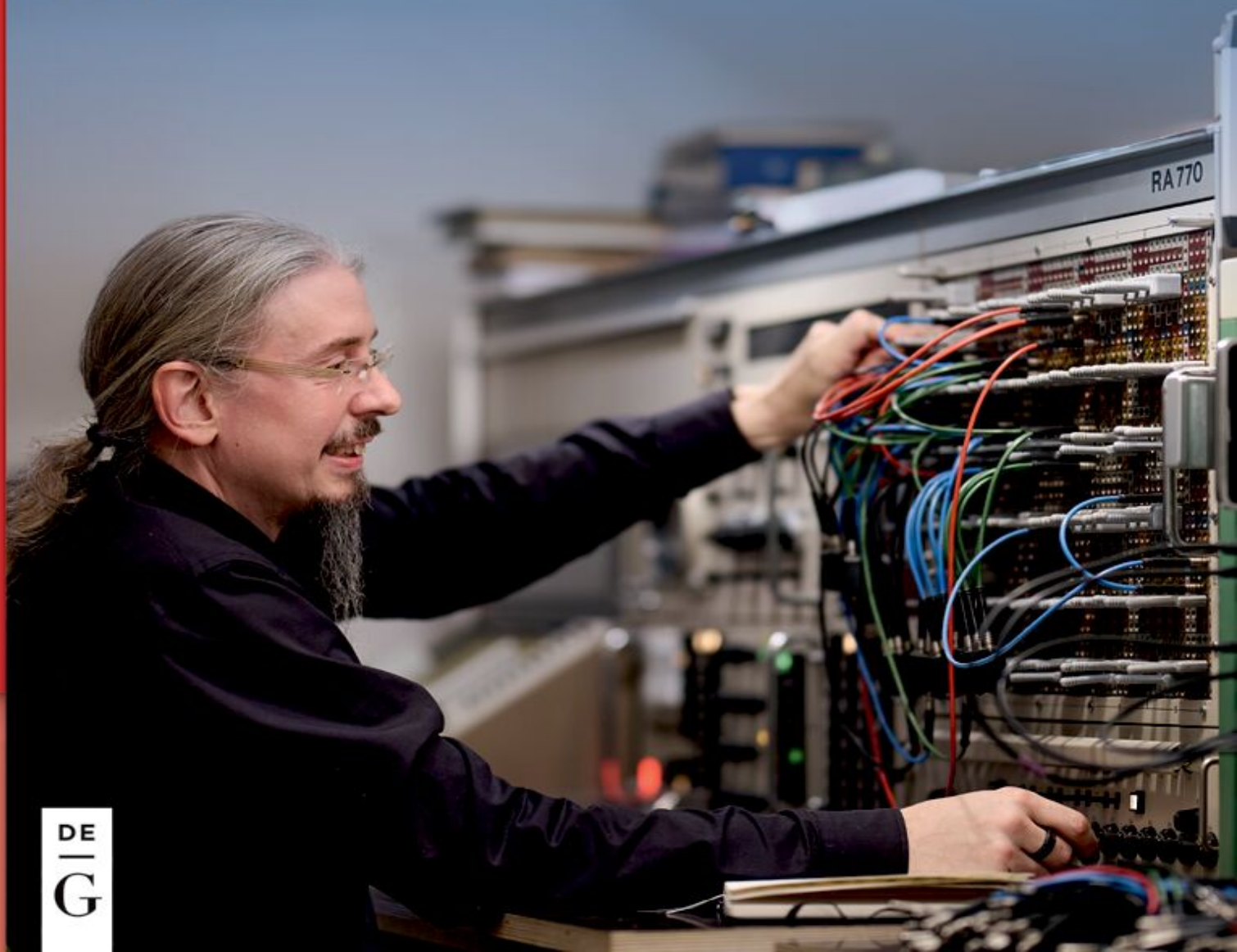
DE GRUYTER GRADUATE

Bernd Ulmann

ANALOG COMPUTING

DEVELOPMENT, PROGRAMMING, APPLICATIONS, AND FUTURE DIRECTIONS

3RD EDITION



DE G

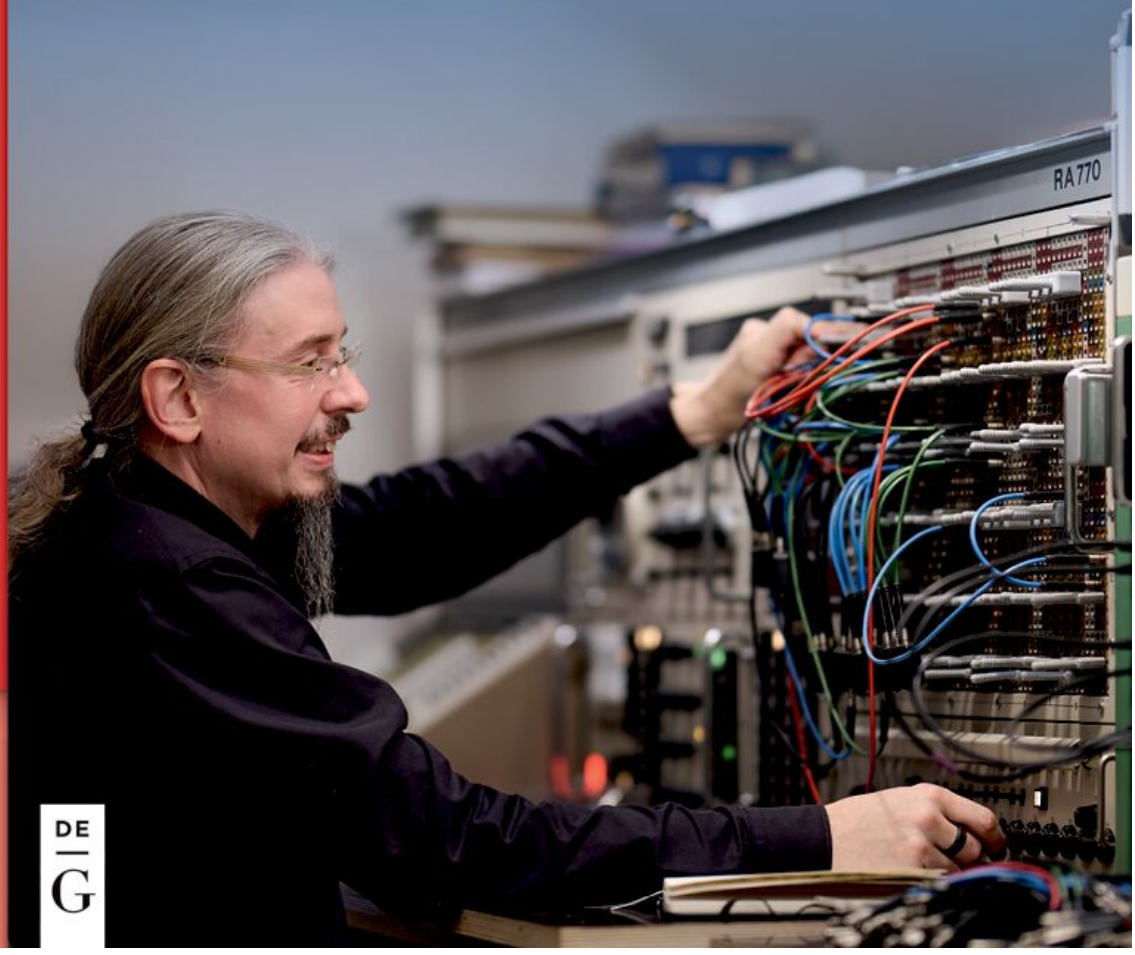
DE GRUYTER GRADUATE

Bernd Ulmann

ANALOG COMPUTING

DEVELOPMENT, PROGRAMMING, APPLICATIONS, AND FUTURE DIRECTIONS

3RD EDITION



DE G

Bernd Ulmann

Analog Computing

Bernd Ulmann

Analog Computing

DE GRUYTER

ISBN 9783119146289
e-ISBN (PDF) 9783112210178
e-ISBN (EPUB) 9783112210505

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at http://dnb.dnb.de.

© 2026 Walter de Gruyter GmbH, Berlin/Boston

Contents

Frontmatter

Acknowledgments

Preface to the 2nd edition

Preface to the 3rd edition

Contents

Epigraph

1 Introduction

1.1 Outline

1.2 Computing before computers

1.3 The notion of analog computing

1.4 Direct and indirect analogies

1.5 The computational power of analog computers

2 Mechanical analog computers

2.1 Astrolabes

2.2 The Antikythera mechanism

2.3 Slide rules

2.4 Planimeters

2.5 Mechanical computing elements
2.5.1 Function generation
2.5.2 Differential gears
2.5.3 Integrators
2.5.4 Multipliers

2.6 Harmonic synthesizers and analyzers

2.7 Mechanical fire control systems

2.8 Differential analyzers

3 The first electronic analog computers

3.1 Helmut Hoelzer
3.1.1 The “Mischgerät”
3.1.2 Hoelzer’s analog computer

3.2 George A. Philbrick’s Polyphemus

3.3 Electronic fire control systems

3.4 MIT

3.5 Anacom and the Caltech Computer

4 Basic computing elements

4.1 Operational amplifiers
4.1.1 Early operational amplifiers
4.1.2 Drift stabilization

4.2 Summers

4.3 Integrators

4.4 Coefficient potentiometers

4.5 Function generators

4.5.1 Servo function generators
4.5.2 Curve followers
4.5.3 Photoformers
4.5.4 Varistor function generators
4.5.5 Photocells
4.5.6 Diode function generators
4.5.7 Inverse functions
4.5.8 Functions of two variables
4.5.9 Function generation on hybrid computers

4.6 Multiplication

4.6.1 Servo multipliers
4.6.2 Crossed-fields electron-beam multiplier
4.6.3 Hyperbolic field multiplier
4.6.4 Other multiplication tubes
4.6.5 Time division multipliers
4.6.6 Logarithmic multipliers
4.6.7 Quarter square multipliers
4.6.8 Other multiplication schemes
4.6.9 Hybrid multipliers
4.6.10 Modern multipliers

4.7 Division and square root

4.8 Comparators

4.9 Limiters

4.10 Resolvers

4.11 Time delay

4.12 Random noise generators

4.13 Output devices

5 The anatomy of a classic analog computer

5.1 Analog patch panel

5.2 Function generators

5.3 Digital patch panel and controls

5.4 Readout

5.5 Control

5.6 Performing a computation

6 Some typical analog computers

6.1 Telefunken RA 1

6.2 GAP/R analog computers

6.3 EAI 231R

6.4 Early transistorized systems

6.5 Later analog computers

6.6 THE ANALOG THING

6.7 LUCIDAC and REDAC

7 Programming

7.1 Basic approach

7.2 Kelvin’s feedback technique

7.3 Substitution method

7.4 Partial differential equations

7.4.1 Quotient of differences

7.4.2 Separation of variables

7.5 Scaling

8 Programming examples

8.1 Solving $\ddot{y} + \omega^2 y = 0$

8.2 Sweep generator

8.3 Mass-spring-damper system

8.4 Predator and prey

8.5 Simulation of an epidemic

8.6 Bouncing ball

8.7 Car suspension

8.8 Lorenz attractor

8.9 Mathieu’s equation

8.10 Projection of rotating bodies

8.11 Conformal mapping

8.12 Hindmarsh-Rose neuron model

9 Digital control units

10 Hybrid computers

10.1 Systems

10.2 Programming

10.3 ML training example

10.4 Stochastic ODE on a hybrid computer

11 Autopatch systems

12 Digital differential analyzers

12.1 Basic computing elements

12.1.1 Integrators

12.1.2 Servos

12.1.3 Summers

12.1.4 Additional elements

12.2 Programming examples

12.3 Problems

12.4 Systems

12.4.1 MADDIDA

12.4.2 Bendix D-12

12.4.3 CORSAIR
12.4.4 TRICE

13 Stochastic computing

14 Simulation of analog computers

14.1 Basics

14.2 DDA programming system for the IBM 7074

14.3 CSMP

14.4 DARE

14.5 Modern approaches

15 Applications

15.1 Mathematics
15.1.1 Differential equations
15.1.2 Multiple integrals
15.1.3 Integral equations
15.1.4 Roots of polynomials
15.1.5 Orthogonal functions
15.1.6 Algebraic functions of complex variables
15.1.7 Linear algebra
15.1.8 Eigenvalues and -vectors
15.1.9 Fourier synthesis and analysis
15.1.10 Random processes and Monte-Carlo simulations
15.1.11 Optimization and operational research
15.1.12 Pipeline networks

15.1.13 Display of complex shapes

15.2 Physics

15.2.1 Orbit calculations
15.2.2 Particle trajectories and plasma physics
15.2.3 Optics
15.2.4 Heat-transfer
15.2.5 Fallout prediction and nuclear weapons effects
15.2.6 Semiconductor research
15.2.7 Ferromagnetic films

15.3 Chemistry

15.3.1 Reaction kinetics
15.3.2 Quantum chemistry and mechanics

15.4 Mechanics and engineering

15.4.1 Vibrations
15.4.2 Shock absorbers
15.4.3 Earthquake simulation
15.4.4 Bending beams and similar problems
15.4.5 Rotating systems and gears
15.4.6 Compressors
15.4.7 Crank mechanisms and linkages
15.4.8 Non-destructive testing
15.4.9 Ductile deformation
15.4.10 Pneumatic and hydraulic systems
15.4.11 Control of machine tools
15.4.12 Servo systems

15.5 Color matching

15.6 Nuclear technology

- 15.6.1 Research
- 15.6.2 Reactor and neutron kinetics
- 15.6.3 Training
- 15.6.4 Control
- 15.6.5 Enrichment
- 15.6.6 Planning of nuclear power plants

15.7 Biology and medicine

- 15.7.1 Ecosystems
- 15.7.2 Metabolism and pharmacokinetics
- 15.7.3 Cardiovascular systems
- 15.7.4 Closed loop control studies
- 15.7.5 Neurophysiology
- 15.7.6 Epidemiology
- 15.7.7 Aerospace medicine
- 15.7.8 Locomotor systems
- 15.7.9 Dosimetry and related fields
- 15.7.10 Training

15.8 Geology and marine science

- 15.8.1 Oil and gas reservoirs
- 15.8.2 Seismology
- 15.8.3 Ray tracing

15.9 Economics

15.10 Power engineering

- 15.10.1 Generators
- 15.10.2 Transformers

15.10.3 Power inverters and rectifiers

15.10.4 Transmission lines

15.10.5 Power grid simulation

15.10.6 Frequency control

15.10.7 Power station simulation

15.10.8 Dispatch computers

15.11 Electronics and telecommunications

15.11.1 Circuit simulation

15.11.2 Frequency response

15.11.3 Filter design

15.11.4 Modulators and demodulators

15.11.5 Antenna and radar systems

15.12 Automation

15.12.1 Data processing

15.12.2 Correlation analysis

15.12.3 Closed loop control and servo systems

15.12.4 Sampling systems

15.12.5 Embedded systems

15.13 Chemical process engineering

15.13.1 Mixing tanks, heat exchangers, evaporators, and distillation columns

15.13.2 Control processes and adaptive control

15.13.3 Parameter determination and optimization

15.13.4 Plant startup simulation

15.14 Transport systems

15.14.1 Automotive engineering

15.14.2 Railway vehicles
15.14.3 Hovercrafts and maglevs
15.14.4 Nautics

15.15 Aeronautical engineering
15.15.1 Landing gears
15.15.2 Aircraft arresting gear systems
15.15.3 Jet engines
15.15.4 Helicopters
15.15.5 Flutter simulations
15.15.6 Flight simulation
15.15.7 Airborne simulators
15.15.8 Guidance and control
15.15.9 Miscellaneous

15.16 Rocketry
15.16.1 Rocket motor simulation
15.16.2 Rocket simulation
15.16.3 Real-time data analysis
15.16.4 Spacecraft maneuvers
15.16.5 Mercury, Gemini, and Apollo

15.17 Military applications

15.18 Education

15.19 Voice recognition

15.20 Arts, entertainment, and music
15.20.1 Arts
15.20.2 Entertainment

15.20.3 Music

15.21 Analog computer centers

16 Analog and hybrid computing in the 21st century

16.1 Challenges

16.2 Applications

A THE ANALOG THING explained

Bibliography

Index

Frontmatter

Dedication

To my beloved wife Rikka

Acknowledgments

© 2026 Walter de Gruyter GmbH, Berlin/Boston 2026 Walter de Gruyter GmbH, Berlin/Boston

This book would not have been possible without the support and help of many people. First of all, I would like to thank my wife Rikka Mitsam, who not only did a lot of proofreading and a terrific job in preparing many of the pen-and-ink drawings, but also never complained about being neglected although I lived the last months more or less in seclusion writing this book.

I am particularly grateful for the support and help of Jens Breitenbach, who did a magnificent job at proofreading and provided many suggestions and improvements enhancing the text significantly. He also spotted many LaTeX-sins of mine, thus improving the overall appearance of this book considerably.

In addition to that, I would like to thank Benjamin Barnickel, Arne Charlet, Daniela Koch, Theresa Szczepanski, Dr. Reinhard Steffens and Francis Massen for their invaluable help in proofreading.

Additionally, I would like to thank Tore Sinding Bekkedal, who took the picture shown in →figure 8.35, Tim Robinson, who built the incredible Meccano-Differential Analyzer shown in →figure 2.24, Tibor Florestan Pluto, who took the photo shown on the title page, Robert Limes, who took the picture shown in →figure 10.4, and

Bruce Baker, who donated the pictures shown in →figures 15.84, →15.88, and →15.89, for their permissions to use the aforementioned pictures.

Without the continuous encouragement of Dr. habil. Karl Schlagenhauf and his invaluable suggestions this book might very well not exist at all.

Last but not least, I would like to thank Prof. Dr. Wolfgang Giloi, Prof. Dr. Rudolf Lauber, Dr. Adolf Kley, Prof. Dr. Günter Meyer-Brötz and Manfred Klittich for sharing their memories of the early days of analog computing at Telefunken, etc. Furthermore, Jeroen Brinkmann introduced me to the “Hammer Computer” and Bert Brouwer offered many insights into solving partial differential equations in conjunction with heat transfer problems.

This book was typeset with LaTeX. Most schematics were drawn using EAGLE and Arno Jacob’s wonderful analog computing symbol library, other vector graphics were created with `xfig`.

Registered names, trademarks, designations, etc., used in this book, even when not specifically marked as such, are not to be considered unprotected by law.

Preface to the 2nd edition

This second edition of “Analog Computing” would not exist if it wasn’t for Dr. Damiano Sacco, acquisition editor at DeGruyter, who asked me to prepare a new edition of “Analog Computing”, which has established itself as a standard textbook since it was first published in 2013.

A lot has happened since then in the world of analog computing. First of all many “new” historic sources, papers, and photographs have become available since 2013. Accordingly, the bibliography of this 2nd edition has been expanded by more then 200 additional sources. Second, and even more importantly, interest in analog computing has grown enormously. This computing paradigm is about to change the world of computing in the near future with a plethora of interesting and commercially important applications ranging from low power computing as required for medical implants to high performance computing, artificial intelligence, and many, many more. It seems even plausible that analog computers might be able to do things typically ascribed to quantum computers while being much simpler to implement, run, and program.

Bringing back analog computers in much more advanced forms than their historic ancestors will change the world of computing drastically and forever. They will not replace the ubiquitous stored-program digital computers, but they will complement them, thus making it possible to solve problems that are currently out of reach for stand-alone digital computers.

The author is especially indebted to Dr. Chris Giles for many valuable discussions and his meticulous proofreading of this 2nd edition and to Nicole Matje for her valuable corrections and suggestions. The author would also like to thank Achim Dassow for information about the QK-329 beam-deflection tube and additional information on early multiplication devices. Rainer Glaschick provided much background information on early differential analyzers and on the hyperbolic field tube. Oliver Bach also proofread the book and took special care of the bibliography making sure that it is consistent and correct. Prof. Dr. Dirk Killat contributed →figure 14.11. The new cover picture

was taken by Tibor Florestan Pluto and shows the author with a GTE-22 analog computer from the late 1960s.

Preface to the 3rd edition

I, the author, am incredibly excited about this third edition of "Analog Computing" as it shows that there is a growing interest in analog and hybrid computing, my favorite topics in life. :-)

This edition contains a wealth of new material, including two new chapters on digital control units (→section 9) and *autopatch* systems (→section 11) as well as a new appendix containing a comprehensive description of the circuitry of *THE ANALOG THING*, a small educational computer which has been in widespread use for the last couple of years (→section A). The section on the Lorenz system (→section 8.8) has been completely rewritten with a thorough exposition of the scaling process. The section on the Caltech computer (→section 3.5) also has been considerably extended. Spelling is now according to American English, although all dates are in dd.mm.yyyy format.

Apart from this the bibliography has been substantially expanded and now contains 1265 entries, including many additional application examples and in-depth sources.[1]

This book would not have been possible without the support of many friends, family members, and colleagues. First of all, I would like to thank my beloved wife Rikka Mitsam for her patience and never ceasing support of my obsession with analog and hybrid computing. I am also greatly indebted to Ute Skambraks[2] from DeGruyter – without her support there would have been no 3rd edition of this book at all. Dr. Chris Giles not only

enthusiastically proofread this book but also made many invaluable suggestions and improvements which greatly enhanced the quality of the book. In addition to this he always had an open ear when I needed someone to discuss technical topics in depth. Nicole Matje, Oliver Bach, Stefan Wolfrum, Shrish Roy did a terrific proofreading job.

I am also greatly indebted to Daniel Ruepp who spotted and corrected many subtle and not so subtle mistakes, ranging from simple grammatical blunders and sign errors to outright mathematical sins.

Joachim Wagner worked wonders with many of the new pictures included in this 3rd edition, especially in cases where only very grainy, old pictures were available. I would also like to thank Marco Guerini for discussions on analog computing in general as well as for several historic papers he sent me. Christopher Howard made me aware of a special form of servo function generators used in early flight simulators and even took detailed photos on my behalf of such devices. Dr. Anton Gsandtner sent me a wealth of information about the work done in hybrid computing at the TU Wien. Dag Spicer from the *Computer History Museum* in Moffett Field gave me access to detailed manuals for a very early *REAC* system. Last but not least, I would like to thank Prof. Dr. Olivier Bournez (École Polytechnique) for long and fruitful discussions on computational complexity and related fields.

I would also like to thank Barbara Hoelzer-Beck, Helmut Hoelzer's daughter, for her support and her incredible hospitality Rikka and I experienced during a week-long visit a few years ago. She also granted the rights for →figure 3.18 showing her father working on the replica of his analog computer in the 1990s. Thanks are also due to Mrs. Shelley Hollingdrake and Dr. Graham Clayton of the

Society of Dyers and Colourists for the wonderful picture of COMIC (15.40) and the permission to use it in this book. JOHANNES FRITZSCHE took the author's photo used on the book cover.

The cover picture of this edition was taken by TIBOR FLORESTAN PLUTO and shows the me with my beloved Telefunken RA770 precision analog computer, built in 1966.

The schematics shown in this book now appear in three different styles (not counting schematics and computer setups replicated from literature). The earliest ones, dating back to 1st edition from 2009, were mostly drawn with `xfig`, a rather basic X-windows vector graphic program. Later schematics added in the 2nd edition were drawn using *Eagle*, a CAD program once in rather widespread use. Things have changed a lot since then and schematics added in this 3rd edition were drawn using *KiCAD*, a cross-platform open source electronics design automation suite.[3]

I hope reading this 3rd edition will be as much of a pleasure and fun as writing it was.

Happy analog computing! :-)

Contents

1 Introduction
- 1.1 Outline
- 1.2 Computing before computers
- 1.3 The notion of analog computing
- 1.4 Direct and indirect analogies
- 1.5 The computational power of analog computers

2 Mechanical analog computers
- 2.1 Astrolabes
- 2.2 The Antikythera mechanism
- 2.3 Slide rules
- 2.4 Planimeters
- 2.5 Mechanical computing elements
  - 2.5.1 Function generation
  - 2.5.2 Differential gears
  - 2.5.3 Integrators
  - 2.5.4 Multipliers
- 2.6 Harmonic synthesizers and analyzers
- 2.7 Mechanical fire control systems
- 2.8 Differential analyzers

3 The first electronic analog computers
- 3.1 Helmut Hoelzer
  - 3.1.1 The “Mischgerät”
  - 3.1.2 Hoelzer’s analog computer
- 3.2 George A. Philbrick’s Polyphemus
- 3.3 Electronic fire control systems
- 3.4 MIT
- 3.5 Anacom and the Caltech Computer

4 Basic computing elements
- 4.1 Operational amplifiers

4.1.1 Early operational amplifiers
4.1.2 Drift stabilization
4.2 Summers
4.3 Integrators
4.4 Coefficient potentiometers
4.5 Function generators
4.5.1 Servo function generators
4.5.2 Curve followers
4.5.3 Photoformers
4.5.4 Varistor function generators
4.5.5 Photocells
4.5.6 Diode function generators
4.5.7 Inverse functions
4.5.8 Functions of two variables
4.5.9 Function generation on hybrid computers
4.6 Multiplication
4.6.1 Servo multipliers
4.6.2 Crossed-fields electron-beam multiplier
4.6.3 Hyperbolic field multiplier
4.6.4 Other multiplication tubes
4.6.5 Time division multipliers
4.6.6 Logarithmic multipliers
4.6.7 Quarter square multipliers
4.6.8 Other multiplication schemes
4.6.9 Hybrid multipliers
4.6.10 Modern multipliers
4.7 Division and square root
4.8 Comparators
4.9 Limiters
4.10 Resolvers
4.11 Time delay
4.12 Random noise generators
4.13 Output devices

5 The anatomy of a classic analog computer
- 5.1 Analog patch panel
- 5.2 Function generators
- 5.3 Digital patch panel and controls
- 5.4 Readout
- 5.5 Control
- 5.6 Performing a computation

6 Some typical analog computers
- 6.1 Telefunken RA 1
- 6.2 GAP/R analog computers
- 6.3 EAI 231R
- 6.4 Early transistorized systems
- 6.5 Later analog computers
- 6.6 THE ANALOG THING
- 6.7 LUCIDAC and REDAC

7 Programming
- 7.1 Basic approach
- 7.2 Kelvin's feedback technique
- 7.3 Substitution method
- 7.4 Partial differential equations
  - 7.4.1 Quotient of differences
  - 7.4.2 Separation of variables
- 7.5 Scaling

8 Programming examples
- 8.1 Solving $\ddot{y} + \omega^2 y = 0$
- 8.2 Sweep generator
- 8.3 Mass-spring-damper system
- 8.4 Predator and prey
- 8.5 Simulation of an epidemic
- 8.6 Bouncing ball
- 8.7 Car suspension
- 8.8 Lorenz attractor
- 8.9 Mathieu's equation

8.10 Projection of rotating bodies
8.11 Conformal mapping
8.12 Hindmarsh-Rose neuron model
9 Digital control units
10 Hybrid computers
10.1 Systems
10.2 Programming
10.3 ML training example
10.4 Stochastic ODE on a hybrid computer
11 Autopatch systems
12 Digital differential analyzers
12.1 Basic computing elements
12.1.1 Integrators
12.1.2 Servos
12.1.3 Summers
12.1.4 Additional elements
12.2 Programming examples
12.3 Problems
12.4 Systems
12.4.1 MADDIDA
12.4.2 Bendix D-12
12.4.3 CORSAIR
12.4.4 TRICE
13 Stochastic computing
14 Simulation of analog computers
14.1 Basics
14.2 DDA programming system for the IBM 7074
14.3 CSMP
14.4 DARE
14.5 Modern approaches
15 Applications
15.1 Mathematics
15.1.1 Differential equations

15.1.2 Multiple integrals
15.1.3 Integral equations
15.1.4 Roots of polynomials
15.1.5 Orthogonal functions
15.1.6 Algebraic functions of complex variables
15.1.7 Linear algebra
15.1.8 Eigenvalues and -vectors
15.1.9 Fourier synthesis and analysis
15.1.10 Random processes and Monte-Carlo simulations
15.1.11 Optimization and operational research
15.1.12 Pipeline networks
15.1.13 Display of complex shapes
15.2 Physics
15.2.1 Orbit calculations
15.2.2 Particle trajectories and plasma physics
15.2.3 Optics
15.2.4 Heat-transfer
15.2.5 Fallout prediction and nuclear weapons effects
15.2.6 Semiconductor research
15.2.7 Ferromagnetic films
15.3 Chemistry
15.3.1 Reaction kinetics
15.3.2 Quantum chemistry and mechanics
15.4 Mechanics and engineering
15.4.1 Vibrations
15.4.2 Shock absorbers
15.4.3 Earthquake simulation
15.4.4 Bending beams and similar problems
15.4.5 Rotating systems and gears
15.4.6 Compressors
15.4.7 Crank mechanisms and linkages

15.4.8 Non-destructive testing
15.4.9 Ductile deformation
15.4.10 Pneumatic and hydraulic systems
15.4.11 Control of machine tools
15.4.12 Servo systems
15.5 Color matching
15.6 Nuclear technology
15.6.1 Research
15.6.2 Reactor and neutron kinetics
15.6.3 Training
15.6.4 Control
15.6.5 Enrichment
15.6.6 Planning of nuclear power plants
15.7 Biology and medicine
15.7.1 Ecosystems
15.7.2 Metabolism and pharmacokinetics
15.7.3 Cardiovascular systems
15.7.4 Closed loop control studies
15.7.5 Neurophysiology
15.7.6 Epidemiology
15.7.7 Aerospace medicine
15.7.8 Locomotor systems
15.7.9 Dosimetry and related fields
15.7.10 Training
15.8 Geology and marine science
15.8.1 Oil and gas reservoirs
15.8.2 Seismology
15.8.3 Ray tracing
15.9 Economics
15.10 Power engineering
15.10.1 Generators
15.10.2 Transformers
15.10.3 Power inverters and rectifiers

15.10.4 Transmission lines
15.10.5 Power grid simulation
15.10.6 Frequency control
15.10.7 Power station simulation
15.10.8 Dispatch computers
15.11 Electronics and telecommunications
15.11.1 Circuit simulation
15.11.2 Frequency response
15.11.3 Filter design
15.11.4 Modulators and demodulators
15.11.5 Antenna and radar systems
15.12 Automation
15.12.1 Data processing
15.12.2 Correlation analysis
15.12.3 Closed loop control and servo systems
15.12.4 Sampling systems
15.12.5 Embedded systems
15.13 Chemical process engineering
15.13.1 Mixing tanks, heat exchangers, evaporators, and distillation columns
15.13.2 Control processes and adaptive control
15.13.3 Parameter determination and optimization
15.13.4 Plant startup simulation
15.14 Transport systems
15.14.1 Automotive engineering
15.14.2 Railway vehicles
15.14.3 Hovercrafts and maglevs
15.14.4 Nautics
15.15 Aeronautical engineering
15.15.1 Landing gears
15.15.2 Aircraft arresting gear systems
15.15.3 Jet engines
15.15.4 Helicopters

15.15.5 Flutter simulations
15.15.6 Flight simulation
15.15.7 Airborne simulators
15.15.8 Guidance and control
15.15.9 Miscellaneous
15.16 Rocketry
15.16.1 Rocket motor simulation
15.16.2 Rocket simulation
15.16.3 Real-time data analysis
15.16.4 Spacecraft maneuvers
15.16.5 Mercury, Gemini, and Apollo
15.17 Military applications
15.18 Education
15.19 Voice recognition
15.20 Arts, entertainment, and music
15.20.1 Arts
15.20.2 Entertainment
15.20.3 Music
15.21 Analog computer centers
16 Analog and hybrid computing in the 21st century
16.1 Challenges
16.2 Applications
A THE ANALOG THING explained
Bibliography
Index

Epigraph

“An analog computer is a thing of beauty and a joy forever.”[1]

1 Introduction

1.1 Outline

A book on analog computing and analog computers? You might ask: Isn't that 80 years late?[1] No, it isn't – although the beautiful analog computers of the past are long since history and only few have been preserved in museum collections, the idea of analog computing is still a marvel of elegance and the following chapters will show that it has a bright and fruitful future.

The intention of this book is twofold: It gives a comprehensive description of the history and technology of classic analog computing but also shows the particular strengths of the analog computation paradigm, which, combined with current state of the art digital circuitry, will find applications in areas such as low power computing, high performance computing, and maybe most importantly in *artificial intelligence* (*AI*) and many other contemporary and economically important fields.

The following chapters first introduce the notion of analog computing before describing the early development of analog computers starting with mechanical analog computers like the Antikythera mechanism, which was built around 100 B. C., and ending with the first analog electronic[2] analog computers developed by Helmut Hoelzer in Germany and George A. Philbrick and others in the United States.

Next, the basic elements of a typical analog computer are described in detail followed by two chapters showing their anatomy, ranging from classic systems to recent implementations, and giving examples of some systems.

The next chapter gives an introduction to analog computer programming[3] followed by a number of practical programming examples ranging from the solution of simple differential equations to the simulation of more complex and nonlinear systems such as the HINDMARSH-ROSE model of a spiking and bursting neuron.

The next two chapters treat digital control units and hybrid computers (analog computers coupled with stored-program digital computers) followed by a chapter describing various approaches toward autopatch systems allowing analog computer to be automatically reconfigured under control of an attached digital computer. This in turn is followed by a treatment of *Digital Differential Analyzers* (digital implementations of analog computers), a chapter on stochastic computing, and a chapter on the simulation of analog computers on classic stored-program digital computers.

The following and largest chapter covers a plethora of classic and current applications of analog and hybrid computers while the last chapter of this book covers the bright future of these technologies in the 21st century with application areas.

1.2 Computing before computers

One of the most important steps in human history was the development of suitable number representations including a zero which allowed for simple algorithms to perform basic operations such as addition, subtraction, multiplication, and division.[4] However, performing repeated computations manually was, and still is, a tedious and error prone process which limited technological progress quite substantially before the advent of analog (and digital) computers.[5] It took humanity a long time to develop mathematical concepts capable of describing natural

processes, including complex dynamic systems. The most important concept being that of (coupled) *differential equations* (*DEQ*s).

This class of equations contains one or several unknown functions and their respective derivatives. Solving these analytically is only possible for fairly simple problems and even then it is an often convoluted and complex process requiring a very high level of mathematical prowess. As soon as the problems get more complicated there are no analytical solutions whatsoever, leaving only numerical schemes for their approximate solution.

Performing even something as simple as an *Euler integration scheme*[6] manually is an incredible chore and even a single error in one of the many hundreds of thousands of steps will invalidate the result.

Since problems described by differential equations are of utmost importance in all fields of science and technology, better means for solving them were sought. Only those who have had to use tables of logarithms, slide rules or mechanical four-species calculators for doing actual work can fully understand the computational revolution started by the first analog computers in the 19th century. Performing a Fourier[7] transform manually was a very tedious manual process. With the advent of *harmonic analyzers* this could be done in a matter of hours or even minutes using some devilishly clever mechanical devices.

The development of computing or simulation machines followed two paths: Analog computers, the main topic of this book, and the now dominant stored-program digital computers, pioneered by Charles Babbage[8] in the form of his *Difference Engine* and the later *Analytical Engine*, which, unfortunately, was never built and thus remained an elusive dream.

Interestingly, the idea of solving problems by actually building *analogs*[9] in the form of coupled mechanical elements such as differential gears, etc., is much older than the now prevailing algorithmic approach. The oldest example of such an analog goes back at least two thousand years as the *Antikythera mechanism*[10] shows. After a long and deplorable hiatus, the idea of computing by mechanical means came up again. The first machines were two and four-species mechanical calculators such as those developed by Wilhelm Schickard[11] in 1623 and Blaise Pascal[12] in 1642. These developments greatly simplified the task of performing basic arithmetic operations but did not help solving intricate differential equations directly.

These developments gathered momentum in the 19th century driven by the computational needs of a time of rapid technological development and an incredible technology optimism. This gave rise to the first mechanical analog computers capable of analyzing and predicting the behavior of dynamic systems such as sea tides. These, in turn, led to more general purpose mechanical analog computers which allowed problems that were basically intractable by manual means to be solved.

However, even in the late 1940s the term *computer* was still associated with a human performing computations manually, with the possible aid of tables of logarithms, slide rules, or mechanical four-species calculators. →Figure 1.1 shows human computers working at the *NACA*[13] High Speed Flight Station in 1949. The pictures taped to the diagonal beam in the room show the purpose of these computations – advancing the state of the art in the field of aviation and later space flight.



Figure 1.1. Human Computers at the NACA High Speed Flight Station Computer Room in 1949 (NACA (NASA) – Dryden Flight Research Center Photo Collection)

One of the most famous (and last) projects to be performed by manual calculation was the *Mathematical Tables Project* started in 1938 during the great depression under the auspices of the *Works Progress Administration* (*WPA*). It employed 450 unemployed and unskilled workers to calculate mathematical tables including logarithms, exponential and trigonometric functions. Using a number of mechanical calculators including some early IBM punch card machines the project took ten years to complete a total of 28 volumes published by Columbia

University Press. The *manual computation* group alone consisted of “97 computers” and “10 clerks”.[14] These tables were not intended as an end in themselves, their purpose was to be used as tools in other, much more complex projects. However, the first stored program digital computers were available in 1948, rendering the result of these ten laborious years obsolete in no time and bringing the era of human computers to an end.

Another impressive example of tedious manual calculations arising from mechanical engineering tasks is described in [→ Shute 1956, p. 74 f.]. The problem at hand was the calculation of the loads acting on connecting wires within the *R100* air ship:

> “*The forces and bending moments in the members could then be calculated by the solution of a lengthy simultaneous equation containing up to seven unknown quantities; this work usually occupied two calculators about a week, using a Fuller slide rule and working in pairs to check for arithmetical mistakes. In the solution it was usual to find a compression force in one or two of the radial wires; the whole process then had to be begun again using a different selection of wires*.”

Some (electro-)mechanical analog computers, especially in avionics systems, required high-precision cams that were also complicated to design. In the 1950s digital computers such as the Bendix *G-15* were used for calculations like this since manual approaches were utterly impractical as Huggins notes:[15]

> “*The Bendix G-15 General Purpose Computer was put to work, to complete the [about 1400] radii. Hundreds of thousands of equations that would take an engineer 3 years to calculate solutions for and possibly a 4th year to check the solutions, were solved in 40 hours*.”

The influence that the analog computers described in the following chapters had on the overall technological advancement of humanity cannot be overrated. It is safe to say that without these marvelous machines most of the technology

we take for granted nowadays would either not exist or would have been developed decades later when digital computers became much faster than their early incarnations.

As will be shown below, there are still areas of application where analog computers, in their most recent incarnations, outperform digital computers not only with respect to their sheer computational power but also and maybe even more importantly in terms of energy efficiency.

1.3 The notion of analog computing

First of all it should be noted that the common distinction between *digital* and *analog* computers, based on the way values are represented, is not correct. It is often said that digital computers differ from analog computers by their way of representing numbers as sequences of bits (binary digits), while electronic analog computers work with continuous voltages or currents to represent variables. This erroneous view has even found its way into some encyclopedias.

Apart from the fact that even voltages or currents are not really continuous – eventually an operation like integration boils down to charging a capacitor with discrete electron charges – some analog computers have used a bit-wise value representation and have been implemented using purely digital elements.

If the type of values used in a computation – discrete versus continuous – is not the distinguishing feature, what else could be used to differentiate between *digital* and *analog* computers? It turns out that the difference is to be found in the structure of these two classes of machines: In our modern sense of the word, a digital computer's constituent elements have a fixed structure and it solves problems by executing a sequence (or sequences) of instructions that implement an algorithm. These instructions

are read from some kind of memory, thus, a better term for this kind of computing machine would be *stored-program digital computer* since this describes both features of such a machine: Its ability to execute instructions fetched from a memory subsystem and working with numbers that are represented as streams of digits.[16]

An analog computer on the other hand is based on a completely different paradigm: Its internal structure is not fixed – in fact, a problem is solved on such a machine by changing its structure in a suitable way to generate a *model*, an *analog* of the problem.[17] This analog is then used to *analyze* or *simulate* the problem to be solved.[18]

Thus, the structure of an analog computer that has been set up to tackle a specific problem represents the problem itself, while a stored-program digital computer keeps its structure and only its controlling program changes. This is summarized by Charlesworth[19] as follows:

> "*An analogue computer is a piece of equipment whose component parts can be arranged to satisfy a given set of equations, usually simultaneous ordinary differential equations [(ODEs)]*."

Similarly [→ Berkeley et al. 1956, p. 75] state that

> "*[a]nalog computers, as the name is intended to imply, compute by means of setups that are analog of the problems to be solved*."

→ Figure 1.2 shows this basic difference in architecture and operation between digital and analog computers.

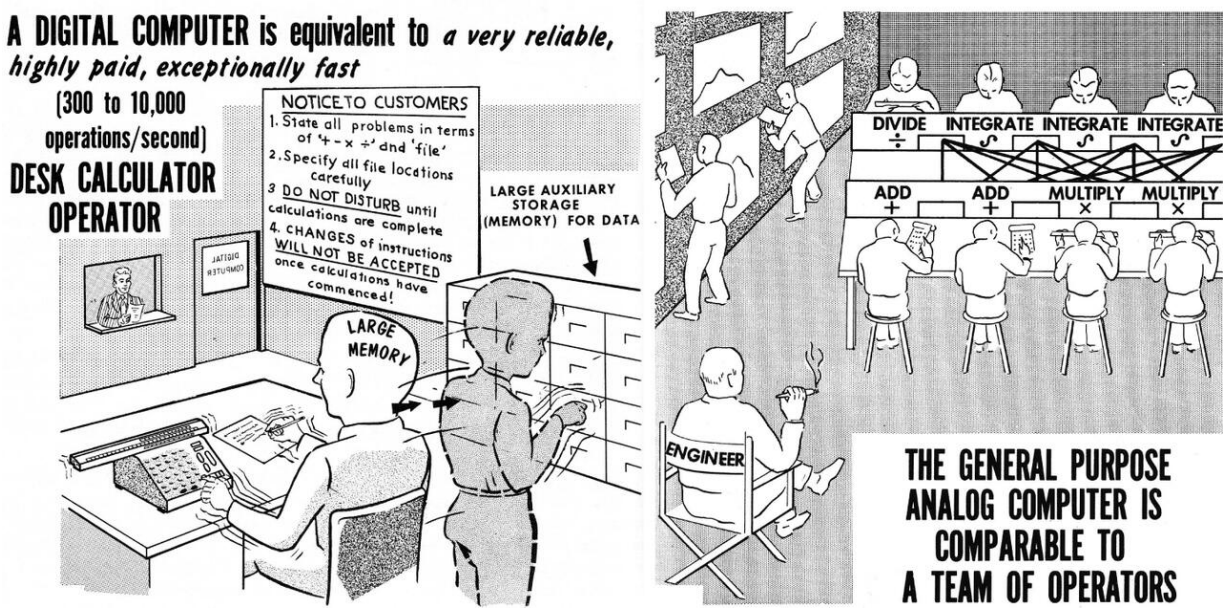


Figure 1.2. Comparison of the basic structure of stored-program digital computers and analog computers (see [→Truitt et al. 1960, p. 1-40] and [→Truitt et al. 1960, p. 1-41])

Consequently, it is perfectly possible to build digital analog computers and this has been done in several ways.[20] In fact such machines may play a substantial role in the future when high precision is of the utmost importance and energy efficiency is a secondary consideration.

Employing the techniques of building models, analogs of problems to be solved or analyzed, which have been developed through more than 50 years in the context of our current digital technology, can and will lead to systems with exceptional computational power as well as low power consumption.

→Table 1.1 shows the four basic possible combinations of analog/digital implementation technology and stored-program control vs. setting up an analog. Of these combinations only three are of practical interest:

Table 1.1 Types of computing machines based on an analog/digital electronic implementation with control based on either a stored-program concept or the implementation of an analog.

	Basic technology	
	Analog electronic	**Digital electronic**
Stored-program control	N/A	stored-program (memory programmed) digital computer
Setting up an analog	traditional analog electronic analog computer	digital differential analyzer

1. The modern stored-program digital computer,
2. the traditional analog electronic analog computer, which will be called an *analog computer* for simplicity in the following text, and, finally,
3. the *Digital Differential Analyzer* (*DDA*) which will be described in greater detail in →chapter 12.[21]

1.4 Direct and indirect analogies

When talking about analogies, it is necessary to distinguish between *direct* and *indirect* analogies. These two terms describe two extremes of abstraction levels in building analogies.

In the strict sense of the word direct analogies are models that are based on the same physical principles as the underlying problem to be solved just with a different scaling regarding size or time of a simulation. Well-known examples of such direct analogies are the determination of minimal surfaces using soap films,[22] the evaluation of tensile structures as they are used for roof structures like the one built for the Olympic stadium in Munich,[23] wind tunnel models for the evaluation of aerodynamic

properties of aircraft and rockets and many more. Other direct analogs employed *electrolytic tanks*. These are reservoirs filled with an electrolytic liquid with embedded electrodes to generate a desired potential distribution within the liquid. Using a two- or three-dimensional sensor carriage, much like an *xy*-plotter, the potential at any given coordinate within the tank can be determined. Such electrolytic tanks were widely used to solve problems in nuclear engineering, etc.

Over time the notion of direct analog analogies was also applied to computers consisting of networks of passive electronic components such as resistors, capacitors, and inductors. [→ Paschkis et al. 1968, p. 5] define a direct analog computer as being

> "*based on the identity of the equations describing two or more systems and carrying out measurements on that system which appears most convenient for that purpose [...] In the direct analog, there is a one-to-one relationship between the (passive) components and the physical properties of the several parts of the prime system*."

Due to their very nature such direct analogs are not very versatile and were often built and employed for a highly specific purpose.[24] → Figure 1.3 shows a wonderful example for a direct analogy, a string-weight model used by Antoni Gaudí[25] during the design and construction of the the Colónia Güell church. This model was built in 1908 at a scale of 1:10 with the weights scaled down by a factor of 10^{-4}. Such models are hanging down and simulate the pressure forces acting on pillars and columns by corresponding tensile forces through the maze of strings with their attached weights.[26]

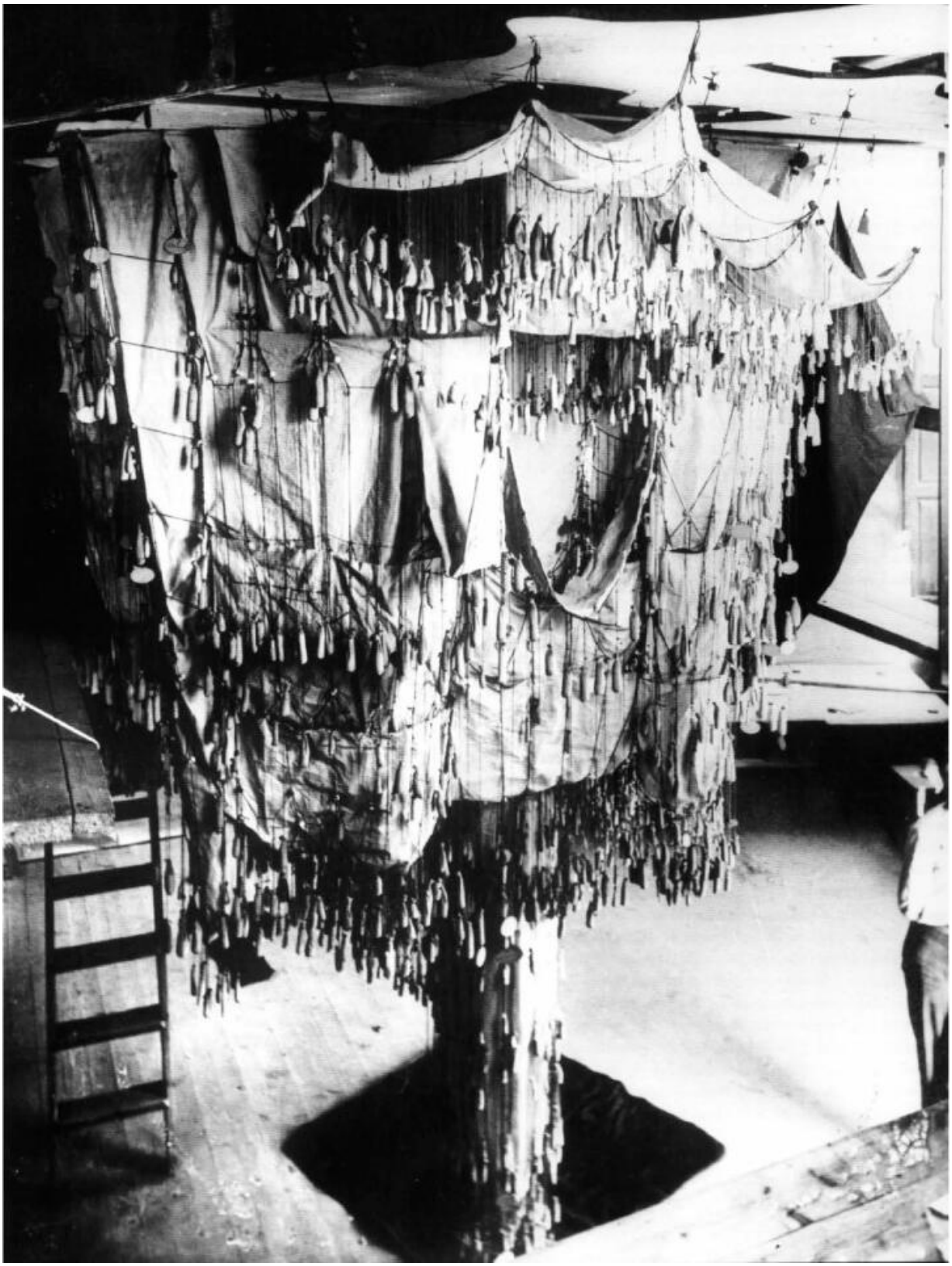


Figure 1.3. Scale model for the Colonia Güell church

In contrast, indirect analogies exhibit a much higher degree of abstraction and are thus much more versatile. The – mostly analog electronic – analog computers used for setting up indirect analogies are truly universal machines and cover a wide range of possible applications.[27] This higher level of abstraction makes the programming of this class of machines quite challenging since their setup does not bear any direct resemblance of the problem to be solved. Therefore a thorough mathematical description of the basic problem is required as a precondition for programming an indirect analog computer,[28] as in the case of our modern stored-program digital computers.

Nevertheless, the level of abstraction required for the successful application of analog computers is still relatively small compared with the algorithmic approach of stored-program digital computers. Last but not least, analog computers, be they direct or indirect, are models.

Due to the fact that analog computers work by acting as a model for a given problem that is represented by direct or indirect means, the amount of circuitry necessary for a simulation is determined by the complexity of the underlying problem.

Accordingly, analog computers are not capable of the trade off between time to solution on the one hand and complexity of the underlying problem on the other that is characteristic of stored-program digital computers. This is both a curse and a blessing: The curse being that an analog computer consisting of a given number of computing elements cannot solve a problem that requires more computing elements to be implemented, the blessing that the time to solution on an analog computer is more or less constant and is not related to the size of the underlying problem.

Thus, large problems require large analog computers regardless of the acceptable time to solution – some classic

problem areas, especially those found in aerospace and applications in the chemical industry, required well over 1 000 computing elements resulting in substantial, if not giant, analog computers.

In addition to this, a stored-program computer can always exchange compute time for precision – something an analog computer also cannot normally do.[29] The precision of an analog computer is given by its particular implementation and typically does not exceed about three to four decimal places for the variables involved in a computation.[30]

1.5 The computational power of analog computers

One of the first questions often encountered with analog computing is about *computing* as such. What is computing? Is everything in the universe performing actual computations? This is a very philosophical question which will be answered differently by different people and different schools of thought. Wikipedia[31] defines computing as "*any goal-oriented activity requiring, benefiting from, or creating computing machinery*" but also notes that the term is synonymous with *counting* and *calculating*, complicating things quite substantially.

Things seem to be a bit more clearly defined when it comes to computing machines, i. e., *computers*, which are typically assumed to be *programmable*. This is, of course, true for analog computers. When the definition of a computer requires the execution of sequences of instructions, then an analog computer could no longer be considered a computer as it does neither execute individual operations, nor does it have a memory system in the classic sense. Accordingly, the requirement of executing sequences of instructions should be dropped as a defining

element of a computer. Instead, the notion of *physical computing*, i. e., the application of physical processes and systems to computation, should complement that of classic stored-program digital computers.[32]

For the purposes of the following chapters the term computing will be interpreted as the use of any system capable of solving a problem represented by a mathematical description regardless of its inner workings. So, even soap bubbles will be considered as performing computations when they are part of some problem related setup with a goal such as minimizing surface area, etc.

How powerful are analog computers? Since a basic logic gate such as an inverting *AND* gate (a *NAND*) can be implemented on an analog computer, one could, at least in theory, simulate a stored program digital computer in a gate by gate fashion on an analog computer. Of course, this is as trivial as it is impractical and gives no real insight into the computational power of an analog computer but shows that an analog computer is in some sense *Turing complete*, i. e., it is capable of computing every Turing[33] computable function. However, this is unsatisfying as it does not do justice to the characteristics of analog computers and their applications.

The first one to develop a mathematical theory of the then new mechanical differential analyzer was Claude Elwood Shannon[34] in 1941.[35] He coined the term *General Purpose Analog Computer* (*GPAC*), a machine containing an unlimited number of integrators, adders, function units (input tables), and gear boxes.[36] He then defines a fundamental solvability condition for systems of ordinary differential equations extending the argument to the generation of (rational) functions, the approximation of constant multiplicative factors by gear ratios, etc.[37]

Such a GPAC can compute polynomials, exponential functions, trigonometric functions such as sine, cosine, etc., as well as all *differential algebraic* functions.[38] [→Graça et al. 2011] show that the class of these functions is closed under addition, subtraction, multiplication, division, composition, derivation, and inversion. In addition to this, it is shown that some initial value problems of polynomial ODEs can in fact simulate a universal Turing machine.[39]

In 1974 Marian Boykan Pour-El[40] published a paper[41] that extended Shannon's ideas on the GPAC. The first insight is that the class of functions which can be generated by analog computers is very large, although there are functions, such as $\frac{1}{\Gamma(x)}$ which are elusive to an analog computer.[42] [→Reif 2007] focuses on mechanical computation including the *reaction-diffusion chemical computer* as introduced by [→Adamatzky et al. 2005], mechanical self-assembly processes, *DNA computing*, etc.

In 1981 Lee Albert Rubel[43] extended Boykan Pour-El's work and showed the existence of an algebraic *universal differential equation* of fourth-order which approximates any continuous function to a degree determined by a continuous error function and even gives a practical example for such a function.[44] Eight years later he showed that, determinism holding[45], "*analytic outputs of a C^∞ GPAC are computable by a digital computer*".[46] In 1993 [→Rubel 1993] introduced the *extended analog computer* (*EAC*) as a theoretical construct, far surpassing Shannon's GPAC by allowing an arbitrary number of free variables with respect to integration (instead of time only). A more restricted extension of the GPAC, the *feedforward GPAC* (*FF-GPAC*) is introduced in [→Graça et al. 2003].

In a 2017 paper Bournez et al. define the class of *P* in terms of DEQs with a polynomial right hand side.[47] This eventually leads to the realization that "*any physically realistic (macroscopic)*

computer is equivalent to Turing machines both at a computability and at a computational complexity level."

Interestingly, Peter Coveney[48] argues that chaotic dynamical systems can not be fully understood by means of digital computers due to the uneven spacing and sparsity of IEEE floating point numbers and gives a practical example in the form of the generalized Bernoulli map.[49] This result hints at the possibility that analog computers might, in fact, exceed digital computers in terms of computational power, at least for certain problem classes, and contradicts the work of Rubel.

There are also arguments for the ability of analog computation to exceed the Turing limit although most probably possible only in a theoretical sense. Prime examples for this are given in [→ Mycka 2005] and by Hava T. Siegelmann[50] in [→ Siegelmann 1995] and [→ Siegelmann 1999]. However, these latter considerations require the existence of real numbers and their exploitability in analog computers.

Given all of the above it can be safely assumed that analog computers are at least Turing complete and there are even arguments that these marvelous machines might actually exceed stored-program digital computers, maybe paving the way to *hypercomputation*.[51]

2 Mechanical analog computers

The earliest analog computers were mechanical in their very nature but were far from being simple. In fact many mechanical analog computers were successfully employed to tackle complex problems ranging from peaceful tide computations to war-time applications like bomb trajectories, fire control, etc. The following sections give a short overview of the era of mechanical analog computers without going too much into detail since mechanical analogs will serve just as a prelude to this book's main theme of electronic analog computers.

2.1 Astrolabes

As early as about 150 B. C. the basics of *astrolabes* were developed. Such devices are basically inclinometers with some additional mechanics to model basic properties of spherical astronomy. Astrolabes are based on the apparent motion of celestial bodies, i. e., the observation that the paths described by stars in the sky are basically circles. Thus, the most common type of astrolabe is the *planispheric astrolabe*, developed in medieval times, which projects the firmament to the equatorial plane.

Using such an instrument it is possible to determine the position of some celestial bodies at a given time. As a navigational tool the planispheric astrolabe is far too imprecise. Nevertheless, it has been used to roughly locate the stars for getting navigational fixes. Detailed information about astrolabes can be found in [→Dodd 1969] and [→Morrison 2007].

2.2 The Antikythera mechanism

More than 120 years ago, in 1900, sponge divers found a lump of corroded gears in a Roman shipwreck, which carried treasures from Greece dating back to about 100 B. C. It turned out that these were the remains of one of the most complicated mechanical and mathematical devices ever. Due to its location near the Greek island Antikythera

(*Αντικύθηρα*) this impressive machine became known as the *Antikythera mechanism*. →Figure 2.1 shows the main fragment of this early analog computer in its current state of preservation.[1]



Figure 2.1. Main fragment of the Antikythera mechanism as displayed in the National Archaeological Museum, Athens, Greece

Intrigued by this find, Derek de Solla Price[2] started investigating the inner workings of this device and summarized his astonishing discoveries as follows:[3]

> "*It is a bit frightening to know that just before the fall of their great civilization the ancient Greeks had come so close to our age, not only in their thought, but also in their scientific technology*."

It turned out that the Antikythera mechanism was ahead of its time by at least 1 000 years. It is of such high complexity that recent research using modern X-ray tomography techniques, etc.[4] continues to deliver new insights. New capabilities and details were discovered as late as in 2021/2022.[5] The device modeled the movements of several celestial bodies, even taking into account various anomalies, which required differential gears[6] and much more complicated epicyclic gearing. This astonishing complexity of the Antikythera mechanism led Mike Edmunds[7] to the following statement:

> "*Nothing as sophisticated and complex is known for another thousand years. This machine rewrites the history of technology. It is a witness to a revolution in human thought*."

This intricate mechanism allowed the calculation of sun and moon positions at given dates and phases of the moon as well as the prediction of solar and lunar eclipses.[8] The implementation of these functions required more than 30 gears, manufactured with extraordinary precision.[9]

2.3 Slide rules

One of the most common, simplest and well-known analog computers is the *slide rule*,[10] which comes in basically three configurations, *linear, circular*, and helical.[11] The basic idea of a slide rule is to reduce the problem of multiplication and division to that of addition and subtraction by employing logarithmically divided scales that may be displaced accordingly to each other in a lateral direction. The analog setup in this case is to mechanize the relation

$$\log(ab) = \log(a) + \log(b)$$

by using two logarithmic scales. Following the development of the logarithm by John Napier[12] and Henry Briggs,[13] who introduced the base 10 for logarithms, it was William Oughtred[14] who described the principle of the slide rule in his seminal two publications *The Circles of Proportion, and the Horizontall Instrument* and *Two rulers of proportion*.[15] An early *scale* is shown in →figure 2.2. Working with such a scale required a divider to transfer lengths from one of its engraved scales to another, a tedious and error-prone process that was greatly simplified by the introduction of a slider containing several scales, which led to the then ubiquitous slide rule.



Figure 2.2. Typical scale (probably 18th century)

→Figure 2.3 shows one of the last, most complex and most versatile slide rules ever built, a Faber Castell 2/83N. Its three main parts are clearly visible:

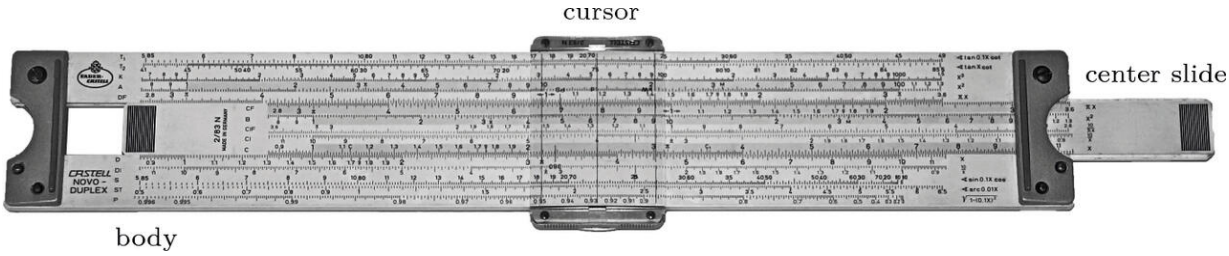


Figure 2.3. Faber Castell slide rule model 2/83N

– The *body*, which consists of the top and bottom *stator* or *stock*. The two stators are held together by two *end braces* or *end brackets*.

– The *center slide*, which can be moved laterally with respect to the body.
– The *cursor*, which slides in grooves of the body.

While simple slide rules only feature a couple of scales, complex ones as the 2/83N have up to 30 and more scales, which implement functions far beyond multiplication and division.[16] Using these scales, trigonometric functions, exponentiation, etc., can be evaluated. Until pocket calculators took over in the 1970s,[17] slide rules were as widely used as they were centuries ago. The following quotation from Josef Vojtěch Sedláček[18] shows this quite strikingly:[19]

> "*It is said that the use of the slide rule in England is so widespread that no tailor makes a pair of trousers without including a pocket just for carrying a 'sliding rule'. During such a time, it is difficult to understand why the slide rule does not enjoy such well-deserved recognition in our own country*."

Slide rules were essential tools for the scientific and technological progress of the last three centuries ranging from mathematics, civil engineering, commercial applications, electronics, chemistry, life sciences, etc., up to applications in aerospace technology.[20,21]

Although they were rendered more or less obsolete by pocket calculators 50 years ago, there are still areas of application where slide rules are employed regularly. For example, many aviators still use a flight computer like the E-6B, a special form of a circular slide rule, that allows the calculation of ground speed, i. e., the speed of an aircraft corrected for wind effects, and many other crucial parameters.[22]

→ Figure 2.4 shows a strange special purpose circular slide rule that was deployed in large amounts during the Cold War – a *Nuclear Weapon Effects Computer*, which allowed rough estimates of fatalities and damage should a nuclear air burst occur.

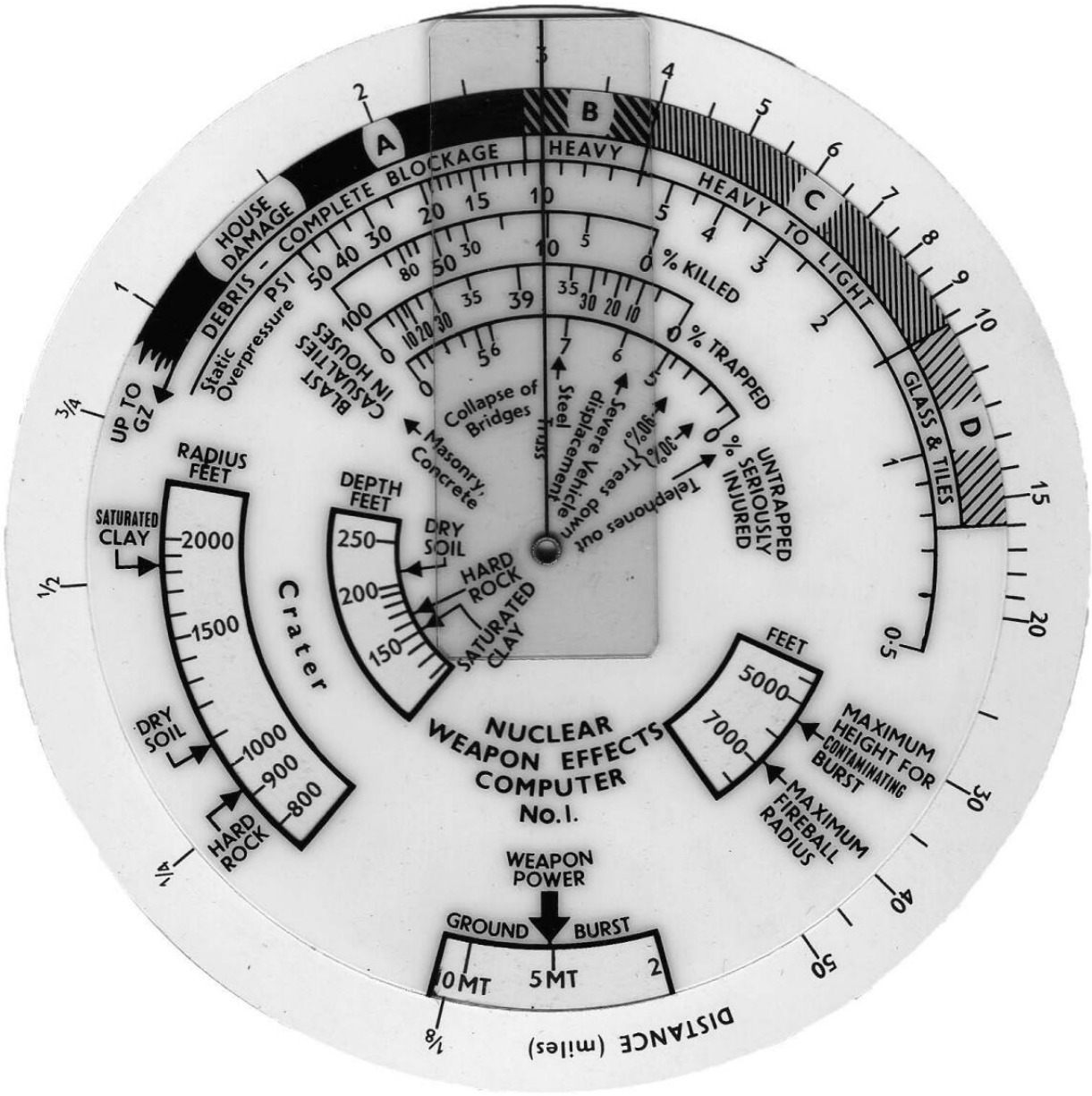


Figure 2.4. Nuclear Weapon Effects Computer

2.4 Planimeters

Planimeters are fascinating instruments. Their purpose, most aptly described by [→ Henrici 1894, p. 497], is the following:

> "*The object of a planimeter is to measure an area; it has, therefore, to solve a geometrical problem by mechanical means*."

Measuring areas enclosed by some “good-natured” boundary curve is an important task in many branches of science as well as in commercial applications, registers of real estate and many more. A typical early application was to analyze pressure/volume indicator diagrams[23] as those written by recording steam engine indicators,[24] which requires the determination of the area enclosed by a curve, which is either plotted in a Cartesian or more often a polar coordinate system. A simple and direct method for performing this task is to cut out the area to be determined and weigh the resulting piece of paper yielding quite good results. Although this method is sometimes still used by chemistry students who regularly have to determine integrals over curves generated by spectrometers and the like, it is not really suitable for everyday usage.

As early as 1814 J. M. Hermann,[27] a Bavarian engineer, invented a planimeter that was built, after improvement by Lämmle, in about 1817. Unfortunately, this instrument seems to have gone unnoticed by his contemporaries and had no obvious influence on subsequent developments.[28] In 1824 an Italian professor for mathematics, Tito Gonnella,[29] invented a *wheel-and-cone planimeter* that used a friction-wheel integrator (see →section 2.5.3) to perform the necessary integration.[30] The first planimeter that was put into production was a device developed by a Swiss engineer named Johannes Oppikofer,[31] who developed two wheel-and-cone planimeters in 1827 and 1836 and a planimeter based on a friction-wheel rolling on a disk in 1849. In fact, there is a plethora of different planimeter principles and implementation variants.

