-17
Test Dataset #2 (grey): from 2020-08-19 to 2020-08-26

Test Dataset #1
from 2019-11-12 to 2019-11-19

Test Dataset #2
from 2020-08-19 to 2020-08-26

StdDev vs Min

StdDev vs Mean

StdDev vs Max

Fig. 1 **a** Time Series plot of motor current (MHS_C_FDM_MTR) of the MHS instrument onboard Metop-C. **b** Three projections used to display Outlier Detection results for two different Test Datasets and the same Nominal Dataset

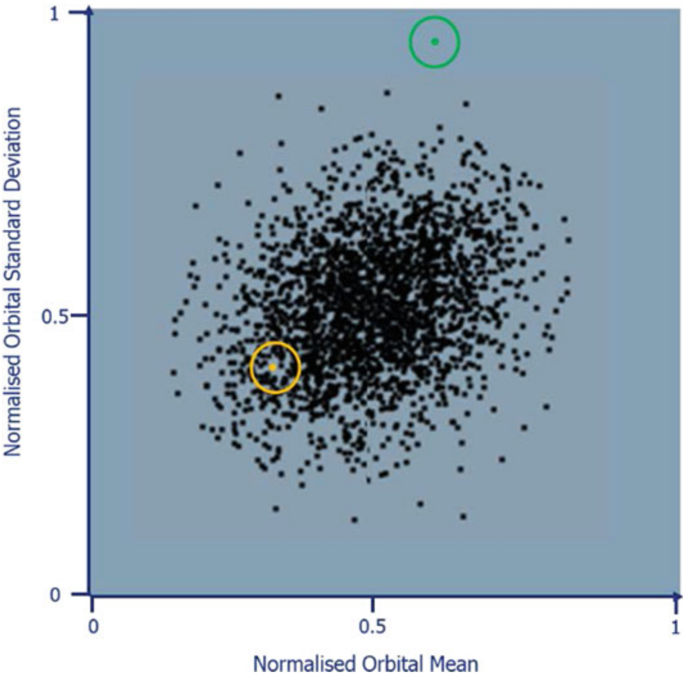


Fig. 2 Example of misclassification

Alpha determines the value of OTD above which OTD is shrunk and below which OTD is stretched.

Beta determines how strong this scaling is as OTD moves away from alpha. Thanks to the normalisation of the Nominal Dataset, the values of *alpha* and *beta* are independent of the Nominal Dataset. By process of trial and error, the values of *alpha* $= 0.05$ and *beta* $= 0.5$ were found to consistently produce results which were subjectively good.

The final step is to assign a non-binary classification to each data point in the Test Dataset. Based on the distance to our Test data point's nearest neighbour in the Nominal Dataset compared against OTD, the following classification is used.

$$\begin{aligned}
&Green < OTD \\
&OTD \leq Yellow < 3*OTD \\
&3*OTD \leq Orange < 6*OTD \\
&6*OTD \leq Red
\end{aligned}$$

The non-binary classification allows an additional layer of logic to be applied at the reporting stage. Specifically, the operator can be alerted only when the percentage of Test data points flagged Yellow, Orange or Red is above a certain threshold. A single data point flagged Yellow is of no interest, however a significant proportion of the data points being flagged Yellow is.

These reporting thresholds have been set to 30% for Yellow and 5% for Orange and are effectively removed for Red.

2.2 Algorithm Development

The algorithm was developed and tuned by process of trial and error. To support this development approach, three of the possible six 2-D projections of the dataset were used for visualisation of the results in "splatter plots", namely StdDev vs Min, StdDev vs Mean and StdDev vs Max as shown in Fig. 1b. The other three projections could theoretically be necessary to fully understand the algorithm results, however based on empirical data they were found to be consistently superfluous. Any outliers in the Min vs Max, Min vs Mean or Max vs Mean projections were always apparent in the Standard Deviation dimension, and so captured by at least one of the three existing projections.

In each iteration, splatter plots for many different parameters were created, and the classification of the Test data points was subjectively assessed. The algorithm and any thresholds were then tweaked to eliminate anything considered to be a misclassification. The process was repeated until the algorithm produced results which were consistently judged to be correctly classified. Members of the FCT performed this work, and so very little time was required for each iteration. This meant the initial development and tuning phase took approximately 2 days. The performance of the algorithm is also continuously reviewed as it is used, however since the initial tuning and development period, only the Yellow, Orange and Red thresholds and their respective reporting thresholds have been adjusted. The algorithm itself has not required any significant update.

It should be noted that review of normal time-series plots of the data were deliberately avoided during this tuning and development process. Firstly, this sort of algorithm is intended to detect things which are not necessarily apparent in time-series plots. Secondly, time-series plots would present the data in a manner where engineers may have some preconceptions about what constitutes an inlier and outlier, whereas the splatter plots in Figs. 1b and 2 ensure that the data are presented in an abstract fashion.

2.3 Algorithm Performance

As explained in Sect. 2.2, the algorithm performance has not been formally assessed, and there has not been any systematic optimisation of it via tuning of the various parameters and thresholds – the process was more trial and error with subjective assessment.

It is nevertheless intended to perform a more systematic analysis in the future using Receiver Operating Characteristic (ROC) curves. These are created by plotting the true positive rate against the false positive rate at various threshold settings. A good example of this analysis in the frame of Outlier Detection was by Fuertes et al., 2018 [8]. With three tuneable variables available for optimization of the algorithm (α, β [(1)] and the number of neighbours' κ), the aim would be reaching the highest value of the area under the curve on a ROC plot with a reasonable computational time considering the calculation power available. It should however be noted that although more systematic, the problem remains that the definition of the 'ground truth'—i.e. what is definitively an outlier or inlier—is still subjective.

So far only a quick study has been done, considering only one spacecraft parameter, leading to a Receiver Operating Characteristic of ~97% as shown in Fig. 3. An ideal ROC Curve would trace a vertical line up the y-axis, turning 90 degrees to a horizontal line at a value of 1 on the y-axis. The results here indicate that the algorithm has a slight tendency to misclassify outliers as inliers.

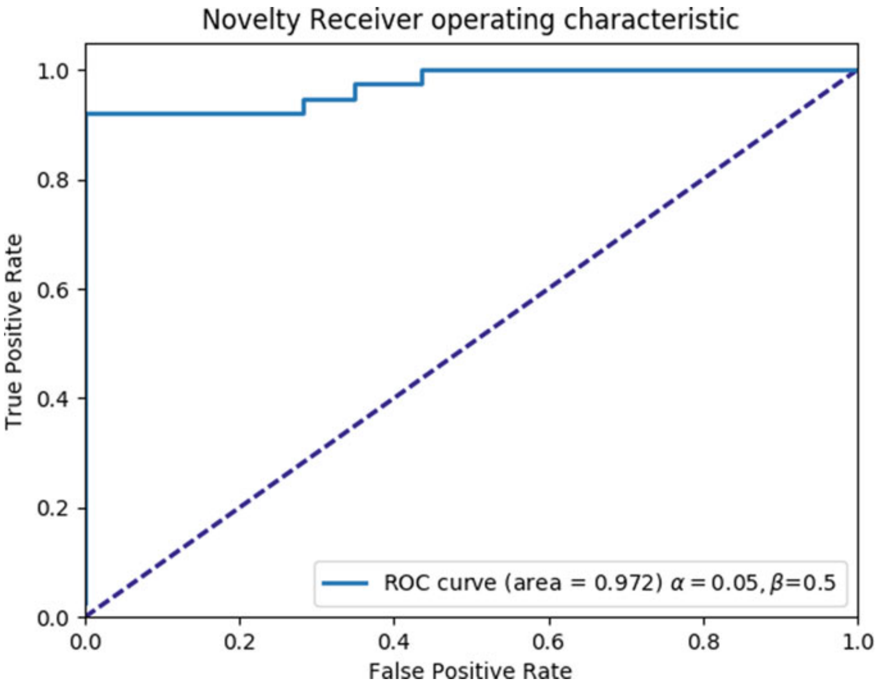


Fig. 3 Receiver operating characteristic (α = 0.05, β = 0.5, κ = 20)

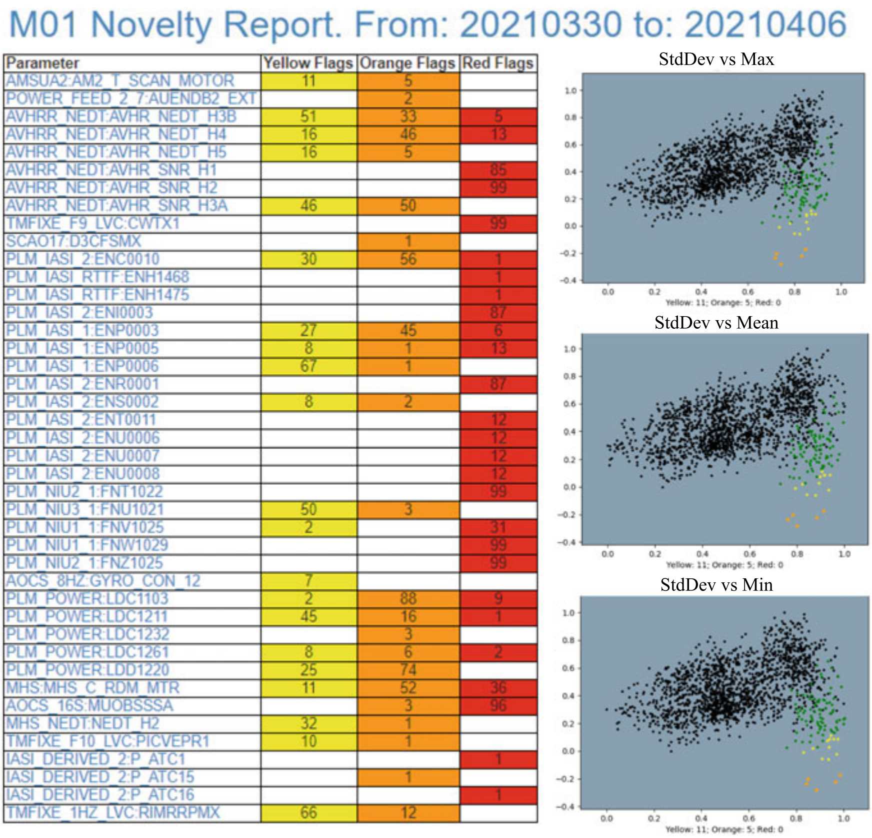
2.4 Beta Testing

Before investing significant resources into implementing Outlier Detection within the operational infrastructure at EUMETSAT, it was necessary to prove the Outlier Detection concept worked practically and identify any limitations and the areas for improvement, which are the main subject of this paper. This trial period involved running the algorithm regularly against real spacecraft telemetry in a beta-testing phase.

Firstly, a simple interface allowing engineers in the FCT to define and maintain the Nominal Dataset was established. Both the Nominal and Test Datasets were extracted from the back-end database of the Telemetry Archive, CHART. CHART also supports a front-end web GUI for plotting parameters. Each time the user requests a plot via this GUI, the data required for the plot is communicated to the server by updating the *http* link. The format of the link such that it is very easy to extract the spacecraft identifier, telemetry point, and time range. A text file of such links along with comments was therefore a suitable way to describe the Nominal Dataset – engineers could browse the archive using the web interface, and then simply copy and paste the links of what they considered nominal data into the text file. The script was therefore updated to read the definition of the Nominal Dataset from such a file. Some parameters may have multiple distinct clusters in the Nominal dataset which represent different ‘nominal’ behaviours, so the script was written in such a way that a single Nominal Dataset for a given parameter could be built from multiple links of the same parameter.

The next problem is how to alert users of outliers. Near Real Time analysis with push notifications or e-mail warnings each time an outlier was detected was considered too complex for the trial period, and could also risk flooding users with so many alarms that they would be simply filtered out and ignored. For this trial period, it was therefore decided to rely on static *HTML* reports. These reports were generated daily but used the previous week’s telemetry as the Test Dataset. This approach has the disadvantage that it requires the end-user to actively open the report; however, this is mitigated by the fact that each daily report covers the previous week, so reports only need to be checked every few days. To make the reports easier to check, they only included parameters on which outliers were detected (or more accurately where any of the reporting thresholds were breached for Test data points flagged Yellow, Orange or Red) and included a summary table which could be quickly reviewed.

An important aspect of Outlier Detection is presenting the results to users such that the “black box” effect is mitigated. Having developed the algorithm using the three 2-D projections (aka ‘splatter plots’) of the data shown in Fig. 4 there was already a readymade solution for this and the splatter plots became an integral part of the daily reports. For each parameter where outliers were reported, the daily reports also contained a CHART compatible http link to the Test Dataset. This allowed operators to quickly start investigations, and also to copy and paste the link into the text file defining the Nominal dataset if it was judged to be a new nominal behaviour.



M01 Novelty Report. From: 20210330 to: 20210406

Parameter	Yellow Flags	Orange Flags	Red Flags
AMSUA2:AM2_T_SCAN_MOTOR	11	5	
POWER_FEED_2_7:AUENDB2_EXT		2	
AVHRR_NEDT:AVHR_NEDT_H3B	51	33	5
AVHRR_NEDT:AVHR_NEDT_H4	16	46	13
AVHRR_NEDT:AVHR_NEDT_H5	16	5	
AVHRR_NEDT:AVHR_SNR_H1			85
AVHRR_NEDT:AVHR_SNR_H2			99
AVHRR_NEDT:AVHR_SNR_H3A	46	50	
TMFIXE_F9_LVC:CWTX1			99
SCAO17:D3CFSMX		1	
PLM_IASI_2:ENC0010	30	56	1
PLM_IASI_RTTF:ENH1468			1
PLM_IASI_RTTF:ENH1475			1
PLM_IASI_2:ENI0003			87
PLM_IASI_1:ENP0003	27	45	6
PLM_IASI_1:ENP0005	8	1	13
PLM_IASI_1:ENP0006	67	1	
PLM_IASI_2:ENR0001			87
PLM_IASI_2:ENS0002	8	2	
PLM_IASI_2:ENT0011			12
PLM_IASI_2:ENU0006			12
PLM_IASI_2:ENU0007			12
PLM_IASI_2:ENU0008			12
PLM_NIU2_1:FNT1022			99
PLM_NIU3_1:FNU1021	50	3	
PLM_NIU1_1:FNV1025	2		31
PLM_NIU1_1:FNW1029			99
PLM_NIU2_1:FNZ1025			99
AOCS_8HZ:GYRO_CON_12	7		
PLM_POWER:LDC1103	2	88	9
PLM_POWER:LDC1211	45	16	1
PLM_POWER:LDC1232		3	
PLM_POWER:LDC1261	8	6	2
PLM_POWER:LDD1220	25	74	
MHS:MHS_C_RDM_MTR	11	52	36
AOCS_16S:MUOBSSSA		3	96
MHS_NEDT:NEDT_H2	32	1	
TMFIXE_F10_LVC:PICVEPR1	10	1	
IASI_DERIVED_2:P_ATC1			1
IASI_DERIVED_2:P_ATC15		1	
IASI_DERIVED_2:P_ATC16			1
TMFIXE_1HZ_LVC:RIMRRPMX	66	12	

StdDev vs Max

StdDev vs Mean

StdDev vs Min

Fig. 4 Sample Outlier Detection Report with Details of First Parameter Shown [AMSUA2_AM2_T_SCAN_MOTOR]

2.5 Summary Lessons Learned from Development and Beta Testing

The most important lesson learned from Development and Beta testing could be summarised as ‘embrace the cult of the imperfect’ when it comes to the Outlier Detection algorithm: There will always be a better Outlier Detection algorithm on the horizon and the best will never arrive. In the meantime, there are many algorithm independent problems to be resolved before an algorithm has any practical use. This implies that as soon as an algorithm is performing adequately, it is preferable to begin working on the practical aspects. Continuing to focus on the underlying algorithm can lead to an endless cycle of dithering. The practical issues found are described below.

The "Black-Box" Trade-off.

Machine Learning algorithms such as Outlier Detection are known to suffer a "Black-Box" effect whereby it can be difficult to understand why a particular response is given to a set of inputs. The three 2-D projections used during the initial development of the algorithm (Fig. 4) are an effective way to allow end-users to understand the algorithm outputs. However, their effectiveness relies on the fact that there are only three of them. This implies that there is a trade-off – while adding a 5th dimension (such as frequency for example) may theoretically allow capturing of more anomalous behaviour, the number of possible projections increases from 6 to 10. Adding a 6th dimension increases the number of possible projections to 15, and so on. At some point, the obfuscation caused by too many projections will outweigh their potential benefits.

An alternative strategy would be to automatically select the 2-D projection which best highlights the reason for the classification, however this requires a plot for each Test data point as the optimal projection will be unique to each.

Alarm Management.

An important aspect of Outlier Detection is alarm management. During the beta-testing phase, it was found that floods of alarms were often received for either planned operations or single anomalies.

One problem was the redundancy in the monitored parameters. The parameters used for on-board limit monitoring were typically taken as a reference; however the objectives of on-board monitoring and Outlier detection are fundamentally different. On-board monitoring is used to trigger a very specific action, whereas Outlier Detection is used to alert the engineer responsible for a specific subsystem that there may be a problem worth investigating. The selection of parameters is discussed in more detail in Sect. 4.

Another problem causing floods of inappropriate alarms were planned operations, predictable geometric events (e.g. Moon entering the FOV of Instruments) and on board anomalies for which there were already explicit alarms. To solve this, it was necessary to implement conditional Outlier Detection, which is described in more detail in Sect. 3.

Rubbish in, Rubbish out.

Any semi-supervised machine-learning algorithm is only as good as the labelling of data in the Nominal dataset. For some systems, the labelling of Nominal Data will be a one-off activity, but for any system which is evolving through aging or development, then Outlier Detection requires continual maintenance of the Nominal Dataset to function. Even with the best intentions, it has been found that this activity tends to be done very irregularly. With the current set-up, engineers found it much easier to completely review the Nominal Dataset once or twice a year than continually maintain them.

This problem is currently under consideration with two possible solutions. Firstly, improving the ergonomics of Nominal Dataset can be achieved by creation of a dedicated GUI. A second option is a periodic autonomous update of the Nominal Datasets. Both this aspects are described in Sect. 6.

3 Influencer/Follower Concept for Conditional Outlier Detection

One of the main improvements recognised during the beta testing phase was the need for some sort of conditional Outlier Detection in order to prevent floods of alarms when a unit enters a non-nominal state for example. However, instead of explicitly identifying the circumstances when outlier detection should or should not be performed on every parameter, it was recognised that conditional outlier detection could be performed by categorising parameters as “influencers” or “followers”. Outlier detection would then only be performed on followers when all their respective influencers are exhibiting nominal behaviour, according to the same outlier detection algorithm.

3.1 Description of the Logic

An “influencer” is a parameter which may have a significant influence on a “follower” and it is set explicitly by Telecommand or autonomously by the on-board software. A typical example is the mode of an instrument, or the on/off status of some onboard equipment. Typically, outliers on influencers should not require reporting as they would anyway be covered by explicit monitoring via traditional OOLs etc.

A “follower”, on contrary, is a parameter under the predominance of an “influencer” – for example the internal current of an instrument or unit.

The “influencer”/ “follower” relationship can be one:one, one:many, many:one or many:many.

A practical example of one:many relationship, is represented in Fig. 5, which shows the internal currents of one of the instruments embarked on MetOp satellites, Microwave Humidity Sounder (MHS), w.r.t. the mode of the instrument.

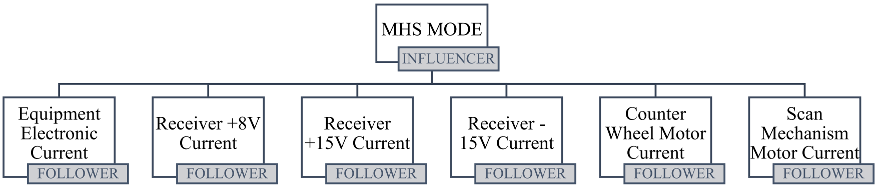


Fig. 5 Influencer (MHS MODE) followers (MHS currents) one:many relationship – MHS instrument

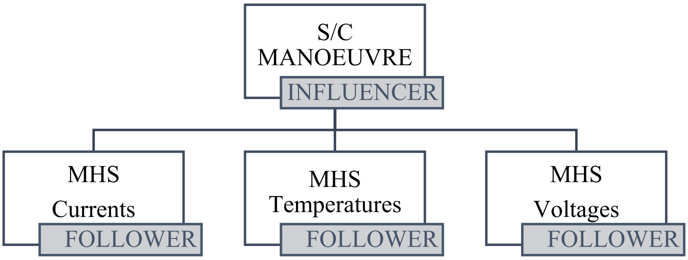


Fig. 6 Influencer (Spacecraft Manoeuvre) followers (MHS currents, Temperatures and Voltages) one:many relationship

In this case, outlier detection is only performed on the internal currents (followers) on those orbits where the MHS mode (influencer) test data is not an outlier. Crucially, this principle doesn't require the on-board equipment to be in a specific state, just the behaviour of its state to be nominal.

CHART also allows the user to define synthetic parameters without restriction. For example, the number of occurrences of astronomical events predicted by on-ground software -such as the intrusion of the Moon in the FOV of an instrument- as well as the completion of an on-board operation –such us a spacecraft manoeuvre or the temporary reconfiguration that follows the detection of an on-board anomaly (FDIR). Such synthetic parameters can be used as additional influencers in order to further suppress outlier detection when outliers can be expected. An example is provided in Fig. 6.

This approach has demonstrably reduced the number of irrelevant outliers polluting reports which greatly improves the overall usability.

3.2 Method

In order to reduce the number of alarms and false positives a multi-parametric interaction has been implemented, as presented in Fig. 7.

The overall logic can be divided in three steps:

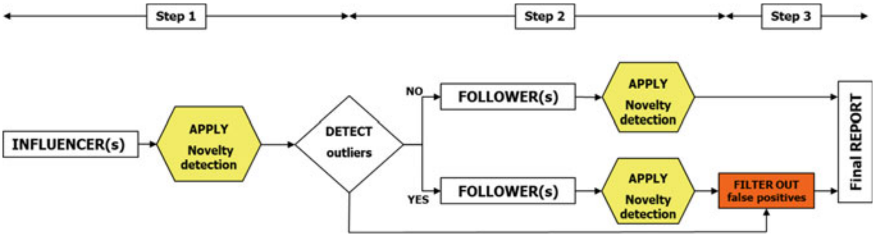


Fig. 7 Conditional Outliers filtering based on Influencers/Followers approach

- Step 1: the outlier detection is initially performed on all the influencers. For each parameter where outliers are identified a link is created in the daily report, as presented in Fig. 4.
- Step 2: the outlier detection is then performed systematically on all the followers, which represent the vast majority of the parameters to be analysed. In case all the influencers are nominal, the algorithm applies no additional filters (logical flow of the upper branch) and the final daily report is made available to the operator. However, if one or more influencers are flagged as non-nominal (logical flow of the lower branch in Fig. 7), an a posteriori condition is applied in order to filter out the Outliers identified on the followers for the same period.
- Step 3: the analysis is performed by the spacecraft expert, who can investigate all the Outliers listed in the final report. At this stage, ideally, none of the anomalies detected for any of the followers should take place during a period of time in which their corresponding influencers are non-nominal. In other words, no false positive should be observed and all the listed outliers should be genuine.

Such multi-parametric interaction has proven to streamline the Outlier detection process significantly, for both LEO and GEO missions. In particular, this approach has been successfully applied and the results are presented in the next paragraph.

3.3 A Practical Example (LEO Mission)

A special operation has been conducted on-board Metop-A in October 2019 and MHS instruments has been reconfigured multiple times in order to evaluate and optimise its performances by changing the velocity of the motor during the scanning phase. Figure 8 reports the profiles of the MHS internal currents.

As expected, each instrument reconfiguration (commanded from ground) is followed by a drop in the internal currents. An indiscriminate execution of the Outlier algorithm over this period would identify positives that shall not require investigation from the flight control team. However, by enabling the followers/influencers functionality, many false positives are filtered out and only the significant outliers are made available to the operator (Fig. 9).

The streamline of the process is clearly highlighted in Fig. 10, where the results of the two different approaches are compared.

It can be observed that:

- 3 parameters: do not appear in the Outlier report anymore
- 1 parameter: the number of orange Outliers has been reduced
- 3 parameters: the number of red Outliers has been reduced

Two parameters, the Scan Mechanism Motor Current (MHS:MHS_C_RDM_MTR) and the Counter Wheel Motor Current (MHS:MHS_C_FDM_MTR), are still included in the final report.

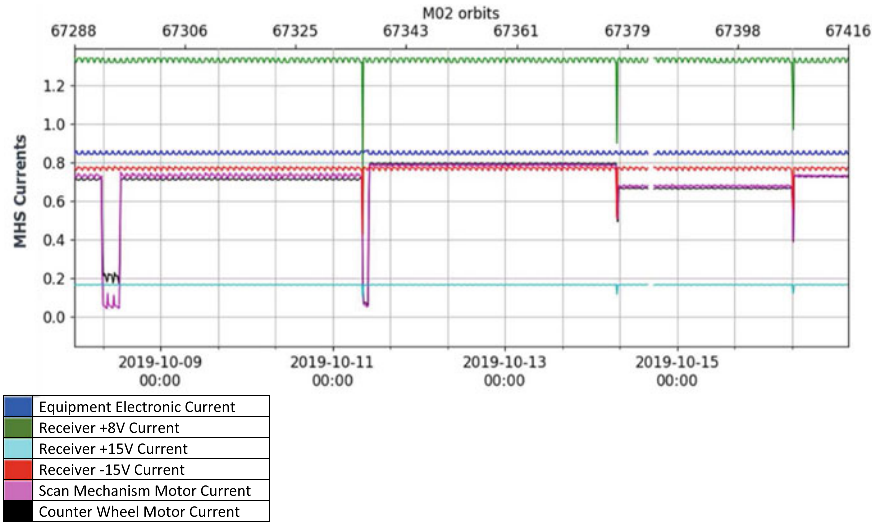


Fig. 8 MHS internal currents (followers) during a planned special operation

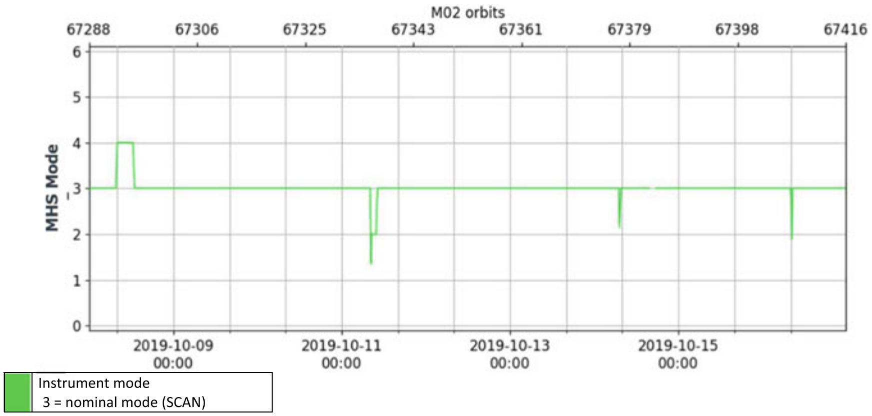
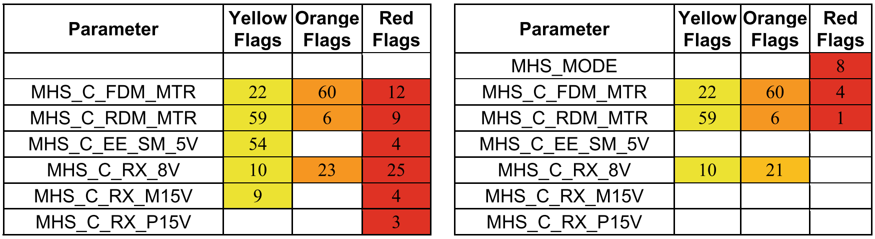


Fig. 9 MHS mode (influencer) during a planned special operation



Parameter	Yellow Flags	Orange Flags	Red Flags
MHS_C_FDM_MTR	22	60	12
MHS_C_RDM_MTR	59	6	9
MHS_C_EE_SM_5V	54		4
MHS_C_RX_8V	10	23	25
MHS_C_RX_M15V	9		4
MHS_C_RX_P15V			3

Parameter	Yellow Flags	Orange Flags	Red Flags
MHS_MODE			8
MHS_C_FDM_MTR	22	60	4
MHS_C_RDM_MTR	59	6	1
MHS_C_EE_SM_5V			
MHS_C_RX_8V	10	21	
MHS_C_RX_M15V			
MHS_C_RX_P15V			

Fig. 10 Outlier detection results without (left) and with (right) the conditional filtering

This is due to the fact that the filtering is only effective in the period of time during which the mode of the instrument is not nominal. The Fig. 9 shows clearly that, apart from some occasional reconfigurations, MHS mode is mostly nominal during the test. However, being the test aimed at evaluating the performances at different velocities of the motor, the two motor currents are consequently flagged as non-nominal.

4 Choice of Parameters

Another key aspect in making practical use of outlier detection is the selection of the relevant spacecraft parameters to be monitored. For a large amount of different observables, in fact, it is not possible to monitor all of them manually, which would require a human effort that is simply not an option. This fundamental problem can be simplified by applying the strategy currently in use at EUMETSAT. This analysis strongly focuses on the experiences of the FCT, but some general concepts can be identified and they can be easily applied to different spacecraft systems.

4.1 Description of the Logic

In order to efficiently identify the critical telemetry to be inspected, two main aspects must be highlighted:

– The maintenance workload: the nominal dataset of a parameter needs to be updated usually due to the aging of the on-board components or due to modification of the on-board configuration. Figure 11, in particular, shows the result of the aging of the GERB IOU through the MSG3 lifetime. It can be seen that the overall trend is the result of two distinct effects: a cumulative degradation due to the nominal use of the spacecraft components plus the eclipses, which are typical of each

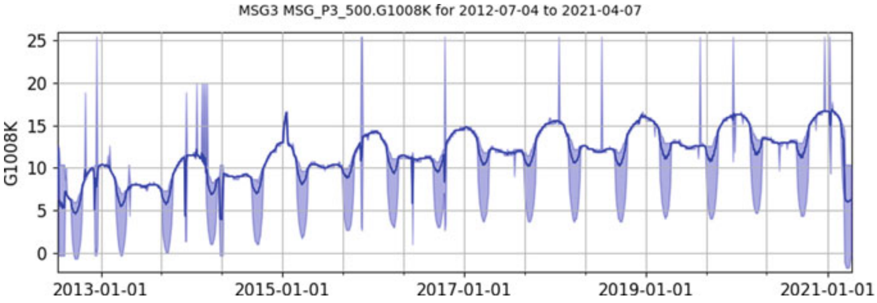


Fig. 11 Increasing temperature of a GERB baseplate temperature (G1008) [degC] over the MSG3 lifetime

geostationary mission. The user is responsible for re-evaluation of what is to be considered nominal based on both his experience and expertise.

- The parameters' hierarchy: based on operational experience, it is generally possible to identify a hierarchy of the "System > Subsystem > Component" kind among the spacecraft parameters. A "System Level" parameter could for example be the Main Bus Current, then a "Unit Level" parameter could be the current that from the main bus reaches an instrument and then a "Component Level" parameter could be the current that reaches one of the different components of the instrument.

The two aspects are strictly related. From one side, it would be very attractive to set a Outlier analysis over as many spacecraft parameters as possible with the lowest hierarchy such to cover all the possible sources of Outliers. However, this would clearly have a huge and impractical maintenance demand.

On the other hand, decreasing the number of significant observables by using only the higher parameters in the hierarchy would also be an insufficient approach.

The optimal choice was found to be a balance of the two extreme aforementioned choices. As shown in Fig. 12, a medium amount of parameters, which are significant at Subsystem level, would not have an excessive maintenance demand and, at the same time, would allow a full coverage of all the spacecraft subsystems.

The solution described provides several advantages. First of all, the usefulness of the detection is not lost as it is clear where the problem lies within the *Subsystem*; therefore, the monitoring is not expected to trigger very often and the subsequent investigation is aimed at evaluating the performances of a limited set of *Components*.

Moreover, the number of parameters to maintain is optimized. The team uses the outlier detection tool to monitor approximately 200 parameters on each of the MetOp satellites.

4.2 A Practical Example (GEO Mission)

A lesson learned from operating the geostationary satellites can be used to practically show the effectiveness of the strategy described in Sect. 4.1.

The generic problem consists in the monitoring of the MSG's primary instrument SEVIRI.

The first step is a systematic identification of a set of hierarchies "*System* > *Subsystem* > *Component*", which for the MSG3 satellite is shown in Fig. 13.

The second step is the selection of a parameter which can be thought of as its position within the hierarchy of the MSG satellite data. In this case, the significant parameter which is neither not too high nor too low in the hierarchy is the instrument current.

It must be underlined that the in-flight experience gained so far within the FCT plays a fundamental role in making an efficient choice of the satellite data, carefully evaluating its position within the hierarchy. But an optimal trade-off can be finally

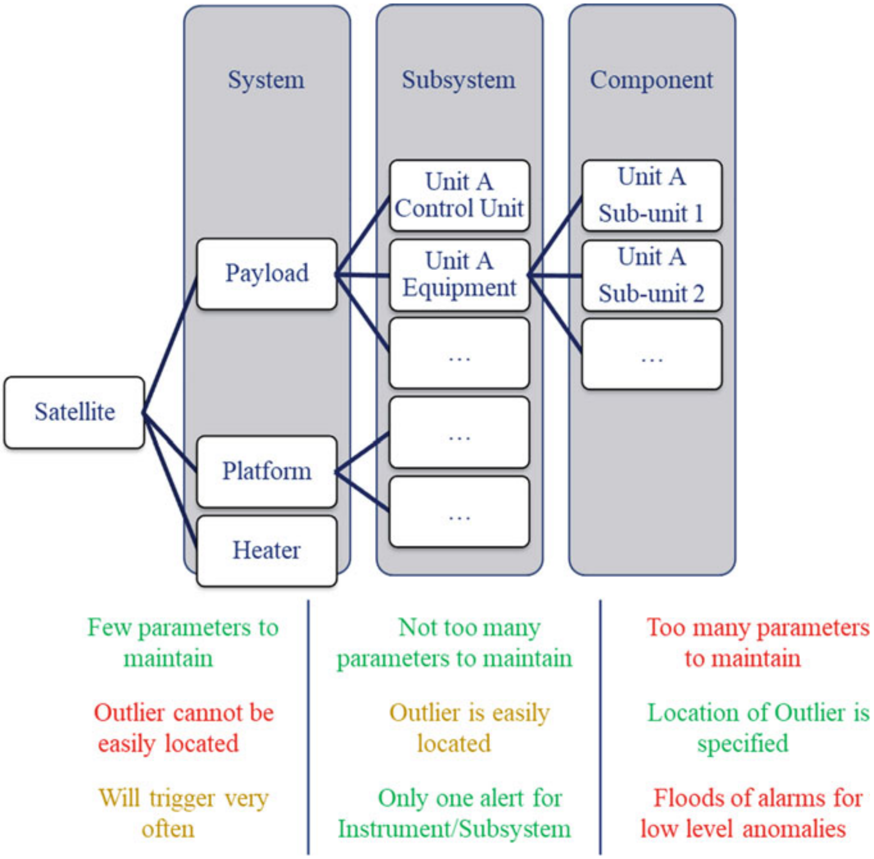

System
Subsystem
Component
Satellite
Payload
Platform
Heater
Unit A Control Unit
Unit A Equipment
…
…
…
Unit A Sub-unit 1
Unit A Sub-unit 2
…
Few parameters to maintain
Outlier cannot be easily located
Will trigger very often
Not too many parameters to maintain
Outlier is easily located
Only one alert for Instrument/Subsystem
Too many parameters to maintain
Location of Outlier is specified
Floods of alarms for low level anomalies

Fig. 12 Pros and Cons of selecting different parameters for monitoring by outlier detection

obtained between the usefulness of the Outlier detection and the effort spent in the maintenance.

5 Selection and Maintenance of the Nominal Data Set

A key factor for the functioning of the algorithm is the choice of the nominal dataset according to which the evaluation of all the properties for the Outliers detection is done. As anticipated already in Sect. 4, for a long mission lifetime all the on board components are inevitably affected by aging, thus the Nominal data need to be periodically updated. This task shall be done efficiently and accurately. A clear example of why the data set need to be maintained is visible from Fig. 14: after more than 10 years in orbit, the mean value of some temperatures changed quite

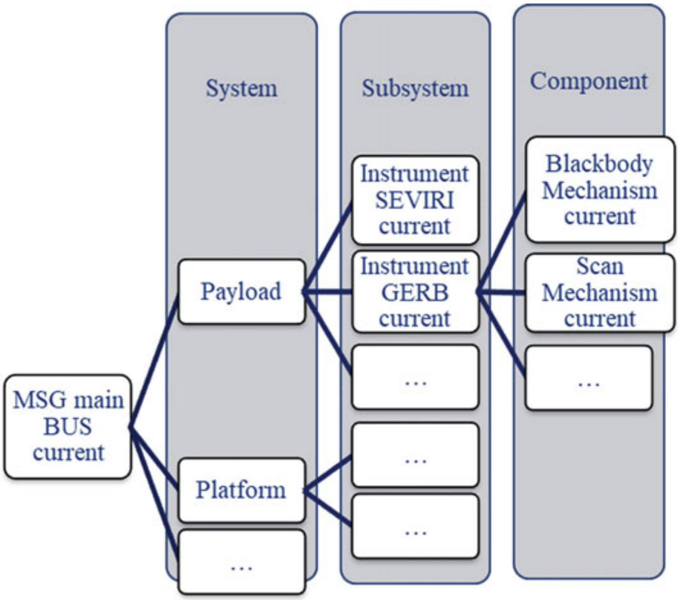


Fig. 13 *System* > *Subsystem* > Component selection w.r.t. SEVIRI instrument on-board MSG3 satellite

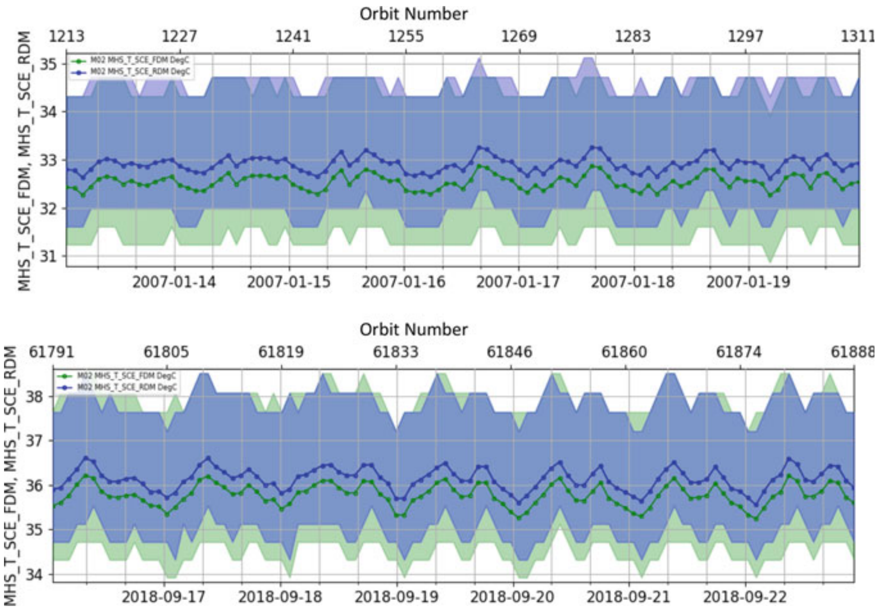


Fig. 14 Example of a nominal aging of a component (MHS internal temperatures on-board Metop-A)

significantly. Obviously, keeping always the same Nominal data would lead to a complete incorrect analysis.

The FCT teams at EUMETSAT can take advantage from the existing CHART framework, which allows engineers to retrieve, process and analyse the spacecraft telemetry; this includes the calculation of orbital statistics, which can subsequently be run through the outlier detection algorithm [3].

5.1 Current Nominal Dataset Maintenance Process

As already described in Sect. 2.3, the Nominal Dataset maintenance process takes advantage of the fact that the CHART front-end web GUI creates links for any plot which contain the spacecraft identifier, telemetry point, and time range. The engineer responsible for the Nominal Dataset maintenance can then simply use the GUI to browse for nominal data and then copy and paste the links of what it is considered nominal data into a master.csv file, noting that the nominal dataset may comprise several non-contiguous links. An embedded Macro is used to make the data compatible with the CHART network. The final step is to transfer the.csv file to the CHART server, so that the daily scheduler can read the new input at the time of its execution. Considering that there are 3.csv files per each spacecraft (*follower*.csv, *influencers*.csv and *follower-influencer-logic*.csv), and considering that the overall process is semiautomatic, the effort spent to generate a new set of input files is quantifiable in about one hour. Based on the experience gained so far, the EPS FCT tends to update these files once a month (3 spacecraft, about 200 parameters per spacecraft). The current approach is then considered a significant improvement w.r.t the initial Beta Testing layout described in Sect. 2, but is also an area where more work is needed to further improve the full process. Ideas to.

5.2 Lesson Learned from GEO Mission

A vast majority of analogue parameters–in particular currents and temperatures- on-board a geostationary satellite are characterized by significant changes during the operational lifetime at least for two reasons:

– Yearly periodic variations, due to the revolution of the Earth around the Sun;
– Two eclipse seasons every year around the equinoxes, due to the geometry of the ecliptic.

An example is presented in Fig. 15, where the solar array current (P1349C) is plotted over 2 years of in-flight operational lifetime of MSG1 satellite.

The choice of a Nominal dataset in this case is not trivial if we want to minimise the effort spent in the maintenance of the parameter dataset.

Three options were identified:

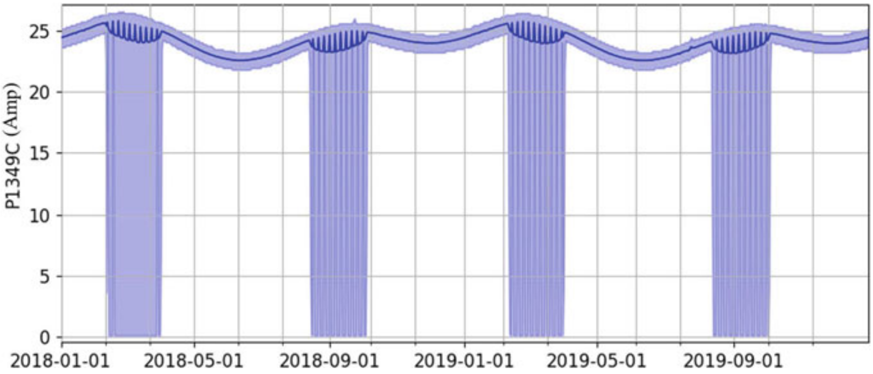


Fig. 15 Variation of the solar array current (P1349C) on-board MSG1from January 2018 to January 2020

- Exclude eclipses from analysis both in the nominal and test dataset. This option was discarded since the eclipse seasons comprehends a high percentage of the year;
- Diversify eclipses from the rest of the year both in nominal and test dataset. This is an attractive alternative that would comprehend a conditional Outlier analysis performed with different nominal datasets. It is quite a complex solution and its feasibility is under evaluation;
- Include eclipses as a nominal dataset in the inputs without applying any diversification. This is the current strategy adopted but it has to be considered as a temporary trade-off. In this scenario, the nominal dataset will contain wider variations and it will be harder to identify Outliers.

By selecting the third option, the overall Outlier detection is theoretically less sensible. However, there are no significant data to quantify this estimate and no simulations have been conducted in this regard. The Outliers identified so far in a validation environment have always indicated a non-nominal behaviour of the components on-board.

In conclusion, even if the theoretical considerations remain valid, the practical experience has proven this approach effective so far. In perspective, the FCT is going to assess the feasibility and the cost–benefit effort of the implementation of the second option.

5.3 Future Nominal Dataset Maintenance Improvements

It is recognised that timely and accurate maintenance of the Nominal Datasets is the most crucial aspect of Outlier Detection which still needs to be addressed. Without continual maintenance of Nominal Datasets, the Outlier Reports become unusable due to the background noise caused by irrelevant outliers. So, the process needs to be as efficient and intuitive as possible. Two different (but not mutually exclusive) strategies are being considered for this.

Automated Nominal Dataset Maintenance.

One of the key problems which remains is updating Nominal Datasets in reaction to slower seasonal or aging evolutions – if this is not done, then the Outlier reports start filling up due to slow evolutions which mask real problems. An automated process to achieve this is currently being considered. Every fourth week, any parameters with outliers would be used as an input. If a parameter has more than about 30% of its Test Dataset flagged as outliers, three trial runs would be performed. Firstly, the 3rd previous week's data is added to the Nominal Dataset and Outlier Detection is re-run. The 3rd previous week is then removed and the 2nd previous week added. Finally the 2nd previous week is removed and the previous week added. If all three trial runs significantly reduce the number of outliers, then it can be assumed that the parameter in question is undergoing a trend to which the responsible Engineer has already been alerted. These previous three weeks of data can then be automatically added to the Nominal dataset. However, before doing so, a check needs to be performed to verify that the new Nominal Data is self- consistent. To do this, outlier detection would be performed on each new Nominal Datapoint using the remaining new Nominal Datapoints as a Nominal Dataset – with a slow trend, each new Nominal Datapoint should be an inlier with respect to the other new Nominal Datapoints.

Since this process adds X new Samples to the Nominal Dataset, it is a good idea to remove X Samples. To do this, the newly added samples could be used as a Nominal Dataset and the original Nominal Dataset becomes a Test Dataset, from which the X largest outliers are identified and removed.

Graphical User Interface.

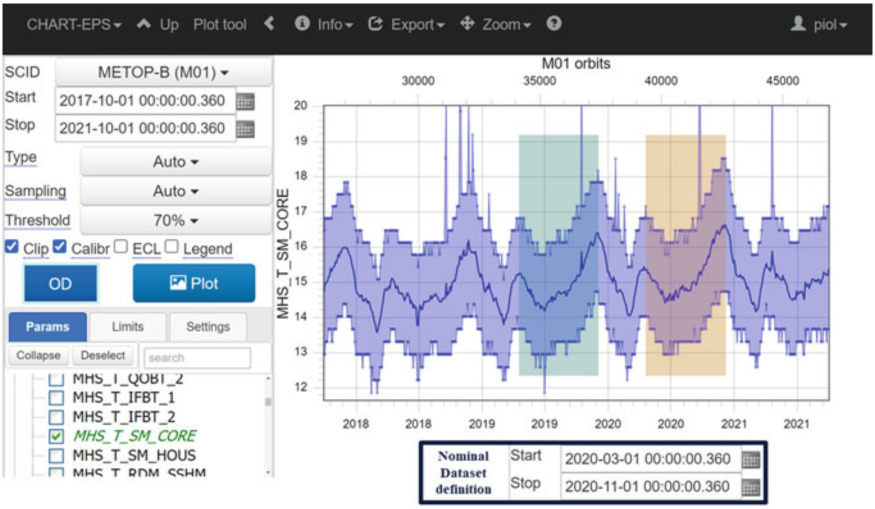
Even with some automated Nominal Dataset Maintenance, visualising and manually defining/adjusting the Nominal Datasets is still necessary. This implies a dedicated GUI to provide an intuitive way to visualise and adjust Nominal Datasets.

Such a GUI would simply provide a time series plot of each parameter, and a way to highlight and adjust the Nominal Dataset by use of shaded areas which can be easily added, removed, moved and resized.

This type of tool is currently under development and some of the functionalities are illustrated in Fig. 16. Ideally, once selected the parameter of interest from the PARAMS list (MHS_T_SM_CORE in the example an MHS internal temperature), the button OD (Outlier Detection) can be clicked to enable the Nominal Dataset definition.

The user can visualise the Nominal Dataset currently in use by simply looking at the highlighted window that appears automatically in the plot tool. In the Fig. 16, this window is marked in green.

If an update is necessary, then the Nominal Dataset Start/Stop time window can be used for a precise selection of the new area; alternatively, a click and drag option is also possible.



SCID METOP-B (M01)
Start 2017-10-01 00:00:00.360
Stop 2021-10-01 00:00:00.360
Type Auto
Sampling Auto
Threshold 70%
☑ Clip ☑ Calibr □ ECL □ Legend

□ MHS_T_QOBT_2
□ MHS_T_IFBT_1
□ MHS_T_IFBT_2
☑ MHS_T_SM_CORE
□ MHS_T_SM_HOUS
□ MHS_T_RDM_SSHM

Nominal Dataset definition
Start 2020-03-01 00:00:00.360
Stop 2020-11-01 00:00:00.360

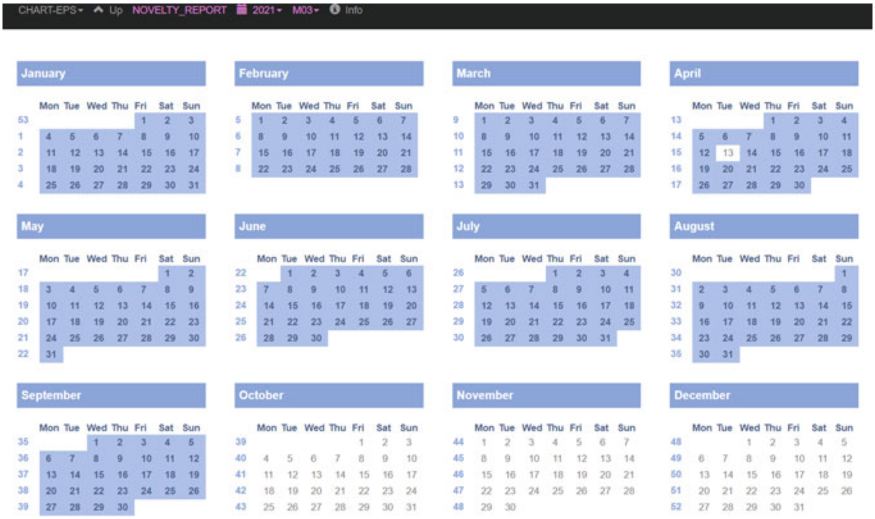
Fig. 16 Outline of nominal dataset maintenance GUI. A time series plot of the selected parameter is shown in blue (MHS_T_SM_CORE). The old Nominal Data is highlighted with green shading and this area would be configurable using either Start/Stop box at the bottom or a click and drag option. The new Nominal Data is highlighted with orange shading instead

Then the new highlighted window, which marks in orange the Nominal Dataset as valid, is automatically displayed and the user receives an immediate visual feedback of the action performed.

At that point, the user can simply acknowledge the changes updating the Nominal Dataset and he can execute the Outlier Detection algorithm with the new inputs. An important aspect to consider for future reference is to give to the user the possibility to select more non-consecutive Nominal Datasets.

As described in Sect. 2, all the results are organised in a.html reports which cover the last seven days of operational data. These files are archived on a daily basis and are made available via CHART reporting tool. In Fig. 17 it is shown the reporting tool organised in a calendar: different spacecraft IDs (M01, M02 or M03) and different calendar years can be selected. The day at which the successful processing takes place is greyed out, otherwise it remains transparent. More in particular, in the example provided all the reports from January 1st to September 30th 2021 have been correctly generated, with the only exception of April 13th.

This approach results in a systematic organisation of the detected outliers; however, this method does not allow the user to easily retrieve the historical data. For this reason, the last functionality that is going to be implemented is the collocation of the results in a dedicated database. It is believed that this type of enhancement will permit an efficient accessibility of the processed data. Eventually, also this functionality will be made available via the GUI.



January

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
53					1	2	3
1	4	5	6	7	8	9	10
2	11	12	13	14	15	16	17
3	18	19	20	21	22	23	24
4	25	26	27	28	29	30	31

February

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
5	1	2	3	4	5	6	7
6	8	9	10	11	12	13	14
7	15	16	17	18	19	20	21
8	22	23	24	25	26	27	28

March

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
9	1	2	3	4	5	6	7
10	8	9	10	11	12	13	14
11	15	16	17	18	19	20	21
12	22	23	24	25	26	27	28
13	29	30	31				

April

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
13				1	2	3	4
14	5	6	7	8	9	10	11
15	12	13	14	15	16	17	18
16	19	20	21	22	23	24	25
17	26	27	28	29	30		

May

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
17						1	2
18	3	4	5	6	7	8	9
19	10	11	12	13	14	15	16
20	17	18	19	20	21	22	23
21	24	25	26	27	28	29	30
22	31						

June

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
22		1	2	3	4	5	6
23	7	8	9	10	11	12	13
24	14	15	16	17	18	19	20
25	21	22	23	24	25	26	27
26	28	29	30				

July

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
26				1	2	3	4
27	5	6	7	8	9	10	11
28	12	13	14	15	16	17	18
29	19	20	21	22	23	24	25
30	26	27	28	29	30	31	

August

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
30							1
31	2	3	4	5	6	7	8
32	9	10	11	12	13	14	15
33	16	17	18	19	20	21	22
34	23	24	25	26	27	28	29
35	30	31					

September

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
35			1	2	3	4	5
36	6	7	8	9	10	11	12
37	13	14	15	16	17	18	19
38	20	21	22	23	24	25	26
39	27	28	29	30			

October

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
39					1	2	3
40	4	5	6	7	8	9	10
41	11	12	13	14	15	16	17
42	18	19	20	21	22	23	24
43	25	26	27	28	29	30	31

November

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
44	1	2	3	4	5	6	7
45	8	9	10	11	12	13	14
46	15	16	17	18	19	20	21
47	22	23	24	25	26	27	28
48	29	30					

December

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
48			1	2	3	4	5
49	6	7	8	9	10	11	12
50	13	14	15	16	17	18	19
51	20	21	22	23	24	25	26
52	27	28	29	30	31		

Fig. 17 Organisation of the outlier detection results for Metop-C from 01/01/2021 to 30/09/2021. Each calendar day is by default transparent. In case the Outlier Detection algorithm report is generated nominally, the day is highlighted with a shading

6 Conclusion

This paper contributes to the implementation of an outlier detection algorithms within a generic operational context. Even though the description of some practical problems and some examples are specific to the operational systems in use at EUMESAT, the analysis offers a set of generic recommendations and guidelines of how to implement this type of solution in any other operational context. The authors believe that the outlier detection algorithms are extremely useful in identifying anomalies before they develop into serious problems on-board; however, compared with developing this or different semi-supervised algorithms, tackling the problems described in this paper has proven to be a much better benefit return on any time and effort invested.

The contributions of this paper are summarized as follows. Independently of the algorithm, three major practical problems need to be addressed:

- The need to minimise cascades of Outliers. In this regard, a influencer/follower concept has been introduced and its effectiveness has been demonstrated;
- The need to think carefully about which parameters are being assessed. In this regard, an effective strategy is based on the System > Subsystem > Component hierarchy has been presented;
- The need to streamline the Nominal Dataset maintenance. The approach introduced has proven to be effective, since the user can easily retrieve the historical data and label the nominal dataset as such. However, the current solution is only

partially automated and the need of a convenient GUI and also a possible automated maintenance of Nominal Data to account for evolving trends has been identified.

These problems are inherent to any semi-supervised algorithm, meaning their resolution will not be invalidated by any subsequent development of the algorithm, or migration to a different one.

Acknowledgements The authors take this opportunity to thank Mike Elson and the other members of the EUMETSAT TSS team for the excellent work spent in developing the CHART toolset in these years and for his patience supporting the younger members of the FCT.

References

1. Martínez-Heras J, Donati A, Kirsch MGF, Schmidt F (2012) New telemetry monitoring paradigm with novelty detection. In: SpaceOps Conference, AIAA, Stockholm, Sweden, pp 11–15.https://doi.org/10.2514/6.2012-1275123
2. O'Meara C, Schlagy L, Faltenbacher L, Wicklerz M (2016) ATHMoS: automated telemetry health monitoring system at GSOC using outlier detection and supervised machine learning. In:14th International conference on space operations. https://doi.org/10.2514/6.2016-2347
3. Trollope E, Dyer R, Francisco S, Miller J, Griso MP, Argemandy A (2018) Analysis of automated techniques for routine monitoring and contingency detection of in-flight LEO operations at EUMETSAT". In: SpaceOps Conference 2018, AIAA, Marseille, France, 2532p. https://doi.org/10.2514/6.2018-2532
4. Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. In: Conference: proceedings of the 2000 ACM SIGMOD international conference on management of data, 16–18 May 2000, Dallas, Texas, USA
5. Martínez-Heras J, Boumghar R, Donati A (2016) Log novelty detection system. In: SpaceOps conference 2016, AIAA, Daejeon, Korea, 2432p. https://doi.org/10.2514/6.2016-2432
6. Iverson DL (2008) System health monitoring for space mission operations. In: 2008 IEEE aerospace conference, IEEE, Big Sky, USA, pp 1–8. https://doi.org/10.1109/AERO.2008.4526646
7. Schulster J, Evill R, Rogissart J, Phillips S, Dyer R, Feldmann N (2018) Charting the future—an offline data analysis & reporting toolkit to support automated decision-making in flight operations. In: SpaceOps conference 2018 AIAA Marseille, France https://doi.org/10.2514/6.2018-2637
8. Fuertes S, Pilaster B, D'Escrivan S (2018) Performance assessment of NOSTRADAMUS & other machine learning-based telemetry monitoring systems on a spacecraft anomalies database. In: 15th International conference on space operations, May 2018

Multivariate Anomaly Detection in Discrete and Continuous Telemetry Signals Using a Sparse Decomposition into a Dictionary

Pierre-Baptiste Lambert, Barbara Pilastre, Jean-Yves Tourneret, Loïc Boussouf, Stéphane d'Escrivan, and Pauline Delande

Abstract This paper presents some research works based on the PhD thesis of B. Pilastre (B. Pilastre, Estimation Parcimonieuse et Apprentissage de Dictionnaires pour la détection d'Anomalies Multivariées dans des Données Mixtes de Télémesure Satellite, PhD Thesis of the university of Toulouse, Nov. 6, 2020.), supported by CNES and Airbus Defence & Space, on a new Anomaly Detection algorithm based on a sparse decomposition into a DICTionary (ADDICT). The proposed method addresses two main challenges related to anomaly detection for satellite telemetry parameters, namely the multivariate processing of these parameters and the mixed continuous and discrete nature of the data. Different variations of the ADDICT algorithm, referred to as C-ADDICT and W-ADDICT, have been investigated differing by the data decomposition term defined using a linear combination of the atoms or its convolutional equivalent. The resulting ADDICT, C-ADDICT and W-ADDICT algorithms have been evaluated on a small representative dataset containing satellite anomalies with an available ground-truth and have shown competitive results with respect to the state-of-the-art. They have also been tested on industrial use-cases, especially regarding online processing (i.e., sequential learning taking into account the feedback of users). The results of these tests are presented in this paper.

P.-B. Lambert (✉) · S. d'Escrivan · P. Delande
Centre National d'Etudes Spatiales (CNES), 18 avenue Edouard Belin, 31400 Toulouse, France
e-mail: pierre-baptiste.lambert@cnes.fr

B. Pilastre · J.-Y. Tourneret
TeSA, 7 Boulevard de la Gare, 31500 Toulouse, France
e-mail: barbara.pilastre@tesa.prd.fr

J.-Y. Tourneret
e-mail: jean-yves.tourneret@enseeiht.fr

J.-Y. Tourneret
Université de Toulouse, IRIT-INP/ENSEEIHT, 2 rue Charles Camichel, 31000 Toulouse, France

L. Boussouf
Airbus Defence and Space, 31 rue des Cosmonautes, 31400 Toulouse, France
e-mail: loic.boussouf@airbus.com

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_12

Keywords Anomaly Detection · Spacecraft Telemetry Monitoring · Sparse Representation · Dictionary Learning

Acronyms/Abbreviations

AD	Anomaly Detection
OOL	Out-Of-Limit
ML	Machine Learning
ADDICT	Anomaly Detection based on a space decomposition into a DICTionary
ROC	Receiver Operating Characteristic

1 Introduction

A main issue in space operations is to monitor housekeeping telemetry and detect potential anomalies as soon as possible. Given the number of satellite sensors (hundreds to thousands), this task needs to be automatically performed using anomaly detection (AD) methods. As traditional monitoring methods such as the Out-Of-Limit (OOL) rule have shown some limits, Machine Learning (ML) methods are being more and more used to improve housekeeping telemetry monitoring through semi-supervised learning: telemetry describing nominal behaviors of spacecraft (without anomalies) is used to build a reference model, which is then compared to newly acquired telemetry (that might include anomalies) to be able to detect never-seen-before abnormal behaviors. Recent methods from the literature have significantly improved the monitoring of the satellite equipment health. However, they are mostly univariate, i.e., they handle each telemetry parameter independently from the others. It is important to consider a multivariate framework in order to take into account possible correlations between telemetry parameters and detect anomalies associated with changes in these correlations, referred to as multivariate or contextual anomalies. This paper studies different AD algorithms that allow multivariate anomalies to be detected.

The paper is organized as follows. Section 2 presents the theory of sparse decompositions into a dictionary and the associated algorithms ADDICT, C-ADDICT and W-ADDICT. Section 3 evaluates the performance of these algorithms on a representative dataset composed of real and simulated spacecraft telemetry in comparison to three state-of-the-art methods. The algorithms are then tested on industrial use-cases brought by CNES and ADS. Conclusions and future works are reported in Sect. 4.

2 The ADDICT Algorithm

2.1 Benefits of a Sparse Decomposition

The ADDICT algorithm stands for "Anomaly Detection Using a DICTionary". It implements the general frame of sparse decompositions presented for univariate AD in [1], and is adapted to the processing of satellite telemetry and multivariate AD with mixed discrete/continuous signals. It was investigated in a PhD thesis [2] following lessons learnt in AD. The objective was to overcome the following concerns:

- The algorithm has to be able to detect multivariate anomalies
- The algorithm has to be able to reduce as much as possible the false alarm rate
- The algorithm has to be able to update knowledge acquired in AD, i.e., to introduce a new normal behaviour resulting from user feedback.

Sparse decompositions attempt at recomposing an input signal as a sparse combination of elementary signals contained in a dictionary. Transforming a signal into another representation space and comparing this transformed signal to normal instances to determine a probability of anomaly occurrence is a common approach for AD. This approach was for instance considered in [3] using functional data decompositions, or in [4] using neural networks autoencoders. While these techniques are able to reconstruct a signal as close as possible to a given input, sparse decompositions allow us to balance reconstruction effort and prefer discarding reconstructions defined by too many dictionary elements using sparsity constraints. These sparse decompositions yield very different signal reconstructions for normal and abnormal signals, allowing the number of false alarms to be reduced significantly.

This paper first presents an overview of ADDICT and its extensions. Additional details can be found in [5].

2.2 Preprocessing

To simplify matricial representations and computations of sparse decompositions, ADDICT requires a data formatting that can be decomposed into two steps: segmentation and vectorization. Each telemetry time series is segmented into windows of a given size w, with the same sampling timestamp and period. The window size required for the segmentation has to be set based on the time granularity one wants to implement for AD. All windows from the different sensors are then concatenated into a vector, referred to as atom, which represents the same multivariate context along discrete and continuous signals.

During the second step of the preprocessing (vectorization), discrete and continuous signals are gathered into the same vector, which will be processed sequentially. Without loss of generality, we assume that discrete components are located at the beginning of the vector, whereas continuous components are included at the end of

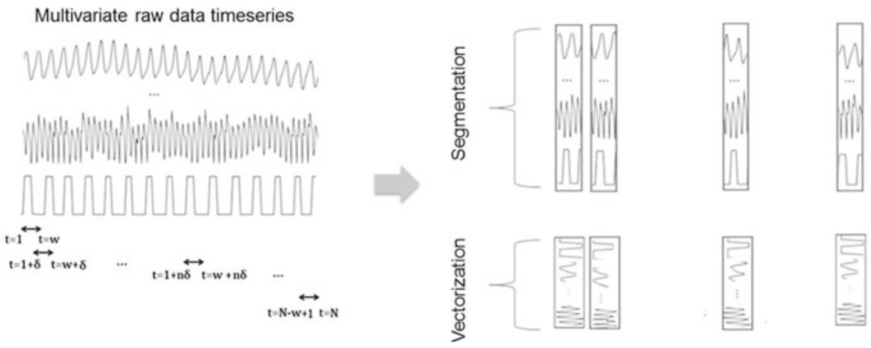


Fig. 1 Preprocessing of raw telemetry data using segmentation and vectorization, leading to a matrix representation that will be used for a sparse decomposition of test signals

this vector. Vectors obtained using normal telemetry time series will be used to build a telemetry dictionary.

The segmentation and vectorization operations are illustrated in Fig. 1 *(with t the timestamp of the telemetry sample, which varies between 1 and N, w the windows size, δ the time step between time windows).*

Note that in the initial version of ADDICT, defining overlapping windows is recommended to build the dictionary (with a fixed time step δ), to make sure binary status changes are properly captured in the preprocessing. Conversely, in the convolutional version of ADDICT, this overlapping is no longer required.

This pre-processing is conducted for any test signal whose anomalies need to be detected and also for dictionary learning purposes. In this paper, the dictionary will consist of the union of segmented/vectorised signals representing nominal behaviours, all stored column-wise.

2.3 Anomaly Detection Using a Sparse Decomposition

2.3.1 Problem Formulation

The sparse decomposition problem can be defined using the decomposition illustrated in Fig. 2, where:

- y is the test signal to be assessed as anomaly or nominal, resulting from the pre-processing presented before,
- Φ is the dictionary, defined as a block diagonal matrix, whose upper-left block contains the discrete components of the dictionary, and whose lower-right block is made of the continuous components. Both blocks have the same number of columns, and the i-th columns in both blocks correspond to the same multivariate context,

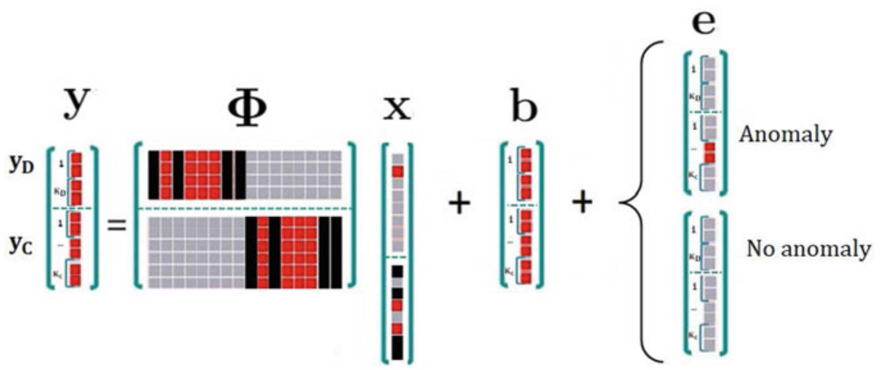


Fig. 2 Sparse decomposition on a dictionary, decomposing a signal as a sparse combination of dictionary atoms corrupted by an additive noise b and a potential anomaly vector e. In this figure, grey cells indicate zeroes, red cells are used for non-zero values, and black cells are atoms discarded for reconstruction (see 2.3.2)

- x is a sparse vector of coefficients, selecting in Φ the atoms used for the reconstruction of y,
- e is an anomaly vector subjected to a sparsity constraint,
- b is an additive noise vector, which is defined as a residual not subjected to any sparsity constraint, able to model the difference between y and the decomposition $\Phi\mathrm{x} + \mathrm{e}$.

The joint determination of the sparse vector x and the anomaly vector e from the observed vector y is the core function of ADDICT. This determination can be done using a two-step approach handling discrete and continuous signals differently. More precisely, the sparse decomposition of ADDICT decomposes y into a sparse combination of dictionary atoms and a noise vector b, with a potential block sparse anomaly vector e, as explained below.

2.3.2 Problem Resolution

As mentioned before, the discrete and continuous components of the observed signal y are handled sequentially. More precisely, the observed signal y is split into two blocks containing its discrete and continuous values as follows:

$$\begin{bmatrix} y_D \\ y_C \end{bmatrix} = \begin{bmatrix} \Phi_\mathrm{D} & 0 \\ 0 & \Phi_\mathrm{C} \end{bmatrix} \begin{bmatrix} x_D \\ x_C \end{bmatrix} + \begin{bmatrix} e_D \\ e_C \end{bmatrix} + \begin{bmatrix} b_D \\ b_C \end{bmatrix} \tag{1}$$

where y_D,x_D, e_D and b_D contain the discrete parts of y, x, e and b (with similar definitions of y_C,x_C, e_C and b_C for the continuous parts of the vectors). The first step of the problem resolution considers a decomposition along the discrete components, i.e., specific discrete atoms are selected that enable a reconstruction with discrete components of the observed signal y_D. In a second step, the continuous components of the selected discrete atoms are used for the continuous reconstruction. Both reconstructions contribute to the noise and anomaly vectors using the discrete and continuous blocks. They are described with more details in the next paragraphs.

Discrete decomposition

A decomposition of the N_D discrete observations along the discrete components of the dictionary is performed first. This is done using the following minimization problem:

$$\arg \min_{x_D \in \beta, e_D \in \mathbb{R}^{\mathrm{N_D}}} \|y_D - \Phi_{\mathrm{D}} x_D - e_D\|_2^2 + b_D \sum_{k=1}^{K_D} \|e_{D,k}\|_2 \tag{2}$$

Note that the block sparsity of the discrete anomaly vector e_D is imposed by the right hand side term and that the different terms are defined as follows:

- $\|e_{D,k}\|_2$ is the Euclidean norm of $e_{D,k}$, which is associated with the kth discrete parameter,
- b_D is a hyperparameter controlling the sparsity of e_D, which is ensured by the last term, reflecting the fact that anomalies are rare and affect few parameters at the same time,
- The sparsity on x_D is controlled by imposing that x_D belongs to the basis $\mathcal{B}$, containing vectors composed of zeros except one value.

This minimization detailed in [5] results in identifying discrete atoms from the dictionary that are the closest to the input signal. If no element from the dictionary is found sufficiently close from y_D, an anomaly affecting the discrete part of the input signal is declared. In the other case, possible discrete atoms are identified and the corresponding continuous atoms are extracted from the dictionary. This subset of the dictionary will be used for the continuous decomposition of the observed signal. This subset of the dictionary is denoted as $\Phi_{\mathcal{M}}$, where $\mathcal{M}$ is the set of k indexes where $\|\hat{e}_{D,k}\|_2$ is not zero. This step discards all the continuous atoms from the dictionary whose discrete parts were not considered in the discrete reconstruction.

Continuous decomposition

After filtering the set of possible continuous atoms from the discrete decomposition step, a sparse reconstruction of the continuous part of the observed vectors is conducted as follows:

$$\min_{x_C, e_C} \frac{1}{2} \|y_c - \Phi_{\mathcal{M}} x_c - e_c\|_2^2 + a_C \|x_C\|_1 + b_C \sum_{k=1}^{K_C} \|e_{C,k}\|_2 \tag{3}$$

A block sparsity is enforced to the vector e_C as for its discrete counterpart e_D. However, an additional sparsity is introduced for x_C controlled by the regularization parameter a_C. This formulation reflects the fact that a nominal continuous signal can be well approximated by a linear combination of few atoms of the dictionary $\Phi_{\mathcal{M}}$ and that anomalies are rare and affect few parameters at the same time.

2.3.3 Anomaly Detection

The discrete and continuous decompositions detailed above lead to the two anomaly vectors $\hat{e}_D$ and $\hat{e}_C$. These vectors are used to detect potential anomalies affecting the telemetry using an anomaly score defined as:

$$a(y) = \begin{cases} -1 & if\ M = \emptyset \\ \left\| \widehat{e_c} \right\|_2 & otherwise \end{cases} \tag{4}$$

An anomaly is then detected using this score as follows:

$$\text{anomaly detected if} \begin{cases} a(\mathbf{y}) = -1(\text{Discrete AD}) \\ \text{or} \\ a(\mathbf{y}) > S_{\text{PFA}}(\text{Continuous AD}) \end{cases} \tag{5}$$

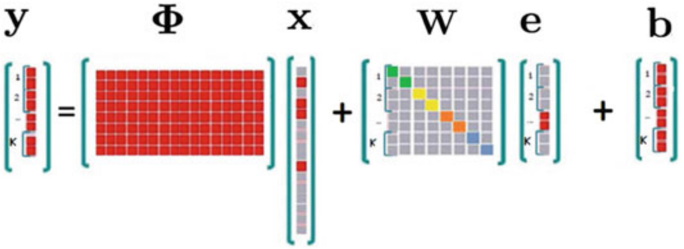
where S_{PFA} is a threshold depending on the probability of false alarm of the anomaly detector (which has to be adjusted by the user). This threshold can be determined using receiver operating characteristic (ROC) curves using a learning database with an available ground-truth.

Note that the detection can be performed parameter by parameter using univariate anomaly signals $e_{D,k}$, k = 1, …, K_D and $e_{C,k}$, k = 1, …, K_C. This allows for identifying the subset of parameters responsible for an anomaly detection inside a larger context, which is essential in an operational environment.

2.3.4 ADDICT Extensions

From the presented ADDICT algorithm, we can derive different extensions that are described in this section and that will be assessed in different use cases in the next sections.

The **W-ADDICT** algorithm proposes to weight the components of the anomaly vector. This weighting allows us to introduce user knowledge for AD and highlight telemetries for which AD should be more sensitive than others. Weights are introduced through a matrix W such that:

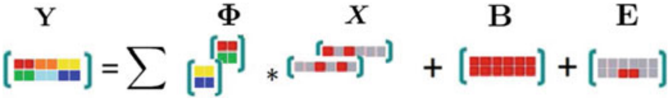


(6)

The **C-ADDICT** algorithm is similar to ADDICT with a convolutive dictionary, i.e., using the following decomposition:

$$Y = \sum_{l} \Phi_1 * x_l + E + B \tag{7}$$

Thanks to the convolutions, time invariance by translation is naturally embedded in the method. Moreover, we have shown [2] that the dictionary learning step is easier. The convolutive decomposition introduced above can be written using a compact matrix form as illustrated below.



(8)

The different versions of ADDICT are tested using different use cases in the next section.

3 Experimental Results

This section presents the experimental results obtained with the previously described algorithms. It firstly focuses on describing the testing methodology and the results of the performance evaluations based on real and simulated spacecraft telemetry, including comparison to other state-of-the-art methods: One-Class SVM (OC-SVM) [6], NOSTRADAMUS [7] and Mixture of Probabilistic Principal Component Analyzers and Categorical Distributions (MPPCAD) [8]. Finally, ADDICT is confronted to industrial use-cases, especially regarding user feedback integration, which is one of the outputs the algorithm intends to provide in operational applications.

3.1 Methodology

The proposed ADDICT, C-ADDICT and W-ADDICT algorithms have been applied to telemetry time-series for spacecraft health monitoring. The training and test datasets consist of 10 different parameters – 3 discrete and 7 continuous - obtained from real and simulated satellite telemetry, detailed in Fig. 3.

In the training stage, the ADDICT (W-ADDICT) dictionary (composed of L = 2000 atoms) and the C-ADDICT dictionary (composed of L = 100 filters) were learnt from 2 months of telemetry describing normal behavior of the spacecraft. This choice represents a compromise between computational complexity and estimation quality, as the number of filters influences the quality of representation of the healthy signals and thus limits the number of false alarms.

In the test stage, the proposed algorithms were evaluated on 18 days of telemetry data with 7 anomaly partitions including: 1 univariate discrete anomaly (Fig. 4, anomaly #1), univariate continuous anomalies (Fig. 4, anomalies #2, #3, #5 and #6), 1 multivariate discrete-discrete anomaly (Fig. 4, anomaly #4) and 1 multivariate continuous-discrete anomaly (Fig. 4, anomaly #7).

The ADDICT, C-ADDICT and W-ADDICT algorithms were compared to 3 state-of-the-art methods which were evaluated on the same dataset:

- The One-Class Support Vector Machine (OC-SVM) [6] tested in a multivariate framework with the ADDICT preprocessing.
- NOSTRADAMUS [7], which is a univariate AD algorithm proposed by CNES based on the OC-SVM method. Each telemetry signal is segmented into time windows of fixed size. Statistical features (mean, minimum, standard deviation, etc.) are then computed on each window. The OC-SVM algorithm is finally used

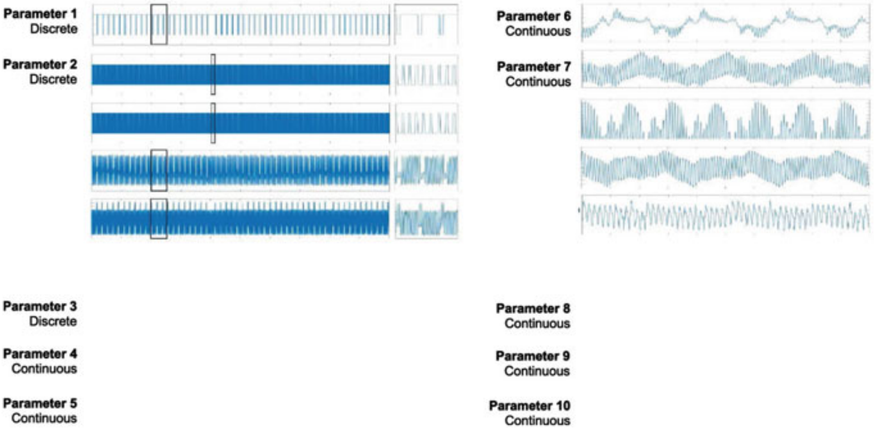


Fig. 3 Extracts of training and test datasets: 10 telemetry parameters including 3 discrete parameters (1 to 3) and 7 continuous parameters (4 to 10). A zoom of parameters 1 to 5 is given to better show their high frequency behaviour

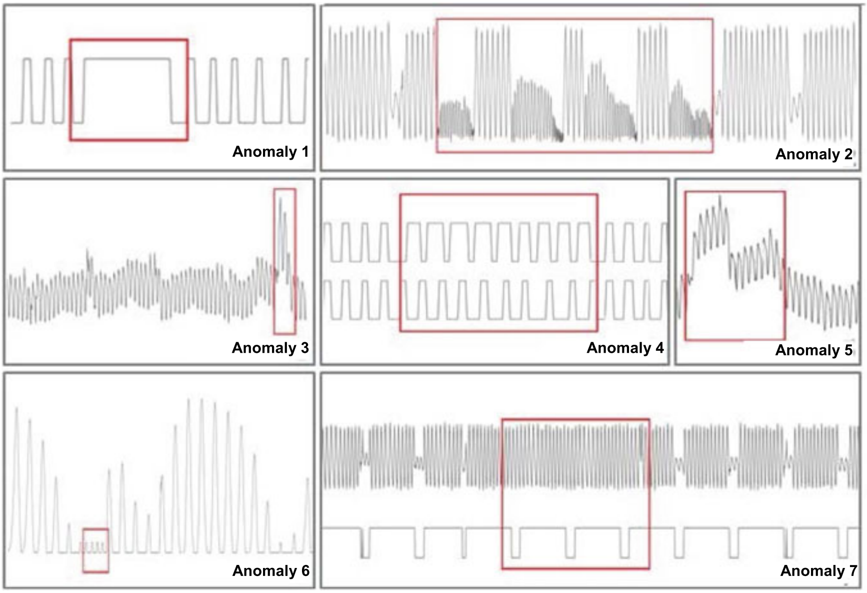


Fig. 4 Examples of univariate (1,2,3,5,6) and multivariate (4,7) anomalies in telemetry data (red boxes)

to define a decision frontier in the feature space containing most of the normal instances, which will be used to detect potential anomalies in new time windows. The NOSTRADAMUS method is able to process continuous and discrete signals using different sets of features.

– The Mixture of Probabilistic Principal Component Analyzers and Categorical Distributions (MPPCAD) algorithm [8], which is a multivariate AD method based on probabilistic clustering and dimensionality reduction that has been applied to the detection of multivariate anomalies in satellite telemetry by JAXA [9].

Table 1 below details the hyperparameters used for the ADDICT, C-ADDICT and W-ADDICT, that have been adjusted by a grid-search based on the ground-truth.

The proposed AD rule (also used in [10]) is based on the estimated anomaly signal $e_k = [e'_{1;k},\ldots,e'_{P;k}]'$, where P is the number of time-series acquired by the telemetry system. An anomaly score is defined as the norm of the anomaly signal, i.e., $a(s_k) = \|e_k\|_2$. This anomaly score is compared to a threshold and a continuous anomaly is

Table 1 Chosen hyperparameters for ADDICT, C-ADDICT and W-ADDICT on the reference dataset

ADDICT	$b_D = 4$	$a_c = 0.06$	$b_c = 0.9$	
C-ADDICT	$\lambda = 0.02$	$\beta = 0.5$		
W-ADDICT	$b_D = 4$	$a_c = 0.06$	$b_c = 0.9$	$\alpha = 0.8$

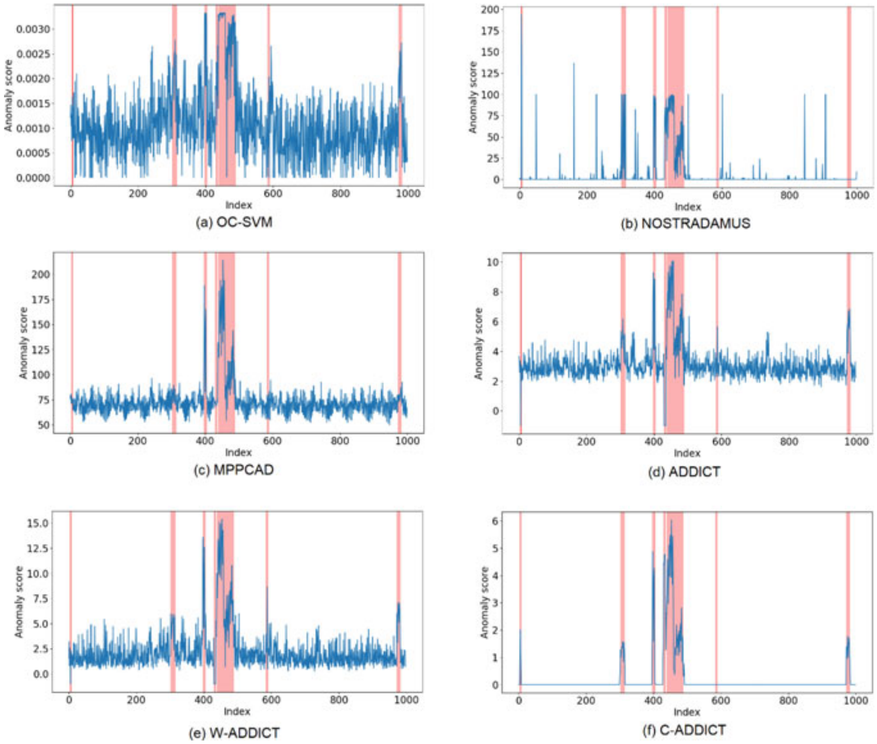
detected if the score exceeds the threshold, i.e.

$$\text{Anomaly detected if } a(s_k) > \text{SPFA} \tag{9}$$

where SPFA is a threshold depending on the probability of false-alarm of the detector. This threshold was tuned by cross validation from data with an available ground truth.

3.2 Performance Evaluation

This section compares the detection performance of the proposed ADDICT, C-ADDICT and W-ADDICT algorithms to the three state-of-the-art methods recalled before. Figure 5 shows the anomaly scores returned by OC-SVM (a), NOSTRADAMUS (b), MPPCAD (c), ADDICT(d), W-ADDICT (e) and C-ADDICT (f) for the ground-truth test dataset with anomaly periods marked by red backgrounds.



(a) OC-SVM

(b) NOSTRADAMUS

(c) MPPCAD

(d) ADDICT

(e) W-ADDICT

(f) C-ADDICT

Fig. 5 Anomaly scores for test signals of the ground-truth dataset with anomaly periods marked by red backgrounds

The MPPCAD algorithm returns low anomaly scores outside the anomaly periods and assigns high anomaly scores for univariate anomalies corresponding to extreme values of the time-series (anomalies #3 and #5, see Fig. 5). However, the scores of other anomalies of the dataset are not high enough to allow detection by this method. The OC-SVM algorithm detects anomalies affecting continuous parameters but fails for discrete univariate anomalies (such as anomaly #1). The NOSTRADAMUS algorithm assigns high scores for all univariate anomalies but fails to detect one of the multivariate anomaly (anomaly #7). In addition, the algorithm returns high scores for significant number of periods without anomaly.

The C-ADDICT algorithm performs very well on assigning high scores only for anomalies periods, both on continuous and discrete data, while other nominal periods are assigned a zero-score, meaning few false alarms compared to the other methods. However, anomaly #6 is not detected by C-ADDICT, while it is detected by OC-SVM, ADDICT and W-ADDICT, with better results for W-ADDICT whose weights are adjusted to each time-series allowing AD to be improved. Note that this anomaly is quite difficult to detect since it has a low amplitude and is observed in a limited time interval. The fact that this anomaly only exists in a small interval will not affect the global probability of detection (as seen in Table 1).

Figure 6 displays the receiver operational characteristics (ROCs) of the five methods. Quantitative results in terms of probability of detection (P_D), probability of false alarm (P_{FA}) and area under the curve (AUC) are also summarized in Table 2. The selected detection thresholds correspond to the points of the ROC curves closest to the "perfect classifier point" (0;1), satisfying the best compromise for spacecraft health monitoring.

AD methods based on standard (ADDICT and W-ADDICT) or convolutional (C-ADDICT) sparse representations appear to be more competitive in these tests. Indeed, ADDICT and W-ADDICT provide high probabilities of detection (P_D =

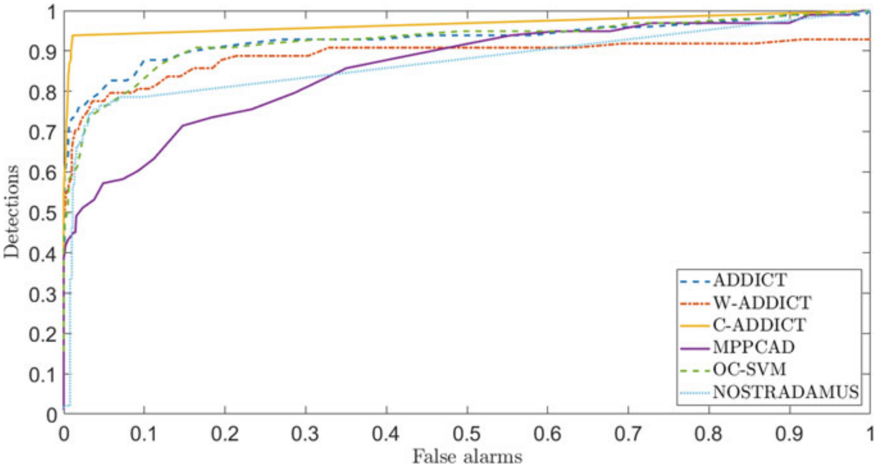


Fig. 6 ROC curves of the different AD methods

Table 2 Values of P_D, P_{FA} and AUC for the different AD methods

Method	Threshold	P_D (%)	P_{FA} (%)	AUC
OC-SVM	0.019	80.9	7	0.9413
MPPCAD	76	81.9	25.9	0.8779
NOSTRADAMUS	29	79.6	6	0.88
ADDICT	4.1	81	3	0.937
W-ADDICT	4.5	85.1	2.7	0.9703
C-ADDICT	0	94.7	1.7	0.9706

81% for ADDICT and $P_D = 85.1\%$ for W-ADDICT) and low probabilities of false alarm ($P_{FA} = 3\%$ for ADDICT and $P_{FA} = 2.7\%$ for W-ADDICT). Despite one missed detection, C-ADDICT seems to be even more competitive with $P_D = 94.7\%$ and $P_{FA} = 1.7\%$.

Note that the other AD methods are not performing as well in part due to the fact that they are not designed for this type of dataset:

– MPPCAD is processing multivariate time windows composed of only one data sample whereas this type of anomalies requires to take into account multiple data samples at once. It would be interesting to test this method with bigger time window sizes.
– NOSTRADAMUS is processing parameters independently and thus fails in detecting multivariate anomalies.
– OC-SVM is able to detect anomalies affecting at least one continuous parameter. However, it does not perform well with the discrete anomalies, which are not detected in this dataset.

These encouraging results obtained with the C-ADDICT algorithm are explained by its translation invariance property, which allows us to detect anomalies appearing at time instants that are different from those observed in the dictionary. Furthermore, an interesting property of C-ADDICT is the possibility of setting the detection threshold to a small value thanks to a better sparsity of the anomaly signal e_k. This represents a real advantage compared to the other methods, for which it is necessary to adjust an appropriate threshold using a ground-truth (which is not always available in operational contexts).

3.2.1 Impact of the Dataset Size on Performances

An important part of performance evaluation for operational applications is how the algorithms scale with the dimension, i.e., how many parameters can be processed jointly while maintaining a good detection performance. Scaling will necessarily heavily impact the computational resources needed and might also impact the capability of ADDICT to detect anomalies affecting few parameters inside a big dataset. A few preliminary tests have been performed using ADDICT, increasing the number of parameters from the original 10 to 20 and finally 30 telemetry parameters. Note that all the additional parameters that have been included in the database did not contain any new anomaly.

The results displayed in Fig. 7 show that up to 30 parameters, the performance of ADDICT does not change significantly. However, the computational load is noticeably impacted by the number of parameters, as shown in Table 3 (note that better results could be obtained through code optimization, planned in future work). This is expected due to the constraints inherent to AD methods using sparse representation techniques. Indeed, sparse representations require to build large-enough dictionaries (i.e., with enough atoms) to ensure a good detection performance. The increase of computation cost is also due to the size of the dictionary atoms, which is proportional to the number of parameters. This limitation is even more significant for C-ADDICT whose execution time is higher than ADDICT (58s versus 9s for 10 telemetry parameters and 1 day of data), due to the use of a convolutional dictionary, which makes it impossible to run multiple detections in parallel.

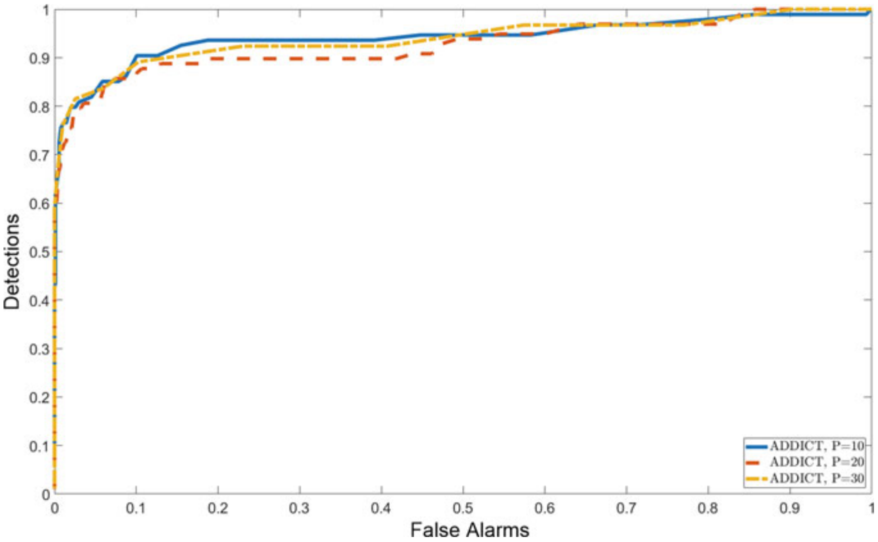


Fig. 7 ROC curves for ADDICT using different dataset sizes

Table 3 Number of dictionary atoms and computation time in respect to the dataset size

Number of parameters	ADDICT/C-ADDICT number of dictionary atoms/filters	ADDICT/C-ADDICT computation time for a 1-day detection phase (seconds)
10	2000/100	9 / 58
20	8000	134
30	15,000	205

3.3 Industrial Use-Cases

Different use-cases have been provided by CNES and ADS in order to validate the use of ADDICT in a space operation context. This section focuses on a specific usecase dedicated to event detection in multivariate telemetry signals (Sect. 3.3.1) and on two use-cases aiming at testing more specifically the "online-learning" feature (or "user feedback integration") of ADDICT (Sect. 3.3.2), which is a highly anticipated point for satellite operators using AD algorithms.

3.3.1 Event Detection in Telemetry

This first use-case study the capability of ADDICT to detect, and even predict, the occurrence of a known (but not predictable) event that can affect the proper functioning of a satellite.

This study has been performed using a telemetry dataset from 3 Airbus-made satellites with about 20 selected parameters each (not necessarily the same parameters for each satellite). About a hundred orbits have been used for capturing the nominal behavior of the satellites using the dictionary learning algorithm and about a hundred of different orbits, during which the looked-for event is known to have happened, have been used for detection purposes.

Examples of telemetry parameters are shown in Figs. 8, 9 and 10. The event occurrences displayed in red usually appear as an extended mode change of some pieces of equipment (discrete parameters: n°1 to 8 in Fig. 8, n°1 in Fig. 9 and n°5 in Fig. 10) or as a major drop and/or frozen-value (null variance) of some continuous parameters (n°13 to 18 in Fig. 8, n°11 to 17 in Fig. 9 and n°11 to 17 in Fig. 10).

The ADDICT and C-ADDICT algorithms have been tested on this use-case with the corresponding configuration:

- ADDICT uses telemetry time-windows of size W = 100 and learnt dictionaries constituted of 8000 atoms of size N = 100K (with K the number of studied parameters).
- The C-ADDICT dictionary is constituted of 200 filters of size M = 100.

The detection score returned by those two algorithms is shown in Figs. 11, 12 and 13 where the red window shows the event occurrence according to satellite experts (ground truth). For the three satellites, a significant increase in the score of the two methods can be observed at the time of the event with respect to the previous nominal orbits, confirming the proper detection of the event of interest. For satellites n°2 and n°3 a noticeable score increase (bigger than the previous average score) is even seen a few orbits prior to the actual event which is explained by an early slight change occurring before the event.

To better understand the detection obtained using ADDICT and C-ADDICT, the two algorithms offer the possibility to analyze the univariate scores to better identify which parameters are affected by an anomaly (or by the event to be detected). The

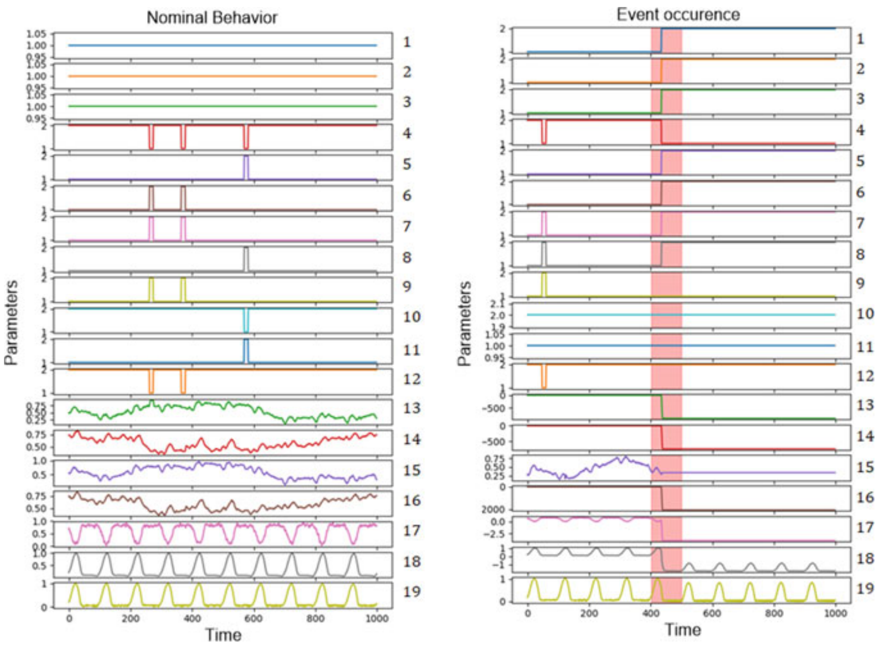


Fig. 8 Telemetry parameters for satellite n°1 with 10 orbits of nominal behavior (left figure) and 10 orbits around the occurrence of the event to be detected (right figure)

scores of satellites n°2 and n°3 are displayed in Fig. 14 suggesting the following comments:

- For satellite n°2, parameters #11 and #13 yield high detection scores before the occurrence of the event explained by the extreme values recorded in the few orbits prior to the event.
- For satellite n°3, scores higher than the average are attributed to all the continuous parameters a few orbits prior to the event and more specifically to parameter #15 which has the highest scores.

Even if these results need to be confirmed by satellite engineers, they prove the capability of ADDICT to detect an unexpected event in telemetry and even its first signs in some of the parameters a few orbits before the event appears. Note that a multivariate analysis is not always necessary to detect an abnormal event as illustrated in this example. Note also that ADDICT offers the possibility to detect which parameters are responsible for the apparition of an unexpected event through the analysis of univariate scores.

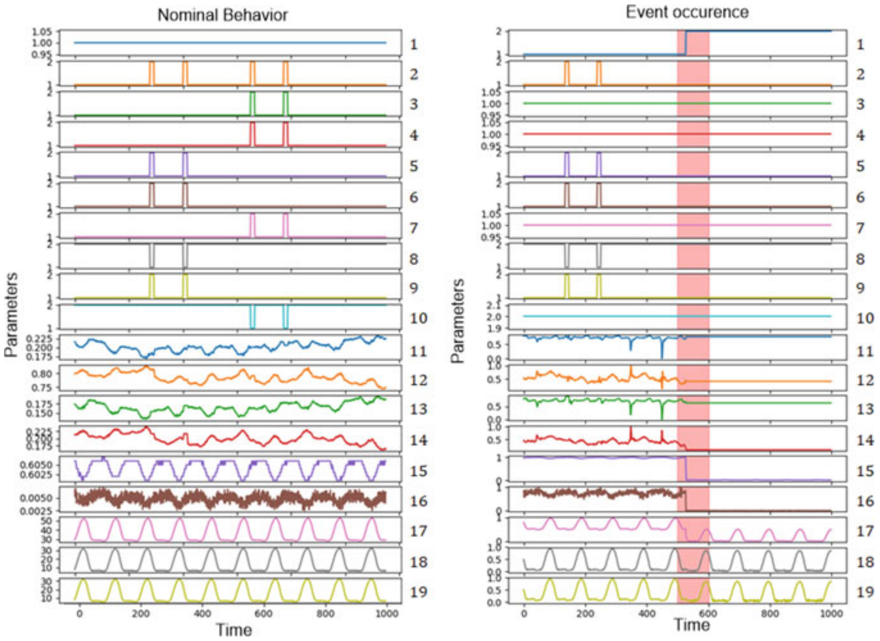


Fig. 9 Telemetry parameters for satellite n°2 with 10 orbits of nominal behavior (left figure) and 10 orbits around the occurrence of the event to be detected (right figure)

3.3.2 User Feedback Integration

Definition

By online-learning we mean the capability of a machine learning algorithm to have its learning models (its dictionaries in the case of ADDICT) updated through user feedback. More precisely, operators flagging the results of the algorithm as bad detections (either a false alarm or a non-detected anomaly) should be able to integrate this information into the learnt models. This possibility offers operators more control over machine-learning AD algorithms by giving them a way to input their expert knowledge in the process. Moreover, these techniques may help to increase the performance of these algorithms over time (allowing more accurate detections and fewer false alarms).

Regarding ADDICT, online-learning is considered through the update of the learnt dictionary by adding wrongly-detected time windows as new atoms of the dictionary. This way, if a time window similar to a previously wrongly detected anomaly is presented to ADDICT, the corresponding telemetry parameters will be more easily reconstructed with the atoms of the updated dictionary and the anomaly score should be greatly reduced.

In the next sections, depending on the use-cases, experiments focus on ADDICT or its weighed or convolutional versions W-ADDICT and C-ADDICT regarding online

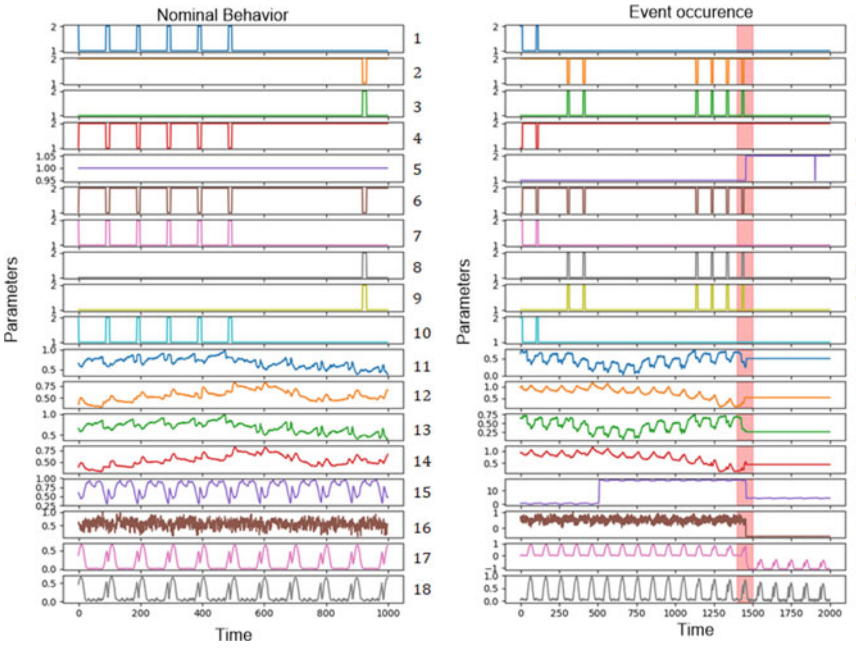


Fig. 10 Telemetry parameters for satellite n°3 with 10 orbits of nominal behavior (left figure) and 10 orbits around the occurrence of the event to be detected (right figure)

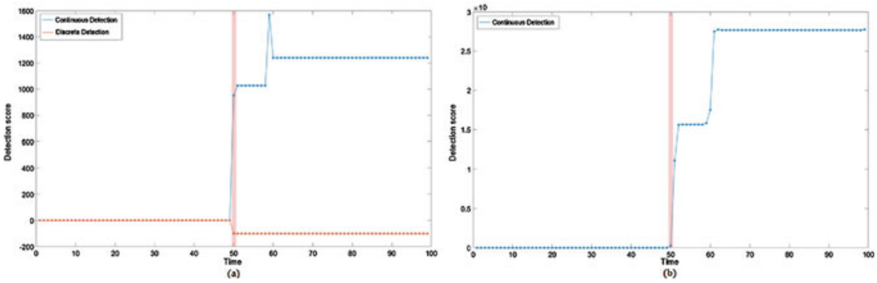


Fig. 11 ADDICT (**a**) and C-ADDICT (**b**) detection score for satellite n°1

learning applications. Other variations of ADDICT could use this online mode as well but have not been tested in this matter yet.

Online-Learning Applied to a Reference Test Database

The database presented in the previous sections has been tested regarding online learning. This database has not been designed specifically for online learning.

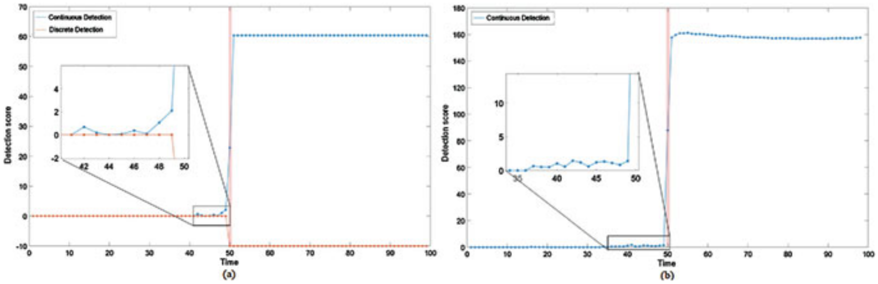


Fig. 12 ADDICT (**a**) and C-ADDICT (**b**) detection score for satellite n°2

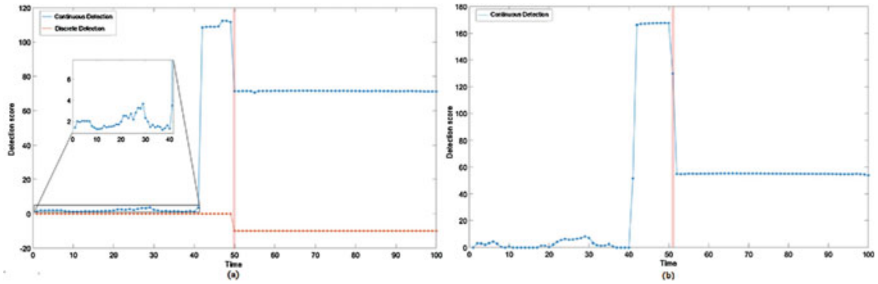


Fig. 13 ADDICT (**a**) and C-ADDICT (**b**) detection score for satellite n°3

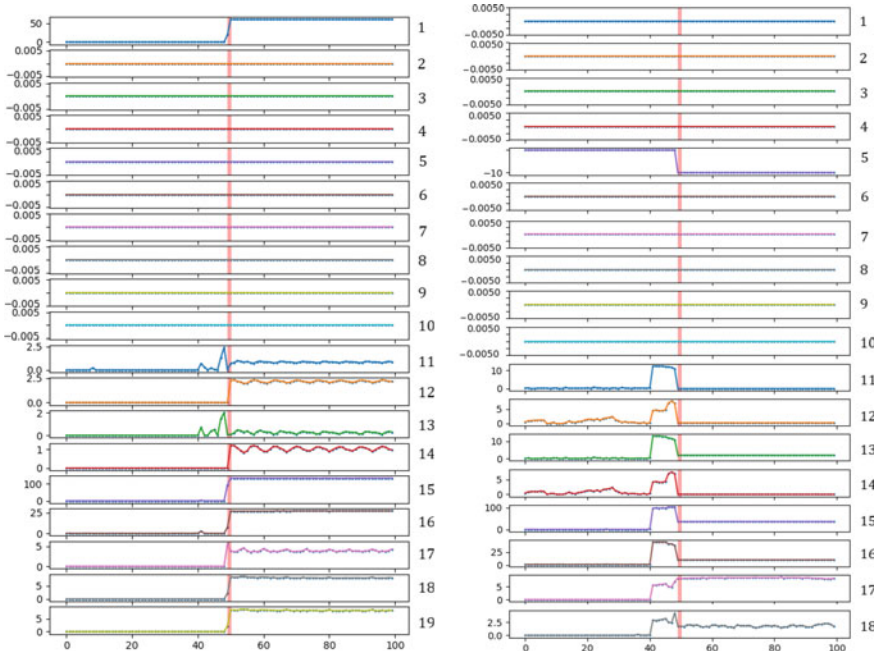


Fig. 14 ADDICT univariate detection score for satellite n°2 (left) and satellite n°3 (right)

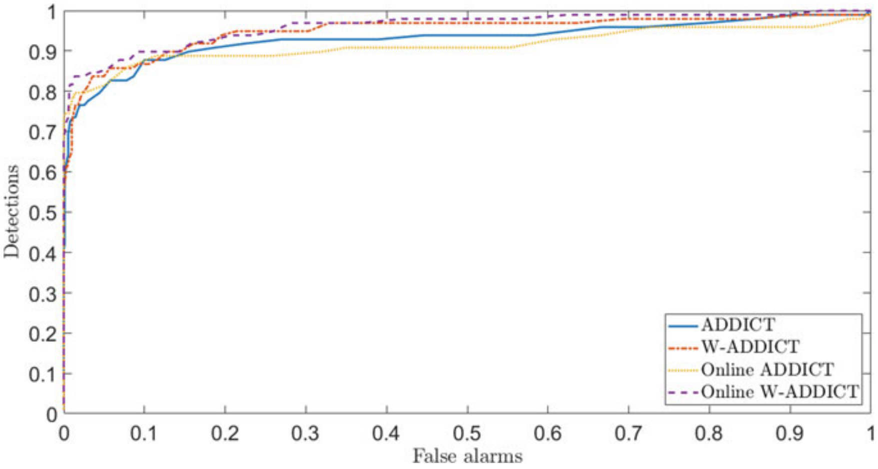


Fig. 15 ROC curve comparison for ADDICT and W-ADDICT with or without online learning

However, it is a representative database of telemetry signals and anomalies that could be found in real-life space operations. Conversely, the use-cases described in the following sections consider telemetry signals for which online learning has been identified as a promising feature beforehand.

Regarding the current experiment, ADDICT and W-ADDICT are confronted through two methodologies:

- Standard mode: learning is performed on the nominal database and the resulting dictionary is evaluated on the test database with respect to the available ground truth. The detection threshold was set to 3.7 for ADDICT and 3.5 for W-ADDICT.
- Online mode: learning is performed on the nominal database. The resulting dictionary is then updated with 152 new atoms corresponding to all time windows wrongly detected as anomalies in the standard mode (i.e., time windows with anomaly scores greater than the detection threshold but labeled nominal in the ground truth). This updated dictionary is then evaluated in the same way as in the first run.

Results are presented in the following ROC curves for ADDICT, W-ADDICT and their respective online versions in Fig. 15 and in Table 4:

Quantitative results show that online learning noticeably reduces the number of false alarms, confirming that including them as atoms of the updated dictionary allows a better reconstruction of similar signals appearing in next occurrences. However, note that some false alarms are still wrongly detected with the updated dictionary.

For ADDICT used with a fixed threshold, the false alarm rate is reduced from 10.2 to 1.2%, with a drop of the good detection rate from 89 to 83%. To keep a same level of good detections, the detection threshold can be set to 3.5 but the false alarm rate is less reduced (7.7% in this case). The results obtained with W-ADDICT show

Table 4 Values of P_D, P_{FA} and Area Under Curve (AUC) for ADDICT and W-ADDICT with or without online learning

Method	Threshold	P_D(%)	P_{FA}(%)	AUC
ADDICT	3.7	89	10.2	0.937
Online ADDICT	3.7	83	1.2	0.947
Online ADDICT	3.5	89	7.7	0.947
W-ADDICT	3.5	88.3	9.9	0.96
Online W-ADDICT	3.5	88.3	4.4	0.9766

a similar reduction in the number of false alarms, going from 9.9 to 4.4% with a better robustness compared to ADDICT.

This first experiment demonstrates that online learning as provided by ADDICT is a significant way of reducing the number of false alarms over time. The next sections consider the interest of online learning for other industrial use-cases.

Orbital Maneuver Detection

The first industrial use-case, brought by CNES, focuses on orbital maneuvers and its impact on telemetry. Indeed, many telemetry parameters appear atypical in comparison to routine telemetry during these operations due to a changing satellite mode, an attitude modification, propeller actuations, tank pressure decrease, etc.

Univariate methods are not well suited to capture this type of contextual scenario:

- If learning data is strictly limited to routine telemetry (i.e., excluding these maneuver operations), the algorithm would detect the signals from the maneuvers as abnormal behaviors in comparison to routine telemetry.
- Conversely, adding these operations to the learning data would probably avoid later false alarms induced by similar maneuvers. However, it would probably degrade the detection capability of the univariate algorithm. For instance, if cases of tank pressure decrease associated with maneuvers are captured as nominal data during the learning phase, a later pressure decrease not linked to such activity would be considered as nominal whereas it could actually be the symptom of a hypothetical propellant tank leak.

ADDICT offers 2 ways of tackling these issues:

1- A multivariate analysis of the telemetry allows the operational context of the satellite to be taken into account. This should help to reduce false alarms (maneuvers being wrongly detected as anomalies) or missed detection (out-of-context telemetry change being wrongly considered nominal) as explained previously.

2- Online learning gives the opportunity to sequentially add the new maneuvers into the dictionary, capturing over time the slight variations between these operational events, allowing for better processing of later maneuvers.

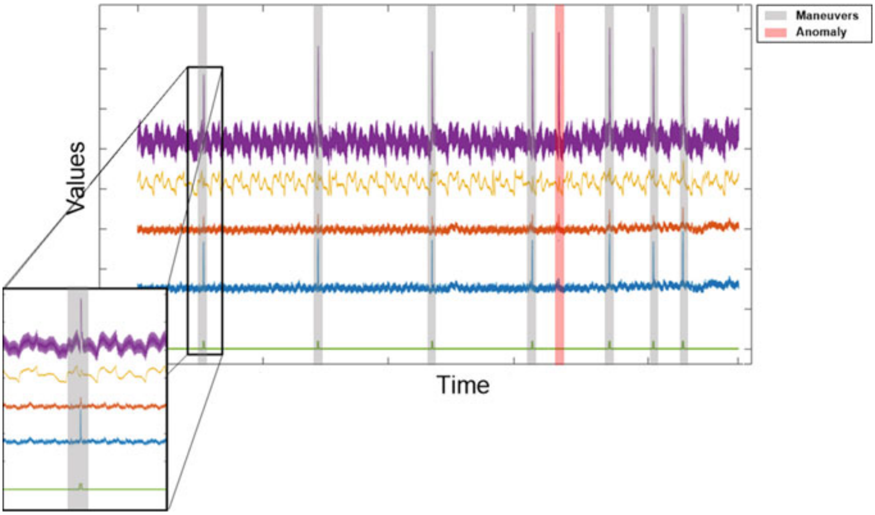


Fig. 16 Orbital maneuver dataset

For this test, a dataset of 5 telemetry parameters from a CNES LEO observation satellite is used and presented in Fig. 16. It consists of 4 continuous signals and 1 discrete signal (in green) representing the maneuver mode. Gray windows in the figure represent expected maneuvers, whereas the red window is a factitious anomaly (i.e., some of the continuous signals behave like a maneuver but the green discrete signal corresponding to the maneuver mode has been modified).

Three sequential scenarios have been tested with ADDICT and C-ADDICT, whose results are presented in Fig. 17:

- Scenario 1: ADDICT and C-ADDICT are trained on routine telemetry exclusively, without any maneuver period considered during learning. This scenario shows that both algorithms detect the maneuvers as anomalous time windows with a significant anomaly score. This is expected as these methods are designed to detect new types of behaviors. Note that for ADDICT a negative anomaly score corresponds to an anomaly detected on a discrete parameter, whereas a positive anomaly score corresponds to an anomaly detected on a continuous parameter.
- Scenario 2: through online learning, the first maneuver period is added in the dictionary and the telemetry is tested with this updated dictionary. The first maneuver is this time well discarded by ADDICT and C-ADDICT. Regarding the other maneuvers and the anomaly, it is already possible with C-ADDICT to choose a detection threshold allowing all expected maneuvers to be considered as nominal and the multivariate anomaly to be detected. Note that ADDICT is not able to provide these results yet.
- Scenario 3: continuing Scenario 2, a second maneuver period is added to the dictionary. This time, the ADDICT scores for the expected maneuvers decrease sufficiently, allowing a threshold to be defined to detect the multivariate anomaly.

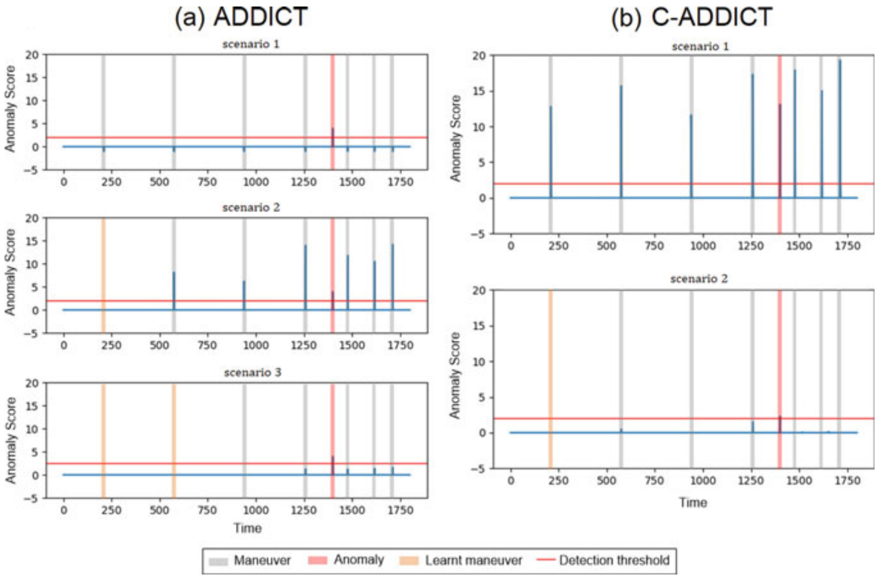


Fig. 17 Orbital maneuver testing scenario results with ADDICT and C-ADDICT

These results demonstrate the interest of a multivariate analysis through sparse estimation methods coupled with online learning, as offered by ADDICT and C-ADDICT, for taking into account known operational contexts and eliminating false alarms associated with them. These capabilities have been demonstrated on orbital maneuvers but can be extended to other recurring satellite activities that are too far from routine to be considered as such but are still nominal (expected behavior). In addition, these results show that ADDICT and C-ADDICT are able to integrate successfully the user-feedback after 1 (C-ADDICT) and 2 iterations (ADDICT).

Telemetry Monitoring During Eclipses

The second industrial use-case, brought by ADS, focuses on integrating different operating modes encountered by a satellite that heavily impacts the way the satellite telemetry behaves.

The considered dataset, shown in Fig. 18, comes from an ADS-manufactured geostationary satellite and consists of 2 continuous signals showing two separated routine behaviors: the first one during annual eclipses periods (gray areas) and the second one during the rest of the year (white areas). The telemetry signals are noticeably different during these two periods.

ADDICT and C-ADDICT have been tested on this dataset using a similar methodology as the one described in the previous use-case: a first testing scenario with learning performed exclusively on the standard operating mode telemetry and other

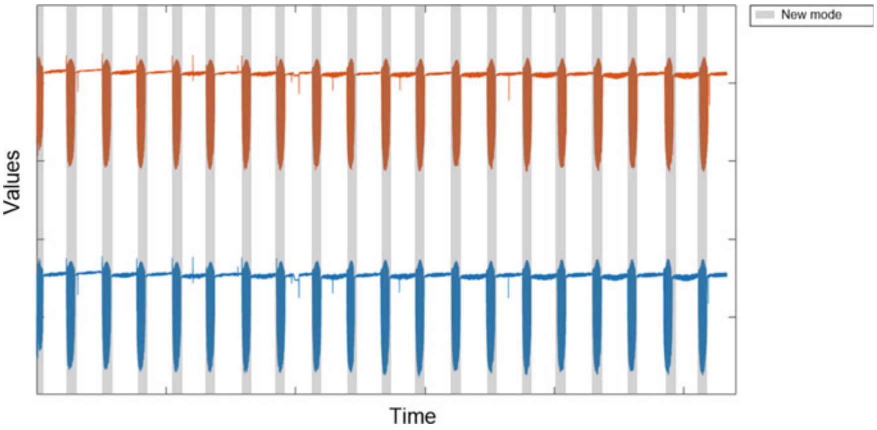


Fig. 18 Eclipse mode dataset

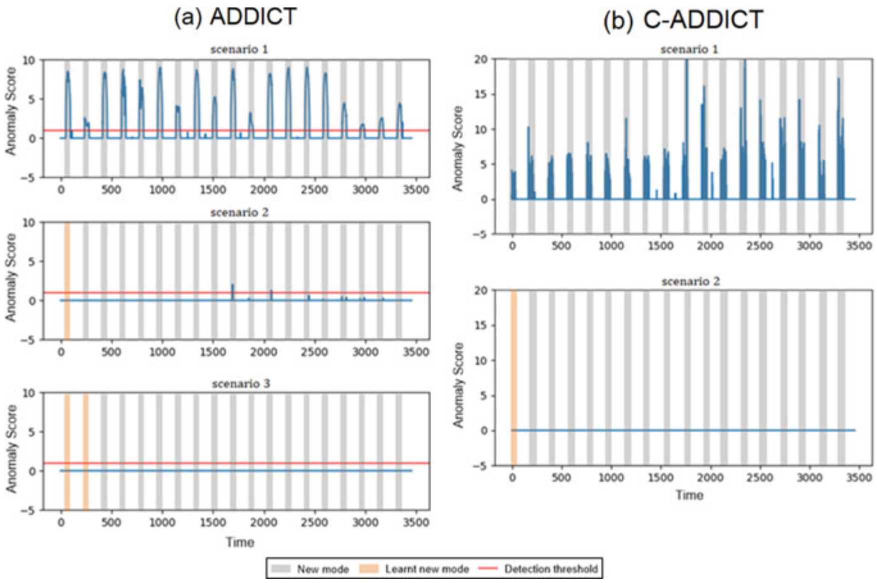


Fig. 19 Eclipse testing scenario results with ADDICT and C-ADDICT

testing scenarios where periods from the eclipse operating modes are sequentially added.

Results of these testing scenarios are presented in Fig. 19 for ADDICT and C-ADDICT. The first scenario shows that ADDICT and C-ADDICT detect, as expected, all eclipse periods as they correspond to a new behavior compared to the dictionary. The second and third scenarios demonstrate that online learning applied to sparse estimation methods are efficient for these kinds of use-cases. Indeed, the eclipse periods are no longer detected by ADDICT after 2 eclipse periods have been added into the dictionary and after only 1 eclipse period for C-ADDICT.

As with the previous use-case, online learning applied with sparse representation methods, as provided by ADDICT and C-ADDICT, is very efficient in capturing the new eclipse mode, effectively eliminating all false alarms after respectively 1 or 2 eclipse periods have been added into the dictionary.

4 Conclusions

A new Anomaly Detection method based on a sparse decomposition into a DICTionary, referred to as ADDICT, was developed during the PhD thesis of B. PILASTRE supported by CNES and Airbus Defence & Space. This paper showed that ADDICT (and its variants) is an efficient method for detecting multivariate satellite anomalies in mixed continuous and discrete telemetry parameters, with competitive results with respect to the state-of-the-art. The performance of ADDICT was evaluated using a real datasets and different use-cases. This paper also demonstrated the interest of using ADDICT in an online mode yielding interesting results in operational contexts (with a reduction of the false alarm rate versus time, the integration of expected operations without detecting anomalies, etc.). Future work includes the industrialization of the ADDICT algorithms and their integration in satellite control centers.

References

1. Adler A et al (2015) Sparse coding with anomaly detection. J Signal Process Syst 79(2):179–188
2. Pilastre B (2020) Estimation parcimonieuse et apprentissage de dictionnaires pour la détection d'Anomalies Multivariées dans des Données Mixtes de Télémesure Satellite. PhD Thesis of the University of Toulouse
3. Barreyre C et al (2019) Statistical methods for outlier detection in space telemetries. In: Space operations: inspiring humankind's future. Springer, Cham, pp 513–547
4. O'Meara C, Schlag L, Wickler M (2018) Applications of deep learning neural networks to satellite telemetry monitoring. In: Proceedings of the conference space operations (SpaceOps'18), May 28-June 1 2018, Marseille, France
5. Pilastre B et al (2019) Multivariate anomaly detection in mixed telemetry time-series using a sparse decomposition. In: Proceedings of the 8th international workshop on computational advances in multi-sensor adaptive processing (CAMSAP), Guadeloupe, France
6. Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC (2001) Estimating the support of a high dimensional distribution. Neural Comput 3(7):1443–1471
7. Fuertes S, Picard G, Tourneret JY, Chaari L, Ferrari A, Richard C (2016) Improving spacecraft health monitoring with automatic anomaly detection techniques. In: Proceedings of the international conference space operations (SpaceOps'16), Daejeon, South Korea

8. Tipping M, Bishop C (1999) Mixtures of probabilistic principal component analyzers. Neural Comput 11:443–482
9. Yairi T, Takeishi N, Oda T, Nakajima Y, Nishimura N, Takata N (2017) A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction. IEEE Trans Aerosp Electron Syst 53:1384–1401
10. Pilastre B, Boussouf L, D'Escrivan S, Tourneret JY (2020) Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning. Signal Process 168:107320

Euclid's Health Monitoring System: Combining and Expanding ESA's Operational Capabilities into New Use Cases

Guillermo Buenadicha, Rui Santos, José Carlos González, Gustavo Marques, and Marco Fresci

Abstract The paper introduces the Health Monitoring System (HMS) for the European Space Agency (ESA) Euclid mission. Euclid, due for launch early 2023 and implementing an extragalactic sky survey relies on tight monitoring of its instrument performance in order to understand and prevent systematics. This requires not only access to a single data source as HouseKeeping Telemetry (HKTM) but also to other input sources that can allow to perform data correlation and cross-matching. The HMS, part of the Science Operations Centre (SOC), covers use cases for the off line monitoring of the health and performance of the instruments but also is a key component in the generation of the entry products for the scientific processing in the mission. It allows storage, access and analysis of time based parametric data and is designed using as baseline the ESA's Analysis and Reporting System (ARES), that provides support for storage, analysis and display of many types of operational house-keeping time based data series. The simplicity of the system architecture and data enhancement is the key for its strength. This concept is being explored by other mission's Science Operations Centres, as BepiColombo, XMM or PLATO, thus hinting new avenues and use cases where cross mission data can be easily exchanged. The paper shows how synergies across different operational areas at ESA create tools and use

G. Buenadicha (✉)
ESA/ESAC, Villanueva de la Cañada, 28692 Madrid, Spain
e-mail: guillermo.buenadicha@esa.int

R. Santos
ESA/ESOC, Robert-Bosch-Straße 5, 64293 Darmstadt, Germany
e-mail: rui.santos@esa.int

J. C. González
Telespazio-Vega Uk S.L. @ESA, Villanueva de la Cañada, 28692 Madrid, Spain
e-mail: Jose.Carlos.Gonzalez.Consuegra@esa.int

G. Marques
CGI, Robert-Bosch-Straße 7, 64293 Darmstadt, Germany
e-mail: gustavo.salvador.marques@cgi.com

M. Fresci
SERCO @ESA, Villanueva de la Cañada, 28692 Madrid, Spain
e-mail: Marco.freschi@esa.int

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_13

cases that provide overall added value. HMS is heavily based on excellent collaboration across ESA's Directorates, and merges know-how and technologies from different operational environments (Mission and Science) into a wider system. The paper will describe the concept and technology of Euclid's HMS and ARES, the data sources, structures and interfaces, and the different operational use cases, focusing highly on the novelty of its use at a Science Operations Centre.

Acronyms/Abbreviations

AND	Alpha Numeric Display
ARES	Analysis and Reporting System
COTS	Commercial Off The Shelf
EC	Euclid Consortium
EDDS	EGOS Data Dissemination System
ESA	European Space Agency
ESAC	European Space Astronomy Centre
ESOC	European Space Operation Centre
ESTRACKESA	Tracking Stations network
GFTS	Generic File Transfer System
HDFS	Hadoop Distributed File System
HMS	Health Monitoring System
HKTM	HouseKeeping TeleMetry
LE1	Level 1 Data Product or Processor
MCS	Mission Control System
MOC	Mission Operations Centre
NISP	Near-Infrared Spectrometer and Photometer
OSS	Operational Sky Survey
OU	Organisational Unit
PUS	Packet Utilization Standard
QLA	Quick Look Analysis
SDC	Science Data Centre
SGS	Science Ground Segment
SOC	Science Operations Centre
SCOS	Spacecraft Control and Operations System
SIS	SOC Interface System
VIS	Visible Imager Instrument
WebMUST	Web Mission Utility & Support Tool

1 Introduction

1.1 Euclid Mission

Euclid is a cosmological mission of the European Space Agency (ESA) designed to map the geometry of the dark universe unveiling the dark matter and energy, due for launch early 2023 [1]. Euclid will conduct from a halo orbit around the Lagrange point number 2 a sky survey during 6 years covering 15,000 square degrees of extragalactic sky, combining data from 2 instruments in the visual (VIS) and near infrared bands (NISP) [2]. Stability and instrument performance is paramount to ensure that the data acquired along the survey is homogeneous and can be processed with same level of quality, so that sufficient galaxies can be extracted to compose a 10ˆ10 catalogue suitable to derive the final cosmological goal of the mission. Euclid's Science Operations Centre (SOC) is based at ESAC (European Space Astronomy Centre), in Spain, where all other SOCs for ESA's scientific missions are also based.

Euclid combines several techniques of investigation, also called cosmological probes, in a very large survey over the extragalactic sky as seen in Fig. 1. Among these cosmological probes, two of them play a major role in the Euclid mission concept and the instrumental approach: the Weak Gravitational Lensing (WL) and the Galaxy Clustering (including Baryon Acoustic Oscillations - BAO). The Weak Lensing measurements consist in observing galaxies distortion caused by gravitational light deflection from unknown and invisible foreground mass concentrations and modified by the expansion of the Universe (cosmic distance ratios). Galaxy clustering stands for any 3-dimension statistical description of clustering and motions of galaxies as function of scales and look back time produced by the combined effects of expansion and gravity on the growth rate of structure. It includes in particular baryon acoustic oscillation (BAO). BAO are a series of wiggles in the matter power spectrum. Individually, WL and Galaxy Clustering are two powerful cosmological

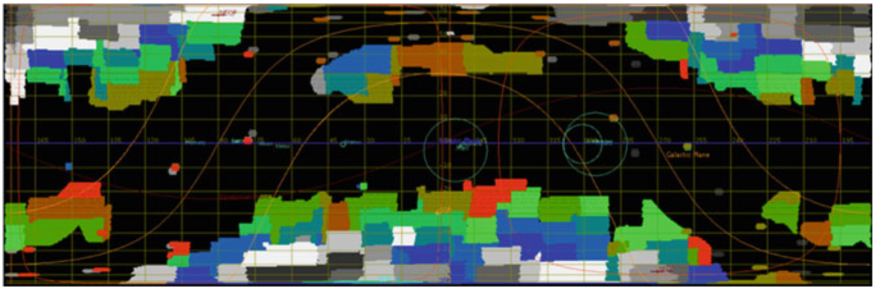


Fig. 1 Euclid Sky Survey, colours represent contiguous patches over the sky, avoidance regions due to galactic and equatorial planes can be seen. Survey is composed of 40,000 fields involving over 160,000 dithers pointings

probes of the Dark Energy. In combination, they will enable control of many undesirable systematic effects and make it possible to break degeneracies in the parameter space of the standard cosmological model.

ESA and the Euclid Consortium (EC) jointly develop the Euclid Mission. ESA has the overarching responsibility for all aspects of the mission and for the fulfilment of the mission requirements and mission objectives. ESA is directly responsible for the development, manufacturing, integration and verification of a spacecraft capable of accommodating the VIS and NISP instruments. ESA is also responsible for the development, procurement, integration and verification of the SOC and MOC. The Euclid Consortium is responsible for the development and timely delivery of the two instruments, NISP and VIS, and for the support during integration and operations. The EC is also responsible for their part of the Euclid Science Ground Segment. Furthermore, the EC responsibility extends to the provision of the ground based photometric surveys and spectroscopy to fulfil its cosmology objectives. The launcher is a Soyuz and is procured by ESA.

1.2 Ground Segment

The MOC (at ESOC) conducts all mission operations of the spacecraft from LEOP to decommissioning, including telecommanding, acquisition of space data and mission safety. The SOC is responsible for the maintenance of the Operational Sky Survey, issuing the scientific planning schedule and the elaboration and distribution of the first level of science data, from the telemetry generated by the spacecraft, called Level 1 (LE1) data. The further elaboration of the data and the generation of the level 2 and level 3 data is performed in the Science Ground Segment (SGS) by a number of data Centres developed by the EC. The SOC is responsible for the operations and the maintenance of the Euclid Archive System with support and contribution of the EC.

The data produced by Euclid, as seen in Fig. 2 is sent daily from Space to ESA's ESTRACK Ground Stations (Malargue or Cebreros) during the Daily Telemetry and Commanding Period (DTCP). From the receiving station the data is sent to ESOC, where the different data components are handled differently. The Science Data (sent from space as files using the CCSDS CFDP protocol) are retransmitted as is received to the SOC, for later processing. The HKTM (both Real Time, generated during the DTCP, and archived, sent through files containing CCSDS PUS packets) is processed at MOC, and eventually converted into parameter timestamped information that is archived in the SCOS 2000 parameter archive. From this archive it is exported for persistence in the ARES system at MOC, and replicated also to the SOC.

The science data files, received at SOC are processed to a first level (uncompressing and reordering of pixel data, adding relevant metadata) so that they can be later fed to the Science Ground Segment Processing, where data is processed at pixel level, calibrated, and galaxy and source extraction is performed to populate the Euclid Catalogue from which higher level science and cosmological probes are

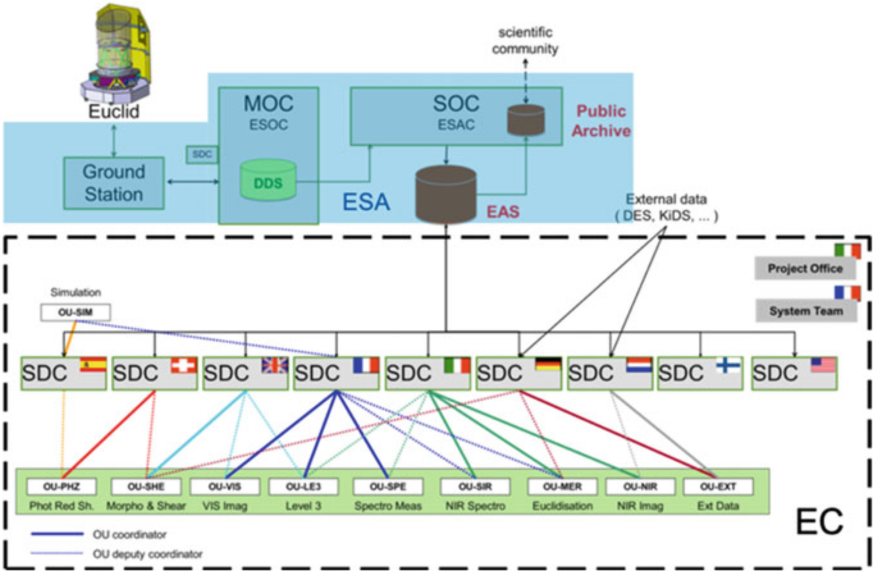


Fig. 2 Euclid Ground Segment. Top blue area are the components provided by ESA, the bottom area is the Science Ground Segment involving the scientific processing of data, by Euclid Consortium. Data for processing is placed in the EAS where it is assigned to an SDC

derived. In parallel, SOC performs a Quick Look Assessment of the data to ensure correctness of the Survey execution. The SGS is a distributed system running over 9 science data Centres across Europe.

2 Parametric Information and ARES

2.1 Parametric Information

A parameter is any characteristic that can help in defining or classifying a particular system. Samples of parameters may be extracted at given times, creating instances of timestamped parameter value series. This concept is key in the monitoring of space missions, where spacecraft and payloads are monitored through parametric information acquired from sensors or produced as part of the On Board SW processing, see an example in Fig. 3. The-time stamped parameters (together with their transport protocol metadata) represent therefore the system from a ground perspective. This information needs to be archived and services provided to ensure access to the data, in order to evaluate suitability of operations on the system as well as identify trends and evolutions.

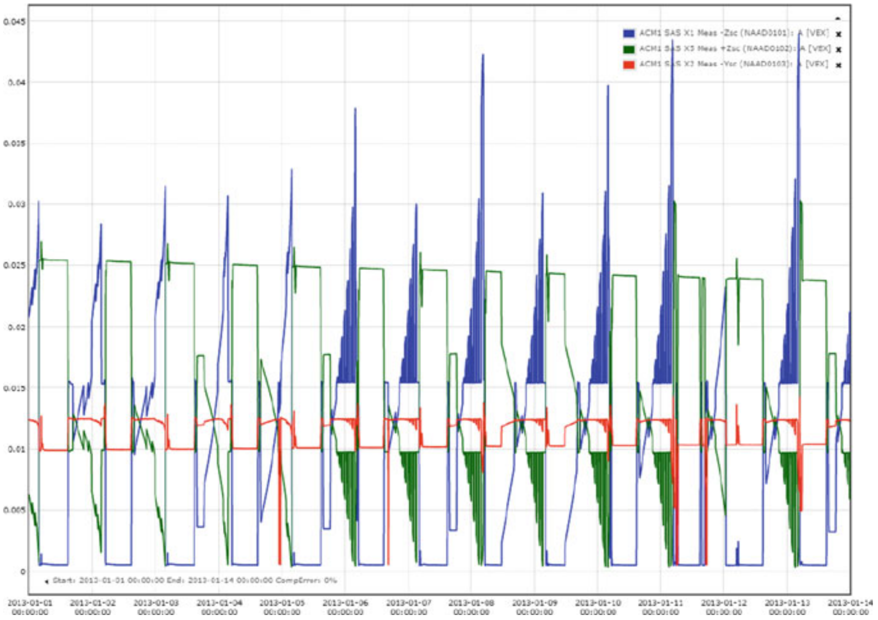


Fig. 3 Example of plotting several parameters along a time range of 2 weeks, allowing feature cross matching. *Source* [6]

As ESA systems have evolved into standardized Telemetry structures in the last few decades, systems oriented to collect, present and analyse the parametric data have been reused and shared across missions. Over the last 20 years the Mission Utility and Support Tools (MUST) has been the main mean used, and still is so by many flying missions. However, as missions grow in complexity also does the volume of produced information, that need to be easily escalated and allow also to adopt new information data types.

2.2 Ares

The ESA/ESOC Analysis and Reporting System (ARES) provides support for offline storage, analysis and display of several types of operational house-keeping data, including: telemetry packet and parameter information; telecommand history; Spacecraft and Mission Control System events. See references [3, 4] for further details. The system is being maintained and augmented with additional features, e.g. recently the system was extended to support the storage and retrieval of files and generic events (which can include the aforementioned MCS events).

One of the key data sources for ARES is the Mission Control System for each mission. Although the interface for ARES is EDDS the end to end system relies

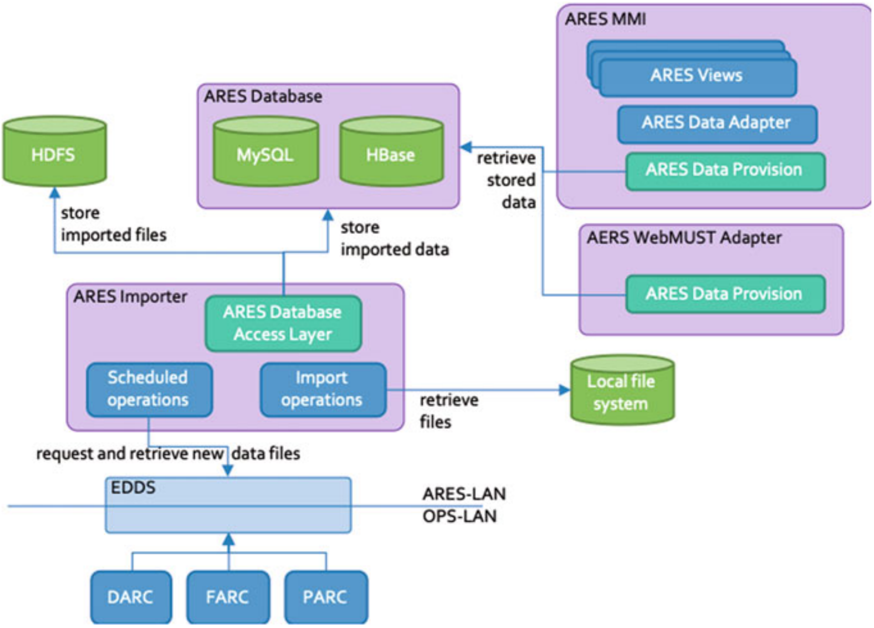


Fig. 4 ARES components showing the interfaces with data provision to consumer applications, importing interface and also the links to the HDFS infrastructure

on other capabilities of the MCS to process and organize the data so it is ready to be injected into the offline storage system. In [5] one can see an example of the configuration and deployment the MCS system (including ARES) for Exomars Trace Gas Orbiter (TGO). As the software used in the process is mostly composed of ESOC common infrastructure systems the process is similar for other missions facilitating the deployment for future missions.

The central element of the ARES system is the ARES database. As seen in Fig. 4, the database is populated with operational data coming from different data sources (PARC, FARC, DARC) through the EDDS, following scheduled operations, or from local files via manual import operations, using the ARES importer. The ARES MMI and other ARES clients access the operational data through the ARES data provision API, in order to retrieve and analyze it.

The ARES importer uses the Yarn component of the Hadoop ecosystem to schedule jobs that import the data from EDDS into the ARES database.

ARES makes use of proven Hadoop functionality of services such as HDFS, Yarn/MapReduce, HBase, Spark and Kafka. Hadoop, with its inherent extensibility as a primary design goal, is able to deal with the large Terabyte and Petabyte level data sets, allowing missions to extend the types of data stored in ARES and their correlations (such as e.g. storing all TM packet information and parameter values; something missions have repeatedly requested) without being concerned with performance degradation or suitability of the data storage system and backup over the mission lifetime. Using Spark, ARES also provides a generic extensible framework for deploying other algorithms, specifically for evaluation of any spacecraft telemetry data, enabling users to design and perform any data computing on the entire stored data set.

Currently all data provided to ARES is mainly file-based (protobuf or CSV), i.e. batch processing. Recently online streaming capabilities have been added (via Kafka pipelines) however this is not yet widely used in operations. For this reason and for simplicity we opted to have no direct sync at the level of Hadoop but decided to go for a simple mechanism where the same source data is distributed to both MOC and SOC ARES importers via file transfer. ARES receives and archives all operational housekeeping data in HDFS ensuring its veracity, integrity and isolation per mission. It is then processed by Yarn/MapReduce and hosted in HBase, not only for easy access, but also keeping it available for later analysis.

There are 3 main mechanisms to access ARES data:

- REST API (MUSTLink)
- ARES Java API
- Python API (PyARES).

The existing data retrieval/visualization clients therefore make use of one of these interfaces to populate their displays.

Nowadays ARES is the main system in use across ESOC Astronomy and Earth Observation missions, some of the ones already using it are: GAIA, Cluster, BepiColombo, Aeolus, Integral, Solar Orbiter, Seosat, Exomars (TGO), and Estrack (Ground Station Data). Upcoming missions using it are: Euclid, Exomars Rover/Surface Platform (RSP) and Juice.

3 Euclid Data Sources

3.1 Data Sources

Although traditionally only Space generated PUS HKTM parameters were considered data sources for parametric monitoring systems, the sources of possible parametric information in the Euclid mission are many:

- **Space Segment:** The HKTM information is generated from Platform and the two instruments, VIS and NISP. It is transported to ground through PUS packets, that are later processed by the MOC MCS as per the Mission Information database (MIB). Euclid will produce a daily volume of 850 Gb of transmitted data, most of it being Science data in compressed format. Over 20,000 parameters are defined in the Spacecraft and Instrument Database, a significant fraction of them repeating at time periods (periodic HKTM).
- **MOC data processing:** The processing of the packets at MOC generate ancillary information that can be timestamped, including parameter out of limit information,

packet metadata information, Mission Control System packets and events so called synthetic parameters, aggregation or combination of any of the previous sources.

- **MOC ground segment:** The MOC ground segment generates products that contribute to the mission and can be also stamped and parametrized: orbital information and estimates of cold gas, consumables or other information, catalogues uploaded to the on-board attitude sensors, others.
- **SOC ground segment:** Similarly, the SOC ground segment generates data values, mostly in the form of the Operational Sky Survey definition that can be converted into parameters. As well the data generated in the SOC Commanding System SCS.
- **SOC data processing:** The SOC is in charge of generating the Level 1 products, later fed into the SGS, and also performing a Quick Look Analysis (QLA) on them. Both processes generate a wealth of metadata and specially QLA reports, that are a collection of parametric information per instrument exposure describing them in terms of instrument stability and performance.
- **SGS data processing:** On one side the Instrument Operation Team will generate reports, sent to SOC for archiving, that can be source of parameters. But beyond that, the main source of information is the processing of the Level 1 data into scientific products, where metadata on the processing itself and data and metadata on the products generated are persisted in the Euclid Archive System. Section 3.2 elaborates further on this.
- **External data sets:** Euclid data is augmented with external science data sets (other sky surveys), but also can be complemented with ancillary data as star catalogues, space weather information, Solar System ephemerids and other info not directly provided by the mission or its ground segments.

In order to explore the Health and Safety of the instruments the information from all these data sources may be relevant. Previous ESA missions as Soil Moisture and Ocean Salinity (SMOS) have already shown the advantages of allowing cross-matching HKTM data with other sources of data, mainly scientific one but also processing metadata, to allow to track and identify source of systematics in final data but also to unveil trends and performances otherwise not visible.