The most successful type of planimeter is the *polar planimeter* that was developed in 1854 by the Swiss mathematician Jacob Amsler-Laffon.[32] →Figure 2.5 shows a typical polar planimeter,[33] which is of a much simpler construction than most of the other instrument types.[34]

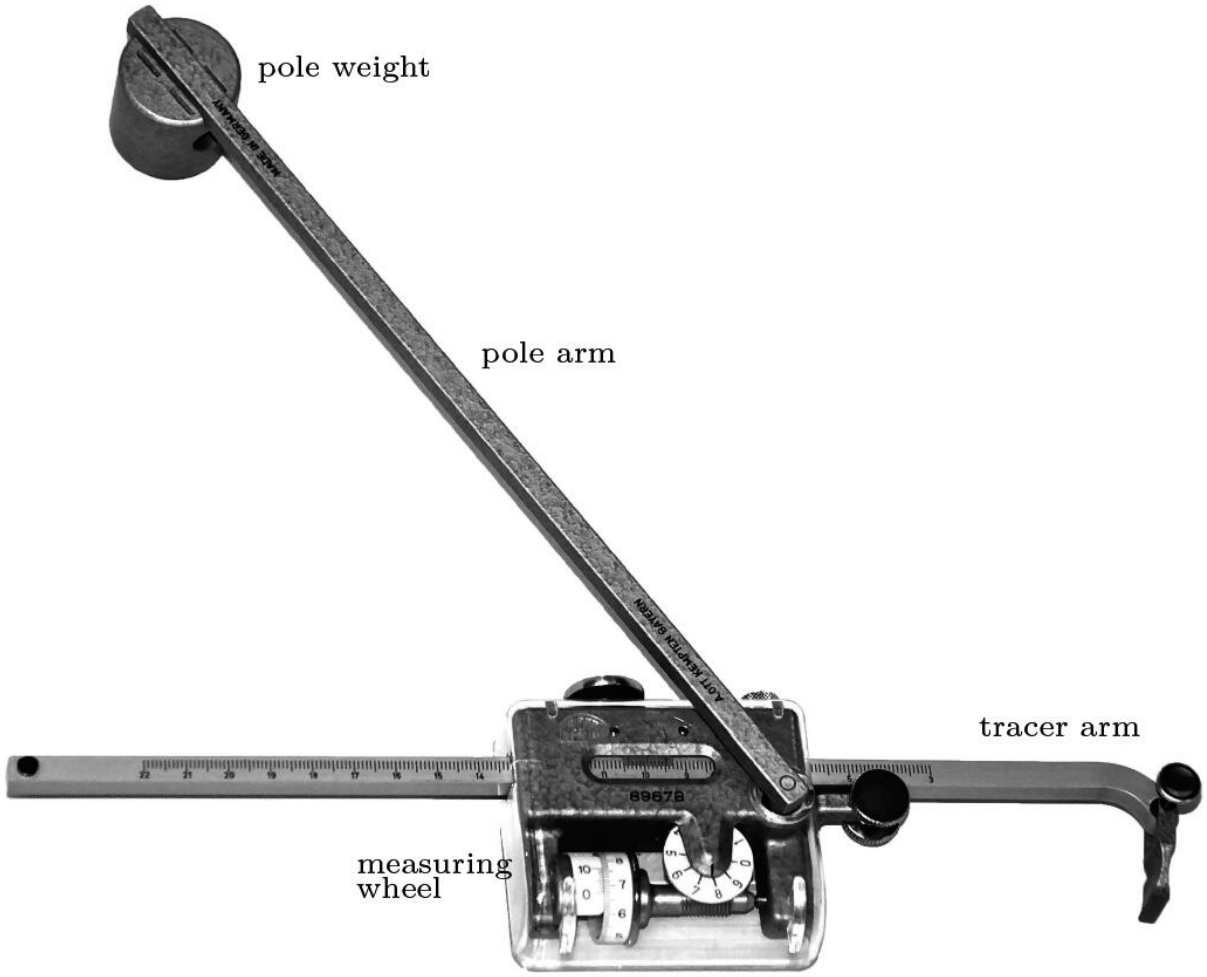


Figure 2.5. Typical polar planimeter made by A. Ott, Bavaria, Germany

The basis of operation for planimeters in general is Green's theorem, which relates a double integral over a closed region, i. e., the area to be determined, to a line integral over the boundary of this region.[35] Thus, a planimeter is a mechanization of

$$\iint \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) \mathrm{d}A = \oint F \mathrm{d}r.$$

Choosing *P* and *Q* in a way that the difference under the left integral equals one yields the area sought. Interestingly, it seems that the first explanation of the operation of planimeters using Green's formula wasn't given until [→Ascoli 1947].[36]

The main parts of a polar planimeter like the one shown in →figure 2.5 are two arms connected with an elbow that is restricted to move on the circumference of a circle. The end of one arm, the *pole arm*, is fixed at a point called *pole* by means of a *pole weight* or a needle, while the end of the other arm, the *tracer arm*, can be moved freely by hand. This end is often formed as a needle or a magnifying lens to simplify tracing the boundary curve of the area to be measured. At the elbow is the integrating or measuring wheel.[37]

To measure the area of a circumscribed figure, the pole is fixed either outside or inside the figure, depending on its size. Then the measuring wheel with its vernier and the counting dial are reset with the needle or lens placed on the starting point of the boundary. Then the boundary line is followed manually in clockwise direction. After reaching the starting point again, the area can be read from the counting dial, the wheel and the vernier. If extended precision is necessary, the same procedure can be applied a second time while following the boundary line counterclockwise,[38] which, of course, yields a result with the opposite sign, the complement.

By setting the position of the elbow, which can normally be shifted along the tracer arm, various scaling values can be set, so areas can be measured in inches as well as in centimeters without the necessity to explicitly convert units.[39]

Determining large areas can be problematic since the counting dial can only represent one significant digit. Thus, as early as 1961 planimeters coupled with electronic counters were built. [→Zuse Z80 1961] describes a linear planimeter[40] that was coupled to an electronic counter capable of processing up to 250 000 impulses per second. The pickup from the measuring wheel was done photoelectrically, thus allowing even better precision than traditional mechanical instruments.

[→Lewin 1972] describes another development that was patented in 1972: Here the position of the tracer arm is sensed by two linear potentiometers generating voltages directly proportional to the current (*x, y*)-position of the tracer. These voltages in turn control an oscillator and some monostable devices. The integration process is then performed basically by a chain of decade counters. This scheme proved to be too complicated and costly for the market. Another quite recent

development is described by [→ LIGHT 1975]. Here the position of a tracer needle or the like is determined by conductive foils, which are placed under the chart containing the curve to be integrated over.[41]

As old as planimeters are, some companies still manufacture polar and linear planimeters, which achieve accuracies of about 1‰. These instruments are still used regularly for surveying and mapping, for determining the area of furs and fabric, etc.

2.5 Mechanical computing elements

All of the instruments shown in the preceding sections are specialized analog computers, capable only of solving just one distinct problem each. This is obviously caused by their fixed structure – a slide rule can only add lengths, so the only variation possible is that of employing different scales to extend this basic operation to multiplication and division and many more.

In the same sense planimeters are specialized instruments for only determining areas. The following sections will now cover some basic mechanical computing elements that can be used to build true – in the sense of their reconfigurability – analog computers, *differential analyzers*.[42]

Typically, mechanical analog computers represent values by rotations of shafts, which interconnect the various computing elements, or by positions of linkages. As simple as most mechanical computing elements seem, they are quite powerful tools. Precision up to 10^{-4} is possible given precise machining of the parts involved and clever scaling of the equations to be solved. In fact, mechanical analog computers even have one advantage over analog electronic analog computers: The former can integrate over every variable whilst the latter can only integrate over time, which then requires some ingenuity to solve partial differential equations[43] and other problems.

2.5.1 Function generation

A common task in simulations is the generation of functions, which are either defined analytically or by measured values. Functions of a single

variable are easily implemented using cams driven by a shaft, the angular position of which represents the input variable. A follower measuring the position of the cam's surface yields the desired function value. The position of this pin can be translated into a rotational motion by a rack and pinion arrangement so this output value can be used as input for another element expecting an angular shaft position as its input value.[44]

Another type of generator for a function of one variable is shown in →figure 2.6. This is a *squaring cam*, which is used to generate a square function $f(x) = x^2$. If dx_1 denotes the element of rotation of the input shaft, the resulting movement of the string wound around the cone is $r_1 dx_1$ where r_1 denotes the average diameter of the cone at the current position of the string. With r_2 denoting the diameter of the drum, the drum rotation resulting from dx_1 is

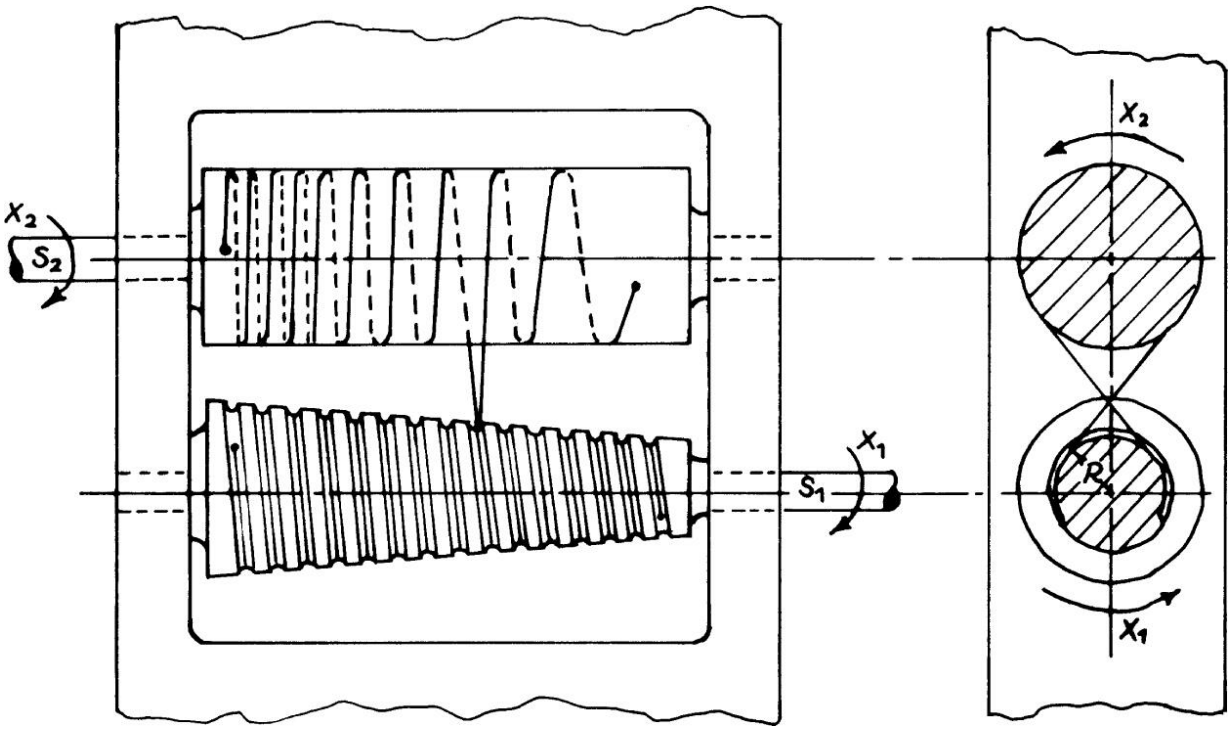


Figure 2.6. Squaring cam yielding $x_2 = x^2$ (cf. [→Svoboda 1948, p. 22])

$$dx_2 = -\frac{r_1 dx_1}{r_2}$$

r_1 is proportional to the angle of rotation x_1 thus

$$dx_2 = -\frac{kx_1 dx_1}{r_2}$$

with *k* being a constant of proportionality. Integration finally yields

$$x_2 = -\frac{k}{2r_2} x_1^2$$

which is the desired square function. This mechanism does, of course, not work down to $x_1 = 0$ – if this is necessary, it may be combined with a differential gear as described in →section 2.5.2.[45]

In many cases, functions of two variables are necessary – →figures 2.7 and →2.8 show a *barrel cam*[46] used to implement a function $x_3 = f(x_1, x_2)$. The value of the input variable x_1 controls the lateral displacement of the three-dimensional barrel cam while the second input variable, x_2, controls the angular position of the cam. The output value is sensed by a follower that gauges the cam's surface. Other implementations feature a movable sensing pin carriage instead of a barrel cam that can be shifted laterally, which allows for a more compact design.

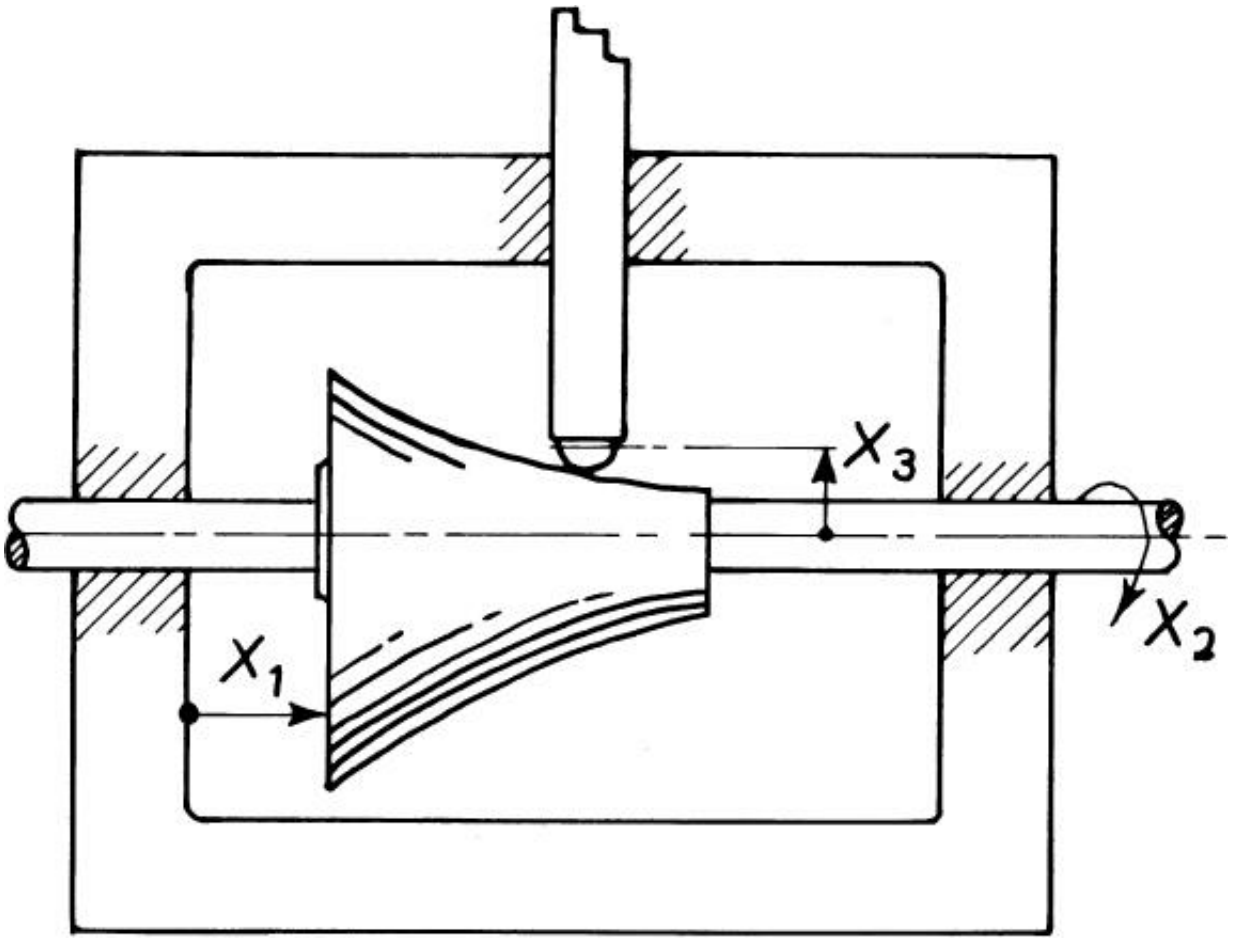


Figure 2.7. Three-dimensional cam as a function generator for a function of two variables $x_3 = f(x_1, x_2)$ (cf. [→ SVOBODA 1948, p. 23])

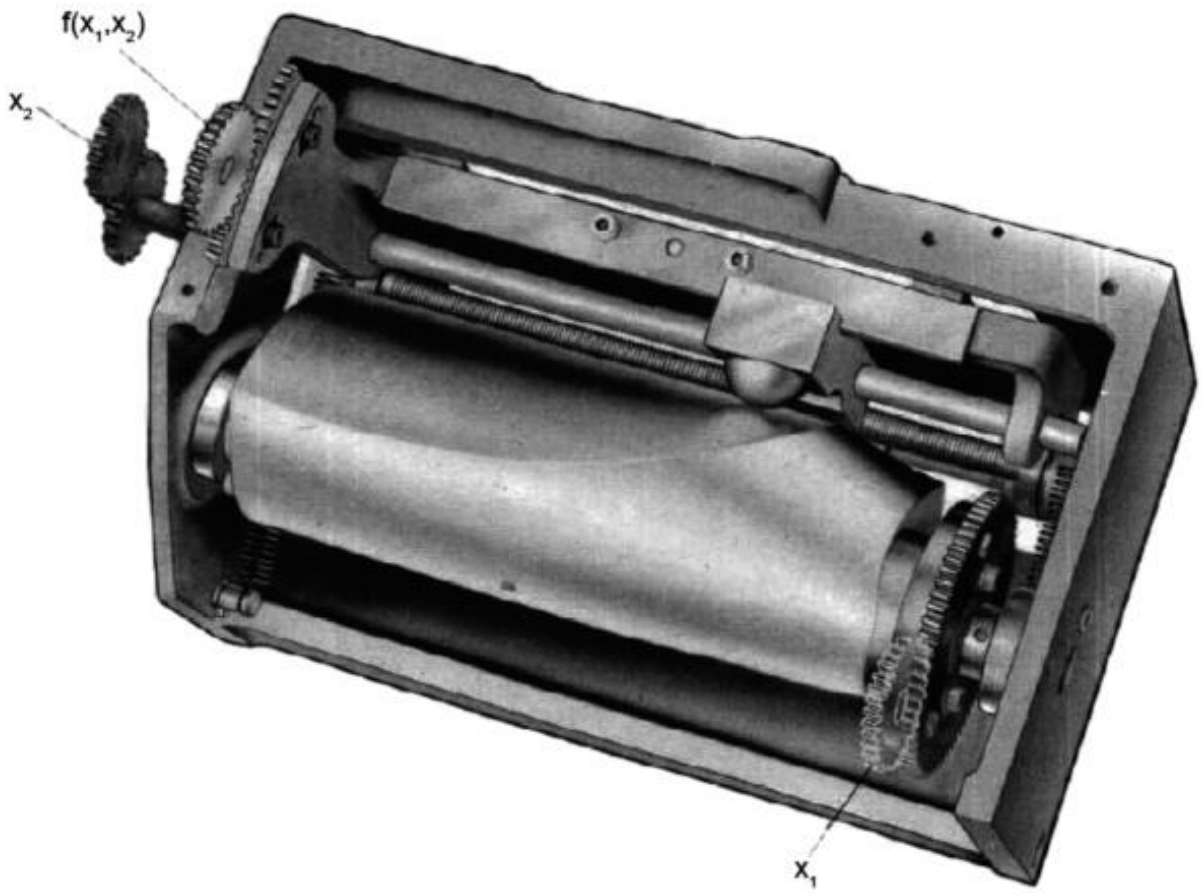


Figure 2.8. The barrel cam – a practical implementation of a function generator yielding $f(x_1, x_2)$ (cf. [→ Bureau of Ordnance 1940, p. 54])

Barrel cams like this were often used in mechanical fire control systems but were also used in some cases in electronic analog computers since the generation of functions of two variables is a complicated task for such a machine. In this case the input shafts for x_1 and x_2 are controlled by electronic servo mechanisms while the output sensing pin is connected to a potentiometer, which delivers a voltage proportional to the desired function value.[47]

An important topic regarding cams and barrel cams is that their surface has to conform to several mechanical constraints to ensure that the sensing pin can follow the surface smoothly and that the pin can never block the cam in its movement due to steep slopes or grooves.

This might sometimes conflict with the mathematical requirements of the computer setup. In such cases, functions may be generated by solving suitable differential equations, which normally requires many more additional computing elements than a straightforward barrel cam implementation.

2.5.2 Differential gears

A barrel cam function generator like this could, in principle, be used to add or subtract two variables, but a simpler, cheaper and more precise mechanism to implement this basic operation exists in form of the *bevel-gear differential*[48] shown in →figure 2.9. This device adds two shaft rotations x_1 and x_2 with some scaling parameters s_1 and s_2, which result from the actual design of the differential, yielding $x_3 = s_1x_1 + s_2x_2$.

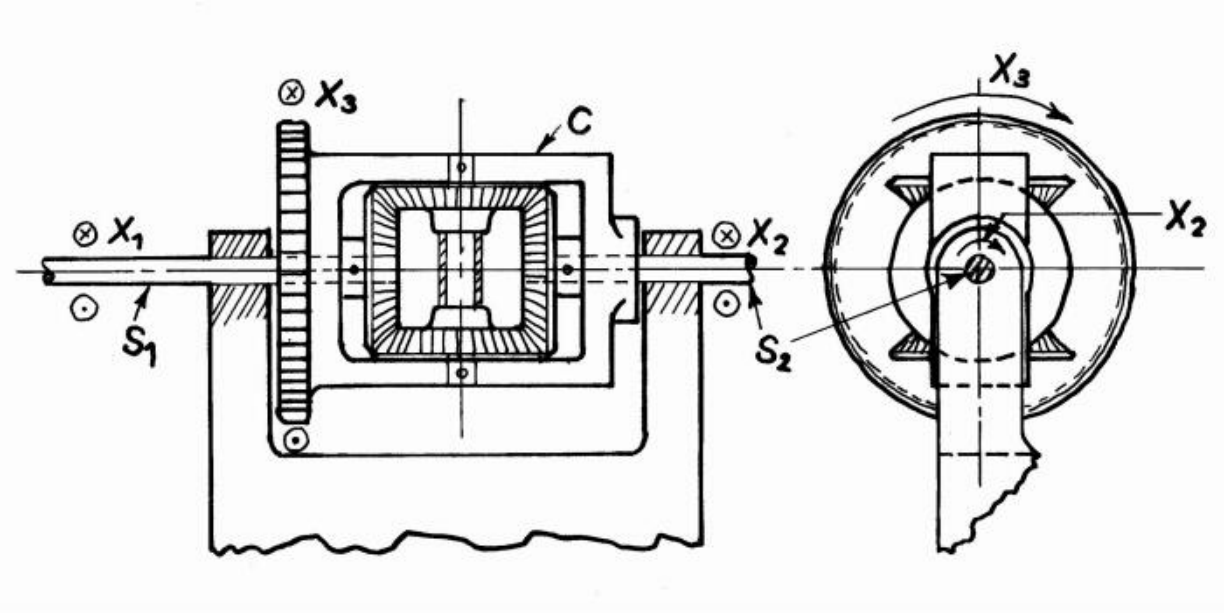


Figure 2.9. Structure of a bevel-gear differential (cf. [→Svoboda 1948, p. 6])

A practical example of such a differential gear is shown in →figure 2.10. This was used in the dead reckoning computer PHI-4A-2 of a Starfighter jet.[49]



Figure 2.10. A differential gear from the dead reckoning computer PHI-4A-2 used in Starfighter jets

2.5.3 Integrators

In contrast to most other machines, integration is a fundamental as well as natural operation for an analog computer. Using mechanical computing elements, the operating principles of integration are remarkably simple.[50] The first mechanical integrators were developed in the early 19th century: In 1814 Johann Martin Hermann developed an integrator consisting of a cone with a wheel rolling on its surface.[51] The position of the wheel on the envelope of the cone represents the values of the function to be integrated while the rotation of the cone corresponds to the variable of integration.[52]

Basically, an integrator mechanizes the calculation of integrals like

$$x_3 = \int x_1 dx_2$$

where x_2 is represented by the rotation of the integrator disk (or cone) while the values of x_1 control the position of the friction-wheel rolling on the disk surface. →Figure 2.11 shows the structure of such an integrator.

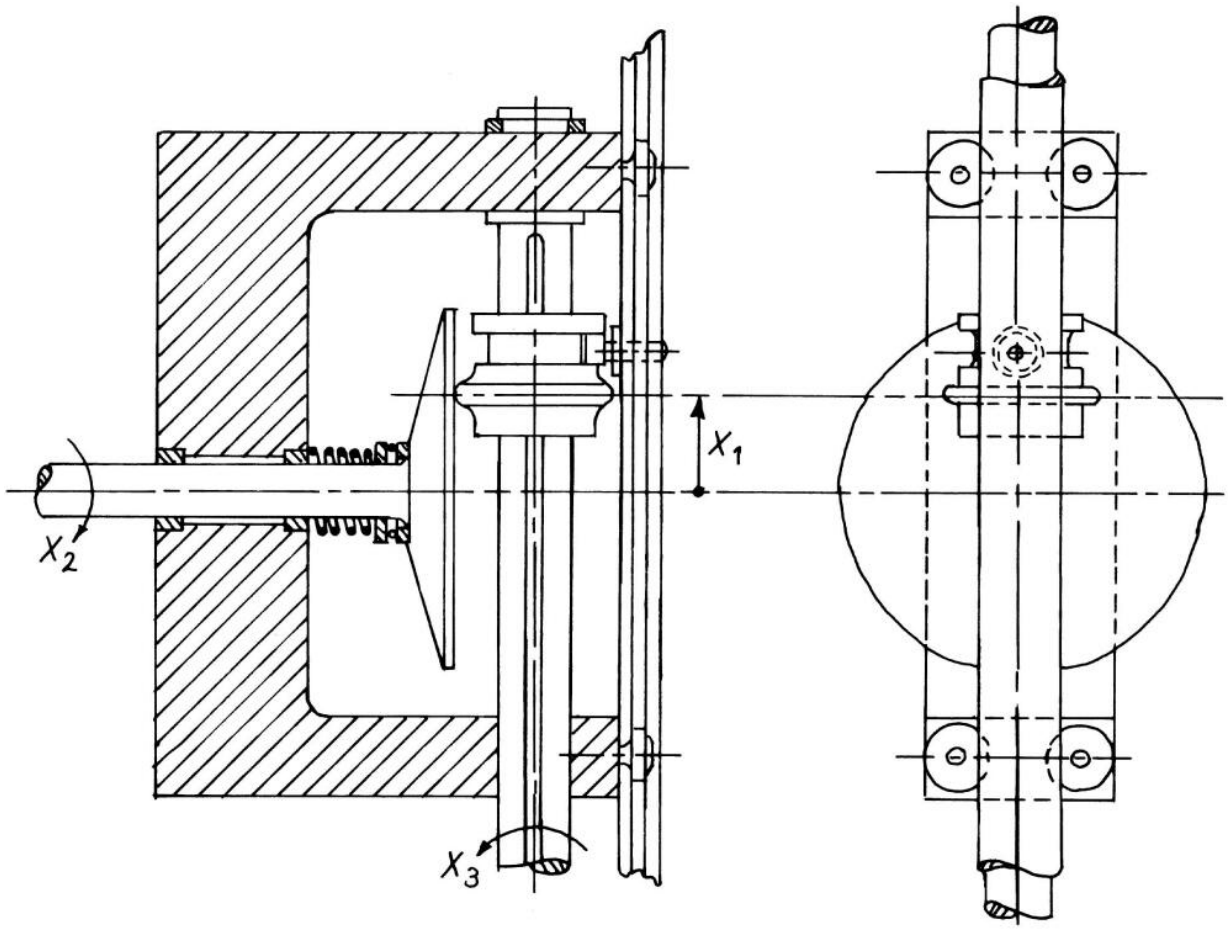


Figure 2.11. A friction-wheel integrator (cf. [→ SVOBODA 1948, p. 24])

Obviously it is dx_3 = kx_1dx_1 where $k = 1/r$ denotes the radius of the friction-wheel, thus yielding

$$x_3 = \int_{x_{2_0}}^{x_{2_1}} \frac{1}{r} x_1 dx_2.$$

→ Figure 2.12 shows a friction-wheel integrator from the *Oslo analyzer*, which was built from 1938–1942 – at its time one of the largest, most precise, and most powerful differential analyzers in the world.[53] On the right-hand side[54] the rotating disk *S* and the friction-wheel *h* with its associated shaft are visible. The wheel position on the surface of the disk with respect to the disk center corresponds to *F* (*x*).

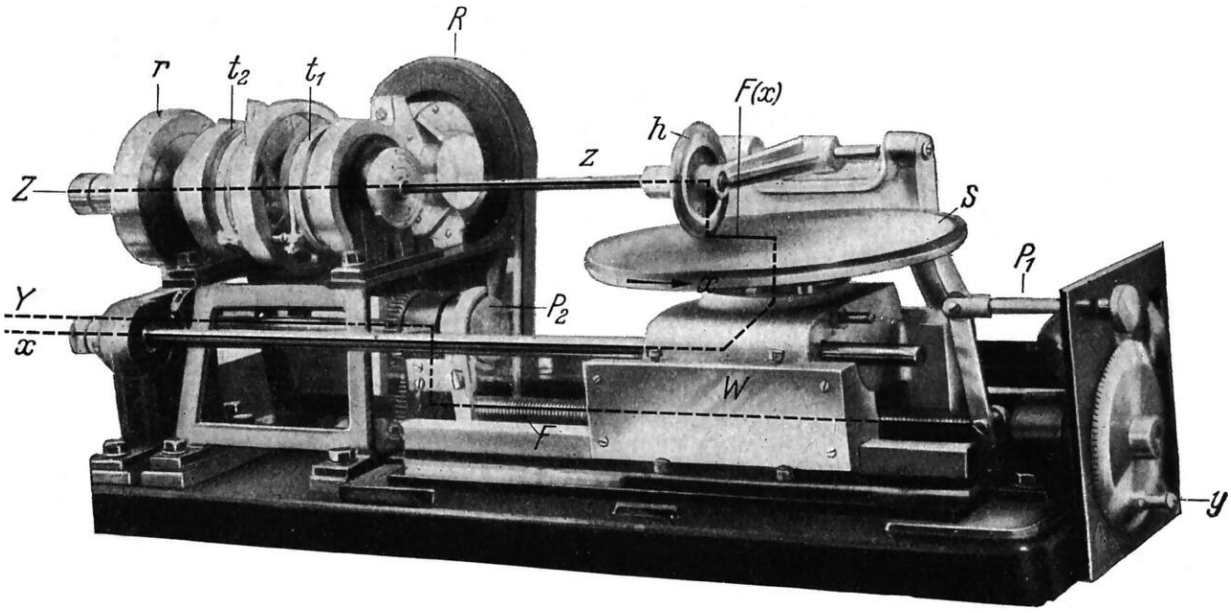


Figure 2.12. Integrator from the Oslo differential analyzer, see → section 2.8 ([→ Willers 1943, p. 237])

In 1876, James Thomson,[55] the brother of William Thomson[56] – later Lord Kelvin – developed an integrator that replaced the friction-wheel with a steel sphere that runs in a movable cage (controlled by x_1).[57] This sphere provides a frictional connection between the rotating disk and a cylinder that acts as a pickup for the integration result x_3. → Figure 2.13 shows a variation of this implementation – two stacked balls, instead of a single sphere, are guided in a movable cage. Such integrators are known as *double-ball integrators*.[58]

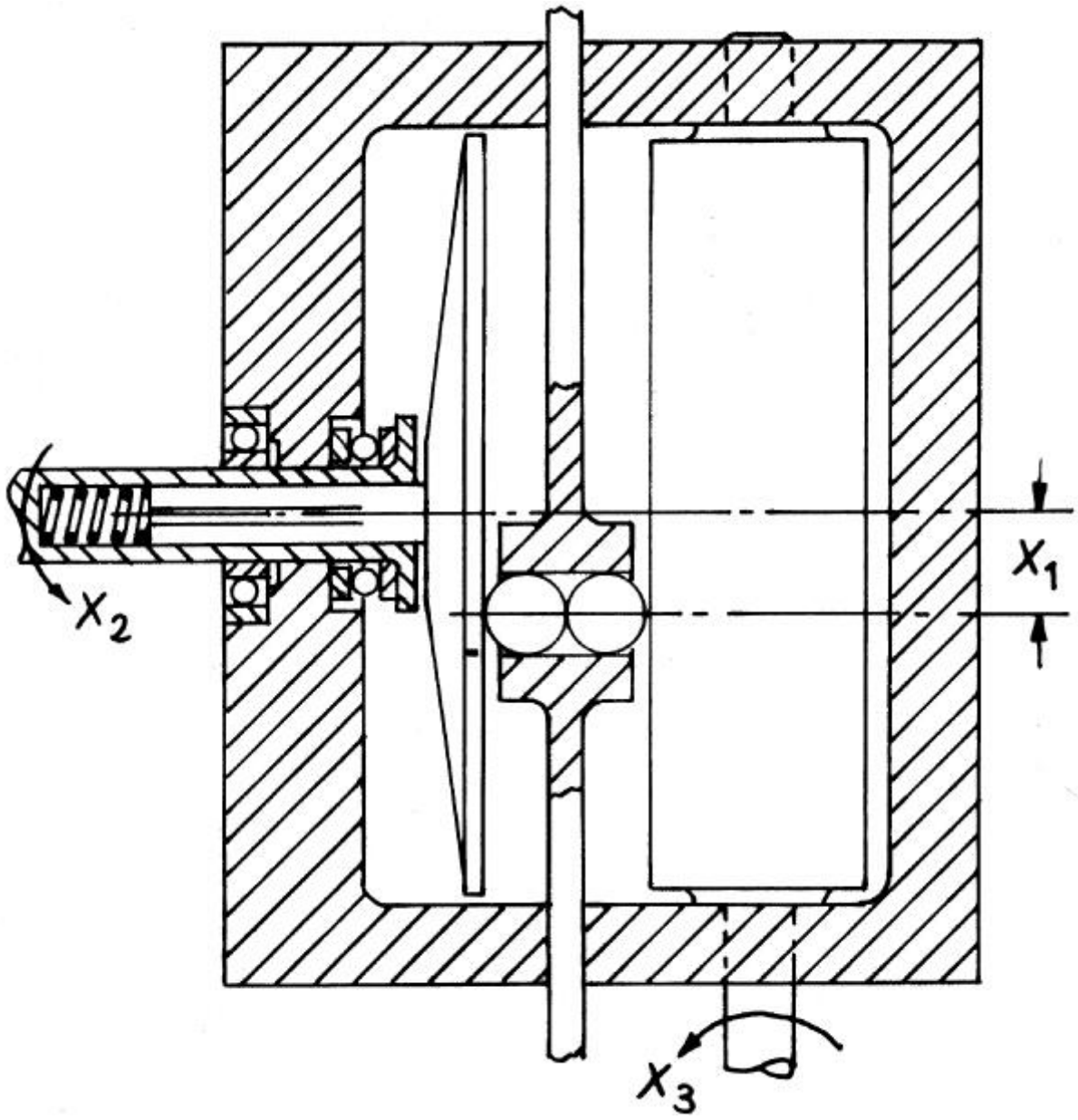


Figure 2.13. Basic principle of operation of a classic double-ball integrator (see [→ Svoboda 1948, p. 25])

Based on this integrator developed by his brother, Lord Kelvin devised the idea of a machine suitable for solving differential equations. The basic concept was to start with the highest derivative in the differential equation to be solved and to generate all of the necessary lower derivatives by repeated integration. Using these derivatives, the highest derivative – that was the starting point for this procedure – can now be

synthesized by combining the lower derivatives in an appropriate way.[59] The discovery of this method is described in [→Thomson 1876] as follows:

> "*But then came a pleasing surprise. Compel agreement between the function fed into the double machine and that given out by it [...] The motion of each will thus be necessarily a solution of [the equation to be solved]. Thus I was led to the conclusion, which was unexpected; and it seems to me very remarkable that the general differential equation of the second order with variable coefficients may be rigorously, continuously, and in a single process solved by a machine*."

Interestingly, Kelvin did not build a practical machine based on this insight, which is all the more puzzling since a (double) ball integrator can transfer sufficient torque to allow a small number of such devices to be chained. Trying to implement this scheme using friction-wheel integrators would have required torque amplifiers, as these integrators can only transmit tiny amounts of torque. Consequently, this brilliant idea of setting up an analog to represent differential equations fell into oblivion. This happened again 45 years later when Udo Knorr[60] published a similar idea in 1921.[61] He explicitly noted that coupled integrators can be used to solve a large group of differential equations of high degree.[62]

2.5.4 Multipliers

Multiplication is a quite difficult task for a mechanical analog computer. While multiplication with a constant can be easily performed with an appropriate gear mechanism or with a friction-wheel integrator where the position of the friction-wheel corresponds to the multiplier, the generalized operation where both input variables may vary is much more difficult to implement.

One common implementation is based on the product rule known from calculus. From (*uv*)' = *u*'*v* + *uv*' it follows by integration that

$$uv = \int u dv + \int v du \tag{2.1}$$

Thus, the multiplication of two variables can be implemented using two interconnected integrators.[63]

→Figure 2.14 shows the structure of a *slide multiplier*. It consists of two carriages sliding vertically and horizontally respectively and a lever

that rotates around a center pin that is fixed to the enclosure. Both carriages and the lever are coupled with another pin that runs in grooves of these three elements. The input variables are x_1, represented by the displacement of the rotating lever's end, and x_2, which corresponds to the horizontal displacement of the second slider. The multiplication result $x_3 = kx_1x_2$, where k denotes a scaling factor that depends on the dimensions of the multiplier, is then available as the vertical displacement of the first slider.

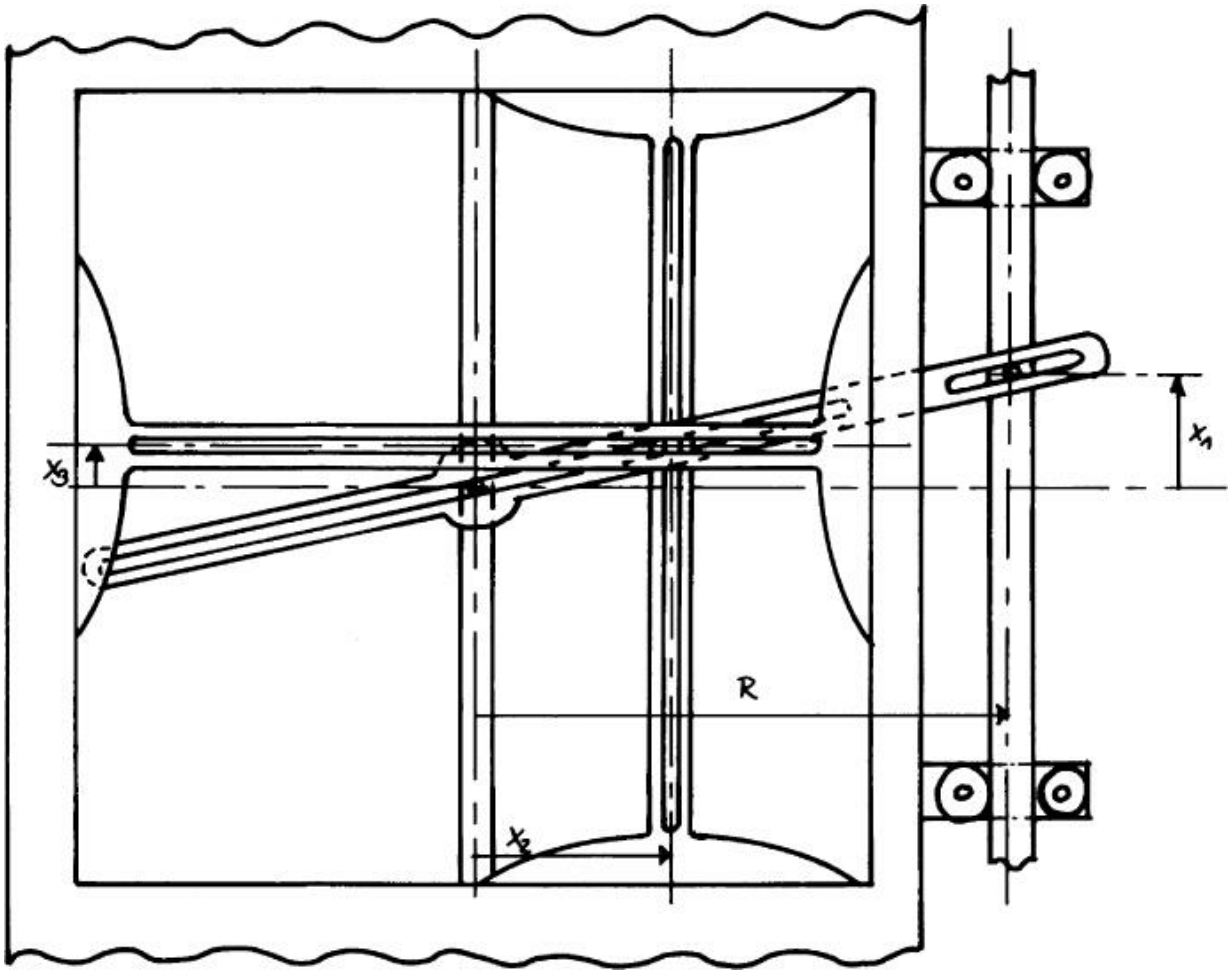


Figure 2.14. Basic structure of a slide multiplier (cf. [→ Svoboda 1948, p. 12])

2.6 Harmonic synthesizers and analyzers

Although Lord Kelvin did not attempt to build a true general purpose mechanical analog computer, he did build some quite complex special purpose analog computers[64] to predict tides by means of harmonic synthesis.[65] Even the earliest sailors had a genuine interest in tide prediction since accurate predictions lead to fewer lay days in harbors. A good description of the term *tide* is given by Kelvin himself:[66]

> "*The tides have something to do with motion of the sea. Rise and fall of the sea is sometimes called a tide; but I see, in the Admiralty Chart of the Firth of Clyde, the whole space between Ailsa Craig and the Ayrshire coast marked 'very little tide here'. Now, we find there a good ten feet rise and fall, and yet we are authoritatively told there is very little tide. The truth is, the word 'tide' as used by sailors at sea means horizontal motion of water; but when used by landsmen or sailors in port, it means vertical motion of the water*."

Tides are caused and influenced by the superposition of a number of effects that are, in fact, harmonic oscillations with different amplitudes, frequencies, and phases.[67] The most important of these effects are the earth's rotation around its own axis, the rotation of the earth around the sun, the rotation of the moon around the earth, the precession of the moon's perigee, the precession of the plane of the moon's orbit, etc.

In 1872 Kelvin developed the first *harmonic synthesizer*, a specialized analog computer for generating harmonics and adding these together to generate a tide prediction.[68] This machine took ten partial tides into account. Its basic construction is shown in →figure 2.15.[69] Using this machine it took only four hours to compute the 1 400 tides that occur during a year for a given harbor. Performing the same task manually took several months.[70]

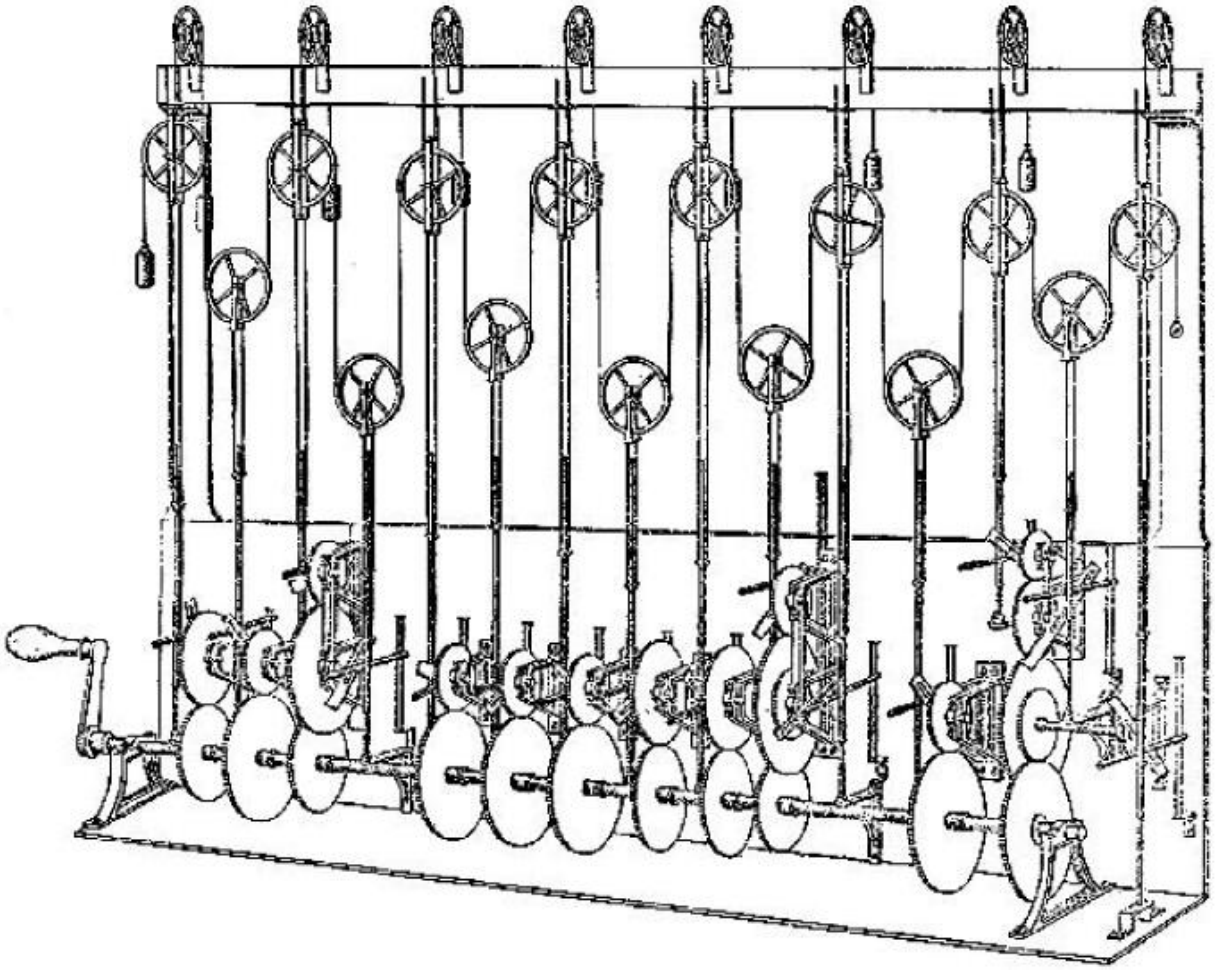


Figure 2.15. Principle of operation of KELVIN's tide predictor (see [→THOMSON 1911])

Generally speaking, a harmonic synthesizer like a tide predictor generates a function *f*(*θ*) based on a given set of harmonics:[71]

$$f(\theta) = a_0 + a_1 \sin(\theta + b_1) + a_2 \sin(2\theta + b_2) + \cdots + a_n \sin(n\theta + b_n)$$

The basic harmonic functions are traditionally generated using a *scotch yoke mechanism* as shown in →figure 2.16. It consists of a boom that is mounted on a shaft so that it can rotate around this mounting point. The angular position of this shaft represents the input variable *θ*. The free end of the boom guides two carriages that are restricted to perform horizontal respectively vertical movements only. The movements of these carriages then represent the values *r* sin(*θ*) and *r* cos(*θ*) where *r*

denotes the radius of the circle described by the free end of the rotating boom.

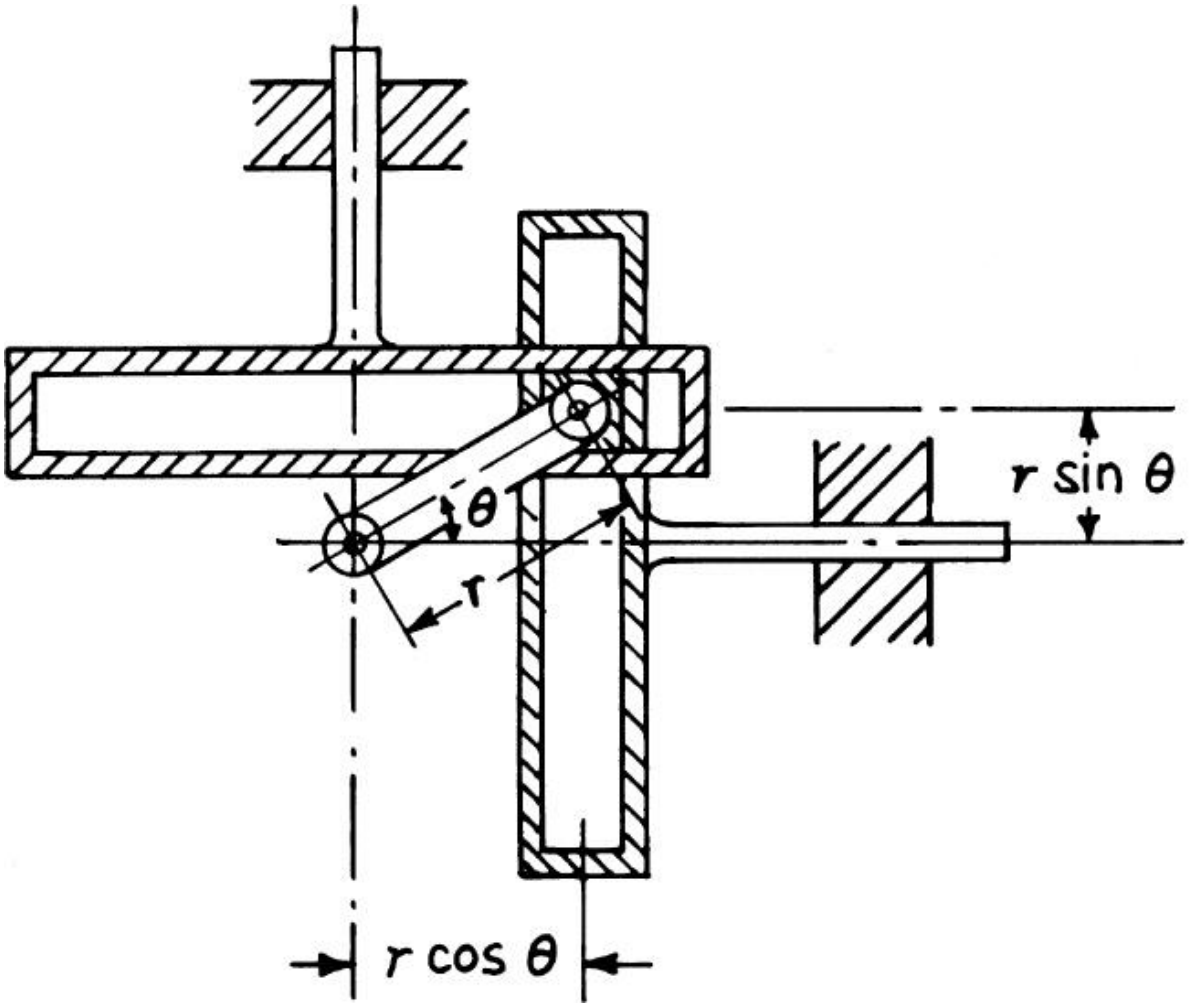


Figure 2.16. A scotch yoke mechanism (cf. [→ KARPLUS et al. 1958, p. 242])

Since the values $r\cos(\theta)$ and $r\sin(\theta)$ are represented by linear displacements, harmonics generated this way cannot be added together using a differential gear. Instead, a steel band of constant length is used, as can be seen in →figure 2.15. While KELVIN's tide predictor only generated a single output function by the superposition of the various harmonics, later harmonic synthesizers often had a more generalized structure like that shown in →figure 2.17.

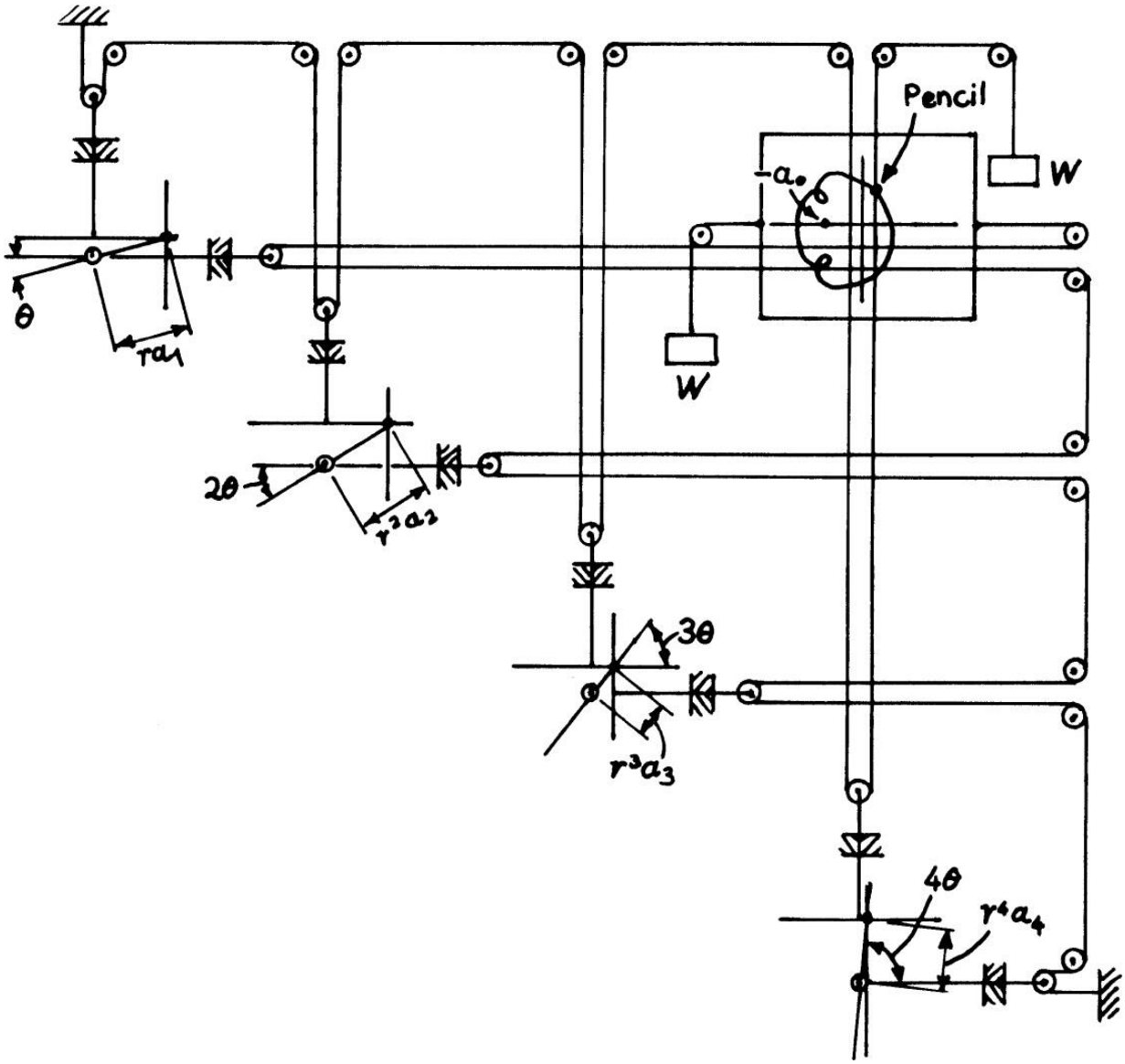


Figure 2.17. Harmonic synthesizer for a fourth-degree equation (cf. [→KARPLUS et al. 1958, p. 242])

This fourth degree harmonic synthesizer consists of four scotch yoke mechanisms as shown in →figure 2.16, each of which generates a sine/cosine-pair and is parameterized by four crank length settings r, r^2, r^3 and r^4 (this rather unusual notation follows [→KARPLUS et al. 1958, p. 242]). Two continuous tapes are threaded through the system adding all sine- and all cosine-values and driving the two coordinate inputs of a plotting mechanism.[72]

Later developments include a device described in [→ Redheffer 1953], which still used scotch yoke mechanisms but with an electronic pickup in form of potentiometers that sensed the positions of the sine/cosine carriages. The output voltages of these potentiometers were then added by means of simple electronic networks and could directly be used to control an *xy*-plotter.

The inverse function of a harmonic synthesizer is performed by a *harmonic analyzer*.[73] These devices perform a Fourier analysis, i. e., they determine the amplitudes and phases of the *harmonics* that comprise a given input signal. The first such harmonic analyzer was developed in 1876 by Lord Kelvin.[74] It consisted of five double-ball integrators and was demonstrated at the Royal Society, configured for a meteorological problem.

Its operation is based on the computation of integrals of the form

$$\int y \cos\left(\tfrac{2\pi\omega x}{c}\right) \mathrm{d}y$$
$$\int y \sin\left(\tfrac{2\pi\omega x}{c}\right) \mathrm{d}y$$

with $\omega \in \{1, 2\}$ (for meteorological applications) or $\omega \in \left\{\frac{39 \cdot 109}{40 \cdot 110}, 1\right\}$ for work on tides.[75]

One of the most complex harmonic analyzers was devised by Albert Abraham Michelson[76] (of Michelson-Morley experiment fame) and Samuel Wesley Stratton.[77] This machine was capable of performing harmonic synthesis as well as analysis with up to 20 harmonic terms – a remarkable feat. Due to its very clever principle of operation and implementation it weighed only 69 kg and could fit on a standard desk.[78] This particular machine came back into the limelight again in the early 2010s when Bill Hammack et al. published a video series detailing on the operation of this device, which they had recently restored.[79]

The determination of Fourier coefficients is also possible using only a planimeter. This startling technique is described in detail in [→ Willers 1943, pp. 171 ff.]. Since this manual process is not only tedious but also time consuming and error-prone, semi-automatic harmonic analyzers based on planimeters have been developed. → Figure 2.18 shows a

planimeter-based harmonic analyzer that was developed and produced by Mader-Ott.[80]

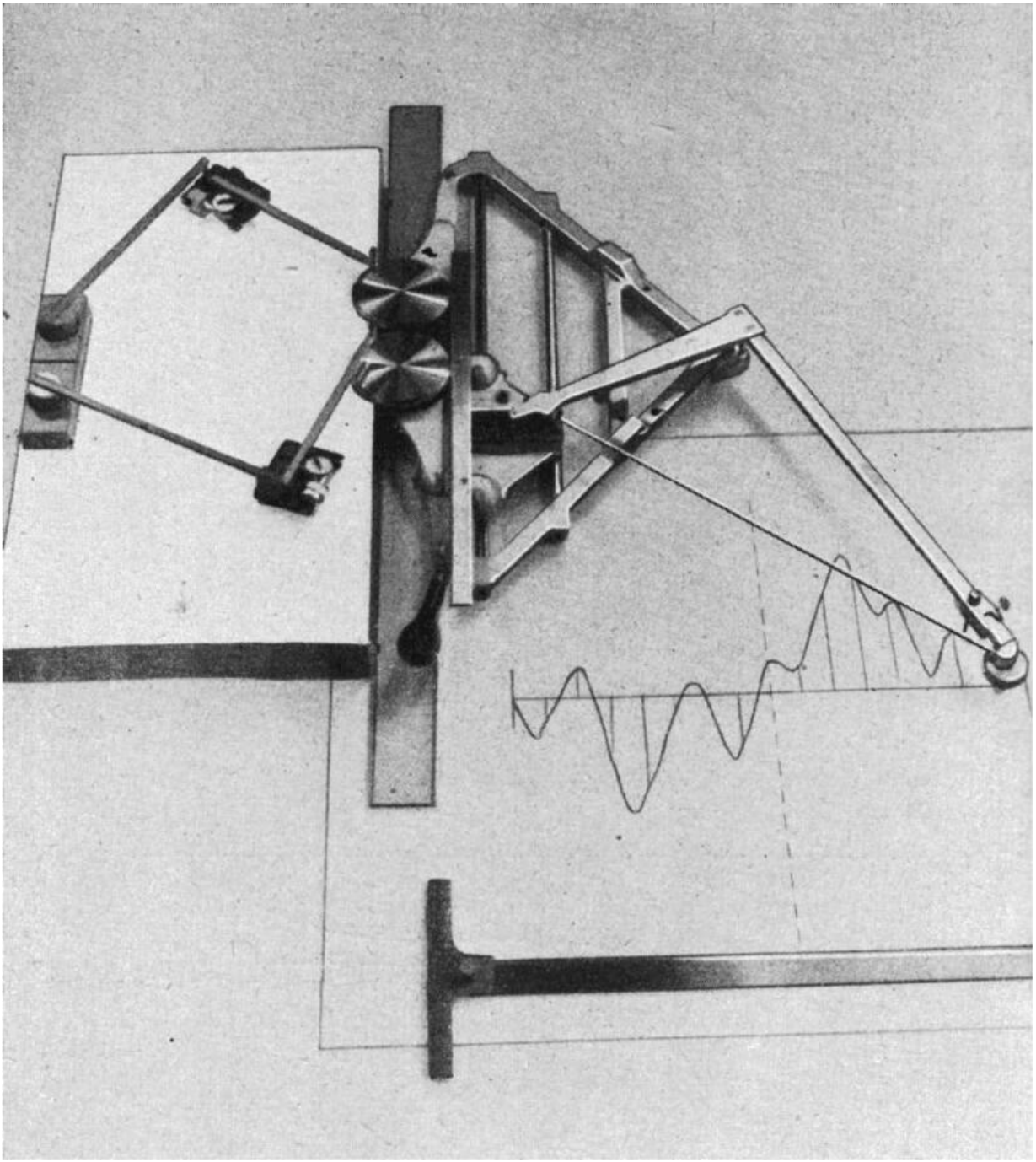


Figure 2.18. Harmonic analyzer made by Mader-Ott (cf. [→Willers 1943, p. 180])

Apart from obvious applications in fields like mechanical engineering where harmonic analyzers were used regularly to analyze complex vibrations and the like, they also played an important role in resource and oil exploration where seismograms must be analyzed to reveal the structure of earth's interior.

2.7 Mechanical fire control systems

Another type of very complex and highly specialized analog computers are *fire control systems*. These military systems perform the following tasks:[81]

> "*Fire control equipment, that takes in indications of targets from optical or radar perception and using extensive calculating equipment puts out directions of bearing and elevation for aiming and time of firing for guns, according to a program that calculates motion of target, motion of the firing vehicle, properties of the air, etc.*"

Until the late 19th century fire control was a simple task given the relatively short ranges of typical gunnery. In previous centuries, firing *broadsides* was a common mode of fighting in naval warfare.[82] During the 19th century shell ranges grew from a few hundred meters up to six miles, which greatly complicated the task of aiming. Long range guns with ranges up to 17 miles were common at the end of World War II[83] – the time of flight of such a shell was in the range of about one minute – resulting in an aiming task so complex that it would have been impossible to hit a target reliably without the support of complex fire control systems, which were implemented as large mechanical analog computers.[84] These machines were not only rugged but also extremely long-lived regarding their service life as [→ Bromley 1984, pp. 1 f.] puts it:

> "*Mechanical analog devices were first used for naval gunnery in World War I, were greatly developed for naval and anti-aircraft gunnery between the wars, were further developed and extended to aircraft systems during World War II, and continued in service in refined versions into the 1970s.*"

Typical effects that have to be taken into account by a fire control system include:

Wind and current effects: Flight times of up to one minute and more make shells susceptible for forces exerted by side winds and the like. Current effects acting on torpedoes are equally, if not even more, complicated to model and predict, mainly due to the long run time of such a weapon.
Heading: Both opponents are normally not at rest but moving (in the case of aircraft very fast).
Environmental conditions: Barometric pressure, temperature, humidity, etc., all influence the path of a shell or torpedo.
Aiming point vs. sensor position: Especially in the case of torpedoes there is normally a mismatch between the position of the sonar system with respect to the periscope and the launching tubes on the side of the attacker and, in the case of sonar contact, the sound source and the desired aiming point on the target vessel.[85]
Signal run time: In case of sonar bearing analysis, the run time of the sound waves can often not be neglected regarding the ships' positions.

→ Figure 2.19 shows the *Torpedo Data Computer Mark-3*[86] developed by the *Arma Corporation*[87] and gives an impression of the immense complexity of such mechanical analog fire control systems. This particular system is deservedly regarded as a masterpiece mechanical analog computer. The two interconnected racks have a width, height, and depth of 160 cm × 115 cm × 100 cm and weigh about 1.5 metric tonnes. The power consumption was quite impressive at 55.7 to 140 A at 115 V operating voltage.

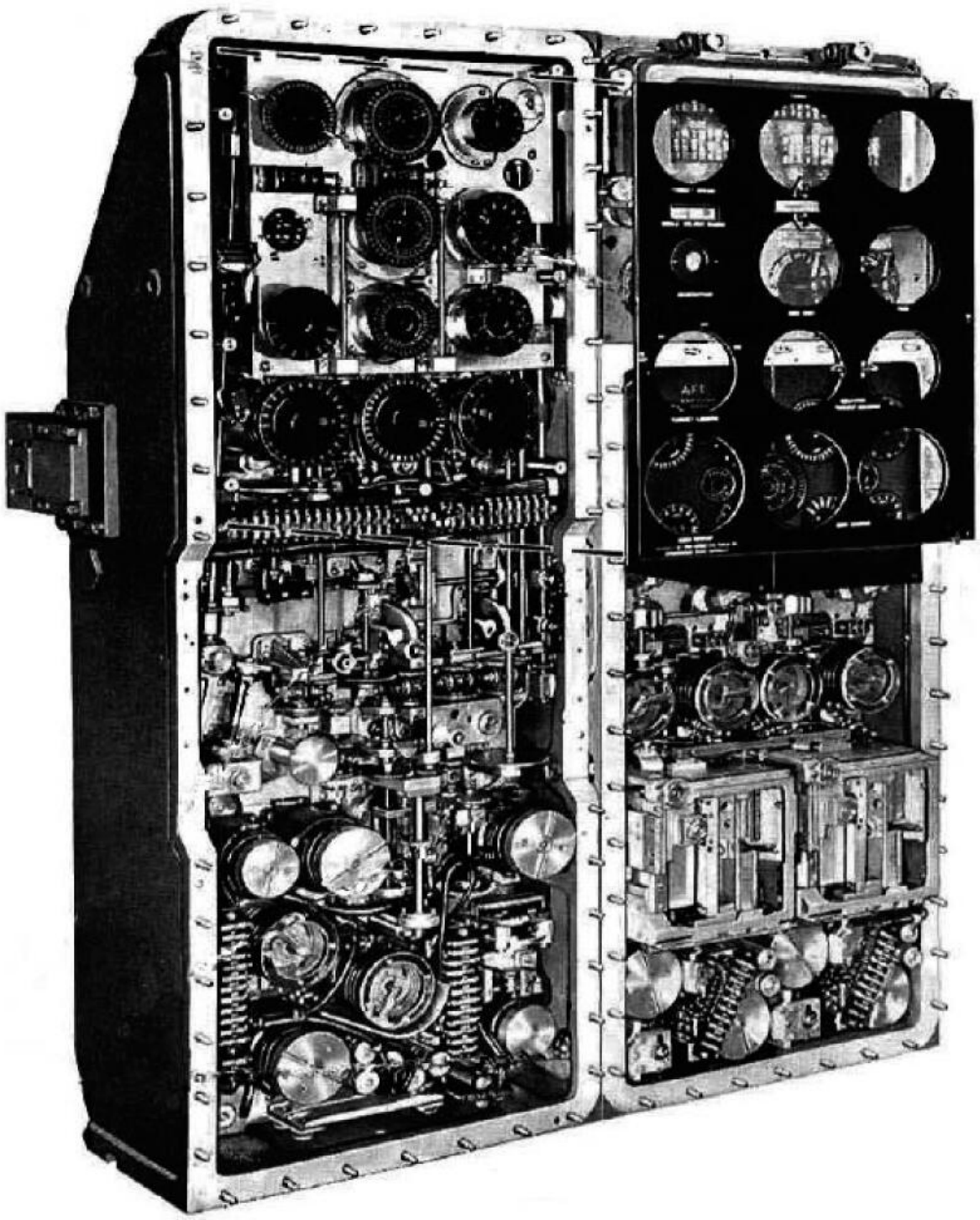


Figure 2.19. TDC Mark-3 with front panels removed (see [→Bureau of Ordnance 1944, p. 34])

A wealth of additional information about fire control systems can be found in [→ Berkeley et al. 1956], [→ Clymer 1993, pp. 286 ff.], [→ Friedman 2008, pp. 16 ff.], [→ Gray et al. 1955], and [→ Admiralty 1943]. [→ Epstein 2024] gives a comprehensive account of technology transfer (theft) with respect to such fire control systems. An overview of the historical development of such systems can be found in [→ murhpy 1960].