3.2 Euclid Archive System and Data Volume

The volume of Euclid science data converts the mission in a Big Data science mission. The way to handle this is through the Euclid Archive System (EAS) [7], that acts both as the Data Processing orchestrator of the SGS and also as science archive to support the processing and to expose the data to the external scientific community. The processing of the data happens over different data releases, that increment volume as the survey is collected from sky, and that enhance the processing techniques, but also aim to seek the optimal data products to achieve the cosmological figure of merit of the mission. This leads to an increase of the volume of data archived (Fig. 5), that

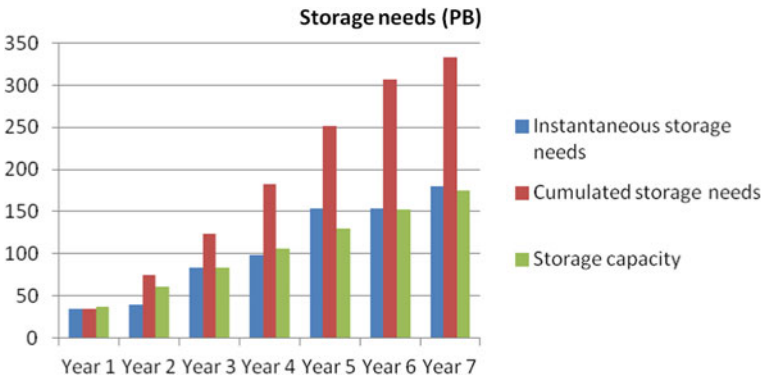


Fig. 5 Storage need related to data volume: Euclid will need the cumulated data along the full mission to conduct the final probe cosmology. Also, several data releases with refined processing are expected. Over 300 PB expected [8]

needs a distributed storage system implemented through EAS Data Storage System, a register of the products generated and coordination of its production through EAS Data Processing System [8], and a component to allow external users to query the scientific results, the EAS Science Archive System. All these systems and multiple processing of the data generate also parametric information linked to the different exposures, and bring relevant information on the instrument performance that could not be obtained through the HKTM or even through a quick look analysis. Due to the huge amount of data involved, not all of it can be used for ingestion in the HMS, so a previous offline assessment of relevance is ongoing to determine which ones will best characterise the system.

4 HMS: System Architecture

4.1 HMS Architecture

Euclid's Health Monitoring System HMS is an element of the SOC that, using the ESOC ARES System as core component, instantiated at ESAC, enhances it to achieve the required use cases. The starting point for the system is to gather as many data sources as defined in Sect. 3 of this paper in order to meet the use cases later described.

A system as HMS needs:

- a way to import data
- data conditioning if required prior to ingestion
- a mechanism for storage
- Mechanisms to provide data to user, through MMI or API.

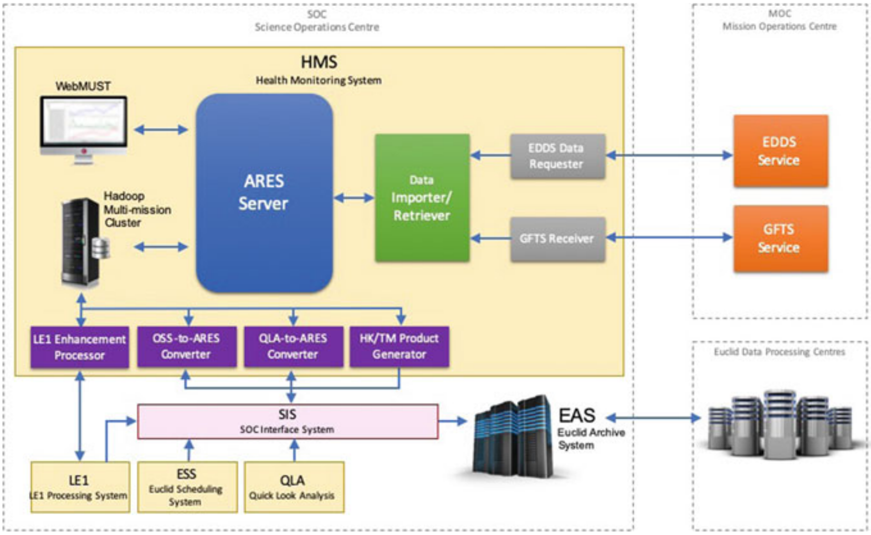


Fig. 6 HMS system architecture, components and interfaces. Diagram represents current state of development

The main requirements are:

- Reliability
- Scalability
- Flexibility
- Friendly User interfaces.

Figure 6 is a representation of the HMS architecture. The system uses a Hadoop backend cluster of physical machines allowing for up to 60 TB of data that can be grown if need be.

An ARES server (ARES@ESAC) deployed in a VM handles the ingestion and data access to the backend, and implements the data services to the clients.

The system ingests data from ESOC. Originally it was intended to implement calls to ESOC EDDS from ARES@ESAC, thus determining the data from the EDDS to be fetched. However, a simpler solution, based on the concept already developed at SMOS was selected, the same response files from EDDS to ARES generated after the calls at ESOC are also sent to the instance of ARES@ESAC. This way both systems are kept synchronized, and the interface becomes naturally unidirectional. If need be, though, the capability to pull data using an EDDS client still exists (covering possible data losses). Data is exchanged using the ESA GFTS service, in the form of protobuf.

Other data sources of data are thereafter acquired. The diagram shows converters for the Operational Sky Survey (OSS) in the form of xml structured data and for the Quick Look Analysis (QLA) JSON reports, already implemented and tested. In order to exchange the data, the SOC Interface System provides stubs to the external systems that connect them with the dedicated conversion tools. These converters perform a simple conversion of data to a *{param, timestamp, value}* csv file, and deposit it in an *InTray* folder of the ARES server. There is no need to declare or define in the ARES system the parameters, thus allowing to grow the data sets with new ones without any extra effort. The parameter description can be adjusted at any time. This simplicity is the basis of the intrinsic power. Parameters can have several types (integers, reals strings, Booleans).

Figure 6 also shows the connection to the Euclid Archive System (EAS), that will allow getting data from the SGS processing. This allows to inject into the system both the information about the processing algorithms and calibration values used in the processing of any given exposure, as well as scientific data extraction from each one (detector characteristics, PSF, galaxy counts, shears, other).

Finally, two components, the LE1 Enhancement Processor and the HKTM Product Generator implement the use cases where the HMS is used not only as a trend analysis and visualization tool, but also as a data generator tool for Euclid LE1.

4.2 Client Applications

The traditional data access system at ESOC was MUST. MUST pulls the data from OPS LAN into a local MySQL database. With the introduction of ARES, some MUST data clients have adapted to be compatible with the newer missions that are supported by ARES, where data extraction can make use of the former MUST web-interface, WebMUST [6] through MUSTLink. This is the baseline client to be used in Euclid, as a web based client that allows data extraction and visualization, as shown in Fig. 7, together with other tools for data correlation and analysis.

For new missions, like Euclid, the analysis of huge datasets of H/K TM data, as well as other information stored in the ARES System, made necessary to use a direct way of accessing the ARES data to perform calculations. The original motivation for this was the need for Machine Learning groups to get direct access to the data, and to execute their algorithms as close to the data as possible. So, being Python the preferred computer language to perform this kind of analysis, ESOC developed the PyARES API, that uses Python to give the user access to the ARES data directly, as well as access to the results from their data processing (see Fig. 8), where the result data comes from custom data pre-processing using the Spark component of the ARES Hadoop infrastructure.

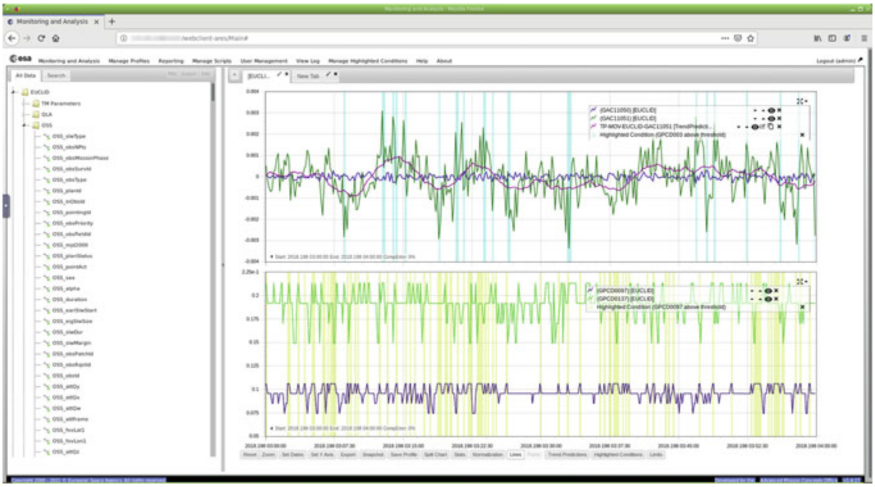
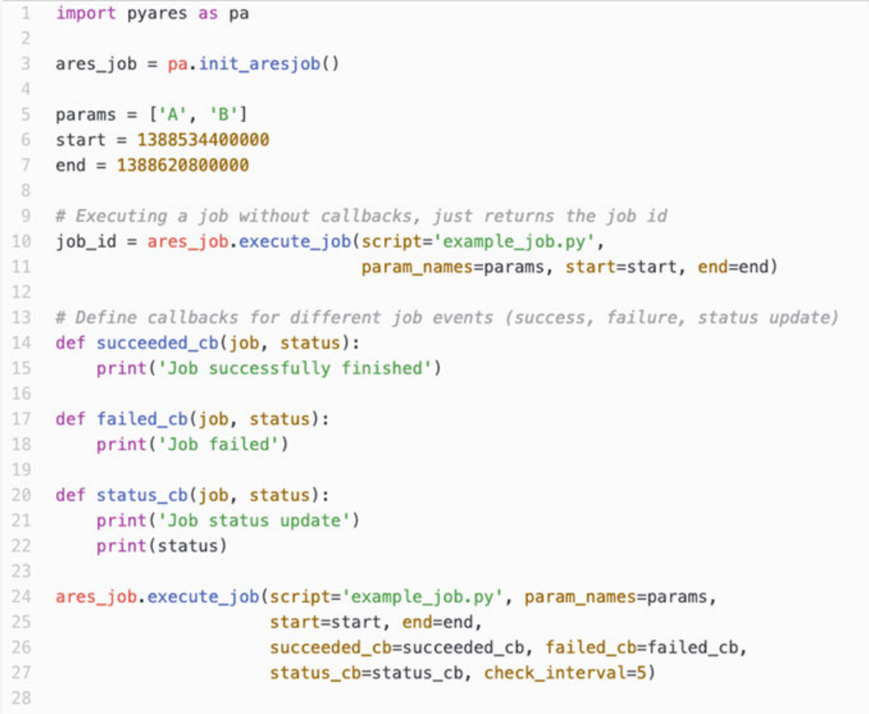
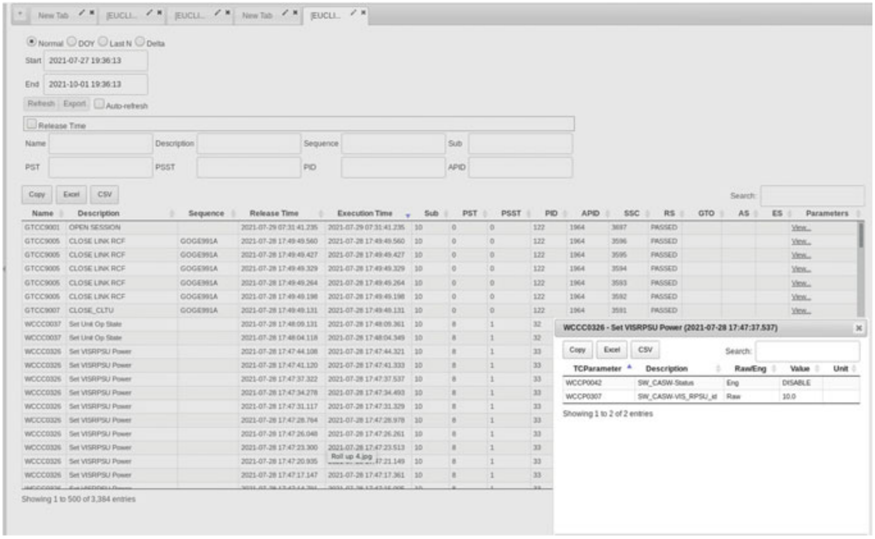


Fig. 7 WebMUST main screen showing the evolution of several parameters (Gaia mission data used for HKTM) and grouping of parameters showing other source parameters as Survey (OSS) on the left pane



```python
import pyares as pa

ares_job = pa.init_aresjob()

params = ['A', 'B']
start = 1388534400000
end = 1388620800000

# Executing a job without callbacks, just returns the job id
job_id = ares_job.execute_job(script='example_job.py',
                              param_names=params, start=start, end=end)

# Define callbacks for different job events (success, failure, status update)
def succeeded_cb(job, status):
    print('Job successfully finished')

def failed_cb(job, status):
    print('Job failed')

def status_cb(job, status):
    print('Job status update')
    print(status)

ares_job.execute_job(script='example_job.py', param_names=params,
                     start=start, end=end,
                     succeeded_cb=succeeded_cb, failed_cb=failed_cb,
                     status_cb=status_cb, check_interval=5)

```

Fig. 8 Example of Python code to execute an ARES job in the cluster



Start 2021-07-27 19:36:13

End 2021-10-01 19:36:13

Name	Description	Sequence	Release Time	Execution Time	Sub	PST	PSST	PID	APID	SSC	RS	GTO	AS	ES	Parameters
GTCC9001	OPEN SESSION		2021-07-29 07:31:41.235	2021-07-29 07:31:41.235	10	0	0	122	1964	3597	PASSED				View...
GTCC9005	CLOSE LINK RCF	GOGE991A	2021-07-28 17:49:49.560	2021-07-28 17:49:49.560	10	0	0	122	1964	3596	PASSED				View...
GTCC9005	CLOSE LINK RCF	GOGE991A	2021-07-28 17:49:49.427	2021-07-28 17:49:49.427	10	0	0	122	1964	3595	PASSED				View...
GTCC9005	CLOSE LINK RCF	GOGE991A	2021-07-28 17:49:49.329	2021-07-28 17:49:49.329	10	0	0	122	1964	3594	PASSED				View...
GTCC9005	CLOSE LINK RCF	GOGE991A	2021-07-28 17:49:49.264	2021-07-28 17:49:49.264	10	0	0	122	1964	3593	PASSED				View...
GTCC9005	CLOSE LINK RCF	GOGE991A	2021-07-28 17:49:49.198	2021-07-28 17:49:49.198	10	0	0	122	1964	3592	PASSED				View...
GTCC9007	CLOSE_CLTU	GOGE991A	2021-07-28 17:49:49.131	2021-07-28 17:49:49.131	10	0	0	122	1964	3591	PASSED				View...
WCCC0037	Set Unit Op State		2021-07-28 17:48:09.131	2021-07-28 17:48:09.361	10	8	1	32							
WCCC0037	Set Unit Op State		2021-07-28 17:48:04.118	2021-07-28 17:48:04.349	10	8	1	32							
WCCC0326	Set VISRPSU Power		2021-07-28 17:47:44.108	2021-07-28 17:47:44.321	10	8	1	33							
WCCC0326	Set VISRPSU Power		2021-07-28 17:47:41.120	2021-07-28 17:47:41.333	10	8	1	33							
WCCC0326	Set VISRPSU Power		2021-07-28 17:47:37.322	2021-07-28 17:47:37.537	10	8	1	33							
WCCC0326	Set VISRPSU Power		2021-07-28 17:47:34.278	2021-07-28 17:47:34.493	10	8	1	33							
WCCC0326	Set VISRPSU Power		2021-07-28 17:47:31.117	2021-07-28 17:47:31.329	10	8	1	33							
WCCC0326	Set VISRPSU Power		2021-07-28 17:47:28.764	2021-07-28 17:47:28.978	10	8	1	33							
WCCC0326	Set VISRPSU Power		2021-07-28 17:47:26.048	2021-07-28 17:47:26.261	10	8	1	33							
WCCC0326	Set VISRPSU Power		2021-07-28 17:47:23.300	2021-07-28 17:47:23.513	10	8	1	33							
WCCC0326	Set VISRPSU Power		2021-07-28 17:47:20.935	Roll up 4.jpg 17:47:21.149	10	8	1	33							
WCCC0326	Set VISRPSU Power		2021-07-28 17:47:17.147	2021-07-28 17:47:17.361	10	8	1	33							

Showing 1 to 500 of 3,384 entries

WCCC0326 - Set VISRPSU Power (2021-07-28 17:47:37.537)

TCParameter	Description	Raw/Eng	Value	Unit
WCCP0042	SW_CASW-Status	Eng	DISABLE	
WCCP0307	SW_CASW-VIS_RPSU_id	Raw	10.0	

Showing 1 to 2 of 2 entries

Fig. 9 Telecommand view in the WebMUST client showing TCs from the Euclid System Validation Test 1.2, including details of the contained telecommand parameters of one instance and uplink status

5 Euclid Use Cases

5.1 HKTM Data Analysis

The main use case for using a system like ARES and WebMUST as our Health Monitoring System is the capability of analysing and correlating a number of HKTM parameters, in order to diagnose or detect problems in the spacecraft, the payload or the scientific data, and trace back their origin. For this, the parametrized HKTM data, as the result of the processing of the raw HK/TM received at MOC, are cloned into our SOC HMS. In case this process fails for any reason, there is still the backup solution of fetching the desired data from the EDDS Service. In any case, these data will be imported into the multi-mission cluster by the ARES Server. A WebMUST client is connected to the ARES Server and the Hadoop cluster to provide a full-featured client to perform these analyses. The HKTM data analysis can be also complemented by analysis of other data elements available at ARES as Telecommand information or TM packet information, as shown in Fig. 9.

5.2 OSS Data Ingestion

In addition to the HKTM parameters received from MOC, the ARES system is capable of storing, analysing and correlating any set of parameters that we ingest in

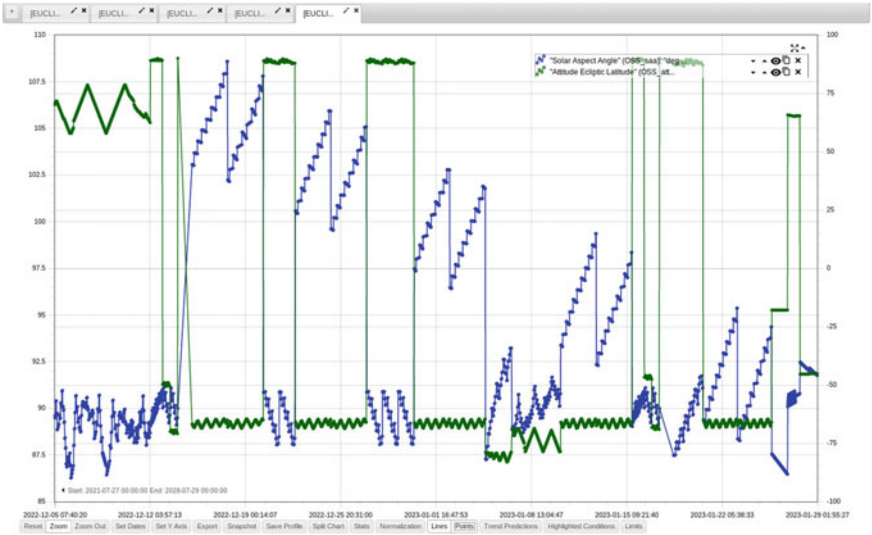


Fig. 10 Solar Aspect Angle versus Ecliptic Latitude of the pointings of the Euclid Survey over 60 days showing different matching features. Each data point corresponds to one observation dither pointing

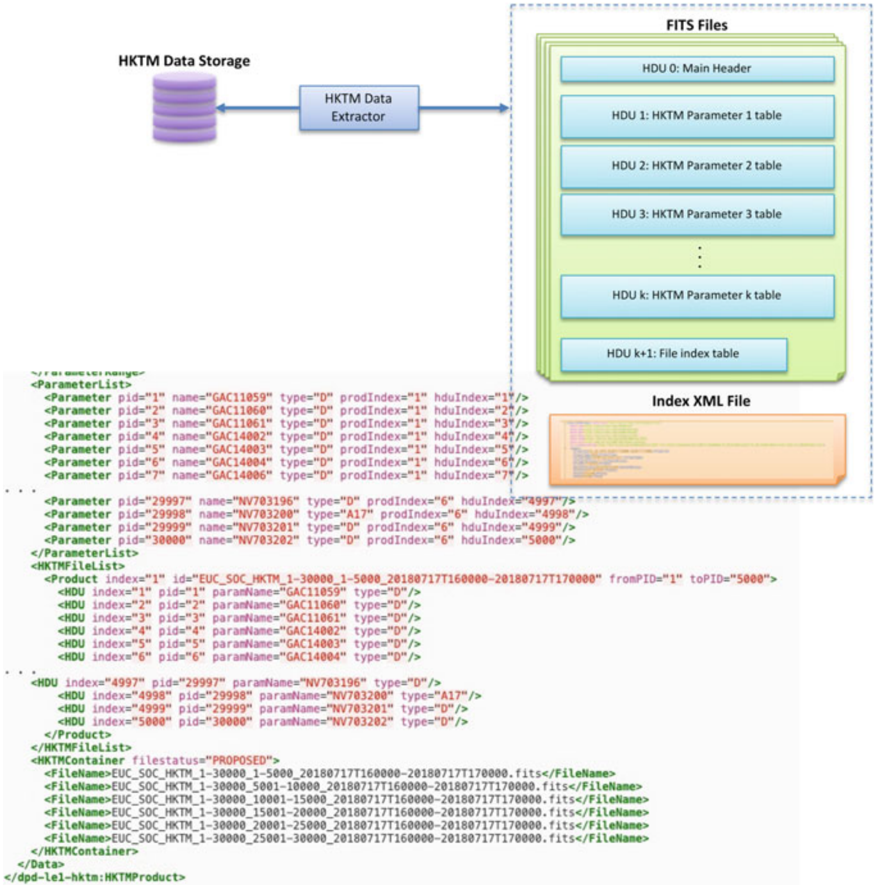
the multi-mission cluster. One of the data sets that we are interested in having available to be able to correlate with the obtained HK/TM information is the Operational Sky Survey (OSS). The OSS is generated by the Euclid Scheduling System (ESS) at SOC as an xml file, upon the reception of a Reference Survey Definition or after any re-planning activities. This SOC product is stored at the EAS, to make it available to all the Processing Functions. From the OSS, we can extract all the pointing positions and conditions, and store this information at the HMS, as shown in Fig. 10. This is done by the OSS to ARES converter.

5.3 QLA Metrics Ingestion

A similar use case can be found with all the parameters resulting from the Quick-Look Analysis of the Level 1 products. The reports generated by the Quick-Look Analysis subsystem (QLA) are in the form of hierarchized JSON files. These reports are structured by product, including analysis and different metrics for each exposure image derived from algorithms run over science data. These files are processed by the so called "flattener", or QLA-to-ARES converter, and the resulting dataset is ingested into the ARES system at the HMS. This process allows us to correlate the resulting metrics from the science data images with the corresponding HK/TM parameters (and also with the survey information at the epoch of the data acquisition).

5.4 HKTM Data Product Generation

The HKTM data is required for some upper level processing pipelines. Since the Euclid data processing is highly de-centralised and distributed, it would not be efficient for those pipelines requiring HK/TM to ask for these data to the SOC HKTM data repository (the HMS). For that purpose, a synthetic product would be built from the last HKTM received data, covering a time span of one hour (to be configured). This data will be generated by the so-called HKTM Product Generator, which will extract directly these data from the multi-mission cluster and will produce FITS files with these data in form of table extensions as seen in Fig. 11. These generated products will be then ingested into the Euclid Archive System (EAS), like any other data


HKTM Data Storage
HKTM Data Extractor
FITS Files
HDU 0: Main Header
HDU 1: HKTM Parameter 1 table
HDU 2: HKTM Parameter 2 table
HDU 3: HKTM Parameter 3 table
HDU k: HKTM Parameter k table
HDU k+1: File index table
Index XML File

```
    </ParameterRange>
    <ParameterList>
      <Parameter pid="1" name="GAC11059" type="D" prodIndex="1" hduIndex="1"/>
      <Parameter pid="2" name="GAC11060" type="D" prodIndex="1" hduIndex="2"/>
      <Parameter pid="3" name="GAC11061" type="D" prodIndex="1" hduIndex="3"/>
      <Parameter pid="4" name="GAC14002" type="D" prodIndex="1" hduIndex="4"/>
      <Parameter pid="5" name="GAC14003" type="D" prodIndex="1" hduIndex="5"/>
      <Parameter pid="6" name="GAC14004" type="D" prodIndex="1" hduIndex="6"/>
      <Parameter pid="7" name="GAC14006" type="D" prodIndex="1" hduIndex="7"/>
. . .
      <Parameter pid="29997" name="NV703196" type="D" prodIndex="6" hduIndex="4997"/>
      <Parameter pid="29998" name="NV703200" type="A17" prodIndex="6" hduIndex="4998"/>
      <Parameter pid="29999" name="NV703201" type="D" prodIndex="6" hduIndex="4999"/>
      <Parameter pid="30000" name="NV703202" type="D" prodIndex="6" hduIndex="5000"/>
    </ParameterList>
    <HKTMFileList>
      <Product index="1" id="EUC_SOC_HKTM_1-30000_1-5000_20180717T160000-20180717T170000" fromPID="1" toPID="5000">
        <HDU index="1" pid="1" paramName="GAC11059" type="D"/>
        <HDU index="2" pid="2" paramName="GAC11060" type="D"/>
        <HDU index="3" pid="3" paramName="GAC11061" type="D"/>
        <HDU index="4" pid="4" paramName="GAC14002" type="D"/>
        <HDU index="5" pid="5" paramName="GAC14003" type="D"/>
        <HDU index="6" pid="6" paramName="GAC14004" type="D"/>
. . .
    <HDU index="4997" pid="29997" paramName="NV703196" type="D"/>
        <HDU index="4998" pid="29998" paramName="NV703200" type="A17"/>
        <HDU index="4999" pid="29999" paramName="NV703201" type="D"/>
        <HDU index="5000" pid="30000" paramName="NV703202" type="D"/>
      </Product>
    </HKTMFileList>
    <HKTMContainer filestatus="PROPOSED">
      <FileName>EUC_SOC_HKTM_1-30000_1-5000_20180717T160000-20180717T170000.fits</FileName>
      <FileName>EUC_SOC_HKTM_1-30000_5001-10000_20180717T160000-20180717T170000.fits</FileName>
      <FileName>EUC_SOC_HKTM_1-30000_10001-15000_20180717T160000-20180717T170000.fits</FileName>
      <FileName>EUC_SOC_HKTM_1-30000_15001-20000_20180717T160000-20180717T170000.fits</FileName>
      <FileName>EUC_SOC_HKTM_1-30000_20001-25000_20180717T160000-20180717T170000.fits</FileName>
      <FileName>EUC_SOC_HKTM_1-30000_25001-30000_20180717T160000-20180717T170000.fits</FileName>
    </HKTMContainer>
  </Data>
</dpd-le1-hktm:HKTMProduct>
```

Fig. 11 HKTM product generation. Full HKTM is provided to the SGS as a FITS product, aggregating all HKTM cut per hour period. This allows later use decoupled from any Telemetry provision service

product in Euclid, where the data processing centres running the different pipelines for their corresponding Processing Functions (PFs) can retrieve them, as well as store new data products, results of the execution of those pipelines.

5.5 Enhancement of LE1 Data

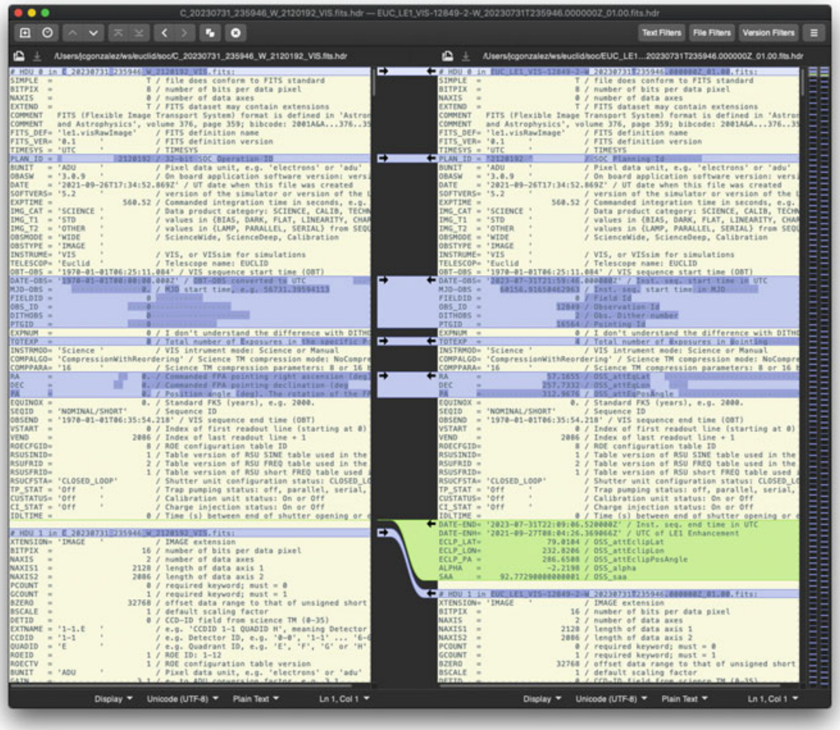
The processing functions in charge of the generation of the Level 2 products for VIS and NISP (photometry and spectrography) need also some information from the HKTM and other data sets (OSS). They have, as main input, the Level 1 product. The metadata is typically used to support querying of products from the EAS for SGS processing, but sometimes also to provide parameters needed in the processing algorithm. Even if the source data (HKTM product or OSS) is available, presenting this as an enhancement helps to ease the upper level design effort. Therefore, the preliminary LE1 products suffer an additional post-processing step: the so called "enhancement" process, performed by the LE1 Enhancement Processor (also known as Auxiliary Processor), illustrated in Fig. 12. In this "enhancement" process, the LE1 product just generated is passed to the LE1 Enhancement Processor, which in turn requests to the multi-mission cluster the additional metadata required by the LE2 Processors (and, potentially, any other higher-level PF), and embeds these metadata into the LE1 product, generating the so-called LE1 Enhanced Product. The enhanced data can be direct retrieval of HMS samples, or complex aggregations, statistical functions or combination of multiple parameters. These products are then stored into the EAS, to make them available to any processing centre for their retrieval.

6 Cross Mission Uses and Evolution

The backend of HMS implemented through the Hadoop cluster is shared with BepiColombo, since mission data can be split making used of name spaces in the Hadoop ecosystem.

6.1 BepiColombo Use Case

ESA's BepiColombo mission to Mercury Science Ground Segment (SGS) [9] is responsible for the following downlink activities: data acquisition, data processing, quick-look analysis and archiving. The system is developed by the SGS group internally but sometimes is completed by third party tools already exercised in other satellite missions. The SGS downlink system consists of the following main functions:



C_20230731_235946_W_2120192_VIS.fits.hdr — EUC_LE1_VIS-12849-2-W_20230731T235946.000000Z_01.00.fits.hdr

/Users/jcgonzalez/ws/euclid/soc/C_20230731_235946_W_2120192_VIS.fits.hdr

```
# HDU 0 in C_20230731_235946_W_2120192_VIS.fits:
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                    8 / number of bits per data pixel
NAXIS   =                    0 / number of data axes
EXTEND  =                    T / FITS dataset may contain extensions
COMMENT   FITS (Flexible Image Transport System) format is defined in 'Astron
COMMENT   and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..35
FITS_DEF= 'le1.visRawImage'    / FITS definition name
FITS_VER= '0.1     '           / FITS definition version
TIMESYS = 'UTC     '           / TIMESYS
PLAN_ID =              2120192 / 32-bit SOC Operation ID
BUNIT   = 'ADU     '           / Pixel data unit, e.g. 'electrons' or 'adu'
OBASW   = '3.0.9   '           / On board application software version: versi
DATE    = '2021-09-26T17:34:52.869Z' / UT date when this file was created
SOFTVERS= '5.2     '           / version of the simulator or version of the
EXPTIME =               560.52 / Commanded integration time in seconds, e.g.
IMG_CAT = 'SCIENCE '           / Data product category: SCIENCE, CALIB, TECHN
IMG_T1  = 'STD     '           / values in {BIAS, DARK, FLAT, LINEARITY, CHAR
IMG_T2  = 'OTHER   '           / values in {LAMP, PARALLEL, SERIAL} from SEQU
OBSMODE = 'WIDE    '           / ScienceWide, ScienceDeep, Calibration
OBSTYPE = 'IMAGE   '
INSTRUME= 'VIS     '           / VIS, or VISsim for simulations
TELESCOP= 'Euclid  '           / Telescope name: EUCLID
OBT-OBS = '1970-01-01T06:25:11.084' / VIS sequence start time (OBT)
DATE-OBS= '1970-01-01T00:00:00.000Z' / OBT-OBS converted to UTC
MJD-OBS =                   0. / MJD start time, e.g. 56731.39594113
FIELDID =                    0
OBS_ID  =                    0
DITHOBS =                    0
PTGID   =                    0
EXPNUM  =                    0 / I don't understand the difference with DITHO
TOTEXP  =                    0 / Total number of Exposures in the specific Po
INSTRMOD= 'Science '           / VIS intrument mode: Science or Manual
COMPALGO= 'CompressionWithReordering' / Science TM compression mode: NoCompre
COMPPARA= '16      '           / Science TM compression parameters: 8 or 16 b
RA      =                   0. / Commanded FPA pointing right ascension (deg)
DEC     =                   0. / Commanded FPA pointing declination (deg
PA      =                   0. / Position angle (deg). The rotation of the FP
EQUINOX =                   0. / Standard FK5 (years), e.g. 2000.
SEQID   = 'NOMINAL/SHORT'      / Sequence ID
OBSEND  = '1970-01-01T06:35:54.218' / VIS sequence end time (OBT)
VSTART  =                    0 / Index of first readout line (starting at 0)
VEND    =                 2086 / Index of last readout line + 1
ROECFGID=                    8 / ROE configuration table ID
RSUSINID=                    1 / Table version of RSU SINE table used in the
RSUFRID =                    2 / Table version of RSU FREQ table used in the
RSUSFRID=                    1 / Table version of RSU short FREQ table used i
RSUCFSTA= 'CLOSED_LOOP'        / Shutter unit configuration status: CLOSED_LO
TP_STAT = 'Off     '           / Trap pumping status: off, parallel, serial,
CUSTATUS= 'Off     '           / Calibration unit status: On or Off
CI_STAT = 'Off     '           / Charge injection status: On or Off
IDLTIME =                    0 / Time (s) between end of shutter opening or e

# HDU 1 in C_20230731_235946_W_2120192_VIS.fits:
XTENSION= 'IMAGE   '           / IMAGE extension
BITPIX  =                   16 / number of bits per data pixel
NAXIS   =                    2 / number of data axes
NAXIS1  =                 2128 / length of data axis 1
NAXIS2  =                 2086 / length of data axis 2
PCOUNT  =                    0 / required keyword; must = 0
GCOUNT  =                    1 / required keyword; must = 1
BZERO   =                32768 / offset data range to that of unsigned short
BSCALE  =                    1 / default scaling factor
DETID   =                    0 / CCD-ID field from science TM (0-35)
EXTNAME = '1-1.E   '           / e.g. 'CCDID 1-1 QUADID H', meaning Detector
CCDID   = '1-1     '           / e.g. Detector ID, e.g. '0-0', '1-1' ... '6-6
QUADID  = 'E       '           / e.g. Quadrant ID, e.g. 'E', 'F', 'G' or 'H'
ROEID   =                    1 / ROE ID: 1-12
ROECTV  =                    1 / ROE configuration table version
BUNIT   = 'ADU     '           / Pixel data unit, e.g. 'electrons' or 'adu'
```

/Users/jcgonzalez/ws/euclid/soc/EUC_LE1...20230731T235946.000000Z_01.00.fits.hdr

```
# HDU 0 in EUC_LE1_VIS-12849-2-W_20230731T235946.000000Z_01.00.fits:
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                    8 / number of bits per data pixel
NAXIS   =                    0 / number of data axes
EXTEND  =                    T / FITS dataset may contain extensions
COMMENT   FITS (Flexible Image Transport System) format is defined in 'Astron
COMMENT   and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..35
FITS_DEF= 'le1.visRawImage'    / FITS definition name
FITS_VER= '0.1     '           / FITS definition version
TIMESYS = 'UTC     '           / TIMESYS
PLAN_ID = '2120192 '           / SOC Planning Id
BUNIT   = 'ADU     '           / Pixel data unit, e.g. 'electrons' or 'adu'
OBASW   = '3.0.9   '           / On board application software version: versi
DATE    = '2021-09-26T17:34:52.869Z' / UT date when this file was created
SOFTVERS= '5.2     '           / version of the simulator or version of the
EXPTIME =               560.52 / Commanded integration time in seconds, e.g.
IMG_CAT = 'SCIENCE '           / Data product category: SCIENCE, CALIB, TECHN
IMG_T1  = 'STD     '           / values in {BIAS, DARK, FLAT, LINEARITY, CHAR
IMG_T2  = 'OTHER   '           / values in {LAMP, PARALLEL, SERIAL} from SEQU
OBSMODE = 'WIDE    '           / ScienceWide, ScienceDeep, Calibration
OBSTYPE = 'IMAGE   '
INSTRUME= 'VIS     '           / VIS, or VISsim for simulations
TELESCOP= 'Euclid  '           / Telescope name: EUCLID
OBT-OBS = '1970-01-01T06:25:11.084' / VIS sequence start time (OBT)
DATE-OBS= '2023-07-31T21:59:46.000000Z' / Inst. seq. start time in UTC
MJD-OBS =     60156.9165046296 / Inst. seq. start time in MJD
FIELDID =                    0 / Field Id
OBS_ID  =                12849 / Observation Id
DITHOBS =                    2 / Obs. Dither number
PTGID   =                16564 / Pointing Id
EXPNUM  =                    0 / I don't understand the difference with DITHO
TOTEXP  =                    4 / Total number of Exposures in Pointing
INSTRMOD= 'Science '           / VIS intrument mode: Science or Manual
COMPALGO= 'CompressionWithReordering' / Science TM compression mode: NoCompre
COMPPARA= '16      '           / Science TM compression parameters: 8 or 16 b
RA      =              57.1655 / OSS_attEqLat
DEC     =             257.7332 / OSS_attEqLon
PA      =             312.9676 / OSS_attEqPosAngle
EQUINOX =                   0. / Standard FK5 (years), e.g. 2000.
SEQID   = 'NOMINAL/SHORT'      / Sequence ID
OBSEND  = '1970-01-01T06:35:54.218' / VIS sequence end time (OBT)
VSTART  =                    0 / Index of first readout line (starting at 0)
VEND    =                 2086 / Index of last readout line + 1
ROECFGID=                    8 / ROE configuration table ID
RSUSINID=                    1 / Table version of RSU SINE table used in the
RSUFRID =                    2 / Table version of RSU FREQ table used in the
RSUSFRID=                    1 / Table version of RSU short FREQ table used i
RSUCFSTA= 'CLOSED_LOOP'        / Shutter unit configuration status: CLOSED_LO
TP_STAT = 'Off     '           / Trap pumping status: off, parallel, serial,
CUSTATUS= 'Off     '           / Calibration unit status: On or Off
CI_STAT = 'Off     '           / Charge injection status: On or Off
IDLTIME =                    0 / Time (s) between end of shutter opening or e
DATE-END= '2023-07-31T22:09:06.520000Z' / Inst. seq. end time in UTC
DATE-ENH= '2021-09-27T08:04:26.369066Z' / UTC of LE1 Enhancement
ECLP_LAT=              79.0104 / OSS_attEclipLat
ECLP_LON=             232.8206 / OSS_attEclipLon
ECLP_PA =             286.6508 / OSS_attEclipPosAngle
ALPHA   =              -2.2198 / OSS_alpha
SAA     =    92.77290000000001 / OSS_saa

# HDU 1 in EUC_LE1_VIS-12849-2-W_20230731T235946.000000Z_01.00.fits:
XTENSION= 'IMAGE   '           / IMAGE extension
BITPIX  =                   16 / number of bits per data pixel
NAXIS   =                    2 / number of data axes
NAXIS1  =                 2128 / length of data axis 1
NAXIS2  =                 2086 / length of data axis 2
PCOUNT  =                    0 / required keyword; must = 0
GCOUNT  =                    1 / required keyword; must = 1
BZERO   =                32768 / offset data range to that of unsigned short
BSCALE  =                    1 / default scaling factor
```

Fig. 12 LE1 product enhancement for a VIS product. The original metadata (left) is enhanced by filling in empty values (in blue), or inserting new ones not previously existing (green)

- Data Acquisition, Storage & Dissemination
- Parameter Storage & Dissemination
- Data Processing
- Quick-Look Analysis
- Monitoring & Control
- Data Distribution.

The first two points are related with the possibility to persists satellite data. In the last years many requirements from users are addressing the importance to have raw and processed data available in a short period. The amount of data is also increasing due to the improvement of data transmission and the use of data compression. Old solutions, like simple relational databases used as repositories for all data are becoming difficult to maintain. Distributed solutions are matching better such type of data domain and the solution offered by ESOC (Cloudera + ARES) is currently a very adaptive and efficient solution.

The infrastructure is based on Cloudera. The infrastructure is quite complex and is composed by HW and SW. After some attempts to use virtual machines it was decided to use physical machines to have good performances and enough resources for handling data coming from satellite missions (the idea is to use the same infrastructure for multi-mission activity). Besides the HW part the infrastructure contains a COTS SW called CDH that is Cloudera's software distribution containing Apache Hadoop and related projects (HIVE, HDFS, Zookeeper etc.). More machines can be added to the infrastructure for increasing the disk space or to resize the cluster. All these components are independent from the satellite data missions we want to manage with this infrastructure for this reason a domain specific software was developed. It provides an easy interface and defines the metadata to join infrastructure and satellite mission's needs. ARES is the software developed at ESA mounted on top of the infrastructure. As said previously, ARES is a sort of interface to the infrastructure that allows the data to be saved according to a certain format, and retrieved and displayed in different ways (ANDs, Graphs etc.). All this is visible in Fig. 13.

The chance to have a scalable distributed storage and processing capabilities (via Spark, as an example) offered the opportunity to think about the migration of some other pieces of mission information inside the same infrastructure. Currently BepiColombo SOC is analysing the costs/advantages to use the infrastructure as repository for the science products, calibration files etc. as described in Fig. 14.

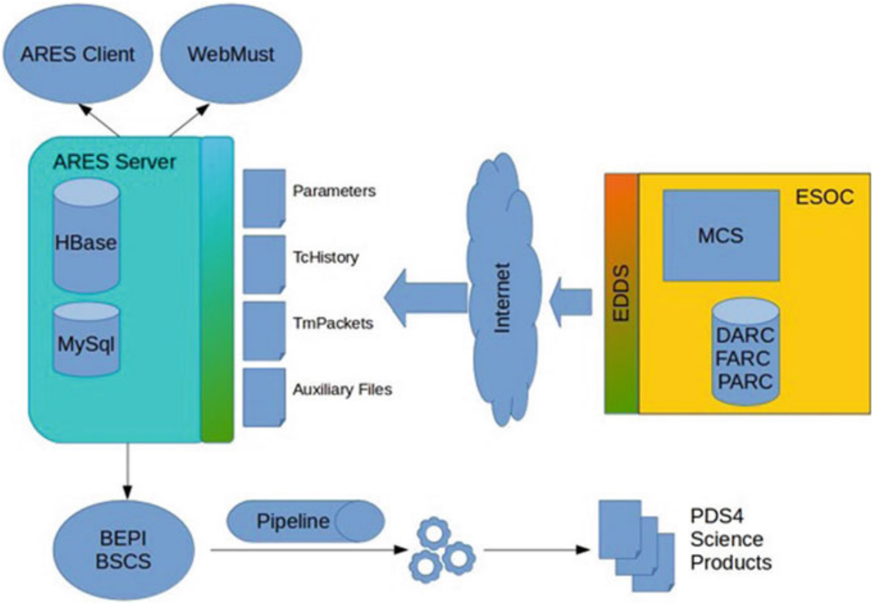


Fig. 13 BepiColombo ARES infrastructure, connecting with BepiColombo Science Operations Control System (BSCS) and getting data from ESOC

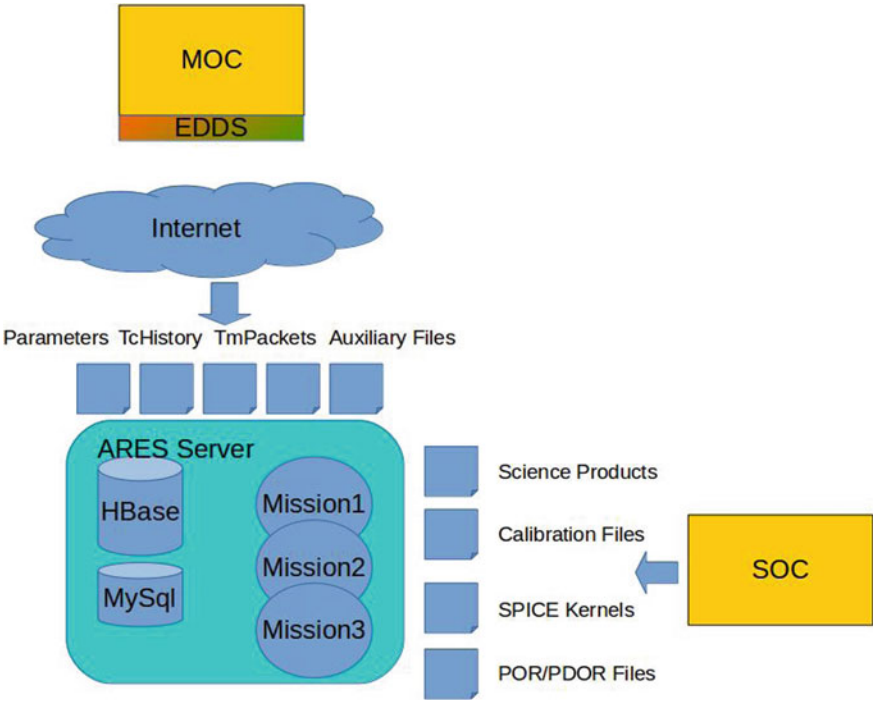


Fig. 14 Different BepiColombo products archived in a shared infrastructure based on the same backend Hadoop cluster

6.2 *Other Missions Use Cases*

XMM Newton is an X-Ray Space observatory flying for more than 20 years. Its systems are aged, and a rejuvenation of the overall Science Ground Segment is being considered while the mission is operational. One task is to gather all parametric sources produced by many different systems (operating systems, languages, data model definitions) into one single system to evaluate its the advantage. XMM has made use of a test installation of HMS for Euclid based on Virtual Machines as Server and Backend, and has been able to conduct a demonstration of the concept in just few days, with minimal effort related to the conversion of data to ARES input files. The XMM-Newton SOC will start migration to ARES from the legacy system during 2022. This will support instrument monitoring at ESAC and by the instrument teams across Europe.

PLATO mission, in development and devoted to the search for exoplanets, is also considering the same approach towards a Health Monitoring System that will allow mixing different sources of data into one common database.

6.3 Enhancements and Future Work

A future evolution for Euclid is the exploitation of the Spark jobs that can be run inside the cluster, thus avoiding this in the external environment. The most evident one is the use case for HKTM product generation, where the creation of FITS files from the time selection of the full HKTM data set can be scheduled and run inside the cluster, and obtain the final product, rather than having to query data and perform the product build externally. Other type of jobs as creation of composed parameters or derived products is under consideration, depending on the results of the system tests being conducted with the Space component and full ground segment. Euclid System Operations Verification Tests are conducted along Q4 of 2021 and Q3 of 2022 and will provide multiple data sets that will allow further insight on capabilities.

Further, it will be interesting to explore the access to the wealth of parametric data in the Euclid Archive System, and perform construction of composed synthetic parameters from the science processing that could give better insight into instrument performance. There are several methods to perform the retrieval of Euclid data products from the EAS. The simplest one is the use of the online web tool DbView. From this tool, once logged on, the user is presented with a page where the tables to explore in the EAS can be selected. By submitting a query request, a table with all the possible Data Products (DPDs) fulfilling the defined criteria is presented. Upon selection of any of those DPDs, the user can get the full list of metadata corresponding to that DPD, or even download the associated data files (if any). A second way to retrieve data products is by using a REST interface designed for that purpose. Finally, a third way to retrieve data products is by using the Metadata Access Layer query capabilities built inside the EAS infrastructure. This is composed by a series of methods and rules in Python, that can be used both for retrieving automatically a series of DPDs, or by an interactive tool that allows the user to specify any selection query to retrieve whatever DPDs fulfil the query.

7 Conclusions

This paper has shown how synergies across two ESA Directorates dealing with operations, Operations (OPS) and Science (SCI), have led to an enhanced use and capabilities of an existing system. Fostering the integration of multiple source of parametric information in a common system enhances the findings of hidden correlations and performances of a space system, specially merging the engineering and scientific domains, and enabling the systems like the HMS based in ARES to conduct other use cases as part of the data processing ground segment. Furthermore, the infrastructure is easily adaptable to be used by other projects, leading to potential synergies and cross mission data exploitation.

References

1. Racca et al (2016) The euclid mission design. Proc SPIE 9904:99040O
2. Buenadicha G, Gómez P, Hoar J, Schwartz J, Scaramella R, Amieux J (2016) Pointing to a survey of pointings: euclid science mission planning. SpaceOps Proc 2016
3. Santos R (2014) How the use of “Big Data” clusters improves off-line data analysis and operations. In: Proceedings of the 13th international conference on space operations (SpaceOps)
4. Santos R, Marques G, Eggleston J (2017) Combining small housekeeping data lakes into a shared big data infrastructure at ESOC-achievements and future evolution. In: Proceedings of the big data from space (BiDS)
5. Montroni G, Pantoquilho M, Santos R (2016) Big data in mission operations, the ExoMars 2016 experience. In: Proceedings of the AIAA Space 2016
6. da Silva J, Donati A (2016) WebMUST evolution. SpaceOps 2016
7. Nieto S, Teodoro P, Giordano F, Racero E, Fernandez M, Noiret D, Salgado J, Altieri B, Merin B, Arviset C (2018) Science exploitation in a big data archive: the euclid scientific archive system. Astron Data Anal Softw Syst (ADASS) (Published)
8. Buenadicha G, Poncet M, Dabin C, Metge JJ, Noddle K, Holliman M, Melchior M, Belikov A, Koppenhoefer J (2015) On behalf of euclid science ground segment system team. In: Euclid: big data from ‘dark’ space, Big Data Spain, 15-16 Oct 2015. Madrid
9. Montagnon E, Budnik F, Casale M, de la Fuente S, Martinez S, Murakami G, Ogawa M, Seki T, Steiger C, Yamashita M (2021) BepiColombo ground segment and mission operations. Space Sci Rev. https://doi.org/10.1007/s11214-021-00805-y

New Questions Opened by the Big Data in the World of the Science Data Processing Centre for Gaia Mission in CNES

Julie Guiraud and Wilhem Roux

Abstract The 16th of July 2019, the ESA's Gaia satellite started his first mission extension after 5 years of producing operational observations (since 25th of July 2014). This mission is the successor of Hipparcos ESA's satellite with the same objective of publishing a catalogue of stars and objects (galaxies, asteroids, etc.) but up to 1 billion objects (against 2.5 million). Gaia catalogue will determine the position, the distance and the movement of each object. To achieve this goal, a consortium, called DPAC, has been created to process all the satellite's data composed of more than 450 people mostly in Europe (including scientists and engineers). 9 Coordination Units (CU) corresponding to dedicated themes and 6 data processing centres (DPCs) have been created. CNES is in charge of 3 scientific CUs (with 7 scientific pipelines) in operations, called DPCC. CNES is in DPAC an important DPC. The first catalogue has been released in September 2016, based on the first year of Gaia observations (2014/2015). The second catalogue has been published on the 25th of April 2018. Over one billion of sources have been processed. Operations are ongoing to prepare the third version of the catalogue (with an early release containing astrometric positions and photometry published on the 3rd of December 2020). This new version of the catalogue will include three years of Gaia observations. All CUs will produce data and the existing scientific algorithms will be improved. This catalogue will be available for the entire scientific community in 2021. DPCC will manage the processing of all delegated CUs for the first time, meanwhile performing daily operations, in a limited time.

On technical point of view, to fulfil the performance constraints, DPCC has chosen in 2010 a solution based on Hadoop technology, emerging from the internet applications such as Facebook or eBay thereby entering the "Big Data" world to process Gaia space observatory data.

On organization point of view, as CNES is processing 3 different CUs for DPAC, DPCC is working with different laboratories, with different countries people, with a

J. Guiraud (✉) · W. Roux
CNES, Centre National d'Etudes Spatiales, Toulouse, France
e-mail: julie.guiraud@cnes.fr

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_14

huge numbers of contributors, dealing with heterogeneous ways of working and with a specific technology (Hadoop is not using SQL language, the design of scientific code has to be compliant). A software development, integration and validation organization has been defined by DPCC to manage all these discrepancies in a homogeneous way.

This organization, tools and methods implemented at DPCC have been part of the successful management of opened questions by the big data world in development of a data processing centre. For the third version of the catalogue, each pipeline executed at DPCC will produce data (and sometimes processed at the same time) which leads DPCC to face now opened questions by the big data world in exploitation of a data processing centre.

After a brief description of the Gaia project and of the CNES involvement in the Gaia data processing, the first part of the presentation will present how DPCC insures the completeness of the input data—more data are delivered by other DPCs, more the risk to have missing data is growing—and the importance of data accounting—how challenging it is to count a such amount of data.

The second part of the presentation will describe how DPCC defines, in collaboration with scientists, a representative dataset to test the new version of each scientific pipeline for the new catalogue version, including how DPCC publishes data to scientists with a dedicated tool to retrieve only part of data. Still in collaboration with scientific teams, how DPCC validates the data produced for the catalogue and how artificial intelligence can help on this verification.

The presentation will conclude with the tools implemented at DPCC to monitor each run (with Hadoop tools or dedicated ones developed by DPCC) part of the full pipeline execution leading to data catalogue, with the objective to optimize the configuration (software development or Hadoop configuration) for the preparation of the next Gaia catalogues.

1 Introduction

The ESA’s Gaia satellite has been launched from the Guiana Space Centre in December 2013 and produces operational observations since the 25th of July 2014. This mission is the successor of Hipparcos ESA’s satellite with the same objective of publishing a catalogue of stars and objects (galaxies, asteroids, etc.) but up to 1 billion objects (against 2.5 million). Gaia catalogue will give astrometric parameters (position, distance and proper motion), spectroscopic parameters (radial and rotational velocities) and full of astrophysical parameters (magnitude, effective temperature, gravity, abundances, etc.…) for a huge amount of objects.

The first catalogue has been released in September 2016, based on the first year of Gaia observations (2014/2015). Over one billion of sources have already been processed. The second version of the catalogue has been published for the entire scientific community since the 25th of April 2018 (based on observations from 2014 to mid-2016). Gaia Data Release 3 is split into two parts: the early release called early Data Release 3 (Gaia eDR3) and the full Gaia Data Release 3. The eDR3 contains full astrometric solution for around 1,800 billion sources (against 1,692 billion in DR2) based on observations from the beginning of the operations until May 2017, and is available since the 3rd of December 2020 [1]. The full one publication is scheduled for the first semester of 2022. A fourth and a fifth data releases are expected later in order to include last observations until the end of the mission, and several other processed data.

2 Gaia Mission

1. *The Gaia project*

Gaia is a space astronomy mission of ESA with a main objective to publish catalogues, for the entire scientific community. Each version contains the astrometry (the measurement of stellar position, parallax, and proper motion), photometry (the measurement of photometric magnitudes) and spectroscopy (for the acquisition of radial velocities and astrophysical parameters). Gaia will measure and distribute the positions, distances and physical characteristics of more than one billion stars in our galaxy and other objects beyond.

The Gaia satellite has been developed by Airbus Defence & Space (Toulouse, France), the control (shared with other ESA's missions) and mission centres are under ESOC responsibility (Darmstadt, Germany). It has been launched on the 19th December 2013 by a Soyouz-Fregat launcher from Guiana Space Centre, Kourou in French Guiana. The two-ton satellite is 1.5 million kilometres from Earth since orbit insertion performed during mid-January 2014 around the L2 Lagrange point. The mission was foreseen to last 5 years.

Raw data is received in ESA's stations (Cebreros, Malargüe and New Norcia) part of the ESTRAK network ground stations, sent to the Mission Operation Centre in ESOC and then transferred to the Science Operation Centre ESAC (European Space Astronomy Centre near Madrid, Spain) for the first science processing.

2. *Data Processing and Analysis Consortium*

The scientific data processing has been delegated to the Data Processing and Analysis Consortium (DPAC), composed of members of the astronomy community, nationally funded. Following an ESA Announcement of Opportunity, the Data Processing and Analysis Consortium (DPAC) has been created in 2006 and represents now about 438 people, engineers and scientists, from 20 countries across Europe and around the world (Brazil, Algeria…) in 2020 (details in Fig. 1). The yearly workload of the Gaia DPAC is about 250 Full Time Equivalents. The Gaia DPAC has been divided into 9 Coordination Units (CU) and 6 Data Processing Centres (DPC) with an executive committee, the DPACE [2].

The data processing centres are located across Europe, in charge of processing one or more CU scientific chains:

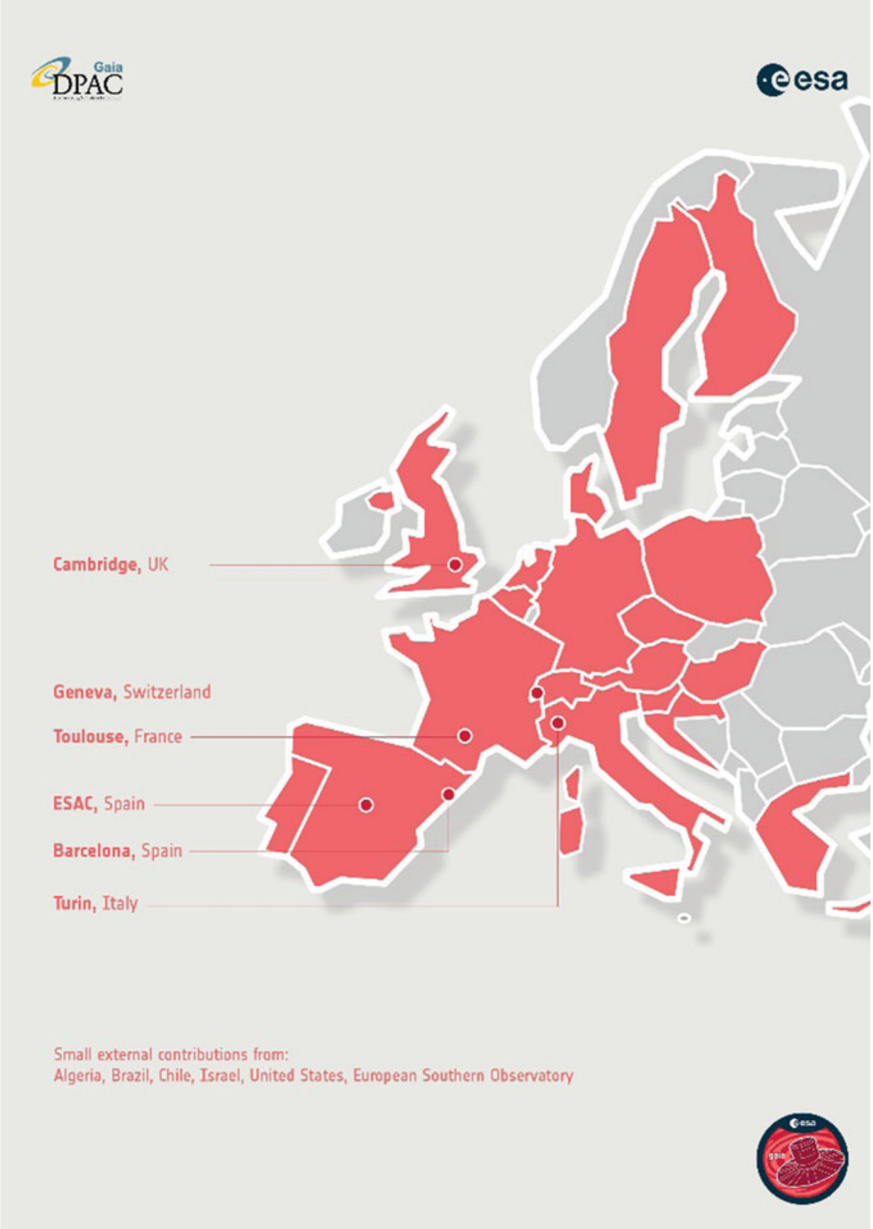

Gaia DPAC
esa
Cambridge, UK
Geneva, Switzerland
Toulouse, France
ESAC, Spain
Barcelona, Spain
Turin, Italy
Small external contributions from:
Algeria, Brazil, Chile, Israel, United States, European Southern Observatory
esa
gaia

Fig. 1 Gaia DPAC—2020—All countries in red participate in Gaia DPAC

- CU1, CU3 (first processing): DPCE at European Space Astronomy Centre (ESAC), Villanueva, Spain.
- CU3: DPCB at Barcelona Supercomputing Centre (BSC) and Centre de Supercomputació de Catalunya (CESCA), Barcelona, Spain.
- CU4, CU6, CU8: DPCC at Centre National d'Etudes Spatiales (CNES), Toulouse, France.
- CU7: DPCG at Observatoire de Genève, Geneva, Switzerland.
- CU5: DPCI at Institute of Astronomy (IoA), Cambridge, England.
- CU3: DPCT at INAF—Astronomical Observatory of Turin (OATO), Torino, Italy.

The scientific code is developed in Java language by the CU members, and then delivered to the corresponding DPCs for integration, system testing and operations.

3. *CNES Involvement in Gaia Data Processing*

CNES is responsible for the technical coordination, quality assurance, integration, validation and operations of the scientific developments for Object Processing (CU4: Solar System Objects, Non-Single Stars, Extended Objects), Spectroscopic Processing (CU6), and Astrophysical Parameters (CU8). It also participates to CU1 as deputy, which is the system architecture unit in charge of the DPAC common tools, definitions and management of the interfaces, system tests and operations coordination.

CNES is in charge of the development, the validation and the operations of the CNES Data Processing Centre (DPCC). The operations are foreseen for 3 years after the end of the extended mission (planned today in the end of 2025) for the final reprocessing of the Gaia catalogue. A total of 5 complete data releases are expected to the final Gaia catalogue.

3 How to Insure the Completeness of the Input Data?

DPCC insures the completeness of the input data (more data are produced and sent by other DPCs, more the risk to have missing data is growing) and the importance of data accounting, but it is challenging to count a such amount of data.

As the data processing centres are spread over Europe, a common tool has been chosen to manage all data exchanges in DPAC: Aspera [3], an IBM tool for managing huge transfers and streaming data. This software is installed in each data processing centre and connected to the central node, which is ESAC in charge of managing data transfers within DPAC. In organization point of view, each data processing centre subscribes to a part of Gaia data which are input data of their own scientific pipelines. Once a data processing centre has produced and validated their Gaia data, they are transferred to the central node and broadcasted to other data processing centre according to the subscriptions. Each transfer is followed up by a ticket on JIRA tool [4] in order to inform all concerned DPAC people.

An XML file is produced and transferred with each delivery (called delivery_form.xml). It is a standardized XML model shared by DPCs, and produced for each extracted data allowing an easier homogenous data transfer managing. Several parameters are defined such as the data processing centre producer, the type of data, the time observation of the processed data… and mainly the solution identifier list. This number is a numeric field which is unique in order to tag data produced within DPAC. The solution identifier is a 64-bit integer encoding on 10 bits the software identifier (unique for each code in each data processing centre), on 11 bits the software version (same software can produce different Gaia data, depending on the version of the software), on 15 bits the day number and on 27 bits the executionId (the unique execution number linked to a processing, same version of software can produce different Gaia data, depending on the inputs data or the parameters for example). After a data transfer, the list of solution identifier supposed received is compared to the solution identifier of each data really received. This verification allows the data processing centre to check if all supposed data have been transferred. At the end, an assessment is given by the data processing centre receiver in the JIRA ticket for traceability.

An internal common format of data was created specially to exchange data between the DPCs: the GBIN format. This is a compressed format, able to be split in many other files, with a datamodel based on Java serialization. This datamodel evolved regularly to add or update many objects, and is necessary for each centre to decode exchanged GBIN files.

A mechanism has been defined to discard some Gaia data, called data qualifier. It is generally used by daily data (cyclic data are validated in once) which can be reprocessed once the data segment is completed with the last version of the code for example, or after an issue detection in Gaia data. In that case, a file called data qualifier, is sent to each data processing centre subscriber containing the source identifiers list of data to be discarded. This file has to be processed independently by each data processing centre to delete or replace old Gaia data. Due to these data modifications, before starting cyclic operations, the completeness of inputs data has to be verified. DPCC stores in a database the information of inputs data received with the following parameters: Gaia data type, data model of the data, Aspera session number, solution identifier list, JIRA ticket of delivery, size of the delivery, number of files delivered, number of objects delivered, path of storage at DPCC.

As DPCC uses the Hadoop system, all the Gaia data have to be inserted in the Hadoop distributed file system in a format readable by Hadoop. This task is a Hadoop query exactly like a processing, meaning at the end of the insertion, a validation has to be made. DPCC developed a script for data accounting. It is also a Hadoop task, meaning it uses resources of the platform to be executed. At the end, a verification of the number of inserted objects compared to the number of rows in Gaia database is made, and is written in the delivery form. These tasks require resources and takes time. More data have to be counted, more time this activity requires. So, this task is done at each data reception and DPCC doesn’t wait for the full input data reception to start validation, in order to save time.

With all these operational steps, DPCC insures that all supposed data are received and inserted in the Hadoop platform. But as different versions of same Gaia data can be sent before starting operations, a wiki page is set up by ESAC (in charge of main database) with all solution identifier to be used for data release preparation, in order to have all DPCs processing the exact same input data.

4 How to Monitor the Execution of Each Processing?

DPCC has implemented tools to monitor each run (with Hadoop tools or dedicated ones developed by DPCC) part of the full pipeline execution leading to data catalogue, with the objective to optimize the configuration (software development or Hadoop configuration) for the preparation of the next Gaia catalogues.

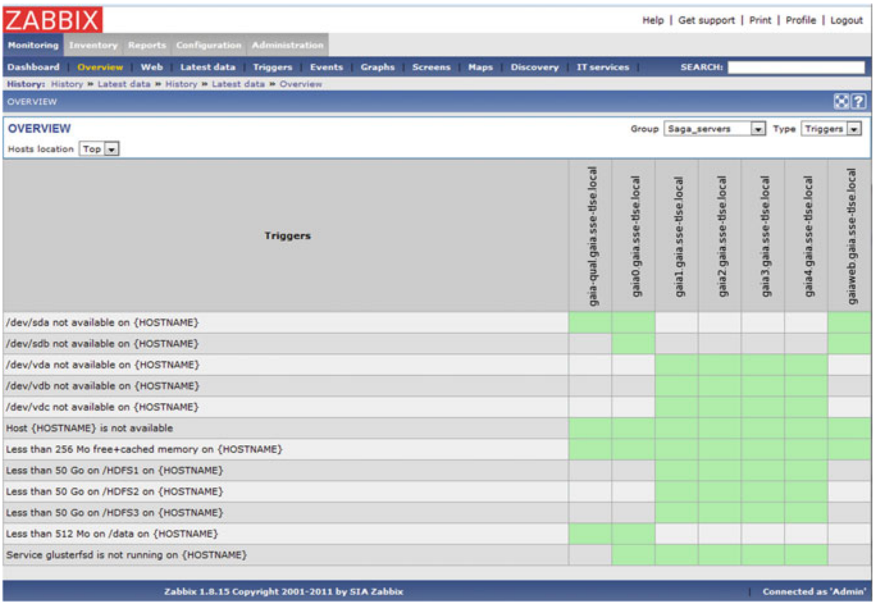
Optimization of the run duration is a major challenge of DPCC, as some chains may run during several months, any weeks saved are welcome. Hadoop gives a lot of statistics, but only at job level. End-to-end performances analyses are complex. Moreover, the processing duration is not a reliable measurement, as it strongly depends on the charge of the cluster, with other chains running at the same time. Several tools have been developed in DPCC to collect CPU time, memory and disk space for several runs issued from qualification phases on several datasets, and estimate the execution duration w.r.t. the final volume of data. At the end of each workflow execution, a “workplan visualizer” tool, relying on an ElasticSearch base [5], is used to quickly analyse the distribution of the jobs on the Hadoop platform. A new tool, named “Wooper” is in operation at DPCC, it automatically collects information from jobs’ logs and execution reports, extract statistics through CSV files, ingest them in a Microsoft Access database, and extrapolate figures according to the number of objects to process, and to the number of cores assigned to the chain. DPCC has implemented or developed the following tools to monitor all workflows of jobs, listed below.

1. *Zabbix*

Zabbix is an open source solution for network monitoring and application monitoring of millions of metrics [6]. DPCC uses it for the supervision of cluster resources, system metrics such as CPU, memory, disk space (example in the Fig. 2). This tool is powerful for IT team to monitor validation and operational platforms.

2. *Yarn*

YARN Dashboards regroups many services: YARN Job History, YARN Resources Manager, YARN Applications and YARN queues [7]. They are different web interfaces to monitor YARN applications and execution of MapReduce jobs. The YARN Job History has the full list of ended jobs and displays the information of each of them such as the status (successful, failed…), counters, time processing, Hadoop configuration, logs… The YARN Applications displays the same information for the current jobs (example in the Fig. 3). The YARN queues provides the view of


ZABBIX
Help | Get support | Print | Profile | Logout
Monitoring
Inventory
Reports
Configuration
Administration
Dashboard | Overview | Web | Latest data | Triggers | Events | Graphs | Screens | Maps | Discovery | IT services |
SEARCH:
History: History » Latest data » History » Latest data » Overview
OVERVIEW

OVERVIEW

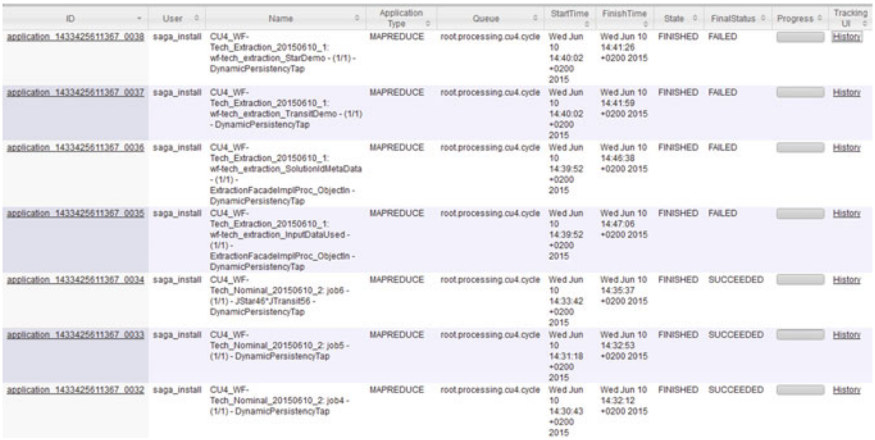
Group Saga_servers Type Triggers

Hosts location Top

Triggers	gaia-qual.gaia.sse-tlse.local	gaia0.gaia.sse-tlse.local	gaia1.gaia.sse-tlse.local	gaia2.gaia.sse-tlse.local	gaia3.gaia.sse-tlse.local	gaia4.gaia.sse-tlse.local	gaiaweb.gaia.sse-tlse.local
/dev/sda not available on {HOSTNAME}	🟩	🟩					🟩
/dev/sdb not available on {HOSTNAME}		🟩					🟩
/dev/vda not available on {HOSTNAME}			🟩	🟩	🟩	🟩	
/dev/vdb not available on {HOSTNAME}			🟩	🟩	🟩	🟩	
/dev/vdc not available on {HOSTNAME}			🟩	🟩	🟩	🟩	
Host {HOSTNAME} is not available	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Less than 256 Mo free+cached memory on {HOSTNAME}	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Less than 50 Go on /HDFS1 on {HOSTNAME}			🟩	🟩	🟩	🟩	
Less than 50 Go on /HDFS2 on {HOSTNAME}			🟩	🟩	🟩	🟩	
Less than 50 Go on /HDFS3 on {HOSTNAME}			🟩	🟩	🟩	🟩	
Less than 512 Mo on /data on {HOSTNAME}	🟩	🟩					
Service glusterfsd is not running on {HOSTNAME}		🟩	🟩	🟩	🟩	🟩	

Zabbix 1.8.15 Copyright 2001-2011 by SIA Zabbix | Connected as 'Admin'

Fig. 2 Zabbix implementation at DPCC

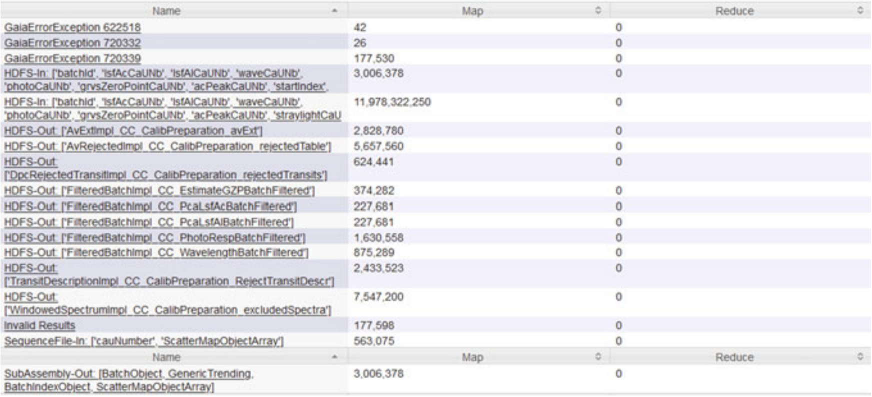


ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1433425611367_0038	saga_install	CU4_WF-Tech_Extraction_20150610_1: wf-tech_extraction_StarDemo - (1/1) - DynamicPersistencyTap	MAPREDUCE	root.processing.cu4.cycle	Wed Jun 10 14:40:02 +0200 2015	Wed Jun 10 14:41:26 +0200 2015	FINISHED	FAILED		History
application_1433425611367_0037	saga_install	CU4_WF-Tech_Extraction_20150610_1: wf-tech_extraction_TransitDemo - (1/1) - DynamicPersistencyTap	MAPREDUCE	root.processing.cu4.cycle	Wed Jun 10 14:40:02 +0200 2015	Wed Jun 10 14:41:59 +0200 2015	FINISHED	FAILED		History
application_1433425611367_0036	saga_install	CU4_WF-Tech_Extraction_20150610_1: wf-tech_extraction_SolutionIdMetaData - (1/1) - ExtractionFacadeImplProc_ObjectIn - DynamicPersistencyTap	MAPREDUCE	root.processing.cu4.cycle	Wed Jun 10 14:39:52 +0200 2015	Wed Jun 10 14:46:38 +0200 2015	FINISHED	FAILED		History
application_1433425611367_0035	saga_install	CU4_WF-Tech_Extraction_20150610_1: wf-tech_extraction_InputDataUsed - (1/1) - ExtractionFacadeImplProc_ObjectIn - DynamicPersistencyTap	MAPREDUCE	root.processing.cu4.cycle	Wed Jun 10 14:39:52 +0200 2015	Wed Jun 10 14:47:06 +0200 2015	FINISHED	FAILED		History
application_1433425611367_0034	saga_install	CU4_WF-Tech_Nominal_20150610_2: job6 - (1/1) - JStar46*JTransit56 - DynamicPersistencyTap	MAPREDUCE	root.processing.cu4.cycle	Wed Jun 10 14:33:42 +0200 2015	Wed Jun 10 14:35:37 +0200 2015	FINISHED	SUCCEEDED		History
application_1433425611367_0033	saga_install	CU4_WF-Tech_Nominal_20150610_2: job5 - (1/1) - DynamicPersistencyTap	MAPREDUCE	root.processing.cu4.cycle	Wed Jun 10 14:31:18 +0200 2015	Wed Jun 10 14:32:53 +0200 2015	FINISHED	SUCCEEDED		History
application_1433425611367_0032	saga_install	CU4_WF-Tech_Nominal_20150610_2: job4 - (1/1) - DynamicPersistencyTap	MAPREDUCE	root.processing.cu4.cycle	Wed Jun 10 14:30:43 +0200 2015	Wed Jun 10 14:32:12 +0200 2015	FINISHED	SUCCEEDED		History

Fig. 3 Example of YARN applications dashboard at DPCC

consumed resources and the resources allocation configuration. The YARN Resource Management gives the current jobs and their respective progressions in real time.

The ability to control counters during and after a job in YARN Applications and YARN Job History, is useful at DPCC to control their evolutions and their final values. They are good indicators to see the first signs of success or failure, especially for the scientific computation jobs. One can find the numbers of objects as inputs and



Name	Map	Reduce
GaiaErrorException 622518	42	0
GaiaErrorException 720332	26	0
GaiaErrorException 720339	177,530	0
HDFS-In: ['batchId', 'lsfAcCaUNb', 'lsfAlCaUNb', 'waveCaUNb', 'photoCaUNb', 'grvsZeroPointCaUNb', 'acPeakCaUNb', 'startIndex',	3,006,378	0
HDFS-In: ['batchId', 'lsfAcCaUNb', 'lsfAlCaUNb', 'waveCaUNb', 'photoCaUNb', 'grvsZeroPointCaUNb', 'acPeakCaUNb', 'straylightCaU	11,978,322,250	0
HDFS-Out: ['AvExtImpl_CC_CalibPreparation_avExt']	2,828,780	0
HDFS-Out: ['AvRejectedImpl_CC_CalibPreparation_rejectedTable']	5,657,560	0
HDFS-Out: ['DpcRejectedTransitImpl_CC_CalibPreparation_rejectedTransits']	624,441	0
HDFS-Out: ['FilteredBatchImpl_CC_EstimateGZPBatchFiltered']	374,282	0
HDFS-Out: ['FilteredBatchImpl_CC_PcaLsfAcBatchFiltered']	227,681	0
HDFS-Out: ['FilteredBatchImpl_CC_PcaLsfAlBatchFiltered']	227,681	0
HDFS-Out: ['FilteredBatchImpl_CC_PhotoRespBatchFiltered']	1,630,558	0
HDFS-Out: ['FilteredBatchImpl_CC_WavelengthBatchFiltered']	875,289	0
HDFS-Out: ['TransitDescriptionImpl_CC_CalibPreparation_RejectTransitDescr']	2,433,523	0
HDFS-Out: ['WindowedSpectrumImpl_CC_CalibPreparation_excludedSpectra']	7,547,200	0
Invalid Results	177,598	0
SequenceFile-In: ['cauNumber', 'ScatterMapObjectArray']	563,075	0

Name	Map	Reduce
SubAssembly-Out: [BatchObject, GenericTrending, BatchIndexObject, ScatterMapObjectArray]	3,006,378	0

Fig. 4 Example of specific counters displayed by Yarn implemented by DPCC

outputs, control the number of exceptions in jobs, and also trace specific counters of raised flags at different steps inside the jobs (example in the Fig. 4).

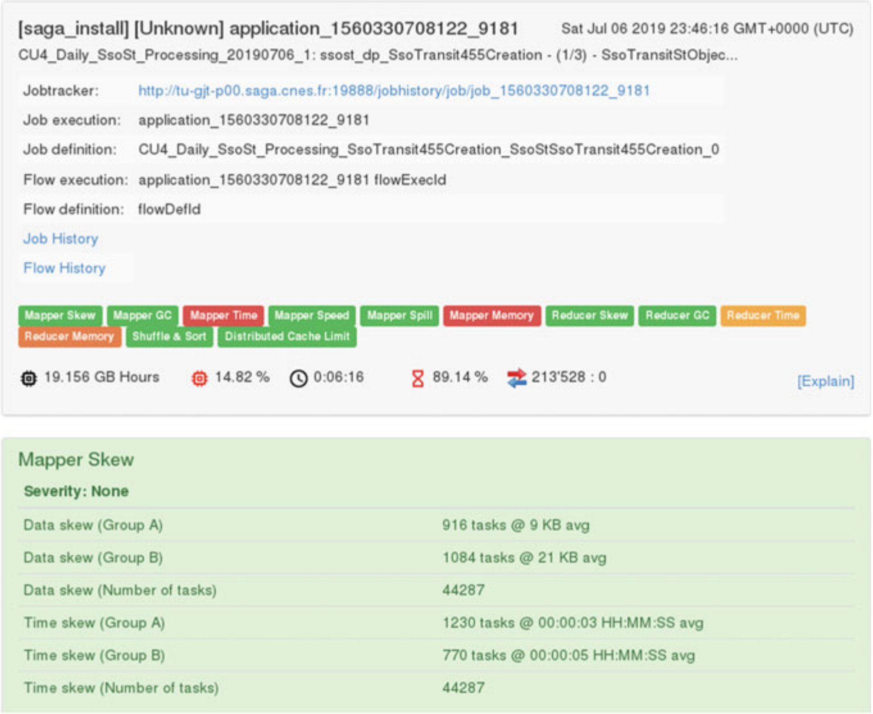
Because of the numerous jobs launched per day, the DPCC is forced to clean the job history beyond 3 months. That's why we have developed our own tools to sum up the useful information at long term (see §5 Wooper).

3. *Dr. Elephant*

Dr Elephant is a performance monitoring and tuning tool for Apache Hadoop and Apache Spark jobs and workflows [8]. This tool, installed on validation and operational platform, retrieves regularly a list of succeed and failed runs through the YARN Resources Manager. The Dr Elephant's dashboard presents detailed diagnostic per job analysing a list of statistics (example in the Fig. 5). These indicators are useful during the qualification and the operations, especially the skew, the garbage collector and the memory for mappers and reducers, because we can tune some Hadoop parameters in consequence to optimize the configuration for a future run.

4. *DPCC Workplan Visualizer*

DPCC Workplan Visualizer is a tool developed at DPCC to monitor the parallelization and time durations in each pipeline's steps of maps and reduces (example in the Fig. 6). This tool allows in particular the detection of a poor distribution through the platform of maps and reduces, and optimize it in consequence thanks to the distribution keys. Thanks to the plots it is also possible to detect the high irregularities, and reduce them increasing the number of maps or reduces with respect to the volume of data to process, in order to streamline the processing and take maximum benefits of the Hadoop platform.



[saga_install] [Unknown] application_1560330708122_9181 Sat Jul 06 2019 23:46:16 GMT+0000 (UTC)

CU4_Daily_SsoSt_Processing_20190706_1: ssost_dp_SsoTransit455Creation - (1/3) - SsoTransitStObjec...

Jobtracker:	http://tu-gjt-p00.saga.cnes.fr:19888/jobhistory/job/job_1560330708122_9181
Job execution:	application_1560330708122_9181
Job definition:	CU4_Daily_SsoSt_Processing_SsoTransit455Creation_SsoStSsoTransit455Creation_0
Flow execution:	application_1560330708122_9181 flowExecId
Flow definition:	flowDefId

Job History

Flow History

19.156 GB Hours 14.82 % 0:06:16 89.14 % 213'528 : 0

Mapper Skew

Severity: None

Data skew (Group A)	916 tasks @ 9 KB avg
Data skew (Group B)	1084 tasks @ 21 KB avg
Data skew (Number of tasks)	44287
Time skew (Group A)	1230 tasks @ 00:00:03 HH:MM:SS avg
Time skew (Group B)	770 tasks @ 00:00:05 HH:MM:SS avg
Time skew (Number of tasks)	44287

Fig. 5 Example of Dr Elephant's windows on an ended processing run

5. *Wooper*

Wooper is a tool developed by DPCC to sum up very useful metrics (example in the Fig. 7), analyse the processing metrics and use it to perform the technical validations. It consists of a Microsoft Access database importing, summarizing and archiving different metrics from Hadoop YARN reports at a workflow level (a chain of Hadoop jobs). They are imported daily to feed this long-term database. We collect less information than in YARN Dashboards, but we can keep this database until the end of DPCC's life.

5 How to Validate Such Amount of Produced Data?

DPCC defines, in collaboration with scientists, a representative dataset to test the new version of each scientific pipeline for the new catalogue version, including how DPCC publishes data to scientist with a dedicated tool to retrieve only part of data. Still in collaboration with scientific teams, DPCC validates the data produced for the catalogue.

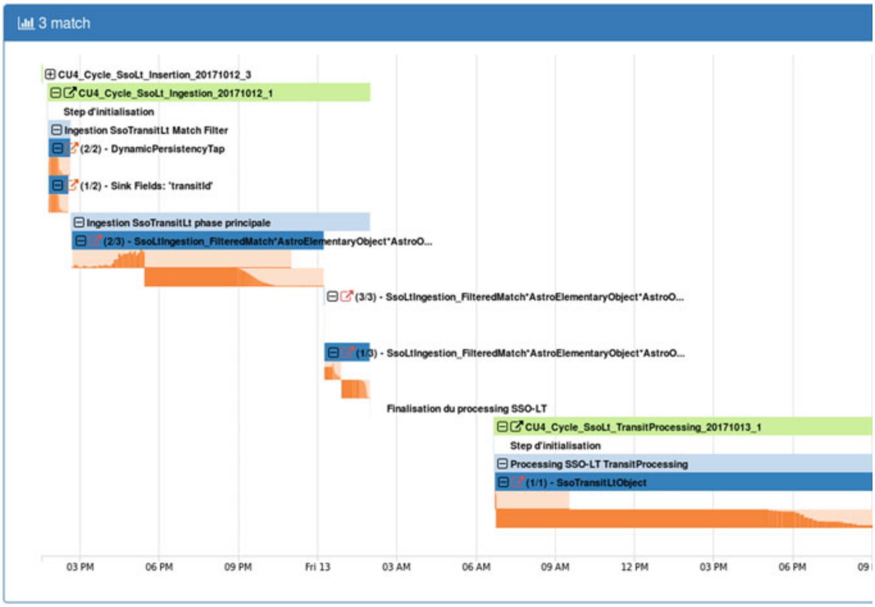
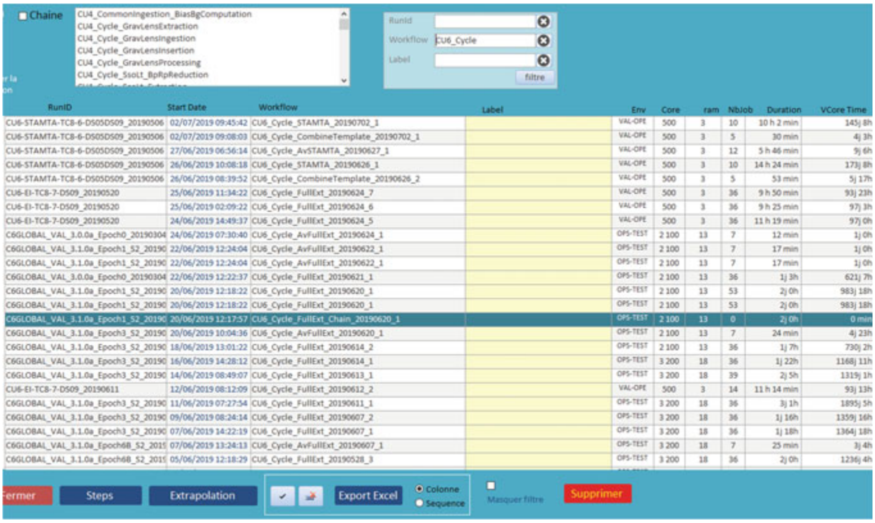

3 match
CU4_Cycle_SsoLt_Insertion_20171012_3
CU4_Cycle_SsoLt_Ingestion_20171012_1
Step d'initialisation
Ingestion SsoTransitLt Match Filter
(2/2) - DynamicPersistencyTap
(1/2) - Sink Fields: 'transitId'
Ingestion SsoTransitLt phase principale
(2/3) - SsoLtIngestion_FilteredMatch*AstroElementaryObject*AstroO...
(3/3) - SsoLtIngestion_FilteredMatch*AstroElementaryObject*AstroO...
(1/3) - SsoLtIngestion_FilteredMatch*AstroElementaryObject*AstroO...
Finalisation du processing SSO-LT
CU4_Cycle_SsoLt_TransitProcessing_20171013_1
Step d'initialisation
Processing SSO-LT TransitProcessing
(1/1) - SsoTransitLtObject
03 PM
06 PM
09 PM
Fri 13
03 AM
06 AM
09 AM
12 PM
03 PM
06 PM
09

Fig. 6 Example of Workplan Analyzer view at DPCC


□ Chaine
CU4_CommonIngestion_BiasBgComputation
CU4_Cycle_GravLensExtraction
CU4_Cycle_GravLensIngestion
CU4_Cycle_GravLensInsertion
CU4_Cycle_GravLensProcessing
CU4_Cycle_SsoLt_BpRpReduction
r la
on
Fermer
Steps
Extrapolation
✓
Export Excel
□ Masquer filtre
Supprimer

RunId
Workflow CU6_Cycle
Label
filtre

● Colonne
○ Sequence

RunID	Start Date	Workflow	Label	Env	Core	ram	NbJob	Duration	VCore Time
CU6-STAMTA-TC8-6-DS05DS09_20190506	02/07/2019 09:45:42	CU6_Cycle_STAMTA_20190702_1		VAL-OPE	500	3	10	10 h 2 min	145j 8h
CU6-STAMTA-TC8-6-DS05DS09_20190506	02/07/2019 09:08:03	CU6_Cycle_CombineTemplate_20190702_1		VAL-OPE	500	3	5	30 min	4j 3h
CU6-STAMTA-TC8-6-DS05DS09_20190506	27/06/2019 06:56:14	CU6_Cycle_AvSTAMTA_20190627_1		VAL-OPE	500	3	12	5 h 46 min	9j 6h
CU6-STAMTA-TC8-6-DS05DS09_20190506	26/06/2019 10:08:18	CU6_Cycle_STAMTA_20190626_1		VAL-OPE	500	3	10	14 h 24 min	173j 8h
CU6-STAMTA-TC8-6-DS05DS09_20190506	26/06/2019 08:39:52	CU6_Cycle_CombineTemplate_20190626_2		VAL-OPE	500	3	5	53 min	5j 17h
CU6-EI-TC8-7-DS09_20190520	25/06/2019 11:34:22	CU6_Cycle_FullExt_20190624_7		VAL-OPE	500	3	36	9 h 50 min	93j 23h
CU6-EI-TC8-7-DS09_20190520	25/06/2019 02:09:22	CU6_Cycle_FullExt_20190624_6		VAL-OPE	500	3	36	9 h 25 min	97j 3h
CU6-EI-TC8-7-DS09_20190520	24/06/2019 14:49:37	CU6_Cycle_FullExt_20190624_5		VAL-OPE	500	3	36	11 h 19 min	97j 0h
C6GLOBAL_VAL_3.0.0a_Epoch0_20190304	24/06/2019 07:30:40	CU6_Cycle_AvFullExt_20190624_1		OPS-TEST	2 100	13	7	12 min	1j 0h
C6GLOBAL_VAL_3.1.0a_Epoch1_S2_20190	22/06/2019 12:24:04	CU6_Cycle_AvFullExt_20190622_1		OPS-TEST	2 100	13	7	17 min	1j 0h
C6GLOBAL_VAL_3.1.0a_Epoch1_S2_20190	22/06/2019 12:24:04	CU6_Cycle_AvFullExt_20190622_1		OPS-TEST	2 100	13	7	17 min	1j 0h
C6GLOBAL_VAL_3.0.0a_Epoch0_20190304	22/06/2019 12:22:37	CU6_Cycle_FullExt_20190621_1		OPS-TEST	2 100	13	36	1j 3h	621j 7h
C6GLOBAL_VAL_3.1.0a_Epoch1_S2_20190	20/06/2019 12:18:22	CU6_Cycle_FullExt_20190620_1		OPS-TEST	2 100	13	53	2j 0h	983j 18h
C6GLOBAL_VAL_3.1.0a_Epoch1_S2_20190	20/06/2019 12:18:22	CU6_Cycle_FullExt_20190620_1		OPS-TEST	2 100	13	53	2j 0h	983j 18h
C6GLOBAL_VAL_3.1.0a_Epoch1_S2_20190	20/06/2019 12:17:57	CU6_Cycle_FullExt_Chain_20190620_1		OPS-TEST	2 100	13	0	2j 0h	0 min
C6GLOBAL_VAL_3.1.0a_Epoch3_S2_20190	20/06/2019 10:04:36	CU6_Cycle_AvFullExt_20190620_1		OPS-TEST	2 100	13	7	24 min	4j 23h
C6GLOBAL_VAL_3.1.0a_Epoch3_S2_20190	18/06/2019 13:01:22	CU6_Cycle_FullExt_20190614_2		OPS-TEST	2 100	13	36	1j 7h	730j 2h
C6GLOBAL_VAL_3.1.0a_Epoch3_S2_20190	16/06/2019 14:28:12	CU6_Cycle_FullExt_20190614_1		OPS-TEST	3 200	18	36	1j 22h	1168j 11h
C6GLOBAL_VAL_3.1.0a_Epoch3_S2_20190	14/06/2019 08:49:07	CU6_Cycle_FullExt_20190613_1		OPS-TEST	3 200	18	39	2j 5h	1319j 1h
CU6-EI-TC8-7-DS09_20190611	12/06/2019 08:12:09	CU6_Cycle_FullExt_20190612_2		VAL-OPE	500	3	14	11 h 14 min	93j 13h
C6GLOBAL_VAL_3.1.0a_Epoch3_S2_20190	11/06/2019 07:27:54	CU6_Cycle_FullExt_20190611_1		OPS-TEST	3 200	18	36	3j 1h	1895j 5h
C6GLOBAL_VAL_3.1.0a_Epoch3_S2_20190	09/06/2019 08:24:14	CU6_Cycle_FullExt_20190607_2		OPS-TEST	3 200	18	36	1j 16h	1359j 16h
C6GLOBAL_VAL_3.1.0a_Epoch3_S2_20190	07/06/2019 14:22:19	CU6_Cycle_FullExt_20190607_1		OPS-TEST	3 200	18	36	1j 18h	1364j 18h
C6GLOBAL_VAL_3.1.0a_Epoch6B_S2_201	07/06/2019 13:24:13	CU6_Cycle_AvFullExt_20190607_1		OPS-TEST	3 200	18	7	25 min	3j 4h
C6GLOBAL_VAL_3.1.0a_Epoch6B_S2_201	05/06/2019 12:18:29	CU6_Cycle_FullExt_20190528_3		OPS-TEST	3 200	18	36	2j 0h	1236j 4h

Fig. 7 Wooper view of runs

1. *DPCC Validation*

Each time a data processing is ended, the first activity at DPCC is the numerical and statistical validation of the processing results and analysis of processing metrics. DPCC members are engineers and not scientists, so it's impossible to assess the validity of the results on a scientific point of view but the first technical analysis done avoids loosing time to work on wrong data.

The processing metrics analysis role is to verify the coherence of run characteristics linked with the platform, such as the processing time per each scientific job, CPUs' consumption, disk space used… For example, if a run lasts 2 weeks and after a patch application or a parameter configuration change, the next run lasts 10 min, we can conclude there is a problem with the change and nothing has been really processed. In another example, if a new field is added in the datamodel and after this modification (and only this one), the results of the next run uses less disk space instead of more, we can also conclude the modification has not been correctly applied.

The numerical validation consists in checking if the Hadoop YARN counters seem compliant with the definition of the test or run. For example, it's impossible to have more sources in output than in input (we should obtain around the same number of sources in input and in output) or it implies there is duplicate data. In another example, if a job hasn't produced any results or if more sources are published than the sources number in the Validation Source Table, we can conclude there is a technical problem. Moreover, the counters of exceptions or specific flags implemented by DPCC and raised during jobs can be used in order to validate comparing them to theoretical values.

The statistical validation method goal is to compute percentages and create plots of trends to give a first assessment on the results. For example, the percentage of rejected sources during the processing should be more or less the same percentage of rejected sources by general Gaia processing (around 11% for catalogue 3). Another example, the Gaia satellite observes between 20 and 40 times the same sources since the launch. At the end of a processing, the number of observations per processed source is checked, compared to the number of satellite observations and plotted to observe the trends (see Fig. 8).

2. *DPCC Publication*

After this primary analysis, DPCC send to the scientific team the results of objects processing included in the VST list for scientific validation. In order to publish the results, DPCC has implemented a CNES tool called SiTools2 which allow data access in its data centres and from scientific laboratories spread over Europe through a web portal, called Gaiaweb.

SiTools2 is an open source web application, developed by CNES [9]. Its purpose is to propose a data access layer as a new tool of an existing archival system. It is a convenient tool for scientific communities which have to provide data access to their data processing or data results. That's why CNES is not the only one to exploit it.

The main goal of this tool is to publish data (results of processing pipeline, logs of the run and execution reports with information on the configuration, the code

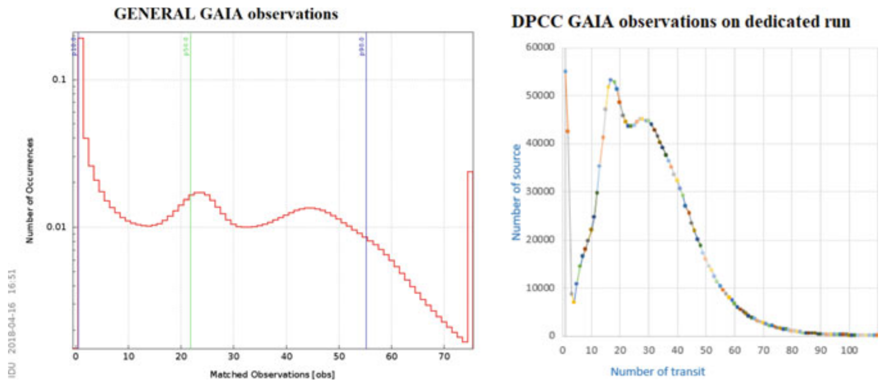
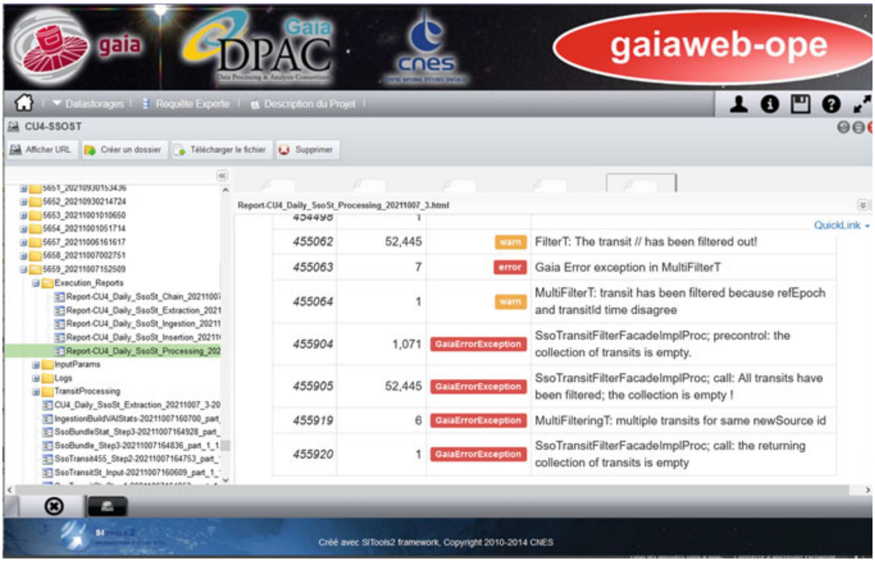


Fig. 8 Observation of a comparable double peaks structure before (left) and after processing (right)

version and dataset used in order to be replayed if needed) in directories (example in the Fig. 9).

In addition, Gaiaweb may be used to fetch data into DPCC Hadoop file system directly by the user. This mechanism is called "Expert Request". The requests are executed on the Hadoop platform on all data (not just a subset). The software operation is:

- The user creates a request on Gaiaweb MMI.
- Gaiaweb generates a JSON file.



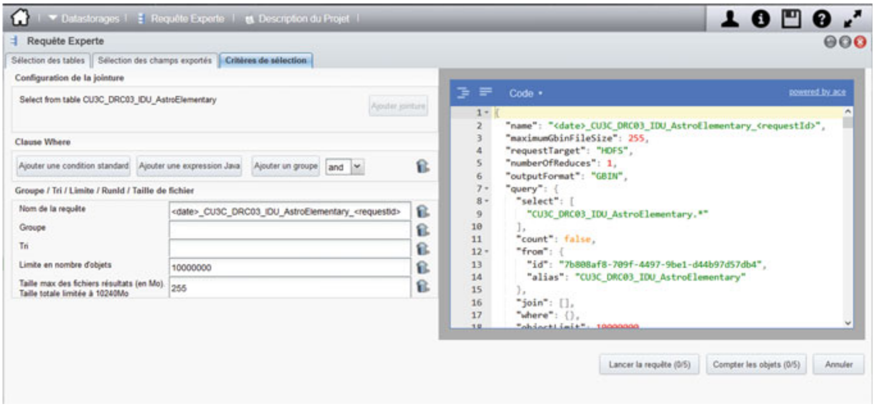
5651_20210930153436
5652_20210930214724
5653_20211001010650
5654_20211001051714
5657_20211006161617
5658_20211007002751
5659_20211007152509
Execution_Reports
Report-CU4_Daily_SsoSt_Chain_2021100
Report-CU4_Daily_SsoSt_Extraction_2021
Report-CU4_Daily_SsoSt_Ingestion_20211
Report-CU4_Daily_SsoSt_Insertion_20211
Report-CU4_Daily_SsoSt_Processing_202
InputParams
Logs
TransitProcessing
CU4_Daily_SsoSt_Extraction_20211007_3-20
IngestionBuildVAIStats-20211007160700_part_
SsoBundleStat_Step3-20211007164928_part_
SsoBundle_Step3-20211007164836_part_1_1
SsoTransit455_Step2-20211007164753_part_
SsoTransitSt_Input-20211007160609_part_1_

Report-CU4_Daily_SsoSt_Processing_20211007_3.html

454498	1		
455062	52,445	warn	FilterT: The transit // has been filtered out!
455063	7	error	Gaia Error exception in MultiFilterT
455064	1	warn	MultiFilterT: transit has been filtered because refEpoch and transitId time disagree
455904	1,071	GaiaErrorException	SsoTransitFilterFacadeImplProc; precontrol: the collection of transits is empty.
455905	52,445	GaiaErrorException	SsoTransitFilterFacadeImplProc; call: All transits have been filtered; the collection is empty !
455919	6	GaiaErrorException	MultiFilteringT: multiple transits for same newSource id
455920	1	GaiaErrorException	SsoTransitFilterFacadeImplProc; call: the returning collection of transits is empty

Créé avec SITools2 framework, Copyright 2010-2014 CNES

Fig. 9 Example of data publication on Gaiaweb



Configuration de la jointure

Select from table CU3C_DRC03_IDU_AstroElementary

Clause Where

Groupe / Tri / Limite / RunId / Taille de fichier

Nom de la requête	<date>_CU3C_DRC03_IDU_AstroElementary_<requestId>
Groupe	
Tri	
Limite en nombre d'objets	10000000
Taille max des fichiers résultats (en Mo). Taille totale limitée à 10240Mo	255

```
{
  "name": "<date>_CU3C_DRC03_IDU_AstroElementary_<requestId>",
  "maximumGbinFileSize": 255,
  "requestTarget": "HDFS",
  "numberOfReduces": 1,
  "outputFormat": "GBIN",
  "query": {
    "select": [
      "CU3C_DRC03_IDU_AstroElementary.*"
    ],
    "count": false,
    "from": {
      "id": "7b808af8-709f-4497-9be1-d44b97d57db4",
      "alias": "CU3C_DRC03_IDU_AstroElementary"
    },
    "join": [],
    "where": {},
    "objectLimit": 10000000
```

Fig. 10 Example of “Expert Request” on Gaiaweb

- A script pulls the JSON folder and retrieves it.
- The script generates a job on the fly and submits it to Hadoop.
- Hadoop runs the job (like any processing).
- The script gets the results and publish them on Gaiaweb.
- Gaiaweb notifies the user by email.
- The user downloads the results, like any data automatic publications.

This tool allows scientists to create on-demand queries with an SQL-like language executed on the operational cluster in order to extract data on full available data on Hadoop File System, on defined criteria (example in the Fig. 10). Also, as the requests are based on a JSON file, the requests can be saved and replayed when necessary.

3. *Scientific Validation*

As it’s a Big Data environment, the dataset selection is crucial: not too much to keep it as a test and not operations, but not too few to avoid missing problematic cases. During the qualification phase of the scientific pipeline, the scientific team selects the candidates (can be single observation, full source… depending on the pipeline and the tests to be done) to construct a validation list, called Validation Source Table. This Gaia table, created by DPCC, contains rows of objects to be processed by the pipeline. Some objects are good candidates with a well-known behaviour thanks to crossed external validation (based on existing catalogues outside Gaia project or parameters publication for example). Their role is to confirm the expected results of the pipeline compared to these catalogues or publications. If the results are comparable, the expected behaviour of the pipeline is confirmed. Others objects are limit candidates. The study of their processing results helps to determine threshold or disable characteristics of objects, in order to publish in the Gaia catalogue only correct results and discard spurious data or outliers. These objects can lead to a pipeline crash, a never converging algorithm or a pipeline’s strange behaviour to be managed by DPCC.

The second method is the internal validation: when data have been previously published in Gaia Data Release, the scientific team compares the current processing results of objects with the previous results. The precision or the residual error is generally improved but the order of magnitude or tendency must be the same.

To avoid algorithm or language bias, scientific team tests two separate codes, working with different implementation of the algorithms, on different hardware configurations and different languages in order to confront both results and to assess if results are comparable. This Validation Source Table list represents less than 10% of the full Gaia sources to be processed which is enough to be representative but not sufficient for science interpretations.

Independently of the validation method (external or internal), the scientific team creates plots of all Validation Source Table results processing. The goal is to analyse the tendency, the distribution or to determine the outliers. The most recurrent plots can be implemented in the Gaiaweb data portal jointly defined by DPCC and scientists, the objective is to provide plots as soon as data are available and to use the potential of DPCC platform to compute them. Increasingly, the scientific teams equip themselves with dedicated computers for the validation, to be able to perform some statistical analysis on the complete results. For a reasonable volume of results to analyse, they can simply use TOPCAT (Tool for OPerations on Catalogues And Tables), that have been updated in order to read GBIN files [10]. In the case of too voluminous data, the teams can develop their own database to ingest results with their own tools, independent of the DPCC.

6 Conclusion

During this phase of qualification, the configuration parameters are tested and validated. Using the Hadoop system requires a dedicated team of experts to monitor and to tune it for the data processing operations. Each parameter influences the processing time. For example, one CU6 pipeline parameter has been set to a prime number instead of random one which leaded to a global processing time divided by 6.

To allow the operations of several pipelines at the same time, DPCC uses the queue mechanism of Hadoop: each pipeline has a maximum of cores and RAM allocated, depending on the resources needed and available. For example, one CU4 pipeline which uses Cassandra database is configured to run on 100 cores maximum. In that case, the pipeline is bounded because Hadoop's requests are too numerous for Cassandra, leading to writing error due to concurrent access to the database.

To process faster the data, the final computation has not to be done on the final reduce, the DPCC's programming team has to develop the Hadoop encapsulation to avoid this configuration.

Finally, the input data used by the pipeline are inserted in the Hadoop file system before the operations. The way these files are inserted is crucial to obtain correct processing performances: too big files will saturate the memory at loading by the pipelines, too small files will generate too many temporary files during the reading of input data by the pipeline. In the same way, the files generated by the pipeline (linked to the number of reduces) have to be well managed (which can be contradictory with the low number of reduces to be avoided). Sometimes a post-processing is requested to recreate bigger output files (in size but with less number of files).

The data management is the key in big data environment. DPCC has now to implement a way of ordering the output data from operational pipelines to be processed faster by other consumer DPCs.

References

1. European Space Agency (2020) Gaia Early Data Release 3 (Gaia EDR3), 3 December 2020. https://www.cosmos.esa.int/web/gaia/earlydr3. Accessed 19 Mar 2021
2. European Space Agency (2021) The DPAC Consortium. https://www.cosmos.esa.int/web/gaia/dpac/consortium. Accessed 19 Mar 2021
3. IBM (2021) Aspera, https://www.ibm.com/products/aspera. Accessed 19 Mar 2021
4. Atlassian (2021) Jira Software, https://www.atlassian.com/software/jira. Accessed 19 Mar 2021
5. Elastic BV (2021) Elasticsearch, https://www.elastic.co/fr/elasticsearch/. Accessed 19 Mar 2021
6. Zabbix LLC (2021) Zabbix Features, https://www.zabbix.com/features. Accessed 19 Mar 2021
7. Apache Software Foundation (2021) Apache Hadoop YARN, 15 June 2021. https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html. Accessed 21 Sept 2021
8. Akshay R (2016) Open sourcing Dr. Elephant, self-serve performance tuning for hadoop and spark, 8 Apr 2016. https://engineering.linkedin.com/blog/2016/04/dr-elephant-open-source-self-serve-performance-tuning-hadoop-spark. Accessed 19 Mar 2021
9. Malapert JC (2011) SITools2: framework for data access layer. Astron Date Anal Softw Syst XX 442:103
10. Taylor M (2021) TOPCAT-GBIN. http://www.star.bris.ac.uk/~mbt/topcat/sun253/inGbin.html. Accessed 22 Sept 2021

Virtual Reality in Support of Space Weather Forecasting

Evridiki V. Ntagiou, Johannes Klug, Juha-Pekka Luntama, and Mehran Sarkarati

Abstract Within the Operations Directorate large amounts of data are produced daily. Their proper visualization is the bridge between the quantitative information in the data and the human intuition and understanding. The Space Weather System in Space Safety Programme produces huge amounts of data products from hundreds of sensors on ground and in space, based on which the Space Weather forecasters make qualitative and quantitative nowcasts and forecasts. Despite the constantly advancing numerical simulation techniques and advanced analysis of the data, human interpretation of the outputs is still pivotal in providing good forecasts. Intuitive data visualisation tools are one of the key techniques to improve their accuracy in the near future. Today, the main means of visualization are 2D graphs projecting the propagation of heliospheric plasma on two orthogonal planes. In this paper, we present the proof-of-concept prototype we have implemented where historic data of Coronal Mass Ejections come to life in an interactive tool based on a Virtual Reality game engine. Employing VR technology, we offer an immersive experience to the forecaster and support educational and promotion activities. The 3D visualisation tool is an innovation that brings new technology in SWE forecasting closely linked to mission operations in ESOC.

Keywords Space weather · Virtual reality · Coronal mass ejection

E. V. Ntagiou (✉) · J. Klug
Applications and Robotics Data Systems Section, Mission Operations Data Systems Division, European Space Operations Centre, Robert-Bosch Str., 64293 Darmstadt, Germany
e-mail: evridiki.ntagiou@esa.int

J. Klug
e-mail: johannes.klug@esa.int

J.-P. Luntama
Space Weather Office, Space Safety Programme Office, European Space Operations Centre, Robert-Bosch Str., 64293 Darmstadt, Germany
e-mail: juha-pekka.luntama@esa.int

M. Sarkarati
Information Technology Department, Directorate of Internal Services, Robert-Bosch Str., 64293 Darmstadt, Germany
e-mail: mehran.sarkarati@esa.int

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_15

Acronyms/Abbreviations

AI	Artificial Intelligence
AR	Augmented Reality
CME	Coronal Mass Ejection
ESA	European Space Agency
ESOC	European Space Operations Centre
EUHFORIA	EUropean Heliospheric FORecasting Information Asset
GNSS	Global Navigation Satellite System
ML	Machine Learning
MR	Mixed Reality
SWE	Space Weather
S2P	Space Safety Programme
SST	Space Surveillance and Tracking
VR	Virtual Reality
XR	Extended Reality

1 Introduction

The term Space Weather (SWE) refers to the changes in the space environment, resulting primarily from changes on the Sun. Space Weather can be defined as: *Space weather is the physical and phenomenological state of natural space environments. The associated discipline aims, through observation, monitoring, analysis and modelling, at understanding and predicting the state of the Sun, the interplanetary and planetary environments, and the solar and non-solar driven perturbations that affect them, and also at forecasting and nowcasting the potential impacts on biological and technological systems* [1].

Space Weather can have a considerable impact on a number of aspects of our everyday life [2] from the electrical power distribution systems, communications and Global Navigation Satellite System (GNSS) positioning accuracy, and even potential impacts on climate. A number of industries can see an effect to the quality of their services due to SWE events and satellite navigation services are majorly affected by solar activity due to disturbances in the ionosphere affecting a number of activities relying on satellite positioning. The impact on orbiting satellites is found on the communications and solar panel performance degradation, functioning of the onboard electronics, and thus in the mission lifetime. Space radiation leads to increased risks to human health on manned space missions. Space weather also has numerous effects on the ground including damage to aircraft electronics, enhanced radiation dose for air passengers and crew, damage and disruption to power distribution networks and pipelines and degradation of HF radio communications.

In preparation of being confronted with potentially hazardous SWE impacts, solar remote sensing and space environment in-situ data are being processed with sophisticated numerical models that will produce forecasts of Space Weather events and their effects to both space and ground segments as well as other domains (e.g. aircraft, power distribution systems etc.). Being able to monitor SWE events, analyse the measurement data and quickly assess the risk from the event for the infrastructure in space or on ground is critical for issuing timely alerts to the end users and taking action to mitigate the impacts. Data visualization is crucial for understanding the outputs of numerical Space Weather models and efficient risk assessment. The current means of visualization of Space Weather phenomena used by scientists and operators are typically focused on 2 dimensional graphs, where quantities like the plasma density or velocity are projected on two orthogonal planes [3, 4]. Improving the visualisation methods can lead to more timely detection of unexpected features which will in turn allow for forecasts of higher accuracy.

1.1 Background

1.1.1 Overview of the Space Safety Programme (S2P)

The objective of the *Space Safety* system is to support the European independent utilization of and access to space for research or services, through providing timely quality data, information, services and knowledge regarding the space environment, the sustainable exploitation of the outer space and the threats particularly regarding hazards to infrastructure in orbit and on the ground. ESA carries out the SSA objectives in successive programmatic steps with a view to achieve a full operational capability over a framework of ten years. ESA is using an architectural breakdown of the objectives into the *three* following segments:

- **Space Surveillance and Tracking (SST) of man-made space objects**. Space Surveillance and Tracking is required in order to respond to the need to maintain the awareness of the population of man-made space objects. SST aims at surveying and tracking of objects in Earth orbit comprising active and inactive satellites, discarded launch stages and fragmentation debris that orbit Earth.
- **Near-Earth Objects (NEO) surveillance and tracking**. The NEO segment of ESA's Space Safety Programme has established precursor services which provide information on asteroids and especially Near-Earth Asteroids.
- **Space Weather (SWE)**. The Space Weather segment deals with changes in the space environment, resulting mainly from changes on the Sun, including modification of the ambient plasma, particulate radiation (electrons, protons and ions), electromagnetic radiation (including radio-waves, visible, UV and X-ray radiation), and the magnetic and electric field. In addition to the Sun, non-solar sources such as galactic cosmic rays, micron-size particulates (from meteoroids and space debris) can all be considered as altering space environment conditions near the

Earth and are also considered as observational objectives to be covered by the SWE segment of the S2P (Fig. 1).

The European system to monitor, predict and disseminate Space Weather information and alerts is being developed in the framework of the ESA Space Safety Programme. The S2P SWE Services are provided through a network consisting of a SWE Service Coordination Centre (SSCC) and five thematic expert service centres (ESCs), supported by the SWE Data Centre (Fig. 2). These services utilise data from a

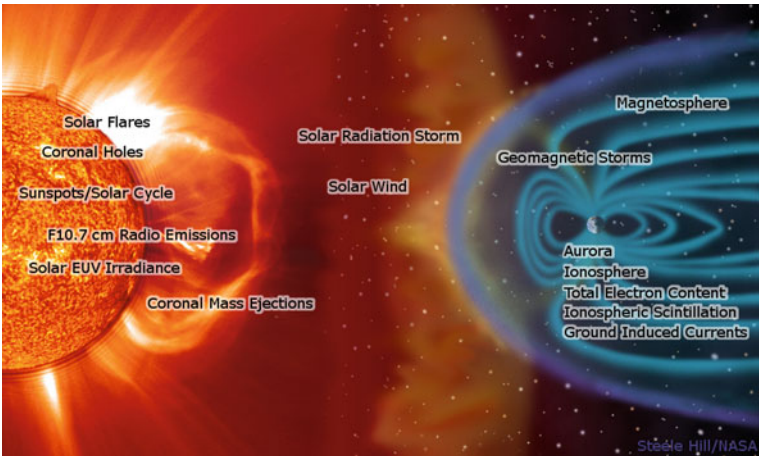


Fig. 1 Space weather phenomena

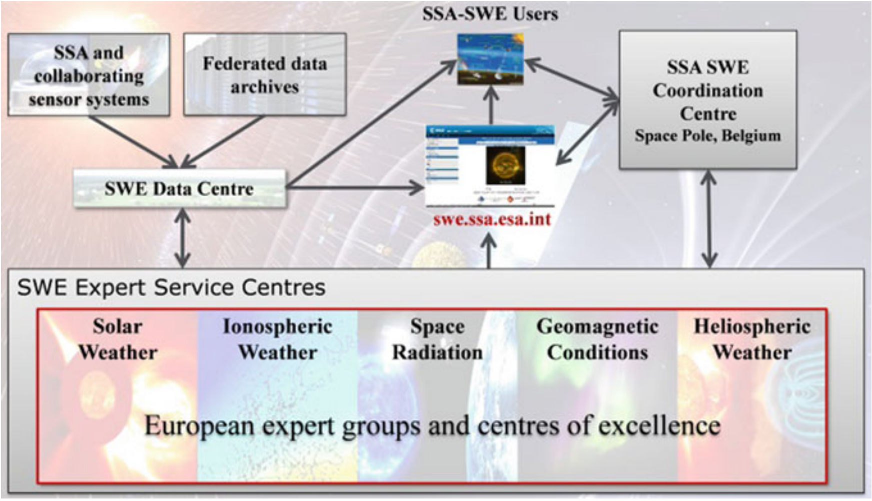


Fig. 2 The S2P SWE service network

wide range of ground-based and space based measurement systems, combined with advanced processing and expertise. Within the SWE Service Network,[1] the ESCs carry out specialist processing beyond level 1 where required.

1.1.2 Extended Reality

Extended reality (XR) is a term referring to all real-and-virtual combined environments and human–machine interactions generated by computer technology and wearables, where the 'X' represents a variable for any current or future spatial computing technologies. XR is the family technology that includes Virtual, Augmented and Mixed Reality (MR) either utilized individually or together:

- Augmented Reality (AR) blends digital data with the real world by rendering artificial object in the real environment.
- Virtual Reality (VR) is the use of computer technology to create a simulated environment and placing the user inside the experience. Instead of viewing a screen, users are immersed and able to interact with the 3D world. In Virtual Reality, the computer uses similar sensors and calculations as in AR.
- Mixed Reality (MR) is the third category, classified between AR and VR and attempts to combine Virtual and Augmented Reality. Like augmented reality, mixed reality also places digital or virtual objects in the real world. However, with mixed reality, users can quickly and easily interact with those digital objects to enhance their experience of reality or improve efficiency with certain tasks.

XR is a rapid growing field being applied in a wide range of ways, such as entertainment, marketing, real-estate, training and remote work.

In this paper, we employ *Virtual Reality*; the technology has been touted for several years now as a technology likely to have a profoundly transformative effect on the way we live and work. It is not a new technology, but recent advances in computational power, storage, graphics processing, and high-resolution displays have helped overcome some of the constraints that have stood in the way of its widespread use. Today's most common use cases centre around consumer products in gaming and entertainment, in the industrial scene and manufacturing areas such as the automotive, aviation or military sector where they are used predominantly in engineering, training and servicing applications. VR is a technology that generates an environment at which physical presence of the user in places in the real world or imagined worlds can be simulated, with objects and scenes that realistically immerse the user in the surroundings; it allows for the user to interact with this generated world, hence it is also referred to as immersive multimedia or computer simulated life. VR creates artificial experiences of the senses, including sight, hearing and touch. State of the art virtual reality environments are displayed either on a computer screen or with special stereoscopic displays also called headsets; some experiences can contain sensory information in addition to the visual one and also use speakers to focus on

[1] https://swe.ssa.esa.int/ssa-space-weather-activities.

realistic sounds. In this technology, the position of the user's eyes are located within the simulated environment. If the user's head turns, the graphics react accordingly. Rather than compositing virtual objects and a real scene, VR technology creates a convincing, interactive world for the user (Fig. 3).

Another important aspect of Virtual Reality technology is the capability of producing *Synthetic Data*, or "any production data applicable to a given situation that are not obtained by direct measurement". Synthetic data is increasingly being used for Machine Learning (ML) applications, a subset of the Artificial Intelligence (AI) family of technologies. An ML model can be trained on a synthetically generated dataset with the intention of using the acquired knowledge to infer conclusions on real data. Using VR one can generate synthetic worlds where collected data can be used for a number of applications e.g. rover localization [5], spacecraft attitude determination [6] etc. Even though the use of synthetic data has not become ubiquitous yet, VR is expected to have a critical role in the production of AI-ready data in the future.

Data visualisation is an offspring of *Visual Communication*, or the bridge between the quantitative information in the data and the human intuition and understanding. Employing an immersive data visualization approach in Virtual Reality, one is able to:

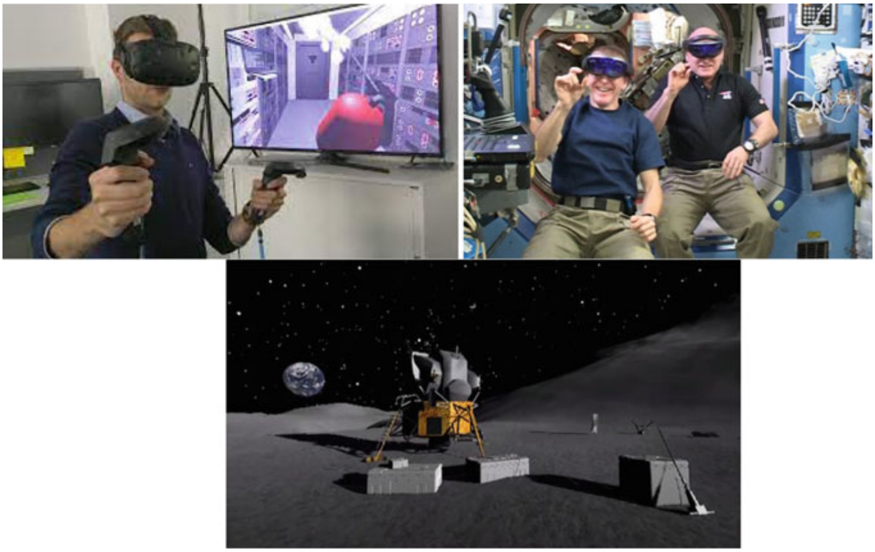


Fig. 3 (*Right*) Researcher with the European Space Agency in Darmstadt, Germany, equipped with a VR headset and motion controllers, demonstrating how astronauts might use virtual reality in the future to train to extinguish a fire inside a lunar habitat. (*Left*) Astronauts on-board the ISS experimenting with Augmented Reality headsets. (*Bottom*) Virtual lunar environment for simulations and trainings

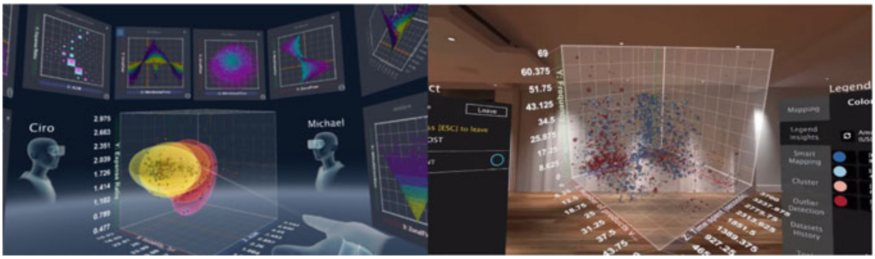


Fig. 4 Examples of next generation immersive data visualisation using Virtual Reality. © Virtualitics

- Make data comprehensible, contextualize results and get new key insights. Complex structures that may exist in 3 or more dimensions are lost when projected to a lower dimensionality display.
- Manipulate, review, interact with the data: ‘Pick up’ a dataset and move around to compare with another.
- Explore new ways to communicate and share findings.
- Be *inside* the data and observe the internal structures from any angle, without only observing the data from outside or see pre-selected 2D projections of the internal structures (Fig. 4).

1.2 Motivation

In Europe’s economy today, numerous sectors can be affected by Space Weather. These range from space-based telecommunications, broadcasting, weather services and navigation, through to power distribution and terrestrial communications, especially at northern latitudes. Space Weather affects global technological systems and societies. Space Weather, or the dynamic conditions on the Sun and in the space environment produce coronal mass ejections, solar energetic particles, and trigger geomagnetic disturbances that impact sensitive technological systems on Earth and in space.

- *Satellite Navigation Services.* One significant effect of the solar activity is found in the disturbances in satellite navigation services, like Galileo, due to SWE effects on the upper atmosphere. This, in turn, can influence industries like aviation, road transport, shipping and any other activities that depend on precise positioning or accurate time synchronisation.
- *Satellites and Humans in Orbit.* For satellites in orbit, the effects of Space Weather are found in the degradation of communications, performance, reliability and overall lifetime. An important example are the solar panels that convert sunlight to electrical power; on most spacecraft, they will be led to the generation of less power over the course of a mission, a degradation which must be taken into account in designing the satellite. Solar particle events substantially accelerate

the solar panel degradation and active space weather conditions can shorten the satellite lifetime. In addition, increased radiation due to Space Weather may lead to increased health risks for astronauts on board the International Space Station in low orbit, and particularly in future on voyages to the Moon or Mars.

- On *Earth*, commercial airlines may also experience disturbances in the navigation and communication, glitches in the aircraft electronics and increased radiation doses to crews (at long-haul aircraft altitudes) during large SWE events. Space weather effects on ground can include damage and disruption to power distribution networks, increased pipeline corrosion and degradation of radio communications.

The Space Weather System is producing a huge amount of data products from hundreds of SWE sensors on ground and in space. The task of a *SWE forecaster* is to use these products to make a qualitative and quantitative Space Weather nowcast and forecast. Being able to monitor SWE events, analyse the measurement data and quickly assess the risk from the event for the infrastructure in space or on ground is critical for issuing alerts to the end users and taking action to mitigate the impacts.

Despite the constantly advancing numerical simulation techniques and Artificial Intelligence (AI) based analysis of the measurement data, human interpretation of the simulation results and data products still provides the best SWE nowcasts and forecasts. Data visualization is crucial for understanding the outputs of numerical space weather models and efficient risk assessment. Today, the main means of visualization of SWE phenomena used by scientists and forecasters are 2D colour maps projecting the propagation of heliospheric plasma on two orthogonal planes[2] (Fig. 5). Intuitive data visualisation tools providing the forecaster a deeper understanding of the solar and heliospheric phenomena in progress are considered as one of the key techniques to improve the accuracy of the space weather forecasts in the near future.

Up-to-date Space Weather information will need to be provided to the end users operated via service providers e.g. Space Weather operators/forecasters. With multiple views of solar activity, scientists can better track the evolution and propagation of solar eruptions, with the goal of improving our understanding of space weather. To better understand the fundamental processes that drive these events, and ultimately improve space weather forecasts, many observation systems on ground and in space monitor the Sun around the clock by different sensors. Each measurement provide unique information about the structures and dynamics in the Sun's behaviour, giving researchers an integrated picture of the conditions driving space weather. Introducing innovative technologies like Augmented and Virtual Reality can result in tools that will allow for an immersive visualisation and will be deployed in multiple forecast centres across Europe while bringing new technology in SWE forecasting closely linked to mission operations in ESOC.

More specifically, introducing VR technology in the SWE analysis process, we are aiming at:

[2] https://swe.ssa.esa.int/ssa-space-weather-activities.
https://swe.ssa.esa.int/fmi-tomoscand-federated.
https://swe.ssa.esa.int/swaci-federated.

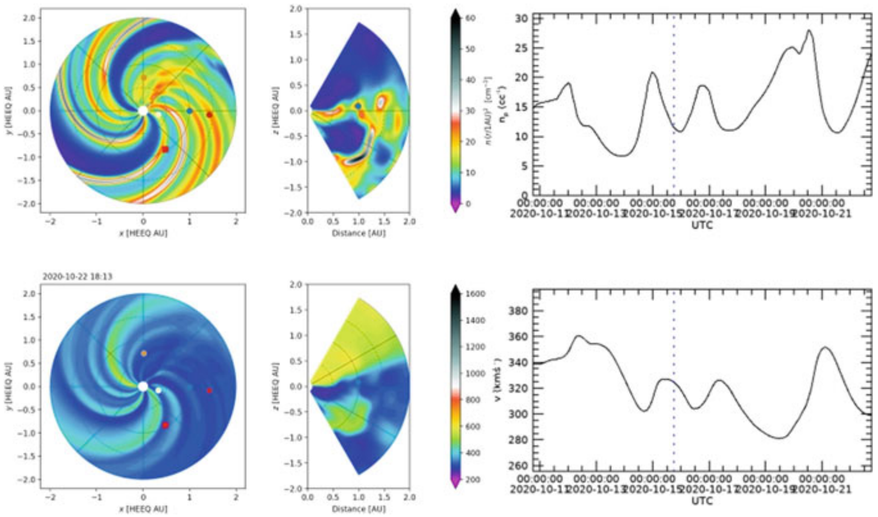


Fig. 5 2D presentation of the plasma density (upper row, **n** – $(r/1\text{ AU})^2$ [cm^{-3}]) and velocity (lower row, km/s) in the solar wind from *EUHFORIA* numerical 3D MHD model for the same day. *Figure Source* https://swe.ssa.esa.int/current-space-weather

- Supporting forecast model development by increasing knowledge about the propagation and unfolding of Coronal Mass Ejections, which is not yet established.
- Providing a visualisation for large dense datasets: Achieve transparency without jeopardizing the amount of information depicted.
- Having a Coronal Mass Ejection in our hands.

The presented prototype is expected to provide Space Weather experts with tools that can act as an enabler of improved SWE forecasts, support the relevant educational and public relations activities, raising awareness and communicating in an intuitive manner main principles of the SWE segment (Fig. 6).

2 Bringing Space Weather Data to Virtual Life

2.1 EUHFORIA Model

Remote sensing data (Fig. 7) are being processed in order to develop sophisticated models that will allow for accurate forecasting of space weather events and their effects to both space and ground segments but also on the surface of the Earth. The European Forecasting Information Asset, or EUHFORIA, is a model developed at KU Leuven that allows for the prediction of Coronal Mass Ejections. Detailed

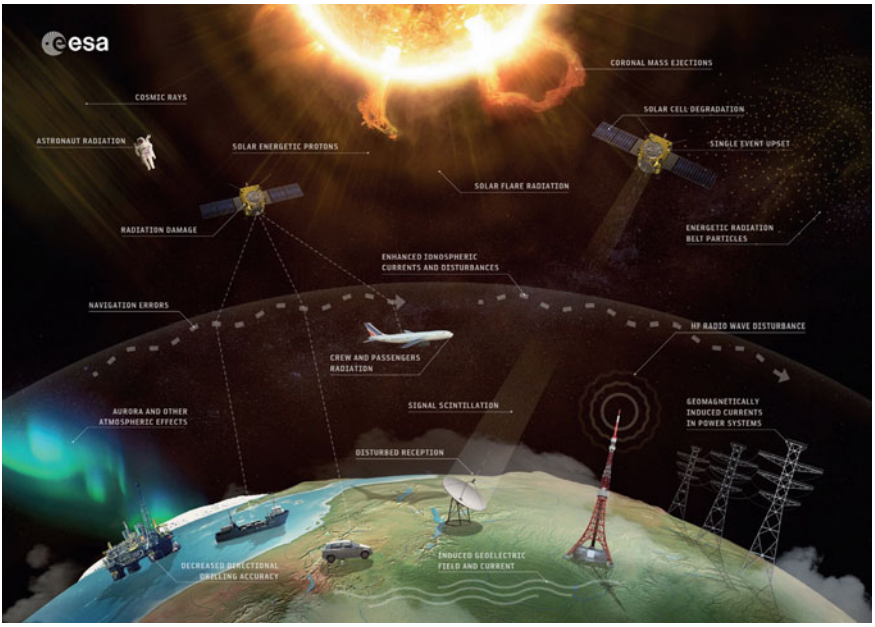

esa
CORONAL MASS EJECTIONS
COSMIC RAYS
SOLAR CELL DEGRADATION
ASTRONAUT RADIATION
SOLAR ENERGETIC PROTONS
SINGLE EVENT UPSET
SOLAR FLARE RADIATION
RADIATION DAMAGE
ENERGETIC RADIATION BELT PARTICLES
ENHANCED IONOSPHERIC CURRENTS AND DISTURBANCES
NAVIGATION ERRORS
HF RADIO WAVE DISTURBANCE
CREW AND PASSENGERS RADIATION
SIGNAL SCINTILLATION
AURORA AND OTHER ATMOSPHERIC EFFECTS
GEOMAGNETICALLY INDUCED CURRENTS IN POWER SYSTEMS
DISTURBED RECEPTION
INDUCED GEOELECTRIC FIELD AND CURRENT
DECREASED DIRECTIONAL DRILLING ACCURACY

Fig. 6 Effects of the space weather phenomena

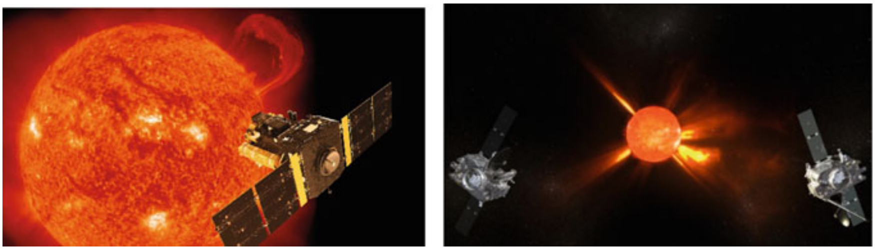


Fig. 7 Depiction of ESA/NASA SOHO mission (left animation) and NASA's STEREO mission (right animation), from which data were collected to produce the EUHFORIA model

descriptions of the model's structure and development methodology can be found in [7, 8].

The elements which are of interest for the VR visualisation regard the outputs of the model, which is a discrete dataset of samples that include information for the full 3D space (in heliocentric coordinates). For each sample, values for the plasma velocity, density, pressure and magnetic field (value and direction) were collected for 4 epochs per day, every six hours *(00, 06, 12, 18)*. The output is stored in numerous compressed files, requiring a pre-processing phase to be transformed to a format suitable to be inserted in the game engine for visualisation.

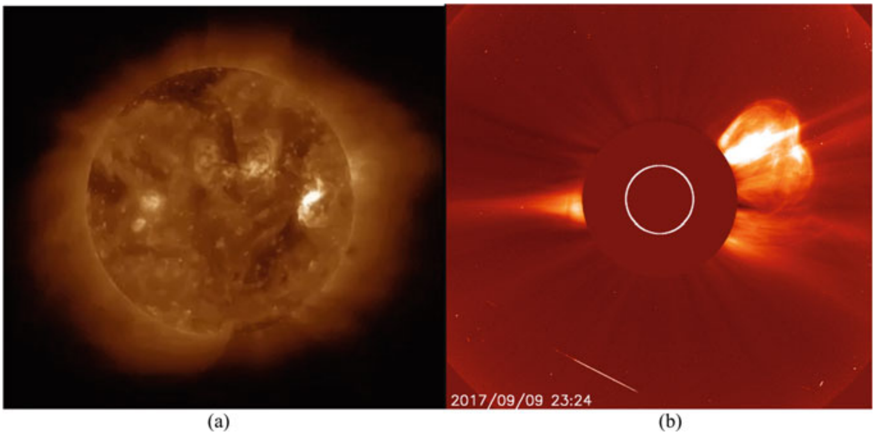


Fig. 8 A coronal mass ejection as captured by **a** NOAA's GOES, **b** ESA/NASA's SOHO mission

2.2 September 2nd–10th 2017: Intense Solar Activity

From 4 to 6 September 2017, heliospheric activity suddenly and drastically increased; the event was initiated from a simple Sunspot which transformed into a complex region ending up in the Sun emitting tens of solar flares and releasing a number of strong Coronal Mass Ejections [9] between September 6th–10th. The flares travelled across the Sun towards the Earth in concert with the Sun's normal rotation. On September 9th, a CME erupted from the Sun before the next day, when an even bigger CME was observed travelling away from the Sun at very high speeds, and was one of the fastest CMEs ever recorded. This CME was not directed towards the Earth but side-swiped Earth's magnetic field, and therefore did not cause significant geomagnetic activity (Fig. 8).

2.3 Proof-of-Concept

The aim of the proof-of-concept that we present in this paper is to be able to visualise the propagation of the plasma from a solar event as well as its interaction with background solar wind in a 3D environment, increasing knowledge about the propagation and unfolding of CMEs which pose a threat to human activities both on Earth and in space. Previous works had made steps towards this direction with the production of a VR experience through a solar storm without interacting with the phenomenon, but only observing it.[3] Another approach is the system presented in [10] where the authors transform the data into 3D space. At ESOC, the Virtual Reality tool was implemented in house taking this idea one step forward by presenting 3D data in VR

[3] https://www-perso.ias.u-psud.fr/solarstormvr/.

as shown in Fig. 9. With this tool, we are able to visualize historic spatiotemporal data from the European EUHFORIA MHD model showing the events that took place on September 2017, in view of providing a prototype that can be used for forecast and risk assessment purposes.

More specifically, the user has the chance to experience a solar storm event and view the evolution of the plasma cloud. The approach followed was the visualization of 3D temporal colour maps, due to its closeness with what the scientists are familiar

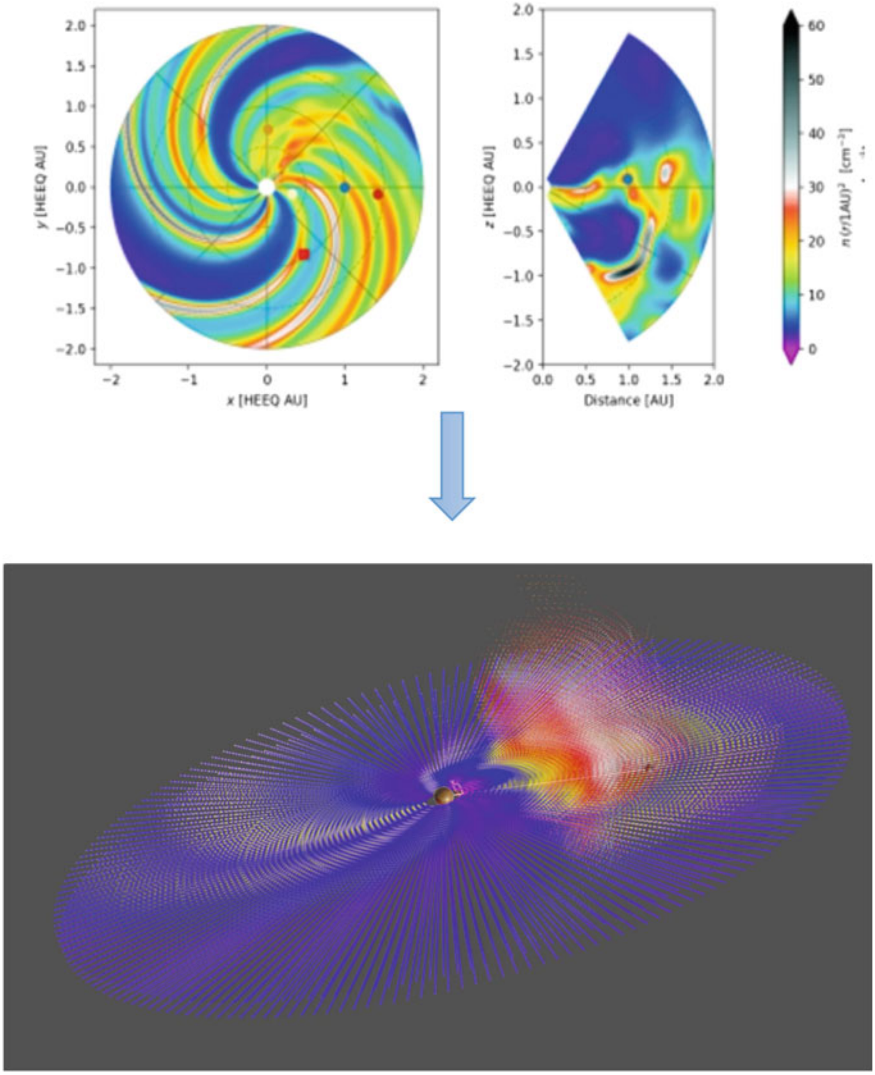


Fig. 9 Visualisation of a CME in unity game engine

with. The tool can be used as a means of a visual comparison among different model parameters or even among different models. The functionality of the tool was selected in close collaboration with potential users from the ESOC Space Weather Office, KU Leuven (EUHFORIA Model developers) and the Royal Observatory of Belgium (Space Weather Forecasters). The capabilities range from the navigation forward and backwards in time by means of epoch steps, to the filtering of the visualised data points at runtime based on their values, etc. A number of challenges were also identified, e.g. handling the volume of the dataset, achieving transparency without jeopardizing the amount of information depicted, etc.

2.4 *Tools and Hardware*

The tool was developed using the dedicated VR equipment at ESOC, in the Advanced Ground Software Applications (AGSA) Lab. The game engine that was used for the design of tool was *Unity*. It is a cross-platform game engine, that gives users the ability to create games and experiences in both 2D and 3D. The scripting API in C# offered by the engine was used for the implementation of the project. In order to view the visualisation, the user has access to an HTC Vive, a virtual reality headset which uses "room scale" tracking technology, allowing them to move in 3D space and use motion-tracked handheld controllers to interact with the environment. The high level architecture of the implemented tool can be found on Fig. 10.

3 Analysis of Initial Results

The implemented tool offers functionalities to the user which allow for the traditional views depicted on Fig. 5 or more complex views as the one shown on Fig. 15a

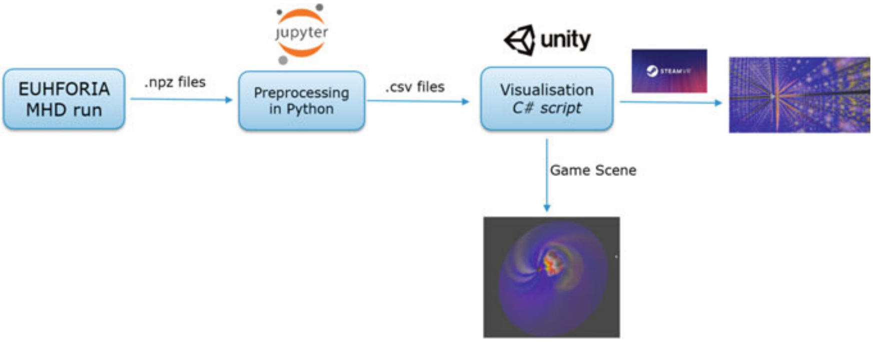


Fig. 10 High-level description of the system architecture

to be presented in an immersive way. Below we summarise the most important functionalities implemented in the prototype:

- *Navigation forward and backward in time by means of epoch steps*: It is of critical importance for the scientists investigating the datasets to be able to move forward and backwards across the different epochs and study the evolution of the phenomena from different angles.
- *Filtering on runtime based on user input values.* The user is able to alter the range of the parameter values depicted by means of setting the maximum and minimum allowed depicted values. The colour scale shall not be adjusted to the new range. In this way, specific phenomena e.g. solar wind or the CMEs can be isolated and investigated.
- *Adjusting intensity/resolution at runtime.* The user is able to highlight specific areas on the 3D space, based on their values by setting the maximum and minimum highlighted values. The data points outside this range will still be depicted, but not highlighted, hence the contrast between the areas of interest and the rest information will be increased. The resolution of the depicted data is allowed to increase/decrease by a factor of 2 on either of the 3 axes: radial, longitude, latitude.
- *Connect Sun—planet under investigation.* A 3D line connection between the Sun and the planet under investigation enables the user to observe the phenomena unfolding in the surrounding area and identifying the effect on the planet.

3.1 Traditional Views Revisited

This approach supports immersive VR presentation that allows the forecaster to look inside the structure of the plasma field, zoom in and inspect small scale structures. The visualisation of the plasma density, velocity and magnetic field together in the same view for a holistic view of the plasma cloud status in an intuitive way, is one of the main challenge of such a visualisation (Figs. 11 and 12).