2.8 Differential analyzers

A central figure in the development of the mechanical and electromechanical *differential analyzer* was Vannevar Bush,[88] who developed a keen interest in analog computing techniques in the early 20th century.[89] He was not aware of the fundamental works of James Thomson and Lord Kelvin[90] and when he was informed about these previous developments in later years he still claimed the intellectual property on his inventions for himself, stating that "*[i]nventors are supposed to produce operative results*" based on the fact that none of his predecessors had successfully built a working general purpose mechanical analog computer.[91]

In 1925, triggered by the increasing computational requirements for solving power grid problems, Bush pursued the idea of a large-scale mechanical general purpose analog computer. Six years later, in 1931, this machine, the first true general purpose mechanical analog computer, was completed[92] at the *Massachusetts Institute of Technology* (*MIT*). It was, in fact, a collection of individual computing elements like those described in the preceding sections. In addition to these components, special units like torque amplifiers, frontlash units[93] and *helical gearboxes* had to be developed. The latter allowed the interconnection of the various computing elements on a central interconnect unit that housed all input and output shafts, which were called *bus rods* (18 in total).

→ Figure 2.20 shows this early differential analyzer.[94] To set up the machine to solve a particular problem, its computing elements had to be connected using these bus rods, a lengthy process. Set up times of several days were not uncommon.[95] One eight hour day of operation was charged at US$ 400.[96] Much of the machine operational time was

devoted to work regarding the stability of long distance lines based on [→Carson 1926].[97] As a computer it proved to be highly influential and successful.[98] This machine and copies like the analyzer shown in →figure 2.21 inspired other researchers to build their own differential analyzers such as those:

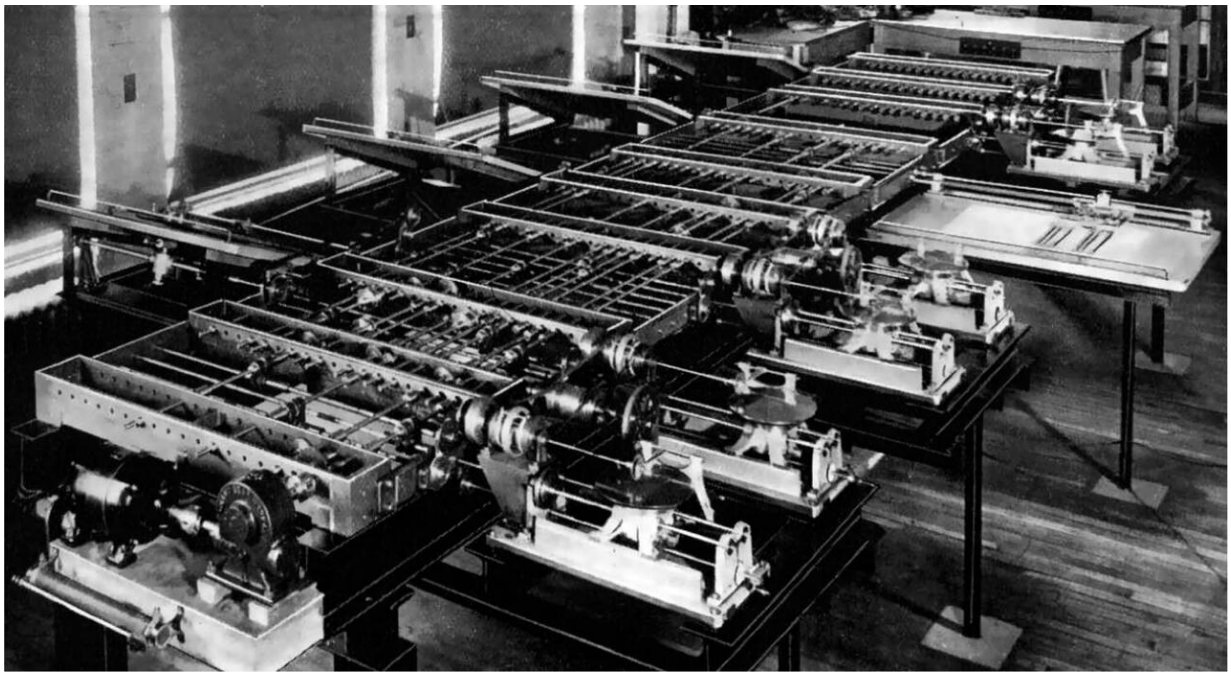


Figure 2.20. Vannevar Bush's differential analyzer (source: [→Meccano 1934/2, p. 443])



Figure 2.21. Another Bush differential analyzer installed at the Aberdeen Proving Ground, Maryland, with Vannevar Bush standing at the machine

- A Meccano based differential analyzer built by Douglas Hartree[99] and Arthur Porter[100] at Manchester University in 1934[101] (the system is shown in →figure 2.22),[102]

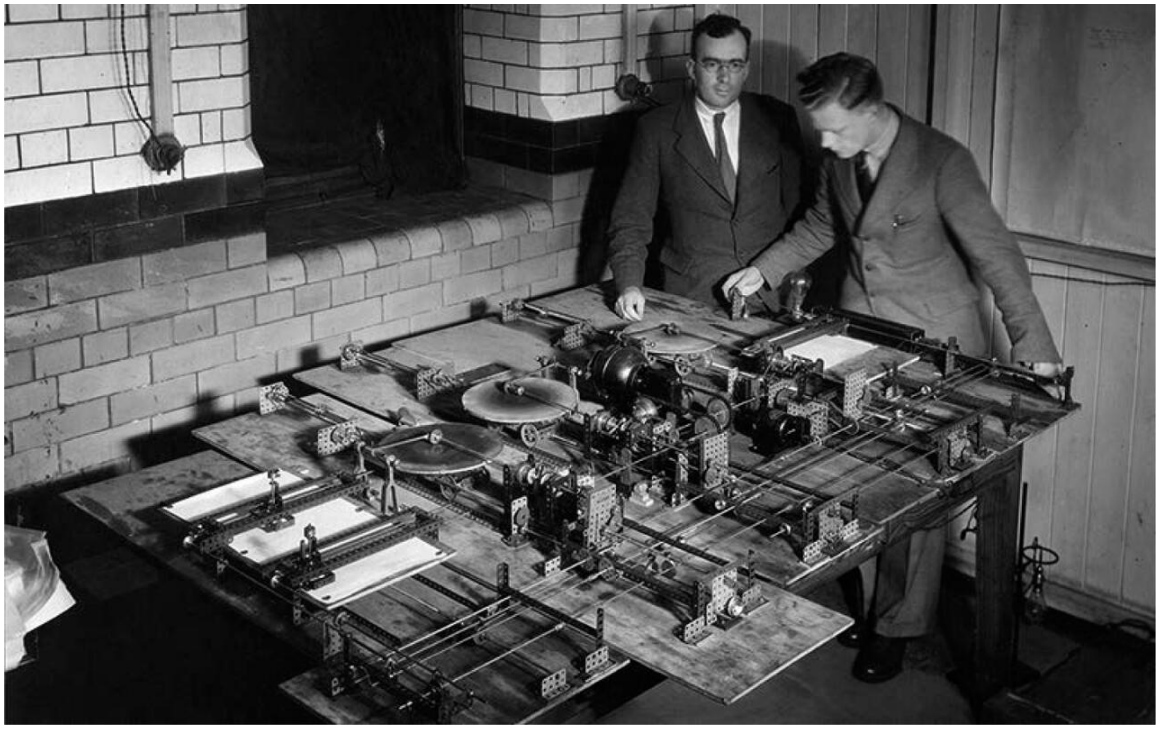


Figure 2.22. The Meccano differential analyzer built by Hartree and Porter

- a similar Meccano based machine built by J. B. Bratt at Cambridge University in 1935,[103]
- the *Oslo Analyzer*,[104] which was built from 1938 to 1942 at Oslo University's Institute of Theoretical Astrophysics under the auspices of Svein Rosseland,[105] who visited the MIT and knew about the MIT differential analyzer,[106]
- a differential analyzer with six integrators that was built and completed in 1938 in Russia,[107]
- the *Integrieranlage IPM-Ott* that was built in Germany[108] (start of development in 1938),
- the giant *Rockefeller*[109] *Differential Analyzer* devised by Vannevar Bush and installed at MIT's *Center for Analysis* in 1942, weighing about 100 tons, and being by far the most complex on this list of machines,[110]
- a "universal integrating machine" (*"Universal-Integriergerät"*) built by the German Askania-Werke in Berlin in the first half of the 1940s,[111]

- a differential analyzer built by General Electric at a cost of $125,000 between 1945 and 1947, which was in use at the University of California Los Angeles,[112]
- and finally a mechanical differential analyzer with only three integrators, which was built as late as 1959 at the *Institut für Angewandte Mathematik und Mechanik*, Friedrich-Schiller University Jena (Germany), under the leadership of Ernst Weinel.[113]

As an example a simple differential analyzer setup is shown in →figure 2.23. The differential analyzer is used to integrate over a function $y = f(x)$, which is provided as a plot on the *input table* shown on the upper left.[114] The integral over $f(x)$ is to be determined between the limits x_1 and x_2. The motor drives three shafts of the differential analyzer: Two shafts control the x-direction movements of the cross-hairs of the input and the pen of the *output table* (upper right) while the third shaft drives the integrator disk, whose rotation and thus time is the variable of integration.

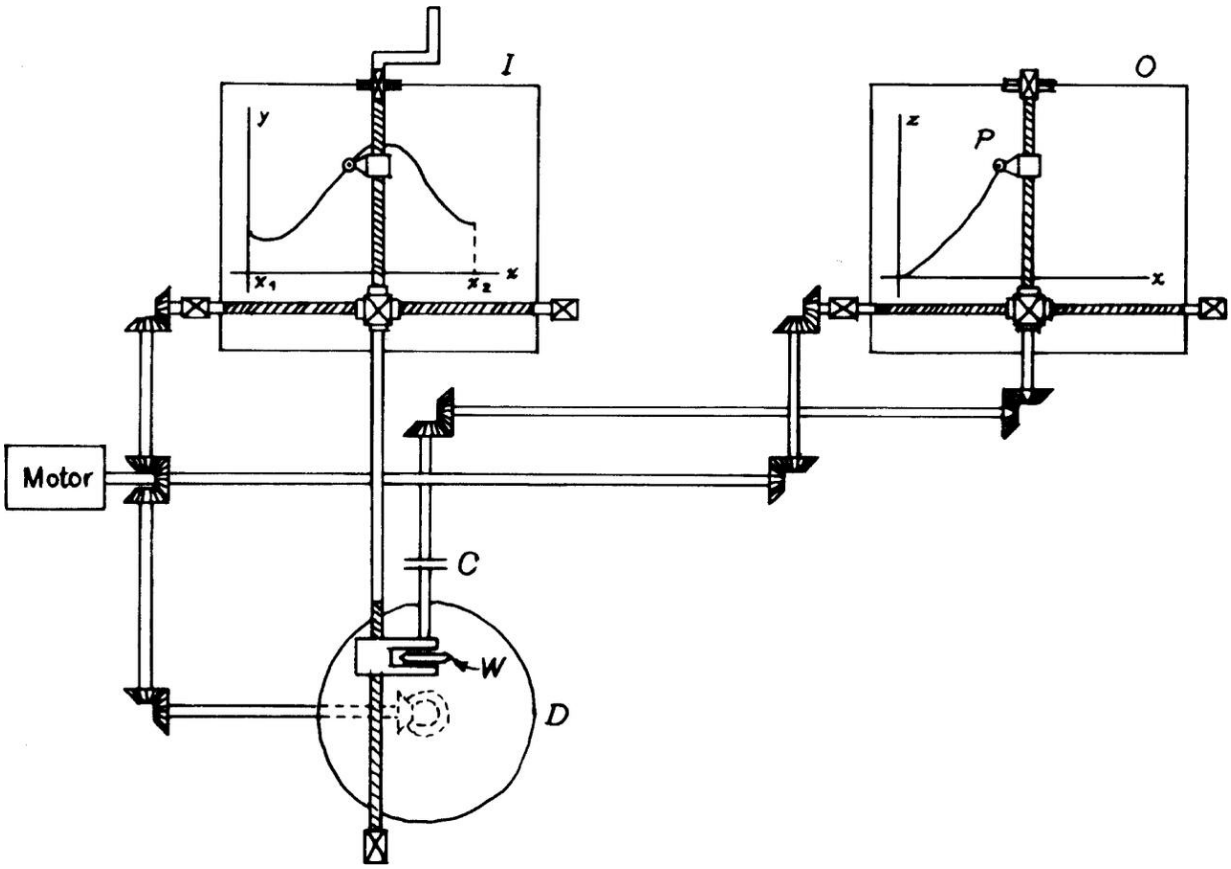


Figure 2.23. A simple differential analyzer setup for integration (cf. [→KARPLUS et al. 1958, p. 190], [→SOROKA 1962, p. 8-10])

In the simplest case a human operator would now track the curve on the input table by turning the crank accordingly to keep the curve in the cross-hairs of the input table's magnifier lens. This in turn changes the displacement of the friction-wheel of the integrator in a way that the angular displacement of *w* represents

$$x = \int_{x_1}^{x_2} f(x)dx.$$

The shaft coupled to the friction-wheel of the integrator finally drives the *y*-input of the output table, effectively plotting the desired integral of *f*(*x*).

Even today mechanical differential analyzers are still fascinating devices. →Figure 2.24 shows a modern implementation of such an

analog computer using Meccano parts, which was developed and built by Tim Robinson in the early 2000s.

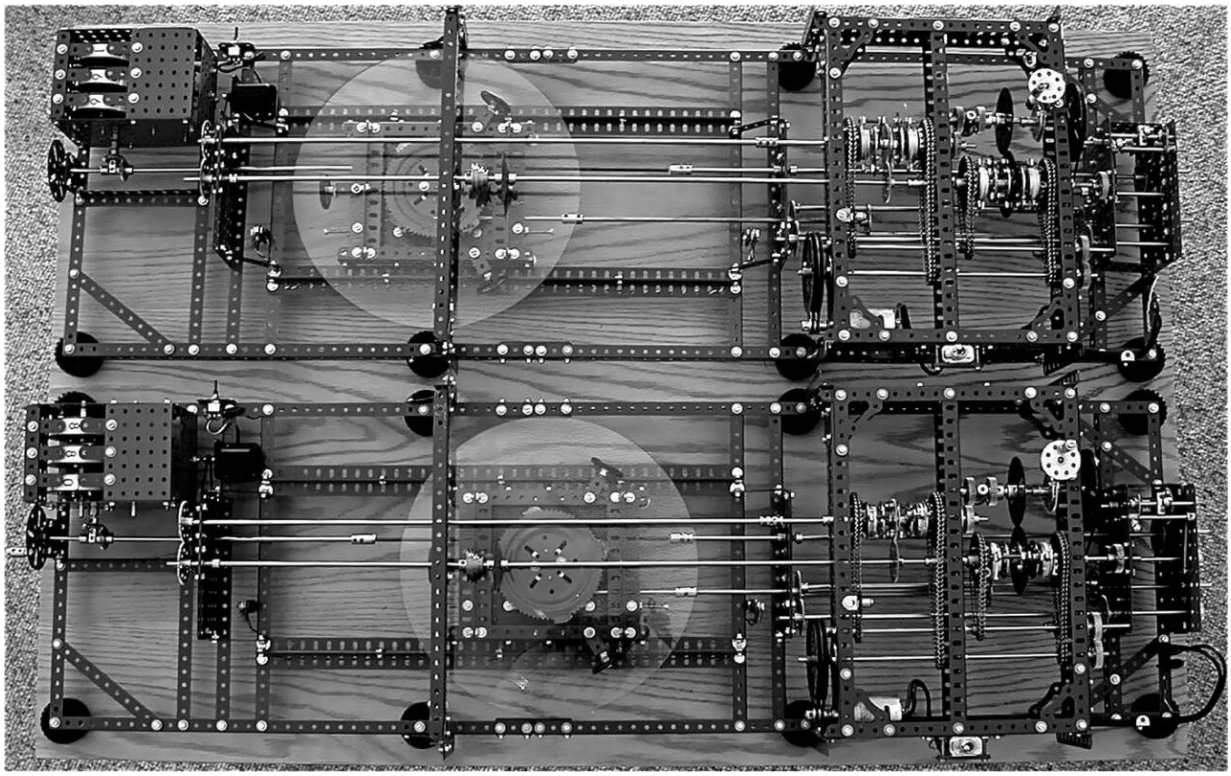


Figure 2.24. Tim Robinson's Meccano differential analyzer (reprinted with permission of Tim Robinson)

Apart from their slow overall operation due to the mechanical nature of the computing elements, the main problem of these differential analyzers were the excessive setup times due to the direct mechanical interconnections between the various computing units. → Figure 2.25 shows the schematic representation of Bush's differential analyzer configured for the Vallarta-Lemaitre cosmic ray problem,[115] which ran for a total of 30 days of computer time.[116] A setup like this could easily require several days to complete and check out – unacceptable for an agile environment and way too expensive for commercially viable applications of this technology.[117]

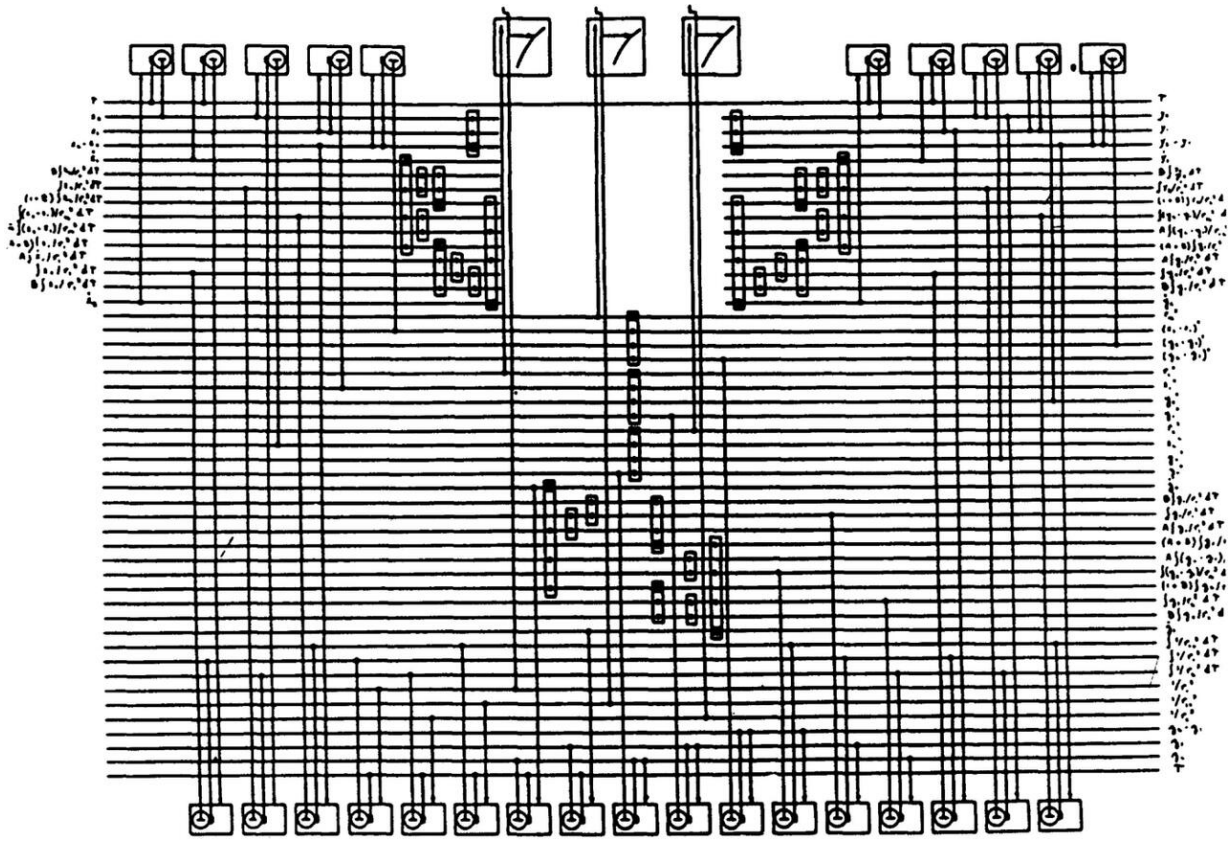


Figure 2.25. Setup of the BUSH differential analyzer for the VALLARTA-LEMAITRE cosmic ray problem (see [→ OWENS 1986, p. 77])

Accordingly, *electromechanical differential analyzers* were conceived, which used intricate rotational sensors and servo mechanisms to render the direct mechanical interconnects obsolete while still using mechanical computing elements. The first such machine was the *ROCKEFELLER differential analyzer*, which VANNEVAR BUSH had planned as a follow up to his first differential analyzer.[118]

Apart from more computing elements and increased precision, special emphasis was put on the capability to "*switch rapidly from one problem to another in the manner of the automatic telephone exchange*".[119] At its heart was a *crossbar switch* like those found in automatic telephone exchanges of that era. Instead of rearranging rotating rods or changing the wiring on a patch panel, the machine could be configured for a specific problem just by devising a "configuration bitstream" for the

crossbar as it would be called today. This configuration data was read into the machine by a paper tape reader.[120]

Of special importance were the rotational sensors required for transforming the rotational outputs of the various computing elements into corresponding voltages. These are described by [→ Berkeley et al. 1956, p. 116] as follows:

> "*[G]reat ingenuity and pains have been devoted to making it a precision mechanism. In particular, the take-off of wheel rotation is effected by an electrostatic* angle indicator. *This imposes virtually no load at all on the turning of the wheel […]*"

The signals generated by these take-offs were then fed into servo circuits that drove motors, which were assigned to the various input shafts of the individual units. The interconnection between these inputs and the servo outputs was accomplished by the aforementioned crossbar switch, which allowed problems to be set up quickly and thus eliminated the main drawback of its predecessor machines. In addition to this, the summers had remotely controllable gear boxes for every single input shaft.[121] Thus, even the change of coefficients could be performed from a central control panel. → Figure 2.26 shows this 100 ton behemoth differential analyzer, which has been aptly described as "*the most important computer in existence in the United States at the end of the war*"[122] due to its work on war related computational problems.[123]

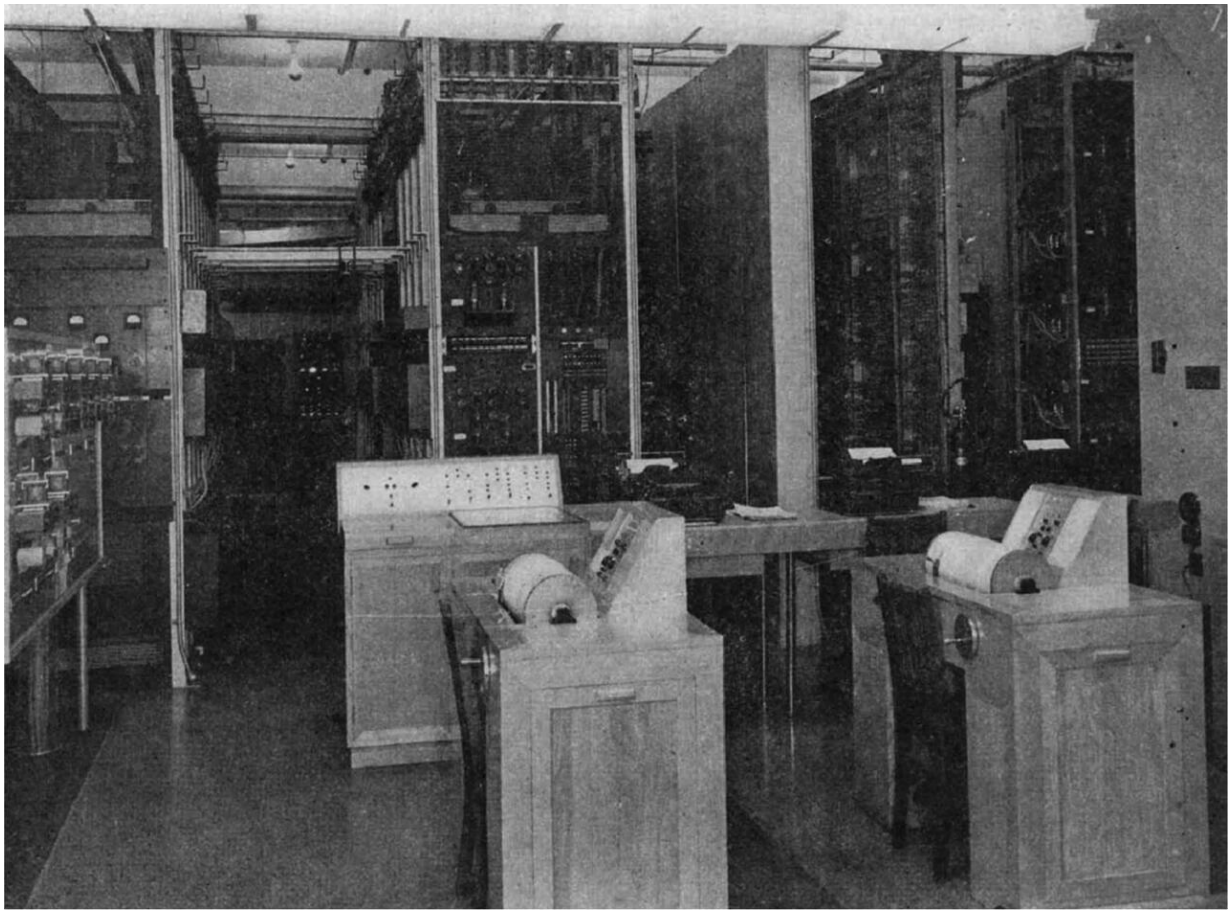


Figure 2.26. The ROCKEFELLER differential analyzer (see [→BUSH et al. 1945, p. 312])

In the following years a number of these electromechanical differential analyzers were built around the world. The *Minden* system is probably the last example of an electromechanical differential analyzer but was not as advanced as BUSH's last machine. In 1948 the German company *Schoppe & Faeser* started the development of a system that was named Minden after the German city in Westphalia where development took place. Of this system three machines were eventually built and sold. One of these machines, containing 20 integrators, was built on behalf of the British Admiralty and installed at the *National Physical Laboratory* (*NPL*). The last of these three systems contained twelve integrators, 20 summers, and ten function tables.[124] The computing elements had a precision of 0.01%. →Figure 2.27 shows a dual integrator unit of a Minden system, while →figure 2.28 gives an overview of a typical Minden

installation.[125] An interesting account of the various and many technical problems encountered with this machine can be found in [→Müller 1956]. A major challenge was the requirement for an inverse mode of operation of the integrators where the disk is driven by the friction wheel which was basically impossible due to the low forces exerted by the wheel.[126] Thus, an electronic servo loop was implemented which proved to be unstable for positions of the wheel too close to the axis of the disk. This resulted in considerable delays and quite some disappointment about the performance and reliability of the overall system.[127]

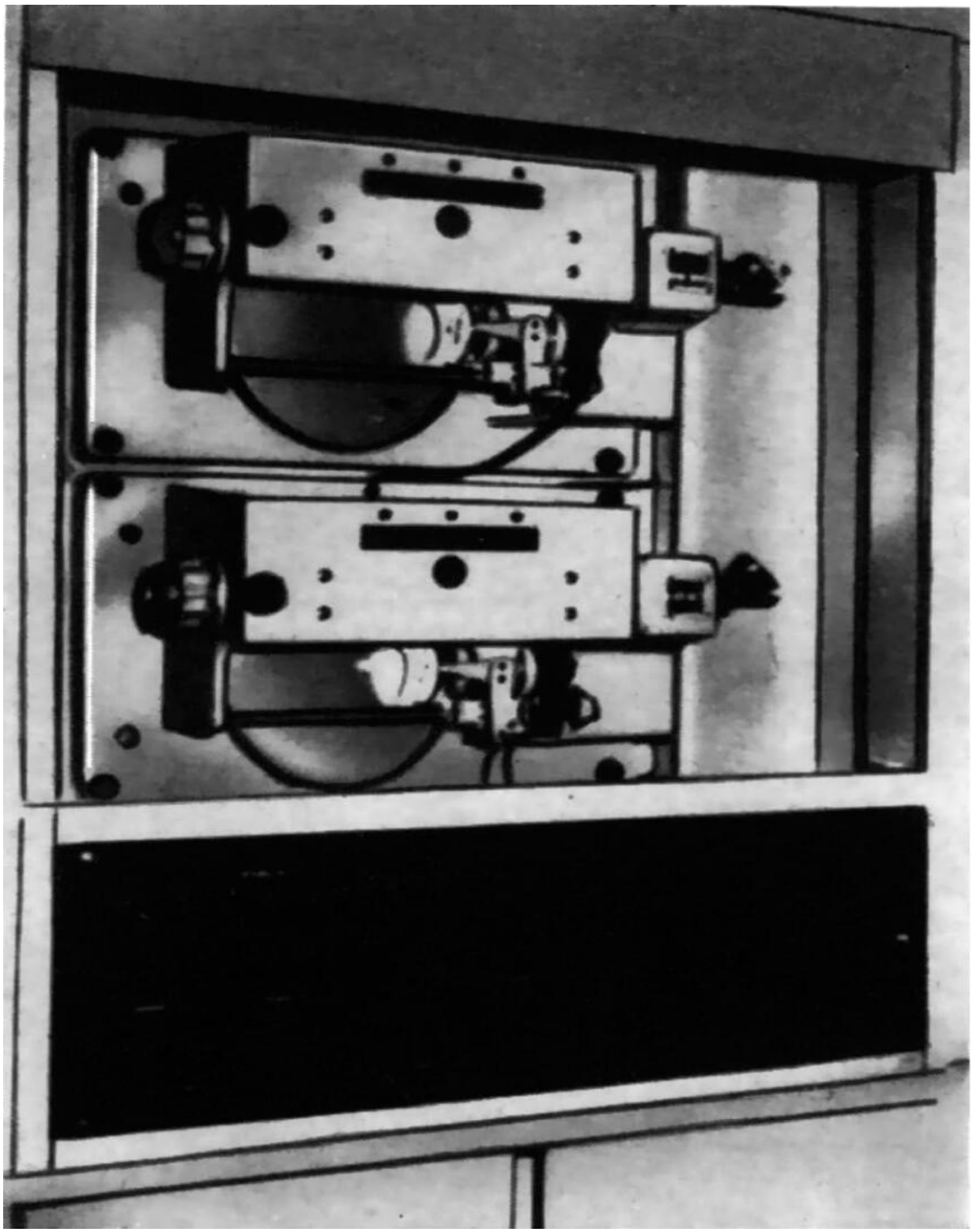


Figure 2.27. Dual integrator of the *Minden* system (see [→Rationalisierungskuratorium 1957, p. 51])

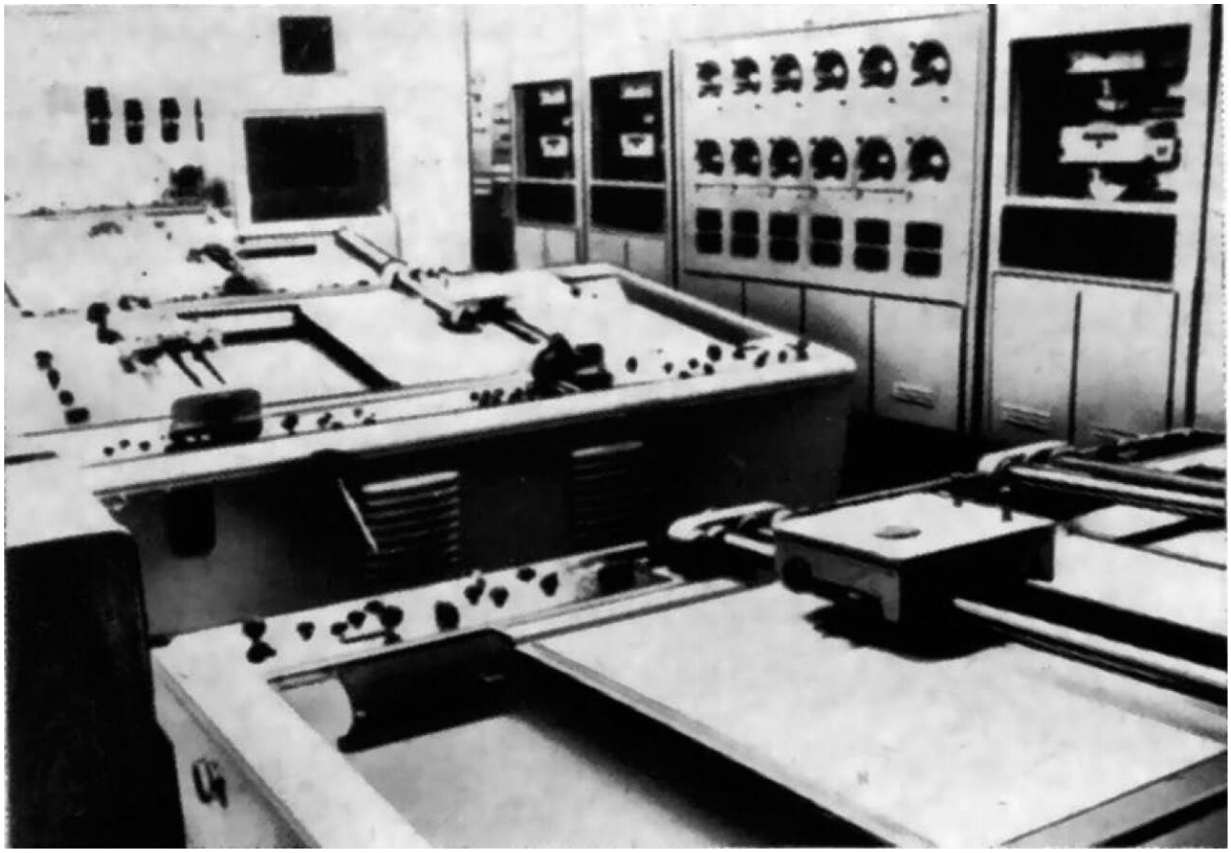


Figure 2.28. Differential analyzer *Minden* (see [→Rationalisierungskuratorium 1957, p. 51])

Another differential analyzer using synchro-motors and servo circuits for connecting the individual computing elements was the *Nordsieck*[128] *computer*, which is described in detail in [→Nordsieck] and [→Killeen 1953].[129]

Despite their advantages over purely mechanical differential analyzers, these electromechanical systems could not compete with the emerging analog electronic analog computers, which offered much greater speed, flexibility, and ease of construction and maintenance. One noteworthy attempt to build a novel electromechanical differential analyzer was undertaken by Hans F. Bückner,[130] who developed the *Integromat* in 1949.[131] This system is especially noteworthy as it uses mechanical computing elements working in a discrete fashion. →Figure 2.29 shows an integrator of this machines. Its basic idea is similar to that of a friction-wheel integrator but instead of continuous position values

and rotations it uses and generates discrete pulses as input and output. At its heart is a horizontal rotating shaft on which a number of slotted disks are mounted. Along this shaft a switch assembly can be moved in such a fashion that the switch is driven by one of these disks at any time. The first disk has one slot, causing a single pulse by means of the switch per revolution of the shaft, the second disk has two slots, etc. The output signals of the switch are *increments* instead of continuous angular positions. The switch position along the shaft corresponds to the position of a friction-wheel along the radius of a continuous integrator. As such this device can be seen as a direct forerunner of later DDAs.[132] Due to its discrete value representation this integrator was also known as *Stufenintegrator* (step integrator).

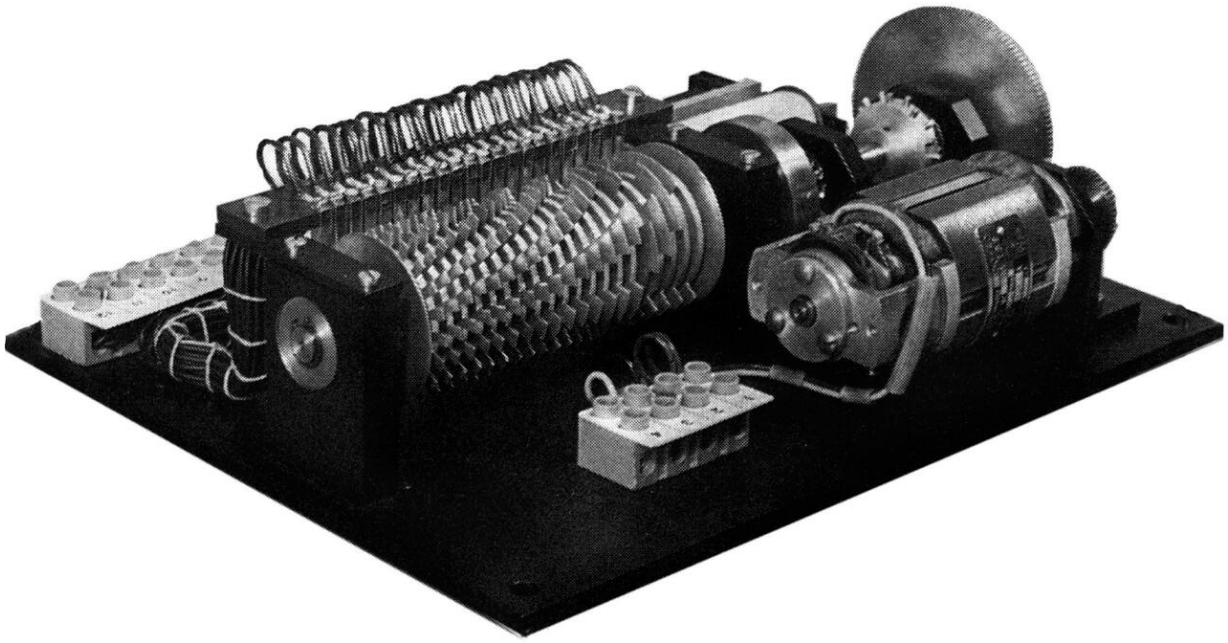


Figure 2.29. Integrator of the Integromat (see [→ ERISMANN 1962, p. 190])

The computing elements of the Integromat were connected by a central patch panel using discrete increment pulses instead of continuous voltages or currents. →Figure 2.30 shows the overall system containing 18 integrators and 12 function generators, built by Schoppe & Faeser. The rationale behind its construction was the simplicity of using only discrete increment pulses instead of transmitting angular positions and

velocities between extremely precise and costly continuously operating electromechanic computing elements. However, it was no commercial success due to two major shortcomings. First of all it was very slow, running at 10 Hz only. This was further worsened by the need to use very small step widths in order to minimize errors. Times of up to 45 minutes for the simulation of a single period of a damped oscillator were not uncommon.[133] In addition to this, its precision left much to be desired, even when running with small step sizes. Even a simple problem, the solution of $\ddot{y} = -y$, which is often called *circle test* when used as a test program since it yields a sine/cosine signal pair that can be used to draw a circle on a two-coordinate output device, showed an error of about 0.5% after only one period.[134] Experiments showed errors in the range of 4.5% after only five periods for typical damped oscillators under consideration – unacceptable for real applications.

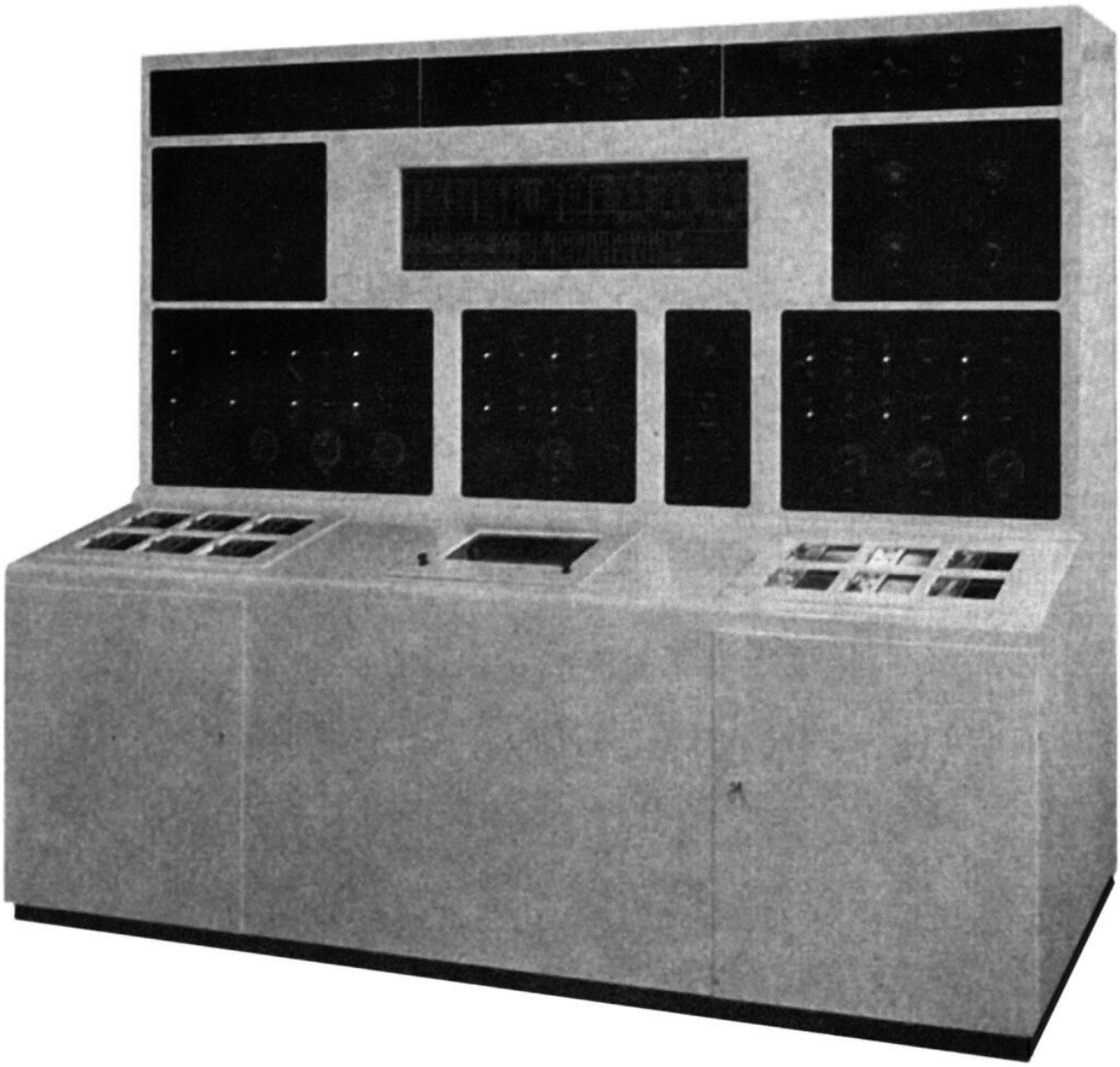


Figure 2.30. Integromat (see [→ ERISMANN 1962, p. 192])

Accordingly, the Integromat could not compete with the already existing electronic analog computers of its time and quickly fell into oblivion.

3 The first electronic analog computers

It is difficult to write about the “first” electronic analog computers because there is no such thing as the single first machine. In fact, the 1930s and 1940s were ripe for the idea of building a differential analyzer based on analog electronic components. Consequently, the following sections focus on five largely independent early developments, namely the works of Helmut Hoelzer, the early developments by George A. Philbrick, the birth of electronic fire control systems, MIT’s first electronic differential analyzer, and finally the Caltech analog computer. These distinct lines of development form the foundation for nearly all of the analog computer developments in subsequent decades.[1]

3.1 Helmut Hoelzer

In 1935 Helmut Hoelzer,[2] a student of electrical engineering at Darmstadt University in Germany, realized that mathematical operations such as integration or differentiation could be implemented quite easily using electronic circuits. Being an avid glider pilot, he knew that there was no device available in the aerospace industry to indicate the true ground speed of an aircraft and comprehended that by measuring the acceleration of an aircraft and integrating over the electric signal from the accelerometer a signal proportional to the velocity should be easily obtained.[3]

Unfortunately, his ideas were neglected for a rather long time – especially by mathematicians who were resistant to the

general idea of “computing” with arcane electronic devices instead of solving complex problems by pure thought. It was not until the development of the *A4 rocket*[4] that he could eventually explore and implement his ideas, which resulted in the *Mischgerät* – the world’s first fully electronic onboard computer for controlling a ballistic missile. Eventually, these developments led to Hoelzer’s development of a general purpose electronic analog computer, which was brought to the United States after World War II and used well into the 1950s during the development of rockets like the Redstone.

3.1.1 The “Mischgerät”

When World War II broke out, Hoelzer was working as an engineer for the renowned German company *Telefunken*,[5] developing radio transmission systems. One evening he was approached by Wernher von Braun[6] and two of his friends, Hermann Steuding and Ernst Steinhoff,[7] who led the department of *Orientierung, Steuerung und Bordinstrumentierung*[8] in *Peenemünde*, where the A4 rocket was under development.[9] They asked him to come to Peenemünde and help designing a radio guidance system for this new ballistic missile.

Hoelzer started working on the proposed radio guidance system shortly thereafter.[10] A particular problem that aroused his interest was the need to compute derivatives and integrals in flight – something completely unheard of before. Based on his ideas from his time at university he decided to use a capacitor as the central element for a circuit capable of differentiation and integration.

The current $i(t)$ charging a capacitor is generally described by

$$i(t) = C\dot{v}(t)$$

where C denotes the capacity and $v(t)$ is the voltage across the capacitor plates at time t.[11] This can also be written as

$$v(t) = \frac{1}{C} \int i(t)dt,$$

so both operations, differentiation and integration, can be implemented using a capacitor. A simple, yet in many cases sufficient approximation is implemented by a passive[12] RC circuit.[13] In the case of differentiation the current flowing through the capacitor can be measured as voltage drop across a sufficiently small resistor, in the case of integration the voltage across the capacitor's plates can be used directly as an approximation for the time integral of some varying input signal.

As simple as this idea looks from today's perspective, its realization was far from easy in the 1940s due to the fact that it requires DC coupled amplifiers because the signals to be amplified and processed further are mainly DC voltages. An ideal DC amplifier would yield an output voltage of 0 Volts if its input voltage is 0 Volts, regardless of its gain, which should be as high as possible. Due to unavoidable imbalances, aging of components, temperature effects, etc., this is not true for a real-world DC coupled amplifier in which the individual amplifier stages are coupled directly. Accordingly, a small error voltage occurring in an early amplifier stage will be propagated and amplified by the following stages, resulting in a non-zero output signal even for an input signal of 0 V.

This precludes the application of such an amplifier for an integrator circuit, since the capacitor would integrate this offset signal, which would not only invalidate the computation but eventually saturate the integrator.

To solve this inherent problem Hoelzer decided to use an AC coupled amplifier instead. This idea solved the drift problem as AC coupled amplifiers employ amplification stages which are not

coupled galvanically but by means of capacitors, which effectively block any DC (drift) voltage. Although this characteristic of AC coupled amplifiers is very desirable, it also makes this kind of amplifier unsuitable for the use with DC (or very low frequency) voltage signals as they occur in the control circuitry of a missile.

So Helmut Hoelzer faced the problem of converting an input DC voltage to an AC voltage suitable for feeding an AC coupled amplifier and rectifying the output voltage again to get the desired DC signal. Transforming DC signals into AC voltage was done by means of a *ring modulator*,[14] whose structure is shown in →figure 3.1. This circuit was described first in the early 1930s and was used widely in radio applications.[15] Remarkably, most of the earliest ring modulators used a very early form of solid state diodes, which were called *Sirutor*[16] in Germany. These devices were developed in 1934 and used a small stack of copper(I) oxide (Cu_2O) pills within a little tube with electrodes attached to its ends. →Figure 3.2 shows a typical Sirutor like those employed by Helmut Hoelzer for his ring modulators.

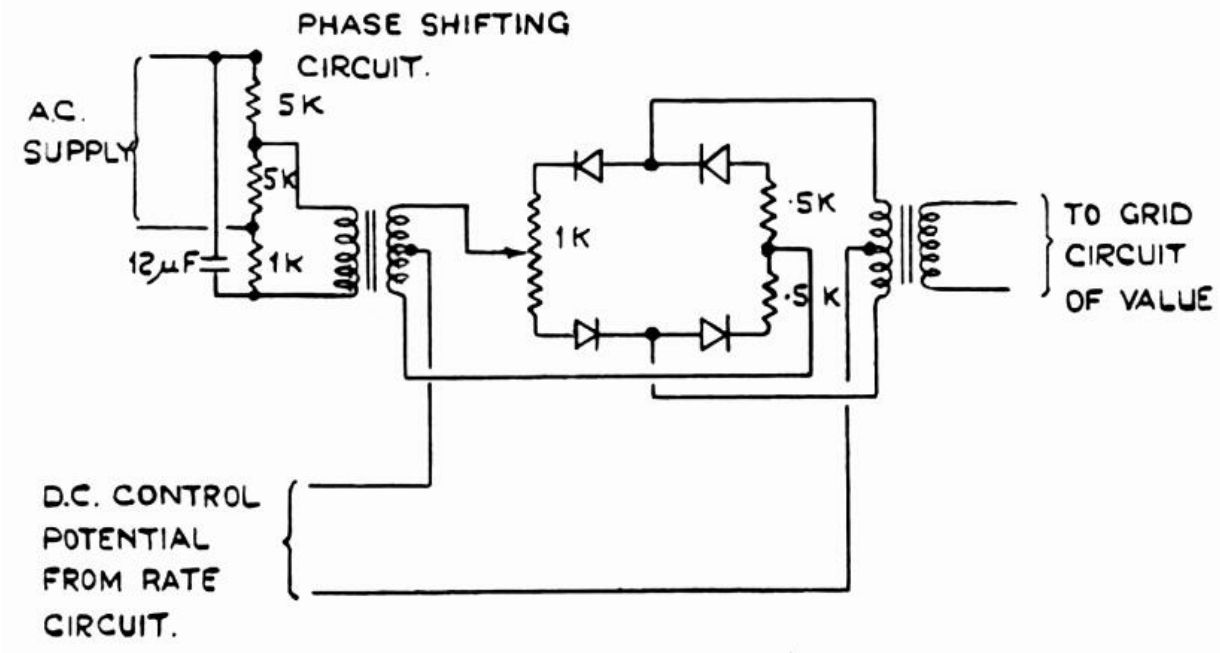


Figure 3.1. Ring modulator circuit used in the Mischgerät (see [→ Hudson et al. 1945, p. 13])



Figure 3.2. Sirutor (scale in centimeters)

In 1940 the first usable radio guidance system for the A4 rocket was devised[17] and Hoelzer turned his interest to the development of a gyro based stabilization and guidance system. Like earlier developments by Robert Goddard[18] the A4 used movable vanes in the rocket's engine exhaust, *exhaust rudders*, as well as *air rudders* to control the flight path of the rocket. The exhaust rudders were the most effective means of control but could only work during the powered ascent phase. After re-entry into earth's atmosphere the air rudders became effective.[19]

This system turned out to be difficult to control due to the sheer size of the A4 and the resulting moment of inertia. The control system had to take this into account to avoid wild

oscillations, which would result in the destruction of the rocket during flight. →Figure 3.3 illustrates this problem: The upper half shows the buildup of oscillations in the case of a simple control system unable to take the moment of inertia into account.[20] The lower half shows the desired behavior of the controller – the oscillation is quickly damped out, which requires more than just a linear controller, it requires the ability to take a derivative term into account.

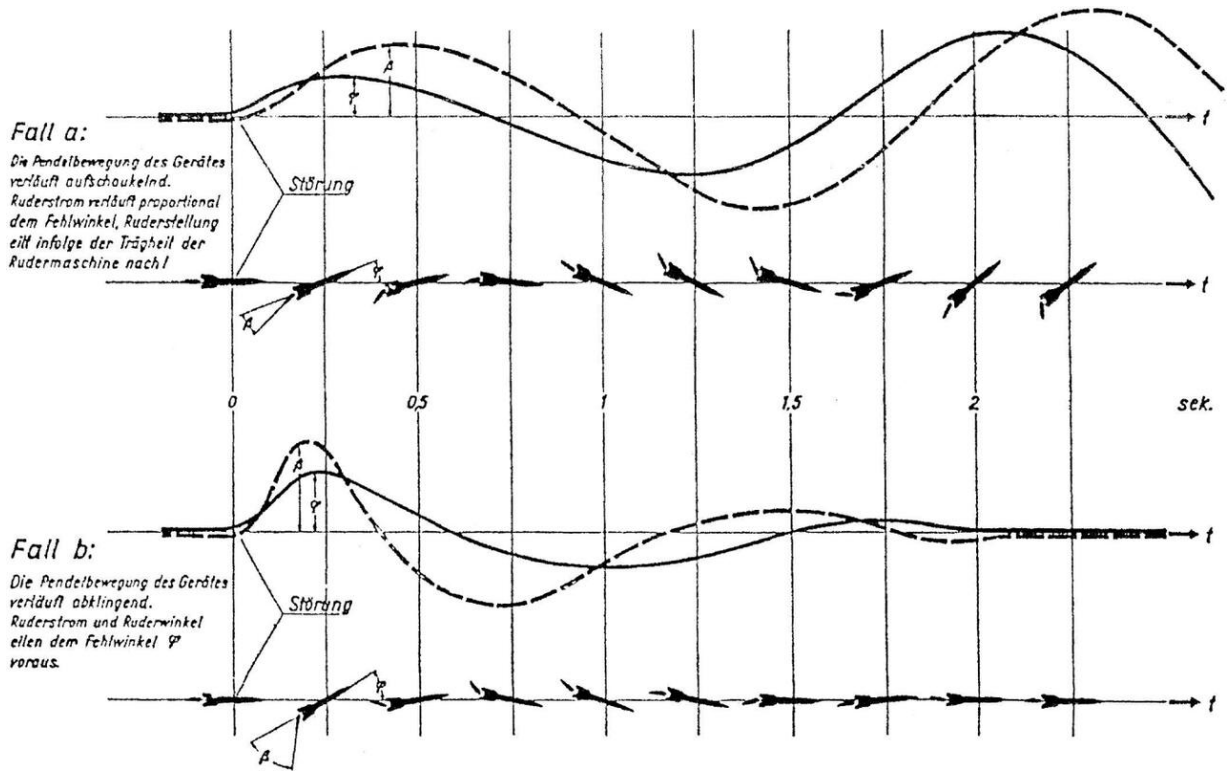


Figure 3.3. Control problem of the A4-rocket ([→N. N. 1945, fig. 82])

At first, the company *Kreiselgeräte GmbH* attempted to develop a gyro based controller, which showed the undesirable behavior shown in the top half of →figure 3.3. This first implementation did not take any derivatives of the signals generated by the gyro

platform into account. Another approach, using bank-and-turn indicators in addition to the main gyro, turned out to be too complex and too costly. After several unsuccessful attempts to build a working control system for the A4, Von Braun approached Helmut Hoelzer as he was aware of the fact that Hoelzer's radio guidance system also needed derivatives and time integrals of sensor output voltages, which were generated using RC combinations instead of costly additional instruments. This turned out to be a viable approach – while the initially proposed bank-and-turn indicators initially cost about US$ 7 000, the simple circuit developed by Hoelzer only cost about US$ 2.50[21] and worked very well even under the severe stresses encountered during the flight of the rocket.

The idea of employing a differentiating circuit to derive rate of change signals from sources such as a gyro system was completely novel at that time and industry representatives were sceptical to say the least. Hoelzer encountered substantial hostility from the companies involved in the development work for the A4. Only the obvious success of his control scheme eventually convinced the management. The resulting control system was dubbed *Mischgerät*, which can be translated as *mixing unit*, a camouflage term invented to disguise the real purpose of this first fully electronic onboard computer.[22]

→Figure 3.4 shows the main parts of the A4 guidance and control system. At the top of the rocket, directly below the war head, is a compartment containing the gyro system, an optional radio receiver/transmitter, and the all important Mischgerät, which controls the servo motors that operate the exhaust and air rudders at the lower end of the rocket.

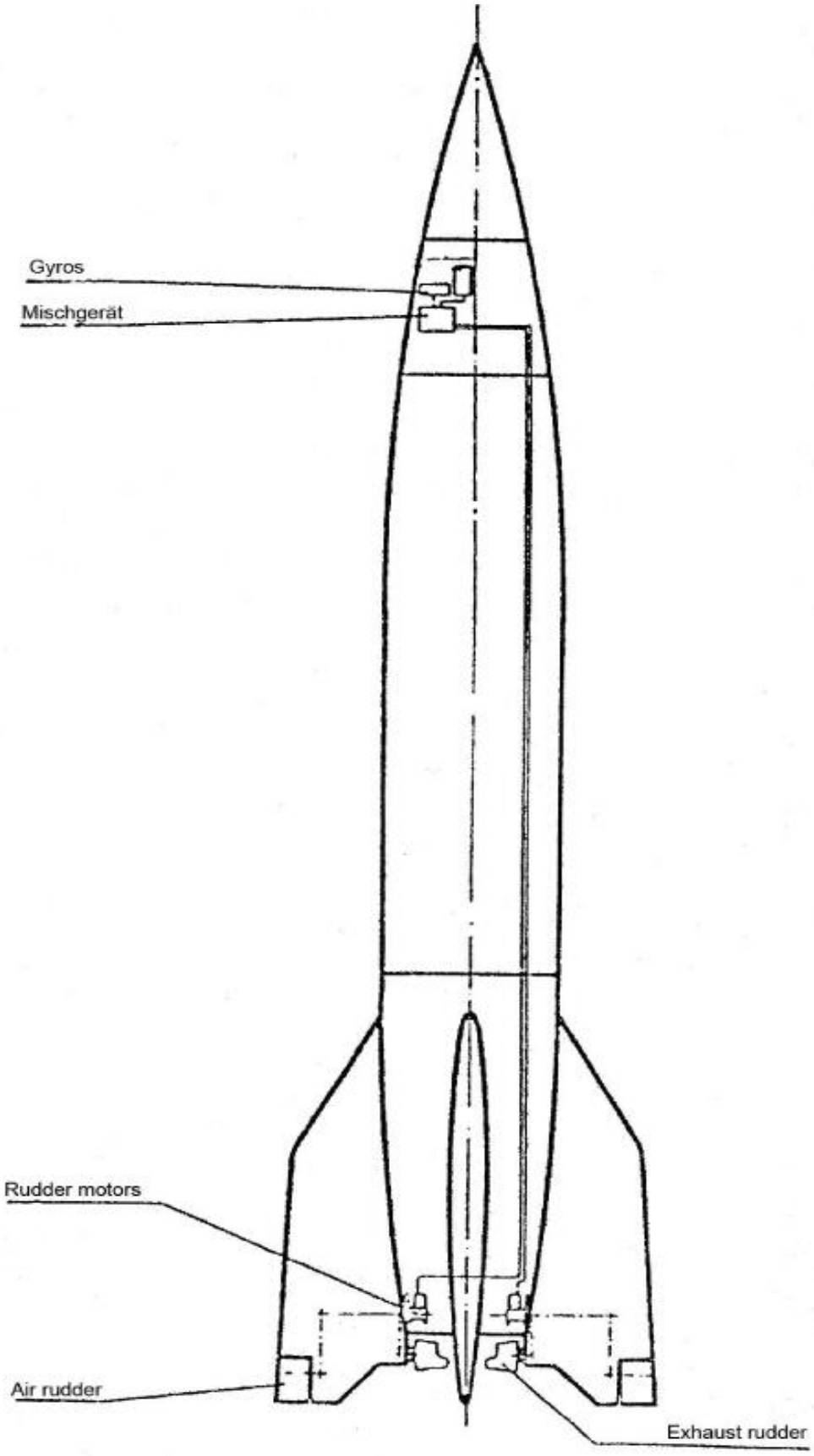

Gyros
Mischgerät
Rudder motors
Air rudder
Exhaust rudder

Figure 3.4. Main elements of the A4 control system ([→N. N. 1945, fig. 83])

A more detailed schematic of the control system is shown in →figure 3.6.[23] Shown on top of the picture is the main bus of the rocket supplying all subsystems with 27 Volts DC. Using two motor-generator units, denoted by *Umf. I* and *Umf. II*,[24] this supply voltage is converted into the voltages required for the gyro motors and the Mischgerät. The two gyros, *Richtgeber D* and *Richtgeber EA* are connected to the Mischgerät, which is shown in the middle of the figure. Using RC combinations denoted by *RC Glied* the input signals are differentiated, which effectively implements a PD controller.[25] The resulting output signals are used to drive the servo motors controlling the rudders. →Figure 3.5 shows the structure of the basic RC combination used in the Mischgerät.

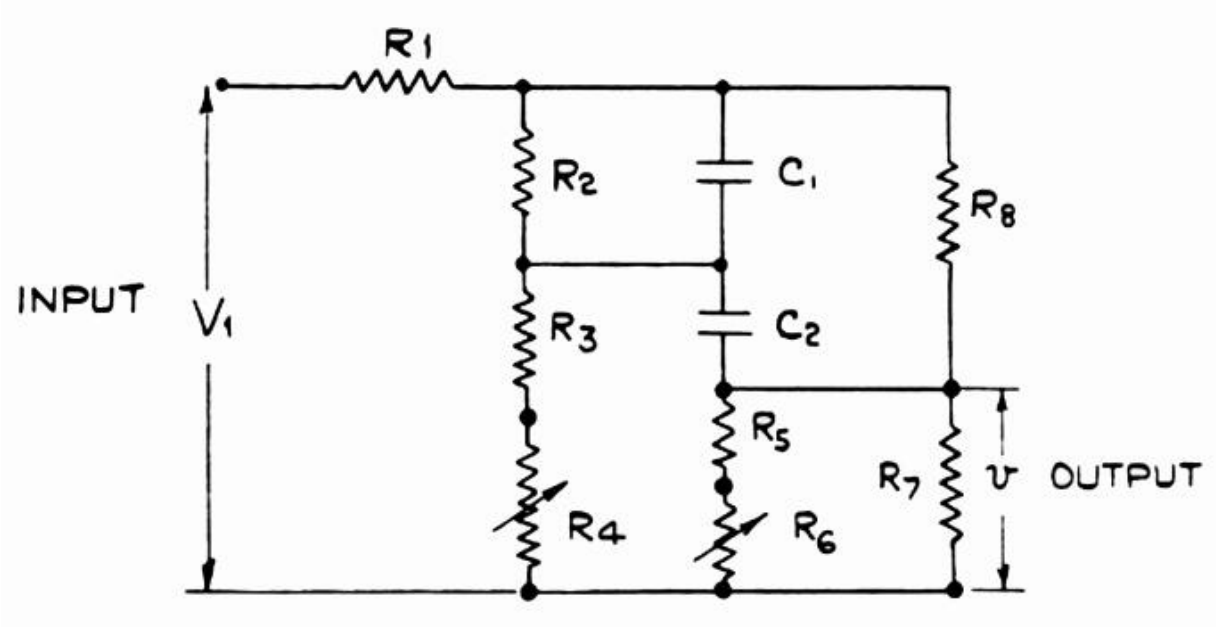


Figure 3.5. Structure of the basic RC combination used in the Mischgerät (see [→ Hudson et al. 1945, p. 13])

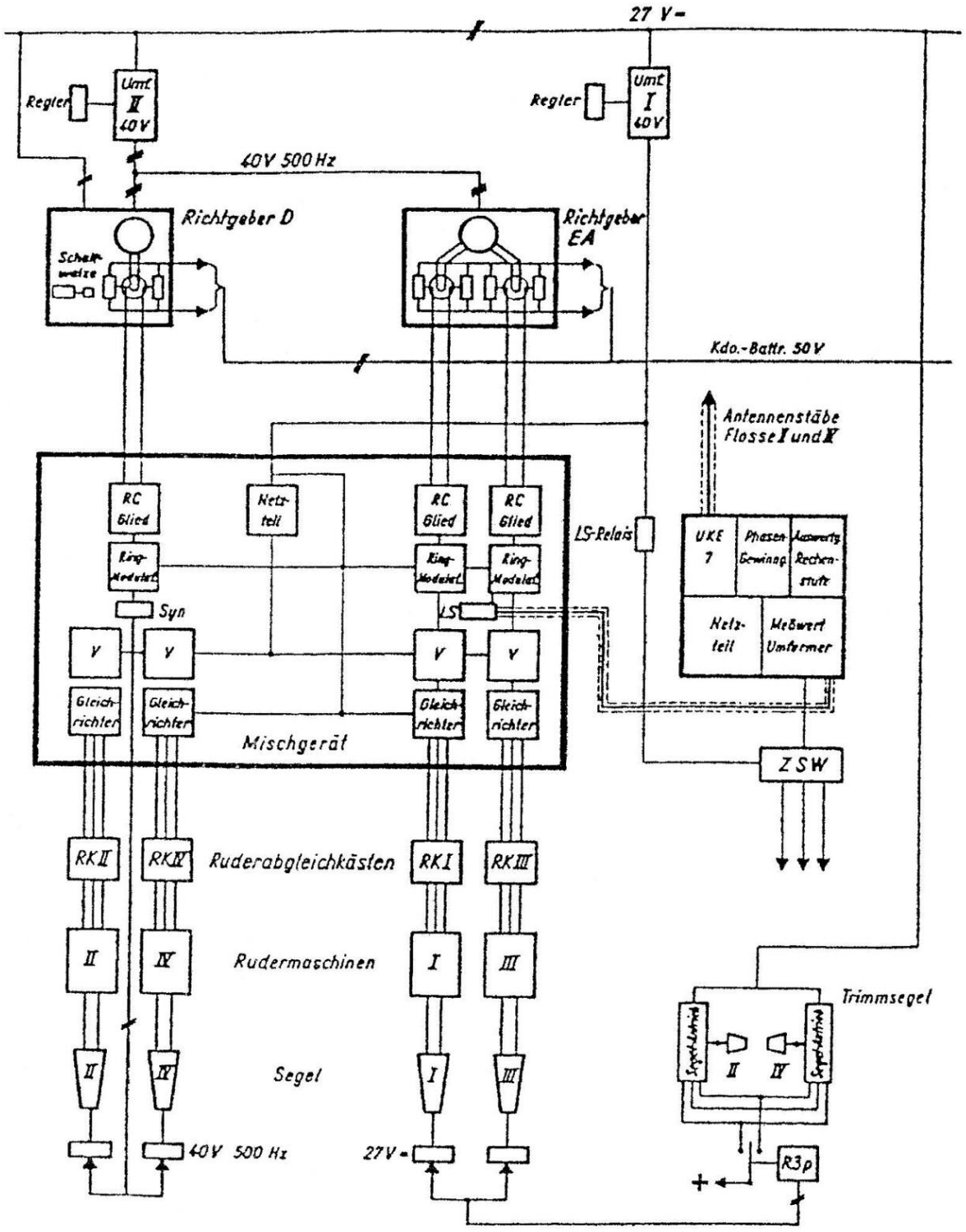


Figure 3.6. Simplified schematic of the A4 control system (see [→N. N. 1945, fig. 80])

→Figure 3.7 is a photograph of the control system's compartment in an A4 rocket. The Mischgerät itself is denoted by *MG, BB* and *ZB* are the main battery and a supplemental battery, and *U II* is the second motor-generator unit powering the gyros, which are installed in one of the other quadrants of this compartment on top of the rocket.