3.2 Filtering

See Fig. 13.

3.3 Adjusting the Contrast

See Fig. 14.

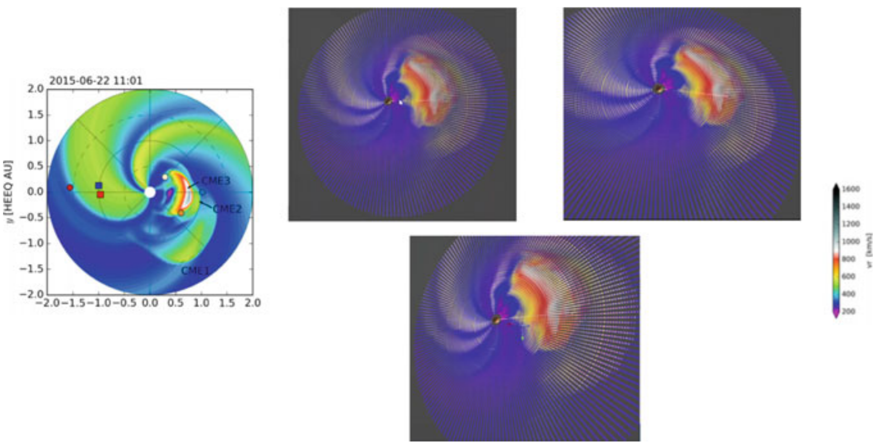


Fig. 11 A demonstration of 3D presentation of the plasma velocity from EUHFORIA model

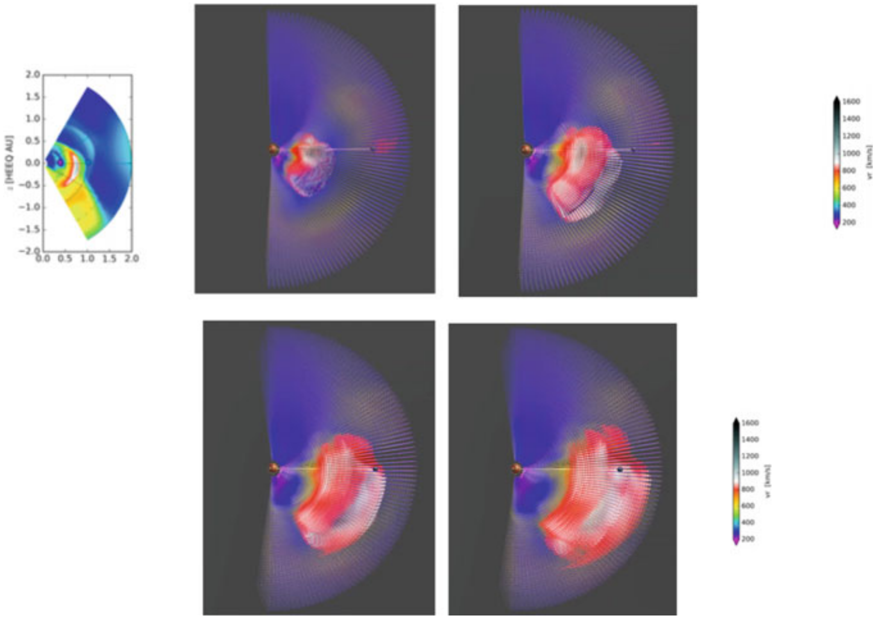


Fig. 12 The evolution of the CME in time

3.4 *"Combination" View*

See Fig. 15.

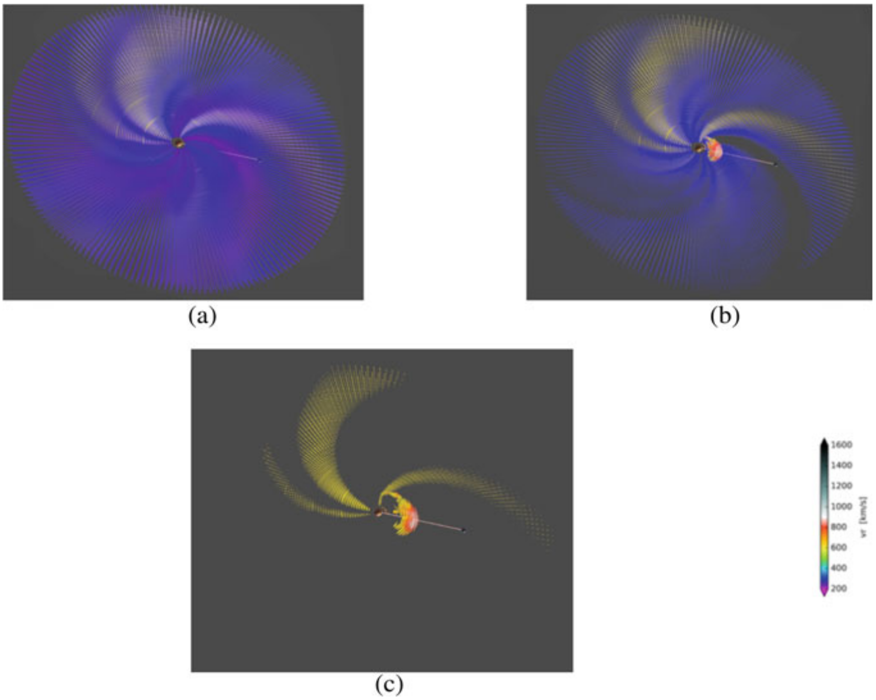


Fig. 13 Different stages of applying low pass filtering in the visualized dataset

3.5 "Inside" the Data

In this view, the user is able to view the Coronal Mass Ejection, or any other visualized phenomenon while being among the data points. Hence, the user can observe the internal microstructures from any angle and gain further insights on the evolution of the phenomenon (Fig. 16).

4 Discussion and Future Works

A number of challenges were identified during the course of the project. The provided dataset consists of a discrete set of points with different values of plasma density, radial velocity etc. The positions of the points do not cover the full continuous 3D space. In order to avoid distorting the information depicted, interpolation among the data points should not be performed, hence the output visualisation is of a *3D grid* form. At the same time, in order to allow for a maximised amount of accurate information to be depicted, ***sampling*** should also be performed very carefully; the user should have the capability to sample the 3D space, by means of increasing/reducing

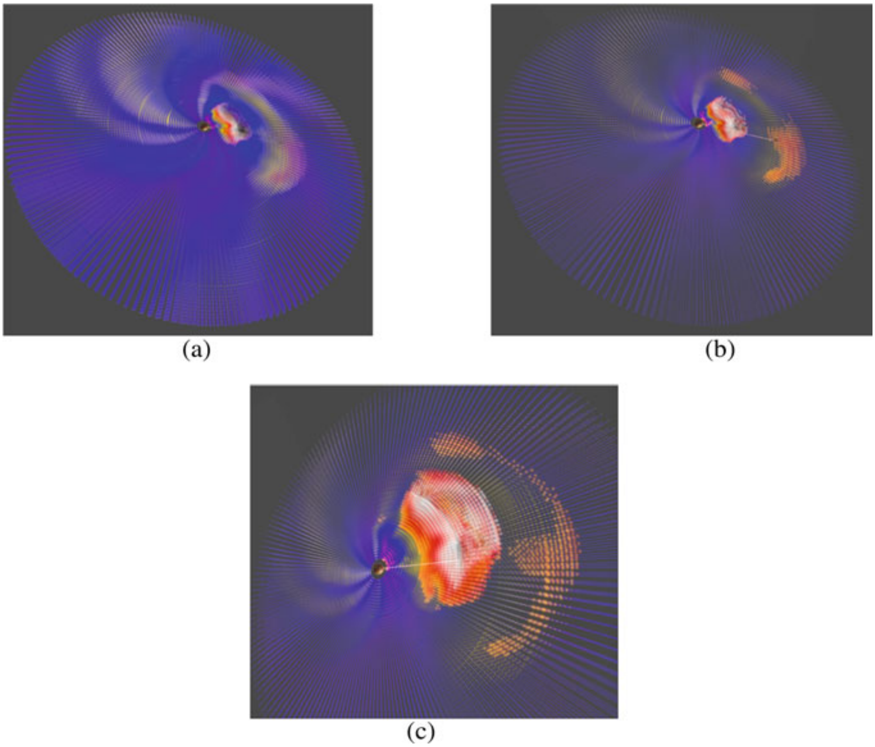


Fig. 14 Adjusting the contrast between data points to increase the focus on specific areas where the events are taking place, without removing the information contained in the remaining 3D space

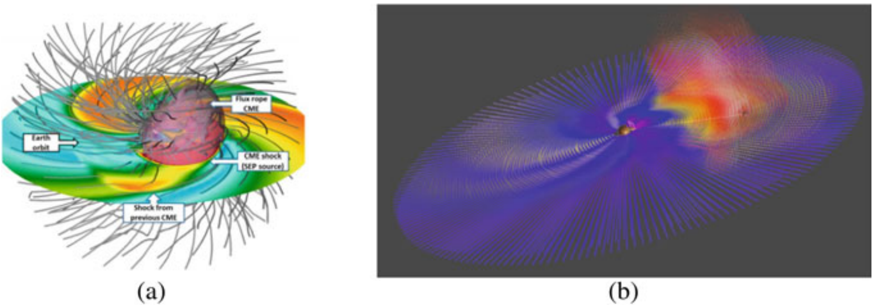


Fig. 15 **a** Visualisation of the equatorial plane and a shockwave. **b** Visualising equatorial plane and the area around the CME low pass filtered, demonstrating a future visualization of the shockwave

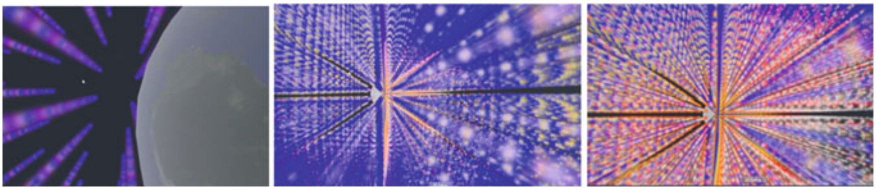


Fig. 16 Evolution of the CME, while the user is placed near the Earth (left), looking towards the Sun (middle) and unfolding the CME phenomenon in time steps (right)

the number of data points shown on radial, latitude or longitude axes. The ***dataset size*** (~40 GB/day) is also sufficiently large to make the rendering process too heavy for the processing unity. Employing colour maps as a means of visualisation comes with inherent challenges as to what type of ***colour scale*** will increase the ability of the user to extract the maximum amount of information from the figure. When using VR, an additional challenge identified is that in order to allow for a high quality user experience, the user should not be overwhelmed in a 'sea of colours' but rather be provided with a level of ***transparency***. A challenge identified during the project was the validation of the visualisation e.g. investigating whether ***artefacts*** are produced due to the view angle of the user. This falls under the wider challenge in the domain of 'Big Data' of ***visualising multi-dimensional, vast data sets*** in a way which allows users to extract insights.

Additional functionality is expected to improve the usability of the tool even further, either by visualising additional domain specific information or by allowing for the user to interact with the data in diverse ways. The visualisation of the ***magnetic field*** and the ***radial velocity gradient*** are top candidates for additional information to be added to the existing plasma velocity visualised information. The magnetic field can be visualised by means of lines or a vector field which will allow for the magnetic field evolution inside the CME-CME interaction to become visible. The shockwave (Fig. 15a) that causes the geomagnetic storm will be a crucial piece of information. Other functionality that will enable maximised information extraction is that of ***measuring distances*** between two points in the 3D space, e.g. using the hand controllers to point in the origin and destination points, or point at areas of interest and ***access relevant information*** in text, charts, graphs etc. Investigating ***real-time pipelines*** for spatiotemporal datasets, allowing for visualisation of incoming streams of data instead of stand alone datasets will increase the applicability of such a tool to a number of domains where multi-dimensional data are continuously collected. Bearing in mind the special equipment needed for VR visualisations and looking into bridging the gap from the currently 3D desktop based visualisation tools, the ***integration with 3D tools*** for desktop users without access to a Virtual Reality equipment will be investigated. At the same time, the final tool should be compatible with other existing Space Weather forecasting models, such as iPIC3D [11].

References

1. Developing the Scientific Basis for Monitoring, Modelling and Predicting Space Weather - COST Action 724 (2009). https://www.cost.eu/publications/developing-the-scientific-basis-for-monitoring-modelling-and-predicting-space-weather/
2. Bothmer V, Daglis I (2007) Space weather physics and effects. Springer
3. Community Coordinated Modeling Center, Integrated Space Weather Analysis System. https://ccmc.gsfc.nasa.gov/iswa/
4. ESA SSA Space Weather Service Network, Heliospheric Weather Expert Service Centre (H-ESC). http://swe.ssa.esa.int/heliospheric-weather
5. Franchi V, Ntagiou E (2022) Planetary rover localisation via surface and orbital image matching. In: The IEEE aerospace conference 2022
6. Proenca P, Gao Y (2020) Deep learning for spacecraft pose estimation from photorealistic rendering. In: The international conference on robotics and automation (ICRA)
7. Pomoell J, Poedts S (2018) EUHFORIA: European heliospheric forecasting information asset. J Space Weather Space Clim 8. https://doi.org/10.1051/swsc/2018020
8. Scolini C, Chané E, Temmer M, Kilpua EKJ, Dissauer K, Veronig AM, Palmerio E, Pomoell J, Dumbović M, Guo J, Rodriguez L, Poedts S (2020) CME-CME interactions as sources of CME geo-effectiveness: the formation of the complex ejecta and intense geomagnetic storm in early September 2017. Astrophys J Suppl 247(1)
9. Kahler SW (1992) Solar flares and coronal mass ejections. Ann Rev Astron Astrophys 30:113–141
10. Travel through a solar storm: a virtual reality experience (2016). https://www-perso.ias.u-psud.fr/solarstormvr/
11. Markidis S, Lapenta G, Rizwan-uddin (2010) Multi-scale simulations of plasma with iPIC3D. J Math Comput Simul 80(7)

PINTA—One Tool to Plan Them All

Rainer Nibler, Jens Hartung, Jonas Krenss, Anna Fürbacher, Falk Mrowka, and Sandra Brogl

Abstract In the recent years, the "Program for INteractive Timeline Analysis" PINTA, developed at the German Space Operation Center (GSOC), was continuously improved and experienced several evolution steps. PINTA is a GUI application running on Windows-based computer systems, whose main purpose is to serve as the anchor tool for a mission planning operation's engineer when generating, modifying or analysing a mission timeline. This is supported by calling automatic planning algorithms of the embedded generic planning library "PLAnningTOol" PLATO, using input of the embedded orbit propagation and event calculation library "SpaceCraft Orbit and GroundTrack Analysis Tool" SCOTA, or its expandability through plugins. PINTA is the generic basis of many semi-automated mission planning systems for past, current and future spacecraft projects operated at GSOC. It is used or has been used for the missions Grace, TET-OOV, FireBird, Grace-FollowOn, Eu:CROPIS and is currently prepared for CubeL. Furthermore, PINTA serves as the timeline analysis tool for validating the TerraSAR-X/TanDEM-X mission planning system. The variety of use cases was further extended to support Launch and Early Orbit Phases (LEOPs) in its special "SoEEditor" configuration as the new generic editing tool for the so-called "Sequence of Events". It was successfully used for the satellites Biros, HAG-1, PAZ, Grace-FollowOn 1 and Grace-FollowOn 2, Eu:Cropis, EDRS-C and is

R. Nibler (✉) · J. Hartung · J. Krenss · A. Fürbacher · F. Mrowka
Deutsches Zentrum für Luft-und Raumfahrt, GSOC, Weßling, Bayern 82234, Germany
e-mail: Rainer.Nibler@dlr.de

J. Hartung
e-mail: Jens.Hartung@dlr.de

J. Krenss
e-mail: Jonas.Krenss@dlr.de

A. Fürbacher
e-mail: Anna.Fuerbacher@dlr.de

F. Mrowka
e-mail: Falk.Mrowka@dlr.de

S. Brogl
Spaceopal GmbH, München, Bayern 80336, Germany
e-mail: sandra.brogl@spaceopal.com

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_16

currently in preparation for EnMAP. In addition to LEOP's, the SoEEditor was also capable of supporting the constellation maneuvers for the TerraSAR-X/TanDEM-X mission. Besides all these use cases, the paper at hand will especially describe how PINTA was even further extended to not only tackle spacecraft-based but also ground-based scheduling. On the one hand it serves as an "On-Call Tool" to support the on-call shifts by automatically generating conflict-free role-based shift plans for all subsystems by considering various constraints like person outages, working hours, role-conflicts, etc. The plan can then be further adapted manually to cope with user change-requests. On the other hand it is used as a "Multi-Mission-Control-Room-and-pass-Scheduler" (MuMiCoRoS) to coordinate the ground-station booking of all LEO (low-earth orbit) satellites: TerraSAR-X, TanDEM-X, TET, Biros, Grace-FollowOn 1 & 2 and Eu:CROPIS. In order to avoid ground-station and operator conflicts between the missions, an automatic and combined plan for all satellites is generated which can then be further modified manually if necessary. As another use case, PINTA (a.k.a. GPT; Galileo Planning Tool) supports the Galileo Service Operation (GSOp). The planning process involves three timelines: a Short-Term Plan (STP), covering the next ten days, two Mid-Term Plans (MTP) for the Operational (OPE) and the Validation (VAL) chain), covering the next 15 weeks, and a Long-Term Plan (LTP), covering the next 15 months. The activities in these timeframes cover all subsystems of Galileo: Flight Ops, Control segment, Mission segment, remote sites, service operations, hardware, software, hosting, network, etc. In order to support the GSOp, numerous additional features, like importers, exporters, interfaces and plugins had to be added to PINTA.

Keywords PINTA · Planning systems · Sequence of events · Galileo planning tool

Acronyms/Abbreviations

BIROS	Bispectral InfraRed Optical System
CCSDS	Consultive Committee for Space Data Systems
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.
EDRS-C	European Data Relay Satellite C as part of EDRS (European Data Relay System)
Eu:CROPIS	Euglena and Combined Regenerative Organic-food Production in Space
GCC	Galileo Control Center
GEO	Geostationary Orbit
GfR	Gesellschaft für Raumfahrtanwendungen mbH
GPT	Galileo Planning Tool
GNSS	Global Navigation Satellite System
GRACE	Gravity Recovery and Climate Experiment
GFO	Gravity Recovery and Climate Experiment–Follow On
GS	Ground Station

GSA	GNSS Service Agency
GSOC	German Space Operations Center
GSOp	Galileo Service Operation
GUI	Graphical User Interface
HAG-1	Hispasat Advanced Generation 1, first Satellite of the OHB SmallGEO generation
LEO	Low Earth Orbit
LEOP	Launch and Early Orbit Phases
LTP	Long-Term Plan
OPE	Operational chain
PINTA	Program for INteractive Timeline Analysis
PLATO	Planning Tool
FOP	Flight Operation Procedure (PROTOS Output (Reference))
MMU	Mass Memory Unit
MPS	Mission Planning System
MTP	Mid-Term-Plan
MuMiCoRoS	Multi-Mission-Control-Room-and-pass-Scheduler
PAZ-1	First Spanish Earth radar observation satellite
PR	Planning Request
SCOTA	Space*C*raft Orbit and groundTrack Analysis tool
SDT	Service desk tool
SoE	Sequence of Events
SPOT	Swath Preview and Ordering Tool
SSF	Saved Stack File (SCOS/GECCOS Output–Reference)
STP	Short-Term-Plan
TanDEM-X	TerraSAR-X-Add-on for Digital Elevation Measurements, “X” for the X frequency band
TerraSAR-X	Terra (Latin for Earth), SAR: Synthetic Aperture Radar, “X” for the X frequency band
TET	Technologie-Erprobungsträger
TSTD	TerraSAR-X/TanDEM-X
V3C	Verlegefähiges Compact-Control-Center
VAL	Validation chain

1 Introduction

The application “**PINTA**” has a history of up to two decades serving for various spacecraft missions operated at the “**German Space Operations Center**” (GSOC). It went through a constant change process which was necessary due to new missions with different requirements. In order to get a better overview of how PINTA and the surrounding mission planning framework, that complements its usability, has evolved, the key components will be first described within Sect. 2 below. Because

PINTA was originally intended to serve as the anchor tool for a "**Mission Planning System**" (MPS), which helps the operations engineer when generating, modifying or analysing a mission timeline, Sect. 3 will deal with this in more detail. This will contain all past and current mission planning systems in which PINTA has played or is still playing a major role.

Another area of application of PINTA, which has been established in recent years, is its "**SoEEditor**" functionality for preparing and maintaining the Sequence of Events before and during the launch and early orbit phases (LEOP) for low-earth as well as geostationary satellite missions at GSOC. Its use case as a SoEEditor will be described in more detail in Sect. 4.

Although PINTA's original purpose was to plan satellite timelines, the paper at hand will especially describe how it was made possible, to not only tackle spacecraft-based but also ground-based scheduling problems.

On the one hand it serves as an "**On-Call Tool**" to support the on-call shifts by automatically generating conflict-free role-based shift plans for all subsystems by considering various constraints like person outages, working hours, role-conflicts, etc. which will be the topic in Sect. 5.

On the other hand it is used as a "**Multi-Mission-Control-Room-and-pass-Scheduler**" (MuMiCoRoS) to coordinate the ground-station booking of all LEO (low-earth orbit) satellites: TerraSAR-X, TanDEM-X, TET, Biros, Grace-FollowOn 1 & 2 and Eu:CROPIS. In order to avoid ground-station and operator conflicts between the missions, an automatic and combined plan for all satellites is generated which can then be further modified manually if necessary. See Sect. 6 for more details.

PINTA, among various competitors, was the tool of choice by the Spaceopal Planning Team, to support the Galileo Service Operation (GSOp) since 2018. This led to a further and previously unknown area of application in which it is now also used as the "**Galileo Planning Tool**". Section 7 gives a deeper insight into this completely new use case and the associated challenges.

Last but not least, in Sect. 8, an outlook to the future challenges and milestones for the development of PINTA, as well as further projects at GSOC, like the successor "**PintaOnWeb**", is given.

2 Components

2.1 PINTA

The application PINTA is named after the fastest of the three ships used by Christopher Columbus in his first voyage across the Atlantic Ocean, La Pinta ("the spotted one") [1] and is also an abbreviation for "**Program for INteractive Timeline Analysis**".

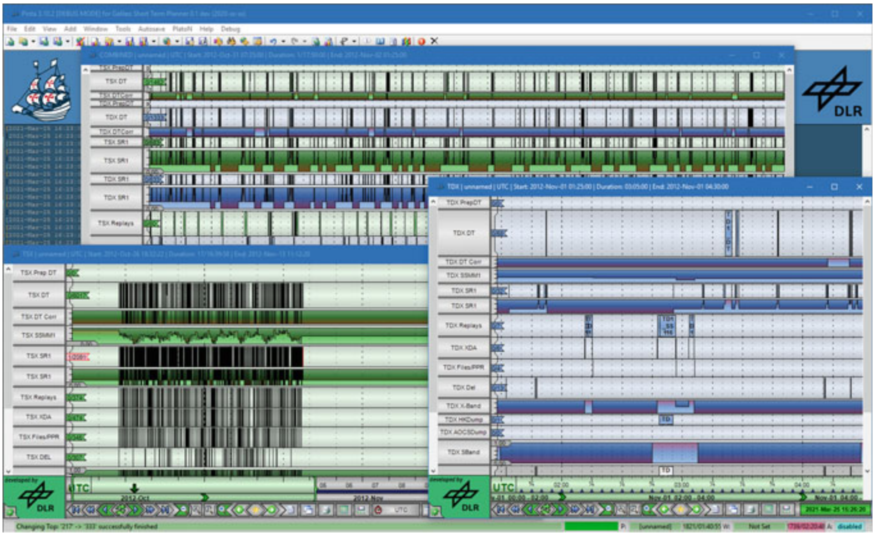


Fig. 1 The PINTA GUI with three different views, to visualize the TSX/TDX mission

PINTA is an interactive GUI application running on Windows-based computer systems. Its main purpose is to serve as the anchor tool for a mission planning operation's engineer when generating, modifying, validating or analysing a mission timeline. An example, how a timeline of the TerraSAR-X/TanDEM-X mission can be visualized with PINTA, is shown in Fig. 1.

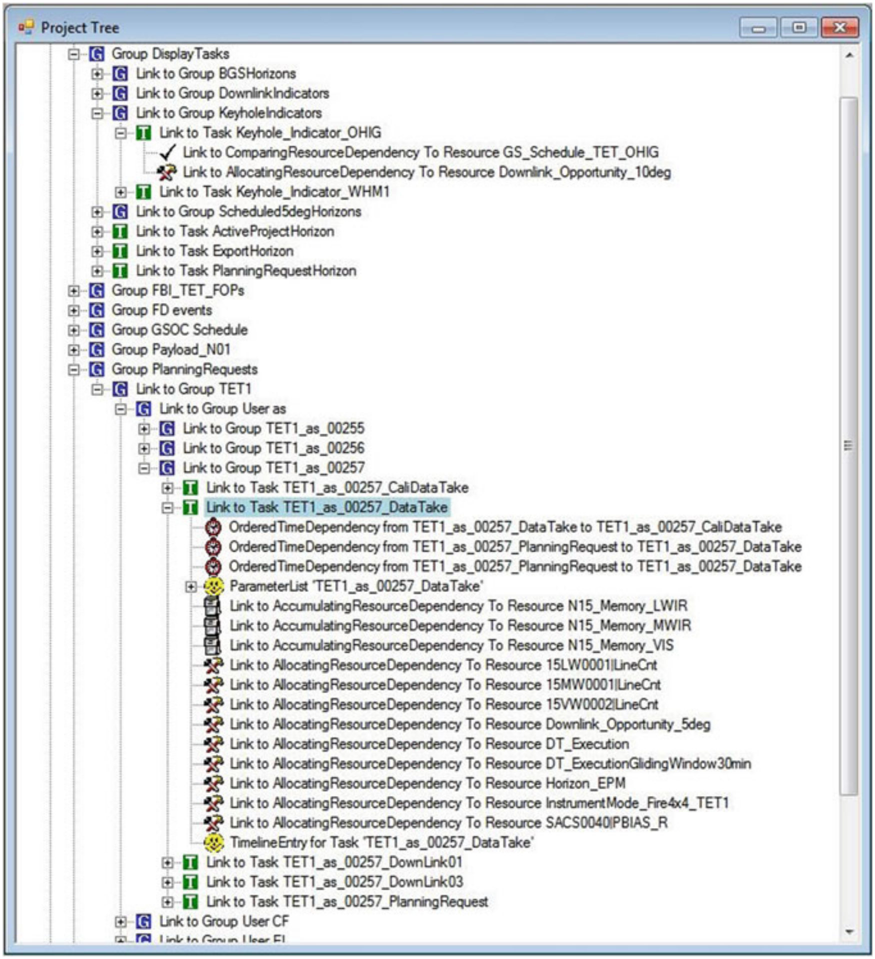
The internal PINTA data structure is called the GSOC planning model or simply the "**Project**". In [2], a much more detailed overview of the modelling language can be found. For a better understanding, the most important objects and relations should be briefly outlined.

Each project consists of generic objects such as "**Groups**", "**Tasks**", "**Parameters**" and "**Resources**". These basic objects can in turn be provided with various conditions as well as interdependencies. The central role in every project are the Tasks, which can be scheduled or un-scheduled by adding/removing "**TimelineEntries**" to/from the Project's "**Timeline**". Furthermore, they can be arranged in hierarchical structures with the help of Groups. The most important constrains are "**OrderedTimeDependencies**" between the Tasks, "**UpperBounds**" respectively "**LowerBounds**" of the Resources and "**Allocating-**", "**Accumulating**"- and "**Comparing-ResourceDependencies**" between the Tasks and Resources, with all but the inter-Task constraints being specified via "**Profiles**" over time.

All the dependencies as well as the Resource modifications/comparisons are updated as soon as a Task is scheduled, un-scheduled or modified. In addition, every further task, including its dependencies, as well as the Resources themselves, that are influenced by the change, can indicate conflicts whether or not the modification led to a conflict-free new Project state.

The typical planning model is composed and displayed in PINTA via the so-called "**Project Tree**". As an example, an excerpt of the planning model for the FireBird mission can be seen in Fig. 2.

In addition to an easy to understand and clear visual representation and the possibility of interactive editing, the possibility to call automatic planning algorithms of the embedded generic planning library PLATO ("**PlanningTool**") or use input of the embedded orbit propagation and event calculation library SCOTA ("**SpaceCraft Orbit and GroundTrack Analysis Tool**") are also included. Furthermore, there is also the option of expanding the scope of functions in a mission-specific manner using a plugin interface.



```
Group DisplayTasks
  Link to Group BGSHorizons
  Link to Group DownlinkIndicators
  Link to Group KeyholeIndicators
    Link to Task Keyhole_Indicator_OHIG
      Link to ComparingResourceDependency To Resource GS_Schedule_TET_OHIG
      Link to AllocatingResourceDependency To Resource Downlink_Opportunity_10deg
    Link to Task Keyhole_Indicator_WHM1
  Link to Group Scheduled5degHorizons
  Link to Task ActiveProjectHorizon
  Link to Task ExportHorizon
  Link to Task PlanningRequestHorizon
Group FBI_TET_FOPs
Group FD events
Group GSOC Schedule
Group Payload_N01
Group PlanningRequests
  Link to Group TET1
    Link to Group User as
      Link to Group TET1_as_00255
      Link to Group TET1_as_00256
      Link to Group TET1_as_00257
        Link to Task TET1_as_00257_CaliDataTake
        Link to Task TET1_as_00257_DataTake
          OrderedTimeDependency from TET1_as_00257_DataTake to TET1_as_00257_CaliDataTake
          OrderedTimeDependency from TET1_as_00257_PlanningRequest to TET1_as_00257_DataTake
          OrderedTimeDependency from TET1_as_00257_PlanningRequest to TET1_as_00257_DataTake
          ParameterList 'TET1_as_00257_DataTake'
          Link to AccumulatingResourceDependency To Resource N15_Memory_LWIR
          Link to AccumulatingResourceDependency To Resource N15_Memory_MWIR
          Link to AccumulatingResourceDependency To Resource N15_Memory_VIS
          Link to AllocatingResourceDependency To Resource 15LW0001|LineCnt
          Link to AllocatingResourceDependency To Resource 15MW0001|LineCnt
          Link to AllocatingResourceDependency To Resource 15VW0002|LineCnt
          Link to AllocatingResourceDependency To Resource Downlink_Opportunity_5deg
          Link to AllocatingResourceDependency To Resource DT_Execution
          Link to AllocatingResourceDependency To Resource DT_ExecutionGlidingWindow30min
          Link to AllocatingResourceDependency To Resource Horizon_EPM
          Link to AllocatingResourceDependency To Resource InstrumentMode_Fire4x4_TET1
          Link to AllocatingResourceDependency To Resource SACS0040|PBIAS_R
          TimelineEntry for Task 'TET1_as_00257_DataTake'
        Link to Task TET1_as_00257_DownLink01
        Link to Task TET1_as_00257_DownLink03
        Link to Task TET1_as_00257_PlanningRequest
    Link to Group User CF
    Link to Group User FI
```

Fig. 2 The PINTA Project Tree displays an excerpt of the planning model for the FireBird mission

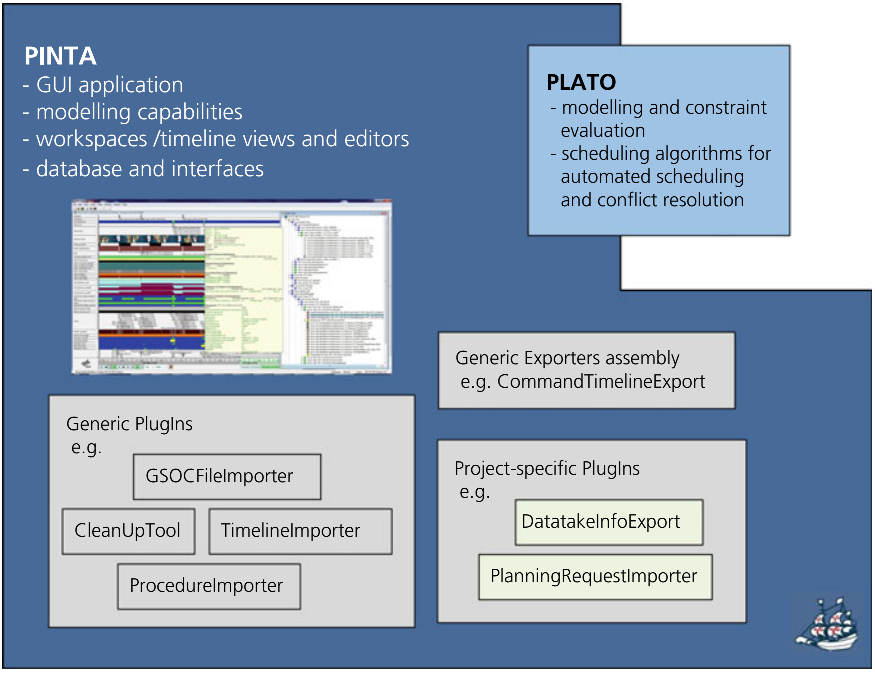

PINTA
- GUI application
- modelling capabilities
- workspaces /timeline views and editors
- database and interfaces
PLATO
- modelling and constraint evaluation
- scheduling algorithms for automated scheduling and conflict resolution
Generic Exporters assembly
e.g. CommandTimelineExport
Generic PlugIns
e.g.
GSOCFileImporter
CleanUpTool
TimelineImporter
ProcedureImporter
Project-specific PlugIns
e.g.
DatatakeInfoExport
PlanningRequestImporter

Fig. 3 The assembly of PINTA

Figure 3 gives an overview of the whole set-up of PINTA with some of its generic as well as project-specific plug-ins and its relation to our planning library PLATO.

2.2 *PLATO*

As already mentioned in the section before, PINTA allows for invoking PLATO (“**Planning Tool**”) algorithms. In general, all the fully automated as well as PINTA-based semi-automated mission planning systems at GSOC make use of these scheduling algorithms to automatically schedule and re-schedule indicator tasks, data-takes, downlinks, etc., depending on the mission-specific use case, in order to prepare a conflict-free timeline for the chosen planning horizon.

These scheduling algorithms however are not implemented from-scratch but are composed from the generic algorithm assembly available in the PLATO library. The core idea of this set of algorithms, filters, etc. is to have a reusable, configurable suite of functionalities that can be variously combined and configured and operate on a planning project available in the GSOC modelling language. A more detailed insight into this approach and an overview of most of the available algorithms and filters can be found in [3].

Regarding typical planning problems at GSOC, just some examples for basic algorithms shall be given here and outlined only very shortly to understand the principle:

- "**ChooseValuesToConsider**": The time range to be forwarded to a sub-algorithm can be determined via various criteria, e.g. respective to the execution time of scheduled horizon Tasks.
- "**ObjectSelection**": Various filters can be applied to determine to which sub-algorithms which of the currently considered Group(s) and/or Task(s) are to be forwarded in which order.
- "**ValueSelection**": When having found the next Task to schedule, various filters can be applied to determine whether and with which execution time it is allowed to be scheduled. The invocation of sub-algorithms with time ranges derived from the new TimelineEntry's execution time is possible.
- "**ConstraintIgnorer**": This allows temporarily deactivating constraints during the execution of a sub-algorithm, which can be necessary to handle circular dependencies. It enables scheduling and/or un-scheduling Tasks with a potential conflict first before trying to repair the solution by scheduling and/or un-scheduling other Tasks.

Further examples for the application of such PLATO algorithms can be found in more detail in [3].

2.3 SCOTA

The main task of SCOTA ("**SpaceCraft Orbit and GroundTrack Analysis Tool**") is to provide mission planning with an easy-to-integrate library that computes multiple orbit-related events. Some important examples are the calculation of a satellite's ground coverage ("swath") during a given time interval, the calculation of ground station contact times, or sun-related events such as the satellite moving in and out of the earth's shadow. These calculations require a library with different layers of complexity, including (but not limited to) modules for mathematics, coordinate and time systems, orbit propagation, and geodesics. A primary design goal of SCOTA is to provide this functionality to all projects supported by mission planning as a single generic library.

SCOTA consists of a layered architecture with the following core components:

- SCOTA.Core is the main assembly; it provides the complete SCOTA functionality through a C# API.
- The SCOTA.App stand-alone application can be started as a (Windows) executable to process input in XML format.
- Two generic SCOTA services encapsulate the access to the *SCOTA.Core* library and thus provide a higher layer of abstraction: SCOTA-Service.Web provides a web service, whereas *SCOTA-Service.ActiveMQ* links SCOTA calculations to an ActiveMQ message broker.

2.4 Plug-Ins

The plug-in interface is used to add further generic as well as mission-specific features to supplement the basic PINTA functionality. The most important generic plug-ins, that are frequently used in a typical mission planning system are the following ones:

- The “**CleanUpTool**”: Since in many projects, further planning is usually based on the results of previous planning, it is often necessary to clean up before or after each planning run, in order to avoid runtime problems due to excessive storage of information in the underlying database. It can be configured for which model content and at what point in time the cleanup should take place. This ensures that only data that no longer has any influence on the current planning status as well as future planning runs is deleted autonomously, while archiving previous states of the model.
- The “**GSOCFileImporter**”: This plug-in is executed to accept input files from other parts of the mission operations ground segment. New Two-Line-Elements and event files from Flight Dynamics, as well as so-called schedule files, are regularly received from the ground station’s planning office, and the latest version of each type is processed at the start of every planning run. The content of the event and schedule files is filtered according to the current configuration of the importer and inserted into the planning model as scheduled tasks, including all the constraints which modify the fill level profiles of the corresponding resources.
- The “**ExecutionTimelineImport**”: With this plug-in, an already finalized and complete timeline is loaded, which consists of so-called FOPs (Flight Operations Procedures) that were generated in a fully automated planning run, e.g. through the mission planning system used for the TerraSAR-X/TanDEM-X mission [4]. Importing all of this data into PINTA allows for visual cross-checking of the consistency and validity of the timeline, which is supported by a preconfigured set of constraints. Another similar use case was implemented for the FireBird mission, where this functionality was reused to ingest input XMLs from the principle camera investigator, which contain a FOP snippet to be used for so-called SystemOrders. For experimental acquisitions taken for these special requests the camera is then not commanded to switch to one of the standard acquisition modes, but instead it is configured with a FOP containing the ingested snippet.

Furthermore, there are numerous other corresponding plug-ins with very specific functionalities for the various mission planning systems. The only examples to be mentioned at this point are those for the FireBird mission, without going into too much detail: the “**PlanningRequestImporter**”, the “**DatatakeInfoFileGenerator**” and the “**EnvelopeRequestImporter**” [5].

Since the plug-in functionality with its dedicated API allows similar additional functions to be easily added, depending on the use cases of the respective project’s use cases, the plug-in mechanism has also proven to be very suitable for adding problem-specific functions when it is not part of a mission planning system of a spacecraft mission. Therefore, the plugin interface will play an important role in all following sections: Mission Planning Systems (Sect. 3), Sequence of Events Editor (Sect. 4), On-Call Tool (Sect. 5), MuMiCoRoS (Sect. 6) and Galileo Planning Tool (Sect. 7).

2.5 TimOnWeb

The graphical display tool TimOnWeb (“**Timeline On Web**”) is used to visualize the mission timeline, that was previously generated by PINTA—of course with the support of PLATO and SCOTA. It provides users, e.g. the flight directors, insight into the current status of the planning model through a website accessible from the World Wide Web.

Figure 4 shows a snapshot of the current TimOnWeb view for TET, one of the two satellites of the FireBird mission. With its help, all requested, ordered and actually planned activities with all their parameters, as well as ground station contacts, sun and shadow phases, can be displayed in a simple and clear manner, through freely configurable task plots. Further information, such as the utilization of hard disk partitions, battery capacities or even ground station elevation masks and ground tracks, can be plotted as well.

Unlike PINTA, TimOnWeb does not implement its own planning model representation. The generic server part is based on the PLATO library, so that extended functions of PLATO such as conflict tracking, filter algorithms, etc. can be used.

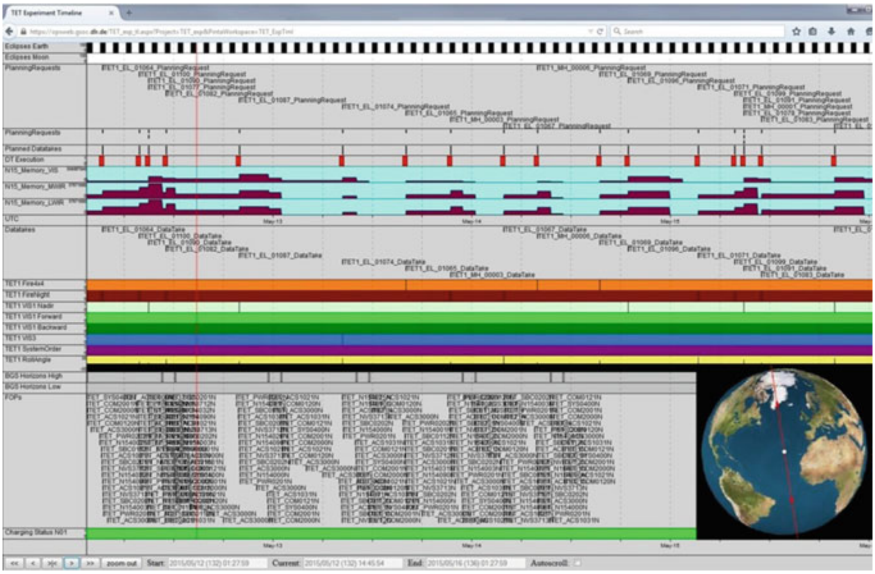


Fig. 4 Snapshot of TimOnWeb, showing an excerpt of the TET Timeline

Since TimOnWeb uses JSON files to transfer the data to be displayed from the server to the client, the tool can also be used to display information from other sources and is therefore not exclusively restricted to be used in combination with PINTA.

In addition, the TimOnWeb client uses state-of-the-art web technologies such as HTML5 and WebGL and relies on open source JavaScript libraries, which enables the application to make use of wide-spread and verified generic functionalities such as jQuery, jQueryUI, Moment and satellite-js. The latter is used, for example, to display ground station elevation diagrams and ground track plots, as shown in the lower right corner of Fig. 4.

A more detailed description with further information on the technical implementation of TimOnWeb can be found in [6].

3 Mission Planning Systems

The main task of PINTA is to serve as a generic basis for many semi-automatic mission planning systems, for past, current and future spacecraft missions, that are operated at GSOC. Figure 5 shows how a typical planning system is structured, how the individual components relate to one another, how everything is embedded within the mission operations ground segment, and which internal as well as external interfaces are commonly used.

PINTA/PLATO in combination with TimOnWeb is used as a semi-automatic and therefore also interactive mission planning framework. Therefore, the daily tasks

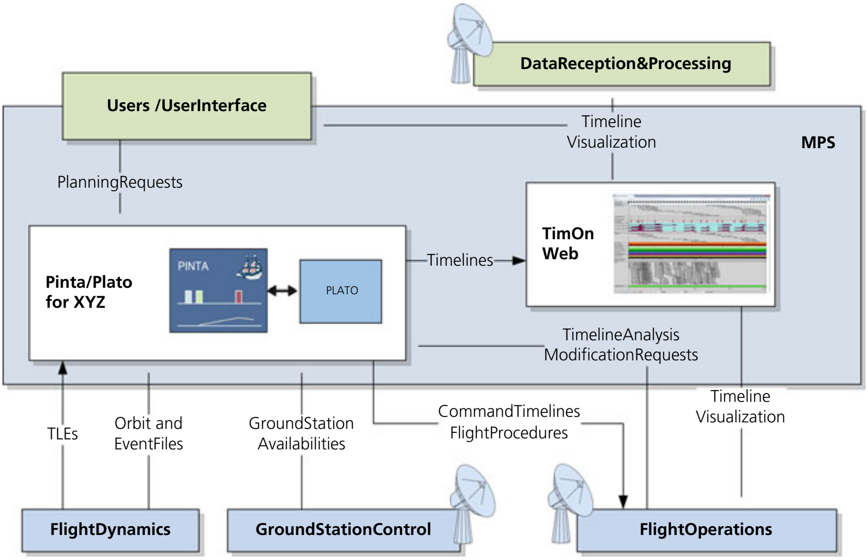
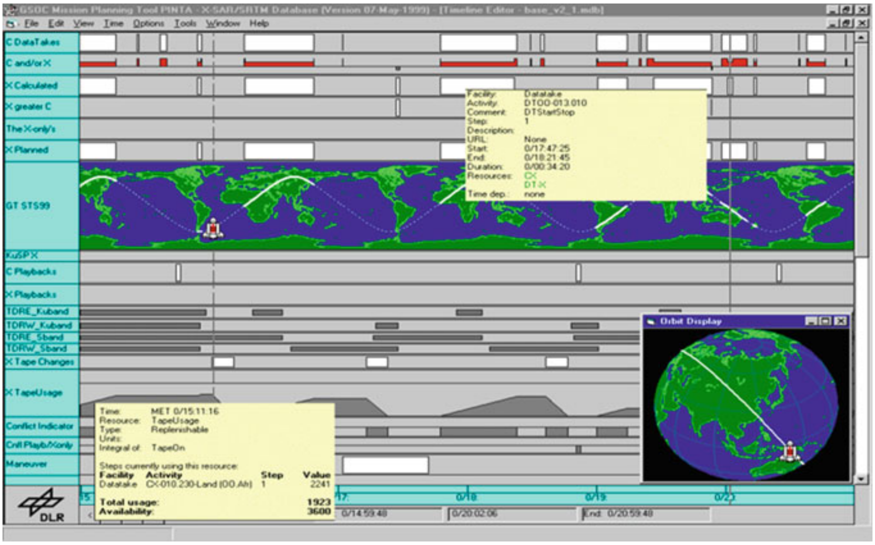


Fig. 5 Generic setup of a mission planning system with PINTA/PLATO and TimOnWeb

of an operators engineer typically are: performing and monitoring the planning runs manually as specified in the according ground operations procedure (GOP), execution of recommendations for off-nominal operations, performing additional procedures for special planning tasks for mission specific payloads, and to serve as a user help-desk for inquiries from the customers.

3.1 GRACE

PINTA is currently in its third version. The predecessor of the current generation was PINTA 2, which was used among others during the GRACE mission. This Gravity Recovery and Climate Experiment was a combined mission of DLR and NASA. Two identical satellites performed detailed measurements of earth's gravity field. The launch was in March 2002 and the end of the science mission was in October 2017. The Main planning Objectives in this mission were the Mass Memory Unit (MMU) Dump to ensure a uninterrupted science data collection, as well as routine activities like sun/moon blinding avoidance and the planning of maintenance and manoeuvre tasks (Fig. 6).



Facility: Datatake
Activity: DTOO-013.010
Comment: DTStartStop
Step: 1
Description:
URL: None
Start: 0/17:47:25
End: 0/18:21:45
Duration: 0/00:34:20
Resources: CX
DT-X
Time dep.: none

Time: MET 0/15:11:16
Resource: TapeUsage
Type: Replenishable
Units:
Integral of: TapeOn

Steps currently using this resource:

Facility	Activity	Step	Value
Datatake	CX-010.230-Land (DO.Ah)	1	2241

Total usage: **1923**
Availability: **3600**

Fig. 6 Pinta 2 GUI, used during the GRACE mission

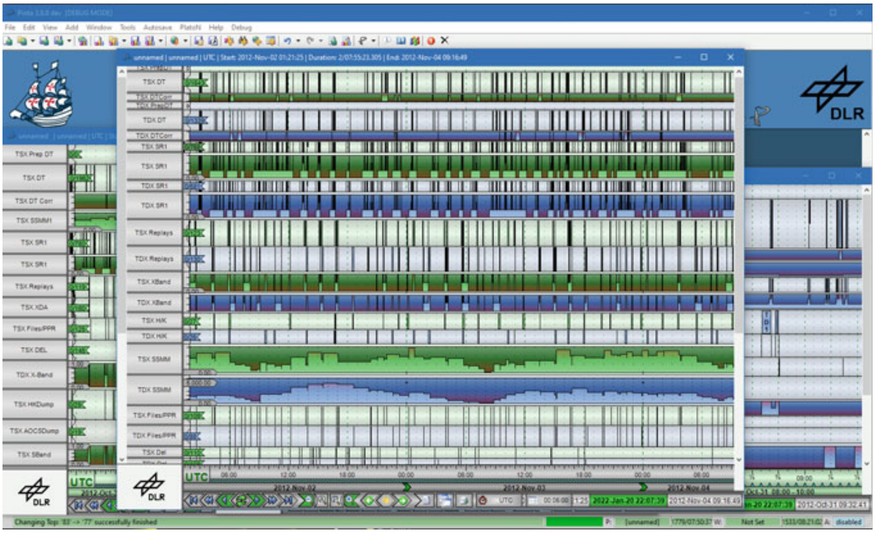
3.2 TerraSAR-X/TanDEM-X

In 2007 the satellite TerraSAR-X (TSX-1) has been launched for the TerraSAR-X mission and in 2010 the mission was completed by the launch of the twin satellite TanDEM-X. When operating in close formation, with distances down to 120 m, these radio satellites are capable of creating stereo-scopic SAR-images which can be used to create a digital elevation model (Fig. 7).

The corresponding planning problem is one of the most complex, we ever had to deal with, at GSOC. It involves data handling, dump scheduling, instrument sleep-level switching, antenna mode changing, attitude mode switching, etc. All together more than 300 different kind of constraints and over 100 types of resources (>1000 in total) need to be checked for over 10,000 activities per day, in order to create a conflict free timeline [4].

One notable difference from all those missions described in this paper is that, in TSTD, PINTA serves as a verification tool only and is not the central planning entity, as the TerraSAR-X/TanDEM-X mission planning system is running unattended as ‘Blackbox’.

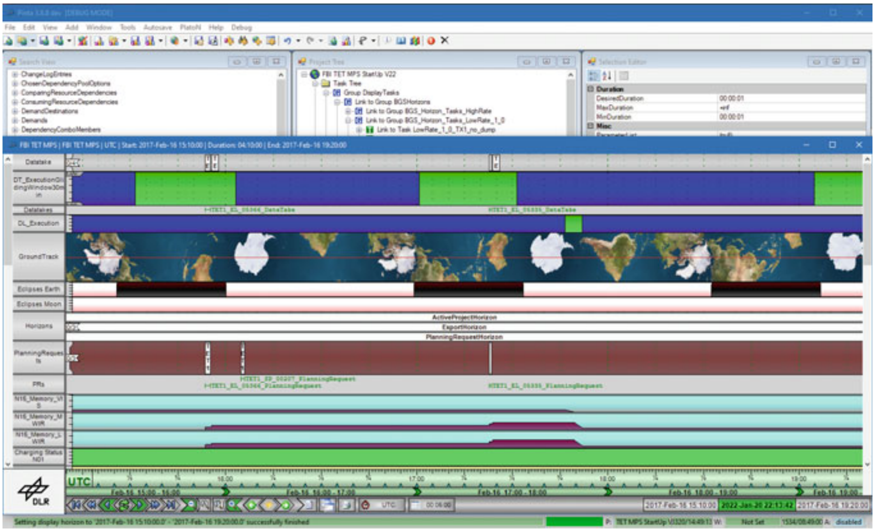
For the verification of the TSTD mission planning system, the planning products, and in particular the execution timeline with the sequence of Flight Operation procedures (FOPs) and the Saved Stack Files (SSF), is loaded in to PINTA to verify the proper export, and therefore commanding of the space crafts [7].



unnamed | unnamed | UTC | Start: 2012-Nov-02 01:21:25 | Duration: 2/07:55:23.305 | End: 2012-Nov-04 09:16:49

Changing Top: '83' -> '77' successfully finished

Fig. 7 PINTA timeline of the TerraSAR-X/TanDEM-X mission


FBI TET MPS | FBI TET MPS | UTC | Start: 2017-Feb-16 15:10:00 | Duration: 04:10:00 | End: 2017-Feb-16 19:20:00

Duration	
DesiredDuration	00:00:01
MaxDuration	+inf
MinDuration	00:00:01

Setting display horizon to '2017-Feb-16 15:10:00.0' - '2017-Feb-16 19:20:00.0' successfully finished

Fig. 8 PINTA timeline of the FireBird mission

3.3 FireBird

The FireBird mission had the goal to deliver science data about worldwide high temperature events to a global science community. It consisted of two infrared, earth observation spacecrafts TET and BIROS. The first one was launched in July 2012 and the second one in June 2016 (Fig. 8).

The main objective of the mission planning system was the generation of consistent and conflict-free timelines in order to command the payload and background sequence operations. Furthermore, the pre-planning and ordering process for acquisitions of the Infrared Camera System had been supported [5, 8].

3.4 GRACE-FO

The Gravity Recovery and Climate Experiment Follow-On (GRACE-FO) is the successor of the GRACE mission with near-identical hardware, that was launched in May 2018 (Fig. 9).

Likewise, this mission also consists of two satellites and the main mission planning objective is the prediction of the MMU fill status, the scheduling of the ground stations contacts as well as the handling of the file dumps, for non-science and science data. The planning of both satellites within one tool is important to coordinate resources, like the usage of the same ground stations.

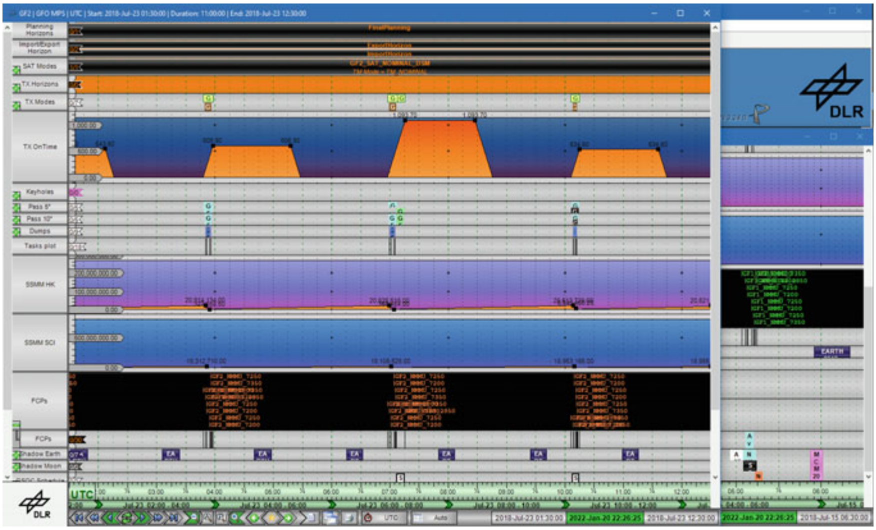


Fig. 9 PINTA timeline of the GRACE-FO mission

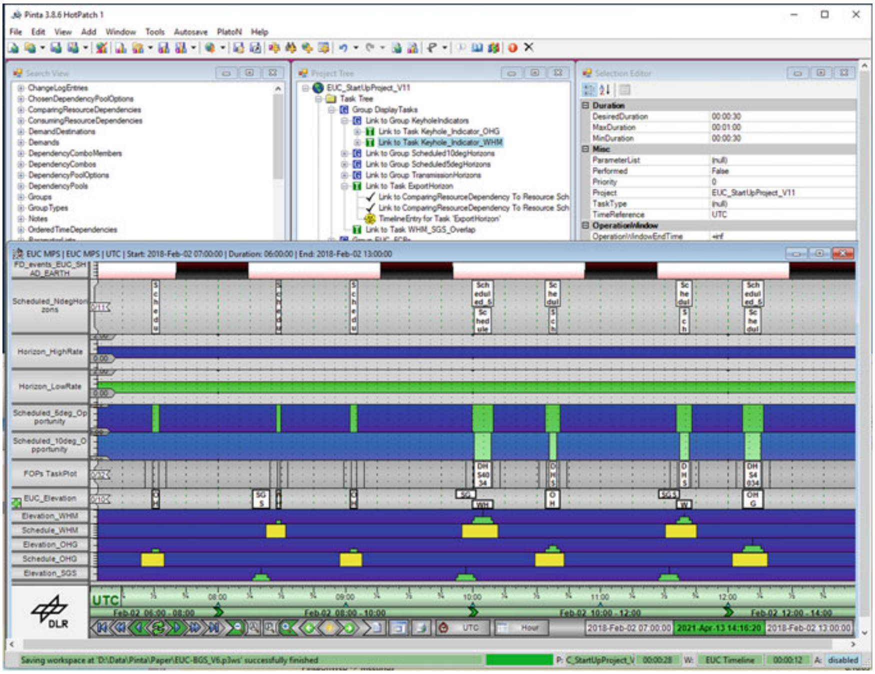
3.5 EuCROPIS

The mission EuCROPIS (Euglena and Combined Regenerative Organic-food Production in Space) was launched in December 2018. The science mission expired at the end of 2019.

On board the DLR mission were the experiments RAMIS (RAdiation Measurement In Space), the NASA-Experiment Power Cell in Space and an on-board-computer developed by DLR. The mission planning system (see Fig. 10) was used to support LEOP-, commissioning- and routine phase with a scheduling of ground station contacts as well as the handling of file dumps for science and non-science data.

3.6 PIXL-1 (Formerly Known as CubeL)

PIXL-1 is the recently launched and first CubeSat where PINTA is used for displaying the timeline and generating the background sequence. The satellite started its mission in January this year and carried a 10 × 10 cm Laser Communication Terminal OSIRIS4CubeSat into the orbit. PIXL-1 is operated by GSOC. The mission planning objectives are the scheduling of ground station contacts, the creation of housekeeping data, GPS data collection and the handling of file dumps. In PINTA all data needs to be exported and exchanged in a project specific format. For PIXL-1 a customized version of the generic PINTA exporter had to be implemented (see Fig. 11).

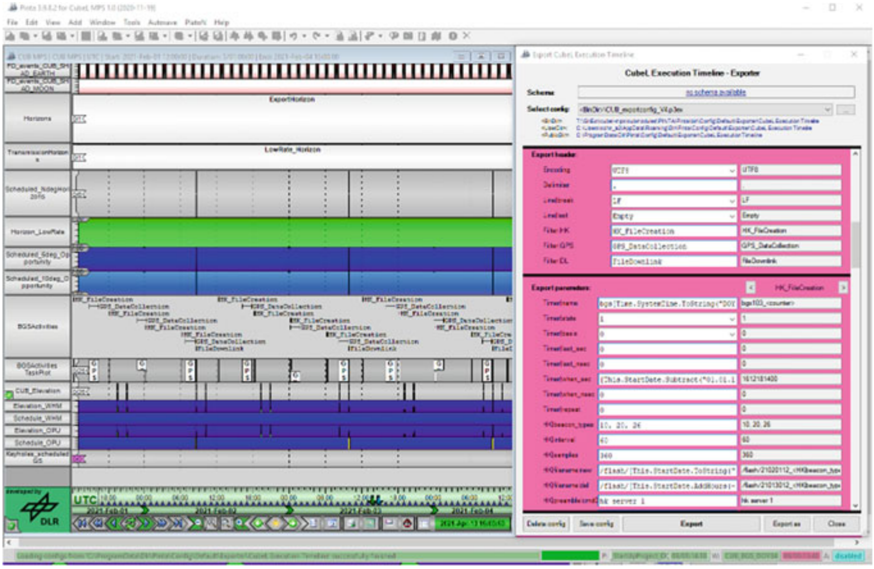

Pinta 3.8.6 HotPatch 1
File Edit View Add Window Tools Autosave PlatoN Help
Search View
ChangeLogEntries
ChosenDependencyPoolOptions
ComparingResourceDependencies
ConsumingResourceDependencies
DemandDestinations
Demands
DependencyComboMembers
DependencyCombos
DependencyPoolOptions
DependencyPools
Groups
GroupTypes
Notes
OrderedTimeDependencies
Project Tree
EUC_StartUpProject_V11
Task Tree
Group DisplayTasks
Link to Group KeyholeIndicators
Link to Task Keyhole_Indicator_OHG
Link to Task Keyhole_Indicator_WHM
Link to Group Scheduled10degHorizons
Link to Group Scheduled5degHorizons
Link to Group TransmissionHorizons
Link to Task ExportHorizon
Link to ComparingResourceDependency To Resource Sch
Link to ComparingResourceDependency To Resource Sch
TimelineEntry for Task 'ExportHorizon'
Link to Task WHM_SGS_Overlap
Selection Editor
EUC MPS | EUC MPS | UTC | Start: 2018-Feb-02 07:00:00 | Duration: 06:00:00 | End: 2018-Feb-02 13:00:00
FD_events_EUC_SHAD_EARTH
Scheduled_NdegHorizons
Horizon_HighRate
Horizon_LowRate
Scheduled_5deg_Opportunity
Scheduled_10deg_Opportunity
FOPs TaskPlot
EUC_Elevation
Elevation_WHM
Schedule_WHM
Elevation_OHG
Schedule_OHG
Elevation_SGS
UTC
08:00
09:00
10:00
11:00
12:00
Feb-02 06:00 - 08:00
Feb-02 08:00 - 10:00
Feb-02 10:00 - 12:00
Feb-02 12:00 - 14:00
DLR
UTC
Hour
2018-Feb-02 07:00:00
2021-Apr-13 14:16:20
2018-Feb-02 13:00:00

Duration	
DesiredDuration	00:00:30
MaxDuration	00:01:00
MinDuration	00:00:30
Misc	
ParameterList	(null)
Performed	False
Priority	0
Project	EUC_StartUpProject_V11
TaskType	(null)
TimeReference	UTC
OperationWindow	
OperationWindowEndTime	+inf

Saving workspace at 'D:\Data\Pinta\Paper\EUC-BGS_V6.p3ws' successfully finished

P: C_StartUpProject_V 00:00:28 W: EUC Timeline 00:00:12 A: disabled

Fig. 10 PINTA timeline of the EuCROPIS mission



CubeL Execution Timeline - Exporter

Schema	no schema available	
Select config:	<BinDir>\CUB_exportconfig_V4.p3ew	

Export header

Encoding	UTF8	UTF8
Delimiter	,	,
Linebreak	LF	LF
Lineend	Empty	Empty
Filter HK	HK_FileCreation	HK_FileCreation
Filter GPS	GPS_DataCollection	GPS_DataCollection
Filter DL	FileDownlink	FileDownlink

Export parameters: HK_FileCreation

Timename	bgs[Time.SystemTime.ToString("DOY	bgs103_<counter>
Timedate	1	1
Timebase	0	0
Timelast_sec	0	0
Timelast_nsec	0	0
Timewhen_sec	{This.StartDate.Subtract("01.01.1	1612181400
Timewhen_nsec	0	0
Timerepeat	0	0
#Qbeacon_types	10, 20, 26	10, 20, 26
#Qinterval	60	60
#Qsamples	360	360
#Qfilename new	/flash/{This.StartDate.ToString("	/flash/21020112_<HKbeacon_typ
#Qfilename old	/flash/{This.StartDate.AddHours(-	/flash/21013012_<HKbeacon_typ
#Qpreamble cmd	hk server 1	hk server 1

Fig. 11 PINTA timeline and execution timeline exporter of the PIXL-1/CubeL mission

3.7 V3C

The objective of the V3C project is the provision and maintenance of a highly mobile compact control centre. This includes satellite command and control as part of the German national capability for "Responsive Space". The goal was to engineer and deliver a compact Mission Operations System, runnable on commodity mobile hardware, enabling fully automated workflow-driven operations of alike user missions.

The main purpose of the Mission Planning System for V3C is the generation of consistent, conflict-free timelines and sequences of flight operations procedures in order to command the payload and background sequence operations of the target spacecraft (for demonstration purposes: BIROS) from all given input items and known constraints. In addition, MPS shall support the V3C operator in the pre-planning and ordering process for acquisitions of the spacecraft imaging payload.

The V3C project comprises three phases: two demonstrations, in which the spacecraft BIROS is used to demonstrate the V3C concept, and afterwards the setup of all V3C components for a new series of to be developed spacecraft. The main MPS task for the demonstrations is to adapt the already existing components used for BIROS in the scope of the FireBird mission to the needs of V3C. The main challenges here are the lack of a connection to the internet, as available during a regular scientific mission like FireBird, and the usage of another payload for V3C than the one used during the FireBird mission [9].

4 Sequence of Events Editor

During the last LEOP's at GSOC, several desired usability improvements for the SoE (Sequence of Events) generation were detected and a lot of new requirements were defined accordingly. It was therefore obvious to consider the possibility of using PINTA for the SoE generation, as well as TimOnWeb for the SoE visualization.

4.1 Overview

In order to meet the special operations requirements during the pre-launch simulation phase, the LEOP phase and the commissioning phase of new satellite missions, many things had to be improved in order to transform PINTA into a usable SoEEditor. One example was the need to extend the PINTA modelling language to handle one main disadvantage with the "OrderedTimeDependencies". Therefore, a template/instance mechanism was introduced for "Tasks" and "Groups" (see [10] for details). Also, an algorithm was implemented to allow the user to reschedule all activities assigned to a conflicting time dependency by only one click or shortcut.

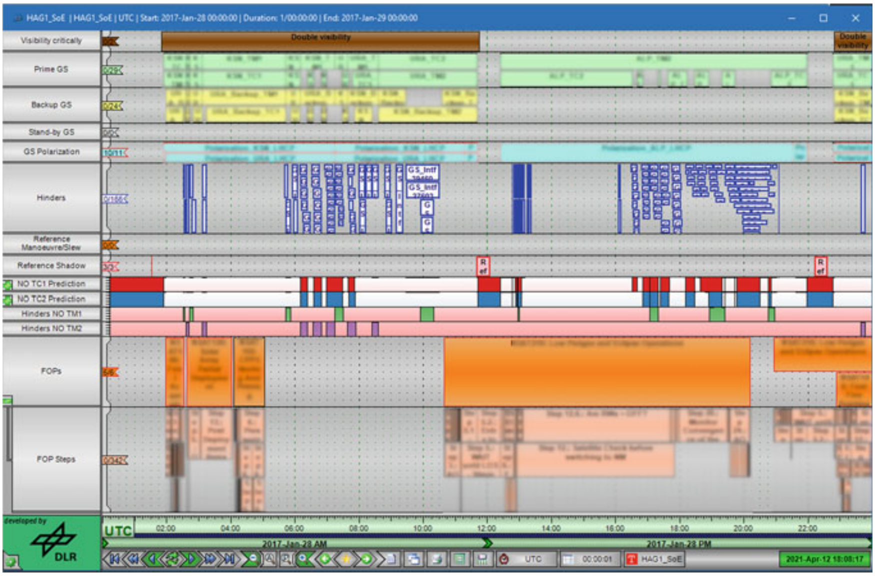


Fig. 12 Sequence of Events for the LEOP of the GEO mission called HAG. Displayed are the first 24 h after the separation of the satellite from the rocket (blurred due to data protection reasons)

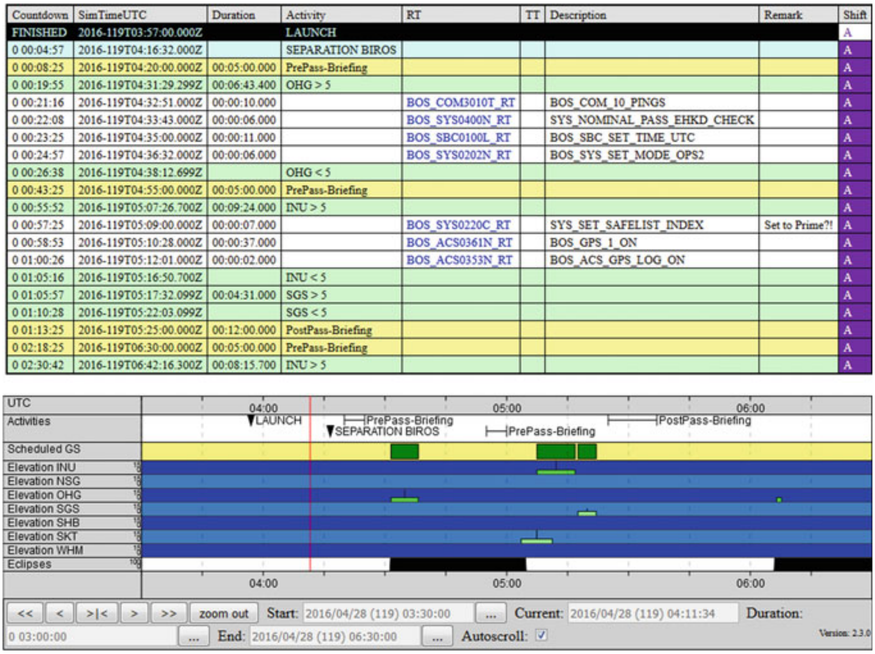
A wide variety of importers had to be developed or modified in order to incorporate FOPs (Flight Operation Procedures), GOPs (Ground Operation Procedures), orbit related events, e.g. ground station visibilities, shadow transitions, or even mission specific events like interfering antenna frequencies with other satellite missions.

In order to reduce the workload for the operator to a minimum, while changing the SoE, it was also necessary to provide an easy-to-use graphical interface. This includes adding FOPs/GOPs and mission-specific events via drag and drop, moving and deleting non-orbit-related events, as well as adjusting time dependencies between different activities.

To share the results of the SoE with LEOP related tools and the operation stuff, several exporters were introduced, reused and extended for e.g. requesting necessary ground station passes or sharing the SoE as PDF files in a graphical and tabular form. An example of the SoEEditor is shown in Fig. 12.

4.2 Displaying

For many missions it is also important that all changes to the SoE are immediately visible in the control room, as well as in the support rooms. For this reason, the interface to TimOnWeb has also been further developed in order to make changes to the timeline immediately visible to all operators (see Fig. 13).



Countdown	SimTimeUTC	Duration	Activity	RT	TT	Description	Remark	Shift
FINISHED	2016-119T03:57:00.000Z		LAUNCH					A
0 00:04:57	2016-119T04:16:32.000Z		SEPARATION BIROS					A
0 00:08:25	2016-119T04:20:00.000Z	00:05:00.000	PrePass-Briefing					A
0 00:19:55	2016-119T04:31:29.299Z	00:06:43.400	OHG > 5					A
0 00:21:16	2016-119T04:32:51.000Z	00:00:10.000		BOS_COM3010T_RT		BOS_COM_10_PINGS		A
0 00:22:08	2016-119T04:33:43.000Z	00:00:06.000		BOS_SYS0400N_RT		SYS_NOMINAL_PASS_EHKD_CHECK		A
0 00:23:25	2016-119T04:35:00.000Z	00:00:11.000		BOS_SBC0100L_RT		BOS_SBC_SET_TIME_UTC		A
0 00:24:57	2016-119T04:36:32.000Z	00:00:06.000		BOS_SYS0202N_RT		BOS_SYS_SET_MODE_OPS2		A
0 00:26:38	2016-119T04:38:12.699Z		OHG < 5					A
0 00:43:25	2016-119T04:55:00.000Z	00:05:00.000	PrePass-Briefing					A
0 00:55:52	2016-119T05:07:26.700Z	00:09:24.000	INU > 5					A
0 00:57:25	2016-119T05:09:00.000Z	00:00:07.000		BOS_SYS0220C_RT		SYS_SET_SAFELIST_INDEX	Set to Prime?!	A
0 00:58:53	2016-119T05:10:28.000Z	00:00:37.000		BOS_ACS0361N_RT		BOS_GPS_1_ON		A
0 01:00:26	2016-119T05:12:01.000Z	00:00:02.000		BOS_ACS0353N_RT		BOS_ACS_GPS_LOG_ON		A
0 01:05:16	2016-119T05:16:50.700Z		INU < 5					A
0 01:05:57	2016-119T05:17:32.099Z	00:04:31.000	SGS > 5					A
0 01:10:28	2016-119T05:22:03.099Z		SGS < 5					A
0 01:13:25	2016-119T05:25:00.000Z	00:12:00.000	PostPass-Briefing					A
0 02:18:25	2016-119T06:30:00.000Z	00:05:00.000	PrePass-Briefing					A
0 02:30:42	2016-119T06:42:16.300Z	00:08:15.700	INU > 5					A

Start: 2016/04/28 (119) 03:30:00
Current: 2016/04/28 (119) 04:11:34
Duration: 0 03:00:00
End: 2016/04/28 (119) 06:30:00
Autoscroll: ☑
Version: 2.3.0

Fig. 13 Sequence of Events for the LEOP of a LEO spacecraft called BIROS. The upper part shows an alphanumeric table with all the past, currently active and upcoming events, during the first ground station contacts after the separation of the spacecraft. The lower part is a graphical presentation of the events on the mission timeline

A more detailed overview of the interaction between Pinta and TimOnWeb with regard to the SoE generation during LEOPs is given to the interested reader in [6].

5 On-Call Tool

PINTA's original purpose was the planning of satellite mission timelines. However, the generation of those timelines is not the only planning problem encountered in regular satellite operation. One such problem is the on-call shift scheduling for multiple subsystems of one or more satellites. The difficulty here lies in the various dependencies and constraints of people and subsystems like person outages, role-conflicts and so on. The on-call tool is used to generate conflict-free role-based shift plans.

5.1 Overview

The on-call plan at the GSOC is usually automatically planned for one year in advance but with the option to re-plan every quarter. For this purpose, PINTA with a specialized PLATO algorithm is used. This algorithm is configured to create a plan where every subsystem role of every satellite which requires on-call support has one person planed at all times. It also has to keep the constraints such as outages of persons, role-conflicts, legal and company requirements in mind and not violate those. There are also optional goals for the algorithm to improve the plan for the staff members. One for example is that the on-call shifts should last a whole week to prevent daily changes of the person who is on-call. Such optional goals are nice to have but not always achievable.

After the automatic planning run, the on-call plan can be displayed, analyzed and manually modified. Such manual modifications of an existing plan are easily done with PINTA using the TimelineView (example plan see Fig. 14). The plan will then be published and displayed on TimOnWeb so that the on-call staff gets the information and the operator knows who to call in case of a contingency.

The operational plan can also be modified by the on-call-coordinator using the timeline in PINTA at any time. These modifications are usually requested by the on-call staff themselves. A change may be necessary if a person on shift becomes unavailable. This ensures that the plan is always up to date.

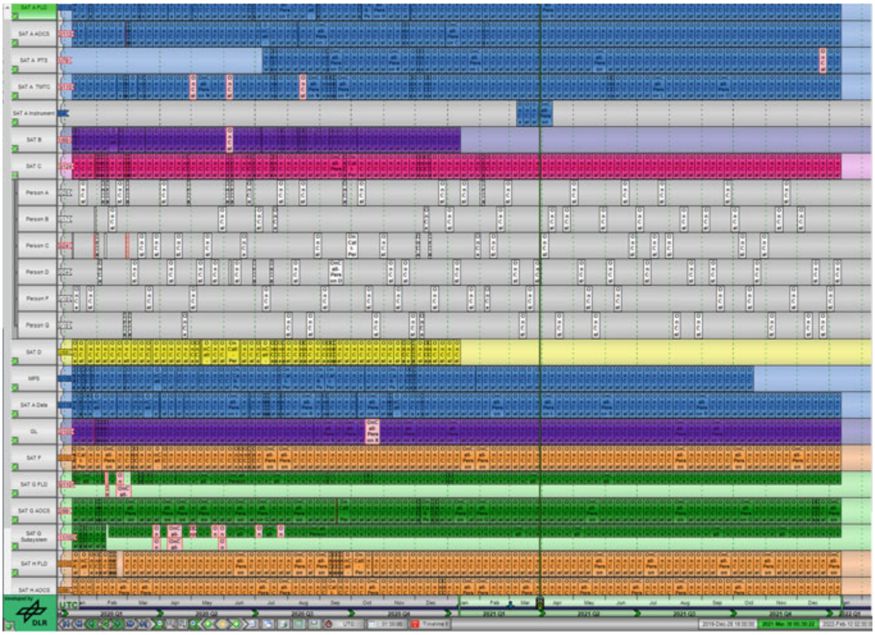


Fig. 14 Example of an On-Call plan over the period of two years with several different satellites and subsystems (renamed due to data protection reasons)

5.2 *Plugins*

There are two special plugins written for the On-Call Tool. The first one help prepare the on-call project for planning. It provides the user with a GUI to add, delete or change staff members and stations. In this context, station stands for a subsystem of a satellite that requires on-call service. This on-call plugin relieves the on-call coordinator by adding, removing and modifying of resources, tasks and dependencies required for the planning and maintaining of the plan. The second plugin is embedded in the Task-Editor GUI and is opened by simply double-clicking of any on-call task within the Timeline View of PINTA. It can be used to easily (de-)select different subsystems of an on-call task of a person.

6 MuMiCoRoS

With an increasing number of satellite missions, a point will eventually be reached where it is no longer possible to treat all missions separately. A new tool was needed to coordinate the ground station planning of the various missions at GSOC, to avoid resource conflicts, such as antenna utilization and operator availability.

6.1 *Overview*

The "**Multi-Mission-Control-Room-and-Pass-Scheduler**" (MuMiCoRoS), based on PINTA, was developed for the weekly ground station planning of all LEO missions at GSOC: TerraSar-X, TanDEM-X, TET, Biros, Grace-FollowOn 1 & 2 and Eu:CROPIS.

It quickly became clear that PINTA was the right software for this task, since it was already possible out-of-the-box to import all required files, such as flight dynamics events, or antenna availability, as these are already part of every mission planning system. The possibility of exporting request files, which in turn is required for ground station booking, was also already available, because this was a necessary requirement for the SoEEditor. In addition, the GUI is flexible enough to be able to be adapted to this planning problem (see Fig. 15).

6.2 *Plugins*

Although PINTA already provided a wide range of functions that were necessary for MuMiCoRoS, an additional plugin was still required to add further functionality. The so-called "**GS-Scheduling**" plugin includes a planning algorithm that is able to

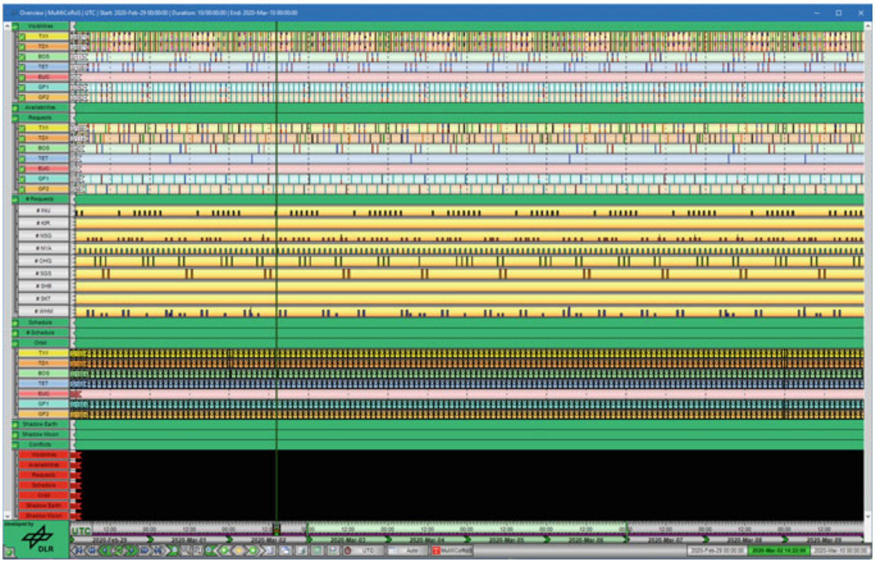
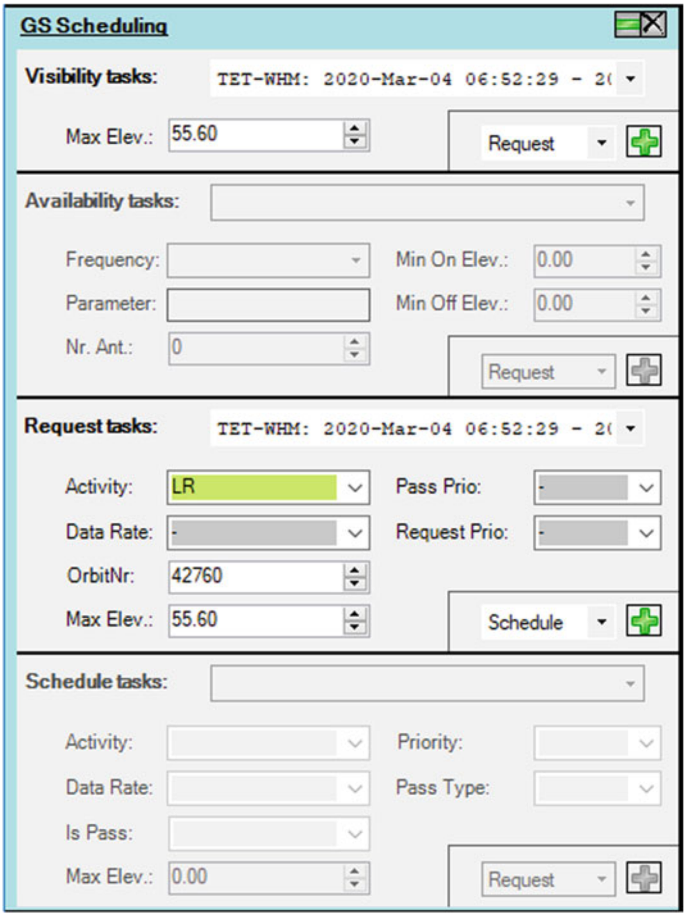


Fig. 15 The main timeline view of MuMiCoRoS. It shows all visibilities for all satellites and ground stations, the availabilities of the ground stations and the currently requested contacts. Resource conflicts, which are very rare and not existent in this plan, are indicated in the bottom plots

generate a conflict-free plan for all missions in a very short amount of time. Each mission has an individual planning strategy, which often changes over time due to changed mission objectives. In addition, it must be guaranteed that the missions do not get in each other's way.

The automatic planning does its job very well and rarely requires manual intervention. Yet, due to unforeseeable events, like broken antennas, to name just the most obvious one, it is still necessary to have the possibility to perform the planning process manually. For this reason, during the development of MuMiCoRoS, great value was placed on the fact that a user can change plans very comfortably and have an overview of all possible side effects. The GS-Scheduling plugins embeds a ground station scheduling specific panel into the Task-Editor's GUI (see Fig. 16). By simply double-clicking on any task within the Timeline View of PINTA, this panel appears and can be used to unscheduled, schedule or edit a property of the requested contacts. Another tool that makes planning easier for the user is the GS Scheduling Editor (see Fig. 17), which can be used for manual scheduling or for getting another look at the plan in tabular form, with the possibility to filter subsets of the whole plan.



GS Scheduling

Visibility tasks: TET-WHM: 2020-Mar-04 06:52:29 - 2(

Max Elev.: 55.60

Availability tasks:

Frequency:

Parameter:

Nr. Ant.: 0

Min On Elev.: 0.00

Min Off Elev.: 0.00

Request tasks: TET-WHM: 2020-Mar-04 06:52:29 - 2(

Activity: LR

Data Rate: -

OrbitNr: 42760

Max Elev.: 55.60

Pass Prio: -

Request Prio: -

Schedule tasks:

Activity:

Data Rate:

Is Pass:

Max Elev.: 0.00

Priority:

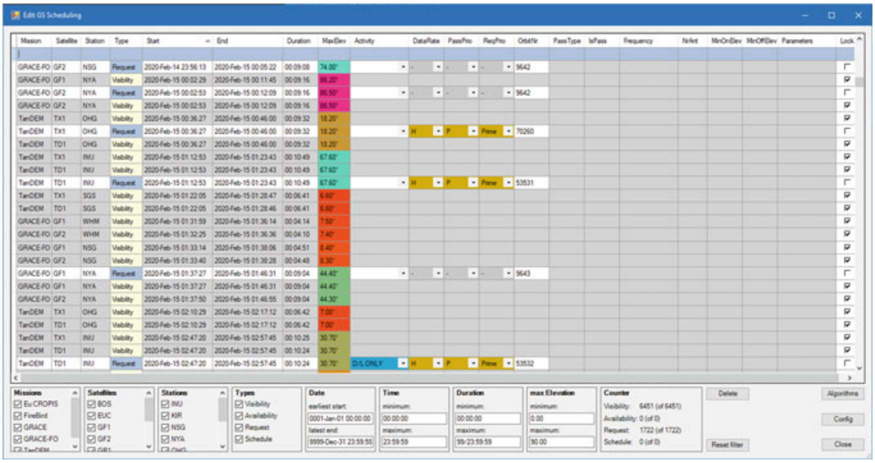
Pass Type:

Fig. 16 GS scheduling plugin embedded within the task editor

7 Galileo Planning Tool

Galileo is one of the four Global Navigation Satellite Systems (GNSS) worldwide, which went live in 2016. The (not yet) fully established constellation and infrastructure of Galileo will consist of 30 satellites distributed over three orbital planes, and about 20 sensor stations spread across the globe for monitoring the navigation signals. Furthermore, two Galileo Control Centers (GCC-D and GCC-I), one located in Germany (Oberpfaffenhofen) and one located in Italy (Fucino), the GNSS service center in Spain (Torrejón de Ardoz) and several mission uplink stations and S-Band Stations, are required for satellite monitoring and control.

The Galileo program is under the responsibility of the European Commission, which has delegated responsibility for the provision of the Galileo service to the


Edit GS Scheduling

Mission	Satellite	Station	Type	Start	End	Duration	MaxElev	Activity	DataRate	PassPrio	ReqPrio	OrbitNr	PassType	IsPass	Frequency	NrAnt	MinOnElev	MinOffElev	Parameters	Lock
GRACE-FO	GF2	NSG	Request	2020-Feb-14 23:56:13	2020-Feb-15 00:05:22	00:09:08	74.00°		-	-	-	9642								☐
GRACE-FO	GF1	NYA	Visibility	2020-Feb-15 00:02:29	2020-Feb-15 00:11:45	00:09:16	86.20°													☑
GRACE-FO	GF2	NYA	Request	2020-Feb-15 00:02:53	2020-Feb-15 00:12:09	00:09:16	86.50°		-	-	-	9642								☐
GRACE-FO	GF2	NYA	Visibility	2020-Feb-15 00:02:53	2020-Feb-15 00:12:09	00:09:16	86.50°													☑
TanDEM	TX1	OHG	Visibility	2020-Feb-15 00:36:27	2020-Feb-15 00:46:00	00:09:32	18.20°													☑
TanDEM	TX1	OHG	Request	2020-Feb-15 00:36:27	2020-Feb-15 00:46:00	00:09:32	18.20°		H	P	Prime	70260								☐
TanDEM	TD1	OHG	Visibility	2020-Feb-15 00:36:27	2020-Feb-15 00:46:00	00:09:32	18.20°													☑
TanDEM	TX1	INU	Visibility	2020-Feb-15 01:12:53	2020-Feb-15 01:23:43	00:10:49	67.60°													☑
TanDEM	TD1	INU	Visibility	2020-Feb-15 01:12:53	2020-Feb-15 01:23:43	00:10:49	67.60°													☑
TanDEM	TD1	INU	Request	2020-Feb-15 01:12:53	2020-Feb-15 01:23:43	00:10:49	67.60°		H	P	Prime	53531								☐
TanDEM	TX1	SGS	Visibility	2020-Feb-15 01:22:05	2020-Feb-15 01:28:47	00:06:41	6.60°													☑
TanDEM	TD1	SGS	Visibility	2020-Feb-15 01:22:05	2020-Feb-15 01:28:46	00:06:41	6.60°													☑
GRACE-FO	GF1	WHM	Visibility	2020-Feb-15 01:31:59	2020-Feb-15 01:36:14	00:04:14	7.50°													☑
GRACE-FO	GF2	WHM	Visibility	2020-Feb-15 01:32:25	2020-Feb-15 01:36:36	00:04:10	7.40°													☑
GRACE-FO	GF1	NSG	Visibility	2020-Feb-15 01:33:14	2020-Feb-15 01:38:06	00:04:51	8.40°													☑
GRACE-FO	GF2	NSG	Visibility	2020-Feb-15 01:33:40	2020-Feb-15 01:38:28	00:04:48	8.30°													☑
GRACE-FO	GF1	NYA	Request	2020-Feb-15 01:37:27	2020-Feb-15 01:46:31	00:09:04	44.40°		-	-	-	9643								☐
GRACE-FO	GF1	NYA	Visibility	2020-Feb-15 01:37:27	2020-Feb-15 01:46:31	00:09:04	44.40°													☑
GRACE-FO	GF2	NYA	Visibility	2020-Feb-15 01:37:50	2020-Feb-15 01:46:55	00:09:04	44.30°													☑
TanDEM	TX1	OHG	Visibility	2020-Feb-15 02:10:29	2020-Feb-15 02:17:12	00:06:42	7.00°													☑
TanDEM	TD1	OHG	Visibility	2020-Feb-15 02:10:29	2020-Feb-15 02:17:12	00:06:42	7.00°													☑
TanDEM	TX1	INU	Visibility	2020-Feb-15 02:47:20	2020-Feb-15 02:57:45	00:10:25	30.70°													☑
TanDEM	TD1	INU	Visibility	2020-Feb-15 02:47:20	2020-Feb-15 02:57:45	00:10:24	30.70°													☑
TanDEM	TD1	INU	Request	2020-Feb-15 02:47:20	2020-Feb-15 02:57:45	00:10:24	30.70°	D/L ONLY	H	P	Prime	53532								☐

Missions
☑ Eu:CROPIS
☑ FireBird
☑ GRACE
☑ GRACE-FO
☐ TanDEM

Satellites
☑ BOS
☑ EUC
☑ GF1
☑ GF2
☐ GR1

Stations
☑ INU
☑ KIR
☑ NSG
☑ NYA
☐ OHG

Types
☑ Visibility
☑ Availability
☑ Request
☑ Schedule

Date
earliest start: 0001-Jan-01 00:00:00
latest end: 9999-Dec-31 23:59:59

Time
minimum: 00:00:00
maximum: 23:59:59

Duration
minimum: 00:00:00
maximum: 99/23:59:59

max Elevation
minimum: 0.00
maximum: 90.00

Counter
Visibility: 6451 (of 6451)
Availability: 0 (of 0)
Request: 1722 (of 1722)
Schedule: 0 (of 0)

Fig. 17 GS scheduling editor plugin for manual scheduling, triggering an automatic scheduling run, or simply for having a better clarity over the whole timeline in tabular form with the possibility to apply filters

GNSS Service Agency (GSA). The service provision comprises various contracts, including the Galileo Service Operations (GSOp) contract, which was awarded to Spaceopal GmbH in 2016. The Munich based, Spaceopal GmbH is a joint venture founded in 2009 by the two partners DLR Gesellschaft für Raumfahrtanwendungen (GfR) mbH and Telespazio S.p.A., which itself is a joint venture owned by Leonardo and Thales Group. As a service operator of the Galileo system, Spaceopal GmbH offers users worldwide high-quality navigation and time measurement. Its planning team works with multiple subcontractors to ensure service operations for the Galileo fleet of spacecrafts and the infrastructure. The primary goal herby is of course, to ensure that operation, maintenance and further development of the system is performed smoothly and without interruptions at any time, so that a continuous global navigation service can be provided. Users all around the world are already using Galileo for single and dual frequency positioning down to the m-level [11].

PINTA, among various competitors, was the tool of choice by the Spaceopal Planning Team which fits their needs the closest. It is used to support the Galileo Service Operation (GSOp) since 2018 and will also be referred to as the Galileo Planning Tool (GPT) in this context of this section. The following subsections will first give an overview of the associated planning problem and then describe technical implementation aspects in more detail. To get a more comprehensive overview of the Galileo planning process see [11, 12].

7.1 Overview

The planning process at Galileo includes three timelines that differ in granularity (see Fig. 18). These timelines are divided into four plans that are constantly updated within firmly defined moving timeframes:

- "**LTP**": A Long-Term Plan for the next 15 months that contains all the high-level activities with a rough allocation of time.
- "**MTP**": Two Mid-Term Plans (MTP-OPE and MTP-VAL) for the next 15 weeks, one for the operational chain (OPE) and one for the validation chain (VAL).
- "**STP**": A Short-Term Plan for the next ten days with the best granularity and therefore highest temporal accuracy down to minutes.

Activities that are scheduled during these timelines cover all Galileo subsystems: flight ops, control segment, mission segment, remote locations, service operations, hardware, software, hosting, network, etc.

For obvious reasons, these plans are interdependent and it must be ensured that all information details are shared correctly between all the different plans at all times. The most reliable way, to achieve this goal, is that these plans must be processed using the same tool, which also provides functionalities that can be used to link the plans together. Furthermore, displaying activities in a clear and easy to grasp way, are a basic requirement to reduce human error. This requires a hierarchical and logical arrangement of the activities, to provide a quick overview about impacts, relationships, delays and especially planning states like approvals, cancellations and the final execution of the activities. An example of the main Timelines within the GPT can be seen in Fig. 20 and one of the color-coding definitions is shown in Fig. 19. Similar ones exist for LTP and STP. A better insight into how much effort was required to bring all these plans together is explained in [11].

Although PINTA already provided a large number of generic functionalities due to its development history, especially as a mission planning system, it was still necessary to undertake further generic as well as specific extensions to support the Galileo service operations. These will be dealt within the following.

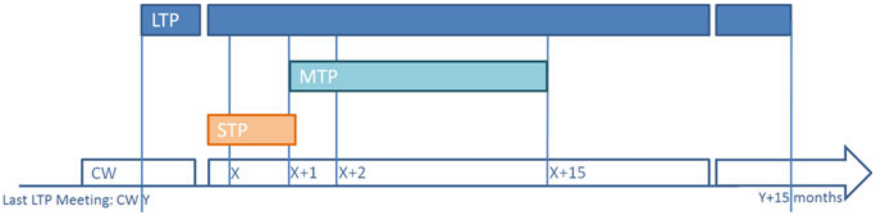
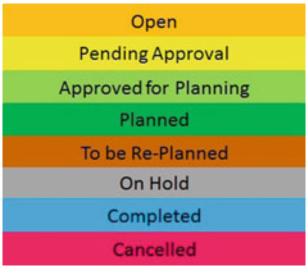


Fig. 18 Coverage of the LTP, MTP and STP timeframes, which is shifted every calendar week



Open
Pending Approval
Approved for Planning
Planned
To be Re-Planned
On Hold
Completed
Cancelled

Fig. 19 GPT states color coding schema for the MTP

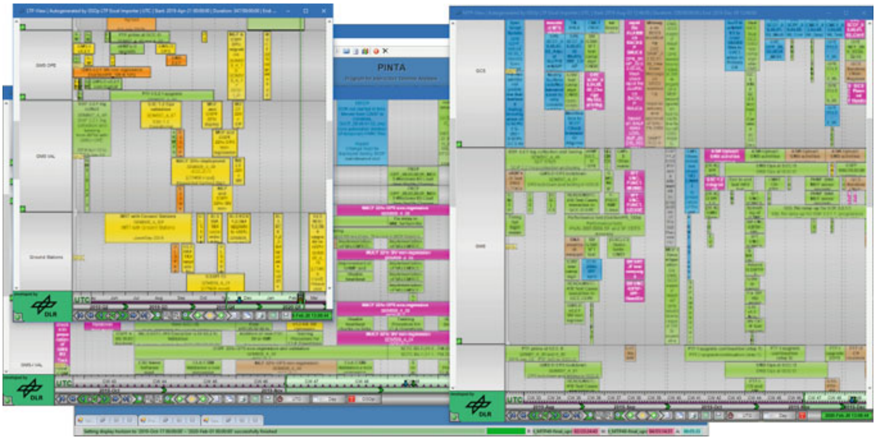
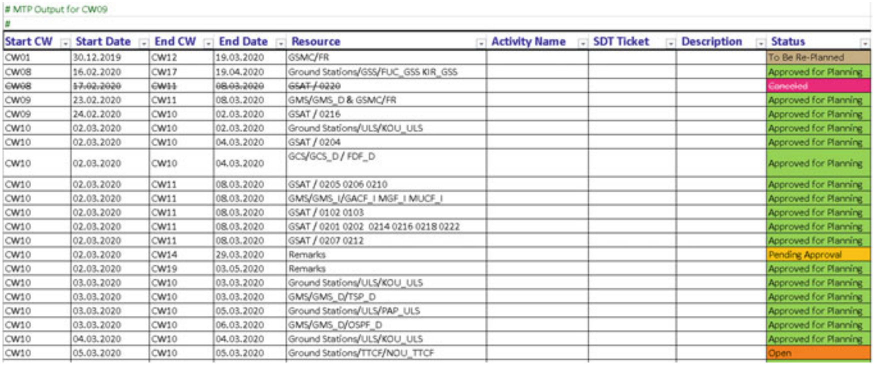


Fig. 20 Screenshot of the GPT showing the LTP (left), MTP-VAL (middle) and MTP-OPE (right) in different views and the main window in the background (task boxes were pixelated due to data protection reasons)

7.2 Generic Exporter

Due to the continuously growing number of projects PINTA has to handle and the hereby involved requirements to support more and more different export formats, the decision was made to completely redesign its export functionalities. This redesign has already been described in [6] together with the interaction between the exporter and the PINTA scripting language, which in turn provides the functionality that every single export value can easily be configured within the GUI. All possible values from the whole PINTA data model structure or even any arithmetic or logic combination of these values can be defined as export values via this scripting language.

In order to comply with the requirements for being a usable Galileo Planning Tool, the generic exporter was further improved. With the primary goal to generate graphical exports, with the same look and feel as the appearance of PINTA timeline views (see Fig. 20), the following file formats were implemented:



MTP Output for CW09
#

Start CW	Start Date	End CW	End Date	Resource	Activity Name	SDT Ticket	Description	Status
CW01	30.12.2019	CW12	19.03.2020	GSMC/FR				To Be Re-Planned
CW08	16.02.2020	CW17	19.04.2020	Ground Stations/GSS/FUC_GSS KIR_GSS				Approved for Planning
~~CW08~~	~~17.02.2020~~	~~CW11~~	~~08.03.2020~~	~~GSAT / 0220~~				Canceled
CW09	23.02.2020	CW11	08.03.2020	GMS/GMS_D & GSMC/FR				Approved for Planning
CW09	24.02.2020	CW10	02.03.2020	GSAT / 0216				Approved for Planning
CW10	02.03.2020	CW10	02.03.2020	Ground Stations/ULS/KOU_ULS				Approved for Planning
CW10	02.03.2020	CW10	04.03.2020	GSAT / 0204				Approved for Planning
CW10	02.03.2020	CW10	04.03.2020	GCS/GCS_D / FDF_D				Approved for Planning
CW10	02.03.2020	CW11	08.03.2020	GSAT / 0205 0206 0210				Approved for Planning
CW10	02.03.2020	CW11	08.03.2020	GMS/GMS_I/GACF_I MGF_I MUCF_I				Approved for Planning
CW10	02.03.2020	CW11	08.03.2020	GSAT / 0102 0103				Approved for Planning
CW10	02.03.2020	CW11	08.03.2020	GSAT / 0201 0202 0214 0216 0218 0222				Approved for Planning
CW10	02.03.2020	CW11	08.03.2020	GSAT / 0207 0212				Approved for Planning
CW10	02.03.2020	CW14	29.03.2020	Remarks				Pending Approval
CW10	02.03.2020	CW19	03.05.2020	Remarks				Approved for Planning
CW10	03.03.2020	CW10	03.03.2020	Ground Stations/ULS/KOU_ULS				Approved for Planning
CW10	03.03.2020	CW10	03.03.2020	GMS/GMS_D/TSP_D				Approved for Planning
CW10	03.03.2020	CW10	05.03.2020	Ground Stations/ULS/PAP_ULS				Approved for Planning
CW10	03.03.2020	CW10	06.03.2020	GMS/GMS_D/OSPF_D				Approved for Planning
CW10	04.03.2020	CW10	04.03.2020	Ground Stations/ULS/KOU_ULS				Approved for Planning
CW10	05.03.2020	CW10	05.03.2020	Ground Stations/TTCF/NOU_TTCF				Open

Fig. 21 Screenshot of the XLSX Tasks Exporter output (blanked fields and removed columns due to data protection reasons)

- "**PINTA 4 Workspace**": Creates the database files (*.p4ws) for TimOnWeb workspaces (version < 2.6), based on the configuration of a pre-defined PINTA timeline view.
- "**TimOnWeb2 Workspace**": Generates a corresponding database file (*.t2ws) for newer TimOnWeb workspaces (version ≥ 2.6) with an even better match in appearance.
- "**Graphical Pdf**": Creates a PDF-Document (*.pdf), again with the intention to mimic the appearance of a PINTA timeline view.

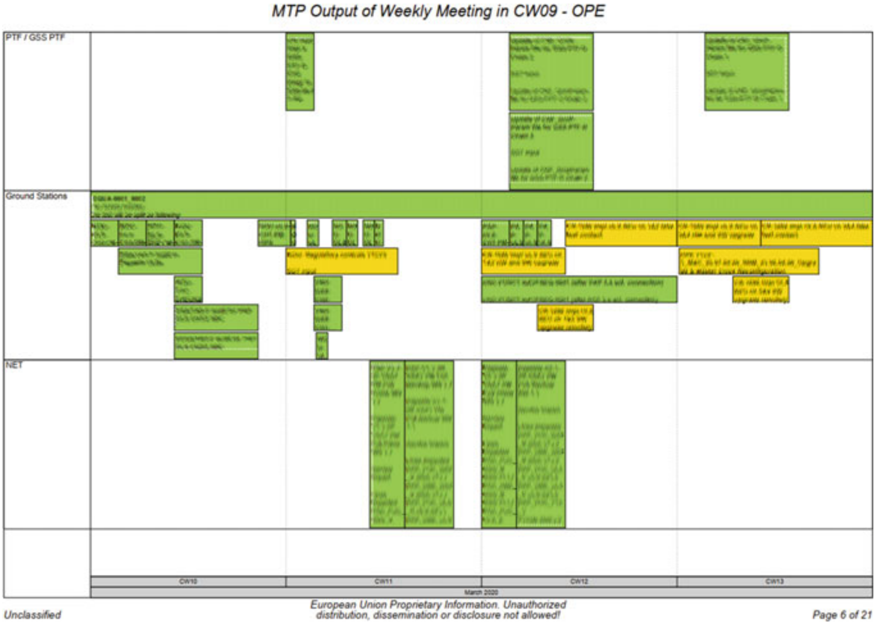
These exporters together with the already existing "**XLSX Task Table**" exporter provide all necessary file formats, which are required to distribute the current schedules of the LTP and the two MTP chains to hundreds of mail recipients after each planning meeting, in three different file formats.

To look uniform and clear, a lot of effort has been put into harmonizing the exported pdf-files (see Fig. 22), the xlsx-files (see Fig. 21) and the configuration of the TimOnWeb workspace (see Fig. 23) with the GPT workspace (see Fig. 20).

7.3 Change-Log Exporter/Importer

When transferring data from one database into another one by hand it can be very time consuming and is also a source of human error with high probability, especially when losing concentration. This kind of manual interaction needs to be avoided in any case. For that reason, PINTA has a built-in exporter and importer for transferring database changes from one PINTA project to another one.

The "**Change-Logs**" are the foundation of the "**Undo/Redo**" concept. Every single change and accordingly the complete history of changes within the project are tracked separately and saved as so-called "**Change-Log-Entries**". Whenever a


MTP Output of Weekly Meeting in CW09 - OPE

Unclassified

European Union Proprietary Information. Unauthorized distribution, dissemination or disclosure not allowed!

Page 6 of 21

Fig. 22 Screenshot of an example page of the PDF-Exporter output (blurred due to data protection reasons)

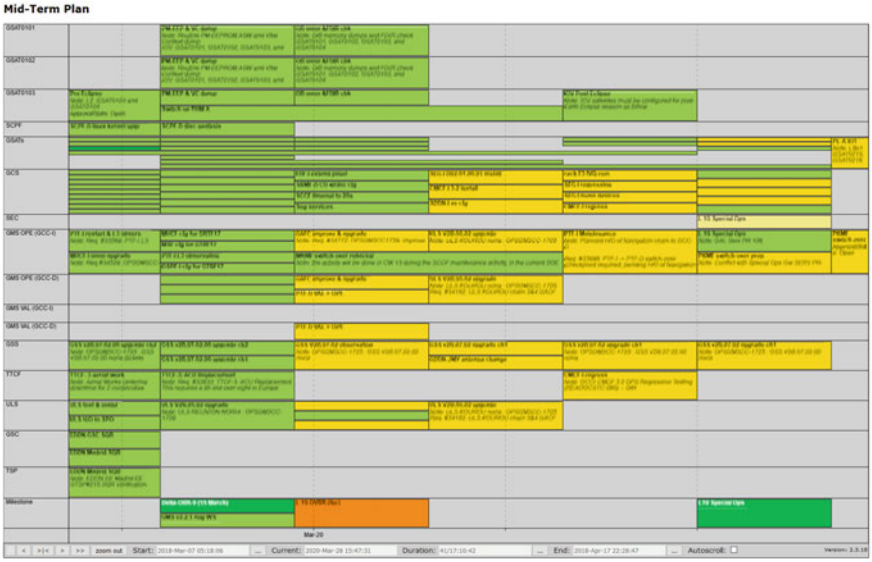

Mid-Term Plan

Fig. 23 Screenshot of the the TimOnWeb workspace with MTP data provided by the GPT (blurred due to data protection reasons)

project is saved, the Change-Log-Entries are saved as well and every change can be reversed even after restarting the program.

Every modification is represented as one specific change log of one of the following types:

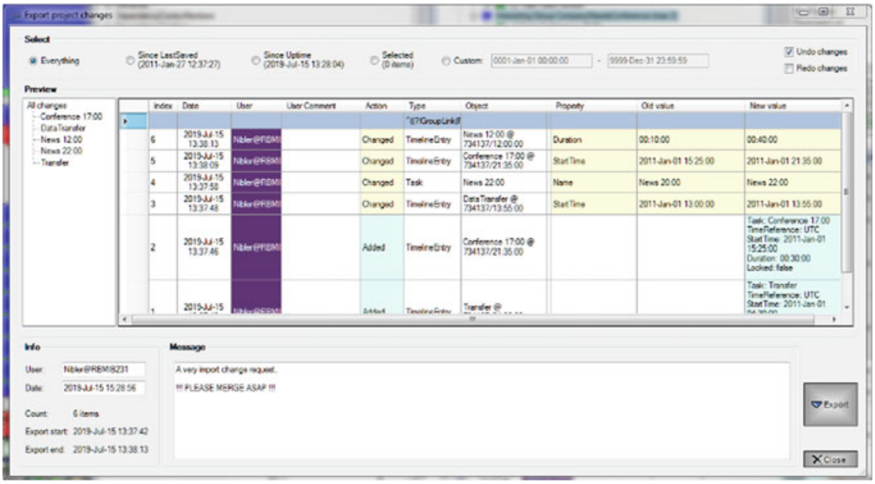
- **Changed**: An object property has been changed.
- **Added**: A new object has been added.
- **Deleted**: An object has been deleted.

The built-in "**Change-Log-Editor**" provides functionalities to visualize, filter, undo, redo, enable, disable and clean-up Change-Logs for the current project (see Fig. 24). The "**Change-Log-Exporter**" can be used to export change logs that were made within the project. These files can then be used as patches, to update the database of another project.

Accordingly, the "**Change-Log-Importer**" is used to import these changes and merge them into the database of the current project.

Merging change-sets, means synchronising database, which is a tough problem in computer science. If for example multiple patches are imported which are assumed to be created by different people, there is always a high chance for conflicts, probably because nobody is aware of the changes made by everybody else. This could lead to the same property being modified in two different incompatible ways, to an object getting deleted twice, or to an object getting changed, which has already been deleted by another user. To deal with this kind of problems a user interaction is required to fix the problem in one or another way.

- **Ignore**: The current conflict will be ignored by disabling the Change-Log-Entry. The state of the database will not be affected by this change.



● Everything ○ Since LastSaved (2011-Jan-27 12:37:27) ○ Since Uptime (2019-Jul-15 13:28:04) ○ Selected (0 items) ○ Custom: 0001-Jan-01 00:00:00 - 9999-Dec-31 23:59:59

☑ Undo changes
□ Redo changes

All changes
- Conference 17:00
- DataTransfer
- News 12:00
- News 22:00
- Transfer

Index	Date	User	User Comment	Action	Type	Object	Property	Old value	New value
					"{(?!GroupLinkI!				
6	2019-Jul-15 13:38:13	Nibler@RBMI		Changed	TimelineEntry	News 12:00 @ 734137/12:00:00	Duration	00:10:00	00:40:00
5	2019-Jul-15 13:38:09	Nibler@RBMI		Changed	TimelineEntry	Conference 17:00 @ 734137/21:35:00	StartTime	2011-Jan-01 15:25:00	2011-Jan-01 21:35:00
4	2019-Jul-15 13:37:50	Nibler@RBMI		Changed	Task	News 22:00	Name	News 20:00	News 22:00
3	2019-Jul-15 13:37:48	Nibler@RBMI		Changed	TimelineEntry	DataTransfer @ 734137/13:55:00	StartTime	2011-Jan-01 13:00:00	2011-Jan-01 13:55:00
2	2019-Jul-15 13:37:46	Nibler@RBMI		Added	TimelineEntry	Conference 17:00 @ 734137/21:35:00			Task: Conference 17:00 TimeReference: UTC StartTime: 2011-Jan-01 15:25:00 Duration: 00:30:00 Locked: false
1	2019-Jul-15	[illegible]		Added	TimelineEntry	Transfer @			Task: Transfer TimeReference: UTC StartTime: 2011-Jan-01

User: Nibler@RBMI8231
Date: 2019-Jul-15 15:28:56
Count: 6 items
Export start: 2019-Jul-15 13:37:42
Export end: 2019-Jul-15 13:38:13

A very import change request.

!!! PLEASE MERGE ASAP !!!

Fig. 24 Screenshot of the PINTA Change-Log exporter, showing a patch with changes that can be exported

- **Repair**: If changes lead to a situation were different valid states are possible, the user can interactively decide which solution to consider as the correct one.
- **Cancel**: The merging of the current change will be cancelled and the whole patch will be reverted, without causing any changes to the database.

Specifically, for the GPT, the functionality of the Change-Log Exporter and Importer had to be expanded even more. Although the export and import of patches works smoothly, the problem hereby is, that the content of the files is not in a human-readable form, but very technical. A way had to be found to make patches readable, and to make things even more complicated, even changeable by the user. For that reason, an interface was developed to the so-called "**MTP-Tool**", an EXCEL macro sheet, developed by Spaceopal. A detailed description of this tool and corresponding workflows can be found in [11].

7.4 Service-Tree Importer

The "**Service-Tree**" is a hierarchical tree-structure for all ground, navigation and space related services that are available at Galileo. The granularity of this tree can be as deep as seven or more hierarchy levels, resulting in hundreds of entries. Every activity within the GPT can be assigned to any number of services. To make things even more challenging, the structure of this tree is not fixed, but changes on a regular basis.

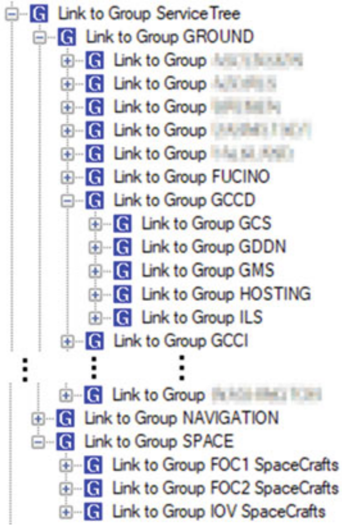
Fortunately, this flexible tree-like behavior is one of the basic principles of how the PINTA database is designed. Briefly described, the model consists of objects like "**Groups**", "**Tasks**", "**Parameters**" and "**Resources**" which are connected with "**Constraints**". All these model objects can be arranged in a hierarchical group structure, as can be seen in Fig. 25. For a more detailed overview of the GSOC planning modelling language have a look at [6] or [2].

In order to keep up with the changes to the Galileo Service Tree, it was therefore necessary to develop a plugin that automatically applies updates to the PINTA Project Tree.

7.5 Task-Editor

Within PINTA the Task-Editor is the main tool to display the details of individual tasks and to make manual modifications. Most of the already available generic functionalities like changing the name or comment, writing notes, moving the tasks start and end time, or tracking changes of individual users are very helpful but were not enough to support the Galileo planning process.

As already mentioned in the subsection about PINTA-Plugins (see Fig. 26) it is possible to embed mission-specific extension panels into the Task-Editor's GUI.



```
⊟ Link to Group ServiceTree
  ⊟ Link to Group GROUND
    ⊞ Link to Group [redacted]
    ⊞ Link to Group [redacted]
    ⊞ Link to Group [redacted]
    ⊞ Link to Group [redacted]
    ⊞ Link to Group [redacted]
    ⊞ Link to Group FUCINO
    ⊟ Link to Group GCCD
      ⊞ Link to Group GCS
      ⊞ Link to Group GDDN
      ⊞ Link to Group GMS
      ⊞ Link to Group HOSTING
      ⊞ Link to Group ILS
    ⊞ Link to Group GCCI
⋮   ⋮     ⋮
    ⊞ Link to Group [redacted]
  ⊞ Link to Group NAVIGATION
  ⊟ Link to Group SPACE
    ⊞ Link to Group FOC1 SpaceCrafts
    ⊞ Link to Group FOC2 SpaceCrafts
    ⊞ Link to Group IOV SpaceCrafts
```

Fig. 25 PINTA Project Tree, containing the whole Galileo service tree

The so-called “**GSOp PR-Editor**” is a new plugin that was written to fulfil the requirements for the GPT. Not only simple extensions such as various ticket IDs, additional description fields, information about the originator, or flags like “Service Impact”, “Urgency” or “Hazardous Condition”, with pre-defined values had to be implemented. To make the user’s work easier, faster, and less prone to errors, more complex functionality was needed.

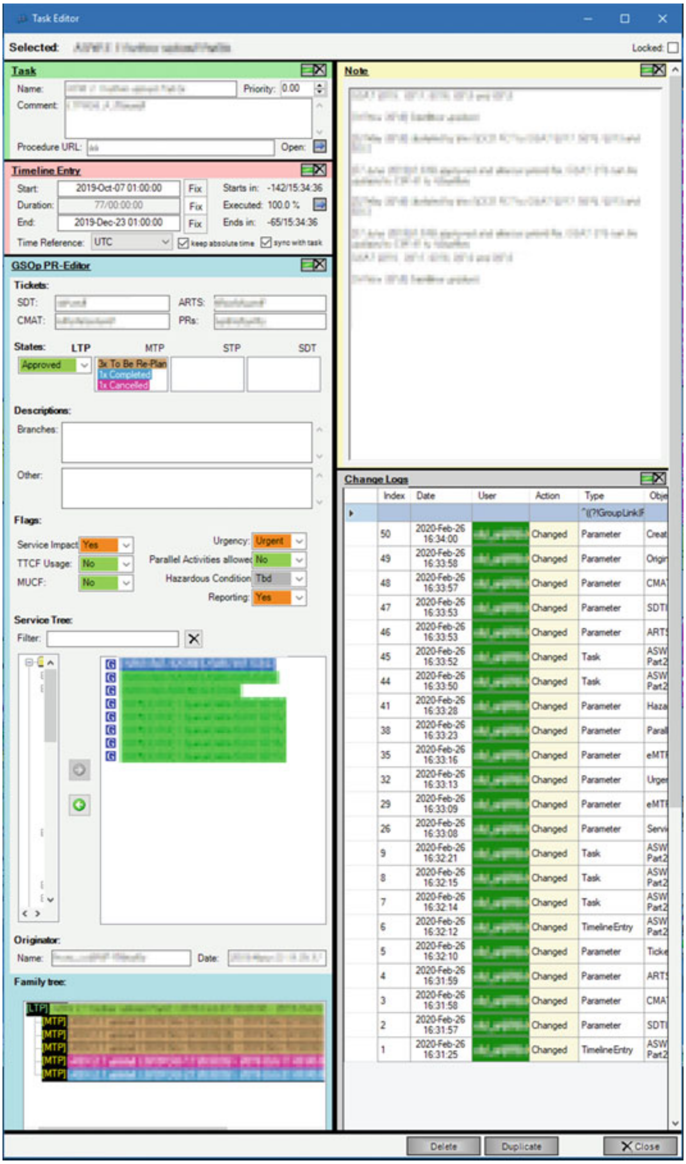
The “**Service Tree**” section contains the full tree-structure of all the services that are available at Galileo. And every one of them can be assigned or unassigned to every activity, which also affects all child services in a recursive way. This can conveniently be done by selecting and clicking a button.

Within the “**Family Tree**” all the parent–child relationships of activities are shown, even if they belong to different timelines, and highlighted according to their planning state. Furthermore, this tree can also be used to create or delete these relations between activities easily via drag and drop.

The “**States**” section can be used to change the planning state of the current activity. Besides, it also shows a summary of all the planning states of all parent and child activities, which gives the user a fast and easy to grasp overview of the relationships.

7.6 SDT Interface

Most of the inputs for the GPT, mainly for MTP-VAL and MTP-OPE, come from another tool, the so-called Service Desk Tool (SDT). Basically, all the activities, like



Task Editor

Selected: [illegible] Locked: □

Task

Name: [illegible] Priority: 0.00

Comment: [illegible]

Procedure URL: [illegible] Open:

Timeline Entry

Start: 2019-Oct-07 01:00:00 Fix Starts in: -142/15:34:36

Duration: 77/00:00:00 Fix Executed: 100.0 %

End: 2019-Dec-23 01:00:00 Fix Ends in: -65/15:34:36

Time Reference: UTC ☑ keep absolute time ☑ sync with task

GSOp PR-Editor

Tickets:

SDT: [illegible] ARTS: [illegible]

CMAT: [illegible] PRs: [illegible]

States: LTP MTP STP SDT

Approved | 3x To Be Re-Plan
1x Completed
1x Cancelled

Descriptions:

Branches:

Other:

Flags:

Service Impact: Yes Urgency: Urgent

TTCF Usage: No Parallel Activities allowed: No

MUCF: No Hazardous Condition: Tbd

Reporting: Yes

Service Tree:

Filter:

Originator:

Name: [illegible] Date: [illegible]

Family tree:

[LTP]

[MTP]

[MTP]

[MTP]

[MTP]

[MTP]

Note

Change Logs

Index	Date	User	Action	Type	Obje
				^((?!GroupLinkF	
50	2020-Feb-26 16:34:00	[illegible]	Changed	Parameter	Creat
49	2020-Feb-26 16:33:58	[illegible]	Changed	Parameter	Origir
48	2020-Feb-26 16:33:57	[illegible]	Changed	Parameter	CMA
47	2020-Feb-26 16:33:53	[illegible]	Changed	Parameter	SDTI
46	2020-Feb-26 16:33:53	[illegible]	Changed	Parameter	ARTS
45	2020-Feb-26 16:33:52	[illegible]	Changed	Task	ASW Part2
44	2020-Feb-26 16:33:50	[illegible]	Changed	Task	ASW Part2
41	2020-Feb-26 16:33:28	[illegible]	Changed	Parameter	Haza
38	2020-Feb-26 16:33:23	[illegible]	Changed	Parameter	Parall
35	2020-Feb-26 16:33:16	[illegible]	Changed	Parameter	eMTI
32	2020-Feb-26 16:33:13	[illegible]	Changed	Parameter	Urger
29	2020-Feb-26 16:33:09	[illegible]	Changed	Parameter	eMTI
26	2020-Feb-26 16:33:08	[illegible]	Changed	Parameter	Servi
9	2020-Feb-26 16:32:21	[illegible]	Changed	Task	ASW Part2
8	2020-Feb-26 16:32:15	[illegible]	Changed	Task	ASW Part2
7	2020-Feb-26 16:32:14	[illegible]	Changed	Task	ASW Part2
6	2020-Feb-26 16:32:12	[illegible]	Changed	TimelineEntry	ASW Part2
5	2020-Feb-26 16:32:10	[illegible]	Changed	Parameter	Ticke
4	2020-Feb-26 16:31:59	[illegible]	Changed	Parameter	ARTS
3	2020-Feb-26 16:31:58	[illegible]	Changed	Parameter	CMA
2	2020-Feb-26 16:31:57	[illegible]	Changed	Parameter	SDTI
1	2020-Feb-26 16:31:25	[illegible]	Changed	TimelineEntry	ASW Part2

Fig. 26 Screenshot of the the GPT Task-Editor, with the embedded PR-Editor plugin (blurred due to data protection reasons)

support, resource bookings, system changes, maintenance, etc., that take place at Galileo are tracked within SDT.

As a logical consequence, to make manual data exchange superfluous, an interface to the SDT is the next step, which is currently under development. This would complete the entire flow of information from the creation of a ticket, the approval process, the execution and the final reporting to an overall system. This topic is explained in more detail in [11, 12].

From the implementation point of view for the GPT this means that in addition to LTP, MTP and STP an additional timeline for SDT is required. Since this hardly differs from the existing plans in terms of modeling, graphical representation and required user interaction, the effort for this is relatively low. However, the export or import of the data is more complicated in order to keep the GPT and the SDT in a permanently consistent state. The work on the interface will therefore continue to be challenging …

8 Conclusions and Outlook

In summary, we can say that we see ourselves as well positioned with our GSOC mission planning tool suite that has been developed within and around PINTA. During decades of mission planning experience at GSOC, PINTA has proven to be flexible, reliable and generically applicable for solving various mission planning problems with a different degree of automation. The range of applications here is not restricted to serving single spacecraft missions, but also comprises solving multi-mission planning problems like the allocation of on-call shift personnel and supporting the sharing of ground resources such as control room and ground station availabilities, or the Galileo service operations planning, as has been explained in this paper.

Furthermore, PINTA can be easily combined with the other components and libraries of the GSOC mission planning tool suite, such as the timeline web viewer TimOnWeb, the planning library PLATO, the event-calculation library SCOTA, or the so-called "Swath Preview and Ordering Tool" (SPOT) [5]. Of course, that is just as important as the well-established integration with the other GSOC sub-systems, such as flight dynamics, satellite and ground station monitoring and control, via generic interfaces.

What is not to be underestimated is the fact that the approach of having a comfortable in-house development for all the above—mentioned mission planning tool suite is much appreciated. This has proven to be beneficial as it easily allows for including direct feedback from operations personnel as well as other mission responsible persons and creates expertise for estimating probable future requirements. And what should also not be kept secret, even if nobody likes to talk about it, is that the biggest advantage of controlling your own software is that debugging and troubleshooting is much easier und way faster.

While in the last years the focus of the ongoing evolution of PINTA lay on modernization, enhancing the robustness and extending the features, also the enhancement of the user friendliness became more and more important. This comprises simplifying workflows, improving the GUI functionalities, performance enhancements, provision of simplified and extended information export capabilities, and, last but not least, documentation. For instance, e.g. a convenient handbook and the provision of trainings and instruction material have to be promoted.

Among the big milestones for the next years, one is to establish the connectivity to the upcoming generic GSOC "**Reactive Planning framework**" (formerly known and presented as the "**Incremental Planning System**" [13]), making PINTA an optional front-end of this, to allow for having interactive mission planning systems also basing on the same integrated, widely configurable and scalable tool suite as already foreseen for the fully automated use cases. Further, not less important, development tasks will be accompanying common developments of the GSOC modelling language as well as the international standardization of interfaces and mission planning and scheduling services in the CCSDS environment [14].

Nevertheless, in parallel, of course, there are always modifications for specific project and user needs wherever requested and viable. Having the control about the development of the software tool-suite it is ensured that further modifications, feature enhancements and functionality extensions will not be missed to stay well-prepared for and in the future as well. Above all, one big future goal is the integration of PINTA and TimOnWeb into one application becoming "**PintaOnWeb**".

Lots of ambitious waters have been sailed so far [1], but the journey won't end here. Not too far away and already visible on the horizon is the next adventure, when PINTA will surf the web via PintaOnWeb.

To be continued …

References

1. Wikipedia article "Pinta (ship)". https://en.wikipedia.org/wiki/Pinta_(ship)
2. GSOC planning modelling language. http://www.dlr.de/rb/Portaldata/38/Resources/dokumente/GSOC_dokumente/RB-MIB/GSOC_Modelling_Language.pdf. Cited on 2017–06–07
3. Lenzen C, Wörle MT, Mrowka F, Spörl A, Klaehn R (2012) The algorithm assembly set of PLATO. In: 12th international conference on space operations
4. Lenzen C, Wörle MT, Mrowka F, Geyer MP, Klaehn R (2011) Automated scheduling for TerraSAR-X/TanDEM-X. In: 7th international workshop for planning and scheduling in space
5. Wörle MT, Spörl A, Hartung JH, Lenzen C, Mrowka F (2016) The mission planning system for the firebird spacecraft constellation. In: 14th international conference on space operations
6. Nibler R, Mrowka F, Wörle MT, Hartung JH, Lenzen C (2017) PINTA and TimOnWeb–(more than) generic user interfaces for various planning problems. In: 10th international workshop on planning and scheduling for space
7. Geyer MP, Mrowka F, Lenzen C (2010) TerraSAR-X/TanDEM-X mission planning—handling satellites in close formation
8. Spörl A, Lenzen C, Wörle MT, Hartung JH, Braun A, Wickler M (2014) Mission planning system for the TET-1 OnOrbitVerification mission. In: 13th international conference on space operations
9. Gärtner SA, Harder N, Hartung JH, Hobsch M, Weigel M (2021) A mobile and compact control center for quick decentral satellite access. SpaceOps-2021, 4 x 1288

10. Hartung JH, Nibler R, Peat C, Spörl A, Wörle MT, Lenzen C (2016) GSOC SoE-Editor 2.0: a generic sequence of events tool. In: 14th international conference on space operations
11. Brogl S, Manailescu S, Gutierrez J, Ballweg R, Mrowka F (2020) How Galileo planning became automated. In: 16th international conference on space operations, SpaceOps 2020, Cape Town, South Africa, 18th–22nd May 2020, SpaceOps-2020, 6,2,1x249
12. Carandente V, Brogl S, Cadenas Gorgojo R, Roßgotterer R, Docal Lareu D, Ballweg R (2020) Integrated GSOp planning architecture. In: 16th international conference on space operations, SpaceOps 2020, Cape Town, South Africa, 18th–22nd May 2020, 6,2,2x246
13. Wörle MT, Lenzen C, Göttfert T, Spörl A, Grishechkin B, Mrowka F, Wickler M (2014) The incremental planning system—GSOC's next generation mission planning framework. In: 13th international conference on space operations
14. Mission Planning and Scheduling Concept, Draft Green Book (2016) Report concerning space data system standards

Gbps High Speed Antenna Arraying for Ground-Based Network

Howard Garon, Obadiah Kegege, David Caruth, Victor Sank, Frank Stocklin, Brent Andres, and Nancy Huynh

Abstract Combining the output signals from two or more ground station antennas can increase the gain of the received signal, providing the critical flexibility to increase the science data rate from space missions. NASA's Near Space Network (NSN) has developed a gigabits/sec high rate antenna arraying system, based on the coherent combination of signals derived from multiple directive antennas. This arraying system is called the "High Data Rate Signal Combiner (HDRSC)." This arraying design approach/technology has been used previously at very low data rates. This work, however, focuses on gigabits/sec high rate antenna arraying system architecture. When coherently combining just two signals there is ideally a doubling of power, i.e., a 3 dB signal-to-noise improvement. Arraying of small antennas can easily outperform a single large aperture antenna not only in radio-frequency performance but also in a substantial reduction of cost. This paper covers the design transition for the arraying approach from post- to pre-detection, hardware architecture starting from separate and distinct analog-to-digital converters to integrated tiles within a single semiconductor chip package, a rigorous and evolving test philosophy, and results.

Keywords Antenna arraying · Gbps · Near space network · Signal

1 Introduction

NASA's Near Space Network (NSN) has been developing Ka-band high-rate antenna arraying, which enables two or more antennas to function as one larger antenna capable of receiving science data at higher data rates than they could individually. Antenna arraying coherently combines radio frequency signals from a group of smaller antennas to produce a signal comparable to the signal produced by a single, larger antenna with more gain. For a number of reasons, building a single larger antenna may be impractical, especially if the site already has multiple resources in

H. Garon (✉) · O. Kegege · D. Caruth · V. Sank · F. Stocklin · B. Andres · N. Huynh
NASA Goddard Space Flight Center, 8800 Greenbelt Road, Greenbelt, MD 20771, USA
e-mail: howard.garon@nasa.gov

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_17

close proximity to each other. Applying this technology to assets already in place will enable the NASA to support missions with higher data demands sooner rather than later. The preliminary NSN antenna arraying analyses were focused on solving the problem of daily data downlink/volume and coverage needed to meet the requirements of the Roman Space Telescope. Initially, Roman Space Telescope had planned to use the Deep Space Network (DSN). A major sticking point was trying to get the DSN to firmly commit to providing the daily hours of coverage needed to meet the daily volume of 11.5 Terabits (Tbit). Working with Roman Space Telescope, NSN developed a solution based on 18-m antenna arraying to increase the data rate transmitted from Roman Space Telescope without impacting the Radio Frequency (RF) design of the spacecraft itself. In addition, the recommendation incorporated multiple data rates for the Roman Space Telescope mission to accommodate the atmospheric attenuation variation during any given pass. This strategy of “Arraying two 18-m Antennas” allowed the 18-m dedicated antenna option to provide well in excess of the 11.5 Tbit/day data volume requirement. To achieve this, two ground terminals are required—one in the northern hemisphere and the other one in the southern hemisphere to account for the earth’s rotation, seasonal tilt and the Halo orbit of Roman Space Telescope around the earth/sun L2 libration point. A number of ground station locations were studied and the recommended locations were White Sands, NM (WSC) and South Africa (SA). The resulting coverage exceeds an aggregate of 18 h/day. Having a minimal arraying capability at 600 Mbit/s increases the daily volume captured to ~30 Tbits per day.

Figure 1 shows an overview of the NSN antenna arraying system. While NASA has experimented with antenna arraying since the 1980s, the NSN arraying hardware surpasses the data rate capabilities of previous arraying technologies. Appropriately named the High Data Rate Signal Combiner (HDRSC), the HDRSC employs the latest cutting edge silicon technologies to calculate the time delay between signals

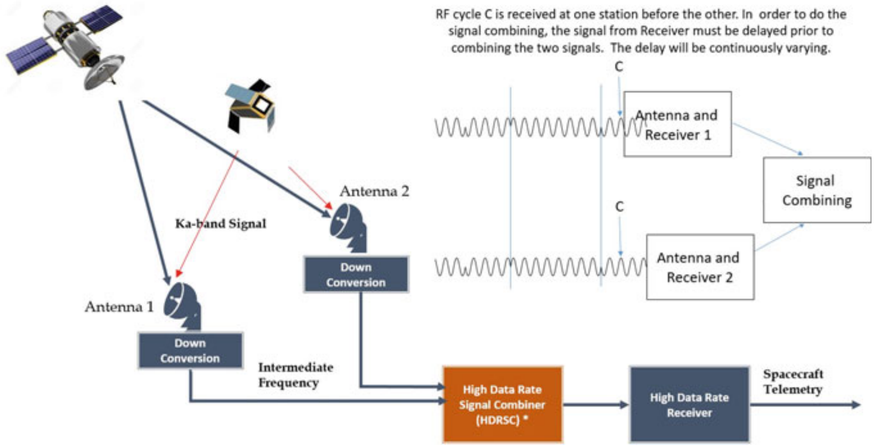

RF cycle C is received at one station before the other. In order to do the signal combining, the signal from Receiver must be delayed prior to combining the two signals. The delay will be continuously varying.

Fig. 1 Overview of the high speed antenna arraying

of spatially disparate antennas. After first converting the incoming signals to the digital domain, the unit determines the time difference between the antenna arrivals. Knowing the time delay to within utmost accuracy, the unit while still in the digital domain, adds the signals. The resulting coherently combined signal is then converted back to the analog domain. Updates to the time difference estimates are accomplished on the order of milliseconds so that the whole process can be completed in real time during a spacecraft pass. Without these real time calculations, the signals, and therefore the data, would be unintelligible. The HDRSC presents NASA with a capability that is both innovative and cost effective.

This paper discusses the systems architecture, development, and demonstration of the high data rate arraying for ground-based network. As NASA's Artemis missions move toward a planned lunar landing in 2024, improving communications between spacecraft and ground has become even more important. The recent breakthrough in the development of HDRSC will allow the network to bring gigabits per second (Gbps) of data back to Earth, expanding our discovery capabilities. Data requirements continue to grow as advanced science instruments are developed and deployed on NASA missions. Ground antennas and systems need increased capacity and capability to capture the immense amount of data coming from these missions.

2 HDRSC Systems Architecture and Development

2.1 Modeling the Arraying System

Figure 2 shows the MATLAB/Simulink model of the High Speed Antenna Arraying system. The model uses a test source representing the spacecraft and channel impediments. It incorporates both carrier and phase instabilities, and relative temporal displacement between spacecraft and antennas. It also injects uncorrelated AWGN noise. The FPGA primary cores include channels A and B carrier and phase recovery cores, the correlator core, and the output formatter core. The VHDL code for the correlator core was generated manually and embedded on the FPGA development board for testing.

As illustrated within Fig. 2, the initial analyses focused on a post-detection scheme with minimal capability to perform arraying between a pair of antennas at 600 Mbit/s, anticipating that the antennas could be separated by up to 5 km. The final working prototype pre-detection design eventually proved to have the capability to array signals beyond ~1 Gsymbols per second (Gsps) while allowing up to 10 km between antennas. The prototype is scalable and allows arraying up to six antennas while simultaneously providing a built-in test capability.

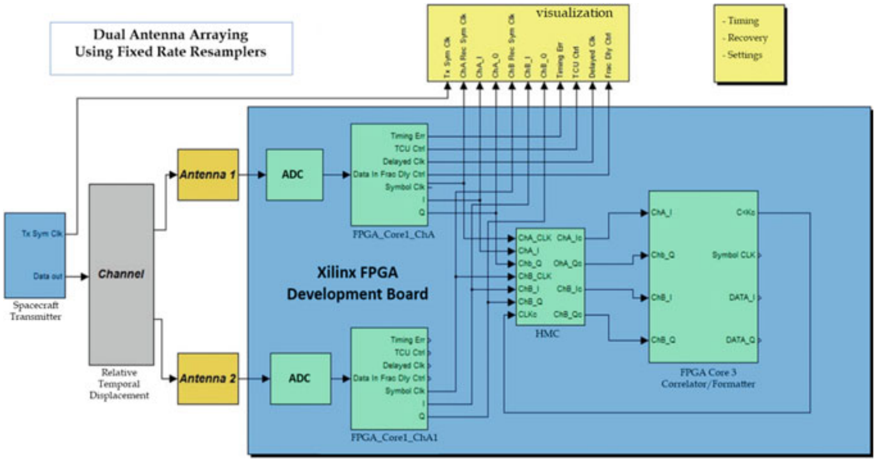


Fig. 2 High speed antenna arraying MATLAB/Simulink model

2.2 Arraying System Architecture

A generic description of the High Data Rate Signal Combiner (HDRSC) relies upon an extremely efficient and tightly-constructed cross-correlator. Referring to Fig. 3, two analog radio-frequency (RF) streams are presented on input to the HDRSC, each representing the output of a spatially disparate antenna prior to any signal processing beyond down-conversion. Each analog channel is converted to a digital stream using an analog-to-digital converter (ADC), their respective output each stored within a circular memory buffer. The pair of circular memory buffers are then simultaneously

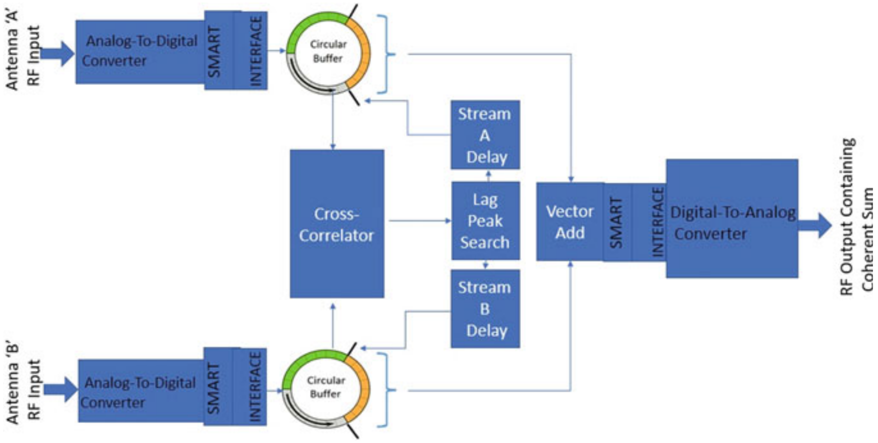


Fig. 3 Generic layout of overall high data rate signal combiner (HDRSC)

accessed, one digital stream from each buffer directed for processing by the cross-correlator.

Still another pair of digital streams, also respectively derived from the same circular buffers, serve as input to a digital vector adder. For the classes of modulated and encoded signals that are ordinarily employed in practice, the cross-correlator output will contain a sharply defined spectral peak. The more complicated the signal as modulated and emitted by the original signal source, the more sharply defined the peak will be. The location of the peak within the cross-correlator output spectrum dictates the time-delay or signal lag between the pair of signals. The information obtained from a search of the cross-correlator spectral output for this peak determines precisely where in the respective circular buffers the digital data streams used for the vector adder are derived. With this proper time offset obtained and applied between the signals, the signals are digitally combined in the vector adder to form a single digital signal output stream. The combined digital stream is converted back into the analog domain using a digital-to-analog converter.

The algorithm described belongs to a class of signal processing often referred to as “pre-detection”. The advantages of this approach are numerous. The only analog signal pre-processing required at the antenna is down-conversion from the original source carrier frequency to an intermediate frequency (IF) to facilitate handling and minimize signal loss. The spatially-separated antennas are not required to be identical, nor are the antenna signals as presented at the input to the HDRSC required to be of the same power magnitude. With this approach, performance is dictated by the properties of the cross-correlator with the primary error driven by phase differentials in the time-domain and dispersion in the frequency-domain (e.g., Doppler). As with most correlation-type algorithms, the HDRSC can construct the cross-correlation peak even if the source signal is buried in the noise. Whether the signal is subsequently detected by further processing in a conventional RF receiver is determined by how much gain is derived from the signal combiner and the characteristics of the receiver for the given modulation and encoder scheme. Our hardware implementation of the HDRSC was first realized to demonstrate the viability of our concept.

3 HDRSC Test Philosophy

The Spacecraft Pass Simulator (SPS), shown in Fig. 4, was used to simulate the signals required for testing the High Data Rate Signal Combiner (HDRSC). The SPS replicates a complete spacecraft pass as it would appear to two fixed antennas from acquisition (AOS) through loss of signal (LOS).

Using a SAFRAN High Data Rate (HDR) test modulator as input, the Spacecraft Pass Stimulator (SPS) outputs two correlated signals each with independent (uncorrelated) noise at either the 1200 MHz or 2400 MHz IF. The test modulator has the capability to output at data rates up to 2 Gbps. The SPS directly outputs these two signals to the High Data Rate Signal Combiner (HDRSC) on the two separate IF

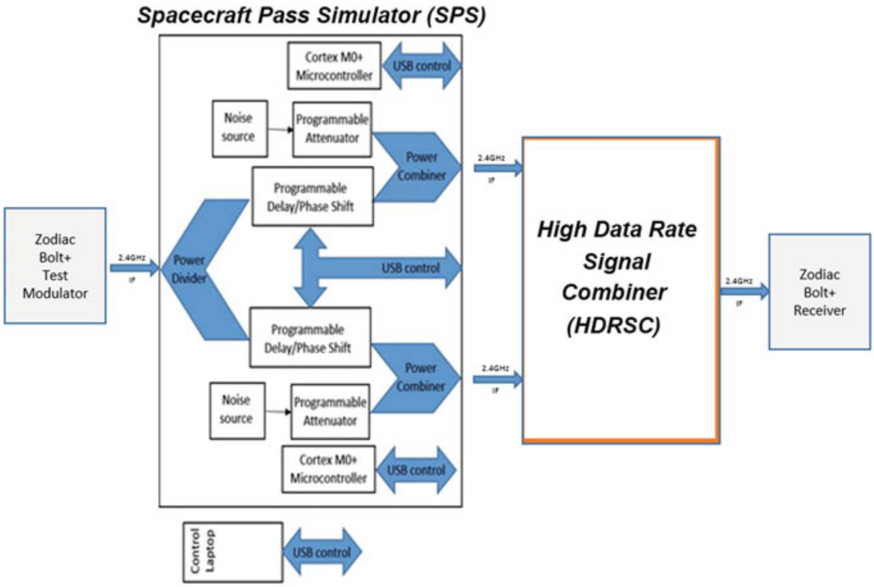


Fig. 4 End-to-end test for the high data rate signal combiner

paths. The coherently combined result as derived within the HDRSC may then be presented at IF to a High Data Rate (HDR) receiver where the signal is subsequently demodulated and decoded using conventional means.

4 HDRSC Test Results

The demonstration as constructed to establish viability of the design concept relied upon the Spacecraft Pass Simulator (SPS) in conjunction with the Zodiac HDR Bolt+ in precisely the same test configuration as described immediately above. This test setup was augmented in order to fully demonstrate the speed at which time-domain coherent results are available at the output of HDRSC as result of any dynamic changes in the HDRSC dual channel input at either or both channel inputs.

Starting with a Roman Space Telescope replica signal (600 Mbps, OQPSK, LDPC 7/8, CCSDS randomized), the basic demonstration consisted of dropping the signal bit-energy to noise ratio (Eb/No) on both channels until the Bolt+ was completely unable to maintain carrier lock on either channel. Almost immediately (approximately 150 ms) after connecting both channels to their respective inputs on the HDRSC, the Bolt+ achieved both bit and symbol lock. We were able to duplicate this scenario regardless of the temporal lag between the two signals up to the maximum prototype design differential (±16 μs). This equates to the maximum lag expected between a pair of ground stations spaced 5 km apart. The demonstration incorporated signal pairs at high Eb/No as well in order to illustrate the range of signal dynamic range handling without HDRSC output impairment. In stark contrast to the high Eb/No test, one channel at a time was abruptly removed from the HDRSC input and the HDRSC output automatically reverted to the other channel without interruption. The output of the HDRSC, of course, was without the coherent gain that could be derived from both channels on input. The test description above met the requirements for the Roman Space Telescope mission. Other test sets performed validated that the HDRSC can handle signal modulation data rates up to 1.2 Gsps given a maximum antenna separation of 5 km (km).

5 Analysis of Roman Space Telescope Downlink with Antenna Arraying

The plots on Fig. 5 represents 24 h of earth visibility from the proposed Halo orbit. The colored coverage areas were obtained by running 1 year worth of coverage data. This includes the annual tilt of the Earth's orbit in conjunction with the Halo orbit for Roman Space Telescope about L2—the earth/sun libration point. It is important to note that both a northern and southern latitude ground stations are required to provide significant complementary geometric coverage.

Using arraying enables the downlink data rate to be increased to 600 Mbs for Roman Space Telescope. The current daily volume requirement is 11.5 Tbs/day. Implementing arraying at WSC only would increase the daily volume collection to ~29 Tbs/day (assumes 300 Mbs at SA). Implementing arraying at WSC & SA would increase the daily volume collection to ~38.8 Tbs/day. This allows increased

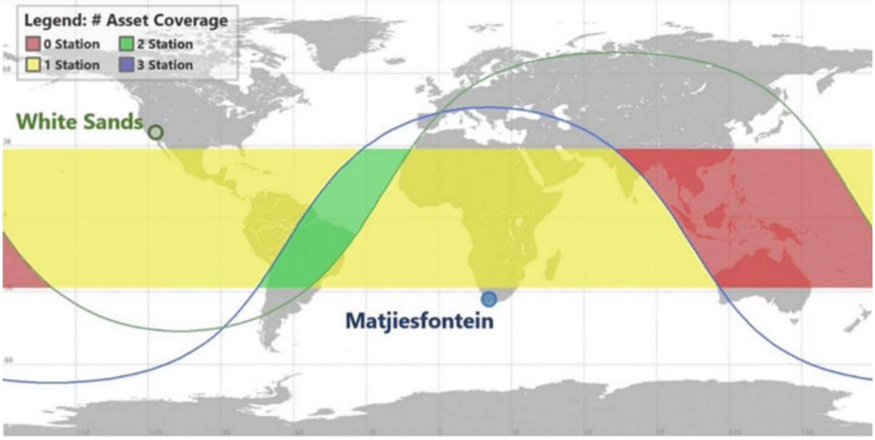


Fig. 5 Roman space telescope coverage with the assumption of arraying ground station at white sand (WSC) and South Africa (SA)

science recovery with additional instrumentation or shorter contact times at WSC/SA allowing these terminals to provide support to other missions.

As summarized in Table 1 (3A and 3B), arraying two 18-m antennas at White Sands without upgrades with 50° K low noise amplifier (LNA), can achieve 584.8

Table 1 Roman space telescope 18-m Ka-band downlink margin summary (eight scenarios)

Link ID	Description	Information rate	Margin (notes 1 and 2)
1A	Ka-band downlink via 18-m WS1 At 10° elevation angle and 95% rain availability	293.1 Mbps	1.0 dB
1B	Ka-band downlink via 18-m WS1 At 20° elevation angle and 95% rain availability	451.9 Mbps	1.0 dB
2A	Ka-band downlink via upgraded 18-m antennas at white sands (with a new 50° K LNA) At 10° elevation angle and 95% rain availability	422.7 Mbps	1.0 dB
2B	Ka-band downlink via upgraded 18-m antennas at white sands (with a new 50° K LNA) At 20° elevation angle and 95% rain availability	801.7 Mbps	1.0 dB
3A	Ka-band downlink via arraying two 18-m antennas at white sands At 10° elevation angle and 95% rain availability	584.8 Mbps	1.0 dB
3B	Ka-band downlink via arraying two 18-m antennas at white sands At 20° elevation angle and 95% rain availability	801.6 Mbps	1.0 dB
4A	Ka-band downlink via arraying two upgraded 18-m antennas at white sands (with a new 50° K LNA) At 10° elevation angle and 95% rain availability	751.6 Mbps	1.0 dB
4B	Ka-band downlink via arraying two upgraded 18-m antennas at white sands (with a new 50° K LNA) At 20° elevation angle and 95% rain availability	1.41 Gbps	1.0 dB

Notes

1. Modulation = OQPSK, Coding = Rate 7/8 LDPC, Roman Space Telescope EIRP = 66.4 dBW
2. The Ka-band downlink margin was calculated at BER of 10^{-8} at the output of the rate 7/8 LDPC decoder
3. Margin is relative to required BER for each scenario and does NOT include any required performance margin

Mbps at 10° elevation and 801.6 Mbps at 20° elevation. Considering antenna upgrades with 50° K LNA and arraying, taking the worst case scenario at 10° elevation (Table 1, link ID 4A), we can achieve of 751.6 Mbps. Table 2 shows the link analysis that considers 50° K LNA and arraying, corresponding to link ID 4A above.

Table 2 Roman space telescope link calculation with arraying

Link 4A. Roman space telescope 751.6 Mbps Ka-band downlink via arraying two upgraded 18-m antennas at white sands (with a new 50° K LNA)
(S/C Antenna Size: 1.2-m; OQPSK; Rate 7/8 LDPC; BER = 10^{-8}; 10° elevation; 95% availability)

Parameter	Value	Remarks
01. User spacecraft transmitter power (dBW)	22.04	Note A; 160.00 WATTS
02. User spacecraft passive loss (dB)	2.50	Note A
03. User spacecraft antenna gain (dBi)	47.85	Note A; 1.2-m HGA
04. User spacecraft pointing loss (dB)	1.00	Note A
05. User spacecrafT EIRP (dBWi)	66.39	1-2+3-4
06. Polarization loss (dB)	0.10	Note A
07. Free space loss (dB)	245.02	Note B; ALT: 1,600,000.0 km EL: 10.00 DEG
08. Atmospheric loss (dB)	0.98	Note B
09. Rain attenuation (dB)	1.75	Note B; ITU Model; EXC: 5.00% ITU Rate 0.01%: 35.50 MM/HR RHGT: 4.74 km
10. Scintillation/multipath loss (dB)	0.27	Note B; ITU Model; EXC: 5.00% GS DIAM: 18.00 M, EFF: 56.00%
11. Cloud attenuation (dB)	0.00	Note B
12. Total propagation effects (dB)	2.75	Note B; ITU Model

(continued)

Table 2 (continued)

Link 4A. Roman space telescope 751.6 Mbps Ka-band downlink via arraying two upgraded 18-m antennas at white sands (with a new 50° K LNA)
(S/C Antenna Size: 1.2-m; OQPSK; Rate 7/8 LDPC; BER = 10^{-8}; 10° elevation; 95% availability)

Parameter	Value	Remarks
13. Clear sky G/T (dB/Degrees-K)	49.20	Note A; @ 10°EL
13a. Arraying gain (dB)	3.00	Note A
14. System noise increase due to atmospherics (dB)	1.96	Note B
15. Ground station G/T (dB/Degrees-K)	50.24	13+13a-14
16. Boltzmann's constant (dBW/(Hz*K))	-228.60	Constant
17. Received carrier to noise density (dB Hz)	97.36	5-6-7-12+15-16
18. Modulation loss (dB)	0.00	Note A
19. TOT information rate-(751.6 Mbps) (dB bps)	88.76	Note A
20. Differential encoding/decoding loss (dB)	0.00	Note A
21. User constraint loss (dB)	0.00	Note C
22. Received Eb/No (dB)	8.60	17-18-19-20-21
23. Implementation loss (dB)	3.50	Note A
24. Required Eb/No AT rate 7/8 LDPC decoder (dB)	4.10	Note A; BER = 1.00E-08
25. Required performance margin (dB)	0.00	Note A
26. Margin (dB)	1.00	22-23-24-25

Note A: Parameter from user input—subject to change
Note B: From class analysis if computed
Note C: Parameter not considered in this analysis

6 Conclusion and Future Work

With the current hardware configuration, the HDRSC is highly integrated on a single commercial FPGA carrier board and allows for a full built-in test capability. The prototype unit fits within a single (1U) rack chassis less than 25 cm deep. and can array a pair of correlated signals with modulation data rates up to 1 Gsps. The unit presently accommodates a pair of antennas separated by up to 10 km. Both the signal data rate and antenna separation handling capability could be increased still further with an upgrade modification to the chosen FPGA The unit is fully automatic, both digital modulation mode agnostic and spread spectrum agnostic. The HDRSC only requires an operator to point the antennas to be arrayed at the signal source.

Lab demonstration of the High Data Rate Signal Combiner (HDRSC) successfully showed the robustness of the system. The HDRSC system automatically detects the delay between the antenna channels and coherently adds the signals to achieve almost a 3 dB gain over a single antenna (two channels). Future work includes the fabrication of the operational system, ground station integration design, and installation and testing at antenna sites, and performance testing including live operational tests.

Acknowledgements Authors would like to acknowledge the support from both the (Roman Space Telescope) mission and NASA's Near Space Network (NSN).

A Novel Alternative to Bundle Protocol for Handling Data Transmission Across Disruption-Tolerant Networks

Caitlyn A. K. Singam

Abstract Communication networks are prone to disruption due to inherent uncertainties such as environmental conditions, system outages, and other factors. However, current communication protocols for state-of-the-art disruption-tolerant networks (DTNs) designed to withstand such conditions are not yet optimized for high performance over long distances, such as those encountered in deep space. Current DTN communication protocols have been documented in the literature as inherently assuming relatively low levels of signal loss, not accounting for end-to-end error rate, and presuming a lack of performance constraints governing optimal communication function. However, these assumptions and constraints frequently do not hold true outside of theoretical scenarios; therefore, there is a need for an improved communication protocol that has the ability to minimize data loss to tolerable levels over an unstable and error-prone communication link. Furthermore, any novel communication protocol should also be able to optimize transmission time: this is because current communication networks for parts of space prone to signal disruptions, particularly deep space, are fairly slow and have a low data rate, since transmitters have to trade speed for accuracy when transmitting data at a particular power level directly from deep space to Earth. Bundle protocol (BP) is an experimental protocol for handling packet transmission through DTN networks that has a number of vocal proponents in the academic and the aerospace community; however, as noted by authors of the protocol, there are a number of key areas of concern associated with BP approach, including, but not limited to, high vulnerability to denial of service (DoS) attacks and issues efficiently handling congestion and flow control schemes implemented across highly variable delay environments. BP, as a protocol which "sits at the application layer of some number of constituent internets", also utilizes internet protocols such as Transmission Control Protocol/Internet Protocol (TCP/IP) and similar alternatives to handle lower-level management of data transfer, and thus inherits the limitations associated with the implementations of such approaches (as well as those that emerge at the interface of protocols at each level), creating further vulnerabilities for potential exploitation by nefarious agents or reductions in system performance due to poor environmental conditions.

C. A. K. Singam (✉)
Department of Bioengineering, University of Maryland, College Park, Maryland 20742, USA
e-mail: csingam@terpmail.umd.edu

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_18

This work concerns the development of a novel protocol for data transmission across delay/disruption-tolerant networks, which is presented as an alternative to the bundle protocol standard. The alternative proposed herein seeks to address some of the limitations seen in bundle protocol and provide a DTN networking option with wider usability, better reliability, and improved immunity to DoS attacks. In particular, the efficacy of the proposed approach, in terms of maintaining both data integrity and transmission speed, was evaluated via simulation against BP and a set of other alternative DTN data handling methodologies from the literature and demonstrated a statistically significant improvement in performance compared to BP and other canonical communication protocols. The result is presented herein in terms of its ramifications for future DTN implementations.

Keywords Communications · Network · Signal optimization · Bundle protocol · Disruption tolerant networking · DTN

Acronyms/Abbreviations

BP	Bundle Protocol
CCSDS	Consultative Committee for Space Data Systems
COA	Course of Action
DoS	Denial of Service
DTN	Disruption-Tolerant Network(ing)
MATLAB	Matrix Laboratory Software
MAVF	Multi-Attribute Value Function
TCP/IP	Transmission Control Protocol/Internet Protocol

1 Introduction

The objective of this analysis was to simulate the performance of three different ad-hoc protocols for disruption-tolerant networking—i.e., the transfer of information through a network of nodes in contexts prone to signal interruption/signal degradation—and to perform a trade-off analysis that would yield a recommendation of the best course of action (COA) for space-based network communication.

This is important for space-based networks in particular since transmitting information over long distances—e.g. directly from the initial node to the destination node—will result in the terminal signal being relatively weak. This is problematic in a high noise (disruption-prone) environment since it will likely result in packet degradation or loss unless signal power is increased to compensate. Given that changing signal power for each transmission is impractical when one's network is located in space, optimization of the signal route is the best means of ensuring the transmitted signal reaches its destination rapidly and with maximum fidelity.

The recommended COA, as determined at the end of the analysis, is one that optimizes performance (in terms of selected parameters) in a fashion that minimizes error during transmission and transmission time to the greatest extent possible. This analysis takes into consideration the relative value of transmission time and transmission error associated with space-based communication systems (for instance, a scientific mission—the most likely type of user for a space-based relay network [1]—would prioritize data integrity much higher than transmission time since even though a longer transmission time equals greater cost, a lower level of data integrity could result in the mission's scientific objective being compromised [2]) and provides a recommendation accordingly.

In order to evaluate the different design options, two different principal metrics of interest were be taken into account: percent error on receipt (a representation of packet integrity) and transmission time (a representation of transmission speed).

At the end of the analysis, the calculated values for percent error and transmission time for each design option were combined together using a multi-attribute value function (MAVF) to yield a single quantitative value representing the relative value of each solution. The MAVF can produce values ranging from 0 (representing the worst option) to 1 (the optimal design option). The analysis objective was considered satisfied when the design option with the highest MAVF ranking was identified.

Three different design options were considered: bundle protocol, the current state of the art routing protocol for DTN [3] which picks the route that strikes a balance between distance and signal quality; distance-based Dijkstra, which selects a route that minimizes the distance travelled during transmission; and a novel value-based approach, which selects a route with the best overall signal quality and based on the overall value of the message to the user on receipt after accounting for incurred errors and transmission delays. The novel approach described herein is the "signal quality-based Dijkstra", a shorthand for the fact that it seeks to minimize the errors incurred during transmission à la the Dijkstra algorithm.

To evaluate the different design options, a generic/hypothetical space-based network comprised of 10 satellites was generated to route data from an initial node to a destination node. Apart from the distance between the initial node and destination node, which was fixed, the distance and signal quality for the link between any two nodes in the network was instantiated randomly. A Monte Carlo simulation was then used to simulate the routing of 500 packets through the network, with each packet representing one 'sample', or iteration of the Monte Carlo simulation, using the three different routing protocols being evaluated. The Monte Carlo simulation was used to vary the signal quality and inter-node distance associated with each link in the network each step, in order to represent variations due to environmental phenomena and orbital movement.

At the end of each iteration, the transmission time for each packet (based on the distance traveled by the packet, and given that the packet is transmitted as an electromagnetic wave traveling at the speed of light) and the packet's state (intact or damaged) was determined. The aggregate data from the entire simulation was then used to calculate the mean transmission time and percent error for the overall sample (determined based on the terminal state of each packet on receipt), both of which were in turn fed into a MAVF. The recommended design option was chosen based on which option was associated with the highest MAVF output value. Welch's t-tests were also used to confirm that the differences in metric values observed were in fact statistically significant: i.e. that there was enough of a difference in the performance of the different options for the choice to have a significant impact on overall system performance.

2 System Description

Since the purpose of this analysis was to determine the relative performance of different disruption-tolerant networking protocols for a generic space-based use case, the analysis used a generic, hypothetical network as its system of interest.

The modeled system (as depicted in the domain block definition diagram shown in Fig. 1) was comprised of a network of 10 space-based relay satellites, located at different distances from each other and from the Earth-based ground station. The satellites, which were modeled as nodes in a graph, all had communication links with one another, as well as with an initial node (the space-based asset generating the data being transmitted) and a terminal node (the ground station). The distances in between nodes, as well as the signal quality values for the links between nodes, were determined randomly; the one exception to this was the distance between the transmitting node (noted as a 'deep space asset/probe' in Fig. 1) and the receiving ground station, which were placed at a fixed distance from one another in order to set the scale (i.e. maximum distance) the network would operate at. The value used for the internode distance between the initial and destination nodes was the

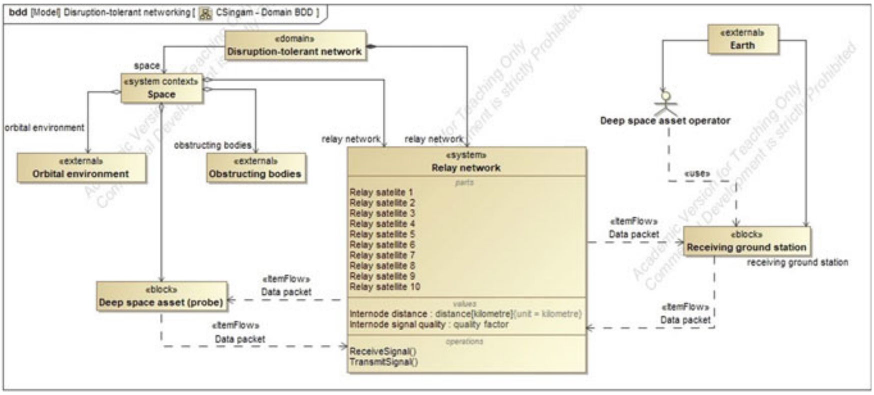


Fig. 1 A domain block definition diagram showing the principal components of the analyzed system

maximum distance from Earth to Titan, 1.27 × 10^9 km, as an actual mission—the Cassini-Huygens mission—transmitted data back to Earth at that distance [4], and the distance thus is an accurate representation of the conditions that would have to be handled by a space-based network attempting to implement any of the tested routing protocols on a practical basis.

The signal receipt and signal transmission operations that are shown as operations of the satellites within the relay network system in Fig. 1, when taken in aggregate for all the satellites within the network, represent the system behaviour dictated by the network's routing protocol. The system's ability to transmit and receive signals among its component satellites is influenced by two key input factors, internode distance (expressed in kilometers) and internode signal quality (expressed as a percentage of the theoretical ideal for signal performance).

Internode distance, determined by the user's existing asset placement, is a key factor since the time it takes to transmit a signal from point A to point B is, inherently, tied to distance: since the electromagnetic waves used to convey signals travel at the speed of light, c, the time taken to travel between node A and node B is dictated by the distance between the two nodes divided by c. Links with shorter internode distances are thus preferable to those with longer distances, as shorter transmission time is always preferable.

Signal quality is also of interest due to the fact that it is a direct reflection of the role environmental context plays in determining the quality of a link and how much of the transmitted signal gets successfully received by the end node. It represents link quality as a percent of the theoretical ideal/maximum, ranging in value from 0% to 100%, and can be calculated based on historical link performance data and information from environmental models. Though higher signal quality is always preferable, signal quality levels near the theoretical maximum of 100% are rare due to the inherent noisiness of a real-world environment. Disruption-tolerant networks need to take into account, or alternatively be resilient against, the effects of varying and/or poor signal quality lest the data payloads they are transmitting be lost or damaged.

Together, these two factors influence the two main metrics of interest for any transmission that moves through the system: percent error on receipt and transmission time.

Percent error on receipt, or the percentage of packets out of the total number initially transmitted which are damaged and/or lost during transmission, can range from 0 to 100%, but should be as low as possible (i.e. as close to 0% as possible) in order to respect the need for data integrity during transmission. This metric is viable for evaluating the performance of real-world networks since the number of packets and the number of bytes per packet in a given transmission will be defined and kept constant for a particular mission. Thus, when the users of the Earth-based ground station receive packets related to a given mission, they can compare the number of packets (and bytes) received with the number that should have been sent by the mission. Given that the entire point of a transmission is to ensure that the terminal node receives the information that was being transmitted, ensuring that percent error is as low as possible, and that as much of the data payload as possible is intact, is a high priority.

Transmission time, the amount of time (in hours) it takes for a single packet to travel from the initial transmitting node to the terminal destination (receiving) node, has a theoretical minimum dictated by the time it would take a pulse traveling at the speed of light to travel the straight-line path from initial to terminal node. However, since relay networks route packets through non-linear paths to their destination and increase transmission time beyond the minimum, it is thus important to consider how much additional transmission time the protocol incurs and attempt to keep the overall transmission time as close to the theoretical minimum as possible. For missions where data packets may contain time-sensitive commands, it is imperative to ensure that transmission time does not become excessively long.

Percent error and transmission time are the primary criteria by which the different routing protocols (the design options being evaluated) used by the network can be evaluated, and will be the focus of this analysis.

3 Design Options

The three design options being evaluated as part of this analysis are bundle protocol, distance-based Dijkstra, and signal-quality based Dijkstra. Their relative performance with respect to the two metrics of interest (described above) is shown in Table 1. The exact metric values for each design option were determined via the Monte Carlo simulation.

The details of how each of the design options work, and the parameters prioritized by each, are presented below.

3.1 Bundle Protocol

Bundle protocol is reflective of the store-and-forward methodology put forth by the Consultative Committee for Space Data Systems (CCSDS) [5] for packet routing.

Table 1 A comparison of the relative performance of the design options being evaluated, with regards to the metrics of interest

Metric	Design options		
	Bundle protocol	Distance-based Dijkstra	Quality-based Dijkstra
% error	Medium	High (suboptimal)	Low (optimal)
Transmission time	Medium	Low (optimal)	Medium to high (suboptimal)

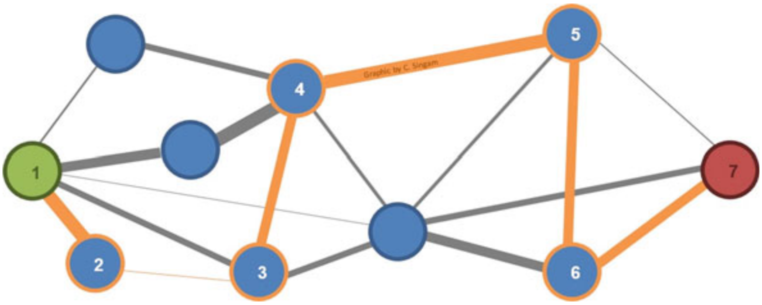


Fig. 2 An example of how bundle protocol routes information through a network from the initial (green) node to the terminal (red) node, with nodes numbered in order of visitation and the path marked in orange. Line widths correspond to signal quality (thicker lines correspond to higher quality links), and distances are to scale

As shown in Fig. 2, in a network following bundle protocol, packets are transmitted from the node they are the currently on to the adjacent/neighbouring node that meets the criteria of being A) closer than the current node to the destination node, and B) having the highest signal quality of the nodes that meet criterion A. This process is repeated until the packet reaches its destination. This typically results in the packets being routed through more nodes than in either of the other two methods, though depending on the length of the links used, this may not necessarily correspond to a longer transmission time or worse signal quality.

3.2 Distance-Based Dijkstra

This design option uses Dijkstra's algorithm to find the path with the shortest distance to the terminal node, which travels through at least one intervening relay node. The stipulation that the recommended path include at least one relay node is to eliminate direct-to-Earth transmission routes, which have already been shown in the literature [6] to be outperformed by relays. This is achieved by instantiating the edge costs of each link as the corresponding internode distance, apart from the link from the initial node to the terminal node, which is instantiated such that the edge cost is significantly higher than the edge cost of any other link and thus resulting in Dijkstra's algorithm rejecting the direct route as a possible path.

The results of this algorithm (exemplified in Fig. 3) usually yield a path that is as close to the straight-line path (marked using a dashed line in the diagram) as possible, resulting in a short transmission time but frequently resulting in a path that includes low-quality links that degrade the signal prior to receipt.

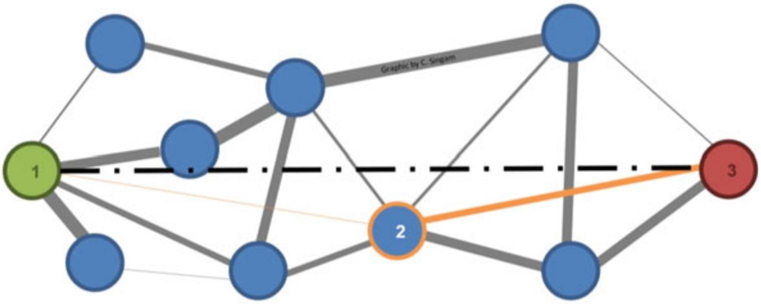


Fig. 3 An example of how the distance-based Dijkstra method routes information through a network from the initial (green) node to the terminal (red) node, with nodes numbered in order of visitation and the path marked in orange. Line widths correspond to signal quality (thicker lines correspond to higher quality links), and distances between nodes are to scale. The dashed line marks the straight-line path from the initial node to the destination

3.3 Signal Quality-Based Dijkstra

This design option evaluates the highest-quality path through the network using the Dijkstra algorithm, using 1—[internode signal quality as a decimal probability] as calculated between each pair of nodes, rather than over an entire path, as edge costs for the network. Since the Dijkstra algorithm selects the path with the minimum cost, and it is desired to maximize the signal quality seen on the chosen route, the edge costs are instantiated as the complement of the parameter of interest: in this case, signal quality.

Since this algorithm uses Dijkstra shortest-path algorithm with complement edge costs, rather than a longest-path algorithm in conjunction with a graph that uses signal quality directly as the edge cost, the results of this algorithm favour paths (exemplified in Fig. 4) with relatively few links and high signal quality. Consequently, it is good at minimizing percent error but since the algorithm favours fewer links, not necessarily shorter ones, it is suboptimal at minimizing transmission time.

4 Simulation Description

The primary objective of this analysis was to identify the mean values associated with each of the design options for the performance metrics of interest, so that a MAVF analysis could be performed and the design options could be quantitatively ranked. However, given that using signal quality-based Dijkstra for space-based networking contexts is, to the best of the author's knowledge, a concept of the author's own devising, no data exists on the performance of such methodologies for the desired use case and in the desired system context. However, given the extravagant financial and scheduling burden that would be involved in constructing any sort

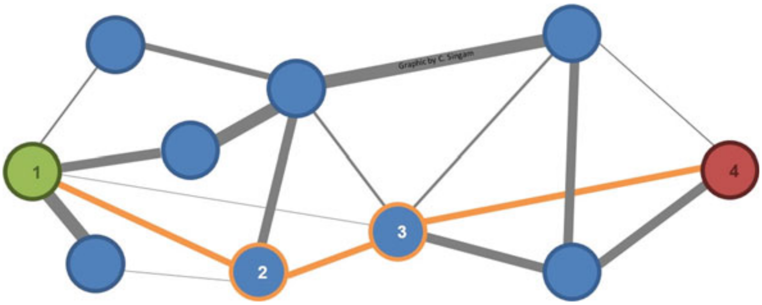


Fig. 4 An example of how signal quality-based Dijkstra routes information through a network from the initial (green) node to the terminal (red) node, with nodes numbered in order of visitation and the path marked in orange. Line widths correspond to signal quality (thicker lines correspond to higher quality), and distances are to scale

of reasonable prototype of a space-based network—or otherwise acquiring even temporary access to existing networks [7]—for testing purposes, obtaining real-world data on the performance of these protocols is, disappointingly, highly impractical. Fortunately for the aerospace community, creating a model of a space-based network and simulating the movement of packets through it using various routing protocols is significantly more practical, providing the data that allows the age-old quandary (of which packet routing protocol performs best in a space-based context) at the heart of this analysis to be answered satisfactorily.

To that end, a multi-step approach was taken to develop a model of sufficient faithfulness to reality as to accurately test the mettle of the three design options of interest. Firstly, the positions of the initial and destination nodes were established in a MATLAB model; the relay network satellites were randomly instantiated so that any one satellite had an equal probability of appearing anywhere between the initial node and destination node.

After each node had been linked with each other node, and all the links had been assigned distances and random signal qualities, the movement of 500 packets through the relay network was simulated using the Monte Carlo simulation method. The packet state and transmission time were recorded for each packet when it was simulated as having reached its destination. Once all 500 packets were simulated as having traveled through the system, the percent error for the population was calculated by totaling the number of packets recorded as having been lost and dividing that by the sample size (500 packets). The mean transmission time associated with each design option was also calculated.

Since each run of the Monte Carlo simulation is associated with a different randomly instantiated graph due to the way the simulation setup was implemented in the MATLAB code, it is possible to run the simulation code multiple times in order to gain an accurate image of how all three design options perform across several different networks. Due to the processing power and time needed to run each simulation, a relatively small sample of 5 runs was used. The values for percent error and transmission time for each run were collected in Excel, and the mean, standard deviation, and standard error of the mean for the two metrics of interest were calculated for each of the design options.

The obtained 5-run means for percent error and transmission time were subsequently fed into a MAVF for final analysis [8]. The metrics were assigned preference weights per the Parnell swing weight matrix [9]: percent error, as a Parnell 'mission critical/large effect' parameter (a metric which needs to be optimized in order to ensure mission success and which mission success is sensitive to) was assigned a weight of 100; transmission time, as a 'mission effectiveness/small effect' parameter (a metric which can be used to compare the relative worth of design options, but which mission success is not as sensitive to) was assigned a weight of 20. The results of the MAVF for each design option were then directly compared and used to make the final recommendation. Additionally, as a confirmatory measure for whether the designs performed significantly differently from one another, the 5-run data means and standard deviations were used to perform one-tailed Aspin-Welch t-tests [10] to identify whether or not the differences seen between design options for both transmission time and percent error were statistically significant ($p < 0.05$). The equation for the t-statistic comparing the means of samples from two different populations (population 1 and population 2) is:

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{1}$$

where $\overline{x}$ is the mean for a sample, s is the sample's standard deviation, and n is the sample size.

The workflow and methodology for the trade-off analysis is summarized in Fig. 5.

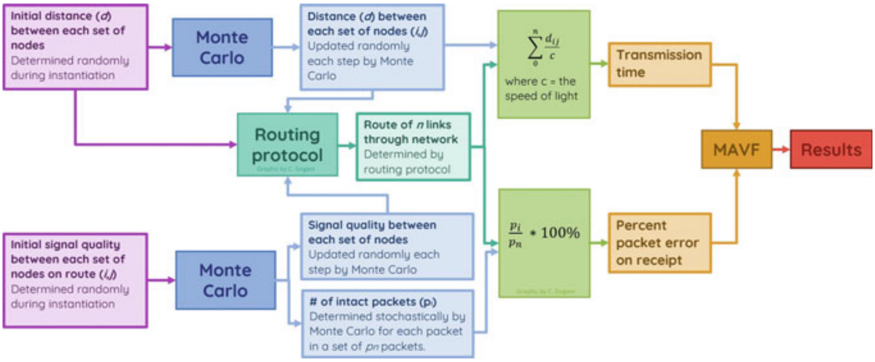

Initial distance (d) between each set of nodes
Determined randomly during instantiation
Monte Carlo
Distance (d) between each set of nodes (i,j)
Updated randomly each step by Monte Carlo
$\sum_{0}^{n} \frac{d_{ij}}{c}$
where c = the speed of light
Transmission time
Routing protocol
Route of n links through network
Determined by routing protocol
MAVF
Results
Initial signal quality between each set of nodes on route (i,j)
Determined randomly during instantiation
Monte Carlo
Signal quality between each set of nodes
Updated randomly each step by Monte Carlo
# of intact packets (pᵢ)
Determined stochastically by Monte Carlo for each packet in a set of pn packets.
$\frac{p_i}{p_n} * 100\%$
Percent packet error on receipt

Fig. 5 A combined response model diagram for the entire analysis, showing the initial input factors (purple) and intermediate factors (blue) produced by the Monte Carlo simulation and fed into the routing protocols being evaluated (turquoise). The equations used to calculate the metrics of interest (green) as they are fed into the MAVF (yellow) are also shown

5 Supporting Methods

5.1 *Network Graph Models*

The relay network that the packets are routed through is represented in the programmatic implementation as a digraph, which is compatible with MATLAB's Dijkstra implementation via the `shortestpath` function [11]. The use of network graphs to model satellite networks is not unusual [12, 13]; however, such graphs are typically used to model control capabilities rather than routing.

The two network digraphs used in this analysis are nearly identical to each other for any given run of the MATLAB program. Each accepts the list of nodes (the 10 relay satellites, the probe, and the ground station) as inputs, as well as arrays representing the links in between nodes (the same for both). The output of these both these models are the fully instantiated and functional digraphs that the Monte Carlo simulation can route its packets through and that the Dijkstra algorithm can be applied to.

As discussed previously, to prevent the direct-to-Earth link (i.e. straight line path from initial to destination node) from being used , the edge cost for the link associated with the direct-to-Earth route on each graph is assigned a value that far exceed the maximum possible value for the edge costs on the other links.

The only difference between the two digraphs is the parameter used as the edge cost for each.

Distance-based graph

Before the graph models are instantiated, each node is assigned an x and y coordinate on the Cartesian coordinate plane (with units in kilometers). Though the coordinates for the initial node and the destination node are fixed so that they are diametrically apart on the coordinate plane and at Titan distance from one another, the relay network's satellites are all assigned coordinates randomly, per a uniform distribution. The lower limit on coordinate distances is defined as 10^4 km (the same as low Earth orbit [14]), and the upper limit is defined by the location of the transmitting probe.

The distance-based digraph uses the randomly assigned position of each node to calculate internode distances and assign edge costs for each link (with the exception of the direct-to-Earth link, as described above). The distance formula is:

$$Distance_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \tag{2}$$

For convenience in implementation, since the metric of interest is transmission time (not transmission distance)—which is distance divided by a constant (the speed of light)—the edge costs for the distance-based digraph are actually assigned transmission times as edge costs. This does not result in any practical difference in how the model functions or how the Monte Carlo simulation interacts with it; it simply makes programmatic implementation easier.

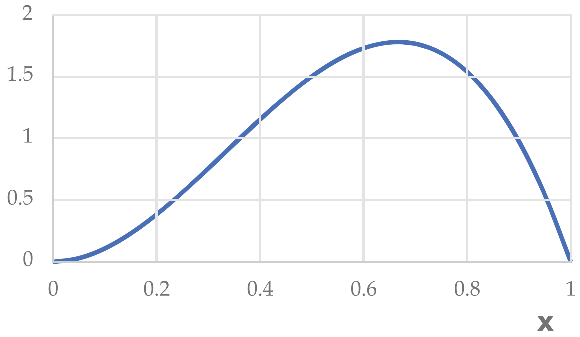


Fig. 6 Beta distribution, a = 3, b = 2. Graph generated using Excel and data from Casio’s Keisan online calculator [15]

Signal quality-based graph

Since signal quality can theoretically vary between 0 and 1 (when expressed as decimals instead of as percentages), but must to skew slightly more to the right than a normal distribution (since satellites are designed with antennas, etc. that are optimized for performance in their environments), a beta distribution was used to randomly determine the signal quality to each link in the model. Figure 6 shows the beta distribution with the parameters selected for this analysis (a = 3, b = 2).

The complement of the signal quality values produced by the beta distribution are then calculated using 1-[signal quality] and assigned to the corresponding links as edge costs.

5.2 *Monte Carlo Simulation*

The Monte Carlo simulation used in this analysis follows standard implementation methodology presented in the literature [16]. The inputs to the simulation are the fully instantiated digraphs described above, with edge costs reflecting the randomly assigned internode distances and signal qualities.

Each iteration of the simulation, a packet is routed through the network as represented by the digraphs. With each step, the packet advances from the node it is currently on to the next node in the path dictated by each routing protocol. In effect, there are three ‘copies’ of the packet moving through the network simultaneously—each subject to a different protocol, but all experiencing the same conditions. Each ‘hop’ between nodes is associated with transmission time—added to a running total that is recorded when the packet reaches its destination—and the possibility that the packet has been lost/damaged mid-transmission, which is determined stochastically by comparing a randomly generated decimal (between 0 and 1) with the assigned signal quality (not to be confused with the edge cost) of the link that has been traveled over. If the random decimal is lower than the link quality, the packet is deemed to have survived the ‘hop’, else it is recorded as having been irreparably damaged/lost. For the sake of data collection (and avoiding the complexities associated with censored data

[17]), 'lost' packets are not removed from the transmission queue and are simulated like intact packets all the way to their destination.

The edge costs are updated each step to represent environmental fluctuations; they are all randomly assigned new values per a normal distribution centered around their initial (i.e. at the start of the simulation) or 'default' value. The digraphs are re-established accordingly with the updated edge costs, and each of the routing protocols re-evaluate their recommended paths based on the node their 'copy' of the packet is currently on and the new edge costs.

Once all three 'copies' of the packet reach their destination, the transmission time (calculated based on distance traveled and independent of simulation time) and packet state for each copy is recorded. The digraphs are then reset to their state at the start of the simulation after each iteration for the sake of consistency between samples/packets, and the iteration ends. Each run of the Monte Carlo simulation simulates the movement of 500 packets through the system (i.e. there are 500 samples/iterations). This was determined to be a sufficient number of samples based on a cumulative running mean (CRM) plot for transmission time for all three routing protocols being evaluated: the mean for bundle protocol, the most variable of the three, stabilized after around 250 samples, which means that 500 samples provides a wide contingency margin (Fig. 7).

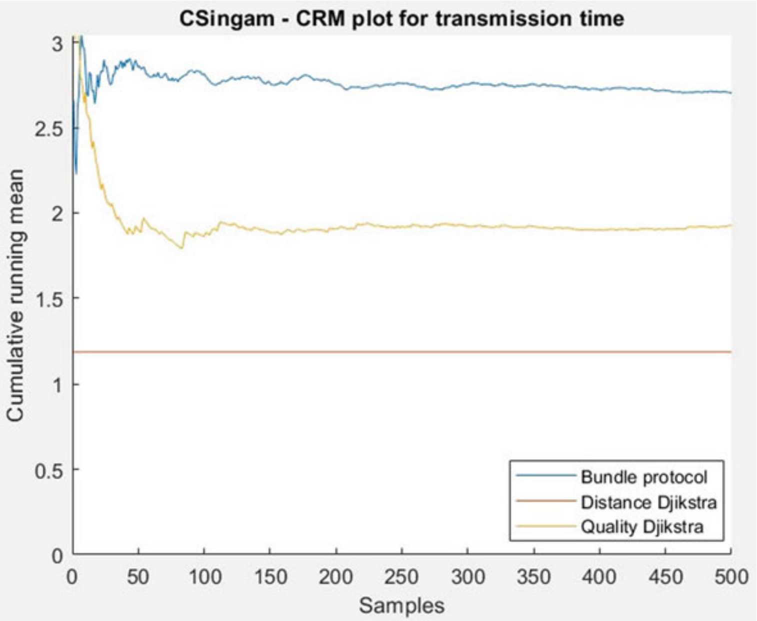


Fig. 7 Cumulative running mean plot for the Monte Carlo simulation, showing stabilization of the mean for all three protocols after around 250 samples

6 Simulation Verification and Validation

6.1 Verification and Validation Methodology

Given that the foundation for the analysis presented herein is a simulation, and thus a generalized representation of the actual behaviour of a real-world satellite network, it was necessary to perform verification and validation testing in order to ensure that the simulation was able to produce accurate and reliable results on a consistent basis as ascertained via verification testing, and that the simulation was able to provide the data needed to compare the performance of the different routing protocols which was verified via validation testing.

To this end, a series of element-level and system-level formal verification and validation tests were performed in order to make a final determination of whether the simulation was of an acceptable level of accuracy.

Development and testing, as with many software projects, followed a spiral development pattern, with individual elements being developed and tested before integration and testing of the larger integrated system. The deterministic module, which is the simpler of the two modules, was developed and white-box tested first, followed by the stochastic module. Due to the presence of a shared interface between the two modules—where parameters from the stochastic module are passed to the deterministic module to perform calculations within a single Monte Carlo event step—the stochastic module's development was nonetheless closely tied to that of the deterministic module, with both elements white-box tested simultaneously. Once integration of the deterministic module and the stochastic module was complete, an element-level integration black box test was performed. Given the successful black-box integration test the system was verified and validated via a final system-level black-box test.

Verification was be performed against the system requirements, and validation was done based on whether or not the simulation was able to provide the requisite information to perform the trade-off analysis the simulation was designed to support.

The elemental testing was conducted informally through code inspections and demonstrations to ensure that the code was functional, whilst the black-box validation testing of the integrated system was conducted formally (involving the development of formal test reports and collection of test data).

The measures of effectiveness (MOEs) that were be used to evaluate the simulation are described in Table 2.

6.2 Verification and Validation Success

The simulation was run through a number of test cases (Tables 3 and 4) as part of the black-box testing process, and the results compared to the manually-derived solutions for recommended route, transmission time, and predicted percent error for a particular network configuration. Path recommendations were verified based on

Table 2 The list of criteria used to evaluation whether the DTN simulation met the standard for acceptability

MOE	Definition
Requirements compliance	Percentage of system requirements met by the simulation, as determined via verification testing
Requirement deviance	Percentage of system requirements for which the system has received waiver(s). This is separate from requirements non-compliance, as deviations from the requirements require justification and stakeholder buy-in (specifically from the user)
Deterministic mode accuracy—protocol percent error	The deterministic mode result for protocol percent error divided by the result determined via manual calculations. (Fraction expressed as a percentage)
Deterministic mode accuracy—transmission time	The deterministic mode result for transmission time divided by the result determined via manual calculations. (Fraction expressed as a percentage)
Fidelity of protocol implementation	Accuracy of the paths recommended by the simulation's implementation of each routing protocol. This MOE is calculated by manually tracing the path recommended by each protocol, and comparing the ordered list of nodes visited with the path generated by the simulation's implementation of the same routing protocol. The fraction of nodes in the simulated path that match the order and ID of the manually identified path (out of the total number of nodes in the path) represents fidelity
Stochastic fidelity—protocol percent error	The average standard deviation error seen in a stochastic result for the metric of protocol percent error for a given set of inputs, relative to the deterministic result for those same inputs
Stochastic fidelity—transmission time	The average standard deviation error seen in a stochastic result for the metric of transmission time for a given set of inputs, relative to the deterministic result for those same inputs
Utility for trade-off analysis	The number of metrics produced by the simulation that can be used in the trade-off analysis, divided by the number of parameters needed to perform the trade-off analysis (i.e. evaluating if the simulation can support user needs)

the most common observed path for each routing protocol, compared against the average network state across all iterations. If the most common recommended path matched the expected path then there was said to be no deviation.

Both the deterministic and stochastic components of the simulation's calculation functions were verified and validated. The deterministic mode results were expected

Table 3 Part of the set of test cases used to exercise the deterministic mode of the DTN simulation

Test case	Max satellite position (km)	Transmitter position (km)	Min. link quality	Max. link quality
1	100	100	0	1
2	100	1,000,000	0	1
3	1,000,000	100	0	1
4	1,000,000	1,000,000	0	1
5	100	100	0.4	0.5
6	100	1,000,000	0.4	0.5
7	1,000,000	100	0.4	0.5
8	1,000,000	1,000,000	0.4	0.5

Table 4 Part of the set of verification test cases used to exercise the stochastic mode of the DTN simulation

Test case	Max satellite position (km)	Transmitter position (km)	Min. link quality	Max. link quality	Monte Carlo iterations
1	100	100	0	1	1000
2	100	1,000,000	0	1	1000
3	1,000,000	100	0	1	1000
4	1,000,000	1,000,000	0	1	1000
5	100	100	0.4	0.5	1000
6	100	1,000,000	0.4	0.5	1000
7	1,000,000	100	0.4	0.5	1000
8	1,000,000	1,000,000	0.4	0.5	1000
9	1,000,000	1,000,000	0	1	500
10	1,000,000	1,000,000	0	1	10

to match the manually calculated values exactly (or to be within 0.1% of the manually-calculated values in the case of rounded values). In the Monte Carlo results, the manually-derived values were compared to the steady-state Monte Carlo values (after running the simulation for 1000 iterations per test case) to see if they stabilized within ±5% of the predicted value.

Testing revealed that the final version of the simulation successfully passed all tests without issue and within acceptable time limits, as evinced by the results shown in Tables 5 and 6.

Table 5 Some of results obtained after black-box testing the deterministic components of the DTN simulation, showing T (transmission time) and E (percent error). N represents no observed deviation; the tilde symbol (~) indicates an observed result that is identical to the expected result

	Expected T/observed T/deviation			Expected E/observed E/deviation			Expected path/observed path/deviation		
Test case	Bundle protocol	Distance Dijkstra	Quality Dijkstra	Bundle protocol	Distance Dijkstra	Quality Dijkstra	Bundle Protocol	Distance Dijkstra	Quality Dijkstra
1	3.7e−7/~/0	1.1e−7/~/0	1.3e−7/~/0	45/~/0	12/~/0	23/~/0	1,3,11,12/~/N	1,6,12,/~/	1,9,12/~/
2	2.8e−3/~/0	1.3e−3/~/0	2.4e−3/~/0	23/~/0	48/~/0	29/~/0	11,6,10,8,3,12/~/N	1,8,12/~/	10,6,12/~/N
3	1.4e−3/~/0	1.1e−3/~/0	1.4e−3/~/0	7/~/0	47/~/0	3/~/0	2,6,3,1/~/N	1,5,3,12/~/N	1,5,3,12/~/N
4	1.8e−3/~/0	1.4e−4/~/0	1.8e−3/~/0	9/~/0	57/~/0	4/~/0	1,11,3,12/~/N	1,3,12/~/	1,10,2,12/~/N
5	1.5e−3/~/0	1.2e−3/~/0	1.5e−3/~/0	19/~/0	22/~/0	19/~/0	1,2,12/~/N	1,6,12/~/N	1,3,12/~/N
6	1.7e−4/~/0	1.1e−4/~/0	1.7e−4/~/0	62/~/0	53/~/0	8/~/0	1,4,8,6,12/~/N	1,9,12/~/	1,11,12/~/ N
7	3.6e−7/~/0	1.2e−7/~/0	1.4e−7/~/0	12/~/0	51/~/0	3/~/0	1,8,3,10,4,12/~/N	1,11,12/~/ N	1,2.12/~/
8	2.1e−3/~/0	1.7e−3/~/0	1.9e−3/~/0	56/~/0	34/~/0	34/~/0	1,3,11,2,12/~/N	1,5,12/~/	1,5,12/~/N

Table 6 Some of results obtained after black-box testing the stochastic components of the DTN simulation, showing T (transmission time) and E (percent error). N represents no observed deviation; the tilde symbol (~) indicates an observed result that is identical to the expected result

	Expected T/observed T/deviation			Expected E/observed E/deviation			Expected path/observed path/deviation		
Test case	Bundle protocol	Distance Dijkstra	Quality Dijkstra	Bundle protocol	Distance Dijkstra	Quality Dijkstra	Bundle protocol	Distance Dijkstra	Quality Dijkstra
1	3.8e−07/~/1	1.3e−07/~/2	1.4e−07/~/3	34/~/2	12/~/3	28/~/2	1,4,10,12/~/N	1,5,12,/~/N	1,10,12/~/N
2	2.8e−03/~/1	1.2e−03/~/1	2.1e−03/~/1	23/~/3	48/~/2	29/~/2	10,6,2,3,12/~/N	1,7,12/~/N	10,4,12/~/N
3	1.9e−03/~/1	1.0e−03/~/2	1.3e−03/~/1	6/~/2	51/~/1	2/~/0	3,6,2,1/~/N	1,4,11,12/~/N	1,10,7,12/~/N
4	5.5e−04/~/1	9.6e−05/~/1	3.1e−04/~/1	59/~/3	84/~/4	36/~/2	1,10,2,12/~/N	1,9,12/~/N	1,3,12/~/N
5	1.6e−03/~/1	1.3e−03/~/1	1.7e−03/~/2	19/~/2	22/~/0	19/~/1	1,3,9,11,8,12/~/N	1,11,12/~/N	1,4,12/~/N
6	1.6e−04/~/1	1.1e−04/~/1	1.6e−04/~/1	62/~/4	53/~/4	8/~/2	1,3,9,11,8,4,2,7,12/~/N	1,7,12/~/N	1,4,12/~/N
7	3.7e−07/~/1	1.0e−07/~/2	1.3e−07/~/1	12/~/3	51/~/2	3/~/3	1,3,11,2,7,12/~/N	1,3,2 12/~/N	1,2,8,0.12/~/N
8	2.8e−03/~/1	1.2e−03/~/1	1.7e−03/~/1	56/~/4	34/~/4	34/~/3	1,11,3,,9,8,12/~/N	1,2,12/~/N	1,2,12/~/N
9	5.5e−04/~/2	9.6e−05/~/2	3.0e−04/~/2	56.6/~/1	80/~/4	38/~/4	1,10,2,12/~/N	1,9,12/~/N	1,3,12/~/N
10	5.2e−4/5.5e−4/12	4.8e−5/4.5e−5/17	1.6e−04/2.1e−4/21	56/40/16	84/90/14	40/36/4	1,3,11,2,12/~/N	1,7,12/~/N	1,4,12/~/N

7 Analysis

The results for a single run of the Monte Carlo simulation are summarized in Table 7. As seen from the results in the table, bundle protocol performed moderately well in terms of percent error but was the worst in terms of transmission time, while distance-based Dijkstra performed the best in terms of transmission time (with minimal variation as well) despite being the worst in terms of percent error. Quality-based Dijkstra, however, outpaced both the other two design options in terms of percent error and was better than bundle protocol, though not distance-based Dijkstra, in terms of transmission time.

Figures 8, 9 and 10 show the most frequent route taken by packets following bundle protocol, distance-based Dijkstra, and signal quality-based Dijkstra respectively for a given network. As is evident from the graphics, the MATLAB implementations of all three protocols behaved in the manner expected, which provides confidence that they were implemented correctly.

Since these are results for a single network, however, they do not reflect the variability in performance seen for each of the design options across different network configurations. Table 8 shows the performance of each of the three design options across five separate runs of the Monte Carlo simulation (and five different networks). The performance of the three routing protocols relative to each other remains consistent, with signal quality-based Dijkstra showing the best results for percent error, distance-based Dijkstra performing the best in transmission speed, and bundle protocol showing moderate results for both data integrity and the worst transmission time.

Notably, distance-based Dijkstra still performs worse than the current state-of-the-art methodology, bundle protocol, in terms of percent error. The other alternative protocol being evaluated, signal quality-based Dijkstra, performs better than bundle protocol across both metrics.

Table 9 shows the results of the t-test, confirming that all of the differences observed in Table 8 between the different design options are in fact statistically significant (i.e. that choosing one option over another would result in a substantial difference in the metric) for both percent error and transmission time. Thus, performing a MAVF analysis is reasonable since it has been established that the choice of design option does have a statistically significant impact on the metrics of interest.

Table 7 Summary of the results from a single run of the Monte Carlo simulation, showing the mean, standard deviation, and standard error of the mean for the two metrics of interest

Method	Percent error (%)	Transmission time (h)		
	Mean	Mean	Standard deviation	Standard error
Bundle protocol	57.6	2.703	0.994	0.0444
Distance-based Dijkstra	77.8	1.1882	0.000129	5.759e−06
Quality-based Dijkstra	36.4	1.926	0.970	0.0434

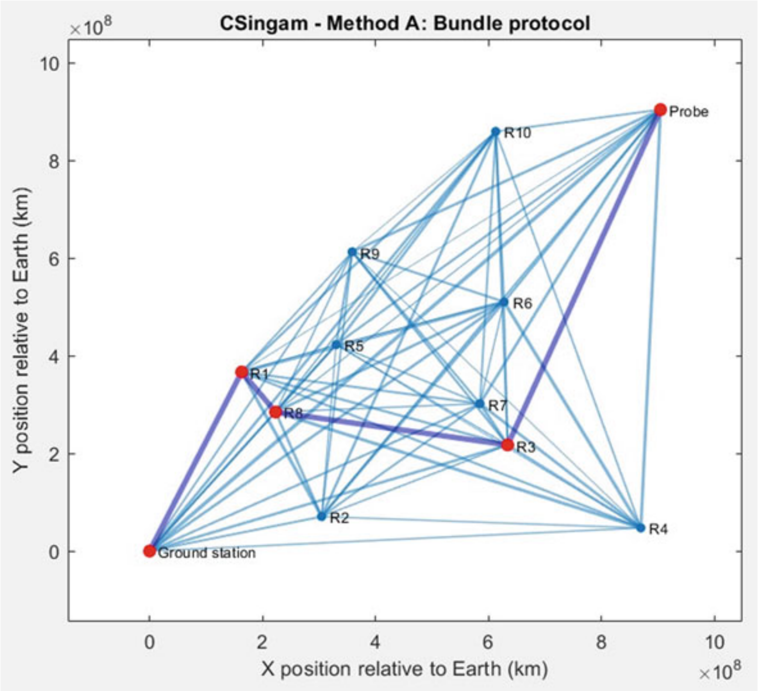


Fig. 8 Most frequent path taken by packets using bundle protocol for the network associated with the results in Table 2

Based on the results seen in Tables 7 and 8, and taking the bundle protocol results as a benchmark for performance (what with it being the current preferred routing protocol for DTN contexts), it is almost unnecessary to perform a MAVF analysis since signal quality-based Dijkstra outperforms bundle protocol across both metrics whereas the other methodology, distance-based Dijkstra, only outperforms bundle protocol in one metric (transmit time) and in fact performs significantly (as shown in Table 9) worse than baseline with regards to percent error.

Nonetheless, the MAVF rankings are provided in Table 10 to provide clear, unequivocal rankings of each design option.

The MAVF results show quality-based Dijkstra to be the best option, with a MAVF value that is over three times higher than the 2nd best option (bundle protocol). Distance-based Dijkstra ranks the worst despite having the best value for transmit time.

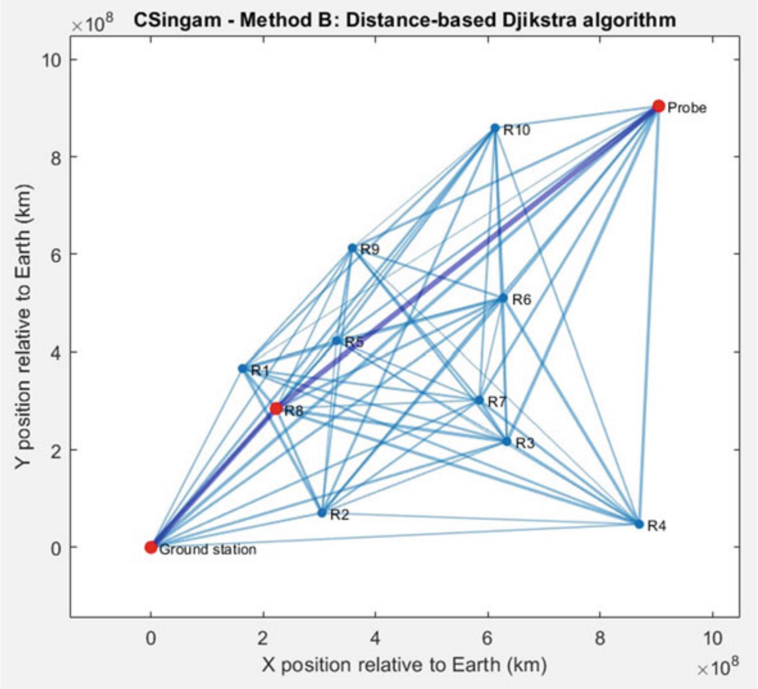


Fig. 9 Most frequent path taken by packets using distance-based Dijkstra for the network associated with the Table 2 results

8 Recommendations

Since the MAVF results are meant to serve as a means of quantifying the relative practical value of the design options, it is also worth noting the methodologies which are not worth implementing (i.e. that performed worse than the baseline). Distance-based Dijkstra performed worse than the baseline in a mission-critical metric (percent error), effectively eliminating any value the protocol's short transmission time might have had—after all, a rapid transmission has little utility if it risks compromising mission success significantly more than is considered standard.

Thus, it would be more accurate to assign an effective value of 0 to for the metric of transmission time for the distance-based Dijkstra methodology; this would yield the corrected MAVF table seen in Table 11.

The recommended course of action is to use signal-based Dijkstra as the routing protocol for space-based DTN applications. In addition to being the best option to minimize transmission error, it also performs moderately well in terms of transmission time and outperforms the current standard for routing protocols, bundle protocol, across both metrics. The Welch's t-test results indicated that switching a network over from using either of the two other design options to using signal quality-based Dijkstra would yield a statistically significant change in metrics, most notably an

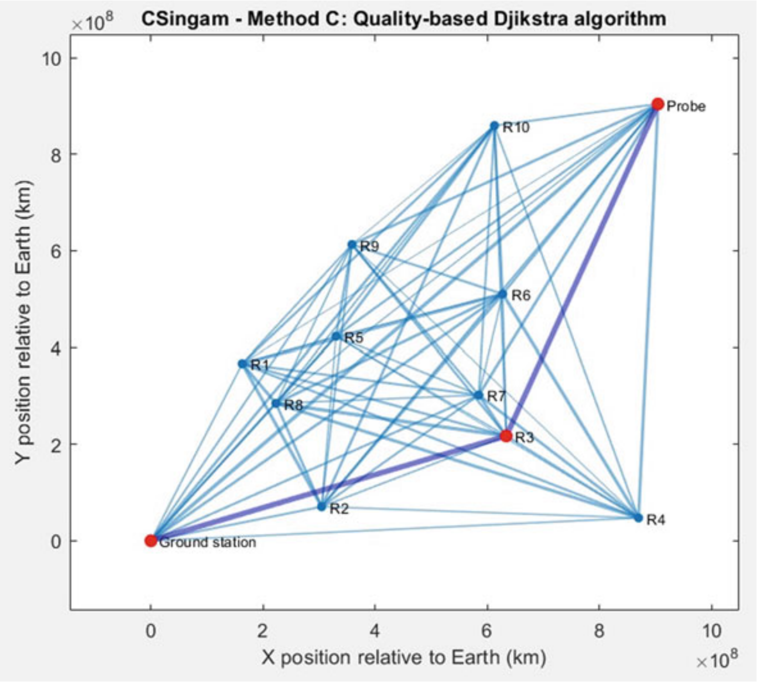


Fig. 10 Most frequent path taken by packets using quality-based Dijkstra for the network associated with the Table 2 results

Table 8 Summary of the results from five runs of the Monte Carlo simulation, showing the mean, standard deviation, and standard error of the mean for the two metrics of interest

Method	Percent error (%)			Transmission time (h)		
	Mean	Standard deviation	Standard error	Mean	Standard deviation	Standard error
Bundle protocol	56.440	7.410	3.314	3.120	0.684	0.306
Distance-based Dijkstra	64.200	4.864	2.175	1.189	0.004	0.002
Quality-based Dijkstra	41.040	4.498	2.011	1.820	0.280	0.125

increase in data integrity on receipt (the more critical of the two parameters). Given that routing protocols are implemented in networks via software, and that software updates can be readily pushed remotely to satellites (by virtue of being communication instruments in and of themselves), it is feasible to implement a change in routing protocols without any hardware modifications to the system.

Table 9 Welch's t-test results comparing each of the design options with each of the other design options for both percent error and transmission time. Green cells indicate that the calculated p-value meets the standard for statistical significance ($p < 0.05$)

Welch t-test results: percent error				Welch t-test results: transmission time			
Vs	Bundle protocol	Distance-based Dijkstra	Quality-based Dijkstra	Vs	Bundle protocol	Distance-based Dijkstra	Quality-based Dijkstra
Bundle protocol	–	0.0914	0.0102	Bundle protocol	–	0.00170	0.00500
Distance-based Dijkstra	0.0914	–	8.758E−05	Distance-based Dijkstra	0.00170	–	0.00437
Quality-based Dijkstra	0.0102	8.758E-05	–	Quality-based Dijkstra	0.00500	0.00437	–

Table 10 MAVF results for each of the three design options based on the metric means from the 5-run dataset

MAVF analysis			
Design option	Value based on percent error	Value based on transmit time	MAVF
Bundle protocol	0.33506	0	0.279217
Distance Dijkstra	0	1	0.166667
Quality Dijkstra	1	0.672774	0.945462

Table 11 The MAVF analysis results, with the values corrected for practicality

MAVF analysis			
Design option	Value based on percent error	Value based on transmit time	MAVF
Bundle protocol	0.33506	0	0.279217
Distance Dijkstra	0	0	0
Quality Dijkstra	1	0.672774	0.945462

9 Conclusions

The area of signal routing for disruption-tolerant networks is an area of active research and burgeoning interest, particularly with the varied applications for both DTNs in general and optimized signal routing approaches in particular. Given the ubiquity of communication systems in globalized society and the inevitability of unforeseen circumstances, absent a surprise confirmation of the existence of Laplace's demon, it is to be expected that existing interest in transferring information efficiently in adverse environments and variable conditions will continue into the foreseeable future as well.

The proposed signal quality-based Dijkstra routing method has been shown to be a methodology that performs well across both DTNs and standard-condition (continuous connectivity) scenarios, making it an ideal candidate for implementation across a variety of applications. As a methodology that has been shown to surpass the performance of existing standards, and that demonstrates benefits in terms of key metrics, security against DoS attacks, and from a network architecture standpoint—including improving network security posture, minimizing individual user risk, and increasing resiliency against sub-optimal traffic scenarios—it offers great promise for use with future DTN implementations. It is anticipated that the methodology can be refined further in future work so as to provide even greater flexibility and performance across any number of different DTN-relevant contexts, as well as greater security against hostile actor takeover.

References

1. McMahon A, Farrell S (2009) Delay-and disruption-tolerant networking. IEEE Internet Comput 13(6):82–87
2. Gertz M, Csaba G (2002) Monitoring mission critical data for integrity and availability. In: Working conference on integrity and internal control in information systems. Springer, Boston, MA, pp 189–201
3. Caini C, Cruickshank H, Farrell S, Marchese M (2011) Delay-and disruption-tolerant networking (DTN): an alternative solution for future satellite networking applications. Proc IEEE 99(11):1980–1997
4. Matson DL, Spilker LJ, Lebreton JP (2003) The Cassini/Huygens mission to the Saturnian system. In: The Cassini-Huygens mission. Springer, Dordrecht, pp 1–58
5. CCSDS Secretariat (2010) Rationale, scenarios, and requirements for DTN in space. CCSDS, Washington, D.C.
6. Ibid
7. NASA (2009) NASA's mission operations and communications services. AO NNH09ZDA007O, January 2009. https://deepspace.jpl.nasa.gov/files/6_NASA_MOCS_2014_10_01_14.pdf
8. Dyer JS, Sarin RK (1979) Measurable multiattribute value functions. Oper Res 27(4):810–822
9. Parnell GS, Trainor TE (2009) 2.3. 1 using the swing weight matrix to weight multiple objectives. INCOSE Int Symp 19(1):283–298
10. NCSS. Two sample T-tests allowing unequal variance. https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/PASS/Two-Sample_T-Tests_Allowing_Unequal_Variance.pdf
11. MathWorks. "digraph: Graph with directed edges" (n.d.). https://www.mathworks.com/help/matlab/ref/digraph.html
12. Ulybyshev Y (1998) Long-term formation keeping of satellite constellation using linear-quadratic controller. J Guid Control Dyn 21(1):109–115
13. Zhou J, Hu Q, Friswell MI (2013) Decentralized finite time attitude synchronization control of satellite formation flying. J Guid Control Dyn 36(1):185–195
14. Chang HS, Kim BW, Lee CG, Min SL, Choi Y, Yang HS, Kim CS et al (1998) FSA-based link assignment and routing in low-earth orbit satellite networks. IEEE Trans Veh Technol 47(3):1037–1048
15. Casio. "Keisan Calculator" (n.d.) https://keisan.casio.com/exec/system/1180573226
16. Paxton P, Curran PJ, Bollen KA, Kirby J, Chen F (2001) Monte Carlo experiments: design and implementation. Struct Equ Model 8(2):287–312
17. Buckley J, James I (1979) Linear regression with censored data. Biometrika 66(3):429–436

A Mobile and Compact Control Center for Quick Decentral Satellite Access

Stefan A. Gärtner, Norbert Harder, Jens H. Hartung, Markus Hobsch, and Martin Weigel

Abstract Compact and inexpensive Earth observation satellites in low Earth orbit are now routinely developed by universities, "New Space" businesses, and space agencies. They enable new opportunities for fast turnaround times of imaging data takes, which is e.g. particularly important for disaster response. For this kind of satellites and the missions enabled by them a ground system exhibiting the same characteristics, namely being compact and mobile, yet inexpensive and flexible, is desired. We present DLR's approach for the provisioning of a ground segment fit for these kinds of "Responsive Space" missions. The objective of this project consists of the engineering, delivery, and demonstration of a compact and yet complete Mission Operations System, runnable on commodity mobile hardware, enabling fully automated workflow-driven operations of alike missions from anywhere in the world with access to a ground station or ground station network. Just as disasters strike suddenly, the ground segment needs to be set up and spun up in a timely manner. This leads to the requirement of being able to quickly roll out the system on new hardware, possibly even several of these systems in parallel. Our paper provides insight on how we perform the automatic deployment and provisioning. Because the system is supposed to be decentralized and used in the field, particular challenges need to be overcome resulting from the lack of all of the infrastructure typically present in conventional control centers, such as network connectivity. An embedded Flight Dynamics System is taking care of automated orbit determination and related event generation to support the mission needs and maneuver capabilities. Special effort is made to cope with auxiliary data that may not be updated on a regular basis in a closed mission environment. The feasibility of the concept is demonstrated by a first system deployment as drop-in replacement for the existing conventional

S. A. Gärtner (✉) · N. Harder · J. H. Hartung · M. Hobsch · M. Weigel
DLR Oberpfaffenhofen, German Space Operations Center GSOC, 82234 Weßling, Germany
e-mail: stefan.gaertner@dlr.de

S. A. Gärtner · N. Harder · J. H. Hartung · M. Hobsch
Mission Technology Department, DLR German Aerospace Center, Münchener Straße 20, 82234 Weßling, Germany

M. Weigel
Space Flight Technology Department, DLR German Aerospace Center, Münchener Straße 20, 82234 Weßling, Germany

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_19

Mission Operations System for DLR's BIROS satellite at the GSOC control center. A second demonstration campaign is performed from a remote location without access to control center infrastructure.

Abbreviations

CCSDS	Consultative Committee for Space Data Systems
CI/CD	Continuous Integration/Continuous Deployment
COTS	Commercial Off-The-Shelf
DLR	German Aerospace Center/Deutsches Zentrum für Luft- und Raumfahrt
ERP	Earth Rotation Parameter
ESA	European Space Agency
FDS	Flight Dynamics System
GDS	Ground Data System
GECCOS	GSOC Enhanced Command and Control System for Operating Spacecraft
GNSS	Global Navigation Satellite System
GSOC	German Space Operations Center
ICRF	International Celestial Reference Frame
LEO	Low Earth Orbit
MCS	Monitoring and Control System
MIB	Mission Information Base
MPS	Mission Planning System
NCTRS	Network Control and TM/TC Router System
NTP	Network Time Protocol
OD	Orbit Determination
ProToS	Procedure Tool Suite
SCOS	Satellite Control and Operation System
SLE	Space Link Extension
SSB	SLE Switch Board
SSF	Saved Stack File
TC	Telecommand
TLE	Two-Line Elements
TM	Telemetry
V3C	"Verlegefähiges" (mobile) Compact Control Center
VM	Virtual Machine
XML	Extensible Markup Language

1 Introduction

One of the exciting new topics in space operations is the emerging market of small, inexpensive, flexible to use satellites which are deployed into low-Earth orbits. Ground segments for these missions should reflect this new paradigm of "Responsive Space" by being equally compact, inexpensive and fast to roll out, easily maintainable, and providing straight-forward operability for small staff. Ideally, such a ground segment can be operated right where the acquired payload data is needed most, for example in disaster areas of the world where quickly unfolding events demand timely access to information.

In this paper we show the design, implementation, and utilization of such a Mission Operations System and how it lends itself to a *compact* and *mobile* ground segment.

1.1 Use Cases

Typically, ground segments for satellite operations are set up in mission control centers, occupy a dedicated control room and are closely tied to the infrastructure provided by those centers. Recent developments of virtualizing control center infrastructure [1] allow flexible reallocation of control rooms and resources, but the Mission Operations Systems running on top of this virtualized infrastructure still do not lend themselves to usage outside of the controlled environment of a control center. Being able to move a Mission Operations System out of a stationary control center enables the realization of several use cases identified in the following.

1.1.1 Disaster Response

Disasters like wild fires or flooding often strike suddenly and violently. Situational awareness is of paramount importance for organization of rescue teams, early detection of hazardous developments and mitigating risks. Often, communication and other infrastructure in affected areas is damaged or destroyed. Giving a single operator the ability to flexibly acquire imaging data right where it is needed allows to quickly adapt to the dynamically evolving situation. A mobile and compact system enabling such an approach can be one of many building blocks for successful disaster response.

1.1.2 Security and Defense

Being able to quickly identify and react to threats is made possible by providing reconnaissance data in the field. One or more mobile and compact control centers can be used to receive and display imaging data where needed. It is possible to use such a system in conjunction with a classical control center that centrally plans and commands data takes. In light of resilience and the possible loss of the primary or backup control centers such mobile systems can be held ready or deployed quickly in order to provide distributed commanding capability. These systems fit well into the context of "Responsive Space" missions as they provide the ground segment counterpart to responsive launch and space capabilities.

1.1.3 Scientific Missions

Scientific missions for which autonomous and distributed commanding capabilities are viable can make use of a compact and mobile ground segment that provides principal investigators a high amount of direct control over their satellite payloads. Such a ground segment can be set up directly at a research institute or other appropriate location, possibly relaying antenna access from a main control center.

1.1.4 Education

Universities building and operating their own satellites can benefit from the educational value that such a compact system provides. Not only is the system operable in-house by a small number of students, but students also get the possibility to learn the basic concepts of spacecraft operations with an easy-to-use, easy-to-expand and yet complete system. By being able to recreate the whole system from scratch in a matter of minutes barriers for experimentation are lowered because the system can be quickly brought into a state known to be usable.

1.1.5 Limits of a Mobile and Compact Control Center

It is to note that a mobile and compact control center is not equally well suited for all types of missions or operational tasks: The more central coordination is needed and the fewer routine tasks are performed, the less applicable is such a system. Large satellites with abundant operational modes and many, possibly quite different, payloads are best operated from a dedicated classical control center. However, it might still make sense to employ a mobile control center for certain tasks or scenarios and thus combine the two approaches.

1.2 Design Drivers

The following high-level design drivers guide the development of the so-called V3C ("Verlegefähiges"[1] Compact Control Center) ground segment:

- The ground segment shall be movable between different places of operation.
- The ground segment shall operate independently, i.e. it shall not be deeply integrated into facilities and infrastructure and shall not rely on permanent Internet access.
- External interfaces (e.g. to ground stations) shall be standards-compliant if such standards exist.
- Copies and updates of the ground segment shall be able to be put into operation quickly.
- The ground segment shall be operated by one operator.
- The ground segment shall enable demonstration operations of the BIROS satellite's optical imaging payload.

1.3 Compactness and Mobility

Achieving *compactness* with respect to ground systems is made possible by the focus on specific mission types as outlined above: Satellites are single-purpose; therefore, no special operation modes need to be supported. Contingency handling is kept to a minimum, not only for ground but also for space segment contingencies. The remaining routine operations tasks are predisposed to automation. This in turn allows a workflow-driven operations approach that can be executed by a single operator on a single console. The drastically reduced number of consoles, the lack of redundancy and the single-purpose space segment allow the whole ground segment to be implemented on commodity hardware like a single laptop.

Compactness is necessary but not sufficient to achieve *mobility*. A ground segment is usually embedded into a larger infrastructure—the control center—that provides multi-mission services, ground station connectivity and facilities. The V3C ground segment is built from components derived from the GSOC multi-mission tool suite. As such, they are ideally tailored to each other and can be integrated into a self-contained assembly with as little external dependencies as possible. Some multi-mission services need to be stripped down and integrated. Ground station connectivity still needs to be provided externally, albeit the system is setup assuming a standardized antenna interface like CCSDS SLE [2]. This allows plugging the ground segment in any ground station network or single antenna implementing the standard.

[1] German expression for a system that is mobile in the sense that it can be deployed at different places.

1.4 Demonstration Campaigns

It should be noted that the ground segment is not tailored to a specific mission but set up in a way to be adaptable to satellites belonging to the class of small, single-purpose LEO space vehicles as described above. Nonetheless, the whole concept is demonstrated in two campaigns currently in preparation with an existing satellite operated by GSOC, namely BIROS [3]. With its specifications[2] BIROS fits the class of satellites targeted by this ground segment.

BIROS houses an optical imaging payload and we show the execution of an exemplary workflow by means of the following scenario: A list of imaging opportunities is presented to the operator after his or her request for imaging of a certain geographical area. Satellite maneuvers, telecommand and telemetry contacts using the connected ground station network are calculated, telecommands are generated and radiated to the spacecraft and their execution is monitored. Imaging data is downlinked, processed and presented to the operator. The whole workflow is performed on the single-laptop hardware and thus serves as demonstration of feasibility and compactness. As next step we show the mobility of the system by taking it off-site for performing the same workflow again, this time without the backing GSOC infrastructure.

The first demonstration is concerned with verifying the compactness of the system: The demonstration operations team performs above workflow from inside GSOC, using the existing network and ground station connections. For this demonstration the regular BIROS Mission Operations System is replaced with the V3C system, see Sect. 4 for details. The second demonstration is concerned with verifying the mobility of the system: The demonstration operations team performs above workflow from outside GSOC without being able to resort to its infrastructure or facilities. The current candidate for the remote demonstration site is an antenna at Weilheim ground station.

2 Ground Segment Overview

2.1 System Overview

As baseline for the ground segment and operations concept the BIROS project is used due to the BIROS satellite being the spacecraft targeted for the demonstration campaigns. This allows reuse of existing components with minimal adaptions to project-specifics as well as familiarity of the operators running the demonstration campaign. Because the existing BIROS system is still in place it is ensured that

[2] Polar sun-synchronous orbit, altitude 510 km, mass $\approx$ 130 kg, infrared and optical imaging payloads.

operations of the satellite can be resumed from this system at any time, especially during spacecraft or ground contingencies.

The system design shown in Fig. 1 is a starting point which evolves in accordance with the project's agile approach. Starting with the demonstrations a potentially usable Mission Operations System is available at any point in time. The following sections detail the composition of each subsystem and how they interface with each other. The concrete system deployment is described in Sect. 3 and integration into GSOC is shown in Sect. 4.1.

Development of the space segment or ground stations (GDS) is not part of this project. The first demonstration campaign uses V3C as plug-in replacement for the existing Mission Operations System. Therefore, some of the existing interfaces have to be served (see Sect. 2.5 for details). With respect to the overall ground segment design this means that a bespoke protocol (NCTRS) is used for ground station connectivity, which will later be replaced with a standards-compliant protocol (SLE) by integrating one of the GDS components (labeled SSB in Fig. 1 and indicated by the dashed lines in Fig. 1) into the system.

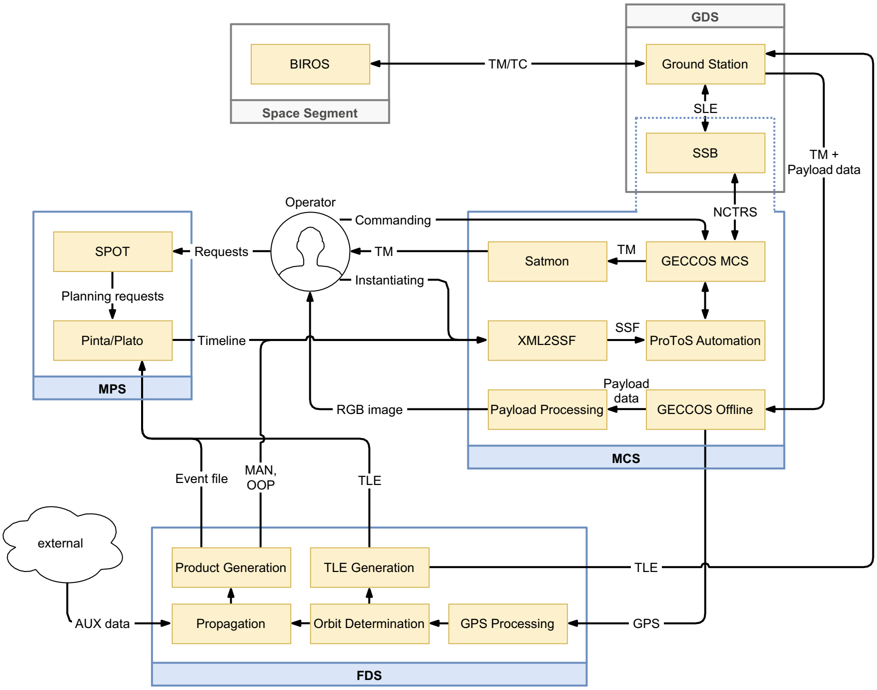


Fig. 1 Overview of the V3C ground segment system design

2.2 Monitoring and Control System

The Monitoring and Control System (MCS) is the subsystem that enables the collection, interpretation and archival of satellite telemetry (TM) as well as the preparation and release of telecommands (TC). It is closely coupled with other subsystems like the Ground Data System which forwards telemetry and telecommands to the ground stations and the Flight Dynamics Systems for which it extracts data necessary for orbit calculations. These orbit calculations are then in turn used for scheduling activities by the Mission Planning System. The Monitoring and Control System consists of the following components:

- GECCOS as real-time Monitoring and Control software and recorded TM (i.e. non-real time, "offline") processing system,
- Satmon server and client as display system,
- ProToS as procedure creation, instantiation and automation tool,
- XML2SSF Merger for merging flight procedures with parameter values from several sources,
- Mission Information Base as the common source for TM and TC definitions.

Figure 2 shows a schematic overview of the Monitoring and Control system and its interfaces. Interfaces to subsystems external to MCS are shown with red arrows.

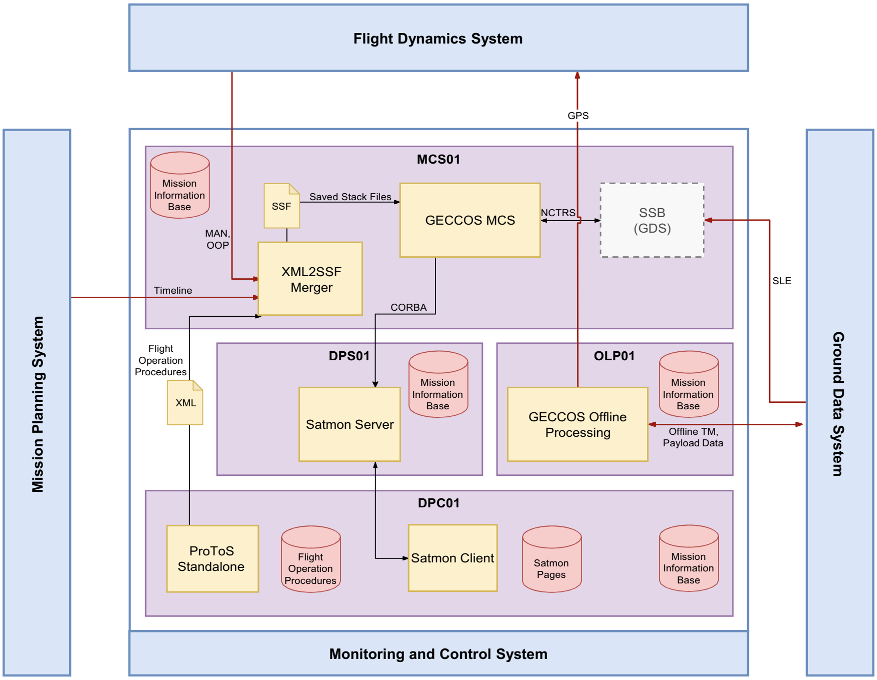


Fig. 2 Overview of the Monitoring and Control System

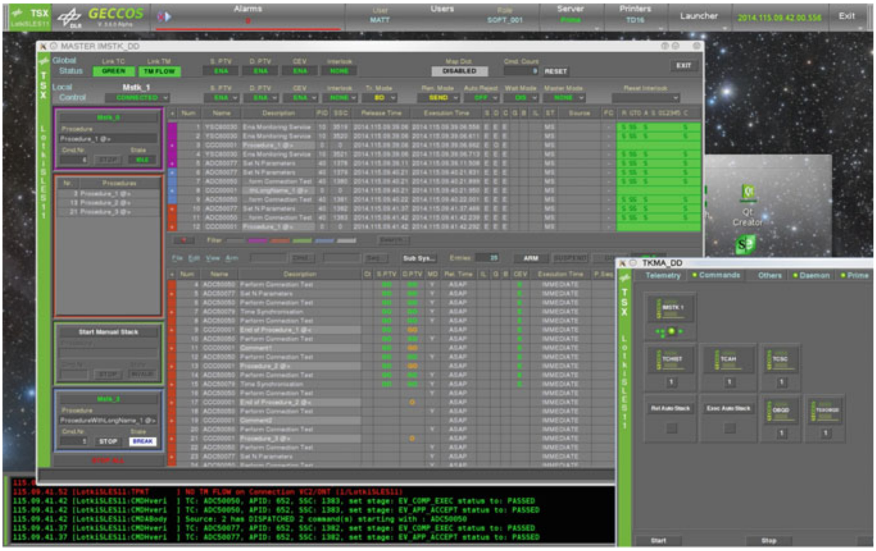
The SSB component is part of the Ground Data System and is located there for the first demonstration. It will be moved into the MCS as shown for the second demonstration, thereby paving the way to a standards-compliant antenna interface.

2.2.1 GECCOS

GECCOS [4] is GSOC's custom Monitoring and Control software and has been branched off from ESA's SCOS 2000 which is widely used by ESA for the monitoring and control of ECSS-E-70-41 [5] compatible missions. GECCOS is capable to read telemetry and send telecommands using the bespoke NCTRS interface or can ingest telemetry from files. It therefore serves as both, a real-time MCS and an Offline Processing System. The bespoke NCTRS protocol is translated to the standard SLE protocol for communication with ground stations by means of a component labeled SSB (SLE Switch Board). As shown in Fig. 2 GECCOS interfaces with nearly all other MCS components and several other subsystems. Figure 3 shows the Manual Stack, which is one of various sub-applications of GECCOS responsible for the preparation and release of telecommands.

2.2.2 Satmon

Satmon [6] is the main tool used for the visualization of telemetry parameters. The Satmon client is able to display the incoming telemetry to users in real-time and



```
115.09.41.52 [LotkiSLES11:TPKT      ] NO TM FLOW on Connection VC2/ONT (1/LotkiSLES11)
115.09.41.42 [LotkiSLES11:CMDHveri  ] TC: ADC50050, APID: 652, SSC: 1383, set stage: EV_COMP_EXEC status to: PASSED
115.09.41.42 [LotkiSLES11:CMDHveri  ] TC: ADC50050, APID: 652, SSC: 1383, set stage: EV_APP_ACCEPT status to: PASSED
115.09.41.42 [LotkiSLES11:CMDABody  ] Source: 2 has DISPATCHED 2 command(s) starting with : ADC50050
115.09.41.37 [LotkiSLES11:CMDHveri  ] TC: ADC50077, APID: 652, SSC: 1382, set stage: EV_COMP_EXEC status to: PASSED
115.09.41.37 [LotkiSLES11:CMDHveri  ] TC: ADC50077, APID: 652, SSC: 1382, set stage: EV_APP_ACCEPT status to: PASSED
```

Fig. 3 GECCOS screenshot showing the Manual Stack

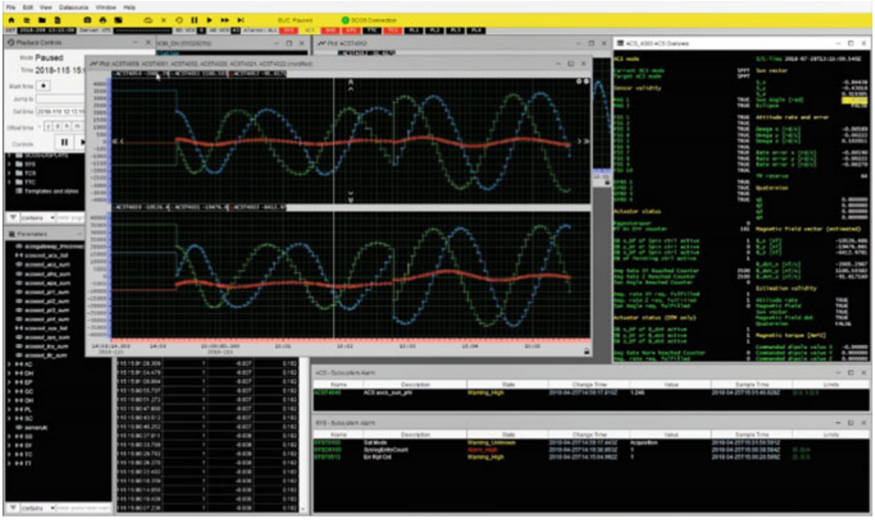


Fig. 4 Satmon screenshot demonstrating different telemetry views and data plots

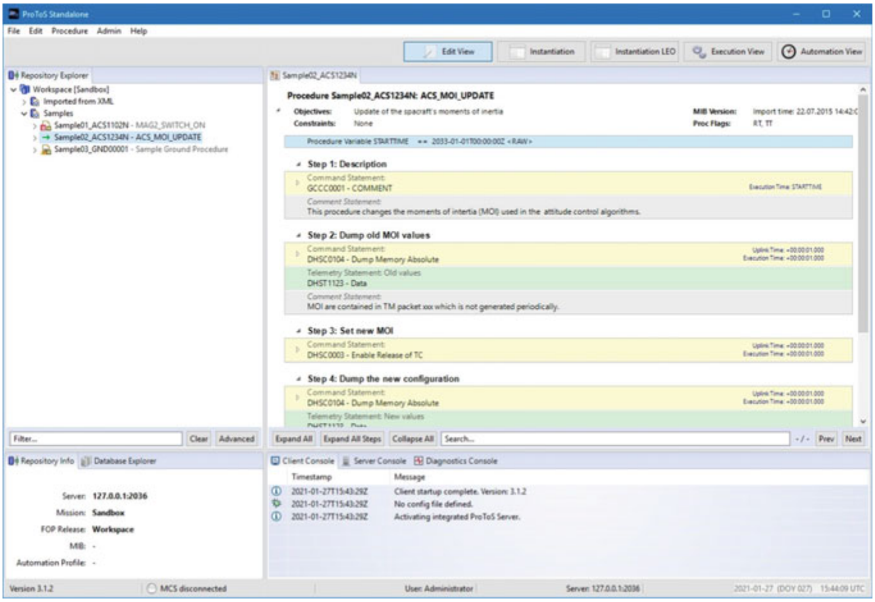
provides fast access to the history of received parameters. For this task it provides many tools to display telemetry such as lists, aggregated parameter pages, purpose-built overview pages, procedure pages, interactive plots as well as reactive flow charts. An integrated editor enables the user to customize and create telemetry overviews. Figure 4 shows some of the possible displays. A highly efficient telemetry database optimized for high storage density and low retrieval latency backs the Satmon client on the server side.

2.2.3 ProToS

The Procedure Tool Suite (ProToS) [7] is a software solution developed at GSOC. Its purpose is to support the creation and execution of satellite test and flight operations procedures and to provide an automation framework for complex operational scenarios. A screenshot of the tool is depicted in Fig. 5.

For the demonstration campaigns ProToS accesses the flight operations procedure database of the BIROS flight operations system and enables the operator to instantiate editable command parameters of these procedures before handing them off to the XML2SSF Merger. ProToS is also used for authoring and validating new procedures supporting the demonstration scenarios.

Instantiated procedures are executed manually by the operator for the demonstration campaigns, but it is planned to leverage both ProToS’ execution and automation framework later in the project to better support workflow-driven operations: The ProToS execution engine connects directly to GECCOS and supervises procedure execution. This enables execution of branching procedures, which is not possible in



Procedure Sample02_ACS1234N: ACS_MOI_UPDATE

Objectives: Update of the spacraft's moments of inertia
Constraints: None
MIB Version: Import time: 22.07.2015 14:42:0
Proc Flags: RT, TT

Procedure Variable STARTTIME == 2033-01-01T00:00:00Z <RAW>

Step 1: Description

Command Statement:
GCCC0001 - COMMENT
Execution Time: STARTTIME

Comment Statement:
This procedure changes the moments of intertia (MOI) used in the attitude control algorithms.

Step 2: Dump old MOI values

Command Statement:
DHSC0104 - Dump Memory Absolute
Uplink Time: +00:00:01.000
Execution Time: +00:00:01.000

Telemetry Statement: Old values
DHST1123 - Data

Comment Statement:
MOI are contained in TM packet xxx which is not generated periodically.

Step 3: Set new MOI

Command Statement:
DHSC0003 - Enable Release of TC
Uplink Time: +00:00:01.000
Execution Time: +00:00:01.000

Step 4: Dump the new configuration

Command Statement:
DHSC0104 - Dump Memory Absolute
Uplink Time: +00:00:01.000
Execution Time: +00:00:01.000

Telemetry Statement: New values

Server: **127.0.0.1:2036**
Mission: **Sandbox**
FOP Release: **Workspace**
MIB: -
Automation Profile: -

```
Timestamp              Message
2021-01-27T15:43:29Z   Client startup complete. Version: 3.1.2
2021-01-27T15:43:29Z   No config file defined.
2021-01-27T15:43:29Z   Activating integrated ProToS Server.
```

Version 3.1.2 | MCS disconnected | User: Administrator | Server: 127.0.0.1:2036 | 2021-01-27 (DOY 027) 15:44:09 UTC

Fig. 5 Screenshot of ProToS showing the tabular view of an example flight procedure

an automated way when using Saved Stack Files—the native GECCOS format. The scriptable automation engine can react to different events and trigger execution of procedures as needed [8]. The engineering of the automation requirements and their implementation is future work during the course of this project.

2.2.4 Other Components

XML2SSF Merger: The XML2SSF Merger merges flight procedure templates in XML format with parameter values in order to provide instantiated flight procedures in a format suitable for GECCOS (SSF). Parameter values are ingested from operators via ProToS, from the Mission Planning timeline, or from Flight Dynamics maneuver activities.

Mission Information Base: The Mission Information Base (MIB) is used by all of the tools above and is the central source for telemetry and telecommand definitions.

SLE Switchboard: The SLE Switchboard (SSB) is a software to support the interface with various types of ground station equipment. It acts as a protocol bridge between the GECCOS-bespoke NCTRS protocol and the standard SLE protocol supported by ground stations throughout the world supporting acquisition of telemetry and sending of telecommands.

2.3 Mission Planning System

The main purpose of the Mission Planning System (MPS) in context of this project is the generation of consistent, conflict-free timelines and sequences of flight operations procedures (FOPs) in order to command the payload and background sequence operations of the target spacecraft from all given input items and known constraints. In addition, MPS shall support the operator in the pre-planning and ordering process for acquisitions of the spacecraft imaging payload.

The main MPS task for the demonstrations is to adapt the already existing MPS components used for BIROS in the scope of the FireBird mission [9] to the needs of the V3C project. The main challenges here are the lack of a connection to the Internet, as available during a regular scientific mission like FireBird, and the usage of a different payload for the demonstration campaigns than the one used during the FireBird mission. The primary payload for the FireBird mission is a scanline infrared imaging device, whereas the demonstrations use the secondary payload optical camera that yields area images. Figure 6 gives an overview of the MPS components and interfaces and how they interact with the operator and the Flight Dynamics and Monitoring and Control subsystems. It can be seen that the MPS comprises two components which will be further described in the following. Ground station scheduling can either be an external component or integrated into MPS.

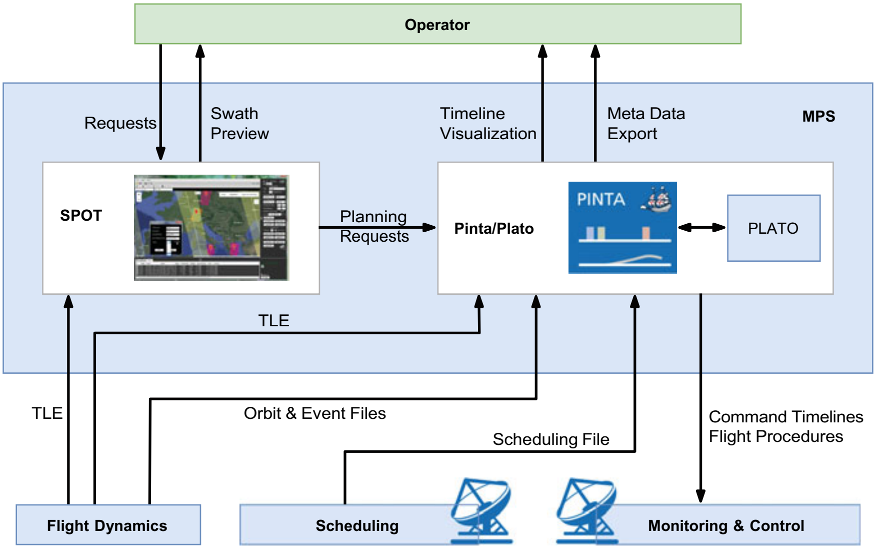
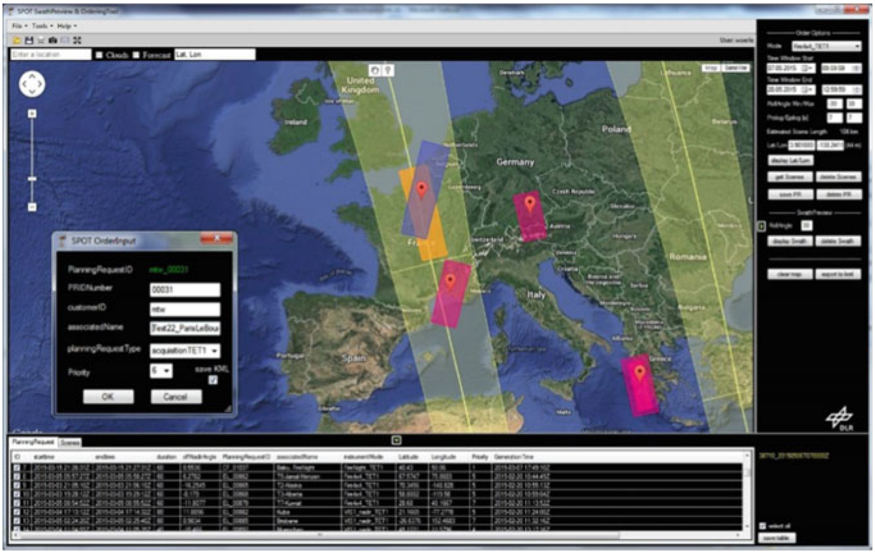


Fig. 6 Schematic view of the Mission Planning System components and interfaces



SPOT OrderInput

PlanningRequestID mtw_00031

PRIDNumber 00031

customerID mtw

associatedName Test22_ParisLeBou

planningRequestType acquisitionTET1

Priority 6

save KML ☑

Fig. 7 Snapshot of the Swath Preview and Ordering Tool (SPOT)

2.3.1 Swath Preview and Ordering Tool (SPOT)

The GSOC Swath Preview and Ordering Tool (SPOT) is used for the calculation and visualization of upcoming acquisition opportunities for areas of interest and for the generation of consistent planning requests for the spacecraft controlled by V3C on the one hand and the visualization of the spacecraft swaths for the pre-planning of campaigns on the other hand.

SPOT provides a graphical user interface using GSOC's SCOTA library and embedding Google Maps[3] or OpenStreetMap.[4] The maps are replaced with an offline solution for this project due to the potential lack of Internet connectivity. SPOT allows to calculate target visibilities based on the latest two-line elements from Flight Dynamics and to prepare consistent planning requests from the chosen target acquisition opportunities, which are to be sent to the core planning system and contain all necessary planning information. A snapshot of the current version and layout of the SPOT GUI can be seen in Fig. 7.

2.3.2 Pinta/Plato

The planning runs for the project are performed with this tool for semi-automated planning and scheduling and timeline export. It is based on GSOC's generic Program

[3] https://maps.google.com/.

[4] https://www.openstreetmap.org/.

for Interactive Timeline Analysis (PINTA) and the GSOC PLATO library and maintains the current planning model based on the latest inputs with project-specific, configurable algorithms and plug-ins. At the beginning of every planning run, Pinta/Plato collects all relevant information:

- the planning requests from the operator that have been created via SPOT,
- the information about scheduled ground station contacts—obtained through a central scheduling office if available or through automated or interactive user selection,
- the orbit-related information from Flight Dynamics, containing e.g. the timestamps of shadow events and ground station visibilities,
- the current two-line elements from Flight Dynamics.

By applying a dedicated combination of configurable planning algorithms, Pinta/Plato enables scheduling timeline entries of all necessary flight operations procedures to run the payload operations and background sequence tasks of the mission and generating conflict-free timelines. The execution timeline consisting of a sequence of flight operations procedures and their command parameters are handed over to the XML2SSF Merger. The operator has the possibility to check and modify the results of a planning run by viewing the graphical timeline representation provided by PINTA. A snapshot of the current version of the timeline view can be seen in Fig. 8.

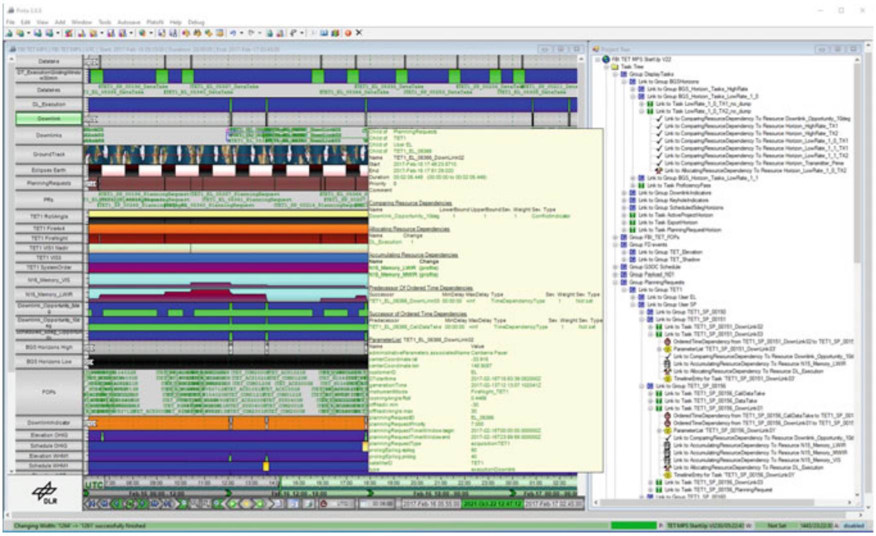


Fig. 8 Snapshot of the Program for Interactive Timeline Analysis (PINTA): timeline and resource plots, detailed activity information and a tabular view into the planning project structure

2.4 Flight Dynamics System

The Flight Dynamics System (FDS) is responsible for computing any information related to the satellite orbit position and attitude. A LEO satellite equipped with a GNSS receiver shall serve as baseline in the context of this project. Transponder ranging presents a second source of navigation measurements, that will be used during launch and early orbit phase and as fallback. In this basic form the FDS tasks include tracking data pre-processing and conversion, orbit determination (OD), and generation of orbit-related products like events and two-line elements (TLE) for observing ground stations. The orbit information also serves as essential input for planning of maneuver activities or Earth observation campaigns.

Depending on the specific satellite payload the FDS may further generate additional data products. The satellite payload also drives the requirements for product quality, like the ensured orbit determination and prediction accuracy. In an enclosed network environment, the FDS may not receive updates of auxiliary data for Earth rotation parameters and solar flux predictions. In the following, the maximum position errors are analyzed in the case of missing updates of the auxiliary data.

2.4.1 Solar Activity

The density models of the residual atmosphere require as input the solar radiation flux at 10.7 cm wavelength, the 90-day average value, as well as the Kp-index and ap-index of geomagnetic activity. Without current data on solar activity, mean values can be applied as typical values. The estimated drag coefficient will compensate for the average atmospheric drag the satellite encountered during the observation arc. In this way long periodic changes are covered, and in particular the 11-year solar cycle.

To validate this, the daily orbit determination runs of the BIROS satellite have been reprocessed with mean solar flux data. The first four years of BIROS operations coincide with minimum solar activity. For the most part the average flux values applied for orbit determination exceed the actual solar activity. This is compensated by lower estimates of the drag coefficient during this time period of low solar activity. In total 1353 orbit records from the first four years of BIROS operations are determined with constant mean values for solar activity, and further propagated for a prediction time of up to four days. The corresponding orbit records from Mission Operations are propagated for the same time periods using the daily prediction file on solar activity. The position difference from the 1353 ephemeris pairs allows for a statistical analysis of position errors. As can be seen from Fig. 9 the mean position errors in radial, along-track and cross-track direction mostly cancel out for the large number of OD cases. The 1σ error bounds are plotted with solid blue lines. After three days of orbit prediction the 1σ position error is about 4.15 km, mostly in along-track direction. Note that these statistical errors are derived from a period of low solar activity.

The daily prediction files from January 2001 till September 2020 are compared against the timeline of historical flux data. The standard deviation of the difference

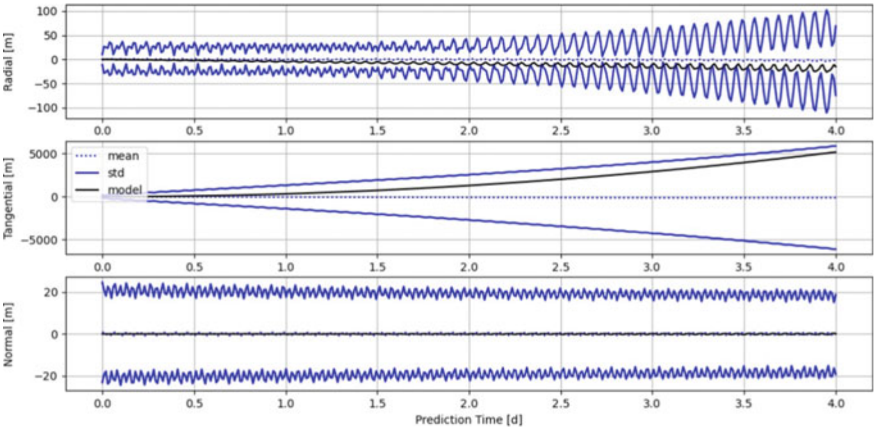


Fig. 9 Position differences from average solar flux data derived from past mission data and error model

between forecast and observation is evaluated depending on the prediction time and in time intervals of one day. The standard deviation corresponds to the 1σ error of the solar flux prediction. In 68% of the cases the actual value is within the 1σ interval, in 95% within the 2σ interval, in 99.7% within the 3σ interval, etc. The 2σ bound is applied for estimation of the maximum orbit errors, refer to Table 1. The expected position errors are to be estimated by means of a circular polar orbit. The orbit is propagated numerically over a period of three days with constant mean values of solar activity, and then over the same period with solar activity set to the highest expected solar activity (MEAN $+ 2 \cdot$ STD). A corresponding 2σ position error is computed from the difference of the two orbit predictions, e.g. in 95% of the cases a lower solar activity is to be expected, and consequently lower position errors, and in 5% of the cases the solar activity is higher with corresponding higher position errors than indicated. When assuming a higher solar activity the satellite experiences larger atmospheric drag. The position errors in flight direction grow fastest with an exponential increase, followed by the position errors in radial direction. Until the end of the prediction period of three days, there are only small differences in normal direction. The 3-dimensional position error after three days is about 60 km.

Table 1 Solar flux data applied for maximum error estimation (2σ errors)

	MEAN	STD			
		Day 1	Day 2	Day 3	Day 4
F10.7 flux	221.3 sfu	6.7 sfu	8.6 sfu	10.5 sfu	12.3 sfu
Average F10.7 flux	213.8 sfu	0.28 sfu	0.31 sfu	0.37 sfu	0.47 sfu
Kp-Index	4.92	1.45	1.41	1.38	1.38
ap-Index	39.67	10.3	11.1	11.4	11.4

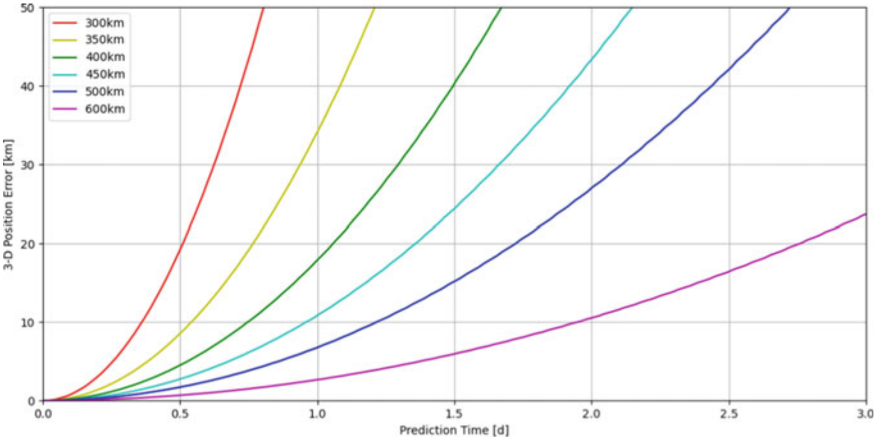


Fig. 10 2σ position errors at various orbit heights, due to missing forecasts of solar activity

Figure 10 illustrates the dependency of the maximum 3-D position error from orbit height. Depending on the required orbit accuracy, orbits created without daily forecasts of solar activity may be used only over a limited prediction time. Besides the strong dependence on altitude, the satellite's area-to-mass ratio and drag coefficient also have an influence. In this example a ratio of 100 kg m^{-2} and a drag coefficient of 1.5 are applied.

2.4.2 Earth Rotation Parameters and Leap Seconds

Missing updates of the Earth rotation parameters (ERP) lead to a faulty coordinate transformation between Earth-fixed frames and inertial reference frames. The maximum position errors are estimated without consideration of further orbit prediction errors.

The pole coordinates (x-pole, y-pole) describe the precise position of the Earth's rotation axis, which follows a circle with a radius angle of $2.91 \cdot 10^{-6}$ rad. The maximum position error scales with the distance to the center of the Earth. For example, it is about 20 m at an altitude of 500 km.

The time difference UT1–UTC defines the phase of Earth rotation with respect to the vernal equinox. A time error is translated into an angular error by the Earth rotation rate of $7.292\,115 \cdot 10^{-5}$ rad s^{-1}. A new leap second is typically introduced when the time difference is about $+0.5$ s or -0.5 s. Thus, the maximum position error shall be estimated for an unconsidered time difference UT1–UTC of half a second. The corresponding angular error is scaled with the distance to the center of the Earth, e.g. about 250 m for an orbit height of 500 km and a UT1–UTC error of 0.5 s. The transformation between Earth-fixed and inertial coordinate systems is needed several times. Station coordinates (transponder ranging) or on-board GNSS measurements are defined in Earth-fixed coordinates. Orbit elements are defined in an inertial coordinate system, and numerical orbit propagation for orbit determination and orbit prediction is performed in an inertial system. Therefore, the estimated error applies for all data products derived from inertial coordinate frames: TLE, osculating orbit elements, orbit ephemeris in inertial frame, derived close approach warnings, etc.

If data products are derived from orbit information in Earth-fixed frame, like orbit ephemeris, the transformation errors to and from inertial frame may cancel out to some extent. To demonstrate this, a single orbit determination run for the BIROS satellite from 2017-03-12 is re-run with and without ERP data. At that date the time difference UT1–UTC was approximately +0.5 s. For the OD run with missing ERP data, the time differences UT1–UTC and the pole coordinates are set to zero. The observation arc covers 24 h of on-board GNSS observations.

The position differences of the two solutions are compared in True-of-Date inertial coordinate frame and ICRF2000 Earth-fixed frame (International Celestial Reference Frame). A good fraction of the transformation errors cancels out for the Earth-fixed frame during the observation arc (see Fig. 11). The right ascension of the ascending node is shifted by approximately 0.002° for the orbit estimated without ERP data, corresponding to the Earth rotation angle during half a second. The estimated orbit elements are propagated for a further prediction time of up to three days. The position errors in Earth-fixed frame start to grow outside the observation arc. For longer propagation times the position errors exceed the level of the transformation errors, both in Earth-fixed and inertial coordinate frames.

Like errors in the time difference UT1–UTC, missing leap seconds lead to faulty coordinate transformations, and the position error can be estimated in the same way. If new leap seconds are not equally introduced to all sub-systems of the ground and space segment, relative time differences arise. The position error due to relative time errors results from the orbit velocity, e.g. 7.6 km per leap second at an orbit height of 500 km.

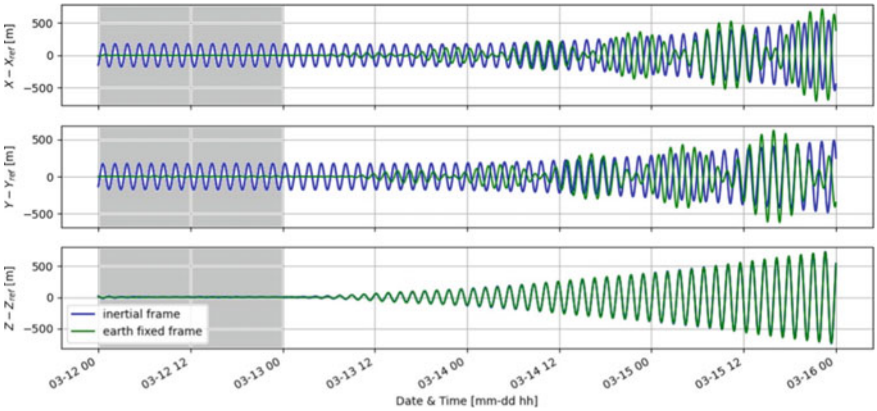


Fig. 11 Position errors due to missing Earth rotation parameter (observation arc in grey)

2.5 External Interfaces

Although V3C aims to be as self-sufficient as possible, any real-world deployment needs to interface with external systems. One external interface that is always necessary is the connection to a ground station or ground station network. Ground station connectivity is achieved by means of a bespoke protocol (NCTRS) for the first demonstration and by a standard protocol (SLE) for the second demonstration.

Another required external data source is provisioning of the current time. Laptop clocks are prone to drift and are not suitable as sole time source. Clock synchronization of at least one second accuracy has to be ensured, e.g. by regularly querying a "Network Time Protocol" (NTP) time server accessible via network or by a dedicated hardware clock utilizing GNSS or other time synchronization signals like DCF77.

For any upcoming user mission, the effect of missing or sporadic auxiliary data for Flight Dynamics due to intermittent Internet connectivity has been studied in Sect. 2.4. These effects will affect the mission design, leading to compromises between orbit position errors and the degree of self-containedness. Missing solar flux auxiliary data have the largest impact on the possible error, accumulating to 60 km in three days for an orbit height of 500 km in a maximum error estimation. These errors for example translate to deviations in the imaged area for an optical payload or to variations of ground station acquisition times. Up-to-date data are injected into the system at the beginning of each demonstration campaign. Possible errors during the short demonstration operations timeframe, which lasts less than three days, are deemed acceptable.

In addition to the external interfaces outlined above, the demonstration campaigns mandate further interfaces as further detailed in Sect. 4.1: Synchronization of certain products, like telecommand history, into the existing BIROS Mission Operations System is necessary in order to ensure the safety of the spacecraft and aid analysis in case mission control is exerted again from the existing BIROS system.

3 Deployment Concept

The deployment concept for this project, i.e. how the systems comprising the ground segment are rolled out on physical or virtual hardware and how they connect to each other, shall be outlined in this section.

3.1 Hardware

Owing to the mobility of V3C all components are integrated into a single mobile hardware, namely a commercial off-the-shelf (COTS) laptop. Its specifications (128 GiB RAM, 8 core CPU, 4 TB SSD) provide enough headroom for flexible adaption to a user mission and for the integration of more automation functionality. Although the demonstration system runs on powerful hardware because it is used as development testbed, deployment of the ground system is not closely tied to this particular piece of hardware: Project components are routinely rolled out on developer's machines without any changes compared to the production environment. This proves beneficial during the SARS-CoV-2 pandemic, where access to physical hardware is limited, also see Sect. 5. The strict separation between hardware and ground system deployment information, which will be detailed in the next section, will also allow shipping V3C as USB stick or download image, ready to be rolled out on user hardware on demand. The latter step is enabled by the chosen system design and will be further fleshed out during and after the two planned demonstration campaigns.

The COTS hardware acts as a host for a number of virtual machines, in which the system components are deployed. No component runs on the physical machine directly. This allows full infrastructure control without imposing too strict requirements on the hosting hardware. The physical host machine runs an Ubuntu Linux[5] operating system and uses Oracle VirtualBox[6] for virtualization. Both are no strict requirements: It is possible to use Microsoft Windows as host system without any changes. Replacing the virtualization solution is possible in principle, but might require some changes.

The system is built without redundancy in mind, neither for single VMs nor for the physical host. This simplifies workflows and lowers hardware requirements significantly but comes with the risk of losing satellite operability due to hardware or software failure. However, the project enables spinning up a completely new ground segment quickly in response to such a situation or have a second system running in parallel as hot redundancy. At this point in time redundancy concepts, especially state synchronization, are not further fleshed out. Yet, the system design and possible deployment options allow novel redundancy concepts not usually found within classical control centers, e.g. ad-hoc physically distributed places of operations via download of the control center or using cloud infrastructure. The demonstration campaigns are set up in a way that in case of failure the existing BIROS Mission Operations System can take over.

3.2 Infrastructure as Code

"Infrastructure as code" is a phrase that typically denotes the notion of defining data center hardware and provisioning with machine-readable files. Although this project only uses a single physical hardware the same technique can be used to describe the virtual infrastructure. The machine-readable definition files are treated like program source code and thus are checked into a version-controlled repository for development. Ad-hoc machine configuration, e.g. by logging in on a machine and

[5] https://ubuntu.com/.

[6] https://www.virtualbox.org/.

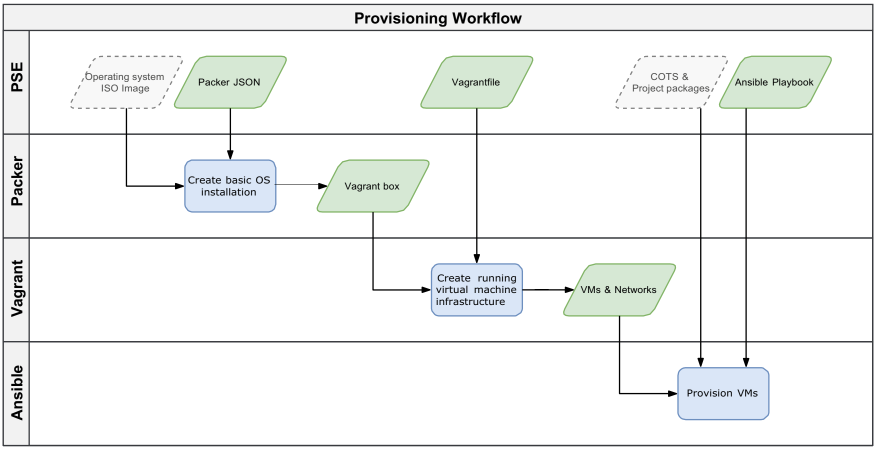


Fig. 12 Provisioning workflow for the roll-out of the V3C system. “PSE” is the Project System Engineer

performing configuration changes, is only possible during development and even in this case changes are only temporary. A continuous integration lifecycle recreates the whole infrastructure upon code changes, thereby nullifying any manual changes.

The system deployment is realized with three open source tools: HashiCorp Packer,[7] HashiCorp Vagrant[8] and Red Hat Ansible.[9] Figure 12 shows how these tools are used in the context of this project for provisioning the whole system. Once the system is rolled out none of these tools are needed any longer. The steps necessary for running a system without these tools need to be identified as part of the work following the demonstration campaigns.

3.2.1 HashiCorp Packer

Packer is used for creating basic virtual machine images for specific operating systems. A JSON file describes the virtualized hardware and how an operating system is installed into this virtual hardware, starting from the ISO images obtained from the operating system manufacturer. The resulting boxes are meant to be generic in order to serve as basis for possibly creating multiple machines from the same box.

[7] https://www.packer.io/.

[8] https://www.vagrantup.com/.

[9] https://www.ansible.com/.

3.2.2 HashiCorp Vagrant

Vagrant creates concrete virtual machines by specializing the boxes created in the previous step: It modifies the virtualized hardware resources according to the VM's role, assigns network interfaces and network addresses, mounts storage space (possibly shared between VMs), and triggers execution of the provisioning step with Ansible. It also defines the network rules governing the means by which the VMs may communicate with the outside world, i.e. with other physical machines.

3.2.3 Red Hat Ansible

Ansible works with the virtual machines created in the previous step and puts them in the desired state described by a so-called Playbook file. This involves installing required COTS and specialized project software packages, creating users and groups, configuring the systems and setting up the necessary directories, services and other resources as needed. Ansible is an agentless provisioning system, which means that no software agent is required to run on the machines to be provisioned. Ansible is executed on a control node and connects to each controlled node. For the role of the Ansible control node a dedicated VM was created in the previous step instead of executing Ansible on the physical host machine. This design is chosen in accordance with the policy to keep requirements for the host machine low.

3.3 Virtual Infrastructure Deployment

The infrastructure that is set up using the previously mentioned provisioning tools is shown in Fig. 13. All VMs are put in the same network 10.x.x.0/24, which is a private network that allows communication of the VMs with each other. Communication with any outside network is performed by forwarding specific ports from the physical host machine network interface to the appropriate VM. Runtime file data that need to be persisted, such as telecommand history data, and file data shared between VMs is exchanged via a Vagrant synchronized folder which is realized as an Oracle VirtualBox shared folder.

3.4 Development Environment

For implementing the V3C ground segment a dedicated development environment has been set up. It runs on the same project hardware as the final product and allows all engineers to develop their systems in parallel. In fact, the final product is merely an artifact of the development environment and itself fully integrated into the same

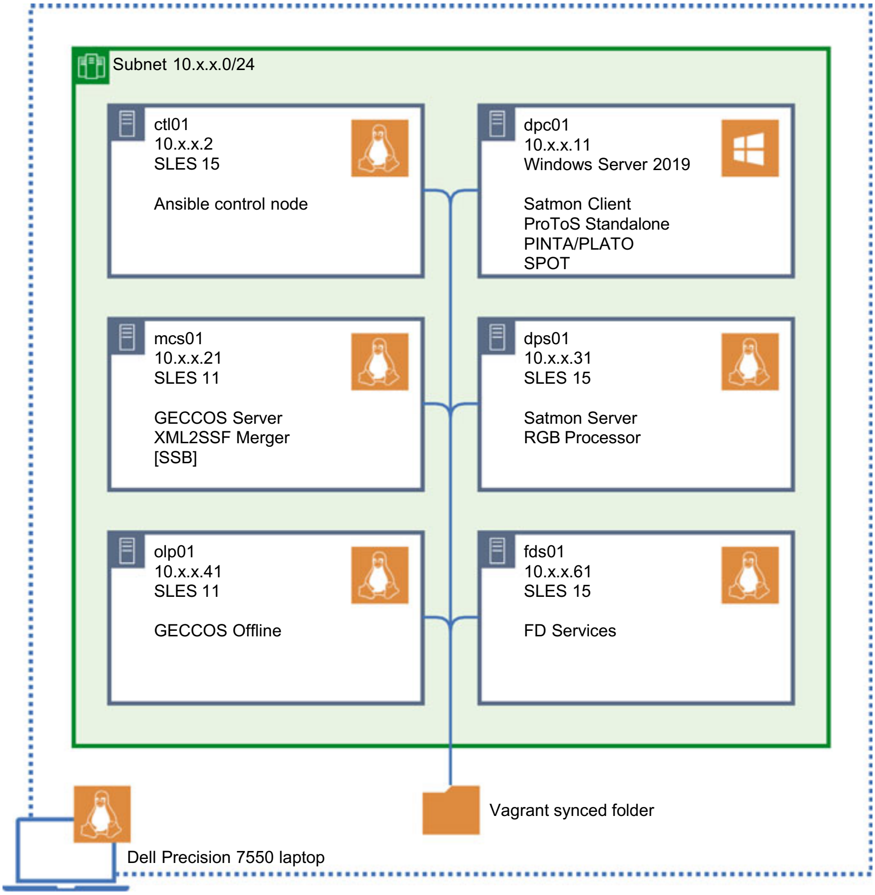


Fig. 13 Ground system infrastructure deployment

automatic deployment lifecycle as the test systems provided by the development environment. Because all infrastructure is handled as code, typical software engineering tools can be used:

The infrastructure definitions and dependencies are checked into a Git[10] repository managed by an in-house GitLab[11] instance. Commits to specific branches trigger different Continuous Integration/Continuous Deployment (CI/CD) pipelines that transform the committed code into running and provisioned infrastructure. Figure 14 shows how this enables automatic creation and removal of several simulation or development environments. Each of these environments represents a complete ground segment and has its own separated network assigned. One of them, the OPS environment, is specially protected and tested more rigorously through a staging environment

[10] https://git-scm.com/.

[11] https://gitlab.com/.

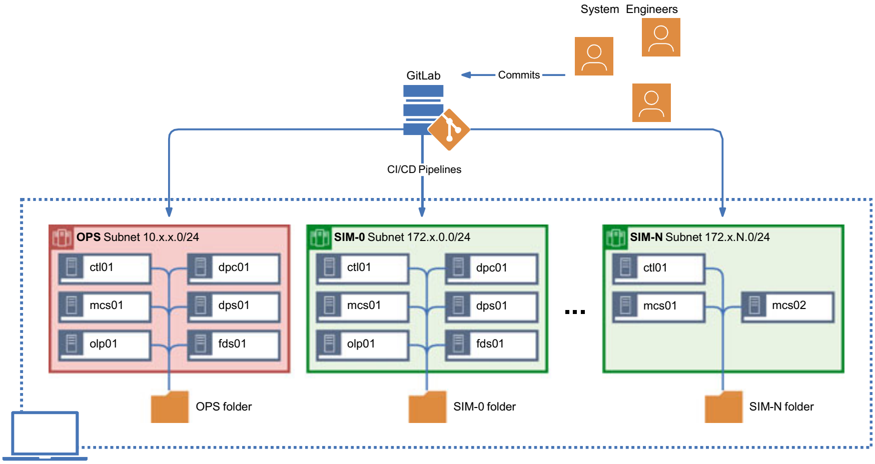


Fig. 14 Automatically managed V3C development environment with an operational system and a variable number of simulation or development systems

because it is the environment supposed to be used operationally. The OPS environment can be recreated into a known state directly from the backing infrastructure code at any point in time. All other environments are for the engineers to work with and are rather short-lived. They may be tampered with and also can be easily recreated at any point in time. Changes can only be made persistent by committing them to the Git repository.

4 BIROS Demonstration Operations

The whole concept of a mobile and compact Mission Operations Systems as outlined in this paper has been successfully demonstrated by performing operations of the BIROS satellite during a first demonstration campaign. Integration of V3C into existing infrastructure at GSOC as well as measures undertaken to ensure spacecraft safety and safety of the regular BIROS Mission Operations System are detailed in the following sections. At the time of writing the first demonstration campaign is still ongoing: The integration concept has been implemented such that telemetry and payload data is processed regularly on the new system and subsequent orbit determination provides the same solutions as the regular system. BIROS was commanded successfully using real-time telecommands as well as telecommands tagged with execution time stamps. Blind acquisitions, where the satellite telemetry transmitter is initially switched off and thus the satellite needs to be commanded *in the blind*, were performed smoothly. Safety of BIROS and its regular Mission Operations Systems was guaranteed at all times. This was proven by interleaving spacecraft operations

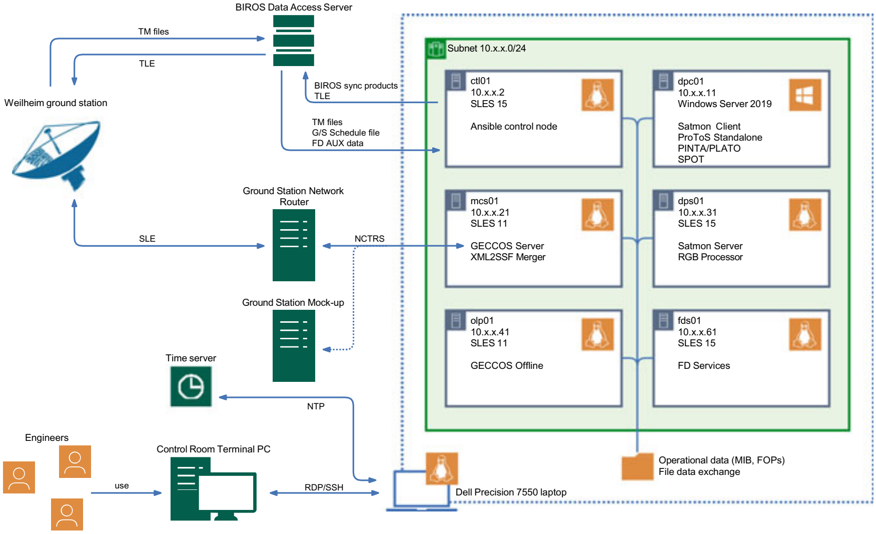


Fig. 15 Integration of V3C into existing infrastructure at GSOC for the BIROS satellite

from both systems without any issues. Although all bits and pieces have been demonstrated successfully with the satellite in the loop, the only outstanding demonstration task is that of a full cycle from image order to uplink to downlink to processing and display. Due to a spacecraft outage unrelated to our efforts this last task could not be performed yet.

4.1 *Integration Concept*

The integration concept for demonstration from GSOC is depicted in Fig. 15. All operations were performed on the V3C hardware described in Sect. 3.1 which was set up in one of the GSOC control rooms. Due to limited screen space and ergonomics all access to the V3C hardware was performed by graphical (Remote Desktop Protocol, RDP) or textual (Secure Shell, SSH) remote connections from one of the terminal computers at the consoles in the control room. The system has been set up with a minimum number of external interfaces in order to integrate into the GSOC infrastructure for the first demonstration: real-time connection to the ground station via NCTRS protocol, connection to a network time server via NTP, file exchange to and from the data access server of the regular BIROS Mission Operations System as central data integration hub, graphical and textual remote access to all systems for the engineers and operators. File exchange data comprise orbit information provided to the ground station for antenna pointing, synchronization products to the existing BIROS system in order to resume commanding from there at any time, recorded telemetry files from the ground station, auxiliary Flight Dynamics data as described in Sect. 2.4 and a ground station scheduling plan because of non-exclusive access to the antenna.

4.1.1 Safety of BIROS and the Existing Mission Operations System

Integrating a new Mission Operations Systems with different redundancy characteristics into an existing system could prove problematic in case of outages. Therefore, measures had to be taken that such an outage does not affect the regular system and that commanding of the satellite can be resumed using the regular system at any time. The latter is ensured by the arrow labeled "BIROS sync products" in Fig. 15. These products comprise lists of sent telecommands and their acknowledgements, the onboard queue model and telemetry check results. Recorded ground station telemetry files are delivered to the existing system and to V3C in parallel, such that spacecraft telemetry is always available even in case of failures. File transfers are operated in *pull mode*, where V3C pulls file data from a reliable data hub and thus does not negatively impact operational file transfers of the existing system in case of a failure.

Extensive testing of all internal and external interfaces according to a test plan ensures safety of the spacecraft itself. A ground station and satellite mock-up took place of the real systems during first tests due to lack of access to a BIROS simulator or engineering model. Data flow tests with Weilheim ground station and finally telemetry and telecommanding tests of increasing complexity were then performed successfully with the BIROS flight model.

5 SARS-CoV-2 Challenges

The project faced (and still faces) some challenges due to the COVID-19 pandemic: The project officially started in October 2020 with some preliminary work being done since April 2020, i.e. the project was influenced by the pandemic and lockdown restrictions right from the start. The project team members never had the opportunity of meeting face-to-face with each other, except for two people at the same time. All work meetings were conducted via teleconferences, telephone, chat systems or e-mail contact. With a general work-from-home order by DLR the project had to be setup in a way no other ground system project has been setup before. Owing to the particular concept of V3C, namely designing and implementing a compact and mobile ground system, these restrictions actually turned out to be beneficial from a technical viewpoint as will be detailed in the following.

Because the project uses an agile approach, requirements engineering, system design, implementation, and tests do not happen in subsequent phases but continually and cyclically throughout the project lifetime. This means that implementation work already commenced almost immediately after project kick-off. At this point no project hardware had been procured yet, much less integrated into GSOC's network landscape. As laid out in Sect. 3.1 the ground system shall be largely independent of the hosting hardware. Therefore, a development environment was set up on each engineer's computer which they used to work from home. The development environment allowed each engineer to work with the complete ground system as implemented so far, which was executed on his or her own computer without resorting to any GSOC control center infrastructure. The same "infrastructure-as-code" techniques were used for the home development environments as are now used for the development environment on the project hardware. All ground system infrastructure was defined in a common Git repository right from the start, made available on each developer's machine, and designed in a robust way to overcome the differences between the developer machines. These were not only differences in hardware specifications, like the amount of available memory, but also in the software environment, like different operating systems. By being forced to make available the development environment not on one centrally accessible system but on each developer system the whole environment was portable since the beginning, thereby fulfilling one of the project goals.

Once the project hardware was delivered, the same development environment was rolled out there. Because the network setup was not yet ready at this moment, the hardware was placed outside of GSOC at first, allowing remote access to all engineers. Later, the hardware was physically moved to GSOC and connected to a network accessible from home through a virtual private network. This was already a first successful test for the mobility of the system as is stated project goal. The restrictions imposed by the pandemic almost automatically aligned with some of the project goals and helped implement them right from the start.

Of course, a development system is not comparable to the final product in every respect: While at first the development system closely resembled the setup of the final product because it allowed working from home, later on a more elaborated development environment fitting into GSOC infrastructure was needed, which is described in Sect. 3.4. The final setup allows multiple engineers to work on the same system, which was not possible with the work-from-home solution.

6 Conclusion and Outlook

We have shown a system design and implementation of a compact and mobile ground segment on a COTS laptop for small satellites in low-Earth orbit. It can be put to use anywhere in the world with minimal connectivity required except for a ground station, which could be equally compact and mobile in principle. Compromises in mission design due to intermittent connectivity have been assessed. Two first demonstration campaigns are currently ongoing or in preparation, showing the feasibility of this concept with the optical imaging payload of the BIROS satellite. How to set up and roll out such a system in an automatic fashion has been shown in detail, as well as how the current global pandemic influenced and even partly helped the project.

Automation functionality for fully automatic workflows initiated by a single operator will be implemented, some of it already for the second demonstration. New deployment concepts like putting the ground segment on a USB stick, making it available for download, or putting it in the cloud will be assessed, as well as the new redundancy possibilities that arise from these concepts. Combining this compact and mobile Mission Operations System with an equally compact and mobile ground station also leads to fascinating new possibilities for space operations.

Acknowledgements Our special thanks go to Martin Wickler for initiating this project, providing core ideas and securing initial funding through DLR's German Space Operations Center GSOC. We sincerely appreciate project funding provided by Wolfgang Jung through the DLR Responsive Space Cluster Competence Center RSC^3 with support by the German Federal Ministry of Defence. Finally, we thank the team members and supporting staff bringing this project to life.

References

1. Singer T (2016) Implementation of the true multi-mission control room at GSOC. In: SpaceOps conferences, American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2016-2454
2. CCSDS (2006) Cross support concept—Part 1: Space Link Extension. https://public.ccsds.org/Pubs/910x3g3.pdf, Green Book, 910.3-G-3
3. Reile H, Lorenz E, Terzibaschian T (2013) The FireBird mission—a scientific mission for Earth observation and hot spot detection. In: Digest of the 9th International Symposium of the International Academy of Astronautics, Wissenschaft und Technik Verlag. https://elib.dlr.de/83866/
4. Stangl C, Lotko B, Geyer MP, Oswald M, Braun A (2000) GECCOS—the new monitoring and control system at DLR-GSOC for space operations, based on SCOS-2000. In: SpaceOps conferences, American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2014-1602
5. ECSS (2003) Ground systems and operations—telemetry and telecommand packet utilization. ECSS-E-70-41A
6. Peat C, Hofmann H (2004) SATMON—a generic user interface for satellite control. In: SpaceOps conferences, American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2004-316-163
7. Beck T, Schlag L, Hamacher JP (2016) ProToS: next generation procedure tool suite for creation, execution and automation of flight control procedures. In: SpaceOps conferences, American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2016-2374
8. Hamacher JP, Beck T (2018) ProToS: automation of flight control procedures for the European Data Relay System. In: SpaceOps conferences, American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2018-2541
9. Wörle MT, Spörl AK, Hartung JH, Lenzen C, Mrowka F (2016) The Mission Planning System for the Firebird spacecraft constellation. In: SpaceOps conferences, American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2016-2621

SEC_LAB: A Secure Communications Testbed for Space Missions

Marcus Wallum, Daniel Fischer, Jadwiga Nowotnik, Łukasz Pieczonka, and Mariusz Tkaczyk

Abstract The Consultative Committee for Space Data Systems (CCSDS) presents several standards and informational reports concerning the security of the space to ground data link, in particular through the specification of a Space Data Link Security (SDLS) Protocol, which enables authentication and encryption for missions utilising standard Space Data Link Protocols (SDLP). SDLS Extended Procedures (EP) specify services, procedures and data structures to manage, monitor and control associated security primitives. A flexible and representative test environment is required in order to assess the suitability of these protocols and also of standard terrestrial secure communication technologies in order to de-risk and promote their uptake for future missions. This paper presents work undertaken to design and implement a representative testbed laboratory for the testing of CCSDS secure protocols and the suitability of IP-based terrestrial network and security components for protection of the space datalink and improved communications flexibility. The developed Secure Communications Laboratory (SEC_LAB) is a virtual testbed which simulates a real space link based on ESA's mission control (MICONYS) and test and validation (TEVALIS) software infrastructure. Several use cases, tests and scenarios have been explored and quantified test results provide insights and recommendations. The feasibility of utilising encapsulated terrestrial networking technologies including

M. Wallum (✉) · D. Fischer
Ground Systems Engineering & Innovation Department, Directorate of Operations, European Space Agency, Robert-Bosch-Str. 5, 64392 Darmstadt, Germany
e-mail: marcus.wallum@esa.int

D. Fischer
e-mail: daniel.fischer@esa.int

J. Nowotnik · Ł. Pieczonka · M. Tkaczyk
Newind Inc., ul. Ostrowskiego 7, 53-238 Wrocław, Poland
e-mail: jadwiga.nowotnik@newind.pl

Ł. Pieczonka
e-mail: lukasz.pieczonka@newind.pl

M. Tkaczyk
e-mail: mariusz.tkaczyk@newind.pl

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_20

encapsulated IP/IPSec over CCSDS protocols, Virtual Private Networks (VPN), Software Defined Networking (SDN) and host fingerprinting is demonstrated. Several lightweight encryption and authentication algorithms are tested to measure overheads and performance impacts, identifying the symmetric block ciphers SPECK and Advanced Encryption Standard (AES) as suitable options. Optimization of on-board processors for particular algorithms is identified as a potentially important factor. Simulated disruption tests indicate that terrestrial protocols are susceptible to disruption, in particular those with handshake authentication operations. Throughput testing for implementations of Galois/Counter Mode (GCM) and Cipher-based Message Authentication (CMAC) indicate optimal sizes for Telecommand (TC) and Telemetry (TM) transfer frame throughput. SDLS EP procedures are validated and a number of risks and mitigation measures are discussed. The output of the activity supports de-risking and informed decision making for investments in adapting existing control systems and securing the data link, thereby addressing a key security threat for future space missions. Next steps and potential future work in the domain include maturation of the testbed and implementation and testing of secure protocols for the next generation of mission control software (European Ground Operations System–Common Core (EGS-CC)).

Keywords Communication · Security · Laboratory · Spacelink · Testing

Acronyms/Abbreviations

CPU	Central Processing Unit
CMAC	Cipher-based Message Authentication Code
CMD	Command
CCSDS	Consultative Committee for Space Data Systems
(D)TLS	Datagram Transport Layer Security
DTN	Delay Tolerant Networking
TEVALIS	ESA Ground Test and Validation System infrastructure
MICONYS	ESA Mission Control System infrastructure
EGS-CC	European Ground Systems–Common Core
EP	Extended Procedures
GCM	Galois/Counter Mode
GEO	Geostationary Orbit
GSTVi	Ground Systems Test and Validation infrastructure
GT APP	Ground Test Application
IP	Internet Protocol
IPSec	Internet Protocol Security
KM	Key Management
LEO	Low Earth Orbit
LAN	Local Area Network
MTU	Maximum Transmission Unit

MEO	Medium Earth Orbit
M&C	Monitor and Control
OVS	Open vSwitch
OF	OpenFlow
OWLT	One Way Light Time
PT APP	Processing Test Application
SCOS 2000	Satellite Control and Operation System 2000
SEC_LAB	Secure Communications Laboratory
SA	Security Association
SDB	Security DataBase
SPE	Sent Packet Encapsulation
SDN	Software Defined Networking
SDLP	Space Data Link Protocol
SDLS	Space Data Link Security
SLE	Space Link Extension
SSME	Spacecraft Specific Model Extension
TC	Telecommand
TM	Telemetry
UDP	User Datagram Protocol
VPN	Virtual Private Network
WAN	Wide Area Network

1 Introduction

1.1 Background

Space systems and the data, products and services they provide are increasingly relied upon by society, increasing their attractiveness as targets for adversaries. One of the exposed components of a typical architecture is the ground to space communications link—a critical interface which, for most unclassified missions, has been traditionally considered sufficiently protected through the high entry barriers to the space domain in terms of cost and complexity. With these entry barriers lowering, means for proper protection through encryption and authentication are key for both the protection of the spacecraft command uplink and the downlinked data.

The Consultative Committee for Space Data Systems (CCSDS) presents several standards and informational reports concerning the security of the space to ground data link, in particular through the specification of a Space Data Link Security (SDLS) Protocol [1], which enables authentication and encryption for missions utilising compliant Space Data Link Protocols (SDLP) [2, 3]. Supporting Extended Procedures (EP) [4] specify services, procedures and data structures to manage, monitor and control associated security primitives. However today, the uptake and implementation of these protocols in real missions remains limited, due to the aforementioned traditional perception and associated cost and risk considerations ("has not flown, will never fly"). Similarly, common protocols and controls which have been long used in terrestrial networks such as Internet Protocol Security (IPSec) [5] Virtual Private Networks (VPN), Transport Layer Security (TLS), as well as modern architectures such as those supported by Software Defined Networking (SDN), face both technical and managerial challenges when it comes to spin-in for use in the space domain.

A flexible and representative test environment is therefore required in order to assess the suitability of these protocols and technologies in order to de-risk and promote their uptake for future missions.

1.2 Contribution

In this paper we present a development undertaken by the European Space Agency (ESA) which answers to the above mentioned need, through the provision of a testbed or 'Secure Communications Laboratory' (SEC_LAB). SEC_LAB provides a modular testbed architecture, integrated with ESA's mission control (MICONYS [6]) and test and validation (TEVALIS [7]) software infrastructure. SEC_LAB is the result of research and development work conducted under sequential projects, each focusing on particular aspects of implementation and testing of a secure ground to space communications stack. These include the following:

- Implementation of a Software Defined Networking (SDN) concept for separation of the control and data planes in the ground to space network and enabling of fingerprinting data source signatures.
- Implementation of IP over CCSDS SDLP and testing of different security protocols including IPSec and (Datagram) Transport Layer Security (DTLS) [8] over the space link.
- Analysis of security protocol robustness in the presence of simulated disruption.
- Testing of various lightweight cryptographic algorithms implemented for CCSDS SDLS and measurement of associated performance impact.
- Implementation and testing of CCSDS SDLS Extended Procedures including encryption Key Management (KM), Security Associations (SA) management and Monitoring and Control (M&C) services.

Through configuration, this allows testing of a standards-compliant reference system under varying conditions. The SEC_LAB development has since been utilised to; (i) communicate and discuss with mission and security stakeholders various considerations for implementation, (ii) verify and validate the secure protocols and standards and (iii) provide test data related to different configurations for validation and comparison by implementing missions.

1.3 Paper Organisation

The remainder of this paper is organised as follows: In Sect. 2 the overall SEC_LAB architecture and approach to realising the required functionalities is described. Section 3 covers the approach to utilisation of the SEC_LAB and main tests conducted. Section 4 presents the results of the development, the conducted tests and discusses findings. Finally, Sect. 5 provides overall conclusions and possible future development areas.

2 Material and Methods

2.1 Overall Solution Architecture

The solution is hosted on virtual machines and networks and is realised by both existing third party solutions and custom developed software modules. A high level logical architecture and data flow is depicted in the below Fig. 1.

The testbed is segregated into different networks, representative of a ground segment (Ground site), space-to-ground data link (SPACELINK) and space segment (Space site). A scenario of possible LANs (Local Area Networks) within both the ground and space segments is envisaged and flexible data exchange between two sites at a network address level is enabled through IP protocol encapsulation into SDLP/SDLS. The SDN elements of the system consist of the following components:

- OVS: OpenFlow VSwitch, allowing data transmission across ethernet ports, controlled by management software which enables host and network traffic filtering based on rules.
- Floodlight SDN Controller: the management console which manages the network rules applied by the OVS. Communication is via the OpenFlow (OF) programmable network protocol.
- SDN Dongle: custom developed software service, which enables hosts authentication.

In addition to the SDN Dongle, further custom developed elements of the system include:

- Ground and Processing Test Application(s): custom software applications with various functions depending on the configuration of the test bed and the tests to be conducted.
- SEC_COM Router: this software module intermediates in data exchange between the local network and the space-to-ground datalink components (see Sect. 2.1).

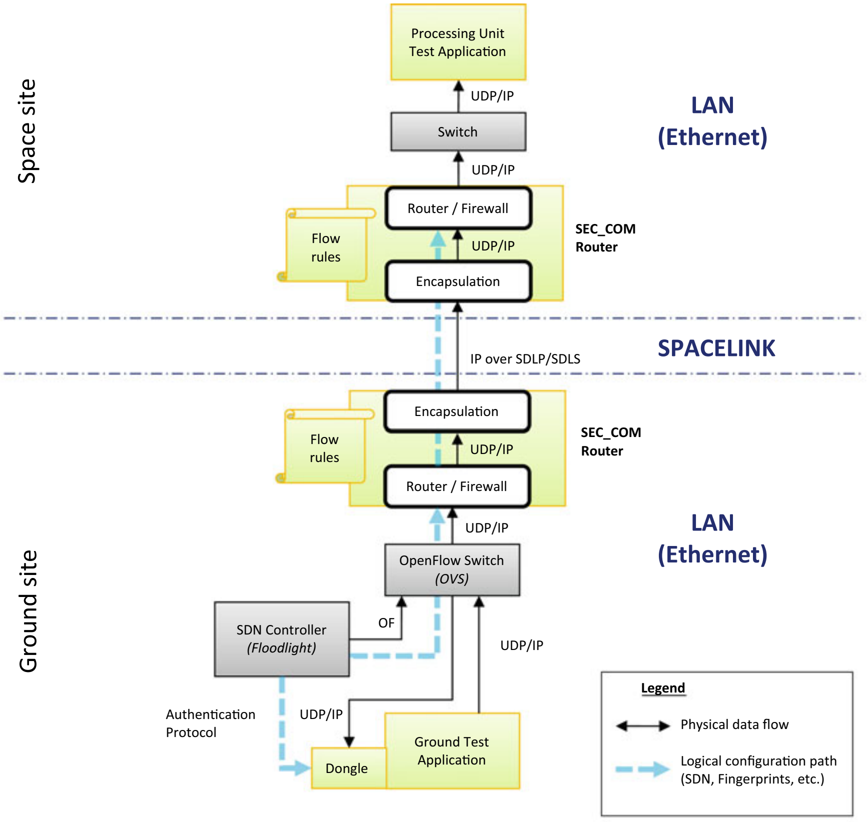


Fig. 1 SEC_LAB high level logical architecture and data flow

The CCSDS SDLP-compliant space-to-ground data link is realised by ESA's mission control (MICONYS) and test and validation (TEVALIS) software infrastructure, in particular the Satellite Control and Operation System 2000 (SCOS-2000) mission control system and Ground Systems Test and Validation infrastructure (GSTVi) simulator software. Some extensions to the SCOS-2000 software, in particular the Packetiser library, and to the GSTVi software, in particular modifications to manage encapsulation packets by the Spacecraft Specific Model Extension (SSME) were required, in order to support encapsulation packets, otherwise the software was used as-is. Encapsulation Packets, rather than Space Packets (both valid options for SDLP/S), were used in order to support IP encapsulation.

The physical architecture of the testbed is depicted in Fig. 2 and is implemented within a virtualised environment, providing flexibility in terms of configuration. Control and forwarding planes (indicated by red and black wires) are decoupled to realise the SDN approach and provide a management/configuration capability.

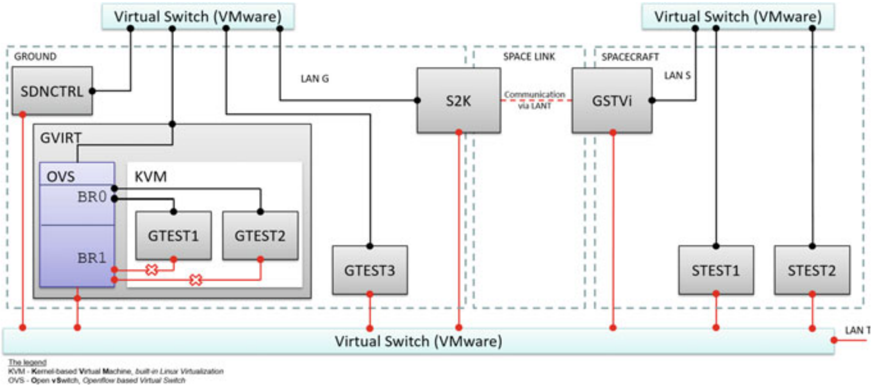


Fig. 2 SEC_LAB physical architecture

OVS integrates with the physical network utilising Linux network adapters and the ports of the virtual network adapters. The hosts with ground test applications (GTEST) for SDN purposes, were virtualized via Kernel-based Virtual Machines (KVM), which in turn were installed together with OVS on the same VMWare host (GVIRT).

The testbed is modular and some modules may be removed depending on the required capabilities to support specific tests, e.g. SDNCTRL and GVIRT can be removed if SDN functionalities are not required.

Various test applications (on GTEST and STEST hosts in Fig. 2) were developed to support different tests and research goals. Functions included:

- Executing transmission of files and data between ground and space segment networks.
- Configuring operation modes e.g. clear mode, authentication, encryption, authenticated encryption.
- Encryption and decryption for DTLS and IPSec protocol testing.
- Logging of efficiency metrics including timestamps, overheads, transmission restarts.
- Measurement and logging of throughput of various encryption algorithms and mode implementations.
- Receiving, sending and processing of SDLS EP commands/Protocol Data Units (PDUs).
- Management of the security database (for SDLS EP).