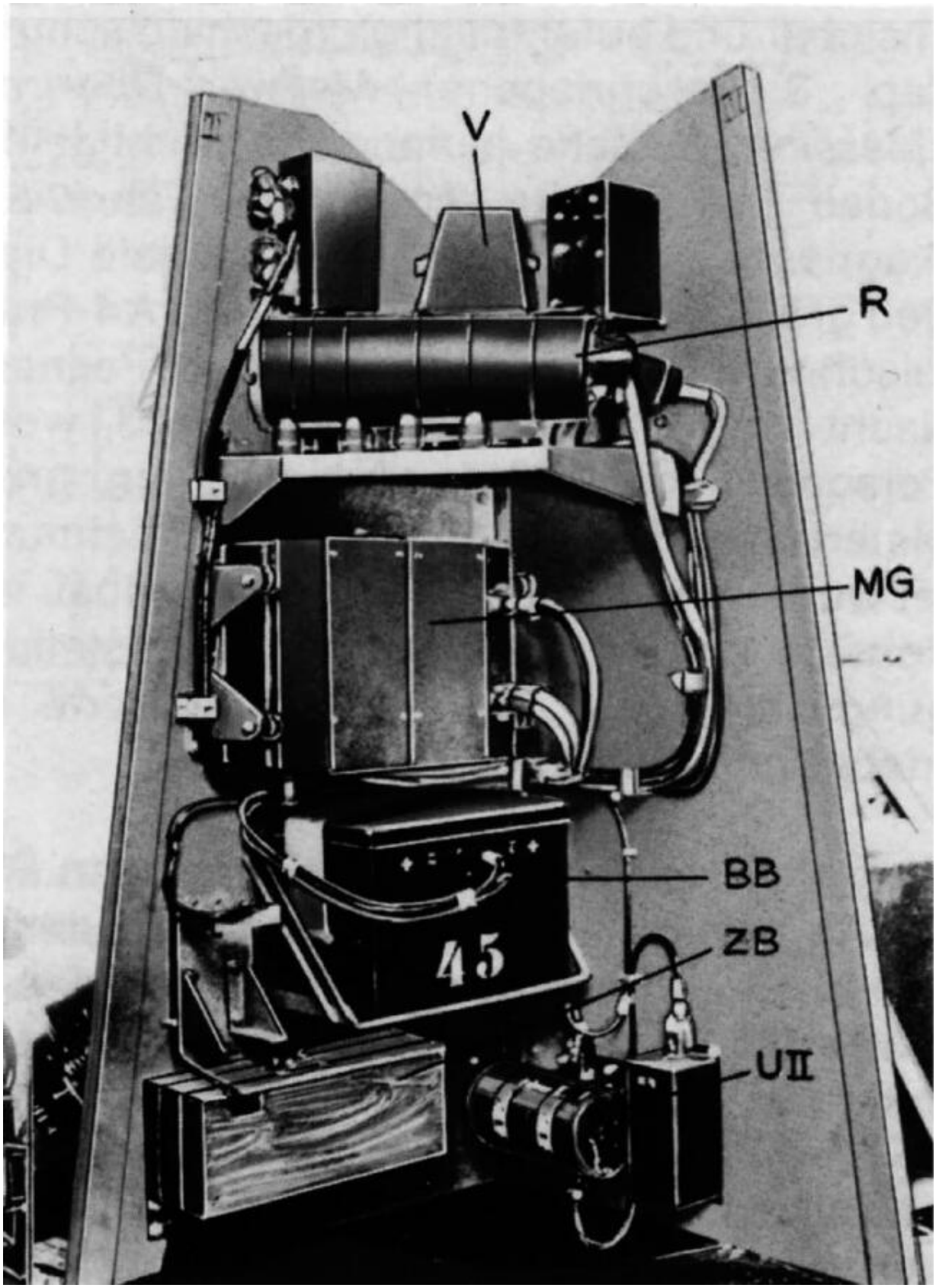

V
R
MG
BB
ZB
UII
45

Figure 3.7. The installed Mischgerät ([→Trenkle 1982, p. 134])

The Mischgerät itself is shown in →figure 3.8 – its small size and rugged design are as obvious as they are remarkable. It is built on a base plate holding five plug-in modules, two at the bottom and three at the top. The empty space on the bottom right contains the connectors to the various rocket subsystems.



Figure 3.8. The Mischgerät (photo: Adri de Keijzer, reprinted with permission)

This device was described by [→ Tomayko 2000, p. 15] as the "*first fully electronic active control system*" and it had great influence on further developments as this quotation from [→ Bilstein 2003, p. 243] shows:

> "*Further work by other Peenemünde veterans and an analog guidance computer devised with American researchers at the Redstone Arsenal culminated in the ST-80, the stabilized platform, inertial guidance system installed in the Army's 1954 Redstone missile […] The ST-80 of the Redstone evolved into Jupiter's ST-90 (1957) […]*"

The *control computer* of the *Redstone* missile was based on magnetic amplifiers and RC networks. It received an attitude error signal from the gyro platform *LEV-3* and filtered these to eliminate the effects of bending moments of the rocket structure. This was necessary as tests showed an interaction of the second bending mode with the yaw and pitch axis control when running without this signal filtering stage. The resulting signals were then differentiated yielding angular velocity signals, etc., which were then amplified in order to steer the exhaust vanes of the rocket.[26] Thus, the Redstone control computer resembled the Mischgerät quite closely. Even the much later *Saturn V* rocket of Apollo fame used an analog *flight control computer* that took commands from the *LVDC* (*Lunar Vehicle Digital Computer*) as well as signals from various other sources and generated the required control signals for the gimbaled rocket engines of the three Saturn V stages.[27]

3.1.2 Hoelzer's analog computer

In parallel with his work on the Mischgerät and initially unnoticed by his supervisor, Helmut Hoelzer started the development of a fully electronic motion simulator analog computer in 1941.[28] This machine was also based on his idea of

using RC combinations to perform integration and differentiation. He was confronted again with many objections and the prejudice by other people. Hermann Steuding once commented about Hoelzer's ideas:[29]

> "*Young man, when I compute something, the results will be correct and I do not need a machine to verify it. By the way, machines cannot do this*."

He was even forced by his supervisor to stop fiddling around with this electronic contraption and concentrate on his real work, as he remembers vividly:[30]

> "*My boss came to the laboratory where he saw the electronic contraption. He said: 'Hoelzer, stop playing with this toy and do your work!' I gave the only answer possible under these circumstances: 'Yes, sir!' Next morning, nothing was left in the lab – the computer had been moved to a small, windowless room. […] Nevertheless, it seemed important to me to develop a system that could aid the development of rocket control systems significantly. […] When it finally worked, everything was forgiven*."[31]

Further development on the analog computer continued in secret. After its completion it not only proved to be a valuable tool, but moreover it played a central role in the A4 development process. Without this analog computer, many problems, especially in the area of guidance and control of a ballistic missile like the A4, could not have been solved, at least not in time. In a short span of time many departments in Peenemünde used this machine actively to solve a variety of problems – even the word *analog* used to denote the setup of an analogon became part of engineering terminology.[32] The influence of this computer cannot be overestimated as [→ Neufeld 2007, p. 133] makes clear. He describes this computer as "*a fundamental innovation that really made a mass-produced guidance system possible*".

Since this analog computer was intended to be a research instrument it had to be more precise than the simple RC combinations used in the Mischgerät for integration and differentiation. HOELZER therefore used a controlled current source to charge the capacitor in the case of integration while a controlled voltage source was necessary for the differentiator. Both sources were implemented using AC coupled amplifiers like those being used in the Mischgerät.[33]

→ Figure 3.9 shows the basic schematic of the differentiator developed by HOELZER for his analog computer.[34] The input values for this computing element are AC voltages that are coupled inductively into the computing element and are amplified by a triode based input stage. A synchronous demodulator in the anode circuit of this tube rectifies the amplified input AC signal. The resulting DC signal is then used to charge the capacitor *C* while the charging current, representing the derivative to be determined, causes a small voltage across resistor *r*. This voltage is then converted back into an AC voltage by a ring modulator, the output of which drives the grid of the output amplifier stage that in turn feeds several output transformers connected in series.

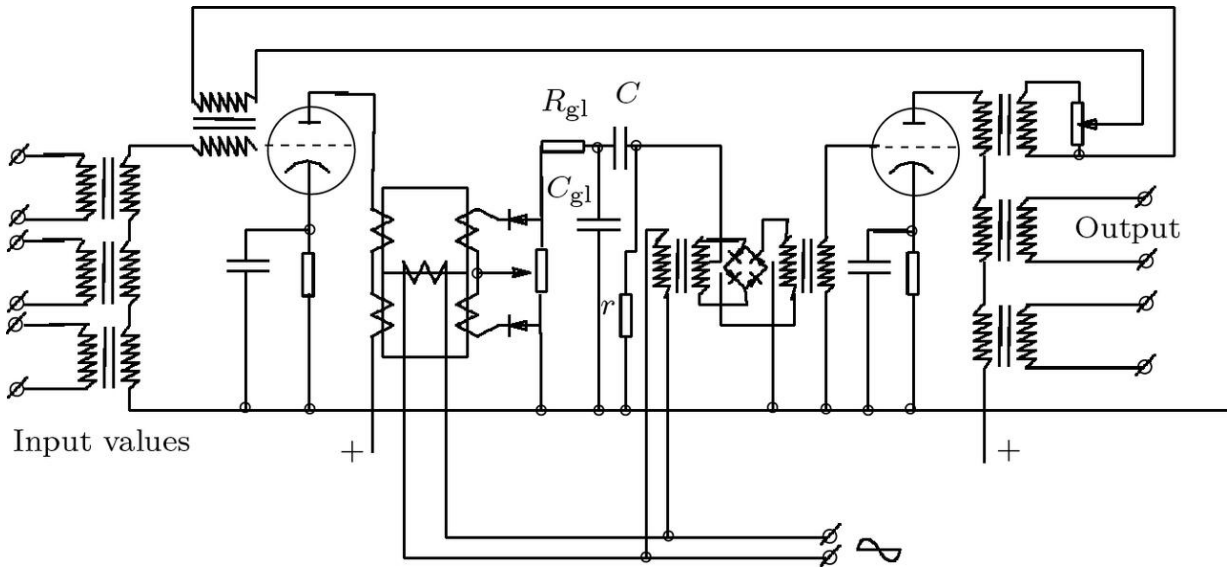


Figure 3.9. Schematic of HOELZER's differentiator circuit (cf. [→HOELZER 1946/1, fig. I, 13])

Apart from this demodulation and modulation scheme there are three noteworthy aspects in this circuit:

1. One of the outputs is used to generate a feedback signal, the amplitude of which can be set with the potentiometer shown in the upper right corner of →figure 3.9.
2. The modulator and demodulator are both driven by a common carrier signal.
3. Finally, the differentiator employs a simple low pass filter consisting of R_{gl} and C_{gl}[35] in order to suppress excessive noise.[36]

The integrator circuit differs from the differentiator circuit only in a few respects: The low pass filter consisting of R_{gl} and C_{gl} can be omitted since the operation of integration itself acts as a low pass filter. In addition to that, the output stage is fed directly with the modulated voltage across the capacitor plates, so the resistor r is also no longer required. →Figure 3.10 shows the

basic schematic of the integrator used in HOELZER's analog computer.

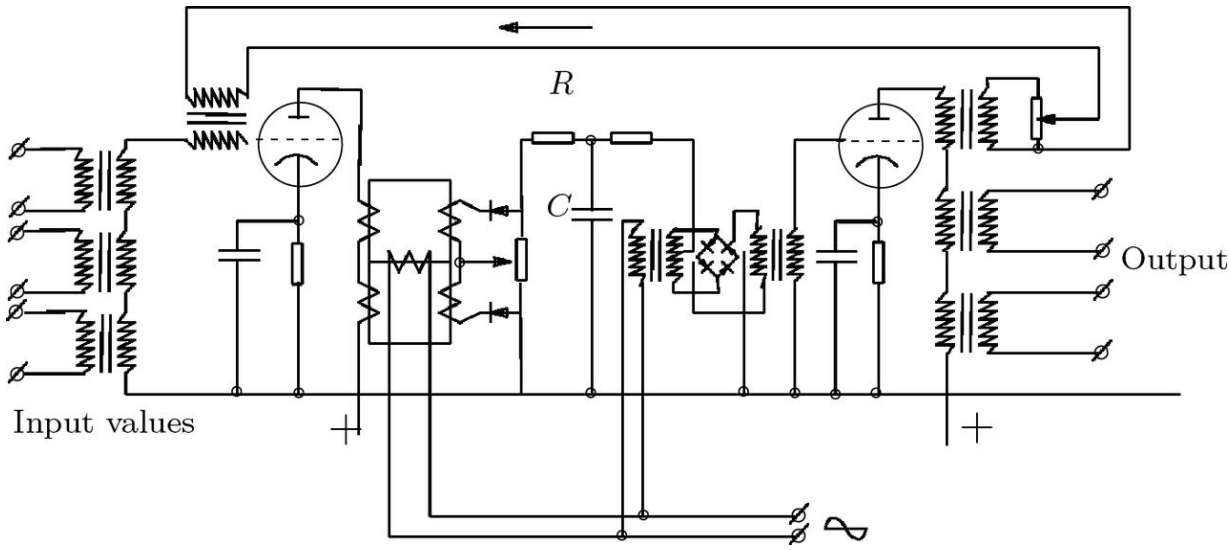


Figure 3.10. Schematic of HOELZER's integrator circuit (cf. [→HOELZER 1946/1, fig. I, 11])

At Peenemünde other systems, such as the *Wasserfall* rocket, an anti-aircraft surface-to-air rocket, were developed.[37] Although this weapon system was not completed in time to see any action, its development was rather advanced, even an electronic simulator using integrator circuits as shown in →figure 3.11 existed. The similarity of this circuit to those developed by HOELZER is obvious. The AC input signal is first amplified by a pentode then rectified using a synchronous rectifier, the output of which feeds the actual integration RC combination. The voltage across the capacitor is then fed into a modulator feeding an output amplifier stage using a second pentode. The main difference between →figure 3.10 and 3.11 is how input and feedback signals are connected. While HOELZER's circuits use

coupling transformers, this circuit couples signals directly to the input stage.

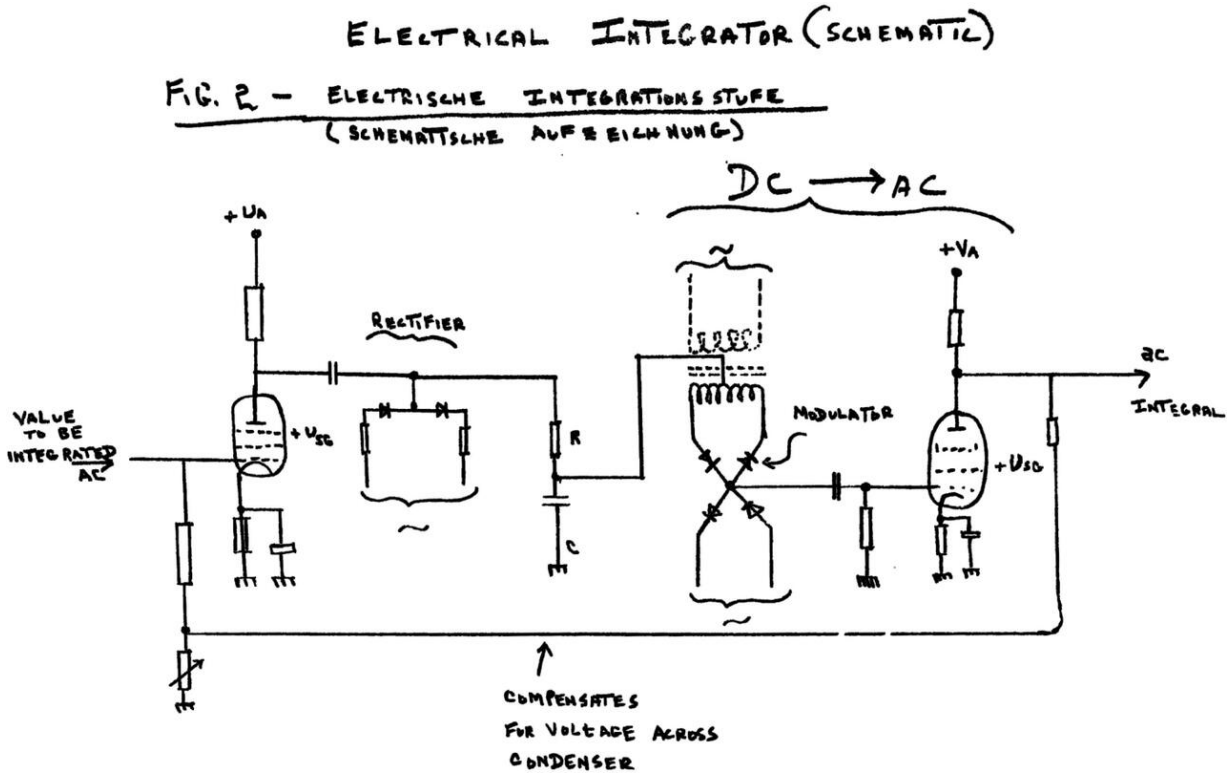


Figure 3.11. Integrator used in the *Wasserfall* simulator (see [→Peenemünde, p. 280])

The analog computer also contained additional computing elements such as multipliers, dividers, and square root function generators.[38]

Especially interesting is a servo circuit suggested by HOELZER to implement division and other operations, since this device is a direct forerunner of the later servo function generators and multipliers described in →sections 4.5.1 and →4.6.1. →Figure 3.12 shows the basic structure of this computing device. It essentially forms a bridge circuit with a servo motor driving two potentiometers R_1 and R_{out} as its central element. R_1, the fixed

resistor R_2, and the two input transformers form the actual bridge circuit. Every imbalance of this bridge will yield an error voltage that is amplified by the Amplifier *A*, which in turn drives the servo motor canceling out the imbalance by readjusting R_1 according to the current input.

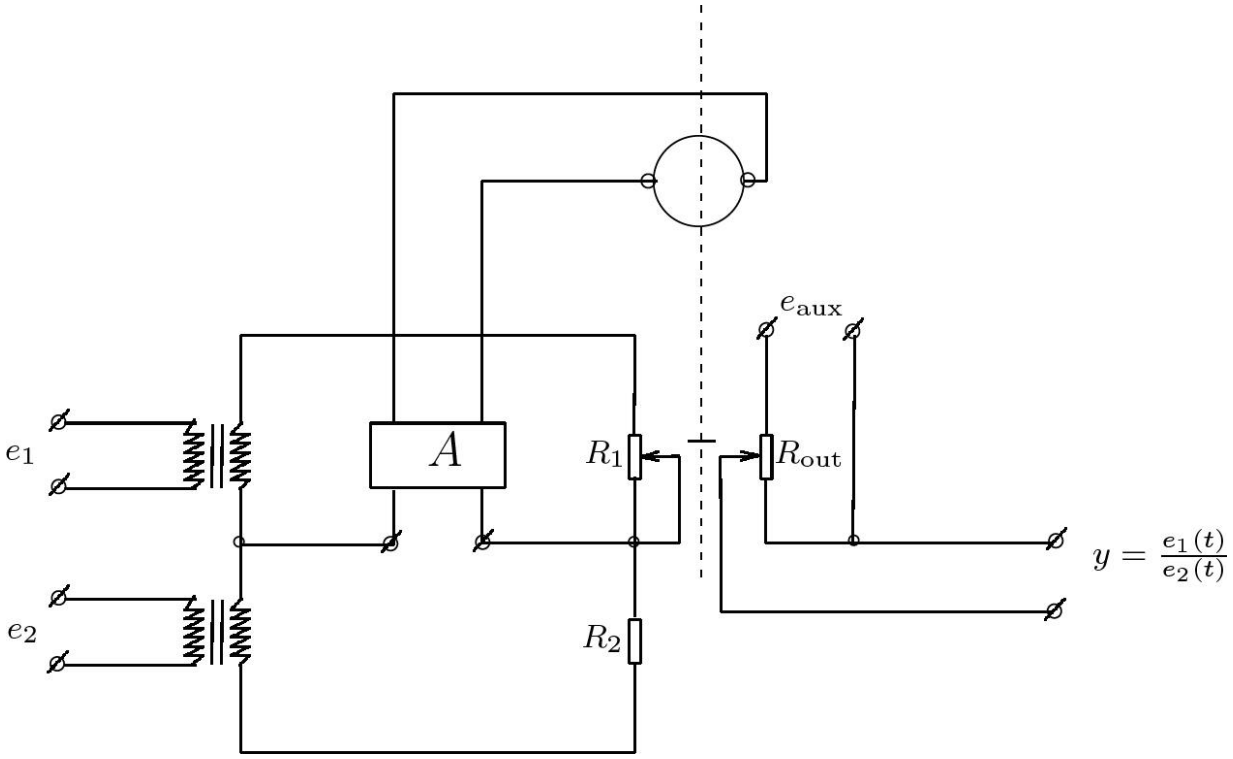


Figure 3.12. Simplified schematic of HOELZER's self-adjusting bridge (cf. [→ HOELZER 1946/1, fig. II, 11])

Since R_1 and R_{out} are driven in tandem this self-adjusting bridge yields

$$y = \frac{e_1(t)}{e_2(t)}$$

at the output of the voltage divider R_{out}, which needs an auxiliary supply voltage e_{aux}.

However, this divider circuit was not implemented during war time since the necessary servo motor could not be obtained. Instead Hoelzer implemented a divider based on the solution of the differential equation

$$\frac{1}{A}\dot{y} + ax = b.$$

where *A* denotes the gain of the amplifier used in the setup. *A* should be as high as possible to minimize errors. The multiplication *ax* was implemented by two ring modulators connected in series.

→ Figure 3.13 shows an improved integrator circuit that also was proposed by Hoelzer. The novel idea here is to simplify the integrator by using only a single AC coupled amplifier instead of two such stages. Since the integration capacitor needs a DC current to be charged, Hoelzer's idea was to place the capacitor at one end of a ring modulator while the other side of the ring modulator was used to close a feedback loop over the amplifier. This idea anticipated the basic integrator design that would dominate analog electronic analog computers for the following decades.

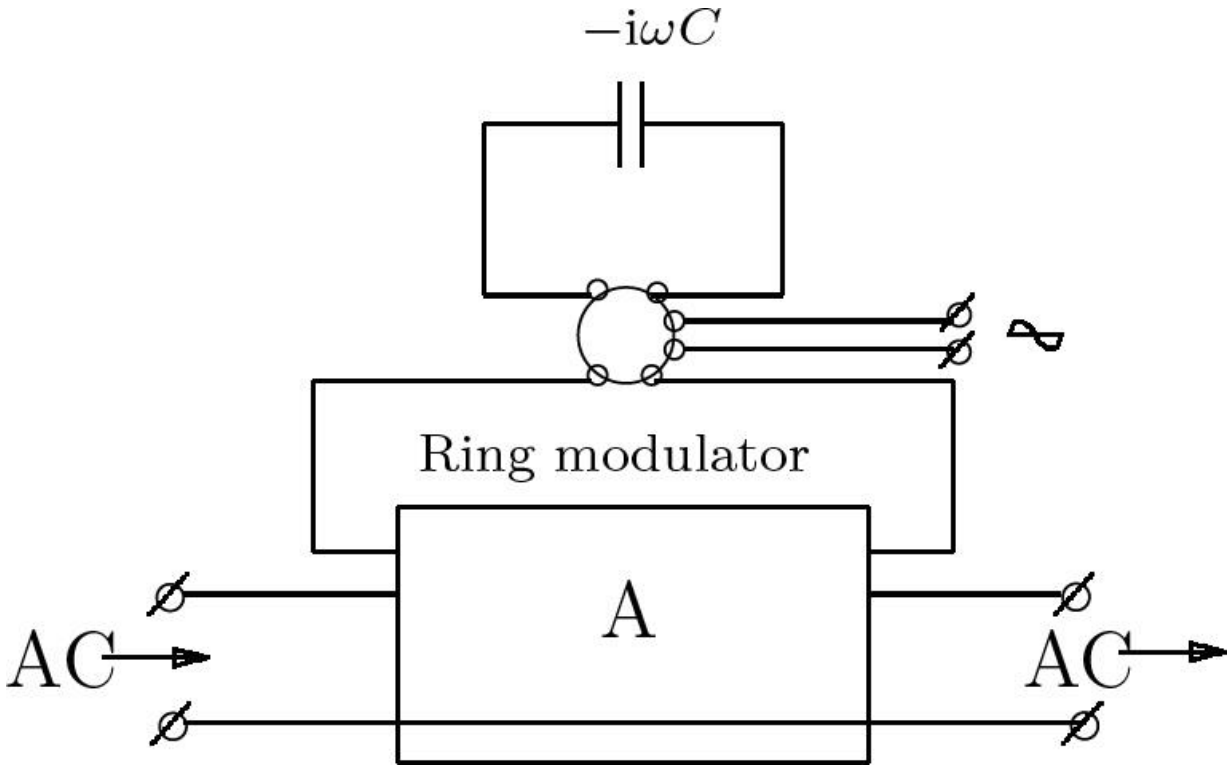


Figure 3.13. An improved integrator circuit (cf. [→HOELZER 1946/1, fig. II, 11])

Unfortunately, this circuit was also not realized during World War II since integrators based on the original design (→figure 3.10) had already been built and worked reliably and satisfactorily.

HOELZER gives a good example how this first analog computer was programmed in his 1946 dissertation[39] where he outlines three different approaches for solving the following differential equation of fourth degree:

$$\dddot{y} + ay + b\ddot{y} + c\dot{y} + dy = f(t). \tag{3.1}$$

→Figure 3.14 shows a straightforward approach using four differentiating elements in series to derive $\dot{y}$, $\ddot{y}$, $y\square$, and $y\square$ from

y. The required *y* is generated by adding these derivatives (multiplied by proper coefficients) and an input function *f*(*t*).[40]

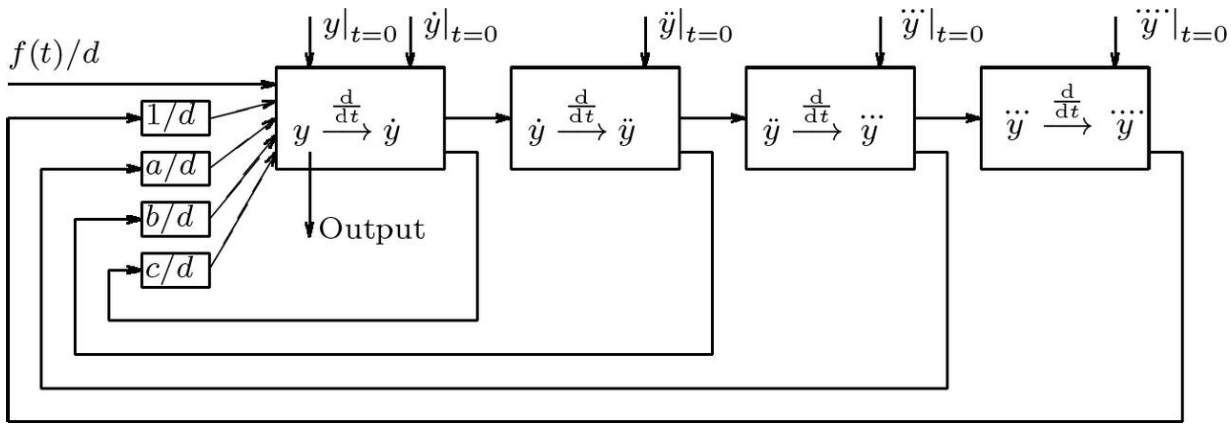


Figure 3.14. Solution of a differential equation of fourth degree $\ddddot{y} + ay + b\dddot{y} + c\ddot{y} + d\dot{y} = f(t)$. with a feedback circuit employing four differentiators (cf. [→HOELZER 1946/1, fig. II, 2])

As straightforward as this solution is, it has a severe drawback: The differentiator chain will increase noise in the variables used, despite the low pass filters employed in each differentiator. Being aware of this fact HOELZER proposed a second approach using four integrators, which is shown in →figure 3.15. This setup is the same any later analog computer programmer would have proposed. In fact, this computer setup results from the application of the KELVIN feedback technique,[41] which HOELZER probably wasn't aware of.

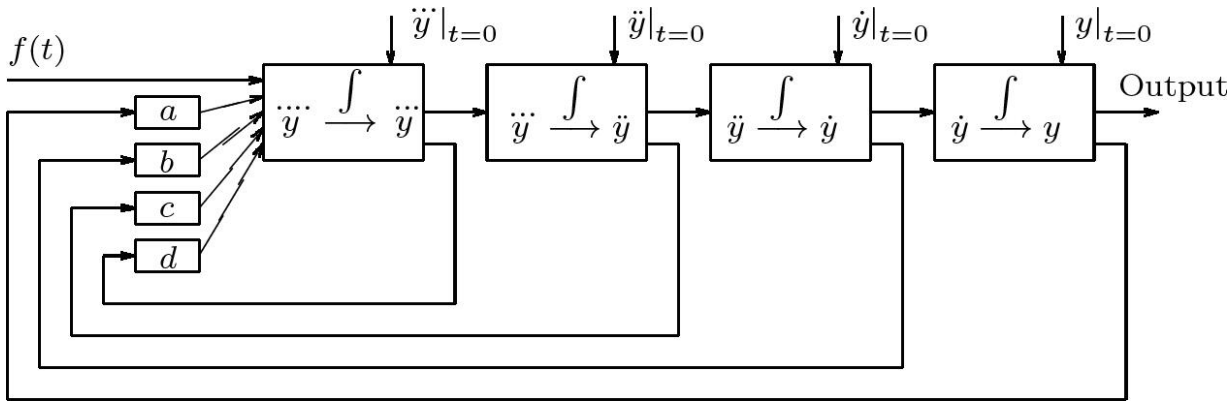


Figure 3.15. Solution of a differential equation of fourth degree $\dddot{y} + ay + b\ddot{y} + c\dot{y} + dy = f(t)$. with a feedback circuit employing four integrators (cf. [→HOELZER 1946/1, fig. II, 1])

Unfortunately, this solution was not without problems, either. It solved the inherent noise problem of the purely differentiator based approach but introduced yet another problem due to the non-ideal characteristics of the computing elements, namely the capacitors: Every integrator introduced some amount of drift due to leakage in its integration capacitor. Thus, this circuit was also not perfect for solving equation (→3.1).

To alleviate these problems HOELZER suggested a third setup shown in →figure 3.16. This approach is a combination of the integrator and differentiator based setups and uses two differentiators as well as two integrators, thus minimizing the negative effects of each type of computing element.[42]

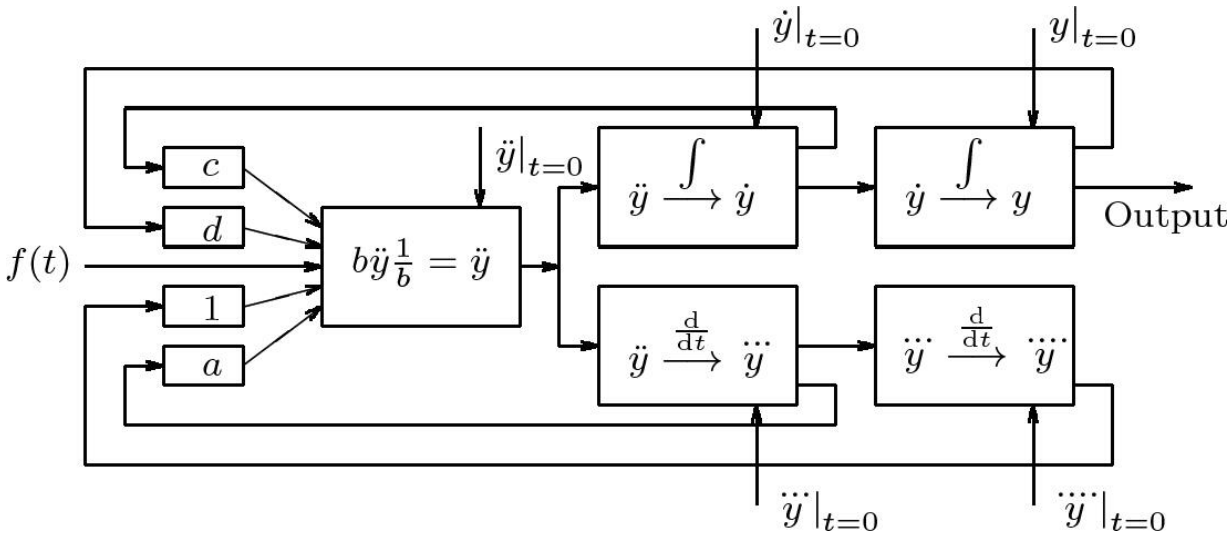


Figure 3.16. Solution of a differential equation of fourth degree $\ddddot{y} + ay + b\ddot{y} + c\dot{y} + dy = f(t)$. with a feedback circuit employing two differentiators and two integrators (cf. [→HOELZER 1946/1, fig. II, 3])

→Figure 3.17 shows the analog rocket motion simulation computer as it was built in the first half of the 1940s. The three top frames contain various computing elements such as integrators, differentiators, a model transmitter, and meters for the various angles, flight time, etc. Below is an electromechanical function generator based on multiple cams mounted on a central shaft driven by a motor which made variable coefficient simulations possible. The bottom compartment contains the power supplies and a motor-generator unit producing the carrier frequency signal for the modulators and demodulators.[43] However, this machine was not a true general purpose analog computer as its computing elements could not be connected freely. Instead it featured a fixed internal structure optimized for its task of simulating rocket motion.

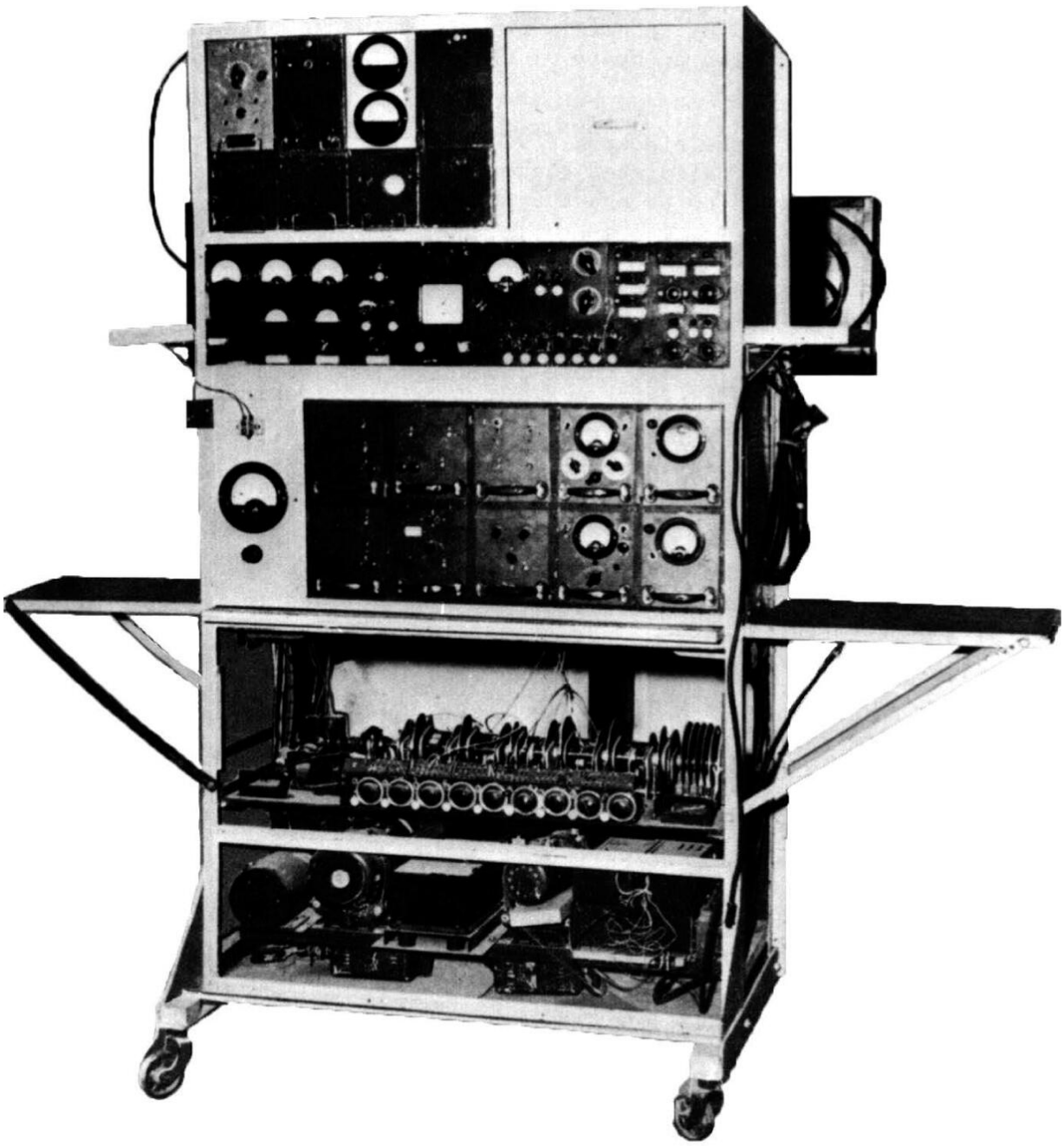


Figure 3.17. HOELZER's analog computer for rocket motion simulation as it was found after World War II (source: NASA, Marshall Space Flight Center)

Two of these analog computers were eventuall built,[44] one of which was taken to the United States where it was used until

about 1955.[45] It was used for various early rocket development projects such as the *Hermes* rocket. Based on this machine an improved version was developed under VON BRAUN in 1950 that was used for about ten years and proved to be a valuable tool in the development of the *Redstone* and *Jupiter* rockets as well as for the design of the first satellite of the United States, *Explorer I*.[46]

In the 1990s HELMUT HOELZER built a replica of his analog computer for a museum exhibition. →Figure 3.18 shows him working on this machine. It slightly differs from the original system as some parts were difficult or impossible to procure more than fifty years later. One notable problem was the rotary converter used in the original machine which, among other voltages, generated the sine signals used in the modulator circuits. This was substituted by a suitable vacuum tube circuit, something that would have been also possible in the 1940s.

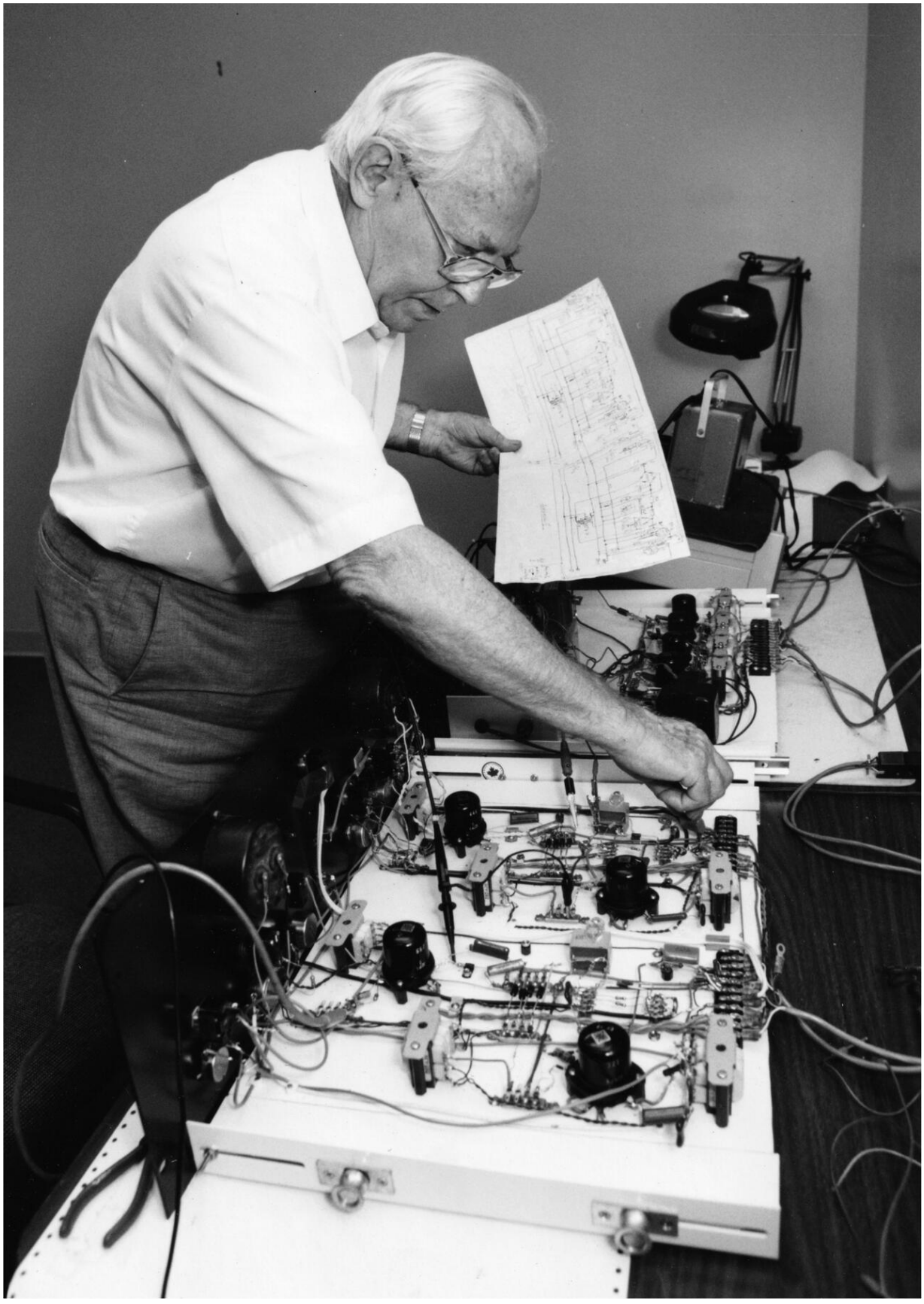


Figure 3.18. Helmut Hoelzer working on the replica of his analog computer in the 1990s; one can even see two diode bridge modulators in the schematic he holds in his left hand (the author would like to thank Barbara Hoelzer-Beck for permission to use this picture)

A noteworthy detail visible in this picture are the *Stahlröhren*, vacuum tubes developed by Telefunken in the 1930s. These tubes were quite special as they had no glass envelope but were housed in a steel enclosure with the actual tube system mounted horizontally. This not only resulted in a very high mechanical stability and good shielding against external electromagnetic fields, but also made these tubes especially well suited for RF applications as the leads from the socket to the various electrodes in the tube system were much shorter than with traditional vacuum tubes. Originally developed for radio applications, these tubes were used extensively in the Mischgerät, military transmitters, and receivers, as well as in Hoelzer's analog computer.[47]

[→ Hosenthien et al. 1962, pp. 453 ff.] describe a follow-up system dubbed *ac analog computer* that used suppressed-carrier amplitude modulation to represent values and a multi-cam function generator to introduce time-varying coefficients into a computation. It surpassed its predecessor system quite substantially by allowing the computing elements to be interconnected by means of patch cables, thus offering a much higher degree of flexibility. The basic computing elements were also improved with respect to their accuracy.

3.2 George A. Philbrick's Polyphemus

In 1938 GEORGE A. PHILBRICK,[48] who worked for Foxboro,[49] wrote a proposal describing a novel *simulator*[50] for process control systems. The main goals of the development were stated as follows:[51]

> "*We attempt to describe a method for the rapid and easy solution of problems which arise in connection with the technical study of process control. Also included is an electrically operated unit capable of disclosing the behaviour of controlled systems as influenced by their various physical characteristics*."

PHILBRICK started with simple analogies like a capacitor modeling a tank, a resistor representing a valve restricting the flow of a medium, etc. →Figure 3.19 shows an example used by PHILBRICK in 1938. Depicted are three tanks T_1, T_2, and T_3 connected by pipes P_1 and P_2, which contain valves V_1 and V_2. These series connected tanks can be filled via P_{in} and drained by P_{out}. The latter pipe also contains a valve V_{out}.

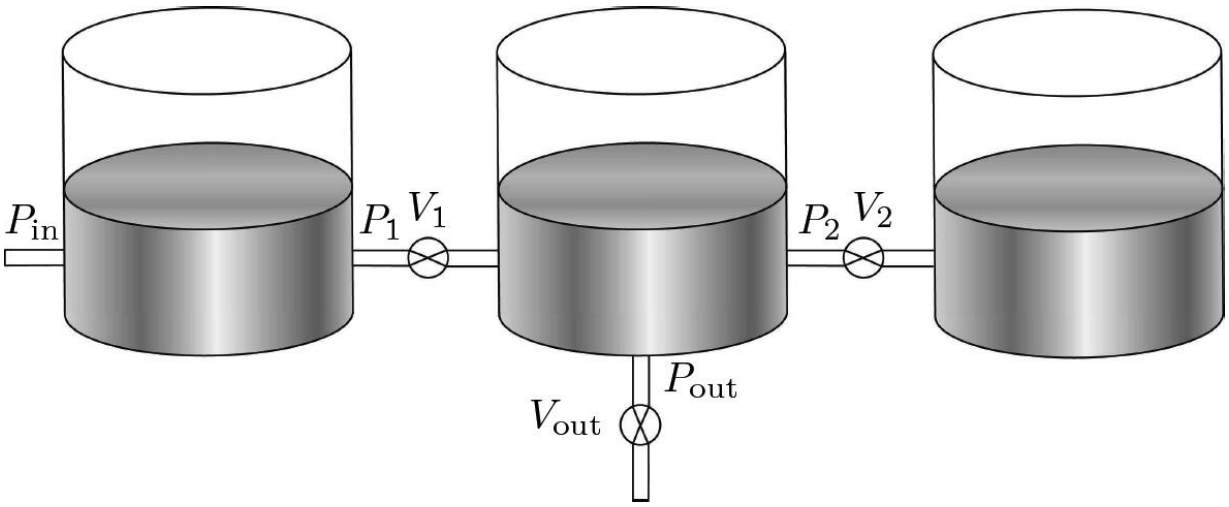


Figure 3.19. Three interconnected tanks (cf. [→HOLST 1982, p. 146])

At first, he proposed to model this three tank system by means of passive electronic components, with resistors representing the valves and capacitors representing the tanks as shown in →figure 3.20.

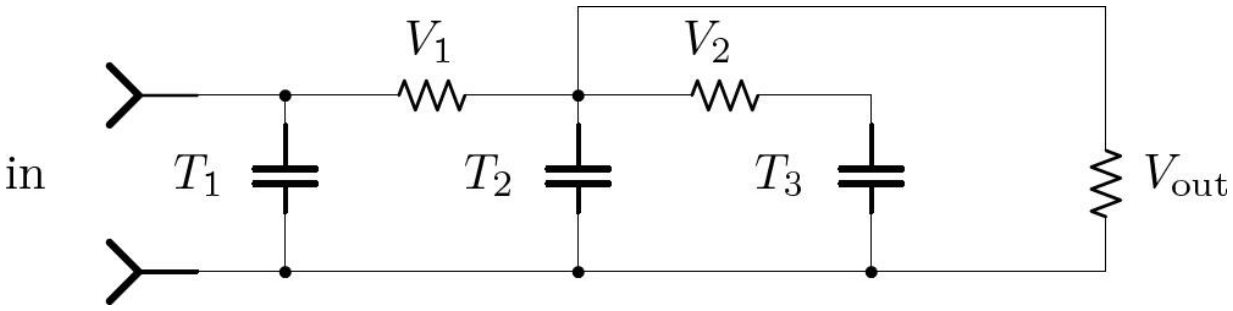


Figure 3.20. Simulation circuit for the three interconnected tanks of →figure 3.19 (cf. [→Holst 1982, p. 146])

Of course more complex simulations would require some kind of amplifier so Philbrick was faced with pretty much the same problem as Hoelzer in Germany: No such devices existed in the late 1930s and state of the art DC amplifiers showed excessive drift, making precise simulations impossible. Consequently, like Hoelzer, Philbrick decided to use AC amplifiers in some of his simulators[52] but omitted the modulator and synchronous demodulator circuits since his simulator was intended to be used in a mode of operation now known as *repetitive operation*. In this mode a simulation is run automatically over and over again in quick succession which makes it possible to display the solution of the problem as a flicker-free graph on an oscilloscope. Changing the settings of the potentiometers representing coefficients of the system being simulated causes this graph to change immediately, making this an incredibly powerful tool for investigating the behavior of dynamic systems.

Based on these ideas PHILBRICK set out to develop a process simulator that was eventually called *Polyphemus* due to its appearance, with a single oscilloscope mounted in the top section of a 19-inch rack holding the simulator's components. With some imagination this setup looked a bit like Polyphemus, the Cyclops of the Odyssey.

Although the basic structure of this simulator was fixed, it could be adapted to different simulation tasks, within limits, by means of a removable cardboard front plate. This plate contained a schematic diagram showing the structure of the process to be simulated and annotations for the various control elements of the simulator like potentiometers, etc. →Figure 3.21 shows Polyphemus with a faceplate representing a system consisting of two liquid baths with stirrers as well as steam and cold water inlets.[53] Other face plates depicting pneumatic controllers, etc., were also available.

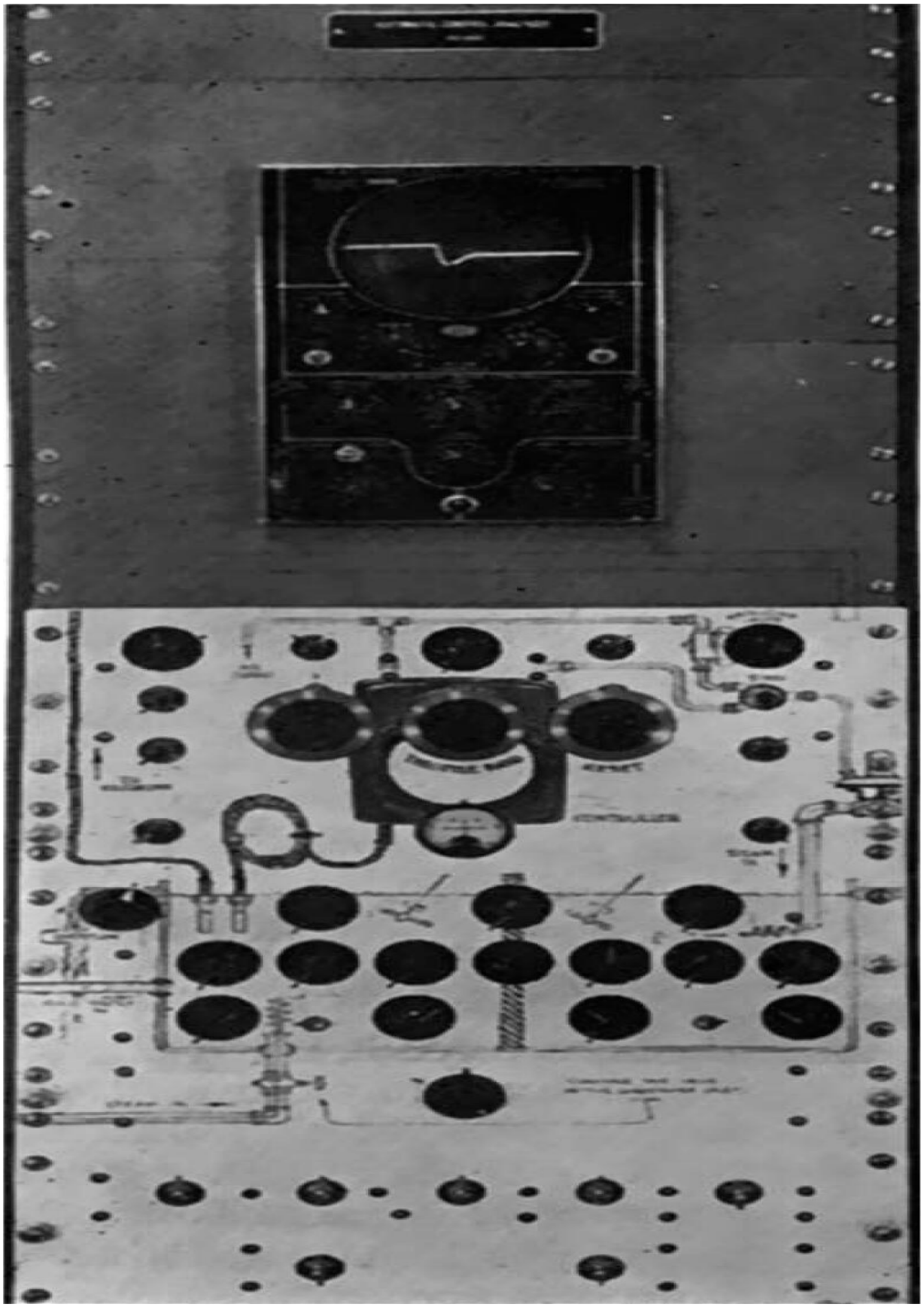


Figure 3.21. George A. Philbrick's Polyphemus (see [→ Philbrick 1948, p. 108])

Although Polyphemus was intended as a tool for basic research it turned out to be much more versatile and useful. Eventually, it was mainly used as a training simulator and as an aid for sales people that could easily be used to demonstrate the behavior of a proposed process controller to prospective customers.[54] In retrospect Polyphemus has been described as follows:[55]

> "*In those days, an electronic analog machine was a pioneering venture. The Kelvin-Bush differential analysers were mechanical, expensive, and very large. Appropriate electronic techniques, if they existed, were not available. Nevertheless, it was evident that only this medium offered the required flexibility and speed, not to mention economy*."

This world's first all electronic process simulator is now part of the collection of the *Smithsonian Institution* – its successor, featuring two oscilloscopes and thus no longer looking like a Cyclops, was in active use at Foxboro well into the 1980s.[56]

To get an impression of the complexity of the circuits developed by George A. Philbrick for his numerous process simulators, →figure 3.22 shows the schematic diagram of a seven stage circuit that models a process controller – particularly noteworthy are the two diodes for implementing nonlinear responses as well as the DC coupled partial circuits.

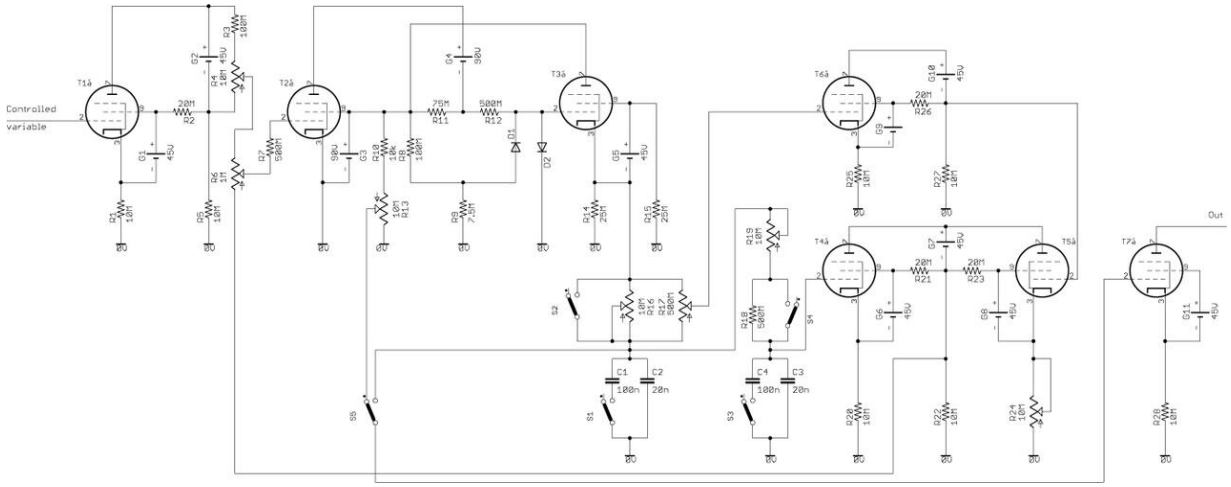


Figure 3.22. Seven-stage amplifier from GEORGE A. PHILBRICK's laboratory notebook, March 11, 1940 (see [→HOLST 1982, p. 154])

In 1946 GEORGE A. PHILBRICK founded a company named *GAP/R*, short for *George A. Philbrick Researches*, that developed and sold products ranging from operational amplifiers[57] to fully fledged analog computers and simulators. These owed much to his early experiences with Polyphemus and its successors. One of the most famous products of his company was the first commercially available operational amplifier, the *K2-W*,[58] which shaped a whole industry. In 1966 GAP/R merged with *Teledyne* to form *Teledyne-Philbrick*.

3.3 Electronic fire control systems

The decline of mechanical fire control systems like those described in →section 2.7 had already started before the advent of World War II.[59] This was not due to insufficient accuracy or precision or even slow response times – in fact, the military feared that in the case of a war the sheer complexity of these machines would prevent their production in sufficient quantities. The main obstacles were that the necessary parts and resources

would be short in supply in such a situation and the necessary highly skilled craftsmen would not be available due to other war effort demands. To make things even worse the rapid development of radar systems also rendered the classic fire control systems obsolete since they could not be directly coupled with radar stations.

In 1940 David B. Parkinson, who worked for the *Bell Telephone Laboratories* (*BTL*), and his coworkers were developing an *automatic level recorder*, a kind of strip-chart recorder employing a logarithmic scale. To drive the recorder's pen a servo mechanism based on a feedback scheme was necessary which inspired Parkinson to think about using this technique for the automatic control of azimuth and elevation of a gun.[60] His supervisor, Clarence A. Lovell, was enthusiastic about this idea and together they started a thorough investigation of the necessary technology and mathematics.[61]

It was decided to use DC voltages to represent the various variables in a fire control calculation. Coordinates of the target and the gun were represented in a three-dimensional Cartesian coordinate system requiring computer circuits for implementing trigonometric functions, summing, differentiation, etc. Target speed components would be generated by electronic differentiation of the position signals. Parkinson summarized the main features of the proposed fire control computer as follows:[62]

> "*It required (1) a means of solving equations electrically (potentiometers), (2) a means of deriving rate for prediction (an electrical differentiator), and (3) a means of moving the guns in response to firing solutions*."

The patent application[63] states this explicitly:

> "*An important form of the invention is a device which, when supplied with electrical voltages proportional to two sides of a triangle, will set itself to indicate an angle of the triangle and to produce a voltage proportional to the other side of the triangle*.
>
> *Another form of the invention is a device which, when supplied with voltages proportional to the rectangular coordinates of a point, will set itself to indicate the polar coordinates of the point*."

The basic problem of aiming a gun is shown in →figure 3.23. The instantaneous Cartesian coordinates of a target *T*, provided by the tracking radar, with respect to the position *D* of the gun are given as *x, y*, and *z*. Based on these, the gun's azimuth and elevation angles *α* and *ε* have to be computed, taking into account the future predicted position of the target and the ballistic characteristics of the shell.

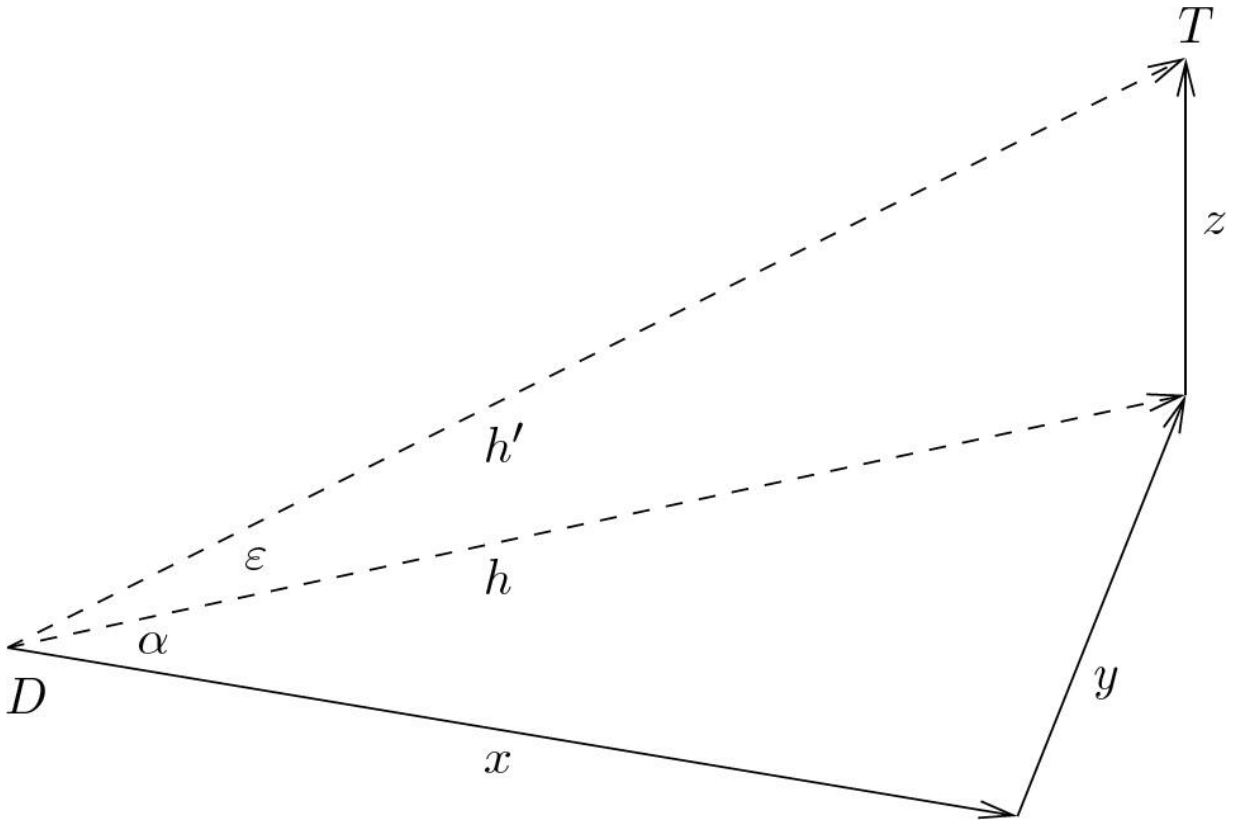


Figure 3.23. Basic fire control task

→Figure 3.24 shows the structure of a trigonometric function generator developed by LOVELL, PARKINSON, and WEBER, which was used in their electronic fire control computer. The motor on the far left is running with constant speed and drives two bevel gears 3 and 4, which rotate in clockwise and counterclockwise direction, respectively. Two electromagnetic clutches, 5 and 6, can be activated by a control signal, thus controlling which gear will be coupled to the vertical shaft, which in turn drives the two potentiometers shown at the top left of the figure.

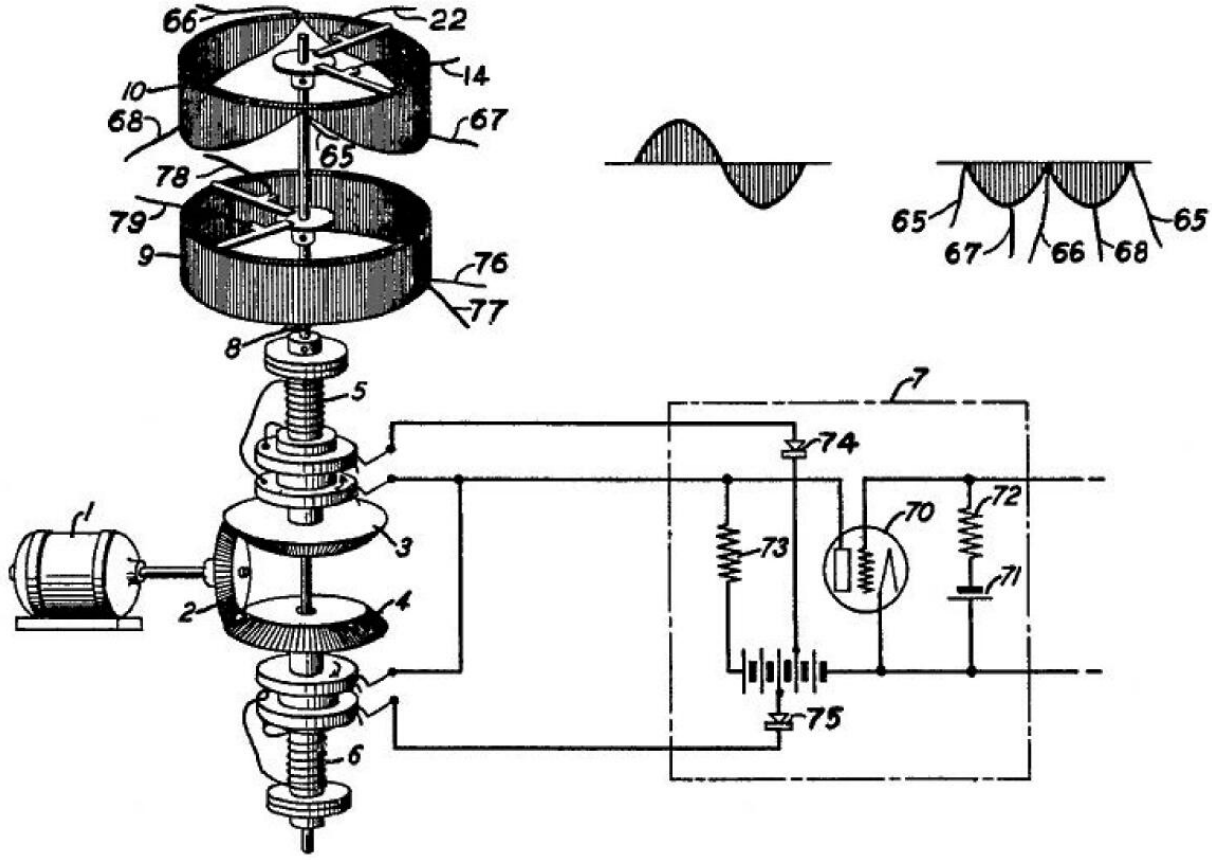


Figure 3.24. Sine/cosine generator used in the T-10 fire control computer ([→LOVELL et al. 1946, fig. 1/2])

The control signals for these two clutches were generated by the comparator circuit shown in the box denoted by 7. This circuit can generate three different control signal combinations: Upper clutch, lower clutch, or no clutch at all will be engaged. These states are controlled by the input signal connected to the appropriately biased grid of the triode labeled 70.

It should be noted that the lower of the two potentiometers shown is a linear potentiometer yielding a voltage varying linearly with the angular position of its wiper when connected as a voltage divider, while the upper potentiometer is of a special type, wound to deliver a voltage corresponding to the desired trigonometric function of the wiper angle.

The combination of this circuit with an amplifier finally yields a servo system that can be used to automatically compute sin(α), etc., given an angle α. This forced Lovell, Parkinson, and Weber to develop an amplifier circuit as shown in →figure 3.25, which is a precursor of later *operational amplifiers*.[64]

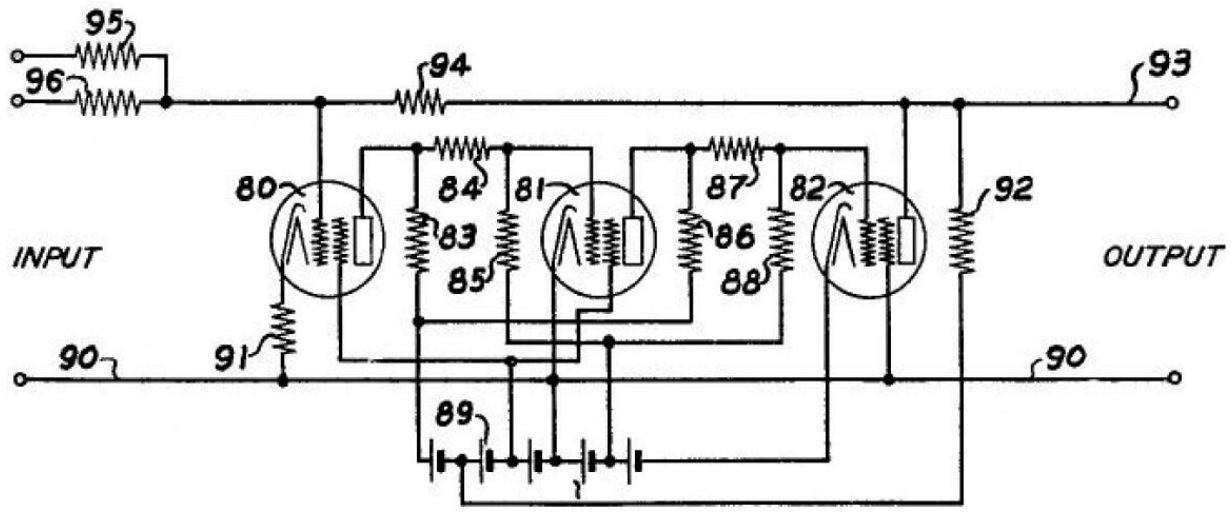


Figure 3.25. Operational amplifier used in the T-10 fire control computer ([→Lovell et al. 1946, fig. 5])

This amplifier acts as an inverting summer[65] with two inputs connected to the resistors 95 and 96 and a feedback resistor 94. Using this amplifier and the device shown in →figure 3.24, a servo system can now be implemented by connecting the output of this summer to the input of the comparator controlling the two clutches. The inputs of the summer are connected to the output of the linear potentiometer and a circuit yielding *α*. Whenever *α* deviates from the angular position of the potentiometers' wiper the summer will generate a positive or negative output signal thus forcing the comparator to engage either the upper or the lower clutch to minimize the error between the wipers' position and the value *α*.

Lovell, Parkinson, and Weber realized that the computing elements they proposed were not limited to fire control but could be used to solve a variety of problems as the following quote from [→Mindell 1995, p. 74] shows:

> "*A digression from the principal subject is made to comment that the use of servo mechanisms to solve simultaneous systems of equations is feasible and, in a large number of cases, practicable. This fact may lead to the application of this type of mechanism to the solution of many types of problems dissociated from the one in question.*"

The actual circuit developed for the implementation of a coordinate transformation relies on the observation that

$$x = h\ \cos(\alpha) \tag{3.2}$$

$$y = h\ \sin(\alpha) \tag{3.3}$$

given the triangle *x, y, h* in →figure 3.23. (→3.2) and (→3.3) can be expanded to

$$x \sin(\alpha) = h \cos(\alpha) \sin(\alpha)$$
$$y \cos(\alpha) = h \sin(\alpha) \cos(\alpha)$$

yielding $x \sin(\alpha) - y \cos(\alpha) = 0$, which is used to control the basic servo circuit of the coordinate resolver shown in →figure 3.26.

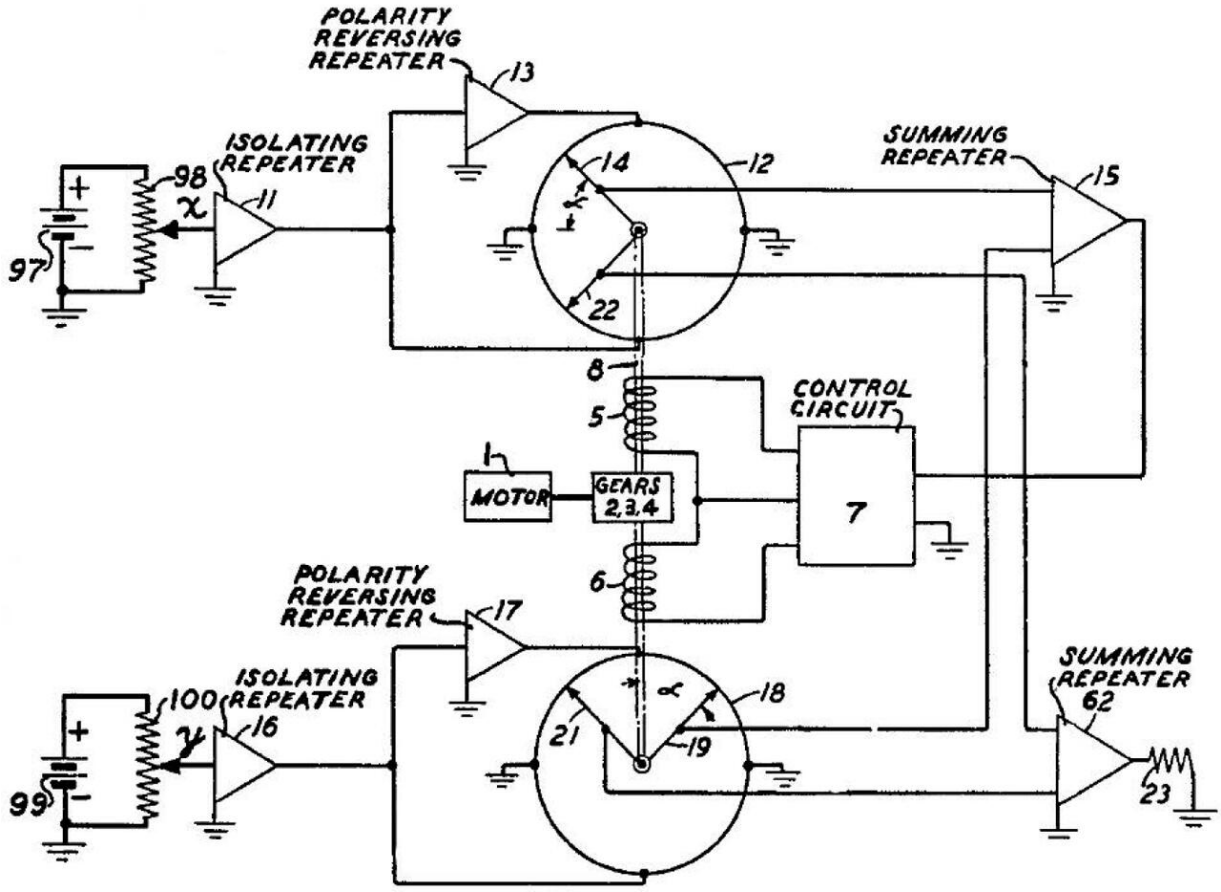


Figure 3.26. Coordinate resolver used in the T-10 fire control computer ([→Lovell et al. 1946, fig. 4])

This circuit has two inputs *x* and *y*, which are connected to two specially wound potentiometers 12 and 18 with four connections each. The two *polarity reversing repeater* devices are in fact summers with only one input each and gain 1.[66] Both potentiometers are connected to −*x*, ground (representing the

value 0), *x*, 0 and −*y*, 0, *y*, 0 respectively. Each potentiometer has two wipers mounted at a relative angle of $\pi/2$ driven by a common shaft that can be connected to one of the two bevel gears which are in turn driven by a motor by means of the two clutches denoted by 5 and 6.

The potentiometers are wound in such a way that their resistance represents $\sin(\varphi)$ and $\cos(\varphi)$ with φ denoting the wiper angle (due to the two wipers mounted at a relative angle of $\pi/2$ with respect to each other, both functions are generated simultaneously using only one such potentiometer). Thus, the wipers 14 and 22 yield $x \sin(\alpha)$ and $x \cos(\alpha)$ while the lower potentiometer generates $-y \cos(\alpha)$ and $y \sin(\alpha)$ at its wipers 19 and 21, respectively. The summing *repeater* 15 now computes the value

$$c = x \sin(\alpha) - y \cos(\alpha), \tag{3.4}$$

which is used to control the comparator 7 driving the two clutches, thus closing the servo loop. As long as *c* equals zero, both clutches are disengaged. As soon as *c* deviates from zero, one of the clutches will be activated, thus connecting the central shaft driving the two potentiometers to the motor, which will in turn minimize the error term *c* until it reaches zero again. Thus, the angular position of the wipers is α, which can be used to control the gun’s azimuth.

The second summing repeater, 62, computes $h = x \cos(\alpha) + y \sin(\alpha)$, which is the horizontal range to the target. Using this value and *z* as inputs for a second, identical circuit, eventually yields *h*' and the elevation angle ε represented by its potentiometer shaft position. It was even suggested to make the motors driving these two sub-circuits strong enough so that they could actually drive the gun directly to point it to the target *T*.

The resulting fire control computer *T-10* consisted of four such servo circuits as well as 30 summing amplifiers and five power supplies. Since the voltages representing the Cartesian coordinates of the target with respect to the gun's location often changed quite slowly, the amplifiers had to be DC coupled amplifiers – thus introducing unavoidable drift effects into the calculations. Assuming that amplifier 15 of →figure 3.26 exhibits a drift error voltage *e* this would cause the servo circuit to try making

$$c = x\sin(\alpha) - y\cos(\alpha) + e$$

zero instead of the term (→3.4). This, in turn, results in non-negligible errors regarding *α, ε, h* and *h*'. Accordingly, the T-10 computer featured additional inputs to its summing amplifiers, which were connected to potentiometers. These had to be adjusted manually to minimize the drift voltage *e* of each amplifier. In addition to that problem, the T-10 relied on the computation of derivatives, thus introducing additional noise into the control signals used to drive the servo circuits. This required the implementation of complex filter networks, further complicating its circuitry.

In spite of these shortcomings, the T-10, which was eventually produced under the designation *M-9* and coupled directly to a *SCR-584* radar system, proved to be greatly superior to the earlier mechanical fire control computers. Due to its easier manufacturing, it was decided on February 12, 1942 to stop all developments of mechanical fire control systems and to concentrate on all electronic devices in the future.[67] The hopes were high regarding the M-9:[68]

> "*The M-9 Director, electrically operated, is, we feel in Ordnance, one of the greatest advances in the art of fire control made during this war, and we anticipate from the M-9 Director very great things as the war goes on*."

These hopes were not in vain as it turned out – the M-9 was deployed in large numbers[69] to Great Britain to shoot down *V1* German flying bombs. Initially 10 to 34 percent of the incoming flying bombs could be downed automatically and sometimes even higher rates were achieved:[70]

> "*In a single week in August, the Germans launched 91 V1's from the Antwerp area, and heavy guns controlled by M-9's destroyed 89 of them*."

Using the M-9, coupled with the use of proximity fuses, brought the number of shells required to hit an enemy aircraft down from many thousand to a mere 100. Hitting an incoming V1 flying bomb required about 200 shells.[71]

Later developments such as the experimental fire control computer *T-15* took many more parameters and effects into account, such as drift of the shell during flight, windage correction, muzzle velocity, air density, etc.[72] The T-15 performed all computations in polar coordinates, thus eliminating the coordinate system transformations otherwise necessary.[73]

The technologies developed for these fire control computers formed the basis for following developments like a trainer for aerial gunners[74] that already used continuously operating servo systems in contrast to the controlled clutches of the T-10. Even more complex was the *Flight Training Apparatus* developed in 1948 by Dehmel,[75] which implemented the basic operation of integration by means of a servo system controlling a potentiometer, the angular wiper position of which corresponds to the integral of the function applied to the servo circuit, etc.

3.4 MIT

In the fall of 1945 A. B. MACNEE set out to develop an all electronic analog computer at the MIT because[76]

> "*[…] it was felt that there was considerable need for a differential analyser of somewhat different characteristics from any then in existence or under development. There appeared to be the need for a machine having the following characteristics: (a) moderate accuracy, of perhaps 1 to 10 per cent, (b) much lower cost than existing differential analysers, (c) the ability to handle every type of ordinary differential equation, (d) high speed of operation, (e) above all, extreme flexibility in order to permit the rapid investigation of wide ranges of equation parameters and initial conditions*."

These goals summarize the main advantages of an all electronic approach to building analog computers. The high-speed operation[77] mentioned is a very important feature since it allows the behavior of systems modeled on the analog computer to be explored in a highly interactive way and often even faster than real time. The operator can change the parameters of a simulation and immediately see the effects these changes have on the system dynamics.

→Figure 3.27 shows parts of the system developed by MACNEE, consisting of integrators, summers, a high-speed function generator,[78] and a high-speed multiplier.[79] A detailed account of this system can be found in [→MACNEE 1949].

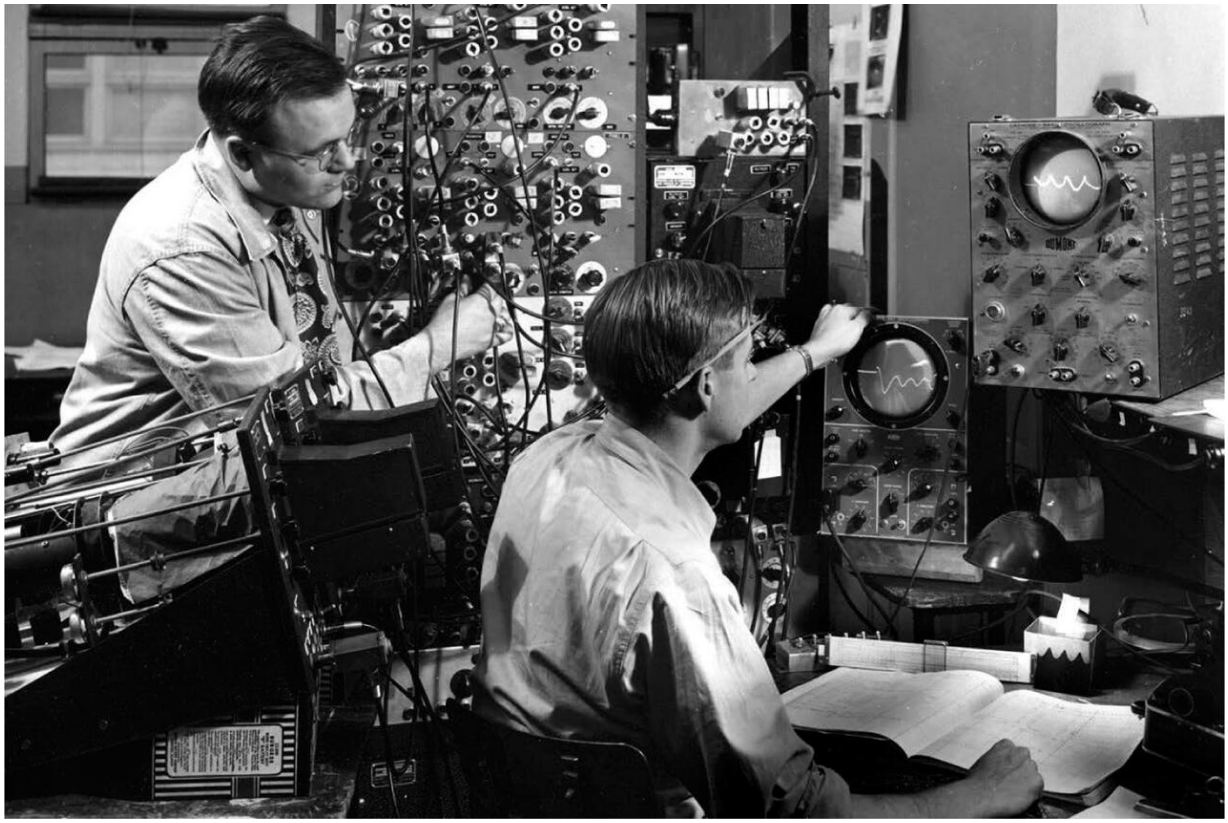


Figure 3.27. MACNEE's high-speed analog computer

One of the novel ideas employed in this computer was that it ran in a repetitive mode of operation with a repetition rate of 60 Hz between individual computer runs, thus allowing a flicker-free picture to be displayed on an oscilloscope. Apart from the fact that this greatly facilitated interactive operation where an operator can change various parameters of a simulation, it had the additional benefit that AC coupled amplifiers could be used in the computer since even constant values would continuously reset 60 times a second due to the repetitive operation. Thus, it was possible to completely avoid using DC coupled amplifiers with their inevitable drift.

As an example of the capabilities of this early machine, →figure 3.28 shows a particular solution for the *MATHIEU equation*[80]

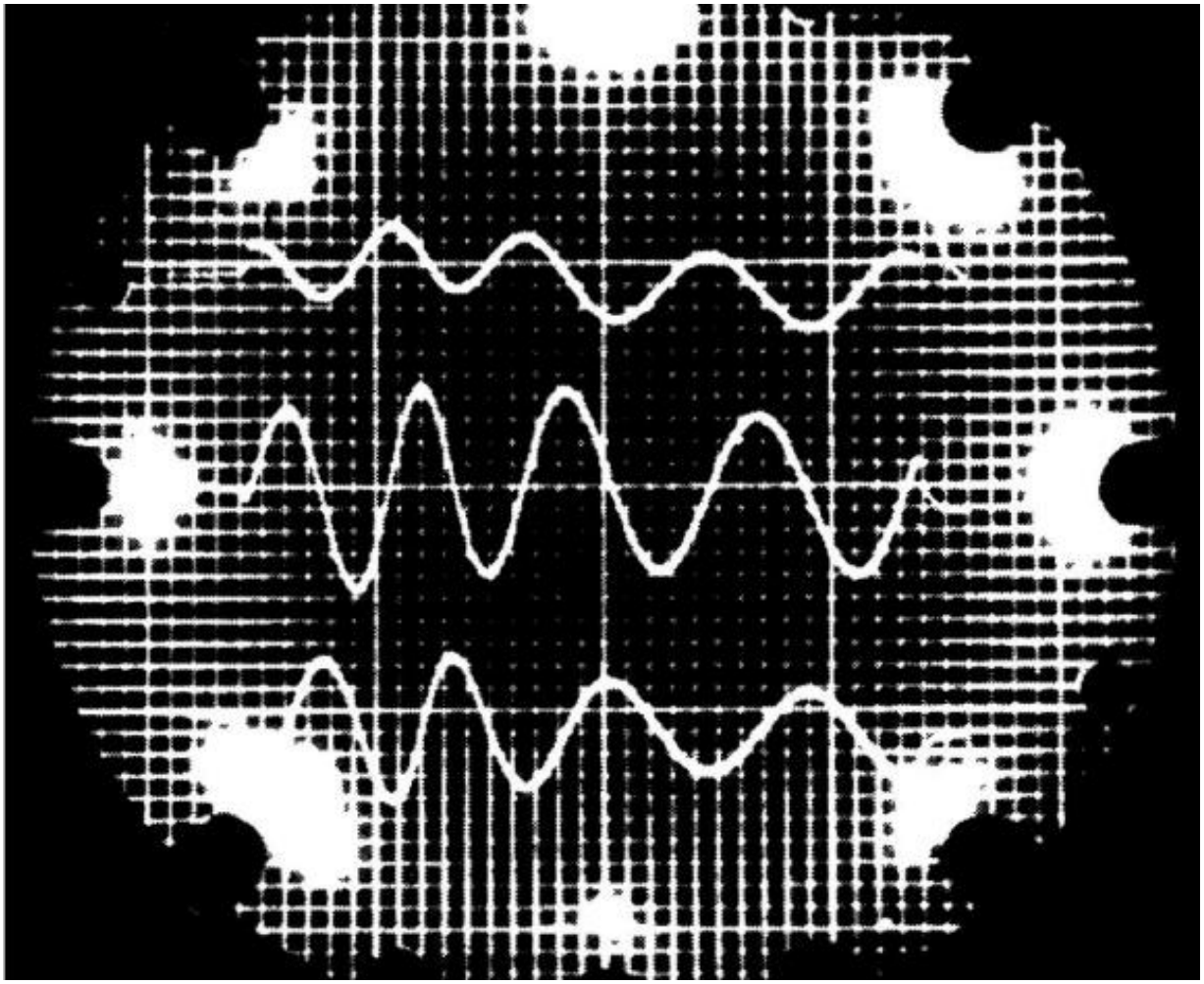


Figure 3.28. A particular solution of the MATHIEU equation (→3.5) obtained with MACNEE's all electronic analog computer, 1948

$$\ddot{I} + \omega_0^2(1 + \varepsilon \cos(\omega_m t))I = 0 \tag{3.5}$$

obtained by this electronic analog computer. The three curves shown on the oscilloscope's display are I, $-\dot{I}$, and $\ddot{I}$.

No other technology in the 1940s would have allowed the solution of such an equation in only 1/60 of a second. Machines like this MIT analog computer made it crystal clear that the days of mechanical and electromechanical analog computers were gone and the future belonged to high-speed electronic devices.

These machines were destined to become crucial elements for the technological progress of the 1950s to the late 1970s.

3.5 Anacom and the Caltech Computer

Westinghouse Electric Corporation, founded in 1886 by George Westinghouse,[81] was a leading force in the introduction of electricity and played a major role in the (rather infamous) *war of currents*.[82] Planning, construction, and operation of large scale electric networks is a highly complex task due to the intricate dynamics of such systems and the variety of unpredictable external influences such as changes in power demand, power production, lightning strikes, etc. Accordingly, Westinghouse used specialized analog computers such as the *DC Calculating Board* which was developed shortly after Word War I.[83] This system was obviously not capable of dealing with the all-important phase angles in AC networks. This gave rise to the development of a much more sophisticated *AC Calculating Board*[84] that was built in 1929 and used until the 1960s:[85]

> "*The board consists of fifteen cabinets of standard steel office desk construction, in which are located the various units of resistance, reactance, capacitance, power sources for simulating synchronous apparatus, cord-and-jack assembled to permit reproducing any miniature system network to be studied, and suitable current and voltage receptacles whereby the metering equipment may be inserted in any component of the miniature network. In the middle is located a desk containing the metering equipment* [for reading phase-angle, voltage, current, and power], *as well as ordinary office desks for the use of the operators. A synchronous motor-driven generator set supplied three-phase, 220-volt, 440-cycle power. Each source of power, representing a generator station, consists of a three-phase, 220-volt, 100% induction regulator and a phase shifter*."[86]

In the 1940s, Gilbert D. McCann[87] started the development of a general purpose analog computer (*GPAC*) which was in its pilot

stage when he left for *Caltech* (*California Institute of Technology*) as a faculty member in electrical engineering.[88] This machine was to become the *Anacom*, short for *Analog Computer*. The pilot implementation was used by Edwin L. Harder[89] in his Ph.D. thesis who would become McCann's successor as consulting transmission engineer and director of lightning studies at Westinghouse and would continue the development of the machine which would be extended considerably over many years to come.[90] → Figure 3.29 shows Harder at the Anacom pilot.

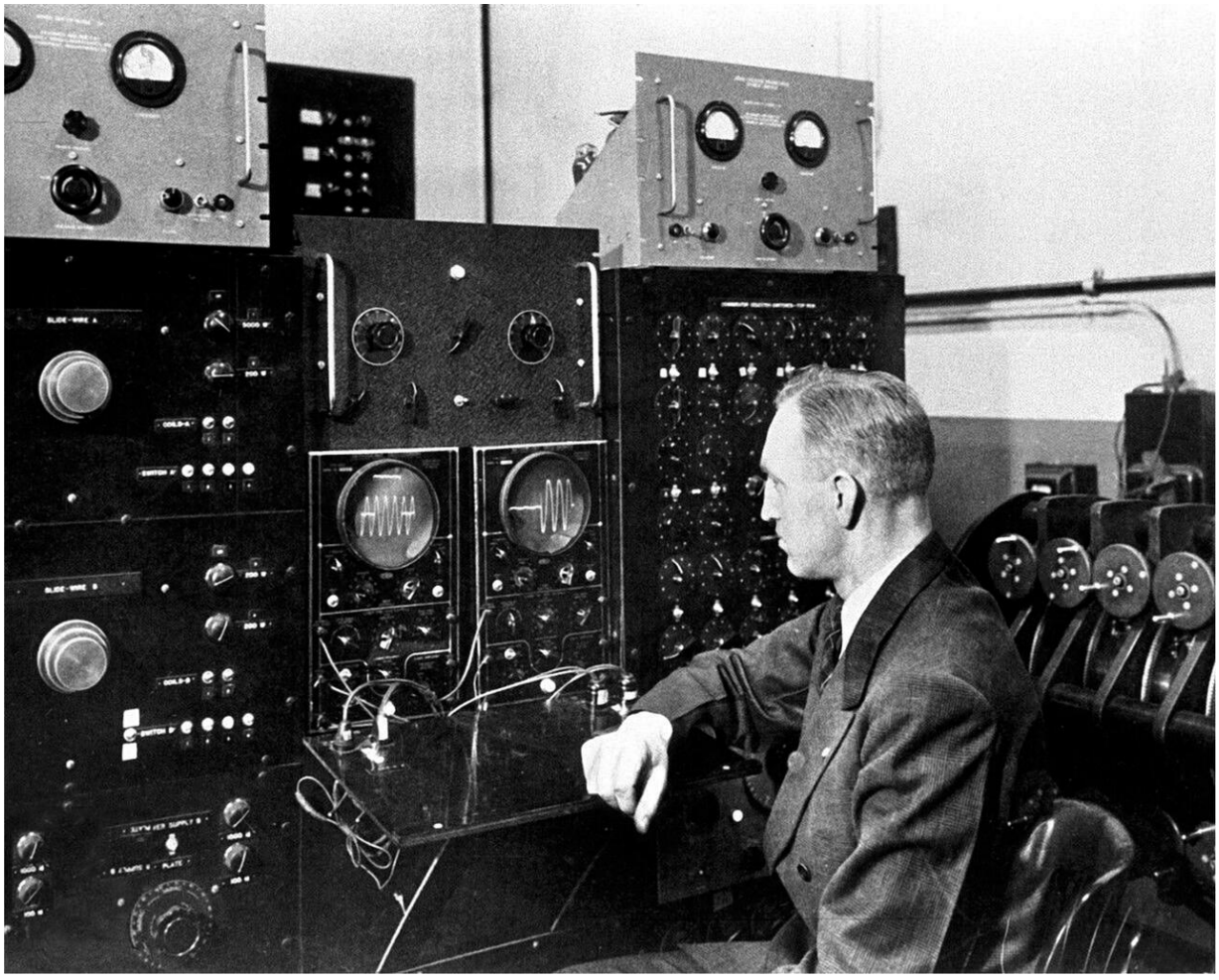


Figure 3.29. Edwin L. Harder at an Anacom pilot machine in 1946

With McCann at Caltech and Harder at Westinghouse it was soon decided to cooperate on the further development of the Anacom.[91] The total construction cost was estimated at 300,000 USD but grew to 500,000 USD eventually. This cooperation resulted in two nearly identical systems being built. The only minor differences were in mechanical construction and the complement of computing elements implemented at each location.[92] The Caltech machine was in fact the first development and construction task of the Caltech *Analysis Laboratory*, established in 1946, and was put into active service as early as 1947 in a partially completed state.[93] Even in this early stage of development the system was successfully applied to solve problems for the *Jet Propulsion Laboratory*[94] (*JPL*).

Compared with the early machines described so far, the *Anacom* and its sibling, the *Caltech computer*, were true behemoths weighing in at about 15 tons each, occupying a considerable amount of floor space.

Anacom proved to be an incredibly powerful tool capable of treating a wide variety of problems ranging from the design of electric networks to mechanical systems, airplane design, and many more. At Westinghouse it became "*the most important computational tool [...] and was used by almost every manufacturing division*."[95] It quickly replaced the older DC and AC Calculating Boards. In fact, the machine should become the most successful of the early analog computers with around 20 to 30 copies having been built and used around the world:[96]

> "*[...] GE* [General Electric] *has one now. The people that run this Anacom wrote to tell me of other systems that existed. They were pretty sure CESI had one in Italy. And I know there's one in England. I know there are others, but we have no complete listing of them*."[97]

It is highly remarkable that Anacom was used and constantly improved over an incredibly long period of time. These improvements resulted in *Anacom II* in the 1970s, followed by *Anacom III* around 1980. While the core components remained essentially unchanged, several substantial improvements and additions prompted the generational name changes. One of the first changes to Anacom was the replacement of the plethora of mechanical switches by electronic vacuum tube based devices, which increased the overall reliability considerably. Anacom II featured an improved patch panel using three-phase connectors to speed up setup times. Eventually, Anacom III was coupled with an *HP* (*Hewlett Packard*) minicomputer, thus basically forming a true hybrid computer.[98] This setup even allowed statistical studies being performed.[99] Anacom III was eventually bought by *Asea Brown Boveri* in the 1980s and used until 1991.[100]

The Anacom differed substantially in its structure from the machines described previously as it used mainly passive components such as individual resistors, capacitors, inductors, transformers[101] as well as operational amplifiers and multipliers as its primary computing elements.[102] The resistors had values of 2, 4, 6, 8, 10 to 100 in steps of 10 ohm, 100 to 1000 in steps of 100 ohm, and 1000 to 6000 ohm in steps of 1000 ohm, the capacitors had capacitances of 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, and 2 μF, while the inductors featured inductances of 0.06 H up to 0.9 H in steps of 0.06 H with a remarkable high quality factor Q = 100.[103] The multipliers were of quite extraordinary construction, based on a double carrier modulation scheme followed by a detector circuit.[104]

The system was aptly named a *Direct Analogy Electrical Analog Computer* and resembled more of an intricate electric Tinkertoy kit than an abstract mathematical machine.[105]

→Figure 3.30 shows about half of the computing and control elements of the Anacom system located at Caltech. A particularly noteworthy device is the synchronous switch allowing for the introduction of transient conditions in a simulated system visible in the lower left of the picture. It is shown in detail in →figure 3.31 and was driven by a synchronous motor coupled by a gear box to a central rotating shaft on which rotating drums were mounted, which in turn were contacted by silver-graphite brushes. Early on, this switch proved to be rather unreliable and required a lot of maintenance. It was among the first modules of the system to be replaced by electronic switches controlled by a central timing sawtooth signal.[106] Further remarkable computing elements of Anacom were nonlinear resistors which basically resembled later diode function generators[107] and a *photoelectric arbitrary function device* (a *photoformer*).[108]

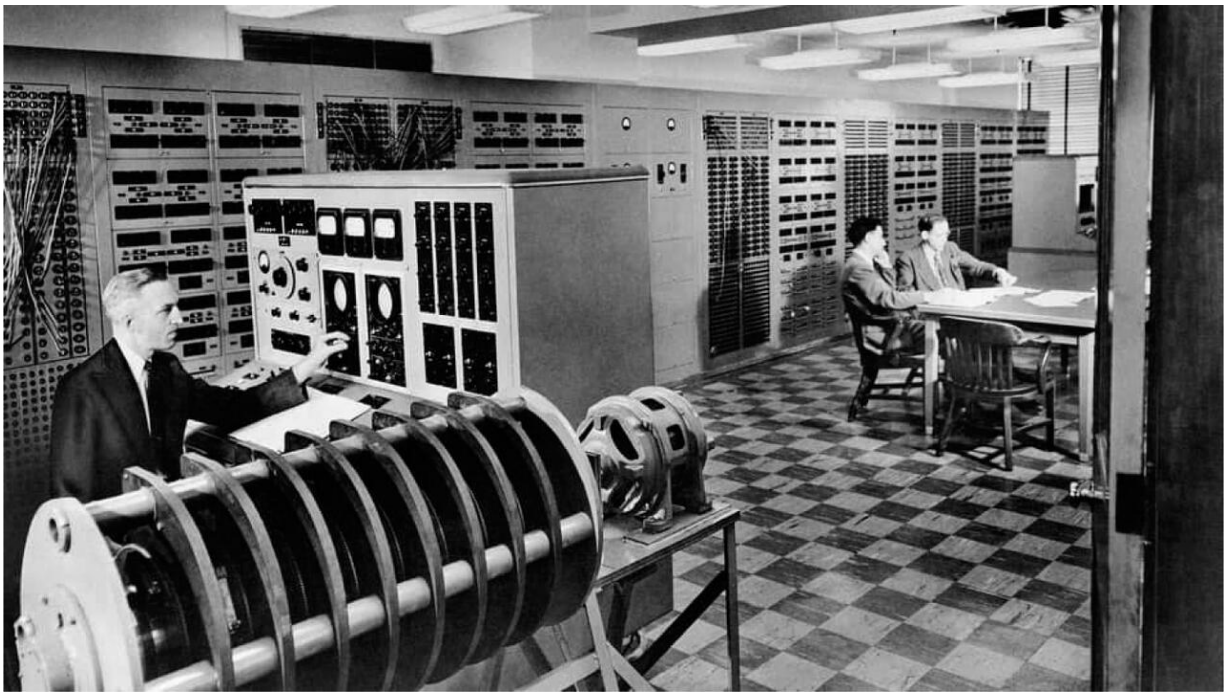


Figure 3.30. The Caltech Anacom

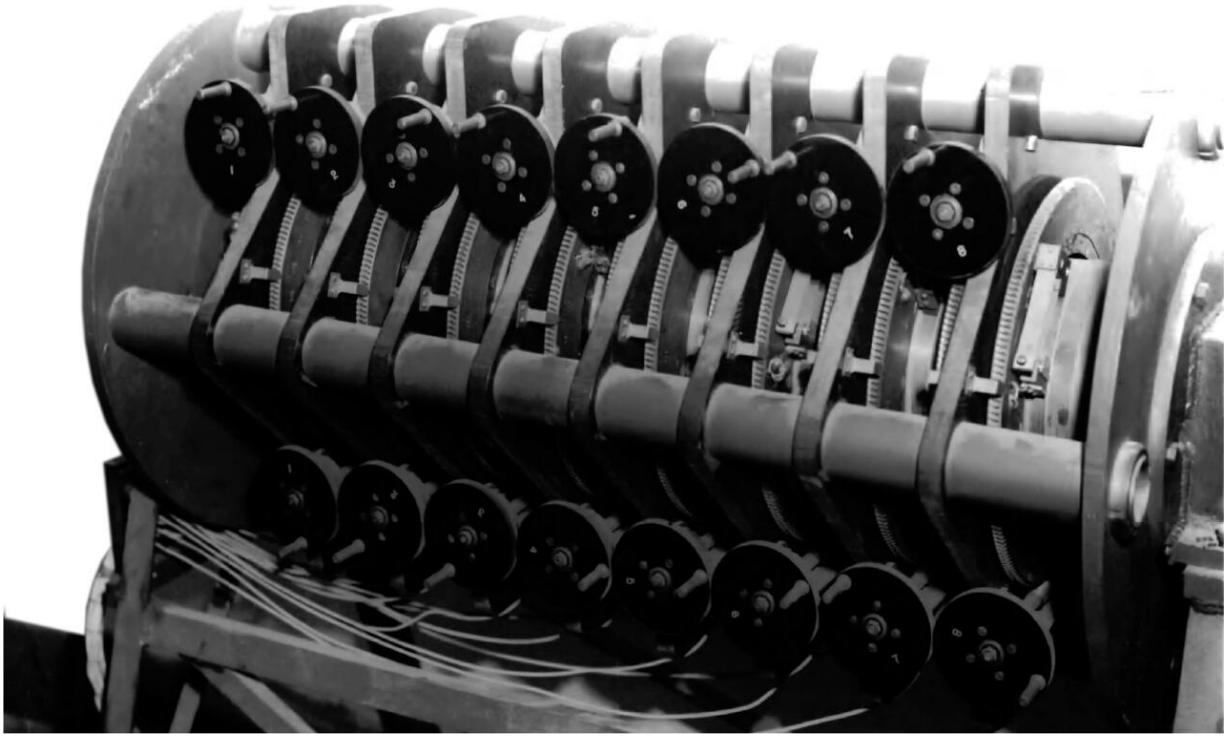


Figure 3.31. Initial version of the synchronous switch used in Anacom (see [→Oodrys 1953, p. 52])

An exhaustive error analysis of the overall Anacom system can be found in [→Dixon 1952]. Errors under consideration in this work are twofold: those caused by the non-ideal computer components and, more interestingly, errors resulting from solving problems involving continuously distributed physical properties such as bending cantilever beams, etc., by discrete elements in a lumped parameter representation. The overall system was reported to achieve an error of less than 2%.[109]

The Anacom systems were used to treat an incredibly wide range of problems such as oil flow problems, bearing lubrication problems, diffusion problems, nuclear reactor design,[110] regulating systems, aerodynamic equations of flight of a high-speed airplane, simultaneous equations, simulation of magnetic

amplifiers with and without self-saturation, corona effects on high voltages lines, and many more.[111]

However, probably the most interesting class of problems involved the effects of lightning strikes on power lines. The simulation of lightning is described in [→Harder 1952] while [→Oodrys 1953, pp. 74 ff.] gives a detailed account of a lightning protection problem in high-voltage stations. These types of problems proved to be computationally so demanding that Anacom was actively used for their solution until 1990.[112] [→Aspray et al. 1991] write:

> "*Utilities all over the country would use [Anacom] for the lightning and switching transient studies on their systems. [...] Westinghouse took on the contract to develop the first 500 kV system for Virginia Electric Power Company VEPCP, and then for Allegheny Systems [...]*"

→Figure 3.32 shows a typical setup of the Caltech analog computer. The rhombus shape represents a function generator,[113] the rectangular box below is a multiplier (with implicit sign inversion as denoted by the $-M$), while the small box on the bottom is a summer. The circular component on the left denotes the forcing function generator, which produces a time-dependent forcing function as an input to the computer setup. This setup solves the differential equation

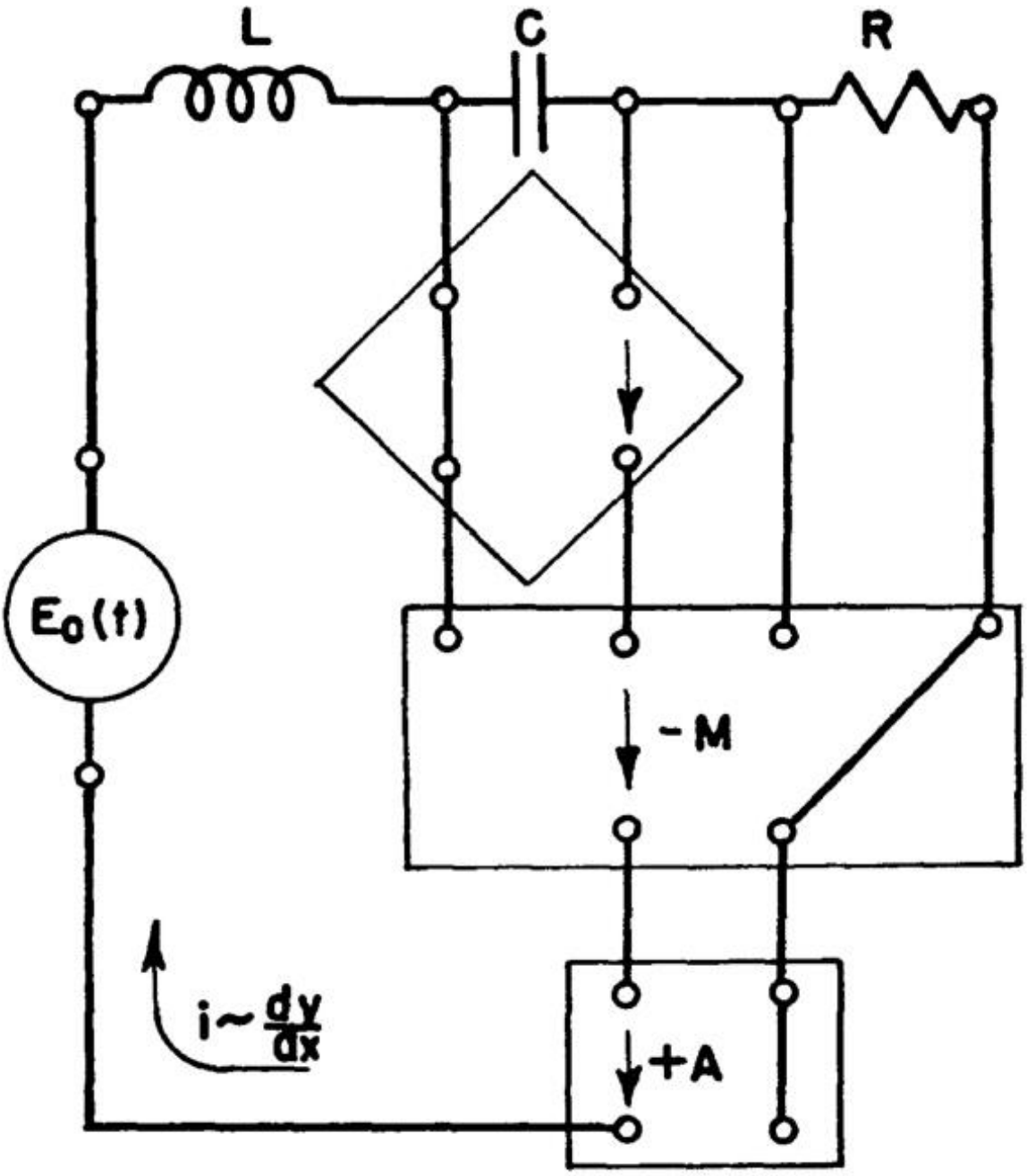


Figure 3.32. Setup of equation (→3.6) transformed into (→3.7) on the Caltech analog computer (see [→McCann 1949, p. 505])

$$F(x) = k_1 \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + f(y) \frac{\mathrm{d}y}{\mathrm{d}x} + k_2 y \tag{3.6}$$

by implementing its electric equivalent

$$E_{0}(t) = L_{\ddot{q}} + \tfrac{\mathrm{AMR}}{C} f_{1}(q)\dot{q} + \tfrac{q}{G}\,. \tag{3.7}$$

It was also possible to solve partial differential equations on the Caltech analog computer by discretizing space (or time) along one or more axes resulting in a finite difference mesh-like network of (mostly) passive components. Due to the limited number of such passive computing components these discretizations were pretty coarse but nevertheless useful for a variety of problems. →Figure 3.33 shows a typical setup of this machine for a partial differential equation. The system was successfully applied to 4$^{\mathrm{th}}$ order partial differential equations as well as to systems of simultaneous partial differential equations.

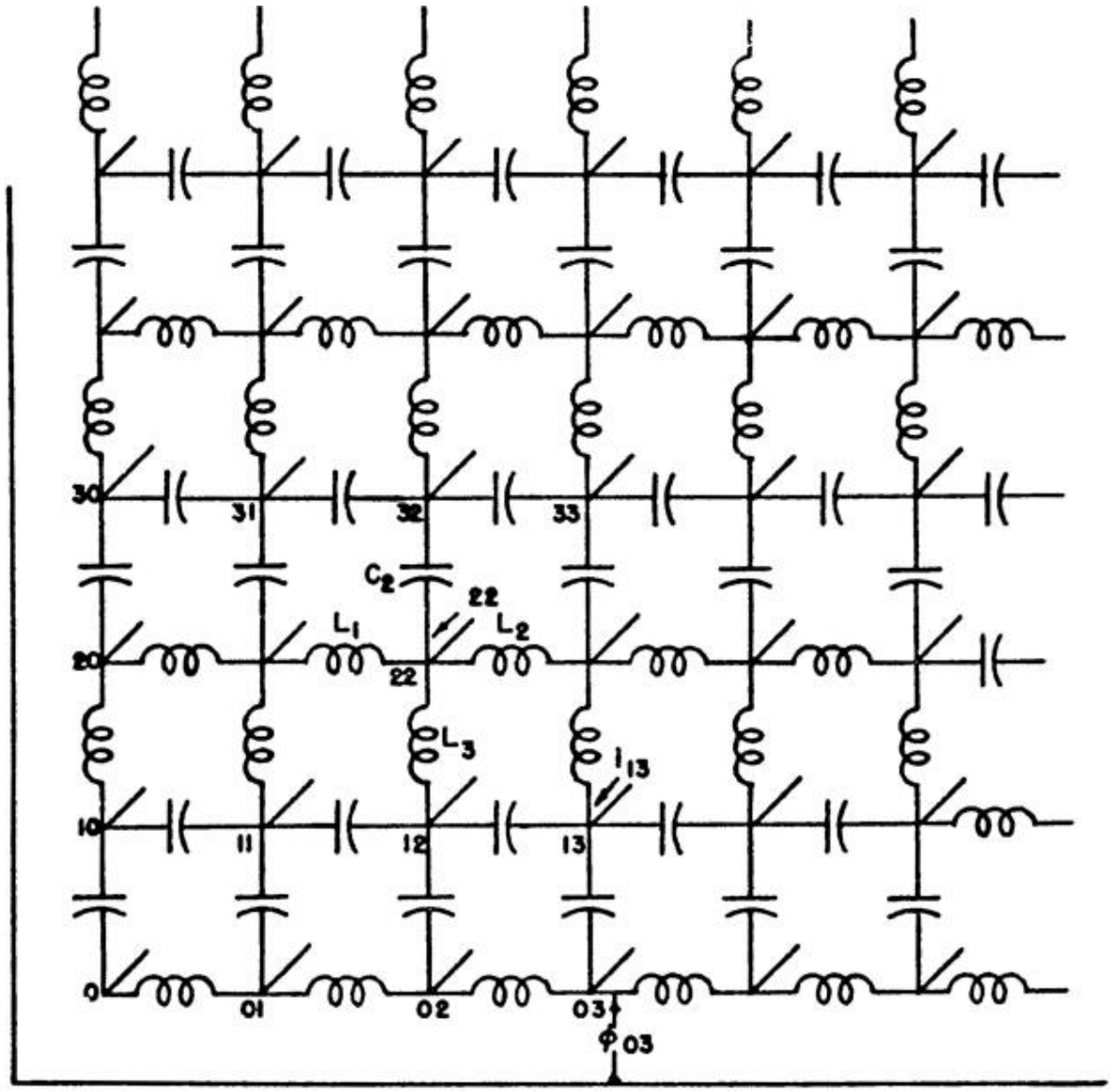


Figure 3.33. Setup of the Caltech analog computer for solving the finite difference equivalent of a partial differential equation (see [→McCann 1949, p. 507])

Another interesting example is shown in →figure 3.34 where an Anacom model of half of a model airplane wing consisting of ten coupled cells is depicted. This was part of a study on bending moments acting on airplane wings.[114]

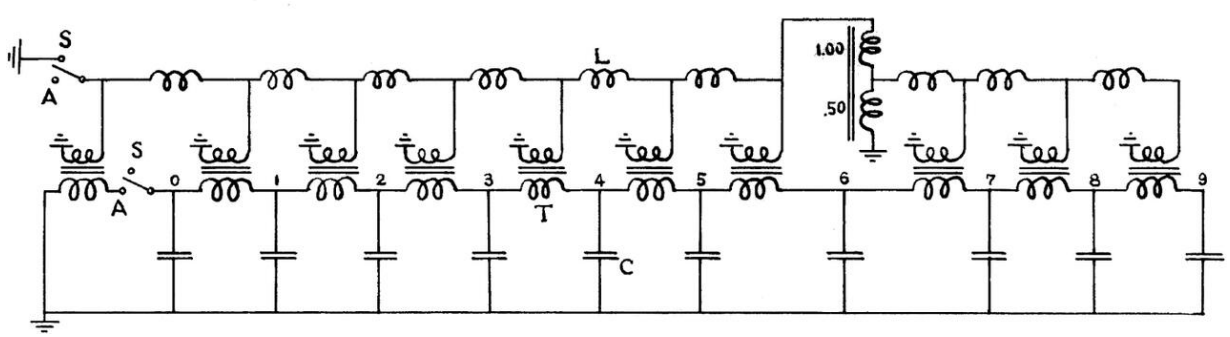


Figure 3.34. Ten-cell circuit representing half of a model airplane wing (see [→ Dixon 1952, p. 53].)

It is interesting to note that the high degree of interactivity typical of analog computers was put into use and mentioned as early as in 1949. [→ McCann 1949, p. 507] notes:

> "*Since circuit element settings and inserted mesh currents can be changed rapidly, iterative methods of solution can be set up in a very short time*"

The wealth of applications of this remarkable analog computer is described in [→ McCann 1949] and [→ McCann et al. 1949]. Among these were servomechanisms and automatic control systems, missile guidance, and autopilot related problems, the analysis of mechanical vibrations, the solution of linear and nonlinear ordinary differential equations, partial differential equations describing the behavior of elastic beams and plates, and many more.

Richard H. MacNeal especially remembers the work on beams as follows:[115]

> "*[...] McCann [...] along with Harry E. Criner, had worked out how to model beams with electric analog computers, using transformers as a primary agent. For vibrations, you would use inductors for stiffness and capacitors for mass. [...] In those days, aircraft were analyzed as beams. There was one beam for the fuselage, one beam for the left wing [...] One of the most serious*

problems in aircraft structural analysis is the flutter problem, which is the instability issue. We got to the point where we could do that fairly well."

Following this brief description of a few of the earliest electronic analog computers the following chapter will now focus on the basic computing elements employed in analog electronic analog computers past, present, and future alike.

4 Basic computing elements

The following chapters give an overview of the most important computing elements found in a typical analog electronic analog computer. Some functions, such as multiplication, can be implemented in very different ways. Although often only one implementation variant prevails, other approaches will be also presented to show the wide range of possible implementations.[1]

An analog computer typically relies on voltages[2] for representing and transmitting values between the various computing elements, which constitute a computer setup for a particular problem. These voltages can only take on values within a certain interval. For most vacuum tube based analog computers this interval ranges from −100 V to +100 V while most (but not all) transistorized analog computers and later machines used values in the range of −10 … 10 V.[3] To simplify things it is common to think of variables occurring in an analog computer setup to be in the interval [−1, 1] regardless of the actual voltages used in a particular machine. The bounds of this interval are called *machine units* and correspond to the actual minimum and maximum voltages for any given machine. Thus, a variable represented by +50 V on a ±100 V machine is said to have the value 0.5. The same value would describe a variable represented by 5 V on a ±10 V transistorized system. All calculations are thus with respect to these machine units of ±1 and never with respect to the actual range of voltages used by a particular analog computer.

All computing elements described in the following sections will work on normalized values in the range of ±1 machine unit.

4.1 Operational amplifiers

The central element of every electronic analog computer is the *operational amplifier*, which forms the heart of most computing circuits such as summers, integrators, etc. This central role of the operational amplifier led to the following remark by [→Truitt et al. 1960, p. 2-58]:

> "*The Operational Amplifier is the King of Analog Computing Components*".

It was clear from the beginning of the development of electronic analog computers that high gain amplifiers capable of working on DC voltages with negligible error voltages caused by drift effects and the like were required for usable general purpose machines. Hoelzer, Philbrick, Macnee, and all of the other pioneers were plagued by the deficiencies of their respective amplifiers. Hoelzer overcame these problems by resorting to AC coupled amplifiers operating on modulated AC signals, which required modulator and demodulator circuits, while Macnee solved this problem by building a machine that computed at such high speed that he could use drift free AC amplifiers in his machine.

It was not until 1949 that a practical drift correction technique was developed[4] by Edwin A. Goldberg. This made the implementation of high precision operational amplifiers with extremely high gain and very low drift possible, which led to a revolution in the development of analog electronic analog computers.[5]

The term *operational amplifier* was coined by John Ragazzini[6] in 1947 and defined as follows:[7]

> "*As an amplifier so connected can perform the mathematical operations of arithmetic and calculus on the voltages applied to its input, it is hereafter termed an 'Operational Amplifier'.*"

→Figure 4.1 shows the symbol used today to denote an operational amplifier in a schematic. Typically such an amplifier has two inputs, one inverting and one non-inverting, and one output that represents the amplified difference between the values applied to the inputs. In

most analog computers, the non-inverting input is not available for use in a computer setup and is either connected to ground – representing the value zero – or used for drift compensation. Thus, differing from the standard symbol used in today's electronics, an *open* operational amplifier, i. e., an amplifier that has no elements connecting its output to its input, is normally denoted by the symbol shown in →figure 4.2 in an analog computer setup diagram. This computing element always inverts the sign of its input signal.

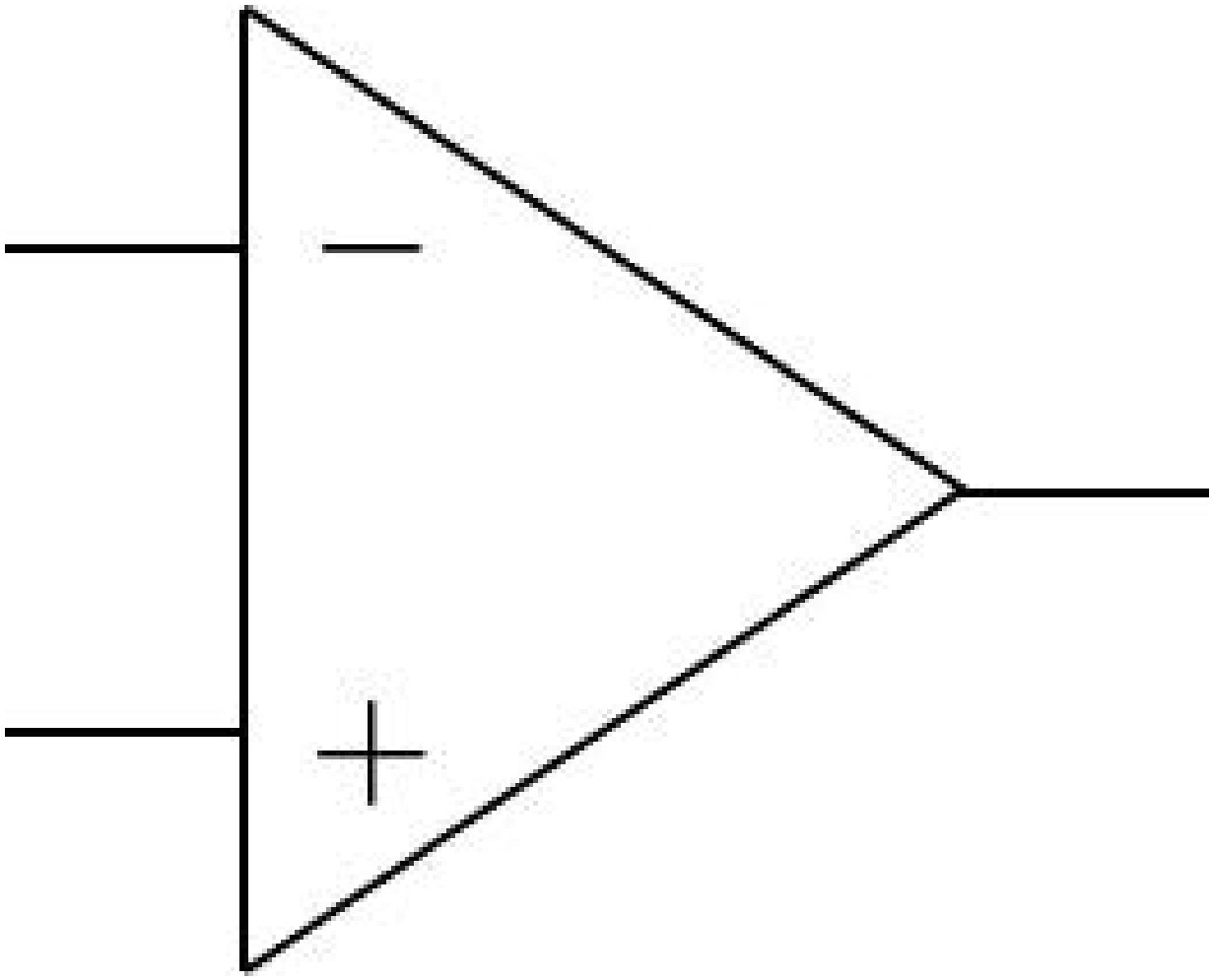


Figure 4.1. Symbol of an operational amplifier

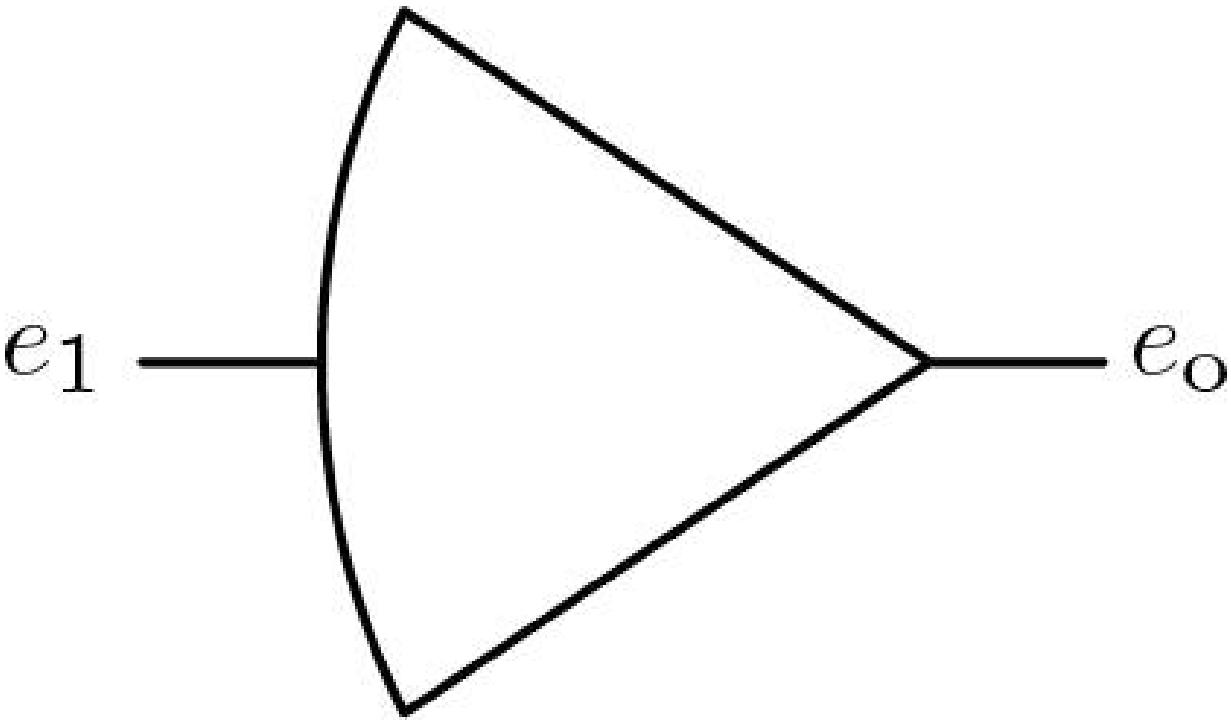


Figure 4.2. Symbol of an *open* or *uncommitted* amplifier

In the early 20th century amplifiers were normally operated in a rather straightforward mode of operation since they did not use any kind of feedback to stabilize their behavior. Thus, these amplifiers were quite unstable and difficult to use since even the slightest changes regarding their components resulted in massively different behavior of the overall amplifier. Then, in 1927, Harold Stephen Black[8] invented the *negative-feedback amplifier* that should become one of the most influential basic circuits in electronics.[9] He was elected as a member of the "Electrical Engineering Hall of Fame" for his seminal work on amplifiers (see [→ Britain 2011]). The idea of negative-feedback was so novel and unusual that it took nine years from Black's patent application to its issuance (see [→ Jung 2006, p. 767]).[10]

At the time, Black was working for the Bell telephone laboratories and his negative-feedback scheme that is taken for granted today allowed the Bell telephone system to successfully implement carrier telephony, which was until then hampered by the poor stability and high distortion of existing amplifiers without

negative feedback; these were prone to errors caused by varying supply voltages, aging of circuit elements (especially the tubes), etc.

→Figure 4.3 shows the basic concept of BLACK's invention: An amplifier consisting of an odd number of inverting stages (resulting in an inverting amplifier) is fed with two signals that are applied to its inverting input: The desired signal *E* itself, which is connected via a series resistor *C* and a feedback signal originating from the output of the amplifier. This signal is connected to the input of the amplifier be means of a feedback resistor *f*. →Figure 4.4 shows this principle using today's standard symbols.

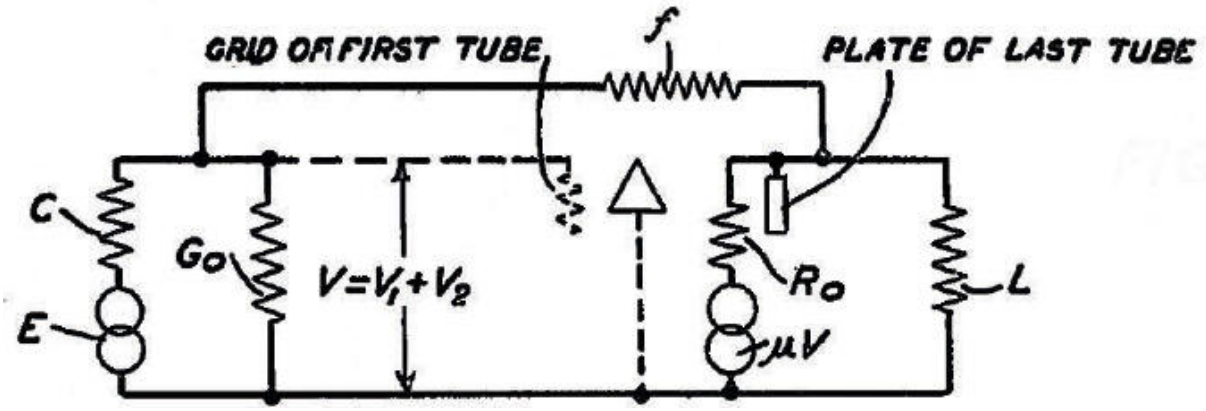


Figure 4.3. The basic principle of negative-feedback according to [→BLACK 1937]

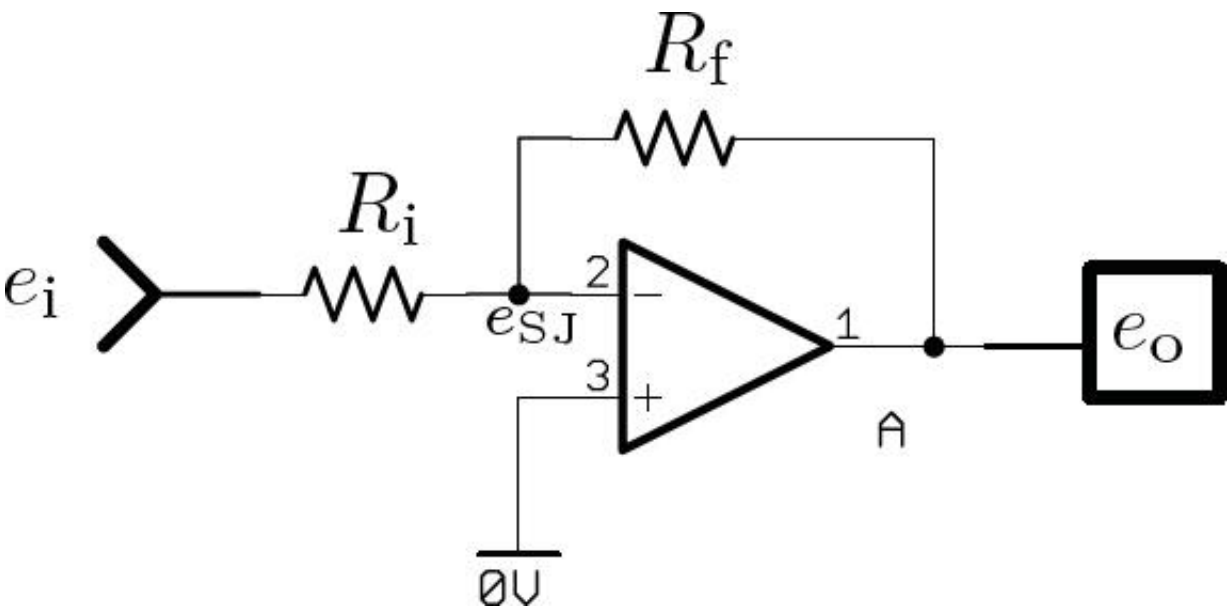


Figure 4.4. Operational amplifier with negative feedback

The idea of feeding back some amount of the output signal of an inverting amplifier to its input allows to trade gain for stability as the following consideration shows: With *A* denoting the gain of the amplifier and e_SJ representing the voltage at the summing junction at its inverting input, as in →figure 4.4, its output voltage is

$$e_o = -Ae_{SJ}$$

yielding

$$e_\mathrm{SJ} = -\tfrac{e_o}{A}. \tag{4.1}$$

The currents at the summing junction are

$$\begin{aligned}
i_i &= \tfrac{e_i - e_\mathrm{SJ}}{R_\mathrm{i}} \quad (\text{input current through } R_\mathrm{i}),\\
i_\mathrm{f} &= \tfrac{e_o - e_\mathrm{SJ}}{R_\mathrm{f}} \quad (\text{feedback current through } R_\mathrm{f}), \text{ and}\\
i_- &\approx 10^{-9}\mathrm{A} \quad (\text{current flowing in the inverting input}).
\end{aligned}$$

Applying KIRCHHOFF's[11] law yields

$$i_- = i_\mathrm{i} + i_\mathrm{f} = \frac{e_\mathrm{i} i - e_\mathrm{SJ}}{R_\mathrm{i}} + i\frac{e_o - e_\mathrm{SJ}}{R_\mathrm{f}}\,. \tag{4.2}$$

Since i_- is negligibly small, a few nA[12] at most, equation (→4.2) can be rearranged

$$\frac{e_\mathrm{i} - e_\mathrm{SJ}}{R_\mathrm{i}} = -\frac{e_\mathrm{o} - e_\mathrm{SJ}}{R_\mathrm{f}}\,.$$

Replacing e_SJ according to (→4.1) yields

$$\frac{e_\mathrm{i} + \frac{e_\mathrm{o}}{A}}{R_\mathrm{i}} = -\frac{e_\mathrm{o} + \frac{e_\mathrm{o}}{A}}{R_\mathrm{f}}\,.$$

Rearranging this equation yields

$$e_\mathrm{o}\left(\frac{1}{AR_\mathrm{i}} + \frac{1}{R_\mathrm{f}} + \frac{1}{AR_\mathrm{f}}\right) = -\frac{e_\mathrm{i}}{R_\mathrm{i}}$$

and thus

$$e_\mathrm{o} = \frac{-\frac{e_\mathrm{i}}{R_\mathrm{i}}}{\frac{R_\mathrm{f} + AR_\mathrm{i} + R_\mathrm{i}}{AR_\mathrm{i}R_\mathrm{f}}} = \frac{-\frac{e_\mathrm{i}}{R_\mathrm{i}} AR_\mathrm{i}R_\mathrm{f}}{R_\mathrm{f} + AR_\mathrm{i} + R_\mathrm{i}}\,.$$

Dividing by R_i and A and rearranging finally yields

$$e_\mathrm{o} = \frac{-\frac{R_\mathrm{f}}{R_\mathrm{i}} e_\mathrm{i}}{1 + \frac{1}{A}\left(\frac{R_\mathrm{f}}{R_\mathrm{i}} + 1\right)} \tag{4.3}$$

Since the gain A is usually very large – some amplifiers used in analog computers reach values up to $A = 10^8 \ldots 10^9$ – and since the ratio $R_\mathrm{f}/R_\mathrm{i}$ is normally quite small, rarely exceeding 10, equation (→4.3) can be further simplified to

$$e_\mathrm{o} = \frac{R_\mathrm{f}}{R_\mathrm{i}} e_\mathrm{i}, \tag{4.4}$$

so the behavior of a high gain amplifier with negligible input current at its inverting input is completely determined by the circuit

elements in the feedback loop and at its input.

4.1.1 Early operational amplifiers

This section focuses on some early and influential operational amplifier designs. Historic references that might be of interest in this context are [→GRAY 1948] and [→CHANCE et al. 1947].[13]

An early dual channel operational amplifier is depicted in →figures 4.5 and →4.6. This amplifier was used in the ground based analog guidance computer of a *Nike* surface to air missile system.[14] The first version of this missile system was deployed from 1954 on well into the early 1960s. This amplifier still shows its roots, which can be traced back to the M-9 fire control computer. In fact, the amplifiers used for Nike were identical to those used in the *M-33* fire control computer, a successor of the M-9.[15]



Figure 4.5. Top view of a typical dual operational amplifier used in the Nike computer



Figure 4.6. Bottom view of the same amplifier

The schematic diagram for a single channel of this operational amplifier is shown in →figure 4.7. It features a now classic differential input stage consisting of both halves of a dual triode. The interstage-coupling is done via the common cathode resistor R1. The non-inverting input is explicitly used as an offset input for (automatic) drift compensation in this application. The following stages are no longer differential and consist of a pentode section and a simple triode output stage for a total of three amplification stages. Thus, the overall amplifier is an inverting amplifier.