2.2 SEC_LAB Router Application

One of the significant custom developed components is the Router, which intermediates in communication between the ethernet network and the space-link. The Router is a software module which allows to customize Linux-based network devices to the requirements of an encapsulation module.

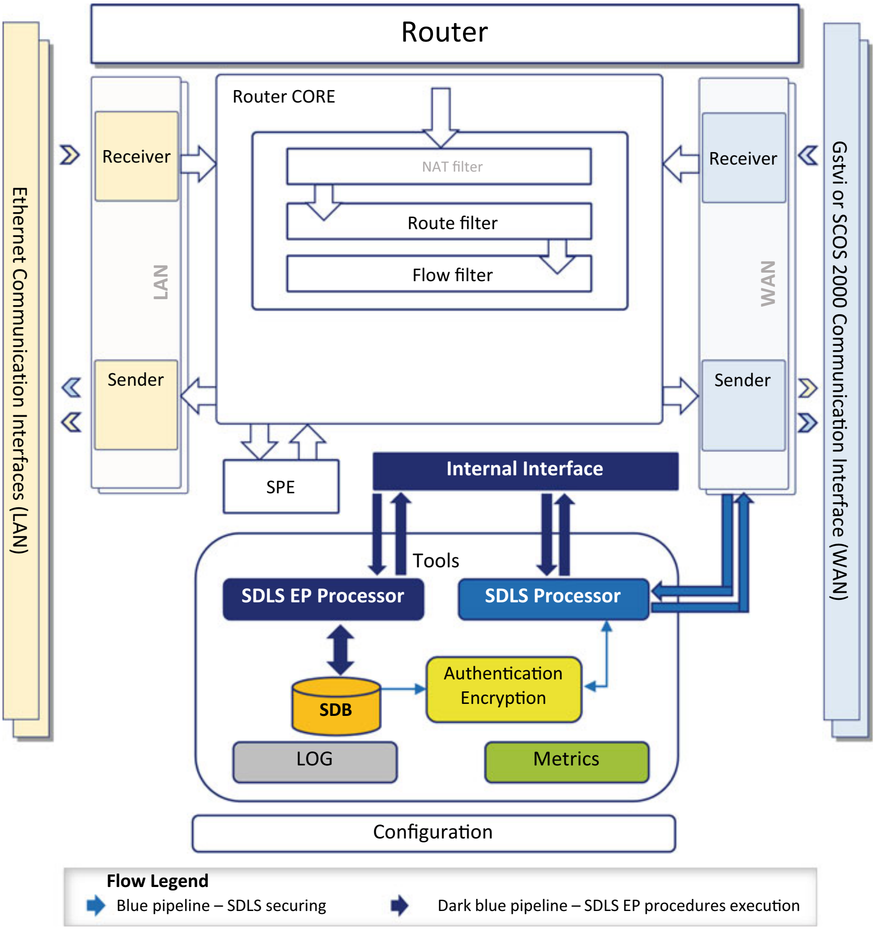


Fig. 3 SEC_LAB router main components and data flows

Individual components of the Router Software consist of modules grouped by function groups as depicted in Fig. 3. The Router works closely with the system components of the network device on which it is implemented, enabling flexible configuration of the encapsulation functionality for many communication interfaces of various network types and protocols. These functions include the following:

- Communication, including receive and send modules, LAN (Local Area Network over Ethernet) and WAN (Wide Area Network over space link) external network interfaces, conveying data to and from the networks and encapsulation/decapsulation of IP (or IPSec, DTLS) into/from SDLP (or SDLS).

- SPE (Sent Packet Encapsulation) is a component responsible for splitting IP packets (IP fragmentation, described in RFC-791 documentation [9], the Internet protocol implements fragmentation functionality), which are larger than maximum transmission unit (MTU), specified at a destination network interface. The splitting operation is run if needed, which typically depends on the data/files sizes sent and the testbed configuration. If there is no need for splitting packets, then conveying is straight forward for further processing. Fragmentation of an IP datagram is necessary when it originates from a network that allows a large packet size and must traverse another network that limits packets to a smaller size to reach its destination.
- Internal interface module responsible for communication with internal data receiver and processing units e.g. processing of SDLS EP commands.
- Network/packet filtering, including Network address translation (NAT), routing table for packet filtering based on source/destination IP and mask, interface, port, protocol type and utilised as required by the testbed configuration.
- Flow filtering for flow rules and forwarding of packets to/from LAN/WAN.
- SDLS Processor implements the security protocol i.e. preparation, reception, encoding/decoding. The Processor detects SDLS frames and forwards to the Internal Interface for further processing, either for SDLS EP operations (SDLS EP Processor) or for encryption/authentication services.
- SDLS EP Processor receives and processes SDLS EP command PDUs and implements the corresponding operation e.g. on the local security database (SDB). Upon execution of the operation, a result is prepared and sent back to the internal interface and on to destination network.
- Authentication/Encryption component provides the set of security services for securing SDLS based communication. This includes configuration for changing SDLS mode e.g. encryption only, authentication only or authenticated encryption (default). This component contains different encryption algorithm libraries which may also be selected by configuration, including AES [10], SPECK [11], TWOFISH [12], PRESENT [13], DESXL [14].
- SDB (Security Database) stores all security data managed by SDLS/EP. The data might be encryption keys, security associations data, logs, etc. The SDB is modified by SDLS Processor. The Authentication/Encryption module reads from the database necessary parameters for securing the spacelink communication through SDLS. Because of the research purpose of the testbed, the SDB is unencrypted and stores data in plain text files for convenience of allowing changes and visibility of responses during experiments.
- Configuration files are local resources which contain the various configurable settings e.g. test application parameters, routing tables, Address Resolution Protocol (ARP) table, etc.

3 Material and Methods

The testbed architecture described in the previous section was designed to be modular in order to support various tests and research oriented questions. The main testing areas may be grouped as follows:

- SDN related operations, including host fingerprinting
- Encapsulated IP over SDLP/SDLS and establishing IPSec VPN over the datalink
- Performance testing of the above
- Lightweight encryption algorithms
- Establishing DTLS over the datalink
- Performance and throughput testing of the above, including stability in simulated disruption
- Validation of SDLS EP and experimentation with EP for IPSec management.

For each of these, various test applications were developed, which were then integrated with the testbed and Router components.

3.1 SDN Tests

Software Defined Networking is a well established approach which enables dynamic, programmable network configuration and which is widely used in terrestrial networks. The objective of the SDN functionality in the testbed was to determine whether 'spin-in' of such technology would be feasible for a space system architecture whereby firewall rules as well as host-level fingerprinting and authentication and packet-level filtering could be configured without impacting performance over a space-to-ground data link. The scenario and dataflow for these tests included the following steps:

- Ground Test Application is sending UDP/IP packets to the Processing Unit Test Application at spacecraft site (as depicted in Fig. 1).
- Packets are sent to the Openflow switch at ground site. If the switch contains an appropriate rule, then it forwards to the router/firewall component.
- Openflow switch asks SDN Controller permission for forwarding.
- SDN Controller authenticates the test application host via the Dongle component. SDN Controller sends the random generated message to the Dongle.
- The Dongle signs the message with its private key, shared only with the host and the SDN Controller. Dongle sends back the digest result (a fingerprint). SDN Controller compares the response with its own calculation and if both results match then sends back to the Openflow switch a rule allowing the transmission. Then the Openflow switch forwards the IP packet to its destination.
- Router/firewall checks the packet source/destination versus its local ruleset and if viable, encapsulation of IP into SDLP process begins. If not, the packet is discarded.

- Following transition, decapsulation and forwarding to the destination host at Space site, no further checks are performed and a classical switch is utilised i.e. the space site LAN is considered trusted. Also no further checks were performed using the OpenFlow switch for returning telecommand data.

Through configuring different hosts and rules, the approach was validated and performance impacts measured (see Sect. 4.1).

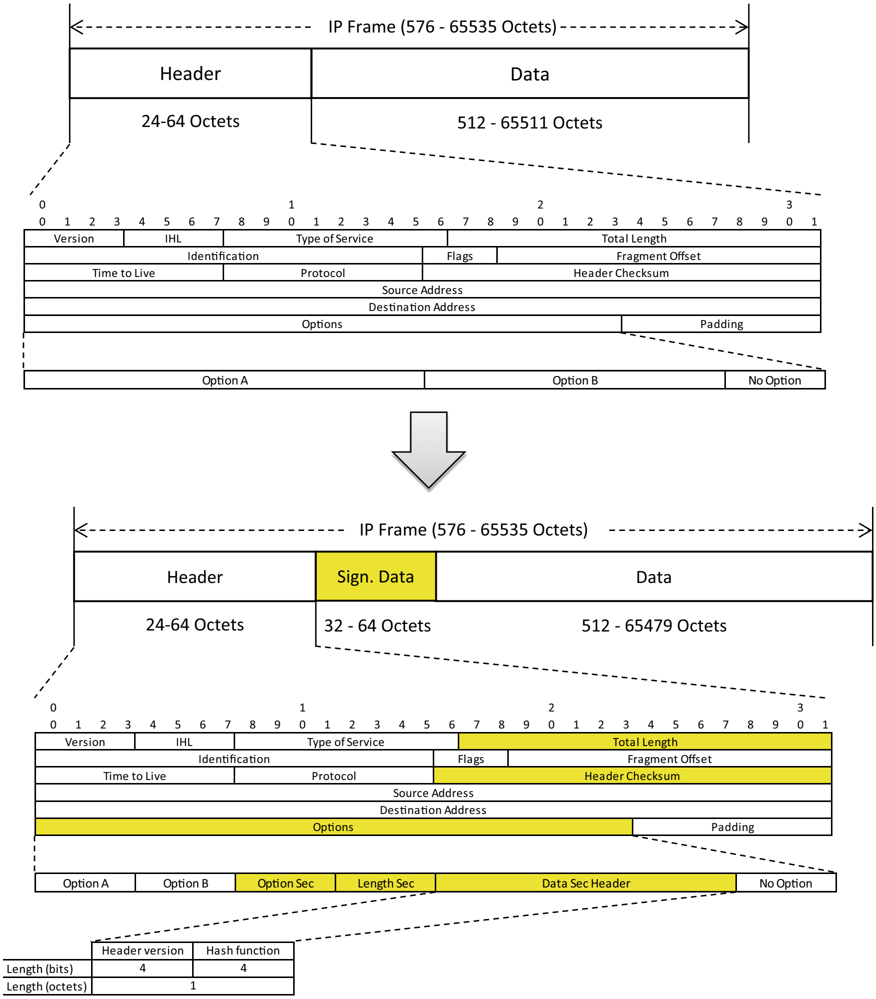
3.2 IP Over SDLP/SDLS and Establishing DTLS and IPSec VPN

The encapsulation of IP over SDLP is standardised by CCSDS [15, 16]. Network layer addressing over the space-to-ground data link enables more advanced addressing and routing scenarios with ground and space segment LANs. The objective of this testing was to investigate if also network layer security implementation such as establishing authentication of IP packets (Sect. 3.2.1), use of (D)TLS protocol and establishing a tunnel through use of the IPSec protocol (Sect. 3.2.2) could also be achievable over SDLP/SDLS.

3.2.1 Signing IP Frames

The testbed and supporting test application were configured to be compliant to CCSDS [5, 6], particularly the Protocol ID is set to 010—Internet Protocol Extension (IPE). As a result there is no option to include additional data such as a digital signature, however the IP standard does allow to add a custom option field to the IP frame header. The implemented approach is therefore to add additional security data inside modified IP frames, which is performed by the encapsulation component of the Router application, prior to the encapsulation operation. The IP frames are then put one-to-one inside encapsulation packets, which are transmitted via SDLP/SDLS frames (TC and TM). At the receiving space site, the reverse process i.e. decapsulation, reading and removal of additional data from the IP frame. The process is transparent for the hosts in each LAN, however during the transmission through the data link and only for the Router component, the approach is not fully compliant with RFC-791 due to the custom use of the optional field.

In detail, during the encapsulation process the new option is added at the end of the option list. The code of the new option is set to 10011111 (the highest possible). Then, the additional security header is added, prepending the octet containing the length of data and option 1 is added together with padding field (zeros). Then the additional security data is added before the data field containing the original IP payload data. The Total Length and Header Checksum fields are recalculated and the process ends. The below figure Fig. 4 depicts this modification data in the IP datagram. The supported authentication process is then similar as in case of SDN, i.e. utilising the security data/secret key, known only to Routers with Encapsulation Modules in the network, a digest is calculated. If the result is equal to the signature read from the IP frame, then the frame is authenticated.



	Header version	Hash function
Length (bits)	4	4
Length (octets)	1	

Fig. 4 IP datagram modification

3.2.2 Establishing IPSec VPN and DTLS Over SDLP

Another widely adopted protocol in terrestrial networks is IPSec. The objective of these tests were to establish IPSec communication over the SDLP/SDLS data link and measure impacts in terms of overheads and stability. For establishing the VPN tunnel a dedicated test application re-using the third party software Strong Swan [17] were integrated with the test bed. This also permitted continued use of UDP and avoidance of the TCP handshake overhead. The VPN channel is successfully established between the ground and space LAN hosts and the IPsec traffic is conveyed through the space-to-ground datalink in the same way as ordinary IP frames.

In addition to IPsec link, DTLS is also supported as a security layer option above the network layer. The corresponding test application is also capable of sending data files from ground to spacecraft host through the spacelink utilising this protocol, which is also UDP/IP based and avoids the handshake required for TCP.

3.3 Lightweight Encryption Algorithms

An active area of research is in the field of lightweight encryption algorithms where applications in the space domain address similar boundary conditions as, for example, embedded systems or IoT sensor networks in terms of constrained resources and a need for minimising computational cost. For the purposes of the SEC_LAB development, a set of candidate algorithms, based on the results of competitions and developments in the cryptography community and their availability as stable implementations at the time, were selected. These were SPECK, TWOFISH, PRESENT and DESXL. AES is also supported, as a very widely adopted algorithm and benchmark. Each algorithm is applied in the testbed through configuration and each is tested in terms of computation overhead, cipher operations and throughput, stability and packet re-send overheads in the presence of simulated disruption. Galois/Counter Mode (GCM) (for authenticated encryption) and Cipher-based Message Authentication Code (CMAC) (for authentication only) were utilised and tested.

3.4 Datalink Degradation Module

One of the challenges in space communications is stability under disruption e.g. bit flips caused by radiation. In order to test the various combinations of algorithms and protocols described above and their stability, the SEC_LAB, through a dedicated module, is capable of disrupting the transmission, i.e. of entire SDLP or SDLS frames being sent through the space-to-ground datalink. The degradation works by changing the bits of transmitting data frames to intentionally raise transmission errors. If transmission errors occur then at the receiving end of the spacelink, while decoding SDLP/SDLS frame the calculated CRC differs compared to the one being sent inside the frame footer. In this case the receiving end identifies the frame as disrupted or corrupted and it is not processed further. The probability of data disruption is configurable separately for TM and TC, and is given by number of bits to be disrupted in the total number of bits, e.g. 3/100000. The associated test application transmits a data file, allowing to measure data transmission time and data overheads in case of disruption of transmission.

3.5 Validation of SDLS Extended Procedures

During the SEC_LAB development, the authoring of the CCSDS SDLS EP standard was ongoing. The SEC_LAB was therefore utilised to fully validate the standard. Through the implementation of all KM, SA Management and M&C services, each procedure of the standard was tested with fully compliant Protocol Data Units (PDU) transferred across the space-to-ground data link. As described in Sect. 2.2 above, dedicated modules were developed for the routing and management of the EP specific packets (SDLS EP Processor) and the execution of procedures is performed against the SDB, enabling monitoring and changing of the various states and managed parameters in the standard e.g. Initialisation Vector, key states, SA states, Virtual Channel IDs, etc. In particular the encryptionSuite and authenticationSuite are configurable with the implemented algorithms mentioned in above Sect. 3.3. After modification within the SDB, the new values are applied in subsequent communications and security operations performed on the testbed.

In addition to the standard procedures validation, extensions to the KM procedures with additional responses were defined to support error handling. As an experimental exercise, a further set of tests were conducted to assess the suitability of the EP for managing IPSec parameters.

4 Results

The resulting SEC_LAB development provides a powerful, flexible capability for conducting testing of several implementation and configuration options for a standards-compliant secure space-to-ground data link. The developed solutions are considered representative enough of real mission systems for the purpose, in particular through the use of the SCOS 2000 mission control system and the GSTVi simulation environment, both used extensively by ESA for real missions. Utilising this development, extensive testing capabilities as introduced in Sect. 3 were used and several insights, results and recommendations were obtained.

4.1 SDN as Viable Technique for Mission Ground Segments

The SDN operations were run with success, and enabled effective packet-level filtering and routing rules to be applied by configuration. Together with the SDN Dongle and Controller components, a host authentication and fingerprinting capability was achieved. As a simulation of an associated threat, by introducing new hosts in the ground site LAN, the effective blocking of traffic from the unauthorised/spoofed host was demonstrated. These tests indicate that SDN techniques are viable solutions for a mission ground segment, however a processing overhead is introduced. In particular, processing time overhead by the SDN and Router components were measured for both TM and TC data and, whilst the overhead increased by a factor of 1.15 for TM and 1.34 for TC, this was on the order of milliseconds. Comparatively, the processing time of the space-to-ground data link software components (SCOS 2000 and GSTVi) account for >98.6% of the overall processing overhead. For these components, the overhead increased by a factor of 2.7 for TM and 1.39 for TC. These values are considered only indications as clearly there are dependencies on, for example, the mission control system software used and likely also the sizing of the network and number of participating hosts (servers, clients). Nevertheless, the concept has been proven and opens the opportunity to flexible, secure networking concepts based on IP based communication. In particular, this could lend itself to distributed operations concepts whereby, for example, remote/distributed operations centres (or simply remote workers) located with different LAN/WAN connectivity as part a ground segment utilise a common space-to-ground data link, which may interface also with an extended local space segment network (for example, on-board the International Space Station). Different operations centres may have responsibility for or access to different subsystems in different space segment networks and with SDN/OpenFlow associated rules and resulting fine-grained access controls are flexible and reconfigurable. Such a concept is illustrated in Fig. 5.

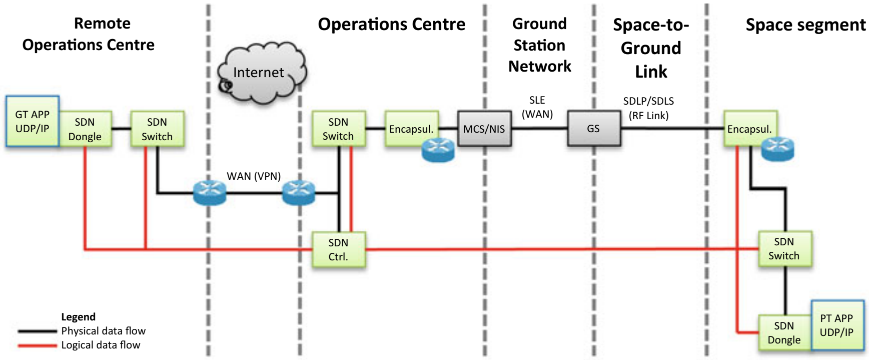


Fig. 5 Possible SDN supported distributed operations concept

Such a distributed operations concept would then be supported by multiple authentication and secure communication layers, i.e. classical VPN across control centres, routing tables, SDN Controllers and switches at local network level in all sites and SDN Dongle authenticating local hosts in all local networks. The communication flow follows the sequence of events mentioned in the previous Sect. 3.1. The option to sign IP datagrams as mentioned in the previous Sect. 3.2.1, presents an alternative means of authentication across ground and space for specific subsystems of local networks even if there is no authentication option implemented at the data link level. The scenario could theoretically also be extended to inter-space segment links.

4.2 IP, DTLS and IPSec Over SDLP/SDLS

As expected, IP over SDLP worked well (it has been implemented in flight e.g. in the Space Communications and Navigation (SCaN) Testbed onboard the International Space Station [18]). As a progression, it was demonstrated that this encapsulated network layer can also be secured using IPSec and that this works over both SDLP and SDLS (implying a double layer of encryption). Further, the DTLS implementation was also successful and supported a security layer option above the network layer. The layered protocols permit flexibility and varying levels of protection across those layers depending on mission specific needs.

4.3 Lightweight Encryption Algorithm Performance Testing

Testing the suitability of the candidate algorithms was approached through measuring the overhead computation time and any drop in throughput performance over the space-to-ground data link. The result of this testing utilising the testbed is depicted in Fig. 6. The indicated time is for transmission of 10 KB data files through TC and 100 KB files through TM. As shown, it was not possible to determine any significant result as the standard deviation was less than one percent, well below the assumed measurement error. Since plenty of factors influence the efficiency of the end-to-end spacelink, in particular the relatively low throughput of the spacelink, processes of SCOS 2000 and GSTVi, the non-real-time operating system, virtualization, etc. it was realised that any bottleneck is not as a result of an encryption algorithm computation overhead. That is, the testbed does not allow to observe an influence of differing computation complexity of encryption algorithms on end-to-end transmission times.

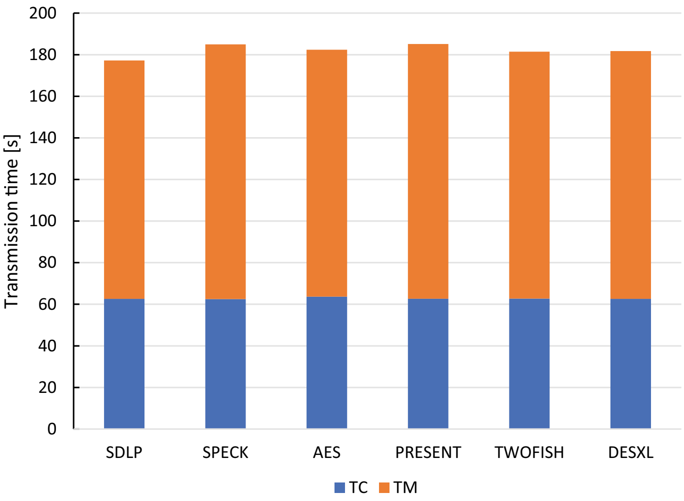


Fig. 6 End-to-end transmission time measured utilising different algorithms

In order to still perform a comparative test, computation was performed utilising a dedicated application which ran each algorithm a set number of times on a local server, outside of the SEC_LAB testbed. The throughput of the local server was approximated and calculated in the follow way:

$$AlgTh = \frac{RoundCnt * DataUnitSize}{RunTime}$$

where:

- AlgTh is a single algorithm throughput, being calculated expressed in number of bytes per second,
- RoundCnt is a number of iterations of encryption operations under a run of a single test,
- DataUnitSize is a size of elementary data unit, being processed by encryption algorithms expressed in number of bytes. Typically 8 or 16 depending on algorithm type (64 or 128 bit block) and,
- RunTime is the time of execution.

Constantly repeating a process (a RoundCnt number of times) simulates work on a big block of data. It also minimizes a participation of other processes other than the test in utilisation of the CPU (a single core). It also forces the CPU to work at its highest clock speed.

The maximum throughput of encryption operations were then approximated and calculated based on the known computational power of the local server CPU and a hypothetical space-based onboard CPU and the algorithm throughput calculated above, as follows:

$$SpaceTh = \frac{SpaceMIPS}{LocalMIPS} * AlgTh * 8$$

where:

- SpaceTh is a single algorithm throughput of the hypothetical space CPU, expressed in number of bits per second,
- SpaceMIPS is the throughput of a single core of a space processor (CPU), expressed in MIPS (Million Instructions Per Second). For the purposes of the test, a random example based on an actual unit with 400 MIPS was selected,
- LocalMIPS is the throughput of a single core of a processor (CPU) used for tests, expressed in MIPS. For the purposes of the test, a set of random Intel (i5, i6, i7), Pentium and Raspberry Pi processors were utilised.
- AlgTh is the above calculated throughput of an algorithm (see above),
- And the result is multiplied by 8 to express it in bits per second.

The results of this testing are summarised in Fig. 7.

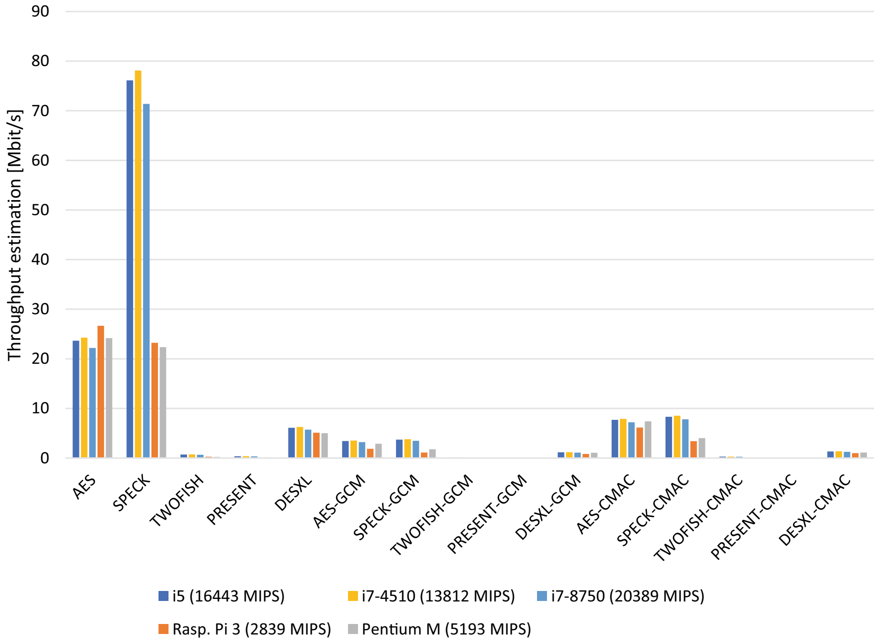


Fig. 7 Throughput estimation of various CPU and various algorithm implementations

These results indicate an order of algorithms from fastest to slowest of SPECK, AES, DESXL, TWOFISH, PRESENT. Notably, the AES algorithm used for the test was system optimized and partially written in the assembler whereas all others were written in C language. Taking this it into account SPECK appears significantly faster. These comparisons are somewhat less significant however when accounting for GCM and CMAC modes of operation. For the TWOFISH and PRESENT algorithms working in GCM or CMAC modes the throughput is low enough (40–200 Kbit/s), that it is almost invisible on the above chart. The results indicate that for an onboard processing unit, optimization for particular algorithms can be an important factor.

In addition, tests were performed by varying the payload data size, again across the different algorithm, CPU and mode implementations. For the sake of brevity, the full results are not reproduced here. In summary, for all conditions it was observed that along with growth of data sizes the throughput increase up to a saturation at approximately 1024 bytes. This indicates that typical telemetry frames (~1072 bytes) have a near optimal size for such cryptographic operations, but for telecommands (~208 bytes) the sizing is less optimal (~15–18% less than the optimal result).

Clearly, these results are indicative and a more precise measurement of onboard CPU throughput estimation would require execution of such tests in a fully representative environment, with the real hardware in the loop.

4.4 Disruption Performance Testing

Testing the stability of the candidate algorithms was approached through injecting deliberate data transmission errors with varying disruption probabilities, across the different protocol, algorithm, and mode implementations. For the sake of brevity, the full results are not reproduced here. An example of the % packet overhead (re-send) for TM data for IP over SDLS, during varying probability of disrupted bits (out of a total 120,000) is depicted in the below Fig. 8.

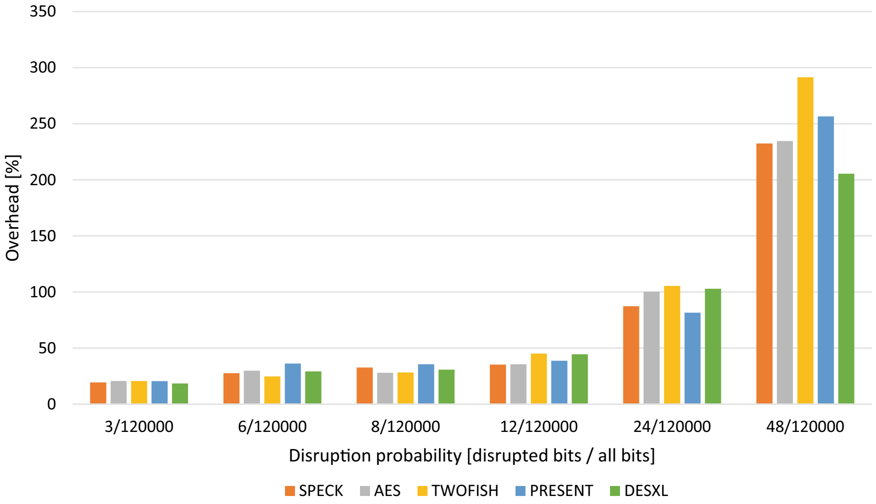


Fig. 8 TM packet overhead for different algorithms and varying disruption probability

There was no observed correlation between encryption algorithms and transmitted data overhead differences affected by spacelink data transmission disruptions. It is worth noting that there is a separation between transmission control and security, i.e. the security layer typically works on an error-free frame, and therefore a disruption test comparing cryptographic algorithms is not entirely realistic, but nevertheless of interest. As expected, there is an exponential rise of packet overhead according to the rise of disruption level. In the case of SDLP/SDLS the probability of disrupted bits for the research was in the range between 8/120000 and 60/120000 for TC and between 3/120000 and 48/120000 for TM. Below the lower boundary the overhead of disruptions was hard to observe. Above the upper boundary the data overhead was above 100% and data transmission was considered as impossible. In specific circumstances the data transmission might be possible, such as in variable disruptions in time. Otherwise during constant disruption rates, if all the SDLP/SDLS frames were disrupted even by single bit, the overall data transmission was impossible.

In the case of IPsec over SDLP/SDLS the high disruption boundary was lowered to the level of 5/120000 for TC and TM because of the additional authentication phase which is very sensitive to any IP datagram loss and also introduces significant overheads upon retries. Above this disruption boundary the IPsec authentication was impossible, due to the necessity of starting the entire process from the beginning and entering an infinite loop.

In case of DTLS over SDLP/SDLS, the high boundary was slightly higher than that for IPsec, at 12/120000. Above this boundary, similarly as in case of IPsec, the authentication phase was impossible to be finished.

The results indicate, as expected, that use of particular terrestrial protocols can face stability issues when subject to disruption, in particular for those with handshake operations which also introduce additional round-trips. For this reason utilisation of such protocols is likely only suitable for shorter One Way Light Times (OWLT), such as those of Low Earth Orbit (LEO) missions, potentially extending to Medium Earth Orbit (MEO) or Geostationary Earth Orbit (GEO) missions provided there is a sufficiently performant data link and throughput. Due to the smaller frame size, TC data was observed as less susceptible to disruption than TM data.

4.5 Results of SDLS EP Validation

The full set of CCSDS SDLS Extended Procedures were implemented and executed. During the implementation, various minor discrepancies and inconsistencies in the draft standard were detected and reported back to the CCSDS working group for incorporation.

One finding was that the majority of KM and SA Management procedures do not contain explicit response steps, for reception of confirmation of command execution, for example related to state changes in the onboard security database. As a result, it was considered there could be a risk of arriving in a non-synchronized state across ground and space. As mitigation, a new Key Inventory service procedure was introduced as an additional KM procedure in the standard. This procedure enables the Initiator to have confirmation of the reception of keys and corresponding key states as a verification for the key management procedures. Combined with handling at other protocol layers (e.g. Communications Operations Procedure-1 (COP-1) [19] protocol to ensure clean frame delivery), the Frame Security Report (FSR), which may be sampled by the Recipient for each received SDLS protected frame and the Dump (Security) Log M&C procedure, the content of which may be configurable on a mission-specific basis, the risk was considered adequately mitigated. Nevertheless, it is considered that such mitigation should be ensured through appropriate operational procedures and the mission specific strategy for error handling and associated operations of an SDLS EP implementation must be defined. These considerations are to be included in an associated CCSDS Green Book on the topic. Extensions to the KM and SA Management procedures with direct responses were developed and validated on the SEC_LAB, and appear viable. As an additional potential risk, it was considered that depending on the SDLS EP Log implementation, it may be possible for an attacker to perform a Denial of Service by saturating the log with, for example, frame rejection errors, thus compromising a possible audit trail. It was considered that this also should be a mission-specific concern to address, through implementation and/or operational procedure.

As an additional research item, the suitability of utilising SDLS EP equivalent procedures for management of IPSec parameters was tested. It was found that in fact the majority of the procedures could be used, with some exceptions and limitations due to differences in managed parameters/fields between the two standards and that this may be interesting for harmonising related operational procedures where potentially different communication layer security management is required within a single system.

5 Conclusions

The developed SEC_LAB capability provides a representative, flexible testbed for research and testing of secure space-to-ground communication solutions. As a result of the development and testing work conducted, several insights have been derived in the context of:

i. reusability of common terrestrial network protocols and techniques, in particular in support of future advanced networking scenarios and IP-based communication, for low OWLT/LEO/MEO missions;
ii. suitability and stability of lightweight encryption algorithms for space system applications and;
iii. the validation and use of the CCSDS SDLS EP standard.

The viability of the tested solutions with real ground segment mission infrastructure is proven. Further, the SEC_LAB has already been utilised to generate test reference data sets for the verification of an SDLS EP implementation for a future Sentinel mission. This is made possible due to the flexibility through configuration of the testbed, to match the managed parameter values for mission implementations. Thereby the goal of the SEC_LAB in terms of providing a test environment for securing space-to-ground communications and de-risking implementations for future missions has been fully achieved.

As envisaged future work, the SEC_LAB can be refactored through the replacement of the SCOS2000 control system with the next generation of EGS-CC based mission control infrastructure [20]. The verification of SDLS/EP would then be beneficial for future missions adopting both the standards and EGS-CC based systems. In addition, ESA already envisage the re-use of the SEC_LAB concept for testing of emerging secure communication standards, in particular the Bundle Protocol Security standard [21] for Delay Tolerant Networking (DTN) scenarios.

References

1. CCSDS (2015) Space data link security protocol. Recommended Standard, Issue 1, CCSDS 355.0-B-1, Blue Book. Washington, D.C., USA: CCSDS, September 2015
2. CCSDS (2015) TM space data link protocol. Recommended Standard, Issue 2 (Blue Book), CCSDS 132.0-B-2. Washington, DC, USA: CCSDS, September 2015
3. CCSDS (2015) TC space data link protocol. Recommended Standard, CCSDS 232.0-B-3, Blue Book, September 2015
4. CCSDS (2020) Space data link security protocol—extended procedures. Recommended Standard, CCSDS 355.1-B-1, Blue Book. Washington, DC, USA: CCSDS, February 2020
5. The Internet Society (2005) Security architecture for IP, RFC-4301 December 2005. https://tools.ietf.org/html/rfc4301. Accessed 08 March 2021
6. ESA (2021) MICONYS mission control system. http://www.esa.int/Enabling_Support/Operations/Ground_Systems_Engineering/MICONYS. Accessed 08 March 2021
7. ESA (2021) TEVALIS ground systems test and validation infrastructure. https://www.esa.int/Enabling_Support/Operations/Ground_Systems_Engineering/TEVALIS. Accessed 08 March 2021
8. Internet Engineering Task Force (2012) Datagram transport layer security version 1.2. RFC-6347, January 2012. https://tools.ietf.org/html/rfc6347. Accessed 08 March 2021
9. Internet Protocol (1981) Darpa internet program, protocol specification. RFC-791. California: Information Sciences Institute University of Southern California, Alington, September 1981. http://www.rfc-base.org/rfc-791.html. Accessed 08 March 2021
10. National Institute of Standards and Technology (2001) Federal information processing standards publications, advanced encryption Standard, November 2001. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf. Accessed 08 March 2021
11. Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L (2013) NSA, the Simon and speck families of lightweight block ciphers, June 2013
12. Schneier B, Kelsey J, Whiting D, Wagner D, Hall C, Ferguson N (1998) Twofish: a 128-bit block cipher
13. Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJB, Seurin Y, Vikkelsoe C (2007) PRESENT: an ultra-lightweight block cipher
14. Kitsos P, Sklavos N, Parousi M, Skodras AN (2010) A comparative study of hardware architectures for lightweight block ciphers. Computer Science, Hellenic Open University
15. CCSDS (2012) IP over CCSDS space links. Recommended Standard, CCSDS 702.1-B-1, Blue Book, September 2012

16. CCSDS (2009) Encapsulation service. Recommended Standard, CCSDS 133.1-B-2, Blue Book, October 2009
17. StrongSwan (2021) OpenSource IPSec-based VPN solution. https://www.strongswan.org/. Accessed 08 March 2021
18. Brooks D et al (2021) In-space networking on NASA's SCaN Testbed. AIAA. https://ntrs.nasa.gov/api/citations/20170001296/downloads/20170001296.pdf. Accessed 08 March 2021
19. CCSDS (2010) Communications operation procedure-1. Recommended Standard, CCSDS 232.1-B-2, Washington, DC, USA: CCSDS, September 2010
20. Pecchioli M, Walsh A (2021) The EGS-CC based mission control infrastructure at ESOC. https://indico.esa.int/event/180/contributions/1367/attachments/1248/1473/0930_Pecchioli_-_Paper.pdf. Accessed 08 March 2021
21. Birrane E, McKeever K (2021) Bundle protocol security specification draft-ietf-dtn-bpsec-27, February 2021. https://tools.ietf.org/html/draft-ietf-dtn-bpsec-27. Accessed 08 March 2021

Flight Operations

ISS Payload Operations Training Throughout the COVID-19 Pandemic: Impacts, Opportunities and Solutions

Craig Cruzen and Jeff Montgomery

Abstract The onset of the COVID-19 pandemic brought a dramatic and rapid transformation to almost every aspect of humanity. The world's space agencies and their missions were not immune to the wide-sweeping changes. One discipline principally affected was mission operations and the various groups supporting that function. Mission support teams, especially for complex and crewed missions like the International Space Station (ISS) were forced to rethink how and where control center staff performed their vital work. Operations training—an essential element to mission ops, had unique hurdles to overcome. Operations training is responsible for preparing astronaut crews for their missions, training and certifying flight controllers, as well as ensuring that new team members are ready to join their colleagues. Every element of training was impacted during the pandemic. From orientation and introductory classes for new controllers, simulations, and advanced lessons, On the Job Training (OJT) and final evaluations; all aspects faced challenges. Trainers at NASA's Marshall Space Flight Center in Huntsville, Alabama were forced to become more efficient with trainees and resources to continue supporting ISS payload operations. The pandemic arrived in the USA in March 2020. Immediately, NASA mandated that the support for ISS real-time operations was critical. As a result, physical access to key facilities was restricted. Trainers and trainees had to quickly shift to 100% remote learning. In the short term, this was not a problem. However, instructors discovered lessons they were accustomed to delivering in a classroom environment often did not translate to remote teaching. Another hurdle to operations training was the mandate that all simulations could only be held remotely. The logistics of even small simulations proved to be challenging due to Information Technology (IT) restrictions and public internet limitations. With simulations essentially halted, as well as the restrictions on most OJT, trainees were essentially stopped in their advancement towards

C. Cruzen (✉)
ISS Payload Operations Director, NASA Marshall Space Flight Center, Huntsville, AL, USA
e-mail: craig.cruzen@nasa.gov

J. Montgomery
Space Systems Operations Training Team Lead, Teledyne Brown Engineering, Huntsville, AL, USA
e-mail: jeff.montgomery-1@nasa.gov

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_21

certification. Once limitations were identified, trainers prioritized new options. Transitioning to all electronic learning materials was a relatively easy fix. Teaching to large groups took additional shifts in the training paradigm. Methods for preparing astronauts for their missions were revised. Simulation supervisors found efficient techniques to provide realistic training experiences. Communication and coordination with management was essential. In every case, the payload operations instructors found novel solutions to all functions listed. This paper discusses the factors and solutions payloads operations trainers found to keep scientific research on the ISS flying forward to mission success.

Keywords ISS · MSFC · Operations · Payloads · Training · COVID · Pandemic

Acronyms/Abbreviations

HOSC	Huntsville Operations Support Center
ISS	International Space Station
IT	Information Technology
MSFC	Marshall Space Flight Center
NASA	National Aeronautics and Space Administration
OJT	On the Job Training
POIC	Payload Operations and Integration Center
PPE	Personal Protection Equipment
Remote IVoDS	Remote Internet Voice Distribution System
SPG	Sim Planning Group
VPN	Virtual Private Network

1 Introduction

The onset and the protracted duration of the COVID-19 pandemic brought about a dramatic and rapid transformation to almost every aspect of humanity. The world's space agencies and their missions were not immune to the unpredictable and wide-sweeping changes. One discipline principally affected was mission operations and the various groups supporting that critical function. Mission support teams, especially for complex and crewed missions like the ISS, were forced to rapidly rethink how and where control center staff performed their vital work (Fig. 1).

Flight controllers and engineers at NASA's Marshall Space Flight Center (MSFC) in Huntsville, Alabama are responsible for operating the vast array of scientific experiments [1]. They are also responsible for payload support equipment, including data and video systems onboard the ISS and well as a complex ground control and distribution network. This team has been supporting 24 × 7 real-time operation

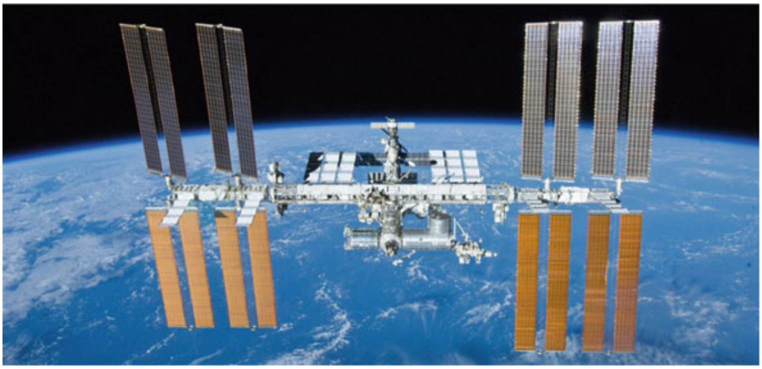


Fig. 1 The International Space Station

of ISS payload operations since 2001. Flight controllers, engineers, and operations professionals at the Payload Operations and Integration Center (POIC) (see Fig. 2) work in close coordination with the Mission Control Center in Houston, as well as international partners around the globe to support ongoing scientific utilization of the ISS [2].

The full impact of the COVID-19 pandemic arrived in the United States in early March 2020. At that time, NASA mandated that support for ISS real-time operations was critical. Physical access to control centers was restricted to only those people directly supporting operations. Thus, the use of on-site NASA facilities and any non-critical operations activities was restricted [3] (Fig. 3).

Training astronauts for their missions, providing recurring proficiency and currency material to flight controllers, as well as ensuring that new team members are ready to join the console team are important element to mission operations. As a result of COVID restrictions, astronauts, trainees, and trainers had unique hurdles to overcome during the initial months of the pandemic. Every element of

Fig. 2 NASA's ISS Payload Operations Center in Huntsville, Alabama

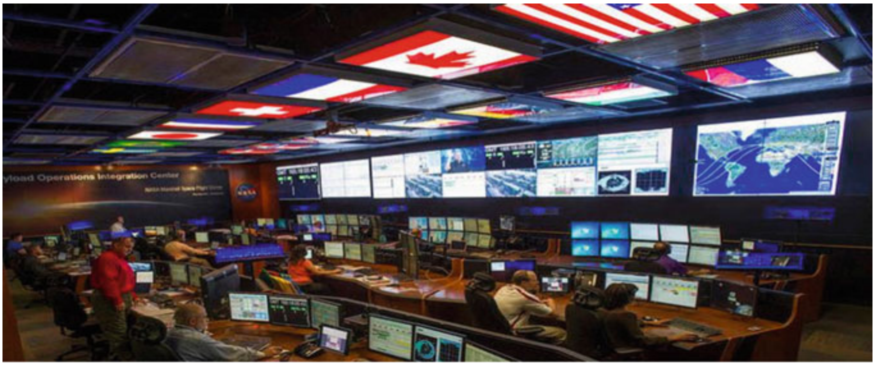


Fig. 3 ISS payload operations center prior to the COVID-19 pandemic (NASA social media photo)

training was impacted including many specialized training facilities. In addition to these limitations, social distancing constraints were implemented and thus trainer-to-trainee, face-to-face interaction were not authorized. From orientation and introductory classes, simulations, advanced lessons, On the Job Training (OJT), and final evaluations;—faced new challenges. NASA operations trainers were forced to rethink, retool, and become more efficient with trainees and training resources in order to continue supporting operations on the ISS.

After a short reaction period, NASA began orchestrating how mission training would be safely restarted. One critical need was astronaut training. ISS mission schedules were impacted, and astronaut preflight training was reengineered to ensure the virus was not carried to the ISS. Likewise, ground controller training adaptation was a major focus. Once limitations were identified and documented, ISS trainers at MSFC discussed, proposed, and prioritized new options and techniques. Transitioning to all electronic learning materials and certifications was a relatively easy fix. Teaching courses to large and diverse groups took additional shifts in the NASA training paradigm. Likewise, methods and procedures for preparing astronauts for their missions were forced to be revised. Simulation supervisors and their teams were also forced to find more efficient techniques to provide realistic training experiences. Communication to, and approval from management was essential. In every case, the ISS payload operations instructors found novel and efficient solutions to all functions listed above. This paper will discuss the factors and solutions that trainers found to keep scientific research on the ISS flying forward to mission success.

2 Impacts and Barriers to Operations Training

2.1 March 2020: Operations with Social Distancing and Remote Ops

Upon declaration of the international COVID-19 pandemic in March of 2020, all US government facilities including NASA field centers were restricted to only mission critical personnel. In the case of NASA and ISS operations, this meant that only astronauts, flight controllers, and engineers were authorized to go on site. In short, only those who were essential for ensuring safe mission continuation [4]. Initially, this did not include trainers who ran simulators, nor did it include trainees who were learning how to become flight controllers. All other NASA employees were instructed to stay at home and work remotely with NASA provided Information Technology (IT) assets. Fortunately, NASA had been working to incorporate remote IT systems for some time, so there were very few problems with the quick transition to telework status.

ISS operations personnel at MSFC were only permitted to go to the Huntsville Operations Support Center (HOSC). Brief visits to offices to retrieve critical items like headsets, computers, or coffee mugs were allowed. Social distancing was instituted in the control rooms. Where previously operators had sat near each other; now they were separated by at least 6 feet between people, and in most cases 12 feet or more (see Fig. 4). This was accomplished by leaving open seats in between console positions, moving some controllers to alternate rooms, and still other operated remotely from their homes thru secure Virtual Private Network (VPN) technology. This use of primary flight control in such a widely dispersed, remote operation at non-government facilities was truly a first for human spaceflight operations [3].

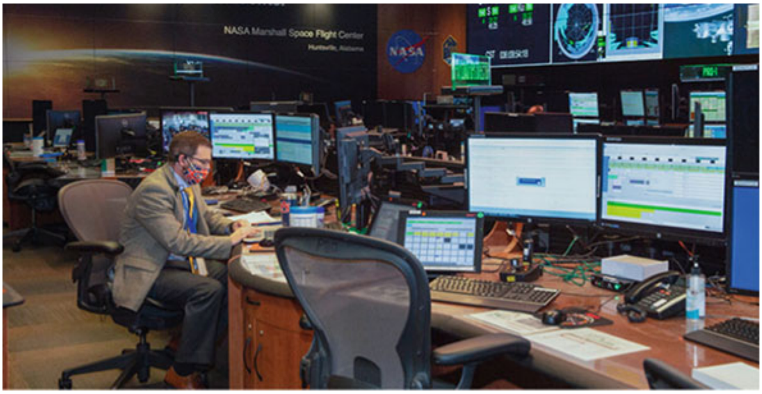


Fig. 4 ISS Payload Operations Center with pandemic protocols and social distancing (public release approved)

Pandemic protocols were documented and compulsory for control teams. These included sanitizing each workstation at the start and end of every shift. NASA provided Personal Protection Equipment (PPE) such as face masks, hand sanitizer and disinfectant wipes. Personnel were always required to stay socially distanced from each other during their shifts. If anyone was potentially exposed to the virus, they were required to quarantine at home for 14 days to ensure there was no reinfection within the control center. This consideration put extreme pressure on the workforce as sometimes these voluntary quarantines weren't discovered until the last minute which led to a rush to find replacements.

These same pandemic protocols caused NASA training to quickly shift to 100% remote learning. As mentioned above, in the first few weeks this was not a problem since NASA had prepared IT resources that facilitated remote access to classes, documents, and teleconferences. However, instructors quickly discovered that often the lessons they were accustomed to delivering in a classroom environment did not translate well to remote learning. The lack of non-verbal communication with the class became a barrier to instruction. Trainees early in their training flows were the hardest hit by these restrictions. New employee orientations were hindered by newly credentialed employees struggling to be assigned computers, badges and learning onboarding steps. Another hurdle was the mandate that simulations could only be held remotely (March thru May 2020). The logistics of planning and executing even small simulations proved to be challenging, almost impossible, due to IT restrictions and public internet limitations. With integrated simulations halted, as well as prohibitions on most OJT, trainees were essentially stopped in their advancement towards certification.

Prior to the pandemic, a significant portion of ISS flight controller training, especially for payload operations, involved traveling to other NASA centers to utilize unique training facilities at JSC and international sites. Starting in March of 2020, that was no longer possible. In most cases trainees were forced to make do with reviewing pictures, videos and electronic procedures without having the opportunity to put their hands on the hardware. This may have been the most significant barrier to training during the pandemic. In some cases, remote video training was made available to new flight controllers, but it remains the exception [3].

An additional barrier was the formal tracking of curriculum progress (i.e. Individual Training Plans). Even though NASA had been transitioning to electronic documents for years, some organizations were still relying on "pen and ink" documents to formalize training completion. The logistics of implementing secure and controlled e-signature documents in a standardized, NASA approved format had to be addressed.

2.2 Spring/Summer 2020: Operations Training Begins to Adapt

As listed above, the barriers to operations training were multi-faceted and daunting. This is reflected in Fig. 5 which shows the decrease in the number of certifications from January thru May 2020.

After impacts to existing training plans were identified, the next step was to determine ways to get training restarted. Following a month of standing down training due to COVID, it became obvious this was going to be a long-term issue. As a management team, we realized it was time to move from reaction mode to planning mode to begin to get critical training executed. This took on increased importance as certified cadre members began to feel the impacts of COVID. The ISS Program could not afford get to a point where there were not enough certified personnel available to continue 24 × 7 operations.

In April of 2020, the MSFC Training Panel, made up of management representatives and senior trainers, held an important strategy meeting to get all Training Leads together to identify areas for training emphasis. This included:

- Identifying those trainees who were closest to certification. These trainees would be given priority for the remainder of their training to get them certified as quickly as possible. The remaining trainees were then grouped based on how complete their Individual Training Plans were.
- Determining how/if existing training could be done remotely.
- Identifying training that must be done in person. Rationale then would be developed on why it must be done in person along with the steps that would be taken to meet NASA COVID restrictions such as social distancing, PPE, cleaning before and after sessions.

Priorities for flight control training were identified and set as follows:

- **Highest Priority: Phase 2 Training**—finishing certifications for console positions where training was approximately 25% complete or more. These trainees needed the following resources:

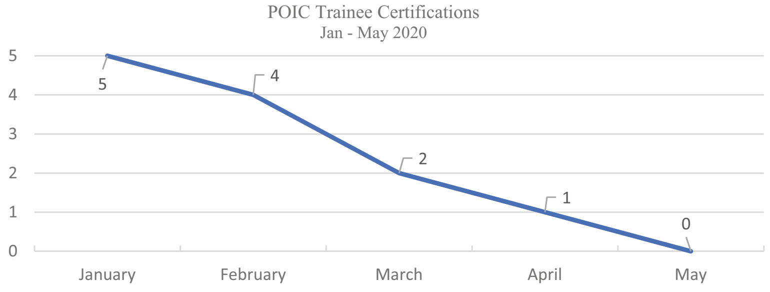


Fig. 5 POIC trainee certifications: January–May 2020

  - Remote Classroom Training
  - Roundtables with mentors and managers
  - Observation OJT
  - Performance OJT.

- **Second Priority: Phase 1 Training**—Initial training for new employees. These trainees needed the following resources:

  - Payload Academy (8-week curriculum to give new hires a common base of NASA and ISS operations fundamentals)
  - Orientation for New Employees
  - After completing these, the trainees moved into the Phase 2 category.

- **Third Priority: Phase 3 Training**—currency and proficiency training for certified cadre. To maintain proficiency and currency requirements, these resources were needed:

  - ISS Emergency Proficiency Training
  - Flight Controller Critical Skills Proficiency Training
  - Proficiency Simulations.

These priorities addressed the need to get trainees who were approximately three months or less from certification through the remainder of their training flow so they could help the certified cadre staff real time operations. This also gave time to adapt Phase 1 training to a telework environment to allow us to begin training new hires and prepare them for Phase 2 training. Finally, setting Phase 3 (certified proficiency) training as a lower priority gave our training team time to modify higher priority training (Phases 2 & 1) to be executed within NASA COVID guidelines while still meeting training objectives. Trainers were then able to use lessons learned from modifying Phase 2 and Phase 1 to make updates to Phase 3 training efficiently. Once the training priorities were identified, the training team began work on the Phase 2 curriculum.

Classroom Based Training: MSFC's Training Integration team worked with discipline training leads to identify ways to make classroom-based training available using Microsoft Teams software [5] in a collaborative environment. MSFC was fortunate to have a professional Instructional Designer with experience tailoring and delivering remote training in a college environment. She used her experience to write a guide on how to facilitate learning remotely. This was very helpful for our instructors and allowed them to master nuances of remote instruction. Training Integration as a group optimized existing lesson presentations into a format better suited for remote delivery. This aided the transition from face-to-face training to remote learning.

As instructors gained experience with Microsoft Teams, they were able to incorporate some of the software features into the lessons to increase student participation. This included using the "raise hand" feature for students asking questions, using the chat feature as collaboration tool during classes and using the participant list as a virtual class roster. Video was also used in some classes to provide a virtual face-to-face environment. Instructors also used the "sharing" feature to show students documents, presentations, and videos.

Another key training tool that received a "make over" were roundtables. These small group meetings are used for both training and knowledge assessment. The format for roundtables prior to COVID-19 was normally face-to-face with either a single student and a trainer or in some cases several students and trainers. Because face-to-face was not an option, roundtables were adapted to use the Microsoft Teams video and chat functions.

Observation OJT is an effective way for a trainee to watch a certified flight controller perform the job they are training for. This is normally done in the control center so the trainees can listen in on voice loops, monitor displays, hear over the wall conversations, and ask questions to their certified counterparts. Some of our non-commanding positions adapted to a format of using remote observation OJT. Trainees would use VPN to log into console tools to monitor operations during real time. Using a product called Remote Internet Voice Distribution System (IVoDs) software, trainees were able to listen to NASA operational voice loops from home. Microsoft Teams was available to serve as a virtual chat between the certified cadre position and the trainee. This allowed questions to be asked both ways. While face-to-face feedback is always best, this remote OJT mode provided an effective means to learn details of the position. Other positions adopted a modified version of face-to-face training to account for social distance requirements. This included having some positions located outside of the primary control room in various support rooms within the control center. This allowed additional personnel without violating social distancing constraints.

Performance OJT is the next level of training where the trainee performs the job while being supervised by a certified flight controller counterpart. Like observation OJT, this is typically done in the control center so the trainees can interact on voice loops, monitor displays, hear over the wall conversations and be in the "hot seat" while their certified counterpart evaluates their performance. During the pandemic, social distancing required us to make changes to how performance OJT occurred. NASA COVID restrictions reduced the number of flight controllers who could be in the control room. Performance OJT is a critical gate in the overall certification process. Obviously, it is important for trainers to be able to watch their trainees in the real-time environment to demonstrate readiness to certify. Because of this, staffing restriction changes were made to ensure we could maintain social distancing while allowing some trainees in the control center for performance OJT. To accommodate other participants, some flight control positions moved operations and associated OJT personnel to support rooms located within the HOSC. This provided access to required tools to perform the job but opened a new challenge of not operating in the same room. These challenges included reduced cognizance of overall operations, missing out on over the wall conversations, and not being able to use visual feedback when communicating with others.

During the pandemic, it was apparent that we needed to not just maintain regular communication between trainers and trainees, but also enhance it. The first task was

Table 1 Microsoft Teams—training implementations

Trainer to trainee one on one meetings	Classroom lessons
Roundtable sessions	Oral evaluations
Simulation following	Real-time flight following
Sim planning group meetings	Training panel meetings

to replace in-person meetings with virtual meetings. Once again, Microsoft Teams proved to be an indispensable tool to facilitate virtual meetings. This product has been used for one-on-one meetings, branch level meetings, as well as board/panel level meetings. The audiovisual capabilities, screen sharing, and chat features has made it easy for our training teams to transition from face-to-face meetings to a virtual meeting environment. Table 1 lists several of the ways that MSFC trainers used Microsoft Teams to augment training.

The Microsoft Teams chat capabilities proved useful as an alternate communication path to email and telephone. This has permitted trainers and trainees to continue discussions and informal training that would normally have occurred in the office environment. This kind of person-to-person contact has been critical in keeping training moving forward to completion during the pandemic. It should also be noted that the "GIF" feature in Microsoft Teams helped to build camaraderie.

Another key to being able to successfully restart training was having a means of communicating obstacles and proposed solutions to senior managers. About two years before the pandemic, the MSFC Training Panel was chartered and established. The Training Panel's role is to meet the training needs required by all MSFC Payload and Mission Operations Division (PMOD). This includes PMOD users, payload developers and the crew. Table 2 lists the Training Panel's functions.

The MSFC Training Panel took on a vital role in resuming training after the start of the pandemic. The panel coordinated with all Training Leads to identify issues and challenges to restarting training. It then set up a small group to define priorities and needs to restart training both in person when required and remotely when possible. The Training Panel provided this report to senior management to help define direction to Training Leads on how required in person training could be resumed while meeting COVID restrictions. The Training Panel also examined alternative ways to deliver existing training remotely. Using experience in delivering

Table 2 MSFC Payload and Mission Operations Division Training Panel primary functions

Ensuring completeness, quality and conduct of all training activities	Develop, review, instruct and implement training products and processes
Perform internal and external coordination to ensure required training objectives are met	Ensuring training policies, processes, and procedures are consistent across disciplines
Provide concurrence on training products prior to formal baseline or external coordination	Establish the baseline for and controls subsequent changes to training products developed in support of mission operations

Table 3 Microsoft Teams virtual classroom capabilities

Desktop sharing by the instructor to show presentations, videos and voice loop audio	Using the raise hand feature to allow students to ask questions
Sharing of video to provide a chance for new trainees to meet one another	Using the chat feature to provide discussion and alternate way to ask/respond to questions

Table 4 POIC prime guidelines for conducting in-person training

Obtain management approval for the event	Identify the names of the person(s) required to be on site along with date, time and location
Report names to management team for contract tracing	Coordinate availability of location for training
Follow NASA guidelines for social distancing and personal protective equipment	Sanitize training areas before and after use

remote classes at the university level, our instructional designers suggested using Microsoft Teams to create a virtual classroom. This virtual classroom provides the capabilities listed in Table 3.

Training Integration also developed a list of tips and techniques that instructors can use to facilitate remote instruction. The team also worked with instructors to update presentations as needed to be more Microsoft Teams friendly.

Training Leads for each discipline played an important role in successfully resuming training as the pandemic continued. They identified key components of their training flows to designate what must be done in person versus what could be done remotely. They worked with NASA and contractor management teams to ensure in person training could and was conducted following MSFC COVID guidelines. A partial list of the guidelines is presented in Table 4.

The overall team's dedication and adherence to the NASA COVID guidelines enabled us to not only safely resume training, but to also continue 24 × 7 × 365 ISS Payload operations. This is reflected in the gradual, yet impressive increase in trainee certifications from May thru December of 2020 in Fig. 6. Note that the drop off in December is typical and expected due to the holiday season.

3 Impacts and Solutions to Flight Control Team Simulation Training

Shortly after NASA implemented its COVID-19 pandemic response, simulations were identified as a critical training need to help trainees be ready for performance OJT and eventual certification. The Sim Team, under Training Panel direction, reconvened the weekly Sim Planning Group meeting remotely using Microsoft Teams. The Sim Planning Group (SPG) is a forum chaired by the Sim Team that meets weekly to plan and script simulation-based training.

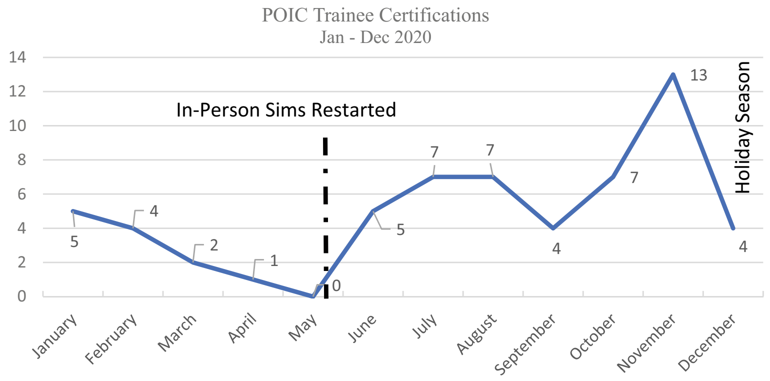


Fig. 6 POIC certifications January–December 2020

In late March of 2020, the SPG met to discuss how/if they could conduct a simulation remotely. Payload Control Area 2 (PCA-2) is the back up control room for POIC. This room is configured similarly to the primary control room (PCA-1) and is normally used for simulation training. It is important to note that early in the COVID response, PCA-2 was maintained as a hot back up in the event PCA-1 needed to be disinfected. Because PCA-2 was not available to support simulations in March and April, the Sim Team and Training Leads considered how to conduct a sim remotely. The first step was to identify limitations of being fully remote. These are listed in Table 5.

The second step was to determine what overall training objectives could be achieved in this constrained simulation environment. The following objectives were identified:

- Practice Failure, Impact, Workaround methodology in response to simulation events.
- Exercise voice loop protocol.
- Practice situational awareness.
- Use applicable Flight Rules, Standard Operating Procedures, and other mission documentation.
- Execute the simulation timeline.

Table 5 POIC limitations of fully remote simulations

Inability to have "over the wall" conversations	Lack of eye contact and other non-verbal cues
Would not have access to multiple large displays used to monitor payload systems	Would have to use remote IVoDS with a subset of normally used communication loops
Would not have access to view simulated payload operations videos	Latency and data quality issues due to different home Internet capabilities

Remote sim sessions were intended to provide a virtual integrated team environment to allow trainees to meet training objectives in the absence of simulations. We conduct a pre-brief on the main voice loop 15 min before starting the sim. The team executed a predetermined portion of a timeline. In all simulations, trainers expect the team to execute the timeline as realistically as possible, performing all coordination/communication required. Remote sims also had less voice loops, so we asked the participants to do all normal coordination (JSC, JAXA, ESA) on their own internal loop. For example, if the POIC Payload Rack Officer needs to call the JSC ISS Electrical Power position, they did so on the PRO loop. The external surrogates likewise responded on that loop.

ISS simulator servers were not used to support these sessions. This was decided because it was unreasonable for trainees to view data on a laptop monitor instead of larger and multiple console displays. This also meant trainees would not have access to Fault Summary, Exception Monitor and other software tools that increase situational awareness. Therefore, when malfunction cases were introduced, they would be given to the cadre via “green card” for telemetry items and/or by calls on the loops (crew calling down, reports to Flight Director, etc.). The overall goal was to provide trainees an opportunity to practice payload operations team skills from remote locations.

Remote sims were executed as follows:

- One hour before the scheduled start time:
  - Trainees and surrogates sign into remote voice loops.
  - Trainers and surrogates sign into Microsoft Teams meeting, used for event coordination.
- 15 min before start, Sim Sup uses the prime voice loop to announce the current sim time so all participants can set their timelines. Payload Ops Director (POD) then gives a pre-sim introduction.
- At the top of the hour, Sim Sup calls session start.
- Sim Sup, trainers, and surrogates coordinate throughout the session:
  - Following script
  - Coordinating sim events
  - Monitoring cadre performance
  - Identifying debrief items
- Perform sim debrief on the prime voice loop.

3.1 April 2020: Fully Remote Simulations

The first remote simulation occurred April 16, 2020 and the results were mixed. The trainees were able to successfully coordinate and execute nominal timeline activities. They were also able to exercise many team skills elements. However, not having simulated data was an impact to commanding position training. Without displays they had to rely on surrogate descriptions of messages and displays to determine malfunctions. Trainees were also not able to practice doing normal commanding. For non-commanding positions, they were able to achieve most planned objectives.

3.2 May 2020: On Site Simulations Resumed with Social Distancing

Due to the mixed results of attempting to execute simulations remotely, the Sim Team began to investigate what would be required to resume in person sim execution. In consultation with the Training Leads and senior management, the following requirements were identified in Table 6.

Despite having trainees spread across several rooms, the first in person simulation was deemed highly successful. The ability to flow simulation data increased the fidelity of training for all trainees. Training Leads and trainees both felt this was a significant improvement over fully remote simulation.

By the summer of 2020, as NASA COVID response became more routine, it was determined PCA-2 could be made available to support training again. While we still could not have the entire cadre of trainees in there at one time, it did allow us to colocate many positions together again. This allowed the trainees to take advantage of using team skills based on non-verbal cues and over the wall conversations. This also helped the Sim Team and surrogates since PCA-2 is equipped with video cameras so trainers can visually monitor trainee performance remotely. This also reduced extra work on the Sim Team in trying to coordinate multiple alternative rooms for each simulation.

3.3 2021: Flight Controller Certification Rates Stabilize

As of the fall of 2021, POIC continues to successfully execute simulations and training based on US government COVID-19 protocols. 2021 POIC certification

Table 6 POIC requirements to restart on site simulation training

Provide social distancing spacing for trainees, surrogates and the Sim team	Provide PPE and cleaning supplies outside designated areas
Locate other areas in the HOSC that could support a control console with an IVoDS unit since PCA-2 use was restricted	Identify walking paths for simulation participants clear from areas used by real time operations
Coordination with real time support to ensure rooms used for simulations did not impact real time operations	Since debriefs would need to be conducted over the loops identify key learning events to minimize debrief time while ensuring effectiveness

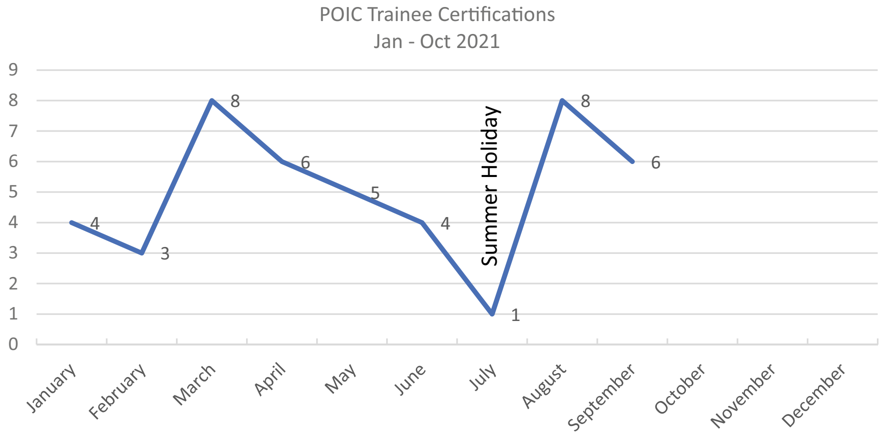


Fig. 7 POIC certifications January–October 2021

results have stabilized compared to 2020 values. As seen in Fig. 7, monthly flight controller certification rates have returned to pre-pandemic values. Note that the drop off in July is expected due to the US summer season.

4 Impacts and Changes to Astronaut Training

For mission critical astronaut training, exposure between flight crews and trainers was cut to a bare minimum (Fig. 8). As discussed above for flight controller training, virtual learning technology was key to minimize flight crew and instructor exposure. Briefing-type lessons were a perfect fit for this and likely will continue in the post-COVID world. This type of training unfortunately loses the face-to-face aspect of meeting with crew but can be a cost savings and much easier to schedule [6].

Remote learning was also successfully used with hardware demonstration lessons. In many cases, instructors sat with the hardware and taught the astronauts via video at their remote location (see Fig. 9). An obvious disadvantage was that the crew did not get to see hardware in person but did get to witness instructor handling hardware and pointing out vital information. This was not ideal learning but was a way to continue mission essential training with lower risk to COVID infection. This opened the door for options of training crew who were assigned late to a mission and may not have much training time. Lessons like this can also be provided while crew is at an international partner site.

Another version of this was “double remote” instruction. In this case, instructors tied in to training from their remote location and developed videos to show and walk through training with crew (likewise at a remote location). This helped instructors by not having to travel for training. It was not ideal as it did not usually allow for

Fig. 8 ISS Crew-1 preparing for their mission during (NASA social media photo)

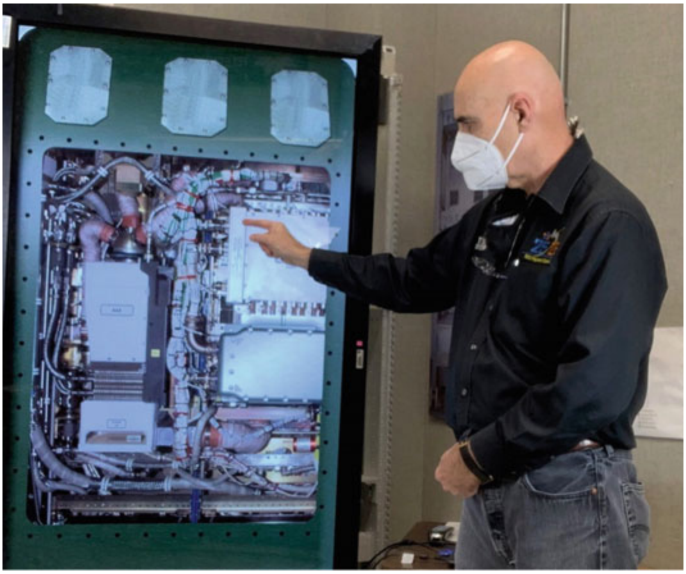


Fig. 9 ISS payload instructor presenting via video [7]

complete lessons. NASA's intent after the pandemic is to return to in person learning when safe to do so.

When in-person learning was possible and allowed, instructors still had to think about teaching from a distance, instead of "shoulder to shoulder" with crew. Some

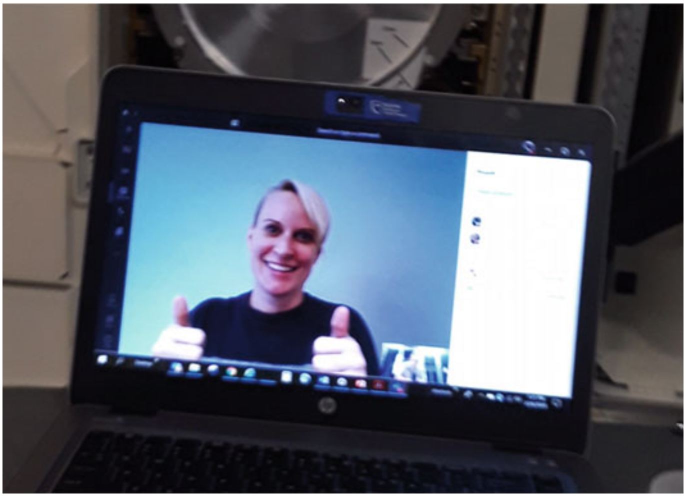


Fig. 10 Astronaut Kate Rubins successfully completes another remote ISS payload class (photo used with permission)

instructors had crew practice with hardware while they observed from a safe distance and provided guidance. This only worked for training where the operations did not include fine details, that is where the instructor had to see precisely what the crew was doing. In other cases, instructors were in a different room but observed via video, explaining detailed operations while crew was following instruction via audio. This method is very similar to how on orbit ops are done for many payloads and was a good learning experience for crew. Some variation of this may become a norm in this type of training—example: first lessons in person; follow on lessons via video/audio instruction to provide more representative of how ops will occur on board (Fig. 10).

Other factors that hampered preflight training had to do with schedule constraints. Often astronaut crew had less time for training due to quarantine requirements. Travel caused some crew to quarantine post travel, only allowing for remote learning vs. in person instruction. Yet another impact to their training involved the reduced number of people involved, and increased workload on the instructors. Prior to the pandemic, instructors and training facilitators attended almost every crew training lesson as a team. The facilitator's role was to greet the crew, assist with logistics, help the instructor stay on time, make sure any distractions were minimized, and then submit a report of the training. In the report, facilitators capture actual start/stop times to determine if we are scheduling the right amount of time, capture a list of attendees, take actions from training as well as any pertinent info from training. When the pandemic started, NASA managers minimized the number of people coming to training. This was for crew and instructor safety, no 'extra' people. So, we quickly moved from sending a facilitator to asking the instructor to take on facilitator role as well. This added a burden on the instructor as they must now manage teaching, facilitating, and fill out a report.

5 Conclusion: Lessons Learned and Efficiencies Gained

The COVID-19 pandemic brought about a dramatic and rapid transformation to space mission operations teams and their various sub-disciplines. Mission support, especially for complex and crewed missions like the ISS, were forced to rapidly rethink how and where control center staff performed their vital work.

Operations training is responsible for preparing astronaut crews for their missions, providing recurring proficiency and currency material to flight controllers, as well as ensuring that new team members are ready to join their colleagues on console. As a result of COVID restrictions, astronauts, trainees and trainers had unique hurdles to overcome during the initial months of the coronavirus pandemic. Every element of training was impacted including many specialized training facilities. In addition to these limitations, social distancing constraints were implemented and thus trainer-to-trainee, face-to-face interaction were not authorized. From orientation and introductory classes, simulations, advanced lessons, OJT, and final evaluations; all aspects faced new challenges. NASA operations trainers were forced to rethink, retool, and become more efficient with trainees and training resources to continue supporting payload operations on the ISS.

Significant barriers to ops training included facilities restricted from use, to limitations on in-person training classes, on site simulations and restrictions to OJT. In many cases, these limitations were overcome by using new remote-training tools like Microsoft Teams, IVODs and electronic documents to minimize person-to-person exposures. In other cases, like simulations and OJT where there is no substitute for the real thing, trainers and management worked together to find smart and safe solutions.

Astronaut pre-flight training was also drastically impacted by the COID-19 restrictions. Here it was even more important to reduce the potential for infections to the crew. In these cases, virtual learning with trainers by video or recorded lessons were utilized to meet many training requirements. These virtual training sessions also resulted in increased schedule flexibility and made training sessions highly efficient. One notable exception was when training the crew on detailed scientific tasks like dissections or plant sampling.

When the pandemic subsides and NASA considers all the lessons learned from the COVID-19 era, we predict that many of the efficiencies gained from virtual training will be retained. However, it must be noted that there can be no substitute for operators or astronauts gaining experience with the actual complex systems and hardware they will use in mission support.

Acknowledgements The authors would like to acknowledge the following groups who have contributed to the success of performing science onboard the International Space Station:

• The astronauts and cosmonauts who live and work aboard the ISS and risk their lives in the pursuit of space exploration.

• The operations teams around the world who monitor and operate the ISS, scientific payloads, and ground facilities around the clock and at the same time plan for future experiments.

• The training teams who work to ensure that the astronauts and ground controllers are prepared to operate the ISS for the benefit of all.

• The scientists and engineers who design the experiments and use the results to improve our way of life on Earth.

References

1. Cruzen C, Gibbs R, Dyer S, Cech J (2005) Expanding remote science operations capabilities onboard the international space station. In: IEEE aerospace conference, March 2005
2. Stetson H, Deitsch D, Cruzen C, Haddock A (2007) Autonomous payload operations onboard the international space station. In: IEEE aerospace conference, March 2007
3. Cruzen C, Montgomery J (2021) ISS payload operations training during the COVID-19 pandemic: impacts and solutions. In: SpaceOps 2021 conference, May 2021
4. MSFC COVID-19 Safe at Work Protocol Guidelines (2020) NASA Safety Document, NASA-MSFC, Spring
5. Microsoft Corporation (2020) Microsoft teams video conferencing software. https://www.microsoft.com/EN-US/microsoft-teams. Accessed 16 March 2021
6. Horvath N, Stinson C (2021) Discussions with NASA payload crew training coordinators. Huntsville, Alabama
7. Blum R (2021) Discussions with NASA payload crew training coordinators. Houston, Texas

Attitude Control on GRACE Follow-On: Experiences from the First Years in Orbit

F. Cossavella, J. Herman, L. Hoffmann, D. Fischer, H. Save, B. Schlepp, and T. Usbeck

Abstract The two satellites for the GRACE (Gravity Recovery And Climate Experiment) Follow-On mission were successfully launched in May 2018 into a polar orbit at an altitude of 491 km. Its predecessor GRACE was operated by the same partners from 2002 until 2017). The mission continues the measurements of the gravity field of the Earth (with emphasis on the time variability) and also delivers radio occultation measurements. The twin satellites are kept at a relative distance of 170 to 270 km and act as probes in the gravity field of the Earth. The inter-satellite distance is measured by a microwave tracking system to an accuracy of 1 µm. A laser ranging interferometer is added as a technology demonstration. Stable and accurate relative pointing, as well as the minimization of disturbance torques, is required in order to optimize scientific results. This poses stringent demands upon attitude control. The performance of the GRACE Follow-on attitude control system will be presented, as well as the special actions and changes that became necessary as the mission evolved. A short description of the sensors and actuators used for attitude control is given and improvements with respect to GRACE are discussed in some more detail. The operational modes are described with a focus on the so-called nominal fine-pointing mode, in which the front ends point towards each other in order to enable microwave- and

F. Cossavella (✉) · J. Herman · L. Hoffmann · B. Schlepp
Deutsches Zentrum Für Luft-und Raumfahrt e.V. (DLR), Oberpfaffenhofen, Germany
e-mail: Fabiana.Cossavella@dlr.de

L. Hoffmann
e-mail: L.Hoffmann@dlr.de

B. Schlepp
e-mail: Benjamin.Schlepp@dlr.de

D. Fischer · T. Usbeck
Airbus Defence and Space - Space Systems, Immenstaad, Germany
e-mail: Denis.Fischer@airbus.com

T. Usbeck
e-mail: Thomas.Usbeck@airbus.com

H. Save
Center for Space Research, The University of Texas, Austin, TX, USA
e-mail: save@csr.utexas.edu

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_22

laser-ranging. The third section opens with a description of special tasks, such as the fine-tuning of the control and monitoring parameters and the complex determination of the satellite's center of gravity. A comparison is made with a tracking model based upon the fuel expenditure from the two tanks that can be determined independently. Several series of involved tests with manual thruster firings were performed in order to characterize the response of the accelerometers to thruster actuations. A description of the design of the tests, their execution and results is presented. A switch to the redundant instrument was made five months after launch on one of the satellites. The consequences for attitude control are discussed in Sect. 4. A method that was developed to cope with a situation where also the redundant GPS receiver would become unavailable is discussed in detail. Conclusions and an outlook for the upcoming years of operations are presented in the last section.

Keywords GRACE-FO · AOCS

Acronyms/Abbreviations

ASM	Acquisition and Safe mode
ASM-RD	ASM Rate Damping
ASM-CP	ASM Coarse Pointing
ASM-SS	ASM Steady State
AOCS	Attitude and Orbit Control System
ACT	Attitude Control Thruster
CESS	Coarse Earth and Sun Sensor
CGPS	Cold Gas Propulsion System
CoM	Center of Mass
CMC	Center of Mass Calibration
DLR	Deutsches Zentrum für Luft- und Raumfahrt–German Aerospace Center
FDIR	Failure Detection, Isolation and Recovery
FDS	Flight Dynamic System
FGM	Fluxgate Magnetometers
GFZ	GeoForschungsZentrum–German Research Center for Geosciences
GPS	Global Positioning System
GRACE	Gravity Recovery And Climate Experiment
GRACE-FO	Grace Follow-on
GF1	Grace Follow-on 1
GF2	Grace Follow-on 2
GSOC	German Space Operations Center
IMU	Inertial Measurement Unit
IPU	Instrument Processing Unit
JPL	Jet Propulsion Laboratory
LEOP	Launch and Early Orbit Phase

LRI	Laser Ranging Interferometer
MTQ	Magnetic Torquers
NOM	Normal Mode
OBC	On-Board Computer
OCM	Orbit Control Mode
OCT	Orbit Control Thruster
OD	Orbit Determination
OOP	On-board Orbit Propagator
PVT	Pressure-Volume-Temperature
RTN	Radial Tangential Normal
STR	Start Tracker
STR1	Start Tracker head 1
STR2	Start Tracker head 2
STR3	Start Tracker head 3
STRE	Start Tracker Electronics
TLEs	Two-Line Elements

1 Introduction

GRACE Follow-On is a scientific co-operation between the USA and Germany, following the model initiated by its predecessor GRACE [1, 2]. The twin satellites were built by Airbus Defence and Space in Germany, under a contract from NASA's Jet Propulsion Laboratory (JPL). Operations are carried-out at the German Space Operations Center (DLR-GSOC) on behalf of the German Research Center for Geosciences (GFZ).

The main scientific goal of the mission is to collect data for creating both static and time-varying maps of the terrestrial gravity field. The satellites were launched in May 2018 at an altitude of 491 km on a circular polar orbit. They are kept at a relative distance of 170 to 270 km and act as probes in the gravity field of the Earth. The inter-satellite distance is measured by a microwave tracking system to an accuracy of 1 μm; a laser ranging instrument (LRI) is added as a technology demonstration and has improved the accuracy of the distance measurements by a factor of 30 [3, 4]. Non-gravitational disturbances can be determined with a SuperStar accelerometer.

The front-ends of the satellites have to point towards each other to enable microwave- and laser-ranging. Stable and accurate relative pointing and the minimization of disturbance torques are required in order to optimize scientific results. This poses stringent demands upon attitude control.

A short description of the sensors and actuators of the attitude and orbit control system (AOCS) is given in the next section. Improvements with respect to GRACE are discussed in some more detail, in particular those for the star cameras and the inertial measurement units.

The performance of the AOCS during nominal operation is evaluated based on the data collected during the first two and a half years in orbit. A description of several special activities necessary to improve the analysis of the scientific data is given with particular emphasis on the consequences for the attitude control of the satellite.

A switch to the redundant instrument was made five months after launch on one of the satellites [5]. This comprised not only a switch of the instrument processing unit, but also of the ultra-stable oscillator, GPS receiver, microwave assembly and electronics. The consequences for attitude control are discussed in the Sect. 4, together with a modified operational approach to bridge several months without on-board GPS data.

2 The Attitude and Orbit Control System

The design of the GRACE-FO satellites benefits from the experience gained from their predecessors. In particular the number and quality of the sensors has been improved [6]. A schematic view of one of the satellites is shown in Fig. 1, a detailed description of the satellite layout can be found in [2].

The Coarse Earth and Sun Sensors (CESS) provides attitude measurements with respect to the direction of Sun and Earth. It consists of six sensor heads allocated along each axis, pair wise in opposite directions, in order to provide a spherical field of view. Each head consists of six thermistors, three for the detection of infrared light and three for the detection of Sun light, providing a 2-out-of-3 redundancy.

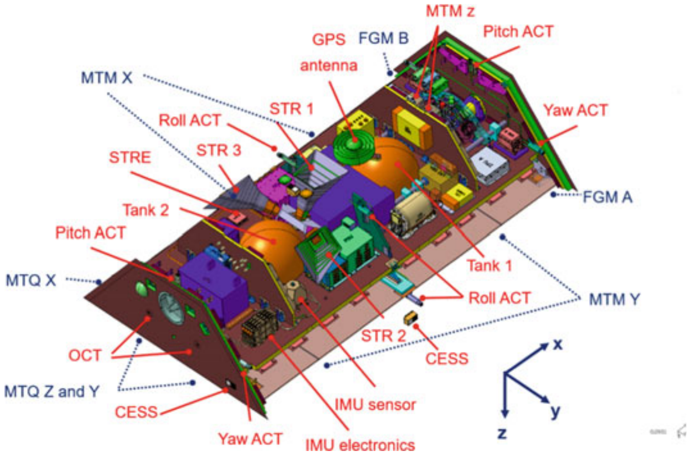


Fig. 1 Layout of one of the satellites (solar panels not shown). The location of the visible AOCS units is shown by red arrows. The dashed blue lines indicate the approximate location of units that are hidden

The magnetic field is measured by one of two fluxgate magnetometers (operated in cold redundancy) with an accuracy of ~300 nT. Each magnetometer consists of three independent magnetic sensors aligned to the satellite axes and operating simultaneously. The measurements are input for control with the magnetic torque rods and are also used as back-up computation of the rotation rates in acquisition and safe mode.

The use of a high-performance Inertial Measurement Unit (IMU) with four independent axes provides accurate rate measurements. A tetrahedral configuration of the inertial sensors offers one-failure tolerance. Three sensors are used during nominal operations, the fourth only after a transition into safe mode. On GRACE a medium performance IMU with three measurement axes was available, providing no redundancy.

GRACE had two star cameras of which only one was actively used for AOCS. GRACE-FO has three star-tracker heads with a separation between their boresights of 80.4° to 100°. The three STRs are used in hot redundancy and are connected to one of two electronic units which are operated in cold redundancy. The data are handled by the on-board computer (OBC) directly, whereas on GRACE they were collected and processed by the instrument processing unit before being handed to the OBC for use by the AOCS. This configuration optimizes the coverage and increases the accuracy of the attitude measurement in all three satellite axes. The new generation of star cameras can handle partial Moon intrusions and the on-board software autonomously delivers fused attitude data from one, two or three cameras depending upon validity.

As on GRACE, GPS data are provided by a dual-band receiver and the data are handled by the processing unit of the microwave instrument (IPU). These data are used to initialize the on-board orbit propagator (OOP) at each AOCS cycle.

The attitude is controlled primarily by a set of cold redundant magnetic torque rods (MTQs) of 27.5 Am^2 each. Each rod has a double coil providing cold redundancy. The rods are aligned with the axes of the satellite and located at maximum distance from the magnetometers in order to minimize disturbances on the magnetic field measurements. The MTQs are supplemented by a cold gas propulsion system with a set of twelve 10 mN attitude control thrusters (ACTs), separated into two branches. Two gas tanks of equal volume are mounted symmetrically along the x-axis of the spacecraft and are connected each to one of the branches, which are operated simultaneously to ensure an even depletion of the tanks. The design is inherited from GRACE with state-of-the-art enhancements and small improvements such as a two-stage pressure regulator that assures a constant feed pressure for the thrusters over the whole range of tank pressures. Two orbit control thrusters of 50 mN thrust, located at the rear part of the satellite, complete the cold gas propulsion system.

2.1 Attitude Modes and Frames of Reference

There are three main AOCS operation modes. The Acquisition and Safe Mode (ASM) is designed to guarantee power and thermal survival of the spacecraft, the Normal Mode (NOM) is the normal operating mode providing accurate three-axis attitude control according to the defined reference, and the Orbit Control Mode (OCM) is used for the necessary orbit change and maintenance maneuvers during the mission.

In addition, a reference attitude can be independently set by defining a frame of reference and a pointing bias. The pointing bias can be defined for each axis as Euler angle and it is added to the target attitude defined by the frame of reference.

The available frames of reference are the Nadir Pointing Frame, the Relative Pointing Frame and the Orbit Control Frame. In Nadir Pointing the z-axis of the satellite is aligned to the nadir direction, the y-axis is perpendicular to the plane defined by the z-axis and the velocity vector, and the x-axis is roughly pointed in the flight direction. The Orbit Control frame aligns the x-axis with the spacecraft velocity vector, the y-axis is perpendicular to the orbit plane and in the direction of the orbit normal vector. The z-axis completes the right-handed system.

As the horn antenna of the microwave instrument is located at the front of the satellite, in order to establish a link the two GRACE-FO spacecraft must point towards each other. When the Nadir frame is selected, a 180° yaw bias must be applied to rotate the leading satellite towards the trailing one. Precise inter-satellite pointing, however, is ensured only by selecting the Relative Pointing frame. This points the + x-axis to the other satellite, following a trajectory generated by an on-board algorithm according to an inter-satellite line of sight reference with the z-axis pointing towards Earth. The mutual distance is kept at 220 ± 50 km, which leads to a continuous small pitch bias of roughly -1°. The slowly varying mutual distance implies that the desired attitude is not a constant. The continuous knowledge of the position of the satellite and its partner is essential for the determination of the Relative Pointing frame. Therefore, each day the latest result of the orbit determination for both satellites are up-loaded to both satellites in form of Two-Line-Elements (TLEs). See also Sect. 4.3.

Attitude biases around the pitch and roll axes are nominally not commanded and used only during the calibration of the K-Band antenna pattern.

2.1.1 Acquisition and Safe Mode

ASM is autonomously entered after a mode drop by FDIR from the higher attitude modes, or after an on-board computer boot. The mode is divided into three sub-modes: rate damping (RD), coarse pointing (CP) and steady state (SS).

The ASM-RD is the point of entry of the ASM, the main task is to reduce the satellite rates to less than 0.2 °/s, no control of the attitude is performed. The spacecraft rotation rate is measured by the IMU or derived from CESS and magnetometer measurements in case of non-availability of IMU data.

In ASM-CP a coarse nominal pointing of the satellite is achieved, with the z-axis of the satellite directed towards Earth and a yaw angle of 0° or 180°, depending on which value is closer to the current attitude.

The ASM-SS keeps the spacecraft in a coarse Earth-pointing attitude. The more powerful battery on GRACE-FO (78Ah name-plate capacity on GRACE-FO compared to 18Ah on GRACE) tolerates temporary yaw deviations of up to 60° in safe mode. No "yaw-steering" concept [7], forcing one of the side panels towards the Sun, is applied on GRACE-FO in ASM-SS.

The ASM mode intrinsically uses the Nadir Pointing frame as reference.

2.1.2 Normal Mode

NOM is the normal operating mode and it provides accurate Earth-pointing attitude according to the defined reference. The reference attitude is derived from the spacecraft position and velocity state vector provided by the GPS or, in case of a GPS outage, by an AOCS on-board orbit propagator.

The normal mode is subdivided into an intermediate acquisition sub-mode (ACQ), the Attitude Hold (AH) and the Fine Pointing (FP) sub-modes.

The NOM-ACQ is the entry point of the normal mode from coarse pointing in ASM or from OCM. The spacecraft acquires a specified 3-axis attitude relative to the selected frame, with an accuracy better than 30 mrad at 3σ confidence level. Rotations of the spacecraft of 180° in yaw are always performed in this mode.

Attitude Hold is entered from NOM-ACQ or from the NOM-FP and it holds the reference 3-axis attitude, with an absolute pointing error (at 3σ confidence level) better than 10 mrad in roll, and better than 5 and 3 mrad in yaw and pitch respectively. The attitude hold mode is used for center of mass calibrations and any other special AOCS operation that could lead to not nominal attitude disturbances, as for example an update of the NOM-FP controller settings or specific thruster tests.

The NOM-FP is the mode that provides high accuracy attitude pointing for science operation. The roll axis is controlled with an accuracy better than 2 mrad, whereas the pitch and yaw axes are controlled with an accuracy better than 0.25 mrad at 3σ confidence level. A science configuration is established once the Relative Pointing is activated in NOM-FP.

The use of TLEs to point towards the partner satellite is on GRACE-FO not restricted to the fine pointing (science) mode. This can be set by command in any NOM sub-mode, thus allowing for extra flexibility when performing special tests.

In all NOM sub-modes, the inertial attitude information is derived from accurate autonomous star sensor measurements and propagated to the current on-board time using measurements of the spacecraft rate from the high accuracy IMU. Attitude measurements from at least two star sensors are fused together, in order to achieve low noise in all axes by eliminating the star sensor boresight noise, which is typically 10 times higher than the cross-axis noise.

The spacecraft rate is also derived from star sensor measurements, as back-up in case of non-availability of IMU data. The CESS is used for FDIR surveillance of Sun/Earth angles in the normal mode.

Actuation is performed mainly by the magnetic torquers, augmented by the cold gas thrusters whenever necessary. The thrusters are operated with different sets of thrust limitation parameters according to the actual state of the attitude and rate errors. Magnetorquer control is based on on-board magnetometer measurements. The NOM-AH mode reflects the GRACE Attitude Hold Mode except for orbit control maneuvers and the NOM-FP mode the GRACE Science Mode [7].

2.1.3 Orbit Control Mode

This mode is used for the necessary orbit change and maintenance maneuvers during the mission under direct ground control. It provides the same principal functionality as the NOM-AH mode and uses the same equipment plus the orbit control thrusters, which are disabled in NOM. The OCM has no sub-modes.

2.2 AOCS Performance

The AOCS on GRACE-FO has shown a very stable behavior since launch. Both satellites have been commanded to NOM-FP ~21 h after separation; GF1 and GF2 have since then maintained fine pointing for more than 98% and 93% of the time, respectively.

The science configuration, i.e. simultaneous NOM-FP and Relative Pointing on both satellites, has been maintained for 86% of the mission life time. It was interrupted by maintenance phases to execute specific AOCS activities or orbit maneuvers, and by AOCS safe modes.

In order to maintain the twin satellites within a relative distance of 220 ± 50 km, five orbit control maneuvers were executed on GF1 and six on GF2 (of which one, however, was a collision avoidance maneuver made in 2018).

The longest interruptions of the science configuration were two prolonged periods in ASM and NOM-Nadir- Pointing on GF2, caused by an outage of the prime instrument processing unit in 2018 that left the AOCS without GPS data for a few months (see Sect. 4) and by an OBC switch to the back-up unit in early 2019 [8]. GF2 has spent a total of ~32 days in ASM, GF1 less than 5. Fuel consumption on GRACE-FO is considerably smaller than it was for GRACE, although the pointing performance is considerably better [3, 6]. In ASM an average gas consumption of 2.5 g/day and 5 g/day has been observed on GF1 and GF2 respectively, smaller than the initially allocated budget of 30 g/day and with some variations according to the angle of the Sun with respect to the orbital plane. This is a factor of 20 to 100 less as compared to GRACE. The nominal modes perform also better on GRACE-FO than on GRACE, with an average expenditure for both satellites of about 0.7 g/day in fine pointing mode (cf. 3–5 g/d in the equivalent Science Mode on GRACE) and ~1 g/day in attitude hold mode (cf. 10 g/d in Attitude Hold Mode on GRACE).

The GRACE-FO satellites were launched with ~32 kg nitrogen gas and have consumed in the first 2.5 years of life about 1.2 kg of fuel. Such a low fuel consumption would allow to operate the satellites for another 60 years, although without an active altitude maintenance they will re-enter in the Earth atmosphere much earlier. The extra fuel can therefore be used to manage the altitude to increase the lifetime and improve the science products.

2.2.1 Fuel Consumption

Two methods are used to track the accumulated gas expenditure: the book-keeping method and the Pressure-Volume-Temperature (PVT) method.

The book-keeping is based on precise tracking of the total amount of time each thruster has been activated, and on knowledge of the thruster feed pressure and mass flow rate of each thruster unit. The mass flow rates at a reference feed pressure of 1.5 bar are known from calibrations of thrusters done on ground and the thruster on-times are recorded on board with one millisecond precision. The feed pressure in each propulsion branch is also available in the spacecraft telemetry. As the mass flow rates scale linearly with the feed pressure, it is possible to track how much gas has been ejected by each thruster over time.

The PVT method uses a combination of on-board and ground information together with the equation of state for a real gas to estimate the amount of mass in each pressure vessel:

$$\mathrm{m} = \frac{\mathrm{P} \cdot \mathrm{V}}{\mathrm{R} \cdot \mathrm{T} \cdot \mathrm{Z}}$$

where P is the pressure of the tank, V its volume, T its temperature, R is the universal gas constant and Z is the compressibility factor. Temperature and pressure of each tank are available in the on-board telemetry, whereas the volume of the vessel has been measured on ground before launch. In this work, a modified van der Waals approach to estimate the compressibility factor is used. The accuracy in the measurement of the gas temperature and pressure is the main source of uncertainty in this method, which can estimate the propellant mass with an accuracy of a few percent. Because the temperature sensors are located on the outer surface of the vessels, the PVT method does not provide accurate results during phases of fast cooling and heating of the nitrogen gas.

The book-keeping method is more suitable for calculating the consumption of the propellant mass but it is less sensitive to small leakages in the system. A comparison between the two methods is used as an indicator for leaks in the propulsion system. In Fig. 2 the estimated propellant expenditure for both tanks on each satellite is shown. Estimates with the book-keeping method show on both satellites a slightly higher consumption of gas from tank 2 (located along the minus x-axis) because of

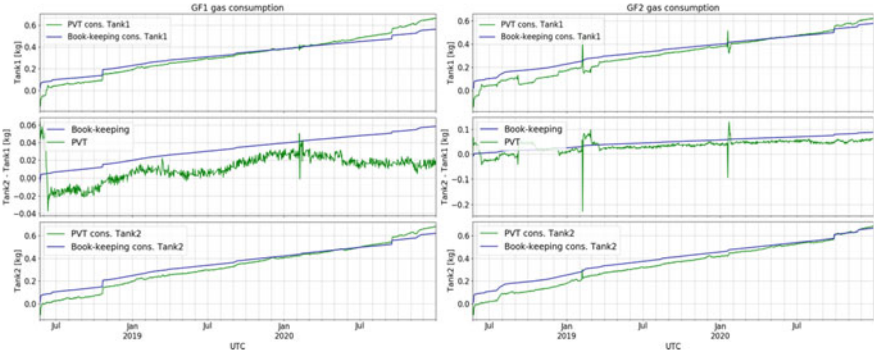


Fig. 2 Expenditure of gas on GF1 (left panel) and GF2 (right panel) as estimated with the PVT and book-keeping methods. The consumption of gas from tank 1 and tank 2 is displayed on the top and bottom panels, respectively. In the middle panel the difference in gas consumption between the two tanks is shown. Both satellites have consumed about 1.2 kg of nitrogen gas, with 25 to 60 g more gas taken from tank 2. Spikes in the used mass estimated from the PVT method coincide with fast cooling phases that followed the switch-off of both science instruments. This happened on GF2 in early 2019 and early 2020, and on GF1 in early 2020. The effect is stronger on tank 1, because it is closer to the instruments that are located near the front side of the satellite

the higher average feed pressure measured on the corresponding branch. The trend is confirmed by the results of the PVT mass estimation for all tanks, within the method uncertainties. On GF1, however, a small divergence between the results of two methods has been observed in the second half of 2020 and it is currently being investigated.

2.2.2 Star Tracker Performance

The boresight of STR1 is close to the −z-axis of the satellite, whereas the boresight of STR2 and STR3 is close to the +y and −y-axis respectively (see Fig. 1). The use of three instead of two camera heads (as on GRACE) improves the availability and accuracy of attitude data about all spacecraft axes. Fused data from all three heads were delivered for approximatively 70% of the time in the first two and a half years of the mission, whereas 0.2% of the time attitude data were derived from a single head only (the other two being blinded by the Sun and the Moon simultaneously; see Table 1). The individual heads deliver valid data for 87–94% of the time.

The performance of GF1 STR2 has to be compared to that of STR3 on GF2, because GF1 flies backwards with a 180° yaw bias with respect to the Nadir frame. The same applies for GF1 STR3 and GF2 STR2.

The precession of the orbital plane has an inertial period of about 320 days (β' cycle), implying that for ~160 days the Sun is on one side and for ~160 days on the other side of the satellite. β' denotes the angle between the orbital plane and the direction to the Sun.

Table 1 Percentage of quaternion samples derived from the fusion of the data from 3, 2 or 1 star tracker heads (upper half) and percentage of valid samples delivered by each star tracker head (lower half) with respect to the unit on-time, after 2.5 years in orbit

	% of on-time	
	GF1	GF2
Fusion type		
3 star cameras	71.06	69.94
2 star cameras	28.77	29.19
1 star camera	0.17	0.17
Validity		
STR 1	93.95	93.28
STR 2	87.70	86.86
STR 3	89.17	88.25

The behavior of each head depends on the phase in the β′ cycle. STR1 is blinded by Sun for β′ around 0° (phase = 0.0 and 0.5), GF1 STR3 and GF2 STR2 are blinded by Sun for β′ around −90° (phase = 0.75), while for GF1 STR2 and GF2 STR3 this happens as β′ approaches +90° (phase = 0.25) (see Fig. 3).

The slightly lower performance of GF2 STR2, as compared to GF1 STR3, is due to an anomalous behavior of the star tracker head in January 2019, when 90% of the measurements were invalid. Although the satellites were going for the first time through a β′ minimum, the same behavior was not observed on GF1 STR3. The anomaly was most likely caused by condensation on the lens of the star tracker 2

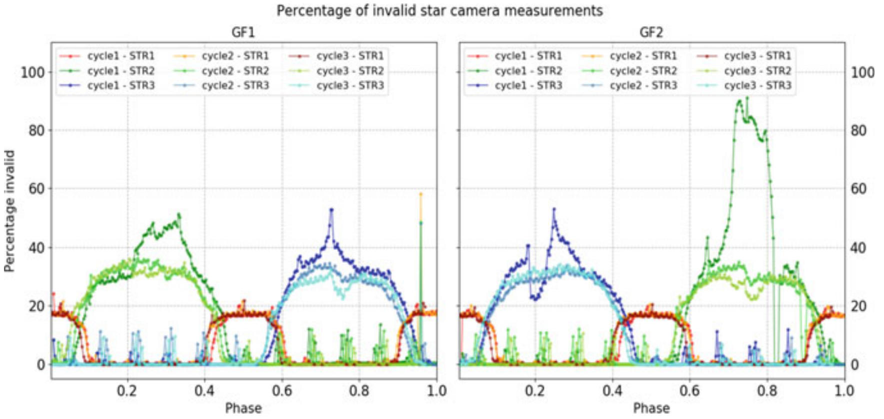


Fig. 3 Percentage per day of invalid measurements of the several star tracker heads as a function of the phase in the β′ cycle. The performance of the STR for GF1 and GF2 is displayed in the left and right panels, respectively. The small peaks of 10% occurring every month on both satellites are due to intrusion of the Moon in the field of view of the star cameras. The high number of invalid STR2 samples during cycle one on GF2, starting at phase = 0.7, was probably caused by a condensation on the lens of the camera. The period with low percentages around phase = 0.2 on GF2 during cycle one is due to a prolonged period in ASM (fewer and shorter blindings due to the larger yaw deviations allowed)

head. This hypothesis is confirmed by the nominal behavior of the star tracker head, when the same phase of the β' cycle was reached in the following cycles (see Fig. 3).

The alignment of each star camera head to the spacecraft frame of reference was determined on ground before launch [2]. Once in orbit, the relative alignment between the STR heads has been measured from the attitude data. The zenith star camera was chosen as a reference. For the side STR sensors, the angle between the in-flight and on-ground calibrated mounting quaternions has been found to be between 0.4 and 0.6 mrad for each analysed quaternion pair. Attitude disturbances have been observed on both satellites when a STR head is excluded from the fusion process, usually because blinded by Sun or Moon. The effect is best visible when two out of three heads are simultaneously excluded and only one of the side cameras remains available. Attitude errors of about 0.5 mrad and occasionally up to 1 mrad were observed on the pitch axis in such cases, whereas nominally the pitch axis is controlled to better than 0.3 mrad. Although this level of disturbance is completely within the control capability of the NOM-FP mode, it led to a higher number of MTQ and pitch thruster actuations and could potentially disturb the science measurements.

The alignments for STR2 and 3 were updated in June 2020 to ensure a smoother transition if control is switched from one configuration to another. The analysis of the attitude errors after the star camera alignments were updated indicates that the existing bias has been significantly reduced, although there is an intrinsic lower accuracy when only one camera head is used. Almost two years of data have been analysed, starting in March 2019. The data set has been divided into 5 sub-sets based on the number of star camera heads available for attitude determination. For each sub-set, the average attitude error along each axis was calculated before and after the update of the star tracker alignments. Only days with nominal attitude and no special AOCS activities have been selected, resulting in a total of roughly 430 days before and 255 days after the update. The result for the pitch axis is shown in Fig. 4. The corresponding mean and standard deviation are reported in Table 2. On GF1 the average pitch error before the update of the alignments was −0.29 mrad when only measurements from STR2 were available, and 0.12 mrad when only measurements from STR3 were available. In contrast, the average attitude error was −0.05 mrad when all cameras were available. After the update, the average changed to 0.02 mrad and −0.10 mrad for the case in which only STR2 and only STR3 were valid, respectively. The standard deviation of the errors did not improve because this is not due to a misalignment but rather due to the lower accuracy of the measurements made with only one head. Similar results have been observed for GF2.

The number of daily actuations of the pitch thrusters has also been reduced after the update of the alignments and on GF1 no day with more than 30 actuations has been observed anymore. The results are summarized in Table 3 for both satellites and are consistent with the observations from the analysis of the attitude errors.

An improvement was also observed for the roll and yaw axes, but it is less significant and therefore not discussed here.

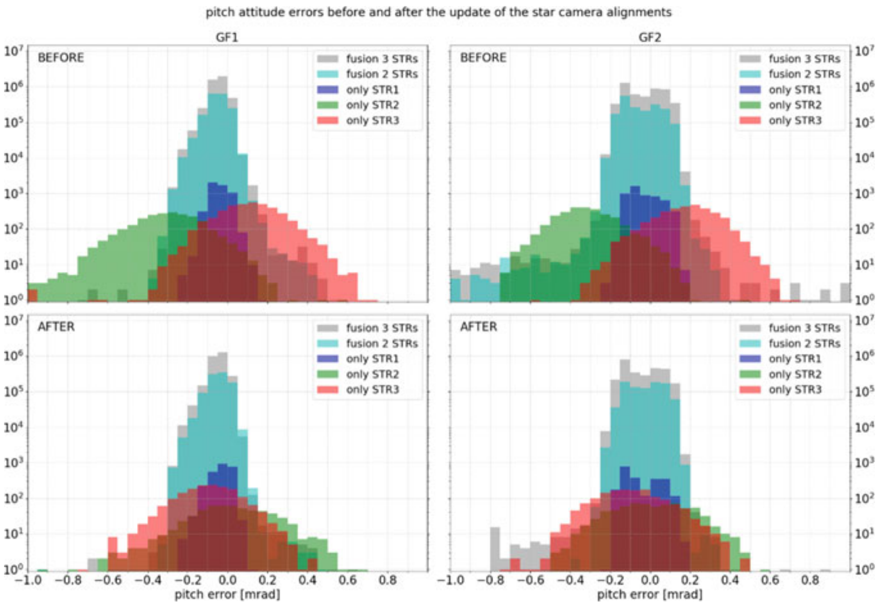


Fig. 4 Distribution of attitude errors along the pitch axis before (top panels) and after (bottom panels) the update of the alignment of the star trackers. On the left side are the attitude errors on GF1 and on the right side the GF2 ones. Due to the considerably smaller size of the data set corresponding to only one star camera head active, a logarithmic scale on the y-axis is used to better display all five data sets at once. Five sets of data are depicted: in grey the attitude error when all STRs are available, in light blue when only two STRs are available and the case when only one star camera is delivering data is displayed in blue, green and red for STR1, STR2 and STR3, respectively

Table 2 Mean and standard deviation of the attitude errors along the pitch axis, as a function of the number of star camera heads used in the fusion process. For each fusion type, values before and after the update of the star trackers alignement are reported

Fusion type		GF1		GF2	
		Before	After	Before	After
3 heads	Mean (mrad)	−0.05	−0.05	−0.03	−0.03
	std (mrad)	0.04	0.04	0.09	0.09
2 heads	Mean (mrad)	−0.04	−0.04	−0.04	−0.02
	std (mrad)	0.05	0.05	0.09	0.08
Only head 1	Mean (mrad)	−0.03	−0.02	−0.04	−0.07
	std (mrad)	0.05	0.04	0.05	0.08
Only head 2	Mean (mrad)	**−0.29**	**0.02**	**−0.34**	**−0.05**
	std (mrad)	0.17	0.21	0.12	0.18
Only head 3	Mean (mrad)	**0.12**	**−0.10**	**0.18**	**−0.09**
	std (mrad)	0.15	0.19	0.15	0.17

Table 3 Percentage of days with up to 10, 20, 30 and more than 30 actuations of the pitch thrusters, as a function of the number of star camera heads used in the fusion process. For each fusion type, values before and after the update of the star trackers alignement are reported

Fusion type	Daily actuations	GF1		GF2	
		Before	After	Before	After
2 heads	0–10	100	98	98.6	98
	10–20	0	2	0.34	2
	20–30	0	0	0.34	0
	>30	0	0	0.68	0
Only head 1	0–10	100	100	100	100
	10–20	0	0	0	0
	20–30	0	0	0	0
	>30	0	0	0	0
Only head 2	0–10	40	94	54	97
	10–20	23	3	26	3
	20–30	15	3	8	0
	>30	**22**	**0**	**12**	**0**
Only head 3	0–10	62	87	66	79
	10–20	20	6.5	10	18
	20–30	8	6.5	9	0
	>30	**10**	**0**	**15**	**3**

3 Special Activities

3.1 Center of Mass Calibration

Non-gravitational accelerations acting on the spacecraft have to be accurately measured in order to remove their effect on the measurement of the intersatellite distance. Therefore, each GRACE-FO spacecraft carries an accelerometer whose proof-mass is aligned to the center of mass (CoM) of the vehicle. Precise knowledge of the location of the center of mass and the capability of keeping it aligned within 100 μm to the proof-mass is required [2]. The satellite layout was optimized in order to control the offset between the CoM and the ACC proof mass to less than 500 μm in all axes. To keep the CoM offset within the desired range, each satellite is equipped with six movable mass trim mechanisms whose rails are parallel to the three axes of the satellite. Each mass weights ~5 kg and can be moved independently in steps of 2.5 μm, allowing for a compensation of the CoM offset up to ±2.16 mm along the x-axis and ±1.74 mm along the y- and z-axis.

The propellant tanks are located on the x-axis at the same distance from the CoM. An unequal fuel usage between them and leakages across the thruster branches are expected to induce over time a drift of the center of mass along the x direction.

The center of mass is measured in orbit by means of a center-of-mass calibration maneuver (CMC). A detailed description of its design and of the analysis method to determine the CoM can be found in [9]. A CMC consists of a periodic angular acceleration along the desired axis imposed on the spacecraft using the magnetic torquers. A constant magnetic torque of ±0.01 Nm is commanded along the selected satellite axis, following a nearly square wave pattern with a pulse width of 5 s and a period of 12 s. Each maneuver consists of 15 of these cycles, for an overall maneuver duration of 180 s. The AOCS executes automatically the maneuver based on the commanded input parameters. Before the calibration starts, the attitude is stabilized by narrowing the dead-bands in all spacecraft axes to 0.5 mrad. These are then set to infinity during the maneuver, resulting in no closed loop attitude control and no usage of attitude thrusters. At the end of a maneuver the closed loop attitude control is resumed automatically. A complete CMC campaign consists of seven maneuvers, two about the roll-axis, two about the yaw-axis and three about the pitch-axis, executed at specific geographic locations.

Two calibration campaigns were executed soon after LEOP. Signature of outgassing in the accelerometer data and an unexpected drift of the center of mass along the x-axis were observed by the science data groups at the Jet Propulsion Laboratory of the Californian Institute of Technology and at the Center for Space Research (CSR) at the University of Texas (Austin). In order to monitor the shift of the center of mass over a complete β' cycle, twelve more campaigns were executed on a monthly basis until October 2019. The offset measured along the x-axis can be compared to the offset expected from the unequal consumption of fuel from the two propellant tanks. The satellite is modelled as five point masses along the x-axis at fixed distance from the CoM [10]. The five bodies are the propellant tanks, the two mass trim mechanisms located along the x-axis and the remaining satellite mass that is fixed and won't change over time. The position of the latter has been estimated by taking as reference a CoM offset measured in orbit. Given the initial outgassing of the satellite, the calibration in February 2019, after the spacecraft went through almost a complete β' cycle and had been illuminated from all sides, has been chosen as reference. The offset of the CoM position calculated from the estimated fuel consumption in the tanks in general represents well the measured one, for both the book-keeping and the PVT methods (see Fig. 5 for GF1). The latter shows some higher fluctuations due to changes of tank pressure and temperature in the different phases of the β' cycle. In February 2020 the trim mass was moved on GF1 in order to maintain the CoM offset within the desired range of ±100 μm. The discrepancy observed in the fuel consumption estimated by the PVT and book-keeping methods (see Sect. 2.2.1) is visible also in this analysis. However, the uncertainties associated with both methods have not been evaluated yet and further investigation is ongoing.

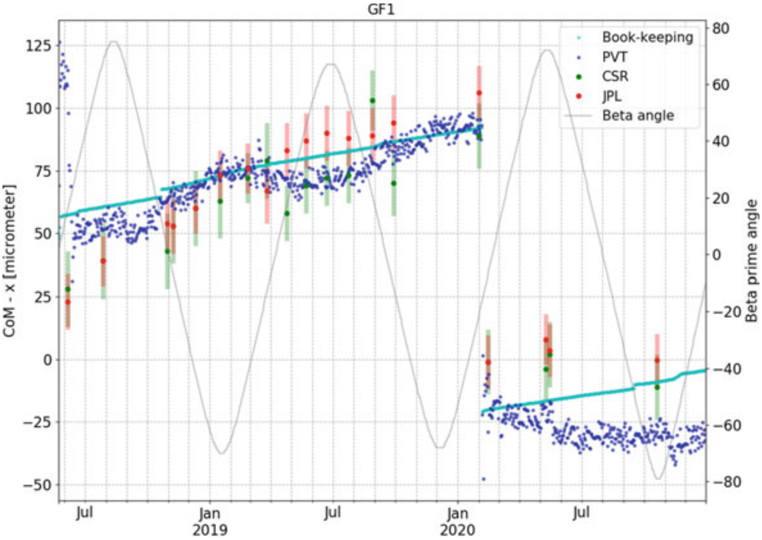


Fig. 5 The offset along the x-axis of the center-of-mass is shown for GF1 over the mission. The CoM offsets measured in orbit during the CMC campaigns are shown in red and green and are based on the estimation from the JPL and CSR science teams, respectively. Superimposed is the estimation of the CoM offset due to the differential consumption of fuel determined with the PVT (dark blue points) and book-keeping (light blue points) methods. The drop in February 2020 is due to a commanded movement of the trim mass to compensate for the accumulate offset. The discrepancy between the PVT and book-keeping methods in the second half of 2020 is currently under investigation. The β′ angle is displayed on the right axis

3.2 Fine Tuning of Monitoring and Control Parameters

It was discovered soon after launch that the accelerometers on GRACE-FO display an improper response at short thruster firings, in particular at roll thruster firings [11]. Roll thruster activity is maximum at the geomagnetic equator, where the control authority of the magnetic torquers is drastically reduced in roll. Ways to reduce the overall thruster activity by making less but slightly larger pulses, thereby temporarily accepting larger deviations in roll, were investigated and tested. This was an iterative process that led to a total of three tests, each with a different combination of thruster and NOM-FP controller settings (see Table 4).

The first step was to modify only the settings for the commanding of the roll and yaw thrusters, including the increase of the minimum thruster on-time for the roll and yaw thrusters from 0.05 s to 0.5 and 0.075 s, respectively. The test performed as expected, the number of roll thruster firings was decreased by 85%, but the daily fuel expenditure increased by 50%. The attitude control performance was not affected.

In a second step the number of roll thruster actuations was even reduced by ~97% over a day. The minimum thruster on-time for the roll axis was increased to 1 s and for the yaw axis to 90 ms, and the angular dead bands for the two axes were increased

Table 4 Controller settings in NOM-FP and AOCS performance during each of the tests carried out in order to reduce the number of roll thruster actuations over a day. Tests one to three were performed over 24 h only. The settings of Test 3 were then maintained over a period of three months

	Min Thr ontime (ms)			Angular dead-band (mrad)			Fuel (grams/d)	Daily number of roll thruster actuations
	Roll	Pitch	Yaw	Roll	Pitch	Yaw		
Default	50	50	50	2.5	0.25	0.25	0.7	160
Test 1	500	50	75	2.5	0.25	0.25	1.2	35
Test 2	1000	50	90	5	0.25	0.5	0.94	5
Test 3	1000	100	100	5	0.5	0.55	0.95	8
3-months	1000	100	100	5	0.5	0.55	1.2	14

to 5 mrad for roll and 0.5 mrad for yaw. The fuel consumption was higher than in the nominal configuration, but only by 34%. The extremely low fuel consumption on GRACE-FO provides enough margin to accept this moderate increase. However, also the number of pitch thruster firings increased (from 1 actuation per day, to 35 actuations per day), because of roll/pitch axis coupling through the deviation moment I_{xy}.

Finally, a third set of settings achieved a reduction of the roll thruster actuations by 95%, did not significantly increase the actuations in pitch axis and showed a fuel consumption comparable to what observed in the previous step. The maximum thruster on-time was increased to 1 s for the roll thrusters and to 100 ms for the yaw and pitch ones, and the angular dead bands were enlarged for all axes. Overall no violation of the pointing requirements was observed. This third set of settings was maintained for about 3 months, from November 2018 until February 2019. Over this time the performance could be estimated over a wider range of orbit conditions. The fuel consumption was higher than what observed during the 24 h test period, with on average 1.2 g of cold gas used in a day. It was also observed that with the new configuration a longer stabilization time in the pitch axis was needed when switching between frames of reference. It was decided to permanently revert all settings to defaults on both satellites in February 2019 when the activities to recover GF2 from the OBC switch took place. A method to calibrate the response of the accelerometer to the thruster actuations was investigated.

3.3 *Tests to Characterize the Accelerometer Response to Thruster Pulses*

A series of thruster tests was executed to model the response of the accelerometers to the actuation of each ACT thruster [12]. Measurements of long pulses were recorded over a range of differential pressures between the two cold gas branches. This could be obtained by spacing out the thrusts by a predefined amount of time.

The backbone of each test is the capability to command the ACTs in open loop. On GRACE-FO it is possible to disable the AOCS control of the thrusters, an FDIR re-enables the on-board control after a configurable amount of time that is by default set to 300 s. During this period the ground operator can command a sequence of thruster activations along the selected axis. The test consists of a sequence of 20 to 60 ACT actuations, each one second long, spaced out by a variable amount of time that ranges between 2 s and 45 min. A different approach is used according to the length of the interval between subsequent thrusts. If the actuations are spaced by less than a minute, they are alternated along the same axis but in opposite directions in order to counteract the attitude deviation created by each single pulse. For sequences requiring some minutes between subsequent firings, the AOCS control on the thrusters is reactivated after each pulse and therefore there is no need to command an alternating sequence. With the thrusters being operated in open-loop, the attitude is controlled only through the MTQs. It is therefore advantageous to execute the tests at geographic locations where the control authority of the magnetic torquer is maximum for the axis along which the ACTs are being fired.

A slight overpressure at the inlet of the attitude thrusters is apparent on branch B on GF1 during nominal operations. The design of the test has therefore to account on GF1 for longer waiting times than on GF2 to allow the pressure to build up to the nominal operational ranges.

On GF1 a total of six tests was executed, with spacings between thrusts of 2 s and of 3, 10, 15, 30 and 45 min. All of them could be executed in NOM-FP over ten days, without interrupting the nominal science data acquisition and with attitude deviations of less than 1.5 mrad in pitch and yaw, and less than 4 mrad in roll. On GF2, a set of six tests was executed with spacings between thrusts of 2 s, 4 s, 7 s, 11 s, 21 s and 15 min. Despite the alternating firing direction and the selection of the optimal geographic location, when commanding thrusts spaced by 4 to 21 s the attitude could drift and attitude errors up to 20 mrad in roll, 4 mrad in yaw and 1.5 mrad in pitch were observed. This part of the tests was therefore executed in the more robust attitude hold mode, resulting however in an interruption of the science mode for about six hours.

The complete set of tests was executed twice, in 2019 and 2020, and the data collected were used by the science data system to calibrate the output of the accelerometer before further using it for the modelling of the gravity field [12].

4 Operations Without Onboard GPS Data

The measurements from the on-board GPS receiver are processed by the IPU yielding a position-velocity–time solution (PVT) at a fixed 2 s interval to the attitude control system. The propagation to the current on-board time is done by a single step, fourth order Runge–Kutta algorithm using a fourth order model of the gravity field. The resulting state-vector is used to generate the so-called reference attitude (see Sect. 2.1). The PVT information is also included in the nominal telemetry stream

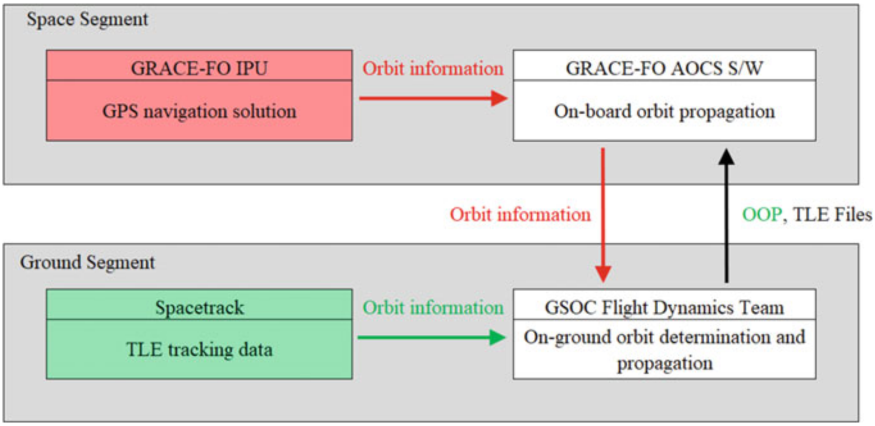


Fig. 6 A simplified diagram for the flow of position and velocity information between space- and ground-segment. The source will be changed to data from SpaceTrack (green box) in case of a prolonged unavailability of the IPU or of the on-board GPS receiver

and used for precise orbit determination from long data arcs. The result for the two satellites is then up-linked at least once per day to both in the form of Two-Line-Elements (TLE). The continuous knowledge of the own and of the other's position enables the accurate pointing required in the science mode (see Sect. 2.1).

The last valid measurement will be used and propagated if for some reason no data from the GPS receiver are available (could be caused by either a receiver, or an IPU outage). The accuracy for the required relative pointing deteriorates rather quickly and an on-board FDIR will trigger a transition into safe mode if the outage persists for more than 24 h. Longer receiver or IPU outages (e.g., due to planned maintenance or a switch-over) can only be bridged in NOM if the on-board orbit propagator (OOP) is updated regularly with fresh information from ground. The uplinked state vector will then no longer be based upon the satellite's own GPS measurements, but is computed from orbit data provided by SpaceTrack[1] (see Fig. 6). This decreases the overall pointing accuracy somewhat due to the lower frequency and quality of the delivered orbit data, and the precision of the on-ground orbit determination and propagation.

In 2018 the prime IPU on GF2 was powered down by an FDIR. A prolonged period of tests was finally followed by the decision to switch to the redundant side. This took several months in total because the complete instrument chain (including not only the IPU, but also the instrument control unit and the microwave tracking system) had to be switched. Orbit determination was based upon SpaceTrack data during this time and OOP updates were sent to GF2 every 2nd, or 3rd day only. No attempt was made to maintain the science configuration as discontinuities of up to 4 mrad in pitch were observed at each update.

[1] https://www.space-track.org/.

This prompted an investigation to see whether a method could be developed to cope with long GPS data outages and still maintain sufficient accuracy to uphold relative pointing and to continue science. An analysis of such a method and of the effects on the inter-satellite pointing is investigated in the following two sections. It will be shown that science can be continued even in the worst-case scenario of a permanent unavailability of on-board GPS data on one of the satellites.

4.1 Improvement When Using Data from SpaceTrack

Standard available TLEs from SpaceTrack had to be used to feed the OOP for GF2 between 2018-07-19 and 2018-10-16, due to unavailability of GPS data. Previous experience shows that a single TLE can be rather imprecise, leading to an apparent discontinuity in the relative position between GF2 and GF1 when the OOP is re-initialized based on such input. This in turn brings about a physical reaction of the attitude control system and might lead to the loss of the inter-satellite link (see Sect. 4.3 for a discussion on the allowed tolerances). Therefore a modified approach, applying a fitting algorithm over several TLE's, was developed.

Two sets of TLEs are normally provided by SpaceTrack each day for each of the GRACE-FO satellites. Taking all TLEs of the last 5 to 7 days and using a SGP4 propagator [13] ephemerides between each of the TLEs' epochs were created. These were then concatenated and handled as standard GPS data, i.e. an orbit determination (OD) over the complete period was made. The OD comprises a batch least-squares fitting algorithm, which allows the determination of a smoother orbit by excluding data from a rather imprecise single TLE. In this way the mentioned discontinuities from one TLE to the next one are strongly reduced.

The method was analysed for a period of three months between July and October 2019 by comparison with the nominal OD that is based upon the on-board GPS data. The results of both types of orbit determinations (state-vectors at a given epoch) were then propagated over ~24 h[2] and the position offsets compared over one orbit. The statistics over 103 days are presented in Table 5 using the RTN-frame (radial-tangential-normal[3]). The largest offset found were 150/4300/160 m in RTN-frame.

The above method requires periods without orbit maneuvers. A different approach would have to be developed for satellite missions with frequent orbit changes, e.g., maneuvers on a daily basis. Also, the analysis was performed over a 100-day period only and during a minimum in the solar cycle. The offsets as shown in Table 5 might differ considerably at other phases in the 11-year solar cycle due to increased solar radiation pressure and atmospheric drag.

[2] The OD results are used to provide on-board orbit products for the next 24 h. The accuracy of these products should be of sufficient quality, which is why an analysis period of 24 h was realized.

[3] Its origin is at the satellite position; the x-axis is aligned with the radial vector that points from the center of the Earth to the satellite (positive outwards), the z-axis is aligned to the normal direction

Table 5 Comparison between orbit determinations based upon on-board GPS data (reference) and from an algorithm fitting up to 14 TLE's (see text). The offsets in the RTN-frame are computed over one orbit after a propagation over a period of 24 h

	R (m)	T (m)	N (m)
Minimum	−150	−3600	−150
Maximum	130	4300	150
Mean	0	10	0
1-Sigma	70	1500	60

4.2 *Accuracy of the On-Board Orbit Propagation*

An analysis of the overall errors associated with the approach as described in Sect. 4.1 was made over a period of three days that had ODs based upon on-board GPS data, as well as from the algorithm using SpaceTrack TLEs available. The results are shown in Fig. 7. A fresh state vector with epoch 17:00 UTC was ingested by the OOP each day. The results of the OOP are then compared with the directly measured GPS position and velocity on-board. It is seen that the position error in tangential direction is significantly larger than in radial, or normal direction. The deviation in velocity, however, is largest in radial direction. The discontinuity after each upload

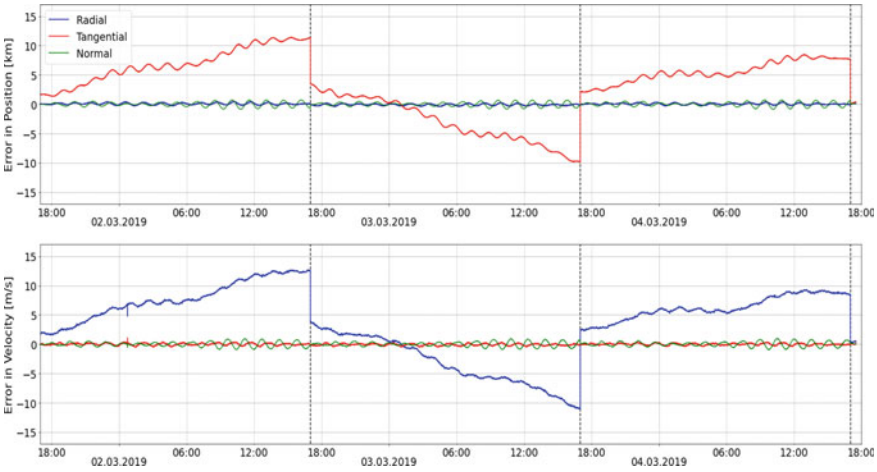


Fig. 7 Position and velocity errors in the RTN-frame are shown over a period of three days. These are computed from a comparison of the on-board GPS measurements with the results of the on-board orbit propagation (OOP). The OOP was re-initialized each day at 17:00 UTC by up-linking the result of an orbit determination (OD) based upon several days of SpaceTrack data. The error in position is dominated by its tangential, in velocity by its radial component

perpendicular to the radial and velocity vector, and the y-axis completes the right-handed system pointing in the tangential direction.

Table 6 The deviation in the RTN-frame of the OOP with respect to the on-board GPS measurements is shown as a function of the upload frequency. Data have been averaged over a three day period

	OOP update frequency	R (km)	T (km)	N (km)
Maximum error	Every two days	0.5	21.0	−0.1
	Once per day	0.5	11.4	0.9
	Twice per day	−0.4	7.0	−0.6
Maximum jump during update	Every two days	<0.1	18.7	<0.1
	Once per day	0.2	12.1	0.7
	Twice per day	0.2	8.1	0.3

is due to the uncertainty in the method of fitting the SpaceTrack TLEs. The small periodic variations that are visible, are related to the position in orbit.

The maximum deviation of the OOP and the magnitude of the jump at upload are of direct interest for science, i.e. for keeping relative pointing within the prescribed limits. These numbers (averaged over three days period) can be found in Table 6 for three different update frequencies. It can be seen that the maximum position error is ~7 km in tangential direction for an update frequency of twice per day (maximum for GRACE FO in the routine phase). Deviations increase to approximately 11 km and 21 km if the uplink frequency goes down to once per day, or once per two days, respectively (e.g., due to problems at the ground station). The maximum jump occurs at the moment that the OOP is re-initialised and its size comparable with the maximum deviations found.

4.3 Analysis of the Guidance Angles

The science mode requires that the front-ends of the satellites point towards each other (relative pointing—see Sect. 2.1.2 for a detailed description). The guidance angles are defined as the difference between the Relative Pointing Frame and the Nadir Pointing Frame and are calculated on-board from a propagation of the uplinked TLEs (also for the other satellite). A rough indication of these three angles is (0, −1°, 180°) for the leader and (0, −1°, 0°) for the follower, but exact guidance depends upon inter-satellite distance, eccentricity, inclination and also upon the quality of the TLE's and their upload frequency. The acceptable errors are dictated by the Laser Ranging Instrument (LRI). A pointing accuracy of ~300 µrad is required for first acquisition, but errors of up to 5 mrad are possible before lock is lost once the link has been established.

The guidance angles in pitch and yaw, and their respective errors, are shown over a period of 5 days when no on-board GPS data were available on one of the satellites (see Fig. 8). The upper two panels show the situation when the OOP was updated twice per day. It can be seen that the errors remain well within the 5 mrad boundary

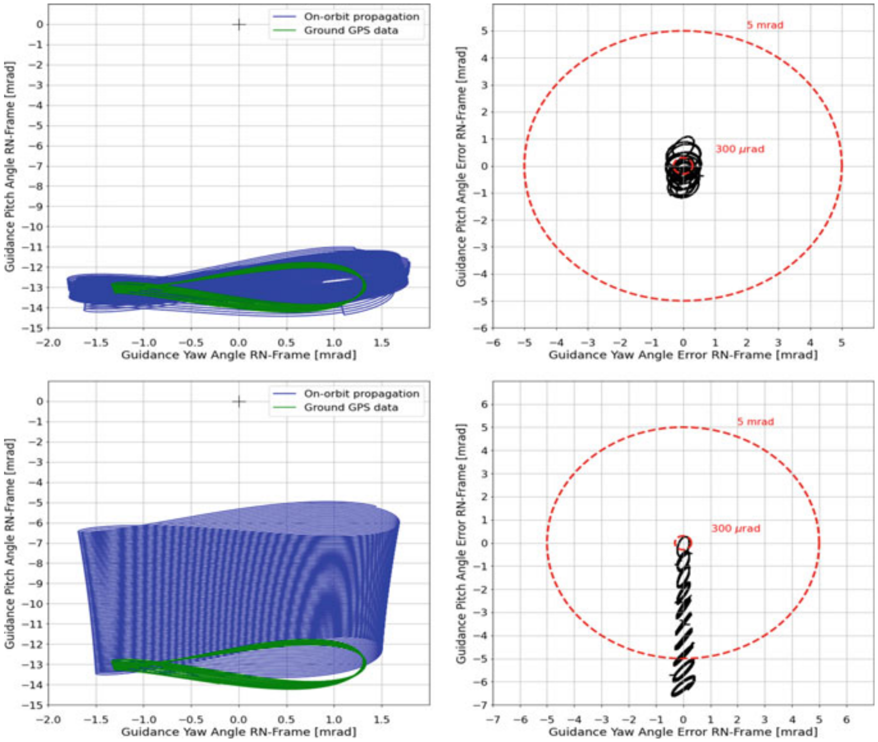


Fig. 8 The guidance angles in pitch and yaw and their errors are shown over a period of 5 days with on-board GPS data missing on one of the satellites. The computation is based upon the OOP, which is re-initialised twice per day in the upper two panels and not at all in the lower two. The RN frame describes the attitude differences between the relative pointing- and nadir- pointing frames. Only a few orbits are shown for clarity

after which LRI lock will be lost. An appreciable part of the time the errors are <300 μrad, implying that the link can be re-established. This is compared in the lower two panels with the case when the OOP is not updated at all. Already after one day a re-acquisition is no longer possible and the limit of 5 mrad is violated after three days. Note that the roll-angle is not very important for the pointing of the LRI and therefore not shown. Also note that the errors in pitch angle are much larger than the ones in yaw. The reason is the larger deviation in position found for the tangential direction (see Fig. 7).

The observed angle deviations are mainly due to the simplified gravitational model of the on-board propagator, together with the fact that the effect of the aerodynamic drag is not included. An overview of the maximum error, as well as the maximum jump, in pitch and yaw during the upload of OOP and TLE data is shown in Table 7.

This analysis, although preliminary because only a short interval near solar minimum could be included, shows that it is possible to continue full science operations even when no on-board GPS data are available on one of the satellites. A next

Table 7 The maximum error and jump in the pitch and yaw guidance angles are shown as a function of the frequency of OOP and TLE update

	OOP/TLE update frequency	Pitch (mrad)	Yaw (mrad)
Maximum error	Every two days	3.2	0.5
	Once per day	2.0	0.6
	Twice per day	1.2	0.6
Maximum jump during update	Every two days	2.7	0.2
	Once per day	1.8	0.1
	Twice per day	1.2	0.1

step could be to investigate the influence of different aerodynamic drag. Also, long-term simulations, including the variation of the angle between the orbital plane and Sun-direction, must be made to demonstrate the feasibility of the described approach. The ability of the instruments to fulfil their performance requirements under these conditions must also be assessed.

5 Conclusions

The performance of the AOCS of the GRACE Follow-on satellites was stable over their first years in orbit, benefiting also from the lessons learned from the GRACE mission. With a remarkably low fuel consumption, the life expectancy based solely on the expenditure of this resource is a factor of ten higher than the initially planned mission duration of five years. A minor difference between the two tanks of the cold gas system has been observed on both satellites, but the uncertainties associated with the methods to estimate the remaining cold gas mass still have to be fully investigated.

The settings for attitude control in normal mode were varied and several thruster tests carried out to help the science team to calibrate the measurements of the accelerometer.

A method to bridge long phases without a functional GPS receiver has been developed and was used on one of the satellites for the several months in which the switch to the redundant instrument chain was prepared. During this time, however, the science data acquisition was interrupted. A preliminary analysis showed that the pointing accuracy will be sufficient to support instrument operations in the worst-case scenario of the loss of the redundant chain. Further tests to improve on the performance will be needed in that case.

Acknowledgements The Gravity Recovery and Climate Experiment Follow-on (GRACE-FO) mission is a partnership between NASA and the German Research Centre for Geosciences (GFZ). Mission Operations for the first 5 years are funded by GFZ and sub-contracted to DLR's German Space Operation Center (GSOC) in Oberpfaffenhofen.

The authors would like to thank the Science Data System teams at JPL and CSR for their help and fruitful discussions during the first years of life of the GRACE Follow-on satellites.

References

1. Tapley BD, Bettadpur S, Watkins M, Reigber C (2004) The gravity recovery and climate experiment: mission overview and early results. Geophys Res Lett 31, Paper L09607
2. Kornfeld RP, Arnold BW, Gross MA, Dahya NT, Klipstein WM, Gath PF, Bettadpur S (2019) GRACE-FO: the gravity recovery and climate experiment follow-on mission. J Spacecr Rocket 56:931–951
3. Landerer FW, Flechtner FM, Save H, Webb FH, Bandikova T, Bertiger WI et al (2020) Extending the global mass change data record: GRACE follow-on instrument and science data performance. Geophys Res Lett 47:e2020GL088306
4. Abich K, Abramovici A, Amparan B, Baatzsch A, Bachman Okihiro B, Barr DC et al (2019) In-orbit performance of the GRACE follow-on laser ranging interferometer. Phys Rev Lett 123:031101
5. Witkowski MM, Gaston R, Shirbacheh M (2021) GRACE follow-on early in-flight challenges. In: 16th international conference on space operations, Cape Town, South Africa, 3–5 May
6. Löw S, Cossavella F, Herman J, Müller K, Davis E, Save H (2019) Attitude and orbit control of the grace satellites at extremely low power. In: IAC-19-B.6.3.2, 70th international astronautical congress (IAC), Washington D.C., United States, 2019, 21–25 October
7. Herman J, Presti D, Codazzi A, Belle C (2004) Attitude control for GRACE. In: ESA SP-548: 18th international symposium on space flight dynamics, Munich, Germany
8. Wirth K, Löw S, Müller K, Snopek K, Gaston R (2021) The challenge and consequences on mission operations after inverting a complex failure management concept in-orbit. In: SpaceOps-2021,3,x1438, 16th international conference on space operations, Cape Town, South Africa, 2021, 3–5 May
9. Wang F, Bettadpur S, Save H, Kruizinga G (2010) Determination of center-of-mass of gravity recovery and climate experiment satellites. J Spacecr Rocket 47:371–379
10. Lex F (2015) Analyse von Anomalien innerhalb des Kaltgas-Systems der low-earth-orbit satellite mission GRACE. Diploma Thesis, Hochschule München
11. McCullough CM, Harvey N, Save H, Landerer FW, Webb F (2019) Accelerometer modeling and calibrations for GRACE-FO. In: AGU fall meeting abstracts, G51B0590M
12. McCullough C, Harvey N, Save H, Bandikova T (2019) Description of calibrated GRACE-FO accelerometer data products (ACT). In: JPL D-103863, NASA Jet Propulsion Laboratory, California Institute of Technology
13. Hoots Felix R, Roehrich Ronald L (1980) Models for propagation of NORAD element sets: Spacetrack rept. no. 3, Aerospace Defense Command, United States Air Force

Ariane 6 Launch System Combined Tests

Charline Dutertre, Luis Escudero, Aline Decadi, Pier Domenico Resta, Julio A. Monreal, and Dirk Riedel

Abstract Ariane 6 is the next heavy European launch system of the Ariane family. It is being developed with the objectives to provide users with high mass performance, mission versatility, operational flexibility, high launch rate and low launch service cost. The flight segment (Launcher System), the ground segment (Launch Base) and the whole Launch System Architecture have already performed the respective Critical Design Reviews. ESA, in its role of Launch System Architect (LSA), is in charge of ensuring the coherence between the Launcher and the Launch Base, validating the Launch System performances and verifying the Launch System requirements so as to reach the above-mentioned objectives. With this goal, the LSA (ESA), the Launcher System Design Authority (ArianeGroup) and the Launch Base Design Authority (CNES) work together on building up an optimised launch operations plan to be validated during the so called Ariane 6 Launch System Combined Tests. The Combined Tests bring together the Ariane 6 Launcher, its Launch Complex and the Launch Range for the first time. The intended test sequences cover one by one all the operations and system configurations (including degraded cases) encountered during the launch campaigns. They will end with the launch facilities revalidation and reconfiguration in view of carrying out the Ariane 6 Maiden Flight. This paper presents the operational logic established to perform the Ariane 6 Combined Tests, the related Launch System, Launcher System and Launch Base tests objectives and the rational for the selected test sequences. The preliminary combined tests, dubbed "Early Combined Tests", which main objective is to mitigate specific risks before the first interfacing between the Launcher Combined Tests Models and the Launch Complex, will be also presented as well as the specific adaptations of the Launch Complex in order to be able to carry out the Combined Tests.

Keywords Launch system · Ariane 6 · Combined tests

C. Dutertre (✉) · L. Escudero · A. Decadi · P. D. Resta · J. A. Monreal
European Space Agency, Paris, France
e-mail: charline.dutertre@esa.int

D. Riedel
ArianeGroup, Paris, France

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_23

Acronyms/Abbreviations

BAF	Encapsulation Building
BAL	Launcher Assembly Building
CDL3	Control Centre
CSG	Guiana Space Centre
ECT	Early Combined Tests
EPCU	Payload Preparation Facilities
ESA	European Space Agency
ESR	Equipped Solid Rocket Booster
EMC	Electro Magnetic Compatibility
LH2	Liquid Hydrogen
LLPM	Low Liquid Propulsion Module
LOX	Liquid Oxygen
LSA	Launch System Architect
RF	Radio Frequency
ULPM	Upper Liquid Propulsion Module

1 Introduction

The Ariane 6 Launch System's development was first decided at the European Space Agency's Council meeting at ministerial level held in 2012. At that time, ESA Member States decided to finance the continuation of the Ariane 5 Mid-life Evolution project in parallel with the execution of Phase A development of a new Launch System named Ariane 6. The definitions of the two Launch Systems were constrained by the objective of maximising the common parts of the Cryogenic Upper Stage based on the Vinci engine. Ariane 6 Launch System development objectives [1] were set to achieve a considerable mission-cost reduction together with maximized launch mission flexibility and performance.

Applying guidelines set by ESA Council, a new governance set-up was designed for the future control of the European launchers sector. It assigns to ESA the dual role of Procurement Entity for both the Ariane 6 Launcher System and Launch Base development activities and of Launch System Architect. ArianeGroup is the Prime Contractor and Design Authority for Launcher System development. CNES is the Prime Contractor and Design Definition Authority for the Launch Base. Finally, the role of future Ariane 6 Launch Service Provider is assigned to Arianespace. Hundreds of European industrial partners bring also their expertise into this joint endeavour through the contribution of thirteen ESA Members States.

The Ariane 6 Operational Concept, presented in [2], targets a significant cost reduction accompanied by increased operational flexibility and versatility together with performance levels matching best forecasts for the 2020s and 2030s. The Ariane 6 Launch System definition is optimised to serve both institutional and commercial markets. This is fundamental in order to deliver a balanced yearly launch service cost in which the public sector plays the role of launch service customer, to whom the Launch Service Provider is bound to offer a launch service price defined up front and agreed by ESA Member States.

This paper presents the implementation of the Ariane 6 Launch System Combined Tests after the already achieved Launch System Critical Design Review phase.

2 Ariane 6 Launch System Description

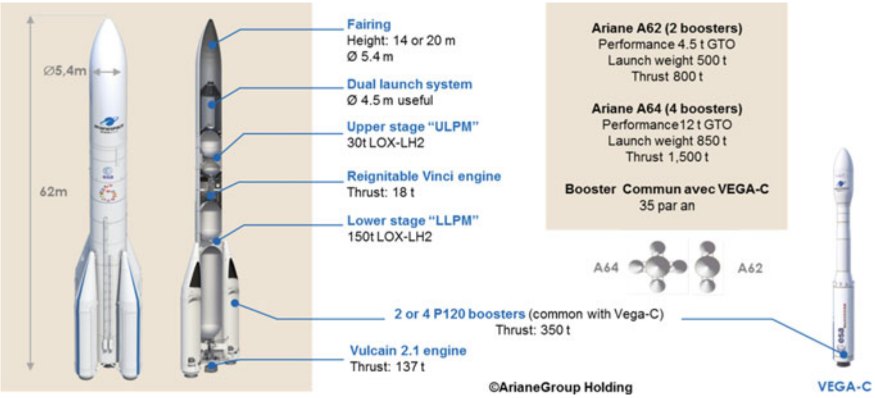
Ariane 6 is a collective adventure based on the passion of the multicultural teams, bringing their expertise, and on an entirely new and optimized industrial organization.

Building on the expertise acquired with Ariane 5 and on years of feedback from its customers, the project teams have developed state of the art and drawn the most efficient organizational standards. Design to cost, simultaneous engineering, maximized standardization up to the development of elements common with the Vega-C launcher, as well as ‘right first time’, are among the fundamental principles embedded. It is based on the Dependability (Reliability, Availability, Maintainability) and Safety approach and involves a series of project milestones which ensure that the progress of development meets all the conditions necessary for attaining the objectives throughout the overall development phase.