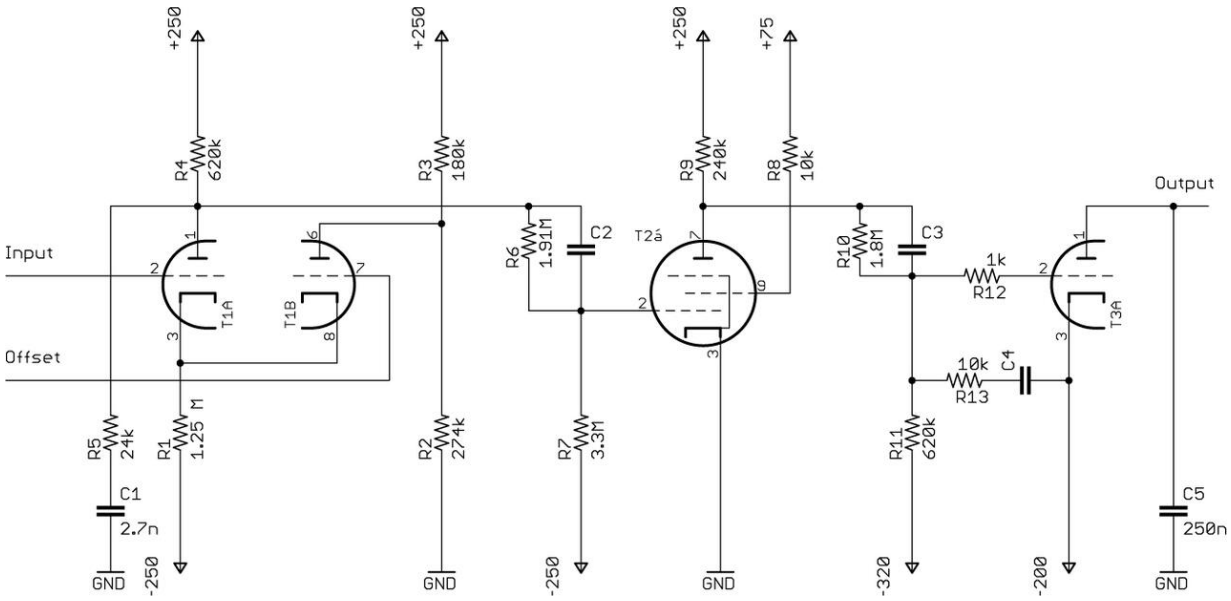


Figure 4.7. Operational amplifier used in the Nike computer

Since the output stage is not of the push-pull type, an external pull-up resistor is necessary. This resistor would be connected between the output triode's anode and the +320 V supply. All in all this amplifier needed five different supply voltages of −320 V, ±250 V, −200 V, and +75 V and has an output voltage range of ±100 V, although the full range can only be maintained up to about 100 Hz,[16] which is not that impressive but more than enough to steer a surface to air missile.

A far simpler yet commercially very successful operational amplifier is shown on the left-hand side of →figure 4.8: The famous *K2-W* amplifier developed by GAP/R, the company GEORGE A. PHILBRICK founded after his early process simulator work at Foxboro.[17] The K2-W amplifier only requires two supply voltages of ±300 V and has an open-loop gain of about 15,000 with an output voltage range of ±50 V at a maximum output load of 1 mA. Rise time is about 2 *µ*s and the input impedance is roughly 100 MΩ resulting in an input current of only about 10 nA.[18]



Figure 4.8. K2-W and K2-X operational amplifiers (GAP/R)

→Figure 4.9 shows the schematic of this pluggable operational amplifier. It is based on two 12AX7 dual triodes and had its market debut in 1953. The first 12AX7 acts as differential input stage with a common cathode resistor.[19]. This input stage is followed by a single triode inverter and a cathode follower driving the output, which is tied by three parallel pull-down resistors to the negative supply voltage. An interesting detail is the neon light bulb in series with a 680 kΩ resistor. This circuit acts as a level shifter.[20]

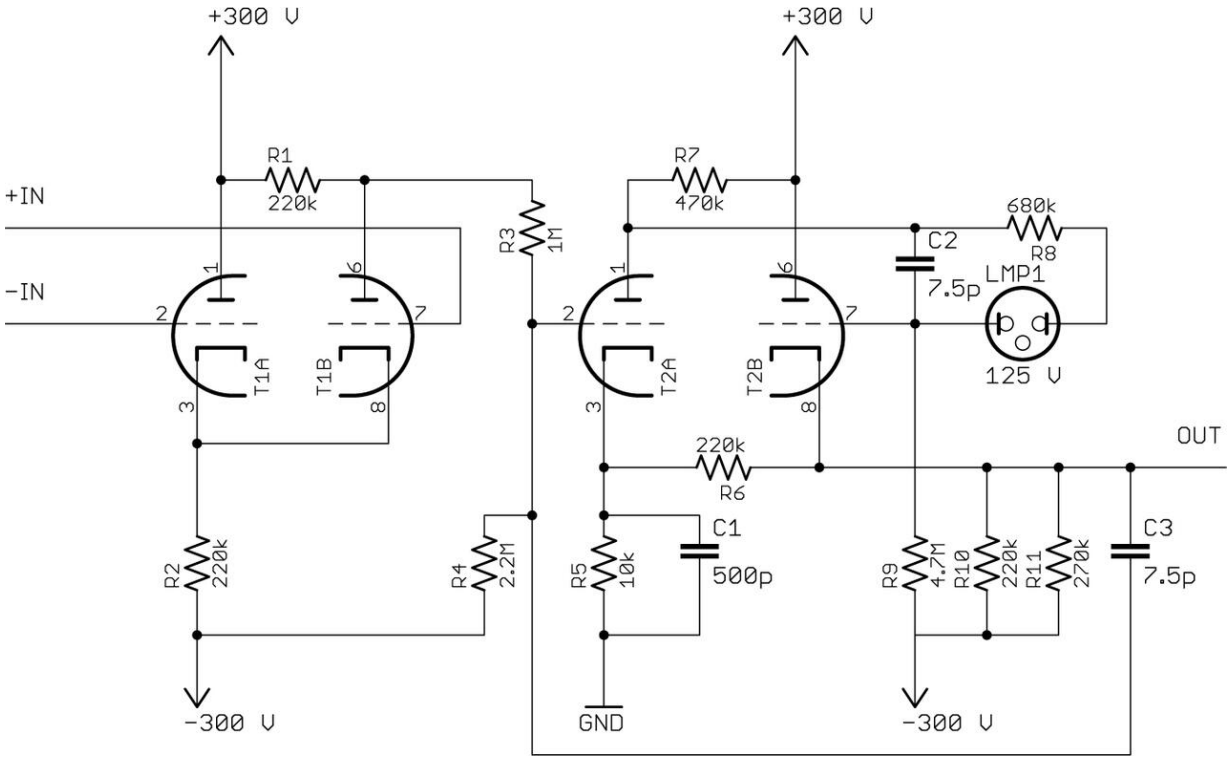


Figure 4.9. Schematic of the K2-W operational amplifier (cf. [→GAP/R K2W])

→Figure 4.8 shows on the right an improved version of the K2-W, the *K2-X* operational amplifier. This device has a significantly faster output signal rise time than the K2-W and features an output voltage swing of ±100 V at 3 mA.[21]

4.1.2 Drift stabilization

All of these early DC coupled operational amplifiers were plagued by drift, i. e., small error signals that propagate through the various amplifier stages and show up with a reasonable amplitude at the output, causing significant errors in computations. Recalling →figure 4.4 and taking into account that the input current even of early vacuum tube based operational amplifiers is negligibly small, it is clear that the voltage e_{SJ} at the summing junction would be zero in the case of a theoretically drift-free amplifier since the input current

i_i flowing through R_i and the feedback current i_f through R_f cancel each other.

Any drift caused by intrinsic effects of the amplifier results in an output signal different from 0 V even if $e_i = 0$. Thus, all drift effects in an amplifier can be modeled by a drift voltage e_d acting on the summing junction of the amplifier via a large input resistor. If one could inject a commensurate voltage with reverse sign at the non-inverting input of the amplifier, it would cancel out the drift error.

This is the basic idea Edwin A. Goldberg describes in his 1949 patent.[22] If one could measure the error voltage due to drift, e_d, at the summing junction and amplify it without introducing further errors with a drift-free amplifier, the resulting output signal could be used to cancel out the drift of the main amplifier. →Figure 4.10 shows the setup proposed by Goldberg.

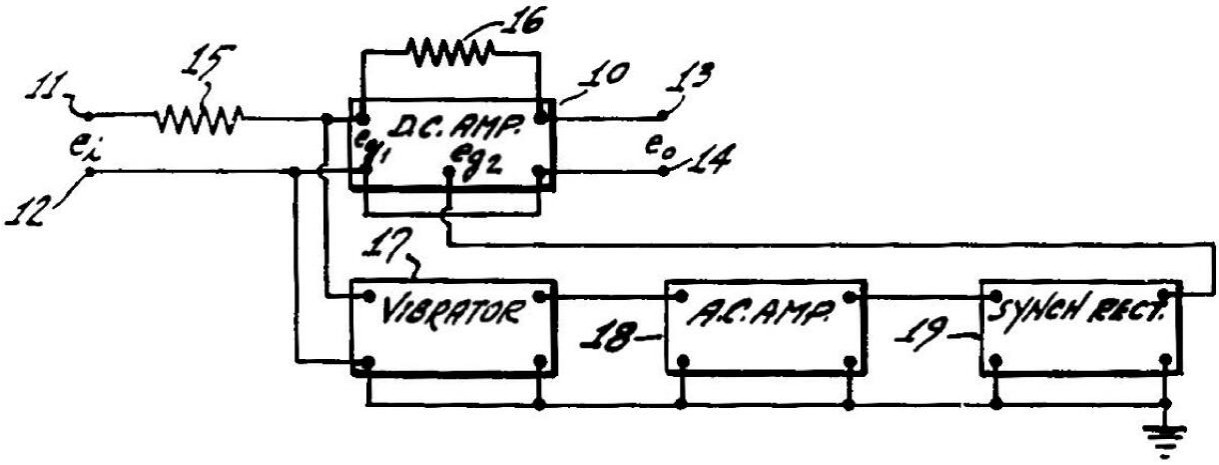


Figure 4.10. Drift stabilization scheme according to Edwin A. Goldberg and Jules Lehmann (cf. [→Goldberg et al. 1954])

The DC amplifier shown in the upper half of the picture is the DC coupled main amplifier whose drift is to be eliminated. To achieve this, the error voltage at the input of this amplifier, the summing junction, is converted to an AC voltage by means of the electromechanical chopper relay[23] in →figure 4.10.

The output signal of this chopper[24] is of rectangular shape swinging between 0 V and the error voltage e_d picked up from the summing junction of the main amplifier. This AC voltage can now be amplified with next to no additional drift error by a separate AC coupled amplifier. The output signal of this amplifier is then rectified (typically by means of a synchronous demodulator driven by the same voltage feeding the chopper) yielding a DC signal that can then be used to compensate for the drift of the main DC amplifier.[25]

This ingenious scheme was quickly put into use in nearly all operational amplifiers used for analog computers. Apart from the automatic drift compensation this *chopper-stabilization* had a second advantage: For low frequency input signals the gains of the main DC amplifier and the stabilizing AC amplifiers essentially add up, yielding a very high gain overall amplifier. As equation (→4.3) shows, a high open loop gain[26] is essential for keeping the errors of a circuit with a negative feedback path low. The amplifier developed by GOLDBERG had an impressive overall DC gain of 150, 000, 000 corresponding to roughly 163 dB.[27] Later developments achieved even higher gains for low frequency input signals.

Electromechanical choppers were used well into the 1970s to stabilize precision operational amplifiers because they exhibit an extremely low resistance when the contacts are closed and nearly infinite resistance in the open state. This came at a price: Most of the commercially available devices had a short life-span. In addition to this, the contact pairs often acted as parasitic thermocouples, thus introducing another error in the drift-compensation circuit. The second problem was finally overcome by using arcane metal compositions for the contact pairs, such as Pt/Rh.

→Figure 4.11 shows a typical 400 Hz chopper made by the German manufacturer *KACO* for use in Telefunken analog computers. The 400 Hz excitation voltage is supplied through a top connector (not visible here) to minimize injection of noise into the amplifier circuit. The top half of the chopper contains the coil, which excites the middle contact to oscillate between the two contacts on

the left and right at the bottom of the relay. The chopper system itself is suspended from a mounting frame by a rubber band, thus minimizing vibration of the overall enclosure.

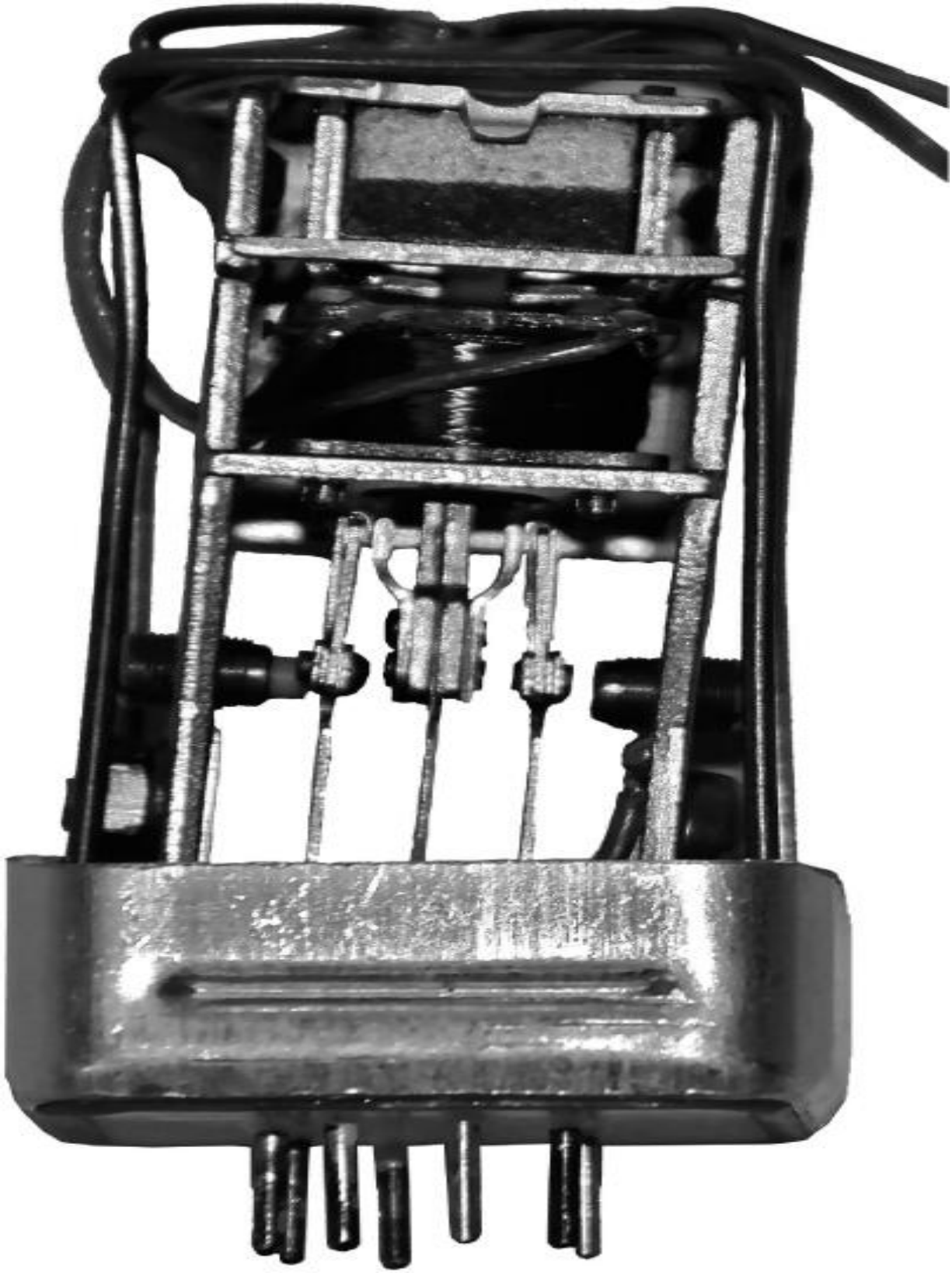


Figure 4.11. Typical electromechanical chopper

Some analog computers used a single AC coupled amplifier for drift stabilization of a large number of operational amplifiers by employing a round robin mode of operation. This reduced cost substantially as the complexity of a single operational amplifier was basically cut in half but required a motor driven rotary switch to connect the single stabilization amplifier to each operational amplifier in sequence. The downside of this approach is that it sacrifices the extremely high gain of a typical chopper-stabilized amplifier. In addition to this, the recovery time from an overload condition is substantially increased.

An example for this technique can be found in the analog computer used in the ground based guidance system for Nike missiles. Here a rotating switch connected one AC coupled amplifier with its associated input chopper and output rectifier sequentially to a number of amplifiers. This required an additional capacitor at each of the operational amplifier inputs used for drift compensation to store the signal generated by the AC amplifier for the duration of one revolution of the rotary switch.

A classic dual drift stabilized operational amplifier is the *Electronic Associates, Inc*. (*EAI*) *16-6H* used in the *PACE*[28] *16-31R* analog computer dating back to the 1950s. It follows the Goldberg approach quite closely as the schematic in →figure 4.12 shows. The circuit in the upper half is a DC coupled operational amplifier with a differential input stage consisting of a dual triode type 5691. This is followed by two rather similar pentode amplifier stages feeding the beam power tube 5881 in the output stage. The bottom half shows the AC coupled amplifier for drift stabilization. Its input is coupled to the inverting input of the DC amplifier by means of a 470k resistor. This signal is then converted to AC by means of the Leeds & Northrup 3338-9 vibrator and is AC coupled to the first 6SJ7 tube. It should be noted that the vibrator is driven by a rather unusual 94 Hz signal generated by a *vibrator drive unit* such as the EAI 21.004. The reason for this might be to avoid the amplification of hum which would be 50 Hz or 60 Hz or a harmonic of these usual power grid frequencies.

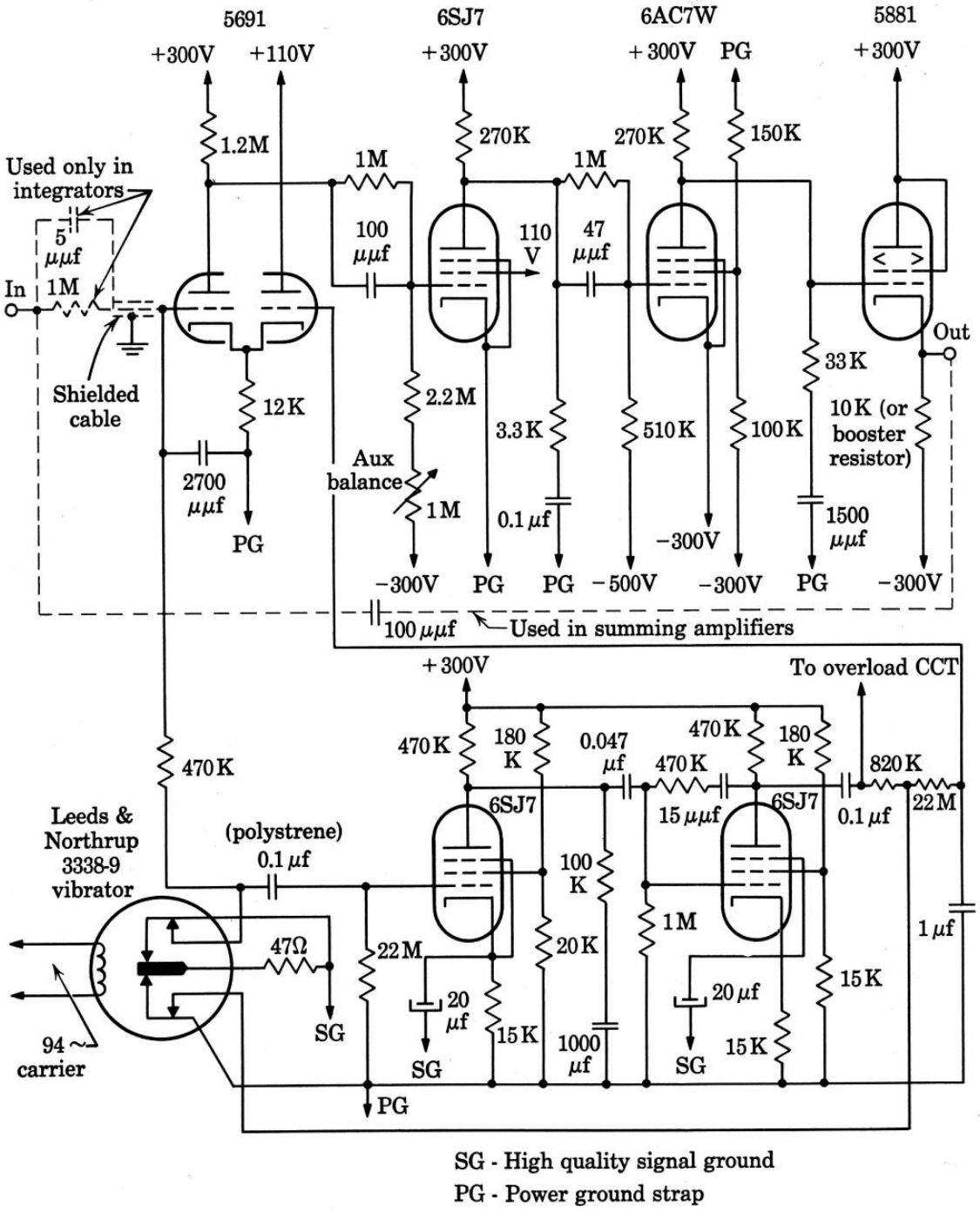


Figure 4.12. EAI 16-6H dual chopper-stabilized operational amplifier (see [→HOWE 1961, p. 114])

The second 6SJ7 is also AC coupled to the plate of the first stage. The output signal is synchronously demodulated by a second set of contacts in the vibrator and smoothed by an RC combination consisting of a 22M resistor and a 1μF capacitor. This output signal is then fed into the non-inverting input of the DC amplifier, canceling out any deviation of its inverting input from zero. It should be noted that EAI analog computers typically featured an intricate grounding system with a *power ground* (*PG*) and a *high quality signal ground* (*SG*).[29] → Figure 4.13 shows the actual implementation of this dual amplifier. Its size and construction are quite remarkable as it is much bigger and more sturdy than other implementations of roughly the same time frame.



Figure 4.13. Implementation of the 16-6H dual operational amplifier

The overall DC gain of this amplifier is $> 2 \cdot 10^5$, the gain of the stabilizer is $> 3 \cdot 10^3$ resulting in an overall DC gain of the amplifier of

$> 6 \cdot 10^8$, a pretty remarkable value.[30] The overall drift is about 75μV per day.[31]

→Figure 4.14 shows another, later vacuum tube operational amplifier with chopper-stabilization. The chopper is the gray rectangle visible in the upper right part of the picture. This particular amplifier is used in a Solartron *Minispace* analog computer.



Figure 4.14. Chopper-stabilized operational amplifier (Solartron)

The schematic of the first fully transistorized chopper-stabilized operational amplifier developed at Telefunken in 1959 is shown in →figure 4.15. Its main developers were Hans Otto Goldmann and Günter Meyer-Brötz.

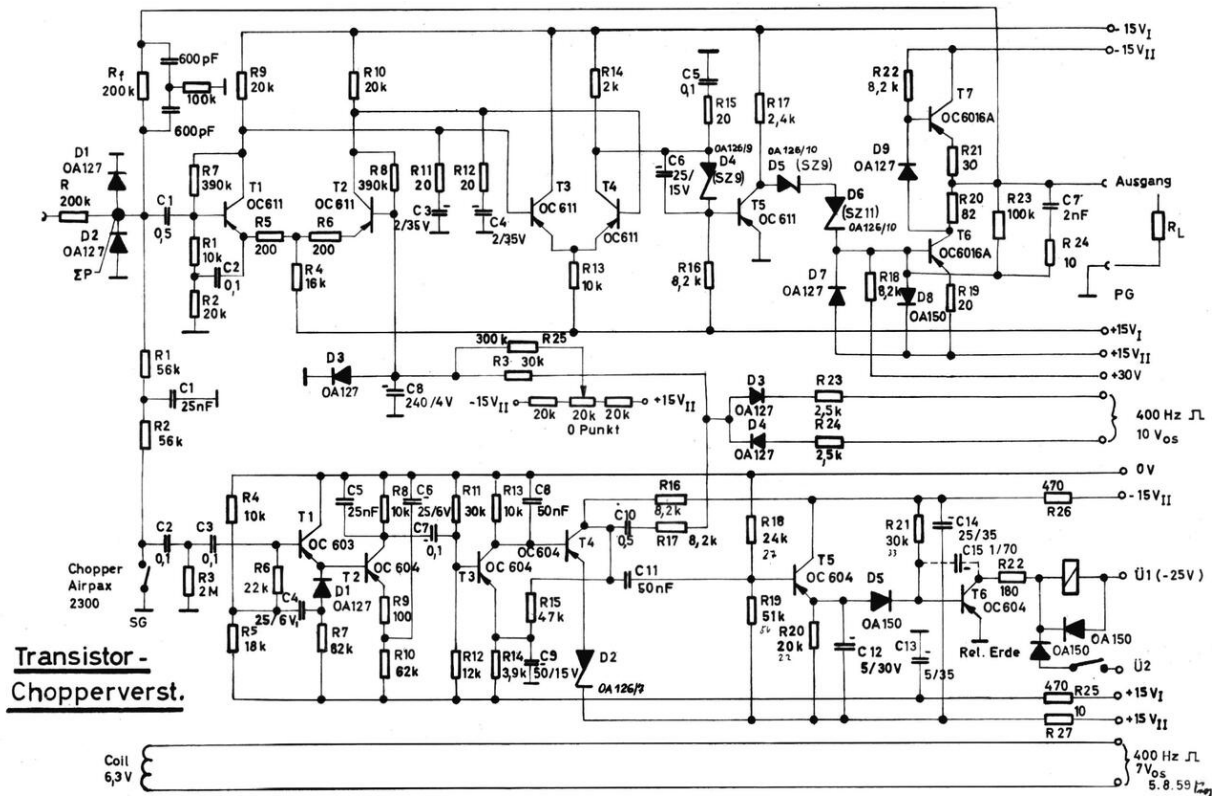


Figure 4.15. Schematic of the first chopper-stabilized transistorized operational amplifier developed at Telefunken in 1959

It is worthwhile to take a closer look at this remarkable circuit. The main amplifier shown in the upper half of the schematic has two differential amplifier stages in series. The first stage, T1 and T2, is the input stage with the base of T1 – this is the inverting input of the operational amplifier – connected to the summing junction by C1. The two anti-parallel diodes D1 and D2 limit the voltage at this point. It should be noted that this main amplifier is AC coupled to the summing junction, so a pure DC signal cannot pass. The base of T2 – the non-inverting input of the amplifier – is fed by the AC coupled amplifier shown in the lower half of the schematic.

T1 and T2 drive a second differential amplifier stage consisting of T3 and T4, which then feeds T5 and finally the differential output stage with the power transistors T6 and T7 (mounted on a heat sink).

The amplifier depicted in this schematic is wired as an inverter, i. e., a summer with a single input. R_f = 200k together with the high-pass T-filter consisting of two 600 pF capacitors and a 100k resistor connected to ground is the feedback path, while the 200k resistor connected to the summing junction is an input with weight 1.

The summing junction is also connected to the AC coupled amplifier shown in the lower half of →figure 4.15 by means of the low-pass T-filter consisting of R1, R2, and C1 and the Airpax chopper. The chopper relay is driven by a 400 Hz square wave signal and repetitively shorts the output of the low-pass to ground, thus creating an AC signal, which is filtered by another high-pass T-filter (C2, C3, R3). This signal is then amplified by the AC coupled amplifier built around the transistors T1, T2, T3, and T4.

The output AC signal at R17 is then rectified by the synchronous rectifier consisting of D3, R23, D4, and R24. This partial circuit is also driven by a 400 Hz square wave signal, which is in a fixed phase relationship to the chopper excitation voltage. A static drift correction voltage can be set by means of the potentiometer labeled “0 Punkt” (zero setting), which is added to the output signal of the AC coupled amplifier. The resulting signal is then fed to the non-inverting input of the main amplifier.

A signal entering this combination of two amplifiers thus takes two paths: Its DC and very low frequency parts are mainly amplified by the AC coupled amplifier and then enter the main amplifier through its non-inverting input, while higher-frequency components are amplified by the DC coupled main amplifier only. Thus, for low frequency and DC signals the overall open loop gain is the product of the gains of both amplifiers.

The output of the AC coupled amplifier is also fed into the circuit consisting of T5, D5, and T6 with their associated passive components. This controls the relay coil shown on the far right. In case of an overload, i. e., the summing junction can no longer be held at 0 V by the DC/AC amplifier combination and the relay will be activated and will signal the overload condition to the control circuits of the analog computer.

The production version of this amplifier (which differs only slightly from this schematic) is shown in →figure 4.16. It was used for more than 10 years in the various fully transistorized Telefunken analog computers. The circuit card on the left side contains the DC coupled main amplifier with its two large germanium output transistors mounted on a cooling plate visible on the top. The card on the right is the drift stabilizing AC coupled amplifier fed with the chopped error signal obtained from the summing junction of the main amplifier and provides the drift correction signal. Not shown is the chopper relay. The overall combined amplifier has a DC gain of up to 10^9 for DC to low frequency signals.[32]

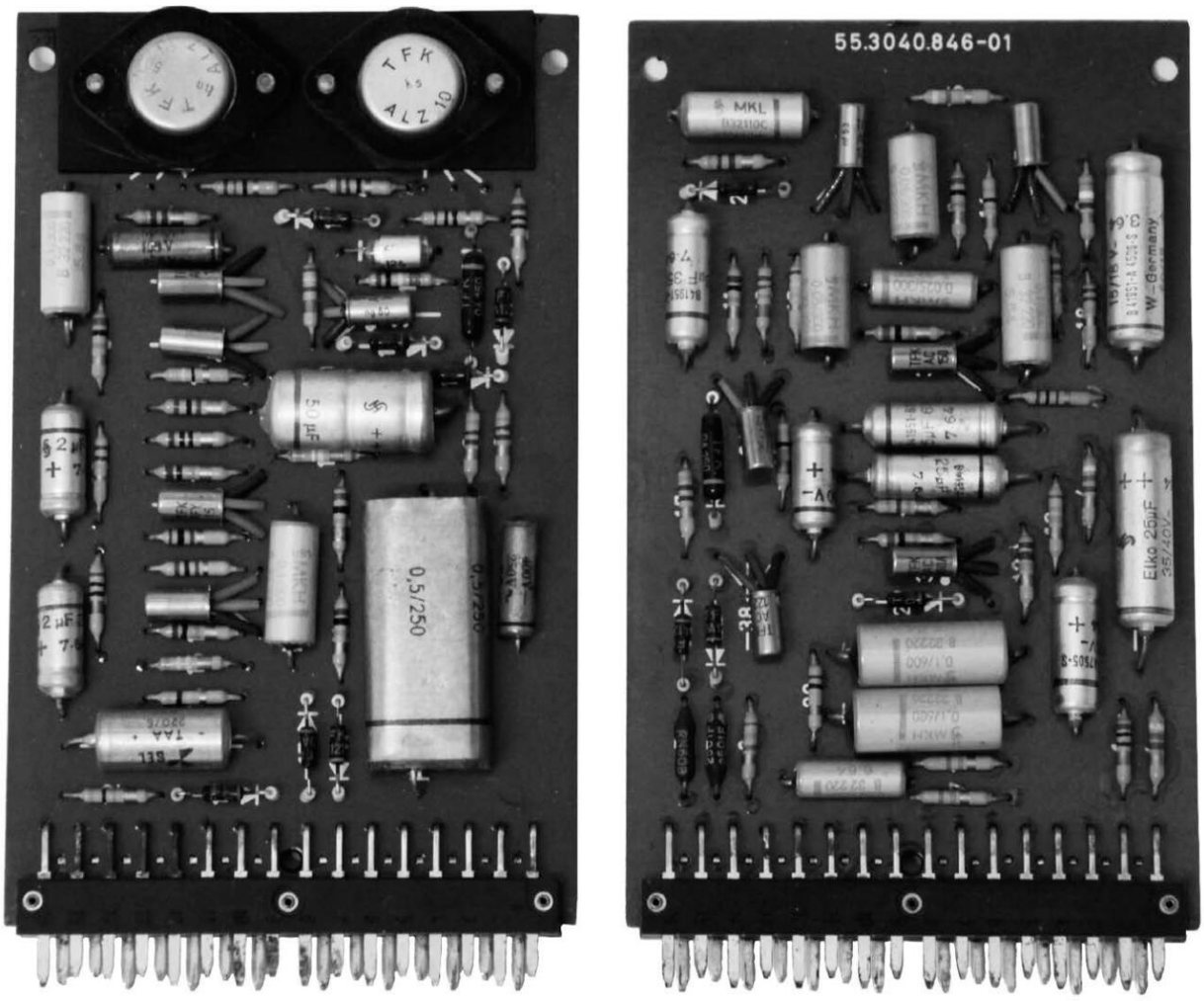


Figure 4.16. Early fully transistorized chopper-stabilized amplifier

→Figure 4.17 shows the successor of this amplifier using an all electronic chopper, thus greatly maximizing MTBF[33] and simplifying its overall construction. This amplifier was used in the Telefunken *RA 770* precision analog computer and has a DC gain of $3 \cdot 10^8$ for low frequency input signals. The electronic chopper and synchronous rectifier are under the shielding shown in the right upper part of the picture. The output stage transistors with their heat sinks are visible in the lower right part of the picture.[34]

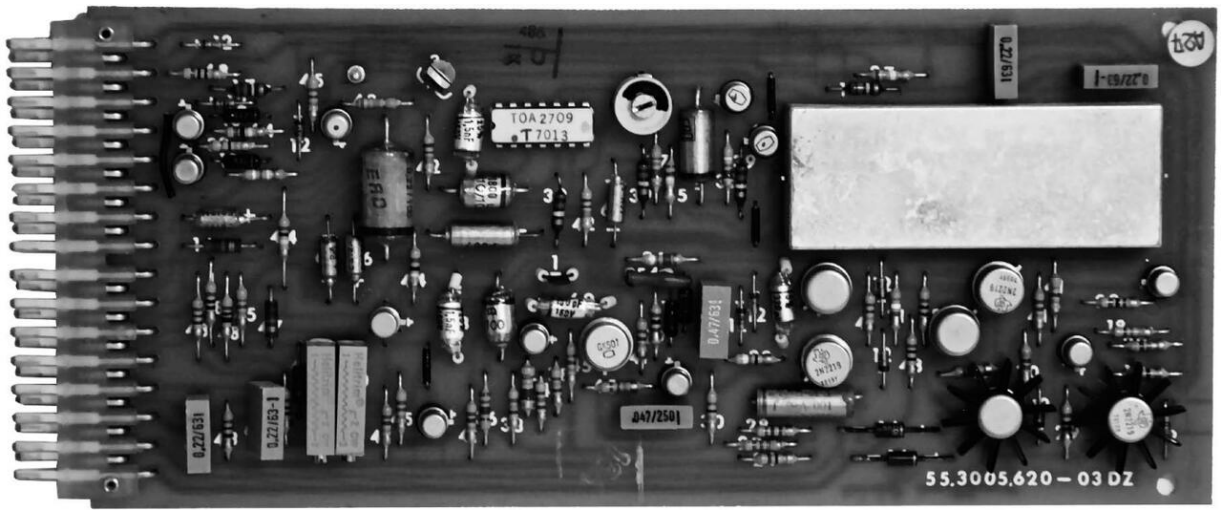


Figure 4.17. Late chopper-stabilized amplifier with all electronic chopper

4.2 Summers

One of the most basic computing elements of an analog computer is the *summer* or *summing amplifier*, which is basically a direct consequence of equations (→4.3) and (→4.4). →Figure 4.18 shows the simplified schematic of one of the earliest summing amplifiers developed by Karl Dale Swartzel[35] in 1941 at *Bell Laboratories* for application in a fire control computer.[36]

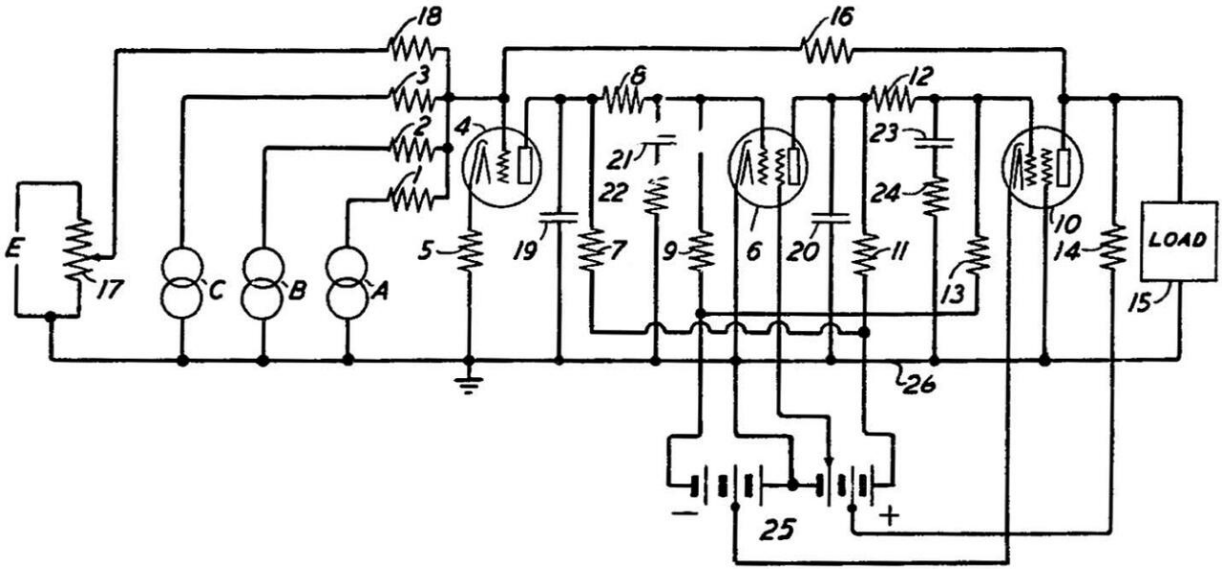


Figure 4.18. Schematic of the “Summing Amplifier” developed by KARL D. SWARTZEL (cf. [→SWARTZEL 1946])

This particular amplifier has three inverting stages employing one triode in the input stage and two tetrodes for the remaining two stages, so the overall amplifier is of the inverting type making a negative feedback scheme as developed by BLACK possible. The summing junction at the grid of the triode input stage (4) is connected to four input resistors 1, 2, 3 and 18 as well as to the feedback resistor 16. Of these four inputs only three are used for signals to be processed while the fourth input is reserved for a manual drift compensation circuit consisting of a voltage source E and a potentiometer 17. This amplifier requires supply voltages of ±350 V, −135 V, +250 V, and +75 V and has an overall gain of about 60,000. The various RC networks between the three amplifier stages stabilize the circuit.

→Figure 4.19 shows an equivalent setup based on an operational amplifier. In this case three inputs e_1, e_2, and e_3 and a connection to the summing junction[37] as well as an output connection are available. The simplified symbol for such a summer is shown in →figure 4.20.

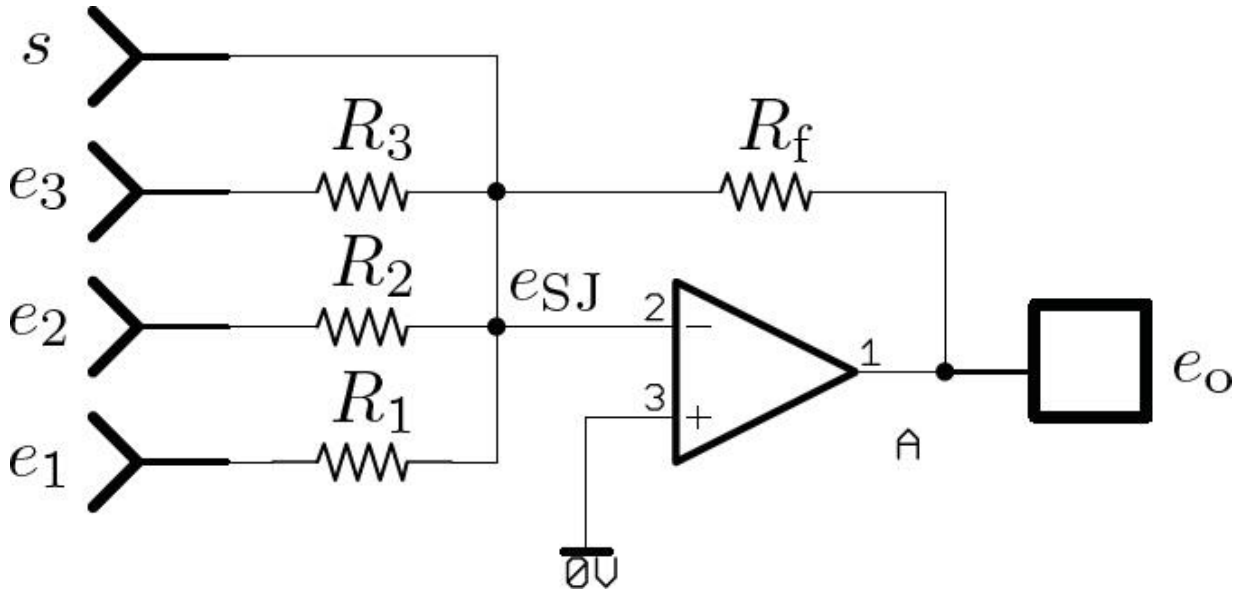


Figure 4.19. Schematic of a three-input summer

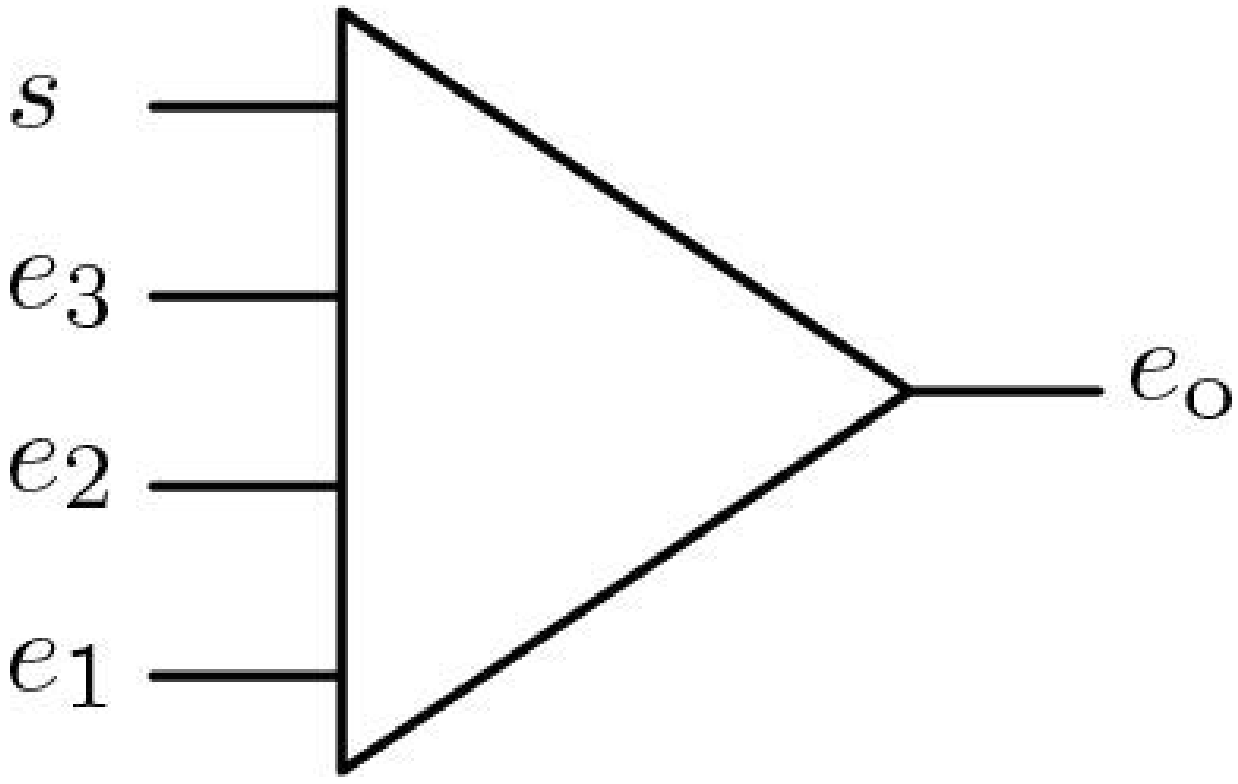


Figure 4.20. Symbol of a summer

Given sufficient open loop gain A of the operational amplifier used, the operation performed by this circuit is basically only determined by the passive elements at its input and in the feedback loop.

Extending equation (→4.4) by additional inputs yields

$$\sum_{i=1}^{n} \frac{e_i}{R_i} = -\frac{e_\mathrm{o}}{R_\mathrm{f}} \tag{4.5}$$

for n inputs with input resistors R_1, ... , R_n and a feedback resistor R_f. Introducing weighting coefficients $a_i = R_\mathrm{f}/R_i$ for all inputs, the output voltage of this circuit is

$$e_\mathrm{o} = -\sum_{i=1}^{n} a_i e_i.$$

Thus, a typical summing amplifier yields the negative of the sum of its weighted input voltages.[38] Typical values for a_i are 1 and 10 although in some cases even weights 4 or 5 can be found.[39] If other weights are required in a computation, inputs can be paralleled or external resistors or potentiometers can be connected to the summing junction input.

→Figure 4.21 shows a typical early summing amplifier – a model K3-A *adding component* made by GAP/R. This device computes the sum of four input voltages e_1, e_2, e_3, and e_4 plus an adjustable constant value e_0 that can be set manually using the potentiometer in the top half of the device. It delivers two output signals with opposite signs satisfying $\pm e = ke_0 + e_1 + e_2 + e_3 + e_4$ with $-1 \le k \le 1$ and $e_0 \in \{0, 5, 50\}$ V.



Figure 4.21. Model K3-A adding component (GAP/R), front and interior view

4.3 Integrators

Replacing the feedback resistor R_f in a summer circuit with a feedback capacitor C yields a circuit capable of integration as shown in →figure 4.22. The feedback current i_f flowing through the capacitor C is determined by $i_f = C\dot{e}_o$. Accordingly, equation (→4.5) can be changed to

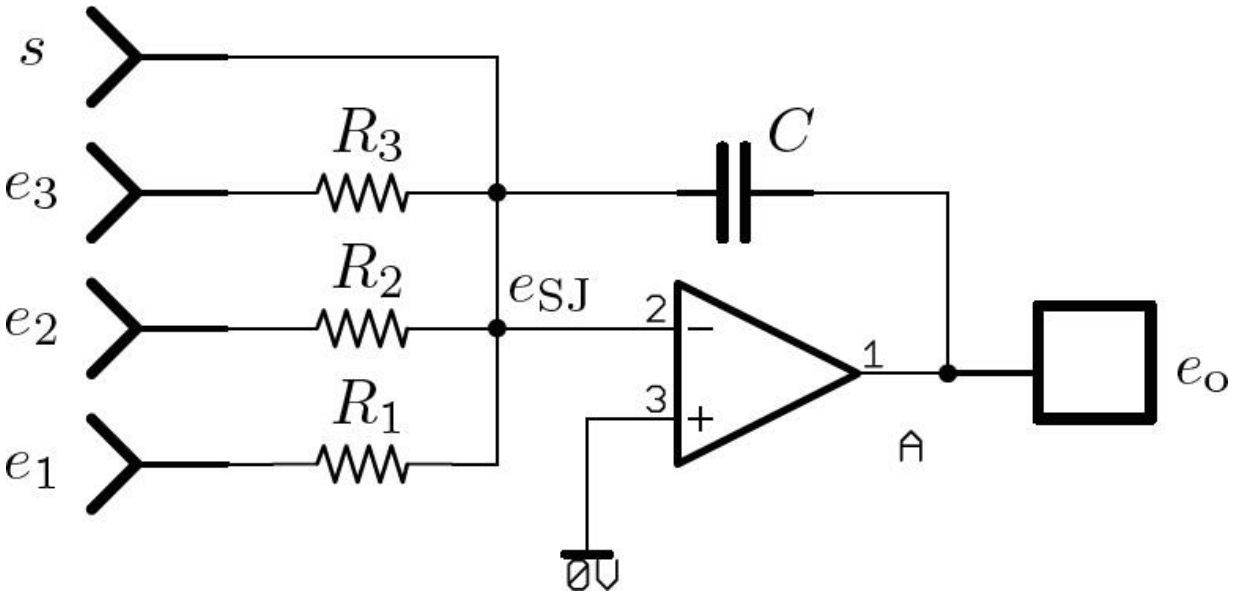


Figure 4.22. Basic structure of an integrator

$$\sum_{i=1}^{n} \frac{e_i}{R_i} = -C\dot{e}_o. \tag{4.6}$$

Redefining the weighting coefficients as $a_i = 1/R_iC$ and solving (→4.6) for e_o yields

$$e_o = \int_0^T \sum_{i=1}^{n} a_i e_i \mathrm{d}t + e(0)$$

where $e(0)$ denotes the *initial condition* of the integrator, i. e., the value of e_0 at time $t = 0$.

→Figure 4.23 shows the symbol used to denote an integrator in an analog computer setup diagram. Since such an integrator is obviously working with time being the free variable it needs some means to reset, or even better preset, the integrating capacitor C to an initial value. In addition to this, additional circuitry is necessary to start and stop the operation of the integrator. →Figure 4.24 shows the simplified schematic of a more realistic integrator as it is used in a typical analog computer.

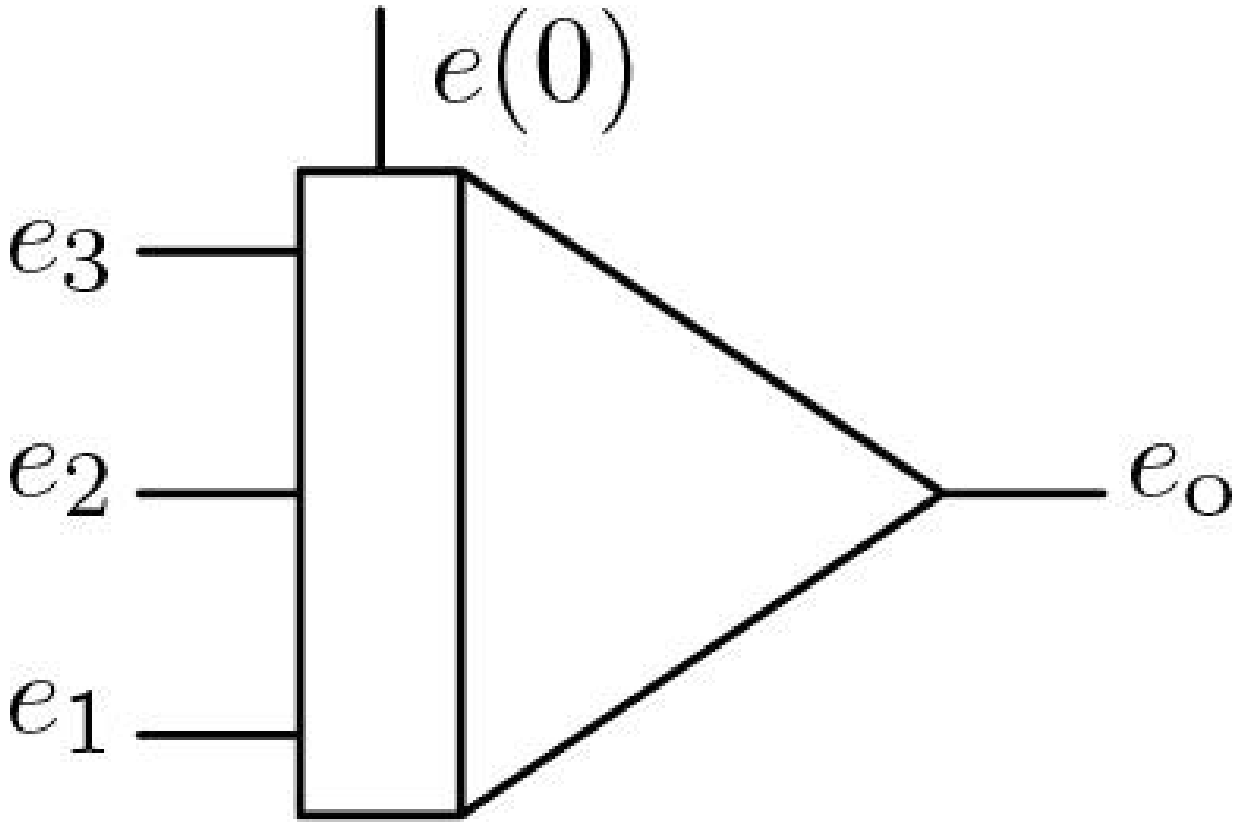


Figure 4.23. Symbol of an integrator with three inputs e_1, e_2, e_3, and initial condition input $e(0)$

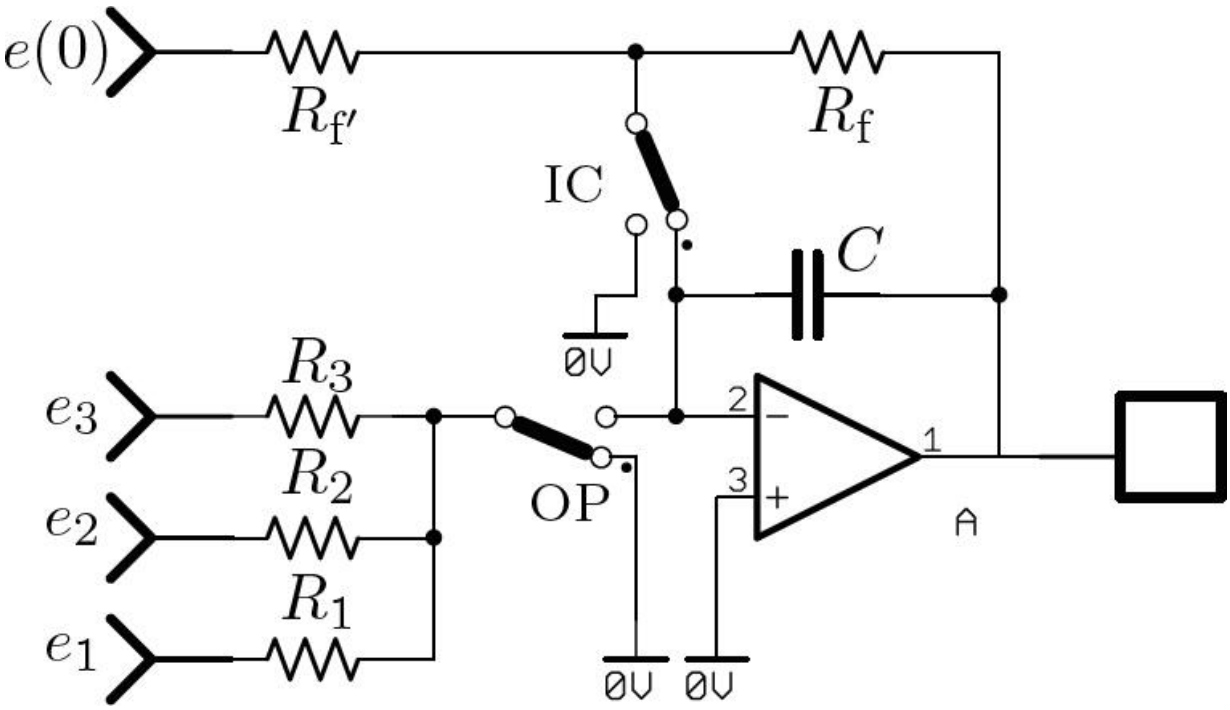


Figure 4.24. More detailed schematic of an integrator showing the control switches *IC* and *OP*

The central control elements are the two switches labeled *IC*, short for *initial condition* and *OP*, short for *operate/halt*. The integrator shown in this figure is currently in its IC mode: The inputs e_1, e_2, and e_3 are grounded through the operate/halt switch while the initial condition switch effectively turns the circuit into a simple summer with only one input $e(0)$. The integration capacitor C is switched in parallel with R_f. Since the summing junction at the inverting input of the operational amplifier is always maintained at 0 V, the capacitor will be charged to the voltage at the output of the amplifier. This, in turn, is determined by the input signal at $e(0)$ and the ratio R_f/R_f', which is typically equal to 1. The capacitor is thus pre-charged to the initial value $-e(0)$.[40]

To switch the integrator into the *OP* mode both switches change their state. This grounds the connection between R_f' and R_f, effectively disabling the input $e(0)$ and connects the input resistors

R_1, R_2, and R_3 to the summing junction of the amplifier. The circuit now performs an integration with respect to time as its free variable.

If the switch OP is reset the integrator enters *halt mode*. In this mode the last value of the integration will be stored.[41] This mode is normally used to pause a computation to read out values of interest from the various elements of the current setup.

In most cases the control switches IC and OP of all integrators in an analog computer setup are controlled by a common set of signals. In some cases it is necessary to control individual integrators or groups of integrators. This makes it possible to solve equations with one group of integrators based on initial conditions generated by a second group. Another typical application for such a more sophisticated control setup is the use of integrators as analog memory cells.

If an integrator is to be operated under individual control the symbol shown in →figure 4.25 is used, which shows the actual inputs controlling the switches IC and OP of the integrator.

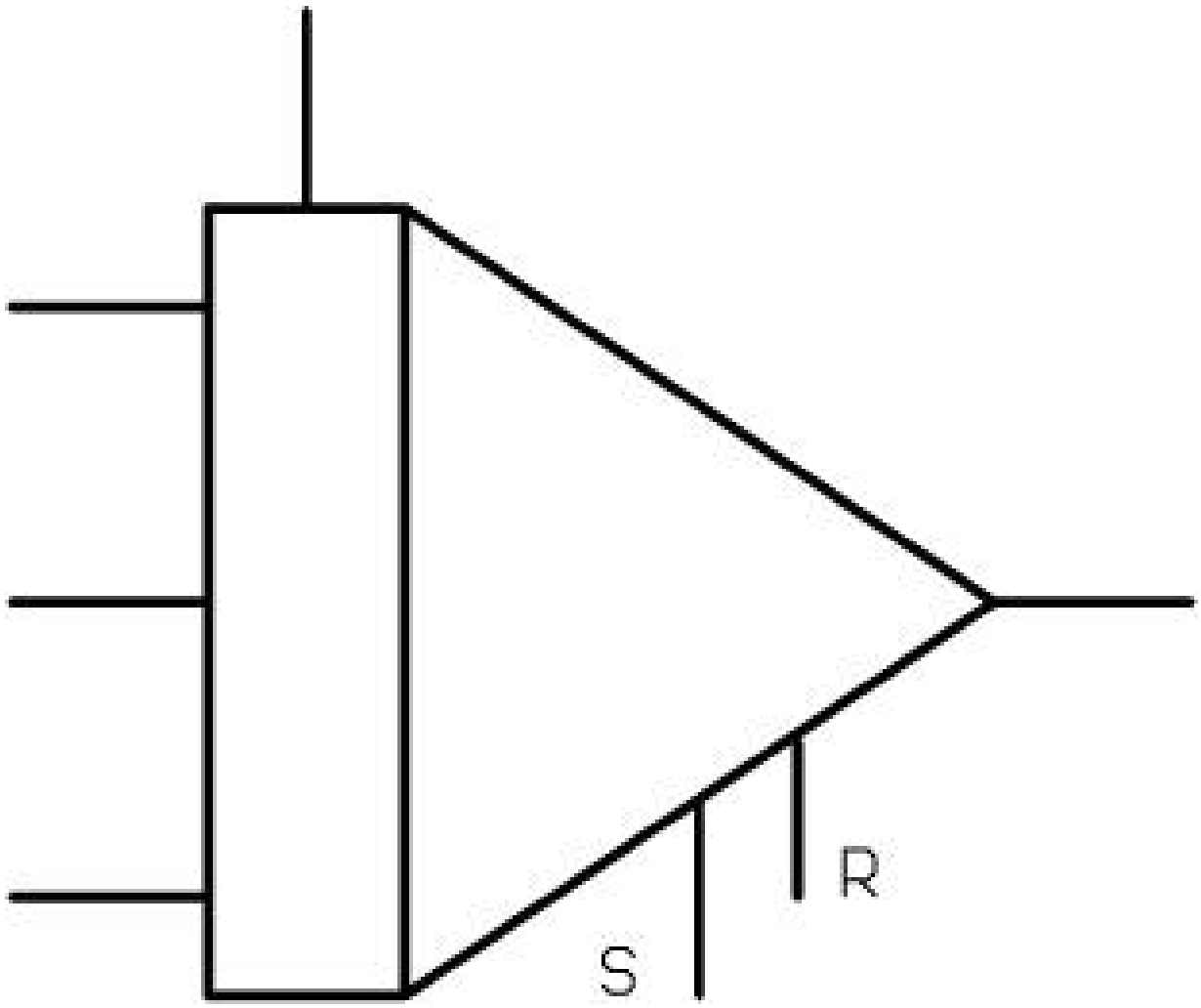


Figure 4.25. Integrator symbol showing control inputs S and R for IC and OP switches

A realistic integrator circuit is still more complex than the schematic shown in →figure 4.24: High-speed integrators use electronic switches in place of the relay contacts shown here. In addition to this, most integrators feature more than one integration capacitor *C*, thus allowing to select different ratios a_i for their operation.[42] Some machines offer as many as four or five different values for *C* (usually powers of ten). Typically, these are selected by shorting certain jacks of an integrator or from a central control panel.

→Figure 4.26 shows a dual integrator made by *Electronic Associates, Inc., EAI* for short. The chassis contains the passive elements for two integrators – the associated operational amplifiers are contained in a second drawer and are not shown here. Each integrator channel contains four capacitors implementing four different time-scale factors.

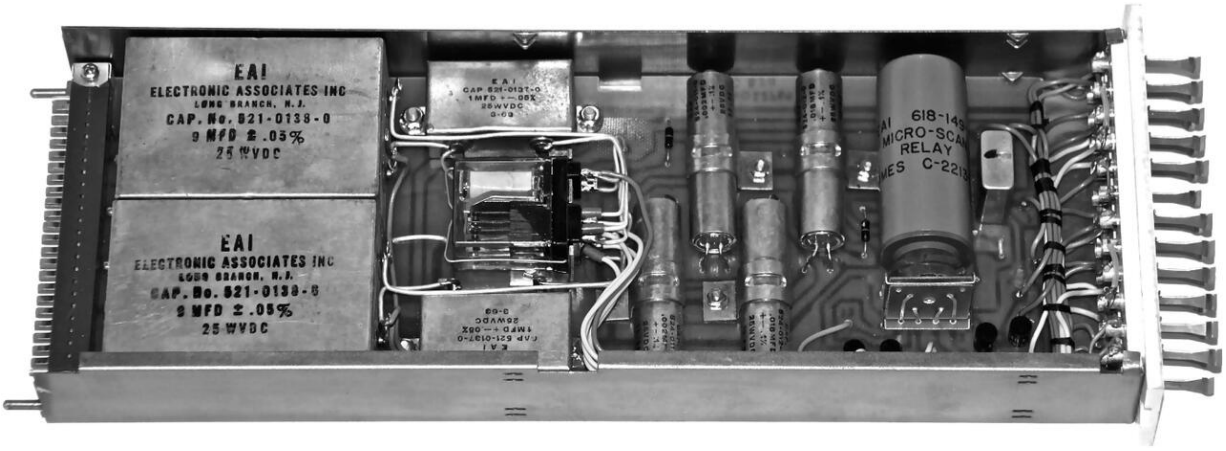


Figure 4.26. EAI dual integrator used in an EAI 580 analog computer

An interesting *hybrid integrator* is described in [→Rogers 1963, p. 13]. It operates on values (x_A, x_D) in a mixed analog/binary representation using a standard integrator in conjunction with a binary accumulator driving an *analog-digital-converter* (*ADC*), both feeding a common summer. The output

$$\int_0^T (x_A,\ x_D)\ \mathrm{d}t = \int_0^T x_A \mathrm{d}t + \sum_{i=1}^{n-1} x_{D_i} \Delta t + x_{D_i}(t - (n-1)\Delta t) \tag{4.7}$$

of this hybrid integrator is again a mixed representation value.[43] The idea was to extend the range of a classic integrator substantially thus simplifying scaling or even making it unnecessary.

4.4 Coefficient potentiometers

The fixed input weights a_i of summers and integrators are not sufficient for representing general coefficients in a computer setup. This is done using *coefficient potentiometers*. In the simplest case these are just potentiometers connected as voltage dividers as shown in →figure 4.27. In an analog computer program such a voltage divider is typically represented by a circle with one input and one output connection.[44] In some cases it is necessary to have a potentiometer with two inputs, which is shown in the right half of the figure.

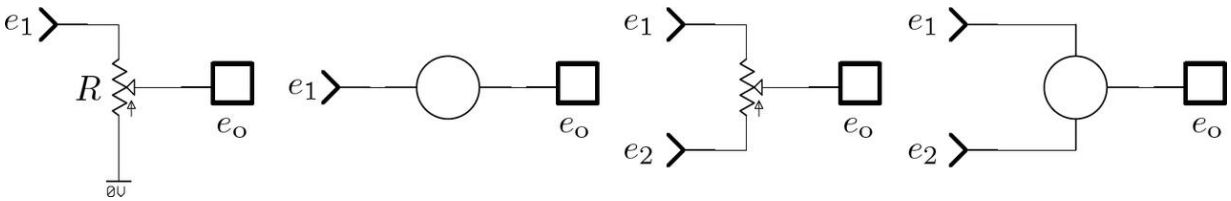


Figure 4.27. Coefficient potentiometers – basic implementation and representation

With α denoting the position of the potentiometer's wiper, the overall resistance R is partitioned into an upper and a lower resistor R_u and R_l connected in series:[45]

$$R_u = (1 - \alpha)R$$
$$R_l = \alpha R$$

Without load applied to the wiper of the potentiometer the operation of such a coefficient potentiometer can be described by

$$\frac{e_o}{e_1} = \frac{R_l}{R} = \alpha.$$

In a traditional analog computer this is a hypothetical case since the following computing elements normally act as a non-negligible load.[46] A computing element connected to the wiper of a

potentiometer acts as a resistive load R_{load} connected in parallel to R_{l}. Thus,

$$R_1 \parallel R_{\text{load}} = \frac{R_1 R_{\text{load}}}{R_1 + R_{\text{load}}}.$$

The total resistance of the coefficient potentiometer with a load connected is then

$$R_{\text{total}} = R_{\text{u}} + R_{\text{l}} \parallel R_{\text{load}} = R_{\text{u}} + \frac{R_{\text{l}} R_{\text{load}}}{R_{\text{l}} + R_{\text{load}}}.$$

The loaded potentiometer can thus be described by

$$\begin{aligned} \frac{e_0}{e_1} &= \frac{R_{\text{l}} \| R_{\text{load}}}{R_{\text{total}}} = \frac{\frac{R_{\text{l}} R_{\text{load}}}{R_{\text{l}} + R_{\text{load}}}}{R_{\text{u}} + \frac{R_{\text{l}} R_{\text{load}}}{R_{\text{l}} + R_{\text{load}}}} = \frac{R_{\text{l}} R_{\text{load}}}{R_{\text{u}}(R_{\text{l}} + R_{\text{load}}) + R_{\text{l}} R_{\text{load}}} \\ &= \frac{R_{\text{l}}}{\frac{R_{\text{u}} R_{\text{l}}}{R_{\text{load}}} + R_{\text{u}} + R_{\text{l}}} = \frac{\alpha R}{(1-\alpha) R \frac{\alpha R}{R_{\text{load}}} + R} \\ &= \frac{\alpha}{(1-\alpha)\alpha \frac{R}{R_{\text{load}}} + 1} \end{aligned} \tag{4.8}$$

→ Figure 4.28 shows the effect of potentiometer loading with respect to the value set by α. It is obvious that setting a potentiometer manually by just looking at the value displayed on its dial will not yield the desired value since the load exerted by the computing elements connected to the potentiometer has to be taken into account.[47]

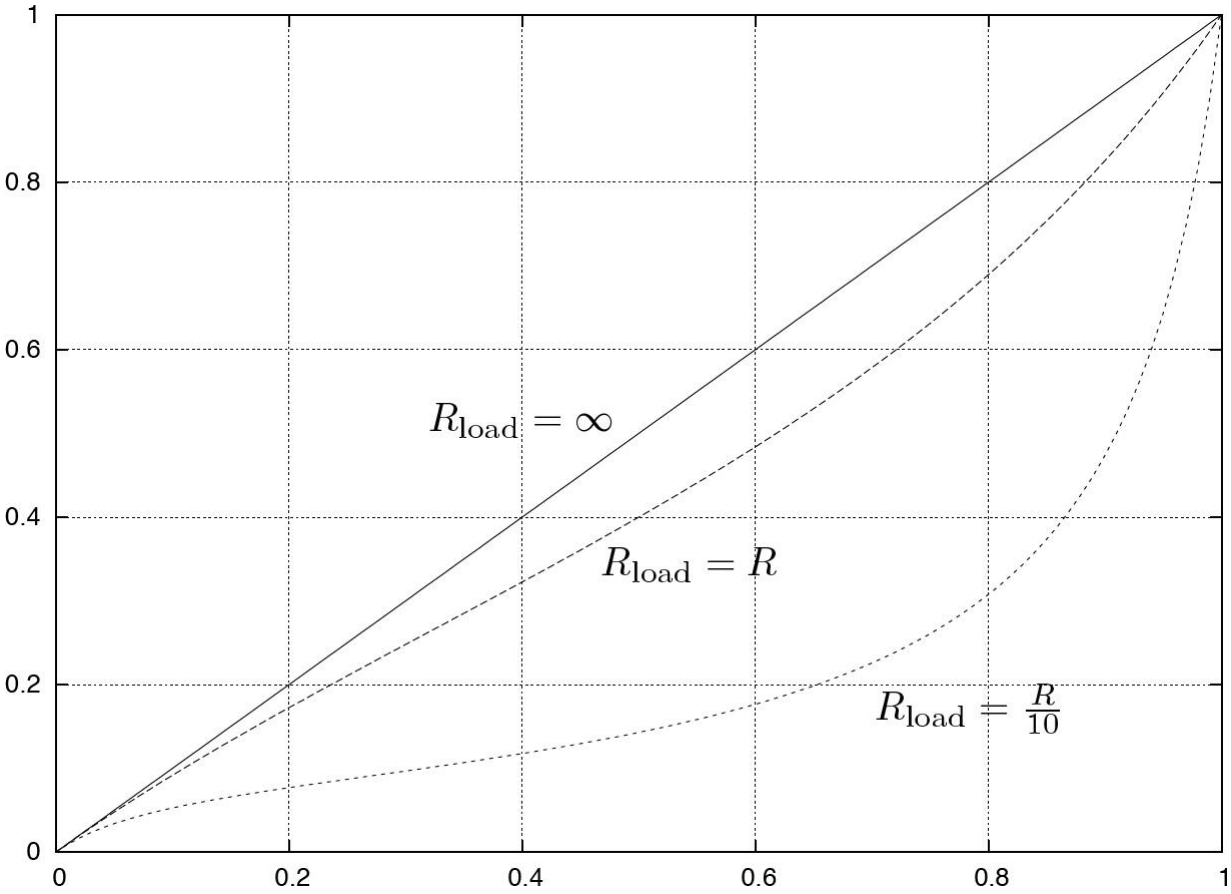


Figure 4.28. Coefficient potentiometer with $R_{\text{load}} = \infty$, $R_{\text{load}} = R$ and $R_{\text{load}} = R/10$

To overcome this problem most analog computers have a special mode of operation, called *potentiometer set* (*Pot Set*). In this mode, which can be selected from the central control panel of the computer, all integrators and summers have their input resistor networks grounded, so no actual computation takes place while every computing element exhibits its actual load to its signal sources – including the various coefficient potentiometers. To set a potentiometer to a precise value in this mode, small and simple systems offer a push button next to each potentiometer that will connect its input to +1 machine unit and connect a precision voltmeter – either a compensation voltmeter[48] or a digital voltmeter – to its wiper when depressed. Turning the potentiometer's dial, the

desired coefficient value can now be set very precisely by means of the voltmeter.