The Launch System results from applying an optimisation-to-cost of operations and concurrent engineering practices in which the Operational Concept plays a central role in constraining Launcher System and Launch Complex performance through the definition of their interfaces. The Ariane 6 Launch System comprises:

- The Launcher System which combines the production facilities and the ground means for integration and control located either in Europe or within the Space Centre perimeter in French Guiana and the launch vehicle elements to be integrated and tested, with the payload integrated and software loaded, ready for filling operations and launch countdown;
- The Launch Base including the Launch Complex (i.e. all infrastructures, facilities and ground means necessary for the assembly of the launch vehicle on its launch pad within the Space Centre perimeter in French Guiana, the filling operations, the countdown and the lift-off), as well as the Launch Range (in particular the safeguard means for ground operations, the flight checks means for launcher tracking, telemetry, safeguard remote control and associated data processing).

Solid and liquid propellants production facilities are also within the French Guiana Space Centre’s geographical perimeter and are the same as those already in use for the exploitation of the Ariane 5 and Vega Launch Systems.



©ArianeGroup Holding

Fig. 1 Ariane 6 Launcher configurations

2.1 The Launcher System

The Launcher System is made up of the three stages as per Fig. 1.

The Central Core comprises:

- The Lower Liquid Propulsion Module (LLPM) H150, equipped with the Vulcain 2.1 engine, diameter 5.4 m, separated tanks, liquid oxygen tank (RLOX) at top position;
- The Upper Liquid Propulsion Module (ULPM) H30, equipped with the Vinci re-ignitable engine, diameter 5.4 m, separated tanks, and burning 30 tons of liquid hydrogen and liquid oxygen.

The lateral Equipped Solid Rockets (ESR) are boosters equipped with solid rocket motors (P120C), diameter 3.4 m, containing 142t of solid propellant. Ariane 6 modularity is ensured thanks to two configurations: Ariane 6 with two boosters called A62, and Ariane 6 with four boosters called A64.

Additional flexibility and versatility options are proposed to guarantee competitive access to space and provide multiple adaptations depending on the mission (e.g. dual-launch system, deorbiting systems when needed, additional kits).

Ariane 6 launcher is modular, flexible and competitive and is the optimum launch solution for commercial and institutional customers. It combines proven solutions with innovation in order to address the changing needs of the market, allied with the unparalleled reliability of the Ariane family.

2.2 The Launch Base

The launch Base is composed by the Launch Complex and the Launch Range adapted made for Ariane 6.

2.2.1 The Launch Complex

The Launch Complex is the infrastructures, facilities and ground means necessary to carry out the integration of the Ariane 6 launcher elements as well as the launch operations. The Launch Complex facilities comprises:

- The Launcher Assembly Building (BAL) for integrating the Central Core;
- The Encapsulation Building (BAF) for integrating the fairing, the dual launch system (in case of dual launch configuration) and the payload(s);
- The Launch Zone including a launch table, a blast deflector, a mobile gantry and a lightning rod system;
- The launch operation control centre (CDL3);
- The transfer means for the transfer of the Central Core, the Upper Composite and the boosters to the Launch Zone;
- Support buildings for the provision of supplies.

The BAL is located in the Preparation Zone (Fig. 5) and is depicted in Fig. 2. The BAL is divided into three main zones:

- On the south side, the unpacking hall, which can receive two containers at a time (containers are evacuated as the campaign proceeds);
- On the north side, the assembly hall with one Central Core assembly line and one Central Core storage line;
- On the west side, the technical rooms.

The Launch Zone enables launcher complete assembly and check-out tests (under the Mobile Gantry), launcher propellant-filling operations, the final functional checks and lift-off. Two symmetric covered exhaust ducts are sized to evacuate the combustion gases of four solid propellant boosters and one cryogenic engine, with a deflector and a water retention area. The launch pad integrates a fixed launch table and a fixed

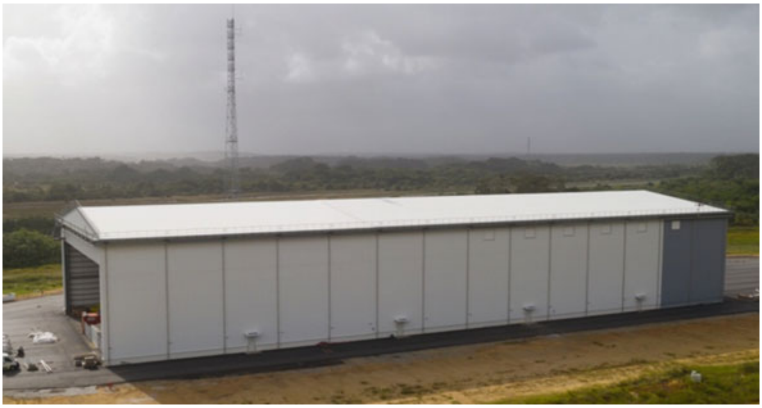


Fig. 2 Launcher assembly building (BAL)

umbilical mast equipped with two cryogenic arms and connection systems between the launch complex and the launcher. The Launch Zone depicted in Fig. 3 is equipped with four lightning masts protecting the launcher, a water tower (for deluge systems only) and a Mobile Gantry.

The Mobile Gantry (Fig. 4) can stand in two different positions: a front position during launcher preparation and a rear position (120 m away) for filling operations, countdown and lift-off. It is a metallic structure with platforms for accessing the

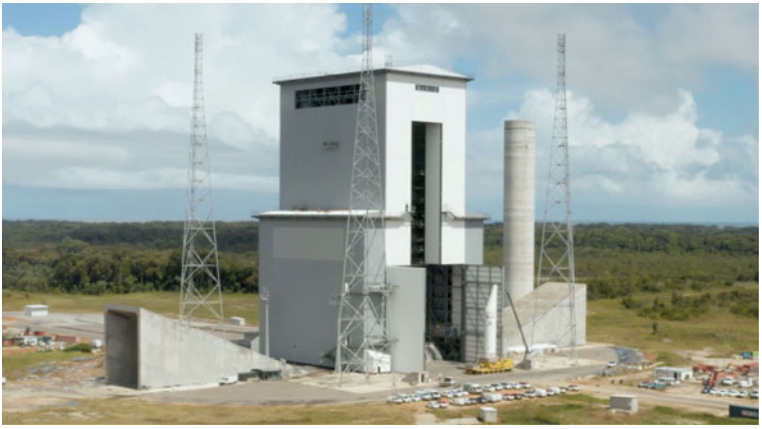


Fig. 3 Ariane 6 Launch Zone (December 2020) ©ESA/CNES/Arianespace

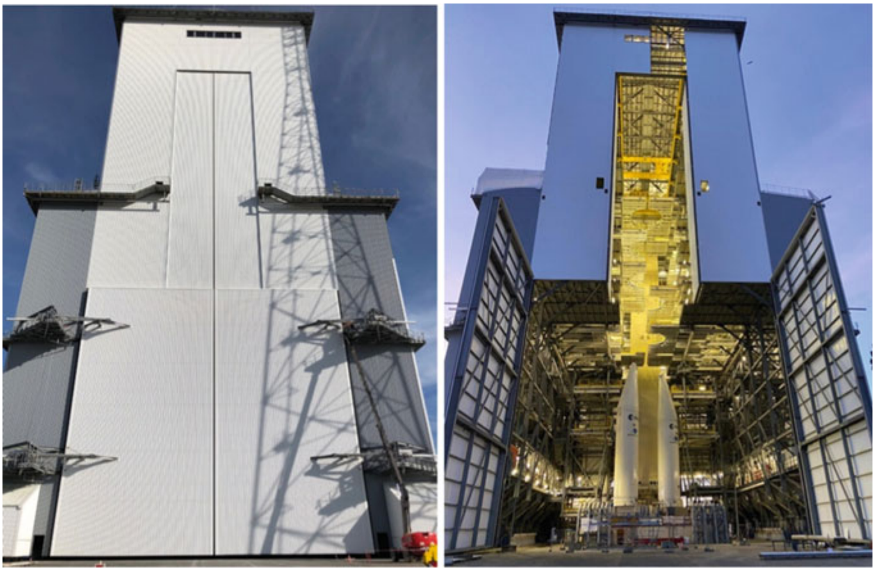


Fig. 4 Ariane 6 Mobile Gantry ©ESA/CNES/Arianespace

relevant levels of the launcher. It enables the assembly of the boosters on the Central Core and the hoisting of the Upper Composite and it provides access to the launcher as late as possible for final preparations.

2.2.2 The Launch Range

The Launch Range provides the necessary services and supports for carrying out a launch campaign including launcher and spacecraft preparation and tests, ensures safety and security of persons, assets, and protection of the environment and performs the safeguard means for ground operations, the flight checks means: tracking of the launcher trajectory and monitors and records the launcher telemetry through the complete launch mission, until satellites separation and launcher passivation. The Launch Range comprises among others means, telecommand, telemetry and tracking stations, telecommunications networks, optronics, weather forecast station, launch mission control centre, payload preparation facilities (EPCU) and auxiliary facilities and services for launch preparation (e.g. transport means, storage areas, chemical and physical laboratories, etc.). The Launch Range means are shared with other Launch Systems operated at the European Spaceport in Kourou (Ariane 5, Vega, Soyuz).

2.3 The Launch System Operational Concept

This concept addresses the operations of the launch campaign at the Full Operational Capability as the target to be achieved. The Ariane 6 Launch System Operational Concept is depicted in Fig. 5.

The launch campaign begins with the first destocking of a launcher element in the frame of a specific launch mission, and includes the following activities:

- Launch Base preparation;
- Central Core integration, preparation and control in horizontal position in BAL;
- Transfer of the Central Core, in horizontal position, from the BAL to the Launch Zone and its verticalization under the Mobile Gantry;
- Transfer of the boosters, mated on their palettes in vertical position, from their Storage Building to the Launch Zone;
- Boosters integration to the Central Core under the Mobile Gantry;
- Ground/board interfaces connections and the readiness tests;
- Payload(s) processing consisting in the payload(s) preparation in the Payload Preparation Facility (EPCU), their transfer to the BAF, and their encapsulation in the Upper Composite in vertical position;
- Upper Composite transfer from to BAF to the Launch Zone, hoisting and mating on top of the launcher under the Mobile Gantry;

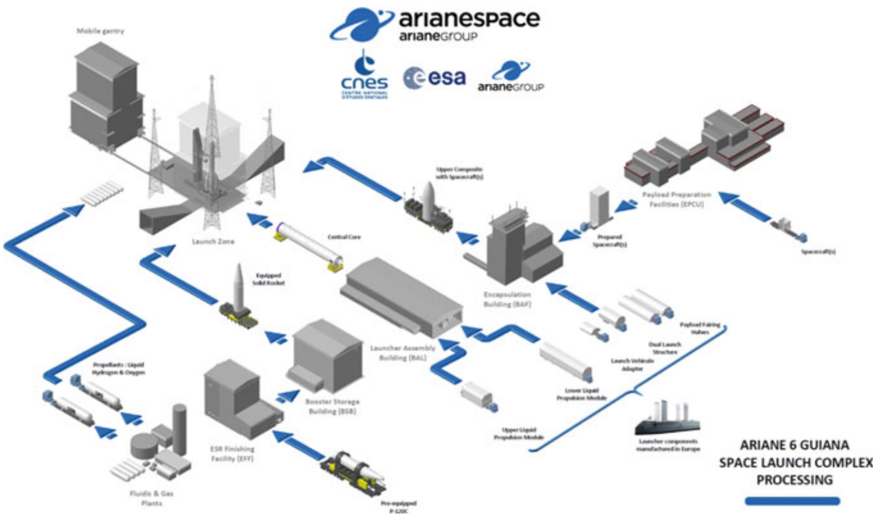

arianespace
arianeGROUP
cnes
esa
arianeGROUP
Mobile gantry
Launch Zone
Central Core
Equipped Solid Rocket
Upper Composite with Spacecraft(s)
Payload Preparation Facilities (EPCU)
Prepared Spacecraft(s)
Spacecraft(s)
Encapsulation Building (BAF)
Launcher Assembly Building (BAL)
Payload Fairing Halves
Dual Launch Structure
Launch Vehicle Adapter
Lower Liquid Propulsion Module
Upper Liquid Propulsion Module
Launcher components manufactured in Europe
Propellants : Liquid Hydrogen & Oxygen
Booster Storage Building (BSB)
ESR Finishing Facility (EFF)
Fluidic & Gas Plants
Pre-equipped P-120C
ARIANE 6 GUIANA SPACE LAUNCH COMPLEX PROCESSING

Fig. 5 Ariane 6 launch system operational concept

- Launch Operations to prepare for chronology and lift-off up to launch vehicle disposal, including Launch Base operations and flight data gathering/post-flight analysis;
- Launch Zone post-flight revalidation and preparation for next launch campaign.

3 Ariane 6 Launch System Qualification Tests

At Launch System level, the "Time-to-Market" high level requirement has driven ESA to establish a very thorough integrated verification and validation plan encompassing the verification and validation of technical and operational performances at the same time. For each Launch System function contributing to operational performances including dependability and safety, the end-to-end verification and validation logic is being established in collaboration with the Launcher System and Launch Base Design Authorities to guarantee that:

- Requirements and functions are fully verified and validated at lowest possible level in the product breakdown structure (PBS),
- Requirements and validation are satisfactorily verified and validated at least ones in the integrated logic,
- Test is the preferred verification method, including scale testing when necessary.

→ **This is the RIGHT FIRST TIME approach at all levels of the development chain**.

ESA, in its role of Ariane 6 Launch System Architect, is in charge of the implementation and management of the Launch System verification and qualification activities for the Maiden Flight up to the Ariane 6 Launch System Operational Qualification, before handing over to exploitation authority the responsibility of Ariane 6 Launch System exploitation and maintenance for the second launch and following launches.

The Launch System validation and verification activities include in particular (but is not limited to):

- Product level testing (Category 1 tests);
- **Launch System Early Combined Tests (ECT) and Combined Tests**;
- Maiden Flight campaign and flight.

Therefore, for Launch System scope, the so-called Early Combined Tests and Combined Tests constitute the main validation and verification steps, completed with some validation and verification during the Maiden Flight campaign and flight.

3.1 Early Combined Tests

The main objectives of the preliminary combined tests, dubbed "Early Combined Tests" (ECT) are the validation of the Launch System performances and the verification of some Launch System critical functions, in anticipation of the Combined Tests, in order to:

- **Mitigate technical risks** that may occur on some of the most critical systems as early as possible and **secure Ariane 6 development schedule** as such,
- Complete the verification of as many requirement's as possible when the relevant verification plan does not require real operational environment so reducing the amplitude and complexity of Combined Tests activities.

ECT selection is based on the criticality of the system (in particular when Ariane 6 presents a particularity with regards to Ariane 5 concept), together with the level of representativeness equivalent to the one required for Combined Tests validation. The ECT constitution is the following:

- Vulcain 2.1 ignition environment;
- Cryogenic Connection Systems disconnection and retraction tests;
- Central Core Deployment and ESRs mating in Launch Zone.

3.1.1 ECT Vulcain 2.1 Ignition Environment

The main objective is to anchor Computer Fluid Dynamics models in order to validate Vulcain 2.1 ground ignition environment by measuring the air speed around Vulcain 2.1 in different conditions.

One of the particularities of Ariane 6 with respect to Ariane 5 is that the Ariane 6 Lower Liquid Propulsion stage engine (Vulcain 2.1) is ignited by ground means (and not by Launcher means) thanks to ground ignition burners. These burners need to have the correct environment (in particular air speed) in order to produce a correct Vulcain

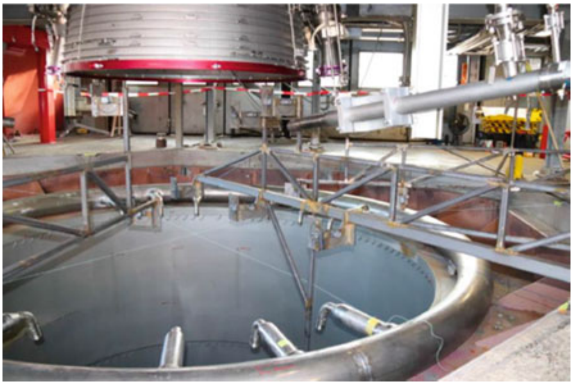


Fig. 6 ECT Vulcain 2.1 test set-up at DLR P5 test bench ©ESA/CNES/Arianespace

2.1 ignition. At the same time, below Vulcain 2.1 there is a Nitrogen Torus, which purpose is to produce the needed aspiration below Vulcain 2.1 in order to extract non-burned hydrogen. The challenge here is to find the correct balance between the good aspiration by the Nitrogen Torus below the Vulcain 2.1 and the correct air speed at burners area for performing an acceptable ignition.

For this reason, it has been deemed necessary to properly anchor Computer Fluid Dynamics models of the area in order to anticipate technical risks on Vulcain 2.1 ignition.

Three phases of ECT Vulcain 2.1 have been retained:

- Air speed measurement around Vulcain 2.1 in Europe Vulcain 2.1 test bench (DLR P5) taking profit of Launcher System Design Authority Vulcain 2.1 engine test campaigns (see Fig. 6);
- Air speed measurement in the Vulcain 2.1 area in the Launch Zone (on the naked table) with just the Nitrogen Torus taking profit of Launch Base Design Authority technical qualification tests (see Fig. 7);

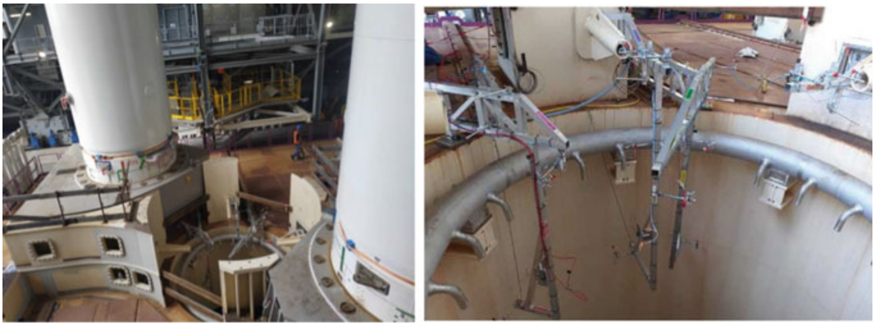


Fig. 7 ECT Vulcain 2.1 test set-up in Launch Zone—naked table ©ESA/CNES/Arianespace

- Air speed measurement in the Vulcain 2.1 area in the Launch Zone in presence of the Launcher mock-up in A62 configuration, taking profit of Launch Base Design Authority technical qualification tests but adding the dummy Central Core.

The two first phases of ECT Vulcain 2.1 have been successfully performed; the third and last one is planned for end of 2021.

3.1.2 ECT on Cryogenic Connection Systems

The main objective is to test the Cryogenic Connection Systems (CCS), which are the systems providing cryogenic fluidic functions to the Launcher, both for ULPM and LLPM. Contrary to Ariane 5, Ariane 6 CCS shall be disconnected in positive time (after ESR ignition order by the Launcher), suppressing the need of purging devices on the Launcher in flight while securing the draining operations in case of last instants Launch abort. A stringent failure mode and risk analysis has been performed on this critical system to suppress any design issue before the Combined Tests. The time needed for the Cryogenic Connection Systems to disconnect and escape the Ariane 6 lift-off corridor is very short, and the consequences of not achieving this goal may be catastrophic for the Ariane 6 Launch System. For this reason, it has been deemed necessary to validate the systems as early as possible.

Two phases of ECT CCS have been retained:

- Taking profit of the Launch Base Design Authority tests performed on Launch Base Cryogenic Connection System in Fos-sur-Mer (South of France), Launcher System elements have been added in order to perform end-to-end validation tests. The test set-up is presented in Fig. 8. These tests have been successfully performed and have validated operations, interface loads and disconnection, retraction and protection of CCS with pressure on the umbilical lines but in ambient temperature.
- The potential impacts of cryogenic environment on the performance of the system will be seen in CCS cryogenic tests in Launch Zone, planned for the end of 2021. Taking profit of the Central Core mock-up (developed for Launch Base technical qualification tests), it is foreseen to install on it the workhorses that have be used in Fos-sur-Mer tests in order to connect ULPM Cryogenic Connection System

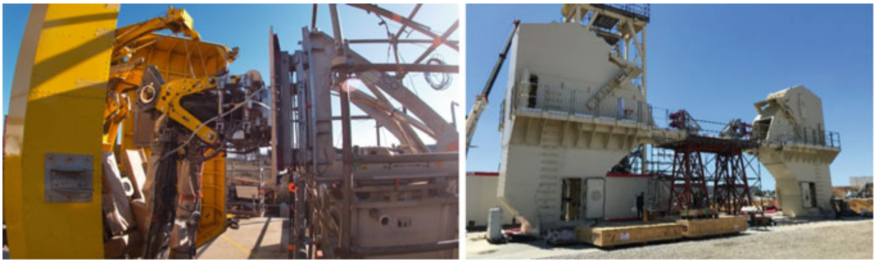


Fig. 8 ECT Fos-sur-Mer test set-up (ULPM on the left, LLPM on the right)

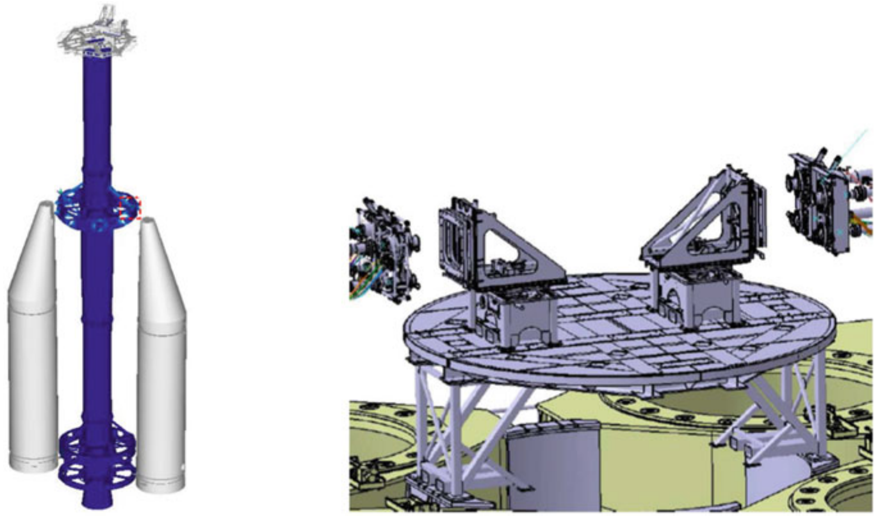


Fig. 9 ECT Launch Zone test set-up (ULPM on the left, LLPM on the right)

on the correct position. Umbilical test mock-ups will be installed on the "board" part in order to circulate LH2 and LOx propellant through umbilicals, in order to achieve cryogenic environment. For LLPM, workhorses will be installed on Launch Zone palettes. Similar umbilicals mock-ups will be used on LLPM to get cryogenic environment. The test set-up is presented in Fig. 9.

3.1.3 Central Core Deployment and ESRs Mating in Launch Zone

The main objective is to validate integration of Ariane 6 boosters (ESR) and Central Core in Launch Zone for A62 configuration by demonstrating the integration predictability, feasibility and repeatability with nominal and extreme geometrical defects, together with a safe disassembly operation.

It is considered that one of the most complex operations on ground is the ESRs mating on the Central Core. For this reason, it was decided to mitigate Combined Tests risks by validating ESR/Central Core integration by means of ESR and Central Core mock-ups as early as possible.

Taking profit of Launch Base technical qualification test campaign, for which a mock-up of the Central Core has been built and is used for the ground means qualification, the validation of the ESR/Central Core mating has been successfully performed end of 2020 in A62 configuration. The test set-up is presented in Fig. 10.

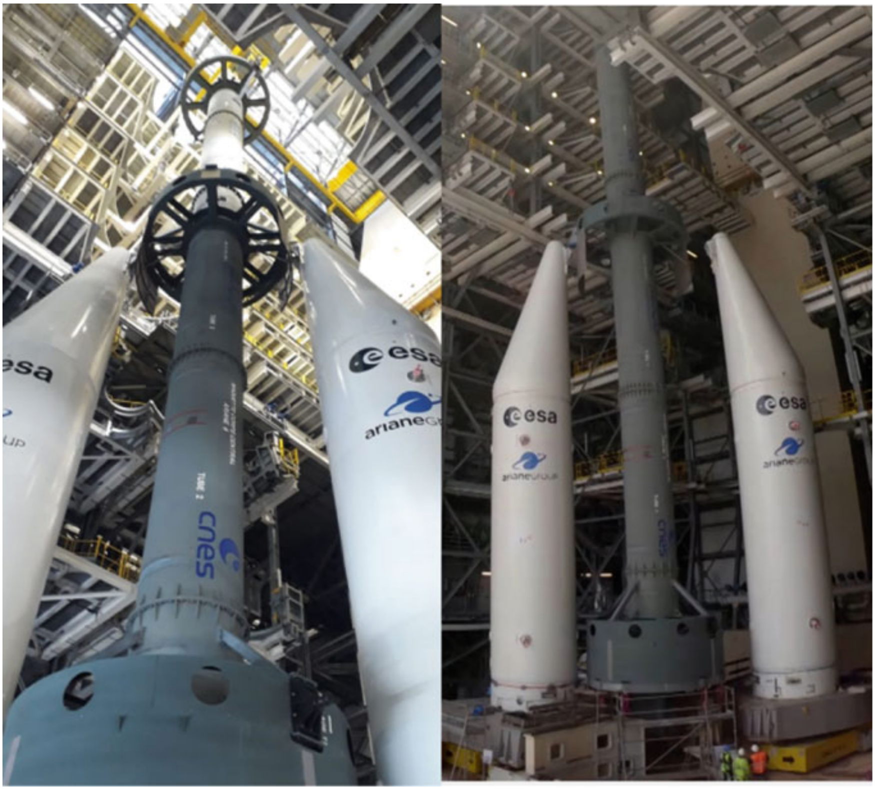


Fig. 10 ECT ESR/Central Core integration test set-up ©ESA/CNES/Arianespace

3.2 Combined Tests

The combined Tests is the Flight Model 0 (FM0) campaign of Ariane 6 which provides a sound basis to enter in the industrial flight campaign from Flight Model 1 (FM1) onwards and giving the foundations to be *Right first time and Ride all times*.

Ariane 6 project is based on a single Combined Tests campaign and a single maiden flight: *Right first time* is a must, tests shall be comprehensive and fully representative of the full operational domain of the Launch System.

The Combined Tests campaign has started in September 2021.

3.2.1 Combined Tests Organisation

The Combined Tests is powered with an intercultural and international team united towards the Maiden Flight being Right first time.

The overall organisation of the different teams participating to the Ariane 6 Launch System Combined Tests is based in the well-known three levels organisation: Operational, Engineering and Decision Authority.

The **Operational Level** is ensured by the Test Conductor Operational Team. It is made of Assistants, Specialty Chiefs and operators in all required Launcher and Launch Complex concerned operational fields (e.g. electrical, fluids, mechanical, propulsion, software, pyrotechnics, launch complex) and respective quality officers. It encompasses two major components:

- The Launch Base Exploitation and Maintenance team: operators from industrial companies resulting from the Launch Base development phase, and also during the Combined Tests to perform operational and maintenance activities;
- The Launcher System Assembly, Integration and Tests team: operators from ArianeGroup for the operations linked to launcher integration, control and tests.

The main tasks of the Operational Level teams are:

- Establishing the Combined Tests specific Operations Plan and the associated Working Instructions, Applicative Software and Operations Sheets;
- Preparing the files for the safety submission for operational phase;
- Conducting the Combined Tests operations.

The **Engineering Level** is ensured by the ESA Combined Tests Manager team together with the Launch System Architect team and the both segments Design Authorities teams. It is made of system and design engineers in all required Launch System, Launcher System and Launch Base concerned definition and performance fields (e.g. electrical, fluids, mechanical, propulsion, software, launch complex, launch range) and respective quality officers. It encompasses four major components:

- Combined Tests Manager ESA team: engineers from ESA (and its Technical Assistance) responsible of the overall coordination of the Combined Tests campaign.
- Launch System Architect team: engineers from ESA (and its Technical Assistance) responsible of the Launch System coherence, who have the authority w.r.t. the Launch System Performances (including Dependability and Safety) and in particular concerning the Launch System and the Launcher to Launch Base interfaces.
- Launcher System Design Authority team: engineers from ArianeGroup (and its industrial partners) responsible of the Launcher System development (including its qualification), who have the authority w.r.t. the Launcher design and in particular concerning the Launcher "limitations of use".
- Launch Base Design Authority team: engineers from CNES (and its industrial partners) responsible of the Launch Base development (including its qualification) who have the authority w.r.t. the Launch Base design and in particular concerning the "limitations of use".

One pillar of the Engineering Level teams is the Ariane 6 Dependability and Safety approach. It aims to identify the exhaustive list of Launch System risks and mitigate them as early as possible, and tests the associated safety barriers in their representative operational use during the Combined Tests. In this frame, a Design-to-RAMS approach has been implemented since the beginning of the project to identify unacceptable failure conditions and hazards scenarios at the earliest, and challenge the Launch System design to fit CSG regulations. This is a pluri-disciplinary and joint effort. Dependability and Safety co-engineering sessions are organized at Launch System level via a dedicated mandate in order to perform the necessary risk/hazard analyses, and implement the required barriers that can be intrinsic to the systems, and/or additional operational means and procedures. The ultimate goal is to mitigate unacceptable failure conditions and hazard scenarios down to an acceptable level. The Dependability and Safety co-engineering team is led by the Launch System Architect team (ESA), and involves the project ArianeGroup and CNES teams. Additional independent Launch System Architect team (ESA) evaluation of the risk level criticality is also performed in order to feed the decision-making process at Launch System management level.

The main tasks of the Engineering Level teams are:

- Establishing and maintaining the Tests Plan, Tests Requests, Measurements Plan and Operational Requirements,
- Analysing the Combined Tests results data and assessing the level of achievement of their respective objectives,
- Approving the test campaigns releases and closures through Test Readiness Reviews and Test Review Boards,
- Establishing the updated versions of the Launch System Dependability and Safety files (i.e. Risk/Hazard analyses, requests for Safety submissions, Conformity files to French Space Law) to demonstrate ultimate compliance to CSG regulations and get the final approval for the overall safety critical systems to be operated at CSG for the standard launch campaign starting with the Maiden Flight.

The **Decision Authority** is ensured by a Steering Committee made of the appropriate Chief Engineers and Managers from the respective entities. They have the final decision for the release of the major tests campaigns and in case of major events during the tests campaigns or of the on-going qualification activities (other than the Combined Tests) (Fig. 11).

3.2.2 Combined Tests Objectives

As the Combined Tests is the Flight Model 0 (FM0) campaign, by definition it allows to de-risk the following launch campaigns thanks to the systematic and strict troubleshooting carried out during the Combined Tests.

And with the LLPM Long Firing Test the first part of the flight (using Vulcain 2.1 engine) will be tested on ground with all the critical functions.

The Launch System Combined Tests aim to conduct a typical Launch Campaign, testing the Launcher System and the Launch Base in representative conditions of

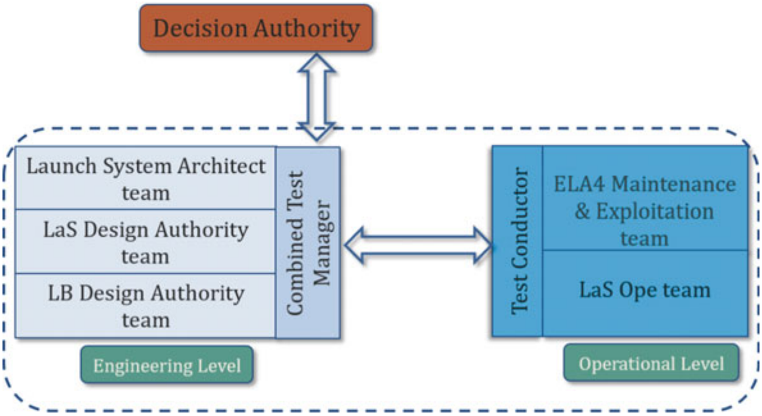


Fig. 11 Ariane 6 launch system combined tests organisation

the overall launch campaign domain, and also introducing specific Combined Tests objectives in order to:

- Reach the maximum level of performances of the critical systems;
- Train teams for potential contingencies, testing off-nominal/degraded cases;
- Verify that all the risks have been mitigated down to an acceptable level of safety with consolidation of the proper dependability and safety margins to guarantee the safety of people and its property, and with demonstration of the compliance to the applicable regulations (e.g. French Space Law).

The Launch System verification and validation logic is based on the classical System Engineering Approach using Function Validations and Requirement Verifications, and can be split in three pillars:

- Phase approach: Launch System Ground Life and Operations;
- Functional Analysis: Function definition and Validation and Control;
- Requirement: Requirement Cascading and Verification and Control.

The objectives of the Combined Tests campaign were therefore defined according to these three pillars: Fig. 12

- Operational validation: The objective of the Combined Tests is to perform and validate all the operational products compliant of the Launch System Operational Concept (Sect. 2.3) required for the Operational Qualification considering nominal and degraded cases.

  To guaranty the competitiveness of Ariane 6, already in the Combined Tests campaign, the operational performance has to be challenged. In this aim, the duration of each test sequences are measured and the resources are optimised.

  Furthermore, the new Launch Base infrastructures including newly developed ground means will prove their efficiency during the Combined Tests.

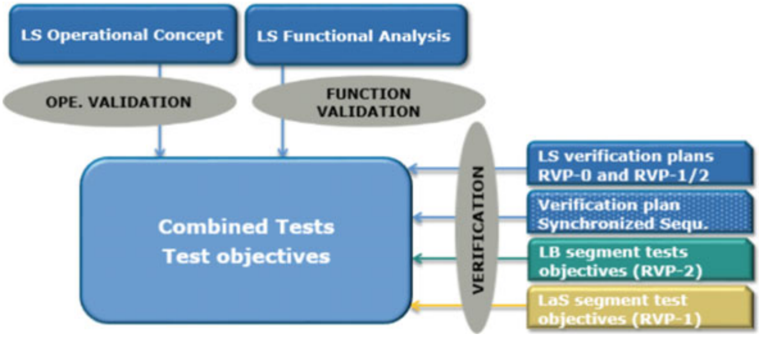

LS Operational Concept
LS Functional Analysis
OPE. VALIDATION
FUNCTION VALIDATION
Combined Tests
Test objectives
VERIFICATION
LS verification plans RVP-0 and RVP-1/2
Verification plan Synchronized Sequ.
LB segment tests objectives (RVP-2)
LaS segment test objectives (RVP-1)

Fig. 12 Ariane 6 LS combined tests objectives

On top, the customisation of the Launcher for the profit of the customer will be encompassed within the Combined Tests: modularity of the launcher between A64/A62 configuration and the flexibility of the mission by changing payload to answer to late customer needs.

- Functional validation: The functional validation of the Launch System is acquired by the exhaustive coverage of the functions identified in the Launch System Functional Analysis for which the Combined Tests are the main contributors.
- Verification of requirements as required by the Launch System Verification Plans and by the Launcher System and the Launch Base Verification Plan. These verifications deal with (but are not limited to):
  - the verification of interfaces between Launcher System elements and the Launch Base installations (mechanical, electrical, fluidic),
  - the verification of environmental requirements,
  - the verification of performance requirements.

3.2.3 Tests Sequences

The main objectives of the Combined Tests being the validation of the Launch System operational product, the test campaign shall be comprehensive (nominal and degraded cases) and fully representative of the full operational domain (limited adaptations). In this sense, the test sequence was defined from a launch campaign adapted in order to embed performance validations and verifications, thus covering the total of the test objectives.

The Ariane 6 Combined Tests is split in six sequences, whose limits correspond to a safe steady state of the Launcher and the Launch Base (Fig. 15).

- The Dummy Payload Encapsulation Test (DPET) is the test campaign of the Upper Composite integration in BAF with specific tests dedicated to:
  - RF and EMC characterization tests,
  - Electrical bonding/grounding verification.

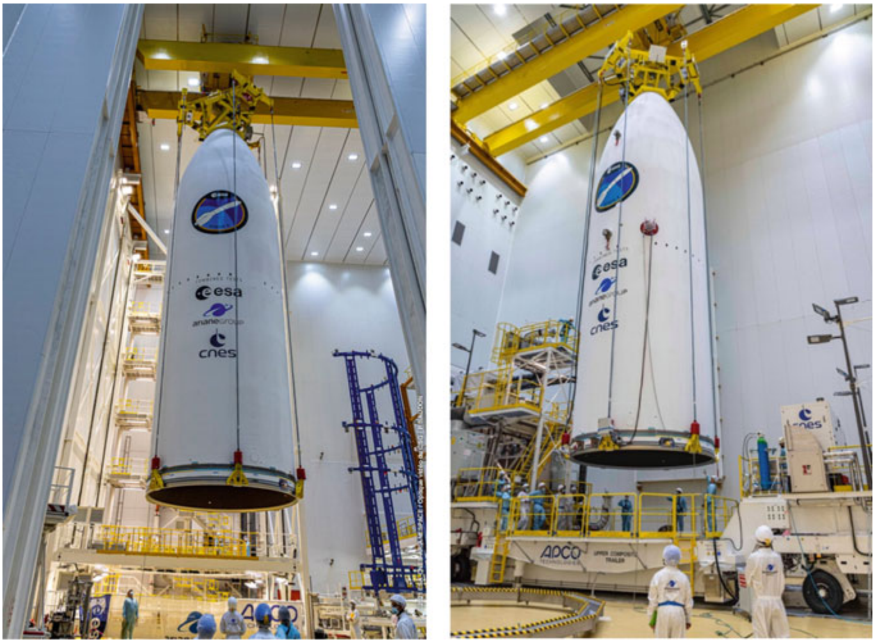


Fig. 13 Dummy payload encapsulation test (DPET) ©ESA/CNES/Arianespace

DPET has been successfully performed in September 2021 (Fig. 13), paving the way to the next phase.

- Combined Test Launcher Integration (CTLI) is the test campaign of the Launcher integration in BAL and in Launch Zone with specific tests dedicated to:
  - Check Out Logic justification,
  - Launcher functional qualification: sign tests and other specific avionics and fluidic tests,
  - Structural leak and conditioning verification of launcher cavities.

  ESR, Central Core and UC transfers validation are also part of this test campaign (Fig. 14).
- The Combined Test LOading (CTLO) and Hot Firing (CTHF) are the test campaigns of the launch chronology in Launch Zone and the LLPM Long Firing Test with specific tests dedicated to:
  - RF and EMC tests characterization tests;
  - Launcher cavities ventilation validation;
  - Chronology/synchronised sequence/loading/draining validation;
  - Vulcain 2.1 ignition environment validation and start-up sequence validation;
  - CCS disconnection validation in ambient and in cryogenic environment;
  - Umbilical disconnection and swivelling validation;

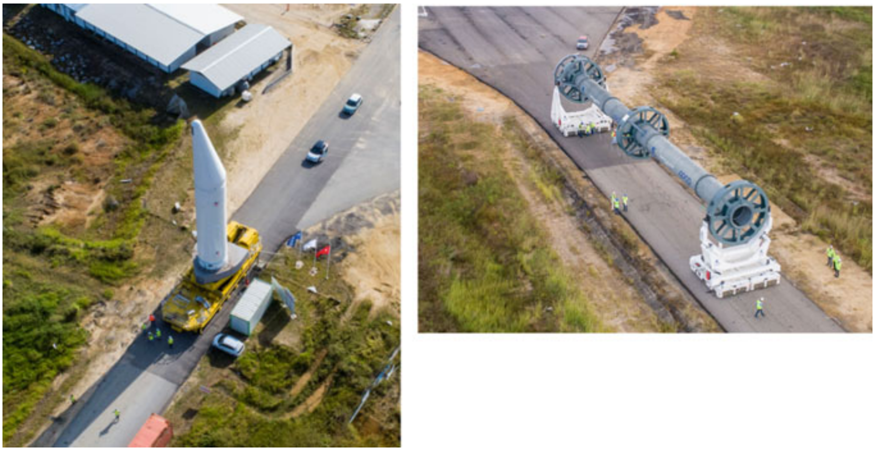


Fig. 14 Transfer test for LB transfer means technical qualification ©ESA/CNES/Arianespace

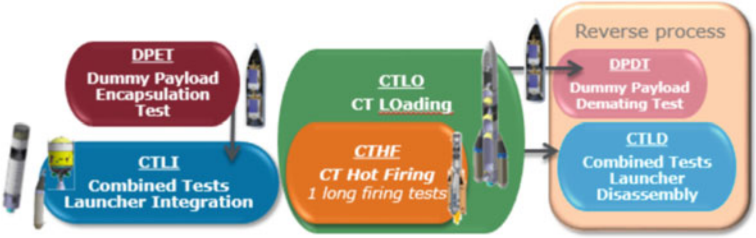

DPET
Dummy Payload Encapsulation Test
CTLI
Combined Tests Launcher Integration
CTLO
CT LOading
CTHF
CT Hot Firing
1 long firing tests
Reverse process
DPDT
Dummy Payload Demating Test
CTLD
Combined Tests Launcher Disassembly

Fig. 15 Overall sequence of the Ariane 6 launch system combined tests

- Qualification of ESR/Central Core mating in A64 configuration.

It is required to perform a least four launch chronologies (loading/deloading): all the draining types (nominal, degraded, emergency) shall be validated during the Combined Tests; therefore, four draining tests are required. The two first chronologies and the Long Firing Test will be performed in A64 configuration and the two last chronologies in A62 configuration.

- The Dummy Payload De-mating Tests (DPDT) is the test campaign of the Upper Composite de-mating in BAF.
- The Combined Test Launcher Disassembly (CTLD) is the test campaign of the Launcher disassembly in Launch Zone and in BAL.

3.2.4 Combined Tests Launcher Model

Test Like You Fly, Fly Like You Test.

In this spirit, the Combined Tests Launcher is a Flight Launcher that overshoots the target for a Flight Model 0 (FM0) campaign.

The Combined Tests Launcher System is composed with:

- Three ESRs mock-up dubbed Pylons equipped with avionic (Fig. 16);
- One ESR with inert propellant and a fixed nozzle;
- One Upper stage ULPM with a non-functional Vinci engine;
- One Lower stage LLPM;
- One Upper Composite composed with long fairing customised for the Combined Tests (Radio Transparent Windows) and one dummy payload (Fig. 16);
- Specific sensors added for the Combined Tests for supporting the qualification activities: almost 700 sensors are installed on the Combined Tests Launcher in addition to the generic functional instrumentation.

The slight adaptations (i.e. inert ESR, non-functional Vinci engine of the Upper Stage) have been agreed for safety reasons, knowing that they do not impact the level of representativeness required for reaching the Launch System qualification and validation objectives.

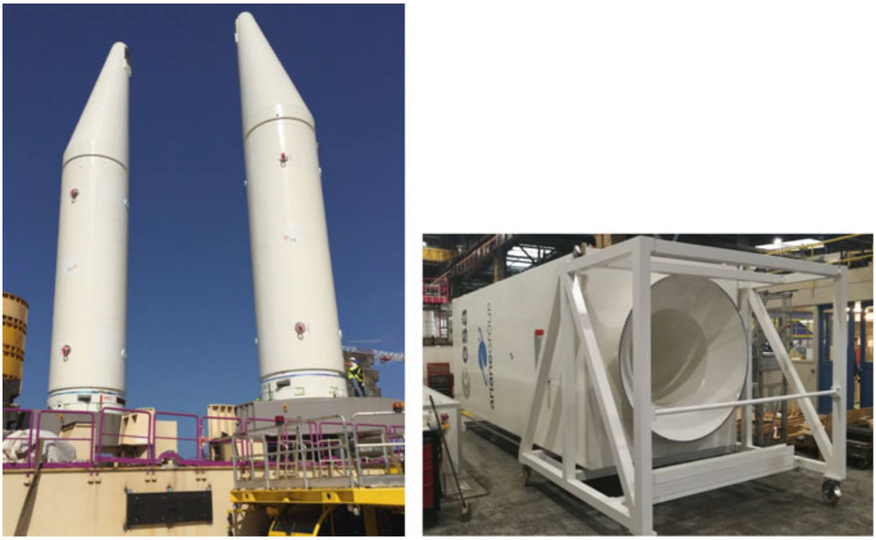


Fig. 16 ESRs mock-ups and dummy payload ©ESA/CNES/Arianespace/ArianeGroup

3.2.5 Launch Complex Adaptations for the Combined Tests

Adaptations of the ground segments are required for the Combined Tests implementation, most of them for the Long Firing Test.

These adaptations have been identified:

- To manage the effect of Vulcain 2.1during the Long Firing Test. Specific Launch Zone protections and specific cooling system have been developed and will be installed for the Long Firing Test.
- To comply to the Dependability and Safety requirements during the Combined Tests and in particular during the Long Firing Test of the Vulcain 2.1. The associated adaptations are:
  - Design a specific firing test safety system for the management of the safety redlines to be monitored;
  - Additional specific analyser lines for detecting potential propellant leakages in the Vulcain 2.1 aft bay.
- To implement specific ground measurements means (e.g. sensors, cameras) for the acquisition function, specific acquisition/archiving systems and specific real time visualisation during the chronologies and the Long Firing Tests.

4 Conclusions

The results achieved so far demonstrate the well-funded of the Ariane 6 development methodology, notably the feasibility of running in parallel various threads of development activities at different levels of the system product breakdown structure while mastering the risks due to such parallelization of activities.

The paradigm test as soon as possible, test at lowest possible level is shown to be effective and makes feasible to master concurrent design of products and operations across contractual boundaries. Moreover, it allows to implement a continuous learning and improvement of operational performance which extremely useful during the development cycle further than being an asset for the achievement of a smooth insertion of the launch system on the market.

The overall Launch System Combined Tests have been presented together with the methodology used for its definition, the early identification of features to be implemented in test specimen and test benches configurations, the definition of comprehensive test objectives. The adoption of Early Combined Tests is a pillar in the development strategy and allows to anticipate the mitigation of any development risk by allocating resources before ahead. That choice is a game changer with respect to the pure waterfall development in which recovery actions are undertaken instead of risk mitigation actions.

The very significant heritage and lessons learnt from previous similar development is another key element of the Ariane 6 test logic and this key element is as much important as Ariane 6 presents some novelty with respect its predecessors. This principle has inspired the definition of accurate testing of each phase including nominal and degraded conditions.

The great attention dedicated to the quality of service offered to future Customer completes the list of pillars that have generated the presented test logic.

The Combined Tests pave the way for the Ariane 6 Maiden Flight foreseen in the second semester of 2022.

Acknowledgements The authors wish to thank ArianeGroup, CNES and ESA teams for their cooperative work effort and their contribution to elaborate this paper.

References

1. Resta PD, Pilchen G, Coulon D et al (2016) The Ariane 6 launch system development status. In: IAC-16-D2.1.35037, 67th international astronautical congress (IAC), Guadalajara, Mexico, 26–30 September 2016
2. Resta PD, Monreal JA, Pouffary B, Lemercier S, Decadi A, Arnoud E (2018) Ariane 6 launch system operational concept: main drivers. In: 15th international conference on space operations (SpaceOps 2018), Marseille, France, May 27–June 01 2018

Orbital and Attitude Control of Spectr-RG Observatory Under Technical Constraints

Natan Eismont, Irina Kovalenko, Vladimir Nazarov, Fedor Korotkov, Maksim Pupkov, Vladislav Zubko, Andreay Poghodin, Pavel Mzhelskiy, Evgeniy Mikhailov, Aleksey Ditrikh, and Andrey Tregubov

Abstract The Spectr-RG space observatory was launched from Baikonur on July 13, 2019, and today is orbiting in the vicinity of the Sun-Earth libration point L2. The planned lifetime of the mission is 6.5 years and includes all-sky survey and pointed observations in the 0.3–15 keV band with the goal to create an X-ray map of the universe. This paper describes technical constraints of the mission, covering launch

N. Eismont · I. Kovalenko (✉) · V. Nazarov · F. Korotkov · M. Pupkov · V. Zubko
Space Research Institute of Russian Academy of Sciences, 84/32 Profsoyuznaya St., Moscow 117997, Russia
e-mail: irina.kovalenko@iki.rssi.ru

N. Eismont
e-mail: neismont@iki.rssi.ru

V. Nazarov
e-mail: vnazarov@romance.iki.rssi.ru

F. Korotkov
e-mail: fkor@romance.iki.rssi.ru

M. Pupkov
e-mail: pupkovmv@student.bmstu.ru

V. Zubko
e-mail: v.zubko@iki.rssi.ru

A. Poghodin · P. Mzhelskiy · E. Mikhailov
Lavochkin Association, 24 Leningradskaya Str., Khimki, Moscow Region 141400, Russia
e-mail: snorry@laspace.ru

P. Mzhelskiy
e-mail: mzhelskiy@laspace.ru

E. Mikhailov
e-mail: mikhaylov@laspace.ru

A. Ditrikh · A. Tregubov
Russian Rocket and Space Corporation Energia, Korolev, Moscow Region 141070, Russia
e-mail: avditrikh2@gmail.ru

A. Tregubov
e-mail: andrey.ait@yandex.ru

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_24

scenario, spacecraft design, ground data relay network, orbital and attitude control systems. First, the orbit correction scheme, applied for communication requirements, is presented. Then, the analysis focuses on a trade-off solution between attitude and orbit control operations. The proposed strategy allows mission to minimize the total propellant consumption by coordinated reaction wheels offloading and reducing the number and cost of the station-keeping manoeuvres to maintain the nominal orbit.

Keywords Spectr-RG · Lagrange point orbit · Attitude control · Station-keeping

Acronyms/Abbreviations

ART-XC	X-ray Space Telescope Designed by IKI RAS
CR3BP	Circular Restricted Three-Body Problem
Delta-v	Velocity Increment
DM3	Fourth Stage of the Proton-M Launch-Vehicle
eROSITA	X-ray Space Telescope Designed by MPE
IKI RAS	Space Research Institute of the Russian Academy of Sciences
LPO	Lagrange Point Orbit
MPE	Max Planck Institute for Extraterrestrial Physics
SEL2	Sun-Earth Lagrange point 2
SRG	Spectrum-Roentgen-Gamma

1 Introduction

Spectr-RG (SRG) is an international X-ray astrophysical observatory, designed by the Space Research Institute of the Russian Academy of Sciences (IKI RAS) in collaboration with Max Planck Institute for Astrophysics (Germany). The spacecraft platform, titled Navigator, was designed by Lavochkin Research and Production Association. SRG was launched on July 13, 2019, from the Baikonur cosmodrome by Proton-M launch vehicle with DM3 upper stage [1]. Three months after the launch, SRG achieved its operational orbit in the vicinity of the Sun-Earth collinear libration point L2 (SEL2), located at a distance of about 1.5 million km from the Earth. The SEL2 orbit choice was determined by its stable thermal conditions and geometry with respect to the Sun, Earth, and Moon, which allows telescopes on board the spacecraft to observe all-sky while avoiding perturbations of these bodies.

The main scientific goal of the SRG mission is to create an X-ray map of the universe, produced by an all-sky survey and pointed observations in the 0.3–15 keV band. The mission is aimed at detecting and observing about 100,000 galaxy clusters, all obscured accreting black holes in nearby galaxies and more than a million of new active galactic nuclei and quasars.

The baseline payload consists of two X-ray telescopes:

- eROSITA (Max Planck Institute for Astrophysics, Germany)
- ART-XC (IKI RAS, Russia).

The full scientific program is planned for 6.5 years: the first four years for the all-sky survey and the remaining time for the point survey of individual sources.

2 SRG Design and Mission Constraints

The mission requirements and constraints are defined by the scientific measurements planned for the onboard instruments, and the technical possibilities of the Navigator platform. The Navigator platform, adapted for SRG from previous missions, has proved its high reliability. The Spectr-R mission is an example which successfully maintained for about 8 years, from 2011 to 2019, on a highly elliptical orbit with the apogee up to 339,000 km, and the perigee of no less than 500 km (the orbital period was around 7.2 days). Another example is the constellation of three Electro-L spacecraft launched into geostationary orbit.

Let us introduce a coordinate system associated with the spacecraft: X_S axis is directed along the line of sight of the telescopes, the Y_S axis is parallel to the rotation axis of the solar panels, and the Z_S axis completes the right-handed set. The spacecraft architecture is shown on Fig. 1.

The technical constraints of the SRG mission are the following:

- **Telemetry data transmission and commands reception**. The medium-gain antenna on board SRG is fixed relative to the spacecraft and is oriented so that its axis coincides with the Z_S axis. The angle between the axis and the direction

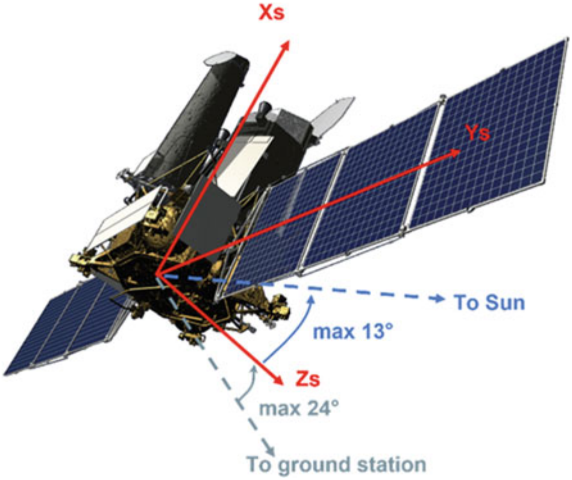


Fig. 1 SRG spacecraft: X_S axis is directed along the line of sight of telescopes, Y_S axis is parallel to the rotation axis of the solar panels, Z_S axis completes the right-handed set

to the ground station, with the distance between the station and SRG is less than 1.8 million km, must not exceed 24° (see Fig. 1). Fulfilling this requirement, the telemetry system provides data relay to the ground station with a rate of at least 512 kbit/s.

- **Thermal conditions**. The thermal insulation of instruments and onboard radiators requires a controlled removal of heat flows. This last condition constrains the angle between the $X_S Z_S$ plane and the direction to the Sun, which must be in the range of 13° (see Fig. 1).
- **Eclipse avoidance**. The orbit choice has to ensure that the spacecraft doesn't fall into the Earth's or Moon's shadows.
- **Ground station network**. The scientific data from the telescopes are initially recorded on board SRG, and then transmitted to ground stations, when the spacecraft is visible. For this purpose, the three ground stations are used: Bear Lakes (64 m antenna dish diameter), Baikonur (12 m) and Ussuriysk (70 m) stations—with an operational elevation angle greater than 7°. To ensure the transmission of all the collected data from the spacecraft, the duration of visibility has to be at least 5 h per day.

3 Operational Orbit

An orbit in the vicinity of SEL2 was proposed as the most suitable for the planned scientific measurements. The orbit choice is detailed in the work [2–4]. The operational orbit satisfies the requirements and constraints outlined in the previous section: all-sky survey and point observations, thermal regime, and radio link. An important advantage of the L2 point location is that the Earth, Moon and Sun remain in one hemisphere relative to the spacecraft, and do not create any noise for observations and any noticeable changes in thermal environment, crucial for the onboard scientific instruments.

The libration-point orbits have first been analysed for space missions by Farquhar [5], and the dynamics near the Lagrange points in the frame of the circular restricted three-body problem has been widely described in the literature, e.g. [6–8]. These methods were used for the Relict-2 mission, designed at IKI, and aimed at observation of the cosmic microwave background radiation from the L2 point [9].

Hereafter, for the mission analysis let us use the rotating reference frame, defined as follows: the origin is at SEL2, the X_{L2}-axis is aligned with the Sun-Earth line and directed to the Sun, the Z_{L2}-axis is orthogonal to the ecliptic plane and directed to the North ecliptic pole, the Y_{L2}-axis completes to the right-handed set.

The launch of SRG was primary planned to be on June 21, 2019. Then, it had to be postponed and the spacecraft was launched on July 13, 2019. The nominal trajectory, for the launch on July 13, 2019 is shown on Figs. 2 and 3. This trajectory is shown in projections to XY_{L2}, and XZ_{L2} planes of the rotating coordinate system, and includes the transfer to the nominal operational orbit from Earth, and then, the numerically propagated orbit over 6.5 years.

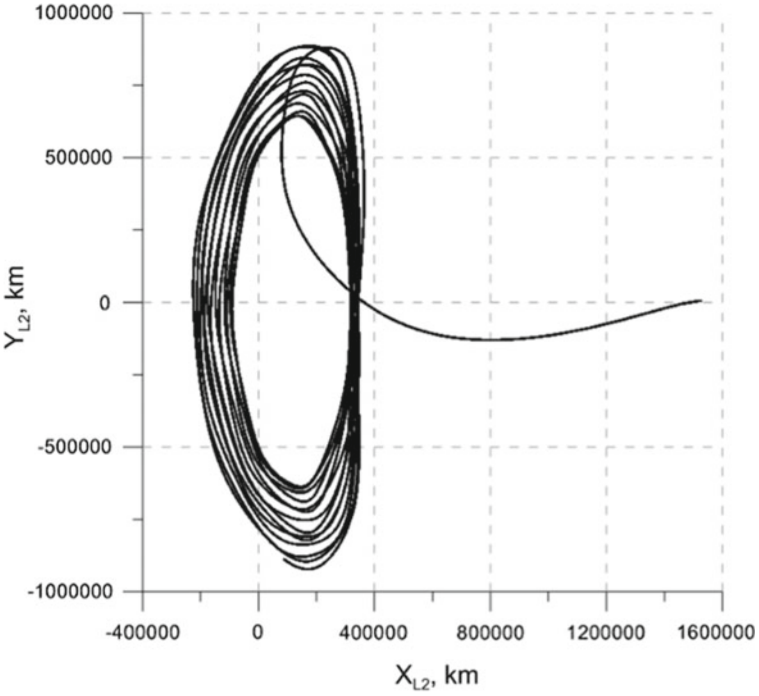


Fig. 2 Projection of the SRG trajectory on ecliptic plane in the Sun-Earth rotating frame over 6.5 years of flight

The choice of trajectory parameters was mostly defined by the requirements on the durations of housekeeping and science data transmission from the spacecraft to the ground stations (Bear Lakes and Ussuriysk), and sending commands to onboard systems. In addition, the tracking operations with the use of the stations and onboard transponders for the range and range rate measurements must be performed. This means that the spacecraft's radio visibility (within the minimal allowed elevation angle above the local horizon of at least 7°) during the mentioned intervals has to be guaranteed during each day of flight.

The change of the launch date has affected the out-of-plane amplitude, along the Z_{L2} axis, and consequently damaged the visibility of the spacecraft from the ground stations.

Figures 4 and 5 show the durations (in hours) of the radio visibility per day for each ground station and the sum of two stations, respectively. The visibility requirements for the first 1000 days of flight are satisfied for duration of visibilities counted as total for the two stations. But it is not the case for the separately considered stations. For instance, the Bear Lakes station is out of the spacecraft visibility from April 28, 2020, to May 18, 2020, and this unfavourable situation would repeat each year. Although the Ussuriysk station can ensure the visibility for these intervals mostly, the

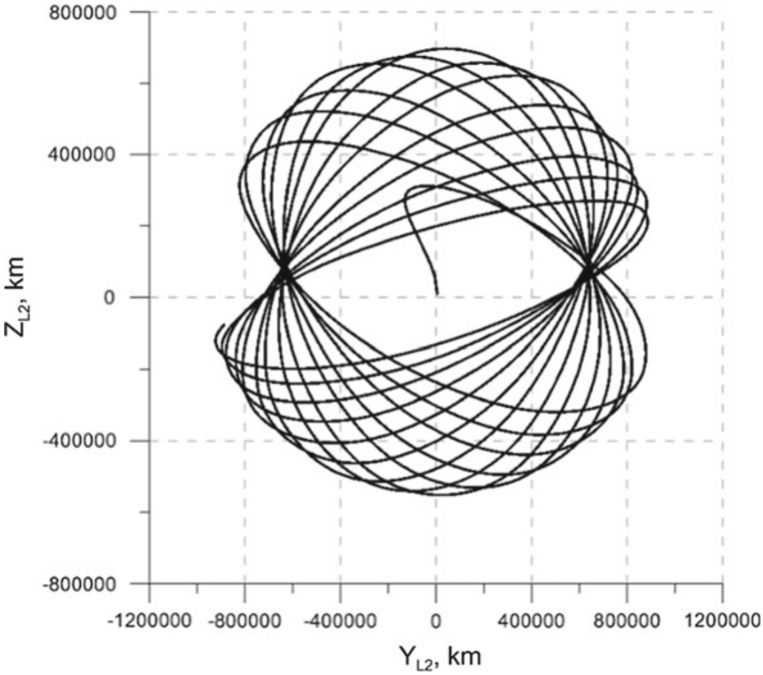


Fig. 3 Projection of the SRG trajectory on YZ_{L2} plane in the solar ecliptic rotating frame over 6.5 years of flight

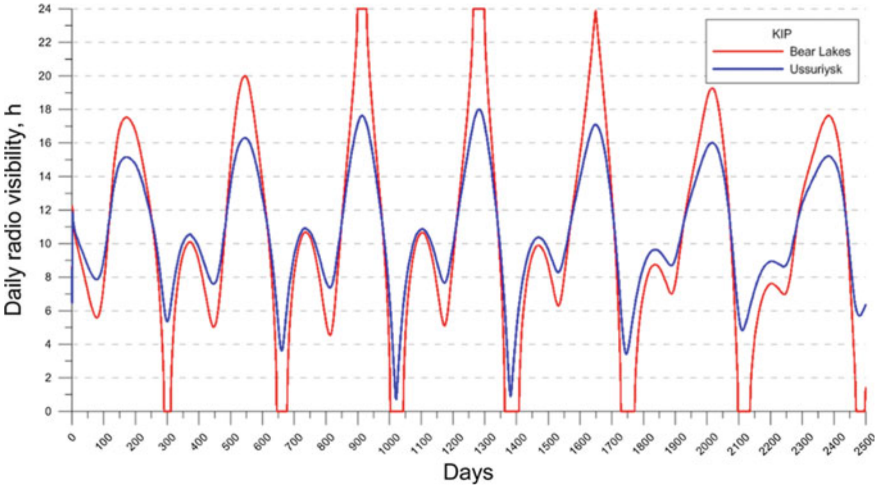


Fig. 4 Visibility duration in hours per day from Bear Lakes (red line) and Ussuriysk (blue line) stations. Days are counted from the SRG launch date: July 13, 2019

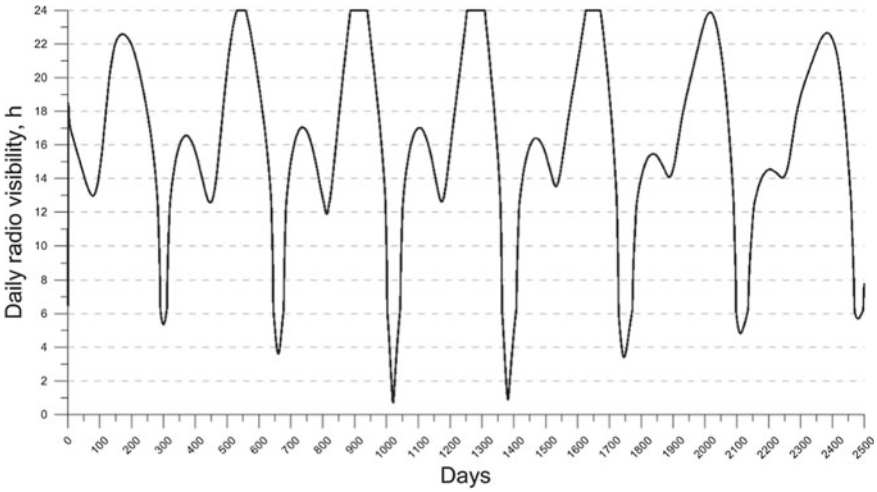


Fig. 5 Total visibility duration in hours per day from Bear Lakes and Ussuriysk stations. Days are counted from the SRG launch date: July 13, 2019

data relay duration would be violated in the spring of 2022 and 2023. To overcome this problem, the following options were considered.

- **Additional stations**. The option with additional ground stations involved the ESA centres in Cebreros (Spain), New Norcia (Australia), and Malargue (Argentina). This solution did not require any orbit change manoeuvres, and consequently any delta-v cost, and did not complicate the mission control operations. Figures 6 and 7 show the durations of the visibility for the Cebreros and New Norcia stations,

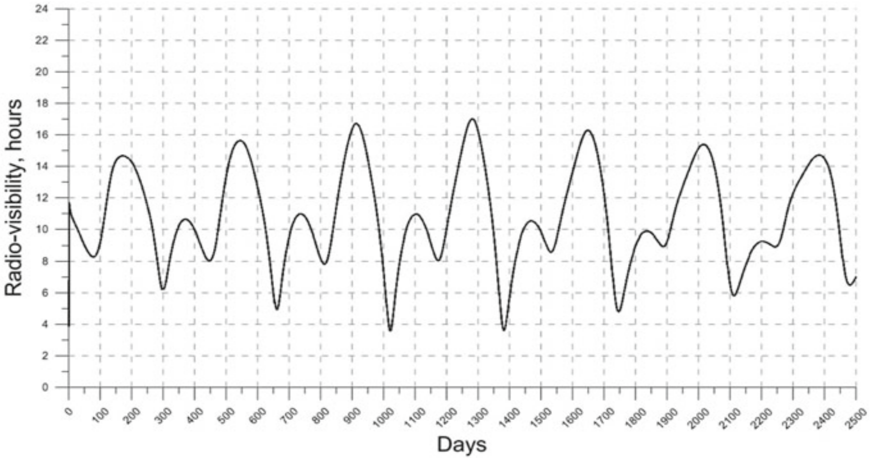


Fig. 6 Visibility duration in hours per day from Cebreros station (days are counted from the launch)

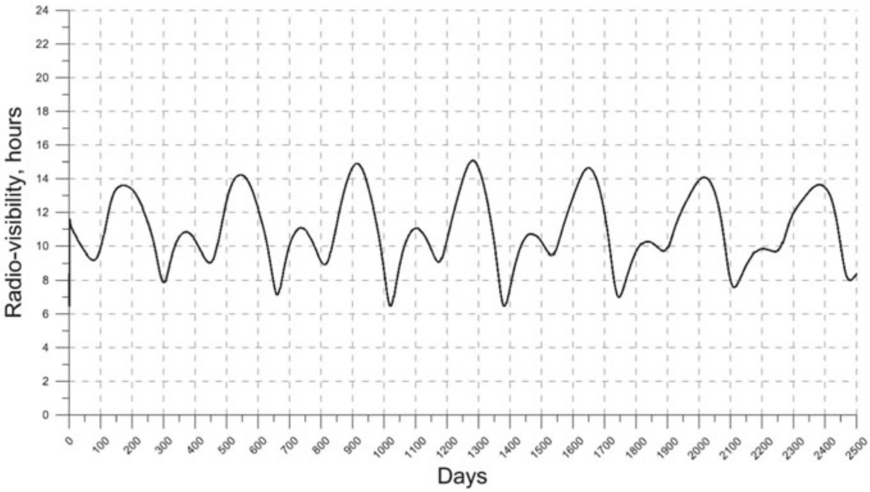


Fig. 7 Visibility duration per day from New Norcia station (days are counted from the launch)

respectively. An important outcome of this solution would be the fact that the stations cannot send commands to the spacecraft. Hence, the following alternative solution with amplitude changing manoeuvres was chosen.

- **Amplitude changing manoeuvres**. This option includes a set of orbit amplitude changing manoeuvres, increasing the visibility in the critical segments of the mission. Each manoeuvre is applied when the spacecraft trajectory crosses the ecliptic plane, in the direction orthogonal to the ecliptic plane [6]. The estimated delta-v cost of this scheme is no more than 65 m/s. Although this solution requires additional delta-v budget and mission control operations, still the available mass of propellant on board spacecraft is high enough to execute further manoeuvres until the end of the mission. This scenario was selected to solve the visibility problem and described in detail in the next section.

4 Amplitude Change Manoeuvres

The first testing manoeuvre of 3 m/s delta-v cost was performed on October 05, 2020, in order to verify a possible pollution of the eROSITA and ART-XC telescopes by the fuel decomposition products. This manoeuvre did not show any damages of the scientific payload, and the next set of operations was planned. Namely, ten subsequent manoeuvres of 6 m/s each have been planned for the following dates: November 23, 2020; February 28, 2021; May 22, 2021; September 01, 2021; November 23, 2021; March 02, 2022; May 22, 2022; September 02, 2022; November 23, 2022; March 02, 2023. This manoeuvres planning can be slightly modified according to further orbit measurements. All the manoeuvres are to be performed near the points where the SRG trajectory crosses the ecliptic plane. The nominal direction of the delta-v impulse is co-directed with the normal vector to the ecliptic plane (when going from the northern to the southern hemisphere) or counter-directed to it (when going from the southern to the northern hemisphere) [6, 10].

Currently, the small testing (October 05, 2020) and the four "big" corrections (23.11.2020, 28.02.2021, 23.05.21, 06.09.21) were performed. Figure 8 shows positions and directions of the manoeuvres on the spacecraft trajectory, and also the propagated trajectory after these manoeuvres. Hence, after performing only these five manoeuvres, the out-of-ecliptic amplitude of the orbit has reduced by about 50,000 km. Figure 9 demonstrates the improvement of the visibility between the initial and corrected orbits.

Figure 10 shows the computed trajectory between all the ten amplitude correction manoeuvres (blue line), the trajectory after manoeuvres with only station-keeping operations (black line), and the initial trajectory (grey dashed line).

The proposed scheme of the set of ten orbit changing manoeuvres reduces the out-of-ecliptic amplitude by about 100 000 km, and consequently enables the daily radio visibility from Ussuriysk station over at least 4 h during all the planned mission

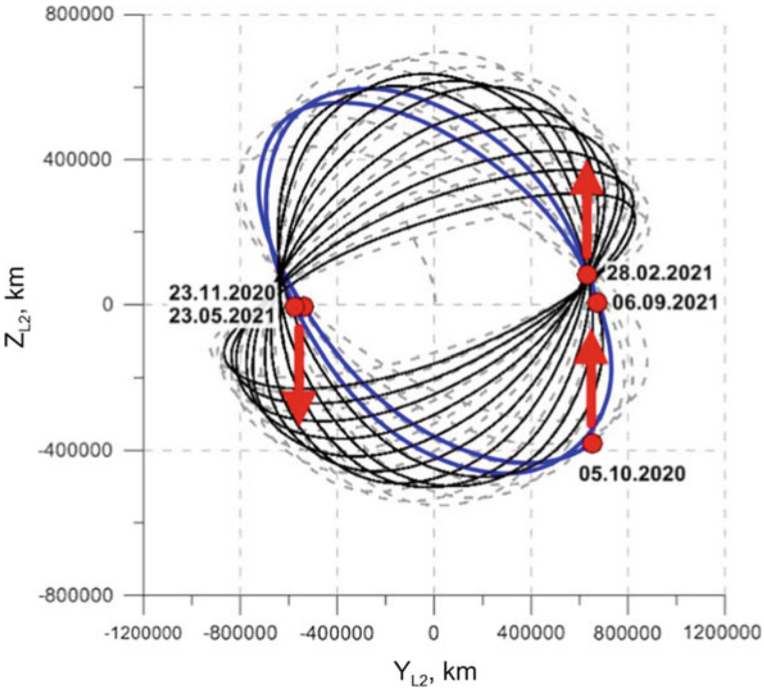


Fig. 8 Dates (in format "day.month.year"), positions and directions of the five first amplitude changing manoeuvres (red arrows). Blue line is the real SRG trajectory, black line—the numerical propagation after these corrections, grey dashed line—the initial SRG trajectory

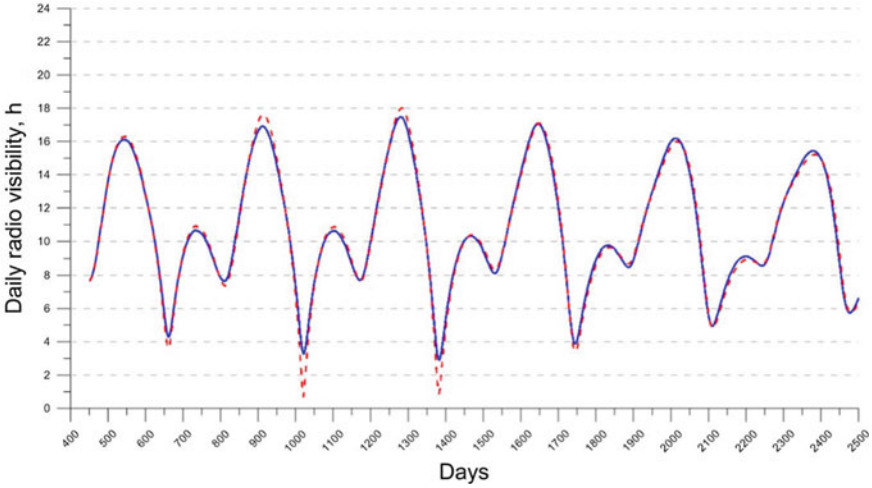


Fig. 9 Daily visibility duration from the Ussuriysk station: red line—initial visibility, blue line—visibility after the first five manoeuvres of the orbit change

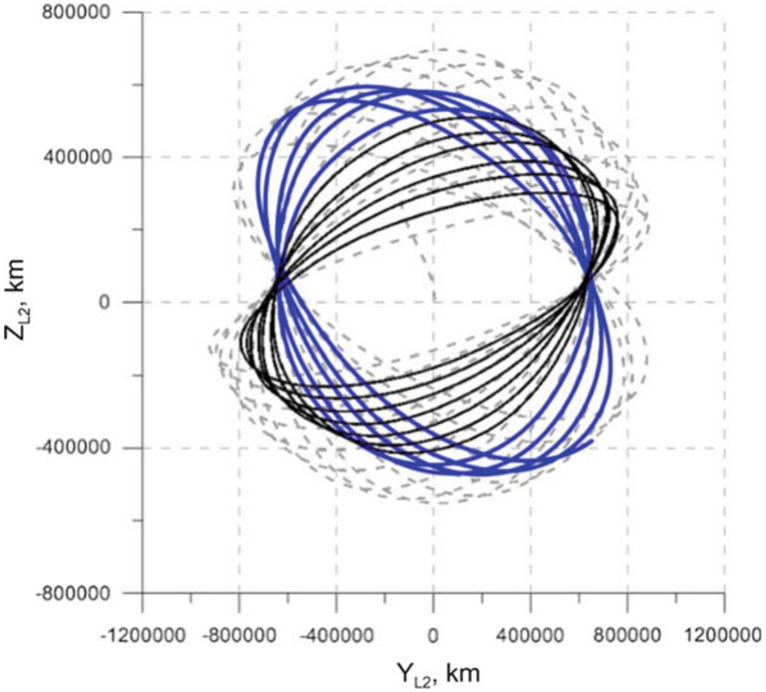


Fig. 10 Computed SRG trajectory between the ten amplitude changing manoeuvres (blue line), trajectory after these corrections (black line), and the initial SRG trajectory (grey dashed line)

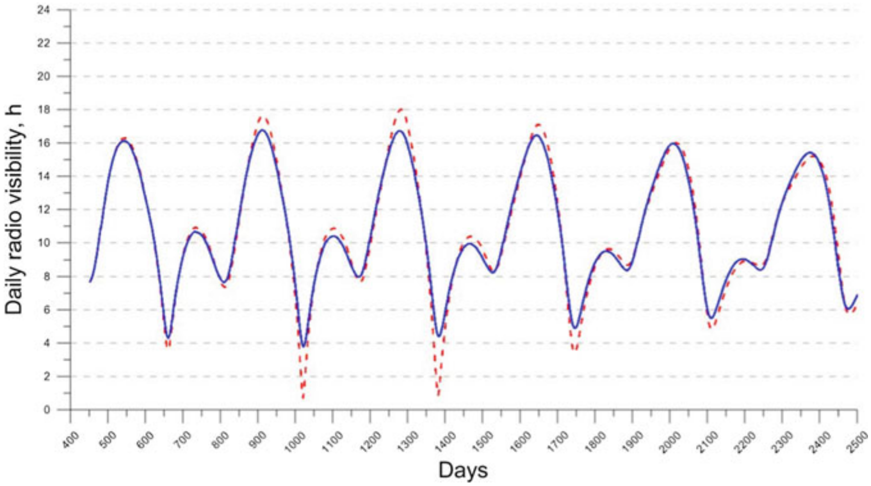


Fig. 11 Daily visibility duration from the Ussuriysk station: red line—initial visibility, blue line—visibility after the ten planned manoeuvres of the orbit change

duration. Figure 11 demonstrates the difference in the visibility between the initial and corrected orbits.

The proposed set of orbit corrections allows mission to fulfil the requirements on the visibility of the SRG spacecraft for scientific data and operational commands relay. The estimated total delta-v cost of these manoeuvres is not exceeded 65 m/s which is acceptable for the total delta-v budget of the mission.

5 Attitude Control Strategy

5.1 Reaction Wheels Desaturation Scheme

To estimate the minimal delta-v required to carry out the mission, let us analyse two types of spacecraft motion control manoeuvres.

The first type are station-keeping manoeuvres to maintain the spacecraft in its operational LPO. These manoeuvres are required due to the instability of the L2 point, and errors of initial orbit insertion and influence of attitude control system operations. In other words, this type of manoeuvres is a part of orbital control. The second type of manoeuvres is a part of the attitude control. The attitude control system includes rocket engines which generate the control torques, but also orbit perturbing forces while desaturating the attitude control reaction wheels (see Fig. 12).

We analyse the possibilities to control these forces simultaneously in order to optimise the total delta-v manoeuvres cost. Our goal is to construct a strategy, allowing

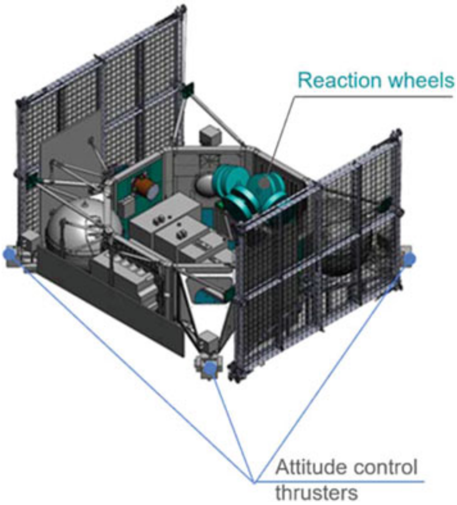


Fig. 12 Navigator-platform [11], the attitude control system of the SRG spacecraft

spacecraft to minimise the perturbations produced by the attitude control operations, and instead use this effect for the L2 orbit station-keeping.

The orbital motion in the vicinity of a libration point has a significant feature: the existence of a direction which is the most sensitive to perturbations. Namely, perturbations along this direction lead to the fastest escape trajectory from the libration-point orbit. At the same time, delta-v component along this direction allows manoeuvres to control the LPO stability. Consequently, the optimal station-keeping manoeuvres to maintain the spacecraft on an orbit about SEL2, have to be applied along this direction. This so-called escape direction $\boldsymbol{u}$ [10], in the Sun-Earth system lies in the ecliptic plane at 28.6° from the X_{L2} axis (see Fig. 13), and does not depend much on the point in orbit.

The forces, produced by the attitude control system, are directed along the X_S axis. At the same time, the thrust, produced by the orbital control engine units, is also directed along the X_S axis. The duration of the offloading wheels is short enough, about 10 s, and is assumed in this analysis as instantaneous thrust.

The systematic X-ray sky survey observations began on December 12, 2019. For these operations, the telescopes axis along X_S of spacecraft rotates with the 4-h period around of spacecraft Z_S axis, which roughly follows the Sun direction. Thus, in half of a year the telescopes scan the whole celestial sphere. This survey will last 4 years during the first phase of the mission, resulting in the full X-ray map of the universe.

The minimal perturbation from the reaction wheels desaturating would be at the moment, when the Z_S rotation axis lies in ecliptic plane and the X_S-axis is orthogonal to the ecliptic plane. On the other hand, these deviating forces can be used as the orbit correction manoeuvres, when the X_S axis during its rotation reaches the plane of ecliptic and its projection to the vector $\boldsymbol{u}$ is the positive or negative, depending on the required direction of the correction impulse (Fig. 13).

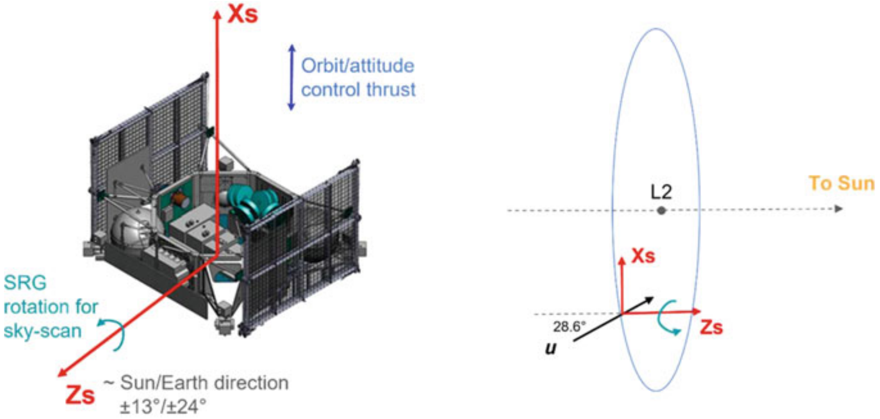

Xs
Orbit/attitude control thrust
SRG rotation for sky-scan
Zs
~ Sun/Earth direction ±13°/±24°
L2
To Sun
Xs
28.6°
u
Zs

Fig. 13 Reaction wheels unloading thrust (left) and escape velocity vector ***u***, forming the angle of 28.6° with respect to the Sun direction (right)

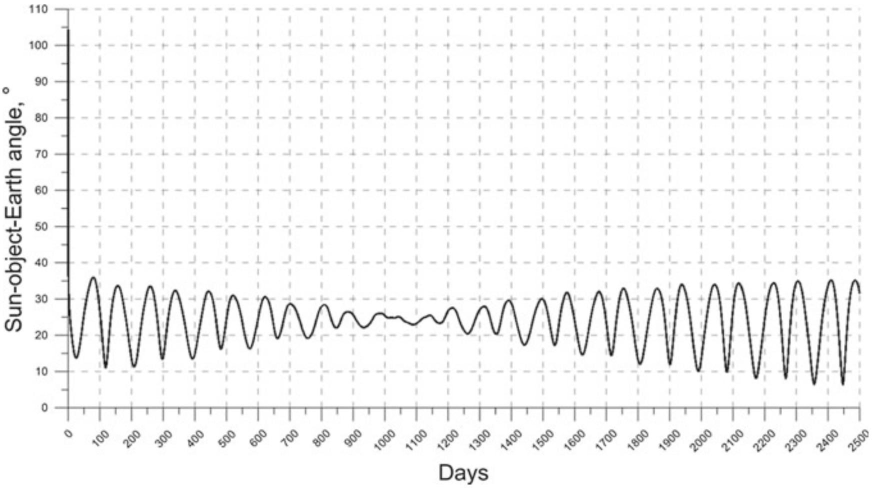


Fig. 14 Sun-spacecraft-Earth angle (days are counted from the launch)

It should be noted however that the spacecraft rotation axis Z_S does not always lie in the ecliptic plane during the attitude control operations. This is due to scientific observation schedule (mostly) and the technical constraints of the spacecraft design (see Sect. 2), namely the admissible angle between the antenna for the radio link (coincided with the Z_S-axis) and the direction to ground stations (within 24°), and also the sunward direction angle (within 13°). It means that the Sun-spacecraft-Earth angle must be less than the sum of the mentioned 13° and 24°. Figure 14 confirms that this constraint is satisfied.

In the situation, when the rotation axis Z_S doesn't lie on the ecliptic plane, the strategy of the control wheels offloading is almost the same, except that instead of the ecliptic plane, the plane formed by the escape velocity vector $\boldsymbol{u}$ and the Z_S rotation axis is considered (Fig. 13).

The attitude control strategy is summarised in the table below.

Condition	Manoeuvre
When Zs-axis lies in the ecliptic plane	
If Xs-axis $\perp$ ecliptic plane ⇨ min. perturbation	Reaction wheel offloading
If Xs-axis lies in ecliptic plane, and Xs-axis projection in $\pm \boldsymbol{u}$ direction	Orbit correction manoeuvre
When Zs-axis is not in the ecliptic plane	
If Xs-axis $\perp$ to Zs $\bigcap \boldsymbol{u}$ plane ⇨ min. perturbation	Reaction wheel offloading
If Xs-axis lies in Zs $\bigcap \boldsymbol{u}$ plane, and Xs-axis projection in $\pm \boldsymbol{u}$ direction	Orbit correction manoeuvre

5.2 Measurements with the Use of Ground Telescopes

In addition to minimization of orbit perturbations by the applied method, another important role in the trajectory control is the involvement in the trajectory measurements of optical tracking by ground-based telescopes. Characteristics of the available for the mission telescopes, e.g. Russian-Turkish 1.5-m Telescope (RTT150) or 1.6-m telescope (AZT33-IK) of Sayan Solar Observatory of ISTP SB RAS, have allowed observing spacecraft as an object of 18–19 magnitude. The sequence of SRG tracking observations done in early phase of the mission, on July 15, 2019, is demonstrated in [12]. Initially, such observations were considered only as auxiliary to the standard radio ranging measurements. But later, processing these observations led to the conclusion that their use allows mission to significantly improve the accuracy of trajectory determination, having the 3σ errors measurements of the order of a tenth of an arcsecond. This result can be explained by the fact that the spacecraft trajectory parameters determination is more sensitive to the errors of measurements of the angular position than to the errors in radial motion measurements, in particular for the orbital control of a spacecraft on an LPO.

6 Mission Extension End-of-Life Disposal Option

The nominal mission program is planned for 6.5 years. However, it is estimated to have a remaining propellant onboard SRG after completing the main scientific program. The remaining propellant would enable to perform a total manoeuvre of 200 m/s [13]. A considered mission extension and a following end-of-life disposal option is a flyby of an asteroid or a comet. According to CNEOS database [14], close approaches to the Earth by known near-Earth objects (NEOs) with distance less than 0.05 au occur with the frequency of more than 50 encounters per year. It allows a large spectre of possible target NEOs for the EOL scenario. As an option presented here, the Apophis asteroid is of particular interest with its close approach to the Earth in 2029. A close approach of the SRG spacecraft with Apophis would make possible to estimate the asteroid's mass with high accuracy.

The Apophis asteroid will approach to the Earth on about 31,000 km [15]. The ephemeris data of Apophis are provided by NASA Horizons interface [16] and used for analysis of SRG flyby options. In this scenario, a one-manoeuvre scenario is simulated, with the following heliocentric graveyard orbit for the SRG mission.

Figure 15 shows the minimum required manoeuvre's cost value according to the date of its application for the transfer to the asteroid. As it can be seen, the ΔV required for the transfer satisfies the estimated 200 m/s budget only for dates between 25.02.2029 and 15.03.2029. The trajectory of the SRG approach to Apophis for the optimal date of the manoeuvre is demonstrated on Fig. 16. In this case the approach takes place before the Apophis perigee, at the distance of about one million km from the Earth, and when the relative velocity of the spacecraft is about 6 km/s.

The proposed option allows to extend the mission, while achieving more favourable conditions for scientific measurements.

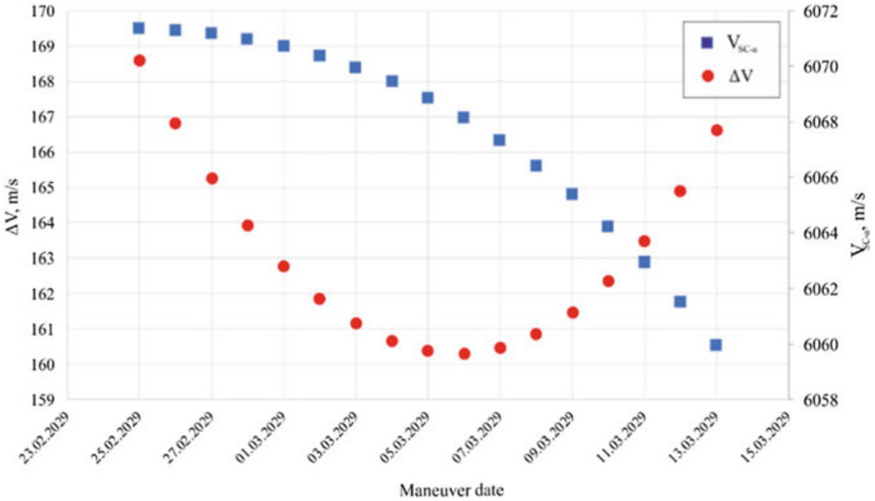


Fig. 15 Minimal ΔV required for the SRG transfer to Apophis on the date of the manoeuvre's application (red dots), and the relative flyby velocity of the spacecraft to the asteroid (blue squares)

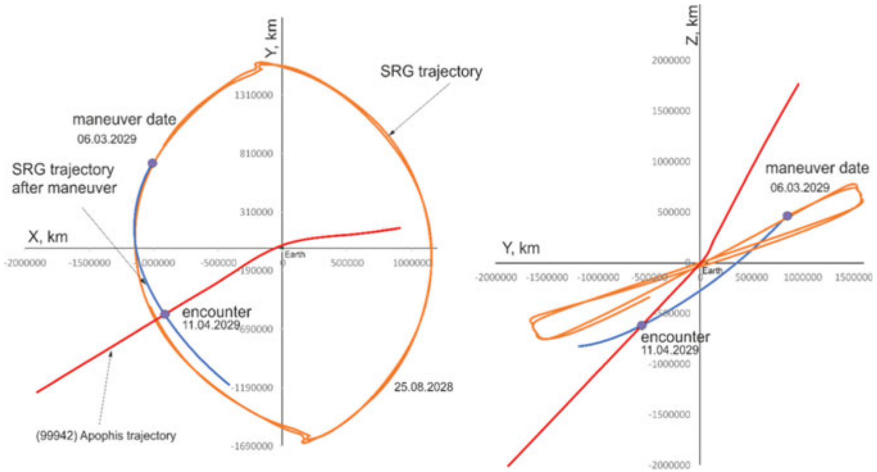


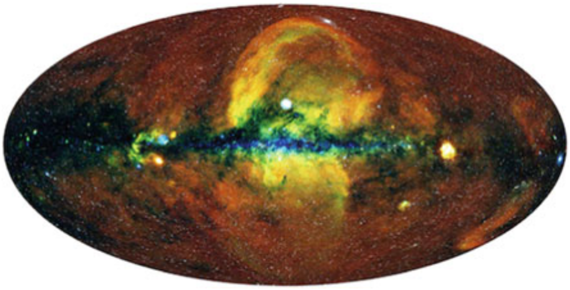
Fig. 16 SRG trajectory about the L2 point (orange line) with the subsequent transfer to Apophis (blue line) for the manoeuvre on 06.03.2029 and the flyby of Apophis on 11.04.2029 in the geocentric inertial equatorial coordinate system

7 Results and Prospective

On July 13, 2019, the SRG space observatory was launched from Baikonur. Since then, the spacecraft has been successfully operating on the LPO, while the onboard telescopes have been providing the unprecedented scientific data about our universe (see Fig. 17) [17].

The operational orbit satisfies all the technical constraints and the scientific goals of the mission. The requirement on the 4-h daily visibility duration from the Russian ground stations is achieved by the on-going set of manoeuvres. The first five manoeuvres were successfully applied on October 05, 2020, November 23, 2020, February 28, 2021, May 23, 2021 and September 6, 2021, without damaging the onboard scientific payload. The next manoeuvres are scheduled until March 02, 2023, resulting in the total delta-v cost of no more 65 m/s.

Fig. 17 The Spectr-RG-eROSITA all-sky map. CREDIT: Jeremy Sanders, Hermann Brunner, Andrea Merloni and the eSASS team (MPE); Eugene Churazov, Marat Gilfanov (on behalf of IKI) [17]



The described in this work strategy of the attitude control was applied to the mission in practice from December 24, 2019, to January 30, 2020. During this time, the momentum wheel desaturations were performed approximately once per week. Each of these sessions lasted about one minute, using less than 10 g of the propellant. The engine thrust during the manoeuvres was always directed close to the normal to the ecliptic plane. Then next, on January 30, 2020, the orbit station-keeping manoeuvre was performed, costing about 0.2 m/s. Hence, the effectiveness of the proposed approach was confirmed, and it was adopted for further flight.

The attitude control strategy described in this work and applied to SRG spacecraft, allows mission to reduce the number of orbit corrections and propellant consumption, and consequently to improve the quality of the sky-survey by the onboard telescopes.

On the next, second phase of the mission, planned for point observations, the telescopes will observe each individual source during about a few days. To reduce the possible orbit perturbations produced by the momentum wheels desaturations, the individual source observations are scheduled between these operations. The optimal planning would be when each subsequent momentum wheels off-loading compensates the orbit perturbations of the previous one, and the projection of the telescope's axis, X_S-axis, during these operations onto the escape direction $\boldsymbol{u}$ is to be zero. Practically, this condition is satisfied when the target object is in a direction orthogonal to the escape vector $\boldsymbol{u}$ (namely in the plane orthogonal to $\boldsymbol{u}$).

In the future, when the main goals of the mission are achieved, the estimated saved propellant on board may be used to extend the mission. In particular, the departure from the SEL2 orbit is vantage to visit a Near-Earth object. A successful example of such a scenario was implemented in ISEE-3 (ICE) mission [18], visited the Giacobini-Zinner comet after its nominal scientific program.

References

1. Spektr-RG launch successful. http://srg.iki.rssi.ru/?page_id=680&lang=en. Accessed 02 Feb 2020
2. Kovalenko ID, Eismont NA (2019) Orbit design for the spectrum-Roentgen-Gamma mission. Acta Astronaut 160:56–61
3. Ilyin IS, Zaslavsky GS, Lavrenov SM, Sazonov VV, Stepanyants VA, Tuchin AG, Tuchin DA, Yaroshevsky VS (2014) Ballistic design of trajectory from a low Earth orbit to a halo-orbit in the vicinity of the L2 point in the Sun-Earth system. Kosmicheskie Issledovaniya 52(6):476–488 (in Russian)
4. Mzhel'skiy PV, Mikhaylov YA (2018) Orbit design of «Spektr-RG» spacecraft under constraints of visibility from Russian ground satations. Preprinty IPM im. M.V. Keldysha 9:18. https://doi.org/10.20948/prepr-2018-9 (in Russian)
5. Farquhar RW (1970) The control and use of libration-point satellites. PhD thesis, Stanford University. Reprinted as NASA Technical Report R-346
6. Canalias Vila E (2007) Contributions to libration orbit mission design using hyperbolic invariant manifolds. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2007. −6 + x + 187 p. https://www.comet-cnes.fr/resource-access/ECanalias_0.pdf
7. Howell KC, Barden BT, Lo MW (1997) Application of dynamical systems theory to trajectory design for a libration point mission. J Astronaut Sci 45(2):161–178

8. Koon WS, Lo MW, Marsden JE, Ross SD (2000) Dynamical systems, the three-body problem and space mission design. In: Equadiff 99, (In 2 volumes), pp 1167–1181
9. Eismont N, Dunham D, Jen S-C, Farquhar RW (1991) Proceedings of the 3rd international symposium on spacecraft flight dynamics, Darmstadt, Germany, Sept. 30–Oct. 4, 1991 (ESA, 1991), p 435
10. Hechler M, Cobos J (2002) Hershel, Planck and Gaia orbit design. In: Paper presented at the libration point orbits and applications, Aiguablava, Spain, 10–14 June 2002
11. Navigator spacecraft platform designed and produced by Lavochkin Association. https://www.laspace.ru/company/products/platforms/navigator/. Accessed 31 Aug 2021
12. https://www.youtube.com/watch?v=0FJsnCXYTh4
13. Eismont N et al (2021) Extension of the earth libration point missions by targeting a spacecraft to near-earth asteroids. In: 7th IAA planetary defence conference
14. NASA, Center for near-earth object studies. https://cneos.jpl.nasa.gov/. Accessed 31 Aug 2021
15. 99942 Apophis (2004 MN4): JPL small-body database browser. https://ssd.jpl.nasa.gov/sbdb.cgi?sstr=99942;orb=1;cov=0;log=0;cad=0#orb (application date 14.04.2021)
16. WebGeocalc: a tool of the navigation and ancillary information facility. https://ssd.jpl.nasa.gov/horizons.cgi. Accessed 14 April 2021
17. Predehl P, Sunyaev R, Becker W, Brunner H, Burenin R, Bykov A, Cherepashchuk A, Chugai N, Churazov E, Doroshenko V, Eismont N, Freyberg M, Gilfanov M, Haberl F, Khabibullin I, Krivonos R, Maitra C, Medvedev P, Merloni A, Wilms J (2020) Detection of large-scale X-ray bubbles in the Milky Way halo
18. Farquhar R, Muhonen D, Church L (1984) Trajectories and orbital maneuvers for the ISEE-3/ICE comet mission. In: Astrodynamics conference, p 1976

MASCOT—A Mobile Lander On-board the Hayabusa2 Spacecraft—Operations on Ryugu

C. Krause, U. Auster, J. P. Bibring, J. Biele, C. Cenac-Morthe, F. Cordero, B. Cozzoni, C. Dudal, D. Embacher, C. Fantinati, H.-H. Fischer, K. H. Glassmeier, D. Granena, M. Grott, J. T. Grundmann, V. Hamm, D. Hercik, T.-M. Ho, R. Jaumann, K. Kayal, J. Knollenberg, O. Küchemann, C. Lange, L. Lorda, M. Maibaum, D. May, Y. Mimasu, A. Moussi, T. Okada, J. Reill, T. Saiki, K. Sasaki, M. Schlotterer, N. Schmitz, N. Toth, Y. Tsuda, S. Ulamec, T. Yoshimitsu, S. Watanabe, F. Wolff, and The MASCOT Team

Abstract MASCOT ('Mobile Asteroid Surface Scout') is a 10 kg mobile surface science package part of JAXA's Hayabusa2 sample return mission. The mission was launched in December 2014 from Tanegashima Space Center, Japan. The Hayabusa2 spacecraft reached the target asteroid in summer 2018. After a mapping phase of the asteroid and a landing site selection process the MASCOT lander was deployed to the surface on the 3rd of October 2018. MASCOT operated successfully for about 17 h

C. Krause (✉) · J. Biele · B. Cozzoni · D. Embacher · C. Fantinati · H.-H. Fischer · K. Kayal · O. Küchemann · M. Maibaum · D. May · S. Ulamec
DLR Microgravity User Support Center (MUSC), Cologne, Germany
e-mail: Christian.Krause@dlr.de

U. Auster · K. H. Glassmeier · D. Hercik
Institut für Geophysik und extraterretrische Physik, Technische Universität Braunschweig, Brunswick, Germany

J. P. Bibring · V. Hamm
Universite de Paris Sud-Orsay, IAS, Orsay, France

C. Cenac-Morthe · C. Dudal · D. Granena · L. Lorda · A. Moussi
CNES Centre National d'Etudes Spatiales, Toulouse, France

F. Cordero
Telespazio VEGA Deutschland GmbH, Darmstadt, Germany

M. Grott · R. Jaumann · J. Knollenberg · N. Schmitz
DLR Institute for Planetary Research, Berlin, Germany

J. T. Grundmann · T.-M. Ho · C. Lange · K. Sasaki · M. Schlotterer · N. Toth
DLR Institute for Space Systems, Bremen, Germany

R. Jaumann
Institute of Geosciences, Freie Universität Berlin, Berlin, Germany

Y. Mimasu · T. Okada · T. Saiki · Y. Tsuda · T. Yoshimitsu · S. Watanabe
ISAS/JAXA, Yoshinodai, Chuo, Sagamihara, Kanagawa, Japan

J. Reill · F. Wolff
DLR Institute of Robotics and Mechatronics, Wessling, Germany

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_25

on the surface of Ryugu. It performed three relocation manoeuvres and one "Mini-Move" and returned 128 MBytes of data. MASCOT has been developed by the German Aerospace Center (DLR) in cooperation with the Centre National d'Etudes Spatiales (CNES). The main objectives were to perform in-situ investigations of the asteroid surface and to support the sampling site selection for the mother spacecraft. These objectives could be reached successfully. On 6th December 2020 Hayabusa2 successfully returned asteroid samples to the Earth.

Keywords Hayabusa2 · MASCOT · Asteroid · Landing · Surface · Ryugu

Acronyms/Abbreviations

CNES	Centre National d'Etudes Spatiales
DLR	Deutsches Zentrum für Luft-und Raumfahrt e.V.
HY2	Hayabusa2
FD	Flight Dynamics
FoV	Field of View
GNC	Guidance, Navigation and Control
IAS	Institut d'Astrophysique Spatiale
JST	Japan Standard Time
MAM	MASCOT Autonomy Manager
MARA	MASCOT Radiometer
MASCam	MASCOT Camera
MASMag	MASCOT Magnetometer
MCC	MASCOT Control Center
MMEGA	MicrOmega
MASCOT	Mobile Asteroid Scout
SSOC	Sagamihara Space Operations Center

1 Introduction

Hayabusa2 is a spacecraft, developed and launched by the Japan Aerospace Exploration Agency (JAXA) dedicated to investigate a primitive asteroid and return samples back to Earth [1]. After launch on December 3rd, 2014 from the Tanegashima Space Center, with an H2A rocket, it went on its cruise to the C-type asteroid (162173) Ryugu (earlier designation was 1999JU3). The spacecraft reached

The MASCOT Team
Cologne, Germany

the target asteroid in summer 2018 and returned the collected samples to Earth on the 6th of December 2020 (JST).

Hayabusa2 is based on the highly successful Hayabusa mission, which was the first spacecraft successfully returning samples from an asteroid to Earth (launched in May 2003 to asteroid (25143) Itokawa). Certain aspects of Hayabusa2 are modified as compared to its predecessor, including the addition of a small lander, MASCOT as science payload.

The MASCOT/Hayabusa2 mission started a new generation of space missions, in a new frame of cooperation, coupling intimately remote sensing at a macroscale, in-situ characterization at a microscopic scale, and return samples for refined laboratory analyses.

MASCOT has been developed by the German Aerospace Center (DLR) in cooperation with the Centre National d'Etudes Spatiales, (CNES) as well as the TU Braunschweig and the Universite de Paris Sud-Orsay. The main objective of MASCOT was to perform in-situ investigations of the asteroid surface, provide ground truth and to support the sampling site selection for the mother spacecraft.

MASCOT is a mobile surface science package with a mass of about 10 kg [2, 3]. After arrival at the target asteroid (162173) Ryugu a detailed mapping phase was performed and the landing site of MASCOT has been selected [4]. The deployment of MASCOT to the asteroids surface took place, as planned, at the beginning of October 2018. After its deployment MASCOT investigated the asteroid surface in detail. Two relocation manoeuvres, self-rightening and a so-called "mini-move" were performed. MASCOT survived two asteroid nights and performed its planned measurements. The scientific results of MASCOT combined with those from the instruments aboard the Hayabusa2 mother spacecraft as well as the returned samples shall allow a comprehensive understanding of asteroid Ryugu and the role of primitive asteroids during the history of the solar system.

2 Mascot Lander System and Payloads

During its mission, MASCOT supported four scientific instruments: a wide-angle camera, a hyperspectral infrared microscope, a radiometer and a magnetometer (see Fig. 1). The camera (MASCam) provided the ground truth for the orbiter remote sensing observations, for measurements by the other lander instruments (radiometer, spectrometer), and the orbiter sampling experiment. It helped characterizing the geological context, mineralogy and physical properties of the surface (e.g. rock and regolith particle size distributions) [5]. The MASCOT camera observations, combined with the MASCOT hyperspectral microscope and radiometer spectral observations, have been designed to cover a wide range of observational scales and to serve as a strong tie point between Hayabusa2's remote sensing science (10^3–10^{-3} m) and sample science (10^{-3}–10^{-6} m) [6]. The radiometer (MARA) determined the surface brightness temperature, the thermal inertia of the surface material and the spectral slope in infrared [7]. The MARA radiometer's FoV has been designed

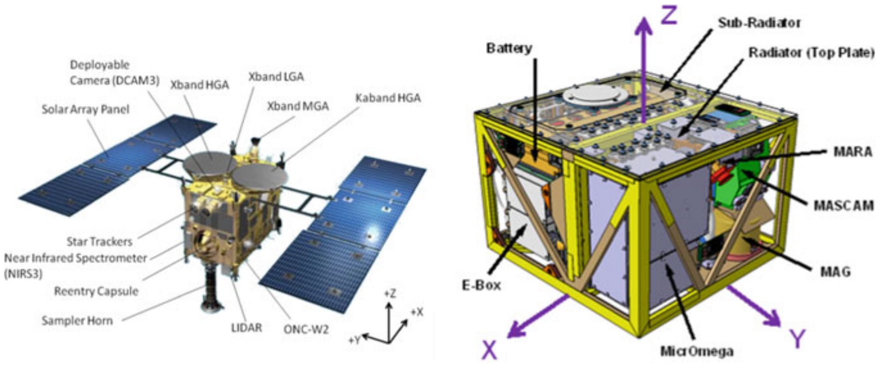


Fig. 1 Left: Overview of Hayabusa2 spacecraft [5], Right: Schematic of MASCOT lander

and calibrated to be within the camera's FoV such that the effects of grain size and boulders in the field of view can be disentangled from the thermal measurements. The magnetometer (MASMag) observed the magnetic field profile during descent and bouncing and for the determination of any global and local magnetization of the asteroid [8]. The hyperspectral IR microscope (MicrOmega) is an instrument for the investigation of the composition of the asteroidal surface at grain scale: regarding minerals (pristine, altered), ice/frosts, and organics as well as the characterization the microscopic structure of the soil [9].

MASCOT was powered by a primary battery which enabled it to investigate the asteroid surface for ideally more than to two asteroid days. MASCOT was further equipped with a mobility mechanism for selfrighting and relocation on the asteroid surface. Due to its shape (compare Fig. 1) and the absence of a dedicated landing gear, it was not predefined which side would face the asteroid once MASCOT came to rest after delivery and bouncing. The GNC sensors would determine the orientation of MASCOT relative to the asteroid surface and the mobility mechanism used to upright MASCOT in its default measurement orientation. The communication of MASCOT to Earth took place via the Hayabusa2 spacecraft as a relay. It is worth noting that because the Hayabusa2 spacecraft was located near the sub-Earth point during the MASCOT mission, contact between MASCOT and the main spacecraft was only possible during the asteroid day times.

Because of the long signal travel times (16 min one way) and the limited lifetime of the MASCOT primary battery, ground loops were not part of the MASCOT nominal operation strategy. Therefore, MASCOT was equipped with the MASCOT Autonomy Manager (MAM), a state machine which scheduled the different MASCOT operations depending on the detected situation, such as:

- Detection of separation from HY2 spacecraft
- Descent of MASCOT
- MASCOT rest on the surface and its orientation towards it
- MASCOT upright, according to its nominal measurement orientation

- Scheduling of science surface activities
- Day/night detection
- Trigger for MASCOT relocation
- Schedule the MASCOT end-of-life activities.

Out of the four instruments aboard MASCOT, MASMag and MARA were permanently switched on and taking measurements. Different settings were adjusted on event by the MAM. The Camera and MicrOmega activities were defined in separate sequences which were scheduled by the MAM on event.

3 Mascot Cruise Phase

Between the Hayabusa2 launch in December 2014 and the approach to asteroid (162173) Ryugu, MASCOT has already performed a large number of operations in space. The main objectives of these operation activities were health status checks of the MASCOT bus and instrument in-flight calibrations. The MASCOT health checks have been performed for an analysis of the functionality of the lander instruments and subsystems. The calibration of the instruments focused on the monitoring of the instrument performance during cruise phase. Additional objectives were to test HY2 and MASCOT interaction as well as preparation tests for the on-asteroid phase. Table 1 shows a chronological overview of MASCOT in-flight activities during the cruise phase.

Table 1 MASCOT in-flight activities during cruise

Activity	Date	Activtiy	Date
MASCOT health check	Jun 2015	MASCOT instrument calibration #2	Nov 2016
HY2-MASCOT communication check	Jun 2015	MASCOT health check	May 2017
MASCOT instrument calibration #1	Sep 2015	MASCOT software upload	July 2017
PRM activation (launch lock)	Sep 2015	MASCOT health check	July 2017
CAM data download	Jan 2016	MASCOT on-asteroid sequence test	Aug 2017
MASCOT health check	Jul 2016	MASCOT instrument calibration #3	Nov 2017
Data transfer test (On-asteroid spacecraft configuration)	Jul 2016	MASCOT on-asteroid MAM test	Nov 2017
MASCOT thermal evaluation	Jul 2016	Data transfer test (On-asteroid spacecraft configuration)	Dec 2017
HY2-MASCOT communication check	Oct 2016		

All telecommands and activities performed in flight and later on the asteroid have been validated beforehand at the MASCOT ground reference model at the MASCOT Control Center (MCC). The telecommands were uplinked to MASCOT from SSOC (Sagamihara Space Operations Center) via the Hayabusa2 orbiter. Hayabusa2 was used as relay for MASCOT telecommands and telemetry throughout the whole mission. Commanding, telemetry processing, evaluation and status assessment of MASCOT was jointly performed in cooperation with all partners JAXA, CNES and DLR in the respective control centers together with IAS and TU Braunschweig. The real-time monitoring of the in-flight activities took place at the MCC while a MASCOT liaison was present at SSOC.

The MASCOT health checks performed in flight showed a nominal status of MASCOT system and instruments. In-flight calibrations of the instruments have been performed to provide the opportunity to monitor instrument performance throughout the cruise phase until the target asteroid was reached.

4 Landing Site Selection

The landing site selection process for MASCOT has been performed in close cooperation with JAXA, as it was not only dependent on input from the instruments of the Hayabusa2 main spacecraft, but also interconnected with the delivery sites of the MINERVA II landers and the selection of the sampling sites [4].

The selection of the MASCOT landing site, a complex process, involving many actors, had to be carried out within about one month, to make use of the data, received after arrival at the asteroid, and being ready for lander delivery in October 2018. In order to be prepared to perform the process in such a short time, it was exercised two times, about a year before the actual events took place.

A major role played the CNES flight dynamics team, analyzing the separation-descent and landing process, including various bouncing scenarios by Monte Carlo analyses [4, 10].

For the landing site of MASCOT several constraints had to be taken into account:

(a) Operational constraints: guaranteeing daylight for 50–70% of the asteroid rotation period, allowing communications link with the main spacecraft and considering thermal restrictions.

(b) Constraints imposed by the Hayabusa2 spacecraft: the main S/C had to stay near the subsolar point and had time constraints related to the use of various Grounds stations on Earth. No overlap between sampling sites and MINERVA II or MASCOT landing sites was acceptable (to avoid the risk of confusion between lander and target marker).

(c) Scientific constraints: due to a relatively homogeneous surface of Ryugu, there were no areas explicitly favored. Nevertheless, scientific relevance of the candidate sites was discussed (Fig. 2).

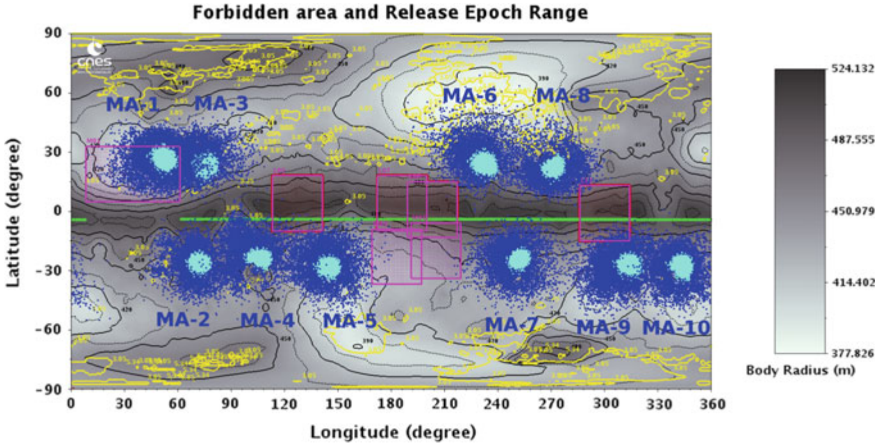


Fig. 2 The possible landing sites of MASCOT determined by the flight dynamics team [4]. The light blue areas are possible first contact points of MASCOT with the surface of Ryugu (CP1). The dark blue areas are the possible settlement points for Mascot after bouncing (SP1). The pink squares were potential activity areas for HY2 spacecraft [credit JAXA]

After analyzing 10 candidate sites, in an iterative process during a team meeting in Toulouse, on August 14th, 2018, one site (MA-9) was selected as most favorable from MASCOT side (Fig. 2). This site has been confirmed by JAXA during the HJST (Hayabusa2 Joint Science Team meeting), August 17th in Sagamihara and was finally adopted for the successful delivery on October 3rd.

5 Mascot Deployment Strategy

For MASCOT deployment HY2 spacecraft slowly descended from its home position at 20 km above the asteroid's surface and released MASCOT at about 41 m (Fig. 3) altitude. The lander performed a free-fall down towards the asteroid surface, touchdown and bounced over the Ryugu surface till it came to rest. After the MASCOT release the HY2 spacecraft ascended to an altitude of 3 km and hovered at this altitude during MASCOT operations. During the descent phase and especially after the separation of MASCOT the HY2 spacecraft took pictures with its on-board cameras to track the pass of MASCOT and to support the localisation of MASCOT on Ryugu's surface [11].

MASCOT landing trajectory prediction and optimization had to take into account the fact that the lander had no anchoring mechanism and was, thus, expected to bounce on the asteroid surface and possibly stop far from its first touchdown point. This is indicated in Fig. 4, where CP1 denotes the first contact point and SP1 the first settling point. MP1 denotes the first measurement position. The simulated trajectories resulting of the Flight Dynamics dispersions analysis are essential inputs to tune at

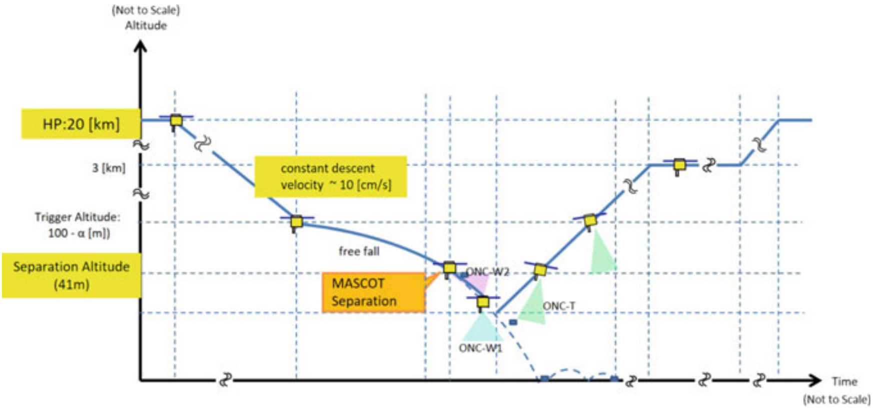


Fig. 3 Schematic image of MASCOT deployment [5] (credit JAXA)

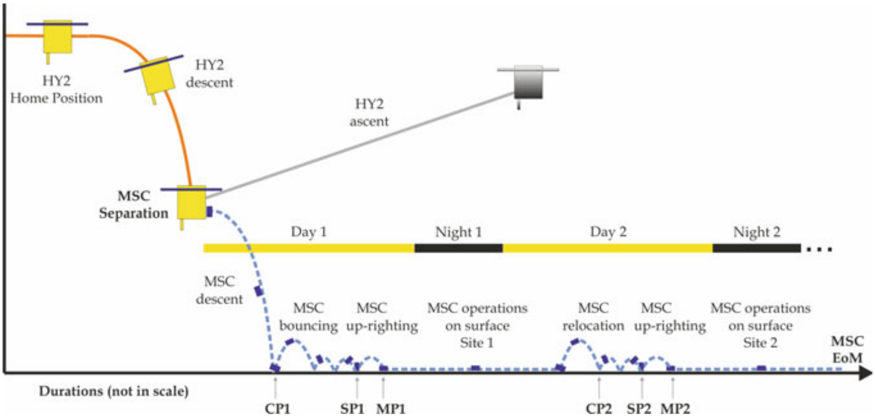


Fig. 4 Sketch of MASCOT and Hayabusa-2 relative movements after separation (credit DLR)

best the exact time of release and the exact position of release, with the intention of maximizing the chances to have MASCOT resting in a suitable place for valuable scientific experiments without endangering Hayabusa2 sampling operations [4].

The deployment concept of MASCOT with its bouncing profile did not allow precision landing on a predefined landing spot but did result in a spread-out landing area instead [10]. This implied that the final settling longitude and latitude could only be predicted with a certain probability. The exact asteroid daytime to start the MASCOT surface operation was not known beforehand and also the exact duration of day and night at the actual resting point had some uncertainty. The planning of MASCOT on-surface operations thus had to rely on minimum and maximum assumptions for the different MASCOT phases like bouncing or surface illumination cycles.

6 MASCOT Landing Preparation and Separation

The general on-asteroid strategy of MASCOT was planned long in advance during the cruise phase. The science sequences were planned, prepared and intensively tested on ground. All sequences needed to be prepared beforehand, as ground interactions with MASCOT were limited. The planning needed to be focused on robustness with the aim to maximize the science return during the limited lifetime of MASCOT. The measurements needed to be timed for an optimal usage of the link budget from MASCOT to the Hayabusa2 spacecraft, so that the data could be stored on-board the HY2 spacecraft before the battery of MASCOT was depleted. As described above the settlement spot of MASCOT could not be predicted precisely, which lead into an uncertainty for the duration from settlement until first sunset, but also in the day/night time duration. The number of attempts and thus the duration to achieve self-righting could not be predicted precisely beforehand, either, introducing an additional timing uncertainty relative to the day/night transition as well as loss and reacquisition of signal of the link to HY2.

The landing site was selected about six weeks before the separation of MASCOT took place. The time between the confirmation of the landing site and the real separation of MASCOT was used to finalise the parameter settings of the payloads and subsystems for the conditions of the targeted landing site. Final tests of the sequences were performed by the operations team at the ground reference model and the final command products were prepared.

The MASCOT separation took place on the 3rd October 2018. The overall activities in space started already days beforehand with preparation activities.

7 MASCOT Landing and On-Asteroid (Ryugu) Phase

The In-flight preparation of the MASCOT Ryugu phase started about 1 week before the separation took place with the upload of the final command sequences for the separation and the on-asteroid phase. These command sequences contained already the final adaptations for the selected landing area.

The final MASCOT switch-on took place on the 29th September 2018 while HY2-Spacecraft was still hovering at its home position 20 km above Ryugu's surface. The activities for MASCOT separation started with the final step for the primary battery depassivation followed by a MASCOT system and payload check. Afterwards MASCOT was not switched off but stayed in idle mode coming along with small activities and checks still powered via the main spacecraft while the HY2 spacecraft started its descent to the targeted release altitude of MASCOT about 40–60 m above the asteroid surface.

Prior to the separation the battery was heated to its targeted release temperatures. The MARA instrument was switched-on about 3 h before separation to adjust the temperatures to its set points. The magnetometer was also switched-on this time to monitor the separation itself and the descent of Mascot to Ryugu's surface.

The MASCOT battery switches were closed shortly before separation. From the moment of separation until the end-of-life MASCOT was powered by its own battery.

MASCOT separation took place on 3rd October 2018 01:57:20 UTC in an altitude of about 41 m above Ryugu's surface. It required about 4.0 ± 1.5 s for MASCOT to leave its support structure (MESS) inside HY2 spacecraft.

30 s after MASCOT separation the HY2 spacecraft started to ascent to an altitude of about 3 km above Ryugu's surface. It continued to stay at this altitude of 3 km until the end of the MASCOT mission and returned afterwards to its home position [3].

During MASCOT descent MASCam, MASMag and MARA performed measurements. The cameras aboard the HY2 spacecraft took pictures in parallel in order to track the pass of MASCOT.

The first contact of MASCOT with the surface of Ryugu occurred 05:51 [mm:ss] after separation. MASCOT bounced over Ryugu's surface for further till it came to rest. The analysis by Jaumann et al. [11] and Scholten et al. [12] showed that the first contact of MASCOT with Ryugu was the shadowed side of a bolder. With combination of the MASCOT images and the HY2 images it was possible to reconstruct the bouncing pass of MASCOT on Ryugu (Fig. 5). After its settlement MASCOT performed a selfrightening manoeuvre to get into its measurement orientation towards the asteroid's surface and the MAM started the first MASCOT

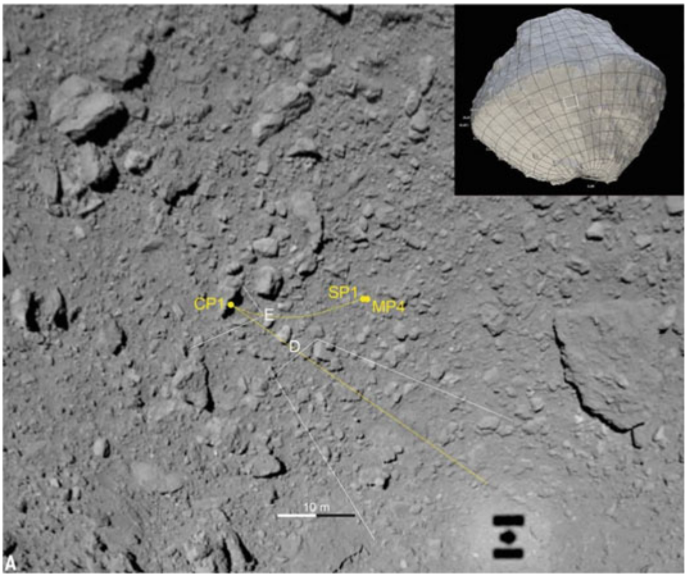


Fig. 5 Reconstruction of MASCOT trajectory towards and on Ryugu, Yellow: MASCOT descent and bouncing trajectory [11]. Image of Hayabusa2 ONC camera (credit JAXA/UTokoyo/Kochi U./Rikkyo U./Nagoya U./Chiba Ins. Tech/Meiji U./U.)

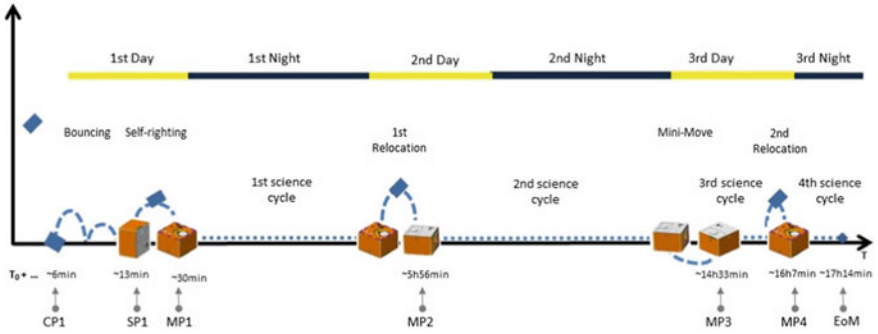


Fig. 6 The schematic MASCOT mission timeline starting from release at T_0 until its end of mission [11]

surface science sequence (Fig. 6). Actually, the GNC system on-board MASCOT detected the orientation wrong for this first measurement position, likely due to the very rough terrain around MASCOT [3]. MASCOT was in this phase upside down oriented but performed constantly its measurement. MASCOT entered in this orientation the first asteroid night in which no communication between spacecraft and lander was possible. With the help of the system and payload data the upside-down orientation could be determined on ground by the MASCOT team. While MASCOT was still on the night side of Ryugu ground commands were sent to the mother spacecraft to force a MASCOT relocation followed by the start the second science cycle to not waste time on the asteroid and to avoid overheating of the MicrOmega bottom side. MicrOmega had just cooled down during the asteroid night while facing to the sky. When MASCOT appeared back in the line of sight to HY2 spacecraft and the commanding link was re-established the respective commands were sent to MASCOT from HY2 spacecraft and the relocation manoeuvre was performed. MASCOT performed a hop of about 70 cm and came to rest in its nominal measurement orientation. The second science measurement cycle was started by the MAM. This second measurement cycle continued during the second asteroid night of MASCOT. All planned measurements could be conducted including also the night MARA and MASCam measurements.

The next intended move of MASCOT was performed after the second science cycle had finished, a so-called MASCOT Mini-Move, 3.6 cm around its z-axis. This slight orientation change of MASCOT was planned to get a slightly different FoV on the same area to allow stereo imaging of the measurement site and attempting also to get different target material into the FoV of MicrOmega.

MASCOT operated meanwhile on the 3rd asteroid day on Ryugu. Before MASCOT entered the 3rd night on Ryugu a last relocation manoeuvre was commanded from ground, attempting to get to another site of the surface. The manoeuvre ended again in an upside-down orientation, and MASCOT entered in this orientation the third night on the asteroid. During this third night the battery of MASCOT depleted. No further signals from MASCOT were received on the next morning. However, the lifetime of MASCOT was near the absolute maximum according to the predictions with a runtime of more than 17 h (Table 2).

Table 2 Main events of MASCOT separation, descent and operation on Ryugu's surface, see also [3]

Time [UTC]	Event	MASCOT operation
		MASMag and MARA are on before separation
01:57:19	NEA firing by HY2 spacecraft	MASMag detects separation
01:57:20	Hayabusa detects disconnection of Mascot	MASCOT moves out of the MESS
01:57:23	MASCOT descent start towards the asteroid	Hayabusa2 released MASCOT with a velocity of ~5.9 cm s^{-1} [11]
01:59:47		MASCam started to acquire descent and bouncing images
02:03:14	The first impact of MASCOT with Ryugu (**CP1**) detected by MASMag followed by a bouncing phase	
02:12:27		MASCam acquired last descent and bouncing images
02:18:51	MASCOT reached its first settlement point (**SP1**)	
02:20:31		MASCOT started to upright, GNC recognized in motion
02:34:20	MASCOT reached its first measurement point (**MP1**)	Start of the 1st science cycle by MAM
03:13:12[a]	End of 1st day and start of 1st night on Ryugu	
07:18:28		A ground command has been sent to HY2 spacecraft to interrupt its science measurement and force the lander to relocate
07:50:04		MASCOT start for 1st relocation (MAM state). Relocation forced by ground command
07:51:38[a]	End of 1st night and start of 2nd day on Ryugu	
08:27:51	MASCOT reached its second measurement point (**MP2**)	The GNC sensors confirmed the correct orientation of the lander and the 2nd science cycle was started

(continued)

Table 2 (continued)

Time [UTC]	Event	MASCOT operation
10:52:06[b]	End of 2nd day and start of 2nd night on Ryugu	
11:16:53		MASMag has been turned off
15:27:11[a]	End of 2nd night and start of 3rd day on Ryugu	
16:28:29	MASCOT at its third measurement point (**MP3**)	MASCOT started "Mini-Move" maneuver and initiated its 3rd science cycle once on MP3
18:05:41	MASCOT reached its 4th measurement point (**MP4**)	MASCOT performed a 2nd relocation once at MP4 executed measurement of MASCam, MicrOmega and shortly MASMag
19:03.57	End of mission (EOM): reception of last MSC HK packets (final link break by horizon occultation)	

[a] Uncertainty of several 10 s because of sensor reading interval
[b] OBC event packet

8 MCC—Ground Segment

Operations were performed at the MASCOT Control Center (MCC) located at DLR, Cologne and with a liaison operator at the Hayabusa2 Control Center (SSOC) in Sagamihara. To include the scientists and engineering community as well as the public at the IAC in Bremen during the landing and operational phase several distributions of telemetry and data products have been established. The main operations were performed by MASCOT operators and subsystem authority representatives in the control room. A liaison operator was located in Sagamihara to be a direct person of contact in any case of connection loss to the MASCOT Control Center. An expertise center was also equipped in CNES premises to support analyses for the landing site selection, with processing capacities dedicated to trajectories commutations and to 3D visualization (Fig. 7).

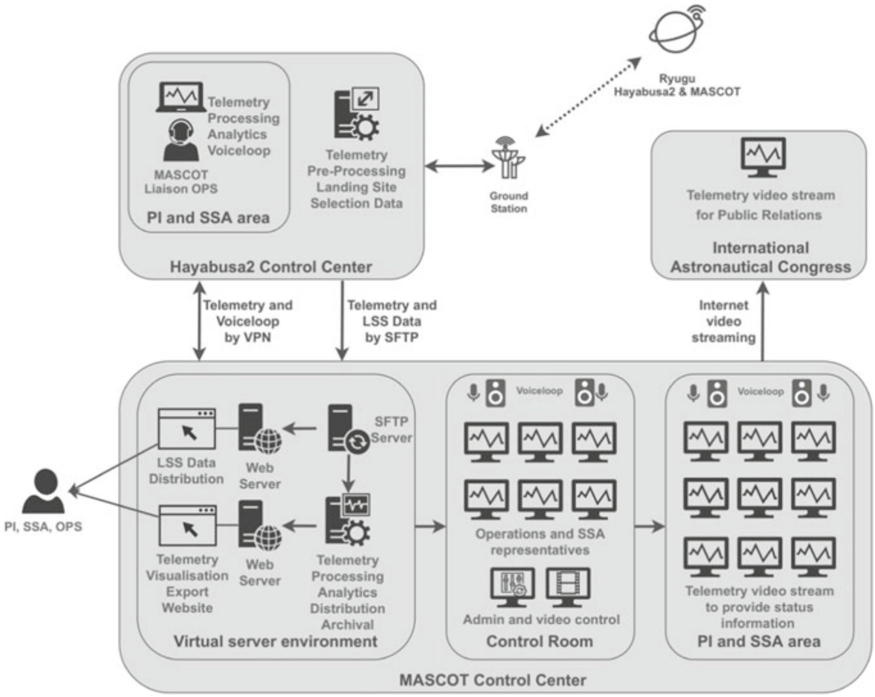


Fig. 7 Schematic on MCC setup, interfaces and data flows during MASCOT operations on Ryugu (credit DLR)

8.1 Telemetry and Landing Site Selection Data

Reception

The telemetry and Landing Site Selection data were received by a server at ISAS that periodically extracted MASCOT relevant and Hayabusa2 related Telemetry from the complete set of Hayabusa2 data sent to Earth. This process ran automatically and produced several growing files that where incrementally received by using the Secure File Transfer Protocol (SFTP) based on Secure Shell (SSH). MASCOT telemetry was provided as a plain CCSDS file and Hayabusa2 telemetry consisted of text files with comma separated values (CSV). The reception rate of MASCOT and Hayabusa2 Telemetry could be freely adjusted up to a rate of one retrieval per minute.

The Landing Site Selection Data was a data set of several types of information. It consisted of data products from all instruments of Hayabusa2. The products were on a scientific level and therefore of standard file types like images, tables, text, video and other. The way of transport was the same as for the telemetry, but updates where made on weekly basis or on request. Because of the large number of files (39,280), folders (577) and the accumulated size of 140 GB a synchronization on file and folder basis was used to only transmit new and changed files and folders during a weekly update. Given the time constraints, the expertise center located in CNES had a direct access to the ISAS server to directly retrieve the data necessary for the time-consuming trajectories optimization and to directly deliver to the Hayabusa2 team the operational data expected, such as the optimized position for the MASCOT release.

Distribution

MASCOT and selected, MASCOT relevant Hayabusa2 telemetry were processed by the MASCOT telemetry servers and could be accessed and used by the clients in in the control room. The plain telemetry was also accessible by the liaison operator in Sagamihara. The operator was equipped with a mobile telemetry server and client to be able to perform the same operations as in the control room. The MASCOT telemetry servers were also capable to provide telemetry data in visualized and exportable formats on a web site. The MASCOT community was able to receive their needed data in pre-configured exports as soon as they were received and processed. Thus, the community was able to follow and analyze their instrument or system and contribute operations with current information. Additionally, to the client and web-based access to the processed telemetry two video streams of selected information were produced. One stream was available in the main scientists and engineering area of the MASCOT Control Center and the second one was published on an internet service to be shown at the International Astronautical Congress (IAC) that was held in Bremen during the landing of MASCOT.

Landing Site Selection Data was synchronized to a Data and Document Management Server of the MCC and available via a web site to all involved parties of the MASCOT project.

8.2 Co-location of the Liaison Operator and Communication

The liaison operator at the Hayabusa2 Control Center in Sagamihara was able to interface personally to the Hayabusa2 operators and had a vital role in the communication of the two control centers. To provide the liaison, the control room and the scientists and engineering area with direct information a voice-loop was used. All involved parties were able to speak to each other no matter of the location of individual. Telemetry and the voice-loop was transported to the liaison by using a Virtual Private Network (VPN) tunnel between the MCC infrastructure and the operator's equipment in the Hayabusa2 Control Center.

9 Summary

The MASCOT mission on Ryugu lasted more than 17 h. This duration exceeded the expectations. MASCOT was successfully separated from the mother spacecraft.

The descent to Ryugu's surface went as expected and could be reconstructed with the data and pictures of MASCOT itself and of HY2 spacecraft. The payloads could perform the main measurement requests or even exceed these.

The Mascot mission was prepared beforehand in great detail considering energy and link budgets. The unknown terrain and uncertainties of final settlement location of MASCOT needed to be considered for the planning scenario and sequencing. The planned activities of MASCOT were robust against the uncertainties faced by the unknown world of Ryugu.

Because of the rough terrain the MASCOT subsystems received wrong determinations of the lander's orientation towards the surface at the beginning of the mission but these could be corrected from ground. The autonomy concept of MASCOT worked out in general with small interactions from ground.

The team experienced an exciting phase with MASCOT being part of the Hayabusa2 project starting with the design; launch and cruise phase and culminating in the successful operation on Ryugu [3].

Acknowledgements The authors would like to thank the complete MASCOT team, including the teams responsible for the subsystems and instruments and the DLR and CNES agencies for supporting MASCOT. Particular thanks go to JAXA for supporting the Lander whenever possible.

References

1. Watanabe S, Tsuda Y, Yoshikawa M et al (2017) Hayabusa2 Mission Overview. Space Sci Rev 208:3–16
2. Ho T-M et al (2017) MASCOT—the Mobile Asteroid Surface Scout Onboard the Hayabusa2 Mission. Space Sci Rev 339–374
3. Ho T-M et al (2021) The MASCOT lander aboard Hayabusa2: the in-situ exploration of NEA (162173) Ryugu. Planet Space Sci 200:105200
4. Lorda L, Canalias E, Martin T, Garmier R, Moussi A, Biele J, Jaumann R, Bibring JP, Grott M, Auster U, Ho T-M, Krause C, Maibaum M, Cozzoni B, Ulamec S, Wolff F, Tsuda Y, Okada T, Mimasu Y (2020) The process for the selection of MASCOT landing site on Ryugu: design, execution and results. Planet Space Sci 194:105086
5. Yamaguchi T, Saiki T, Tanaka S et al (2017) Hayabusa2-Ryugu Proximity Operation Planning and Landing Site Selection. IAC-17-A3.4A.7
6. Jaumann R, Schmitz N, Koncz A et al (2017) The Camera of the MASCOT Asteroid Lander on Board Hayabusa 2. Space Sci Rev 375–400
7. Grott M, Knollenberg J, Borgs B et al (2017) The MASCOT radiometer MARA for the Hayabusa 2 mission. Space Sci Rev 413–431
8. Herčík D, Auster HU, Blum J et al (2017) The MASCOT magnetometer. Space Sci Rev 433–449
9. Bibring JP, Hamm V, Langevin Y et al (2017) The MicrOmega Investigation Onboard Hayabusa2. Space Sci Rev 401–412
10. Lorda L, Canalias E, Garmier R, Torres A, Martin T, Biele J, Moussi A, Ho T-M (2019) Flight Dynamics Analyses to reconstruct MASCOT's trajectory on Ryugu's surface. In: 18th Australian aerospace congress, 24–28 February, Melbourne

11. Jaumann R et al (2019) Images from the surface of asteroid Ryugu show rocks similar to carbonaceous chondrite meteorites. Science 365:817–820
12. Scholten F et al (2019) The descent and bouncing path of the Hayabusa2 lander MASCOT at asteroid (162173) Ryugu, 632. Astron Astrophys 13

Automating International Space Station Robotics Operations Planning: Successes and Challenges

Laura M. Lucier, Kenton C. Kirkpatrick, and Alejandro Ramirez-Serrano

Abstract The National Aeronautics and Space Administration's (NASA's) Robotics Operations branch at the Johnson Space Center is entrusted with the planning and execution of operations on-board the International Space Station (ISS) using the Canadarm2, Dextre, and Mobile Transporter robots. While the most exciting part of the team's work (the aspect most often shown on television or portrayed in movies) is the actual execution of the mission, a significantly greater portion of a robotics flight controller's time is spent performing pre-operations planning. Myriad aspects of each robotic task must be considered in detail prior to execution, including overall task choreography, robot configurations and trajectories, writing of operational procedures, incorporation of protections against operational risks, adherence to flight rules, and advance development of contingency response plans. Recognizing the time demands associated with executing pre-operations planning and struggling to meet increasing call for extra-vehicular robotics operations on-board the ISS, NASA's Robotics Operations branch sought to automate their pre-operations planning process with the goal of decreasing process execution time while preserving (or ideally, improving) the process' level of safety and mission success. To achieve this objective the team employed an agile software development model and prioritized software features that would reduce the process' inherent human risk by allowing the software to autonomously execute those tasks most often performed incorrectly by human operators. This chapter provides an overview of the Robotics Operations branch's pre-operations planning process followed by a description of the automation project and associated software mechanisms and algorithms. Also discussed are

L. M. Lucier (✉) · K. C. Kirkpatrick
Robotics Operations Branch, NASA Johnson Space Center, 2101 E NASA Pkwy, Houston, TX 77058, USA
e-mail: laura.m.lucier@nasa.gov

K. C. Kirkpatrick
e-mail: kenton.c.kirkpatrick@nasa.gov

A. Ramirez-Serrano
Department of Mechanical and Manufacturing Engineering, Schulich School of Engineering, University of Calgary, 2500 University Drive NW, Calgary, AB T2N 1N4, Canada
e-mail: aramirez@ucalgary.ca

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_26

several challenges that were addressed during the software development and adoption phases and the results of the effort.

Keywords Robotics · Mission planning · Agile · Automation · Autonomy

Nomenclature

Φ	Potentials vector
Φ_g	Goal values of potentials
Φ_d	Potential distance to goal
m	Number of potentials
Θ	Joint angles vector
n	Number of joints
j	Joint subscript
J	Jacobian matrix
J^+	Jacobian pseudo-inverse
G	Gain
Fx	Software feature x
Qx	Human error mode x

Acronyms/Abbreviations

ARMD	Automated Robotics Mission Designer
GUI	Graphical User Interface
HRA	Human Reliability Assessment
ISS	International Space Station
NASA	National Aeronautics and Space Administration
PV	Procedure Verification
RPS	Robotics Planning System
SPDM	Special Purpose Dexterous Manipulator (Dextre)
SSRMS	Space Station Remote Manipulator System (Canadarm2)
XML	Extensible Markup Language

1 Introduction

For centuries, automation has been used to render tasks traditionally performed by humans safer, more repeatable, and more efficient. While the tasks humans perform have evolved with the advent of computer control, so too has automation, with today’s smart software tools and human operators cooperating on increasingly complex tasks such as the planning of robotics operations performed in support of human space exploration. This chapter describes the NASA Robotics Operations branch's experiences and lessons learned during a multi-year effort to automate a portion of their pre-operations planning or "mission design" process. An overview of the human-in-the-loop mission design process is given, along with rationale for the need to automate some or all of the process. Section 2 describes the agile approach taken to develop the automation software, with attention paid to how decisions were made regarding which software content to prioritize for inclusion. A description of the various functional algorithms employed to replicate human operator tasks is also given. Results are presented and discussed, followed by a brief conclusion.

1.1 The Robotics Mission Design Process

"Mission design" is the term used to describe the operations planning process that precedes on-orbit execution of ISS robotics activities. Performed by highly trained members of NASA's Robotics Operations branch called "mission designers," the four major phases of the mission design process are shown in Fig. 1.

1. The mission designer determines the trajectories that the robot(s) will follow.
    a. Which robot(s) will be used is dictated by the primary goal of the operation (e.g. large or small payload manipulation, spacewalk support, visual inspection of a spaceship, etc.).
    b. The robots' starting and ending configurations are dictated by preceding and follow-on operations, respectively.
    c. The mission designer must determine appropriate task configurations for the robot(s), taking into consideration (for example) the locations and fields-of-view of the cameras that will be used to monitor the motion and positioning of the robot(s), the starting and goal locations of the payload(s) that will be manipulated, the loads that will be imparted on the robot, the payload and the surrounding ISS structure when the payload

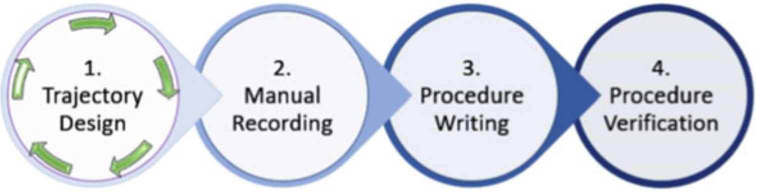


Fig. 1 The ISS robotics mission design process

is extracted/installed, and, for spacewalk support, the reach of the astronaut, the astronaut's line of sight, and the astronaut's access to handrails and other stabilization aids.

d. Once the starting, ending, and task configurations have been determined, the mission designer must determine the safest and most efficient trajectories along which to maneuver the robot(s). All trajectories must obey safety rules such as maintaining a minimum safe distance from the space station. The robot may be moved to its goal position(s) using several possible motion modes. Each mode will result in the robot taking a different path to the goal. The mission designer must choose which mode to use for each maneuver, keeping in mind that some modes are only available when the robot is being operated by an on-orbit astronaut as opposed to being operated by a flight controller from the ground. When operating using frame-of-reference and manual modes, either the shoulder roll or shoulder yaw joint must be "locked" (held stationary) in order to ensure motion of the seven-joint robot arm occurs along a predictable and intended path. Due to the complexity of the ISS work environment, several intermediate reconfigurations are typically needed to maneuver the robot(s) between task configurations. The pre-automation method of planning these intermediate maneuvers was manual iteration/trial and error.

The above work is performed using a computer program that graphically models the ISS's external structure and simulates the kinematics and motion modes of the robots. Called "Robotics Planning System" or RPS, this NASA-proprietary software application runs on a Linux operating system on dedicated computers integrated into NASA's Mission Control Center computer network.

2. The mission designer manually saves their work by entering the intermediate and task configurations for each robot into a spreadsheet using Microsoft Excel on the mission designer's Windows-based laptop. Also recorded are data such as which motion mode has been selected for each maneuver, which base joint locking algorithm will be used, and which display, command, and payload parameter (or motion control) files are to be selected for each maneuver. While tedious and time consuming, this phase of the mission design process is required to record the mission designer's work or save progress between mission design sessions, to pass information between mission designers, and to transfer information from the above-mentioned Linux-based RPS application to the Windows-based procedure authoring application used in the following step of the process.
3. The mission designer develops a procedure that contains step-by-step instructions for executing the operation.

   a. Using the recorded information from Step 2 as a reference, the mission designer manually types the robot configurations and bridging trajectories into a text file input script that was used with a NASA-proprietary software application that ran on the mission designer's laptop.

   b. The mission designer includes instructions for changing motion modes, grasping/releasing payloads, positioning camera focal points to view the motion, etc. in the input script.
   c. The software outputs an MSWord-formatted document containing step-by-step instructions for executing the robotic operation. Due to software limitations, the mission designer must manually edit these steps and add various additional information to the procedure. Additionally output is a text file used to queue commands corresponding to each step of the operational procedure through the Mission Control Center's command and control interface to the ISS.

4. Three robotics flight controllers perform independent reviews of the mission designer's work. This step is required due to NASA's low tolerance for error in safety–critical processes and due to the numerous potential human error modes inherent to the robotics mission design process. This process step is called Procedure Verification (PV), and makes use of a PV Checklist; a document that lists the various functional and safety-related requirements that the procedure and robot configurations and trajectories must meet.

1.2 Process Challenges

Though consisting of just four major steps as described in Sect. 1.1, the mission design process is complex and nuanced. Relying heavily on highly trained humans-in-the-loop for process efficiency and success, mission designers train for approximately one year before their work reaches a skill level that results in consistently safe and useful end products. Up to two years of experience may be required before a mission designer's work consistently results in efficient robot trajectories and configurations that are free of significant errors and do not require re-planning upon review by more experienced personnel. Mission designers must not only be familiar with the robots, their kinematics, and the ISS's external physical environment, but also the numerous operational constraints that must be met and standard operating practises that must be followed to guarantee the safety of the manipulators, payloads, ISS and humans on-board.