When cheap integrated circuit operational amplifiers became available some analog computers employed simple buffer stages to decouple the coefficient potentiometers from the load of the following computing element, thus vastly simplifying the setup procedure.[49]

→ Figure 4.29 shows two typical buffering schemes: The circuit on the left uses an operational amplifier as a simple non-inverting output buffer for the potentiometer *R*. The circuit shown on the right is more versatile and is used for example in the *EAI 2000* hybrid computer. In contrast to a simple voltage divider with buffer amplifier this circuit allows coefficient values to be set in the range −1 … 1 as the following consideration shows: If α is set to 0, the potentiometer's wiper is at its grounded end, thus the non-inverting input of the operational amplifier is grounded and the amplifier acts as a simple inverting amplifier with gain R_f/R_i. Typically $R_f = R_i$ is chosen so that the overall gain is 1. Setting the potentiometer to $\alpha = 1$ causes the wiper to be at the upper end of the potentiometer thus connecting the non-inverting input of the amplifier with the input e_1. Since the amplifier will maintain the voltage e_{SJ} at its inverting summing junction at 0 V, it follows that $e_o = e_1$. So this circuit can be described by[50]

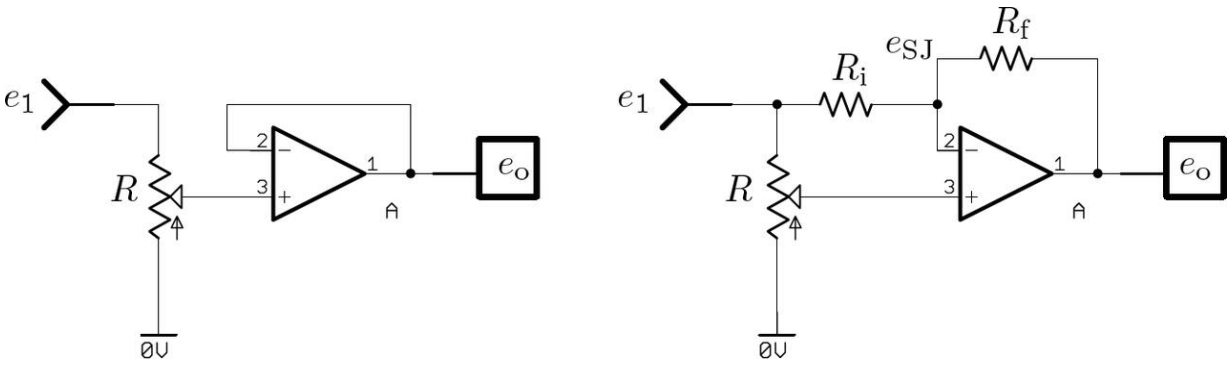


Figure 4.29. Buffered coefficient potentiometers

$$\frac{e_0}{e_1} = 2\alpha - 1$$

Large installations often used servo driven potentiometers that could be set from a central control panel or automatically from an attached paper tape reader or even an attached stored-program computer, thus allowing rapid setup of coefficients. →Figure 4.30 shows a typical servo potentiometer from an *EAI 580* analog computer. The 10-turn wire wound potentiometer is visible on the left with the motor driving a reduction gear visible in the upper right half. The small circuit board below the motor holds the relay used to select the potentiometer for readout. Later hybrid computers[51] often replaced classic potentiometers by digitally controlled attenuators.[52]

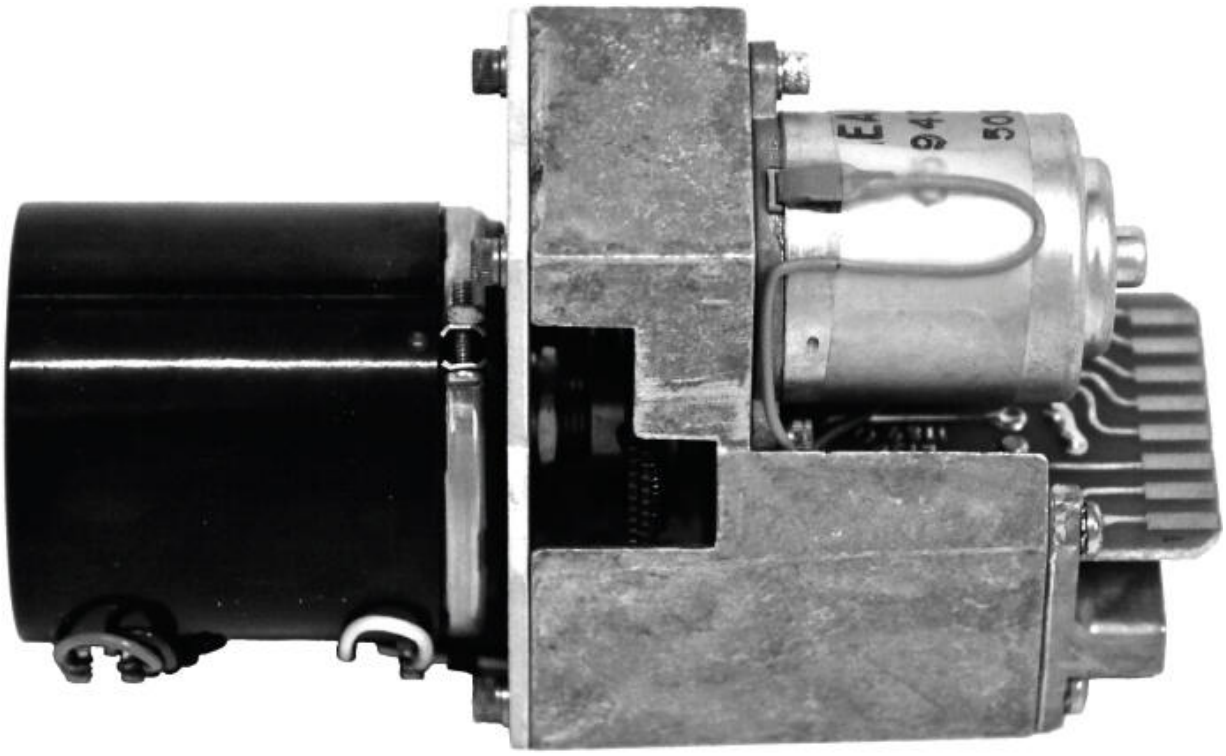


Figure 4.30. EAI servo potentiometer

Using only summers, integrators and coefficient potentiometers, an analog computer would be restricted to the solution of simple linear differential equations of the form

$$\sum_{i=0}^{n} a_i y^{(i)} = 0.$$

To overcome this restriction additional computing elements are required which are described in the following sections.

4.5 Function generators

Another basic task is the generation of a general function $f(x)$. Ideally, such a function could be derived as a solution of a differential equation, but in many cases functions are required which are defined by values obtained from experiments or other sources. In these cases special computing elements called *function generators* are required. Generating arbitrary functions with purely analog electronic means is not a trivial task and often requires a large amount of circuitry.[53] Even more difficult, at least in the common case, is the generation of functions like $f(x_1, x_2, \ldots)$ of more than one variable by analog electronic means. The symbol for representing a function generator with a single input is shown in →figure 4.31.

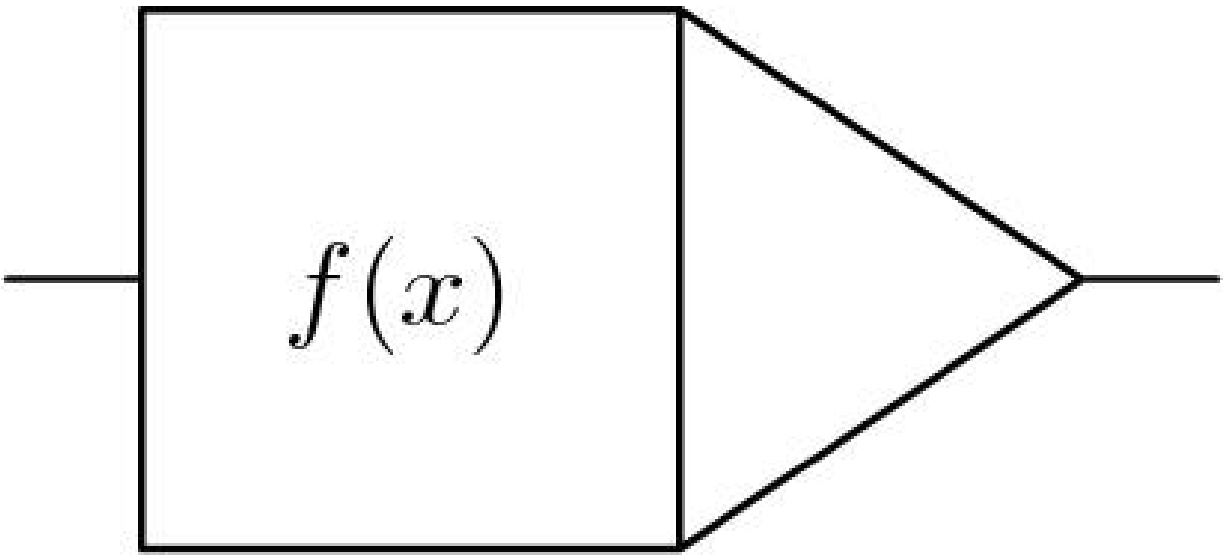


Figure 4.31. Symbol of a function generator

A plethora of different approaches to generate arbitrary functions have been developed over time. The following sections describe some of the more widely used or technologically more interesting implementations.[54]

4.5.1 Servo function generators

The earliest function generators used by Lovell et al. in their fire control computer were based on a servo circuit controlling a motor driving at least two potentiometers mounted on a common shaft. One of these potentiometers is used to generate a feedback value for the servo circuit while the other(s) generate the function(s) desired depending on the angular position of their respective wipers. →Figure 3.24 (→section 3.3) shows such a setup used to generate trigonometric functions.

Other implementations of servo based function generators use cams driven by the servo controlled motor with a feeler pin riding on the cam's surface driving an output potentiometer. Another interesting implementation of a cam potentiometer is described in [→Loveman 1962].

These techniques have the disadvantage that specially wound potentiometers are required and these are delicate instruments and difficult to manufacture. The same is true for machined precision cams. Consequently, early *tapped potentiometers* were employed for function generation. These are based on standard 10-turn wire wound potentiometers that have been modified to feature multiple taps connected equidistantly to the resistive element. Using these taps, the potentiometer can be used as a linear interpolator between the voltages applied to its various taps. The taps' positions correspond to the breakpoints while the voltage applied to the tap represents the desired function value at this particular breakpoint. Using a servo system driving the potentiometer's shaft according to an input value x, such a setup yields a function $f(x)$ which is the result of linear interpolation between the values set at the taps.[55]

→ Figure 4.32 shows a classic servo function generator potentiometer for a trigonometric function such as sine and cosine. The sliding contact can be seen under the transparent cover with its degree scale. It moves along a specially wire wound potentiometer body which can be easily exchanged if different functions are required. A unique feature of this particular device is that up to four coaxial resistive elements can be inserted so that several functions of one common variable can be generated simultaneously. The device shown here is part of a *C-118A Aircraft Simulator Trainer*.[56]



Figure 4.32. Servo function generator potentiometer enclosure and resistive element (photos by CHRISTOPHER HOWARD, used with permission)

4.5.2 Curve followers

Another servo-based function generator is the *curve follower*, which is basically a servo circuit fed by a pickup connected to a *xy*-plotter as shown in →figure 4.33.

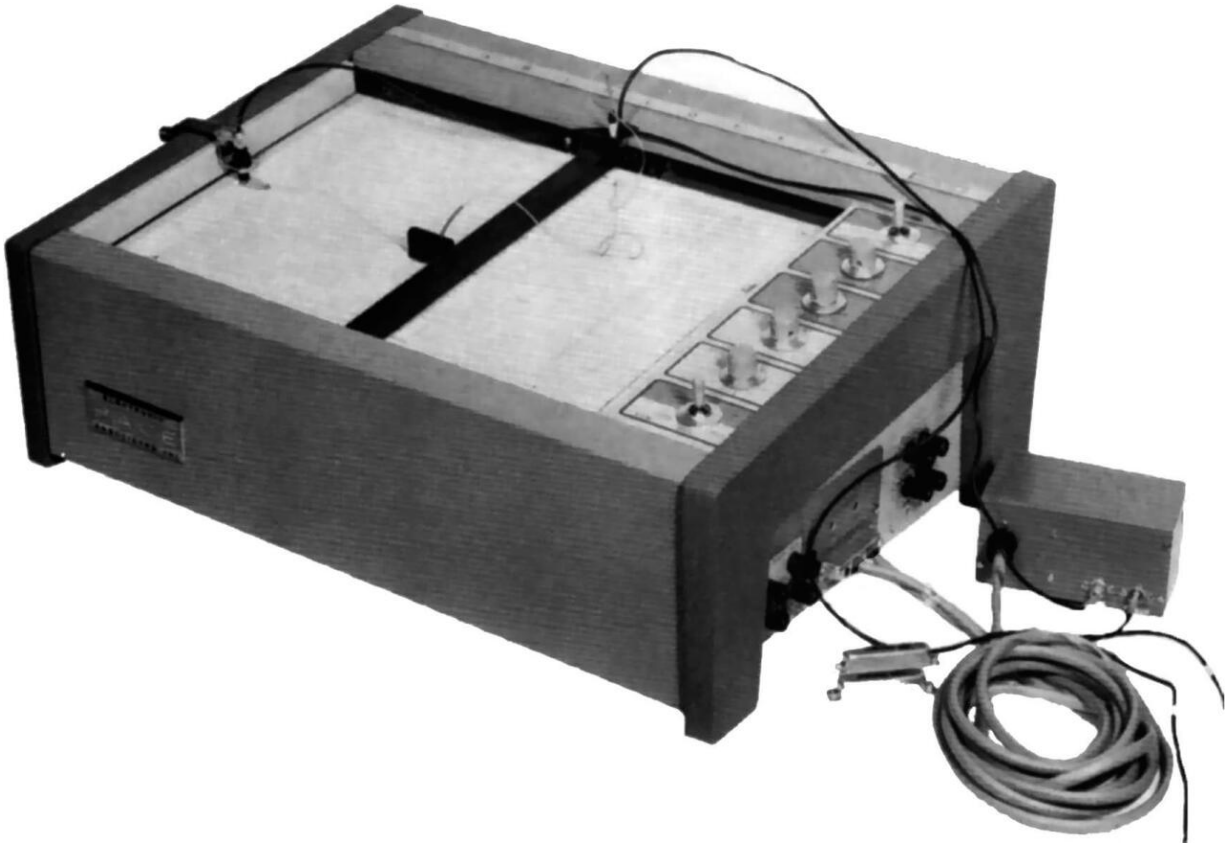


Figure 4.33. EAI curve follower

The basic idea is as simple as it is elegant: The function to be generated is painted or plotted on a sheet of paper with conductive ink. This graph is then placed onto the table of a modified *xy*-plotter that has a pickup coil mounted in place of the pen normally used. An external sine generator is then connected to the conductive trace of the function graph on the paper. A simple servo circuit now feeds the *y*-input of the plotter in such a way that the pickup coil will always be directly over the conducting function graph, while the *x*-input of the plotter is fed with the argument of the function to be generated. The *y* value used to drive the *y*-input of the curve follower now represents the desired value $f(x)$.

Other curve followers used an optical pickup to follow the function graph instead of the conductive ink and inductive pickup setup. The basic principle of operation remains the same.

An interesting variant of a curve follower is shown in →figure 4.34. The function is drawn on a long strip of paper which is transported using a reel-to-reel system. This graph is followed by means of two optical pickups. To achieve a high degree of precision, a *polydromial* representation of the function graph as shown in →figure 4.35 is used. Here, the function is sliced into vertical strips of equal height. Using two optical pickups moving in tandem at a fixed distance such a representation can be scanned without interrupting the output signal when the graph crosses a boundary between vertical strips. The desired value of $f(x)$ with x representing the position along the paper tape is the vertical position of the optical pickups.[57]

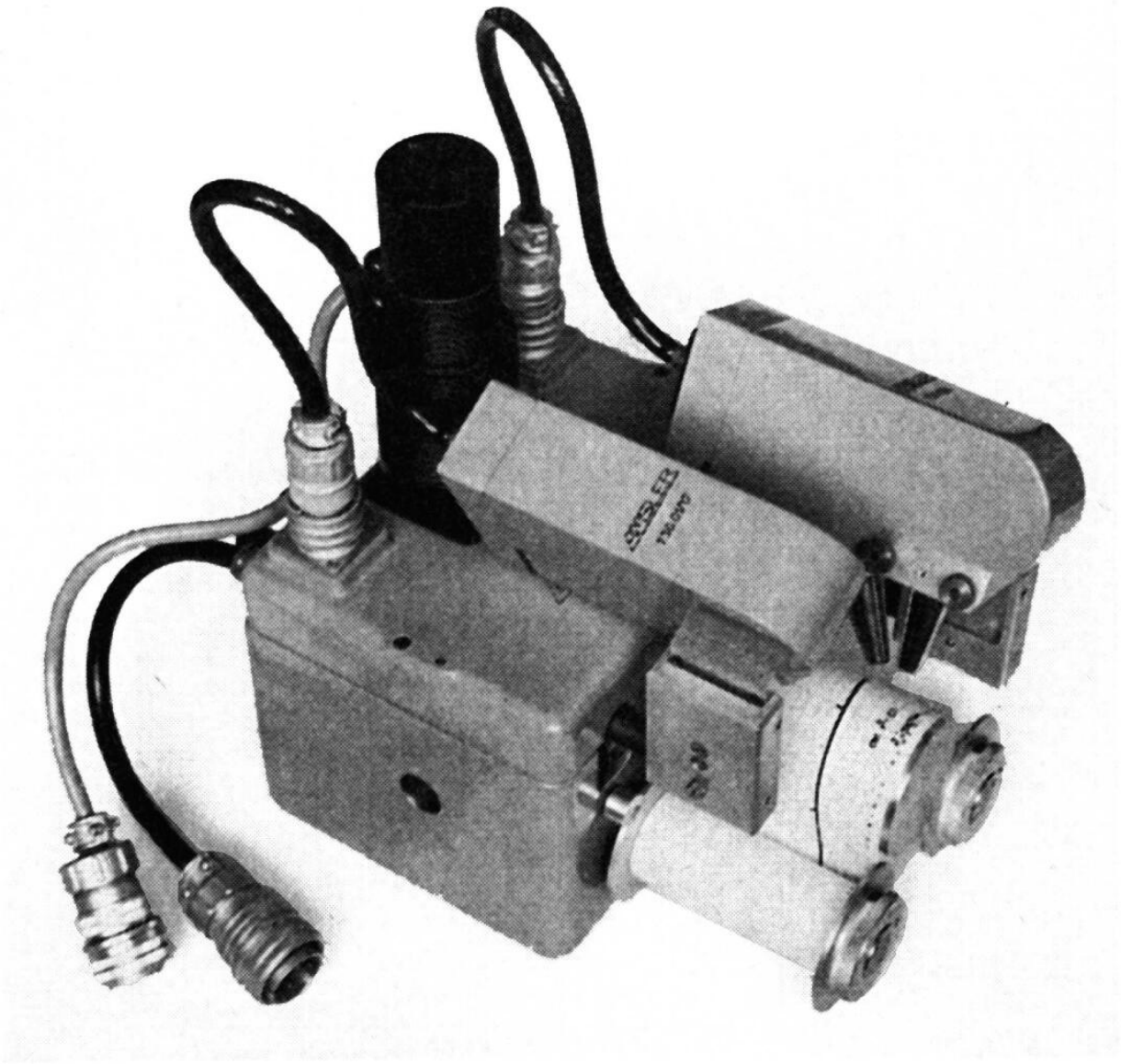


Figure 4.34. Polydrome curve follower (see [→ ERISMANN 1962, p. 207])

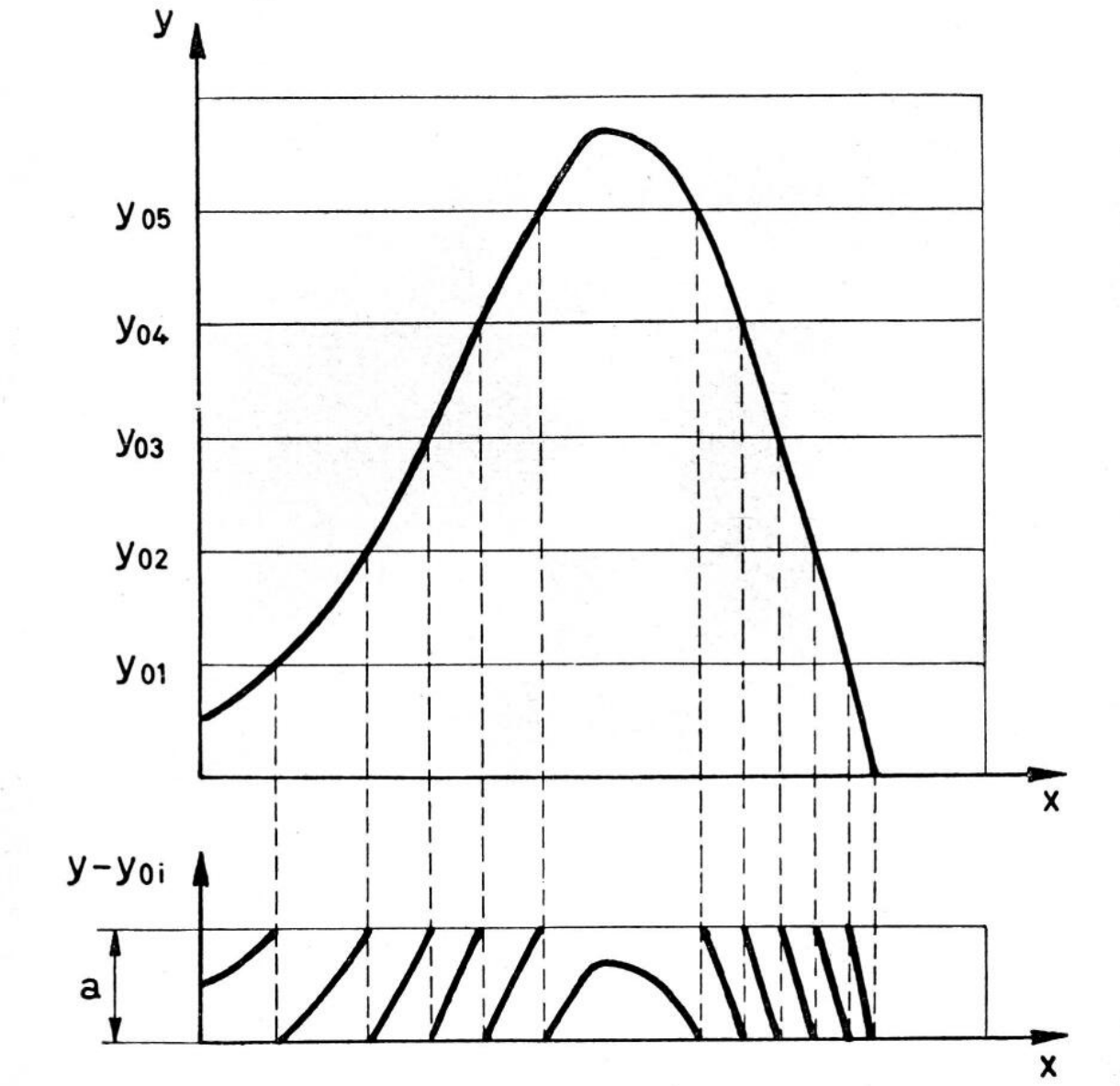


Figure 4.35. Polydromial representation of a function graph (see [→ ERISMANN 1962, p. 205])

4.5.3 Photoformers

As elegant as these electromechanical function generation techniques are, their main disadvantage is that they cannot cope with rapidly varying function arguments *x* due to the relatively large

moving masses. Fortunately the basic curve follower idea can be easily implemented using an all electronic approach like the one shown in →figure 4.36.[58]

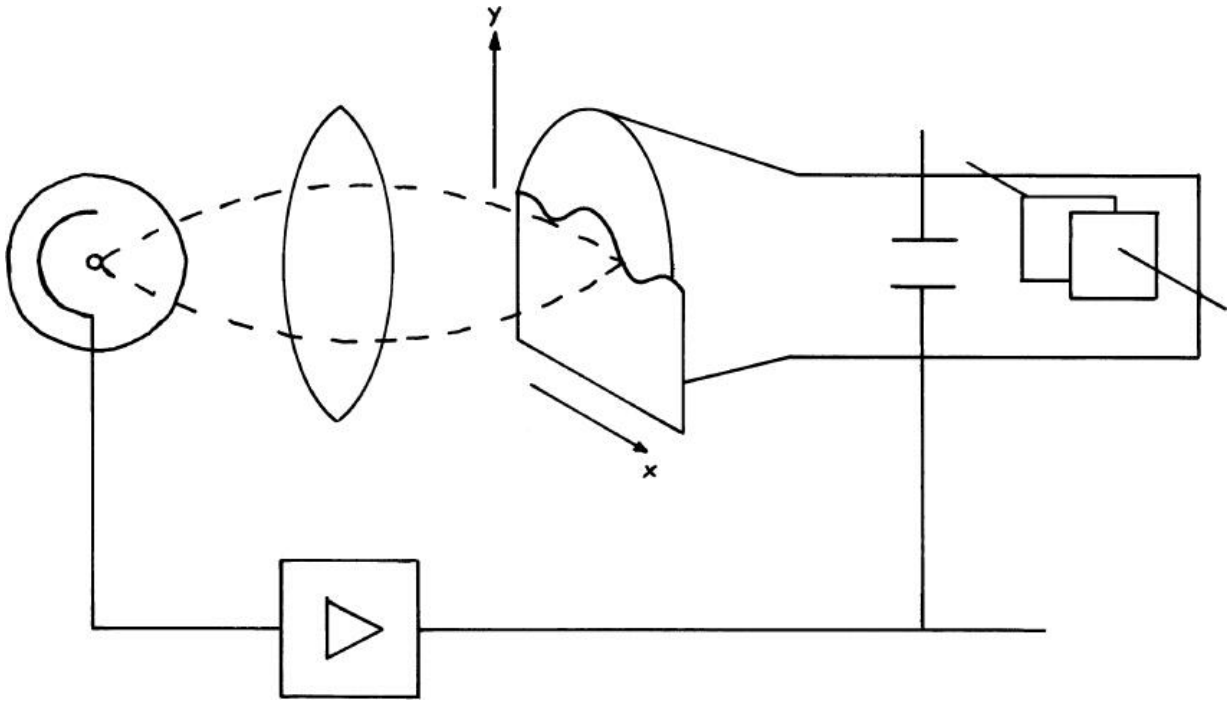


Figure 4.36. Optical function generator (cf. [→HAUG 1960, p. A3])

The function to be generated is represented by an opaque mask placed in front of an oscilloscope tube. The input to the *x*-deflection plates of the display tube is fed by the function argument. Using a photomultiplier and a simple servo circuit, a signal for driving the *y*-deflection plates is now generated so that the spot of light on the face of the display will always be placed at the boundary of the opaque mask; this system effectively generates $y = f(x)$.[59]

4.5.4 Varistor function generators

A different approach to function generation is the application of *varistors*.[60] A varistor is a current sensitive nonlinear resistor

implementing a basic function of the form[61]

$$e_{\mathrm{o}} = a e_1^n$$

with *n* being determined during production. Typical values are *n* = 2 or *n* = 3. The precision of these varistors is quite good. [→Korn 1962, p. 3-75] quotes an accuracy of 0.4 V for the *Quadratron*,[62] a varistor made by *Douglas Aircraft Company*.[63]

Of course having square and cube functions does not give the flexibility of a photoformer, but these varistor circuits were extremely cheap. This made it possible to use these devices to generate functions by truncated power series.[64]

4.5.5 Photocells

A truly unique analog computer was built by Eldo Koenig[65] who received the Alfred Noble Prize[66] for his work in the field of analog computing in 1951. This machine[67] was a special purpose system used for the simulation of magnetic circuits and other nonlinear systems including fluid flow, aerodynamics, etc.

It consisted basically of a large number of nonlinear computing elements based on AgCs/Cs-oxide photo cells exhibiting a rather pronounced nonlinear characteristic curve depending on the voltage across the tube and its illumination. Groups of cells representing similar or equal nonlinear terms in a given problem were illuminated by a shared adjustable source of light. The overall system is shown in →figure 4.37.

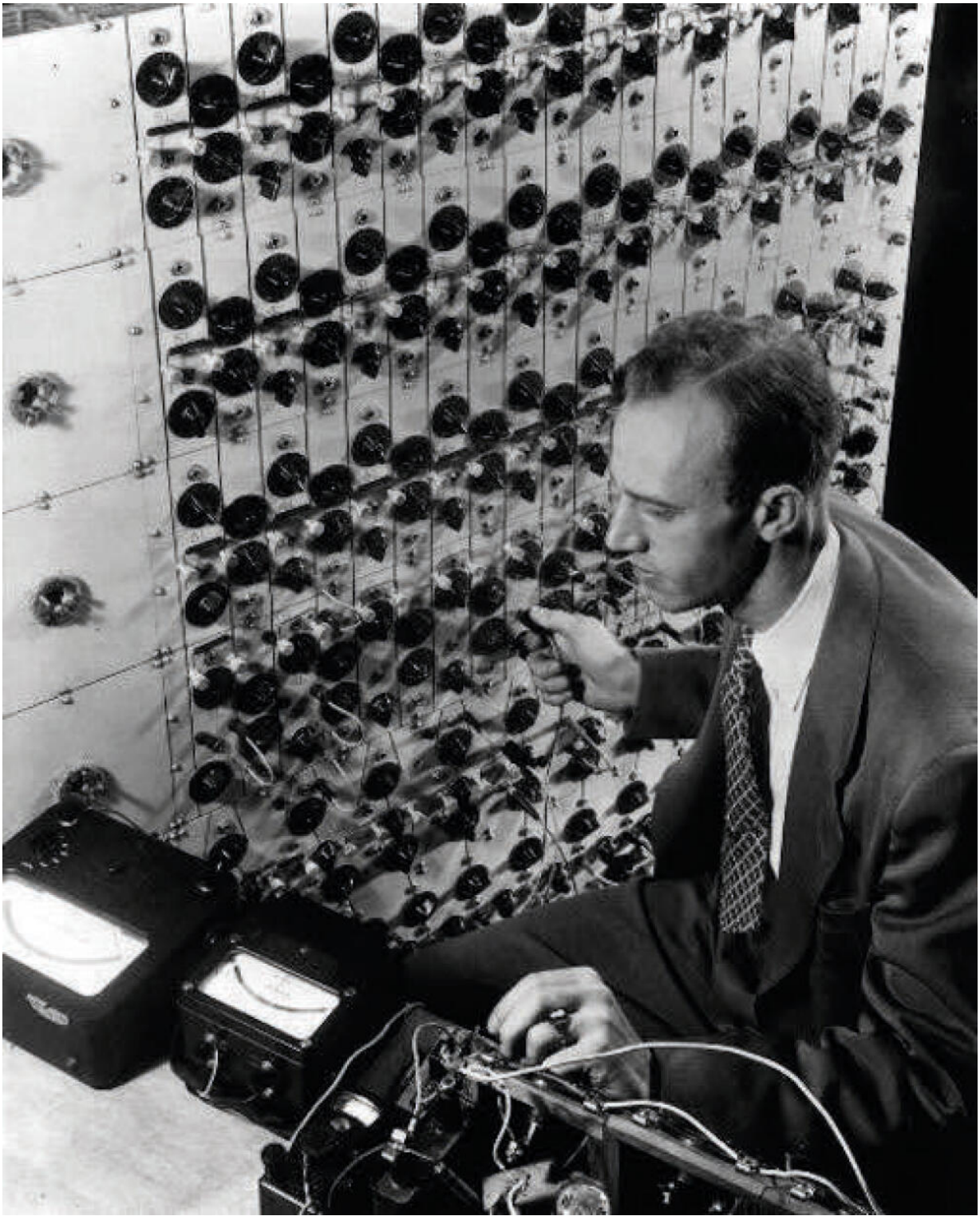


Figure 4.37. ELDO KOENIG in front of his photo cell analog computer

However, the use of photo cells as nonlinear computing elements did not catch on due to a number of problems. First, the characteristic curves of such tubes have so much variation that suitable tubes must be selected. These must then be pre-aged before actual use to minimize drift effects that would render the analog computer solution incorrect. Accordingly, this machine was basically a footnote in the history of analog computing.

4.5.6 Diode function generators

The availability of reliable and cheap germanium, and later silicon diodes gave rise to another technique of implementing function generators, the *diode function generator* or *polygon function generator*. The basic structure of such a circuit is shown in →figure 4.38.

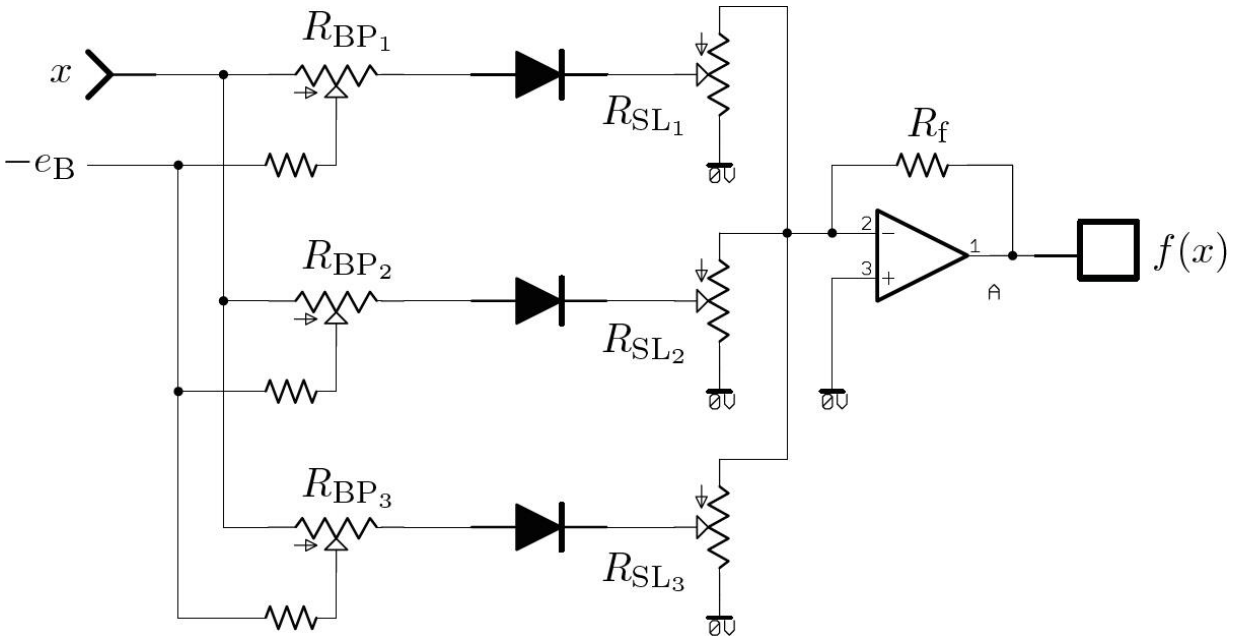


Figure 4.38. Principle of operation of a diode function generator

An ideal diode acts like a switch conducting current only in one direction but not in the reverse. So a simple ideal diode might be

used to implement a function such as

$$f(x) = \begin{cases} ax, & \text{if } x > 0, \\ 0, & \text{otherwise} \end{cases}$$

where the slope *a* can be set by a series resistor. If the voltage at which the diode starts conducting could be shifted arbitrarily, a function could be generated by a piecewise linear approximation using several of these circuits.

The circuit shown in →figure 4.38 is based on this idea: Each of the three diodes starts conducting at a voltage defined by the setting of its associated bias potentiometer R_{BP_i}, which effectively sets a breakpoint position. The voltages at the cathode of each of these diodes are then summed by an operational amplifier with variable input weights defined by the setting of R_{SL_i} and the feedback resistor R_{f}. R_{SL_i} defines the *slope* of the particular partial function.

Of course, a practical diode function generator features many more than just three of these biased diode circuits.[68] A typical implementation of a function generator made by Telefunken is shown in →figure 4.39. This particular example generates $f(x) = x^2$ for values $x \geq 0$ and was used in the various transistorized Telefunken analog computers.

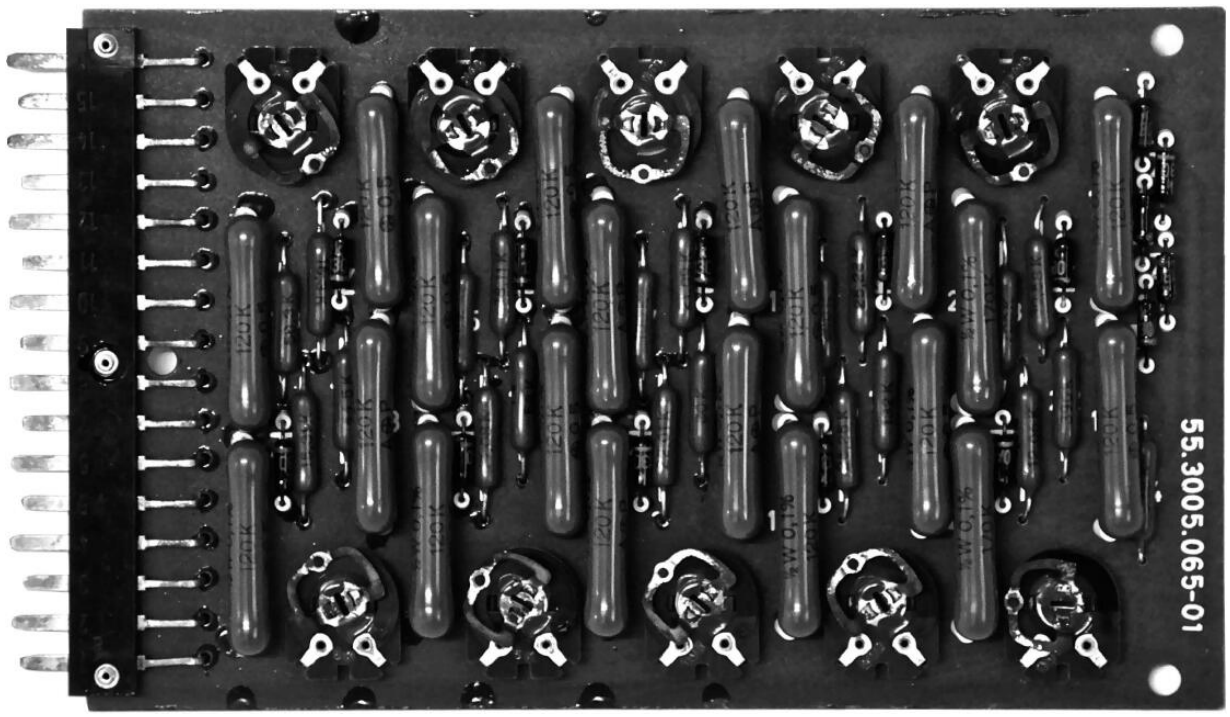


Figure 4.39. Diode function generator implementing $f(x) = x^2$ for $x \geq 0$ (Telefunken)

A more versatile *variable diode function generator*, a *VDFG* (made by EAI), is shown in →figure 4.40. It features ten diode segments with coarse and fine control for the respective breakpoints and slopes. The possibility of defining the breakpoints had the advantage that they could be set to match the function being generated as well as possible, thus minimizing errors. On the other hand, it greatly complicated the overall manual setup procedure. Consequently, other manufacturers implemented diode function generators with 21 equally spaced fixed breakpoints with only the slopes of the individual partial functions to be set manually.

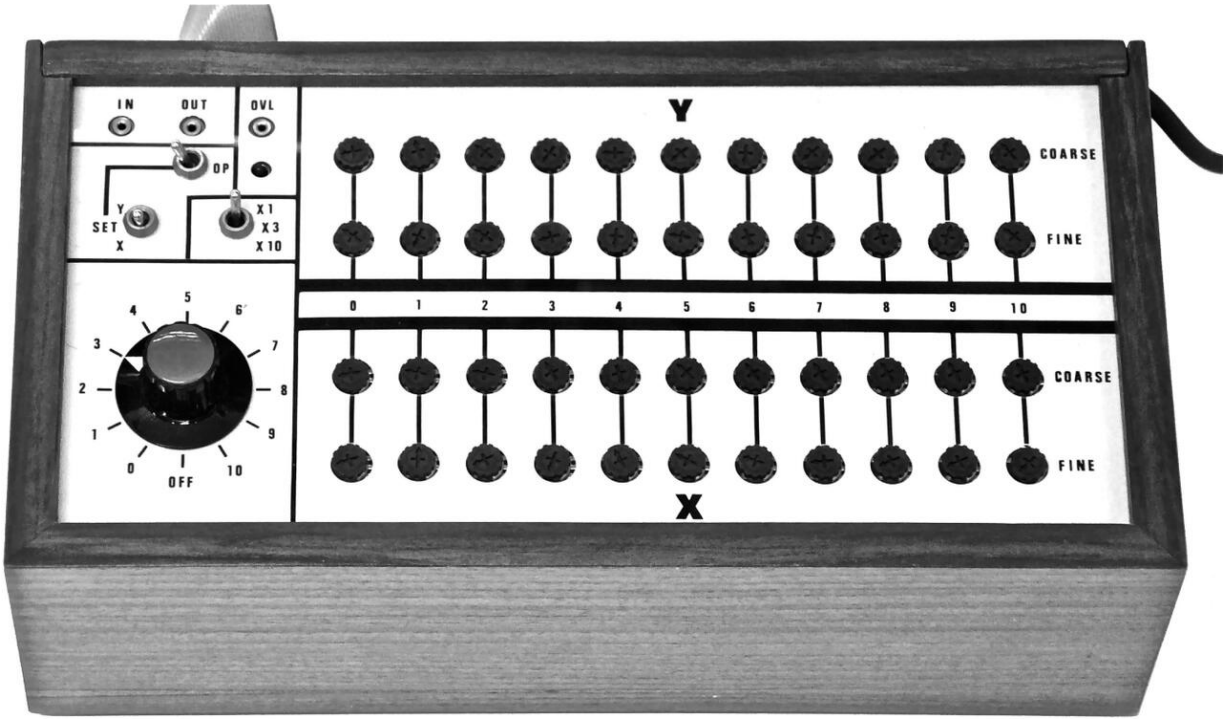


Figure 4.40. EAI function generator

The last and maybe most remarkable successor of these diode function generators is the AD639 integrated circuit which implements the basic trigonometric functions and their inverses with a high degree of precision.[69]

4.5.7 Inverse functions

Many functions can be readily implemented by generating the inverse of another readily available function. Using a function generator in the negative feedback path of an operational amplifier circuit yields its inverse function.[70] → Figure 4.41 shows the basic setup required to generate an inverse function $f^{-1}(x)$ using a function generator $f(y)$ and an *open amplifier*.[71]

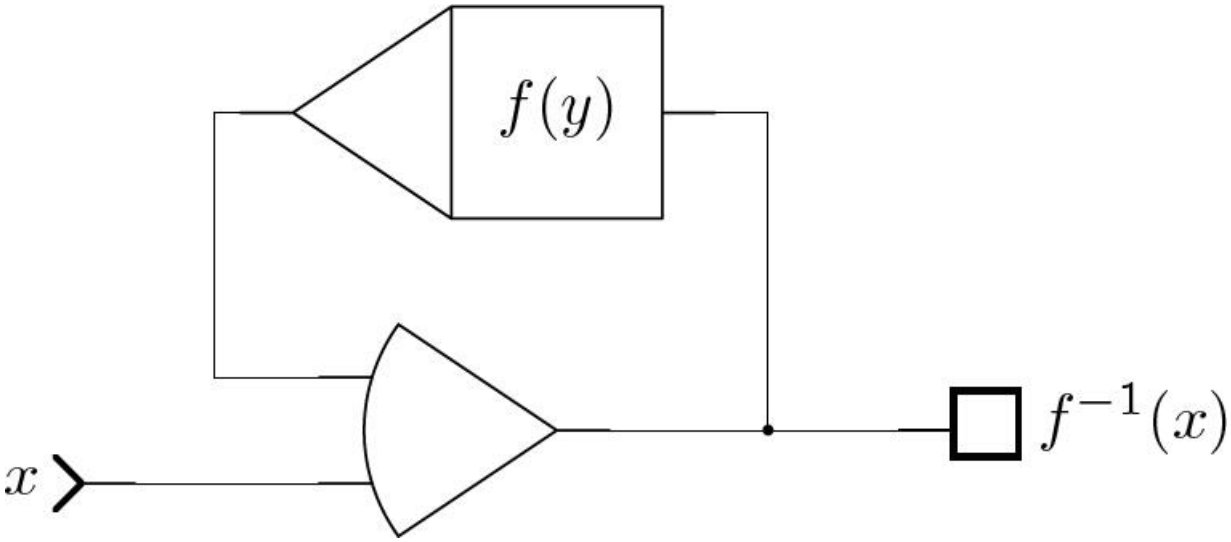


Figure 4.41. Generating an inverse function using an open amplifier

4.5.8 Functions of two variables

Generating functions of more than one variable, the case with two variables being the most common, is difficult using only analog electronics.[72] Some approaches to solve this problem involve the use of three-dimensional cams[73] with appropriate servo motor circuits. Another technique is based vaguely on the curve follower idea: An *xy*-plotter with its pen replaced by a pickup stylus is controlled by its *x*- and *y*-inputs moving a pickup over a specially prepared conductive paper or plastic sheet. On this sheet equipotential contour lines are drawn with conductive ink. These lines are then each connected to adjustable voltage sources. The conductive sheet then performs a two-dimensional linear interpolation between these equipotential lines.

A variation of that scheme is the electrolytic tank. In the simplest case it is just a rectangular bath filled with a conductive fluid. A pickup electrode is then moved through this trough by an *xy*-plotter. Using wires embedded in the bath, equipotential lines can be implemented as above. Even more complicated functions can be represented using a non-flat bottom plate. In some cases even a

three-dimensional movement of the pickup electrode has been employed yielding functions of the form $f(x, y, z)$.

Similar to the mechanical barrel cam[74] is the *contour map* function generator shown in →figure 4.42.[75] The function is represented by a three-dimensional "landscape", which is probed by a tip being moved in x- and y-direction as controlled by the two function arguments. The actual height of the tip touching the contour then corresponds to the desired function value $f(x, y)$.

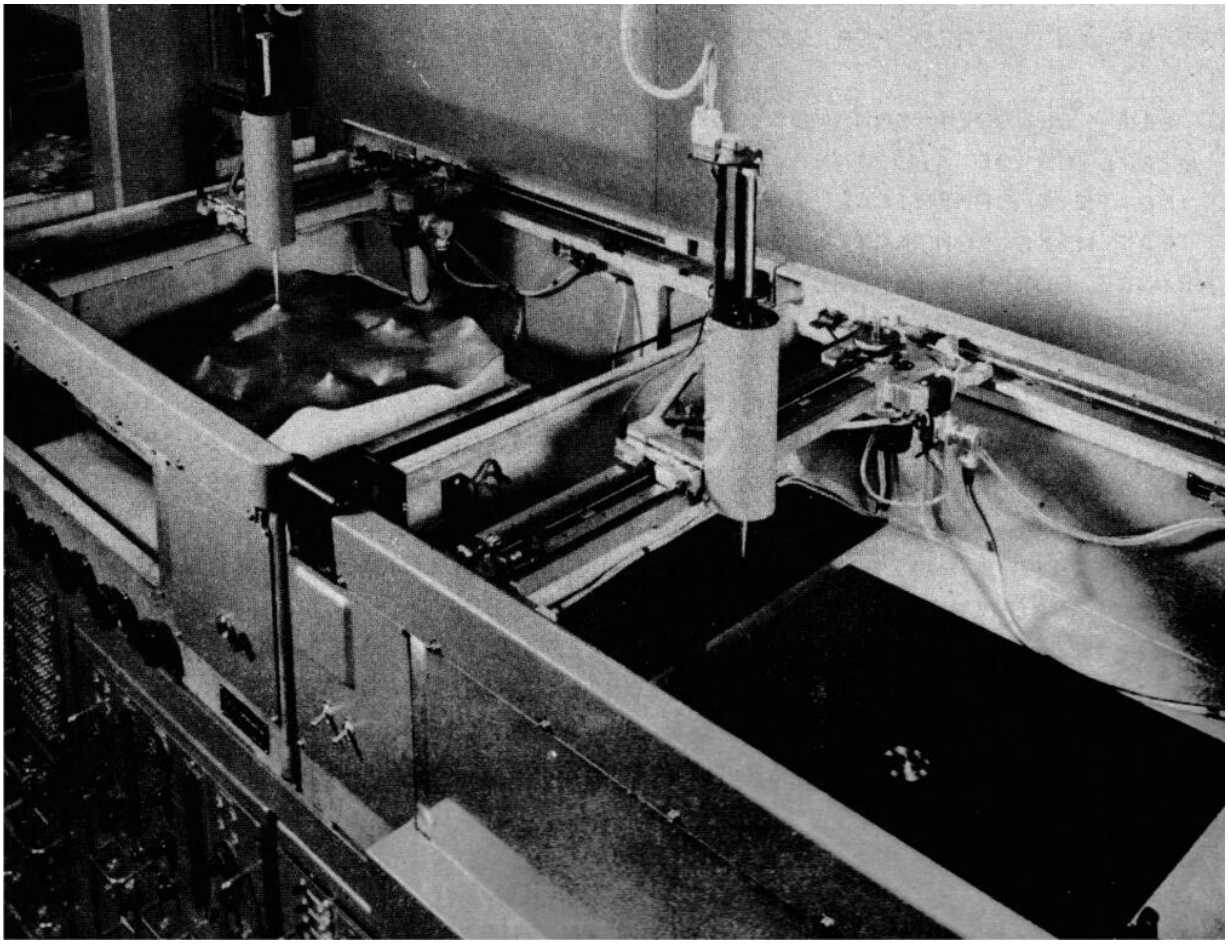


Figure 4.42. Contour map function generator

Yet another possibility is to use a couple of function generators of one variable and feed a tapped potentiometer function generator with their respective outputs, thus linearly interpolating between the

outputs of the individual function generators. In cases where the response times achievable with these servo-based techniques is not sufficient, the use of an oscilloscope tube, a translucent mask, and a photomultiplier has been proposed. The main problem here is the manufacturing of a precise translucent mask representing the function $f(x, y)$ by means of the extinction of light.

4.5.9 Function generation on hybrid computers

Hybrid computers offer interesting possibilities for the generation of arbitrary functions. A typical example for this are EAI's *digitally controlled function generator*s (*DCFG*s) used in the EAI 680 and later systems. These function generators are quite similar to diode based function generators but can be configured under program control from the attached digital computer in a hybrid system and offer up to sixteen support points each. The function is interpolated linearly between support points. A rather remarkable feature of these DCFGs is that they can also be used for the implementation of hysteresis functions.[76] DCFGs can even be coupled in case more support points are required for a certain function.

A problem frequently encountered in many branches of science and engineering is the determination of parameters for a set of coupled DEQs so that their solution matches certain values at given points in time. Using a function generator like a DCFG, this process can be guided by determining an error function which is obtained by subtracting the solution of the set of DEQs over time from the desired solution function implemented on the function generator. This error function may then steer the process of parameter determination in a hybrid computer setup.

In many cases functions of several variables $f(x_0, x_1, \ldots)$ are required which is notoriously difficult to implement on a purely analog system. Such functions arise in many areas including flight simulation of aircraft and rockets, the simulation of nuclear reactors, and many more.[77]

A straightforward approach to this would be to offload the actual function generation to the digital part of a hybrid computer which is typically implemented by table lookups. As simple as this is, it has a number of drawbacks, phase shift being the most important one since the digital computer has to actively perform memory requests under control of an interrupt system. If interpolation between support points is required, this will put an additional burden onto the digital computer increasing its response time and thus potentially slowing down the analog computation.

Accordingly, there were and are systems for the generation of multivariate functions using dedicated memory systems with attached *digital-analog-converter*s (*DAC*s) and *analog-digital-converter*s (*ADC*s). These can be loaded with the desired function values by the digital computer and then perform the function generation and even the interpolation if necessary in a stand-alone fashion, offloading the digital computer considerably and reducing phase shift.

However, the amount of memory required increases rapidly with the resolution and number of input variables. A function $f(x, y, z)$ with 10 bits of resolution for each of its input variables already requires $2^{3 \cdot 10} \approx 10^9$ memory cells which is unsuitable even today. Accordingly, input variables are typically restricted to rather low resolutions such as four or five bits only.

[→ Kleinert 1979/2] describes an interesting technique of “time sharing” nonlinear computing elements in an analog computer setup using sample-and-hold (*SAH*) elements[78] under control of either a digital control unit or the digital computer in a hybrid system. Given sufficiently large bandwidth of the nonlinearities required the actual sharing process can be performed with many kHz. Using this technique, nonlinearities, including function generators, can be used in several places in an analog computer setup.

4.6 Multiplication

The seemingly trivial operation of multiplying two time varying variables is quite complicated to implement. Accordingly, a plethora of different approaches were suggested and used in the past to build usable multipliers for analog computers.[79] Since multiplication can be seen as a function $f(x, y) = xy$ of two arguments, many of the ideas described in the preceding section can be used to implement this operation.[80]

4.6.1 Servo multipliers

An obvious way to implement a multiplier is based on a servo function generator driving a number of linearly wound potentiometers mounted on a common motor driven shaft. The input variable to control the servo circuit is the multiplier x while the ends of the free potentiometers, i. e., those that are not needed for the servo-feedback, are connected to the multiplicands $+y_i$ and $-y_i$.[81] If x and y can take on values in the interval $[-1, 1]$, the multiplier is said to be capable of *four-quadrant operation*.

→ Figure 4.43 shows a five channel servo multiplier made by Solartron: The left half of the device houses the power supply transformers.[82] The power output transistors of the servo circuit driving the motor are mounted on the two large black heat sinks visible in the upper middle of the picture.



Figure 4.43. Solartron servo multiplier

At the heart of the servo multiplier are the six precision potentiometers mounted on a common shaft, visible on the right of the picture. One of these potentiometers is used to generate the feedback signal for the servo circuit while the remaining five potentiometers are available for multiplication. Thus, with one input x, this servo multiplier can generate five products xy_1, xy_2, … , xy_5.

Apart from their mechanical complexity servo multipliers have the inherent drawback that the rate of change of the x is limited due to the inertias involved. If x changes too rapidly, the servo circuit will be unable to follow, introducing errors in the products. Nevertheless, in many applications this restriction turns out to be not a severe one – the advantage of generating a number of products based on a common multiplier often outweighs it. Additionally, the rate of change of the y_i is not limited.

4.6.2 Crossed-fields electron-beam multiplier

One of the first all electronic multiplication schemes is the *crossed-fields electron-beam multiplier* described by [→ Macnee 1948, Sec. IIIA].

The central element of this type of multiplier is an oscilloscope tube with its *x*- and *y*-deflection plates and an additional deflection coil on the tube's neck.[83]

The *x*-deflection plate pair is controlled by the *x*-input voltage causing the electron beam to move horizontally with a velocity $|v_x| \sim x$. The *y* input signal drives a power amplifier which in turn feeds the deflection coil wound around the neck of the tube. This coil generates a magnetic field $B \sim y$, thus deflecting the electron beam in the vertical direction by an amount xy.

The screen of the display tube is divided into an upper and lower half by an opaque mask. Using a photodetector,[84] a servo circuit like the one outlined in →section 4.5.3 can now be setup. This circuit drives the *y*-deflection plates of the oscilloscope tube in such a way that the spot of light will always stay on the edge of the opaque mask. Thus, the output voltage of the servo circuit is in proportion to xy.

4.6.3 Hyperbolic field multiplier

An even more arcane multiplication scheme based on a very special vacuum tube is the *hyperbolic field multiplier*, which was invented at the university of Darmstadt by Friedrich-Wilhelm Gundlach[85] in the early 1950s as a high speed multiplier.[86]

→Figure 4.44 shows the structure of this hyperbolic field tube. It has three sets of deflection plates – from left to right these are a vertical deflection plate pair, four plates generating a hyperbolic electrostatic field, and another pair of deflection plates to deflect the electron beam horizontally. On the far right are two target plates connected to a difference amplifier.

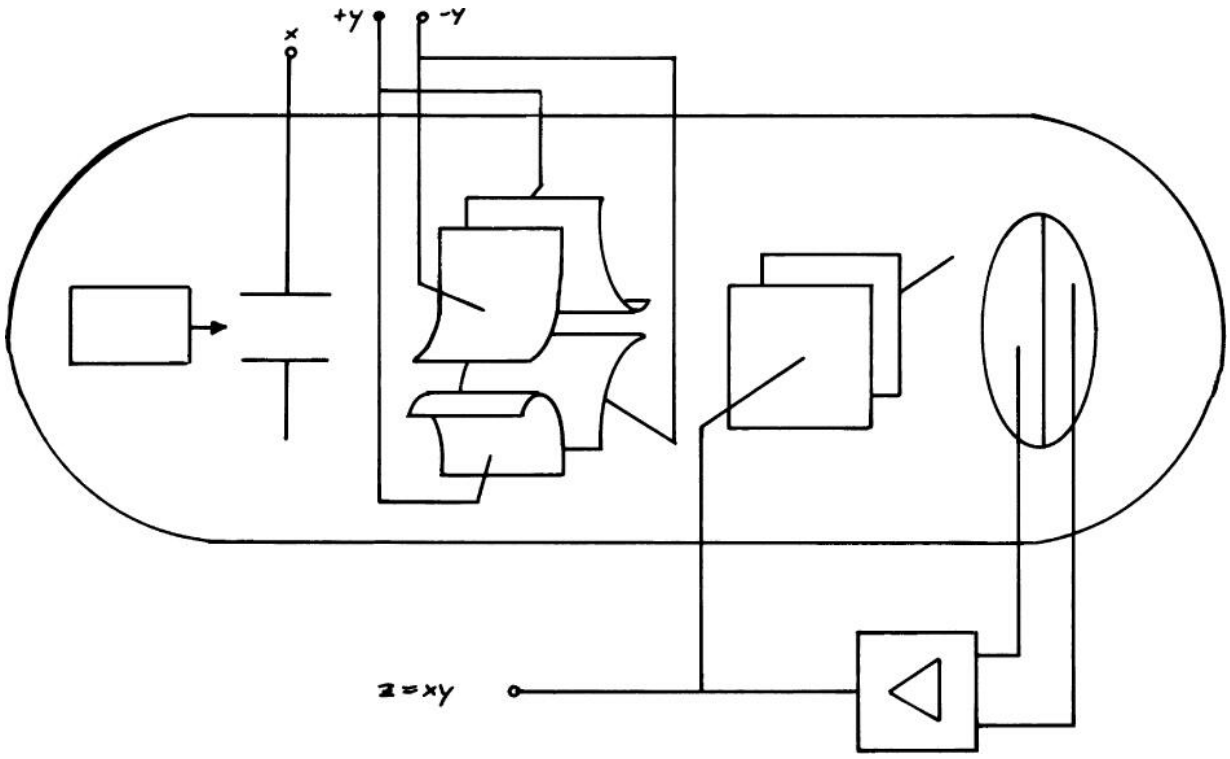


Figure 4.44. Structure of a hyperbolic field tube (cf. [→ HAUG 1960, p. A7])

The vertical deflection plates are driven by the multiplier input x while the multiplicand y is fed in a crosswise fashion to the hyperbolic plates. The field created by these is proportional to $y\eta\xi$ with η and ξ being the coordinates of a coordinate system with its origin in the middle of these four deflection plates.[87]

An electron beam, which has not been deflected by the vertical deflection plates, will cross the hyperbolic field undisturbed since $\eta = \xi = 0$. If the beam has been deflected due to $x \neq 0$, it experiences an additional deflection due to this hyperbolic field resulting in a additional horizontal deflection proportional to xy.[88]

The electron beam now hits the target plates causing a tiny voltage which is amplified by the difference amplifier connected to these plates. The output voltage of this amplifier drives the horizontal deflection plates so that the electron beam will always be

exactly in the middle of the two target plates, so this voltage will be *xy*.

This type of multiplier is capable of operation at high frequencies with respect to both input voltages *x* and *y* due to its purely electrostatic deflection system. [→HAUG 1960, p. A7] mentions a dynamic multiplication error of only about 1% at signal frequencies of 80 kHz – a remarkable value for its time. The complete schematic of a multiplier based on this special tube can be found in [→DHEN 1960, p. 31].[89] →Figure 4.45 shows the only known surviving hyperbolic field tube.[90]

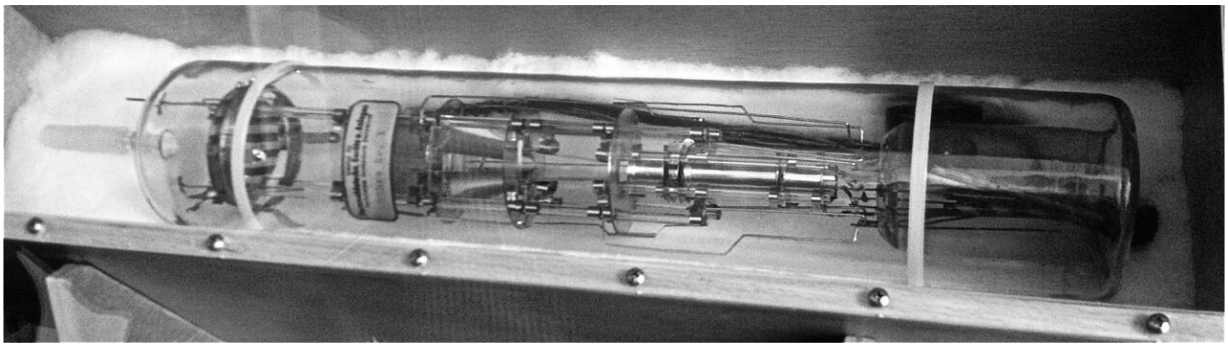


Figure 4.45. Hyperbola tube

4.6.4 Other multiplication tubes

The 1950s saw several other vacuum tube based approaches to multiplication apart from the hyperbolic field tube. One of these devices was developed by ALEXANDER SOMERVILLE. →Figure 4.46 shows the principle of operation. An electron beam with square cross section is deflected by two sets of deflection plates in *x* and *y* direction before hitting a target consisting of four square collector plates. The current measured by each of the meters $I_{1...4}$ depends on the position of the deflected electron beam.

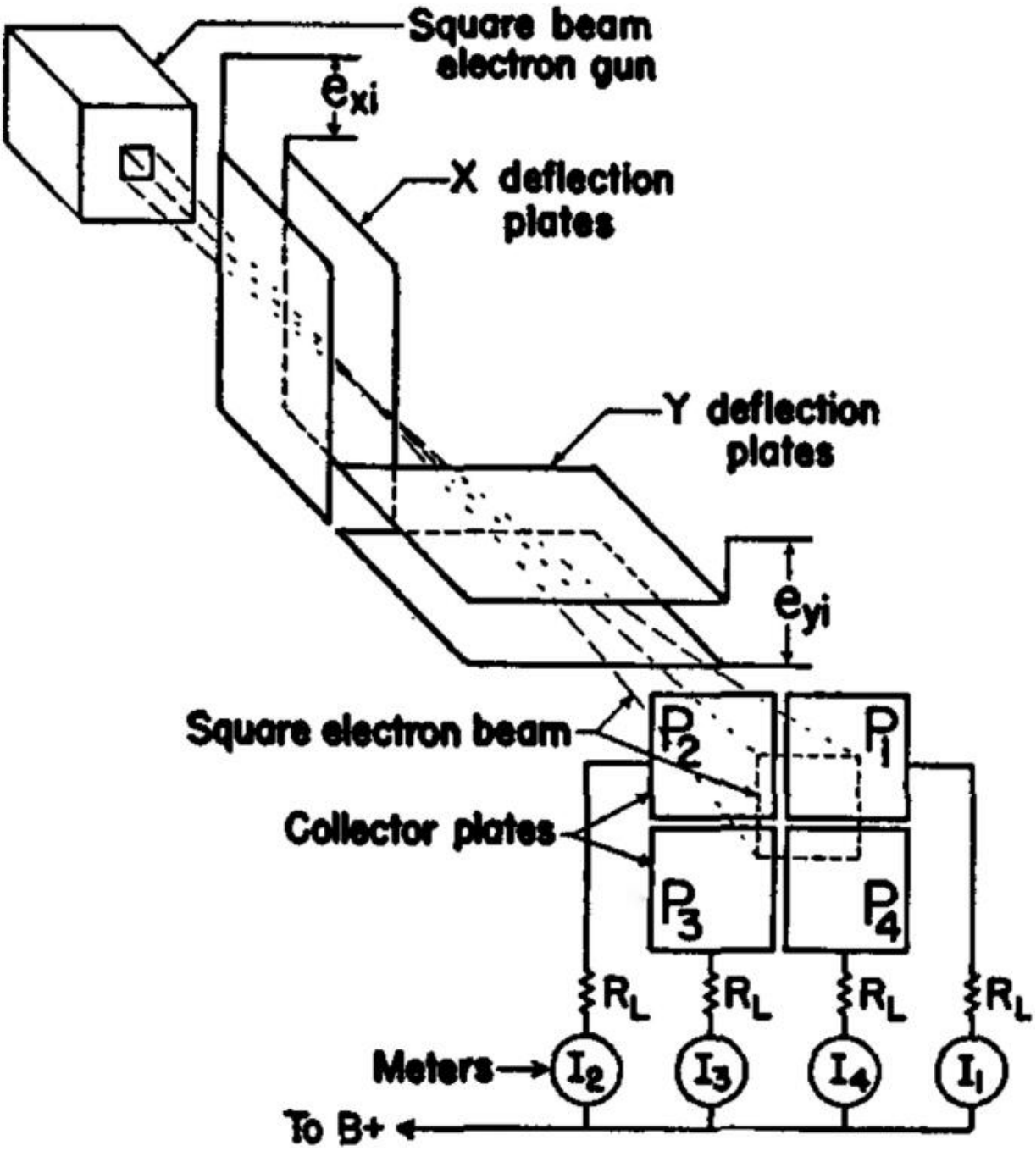


Figure 4.46. Electron beam multiplication tube (see [→ SOMERVILLE 1950, p. 145])

If the beam is not deflected, i. e., $x = y = 0$, it hits the four plates $P_{1...4}$ in a symmetrical way and the currents, taking the different signs of the plates into account, cancel out. If, for example, $x = y = 1$, current

will only flow through P_1 corresponding to the product $xy = 1$, etc. This basic idea can be further improved by employing feedback circuitry to linearize the multiplier.[91]

A similar device developed by E. J. Angelo in 1952 had a bandwidth of 70 kHz and a precision of about 2% of full-scale output. This device used an electron beam with a circular cross section and four square target plates.

4.6.5 Time division multipliers

Another approach to multiplication is the *time division multiplier*,[92] which is a particular implementation of a larger family of *modulation multipliers*.

As described before, a servo multiplier is essentially just a voltage-controlled voltage divider using potentiometers to yield the desired product values. Voltage division can also be achieved by modulating a square wave signal. The amplitude of this signal is controlled by the y input, while x controls the duty cycle.[93] $x = \pm 1$ corresponds to a signal being constantly at the value $\pm y$ while $x = 0$ would result in a duty cycle of $1 : 1$. Generating the average of this square wave signal by means of an RC low-pass filter yields an output voltage of xy.

As in the case of the servo multiplier, this scheme can be readily extended to the computation of multiple products with a common multiplier x. Accordingly, most time division multipliers contain one mark-space modulator circuit driving several amplitude modulators which are controlled by individual inputs y_i.[94] [→ Korn et al. 1954] describe the simple relay-based multiplier shown in →figure 4.47. The variable x is fed with opposite signs to the circuit and one SPDT relay contact switches between $+x$ and $-x$. This signal is fed to a ripple filter consisting of two T-filters, yielding the desired output value $x_o = \frac{xy}{2z}$.

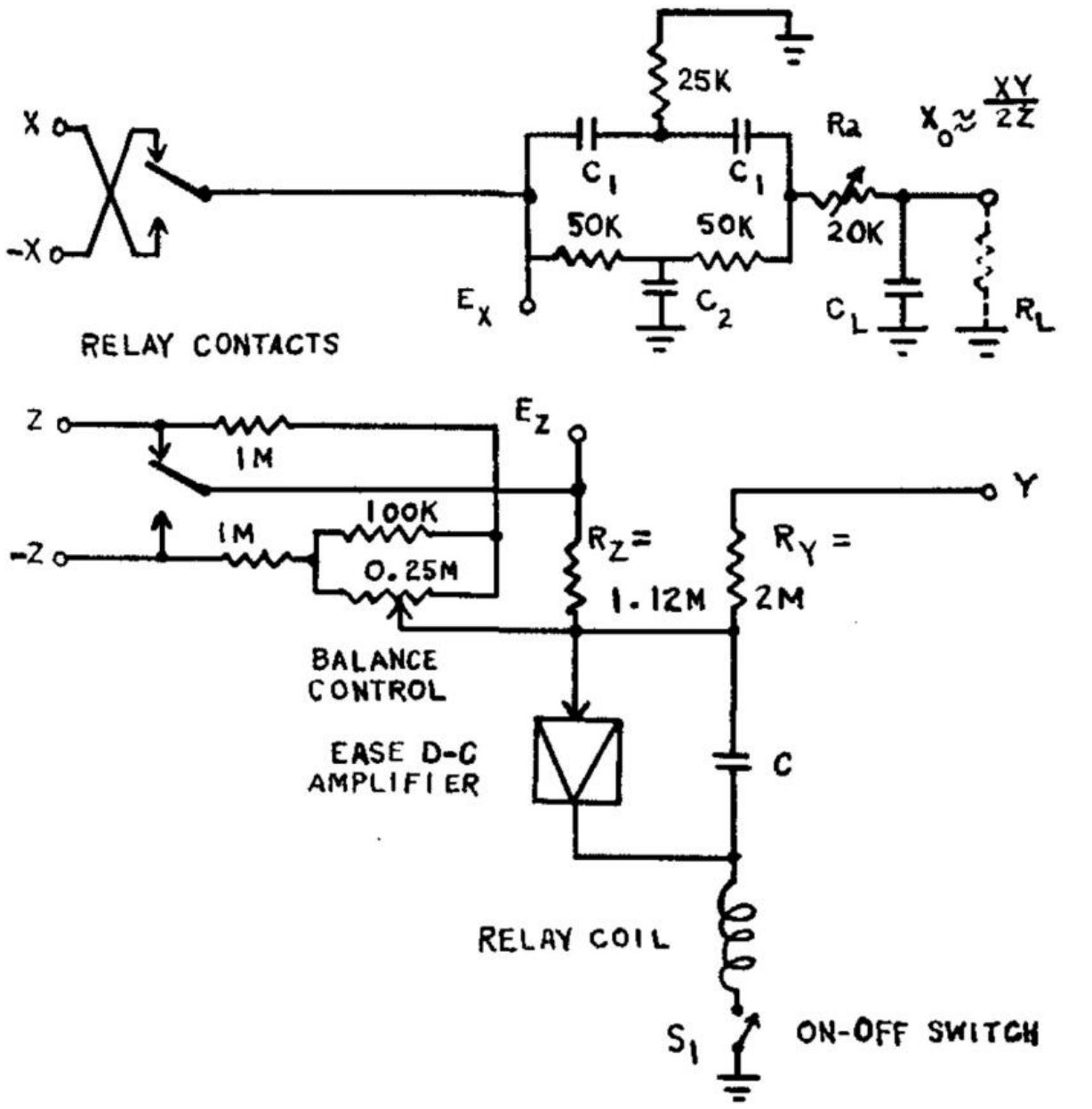


Figure 4.47. Relay based time-division multiplier (see [→ KORN et al. 1954, p. 980])

At the heart of the circuit is an integrator consisting of a DC coupled amplifier with two inputs and a capacitor in its feedback path. The output of this integrator drives the relay. One of the integrator inputs is connected to the term *y* while the other input E_z is connected to +*z* and −*z* by means of a second SPDT relay contact. *z* is

typically the machine unit of the computer. If $y = 0$, the relay will toggle between $+z$ and $-z$ with a 50% duty cycle resulting in $x_o = 0$. When $y \neq 0$, it biases the integrator, which will in turn result in a different duty cycle of the relay operation so that the integrator input value E_z will cancel out y yielding the desired product value at x_o.

A setup like this has some drawbacks such as limited bandwidth, mainly determined by the maximum frequency at which the relay can operate reliably, and limited life time due to wear of the relay contacts.[95] A variation of this idea is the *triangle-integration* multiplier as described in [→ Hartmann et al. 1961].

[→ Pfeiffer 1959] describes an interesting variation of a time division multiplier using a triangular wave instead of a rectangular signal. A simple and rather recent implementation of a two-quadrant time division multiplier based on a sawtooth signal is described in [→ Selvam et al. 1995].

4.6.6 Logarithmic multipliers

An obvious implementation of a multiplier is based on adding logarithms to achieve a multiplication – the very same principle that slide rules depend on. →Figure 4.48 shows the basic setup of a *logarithmic multiplier* which uses three function generators, two yielding the logarithm of an argument, and one yielding the inverse logarithm.[96] An interesting linear-to-logarithmic circuit based on standard vacuum tubes, such as the dual triode 12AX7, can be found in [→ Howard et al. 1953] while [→ Savant et al. 1954] describe a practical multiplier based on this idea.

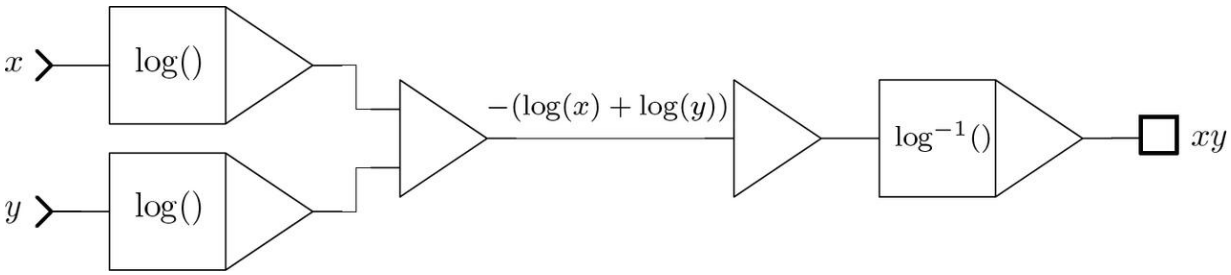


Figure 4.48. Logarithmic multiplier

The main problem of this type of multiplier is the fact that the two variables *x* and *y* are restricted to positive values unless special precautions are taken, such as generating absolute values and restoring the sign of the multiplication result.

4.6.7 Quarter square multipliers

By far the most widely used classic multiplier circuit is the *quarter square multiplier*. It is based on

$$xy = \frac{(x+y)^2-(x-y)^2}{4}$$

which only requires the computation of two square functions,[97] a difference, and a division by 4, which can be easily done with a voltage divider or a suitable feedback resistor of the output amplifier.[98] → Figure 4.49 shows the basic circuitry of such a multiplier.

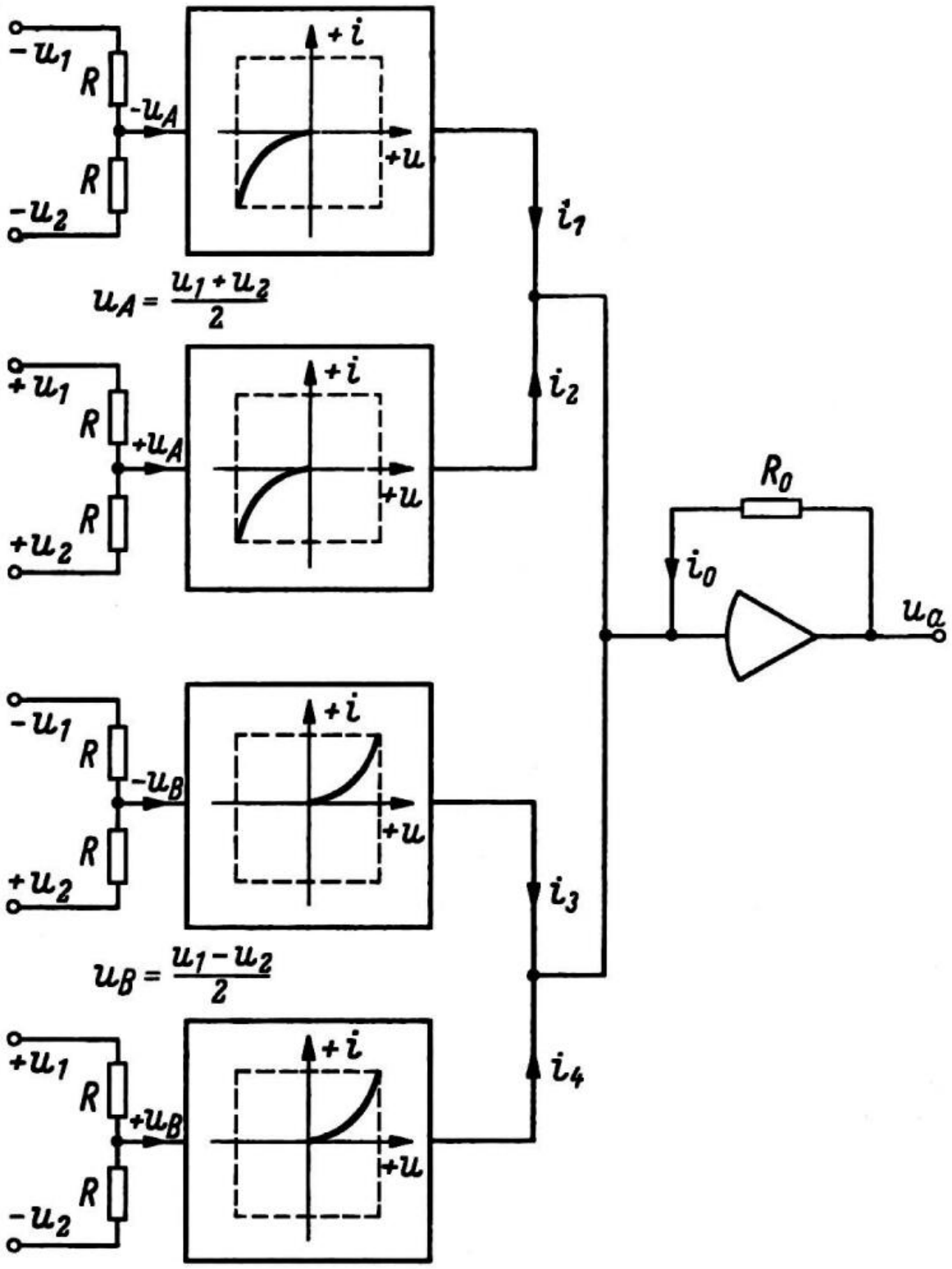


Figure 4.49. Principle of operation of a quarter square multiplier (cf. [→ Giloi et al. 1963, p. 92])

A particular requirement of this type of multiplier is that both variables, the multiplier as well as the multiplicand, have to be supplied with both signs, which typically is not a problem in most analog computer setups due to the sign changing behavior of summers and integrators. The main advantages of this multiplier are its high bandwidth as well as its relatively low price. The only drawback is that there is no way to generate multiple products of a common multiplier as is possible in the case of servo and time division multipliers.

In the late 1940s Raytheon started development of special nonlinear beam-deflection tubes for analog multiplication. The first of these tubes, the *QK-256*, used an electron beam with rectangular cross section that could be deflected before passing through a specially-shaped mask. Depending on the amount of deflection, the beam passing through this mask changed its cross section in a nonlinear way. Based on this tube, the *QK-329* was developed, which allowed the construction of very high bandwidth – up to 80 Mhz – quarter square multipliers.[99]

→ Figure 4.50 shows a typical quarter square multiplier based on two QK-329 tubes. The bandwidth of this setup is primarily limited by the operational amplifiers used in the input and output stages and not by the beam-deflection tube implementing the square function.

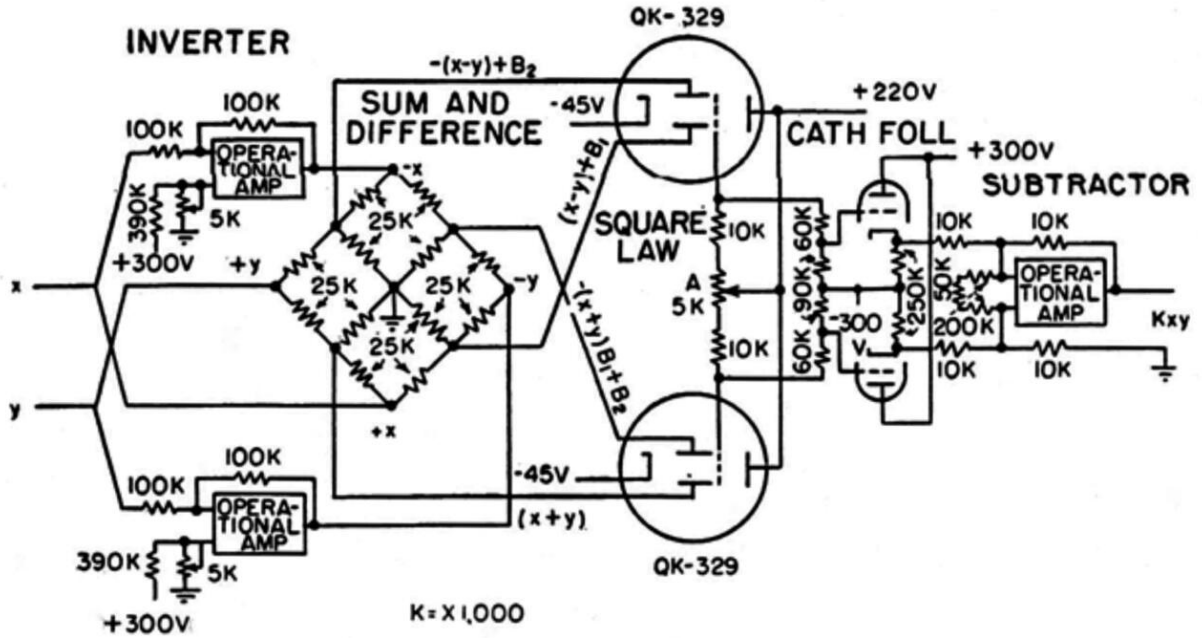


Figure 4.50. Quarter square multiplier based on two QK-329 beam-deflection tubes (see [→Miller et al. 1955, p. 163])

Quadratrons, as described in →section 4.5.4, were also used for low bandwidth (up to about 400 Hz[100]) and low precision quarter square multipliers.[101]

An interesting technique to decrease the errors of quarter square multipliers was applied in EAI's *SIMSTAR* system. Since this type of multiplier is dominated by absolute errors, it is desirable to perform an automatic rescaling of its input values *x* and *y*. Using electronic comparators these inputs can be scaled by a factor of 4 if they are smaller than $\frac{1}{4}$ of the machine unit range, while the multiplier output is automatically scaled down by factors of $\frac{1}{4}$ or $\frac{1}{16}$, respectively.[102]

[→Giloi et al. 1968] describe an interesting computer aided design scheme for the development of quarter square multipliers with ALGOL code. Using an iterative approach they could show that an approximation error as low as 0.03% can be obtained without requiring high precision components.

4.6.8 Other multiplication schemes

Over the years many other multiplications schemes have been developed and implemented. There are *dynamometer multipliers* balancing the forces of multiple electromagnets against each other[103], *strain-gauge multipliers* balancing a strain-gauge with voice coils,[104] *heat-transfer multipliers*,[105] *Hall effect multipliers*,[106] and many more.[107]

As an example, a strain-gauge multiplier is shown in →figure 4.51. It can be considered as a type of servo multiplier. At its heart are four strain gauges forming two separate bridge circuits with their mechanical centers attached to a common point which can be moved by means of a linear actuator. This actuator is controlled by an amplifier which nulls the sum of its *x* input and the output of the bridge circuit at the bottom of the schematic. The bridge at the top yields the desired product *xy*. This setup restricts its *x* input to rather low bandwidth signals (but higher than those in a classic servo multiplier using motors and potentiometers), while the *y* input can carry high bandwidth signals. Adding more bridge circuits to the setup, multiple products can be generated at once.

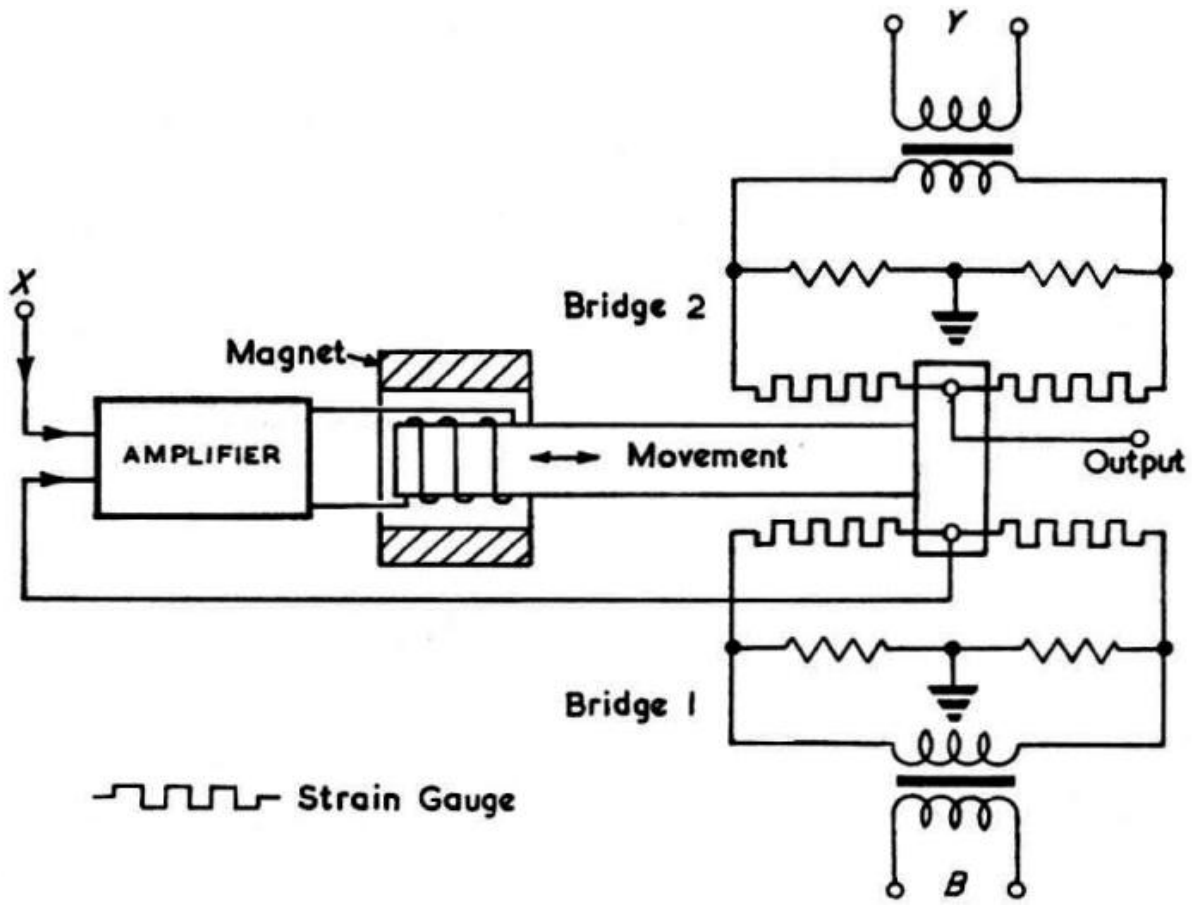


Figure 4.51. Structure of a strain-gauge multiplier (see [→ CZAJKOWSKI 1956/2, p. 352])

A rather unconventional approach to multiplication is the *coincidence multiplier* described in [→ TASINI 1957, pp. 26 ff.]. It is based on a probabilistic approach where the variables to be multiplied are either represented by random pulses, the densities of which represent the values, or by periodic wave forms with incommensurable frequencies. The desired product can then be obtained by a simple AND gate with multiple inputs.[108] As elegant as this approach looks at first glance, it is riddled with problems. It is difficult to generate suitable random or incommensurable signals in the first place. In addition to this, the precision obtained by this method is very low and the output signal is not as stable as required by most applications. Accordingly, this approach never caught on.

An interesting hybrid multiplier technique for analog computers using binary counters and very early digital-analog-converters is described in [→Goldberg 1951].

In the late 1960s analog transconductance multipliers were developed.[109] These are based on the effect that the gain of a transistor depends on its collector current, so that base and collector current can be multiplied by a differential transistor pair followed by a difference amplifier.

4.6.9 Hybrid multipliers

[→Rogers 1963, pp. 12] describes a hybrid multiplier operating on two values represented by a combination of an analog voltage and a binary value (x_A, x_D) and (y_A, y_D). The analog and digital parts are multiplied using a standard analog multiplier and a binary multiplier feeding an ADC. The partial products $x_A y_D$ and $x_D y_A$ feed multiplying ADCs. The resulting four terms are fed into a summer yielding the desired output. The basic idea is to extend the range of the multiplier, thus simplifying scaling.

[→Waldschmidt 1970, pp. 22 ff.] describes a high precision hybrid multiplier based on a Taylor approximation

$$f(x,y) = xy = x_i y_i + y_i \Delta x + x_i \Delta y + \Delta x \Delta y.$$

x and y are analog values which are fed to ADCs yielding a mixed value representation not too dissimilar to the idea described above.[110]

4.6.10 Modern multipliers

Modern analog multipliers are typically based on *Gilbert cells*, also known as *translinear circuit* or *transconductance technique*.[111] It was developed by Barrie Gilbert[112] in the late 1960s[113] and quickly rendered other multiplication circuits and techniques obsolete due to its simplicity and the possibility to implement a complete

multiplier in a single integrated circuit. A typical circuit is shown in →figure 4.52. The two arguments *x* and *y* are fed to the bases of transistors Q_A, Q_B, and Q_C, Q_D, respectively. The actual multiplication is performed using the transistors Q1, Q2 (connected as diodes), Q3, Q4, Q5, and Q6.[114]

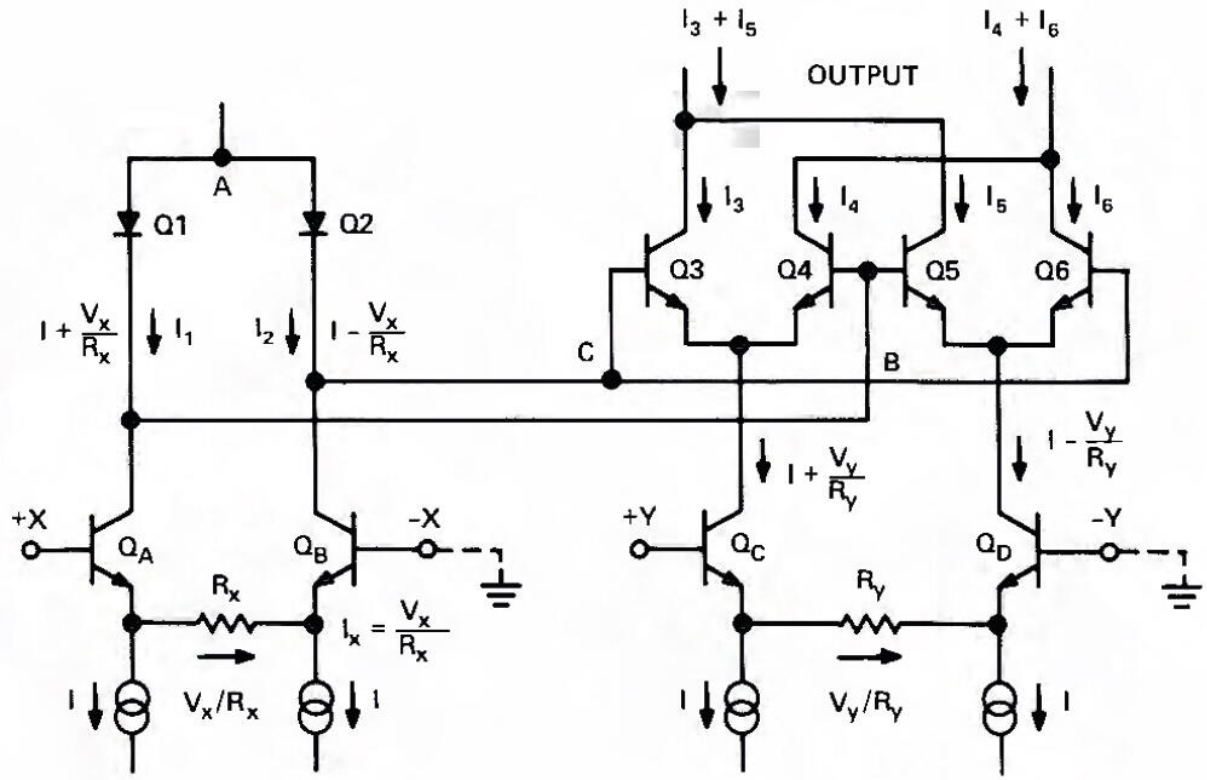


Figure 4.52. Basic structure of a variable transconductance multiplier (see [→Sheingold 1978, p. 38])

More recent developments are *FGMOS* (*floating gate metal oxide semiconductor*) transistor based multipliers as described in [→Keleş et al. 2009] and a novel high precision wide bandwidth *CMOS* (*complementary metal oxide semiconductor*) multiplier as described in [→Heimann et al. 2021].

4.7 Division and square root

Division and square root are normally implemented in a straightforward fashion using a multiplication circuit and an open amplifier to generate the inverse function of multiplication. → Figure 4.53 shows how a division operation can be implemented using this technique by employing a multiplier[115] in the feedback loop of an open amplifier. Connecting the *y* input of the multiplier also to the output of the amplifier sets up a square root function.

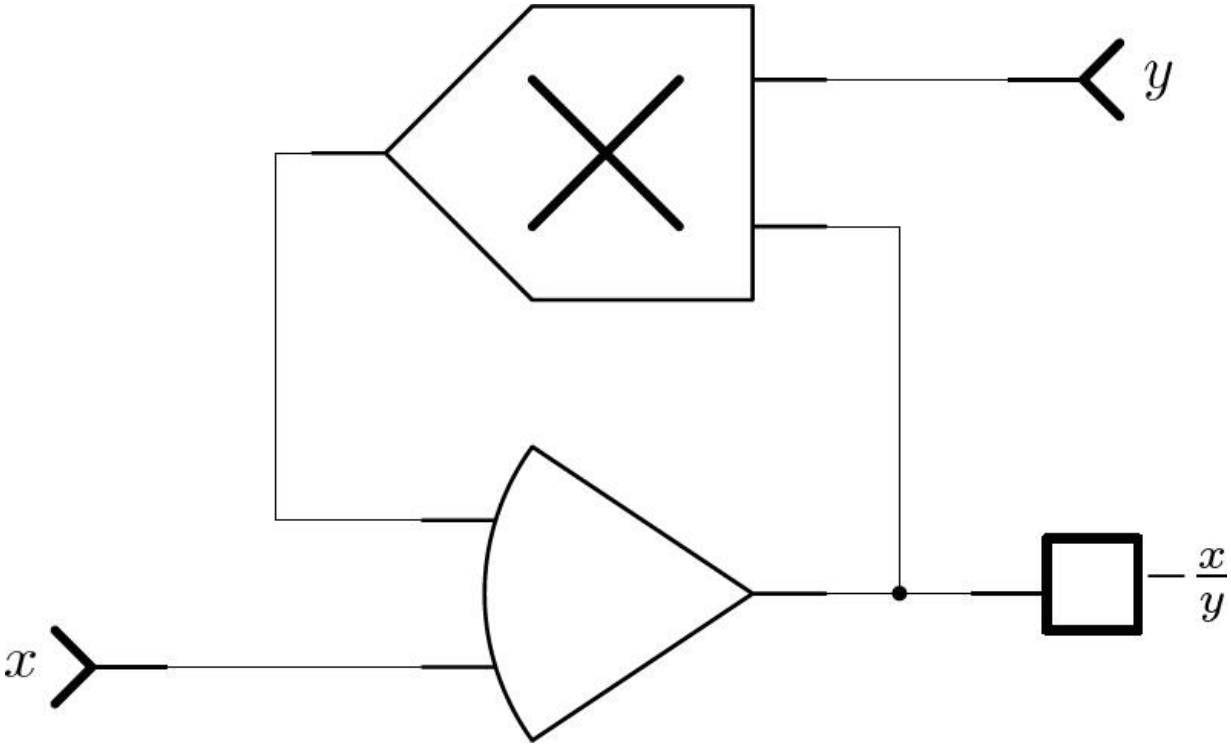


Figure 4.53. Division as inverse function of multiplication

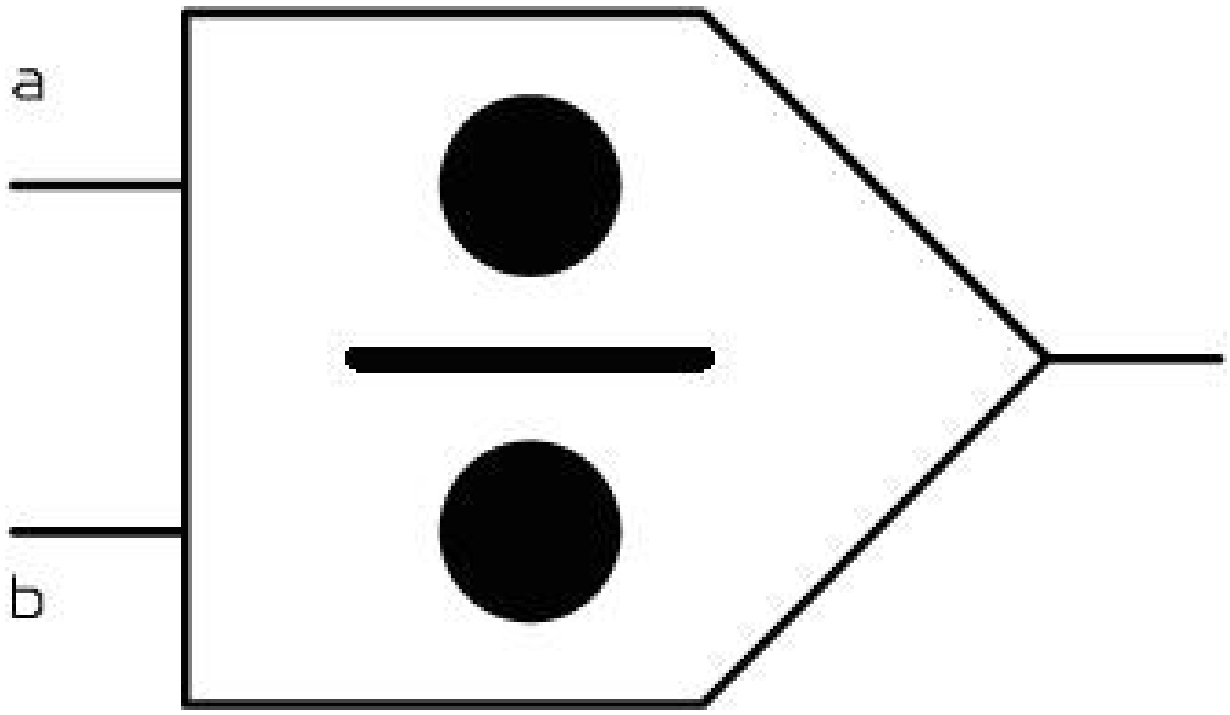


Figure 4.54. Symbol for a divider

It should be noted that circuits using open amplifiers with active elements in the feedback path may tend to unstable operation, i. e., oscillation. To stabilize such circuits it is common to connect a small capacitor in the range of several 10 pF between the summing junction and the output of the amplifier.

4.8 Comparators

Some problems require a means of implementing step functions or to switch a variable from one value to another, etc. This is typically done using a *comparator* in conjunction with analog switches in an analog computer. Basically, a comparator is just an operational amplifier with biased diodes in its feedback loop thus limiting its output typically to two values such as +1 and −1.

→ Figure 4.55 shows the general principle of the operation of a comparator.[116] At its heart is an amplifier featuring typically two inputs with resistors R_1 and R_2.[117] The feedback loop of this amplifier

consists of two diodes biased by two voltage sources e_{b1} and e_{b2} depicted by batteries in this schematic.[118]

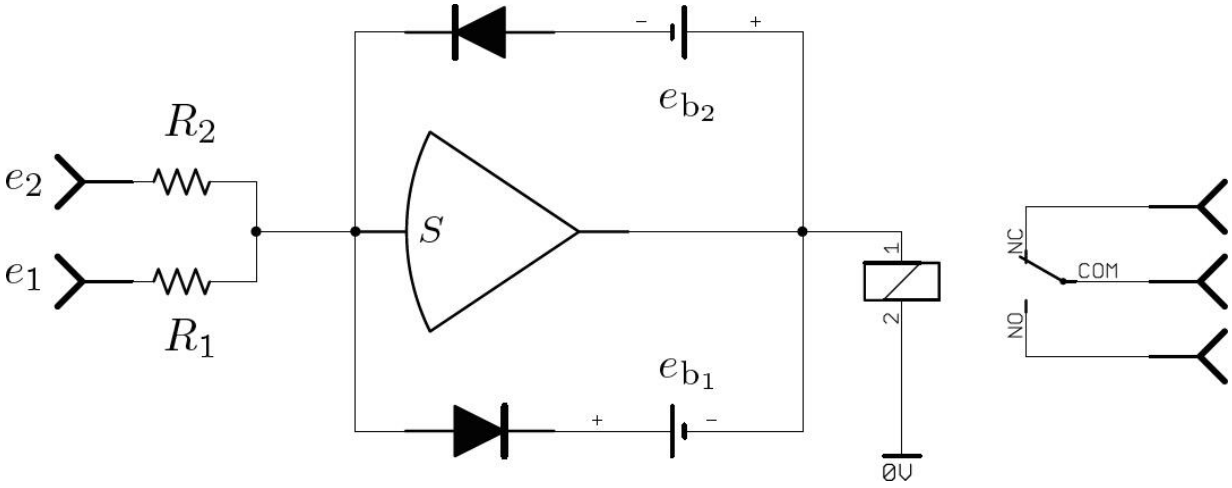


Figure 4.55. Principle of operation of a classic comparator

In the simplest case the output of the amplifier controls a relay. If the sum of the two input signals e_1 and e_2 is greater than zero, the output will be driven to its lower limit while the upper limit will be reached for a sum less than zero. Thus, a comparator can be used to compare the voltages of two (or more) signals.

Small early tabletop computers often did not provide dedicated comparators – the operational amplifiers were too precious to be committed to a specific function like this. Instead their patch panels often featured a couple of free diodes which can be used to setup a comparator based on an open amplifier. Comparators directly driving a relay are often called *relay comparators*.

Larger and more complex analog computers typically contain a number of comparators which do not control a relay directly. Instead the comparators yield a digital output signal which can be used to control a relay driver or to signal certain events to a connected digital computer, etc. Such comparators are typically called *electronic comparators*. →Figures 4.56 and →4.57 show the

symbols for a relay comparator and an electronic comparator yielding a logic output signal, respectively.

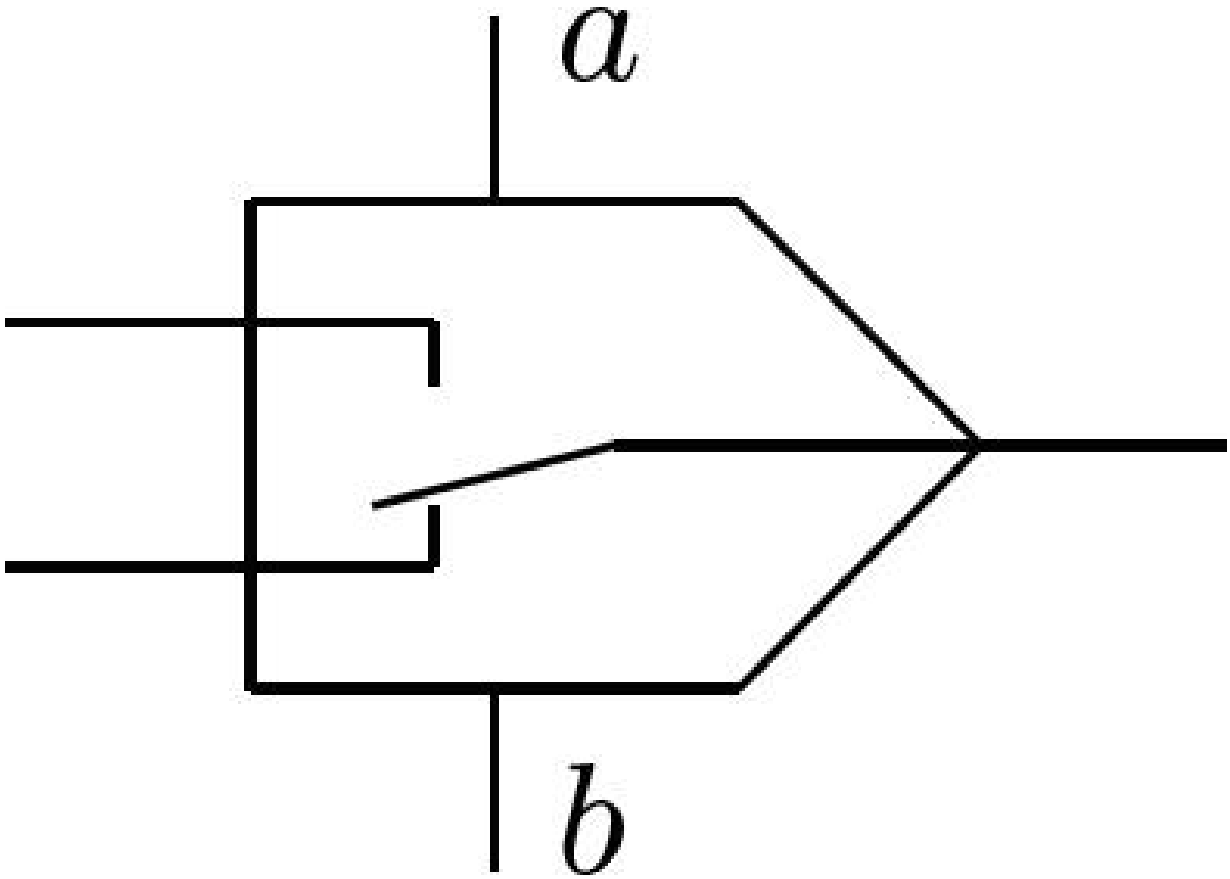


Figure 4.56. Symbol of a relay comparator

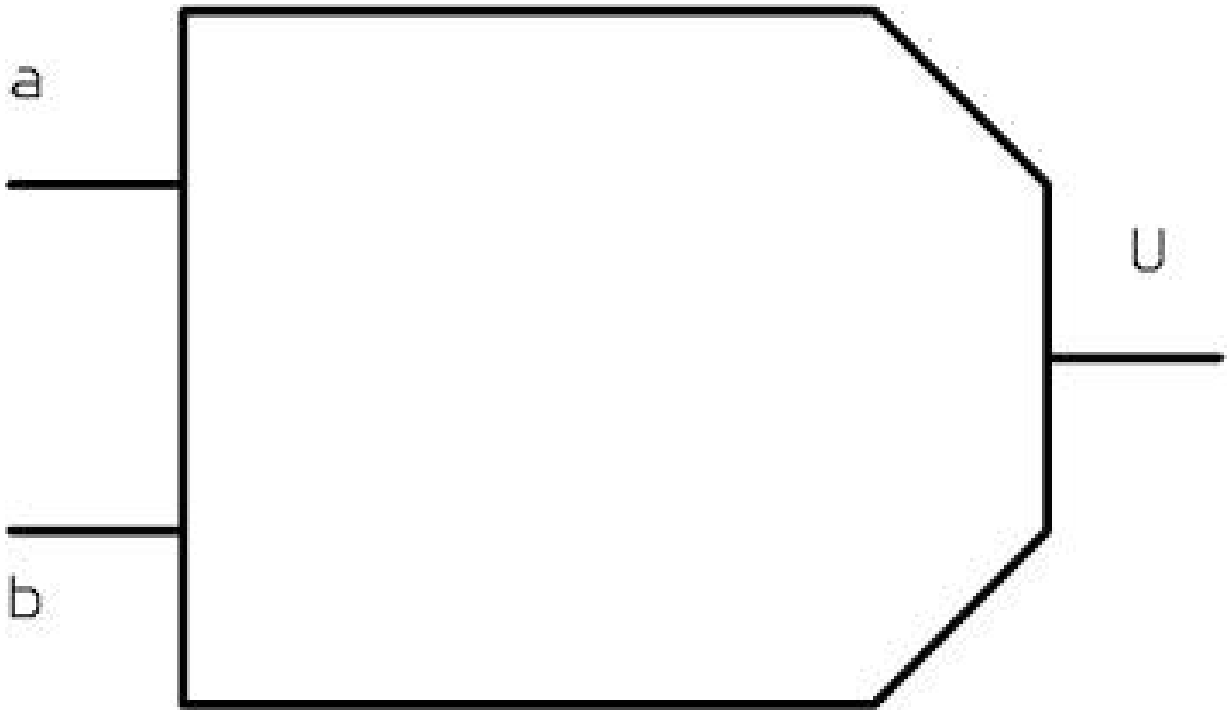


Figure 4.57. Symbol of a comparator

4.9 Limiters

Limiters are used to limit the value of a variable. This is often necessary in simulations of mechanical systems where moving parts can hit stops, etc. The basic circuit of a limiter is similar to that of the amplifier part in a relay comparator. By using appropriately biased diodes forming a feedback path of the operational amplifier, its output voltage can be limited by an upper and lower bound.
→ Figure 4.58 shows the structure of a basic limiter.[119]

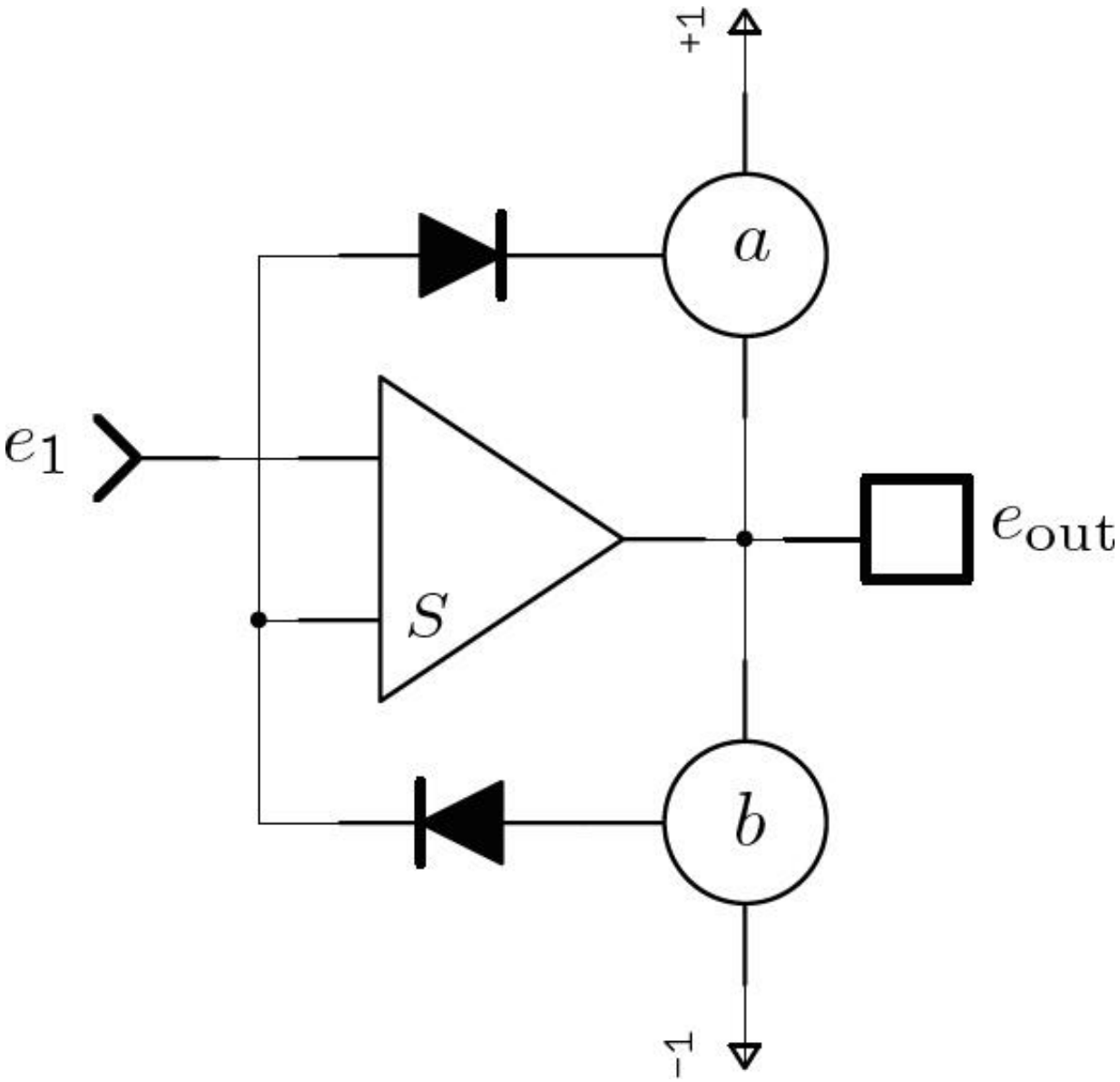


Figure 4.58. Simple limiter circuit

4.10 Resolvers

Resolvers are among the most complex individual computing elements in an analog computer. Typically only large systems contain resolvers as stand-alone units. The purpose of a resolver is the conversion of polar to rectangular coordinates and vice versa. In addition to this, a resolver can also yield the sine and cosine values

of a given input angle. Resolvers also allow the rotation of coordinate systems, which is an especially useful operation in aerospace related problems.

One of the earliest resolvers described in literature is the device developed by Lovell et al.[120] This device is a typical *servo resolver* – this type of resolver was used widely in aerospace applications where coordinate transformations are abundant and the rates of change of the variables to be transformed are normally sufficiently slow to be suitable for an electromechanical approach. Later analog computers also featured all electronic resolvers allowing high-speed operation.

Resolvers are quite complicated and there are too many implementation variants to be described in detail here.[121]

4.11 Time delay

A variety of problems contain structures causing time delays, such as mass flow through pipelines, the transport of pharmaceutical drugs between compartments, etc. Accordingly, time delay units are required for modeling such problems on an analog computer. An ideal time delay unit would act as shown in → figure 4.59: An input signal is delayed by a variable amount Δt. Given an input signal $y(t)$, the output $e_o(t)$ of the time delay unit is then $e_o(t) = y(t - \Delta t)$.[122]

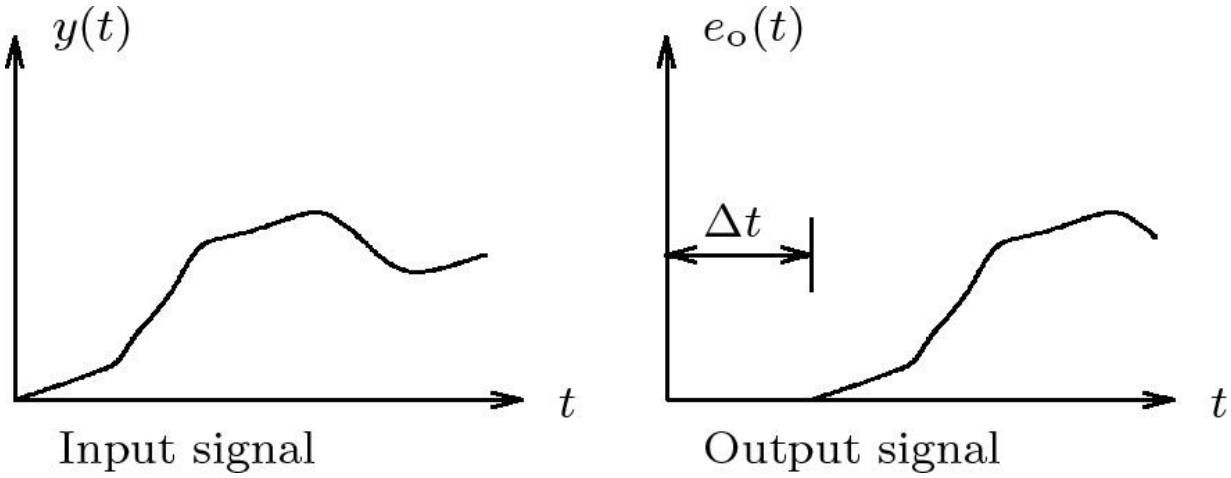


Figure 4.59. Behavior of an ideal time delay

There have been many different approaches to implement time delay functions for use in an analog computer, the *capacitor wheel* as shown in →figure 4.60 being one of the earliest approaches. It consists of a large number of capacitors used as analog storage elements and two stepping switches labeled 1 and 2 running synchronously, implementing a basic sample and hold circuit. The first switch selects a capacitor to be charged while the second selects a different one to be read out. Using the structure 3 the delay can be set in multiples of the stepping frequency. A modern variant of this approach is the *bucket brigade* circuit consisting of alternating pairs of transistor structures, each connected to an associated capacitor.[123]

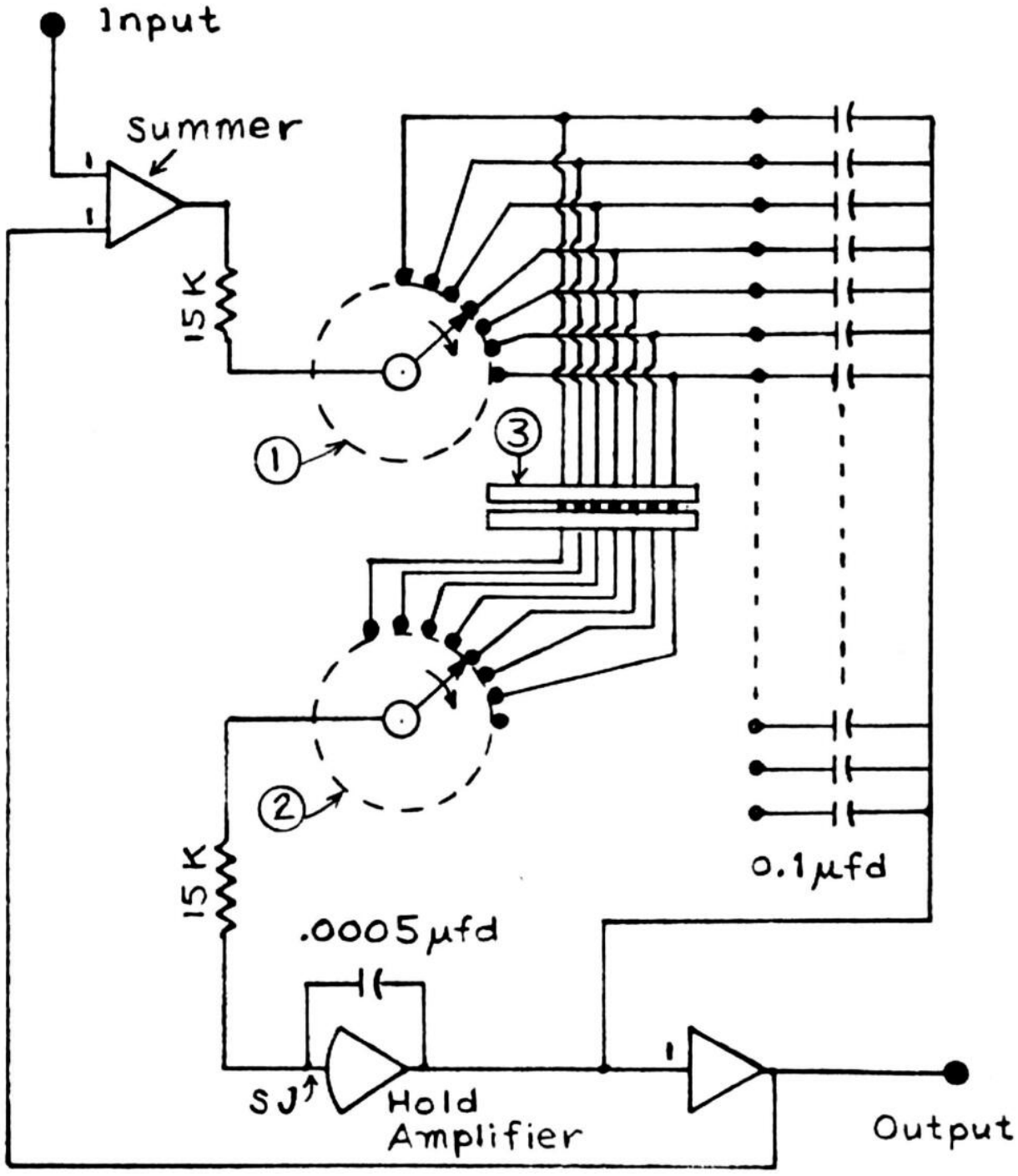


Figure 4.60. Time delay circuit according to [→THOMAS et al. 1969, p. 638]

An alternative implementation of such a sampling time delay consists of a chain of integrators which are cycled through initial

condition and halt mode in an alternating fashion. Each integrator has its initial condition input connected to the output of its predecessor so that a sampled voltage propagates through this chain in a stepwise fashion.

The main disadvantage of such systems is the very limited number of storage elements, effectively limiting the number of points where the function to be delayed is sampled.[124]

The most basic idea from today's point of view is to digitize the input signal with an ADC, store the result in a digital memory system, and read it out with the desired delay time Δt. The digital values read from memory are then fed to a *DAC*. Although this yields very good results regardless of the structure of the input signal,[125] this approach became feasible only in the late 1970s with the advent of cheap *random access memory* integrated circuits (*RAM*). → Figure 4.61 shows a typical digital time delay unit made by EAI in the 1970s.

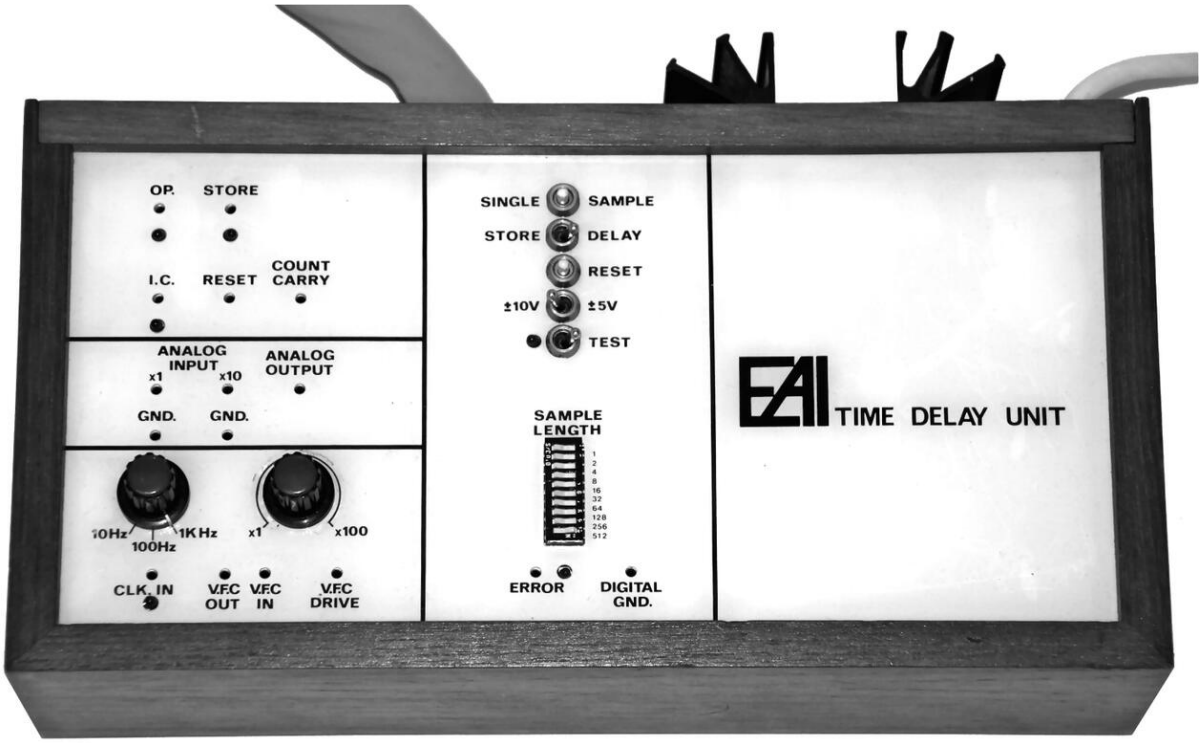


Figure 4.61. EAI time delay unit

All of these approaches discretize the signal to be delayed, thus introducing additional and sometimes unwanted effects in a program. A different approach to time delay operating in a continuous fashion is shown in →figure 4.62. This EAI made system, the *Simulag Unit*, uses a magnetic tape transport with separate write and read heads spaced apart by some fixed distance *d*. An analog input signal is stored on tape using a pulse width modulation technique and read back after a time $t = \frac{d}{v}$ where *v* denotes the velocity of the tape. *d* can be set manually or controlled by an external signal making variable time delays possible.[126]

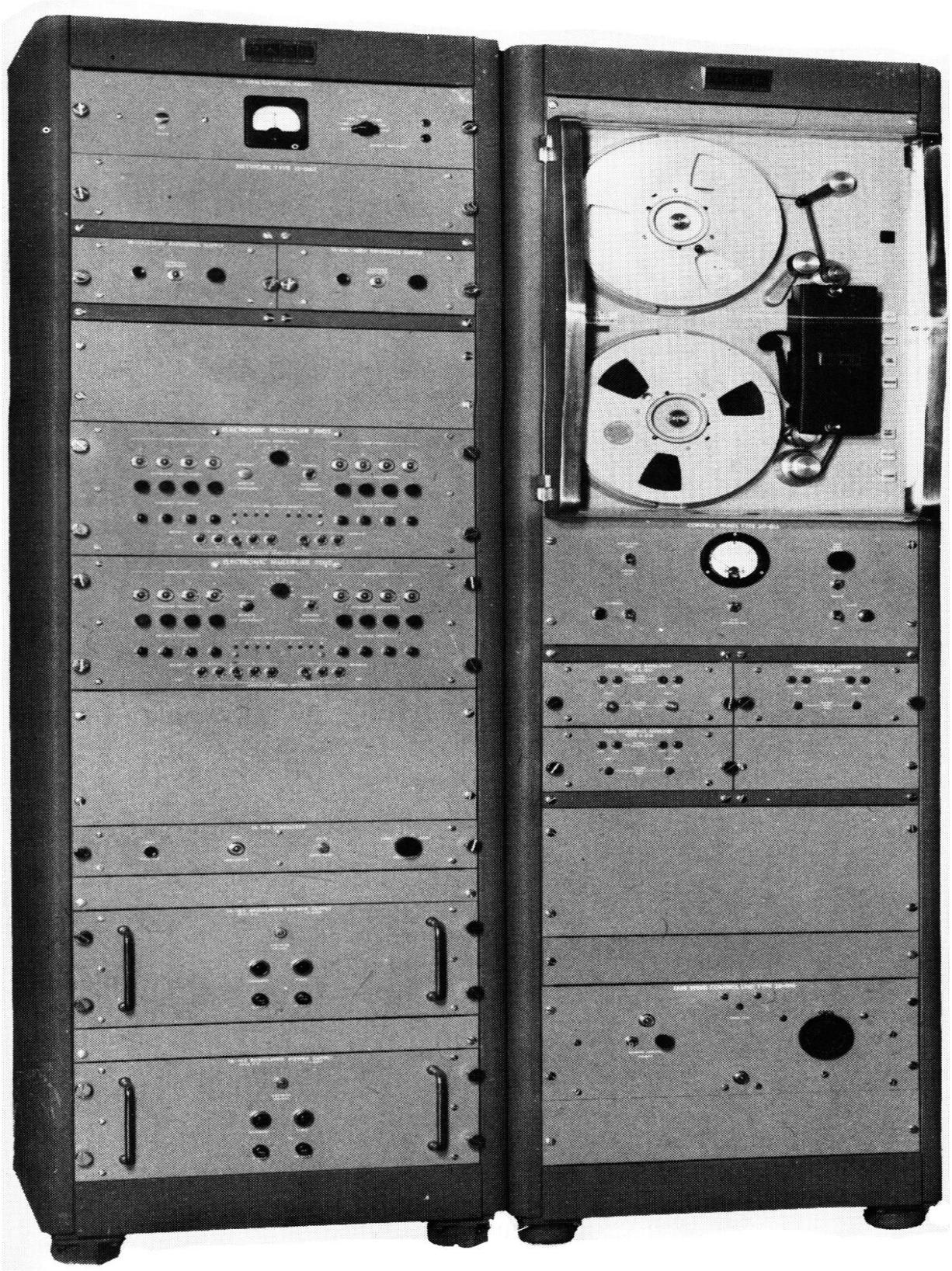


Figure 4.62. EAI time delay device (see [→Howe 1961, p. 261])

Yet another approach is based on the curve follower.[127] The function to be delayed is plotted by a strip-chart recorder. The paper containing the plot is then continuously fed into a one-dimensional curve follower, which in turn delivers the delayed function.[128]

In cases where the substantial cost of a dedicated time delay system could not be justified, delay functions can be set up using *Padé approximations* (named after Henri Padé[129]) or a *Stubbs-Single approximation*. In Laplace[130]-transform notation delays are characterized by the following general transfer function:

$$\frac{Y(s)}{X(s)} = e^{-Ts}$$

→ Figure 4.63 shows the implementation of a 4th-order Stubbs-Single approximation, which is based on the transfer function

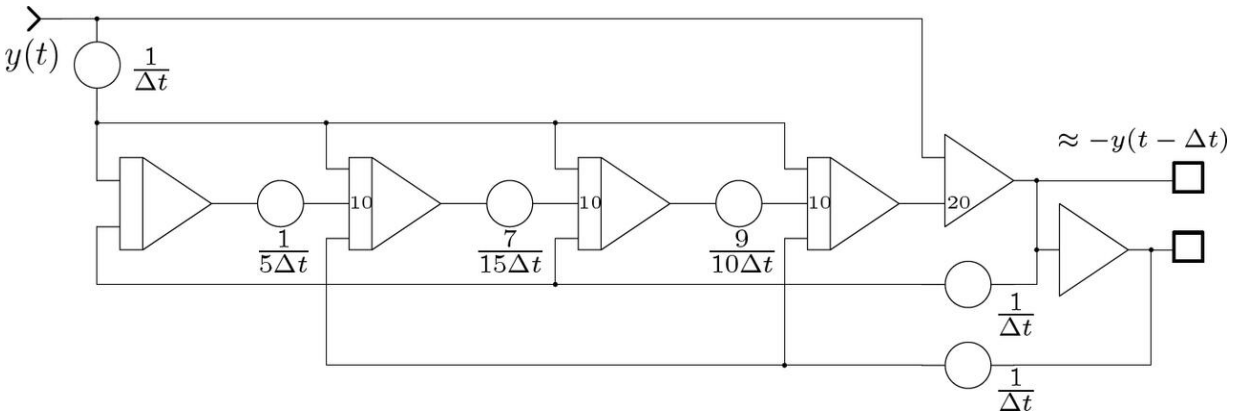


Figure 4.63. 4th-order Stubbs-Single approximation for time delay

$$S_4(s) = \frac{1-\frac{1}{2}Ts+\frac{15}{134}T^2s^2-\frac{13.55}{1072}T^3s^3+\frac{1}{1072}T^4s^4}{1+\frac{1}{2}Ts+\frac{15}{134}T^2s^2+\frac{13.55}{1072}T^3s^3+\frac{1}{1072}T^4s^4}.$$

These delay functions tend to become quite complicated with increasing expectations regarding the fidelity of the delay. Their setup is also quite time consuming due to the many coefficient potentiometers required. More details and examples regarding Padé

and Stubbs-Single approximations can be found in [→Kennedy 1962, pp. 6-5 ff.], [→Carlson et al. 1967, pp. 225 ff.], and [→Stubbs et al. 1954]. A comprehensive description of electronic time delays can be found in [→Ulmann 2023/1, pp. 91–102].

4.12 Random noise generators

Many real-world problems to be analyzed by analog computers involve random processes. Accordingly, sources of random noise are sometimes required in an analog computer.[131] Typical applications include the analysis of correlation functions and delay errors in signal networks, mechanical system analysis, the measurement of spectral densities, the study of optimization processes by sequential random perturbations, and many more.

Typical random sources employed in analog computers are based on physical systems exhibiting a random behavior such as the decay of radioactive isotopes, resistor noise, or recombination noise on PN-junctions in bipolar semiconductors.[132] Often complex demodulation schemes and special function generators are employed to yield the desired probability distribution of the random signal.[133] It is also possible and often advisable to use digital feedback shift registers and digital-analog-converters to produce random voltage signals of certain properties.

A classic low frequency random noise generator, the EAI *PACE 201A*, is