Training therefore includes familiarization with these constraints, lessons in robot kinematics, reviews of past operations and lessons learned in their design and execution, and literally hundreds of hours spent practising the development of trajectories for a range of typical robotic tasks. The training program employs a variety of instructional pedagogies and makes use of various information-delivery mechanisms including formal instructor-led hands-on lessons, completion of computer-based tutorials, self-study, peer learning and collaboration, mentorship, and creation and grading of mock procedures. During this time mission designers also train to become and work as flight controllers (i.e. to support operations execution) or for other assignments such as instructing astronauts in the use of the Canadarm2. It is reasonable to state that approximately 40–60% of the trainee's standard work week is spent developing mission design skills during this one- to two-year period, equating to as much as 2400 h of training before being deemed wholly competent in the craft.

Regardless of mission designer skill level, manual (i.e. non-automation software-supported) ISS robotic trajectory planning as performed by NASA's Robotics Operations branch and described in Sect. 1.1 is a highly iterative process. Pre-automation planning tools permit the mission designer to maneuver the robot(s) in various operational modes, moving one to seven joints at a time. Singularity, self-collision, and environmental collision tracking is performed by the planning tools, but not in a predictive manner. The mission designer must therefore first intuit and then try a solution to test whether or not it will work. With years of experience, increased skill reduces iterations and design time, but this trial-and-error method of finding solutions is grossly inefficient, and must be performed many times for each robotic operation because seldom is the same operation completed more than once. Even for seemingly repetitive tasks, such as relocating a robot from one base location to another, the starting and ending configurations for each operation highly vary due to dependence on the preceding and following tasks and susceptibility to evolving ISS external configuration. Although other portions of the end-to-end mission design process are also time consuming (information gathering, developing the text script used to produce the procedure, procedure verification by multiple personnel, etc.), the trajectory design portion of the process represents on average 30–50% of the total hours required to plan the task [1].

As dictated by ISS safety controls, constraints regarding motion rates, maximum motor driving currents and a requirement for live television views of the motion when operating within five feet of ISS structure drive mission designers to avoid such motion when possible. Trajectories must also avoid regions known as "stay-out zones" and "notification zones," which, if entered, may interfere with the operation of ISS components and payloads (for example, the rotation of shade-tracking radiators or line-of-sight of Earth-monitoring science equipment). Mission designers must also consider such things as payload thermal sensitivities, as some payloads are not powered during portions of the operation, driving emphasis on trajectory efficiency or avoidance of solar shadowing. Payloads may have other sensitivities such as requiring that sensors not be exposed to direct sunlight during maneuvering. Some of these considerations are hard constraints (e.g. avoidance of surrounding structure as well as robot arm singularities and self-collisions), and others are softer constraints where trades must be made (e.g. trade between loss of science due to passing through a science payload's field-of-view versus gain in path efficiency). Still others are related to execution efficiency, such as minimising the number of times the robot is commanded to switch between the joint selected for base joint locking, switch between operational modes, or maneuver close to structure where motor rate and current limits and communication/downlink video requirements apply. Combine this with the vast number of possible solutions (noting that when Dextre is operating on the end of Canadarm2, the result is a 15 degree-of-freedom robotic system that can operate from any of several base locations), and it is easy to appreciate that trajectory and configuration planning for ISS extra-vehicular robotic operations could be considered an art as much as a science.

1.3 The Need for Efficiency and Rationale for Automation

Although ISS assembly is complete (a task which relied heavily on Canadarm2), demand for ISS robotics operations is on the rise. With focus shifted from construction to scientific research, external science equipment arrives and departs the ISS with increasing frequency and depends on the Canadarm2 and Dextre robots for installation, removal and, more and more often, manipulation (e.g. the Roll-Out Solar Array [2] and Robotic Refueling Mission [3, 4] experiments). As the space station ages, there is also escalating need for robotic maintenance and repair of the orbiting outpost. Combined with a desire to realize crew time savings and risk reduction by performing tasks robotically rather than by spacewalking astronauts, these use cases have resulted in a marked increase in the frequency of extra-vehicular ISS robotics operations over the past decade:

- In 2009, ground-based flight controllers performed 100 days of telerobotic operations, sending 16,442 commands to the ISS.
- In 2015, at the start of the mission design process automation effort, this had increased to 128 days and 52,176 commands.
- In 2019, the team performed 170 days of telerobotic operations, sending 105,089 commands.

In addition to performing mission design work for tasks planned for execution in the near term, the Robotics Operations branch is increasingly called upon to perform task assessments for operations that will be performed more than one year in the future, or may, ultimately, not be performed at all. The team must also, on occasion, complete mission design work very quickly, when launch delays cause schedules or order of operations to change or emergencies such as the failure of critical hardware require rapid response and removal/replacement of the failed component.

This steady increase in workload is forecast to continue in coming years and necessitates the ability to plan and perform unique and increasingly complex telerobotic tasks more quickly without growing the size of the workforce. As a result, NASA's Robotics Operations branch constantly seeks and implements efficiencies in both the mission design (pre-execution) and execution phases of their operations. Although notable gains in execution efficiency have been realized (see [5] for examples), these are beyond the scope of this chapter, which instead focuses on the efficiencies gained through partial automation of the mission design process.

Table 1 presents average times required to perform seven typical mission design activities. Consider, for example, the relocation (or "walkoff") of the Canadarm2 from one base location to another. Per Table 1, prior to the introduction of the process automation software (2016 data), a walkoff operation took approximately 20 h for a mission designer to plan, and $3 \times 5 = 15$ h for three flight controllers to verify, for a total cost of 35 worker-hours in the pre-execution phase. The time to execute a walkoff operation is approximately four hours, supported by three flight controllers, for a total of 12 worker-hours cost to the organization, or a ratio of approximately three hours of pre-execution planning cost per one hour of operations execution. For more complex

Table 1 Mission design time savings due to process automation

Mission design activity	2016 (hours)	2018 (hours)	2020 (hours)	Time reduction (%, 2016 vs. 2020)	Hours saved per operation	No. of operations per year[a, b]	Annual savings (hours)
Develop procedure for an SSRMS walkoff	20.22	8.14	6.76	67	13.46	30	403.80
Develop procedure for SPDM stow or unstow	24.63	12.71	10.17	59	14.46	37	535.02
Develop procedure for robotics spacewalk support	39.15	26.86	18.48	53	20.67	6	124.02
Develop procedure to install a payload onto ISS	42.00	30.33	20.19	52	21.81	35	763.35
Verify SSRMS walkoff procedure	5.16	3.08	3.56	31	1.60	90	144.00
Verify SPDM stow or unstow procedure	7.12	4.25	4.52	37	2.60	111	288.60
Verify command scripts for SSRMS walkoff or SPDM stow/unstow	5.00	1.52	1.00	80	4.00	201	804.00

[a] Estimate for number of operations (procedures developed) per year is an average of the number of times each operation (e.g. SSRMS walkoff, payload install) was planned for (not necessarily performed) in 2018 and 2019

[b] Each procedure is verified by a minimum of three flight controllers. Therefore, the estimates for number of verifications is three times the average number of each operation in 2018 and 2019. This is conservative as verifications must be repeated if the initial verification discovers significant errors in the procedure, causing the procedure to be re-worked by the mission designer. This happens relatively frequently, but was not considered in the time savings estimate as frequency statistics were not available

robotics operations such as the installation of payloads, the planning-vs-execution ratio may be much higher.

Recognizing the time demands associated with executing their mission design process and, in 2015, already struggling to meet increasing demands for robotics operations on-board the ISS, NASA's Robotics Operations branch sought to automate the robotics mission design process with the goal of decreasing operations planning time while preserving (or ideally, improving) the process' level of safety and mission success. To this end, in December, 2015, personnel were contracted to develop a software tool that would automate some or all of the mission design process. A description of the resulting software application, christened the "Automated Robotics Mission Designer" (ARMD) and the process undertaken to develop it is provided in the following section.

2 The Automated Robotics Mission Designer (ARMD)

2.1 The Agile Software Development Model

At the time of project approval (December, 2015), the automation software was scheduled to be developed over a two-year period. Four key considerations drove NASA to adopt a Scrum-based [6] agile software development approach for this project:

1. The software development team was allocated to the project as soon as the project was funded; forcing them to wait to begin development work until after comprehensive requirements were written would have resulted in them being re-allocated to other work, potentially delaying the project significantly.
2. The Robotics Operations branch (customer) was eager to begin using the automation software as soon as possible, even if it did not yet include all desired functionality.
3. The software development team were not experts in the field of robotics mission design and the customer was not an expert in writing software requirements. Thus, significant and recurring customer involvement and testing would be necessary throughout development to ensure that the software was performing correctly and meeting customer needs.
4. There was insufficient initial time and funding to incorporate all desired software features, but there was potential for the project to be extended and additional funding approved if the project demonstrated early success. This necessitated delivery of working software as soon as possible and implied iterative software growth.

2.2 *Software Feature Elicitation and Valuation*

As is typical of organizations adopting an agile software development approach, the Robotics Operations branch quickly encountered the challenge of determining which aspects of the mission design process should be prioritized for automation as there were temporal, financial, and philosophical constraints preventing complete automation of the process. Various frameworks exist for performing such decision-making, with several directing the user to prioritize software features which provide the most “value” for the least “cost” (e.g. that provided by Popli et al. [7]). When using such models, value is typically equated to either quantitative measures of the software features’ potential to provide financial gain for the organization, or subjective and qualitative measures of user wants [8]. Cost is often expressed as a quantitative measure of financial outlay, worker-hours, or similar expression of lost opportunity, risk, or investment (e.g. story points [9]).

It is well known that human spaceflight is an inherently risky undertaking, and that, in pursuit of this bold endeavour, NASA has experienced failure, occasionally with deadly consequences such as the *Challenger* and *Columbia* disasters [10, 11]. In both these events, time pressure was identified as a key factor leading to poor decision making [10, 11]. In the case of robotics operations performed on-board the ISS, human error introduced during the mission design phase may have similarly catastrophic results and time pressure is present due to the increasing number of robotic operations demanded by the ISS program each year. This results in less-experienced personnel being assigned to perform robotics mission design work, fewer independent reviews of the work being performed by experienced personnel, and less time available to produce and inspect the work. These factors contribute to the human risk inherent in NASA’s robotics mission design process. In the case of the project at hand, the Robotics Operations branch’s prime goal was to leverage automation software to cope with time pressure and increase productivity (i.e. to reduce the time required to perform various mission design tasks). Believing that any introduction of automation to the process would improve process efficiency, and recognizing that the project would not be considered a success if the solution resulted in increased operational risk, the organization opted to adopt a human reliability assessment-based approach developed by Lucier [12] for identifying (eliciting) potential software features and prioritizing features for inclusion in the automation software.

The project team in February, 2016 began by performing a simplified Human Reliability Assessment (HRA) using task analysis methods described by Kirwan and Ainsworth [13]. The task analysis identified 51 high-level human operator tasks associated with performing the mission design process and 54 human error modes (Qx) that might be realized during the execution of those tasks. The team then identified software features (or high-level requirements) which could be implemented to eliminate the human operator tasks and/or eliminate or reduce the frequency of realizing the various human error modes. During team discussions, it was decided that software features corresponding to 11 tasks/error modes should be screened out of the feature candidate list; two due to the user community’s unwillingness to trust the software to address the error mode/perform the human operator task, six due to the combined user/software developer community's inability to envision an efficient automation solution to address the error mode/perform the human operator task, and three due to determination that failure to perform the task does not have a negative impact on the process' risk level.

From the outset of the project, NASA's Robotics Operations branch felt strongly that the automation software must, at a minimum, be able to autonomously plan robot trajectories as this task typically constitutes the greatest portion of the mission design process' human operator workload. Further, to be useful, all robot trajectories planned by the automation software would have to comply with NASA's various safety constraints; for example, those that require the robots to maintain a minimum distance from the ISS except under very specific circumstances. These fundamental needs led to the identification of six "must-work" functions, or features.

The team in May–June, 2016 employed expert elicitation methods to obtain estimates of the frequency or likelihood of realizing each human error mode. Corresponding software features were then assigned a measure of value (or "Importance") for use with Popli et al.'s [7] model for software feature prioritization. Consistent with Lucier's Reliability Improvement Score framework [12], the most valuable software features were defined as those corresponding to the most frequently realized human error modes and therefore, if implemented in the automation software, most likely to improve the mission design process' reliability. Figure 2 summarizes the seven step process described in this section. Additional details can be found in [12].

Table 2 details the final list of 44 candidate software features (Fx), their value (or Importance) expressed as a Reliability Improvement Score corresponding to

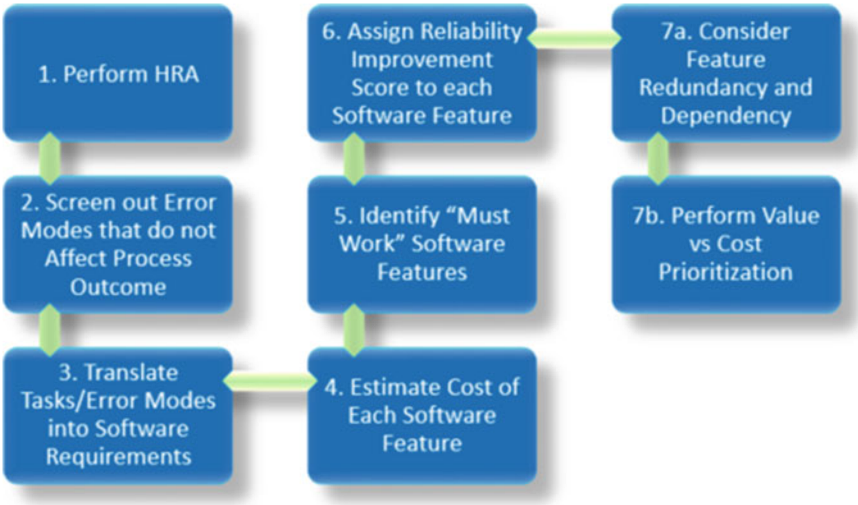

1. Perform HRA
2. Screen out Error Modes that do not Affect Process Outcome
3. Translate Tasks/Error Modes into Software Requirements
4. Estimate Cost of Each Software Feature
5. Identify "Must Work" Software Features
6. Assign Reliability Improvement Score to each Software Feature
7a. Consider Feature Redundancy and Dependency
7b. Perform Value vs Cost Prioritization

Fig. 2 Software feature elicitation and prioritization process

Table 2 Value-versus-cost evaluation results

Feature	Effort, E (cost)	Importance, I (value)	Popli prioritization (I/E)	Feature	Effort, E (cost)	Importance, I (value)	Popli prioritization (I/E)
0. No baseline functionality required				3. Crew commanded modes required			
F73	100	2.8148	0.0281	* F83j	5	0.4348	0.0870
* F63	50	1.0370	0.0207	* F83b	5	0.3636	0.0727
F3	100	1.8667	0.0187	* F82i	100	1.3913	0.0139
F65	100	1.1111	0.0111	* F83l	5	0.2609	0.0522
Sum	350	6.8296	0.0195	Sum	115	2.4506	0.0213
1. Autonomous fixed-base trajectory planning req'd				4. Auto base location determination required			
√ F71	5	4.1481	0.8296	F82ga	100	1.5833	0.0158
* F82fb	25	0.6667	0.0267	F82e	100	1.5000	0.0150
F82gb	75	1.5833	0.0211	F82fa	100	0.6667	0.0067
Sum	105	6.3981	0.0609	Sum	300	3.7500	0.0125
2. Autonomous procedure output required				5. Autonomous camera view/survey required			
√ F83d	5	3.7500	0.7500	F83fb	100	3.7500	0.0375
√ F81g	5	1.7500	0.3500	F83eb	100	3.5833	0.0358
√ F81e	5	1.6667	0.3333	F81c	100	2.7500	0.0275
√ F49	25	3.9259	0.1570	F80cb	100	2.6667	0.0267
F83ea	25	3.5833	0.1433	F82b	100	1.9167	0.0192
√ F47	25	3.4074	0.1363	F80db	75	1.2500	0.0167
√ F80b	25	2.9167	0.1167	F67	100	0.8889	0.0089
√ F80f	25	2.8333	0.1133	F81d	100	0.8696	0.0087
√ F80ca	25	2.6667	0.1067	Sum	775	17.6752	0.0228
√ F80h	25	2.4167	0.0967	F80f required incorporation of F80e F80ca required incorporation of F80b F80e required incorporation of F81j			
√ F82d	25	2.0870	0.0835				
√ F83g	5	0.4167	0.0833				
√ F82c	25	1.8333	0.0733				
√ F83c	25	1.6667	0.0667				
* F83hb	25	1.5833	0.0633				
√ F83k	25	1.5833	0.0633				
√ F81j	50	2.0000	0.0400				
√ F80e	50	1.8333	0.0367				
* F81f	5	0.1667	0.0333				
F82h	75	2.0000	0.0267				
F80da	50	1.2500	0.0250				
* F81a	100	1.5652	0.0157				
Sum	650	46.9022	0.0722				

the average aggregated expert estimations of the frequency of realizing the corresponding human error mode, and their cost in terms of story points, as estimated by the NASA robotics software development team. Also presented is each feature's value-versus-cost score (or Importance-versus-Effort score, using the terminology of Popli et al. [7]). Check marks indicate the 17 features that had been incorporated into the automation software as of July, 2018, the end of the two-year automation software development project. Asterisks indicate the additional nine features that had been incorporated as of January, 2020. Also shown on Table 2 are five baseline capabilities that the software would need to provide in order to permit automation of various discrete operator tasks and the incorporation of corresponding software features. The automation of these baseline capabilities formed five major phases of the robotics mission design process automation project, as indicated in Table 2.

2.3 Software Development, Functionality and Use

Following software feature elicitation and prioritization, the team shifted their focus to iterative rounds of software development, testing, and use. As stated previously, a critical function required of the ARMD software was autonomous path planning. A description of the algorithms used by the software to perform this task follows.

2.3.1 Robotics Path Planning Algorithms

A combination of established path planning methods was used to develop software functions to autonomously plan paths for the robotic manipulators to follow towards goal configurations while avoiding pre-described obstacles. These methodologies included the use of potential fields to determine goal configurations (attractive potentials) and unsafe/undesirable configurations (repulsive potentials) [14], as well as the use of a Dijkstra method for branch path exploration [15, 16].

To identify initial configurations to explore and evaluate, the algorithm uses a Dijkstra-based method to discretize, sort, and explore three-dimensional space with an n-armed robot (in the case of Canadarm2 or the Dextre robots' arms, $n = 7$). Individual seven-degree-of-freedom waypoints to the final goal are also determined using this exploration method, with their cost being evaluated according to constraints such as structural obstacles, ISS payload radiation hazard zones, self-collisions, and singularities, as illustrated in Fig. 3.

To reduce the high computational cost associated with evaluating discrete points for all n-joints across the entire range of allowable motion (in the case of Canadarm2 joints, ±270 degrees; corresponding to 541^7 possible discrete points if the discretization were performed in one-degree increments), the algorithm was designed to first start with a coarse discretization (e.g., 30 degree increments). Once viable paths are found using the coarse discretization, the space in the vicinity of those viable solutions is explored using a finer discretization to determine more efficient paths,

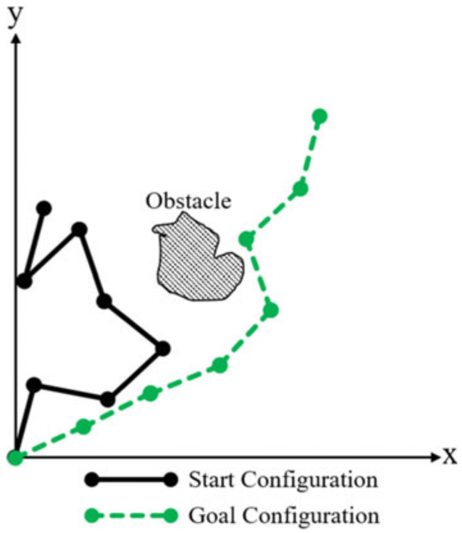


Fig. 3 Seven degree of freedom goal posture determination

as illustrated in Fig. 4. By taking this approach, the algorithm is capable of finding safe and usable solutions within in a timeframe reasonable to the user (on the order of 15–45 min for a typical Canadarm2 operation).

To evaluate waypoints to the goal, and determine joint rates for maneuvering between those waypoints, a potential field application similar to that proposed by Mayorga et al. was employed [14]. For a chosen set of m field potentials and a robotic manipulator with n-joints,

$$\Phi = \{\varphi_1, \varphi_2, \ldots, \varphi_i, \ldots, \varphi_m\} \tag{1}$$

$$\Theta = \left\{\theta_1, \theta_2, \ldots, \theta_j, \ldots, \theta_n\right\} \tag{2}$$

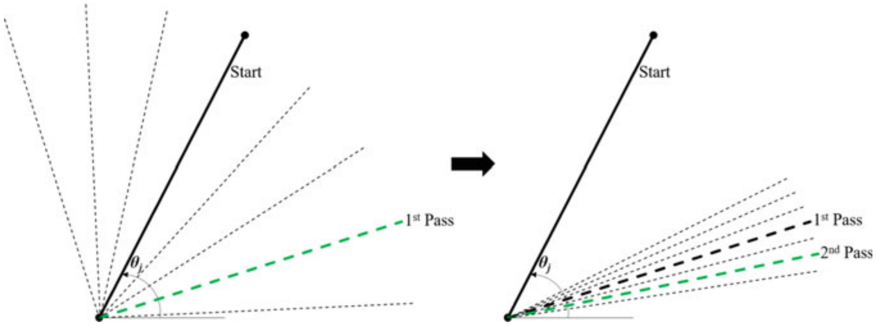


Fig. 4 Path discretization

with

$$\Phi = f(\Theta) \tag{3}$$

Given this defined set of field potentials, the Jacobian matrix can be defined as the time derivative of the potential constraints and leads to the following:

$$\frac{d\Phi}{dt} = \frac{df(\Theta)}{dx} = \frac{\partial f(\Theta)}{\partial \Theta}\frac{d\Theta}{dt} = J(\Theta)\frac{d\Theta}{dt} \tag{4}$$

where

$$J(\Theta) = \frac{\partial f(\Theta)}{\partial \Theta} \tag{5}$$

By taking the pseudo-Inverse of the Jacobian, joint angle rates can be determined relative to the potential rates:

$$\frac{d\Theta}{dt} = J^{+}(\Theta)\frac{d\Phi}{dt} \tag{6}$$

where the pseudo-inverse J^{+} is determined by:

$$J^{+}(\Theta) = J^{T}(\Theta)[J(\Theta)J^{T}(\Theta)]^{-1} \tag{7}$$

With this solution, the instantaneous joint rates that give the desired change to/from the potentials in the field can be stably determined by defining the rate of change according the distance from the potential goal and creating a feedback loop to drive values of the potentials towards their goal as follows:

$$\Phi_d = \Phi - \Phi_g \tag{8}$$

$$\frac{d\Phi}{dt} = -G\Phi_d \tag{9}$$

resulting in

$$\frac{d\Theta}{dt} = J^{+}(\Theta) * (-G\Phi_d) \tag{10}$$

This framework was applied to create the heart of ARMD; an autonomous path planning capability embedded within the Robotics Operations branch's existing RPS application used to perform the mission design process. Previously limited in capability to models of the ISS environment and robot kinematics, the end result is a powerful path planning tool that solves the constraint space to autonomously determine safe and effective robotic trajectories given initial and final robot configurations and a list of obstacles by the user (human mission designer).

2.3.2 Additional ARMD Capabilities

Although trajectory planning is the most time consuming task associated with the mission design process, the Robotics Operations branch desired the automation of additional tasks in an effort to improve overall process efficiency and eliminate tasks most frequently performed erroneously by human mission designers. To this end, the ARMD software incorporates various other notable capabilities, some large and some small. Examples of significant additional functionality include:

- Auto-generation of MSWord-formatted operational procedures. Prior to the automation project, mission designers were required to manually record robot trajectory solutions in a spreadsheet, and transfer the data from the spreadsheet into a text file for use with a second software tool, running on a disparate platform, to produce MSWord-formatted operational procedures (ref. Sect. 1.1). This original process also required the mission designer to manually insert into the text file instructions for intermediate operations such as mode changes and effector grasp/release, with the end result being a procedure that required yet additional manual editing of various parameters that could not be accommodated by the text converter. Conversely, the ARMD software program allows the user to easily select and insert, using a Graphical User Interface (GUI), such intermediate operations into their end-to-end task plan during initial mission design, and in many cases, is capable of autonomously inserting required tasks without any user input. With a single GUI button click, the ARMD software generates the MSWord-formatted procedure with little-to-no manual post-processing required.
- Auto-generation of Extensible Markup Language (XML)-formatted operational procedures. NASA is making increased use of XML-formatted (rather than MSWord-formatted) procedures. XML-formatted procedures provide various ease of use advantages such as decluttering (e.g. by allowing the user to expand or collapse optional or conditional instructions) and real-time procedure customization (e.g. by allowing the user to specify which power channel or equipment string will be used, and subsequently auto-populating the procedure with corresponding instructions). Prior to the automation project, mission designers were required to produce XML-formatted procedures using an XML editor application and library of pre-defined tags. The ARMD application autonomously generates XML-formatted procedures with little-to-no manual post-processing required.
- Auto-generation of command scripts. ISS flight controllers have various options for commanding to the vehicle. Commands may be selected and uplinked one at a time from an inventory of all available ISS commands, or scripts may be used to queue sequences of commands in the order they will be uplinked. Task execution is both more efficient and less error-prone when

commanding using scripts. The ARMD application auto-generates scripts for use with ISS's ground-based, open-loop command script execution tool. ARMD is also capable of auto-generating scripts compatible with ground-based and on-board autonomous (closed-loop) command execution tools capable of performing pre- and post-command condition checking.

- Ability to read/write files for team collaboration and procedure validation. The above-mentioned spreadsheet historically used for manual solution recording and procedure development was also relied upon to record the mission designer's preliminary work and communicate it to other team members. Conversely, ARMD provides the ability to write and read files that capture the mission designer's work and the resulting operational task sequence or choreography. When loaded to the ARMD application, these files configure RPS and the simulated ISS/robotic work scene consistent with the user-selected step in the task sequence. This allows team members to seamlessly pick up with or assess another's work, making mission design and procedure validation both more efficient and less prone to error. This capability also allows mission designers to save optimized trajectory sequences for re-use or use as a starting point when designing similar operations.
- Auto-generation of data files required for engineering analysis. In order to perform various pre-operations analyses (for example, assessments of loads imparted on the ISS solar arrays during robotics motion and the generation of transmission masks to prevent antennas from radiating when the robot is in the vicinity), files containing the robot's configuration (joint angles and end effector location) at fine intervals along every trajectory must be generated. The ARMD software is capable of generating these files nearly instantaneously where previously, the mission designer was required to maneuver the robot(s) in the simulator, through each trajectory, at real-time speed, while the simulator "recorded" the robot configuration at frequent intervals.

Examples of minor additional functionality include:

- Auto-calculation of maneuver times and auto-incorporation of the times into operational procedures. In order to determine how much communications coverage is needed to perform a given maneuver, robotics flight controllers must know the maneuver duration. Prior to the introduction of the ARMD software, mission designers were required to either use a stop-watch to time each maneuver while executing using a simulator at real-time speed, or were required to look up the joint motion rate for each operation in the appropriate arm control software file, determine how far each joint would move during each maneuver, and then calculate the time it would take for the maneuver to complete, considering change in joint angle and rate of change. The mission designer was then required to manually edit the MSWord procedure to indicate the maneuver time for each trajectory.
- Auto-generation of "clearance notes" and calculation of minimum distances between the robot/payload and surrounding structure. When maneuvering within five feet of ISS structure, robotics flight controllers are required to perform visual monitoring of the clearance between affected hardware. To do so, operational procedures contain "clearance notes" alerting the flight controller as to

the minimum clearance expected to be reached during the maneuver, which ISS camera to use to monitor the clearance, and which hardware components to monitor (e.g. clearance between the base boom and a payload pallet). The ARMD software is capable of auto-generating clearance notes and inserting them into operational procedures, though at this time, is limited to identifying and specifying minimum clearances and which hardware to monitor (the software is not yet capable of recommending which camera to use to monitor the clearance, though this capability is planned for a future iteration).

- Instant visual playback of robot motion and clearances for validating trajectories. This feature has significantly reduced the time required to perform procedure verification and to perform feasibility assessments of robot trajectories. Prior to the introduction of ARMD, mission designers were required to perform simulated commanding to the robot system in order to run through each maneuver. This included motion mode changes, loading of control parameters, etc. Alternately, the ARMD application is capable of stepping through motion sequences at the click of a single GUI button while simultaneously displaying to the operator distances between the robots and surrounding ISS structure.

3 Results

As previously stated, the primary goal of the automation project was to reduce the time to perform the robotics mission design process while preserving (or ideally, reducing) the process' error rate. As a result of the effort, the four-phase mission design process described in Sect. 1.1 was altered significantly, as illustrated in Fig. 5.

Because the organization did not have historical data regarding the time required to perform typical mission design tasks, members of the Robotics Operations branch were surveyed in 2016, prior to development of the automation software, in order to estimate baseline time durations for completing seven typical mission design tasks, as

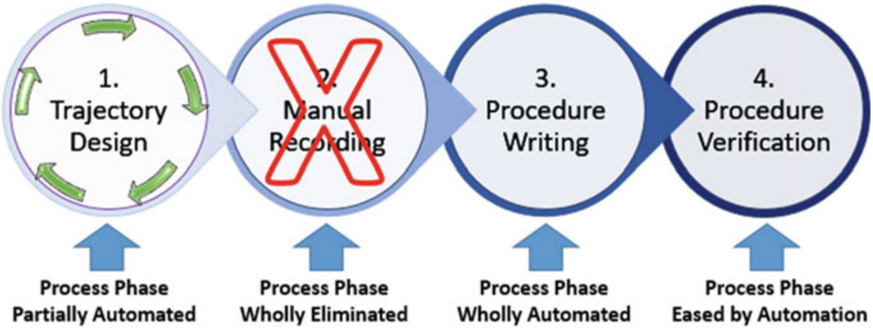

1. Trajectory Design
2. Manual Recording
3. Procedure Writing
4. Procedure Verification
Process Phase Partially Automated
Process Phase Wholly Eliminated
Process Phase Wholly Automated
Process Phase Eased by Automation

Fig. 5 Mission design process post-automation

detailed in Table 1. The team was again surveyed in 2018, after initial introduction and limited use of the ARMD software and again in 2020, by which time the software had been universally accepted by the team and become the de facto method of performing the mission design process. As shown in Table 1, the organization realized a 31–80% reduction in task execution times, resulting in a greater than 3000 worker-hour per year time savings. Considering that the cost of the initial, two-year development project (2016–2018) was approximately 4160 h, the NASA organization recouped their investment within the first 1.5 years of use, representing a high return-on-investment. These results are considered to be conservative as they only represent time savings associated with a sub-set of seven of the many tasks performed using the mission design process.

Members of the Robotics Operations branch were also asked to indicate via four- or five-point Likert scales the frequency at which 54 mission design process errors are realized. Figure 6 details the change in frequency or likelihood for each error mode, Qx, pre- (2016) and post-introduction (2020) of the automation software.

The automation project would be deemed a success only if no error modes experienced a statistically significant increase in error rate, though ideally, the organization desired a decrease in process errors. Fisher's Exact non-parametric tests for statistical significance [17] were performed to assess changes in error rates and resulted in the determination that since introduction of the automation software:

- Seven error modes have been fully eliminated (are no longer realized),
- Four error modes are realized very rarely (appear in Fig. 6 to have a frequency of zero),
- 12 error modes are being realized statistically significantly less often, and
- No error modes exhibited a statistically significant increase in frequency.

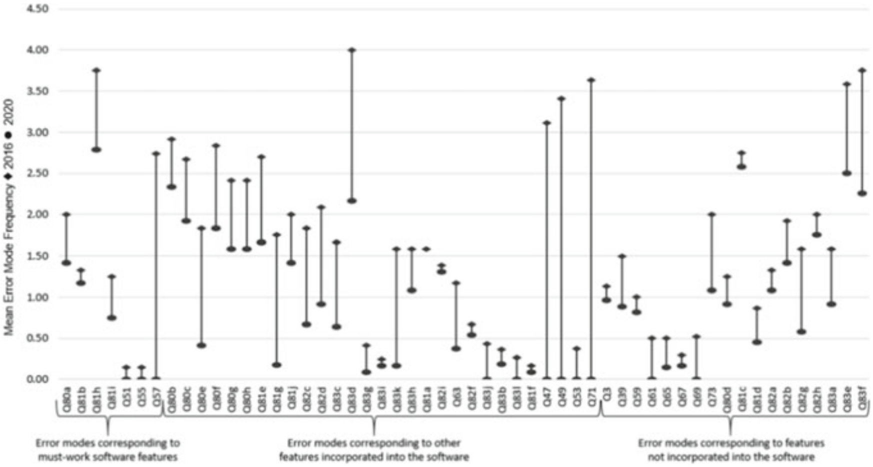


Fig. 6 Error mode frequencies

Of note is that three of the error modes which experienced a significant decrease in frequency do not correspond to features incorporated into the software. In these cases, the team feels that the introduction of the software indirectly contributed to the error mode reduction by way of reducing or eliminating some process tasks and therefore freeing operators to pay more attention to (and therefore more correctly perform) remaining tasks.

4 Discussion

Following brief discussion regarding data validity, this section focuses on two major challenges the team experienced in the development and adoption of the automation software: path planning optimization, and organizational acceptance of automation.

4.1 Data Validity

All data was generated by personnel trained and experienced in performing the mission design process, and was gathered in keeping with best practices in expert elicitation [18] and survey design [19]. Care was taken to minimize participant and researcher bias and statistical analysis methods appropriate to the nature of the data were selected. Though sample sizes were relatively small (26 experts participated in the 2016 survey, 28 in the 2018 survey, and 24 in the 2020 survey), resulting in a 10–11% confidence interval, these sample sizes correspond to participation rates of 67–72% across the total mission designer population. This, combined with the level of agreement seen in responses between participants suggests that the data is a reasonable reflection of opinion of the population as a whole.

4.2 Path Planning Optimization

One particular challenge experienced during the creation of the automation software was the complexity of the cost algorithm required to generate optimal robotics trajectories. As described in Sect. 1.2, there are various considerations that must be taken into account when designing the most efficient and effective path for the robot to follow. Consider, for example, that a longer path far away from structure may ultimately be quicker to traverse than a shorter path close to the ISS, where motion must occur at slower rates and only when video downlink capability is available. Additionally, it is difficult to define software rules to weigh trades such as maneuver efficiency versus loss of science due to allowing the robot to enter a zone within which it will block the field of view of a science payload, not in the least due to the fact that opinions regarding reasonable trades vary amongst experienced mission designers. In response, the ARMD software errs on the side of caution, trading efficiency for safety and mission success, and often causing mission designers to spend time manually optimizing the ARMD solution. An additional efficiency loss comes from the software's lack of knowledge regarding preceding and follow-on tasks, which may require a significantly different robot configuration than that planned by the software when considering a single operation in isolation.

Future refinement of the ARMD software's cost function, as well as incorporation of machine learning capabilities have the potential to significantly improve the software's performance in this regard and save the Robotics Operations branch additional process execution time.

4.3 Organizational Acceptance of Automation

During initial development of the automation software, the team identified two operator tasks which they did not trust the software to perform autonomously. Although the software performs many other operator tasks autonomously, the team has opted to "trust" the software blindly on some and perform verification of the accurate performance of others. This is not perceived to be a poor reflection on the performance of the software or lack of trust in the developers who created it and those who tested it, but rather a reflection of the Robotics Operations branch's sense of responsibility for the quality of the products they produce and a necessity dictated by NASA safety controls, which require that trained operators verify safety–critical aspects of robotics operations.

In addition to many members of the team needing time to develop confidence in the solutions generated by the software, members of the team have varying opinions regarding to what extent those who are new to mission design should be permitted to use the automation software versus learning to perform mission design the "old fashioned" (or manual) way. While some members feel that an overreliance on automation results in unacceptable loss of expertise, others feel that the effort required to develop and maintain such skills is unreasonable given the reliability of the software and workload of the team. Unsurprisingly, resistance to reliance on the automation software is most prevalent amongst mission designers who learned to perform the process prior to its availability, while those who have used the tool since the outset of their mission design training see little value in not using it.

Ultimately, general acceptance and wide-spread use of the software took approximately two years. By adopting an agile development approach, the organization was able to introduce the application in small pieces as its capability grew, which lessened the learning curve for those experienced in and reticent to leave the comfort of familiar, traditional mission design methods. User enticement measures included ensuring that the application automated several of the more tedious chores associated with the mission design process such as procedure and command script generation, as well as provision of comprehensive, on-demand training and reference materials in a variety of formats.

5 Conclusions

This project was successful in meeting NASA's goal of reducing the time required to perform the Robotics Operations branch's mission design process while reducing the process' risk. It delivered a high return-on-investment and demonstrated successful application of automation and autonomy to a high-risk, complex, human-in-the-loop process. It represents a stepping stone towards the increased reliance on automation and autonomy that is planned for NASA's exploration of space beyond low-Earth orbit and has enabled the Robotics Operations branch to better meet increased demand for ISS extra-vehicular robotics operations within the constraints of their existing workforce. The team continues to seek opportunities to further enhance and augment the capability of the ARMD software.

References

1. Mills I (2013) ROBO SPDM resource allocation request. NASA MOD control board presentation
2. Mathewson S (2017) NASA tests flexible roll-out solar array on space station, 20 June 2017. https://www.space.com/37250-roll-out-solar-arrays-on-space-station.html. Accessed 22 Feb 2020
3. NASA (2020) Robotic refueling mission phase 2. https://www.nasa.gov/mission_pages/station/research/experiments/explorer/Investigation.html?#id=920. Accessed 22 Feb 2020
4. NASA (2020) Robotic refueling mission. https://www.nasa.gov/mission_pages/station/research/experiments/explorer/Investigation.html?#id=765. Accessed 22 Feb 2020
5. Lucier LM, Watson WA (2021) International space station (ISS) robotics development operations team results in robotic remote sensing, control, and semi-automated ground control techniques. In: 16th international conference on space operations, Cape Town, South Africa, 2021, 3–5 May
6. Schwaber K, Sutherland J (2012) Software in 30 days: how agile managers beat the odds, delight their customers, and leave competitors in the dust. Wiley, Hoboken
7. Popli R, Chauhan N, Sharma H (2014) Prioritizing user stories in agile environment. In: International conference on issues and challenges in intelligent computing (ICICT), Ghaziabad, India, 2014, 7–8 February
8. Zacarias D (2017) 20 product prioritization techniques: a map and guided tour, 14 October 2017. https://foldingburritos.com/product-prioritization-techniques/. Accessed 22 Feb 2020
9. Cohn M (2005) Agile estimating and planning. Prentice Hall, New Jersey
10. Investigation of the challenger accident: report of the committee on science and technology, House Report 99–1016, Washington, 1986
11. Columbia accident investigation board report, Government Printing Office, Washington, 2003
12. Lucier LM (2019) A human reliability assessment approach to software feature elicitation and valuation: automation of NASA's robotics mission design process. University of Calgary, PhD diss
13. Kirwan B, Ainsworth LK (1992) A guide to task analysis: the task analysis working group. CRC Press, Boca Raton
14. Mayorga RV, Janabi-Sharifi F, Wong AKC (1994) A simple method for the collision avoidance of telerobotic manipulators. In: Proceedings of the 1994 IEEE international conference on robotics and automation, vol 3, San Diego, CA, USA
15. Dijkstra EW (1959) A note on two problems in connexion with graphs. Numerische Mathematik 1

16. Kang HI, Lee B, Kim K (2008) Path planning algorithm using the particle swarm optimization and the improved Dijkstra algorithm. In: 2008 IEEE Pacific-Asia workshop on computational intelligence and industrial application, Wuhan
17. Fisher RA (1944) Statistical methods for research workers, 9th edn. Oliver and Boyd, Edinburgh
18. Cormer MK et al (1984) NUREG/CR-3688: generating human reliability estimates using expert judgement. US Nuclear Regulatory Commission, Albuquerque
19. Czaja R, Blair J (2005) Designing surveys, 2nd edn. Pine Forge Press, Thousand Oaks

Operability on the Europa Clipper Mission: Challenges and Opportunities

Joel Signorelli, Marc A. Sarrel, and Meghana N. Kumar

Abstract Flight and ground system operability has been a focus area on the Europa Clipper Project since early in its formulation phase. This has given the operations team the opportunity to influence the design, with a goal of increasing overall system operability. This paper presents example operability challenges, opportunities, and solutions arising from the Critical Design Review (CDR) system design. The integrated wing assembly design directly couples a scientific instrument (the REASON sounding radar) to the spacecraft's power source (solar array wing panels). Impacts to mission operations of this design include: increased slew durations; solar array pointing constraints during inner cruise, Europa flybys, and orbit trim maneuvers; and stray light intrusions into the stellar reference units' keep out zones. The use of CCSDS File Delivery Protocol (CFDP) Class-2 for reliable downlink of the large volume of Europa Clipper science data is described, along with nominal and off-nominal use cases. The effort to improve post-launch spacecraft visibility by adding a third low-gain antenna to the spacecraft is detailed. The design of the bulk data store has necessitated the implementation of accountable data products (ADPs), accountability identifiers (AIDs), and metadata packets to provide end-to-end science data accountability. To streamline and automate the flight rules generation and checking process, a first order and temporal logic-based solution of expressing flight rules without ambiguity, and whose programmatic implementation can be automated, is proposed. The focus on operability has had a positive influence on Europa Clipper design decisions, although cost, schedule, budget, heritage, and other technical concerns have many times outweighed operability concerns. However, experience to date demonstrates that this approach to operability results in more thorough, balanced consideration of the effect of early design trades and decisions on the operations

J. Signorelli (✉) · M. A. Sarrel · M. N. Kumar
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
e-mail: joel.signorelli@jpl.nasa.gov

M. A. Sarrel
e-mail: marc.a.sarrel@jpl.nasa.gov

M. N. Kumar
e-mail: meghana.n.kumar@jpl.nasa.gov

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_27

phase of a mission than seen in many previous missions, and provides operations development insight into prioritizing work to go.

Keywords Operability · CFDP · Flight rules · Solar array

Acronyms/Abbreviations

ACK	Positive Acknowledgement
ADP	Accountable Data Product
AID	Accountability Identifier
BDS	Bulk Data Store
CCSDS	Consultative Committee for Space Data Systems
CFDP	CCSDS File Delivery Protocol
EOF	End Of File
FBA	Fan Beam Antenna
FreqID	Frequency Identification
JOI	Jupiter Orbit Insertion
KOZ	Keep Out Zone
MiB	MebiByte (1 MiB = 1,024 × 1,024 bytes = 1,048,576 bytes)
NAK	Negative Acknowledgement
OTM	Orbit Trim Maneuver
PDU	Protocol Data Unit
REASON	Radar for Europa Assessment and Sounding: Ocean to Near-Surface
RCS	Reaction Control System
RWA	Reaction Wheel Assembly
SADA	Solar Array Drive Assembly
SOC	State Of Charge
SRU	Stellar Reference Unit
TCM	Trajectory Correction Maneuver

1 Introduction

The concept of “operability” was adopted by the Europa Clipper mission early in its pre-project phase. Much of that early effort involved defining operability, determining how it could be quantified, and arriving at strategies to incorporate operability into the mission’s culture and design. Approximately 150 operability requirements were developed, addressing topics across the entire system, including components of both the ground and flight systems. Reference [1] describes in detail the early operability effort on Europa Clipper. Since that time, the Project has progressed from the formulation phases (Phases A-B) with a preliminary system design, to the implementation phases (Phase-C onward) with a baseline system design.

Recently the focus of the operability effort has shifted to ensuring operability requirements have been properly implemented in the baseline design, and assessing any proposed design changes for operability impacts (both positive and negative). As the design matures and lower-level implementing details are worked out, both "challenges" and "opportunities" to operability have been exposed. Challenges represent design features or characteristics that have the potential to degrade the operability of the system, while opportunities have the potential to enhance the system's operability. This paper presents several example challenges and opportunities uncovered as the flight and ground systems have matured from preliminary to final design.

2 Operability Overview

2.1 Operability Defined

In simple terms, operability is the quality of a system that represents how easy or difficult it is for humans to reliably and efficiently operate that system. Operability has been defined on the Europa Clipper mission as "A feature of the end-to-end system (including flight and ground segments) that enables the ground segment (comprising hardware, software, personnel, and procedures), to operate the space segment during the complete mission lifetime, using a minimum of resources, while maximizing the quality, quantity, and availability (or timeliness of delivery) of mission products, without compromising spacecraft safety."

2.2 Aspects of Operability

The Europa Clipper operability effort has defined nine "aspects of operability" as an attempt to describe the characteristics of an operable system. By maximizing these aspects, the operability of the system should generally increase [1]. The nine aspects of operability used by Europa Clipper are described below.

1. Visibility/observability: the extent to which the system provides the operations team with usable information about the configuration, status, and performance of the system.
2. Commandability/controllability: the extent to which the operations team can place the flight system in the desired state, and produce the desired outcome via commanding.
3. Predictability: the extent to which the operations team is able to predict, with some certainty, the outcome of the execution of a planned event.

4. Flexibility: the extent to which the operations team can reconfigure components to maximize or optimize component utilization, to circumvent anomalous components, provide options, to increase robustness.
5. Robustness: the extent to which the system maintains performance under perturbations, and prevents and contains errors.
6. Autonomy: the extent to which the system manages nominal or contingency operations without ground intervention.
7. Efficiency: the extent to which the operations team can optimize the use of time and resources.
8. Testability: the extent to which the operations team can verify and validate system components and test assets.
9. Tractability: the extent to which the operations team is freed from the need to pay attention to, or "care and feed" the system.

3 Europa Clipper Overview

3.1 Mission Overview

The Europa Clipper mission is formulated, implemented, and operated by a joint Jet Propulsion Laboratory (JPL) and Johns Hopkins Applied Physics Laboratory (APL) project team. The Europa Clipper mission will explore Jupiter's "water world" moon Europa, and investigate its habitability utilizing a set of five remote sensing instruments that cover the electromagnetic spectrum from thermal emission through the ultraviolet, four in-situ fields and particles instruments, a two-channel radar, and a gravity science investigation [2]. Clipper will produce high-resolution images of Europa's surface, determine its composition, look for signs of recent or on-going activity, measure the thickness of the icy shell, search for subsurface lakes, and determine the depth and salinity of Europa's ocean. Additional goals are to characterize the radiation environment near Europa, and investigate scientifically compelling landing sites for a potential future *landed* mission.

The Europa Clipper spacecraft is scheduled to launch in 2024 on an indirect trajectory, using a Falcon Heavy commercial launch vehicle. This 5.5 year long trajectory includes Earth and Mars gravity assists, and will have a perihelion distance of 0.82 AU (Fig. 1).

Because of the intense radiation environment in the vicinity of Europa's orbit, a multiple flyby mission approach was selected over an orbiter (Fig. 2). This minimizes the spacecraft's time near Europa, and hence its exposure to radiation. The 3.8 year long science tour portion of the mission consists of over 50 close flybys of Europa, with many as low as 25 km above the surface. Each encounter is divided into four subphases (Fig. 2): the approach subphase, beginning approximately two days prior to closest approach; the nadir subphase, when the spacecraft is closest to Europa and in a strictly nadir-pointed attitude for science data collection; the departure subphase, extending from the end of the nadir subphase until about two days after

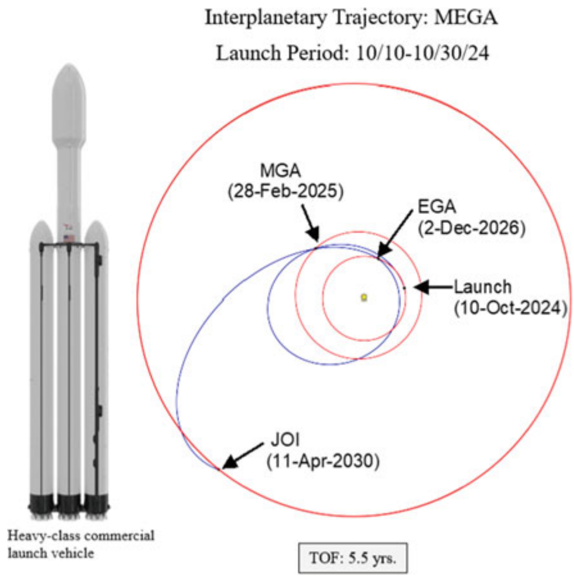


Fig. 1 Europa Clipper launch period and interplanetary trajectory

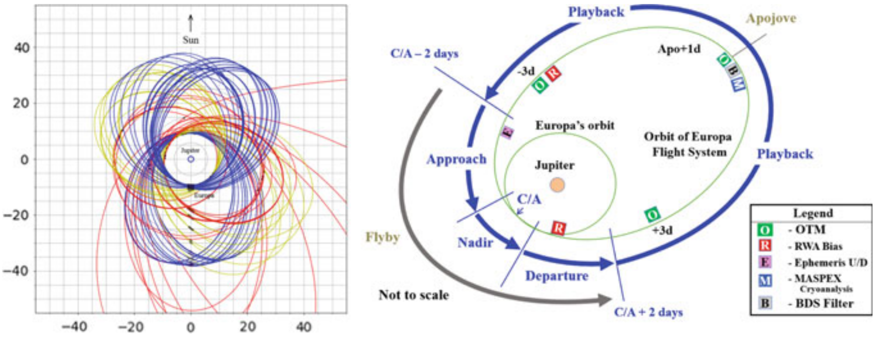


Fig. 2 Multiple Europa flyby approach (left); typical Europa Clipper encounter with Europa (right)

closest approach; and a playback subphase, where the data recorded during the flybys are transmitted to the ground.

Almost all of the of the science data collection occurs in the approach, nadir, and departure subphases, with the exception of some in-situ (fields and particles) data collection and occasional instrument calibrations during the playback phase. Due to downlink data volume limitations and high science data volumes collected on each encounter (approximately 80 Gbits), there may be latencies of up to several weeks before all the science data is returned to the ground from a given encounter. The operations team will prioritize data return based on science-driven priorities in order to receive decisional data in a timely manner early in the playback phase that is “fed forward” to use in design of observation plans that require optimization or information from a prior encounter. Three orbit trim maneuvers (OTMs) are executed during the playback phase to maintain and optimize the flight system’s planned trajectory and the flyby altitude.

3.2 Flight System Overview

The Europa Clipper flight system consists of a spacecraft bus and 9 scientific instruments (Fig. 3), and has a wet mass of 6,001 kg. The spacecraft is solar powered, with two gimbaled solar panel wings (102 m^2 total area) providing 700 W power at end of mission. A 483 Ah (BOL) Li-Ion battery provides power during high-demand periods (e.g., Europa flybys), and during eclipses. Trajectory maintenance and coarse attitude control are provided by twenty-four 25-N bipropellant engines (MMH/NTO). Data storage is provided by a 512 Gibit bulk data store (BDS). Attitude control consists of four reaction wheel assemblies (RWA) for precise attitude control, 24 bipropellant engines, two stellar reference units (SRU), two inertial measurement units (IMU), and four digital Sun sensors. A 20 W (RF) X-band radio will be used for uplink and selective downlink. A 35 W (RF) Ka-band radio will provide the primary link for science data downlink. The telecom subsystem utilizes one high gain antenna (HGA), one medium gain antenna (MGA), three fan beam antennas (FBA), and three low gain antennas (LGA). Thermal control is maintained by an active thermal pump loop, heaters, and louvers. The spacecraft avionics is mostly contained in an aluminum vault, to significantly reduce the radiation total ionizing dose experienced by the avionics.

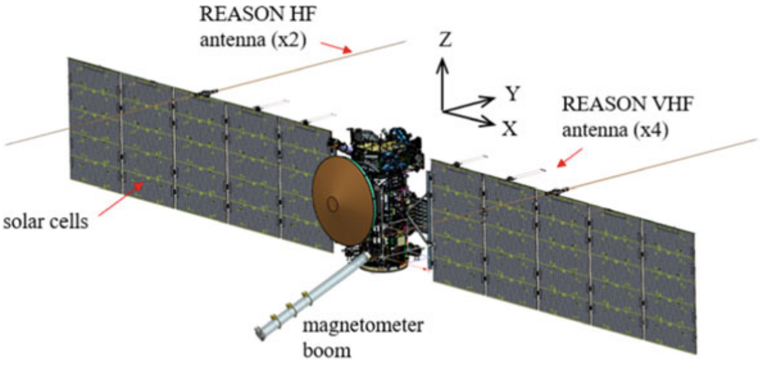


Fig. 3 Europa Clipper flight system configuration

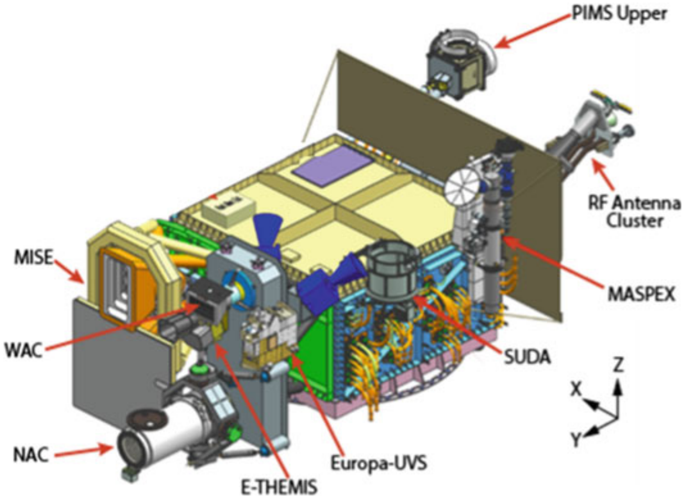


Fig. 4 Europa Clipper NADIR deck mounted instruments

The instrument suite consists of: REASON (dual band ice penetrating radar); EIS-WAC (wide angle camera); EIS-NAC (narrow angle camera); MISE (shortwave infrared spectrometer); MASPEX (neutral mass spectrometer); ECM (magnetometer); PIMS (plasma sensor); Europa-UVS (ultraviolet spectrograph); E-THEMIS (thermal imager); and SUDA (dust detector). The NADIR deck mounted instruments are shown in Fig. 4. In addition, the gravity science investigation is performed via the telecom system, while RADMON performs radiation monitoring. All instruments are body mounted, and the spacecraft points the remote sensing instruments toward nadir during most of the flyby, and points the instruments designed to sample material from Europa itself in the velocity-facing direction at closest approach. The infrared spectrometer, has an internal mirror that further allows it to scan along-track to compensate for target motion when close to Europa. The narrow-angle camera has a 2-axis gimbal to allow for acquisition of stereo coverage and to extend its field of regard to off-nadir targets.

4 Operability Challenges and Opportunities

Presented below are several topic areas that have operability challenges and opportunities associated with them. These have been uncovered as the flight and ground system designs have matured and implementation work has progressed.

4.1 Integrated Wing Assembly

A power source trade study was held back in the spring of 2015 to determine which of the proposed power source options (several radioisotope thermoelectric generator based options, and photovoltaic) would be most appropriate for the Europa Clipper mission. The Project selected a photovoltaic (i.e., solar array) power source as the preferred option. At that time, the Operability Working Group identified several characteristics of the proposed solar powered mission that could impact the operations team including: a post-launch, mission-critical solar array deployment; variable power output as a function of spacecraft attitude, solar distance, solar cell radiation degradation; large inertial properties leading to increased turn slew rates, settling times, and potential for dynamic structural interactions; inner cruise thermal constraints on a solar array; significant solar array configuration coupling with other subsystems (e.g., the REASON radar instrument); potential for added off-Sun attitude constraints; reduced power generated during eclipses and fault (safing) scenarios; and the need to continually articulate the array for optimal power production [1].

In addition, various spacecraft configurations were investigated for accommodating the REASON instrument's VHF and HF antennas onto the spacecraft. Trade studies were performed to decide on the best configuration for both instrument performance, with least impact on the spacecraft structure and the other instruments (e.g., antenna impingement in to fields of view, glint, electromagnetic compatibility, power and data routing, cabling stiffness, etc.). Ultimately, a configuration was chosen that integrated the REASON instrument along the leading edge of the solar array wings. The solar array and REASON instrument now became coupled into an "integrated wing assembly"—the spacecraft's power source was now coupled to an instrument. The REASON accommodation challenge was to provide a sufficiently predictable radar environment for REASON transmission and reception, while the solar array accommodation challenge was to ensure the solar array still deploys properly and provides reliable power. The REASON VHF and UHF antennas are mounted normal to the leading edge of the solar array (Fig. 3).

As the flight system design has matured, a greater understanding of the impacts of these decisions on mission operations has been exposed. Impacts to mission operations of this design include: increased slew durations; solar array pointing constraints during inner cruise; solar array positioning during Europa flybys and during orbit trim maneuvers; and stray light intrusions into the stellar reference units' keep out zones.

4.1.1 Spacecraft Inertia Properties Growth

The Europa Clipper spacecraft, with its 30.5 m tip-to-tip solar array, will be one of the largest solar arrays flown on a deep space mission to date. Figure 5 illustrates a relative comparison between Europa Clipper and three other deep space missions, Dawn, Juno, and Juice.

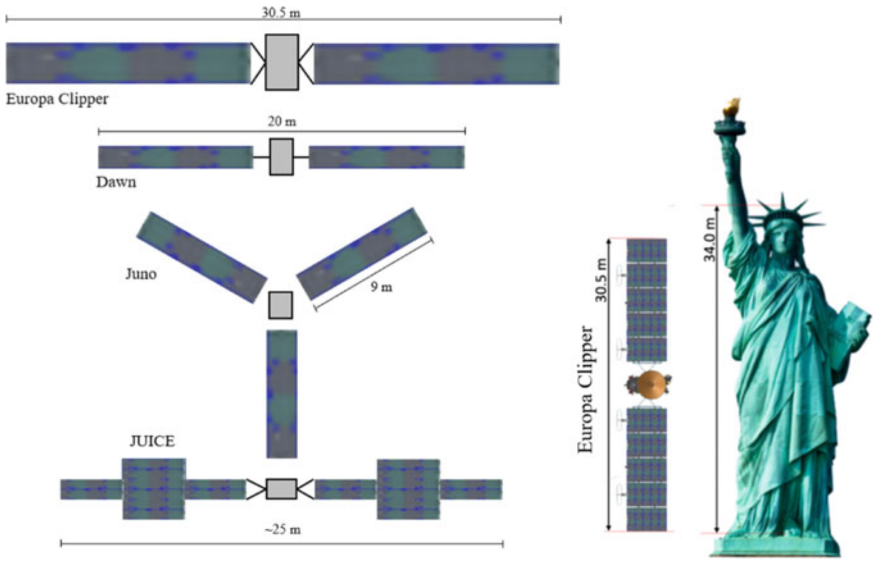


Fig. 5 Solar array relative size comparisons

As the Europa Clipper mission and flight system design have matured, the size of the solar array has grown to support the predicted power and energy load. Since the Preliminary Design Review (PDR), the length of the spacecraft (along the X-axis), has increased 3.5 m, to 30.5 m tip to tip. This has led to the bounding moments of inertia to become [Ixx, Iyy, Izz][1] $= [10437, 54{,}432, 48454] \times 10^4$ kg m^2, representing an approximate 30% increase in spacecraft inertia. A relative comparison of the moments of inertia between Cassini, Dawn, and Europa Clipper is shown in Fig. 6. This increase in inertia properties has resulted in 180° pure body axis slew durations on RWAs of 70 min for X-axis slews, and 90 min for Y and Z axis slews, along with commensurate decreases in RWA torque and momentum margins, in the majority of operational cases. The large increases in inertia necessitated added complexity in the form of scenario-based agilities, some of which are slightly more restrictive, and some slightly less.

These slew durations can negatively impact the *efficiency* and *flexibility* aspects of operability. Although long slew durations can be incorporated into activity planning, they reduce the amount of time available for other activities (e.g., science acquisition, data downlink, battery charging). *Flexibility* to add targets of opportunity activities to a timeline are constrained by time required to perform the slews.

Europa Clipper was designed primarily to acquire science observations while nadir pointed during Europa flybys. Therefore, the ability to incorporate off-nadir observations (e.g., Europa-UVS Jupiter transits and stellar occultations) needs to consider increased slew durations.

[1] MEV, arrays deployed and in power configuration, propellant tanks full.

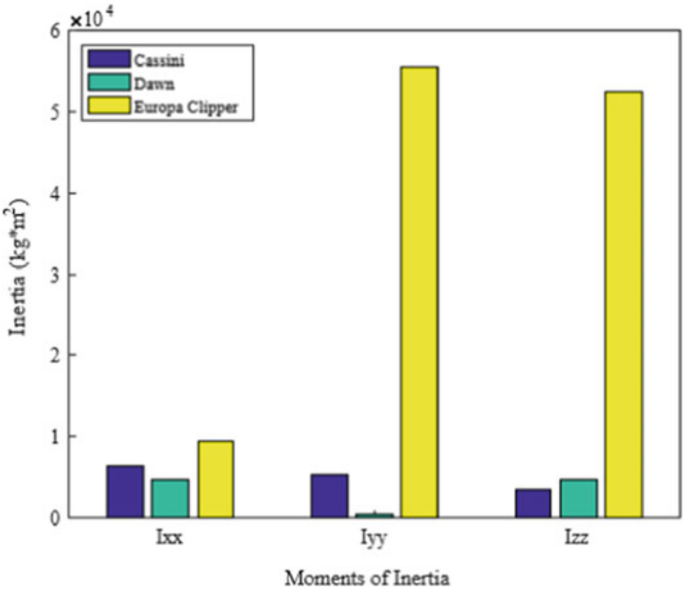


Fig. 6 Relative comparison of the moments of inertia between Cassini, Dawn, and Europa Clipper

During Europa science campaign 1, Europa Clipper Magnetometer calibration rolls are currently planned every 3rd Europa encounter. These consist of 6 revolutions about the X-axis inbound, and 6 revolutions about the Y-axis outbound, ending and beginning at ~165,000 km altitude, respectively. Because of the ponderous slew durations to perform 360° rolls, each pair of calibration rolls is estimated to take ~25 h to perform. The inbound calibration takes the array periodically off-Sun and precludes the use of the HGA, while the outbound calibration precludes use of Ka-band. This negatively impacts the *visibility*, *flexibility*, and *robustness* aspects of operability.

However, the project's high-fidelity mission-level modeling and simulation capability, developed early in the pre-project phase, has been invaluable in rapidly assessing the impacts of these constraints on mission operations [3].

4.1.2 Frequency Identification Activity

Because of the large size of the Europa Clipper solar array and its integration with the REASON instrument (aka, the integrated wing assembly), it is impractical to perform credible, predictive ground-based deployment testing. Predicting and verifying the natural structural frequencies of the integrated wing assembly with reasonable uncertainty is therefore problematic.

Spacecraft attitude controller designs incorporate notch and roll-off filters to stabilize structural modes. Fixed-gain controllers rely on wide notches to accommodate mode frequency variability due to changes in solar array position, assumed frequency knowledge errors, and spacecraft configuration and mass property changes.

Because of this pre-launch frequency knowledge uncertainty, an on-orbit frequency identification activity (FreqID) must be performed. Structural modes in the 0.07–0.7 Hz range are of most interest (<10 modes). The initial post-solar array deployment attitude controller uses a robust, low bandwidth implementation (known as "Super-Safe") which requires a minimum frequency guarantee. But the spacecraft cannot perform fine pointing and moderate slew rates until new control parameters are identified and uploaded.

The first FreqID activity will be performed to identify the first modes of the spacecraft (with deployed solar array, stowed ECM boom, and stowed REASON antennas). This test is done by open-loop firing of the engines to excite modes, at two solar array positions (0°, 90°), while recording high-rate data from the IMU. These data are downlinked to the ground, analyzed, new attitude control parameters are determined, and then uplinked to the spacecraft. This activity is expected to take 2.5 days.

FreqID-1 must be performed before TCM-1 (currently scheduled as early as L + 14 days) can be performed. Also, supportable downlink rates via the LGA decrease quickly after launch, motivating the need to transition to inertial pointing as soon as practicable. Inertial pointing is required to use the higher performance fan beam antennas. The FreqID-2 activity will take place after the ECM boom and REASON's VHF and HF antennas are deployed.

The FreqID activities do not present a long-term (mission duration) challenge to operability. But they do, however, require the operation's team attention and interaction with the spacecraft in the critical post-launch period (decreased *tractability*). Because of the difficulties associated with characterizing the array's dynamics on the ground, *testability* is reduced, thereby making an on-orbit test essential. Because of the constraints imposed by the pre-FreqID spacecraft, the operations team is limited in its ability to interchange post-launch checkout activities, thus limiting their *flexibility*. Downlink telemetry rates will be limited until the spacecraft is able to transition to inertial pointing after FreqID-1 is complete, thereby potentially limiting the operations team's *visibility*.

4.1.3 Solar Array Position During Propulsive Maneuvers

During the Europa Clipper mission, approximately 250 propulsive maneuvers will be performed by the spacecraft, using its twenty-four 25 N reaction control engines. These propulsive maneuvers consist of trajectory correction maneuvers (TCMs), orbit trim maneuvers (OTMs), and Jupiter orbit insertion (JOI). These maneuvers will be performed using a "turn-and-burn" approach where the spacecraft will perform slew(s) to align the +Z axis along the desired ΔV direction, the burn will be executed, and a return slew to the original attitude will be performed. Maneuvers will utilize an on-board reusable "ΔV Behavior" which will be seeded with each maneuver's specific arguments and parameters.

During ΔV maneuvers, the solar arrays must be orientated in the "home position," with the solar cells pointed toward the $-$Y axis, co-aligned with the HGA (Fig. 3). The primary reason that this is the preferred orientation is that dynamically this is the stiffest orientation of the array to thruster inputs along the Z-axis. This enables more accurate pointing control during the maneuvers. The secondary reason for this orientation is to minimize the risk of contamination and heating from RCS plume impingement on the REASON instrument's VHF and HF antennas.

The solar array must be re-oriented to the home position for each of the 250+ propulsive maneuvers. Tens of minutes must be allowed for: post-slew solar array and spacecraft dynamics slew settling + solar array drive assembly (SADA) articulation + settling time after SADA articulation. However, all of this is accounted for in the ΔV Behavior, making for increased operability (*efficiency*).

Reorienting the array to the home position will frequently result in reduced solar input to the array, and reduced battery state of charge, therefore reducing power margins (*robustness*). This reduced available energy can be accounted for during the activity planning and sequence generation process, so should not be a significant impact to operability.

The nominal 6.5 h JOI burn is also performed with the array in the home position. Any decrement to battery SOC due to array position during JOI is well within the battery capacity, since the spacecraft battery capacity was primarily driven by anomaly scenarios directing following a long, 9.2 h eclipse.

4.1.4 Inner Cruise Solar Array Pointing Constraints

"Inner cruise" has been defined as the time period before the flight system crosses 2 AU from the Sun for the final time in the mission. Several additional operations-impacting constraints arise during inner cruise.

Thermal Constraints

The Europa Clipper solar array is being designed for thermal conditions from a 0.82 AU perihelion to 5.6 AU (Jupiter orbit). During inner cruise, the HGA is used as a Sun shield to protect temperature sensitive components of the spacecraft, and certain thermal precautions must also be observed to protect the solar array. Spacecraft orientation is constrained primarily to $-$Y to Sun (i.e., HGA to Sun) ±5°, with the exception of performing trajectory correction maneuvers (TCMs), and some GNC calibration activities such as the FreqID.

The solar array normal must be feathered to at least 41° from the Sun at all times when the spacecraft is <0.95 AU from the Sun (Fig. 7). This requirement prevents solar array temperatures from exceeding allowable limits and causing an undervoltage. In addition, the REASON instrument needs to be canted back at least 66° from the Sun line in order for components of its matching network and edge ground to remain thermally safe.

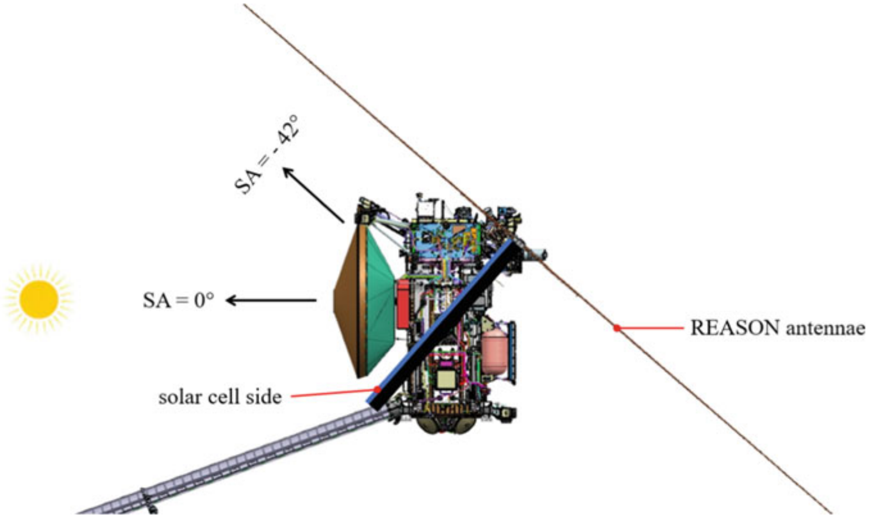


Fig. 7 Inner cruise configuration, solar array offset from Sun line

TCMs During Inner Cruise

Inner cruise constraints also affect how trajectory correction maneuvers are performed while in inner cruise. The Europa Clipper Mission Plan requires that the flight system must be able to perform statistical maneuvers, in any direction, at ≥0.9 AU. Several constraints must be maintained while performing TCMs at 0.9 AU: the array must be offset 41° from the Sun (section "Thermal Constraints"); the array must be held at 0° offset for stability (Sect. 4.1.3); REASON must be >90° from the Sun; and there can be no Sun exposure within ±45° of the SRU keep out zone. Because of this, the standard "ΔV behavior" cannot be used as is.

Each inner cruise TCM will require extra effort by the operations team to design a "non-standard" maneuver that is not directly supported by the ΔV behavior. It will necessitate that a "wrapper" sequence to be developed and tested. This wrapper sequence will contain two turns before the burn, and two turns after, and implement a strategy of pointing the solar array *edge* to Sun while articulating the arrays and slewing to the maneuver attitudes. These non-standard maneuvers will necessitate more effort by the operations team to develop, test, and execute, resulting in decreased *efficiency*, *tractability*, and could introduce more potential for human error.

4.1.5 Solar Array Position During Europa Flybys

In planning Europa flybys, the operations team must balance several conflicting needs including: (1) having the solar array track the Sun to provide power; (2) placing the solar array in the −90° standard closest approach orientation to orient the REASON antennas for nadir sounding observations; and (3) precluding SADA articulation during observations requiring fine pointing stability.

A typical flyby incorporating the above constraints may produce the following scenario [2]:

At ~12 h before closest approach (~165 k km altitude), the spacecraft will turn its HGA away from Earth to point the remote sensing instruments towards Europa. The spacecraft uses a “nadir-Sun” attitude, which twists the spacecraft about the nadir axis to point the solar array directly at the Sun. For the bulk of this period, the SADA is actively tracking the Sun and power is being generated by the array. However, during EIS observations the array must be held in place in order to ensure sufficient pointing stability (i.e., avoid disturbances from SADA stepping/array structural dynamic interactions). If energy is at a premium for a given flyby, it is possible to delay the turn to the next attitude (the “ram-optimized” attitude) to keep full Sun on the arrays slightly longer without impacting too many science observations.

At ~4.4 h before closest approach (~60 k km altitude), the spacecraft will turn to twist the spacecraft about nadir to align the remote sensing instruments’ FOVs with the velocity direction in the “nadir ram-optimized” orientation. During this time the solar array is fixed, and power may still be generated by the array, but in a diminished capacity. Fixing the arrays ensures a highly stable platform for certain science observations.

At ~27 min before closest approach (~5000 km altitude), the array is articulated to the −90° standard closest approach orientation which physically orients the long axis of the REASON antennas in the ±Z direction, with the edge of the array that houses the REASON antennas in +Y direction towards nadir (Fig. 8). Although REASON

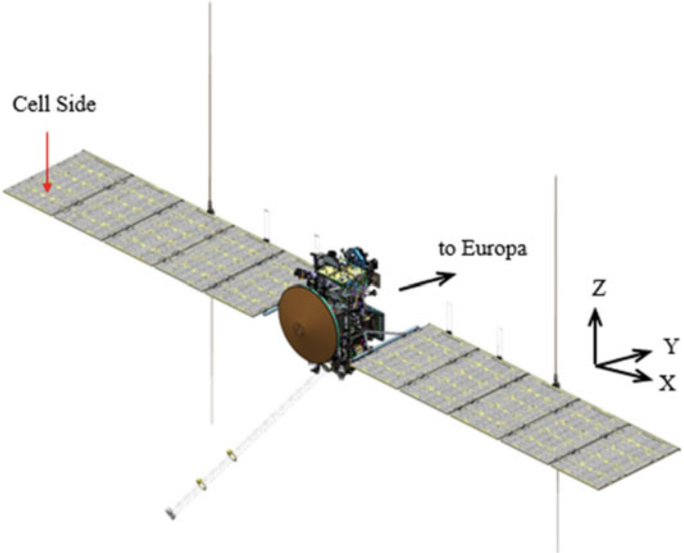


Fig. 8 Solar array orientation during Europa flyby

does not begin sounding until 1000 km altitude, other science observations such as NAC pushbroom measurements require the solar arrays to remain fixed, and there will be no time between these observations and REASON to perform a solar array actuation.

This scenario is then roughly mirrored on the outgoing portion of the flyby. Because of the varying geometric conditions and scientific objectives during the tour, the timing and content of each flyby is different, although inherently similar. Each individual flyby will need to be developed and validated by the operations team on a rapid cadence to support each 14-day encounter.

The above flyby scenario highlights some of the challenging attributes arising from the spacecraft's integrated wing assembly design, where the power source is integrated with an instrument. In order to address these challenges *efficiently* and *predictably*, the operations team needs automated tools which can: (1) accurately predict battery SOC and energy as a function of desired flyby activities, trajectory geometry, and spacecraft attitude; (2) schedule science activities in their required order (meeting their geometric constraints); (3) account for spacecraft and instrument dynamic interactions, slew times, settling times; and (4) check to ensure no flight rules or constraints are violated during the flyby period.

4.2 Large Science Data Volume for Downlink

The Europa Clipper mission is expected to generate over 5 Tbits of scientific and engineering data during the life of the mission. Each Europa flyby alone will generate ~80 Gbits of data. On previous deep space missions a team of Data Management engineers was responsible for downlink data accountability. This team was responsible for reconciling data that was requested for downlink with actual data successfully received on the ground. If requested data was not successfully received, they would need to generate uplink commands to request the missing data be retransmitted to the ground. They would also have to send commands to delete the on-board data once it was successfully received on the ground. This data accountability function could be a tedious, manual, or semi-automated process (decreased *tractability* and *efficiency*).

Europa Clipper will utilize Ka-band downlink to help maximize the downlink telemetry rate. But Ka-band is more susceptible to environmental effects (e.g., rain and wind) than X-band, which could potentially result in more downlink data losses which the Data Management team would need to rectify.

The Europa Clipper project saw an opportunity to help improve the *efficiency* of downlink data accountability by utilizing the CCSDS File Delivery Protocol. Using this protocol would help increase the efficiency of the operations team by allowing them to focus on tasks that actually require their attention, as opposed to doing the routine, repetitive, and manual tasks associated with data accountability of 5 Tbits of data.

4.2.1 CCSDS File Delivery Protocol

The CCSDS File Delivery Protocol (CFDP) is a standard for transmission, retransmission, and acknowledgement of files between ground stations and spacecraft [4]. JPL missions have historically used a variety of mechanisms to accomplish this task. The simplest mechanisms involved transmission from the spacecraft with no possibility of retransmission of missing data. On some missions, retransmission was possible, but was used only infrequently for very important data. More capable home-grown solutions allowed for retransmission on a routine basis, but required a human, using ground software, to build the commands to retransmit data.

The primary benefit of CFDP is that it automates the process of retransmitting missing data, and deleting data from the spacecraft that has successfully been received on the ground. On a routine basis, no manual intervention is needed.

4.2.2 How CFDP Works

The way that CFDP operates is relatively simple. Figure 9 illustrates how a single file is downlinked from the spacecraft and how data is retransmitted. Europa Clipper uses CFDP class 2 with automatic retransmission. CFDP class 1 is the same, but has no retransmission capability. CFDP classes 3 and 4 are similar to classes 1 and 2, but are for relay via one or more waypoints (other spacecraft or ground stations).

Each chunk of information that CFDP sends and receives is called a protocol data unit (PDU). First, the spacecraft sends the metadata PDU for the file. This includes

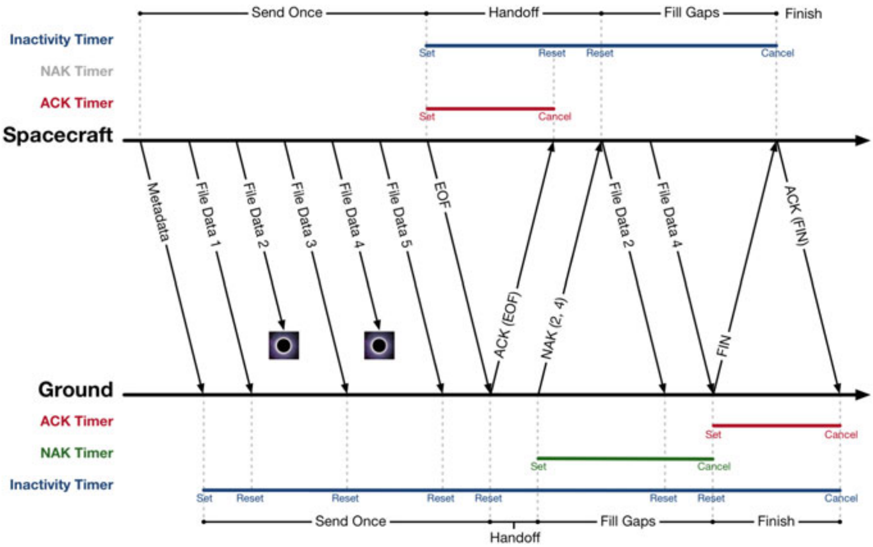


Fig. 9 Single CFDP downlink transaction, with retransmitted data

the file name, the CFDP transaction number and several other pieces of information. Then, the spacecraft sends the file data PDUs. These contain the actual content of the file. In this example, there are five PDUs, but typically, there could be tens or hundreds. Two of the file data PDUs, numbers two and four, are lost in transmission. Finally, it sends the end of file (EOF) PDU, which includes the file checksum for data integrity. The horizontal offset in this diagram between when a PDU is transmitted and when it is received represents the one-way light time between the ground and the spacecraft.

Upon receipt of the EOF PDU, the ground responds with an ACK (EOF) PDU to acknowledge its receipt. Then, in this case, the ground sends a NAK PDU to tell the spacecraft which data must be retransmitted. The spacecraft responds with the missing data. Having now received all File Data PDUs, and having confirmed the file checksum, the ground sends the FIN PDU to tell the spacecraft that it has received the complete file. Upon receipt of the FIN PDU, the spacecraft deletes the file and sends the ACK (FIN) PDU to the ground to complete the transaction.

On a routine basis, no manual intervention is needed. However, there are several operational constraints that have driven the configuration and planned use of CFDP. Had these not been identified and addressed before launch, more manual intervention would have been needed, and the benefits of using CFDP would have been reduced.

4.2.3 CFDP Timers

Timers, on both flight and ground, are an important part of the CFDP protocol. They ensure that transactions will eventually complete, even if some of the PDUs are lost. If the spacecraft sends the EOF PDU, but does not receive an ACK (EOF) PDU from the ground before the ACK timer resets, it will resend the EOF PDU. After a specified number of retries, the lack of the ACK (EOF) PDU from the ground will trigger a CFDP fault response. CFDP fault responses are a routine part of the CFDP protocol and do not indicate any sort of problem or anomaly with the spacecraft.

The CFDP timers only run when the reception of CFDP PDUs is expected. That is, they only run during telecommunication passes. Figure 10 shows the nominal pattern. On the spacecraft, the timers thaw (start running) when the spacecraft transmits the first CFDP PDUs. On the ground, the timers start to run upon first receipt of CFDP PDUs from the spacecraft and end with the last receipt.

Europa Clipper must use its downlink bandwidth efficiently in order to make sure the ground receives all the data collected during the mission. Judicious configuration of timers, and choice of CFDP fault responses were needed to achieve this. As indicated in the CCSDS specification document, the default CFDP fault response for a lack of ACK (EOF) from the ground would be to cancel the transaction. The file would remain on board the spacecraft, but the state of the transaction would be lost. If there were gaps in the file on the ground, the only choice to fill them would be to retransmit the entire file. To avoid this, Europa Clipper changed the fault response in question. After the specified number of retries, the spacecraft will send the ground a warning indicating that the limit has been reached. But, the spacecraft

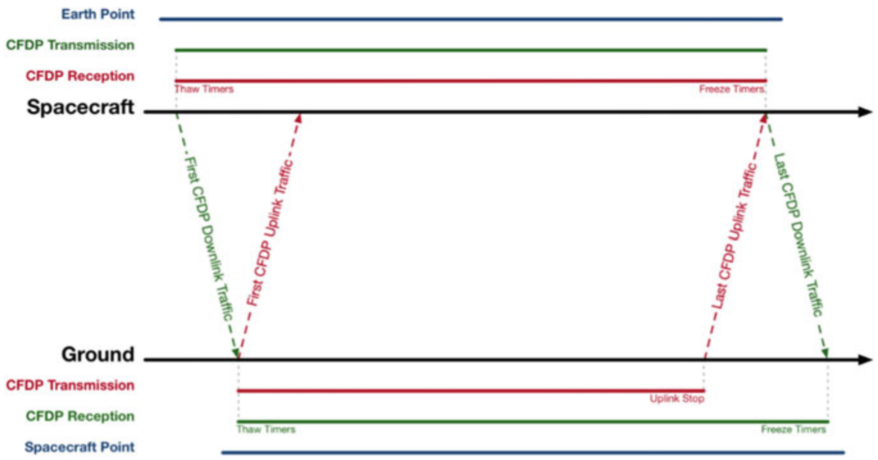


Fig. 10 Pattern of CFDP timer freezes and thaws

will keep sending the EOF indefinitely until the ground intervenes. This preserves the state of the transaction, and the possibility to retransmit only the data that was lost. Fortunately, Europa Clipper has enough on-board data storage to make this strategy viable. Similar approaches were taken with the fault responses for other timers, and for fault conditions unrelated to timers. The rule is to warn the ground, but keep the transaction alive.

4.2.4 CFDP During Solar Conjunction

There are other nominal cases during the mission in which CFDP must be managed to keep downlink efficient. About every 13 months, Jupiter, and therefore the Europa Clipper spacecraft, enters solar conjunction, where it passes close to the Sun as seen from Earth. At such times, telecommunications are disrupted. Europa Clipper uses Ka-band for downlink and X-band for uplink. Ka-band performs better near the Sun than does X-band. When the angle between the spacecraft and the Sun is greater than 3°, both Ka-band and X-band perform acceptably. Between 1° and 3°, X-band is unreliable due to solar scintillation effects on the signal, but Ka-band still works; within 1° neither Ka nor X-band can be relied upon. See Fig. 11 for details. A typical solar conjunction lasts seven or eight days. The first two or three days have downlink communications only, then two or three days of no communication and followed by two or three days of downlink only again. This has several implications for CFDP.

For efficiency, it is desired to utilize the time between 1° and 3° to downlink data, and to continue the use of CFDP class 2 so that missing data is automatically retransmitted. But, within 3° of the Sun, uplink to the spacecraft is not possible, and therefore the ground cannot request retransmission of data or close out downlink CFDP transactions (Fig. 12).

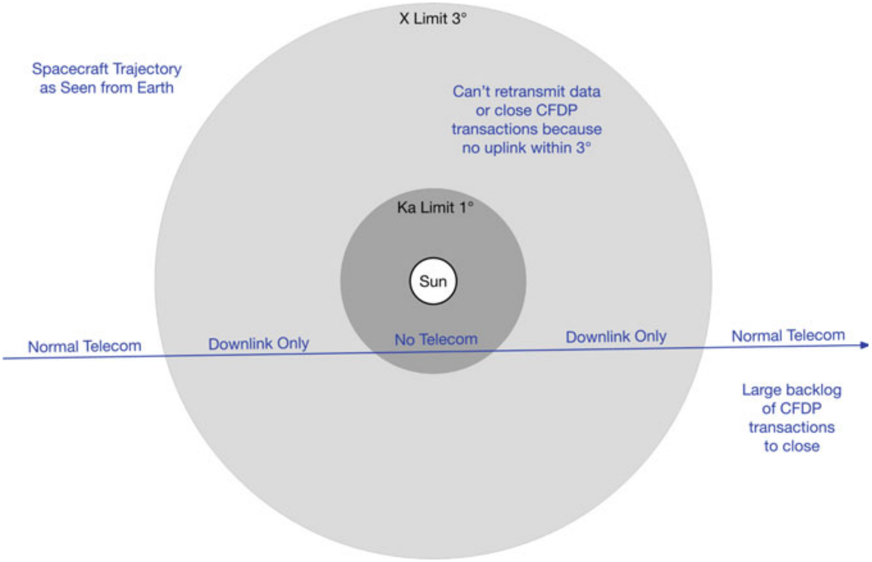

X Limit 3°
Spacecraft Trajectory as Seen from Earth
Can't retransmit data or close CFDP transactions because no uplink within 3°
Ka Limit 1°
Sun
Normal Telecom
Downlink Only
No Telecom
Downlink Only
Normal Telecom
Large backlog of CFDP transactions to close

Fig. 11 Solar conjunction geometry

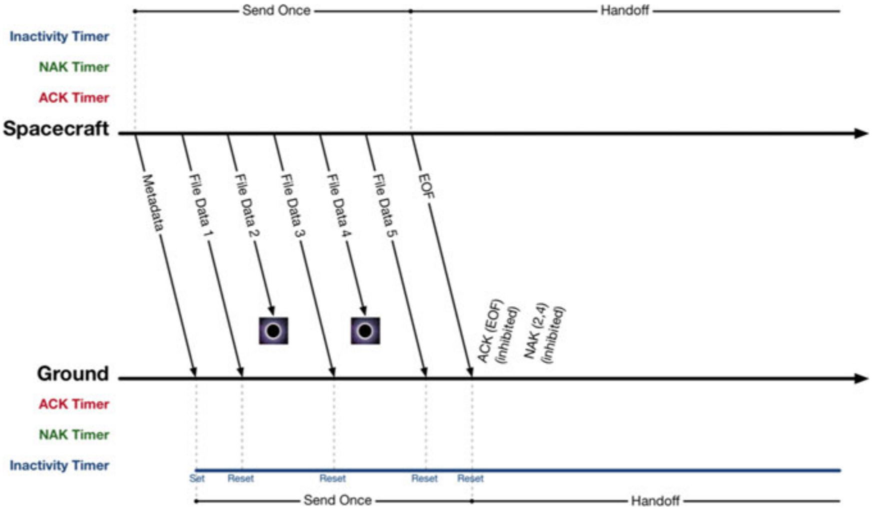


Fig. 12 Single CFDP downlink transaction, with retransmitted data, between 1° and 3° of the Sun

The send once phase happens the same as normally, compare to Fig. 9. But, because the ground cannot transmit to the spacecraft, the ACK (EOF) and NAK (2, 4) are inhibited. After solar conjunction is over, all the previously inhibited ACKs, NAKs and FINs are transmitted to the spacecraft, any missing data is retransmitted and the transactions are closed normally as in Fig. 9. The only timer that runs during

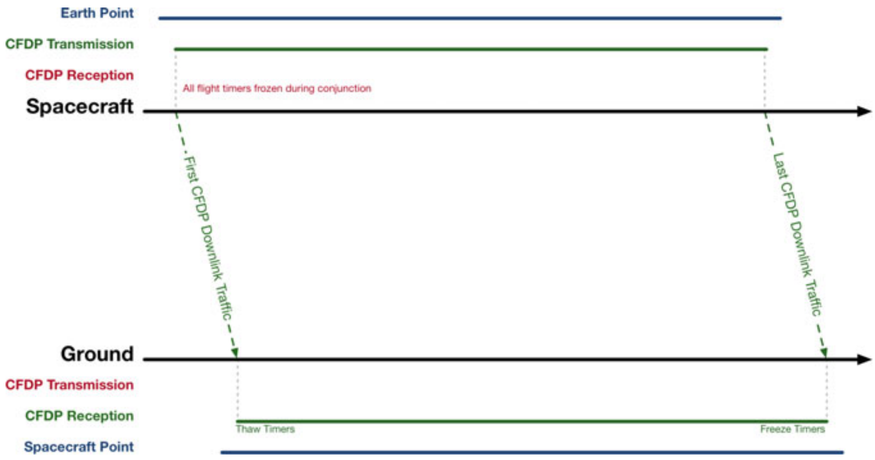


Fig. 13 Pattern of CFDP timer freezes and thaws between 1° and 3° of the Sun

solar conjunction is the ground inactivity timer. The ground inactivity timer will be lengthened so that it gives warning less frequently during solar conjunction. The ACK and NAK timers do not run because the ground cannot send any ACKs, NAKs or FINs. The spacecraft timers do not run at all.

Figure 13 shows the pattern of CFDP timer freezes and thaws for a telecommunication pass during solar conjunction. Only timers on the ground run.

The other implication of using class 2 CFDP during solar conjunction is that the number of open transactions becomes much larger. Normally, there might be one or two hundred open transactions at any given time. During solar conjunction, the number approaches 1,500. During conjunction, transactions are opened at the normal rate, but none are closed. The spacecraft needs some amount of buffer space for each open class 2 transaction. Since memory is statically allocated, the number of allowable open class 2 CFDP transactions must be chosen before launch. The allowable number was increased to 3,000 in order to accommodate solar conjunction with sufficient margin.

If the limit of 3,000 is exceeded, the spacecraft will abandon the oldest transactions to allow buffer space for new transactions. Only the metadata about the transaction is lost; the file contents are preserved on board. However, it will be possible after solar conjunction to resurrect the transaction and pick up where it left off. Even if a transaction is abandoned by the spacecraft, the full state is retained by the ground. After solar conjunction, the ground will resume by sending an ACK (EOF), NAK or FIN as appropriate. We have chosen to encode information into our downlink transaction identifiers such that the spacecraft can determine which file is associated with a transaction just based on the identifier. This allows the spacecraft to resume a transaction, even after it has been abandoned.

All of the above measures help operability (*tractability*) in the sense that they minimize changes to CFDP operation during solar conjunction. The operations team should not have to do anything special during conjunction, other than perhaps changing a few timer settings once at the beginning and once at the end.

4.2.5 CFDP During Inner Cruise

During most of inner cruise, the HGA is used as a thermal shield, necessitating the use of the FBAs, with their lower downlink data rates. All the engineering data that is collected cannot be downlinked as it is generated, leading to a backlog of engineering data on-board. Every year or two during inner cruise, the spacecraft is at opposition to the Sun. During those brief times, the HGA can be used to downlink the backlogged data at high rates. A similar situation occurs when the spacecraft moves beyond two AU from the Sun. Beyond that limit, it can point the HGA at Earth.

However, this causes a problem. Because engineering data files tend to be small, there will be many of them in the backlog. If these can't be closed fast enough, due to round-trip light time, the limit of 3,000 open class 2 CFDP transactions may be exceeded. Also, the engineering file system (EFS) is limited to about 106 Gibits of storage. As explained above, this is to be avoided since this creates extra work for the operations team. Figure 14 illustrates the problem. The red line indicates the number of gigabits of backlogged engineering data, assuming no active management to prevent exceeding the EFS storage capacity. There are five points during the inner cruise when high data rates are available and a large portion of data (and hence a large number of files) are cleared.

There are several possible ways to mitigate this risk of exceeding 3,000 open class 2 CFDP transactions. The same amount of data could be stored in fewer files (i.e.,

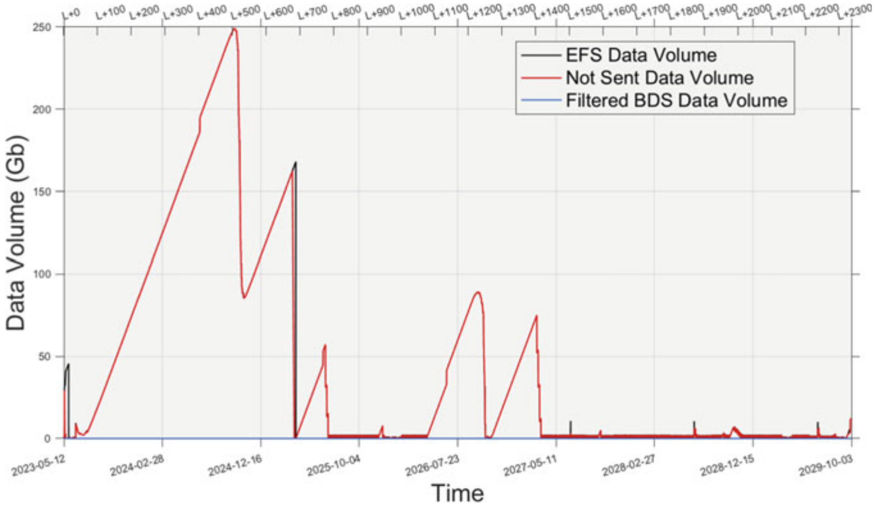


Fig. 14 Backlog of engineering data on the spacecraft during inner cruise (without active management)

make the files larger). This would reduce the number of simultaneous transactions. The downlink data rate could be lowered, and the duration of the telecommunication passes could be increased. That would enable downlink of the same amount of data. Then each file would take more time, and therefore reduce the number of simultaneous transactions. If some combination of the above does not mitigate the risk sufficiently, then the operations team might choose to manually delete some data (that is, never send it to the ground).

All of the above mitigations are intended to minimize the extra workload and operational complexity during day-to-day operations. From an operability perspective, it is desirable to spend more time in development and planning, to help reduce potential operational complexities.

4.2.6 Uplink CFDP During Tour

The spacecraft has a limit of 100 concurrently open uplink CFDP transactions. During tour, we estimate that we may need to uplink close to that number of files, or more, for each background sequence, once every four weeks. An uplink CFDP transaction is really just a mirror image of the downlink transaction (Fig. 15).

One approach to this limit would be to group the files into groups of 100 or less. After each group is uplinked, operations would wait to receive confirmation from the spacecraft that all the transactions had been closed before uplinking the next group of files (Fig. 16). This works, but can lead to inefficiencies of time. Alternately, given that the telecom link for uplinking files is much more reliable than that for downlink,

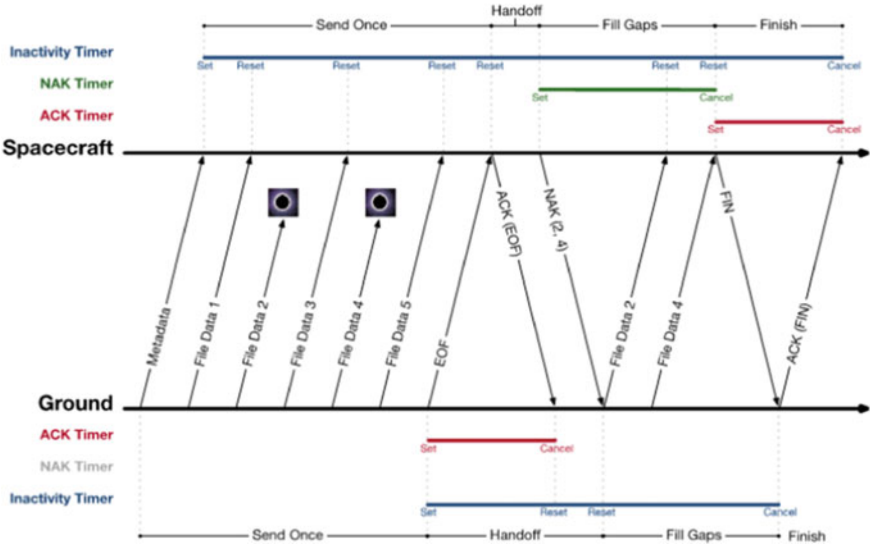


Fig. 15 Single CFDP uplink transaction, with retransmitted data

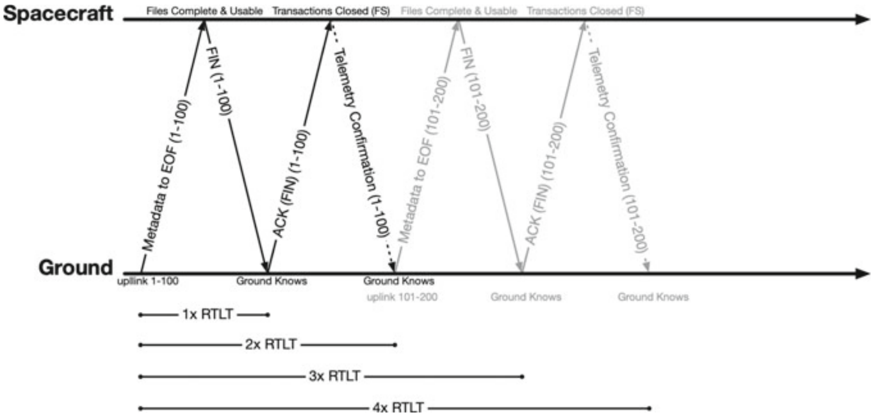


Fig. 16 Grouping of uplink files

we may compress the timeline somewhat and not wait for telemetry confirmation on the ground. This would save roughly one round-trip light time compared to the above. A third possibility would be even more efficient. Given the size and order of the files to be uplinked, it is possible to predict the number of open transactions over time. Assuming no retransmissions, each transaction is open for one round-trip light time. Given that information, we can insert pauses of minimal length to keep the transaction count below the limit of one hundred.

Given that the telecom link for uplinking files is much more reliable than that for downlink, we may compress the timeline somewhat and not wait for telemetry confirmation on the ground. This would save roughly one round-trip light time compared to the above (Fig. 17).

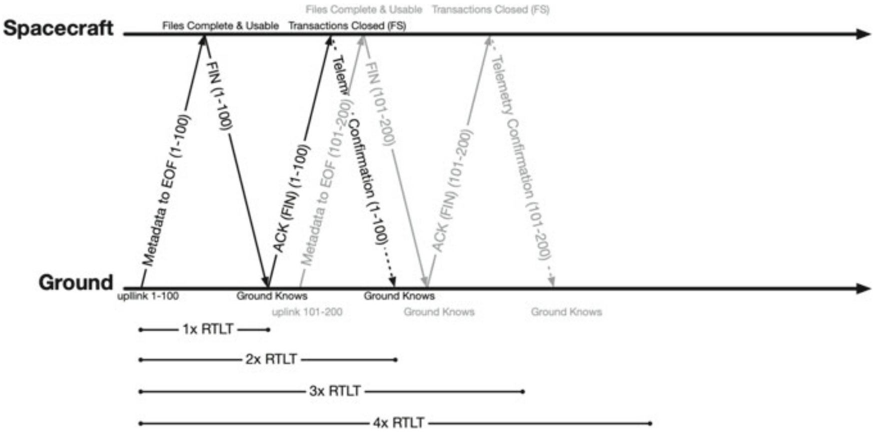


Fig. 17 Compressed grouping of uplink files

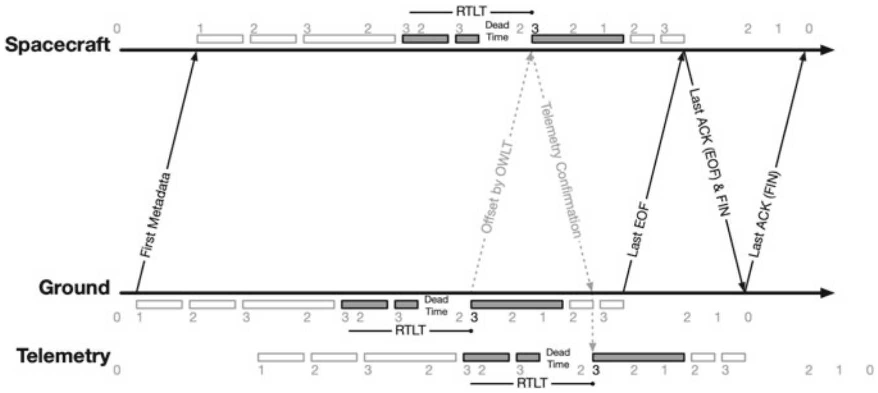

RTLT
Spacecraft
0 1 2 3 2 3 2 3 Dead Time 2 3 2 1 2 3 2 1 0
First Metadata
Offset by OWLT
Telemetry Confirmation
Last EOF
Last ACK (EOF) & FIN
Last ACK (FIN)
Ground
0 1 2 3 2 3 2 3 Dead Time 2 3 2 1 2 3 2 1 0
RTLT
Telemetry
0 1 2 3 2 3 2 3 Dead Time 2 3 2 1 2 3 2 1 0
RTLT

Fig. 18 Minimal uplink grouping

A third possibility would be even more efficient. Given the size and order of the files to be uplinked, it is possible to predict the number of open transactions over time. Assuming no retransmissions, each transaction is open for one round-trip light time. Given that information, we can insert pauses of minimal length to keep the transaction count below the limit of one hundred.

Figure 18 shows an example. For simplicity, the number of uplink transactions is limited to three, rather than one hundred. Each transaction is open for one round-trip light time. So, the number of transactions open at any one time can be found by summing the number of files uplinked within the past round-trip light time. Three transactions are shaded to illustrate. The number of open transactions is shown as it is on the ground, as it is on the spacecraft and as it is seen in telemetry on the ground. The profile is the same, just offset by the light time. Operationally, we never start uplinking a new file unless the number of open transactions is less than three. If it is equal to three, we wait by inserting dead time until the number drops to two.

It is also possible to optimize the order of file uplink to reduce or eliminate dead time. For example, by evenly distributing the smaller files among the larger ones, it may be possible to avoid dead time altogether. This assumes that all the files are of equal priority, and there is no operational need to uplink some files before others.

4.2.7 Efficiencies Gained by Using CFDP for Uplink

CFDP enables some uplink *efficiencies* that were not possible with the legacy uplink system. First, uplink of individual files may span telecom sessions. In the legacy uplink system this was not possible; we had to make sure that the last planned file was complete before the end of a telecom session. For particularly large files, like flight software uploads, we had to break them into smaller files and then reassemble them into the original large file on board the spacecraft. This capability allows us to make more efficient use of the time available for uplink by allowing us to uplink all the way to the end of the telecom session regardless of file boundaries.

Another benefit is that we can uplink files concurrently. We have designed our ground software to have multiple parallel uplink queues. One queue might be used for the files for a background sequence. Files in that queue would be uplinked in order, one at a time. The second queue would be available for higher priority uplinks like a maneuver sequence or an ephemeris update. When there are files in both queues, the ground will alternate uplinking data from the queues in a round-robin fashion. These capabilities remove constraints from planning the uplink of files, and reduce operational complexity.

4.3 Post-Launch Visibility

4.3.1 Europa Clipper Telecom Configuration

The Preliminary Design Review (PDR) telecom design of Europa Clipper included: (1) a 3 m high gain antenna (HGA) boresited along the −Y axis, for high-rate downlink of science and engineering data during the Tour; (2) a medium gain antenna (MGA) co-aligned with the HGA, to provide low-rate telemetry during certain safing situations (allows for relaxed pointing constraints); (3) three fanbeam antennas (FBA) in the −Y/+Z, −Y/−Z, and +Y/−Z directions, to provide roughly spherical coverage, to provide telemetry during inner cruise (and later for gravity science measurements during Tour); and (4) two low gain antennas (LGA) in the +Y/−Z and –Y/+Z directions, to provide telemetry near-Earth and during portions of Inner Cruise. These antenna locations and fields of view (FOV) are shown in Fig. 19.

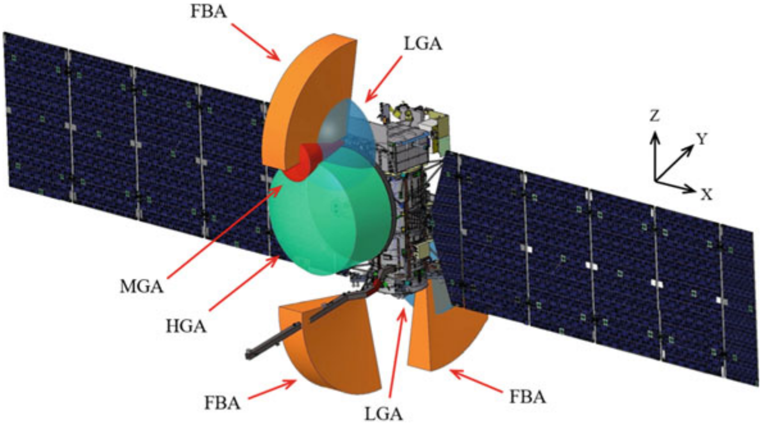


Fig. 19 Europa Clipper preliminary design review telecom antenna configuration and fields-of-view

The two LGA antennas were originally intended to provide telemetry coverage for the Inner Cruise phase of the mission. The broadband LGAs ideally provide coverage of the hemisphere at which they are pointed.

4.3.2 The Post-Launch Visibility Challenge

During post-PDR development and analysis, it was determined that given the Sun-probe-Earth (SPE)/Sun-Earth-probe (SEP) angles of near 90° for several days post launch, and the need to orient the spacecraft with the Z-axis perpendicular to the Sun line (−Y to Sun), the spacecraft would be generally −Z pointed to Earth (within a cone ±20°). As a result, the two original LGA antennas (pointed in the −Y and +Y directions), would be ~90° from the Earth-point line. Figure 20 shows a line-of-sight analysis between the −Y spacecraft axis and the Canberra Deep Space Network (DSN) complex during an example initial acquisition.

The view to Canberra for these antennas oscillates about 90° from either boresight, for the first 40 min or so, and then settles down at almost exactly 90°. At this 90° orientation, downlink signal would fall in the extremes of both LGA antennas. In the regions beyond ~80° from boresight, the spacecraft could introduce significant pattern distortions via spacecraft scattering, which could increase the difficulty in acquiring and maintaining lock. As a result, the downlink telemetry signal could be intermittent at best.

From an operability perspective, this is not desirable. This decreases or eliminates the operations team's *visibility* and situational awareness into spacecraft configuration, health and status, and performance at a critical time in the mission. This decreased *visibility* detracts from their ability to detect anomalous behavior and trends and diagnose the root cause of problems, thus decreasing the operations team's ability to take prompt corrective action.

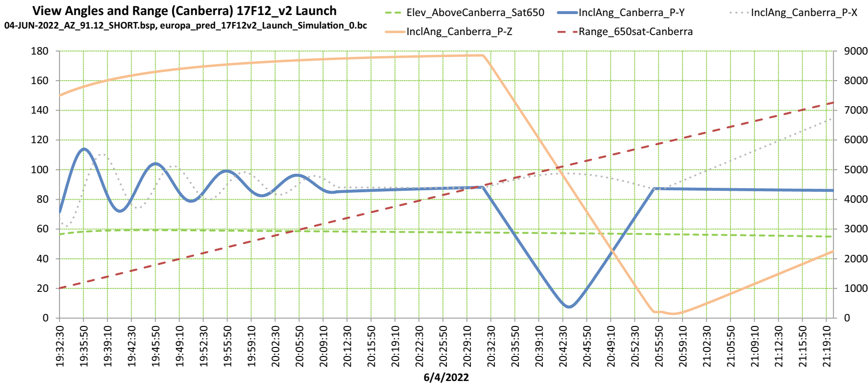


Fig. 20 Example post-launch line of sight between Europa Clipper and Canberra DSN complex

Post launch vehicle separation, the spacecraft enters a 0.1 RPM thermal roll to equalize and thermally condition the solar array hold down release mechanisms prior to solar array deployment. This barbecue roll will cause the LGA that is pointed closest to Earth to swap every 5 min. Even if viewing angles with the ±Y LGAs were sufficient to provide a reliable link, this roll would require the operations team to swap between the two antennas based on line-of-sight to the spacecraft. This decreases the *tractability* aspect of operability, requiring increased operations team interaction with the spacecraft at a critical time during the mission (the launch phase). The *commandability* of the spacecraft should not be affected by the PDR LGA configuration. The short spacecraft-to-DSN range should provide a large uplink margin, even at far off-boresight angles.

In addition, there is a JPL Design Principle (DP) that must be complied with, regarding visibility during critical events (e.g., launch, JOI) [5]. Any gaps in downlink coverage during critical events would violate this design principle. Therefore, the PDR designed LGA configuration did not appear to be very compatible with the post-launch/initial acquisition spacecraft orientation and attitude profile.

4.3.3 Low Gain Antenna Trade

Once this reduced post-launch visibility was uncovered, the Europa Clipper project initiated a trade study in the second half of 2018 to determine the best resolution. The following solutions were examined: (1) keeping the current design; (2) re-orienting the existing LGAs 45° to the ±Z directions; and (3) adding a third LGA.

In the end the decision was made to implement Option 3, adding an additional, −Z facing, LGA. Adding a 3rd LGA would provide reliable communications with less impact in terms of cost and design complexity. This 3rd LGA antenna: improves *visibility* during post-launch, initial acquisition phase, providing reliable communications during launch activities; allows the ops team to receive telemetry after launch and eliminates risk of drop-outs during this critical event; and enables the spacecraft to meet JPL Design Principles.

4.4 Science Data Accountability

Europa Clipper's science instruments acquire the majority of their data during the few hours around each closest approach of Europa. The presence of multiple imaging instruments and a sounding radar requires the data system to handle very high rates of data transfer from the instruments, most of which do not buffer data internally. Data are streamed from the instruments into the bulk data store (BDS) to await downlink. Due to mass, power, and other constraints (such as extremely high radiation dose and rate), the BDS is unable to sort the multiple high-rate input data streams in real time and create traditional data files for each instrument. A firmware solution produces fixed 12 MiB sized BDS files which store the science data. The instruments cannot control what data goes into which files, and there is no directory hierarchy. The BDS is not a full-fledged file system. The resulting BDS files are an amalgamation of data from the instruments, and provide limited traceability to specific science observations. This limited *visibility* into file content complicates the operations team's need to downlink higher priority observations first, as they cannot easily identify in which BDS files they are contained.

Europa Clipper has attempted to simplify the data accountability function by adopting a round-trip, end-to-end system of identifiers to track activities during the planning and sequencing process, and link those activities to the data collected onboard the spacecraft as a result of their execution. The path taken by the data from the time it is collected by the science instruments to when it is delivered back to the science teams is complex. It involves many steps and transformations. Data identifiers can be used as a common thread to help track the data through these many steps.

The basic unit of accountability for Europa Clipper science data is called the accountable data product (ADP). Each ADP has a single unique identifier called an accountability identifier (AID). AIDs are never reused, so the mapping from ADPs to AIDs is guaranteed to be one-to-one. The AID is simply a 32 bit number that is attached to activities, commands, telemetry packets, and science data products. Figure 21 shows the end-to-end life cycle of ADPs.

Science planning starts with each science team choosing a set of planned activities. Each activity is tagged with one or more AIDs. The activities are then translated into commands that are uplinked, and executed by the spacecraft. The AIDs follow the activities to the commands. When executed by the instruments on board the spacecraft, the resulting telemetry packets are also tagged with the same AID. After those telemetry packets are sent to the ground, they are processed, packaged and delivered to the science team as data products according to their AID. The instruments also produce metadata packets that describe the exact number of packets produced.

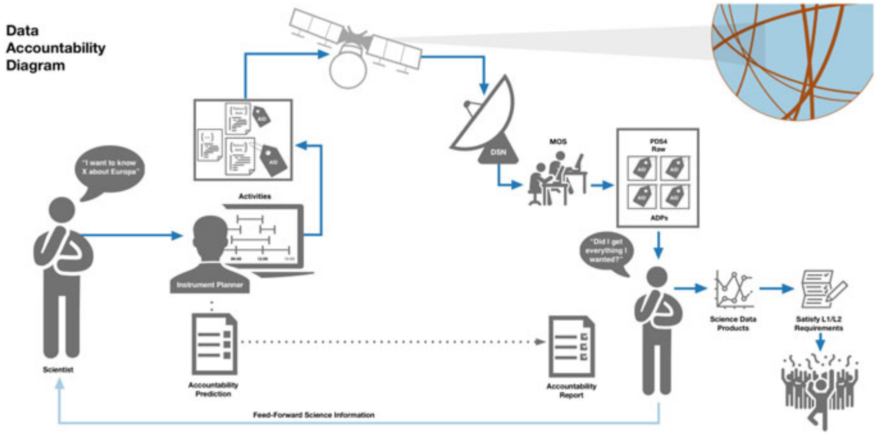

Data Accountability Diagram
"I want to know X about Europa"
Scientist
AID
AID
Activities
Instrument Planner
DSN
MOS
PDS4 Raw
AID
ADPs
"Did I get everything I wanted?"
Science Data Products
Satisfy L1/L2 Requirements
Accountability Prediction
Accountability Report
Feed-Forward Science Information

Fig. 21 Europa Clipper science data accountability overview

If the BDS was a traditional file system, the instruments could have made a file on the BDS for each ADP. That would have made tracking the progress of the data through the downlink system much easier. The ADP metadata packets would not have been necessary at all in this case. In effect, the ADP helps compensate for some of the hardware limitations on the spacecraft. It serves as a virtual file that can be tracked in a similar way as if the instruments could control which data goes into which BDS file.

One of the main benefits of this end-to-end system of accountability is automation. Routine accountability reports can be generated automatically. This gives mission operations personnel easy *visibility* into the data pipeline, and gives early notice of problems. Rather than reactively waiting for reports of problems from the science teams, the mission operators can be ahead of the game and address data problems proactively. Because the same identifiers are used from planning through data delivery, it makes the alignment of planned activities with data, and the demonstration of completion of higher-level science goals, easier.

4.5 Logic-Based Flight Rules System

Europa Clipper is attempting to initiate a logic-based flight rules system as an opportunity to improve the operability of flight rules development, implementation, maintenance, and usage.

4.5.1 Flight Rules Defined

Flight rules represent prohibited spacecraft activities, configurations, or behaviors, and are essential to protecting the health and safety of the flight system or science data return. For example, a flight rule may state that a battery must always have more than thirty percent state of charge, or that the spacecraft should not point at the Sun before the camera lenses are covered. Associated with each flight rule is rationale as to why the flight rule exists, and identifies the potential consequences to the flight system if the rule is violated. Subsystem experts design and author flight rules for each subsystem (e.g., power; guidance, navigation, and control; an instrument such as a magnetometer). These flight rules are later implemented in ground tools that are used to simulate and plan mission activities, and can be integrated with software that checks if any flight rules have been violated.

4.5.2 Operability Challenges with the Current Flight Rules Implementation

Historically, the majority of past flight projects have had subsystem experts author flight rules in a natural language, English. These English-based flight rules are typically stored in an online database or file system, and are later accessed and coded by ground tool developers. However, due to the imprecise nature of natural languages, the intent of the flight rule is often ambiguous, and can also contain an author's implicit but unexpressed assumptions. As a result, there has been a heavy reliance on interpersonal interaction between the subsystem's flight rule authors and the ground tool developers to communicate the precise intent of the rule and translate this intent into mission planning and flight rule checking software. This can be a time-intensive and *inefficient* process.

Given the above flight rule development process, the efficacy of the *testability* of flight rules can be questioned. The correctness of test cases that are created to validate and verify flight rules are dependent on the interpretation of a given flight rule.

Due to the complicated nature of flight rules, it is often determined that a significant subset of flight rules should be checked manually. For example, the Mars Reconnaissance Orbiter has 198 flight rules, all authored in English, of which only 56 are checked automatically by ground tools. Cassini had ~170 flight rules, again all authored in English, of which only half were checked automatically. It is estimated that Europa Clipper will have >300 flight rules. For *efficiency* and *robustness*, it is desired to have all flight rules checked automatically on Europa Clipper.

Experience has shown that flight rules, since they may be difficult to accurately capture in code, may not be updated or corrected as errors are found or if the rule changes. This can lead to many flight rules falsely being triggered as "violated" each time the FR check is performed. The operations team then gets in the pattern of ignoring these nuisance violations, leading to decreased *tractability.*

4.5.3 A Logic-Based Flight Rule Implementation

Due to the deficiencies of the natural language based approach to authoring flight rules, it may be advantageous to identify an expressive, intuitive, non-programmatic solution to author and check flight rules. Programmatic solutions such as using an object-oriented programming language to express flight rules have been dismissed because flight rule authors are subsystem experts, and are typically not trained computer programmers. To this end, we present a first order and temporal logic-based solution to express flight rules without ambiguity, and whose programmatic implementation can be automated. This solution provides a finite, contained set of intuitive logical tools to represent flight rules and does not require flight rule authors to have programming experience.

The contained set of logical tools that can be used to represent flight rules consist of *logical predicates* and *variables* that represent spacecraft events or values. Logical predicates take in variables as inputs and output a Boolean result, and can be thought of as analogous to a Boolean method in Java. Temporal event variables have properties which represent the start time of the event and the end time of the event. Variables can also represent spacecraft values, also known as states, such as battery state of charge. A *condition* is any Boolean evaluable statement that contains a variable. An example of a predicate in this logic-based solution is the *Before* predicate, which compares two conditions or a condition to an event and determines if the first condition ends before the second condition starts, or if the condition ends before the event starts. By nesting atomic predicates and combining predicates with logical connectors, such as *and, or, or not*, complex flight rules can be represented in logic.

For example, a simple hypothetical flight rule may state "do not image a mosaic until the camera cover is off." In this case, assume the variable that represents the camera cover state is *camera_cover_state* and the event that represents a mosaic is *image_mosaic.* Let the convention used be that if the flight rule is true, then it is violated. The *condition* that is being checked is $camera\_cover\_state = off$. The logic based equivalent of this natural language flight rule is:

$$Before(image\_mosaic,\ camera\_cover\_state = on)$$

The explicit definition of Before removes any vagueness from this statement. The designs of the temporal predicates in this solution, including Before, were largely based on Allen's Interval Algebra [6]. The removal of ambiguity increases the *robustness* of this system. By eliminating the recurring discussions to clarify flight rule intent between developers and flight rule authors, the *efficiency* is also increased.

A more complex hypothetical flight rule might state "the camera cover must be off at least ten minutes prior to imaging a mosaic." Let the event that represents the camera calibration activity be *camera_calibration.* The Before predicate can be overloaded to capture this rule:

$$Before(image\_mosaic,\ camera\_lens\_state = on,\ \text{“}<\text{”}, time = 10)$$

This Before predicate will be evaluated to true if the camera lens state is on within ten minutes of the mosaic being imaged. In the previous rule, the Before predicate would have only have evaluated to true if the camera cover state was on in the time step prior to the image mosaic starting. This rule can be complicated further by saying that "the camera cover must be off at least ten minutes prior to any imaging event." In this case, let the different imaging events be represented by the following: *image_mosaic, narrow_angle_image*, *wide_angle_image.* Instead of repeating the Before predicate syntax for each event and using an OR to join the clauses, we can define an OR_GROUP: OR_GROUP(name = "imaging_events", image_mosaic, narrow_angle_image, wide_angle_image). The rule now becomes:

$$Before(\text{“}imaging\_events\text{”},\ camera\_cover\_state = on,\ \text{“}<\text{”}, time = 10)$$

As the examples get more complicated, we begin to rely more heavily on the logic based specification to assist in writing rules intuitively.

While the initial work to define the set of logical tools has been completed, future work remains in order to realize the complete potential of logic-based flight rules, which would further increase operability. For example, adopting a logic-based approach may seem like a daunting task at first, especially for those who are unfamiliar with writing in logic. However, this logic-based approach lends itself to creating simple front-end tools, or GUIs, to facilitate the construction of logic-based flight rules. Customizable third-party visual programming tools, such as Blockly by Google [7], can also be harnessed to design a highly intuitive experience, further increasing the *efficiency* of this system.

This logic-based approach also lays the groundwork for automating the programmatic implementation, checking, verification, and validation of flight rules. The objects created by the front-end authorship tool can be turned into programmable objects automatically, as is done by Blockly. These programmable objects can be distributed to and integrated with mission planning and flight rule checking ground tools. In order for this end-to-end approach to succeed, variables that represent spacecraft events and values must be stored in a central location, and the equations used to calculate these variables in modeling software must be transparent to anyone who works with flight rules. To this end, parallel work on the development of spacecraft state dictionaries is underway.

5 Conclusions

1. Operability has had a positive influence on design decisions on Europa Clipper, although cost, schedule, budget, heritage, and other technical concerns have, at times, outweighed operability concerns. The Europa Clipper mission experience to date demonstrates that adopting operability has resulted in more thorough, balanced consideration of the effect of early design trades and decisions on the operations phase of a mission than seen in many previous missions, and provides operations development insight into prioritizing work to go.
2. As the system design has matured, deeper implications of design decisions have come to light. It has been said "The devil is in the details." Some design decisions (e.g., the integrated wing assembly) will be mitigated mostly by operational constraints, activity designs, ground tool development, and operations team focus and attention. Other design decision consequences have been successfully mitigated by hardware and software solutions: poor post-launch visibility led to the addition of a 3rd LGA; and large science data volume and its structure led to the implementation of CFDP class 2 and the introduction of the accountability identifier.
3. Opportunities have also been taken to improve on past designs and performance, as demonstrated in the adoption of a logic-based automated flight rule development and checking system.

4. The project's high-fidelity mission-level modeling and simulation capability, developed early in the pre-project phase, has been essential in rapidly assessing the impacts of proposed design decisions on mission operations.

Acknowledgements This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

1. Signorelli J, Bindschadler DL, Schimmels KA, Huh SM (2018) Operability engineering for the Europa Clipper mission: formulation phase results and lessons. In: AIAA 15th international conference on space operations, May 2018
2. "Europa Clipper Mission Plan," JPL D-55485, Oct 2019
3. Ferguson E, Wissler S, Bradley B, Maldague P, Ludwinski J, Lawler C (2018) Improving spacecraft design and operability for Europa Clipper through high-fidelity mission-level modeling and simulation. In: AIAA 15th international conference on space operations, May 2018
4. CCSDS File Delivery Protocol (CFDP), CCSDS 727.0-B-4, Blue Book January 2007
5. JPL Design, Verification/Validation & Ops Principles for Flight Systems (Design Principles), Rev. 6. JPL D-43913, Oct. 2012
6. Allen JF (1983) Maintaining knowledge about temporal intervals. University of Rochester. Commun ACM 26(11). [Association for Computing Machinery]
7. Blockly (2020) A JavaScript library for building visual programming editors https://developers.google.com/blockly/. Accessed Feb 2020

Fast Retargetable Goals Driven Approach to Deal with Plan Failures of Spacecraft

Rui Xu, Chao Chen, Zhaoyu Li, Shengying Zhu, and Zixuan Liang

Abstract Plan repair is more preferred over replanning when the agent suffers from plan failures. Most of the existing research work on autonomous plan repair regards the planned action as an instantaneous point, preventing it from being directly applied to the field of spacecraft operation, where is full of concurrent actions with varying duration and resource consumption. In this chapter, a reactive rapid autonomous plan repair algorithm based on retargetable goals, Retargetable Goals Plan Repair Method (ReGPR), is proposed. In ReGPR, a mechanism for transforming a mission plan into a state queue and a method of determining the optimal states, i.e., retargetable goals, based on evaluation are proposed. For transformation, ReGPR discretizes the concurrent actions into two state nodes distributed at their beginnings and ends, which encapsulates much information such as logic, numeric, and duration. Then, the plan repair problem is transformed into the puzzle of state transition by mapping them into the same timeline, which forms the retargetable goals. To determine the optimal goal for recovery, an evaluation criterion including goal reachability and search time estimation is designed. With the help of evaluation, ReGPR gets rid of searching for the recovery plan by the try-error method and finds the possible solution quickly. Several experiments with either logic failures or energy shortage or both were done in the modified Satellite Complex domain to show the performance of ReGPR by comparing it with the corresponding results of the replanning methods and other plan repair methods. And the results desmonstrate that in the repair progress, ReGPR explores fewer state nodes in no more than one round, and its advantage is greater when the plan repair problem becomes more complex.

Keywords Plan failure · Plan repair · Reactive plan · Retargetable goal · Spacecraft plan

R. Xu (✉) · C. Chen · Z. Li · S. Zhu · Z. Liang
School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, P.R. China
e-mail: xurui@bit.edu.cn

Key Laboratory of Autonomous Navigation and Control for Deep Space Exploration, Ministry of Industry and Information Technology, Beijing 100081, P.R. China

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_28

Nomenclature

O	Operator
a	Action
e	Happening
pre_x	Precondition of an element x
eff_x	Effect of an element x
eff_x^+	Add effect of an element x
eff_x^-	Delete effect of an element x
n_x	Name of an element x
p_x	Parameter of an element x
d_x	Duration of an element x
t_x	Occurrence time of an element x
P	Plan
E	Event
Π	Plan repair problem
F	Propositional set
V	Bounded variable set
O	Operator set
I_Π	Initial state of Π
G_P	Goal state of P
S	State of a spacecraft at a certain time
Γ	Regression function
E_S	Evaluation value of S
δ	Goal reachability
t_r	Time for plan repair
Δt	Maximum repair time allowed
h	Heuristic distance between two states
d	Average expansion delay number between the generation and expansion of a state node
$\bar{t}$	Average expansion time of each search node
ReGPR	Retargetable goals plan repair method
TRPG	Temporal relaxed plan graph

1 Introduction

Before the spacecraft starts performing its mission, there is always an operation plan to guide what it should do at the specified time. However, when the spacecraft fails to execute the mission plan in the dynamic and uncertain space environment, the ground staff can hardly be aware of that problem in a short timescale and upload a solution quickly, due to the visibility or long-distance communication between the spacecraft and the ground [1]. For example, because of the long delay, the European Space

Agency team could do little in the strong real-time mission of landing but just waited for the bad news that Philae finally landed in the shadow site after bouncing several times on the surface of the comet 67P, which results in the loss of opportunities to gather more scientific data [2]. Therefore, there is a great need for an on-board autonomous operation to deal with plan failures, so as to enhance the robustness of spacecraft to uncertainty and increase science return [3–5].

Plan repair provides such a possible way to enhance onboard autonomy, which resumes plan execution by utilizing both reactive acting and proactive reasoning on the ongoing plan. In contrast to replanning that derives a brand-new plan from scratch with plan failures, plan repair makes up for the original plan to retain most of the previous efforts [6]. Much research work is inclined to the fact that plan repair is more effective [7–9], though it is theoretically proven that plan repair shares the same complexity with replanning in general, and sometimes could be more difficult [10].

Plan repair results from the uncertain environment, which makes the assumptions of the previously considered plan deviate from reality [1]. According to the characteristics of repair, most of the existing plan repair methods can be divided into the following five categories [11]: (1) Rule matching method [12], which resumes plan execution according to the error fix rules formulated in advance; (2) Local adjustment method [13], which uses the plan result library to find feasible fragments to replace the failed parts in the plan; (3) Unrefinement and refinement method [14], which first removes failed actions in the plan and then searches a recovery plan to make up it; (4) State transition method [15], which skips non-executable actions and finds alternative actions in the state space; (5) New problem construction method [16], which extracts key elements, e.g., groups of activists and constraints, from the failed plan to construct an enhanced planning model, so as to use the existing planner to solve the problem.

The middle three categories of the above methods are widely studied in the plan repair community [17–19]. And the challenge lies in the trade-off between the deletion of non-executable actions in the failed plan and the addition of available actions, in order to modify the original plan as little as possible [20]. As a result, most of the plan repair methods use the try-error manner to resume execution, i.e., trying to add actions after deleting some actions to fix the failed plan, and if it fails, more actions are removed from the plan in turn. For example, van der Krogt [14] built a structure called a removal tree with a depth from 1 to calculate possible infeasible actions. After stripping the removal tree from the failed plan, the remaining parts are sorted by the planning heuristic and refined in turn. If the solution is unfortunately not found during this process, the depth of the removal tree increases until a recovery plan is found, or the whole plan is traversed.

However, the try-error manner could be inefficient. On the one hand, the process of plan repair is blind because that all actions are treated equally, so it is impossible to accurately judge invalid actions. If the recovery plan is implicit at the tail of the plan, all previous actions have to be deleted in turn in this manner until the actions available for plan repair are exposed, which inevitably resulting in the increase of search times and total solution time. On the other hand, the impact of the failure on the original mission goal is ignored. For example, when the rover can see the target rock within its field of view thanks to a blast of wind, it does not need to move forward and then turn to the rock as planned.

To overcome the above inefficiencies, in this chapter, a retargetable goals-driven plan repair method called ReGPR for spacecraft is proposed, whose actions can be durable, concurrent, and resource dependent. In ReGPR, the failed plan is transformed into a state queue by mapping the beginning and end of different actions into the same timeline. For the state in the queue, it contains the expected elements at that time instant in the plan, which is composed of logic, energy level, execution time, and so on, and provides alternative objectives (retargetable goals) for plan repair. And an expression including goal reachability and search time estimation is designed to evaluate the optimal one among the retargetable goals, which is expected to hit the right recovery plan with the minimum cost.

This chapter organizes as follows. Section 2 reviews the definition of the plan repair problem to prepare for the following parts. Section 3 explains how to transform the plan repair problem into the path-finding problem and to get the retargetable goals, and overviews the proposed method. And Sect. 4 describes in detail the evaluation method of the founded retargetable goals. Then Sect. 5 sets up the experiment based on the modified Satellite Complex domain and analyzes results of that. And the last section summarizes this chapter.

2 Preliminaries

Like planning and replanning, plan repair is also a reasoning process about a course of action, i.e., a plan, to achieve a set of goals under specified constraints. The difference lies in that planning and replanning start from scratch, while plan repair is based on the existing plan. Therefore, before introducing the plan repair problem, we give the following related definitions at first.

Definition 1 (*operator*) An operator o is an action framework without instantiated parameters, which is expressed as $o = \langle n_o, p_o, d_o, pre_o, eff_o \rangle$. pre_o indicates the mandatory requirements for execution, and eff_o describes how o changes the world after execution, including the positive effect eff_o^+ and the negative effect eff_o^-.

An operation outlines a capability of an agent and its conditions, providing a model basis for decision-making. In fact, an operator always has a duration, and for simplicity, when making a decision, the durative operator could be divided into three parts: the beginning part, the intermediate part, and the end part. The beginning part is described by the condition of action starting and the instant effect after application, while the intermediate part usually only contains persistent conditions, and the end part covers the preconditions of the end of the action and its final effect. Then, a decision-maker can choose an appropriate operator to fulfill the mission goal and try to instantiate the operator.

Definition 2 (*action*) An action $a = \langle n_a, t_a, p_a, d_a, pre_a, eff_a \rangle$ is an instantiated operator. Apart from the above parameters in an operator, there is a time instant t_a to specify when the action will happen in execution. And its end time can then be calculated by adding the duration with the occurrence time, i.e., $t_a + d_a$.

Once an action is selected to meet the target requirements, the execution conditions of the action also need to be met by other actions or a state.

Definition 3 (*state*) A state is the portrayal of a spacecraft at a certain time, which could be expressed as $S = \{t_S, F, V\}$, where F depicts the logical status of the spacecraft at the time instant t_S, e.g., a camera is turned on, and V depicts the numerical status, e.g., 70% of the fuel is left at t_S.

If all preconditions of action are true in a certain state, the action can then be applied in that state and change that state by performing its effects. And then other new actions can be applied in the derived state until a goal state is reached and all the conditions of actions in the action queue have no conflict. In that case, that action queue is a plan.

Definition 4 (*plan*) A plan $P = \{a_1, a_2, \ldots, a_n\}$ is a time-ordered set of actions and is expected to achieve the mission goals at a certain initial state.

When put into practical use, the derived plan is fragile to adapt to the uncertain space environment like a sudden storm and anomalous events like the failure of magnetorquer, resulting in the conditions of the planned activities are destroyed and goals cannot be achieved, which means a plan failure.

Definition 4 (*event*) An event $E = \{t, e\}$ is a time-specified happening and usually has unsatisfactory results.

For example, {0, (not (camera calibrated))} describes an event that the camera is not initially calibrated. Then, we have the plan repair problem as follows:

Definition 5 (*plan repair problem*) A plan repair problem Π is a six tuple, i.e., $\Pi = \langle P, F, V, O, I_{\Pi}, G_P \rangle$, where O contains uninstantiated operations to model the way and effect that the spacecraft changes the world, while I_{Π} depicts the status of the spacecraft at the start of plan repair, and G_P represents targets that have not yet been achieved.

Then, the plan repair problem is how to synthesis a series of operators from O and instantiate them under the constraints of F and V, so that the spacecraft can reach the state G_P from the specified initial state I_{Π}.

It should be noted that there is more than one target in G_P. In addition to unfulfilled mission goals, the effects of unexecuted planned actions could also be selected as possible objectives for plan repair, with the expectation that the interrupted execution will be resumed by finding a recovery plan to bridge the failure state with the remaining executable parts of the failed plan. Most of the existing plan repair methods use the try-error manner to find a reachable solution from the target set, which has the disadvantage of low efficiency as discussed above.

3 The Retargetable Goals Plan Repair Method

In order to do plan repair for spacecraft, whose actions are durable, concurrent, and resource dependent, a transformation manner to convert an action sequence into a state queue is first introduced to provide possible targets for recovery. And then the derived targets are regarded as retargetable goals, and the goal evaluation expression integrating multiple factors, e.g., reachability and run time of plan repair, is proposed to assist in goal selection for plan repair, so as to improve the efficiency of plan repair.

3.1 *Transformation of the Plan Repair Problem*

There are many subsystems or devices in the spacecraft such as the attitude subsystem and the camera in Fig. 1. The attitude subsystem can change the attitude of the whole spacecraft by turning, which could support the imaging of the camera and the data transmission operation of the antenna. Therefore, they are related to each other, leading to that the actions in the pre-designed plan of the spacecraft can be durable, concurrent, and resource dependent. Recall the expression of the operator in Definition 1, we can synthesize the different preconditions and effects at the boundary of the action at the same time, so as to form a composite state at that time, as shown in Fig. 1. Then, the pre-designed plan can be transformed into a state queue, e.g., $\{S_0, \ldots, S_6\}$ in Fig. 1. When a plan failure happens, the state queue serves as an optional repair goal provider for the plan repair problem, which wants to find a recovery plan from the state S_f after the plan failed to a feasible target, i.e., one state in the set $\{S_4, S_5, S_6\}$.

However, it is redundant for any state in the state queue. Because it is composed of the effect of an action, which includes not only the desired impact but also other information such as energy consumption, system status, and so on. For example, for the action *take_photo* in Fig. 1, its effects include not only obtaining the target photo but also ending the work status of the camera and consuming both storage space and electricity power. Moreover, because all items in the state need to be satisfied, the redundant element will lead to a longer solution time when it is taken as a repair goal for recovery.

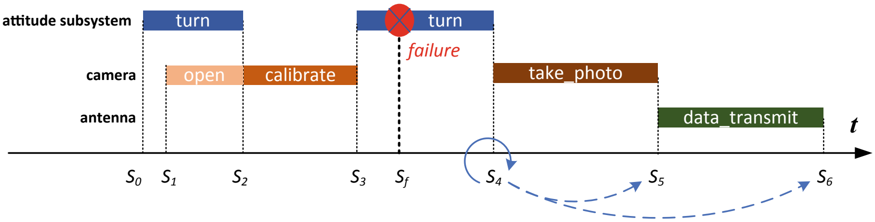


Fig. 1 Transformation diagram of plan repair problem

To overcome that, we use the state regression function $\Gamma(s, a)$ as shown in Eq. 1 to calculate the wanted effect. Where S' is the partial state that presents the minimal set of propositions, which must hold in the state, to achieve the state S by executing the action a.

$$S' = \Gamma(S, a) = S \backslash eff_a^+ \cup eff_a^- \cup pre_a \tag{1}$$

Note that there are not only logical elements in the state variable but also numerical components such as data storage. In regression, the numeric values in S are calculated in reverse at the end effects of the action, e.g., the regressed state should store the battery consumption of the spacecraft used during imaging. Besides, since there are many different types of resources in spacecraft such as fuel and electricity power, and actions can be concurrent, the effects should be merged according to the scheduled execution time [19]. When calculating, numerical effects are classified first according to their type, and then the total consumption of each type is calculated respectively. And since not all types of resources are required by the scheduled actions, we select the numerical conditions that appear in the action as the expected numerical state. Then we get a simplified version of the state queue, which we call the retargetable goals.

Another thing to note is that unforeseen failures may happen at any time during the execution of the plan. When execution encounters a plan failure, it is generally difficult for the spacecraft to terminate all performances immediately and enter a safe mode. Therefore, spacecraft actions are assumed to be non-interrupted in this chapter. Then, the initial state of the plan repair problem is an estimated version, which is given by the end effect of the action closest to the failure in the given plan, as shown in S_4 in Fig. 1. Then the initial state of the plan repair problem is changed from S_f to S_4.

Now, we have the initial state and possible recovery goals for plan repair, then based on the existing spacecraft operators, a plan repair problem can be transformed into a pathfinding problem, which can be solved by finding a path from the initial state to one of the retargetable goals.

3.2 Overview of the Method

Instead of trying to target each regressed state in turn until a repair solution is found, we can directly hit the optimal retargetable goal and find a recovery plan easier. To this end, we develop the retargetable goals plan repair Method, ReGPR, whose pseudocode is shown in detail in Fig. 2.

ReGPR is reactive to plan failures and is fed with the initial state and retargetable goal queue of repair, and gives out a result, either a recovery plan or a failure flag. In ReGPR, at first, the input retargetable goals are evaluated by the evaluation function described in detail in the following part, and the goal with the minimum evaluation value is chosen as the repair target (line 1). Note that there may be two or more goals

```
Algorithm1 Retargetable Goals Plan Repair Method (ReGPR)
Input: initial state $I$ and retargetable goal queue $G_S$
Output: recovery plan $\boldsymbol{P}$ or failure
1.  $G$ ← choose a goal from $G_S$ with a reasonable minimum $E$ value
2.  for $g$ in $G$, do
3.      open ← $I$, closed ← $\varnothing$
4.      while open is not empty, do
5.          $S$ ← remove the node from open with smallest heuristic value
6.          closed ← $S$
7.          $A$ ← find applicable actions in $S$
8.          for $a$ in $A$, do
9.              ss ← a copy of $S$
10.             succ ← apply $a$ in ss
11.             if the heuristic value of succ is 0
12.                 trace the action sequence $\boldsymbol{P}$ to succ and return $\boldsymbol{P}$
13.             else if succ is in open or closed, continue
14.             else open ← succ
15.             end if
16.         end for
17.     end while
18. end for
```

Fig. 2 Pseudocode of the ReGPR method

with the same evaluation value, then the repair target is selected in chronological order (line 2). And once the available repair objective is determined, combined with the initial state of repair, we can use an algorithm similar to A* to solve the plan repair problem by finding a path through the specified initial state I to the selected repair goal g (lines 3–17). This algorithm selects the open node with the lowest heuristic value for expansion (line 5). And the heuristic used here is TRPG [21]. Then, some actions that are satisfied in the selected state are applied, and a new state *succ* is created (line 10). If the goal distance estimation is 0, i.e., *succ* reaches the goal, we find the solution and return it (line 12), otherwise, this node is put into the open list (line 14) except duplicate nodes already in the open list and closed list (line 13).

4 Evaluation of Retargetable Goals

It is ineffective trying to search for a solution path from the initial state to possible retargetable goals one by one. On the contrary, it is necessary to exploit the existing plan and formulate criteria to assess priorities in retargetable goals. To this end,

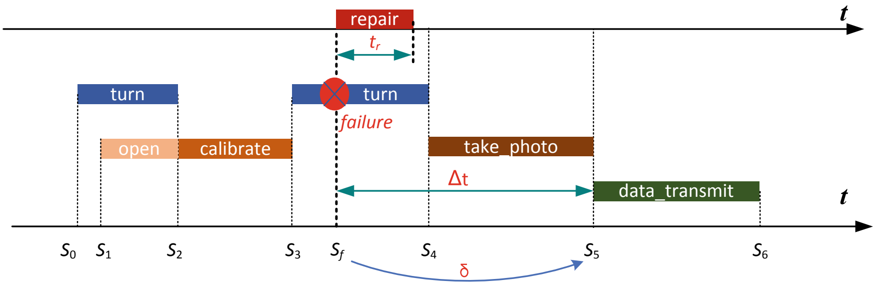


Fig. 3 Element diagram of evaluation criteria of retargetable goals, including the goal reachability δ, run time t_r of repair, and repair deadline Δt

several important things come into mind. First and foremost, if the retargetable goal cannot be reached, the wasted search time can then be utilized to notify the ground control as soon as possible. And if it is reachable, the unfulfilled targets may be repaired. Then, the process of plan repair solving is time-consuming, which is ignored by most of the related research work. In fact, the execution of actions is time-tagged and has strict time constraints. If the relevant time constraints are broken, the availability of unexecuted actions in the original failed plan will be greatly questioned. When execution time is considered, the deadline for plan repair, the fixed time span between the initial state for recovery and the corresponding retargetable goal, has to be mentioned. And finally, once a recovery plan is found, it should be merged with the remaining applicable actions in the failed plan to output a new feasible execution plan for the spacecraft, which also takes time to complete.

Combined with these considerations, we finally designed the evaluation criteria E_S for the retargetable goals, as shown in Eq. 2. And the meaning of these factors can be found in Fig. 3. The retargetable goals can then be sorted by E_S. If the goal is unreachable, there is no need to repair it. Otherwise, the shorter the repair time or the longer the spare time to update a new execution plan, the better, i.e., the smaller the value of E_S, the higher the priority of the goal S.

$$E_S = \delta \cdot \frac{t_r}{\Delta t - t_r} \tag{2}$$

4.1 Estimation of E_S

Obviously, the value of E_S is determined after solving the problem. However, this deviates from the original design intention, i.e., using E_S to guide the path search and speed up the repair process. Therefore, an estimation version of E_S is in need.

4.1.1 Estimation of δ

For goal reachability, we use the TRPG heuristic [21] as a tool to judge whether there is a solution between the initial state and the specified retargetable goal. TRPG is a temporal variant of the relaxed planning graph, which has an alternating structure of the action layer and fact layer and has a timestamp with each layer to specify when the corresponding item is deemed to happen, as shown in Fig. 4. And these layers are connected with each other through the precondition and effect of an action or the empty operator *no-op*. Then one can find a path through the given initial state to the desired goal state in the plan graph. However, it may be time-consuming. Therefore, in order to simplify the problem, several factors are relaxed such as ignoring the deletion effect of actions and exaggerating the numerical effect (for the positive/negative effect, the maximum/minimum value of the variable involved is increased/decreased). Then, a relaxed plan can be easily traced back, which can be used as a heuristic to estimate the remaining effort to achieve the goal.

When called, the TRPG function is fed as a black box by an initial state and the specified retargetable goal and outputs the estimated goal distance h. When it is estimated that there is no path between the two states, i.e., unreachable, the value of h is -1, otherwise, h is the number of actions leading the initial state to the specified goal state. Then, we can determine the value of δ as shown in Eq. 3.

$$\delta = \begin{cases} -1 & h < 0, \\ 1 & otherwise \end{cases} \tag{3}$$

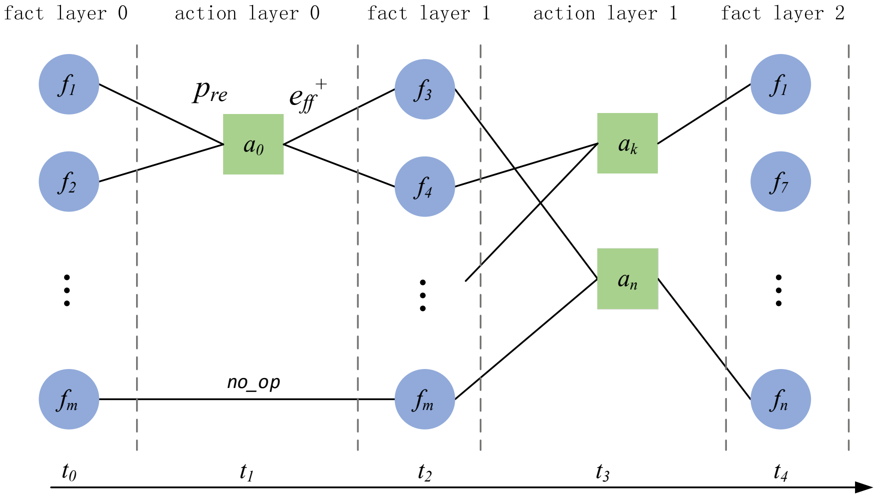


Fig. 4 Illustration of TRPG

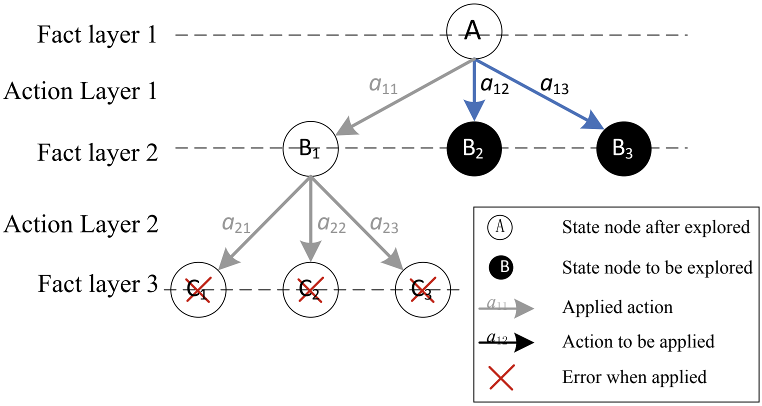


Fig. 5 Illustration of the search vacillation

4.1.2 Estimation of t_r

For repair solution time, when the search method in Fig. 2 is used to solve the plan repair problem, there is a phenomenon called *search vacillation* [22], i.e., there is a gap between the node generation and the node expansion. For example, there are three fact layers and two action layers in Fig. 5. The state nodes B_2 and B_3 are generated before the explored state nodes in the third fact layer. If there are no more actions available for all the three state nodes in the third fact layer, B_2 and B_3 will be used in the next expansion. Assuming that B_2 is selected as the next extension object, there are three nodes are explored between its generation and extension, i.e., C_1, C_2, and $C_{3.}$ Therefore, it can be said that the extended delay of B_2 is 3.

Then, combined with the goal distance estimation of the TRPG heuristic, we can derive the estimation expression of t_r, as shown in Eq. 4. To get the value of each factor, as pointed out in [22], an offline version of the search method is run under the same hardware conditions. And the average of that results of $\bar{t}$ and d are taken as the final value in Eq. 4 to estimate the plan repair time.

$$t_r = h \cdot \bar{t} \cdot d \tag{4}$$

4.1.3 Calculation of Δt

Since there are many durative actions in the plan, a series of expected states distributed at different times can be obtained by Eq. 1. When one of them is selected as a possible repair target, in order to maximize the efforts of an exists to plan, plan repair should be done before the target comes, i.e., to make sure that a recovery plan and a new spacecraft execution plan will be found within the time limit. To capture this, we propose a rolling calculation method to get the value of Δt, as shown in Fig. 6.

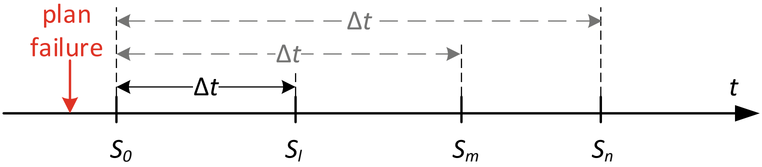


Fig. 6 Illustration of the rolling calculation of Δt

When an expected state is selected for repair, e.g., S_l in Fig. 4, the value of Δt is the time difference in the failed plan between the initial state of repair S_0 and S_l. If S_l fails to resume the plan execution, the next expected state S_m is selected, and the value of Δt is correspondingly updated until the end of the failed plan.

With the estimation of each factor, the value of E_S will be obtained naturally. Each state in the retargetable goal queue will have the corresponding evaluation value. And the one with the minimum evaluation value is chosen to be the repair goal. If there are two or more retargetable goals with the same evaluation value, the repair goal is selected in chronological order, which lays the foundation for rapid repair.

5 Experimental Results and Discussion

In order to evaluate the proposed ReGPR method, the Satellite Complex domain[1] in the International Planning Competition is used in this experiment. These tests were conducted on a series of variants of problems that were initially solved by the POPF2 [23] planner by artificially changing plan execution state at different time instants in the form of an event as defined in Sect. 2. ReGPR was implemented based on POPF2, and all tests were performed on a Debian server with a Xeon(R) E5-2698 v3 @ 2.30 GHz × 16 CPU and 19GiB RAM.

5.1 Experimental Set-Up

In the basic Satellite Complex domain, the spacecraft turns itself to perform several observation tasks with varying slew times and calibration times combining with limited data capacity. In order to adapt to real space operations, this domain is modified as follows: we first designed the power consumption for each activity, so that there are two renewable resources in this field, i.e., electricity power and data storage; then, we introduced an antenna device to support the data transmission operation; finally, we designed a charging operation to replenish the spacecraft with electricity under the consideration of the energy consumption and online supplement. Then,

[1] The domain is available from https://ipc02.icaps-conference.org/CompoDomains/SatelliteComplex.pddl.

with any input of initial and goal states, we can acquire a plan from the planner. And here we use the POPF2 planner as the problem solver to give out an initial feasible plan.

In these tests, plan failures are simulated by changing the environment in the execution context, for example, changing the direction of the satellite sometime before it turns. And the simulated plan failures in the satellite complex domain can generally be classified into two categories: contextual logic changes and resource level changes. For the actions being executed or to be executed, if the context logic changes, e.g., rover location and camera calibration status, or the resource level changes, e.g., insufficient power, their preconditions will be destroyed, and the execution of the plan will be blocked.

To test the performance of ReGPR, we compared it with the replanning method and plan repair method on the same platform. For replanning, we use the POPF2 and lpg-td [24] planner to give out a new plan under a plan failure. And for plan repair, we use a version of ReGPR without goal evaluation, hereinafter referred to as ReGPR-ne. As for performance, we mainly focus on the problem-solving time, the number of explored nodes, and the number of search rounds. If a method has less solution time, fewer explored nodes, and fewer search round, it can be said that the performance of this method is better.

5.2 Results and Discussion

We randomly set the plan failure types of logic dissatisfaction, resource dissatisfaction, and both dissatisfactions at different execution times in the 12 test instances originated from the initial plan output by the POPF2 planner, and got the results as shown in Figs. 7, 8, 9 and 10. In order to reduce the error as much as possible, all experimental results are the arithmetic mean after three runs, and the running time is limited to 120 s.

Compared with the other two replanning methods, the two plan repair methods solve more problems in a shorter time on average. On the one hand, apart from the 8th–12th problems that cannot be solved by all the four methods due to time run out or evaluated as unreachable, POPF2 and lpg-td cannot solve the 7th problem. The unsolvable problems are caused by either too low a power level to start an activity or an unreachable goal in the current state. While POPF2 and lpg-td chase for the mission goal, ReGPR, and ReGPR-ne fulfill only the recent retargetable goal. Therefore, the two plan repair methods have a higher success rate as shown in Table 1. On the other hand, the two plan repair methods have a shorter solving time. According to Table 1, the average total solving time of all these test problems for ReGPR, ReGPR-ne, POPF2, and lpg-td is 0.5667 s, 0.5297 s, 57.9156 s, and 66.7728 s respectively, as shown in Table 1. For the respective solvable problems, the average solving time is 0.8922 s, 0.7817 s, 26.8195 s, and 28.7533 s. And for the unsolvable problems. the results are 0.1107 s, 0.177 s, 101.45 s and 120 s respectively.

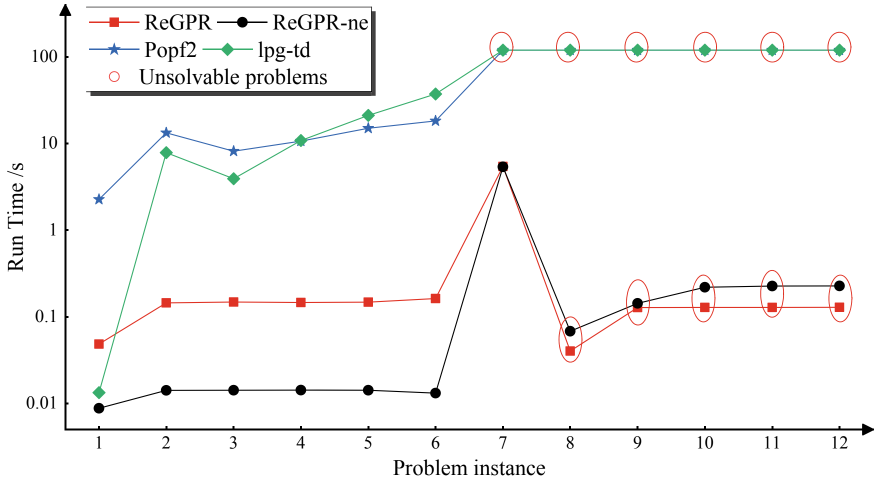


Fig. 7 Run time for the test problems

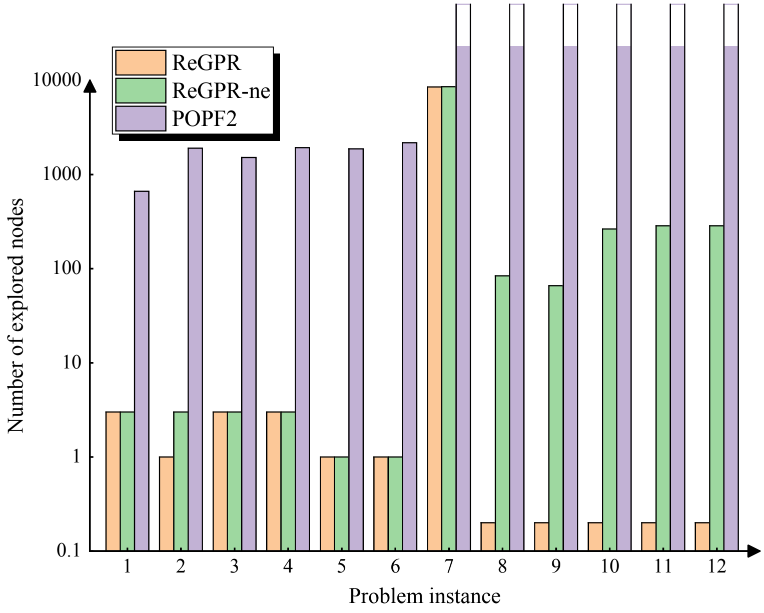


Fig. 8 Number of explored nodes during the search

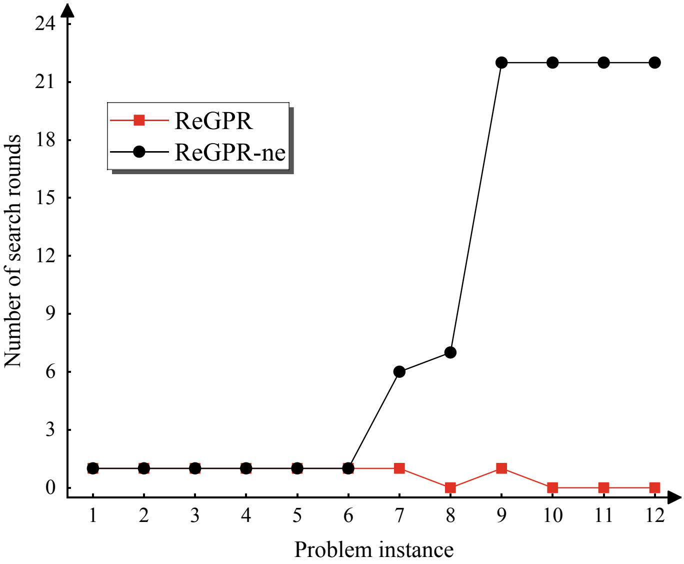


Fig. 9 Number of search rounds during the search

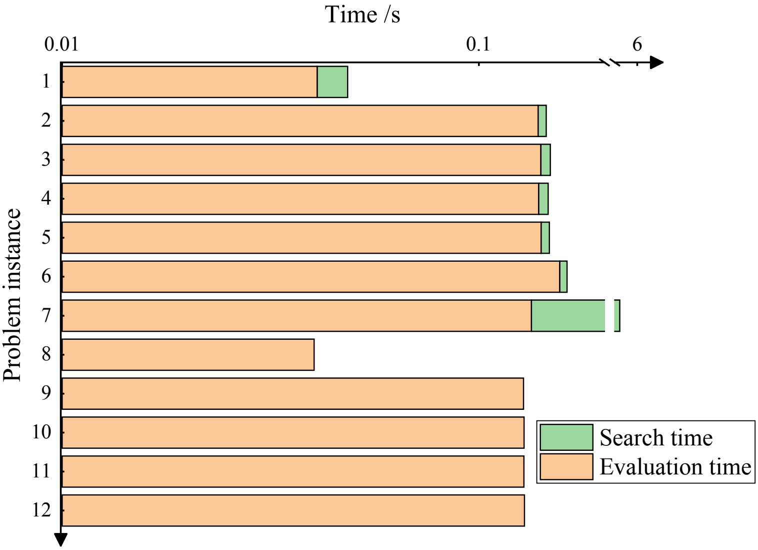


Fig. 10 Different components of ReGPR running time

Table 1 Statistical results of the four different methods

	ReGPR	ReGPR-ne	POPF2	lpg-td
Solved problems	7	7	6	6
Success rate (%)	58.33	58.33	50	50
Average solving time (s)	0.5667	0.5297	65.6447	66.7728
Average solving time (except unsolvable problems) (s)	0.8922	0.7817	26.8195	28.7533
Average solving time (include unsolvable problems) (s)	0.1107	0.1770	120	120

Then it can be concluded that: (1) Plan repair is indeed more efficient than replanning in terms of problem-solving time; (2) ReGPR shares the same success rate with ReGPR-ne, which confirms the correctness of the proposed method in this chapter; (3) The average solving time of ReGPR is 0.86% of POPF2 and 0.79% of lpg-td, and it is 6.95% higher than that of ReGPR-ne, which means that ReGPR has low efficiency in term of the total problem-solving time; (4) However, when the unsolvable problem is considered, the average solving time of ReGPR is 0.09% of POPF2 and lpg-td, and 37.46% lower than that of ReGPR-ne, which means that ReGPR has great advantage in dealing with a unsolvable problem.

There are two knids of problems for the plan repair method among these 12 instances: solvable (1–7 problems) and unsolvable (8–12 problems). It is clear form Fig. 7 that ReGPR takes 76% more time solving problems on average than ReGPR-ne in the solvable problems, while 36% searching time is saved on average in the unsolvable problems compared with ReGPR-ne. The reason is that most of the solvable problems can be resumed in just 1search round as shown in Fig. 9. Then, the method of collecting all redirectable targets and evaluating them must take longer solving time than the method without evaluation. However, when the plan repair problem becomes complex, the search time will increase, and the solution time gap between the two repair methods will narrow. In the 7th problem, the solution time difference between the two plan repair methods is 0.0534 s, which is 0.99% of ReGPR-ne. And when the problem is unsolvable, ReGPR-ne has to try all the targets before declaring failure, while ReGPR just needs evaluate them before the real search progress, result that a lot of search time could be saved. In fact, plan fialures are diverse due to the uncertain space environment, and not all execution failures can be solved autonomously. But if it can be judged to be unsolvable before performing the time consuming search process, the problem could be tackled as soon as possible by delivering it to engineers, to avoid further deterioration of the situation.

Moreover, compared with the other plan repair method ReGPR-ne in the terms of explored nodes during the search, as shown in Fig. 8 (because the source code of lpg-td is unavailable,[2] it is hard to output the result), and search rounds as shown in Fig. 9, the proposed method ReGPR has a significant advantage, especially in the unsolvable problem instances, i.e., the 8th, 10th, 11th, and 12th problems, whose

[2] The homepage of LPG is https://lpg.unibs.it/lpg/.

search rounds is 0. (In order to characterize the special value of 0, 0.2 is specified as 0 explored nodes during the search in the Fig. 8.) The number of rounds here refers to the number of changes to the retargetable goal in the search process of solving a problem. If the number of rounds is 0, it means that the search is not performed. This is in line with the original intention of the method design, that is, the repair goal is evaluated before performing the search, so as to save unnecessary search time.

Besides, in the normal cases in Figs. 8 and 9, ReGPR has 1 search round while ReGPR-ne has different rounds, and ReGPR explored nodes no more than ReGPR-ne during the search. It could be divided into two cases. The first case is the 2nd problem, where both the two plan repair methods search with one retargetable goal, but ReGPR explored just 1 state node while ReGPR-ne explored 3. The reason is that after evaluation, ReGPR found that the third retargetable goal has the lowest evaluation value with lower reachability estimation, while ReGPR-ne searched directly to the first goal and found a feasible recovery plan. And the other case is the 7th problem, where ReGPR explored 8519 nodes in 1 search round within 5.4464 s while ReGPR-ne explored 8584 nodes in 6 rounds within 5.393 s. And the final selected retargetable goals are the same for the two methods. The reason is that after evaluation, ReGPR gets rid of those unreachable goals before performing searching, so that it can directly focus on the targets that are expected to be solved.

As a method of evaluating firstly the optimal goal and then searching for solutions, we also want to know the time spent in each part. As can be seen from Fig. 10, except for the unsolvable problem instances and the 7th problem, where the search process spent more time than the evaluation process, in other cases, the average evaluation time is 2270.88 times longer than the average search time (the average evaluation time 0.1261 s and the average search time is 0.007 s). The reason for this may be that in the face of multiple redirectable targets, although the evaluation process reduces the number of search rounds, it cannot reduce the number of evaluations. Each evaluation process needs to be estimated by a heuristic. And in this chapter, the TRPG heuristic serves that estimation. Before using the TRPG heuristic, initialization is required. Once the initial state or the goal state of the TRPG heuristic is changed, it should be reinitialized. Besides, for spacecraft, apart from the logical part in a state, there is also a numerical part such as power and fuel, and the duration of action can also be variable. All these determine that reinitialization is a time-consuming process. And it may be improved by alternating a heuristic with lower computation effort. However, it can be inferred from the 7th problem that when the problem becomes complex, the search time will dominate the total solving time of the problem. In that case, ReGPR will perform better.

6 Conclusions

Plan failures of a pre-designed plan may inevitably happen due to the uncertainty of the space environment. This chapter proposes a reactive rapid autonomous plan repair algorithm based on retargetable goals, ReGPR, considering space operations are durable, concurrent, and resource dependent. It transforms the spacecraft plan into a time-ordered state queue, extracts retargetable goals for recovery, and gives an evaluation criterion of the retargetable goals. The performance of this method is verified by many tests based on the modified Satellite Complex domain. And the experimental results demonstrate that ReGPR can reduce the number of search rounds and the number of explored nodes in the search, and it has great advantages in complex problems. But the evaluation time dominates the total solving time in most problems. This method can provide technical support for future spacecraft missions to deal with emergencies autonomously.

Acknowledgements The authors would like to thank the editors for their hard work on this chapter. This work was supported by the National Key Research and Development Program of China [grant number 2019YFA0706500]; the National Natural Science Foundation of China [grant numbers 61976020, 62006019], the Civil Aerospace Research Project of China [grant number KJSP2020020302] and Shanghai Aerospace Science and Technology Innovation Fund (SAST).

References

1. Bresina J, Dearden R, Meuleau N, Ramakrishnan S, Smith D (2002) Planning under continuous time and resource uncertainty: a challenge for AI. In: Proceedings of the eighteenth conference on uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., Alberta, Canada, pp 77–84
2. Ulamec S, O'Rourke L, Biele J, Grieger B, Andrés R, Lodiot S, Muñoz P, Charpentier A, Mottola S, Knollenberg J, Lander-Philae R (2017) Operations on comet 67P/Churyumov-Gerasimenko, analysis of wake-up activities and final state. Acta Astronaut 137:38–43
3. Chien S, Doubleday J, Thompson DR, Wagstaff KL, Bellardo J, Francis C, Baumgarten E, Williams A, Yee E, Stanton E, Piug-Suari J, Davies M (2017) Onboard autonomy on the intelligent payload experiment CubeSat mission. J Aerosp Inf Syst 14(6):307–315
4. Cividanes F, Ferreira M, Kucinskis F (2019) On-board automated mission planning for spacecraft autonomy: a survey. IEEE Lat Am Trans 17(6):884–896
5. Silva R, Fratini S, Guerreiro J, Nogueira T, Policella N (2020) Goal-oriented onboard autonomous operations: an OPS-SAT experiment. In: 71st international astronautical congress, IAC 2020, international astronautical federation, IAF, Virtual, Online
6. Chang K-H, Han H, Day WB (1993) A comparison of failure-handling approaches for planning systems—replanning vs. recovery. Appl Intell 3(4):275–300
7. Mohalik SK, Jayaraman MB, Badrinath R, Feljan AV (2018) HIPR: an architecture for iterative plan repair in hierarchical multi-agent systems. J Comput 13(3):351–359
8. Bechon P, Lesire C, Barbier M (2019) Hybrid planning and distributed iterative repair for multi-robot missions with communication losses. Auton Robot 1–27
9. Chen C, Xu R, Zhu S, Li Z, Jiang H (2020) RPRS: a reactive plan repair strategy for rapid response to plan failures of deep space missions. Acta Astronaut 175:155–162
10. Nebel B, Koehler J (1995) Plan reuse versus plan generation: a theoretical and empirical analysis. Artif Intell 76(1–2):427–454
11. Xu R, Chen C, Cui P, Zhu SY, Xu F (2019) Research on spacecraft autonomous mission plan repair. J Astron 40(07):733–741 (in Chinese)
12. Hammond KJ (1990) Explaining and repairing plans that fail. Artif Intell 45(1–2):173–228
13. van der Krogt R, de Weerdt M (2005) Plan repair using a plan library. In: Belgian/Netherlands artificial intelligence conference, Brussels, Belgium, pp 254–259

14. van der Krogt R, de Weerdt M (2005) Plan repair as an extension of planning. In: ICAPS 2005: proceedings of the 15th international conference on automated planning and scheduling. Association for the Advancement of Artificial Intelligence (AAAI), Monterey, California, USA, pp 161–170
15. Guzman C, Castejon P, Onaindia E, Frank J (2015) Reactive execution for solving plan failures in planning control applications. Integr Comput Aided Eng 22(4):343–360
16. Talamadupula K, Smith DE, Cushing W, Kambhampati S (2013) A theory of intra-agent replanning. Arizona State Univ. Tempe Dept. of Computer Science and Engineering
17. Holler D, Bercher P, Behnke G, Biundo S (2020) HTN plan repair via model transformation. In: KI 2020: advances in artificial intelligence. 43rd German conference on AI, 21–25 Sept. 2020. Springer International Publishing, Cham, Switzerland, pp 88–101
18. Wally B, Vyskocil J, Novak P, Huemer C, Indelar R, Kadera P, Mazak-Huemer A, Wimmer M (2021) Leveraging iterative plan refinement for reactive smart manufacturing systems. IEEE Trans Autom Sci Eng 18(1):230–243
19. Chen C, Xu R, Li Z, Zhu S, Cui P (2020) Using plan repair method to cope with plan failures in Chinese orbiting mars mission. In: 71st international astronautical congress, IAC 2020, international astronautical federation, IAF, Virtual, Online
20. Fox M, Gerevini A, Long D, Serina I (2006) Plan stability: replanning versus plan repair. In: Long SS, Borrajo D, McCluskey L (eds) Proceedings of the 26th international conference on automated planning and scheduling (ICAPS 2006). AAAI Press, London, UK, pp 212–221
21. Coles A, Coles A (2017) A temporal relaxed planning graph heuristic for planning with envelopes. In: ICAPS 2017: 27th international conference on automated planning and scheduling. AAAI Press, Pittsburgh, PA, United States, pp 47–55
22. Burns E, Ruml W, Do MB (2013) Heuristic search when time matters. J Artif Intell Res 47(1):697–740
23. Coles AJ, Coles AI, Fox M, Long D (2010) Forward-chaining partial-order planning. In: ICAPS 2010: proceedings of the 20th international conference on automated planning and scheduling. AAAI Press, Toronto, Ontario, Canada
24. Gerevini A, Saetti A, Serina I (2004) Planning with numerical expressions in LPG. In: ECAI-04: proceedings of the 16th European conference on artificial intelligence. IOS-Press, Valencia, Spain, pp 667–671

InSight-SEIS Instrument Deployment Operations on Mars

Charles Yana, Ken Hurst, Laurent Kerjean, Emilien Gaudin, Philippe Lognonné, Ludovic Rochas, Agnès Jullien, Frederique Meunier, Rémi Lapeyre, Nicolas Verdier, Khaled Ali, and Benjamin Jaillant

Abstract On November 26, 2018, the NASA InSight spacecraft successfully landed on Mars. This paper describes the operations that took place on Mars after that landing, focusing on deployment with the SEIS (Seismic Experiment for Interior Structure) seismometer instrument's robotic arm, followed by the associated wind and thermal shield. This is the first time an instrument has ever been deployed on another planet. The operations led by NASA-JPL lasted several weeks, until February 2019. CNES, the French Space Agency, is leading the operations concerning the SEIS instrument. The SEIS instrument consists of two independent 3-axis seismometers: an ultra-sensitive very broad band (VBB) oblique seismometer, and a miniature, short period (SP) seismometer. These sensors measure movements at the surface of Mars and will enable the planet's interior structure to be characterized for the first time. The mission will last for two years (one Martian year). This paper is divided into two parts. The first part describes how the CNES team prepared for the SEIS deployment operations on Mars, through a yearlong training course with NASA-JPL and a rigorous certification process. The many lessons learned during this training process will be mentioned here, especially how operational issues encountered during the tests were taken into account to update the operational processes and the ground tools before the real operations began. The second part of the paper describes the real deployment operations on Mars in detail. The SEIS team was responsible for performing a health assessment on SEIS using the telemetry received, preparing SEIS activities while taking available resources and flight constraints into account, and delivering sequences to be run on Mars. The entire team was operating remotely from JPL, far from home for a long period of time. This paper therefore also includes

C. Yana (✉) · L. Kerjean · L. Rochas · A. Jullien · F. Meunier · R. Lapeyre · N. Verdier
Centre National d'Etudes Spatiales, 18 Avenue Edouard Belin, Toulouse, Cedex 4, France
e-mail: charles.yana@cnes.Fr

K. Hurst · K. Ali
Jet Propulsion Laboratory, 4800 Oak Grove Dr, Pasadena, CA 91109, USA

E. Gaudin · B. Jaillant
Telespazio France, 26 Avenue Jean François Champollion, 31100 Toulouse, France

P. Lognonné
Institut de Physique du Globe de Paris, Université de Paris, 1 rue Jussieu, 75005 Paris, France

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_29

a section on the management of French teams operating from NASA-JPL in California over a three-month period, especially in relation to the human and technical challenges encountered during that critical phase, and how lessons learned from the phase could benefit future missions.

Keywords InSight · Mars · Deployment · Seismometer · Training · Operations

Acronyms/Abbreviations

AFT	Allowable Flight Temperature
ATLO	Assembly, Test and Launch Operations
CAB	Centro de Astrobiologia
C&DH	Command and Data Handling
EDL	Entry, Descent and Landing
FOCSE	French Operations Center for Science and Exploration
ICC	Instrument Context Camera
IDC	Instrument Deployment Camera
IOT	Instrument Operations Team
IPGP	Institut de Physique du Globe de Paris (Université de Paris)
MOS	Mission Operations System
PDL	Payload Downlink Lead
PUL	Payload Uplink Lead
SEIS	Seismic Experiment for Interior Structure
SP	Short Period seismic sensor
SISMOC	SEIS on Mars Operations Center
VBB	Very Broad Band seismic sensor
WTS	Wind and Thermal Shield

1 Introduction

When the NASA InSight spacecraft landed on Mars on November 26, 2018, the JPL operational teams in the Jet Propulsion Laboratory (JPL) control room in Pasadena, CA, celebrated a spectacular and very successful EDL sequence on Mars, but only for a few minutes. They then quickly focused on the "deployment" phase that the successful landing had kicked off, resulting in a fully commissioned lander with instruments operating nominally on the surface of Mars just a few weeks later.

This paper describes how the preparation of InSight instrument operations ramped up until landing day and provides a focus on team training and the technical and operational SEIS system qualifications. While InSight has already commenced its extended mission on Mars, the authors took the opportunity of the 16th SpaceOps Conference to reflect on the instrument deployment phase that was initiated after landing and which, being completely unprecedented, constituted the most challenging part of the operations.

2 The InSight Mission, SEIS and Its Instruments

2.1 The InSight Mission

The InSight mission to Mars was selected by NASA on August 20 2012 as part of the NASA Discovery program, in order to deploy the first geophysical observatory on Mars, providing scientific knowledge that is essential to the understanding of the fundamental processes involved in the formation and evolution of telluric planets.

The InSight lander successfully landed on the surface of Mars on November 26, 2018 and deployed the SEIS instrument on Martian ground on December 19, 2018. SEIS has been successfully commissioned and operating nominally on Mars since then.

SEIS, by measuring seismic activity, the meteorite impact flux and the Phobos tide, will characterize the deep interior structure of Mars, providing information about the thickness and structure of the crust, the composition and structure of the mantle and the size of the core. This instrument is the main experiment onboard the InSight mission. The key scientific requirement is related to the a priori low activity of Mars as compared to the Earth and to the fact that InSight consists of a single seismic station. This implies very low instrumental and station noise in order to not only detect the smallest quakes but also, for the more significant ones (i.e. magnitude >4), to detect multiple Rayleigh wave packets. This multiple Rayleigh wave detection is the key to localizing events. Detection of three wave packets (R1, R2, R3) will determine quake time/distance and Rayleigh wave speed, while detection of two wave packets (R1, R2) will determine quake time/distance for a known wave speed. After localization, seismic models will then be derived from the analysis of seismic phases.

These requirements demand very low instrument noise (~10–9 m/s2 rms in bandwidth 0.01–0.5 Hz), 3 axis measurements and shielding or decorrelation from all environmental noise louder than the instrument noise. The quality and success of the deployment of SEIS are hence pivotal to how well the instrument performs and directly affect the quality of the data produced and supplied to the scientific community.

2.2 The SEIS Instrument

The SEIS is the main instrument of InSight. It is a hybrid seismometer, i.e. both broadband and short period. The sensor itself ("sensor assembly") is deployed on the surface of Mars, while the electronic "E-Box" remains in the lander (Fig. 1).

A shield (WTS) protects the seismometer from the wind and thermal insulation stabilizes its temperature. VBB sensors are located inside the vacuum sphere. A leveling system ensures their horizontality.

At the core of the SEIS instrument are three VBB (Very Broad Band) sensors (to reconstruct three measurement axes) in a titanium sphere. They were developed by the French institutions IPGP and CNES. This sub-system includes three independent mechanical pendulums with their feedback electronic board (inside the E-box), a sphere in a vacuum which filters thermal environments, and which also provides mechanical support, and three proximity electronic (PE) boards (one per pendulum) for measuring the displacement of the pendulum, mounted in a box outside the sphere, and hung on the leveling system (Fig. 2).

Outside the sphere, three short-period sensors and the associated electronics were developed by Imperial College and Oxford University with UK Space Agency (United Kingdom Space Agency) funding. The sphere is mounted on a leveling system (LVL) which couples the seismic sensors with the ground and keeps the sphere and VBBs horizontal. The LVL and the associated electronics were developed

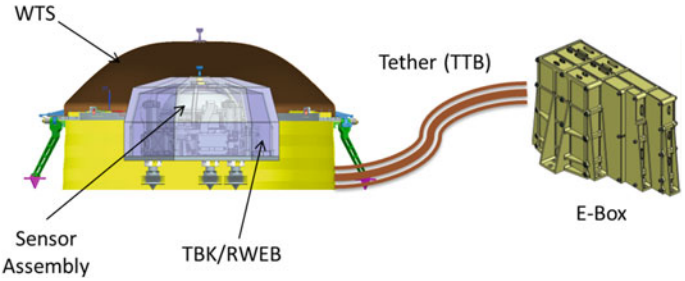


Fig. 1 View of the sensor head deployed on the surface of mars

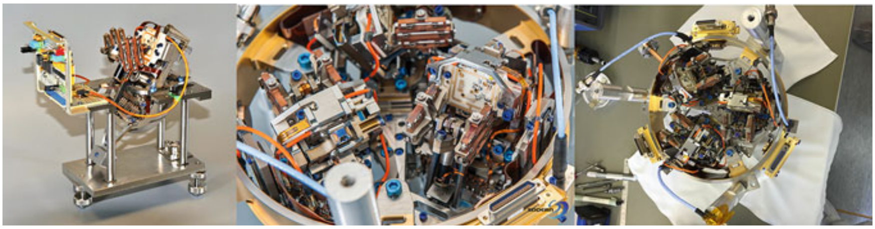


Fig. 2 From left to right: pendulum VBB engineering model (SEIS/InSight), sphere model incorporating three VBB axes, SEIS sphere flight model

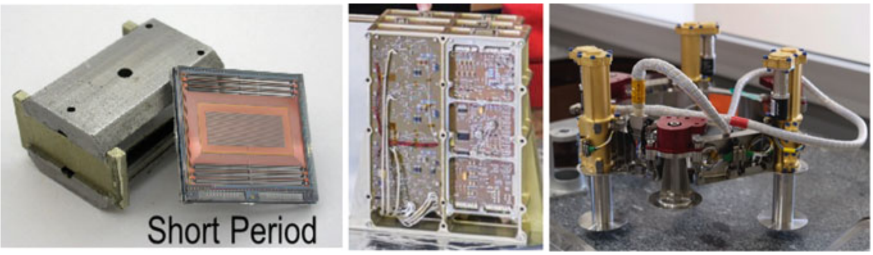


Fig. 3 From left to right: short period sensors, E-box and leveling system

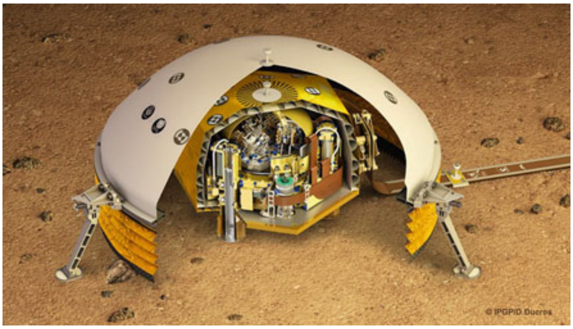
by MPS (Germany) with DLR funding. The electronics box and DC/DC converter board, and the electronic control and acquisition board were developed by ETH (with SSO (Swiss Space Office) funding via ESA/PRODEX), who also integrated electronic boards from other subsystems (Fig. 3).

The atmospheric turbulence shield (Wind and Thermal Shield) deposited on the surface of Mars over the seismometer was developed by JPL/NASA, as well as the tether and tether Storage Box connecting SEIS on Martian ground to the E-box on the lander (located inside a thermal enclosure below the deck) (Fig. 4).

In addition to SEIS comes the APSS (Auxiliary Payload Sensor Suite), which is a set of environmental sensors (wind, pressure, temperature, and magnetic field sensors) considered to be auxiliary payloads, along with the associated electronics (Payload Auxiliary Electronics). This was developed by JPL, UCLA (responsible for the IFG magnetometer) and CAB (responsible for the TWINS wind sensors). During the scientific monitoring phase, CNES took over the operational duties for APSS but did not develop it.

Finally, the SEIS/APSS Flight Software (FSW), shared by both instruments, is used to command and monitor the SEIS and APSS. The science data are also managed by the FSW and buffers used for both the SEIS and APSS are available on the lander to cope with telemetry and UHF limited bandwidth. JPL was responsible for developing the SEIS/APSS FSW from specification written by CNES.

Fig. 4 The SEIS sensor assembly



2.3 *The SEIS Ground Data Segment*

2.3.1 The FOCSE-SISMOC Mission Center

In addition to the SEIS instrument, CNES has also developed a complete operations Ground Segment. It is called SISMOC (SEIS Mars Operations Center) and is located in Toulouse, France, forming part of the FOCSE Center, dedicated to Science and Exploration operations at CNES.

On one hand, the SISMOC offers a set of basic services such as data management, task scheduling and system supervision, constituting the core system of the operation center. On the other hand, the SISMOC includes a set of mission-specific services such as management of high resolution data or correlation of the various clocks. The functional capabilities allocated to the SEIS ground segment are:

- SEIS and APSS health and safety assessment,
- Programming of SEIS/APSS (including management of the downlink bandwidth via configuration of the continuous data processing),
- Onboard seismic event buffer management,
- Various onboard time correlations,
- Detection/characterization of seismic events,
- Detection of meteorite impacts,
- Production, distribution and archiving of L0 and L1 products (uncalibrated and calibrated miniSEED).

The SEIS system has many interfaces, both onboard and on the ground. The lander communicates with the JPL Mission Operations Ground Segment, which is the interface between the SEIS and APSS IOT and the instrument. In addition to communicating with the JPL Ground Segment for data delivery and instrument command uplink, the SEIS Ground Segment communicates with SEIS Science entities for seismic event detection and characterization, and data distribution to the science team. The SEIS Ground Segment also has interfaces with APSS teams for calibration purposes, and pressure and wind data formatting and distribution (Fig. 5).

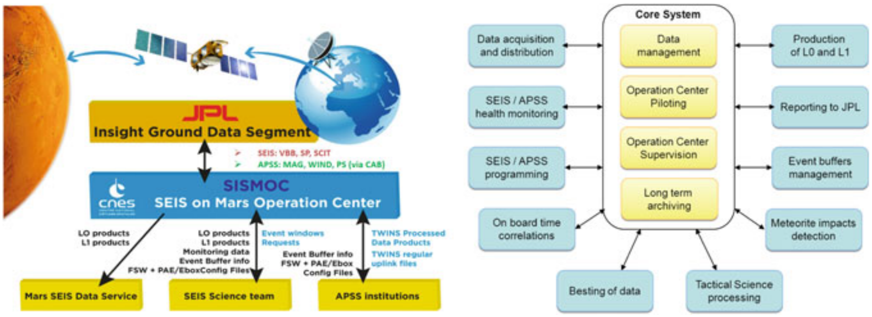


Fig. 5 SISMOC general functional architecture

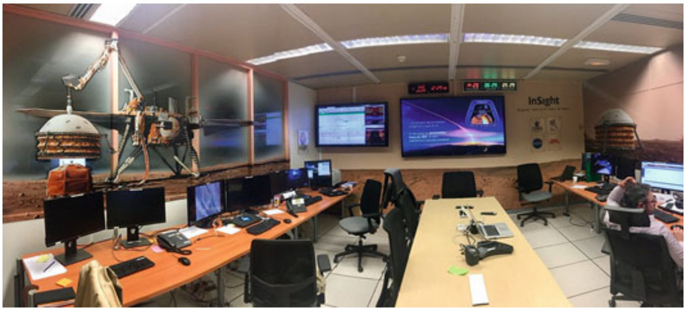


Fig. 6 FOCSE-SISMOC operations center at CNES

2.3.2 SEIS Science Services

In order to achieve these tasks, the SEIS ground segment is organized into two major components: the SISMOC, which is installed in CNES-Toulouse that mainly deals with the engineering operations and the scientific tactical processing (Fig. 6), and the Mars SEIS Data Center (MSDS), developed by IPGP, in charge of producing high-level end scientific products, in order to archive them and to supply scientific products to the scientific community.

SISMOC and MSDS distribute the SEIS and APSS data to the science team, which is also organized with several SEIS Ground Services (Fig. 7), including:

- **Mars Quake Service (MQS)** which is located in the ETH facilities in Zurich, Switzerland. The role of the MQS includes monitoring of the SEIS and APSS data for seismic events, and the preparation of the Mars Seismic Catalog.
- **Mars Structure Service (MSS)** whose role it is to fully characterize the seismic events detected and to elaborate and improve the Mars scientific models.
- The **SISMOC**, which is installed in CNES-Toulouse, which mainly deals with the engineering operations and the scientific tactical processing.
- The **Mars SEIS Data Center (MSDS)**, which is in charge of producing high-level end scientific products, in order to archive them and to distribute scientific products to the scientific community. Even though it is mentioned here, this Center is out of the scope of this document and will not be described in this paper.

2.4 SEIS and APSS Operations

Operations require telemetry to be monitored on a daily basis in order to assess the instrument's health and the data bandwidth for downloading the seismic data recorded onboard to be delivered after processing to the various scientific locations for analysis. In the meantime, the activity plan and associated sequences of commands are prepared for uploading and running on Mars the following week, taking into account the new seismic events detected or other types of requests, such as calibrations and VBB recenterings (Fig. 8).

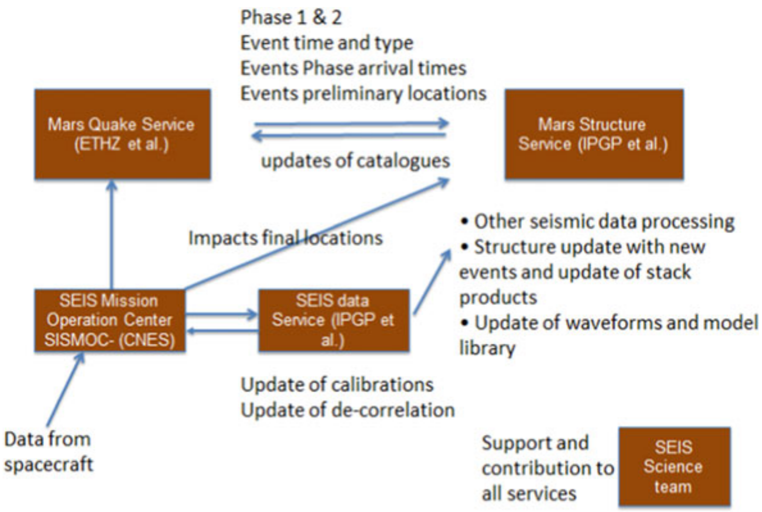

Phase 1 & 2
Event time and type
Events Phase arrival times
Events preliminary locations
Mars Quake Service (ETHZ et al.)
Mars Structure Service (IPGP et al.)
updates of catalogues
Impacts final locations
• Other seismic data processing
• Structure update with new events and update of stack products
• Update of waveforms and model library
SEIS Mission Operation Center SISMOC- (CNES)
SEIS data Service (IPGP et al.)
Update of calibrations
Update of de-correlation
Data from spacecraft
Support and contribution to all services
SEIS Science team

Fig. 7 Joint operations of the SEIS science services

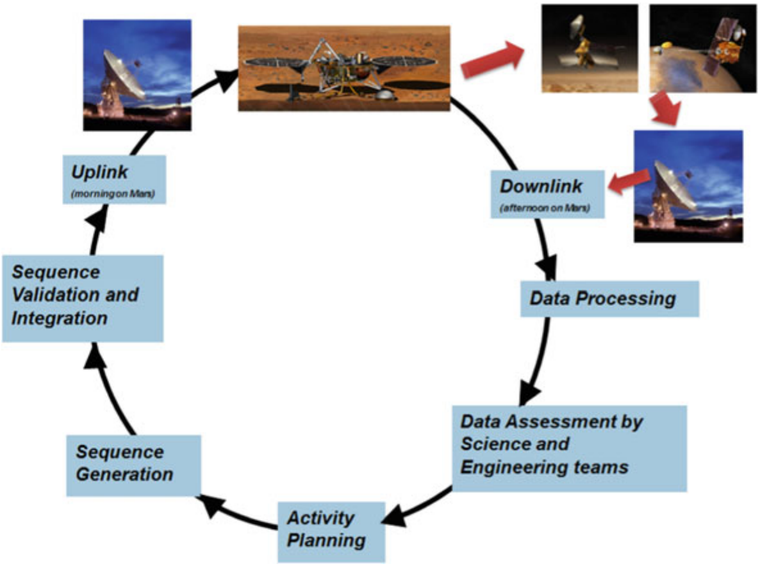


Fig. 8 Daily operations cycle during deployment

Daily monitoring is performed by the SEIS operations team, and the SEIS health and safety assessment is provided to the InSight Mission at JPL using dedicated tools (NSYTReports) and during tactical meetings.

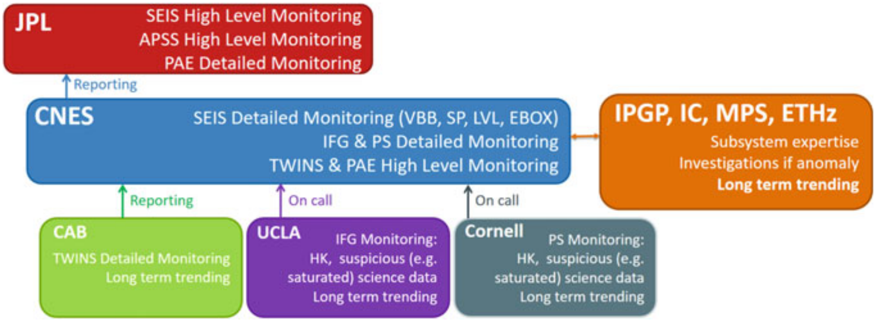

JPL
SEIS High Level Monitoring
APSS High Level Monitoring
PAE Detailed Monitoring
Reporting
CNES
SEIS Detailed Monitoring (VBB, SP, LVL, EBOX)
IFG & PS Detailed Monitoring
TWINS & PAE High Level Monitoring
IPGP, IC, MPS, ETHz
Subsystem expertise
Investigations if anomaly
Long term trending
Reporting
On call
On call
CAB
TWINS Detailed Monitoring
Long term trending
UCLA
IFG Monitoring:
HK, suspicious (e.g. saturated) science data
Long term trending
Cornell
PS Monitoring:
HK, suspicious (e.g. saturated) science data
Long term trending

Fig. 9 SEIS and APSS downlink organization

The SEIS and APSS are monitored jointly by CNES partners. CNES leads the team's overall tasks but requires input and support from partners who participated in the development of the subsystems. CNES performs the daily analysis of SEIS and APSS telemetry, and CAB performs the daily analysis of TWINS telemetry. In case of an issue, such as unexpected or off-nominal behavior, or missing data, CNES can obtain expert support from IPGP (VBBs), Oxford/Imperial College (SPs), ETH (E-box), MPS (LVL), CAB (TWINS), UCLA (IFG), and JPL (FSW) (Fig. 9).

In parallel, the SEIS and APSS operations team prepares daily activity plans for the instrument:

1. At the beginning of the planning process, the SEIS IOT provides input and activity requests for SEIS and APSS, and receives power and CPU time allocations consistent with available resources from Mission Planning at JPL bandwidth.
2. Within this allocation, SEIS IOT determines the activities that can be performed during the uplink cycle (calibrations, seismic event requests, leveling, etc.).
3. Following scientifically-relevant event detections, Events Requests are received from the science team during the planning cycle and are taken into account for the activity plan.
4. Once the activity plan has been defined, sequences, configuration files and parameter tables are drawn up, validated and sent to JPL for bundling and uploading to the spacecraft, before being run on Mars during the days that follow.

CNES leads the tasks, but partners provide support during the uplink process (in addition to science input and requests from the science team): IPGP for the VBBs, Oxford/Imperial College for the SPs, ETH for the E-box, MPS for the LVL, CAB for the TWINS wind sensor, UCLA for the IFG magnetometer and Cornell for the Pressure Sensor. During the deployment phase, the downlink and uplink coordination between CNES and SEIS/APSS partners and subsystems is managed during the daily SEIS tag-up meeting, organized and led by a member of the Operations team (the SEIS Operations System engineer at CNES by default).

Only the CNES Operations team is authorized to interface with JPL for operations: downlink assessment, uplink process. The only exception to this is the CAB team who provides the TWINS downlink assessment directly via the dedicated tool (called NSYTReports) and could at times interact with JPL in operations meetings as needed.

3 The Preparation of Operations

3.1 Operational and Technical Qualification

The InSight mission was selected by NASA in August 2012. Shortly after, in November 2012, a permanent project correspondent for operations activities was appointed. The project correspondent's role was to prepare the operational phase for SEIS and APSS instruments, and provide the SEIS/APSS MOCD (Mission Operation Concept Document) and the SEIS/APSS General Operation Plan documents for the mission. This was conducted in coordination with JPL operations plans. The project correspondent acted as the operational lead, fulfilling the role of SEIS Operations Manager in interface with the project during development for the operational validation of the center and the operations of phase E1 (deployment and commissioning). During the development phase, he coordinated the Operations team and with the technical and operational activities of the different team members and support system. This involved acting as the interface between the SEIS System team and the SEIS Operations team, but also between the JPL MOS (Mission Operations System) team and CNES for all the operational topics and reviews that arose during development. He also ensured that the Operations team members had benefited from sufficient training to support and handle SEIS and APSS Operations.

It was crucial to the success of SEIS and APSS operations that this role be defined and active very early in the development phase. This was necessary in order to anticipate the time when operations would kick off and when the instrument, the lander, the flight software, the ground segment and the science team would all intersect together with the first data recorded on Mars. However, before this happened, the SEIS and APSS system underwent a full operational and technical qualification process, culminating with the Operational Readiness Review held at JPL before real operations commenced.

Several kinds of tests were identified and resulted in the operational qualification of the operators and the mission center for SEIS and APSS. This enabled them to gain knowledge and experience on the instrument, the ground data system, and the operational processes:

- **Operability tests** made it possible to validate command products, the flight software algorithm, the first version of the tools and software, and general SEIS telemetry. They did not involve APSS at that time. Those tests were heavily focused on SEIS and low-level command validation. They took place at CNES.

- **Functional tests** involved a lot more GDS tool functionalities and representative instrument activities that were more representative of what would actually occur on Mars. They enabled SEIS to interact more with other mission subsystems, like InSight C&DH (the lander onboard processor), JPL tools, and relevant duration and data volume modeling for SEIS activities. They took place at CNES.
- **ATLO (Assembly, Test and Launch Operations)** took place at the Lockheed Martin facilities in Denver, CO, where SEIS and APSS were integrated into the spacecraft. Part of the ATLO involved end-to-end functional tests for SEIS. Operations teams participated in two SVT tests (Surface Verification Test, simulating science monitoring activities) and two DST tests (Deployment Scenario Test, simulating deployment operations). The huge advantage of those ATLO tests is that they were performed using Flight Models of SEIS, APSS and the InSight lander. Even though the conditions on Mars (gravity, atmosphere) could not be fully simulated on Earth, those tests were the closest to the reality of future operations. They were held in 2017, roughly one year before landing. Within the SEIS team, future operators prepared all the command products (sequences of commands, configuration files) that were run during those tests, as they would when real operations started. However, the operational processes were not representative of real operations (sequence delivery process, downlink assessment, etc.), especially the operational timeline. The ATLO tests marked the formal end of the technical qualification for SEIS.
- **Operational Readiness Tests (ORT):** operational processes and products were validated during ORTs organized by JPL. SEIS-focused pre-ORTs were organized by CNES, also involving many members of the science team. The following chapters of this paper will take a closer look at the ORTs. The pre-ORTs took place at CNES and generally lasted 3–4 days. The ORTs took place at JPL and each lasted one week in general.

Once all of these tests had been conducted, the SEIS team members were ready to start operating SEIS on Mars (Fig. 10).

As part of the operational qualification, the JPL MOS team organized many Operational Readiness Tests (ORTs) the goal of which is to conduct end-to-end testing of all phases of the mission: launch, cruise, EDL (Entry, Descent and Landing), and surface operations. The SEIS team was involved in the surface operations ORTs, along with other payload teams as well as the Lander team at Lockheed Martin. These tests aimed to be as representative as possible of real operations. Hence the "deployment" ORTs, simulating the first sols of the mission, were held at JPL in Pasadena, CA, where all the teams would work side by side for real operations. Regarding SEIS, they involved both development and operational teams, since both the experts who developed the instrument and the engineers who would operate it on the surface of Mars were set to participate in deployment operations. The science monitoring (or "routine") surface operations, however, only involved operational teams. These teams participated remotely from the SISMOC mission center in Toulouse, France, as would be the case during real operations. The development teams then served as backup and experts in case of anomalies, as for the real "routine" operations (Science

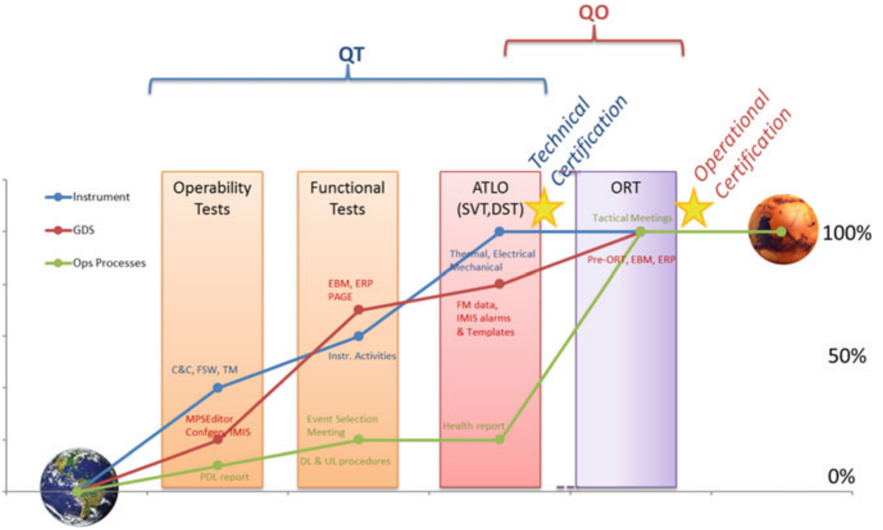
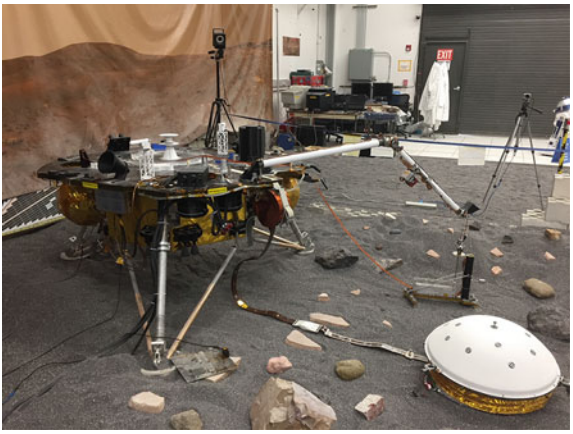


Fig. 10 The SEIS technical and operational qualification process

Monitoring phase, or phase E2). CNES provided an engineering model and a weight model for SEIS that was installed in the JPL InSight testbed and used during ORTs. Those tests were heavily focused on operational processes and realistic timelines for downlink assessment, product deliveries, planning processes, etc. One of the ORTs for surface monitoring simulated anomalies with the payloads and the spacecraft. This was a good training opportunity for the operations teams, reducing stress levels should that anomaly occur during real operations (Fig. 11).

Operational products that were validated during ORTs include a general SEIS site-acceptance process defined with the JPL MOS team (and actually led by them)

Fig. 11 InSight Testbed at JPL used for ORTs



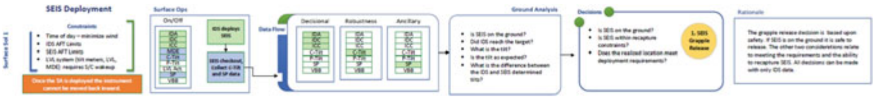

SEIS Deployment
Surface Sol 1
Constraints
Time of day – minimize wind
IDS AFT Limits
SEIS AFT Limits
LVL system (tilt meters, LVL, MDE) requires S/C wakeup
Once the SA is deployed the instrument cannot be moved back inward.
Surface Ops
On/Off
IDA
IDC
ICC
MDE
C-Tilt
P-Tilt
LVL Act
SP
VBB
IDS deploys SEIS
SEIS checkout, Collect C-Tilt and SP data
Data Flow
Decisional
Robustness
Ancillary
Ground Analysis
Is SEIS on the ground?
Did IDS reach the target?
What is the tilt?
Is the tilt as expected?
What is the difference between the IDS and SEIS determined tilts?
Decisions
Is SEIS on the ground?
Is SEIS within recapture constraints?
Does the realized location meet deployment requirements?
1. SEIS Grapple Release
Rationale
The grapple release decision is based upon safety. If SEIS is on the ground it is safe to release. The other two considerations relate to meeting the requirements and the ability to recapture SEIS. All decisions can be made with only IDS data.

Fig. 12 Extract of the SEIS deployment success criteria, defined by J. Ervin (InSight deployment systems) at JPL

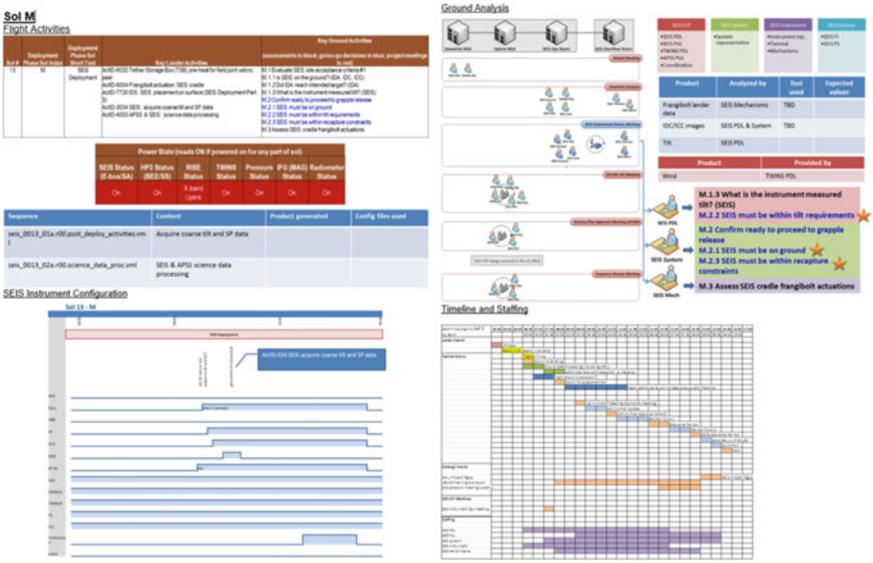
and, in particular, the success criteria for each step of the deployment. Non-reversible decisions therefore had to be made during that phase (deployment of the WTS on top of SEIS for instance) and success criteria based on telemetry were defined to make the operational process run more smoothly, which proved very useful (Fig. 12).

The deployment phase was highly scripted and did not allow for much flexibility in operations, with very limited opportunistic science allowed. Each sol of the mission did correspond to a specific activity, and for sols involving SEIS, a full set of flight and ground activities was compiled in a PowerPoint package. *Flight activities* include pre-defined sequences to be run, expected sensor configuration, and configuration files used. Once flight activities occurred, the *ground activities* section compiled what was set to happen during the assessment process: team members involved, tactical meetings, operational products to assess, decisions to be made, and the associated timeline for operations to ensure consistency.

All activities and operational products were modeled with JPL during the technical qualification process described previously (operability tests, functional tests, ATLO). Modeling means defining the resources required for an activity to occur onboard: duration of the activity, data volume generated, required power, specific constraints, and interoperability with other subsystems. This modeling of all SEIS operational products and activities were a very critical and time-consuming part of the overall validation process, but proved to be key to the success of operations (Fig. 13).

3.2 Team Training

The SEIS operations team followed the JPL certification process for InSight, requiring a core knowledge set and practical on-console training during ORTs before being allowed to participate in real operations on Mars. Since the number of ORTs was limited (only 4 ORTs involving SEIS, each simulating about 5 sols of operations), the challenge was to provide sufficient training opportunities to all six SEIS team members covering the roles of Payload Downlink Lead (PDL) and Payload Uplink Lead (PUL), i.e. the SEIS team members actually on the console in the control rooms during operations. It was decided to involve all team members (PDLs, PULs, but also system engineers, instrument and subsystems engineers, etc.) during all ORTs, and to rely on a lot of shadowing to ensure full benefits could be obtained from the ORTs. This was of course a costly decision, to have the whole team travel to JPL, but it limited the risk of team members being insufficiently qualified. Organizing a lot of



Sol M

Flight Activities

Sol #	Deployment Phase Sol Index	Deployment Phase Sol Short Text	Key Lander Activities	Key Ground Activities (assessments in black; go/no-go decisions in blue; project meetings in red)
13	M	SEIS Deployment	ActID-6020 Tether Storage Box (TSB) pre-heat for field joint velcro peel ActID-6004 Frangibolt actuation: SEIS cradle ActID-7720 IDS: SEIS placement on surface (SEIS Deployment Part 3) ActID-3034 SEIS: acquire coarse tilt and SP data ActID-4000 APSS & SEIS: science data processing	M.1 Evaluate SEIS site acceptance criteria #1 M.1.1 Is SEIS on the ground? (IDA, IDC, ICC) M.1.2 Did IDA reach intended target? (IDA) M.1.3 What is the instrument measured tilt? (SEIS) M.2 Confirm ready to proceed to grapple release M.2.1 SEIS must be on ground M.2.2 SEIS must be within tilt requirements M.2.3 SEIS must be within recapture constraints M.3 Assess SEIS cradle frangibolt actuations

Power State (reads ON if powered on for any part of sol)						
SEIS Status (E-box/SA)	HP3 Status (BEE/SS)	RISE Status	TWINS Status	Pressure Status	IFG (MAG) Status	Radiometer Status
On	On	X-band Uplink	On	On	On	On

Sequence	Content	Product generated	Config files used
seis_0013_01a.r00.post_deploy_activities.vml	Acquire coarse tilt and SP data		
seis_0013_02a.r00.science_data_proc.vml	SEIS & APSS science data processing		

SEIS Instrument Configuration

Ground Analysis

SEIS EST	SEIS System	SEIS Instrument	SEIS Science
•SEIS PDL •SEIS PUL •TWINS PDL •APSS PDL •Coordination	•System representative	•Instrument rep. •Thermal •Mechanisms	•SEIS PI •SEIS PS

Product	Analyzed by	Tool used	Expected values
Frangibolt lander data	SEIS Mechanisms	TBD	
IDC/ICC images	SEIS PDL & System	TBD	
Tilt	SEIS PDL		

Product	Provided by
Wind	TWINS PDL

M.1.3 What is the instrument measured tilt? (SEIS)
M.2.2 SEIS must be within tilt requirements ★
M.2 Confirm ready to proceed to grapple release
M.2.1 SEIS must be on ground ★
M.2.3 SEIS must be within recapture constraints ★
M.3 Assess SEIS cradle frangibolt actuations

Timeline and Staffing

Fig. 13 Daily deployment timeline activities package

operational shadowing was considered to be a stress-reduction action, including for the prime operator, who could thus benefit from the support of the shadow team member during the ORTs.

The ORTs also provided a first idea of the ground segment functionalities and performances, leading, as expected, to a lot of late change requirements. Some tool functionalities proved to be missing or no longer relevant to the actual needs of the operation, and those changes had to be implemented by the GDS development team between ORTs, which is always stressful. The team successfully coped with the situation. What was more difficult to assess however, was the actual load of the system: number of users connected simultaneously, amount of data ingested in the pipelines, and overall infrastructure performances of a ground system still undergoing development (Fig. 14).

Each team member's progress through certification was tracked by the SEIS operations manager and reported to JPL as part of the certification process before flight. The certification matrix provided by JPL and common to all team members participating in operations was also used to ensure all team members could have enough training opportunities to be certified. For instance, all post-landing deployment roles required to:

1. shadow someone in the "prime" seat for ≥3 shifts
2. occupy the "prime" seat with a mentor for ≥3 shifts
3. read/watch all required certification materials
4. be intimately familiar with the procedures for their role.

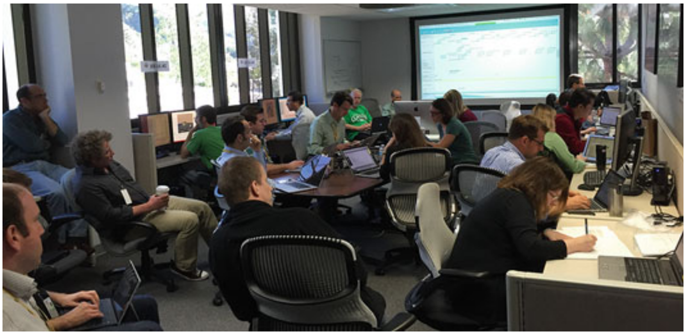
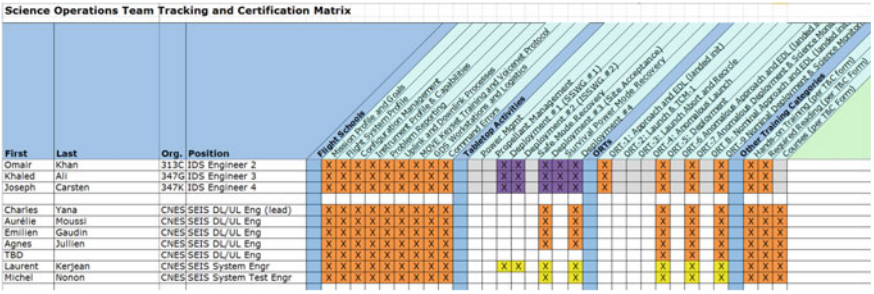


Fig. 14 ORT simulation (pre-COVID) at JPL



Science Operations Team Tracking and Certification Matrix

First	Last	Org.	Position	Flight Schools	Mission Profile and Goals	Flight System Profile	Configuration Management	Instrument Profile & Capabilities	Problem Reporting	Uplink and Downlink Processes	MOVE Keyset Training and Voicenet Protocol	GDS Workstations and Logistics	Command Errors	Tabletop Activities	Power Mgmt	Propellant Management	Deployment #1 (SSWG #1)	Deployment #2 (ISSWG #2)	Safe Mode Recovery	Deployment #3 (Site Acceptance)	Survival Power Mode Recovery	Deployment #4	ORTs	ORT-1: Approach and EDL (landed init)	ORT-2: Launch & TCM-1	ORT-3: Launch Abort and Recycle	ORT-4: Anomalous Launch	ORT-5: Deployment	ORT-6 Anomalous Approach and EDL (landed	ORT-7 Anomalous Deployment & Science Moni	ORT 8- Nominal Approach and EDL (landed init)	ORT-9 Nominal Deployment & Science Monitor	Other Training Categories	Hands-on Training (per T&C form)	Required Reading (per T&C Form)	Courses (per T&C Form)
Omair	Khan	313C	IDS Engineer 2		X	X	X	X	X	X	X	X	X				X	X		X	X	X		X				X		X		X		X	X	
Khaled	Ali	347G	IDS Engineer 3		X	X	X	X	X	X	X	X	X				X	X		X	X	X		X				X		X		X		X	X	
Joseph	Carsten	347K	IDS Engineer 4		X	X	X	X	X	X	X	X	X				X	X		X	X	X		X				X		X		X		X	X	
Charles	Yana	CNES	SEIS DL/UL Eng (lead)		X	X	X	X	X	X	X	X	X							X		X						X		X		X		X	X	X
Aurélie	Moussi	CNES	SEIS DL/UL Eng		X	X	X	X	X	X	X	X	X							X		X						X		X		X		X	X	X
Emilien	Gaudin	CNES	SEIS DL/UL Eng		X	X	X	X	X	X	X	X	X							X		X						X		X		X		X	X	X
Agnes	Jullien	CNES	SEIS DL/UL Eng		X	X	X	X	X	X	X	X	X							X		X						X		X		X		X	X	X
TBD		CNES	SEIS DL/UL Eng		X	X	X	X	X	X	X	X	X															X		X		X		X	X	X
Laurent	Kerjean	CNES	SEIS System Engr		X	X	X	X	X	X	X	X	X				X	X		X		X						X		X		X		X	X	X
Michel	Nonon	CNES	SEIS System Test Engr		X	X	X	X	X	X	X	X	X							X		X						X		X		X		X	X	X

Fig. 15 JPL certification matrix

In the end, all SEIS operators and system team members were successfully qualified before operations commenced on November 26, 2018 (Fig. 15).

3.3 Bathtub Period

One of the difficulties faced by the InSight team was the postponement of the mission launch from 2016 to 2018, due to technical issues with the SEIS instrument in December 2015 (leak from the sphere containing the VBB sensors which required perfect sealing once vacuum has been established inside to ensure good performance from the sensors). With Mars launch opportunities only occurring every two years, the launch slipped to 2018 and resources at JPL were reduced in mid-2016. This resulted in a "bathtub" period of several months during which the JPL MOS team was kept to a minimum, before ramping back up to full capacity mid-2017. During that time, the SEIS instrument team was obviously busy fixing the SEIS sphere (and building a SEIS spare model requested by JPL). The SEIS operations team resources were maintained at the same level, postponing the arrival of additional resources (two additional operators) to 2017. The opportunity was taken to perform more testing, more validation, and more operational product preparation. Pre-ORTs conducted within the SEIS team were organized at CNES during the bathtub period and made it possible to better take the needs of the science team into account concerning operations, seismic data processing and distribution, and logistics during the deployment phase at JPL.

4 The Deployment and Commissioning Operations

4.1 *Deployment Operations*

InSight was finally launched to Mars on May 5 2018, and was set to land on Mars six months and a half later on November 26 2018. During cruise, a heath check out of SEIS was performed, along with all other payload subsystems. The team also took the opportunity to proceed with the first data acquisition and scientific calibrations of the SP sensors.

After landing on Mars on sol 0, several macro-activities were defined:

- Lander initialization and deployment preparation
- Workspace assessment and instrument placement site selection
- SEIS deployment and site acceptance
- WTS deployment
- HP3 deployment and site acceptance.

The second step, dedicated to defining where to deploy the SEIS and HP3 instruments, was set to last 14 sols. However, pre-defined nominal placement sites were chosen ahead of time and consisted of the best sites for SEIS and HP3 on a flat terrain without rocks. Therefore, the instrument placement site selection process could be reduced to just sols as "Site confirmation", providing an additional margin for the following sols. Luckily, the slope of the ground on the landing site happened to be less than 4 degrees, and no rock or obstacle prevented the instruments from being deployed on their pre-defined placement site, so that the whole process could be reduced to 4 sols (Fig. 16).

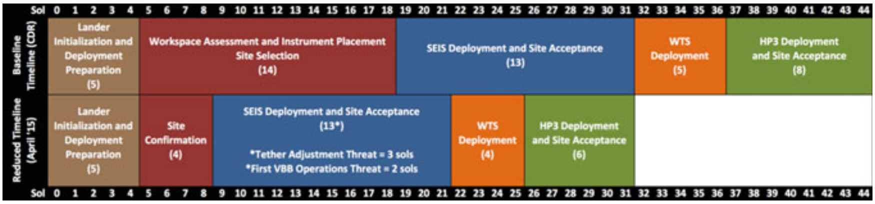


Fig. 16 Initial InSight deployment timeline

The site confirmation assessment, conducted using images taken by the pair of cameras (ICC below the deck of the lander, IDC on the elbow of the robotic arm), was performed in parallel with the first checkout commands sent to the instruments, which proved to be perfectly functional. The temperatures were closely monitored and the sensor assembly, at that time still on the lander deck, experienced temperatures as low as −73 °C during the first Martian nights. The E-box, installed in a thermal enclosure below the deck, was about −6 °C on average.

Checkout was successful, so the SEIS team was able to turn ON the SEIS heater, preventing the temperatures of the sensor assembly to drop below permissible flight temperatures. A few months before landing, it was discovered that the coils of the SEIS sensors were sensitive to cold temperatures, adding a strong constraint to the minimal temperatures the instrument was allowed to experience. On the other hand, turning the heater ON was an energy consuming activity and generated noise on the seismic data. However, while still on the deck, the VBBs sensors were saturated and mostly useless, so only the less sensitive SP sensors, which did not require installation on a perfectly flat site, were turned on and provided the first seismic data recorded on Mars by SEIS on sol 4 (i.e. 4 sols after landing). Calibration of the SP sensors on sol 4 completed the on-deck seismic characterization activity.

An important activity, that proved to be more complex than expected, was the calibration of the tiltmeters embedded in the leveling system, and which were required prior to deployment. Neither sets of tiltmeters (the one on the InSight lander and the one on SEIS LVL) provided consistent measurements, and this puzzling situation led to a requirement for additional measurements. The SEIS tiltmeter was more sensitive to temperature and gravity than expected (it is a spirit level device) and its transfer function had to be recalibrated in order to provide pertinent measurements. This needed to be performed before SEIS was deployed on the ground.

The detailed steps of the SEIS deployment and site acceptance process are as follows:

- SEIS sensor assembly deployment
- SEIS sensor assembly initial leveling
- SEIS VBB sensors first recentering
- SEIS Tether Storage Box opening
- SEIS Load Shunt Assembly opening
- SEIS Leveling low
- Pining Mass Adjustment.

The actual deployment occurred on sol 22, SEIS being off during the activity. The first seismic data on Martian ground was collected on sol 24 by the SP sensors. Before the VBB sensors could record valuable unsaturated seismic data, the whole sensor assembly had to be levelled, meaning the 3 legs of the leveling system had to be activated to bring the sensor assembly to a perfectly horizontal state (less than 0.1° with respect to the horizon), in order to avoid saturation of the VBBs. That activity was carried out on sol 30 and was successful on the first try, with the aggregated tilt being lowered from 2.55° to 0.06°, below the 0.1° target.

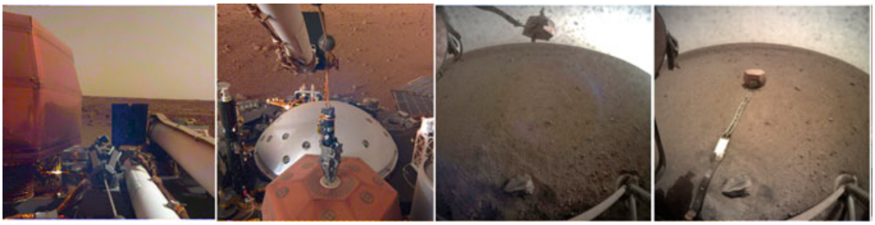


Fig. 17 SEIS sensor assembly deployment

It was now time to start using the heart of the SEIS instrument, the sensitive VBB sensors. The VBBs first had to be recentered, i.e. mechanically balanced by moving a mass along a balancing mechanism using a recentering motor, which would unsaturated the VBBs. That activity was carried out on sol 35 and was also successful. Then followed the tether storage box opening activity on the next sol. The tether (the cable between SEIS and the E-box) was compressed inside a box attached to the lander (the Tether Storage Box), and opening that box allowed the tether to lie flat on Martian ground, without any mechanical constraints from the lander (Fig. 17).

4.2 LSA Opening and Pinning Mass Adjustment

The tether provides electrical connectivity between the SEIS Sensor Assembly, deployed on the Mars regolith about 1.8 m south of the nearest Lander foot, and the SEIS Electronics Box in the thermal enclosure on the Lander. The configuration is shown in Fig. 18. The tether consists of 5 belts each 50.8 mm wide by ~ 0.5 mm thick, made of 5 layers of Kapton™ and 4 layers of copper. The tether is subject to diurnal temperature swings of about 100 °C as it runs across the ground from the lander to the WTS. To isolate the sensor assembly from the thermoelastically-driven motions of the tether, we incorporated an isolation mass called the "Pinning Mass" (PM) clamped to the tether about 40 cm from the sensor assembly and a 90 degree bend up, followed by a horizontal 180 degree bend in the tether called the "Load Shunt Assembly" (LSA) immediately outside the sensor assembly as shown in Fig. 19. The LSA is the primary mean to prevent from thermos elastic effect. The pinning mass was added later in the design to enable a forced opening of the LSA. The first idea was to use the grapple to lift off the pinning mass but this was discovered to be too risky and it was finally preferred to use the scoop to do so.

The LSA must be strong during deployment to resist the pull of the tether hanging between the sensor assembly held in mid-air by the arm, and the lander, and must be extraordinarily weak after deployment in order to decouple the motions of the tether on the ground from the sensor assembly. This was accomplished by heat forming the tether to the desired shape during manufacturing, and clamping the two sides of the horizontal bend together during flight and deployment. The clamping was done

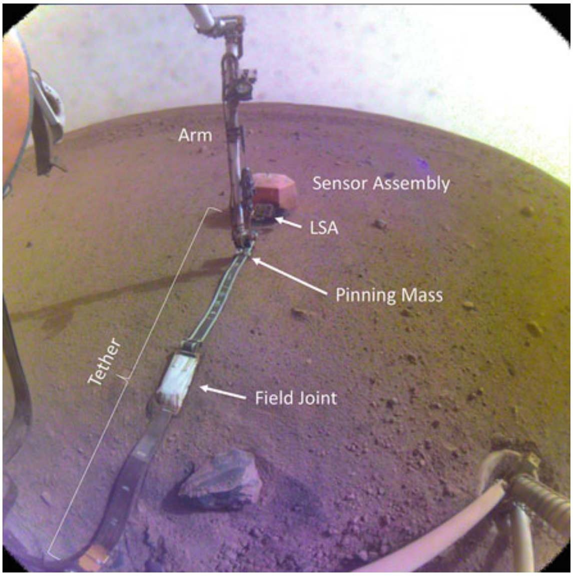


Fig. 18 Left: Context Camera image (fixed, wide angle camera under the lander deck) showing the sensor assembly, the tether, and the arm in the process of adjusting the pinning mass

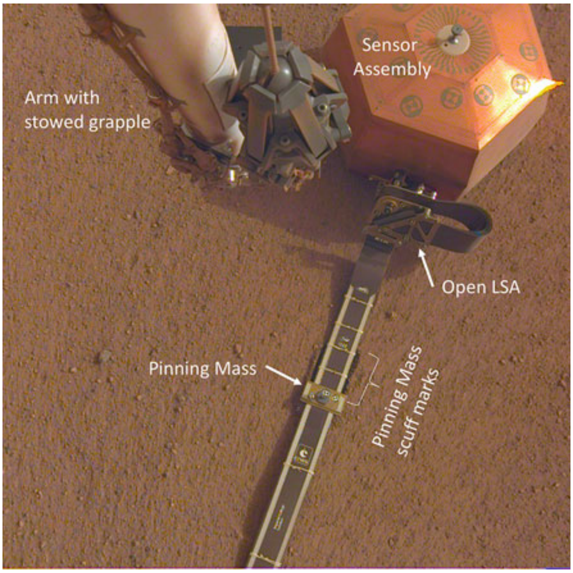


Fig. 19 Deployment Camera image (camera attached to arm) showing the sensor assembly, the tether, the pinning mass, and the LSA after it had been opened

with a breakable bolt (Frangibolt™) that was broken after the deployment team was satisfied with the placement of the sensor assembly on the surface of Mars. After the Frangibolt™ was broken, the two sides of the horizontal bend were separated by dragging the pinning mass back toward the lander using the tip of the scoop on the arm as shown in Fig. 18. The final result is shown in Fig. 19, showing the LSA as clearly open (and the shadow reveals sunlight in the gap) and ready for service.

Moving the pinning mass to open the LSA was, at the time, the most precise use of the arm ever accomplished. It was an iterative procedure, requiring 3 iterations to accomplish the goal of having the LSA clearly open. On the first iteration, the scoop just ticked the top of the pinning mass and only moved it about 2 mm. On the next two iterations, the scoop was lowered by another centimeter to fully engage the pinning mass. The entire tether on the ground moved as a unit during the pinning mass adjustment without generating or enhancing any folds in the tether. After the second iteration, the LSA was open by an amount too small to see in the photographs. We know it was open because of the appearance of a spectral line in the seismometer data corresponding to the ~2.9 Hz fundamental frequency of the open LSA. The third iteration resulted in the LSA being open by about 2 cm at its closest point, which is enough to ensure that the tether will not relax over time and effectively close it.

The effect of opening the LSA can be seen as a notable reduction of background noise on the seismometer in the multi-day spectrogram in Fig. 20.

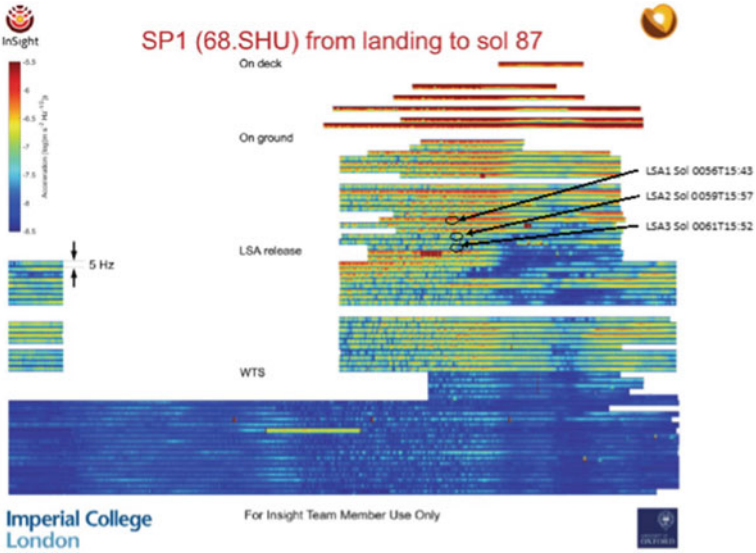


Fig. 20 Multi-day spectrogram from the SP instrument. Note the transition to "cooler" colors after the LSA release, indicating a significant lowering of the background noise level. Iterations of the PM adjustment were done on sols 56, 59, and 61

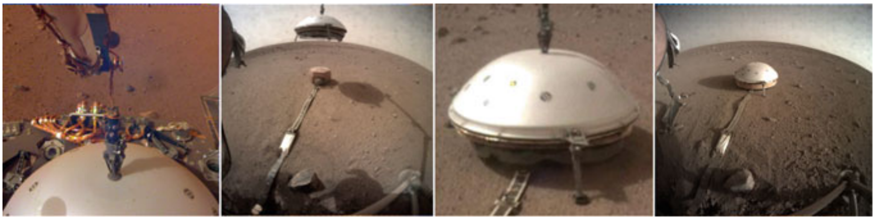


Fig. 21 SEIS wind and thermal shield deployment

4.3 WTS Deployment

The Wind and Thermal Shield (WTS) was deployed on top of SEIS on sol 66. This deployment was critical to the first sols of operations because it made it possible to keep the sensors ON overnight. Before the WTS was deployed, the cold temperatures at night (as low as −80 °C) prevented the sensor assembly, whose AFT limit is −65 °C, from being powered ON, especially because of the sensitive coils. Having the VBB and SP seismic sensors ON continuously (thanks to the igloo effect reducing the temperature gradients inside the WTS) marked the real start of valuable continuous seismic data acquisition.

An unexpected difficulty originated from the WTS skirt, which did not go down as low as expected, as shown on the third image of Fig. 21: the chain mail was (partially) deployed, but not the skirt. That meant that the sensor assembly was not thermally isolated, and dust and wind could therefore penetrate inside the WTS. This situation had to be avoided at all costs, from a performance point of view. Fortunately, gravity and temperature finally had on effect on the skirt which was eventually fully deployed after a couple of days without requiring further action on it (no shaking, no bumping).

After sol 66, the overall temperature increase of the sensor assembly, now beneath the WTS, made it possible to keep the VBB and SP sensors ON overnight on sol 70, and to turn the heater OFF on sol 73, reducing the noise generated by the heater and saving energy. The WTS helped reduce the temperature gradient of the sensor assembly by about 15 °C, which was a lot less stressful for the hardware (Fig. 22).

4.4 Commissioning

The WTS deployment phase was followed by the deployment of the HP3 instrument, provided by DLR, the German Space Agency. In parallel with that phase, SEIS carried on with its commissioning phase, with all sensors on continuously in nominal configuration and the heater OFF. Actual HP3 deployment was carried out on sol 76, and the first steps of SEIS commissioning were programmed on sol 77 and 80, with two consecutive fine tunings of the Thermal Compensation Device Mechanism

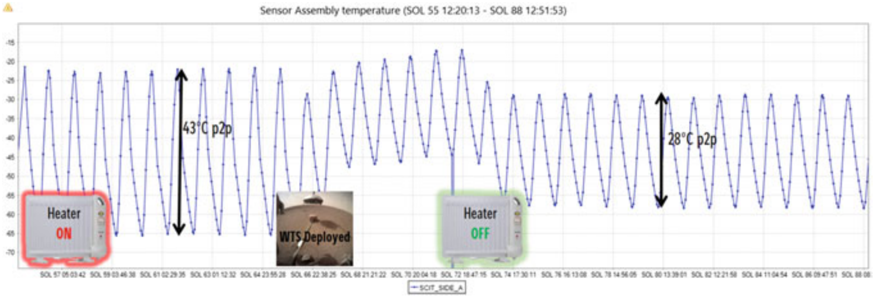


Fig. 22 Temperatures of the SEIS sensor assembly, sols 55–88

(TCDM) of the VBBs, which reduced the peak to peak amplitude of the POS output of the VBB2 from ~8 to ~2 V. VBB1 and VBB3 excursions were also reduced but their initial peak to peak amplitude was less critical than VBB2 (~4 V initially).

The overall pendulum performance was then assessed based on a "F0/Q" characterization and its transfer function was established with several calibrations at different temperatures, using a pre-loaded calibration waveform.

SEIS was then fully commissioned by mid-March 2019 and ready to transition to scientific monitoring operations. It recorded the first Marsquake on April 7, 2019 (sol 128).

In parallel with the SEIS commissioning period, the CNES operators progressively took over APSS operations by shadowing APSS PDL and PUL roles, while JPL engineers were still at the console. This training had already begun during ORTs but the real operations were of course more interesting, especially since the monitoring tools used by CNES were different to those used by JPL. The operational duties after landing were covered by JPL engineers and they took the time to train the CNES engineers during real operations, which was challenging for both teams: for JPL because it occurred during real operations (but APSS commissioning only lasted a few sols and was already over when the transition process started), and for CNES because it occurred in parallel with SEIS commissioning, which required additional staffing capabilities. The training and handover of operational duties was very successful, thanks to the excellent collaboration between teams and the unwavering dedication of the JPL engineers while training the CNES engineers. All technical issues related to APSS prior to the handover were handled by JPL, including operational workarounds and flight software updates.

4.5 *Lessons Learned*

A big part of the success of the SEIS deployment operations can be credited to the excellent synergy between the SEIS development team and the SEIS operations team. Most development (i.e. phases ABCD of the project) team members transitioned off the project after deployment and commissioning operations (phase E1) were over, but they were not actually operating the instrument during those phases. The SEIS operations team took over operational duties when the ORTs started: they covered the roles of Payload Downlink Lead (PDL), in charge of SEIS monitoring, and Payload Uplink Lead (PUL), in charge of SEIS programming and strategic planning. However, it was important that the development team could bring its expertise to the operations, and engineers who developed SEIS and its subsystems (E-box, VBBs, SPs, LVL) participated in phase E1 on-site at JPL. All key decisions were taken collectively within the entire team, depending on the context and the subsystem involved: development team, SEIS PI, operational team, SEIS instrument engineer at JPL, etc. The SEIS development Project Manager and the SEIS PI had the final word for all SEIS-related decisions. That long phase (which lasted more than 3 months) was the perfect opportunity for the operations team to finalize the training process with real data and real Martian conditions, being able to work in an integrated team with the development team. This process worked very well, and no operational or decision-making issue occurred because the entire decision-making process had been defined and rolled out in advance (and no major flight anomaly occurred). By sitting at the console to perform operations during the ORTs, plus the deployment and commissioning operations, the SEIS operators acquired the knowledge and confidence required to be fully autonomous when phase E2 science monitoring operations started and the development team safely transitioned to other activities.

The deployment phase took place at JPL in Pasadena, CA, far from Toulouse in France, where CNES is located. There was a significant number of team members involved in operations, so flying all of them to JPL was an expensive cost item for CNES, and for all its partners (IPGP, Imperial College and Oxford, ETH, MPS, etc.). Some team members had the possibility of spending a lot of time at JPL and volunteered to do so. Others preferred to limit their travels and the time away from home, and some were only required for a limited time and a specific step of the deployment (e.g. the CNES engineer who designed and provided the Frangibolt that latched SEIS to the lander deck was required to attend SEIS deployment operations). All team members were eventually able to participate in the operations at JPL in accordance with their wishes and required attendance. This meant that no one felt “forced” to travel to JPL or stay there for extended periods. Similarly, no one felt frustrated by not being able to attend JPL operations at JPL, because cost limitations were not an issue if the situation was not exploited. There was a good balance between team members and the situation settled well, which greatly contributed to the overall excellent team spirit needed during that critical phase of the mission.

The deployment phase also meant that many team members, about 10 to 12 simultaneously for just CNES (30 to 40 simultaneously in total with the SEIS partners and the science team), had to work together, mostly in the same room, for extended periods, dealing with sensitive operations that could become quite stressful. Rules and guidelines were defined and communicated to the team to avoid conflicts. These guidelines may have seemed obvious, but it was important that team management (especially the project manager and the operations manager) ensured that they were closely followed. Accommodation opportunities were offered (hotel, Airbnb, etc.)

and each team member was free to choose their preferred option, whether it was an apartment shared with colleagues, a hotel room outside of town, far away from the other team members, or an (expensive …) Airbnb close to JPL. This privilege was judged necessary by project management in order to ensure harmonious relations during this extended period of time. Among the rules laid down, the most important one was not blaming or judging in case of an operational mistake or an anomaly with the tools. This proved very useful when the SEIS GDS encountered technical issues during deployment operations. The choice had been made long before operations started to remotely access all CNES tools and software developed for SISMOC. These had been designed for a limited number of users and for handling a limited volume of data, based on the 38Mbits/sol guideline provided by the mission at JPL. In the end, the team grew larger and about two times more users than expected needed to access the tools during deployment. In addition, the daily data volume was sometimes 5 to 7 times greater than the 38Mbits/sol expected. What might appear to be good news for operations and science, proved to be very challenging for a ground segment whose infrastructure and remote access could not handle that many users or that amount of data. Inevitably performance deteriorated with time, and SISMOC performance became an issue about halfway through the deployment phase. The SISMOC team worked hard to ensure SISMOC performance issues did not have a major impact on ongoing operations, requiring a great deal of behind-the-scene workarounds, tool patches and server restarts. This included having on-call personnel available between Christmas and New Year's Day. Operations went well overall, but the period was quite challenging for the GDS teams. Late change requests are to be expected since operational and scientific requirements are subject to change—hopefully not dramatic change—when ORTs and real operations start. One option is therefore to maintain a high level of resources and funding for the GDS team in the last months before operations commence, to accommodate those changes. This is a costly option, but one that can definitely safeguard operations if the project can afford it. In the case of InSight and SEIS GDS, this was amplified by the fact that the operational paradigm changed with the capability to use the ESA TGO satellite as a relay satellite, greatly improving the available bandwidth. This added a significant load on the ground system and impacted its performance.

Another rule was communicated to the teams participating in the operations: make sure fatigue and stress did not affect the team behavior during operations. To do so, project management oversaw that enough rest periods were scheduled for the team members (no more than 4 consecutive workdays for instance), and that enough team members were physically present at JPL simultaneously to accommodate for fatigue or illness. This option was activated approximately 5 times over the entire period, and is likely to have helped prevent operational errors. The rule was a lesson learned directly from the Mars Science Laboratory 90 sols period back in 2012, where general fatigue among the ChemCam instrument team, due to limited resources and stressful operational shifts, led to operational errors.

In addition to serving as a very valuable stress-reduction opportunity due to being as representative of flight conditions as they could be, the ORTs also allowed the SEIS (CNES, IPGP et al.) and InSight (JPL) teams to get to know each other better before real operations started. This greatly improved the general operational flow and real-time discussions, especially between SEIS operators and mission management. The fact that the ORTs generated realistic data was also very valuable. The SEIS EM and simuSEIS simulator integrated into the InSight testbed at JPL proved very useful and should definitely be an example to follow for future missions, as a key to successful operations.

5 Tether Burial Activity

Since the beginning of the Insight mission, scientists noticed quickly random spikes—called glitches—in VBB velocity signal. Those glitches are considered as perturbations degrading the quality of the scientific signal, and despite attempts to filter them out mathematically, they are still a source of noise in the VBBs velocity signal, with up to 150 occurrences on a given sol. First analysis seems to point at the fact that the tether could be the where the glitches are originating from. As the tether is exposed to large temperatures variations (up to −80 °C in one hour when temperatures drop suddenly at night), a thermos-elastic effect is considered as the main explanation.

Early in the project, the idea was raised to use the scoop—located at the end of the robotic arm—to cover the tether with a regolith layer, but it has not been considered as a priority activity, mainly focused on HP3 deployment operations.

After 2 years of nominal mission, NASA took the opportunity of the mission extension to allow to study the feasibility of this approach. CNES (responsible for SEIS and WTS) and JPL (responsible for robotic arm and tether) created a dedicated tether burial working group, to share knowledge, hypothesis and ideas. This working group included operation engineers, geologists as well as weather specialists (the wind has to be taken into account when dropping material from a given height). CNES also relied on its SEIS QM, i.e. Qualification Model, physical installation next to the SISMOC mission center (mainly SEIS QM, spare WTS and tether QM in a sandbox) and a 3D printed scoop to set up tests, and proposed an optimized dump strategy consistent with the available resources and the scientific needs.

5.1 Tether Analysis

Thanks to recent images from ICC camera located below the lander, the tether shape has been analyzed. Pictures show that the tether was not completely in contact with the ground along its entire length, but had an elevated part (like a bridge) around 40 cm from the WTS. The shape has been determined from the tether's shadow. This bridge has been considered as a sensible area to cover, since if a heavy load was laid here, there was a risk that the vertical force applied on this bridge would be reflected by a horizontal thrust in the WTS and SEIS direction, leading to re-contact

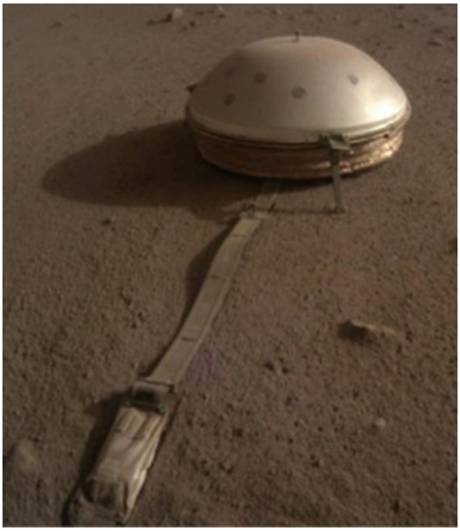


Fig. 23 Tether shape, with the bridge and its shadow

the 2 plates of the LSA (separated during SEIS deployment, see Sect. 4.2, in order to avoid mechanical perturbations from the lander). This risk has been considered as a major consequence, involving to propose a specific strategy to mitigate it (Fig. 23).

5.2 Covering Strategy Phase 1

Taking into account the tether shape constraints, the scoop capacity, the number of possible attempts, and the safety limitations (*Instrument Deployment System*, or IDS, team takes margins for the distance between the scoop and the tether), the first test campaign was focused on covering the part between the WTS and the bridge, close to the instrument.

CNES team performed several tests in the sand box, with WTS, tether and a 3D printed scoop, taking hypothesis about regolith volume in the scoop, and respecting the security distance between scoop and tether/WTS. Different configurations were tested, and all dumps were analyzed and characterized (shape, height, width).

To improve covering area close to the WTS without taking any risk with the scoop, CNES also experienced a deposit on the WTS, regolith sliding along the WTS to fall almost plumb. It is interesting to note that when this idea has been executed on Mars, the unexpected picture of WTS partially cleaned thanks to regolith dump spawned the idea of cleaning solar panels with a similar technic.

Finally, the proposed strategy was to build 5 piles, some made of 1 scoop volume, some made of 2 scoop volumes.

The piles were named after Pyrenean peaks alphabetically to indicate the order of the dumps (Figs. 24 and 25).

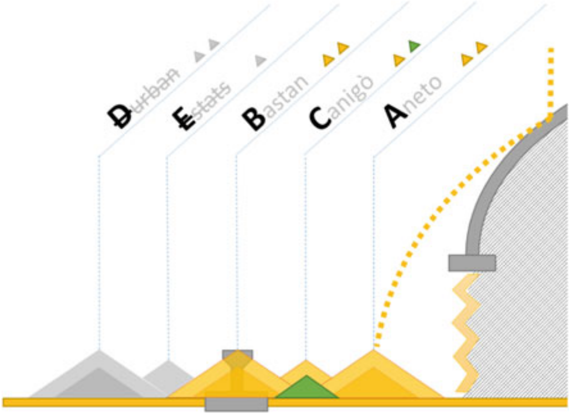


Fig. 24 Dumps strategy agreed by the working group

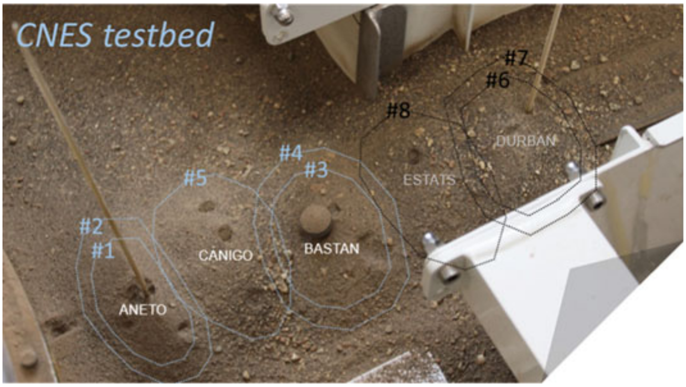


Fig. 25 Dumps results in CNES premises

5.3 *Scrapping*

Dumping regolith with the scoop requires to collect material from the martian surface. A dedicated area, between HP3 and SEIS has been proposed by IDS team and agreed by mission management, since considered not risky and easier to reach with the arm. Before collecting regolith, scrapping activities are mandatory, building piles where the scoop will shovel regolith. This task was entirely managed by the IDS team of JPL engineers (Fig. 26).

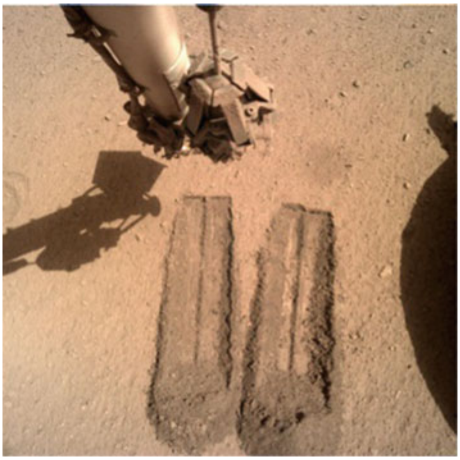


Fig. 26 Scoop scrapped two times

5.4 *Dumps*

Once the overall dump strategy had been agreed by the working group and mission management, it was up to CNES to provide targets to the IDS team at JPL. This process led to agree on a coordinates system with IDS, to ensure having the same reference between CNES and JPL. After each dump, stereo images were taken by the IDC camera located on the arm, providing a 3D model helping to estimate dumps and pile accuracy. Tether burial working group members also managed to provide differences between 3D models, highlighting new dumps and giving accurate value of scoop content (Fig. 27).

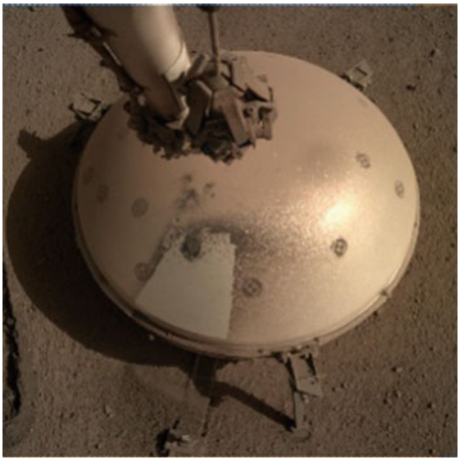


Fig. 27 Sol 816, first dump on WTS

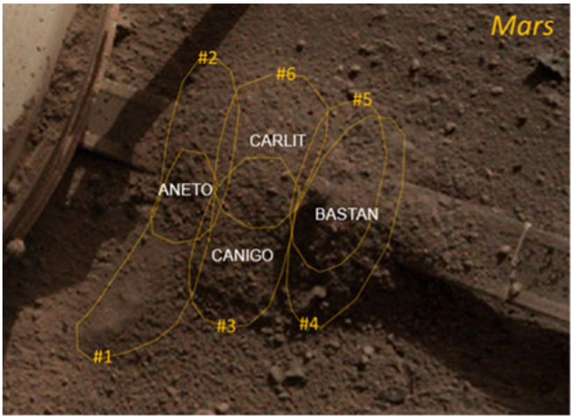


Fig. 28 Six dumps performed on Mars

- First dump on “Aneto” target (sol 816) created a pile slightly on the left with regards to the tether, but close to the expectation. As expected, the regolith sled on the WTS and felt in the area to cover. An unexpected effect was clearly visible on the picture, showing that the deposit cleaned dust off the WTS in its path.
- A second deposit on “Aneto” has been performed sol 850, at a later time to accommodate reduced wind periods, avoiding dispersion of the lighter particles from the regolith.
- Dump #3 targeted “Bastan”, on sol 856. It was the first direct deposit on the tether, meaning more uncertainty on the precision. IDS finally reached the “Canigo” target, with a slight offset on the left.
- Dump #4 targeted “Bastan” again, on sol 863. Improving accuracy thanks to previous dump bias understanding, this deposit reached his target.
- Dump #5 targeted again “Bastan” on sol 870, in a 2 dumps strategy, covering the pining-mass. This dump has been performed in windy conditions, as part of study for potential solar array cleaning operations by saltation.
- Dump #6, on sol 877, was more pragmatic, aiming a new target completing Canigo pile on the right, called “Carlit”.

After 61 sols of operations, 6 dumps were performed, covering the expected part between WTS and pining-mass. The weight put on this section allows the second phase of burial to begin: the bridge section (Fig. 28).

5.5 *Priority to Energy*

The first dump on WTS and the cleaning effect of regolith inspired JPL team to consider using this phenomenon to clean the solar arrays. Indeed, after 2 years on Mars, solar arrays are covered with a considerable dust layer, decreasing the available energy for operations. JPL came up with a proposition to use the saltation effect to

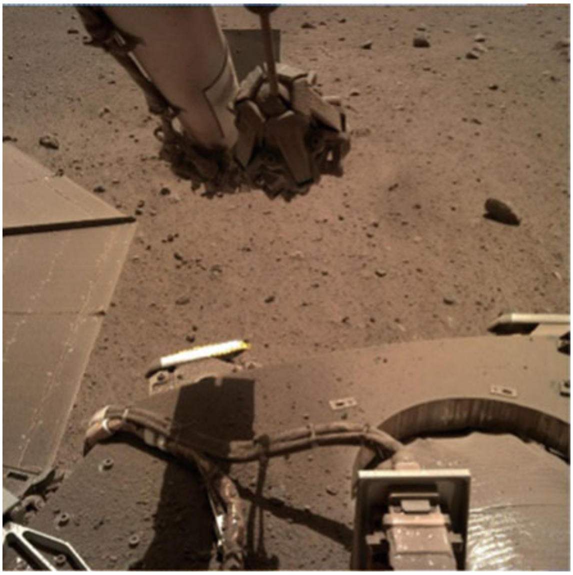


Fig. 29 After regolith deposit on the deck for solar array cleaning

clean the solar arrays, using windy timeframe to drop regolith close to the solar arrays, creating conditions for biggest particles driven by the wind to tear smaller particles from the solar array (Fig. 29). This operation, primordial to extend the mission lifetime, was considered higher priority than the tether burial activity until the solar conjunction set to occur in October 2021. It has been agreed by all that targets Durban and Estats are not required during the first dump campaign.

5.6 Second Phase Preparation

Anticipating the necessity to provide new targets quickly, CNES experienced the second part of tether burial (bridge section) with the same approach (sand box, scoop dumps, observations). The tether available in CNES premises has been shaped to fit as much as possible with the one on Mars, to be representative in term of tether bending and sand dynamic. The conclusion is a 16 dumps strategy, starting from the pinning mass (see Sect. 4.2), in direction of the lander. This strategy involves a lot of regolith acquisition, and IDS team is working on an additional acquisition area to match the needs.

5.7 Tether Burial Activity Synthesis

From an engineering point of view, a first layer of regolith has been successfully put on the tether area between the WTS and the pinning mass. This objective has been reached, taking a significant amount of time due to 2 factors:

- Uplink cycles: with 1 uplink per week, JPL and CNES can only program 1 acquisition and 1 dump per week.
- Unexpected issues: the robotic arm experienced two benign safings during acquisitions. IDS team had to understand the root cause before unsafing the arm and reprogram its activities.

The science team of mission seismologists are interested in evaluating the impact on VBBs velocity signal, put in place a glitches follow-up, and updated the graph regularly. It shows a significant decrease at the beginning of the tether burial activity, then glitches recovered their initial level, showing a mitigated effect of this first phase. The first layer is relatively thin (no more than 1 cm), and a thicker layer could have more positive impact on glitches decrease.

6 Conclusion

The spectacular Mars landing and instrument deployment operations that followed, led to the successful commissioning of the SEIS seismometer on Mars, as well as the APSS weather station. They are still collecting high-quality seismic and atmospheric data on Mars to this day. The first Marsquake was detected shortly after, in April 2019. The success of the deployment operations owes a lot to the long and meticulous preparation of that phase and the associated training process, as well as the excellent collaboration between all teams. As always, the human factor is key, and that once again proved to be right for InSight. This outstanding technical journey was also an amazing human adventure for the teams involved.

Optimal Reaction Wheel Control with Stiction and Resonance Avoidance

Tianyi Zhang and Philip Ferguson

Abstract Reaction wheels are commonly used to provide precision control for spacecraft that require fine pointing. However, vibration disturbances from the mechanical structure of reaction wheels such as stiction and resonance can cause attitude errors that are difficult to compensate. By using an optimal reaction wheel control that keeps the wheel speeds away from the disturbance-related wheel speeds, the effects of these disturbances can be minimized. This paper proposes an optimal control algorithm for reaction wheel assemblies with four or more wheels that minimizes the impact of stiction and resonance by implementing an intelligent wheel torque distribution algorithm. The null space provided by redundant wheels is utilized as an extra degree of freedom for controlling the wheel speeds. The formulated optimization problem is solved by Particle Swarm Optimization (PSO) and the numerical simulations of a 4-wheel spacecraft are carried out to validate the proposed algorithm.

Keywords Torque distribution · PSO · Wheel torque · Optimal control · Satellite

Acronyms/Abbreviations

RWA	Reaction Wheel Assembly
PSO	Particle Swarm Optimization
TRM	Track Rate Mode

T. Zhang (✉) · P. Ferguson
STARLab, University of Manitoba, Winnipeg, MB R3T 5V6, Canada
e-mail: zhangt17@myumanitoba.ca

P. Ferguson
e-mail: Philip.Ferguson@umanitoba.ca

C. Cruzen et al. (eds.), *Space Operations*, Springer Aerospace Technology,
https://doi.org/10.1007/978-3-030-94628-9_30

1 Introduction

For space missions that image celestial bodies or ground targets, high precision and stability for attitude control is required. For example, NEOSSat (Near Earth Object Surveillance Satellite) is used to search, track and image resident space objects (usually spent satellites) around the Earth. One of the observing modes for NEOSSat is TRM, in which the satellite's slew rate and direction matches that of the target object. In TRM, the camera exposes continuously while slewing, so that the target object is a bright dot in the picture while the stars in the background appear as streaks [1]. Using the direction and length of the star streaks, in combination with guide star catalogues, the orbit of the target object can be calculated. Therefore, TRM requires high agility and stability of the attitude control system when imaging target objects [1]. Any attitude disturbances during the tracking slew would result in a blurry image and incorrect tracking results.

While many satellites include redundant reaction wheels in case of one wheel fails, redundant wheels can also enable wheel speed control for the purposes of disturbance avoidance. In this paper, the null space matrix is employed to utilize this redundancy. The torques in the null space always result in a zero net torque in the body frame such that the desired slew is not affected while enabling manipulation of the wheel speeds. The null space torque components work as auxiliary torques such that the spacecraft achieves active control over wheel speeds without disturbing the original attitude dynamics.

On this basis, this paper proposes an emerging optimization problem to solve for the optimal set of null space torques for a given slew, where the optimal null space torques are designed to keep the wheel speeds away from zero-crossing and resonant speeds. Particle Swarm Optimization (PSO) is implemented and the computational results validate this algorithm.

This paper is organized as follows: Sect. 2 briefly reviews the related literature, Sect. 3 provides the problem formulation, Sect. 4 presents the mathematical model of satellite's attitude dynamics and the test results, and Sect. 5 concludes the paper and summarizes the contributions.

2 Related Research

For RWA consisting of more than three reaction wheels, the wheel torque distribution is not unique for a given commanded body torque vector. The simple pseudo inverse algorithm is one of the most common torque distribution algorithms which minimizes the l_2-norm of wheel torques [2]. Markley et al. [3] present a min–max algorithm (also known as l_∞-norm optimal method) where the maximum of the torques among wheels is minimized. Lim and Miotto [2] implemented linear programming to maximize the time between two momentum off-loadings. Mikihiro [4] proposed a hybrid algorithm that utilizes both l_2-norm optimal method and l_∞-norm optimal method.

Schaub and Lappas [5] proposed a power-optimal torque distribution algorithm that reduces the overall RWA power consumptions. Cao et al. [6] present an algorithm that maximizes the allowable slew rate of the spacecraft.

While the previously mentioned papers provide a means of adjusting the instantaneous torque mapping to minimize an objective function, none of them address the disturbances such as stiction and resonance which are induced when wheel speeds approach specific values. Rigger [7] presented a control algorithm that avoids stiction and maximum speed limit by incorporating with null space torque component, but the determination of the null space component is not optimal as the algorithm is implemented only if the wheel approaches certain pre-defined threshold value. Kron et al. [8] designed a PID controller to utilize the null space torque in terms of reaction wheel momentum management, however, as a real-time control algorithm, it cannot guarantee the optimality. This paper explores the optimal null space torque components for a given desired slew by planning the entire wheel speed profile for one or several consecutive.

3 Problem Formulation

3.1 Attitude Dynamics

The mathematical model of attitude dynamics can be expressed as:

$$I_{sc}\dot{\omega}^B + \omega^B \times \left(I_{sc}\omega^B + {h_w}^B\right) + \tau_d = {\tau_w}^B \tag{1}$$

$${\tau_w}^B = -{\dot{h}_w}{}^B = C_{con}{\tau_w}^{RW} \tag{2}$$

where I_{sc} is the spacecraft's 3-by-3 inertia matrix, ω^B is the body rate of the satellite, τ_d is the disturbance torque, and ${h_w}^B$ is the angular momentum of reaction wheels in the body frame. For this paper, a numerical test has been done for a 4-wheel RWA with pyramid configuration whose configuration matrix is:

$$C_{con} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{-1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \tag{3}$$

The moment of inertia of the satellite is set to be $I_{sc} = diag[4, 3, 2]$ kg m^2. The reaction wheels are assumed to be identical with a moment of inertia of 8×10^{-4} kg m^2 each.

3.2 Null Space Torque Component

The torque distribution algorithm for a 3-wheel RWA is found by taking the inverse of the 3-by-3 configuration matrix C_{con}, where the columns of C_{con} indicate the orientations of reaction wheels in the body-fixed frame. The relationship between body torques and wheel torques can be expressed by Eq. (4).

$$ {\tau_w}^B = C_{con} {\tau_w}^{RW} \tag{4} $$

where ${\tau_w}^B$ is a $[3 \times 1]$ vector that contains the desired control torques (provided by reaction wheels) in the body-fixed frame, and ${\tau_w}^{RW}[n \times 1]$ contains the final commanded torques in the reaction wheel frame.

For RWAs consisting of n wheels ($n > 3$), C_{con} is no longer a square matrix and is not invertible. The pseudo inverse (sometimes referred to as the "right Moore–Penrose inverse" [9]) provides an analytical solution that minimizes the l_2-norm of the distributed wheel torques. The pseudo inverse can be stated simply as:

$$ {C_{con}}^{\dagger} = {C_{con}}^T \left(C_{con} {C_{con}}^T\right)^{-1} \tag{5} $$

where ${C_{con}}^{\dagger}$ $[n \times 3]$ is the pseudo inverse of the configuration matrix. In order to leverage the redundancy provided by the redundant wheels, the null space matrix (N) of C_{con} is introduced, such that:

$$ C_{con} N = 0 \tag{6} $$

The null space matrix N is a $[n \times m]$ matrix where $m = n - r$ indicates the degree of redundancy, and r is the rank of C_{con}. Note that r is equal to 3 for RWA with at least 3 noncoplanar wheels. In this paper, the null space torque is defined as:

$$ \tau_{null} = N A \tag{7} $$

where A is a $[m \times 1]$ vector that consists the null space scaling parameters, as defined in [8]. The proposed torque distribution algorithm in this paper augments the pseudo inverse algorithm with the null space torque. The pseudo inverse algorithm provides the basis of torque distribution that guarantees the desired torque is properly distributed and the null space torque is used as an auxiliary torque for controlling wheel speeds without affecting the commanded torque. The null space torques can theoretically take arbitrary values without affecting the desired torque in body-fixed frame but must ensure that the final commanded torques ${\tau_w}^{RW}$ do not exceed the reaction wheel's torque limit. By strategically choosing the values of τ_{null}, we can achieve active control over the wheel speeds such that the undesired speeds are avoided. The overall torque mapping algorithm can thus be stated as:

$$ {\tau_w}^{RW} = {C_{con}}^{\dagger} {\tau_w}^B + N A \tag{8} $$

Since the null space matrix N for a given C_{con} is fixed, the tuning of τ_{null} can only be done by manipulating the scaling vector A.

3.3 Problem Formulation and the Implementation of PSO

For a given desired slew with known initial wheel speeds, the propagation of wheel speeds and satellite body rates can be calculated by incorporating the chosen torque mapping algorithm with the overall attitude dynamic model and integrating it forward in time. With the wheel speed propagation, the optimization problem can be formulated to determine the optimal null space torques:

$$\min_{A(t)} J = \int_{t0}^{tf} \sum_{i=1}^{n} (C_1 e^{-C_2(|\omega_i(t)|-\omega_s)^2} t)dt \tag{9}$$

$$s.t. \begin{cases} \dot{\omega}(t) = {I_{wh}}^{-1}[{C_{con}}^{\dagger}{\tau_w}^B + N \cdot A(t)] \\ -\tau_{\max} \le {C_{con}}^{\dagger}{\tau_w}^B + NA(t) \le \tau_{\max} \\ -\omega_{\max} \le \omega(t) \le \omega_{\max} \end{cases} \tag{10}$$

where t_0 and t_f are the start/end time of the slew, respectively, $\dot{\omega}(t) = [\dot{\omega}_1, \dot{\omega}_2 \ldots \dot{\omega}_n]^T$ contains the wheel accelerations, $\tau_{\max}$ and $\omega_{\max}$ are the maximum available torque and wheel speed, respectively, ${I_{wh}}^{-1}$ is the inverse of the diagonal matrix that consists of reaction wheel inertias, and C_1 and C_2 are pre-defined constants. The exponential term $C_1 e^{-C_2(|\omega_i(t)|-\omega_s)^2}$ is a bell-shaped function that has a maximum at $|\omega_i(t)| = \omega_s$, such that the cost function approaches its maximum when wheel speed is getting close to ω_s, the absolute value of the wheel speed that needs to be avoided. In this paper, ω_s has two values: $\omega_s = 0$ (for stiction avoidance) and $\omega_s = \omega_n$ (for resonance avoidance) where ω_n is the natural frequency of the reaction wheel's fixture. The constant C_1 decides the maximum value of the exponential penalty function, which is recommended to fall within the range [5, 10]. The constant C_2 is suggested to be on the order of 10^{-1}. The exponential term is multiplied by time t because in some cases the undesired wheel speed ω_s cannot be avoided, and in this situation we want it to occur early in the slew such that there is enough time to damp out the effects of disturbances before the end of the slew.

The solution of this optimization problem yields the time series of the optimal null space scaling vector $A(t)$. However, $A(t)$ cannot be solved as a continuous function of time because of the discrete, nonlinear objective function. Therefore, for a "bang-off-bang" type of slew, $A(t)$ is defined by three values:

$$A(t) \begin{cases} A_1, \text{ ramp up period} \\ A_2, \text{ coasting period} \\ A_3, \text{ ramp down period} \end{cases} \tag{11}$$

Note that each of A_1, A_2, and A_3 are each an $[m \times 1]$ vector, solved by PSO. As PSO is an intrinsically non-constrained search algorithm, constraints must be incorporated as additional penalty terms to the objective function [10].

The constraints on the maximum allowable wheel torques/speeds can be converted into:

$$g_1(t) = -({C_{con}}^{\dagger}{\tau_w}^B + NA(t)) - \tau_{\max} \leq 0 \tag{12}$$

$$g_2(t) = {C_{con}}^{\dagger}{\tau_w}^B + NA(t) - \tau_{\max} \leq 0 \tag{13}$$

$$g_3(t) = -\omega(t) - \omega_{\max} \leq 0 \tag{14}$$

$$g_4(t) = \omega(t) - \omega_{\max} \leq 0 \tag{15}$$

and

$$q_m(t) = \max\{0, g_m(\mathrm{t})\}, m = 1, 2, 3, 4 \tag{16}$$

After the constraint terms are added to the original objective function, Eq. (9) becomes:

$$\min_{A(t)} J = \int_{t_0}^{t_f} \left\{ \sum_{i=1}^{n} (C_1 e^{-C_2(|\omega_i(t)| - \omega_s)^2} t + + \sum_{i=1}^{n} \sum_{m=1}^{4} C_3 q_m(t)^{q_m(t)} \right\} dt \tag{17}$$

Note that if a constraint is violated, g_m becomes positive and $q_m = g_m$, thus additional penalties are added. C_3 shall be large enough (i.e., several orders of magnitude larger) compared to C_1 and C_2 as the constraints on maximum wheel torque/speed are not allowed to be violated. Here C_3 is recommended to be on the order of 10^4. The number of particles in the swarm needs to be properly tuned to balance between the computational efficiency and to limit the possibility of PSO returning a local (as opposed to global) minimum. In this paper, the swarm size was fixed at 300.

4 Tests and Results

4.1 *Single Slew Tests*

In the numerical tests, a 60-s slew was planned where the desired quaternion and initial quaternion ware ${q_{des}}^T = [0.1660, 0.8301, -0.3320, -0.4161]$ and ${q_{ini}}^T = [0, 0, 0, 0]$, respectively. The initial wheel speeds were $[100, 100, -100, -100]$ rad/s. For a 4-wheel RWA, the null space scaling parameter A is a scalar and was solved as stated in Sect. 3. Four tests have been done to illustrate the effectiveness of the proposed algorithm on wheel speed control. The first test (test 1) used only the pseudo inverse algorithm for torque mapping, and the resultant wheel speeds are shown in Fig. 1 as a baseline with to compare the new optimization method. The zero-crossing avoidance algorithm and resonance avoidance algorithm was applied to test 2 and test 3 respectively. Note that in test 3 the resonance speed was assumed to be equal to 300 rad/s, such that $\omega_s = \omega_n = 300$ in Eq. (17). The resulting wheel speeds for test 2 and test 3 are shown in Fig. 2 and Fig. 3 respectively. Test 4 included both zero-crossing and resonance avoidance simultaneously and the resulting wheel speeds are indicated in Fig. 4. Table 1 summarizes the null space scaling parameters as found using PSO for each test.

With the given initial wheel speeds and the desired quaternion, there were four zero-crossings and two resonance crossings when using the pseudo inverse algorithm (as shown in Fig. 1). The zero-crossings were eliminated in test 2 where the zero-crossing avoidance algorithm was implemented, but the resonance crossings were still present. As shown in Fig. 2, the resonance crossings have been removed, but the zero-crossings were present.

In test 4, the cost function penalized both $\omega_s = 0$ and $\omega_s = 300$, so that both stiction and resonance were penalized. As indicated in Fig. 4, the algorithm tried to

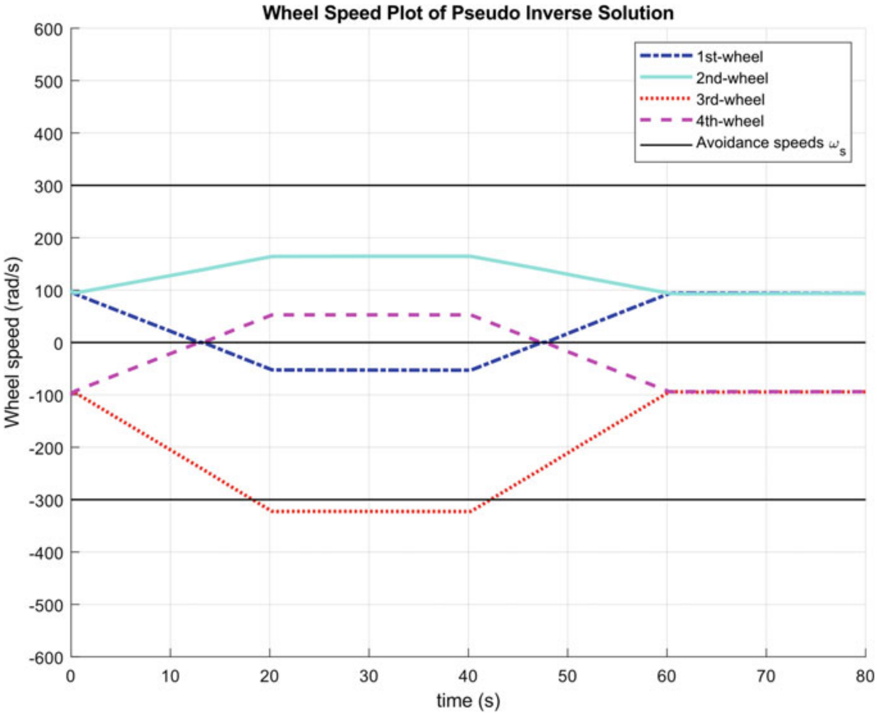


Fig. 1 Wheel speeds of test 1: pseudo inverse algorithm

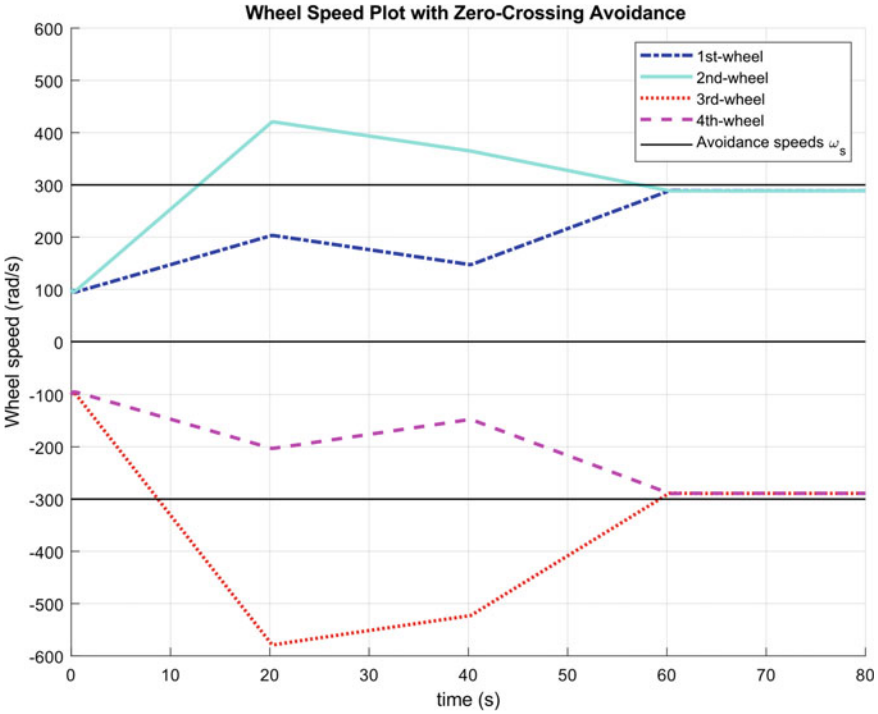


Fig. 2 Wheel speeds of test 2: zero-crossing avoidance algorithm

limit the wheel speeds within the 'safe zone' that is between the zero speed and the resonance speed. However, in this case the resonance on the 3rd wheel could not be avoided but it happened with 13 s left before the slew end so there likely would have been enough time for the controller to damp out the effects of the resonance.

In tests 2, 3 and 4, the runtimes of the optimizer (PSO) were 36.5 s, 43.6 s, and 45.3 s respectively. Therefore, this algorithm is feasible for on-board computation and would not require any inputs from the ground.

4.2 Monte Carlo Test for Successive Slews

To further evaluate the effectiveness of the stiction and resonance avoidance on settling time reduction, a high-fidelity attitude control simulator with reaction wheel as actuators was created, such that the attitude errors due to reaction wheel stiction and resonance were properly simulated. The reaction wheel stiction and resonance were modelled by Eqs. (18) and (19), respectively.

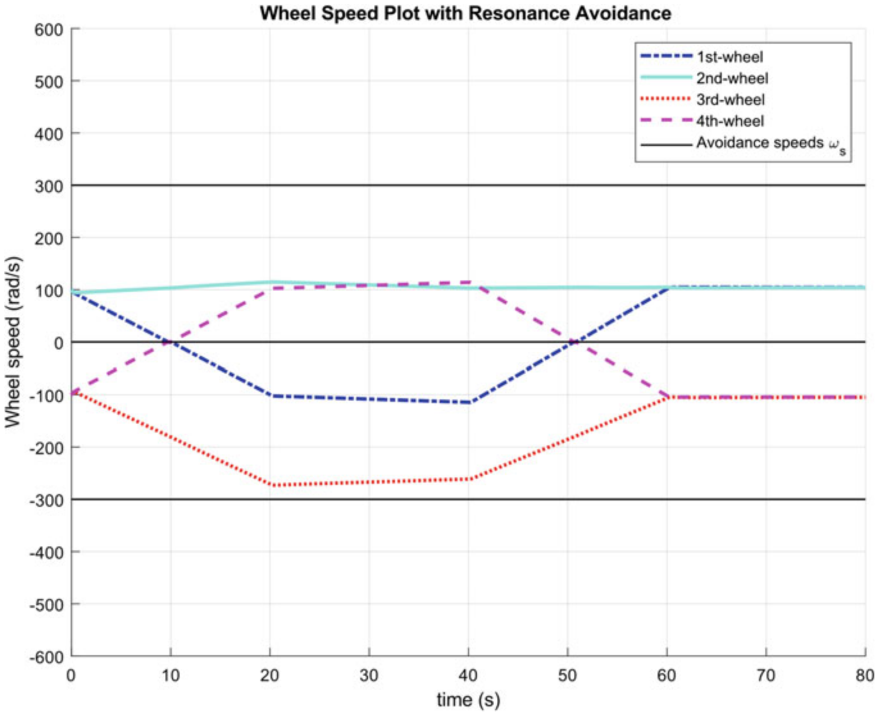


Fig. 3 Wheel speeds of test 3: resonance avoidance algorithm

$$\tau_{stic} = 0.003\frac{\omega(t)}{|\omega(t)|} \tag{18}$$

$$\tau_{res} = 5e^{-(|\omega(t)|-\omega_n)^2}C_{imb}\omega(t)^2\sin(2\pi h\omega(t)t + \alpha) \tag{19}$$

where C_{imb} is the amplitude of the reaction wheel jitter due to the reaction wheel imbalance, h is the wheel-dependent harmonic number and α is the random phase change within $[0, 2\pi]$ [11]. The reaction wheel jitter is amplified when the wheel speed comes to the resonance speed ω_n.

The test assumed that the spacecraft slewed to 5 consecutive targets (therefore requiring 5 successive slews), and each slew started only once the previous slew had properly settled (such that the pointing error dropped below 0.001 rads). The pointing errors due to the disturbances during the slew were compensated by a PD pointing controller. As expected, reaction wheel disturbance torques incurred during the slew induced pointing errors that affect the settling time after the slew. In this test, the RWA was assumed to be identical to the previous tests and the configuration was expressed by Eq. (3). The initial speeds of all reaction wheels were assumed to be identical, which was equal to 5 rad/s.

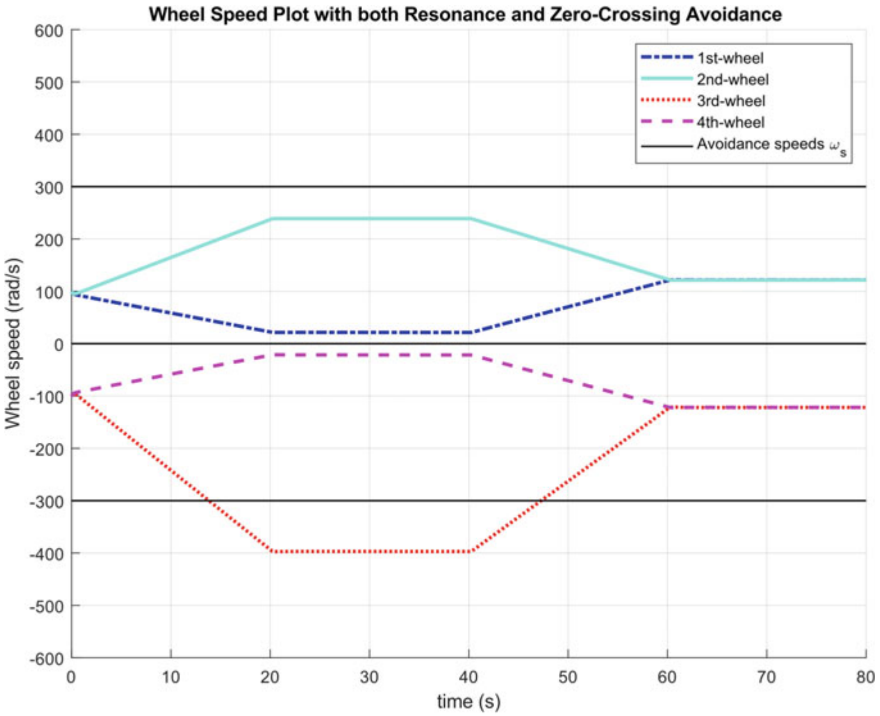


Fig. 4 Wheel speeds of test 4: both resonance and zero-crossing avoidance

Table 1 Null space scaling parameters for each test

	Test 1	Test 2	Test 3	Test 4
A_1	0	−0.0216	0.0042	−0.0064
A_2	0	0.0047	0.0010	0
A_3	0	0.0005	−0.0060	0.0040

The objective function was modified to enable a five-slew sequence optimization. The modified objective function is stated as:

$$\min_{A(t)} J = \int_{t_p}^{t_{p+1}} \left\{ \sum_{i=1}^{n} (C_1 e^{-C_2(|\omega_i(t)|-\omega_s)^2}(t - t_p) + + \sum_{i=1}^{n} \sum_{m=1}^{4} C_3 q_m(t)^{q_m(t)} \right\} dt, \quad p = 1, 2, \ldots, 5 \tag{20}$$

where t_p is the start time of the pth slew. As the slews were assumed to be consecutive, the end time of the previous slew was identical to the start time of the next slew. Both stiction and resonance avoidance were incorporated in the tests such that $\omega_{s,1} = 0$

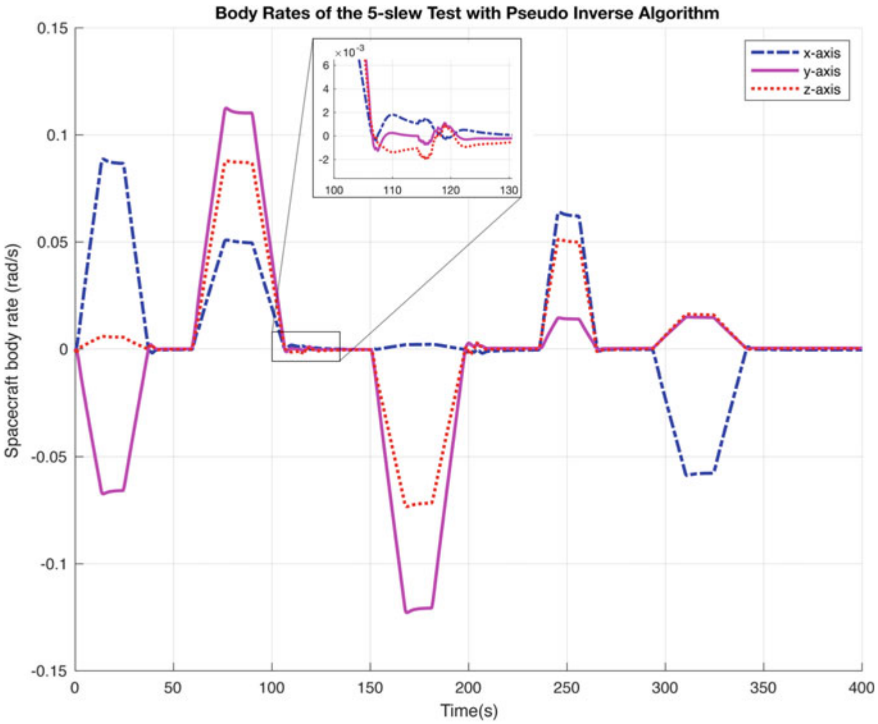


Fig. 5 Body rate profile for five slews using the Pseudo inverse algorithm (both stiction and resonance are present)

and $\omega_{s,2} = \pm 300$ rad/s. Figures 5 and 6 show the body rates of the spacecraft for the five given slews. Figure 5 shows the results from the Pseudo inverse algorithm and Fig. 6 shows the results from the stiction and resonance avoidance algorithm. The wheel speed plots of both cases are presented in Figs. 7 and 8, respectively. The stiction and resonance avoidance algorithm was implemented as introduced in previous section, so that there were three null space scaling parameters for each slew (15 parameters in total for five slews).

As indicated in Fig. 7, at the given initial speed of 5 rad/s, the wheel speeds resulting from the Pseudo inverse algorithm crossed zero near the beginning of every slew, which induced pointing errors (for example, at the time of 108 s in Figs. 5 and 7) and led to considerable settling time. Figure 8 shows the wheel speeds with stiction and resonance avoidance such that the wheel speeds were almost entirely kept within the "safe zone" where neither stiction nor resonance were present. In this scenario, the definition of the "safe zone" differed from test 4. Here, the "safe zone" was in-between the resonance speed and the maximum speed. Overall, the five sample slews settled at the time of 360 s for Pseudo inverse algorithm and 315 s for the stiction and resonance avoidance algorithm. Therefore, the overall settling time was reduced by 45 s for the same set of slews.

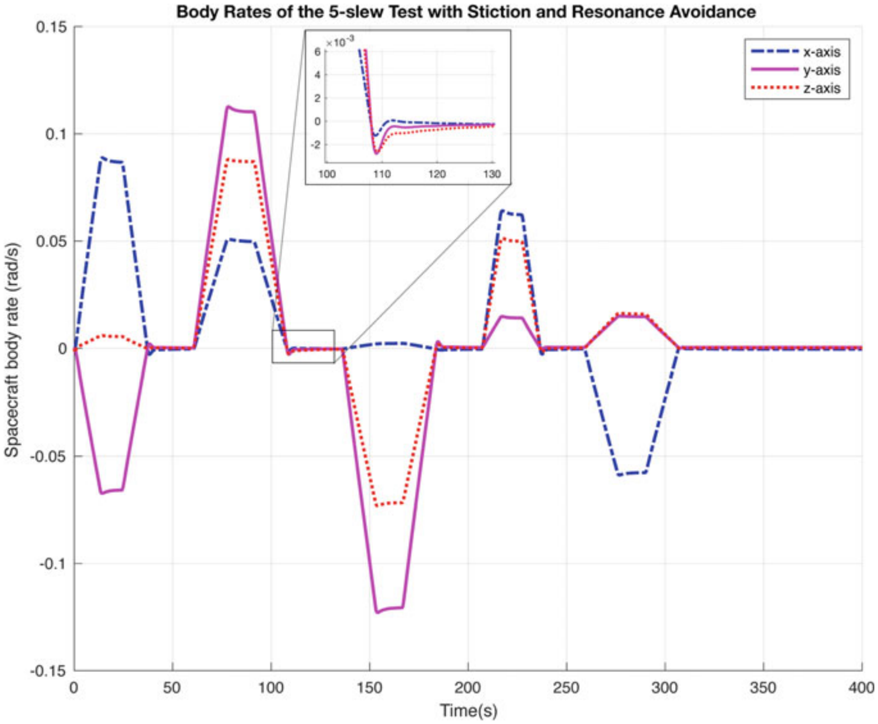


Fig. 6 Body rate profile for five slews using the stiction and resonance avoidance algorithm

A Monte Carlo test was carried out such that the five-slew test was repeated by randomizing 100 sets of five-slew profiles to further validate the proposed stiction and resonance avoidance algorithm. The initial wheel speeds for all tests were assumed to be identical (5 rad/s). The Euler axis of every single slew was a randomized unit vector and the Euler slew angle was randomized between $[0, \pi]$ rad. Again, all the slews in the test were "bang-off-bang" type such that the accelerating and decelerating stage had the same length of time. The length of each single slew was randomized on the range of [25, 50] s. The result of the Monte Carlo test indicated that the average computation time using the PSO optimizer was 62 s, and the average settling time reduction when compared with the pseudo-inverse algorithm was 56.3 s (for five slews).

5 Conclusion

This paper proposed an optimal control algorithm for an RWA with redundant wheels, such that zero wheel speed and resonance wheel speed were avoided as much as possible. This algorithm manipulated the torques in the null space as an extra degree

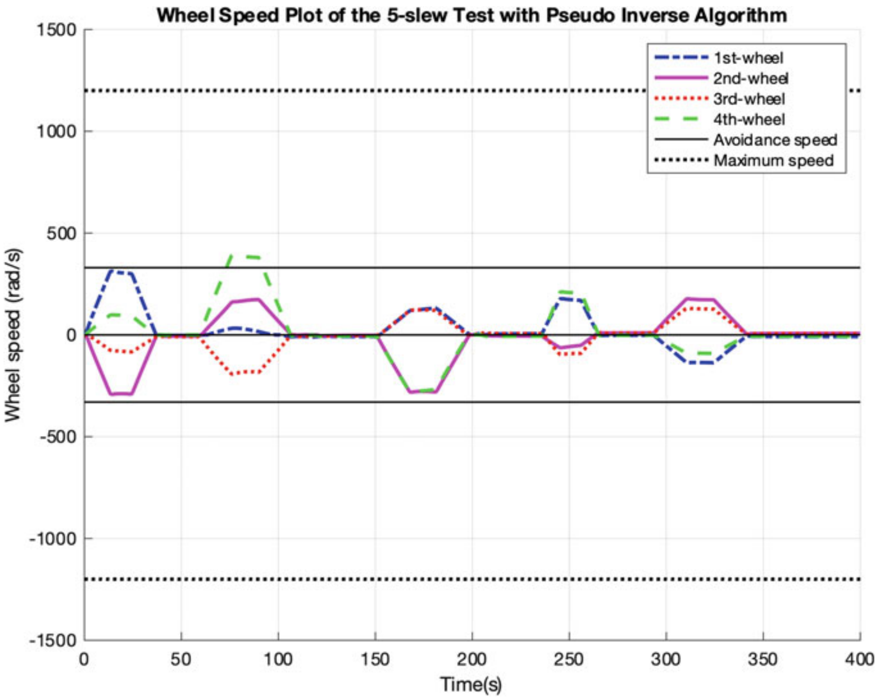


Fig. 7 Wheel speed plot for five consecutive slews using the Pseudo inverse algorithm

of freedom of wheel speed control without affecting the desired slew. The optimal null space torques were determined via the corresponding optimization problem, which was solved by PSO.

The results from the numerical simulations validated the proposed algorithm on zero-crossing avoidance and resonance avoidance. In the sample tests, the zero-crossings or resonances were entirely eliminated by solely implementing the zero-crossing avoidance or resonance avoidance algorithm individually, compared to the pseudo inverse solution. When both stiction and resonance were penalized as presented in test 4, the algorithm adjusted wheel speeds in a compromised manner to minimize the effects of both disturbances. The resultant wheel speeds were mostly maintained within the "safe zone" between the zero-speed and the resonance speeds.

A closed-loop attitude control simulator as well as a high-fidelity reaction wheel model were developed to explore the effects of reaction wheel disturbances on the spacecraft's attitude control agility. A Monte Carlo test was carried out such that random slews were tested to demonstrate how much faster a spacecraft using the proposed wheel speed management algorithm could slew to five successive targets compared to the Pseudo inverse algorithm. The results indicated that the average settling time reduction for five successive slews was 56.3 s.

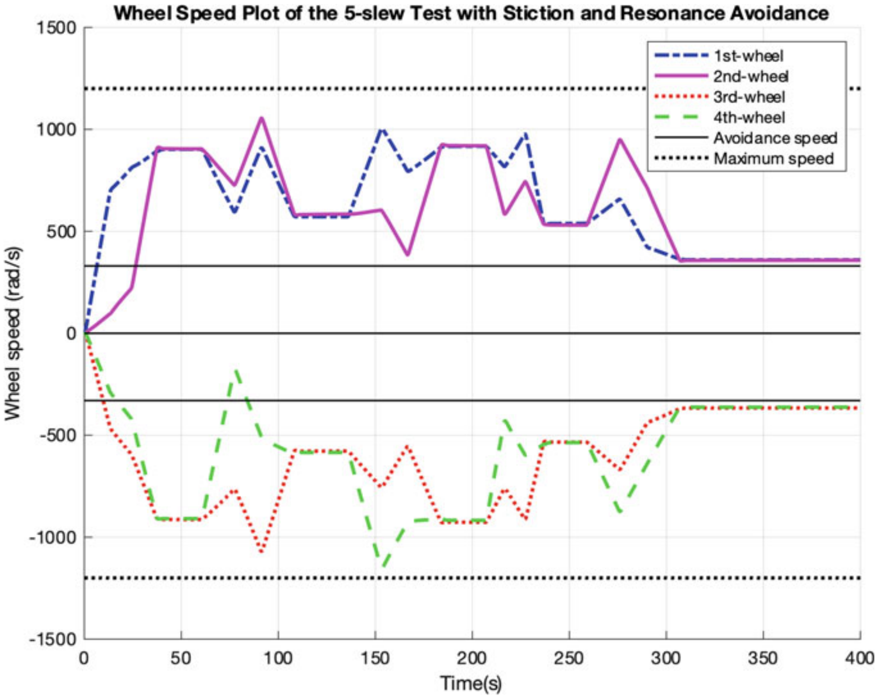


Fig. 8 Wheel speed plot for five consecutive slews using the stiction and resonance avoidance algorithm

References

1. Wallace B et al (2004) The near earth orbit surveillance satellite (NEOSSat). In: International Astronautical Federation—55th international astronautical congress 2004, vol 6, no 613, pp 3750–3758. https://doi.org/10.1117/12.567077
2. Sungyung L, Miotto P (2006) Actuator allocation algorithm using interior linear programming. In: Collection of technical papers—AIAA guidance, navigation, and control conference 2006, vol 2, no August, pp 1015–1029. https://doi.org/10.2514/6.2006-6186
3. Markley FL, Reynolds RG, Liu FX, Lebsock KL (2010) Maximum torque and momentum envelopes for reaction-wheel arrays. J Guid Control Dyn 33(5):1606–1614. https://doi.org/10.2514/1.47235
4. Sugita M (2017) Torque distribution algorithm for effective use of reaction wheel torques and angular momentums. Acta Astronaut 139:18–23. https://doi.org/10.1016/j.actaastro.2017.06.014
5. Schaub H, Lappas VJ (2009) Redundant reaction wheel torque distribution yielding instantaneous L 2 power-optimal attitude control. J Guid Control Dyn 32(4):1269–1276. https://doi.org/10.2514/1.41070
6. Cao X, Yue C, Liu M, Wu B (2016) Time efficient spacecraft maneuver using constrained torque distribution. Acta Astronaut 123:320–329. https://doi.org/10.1016/j.actaastro.2016.03.026
7. Rigger R (2010) On stiction, limit and constraint avoidance for reaction wheel control. In: SpaceOps 2010 conference, no April, pp 1–8. https://doi.org/10.2514/6.2010-1931

8. Kron A, St-Amour A, de Lafontaine J (2014) Four reaction wheels management: algorithms trade-off and tuning drivers for the PROBA-3 mission. In: IFAC proceedings volumes (IFAC-PapersOnline), vol 19, pp 9685–9690. https://doi.org/10.3182/20140824-6-za-1003.00604
9. Dresden A (1920) The fourteenth western meeting of the American mathematical society. Bull Am Math Soc 26(9):385–396. https://doi.org/10.1090/S0002-9904-1920-03322-7
10. Parspoulos KE, Vrahatis MN (2002) Particle swarm optimization method for constrained optimization problems. In: The second Euro-international symposium on computational intelligence, pp 214–220
11. Masterson RA (1999) Development and validation of empirical and analytical reaction wheel disturbance models. MS thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA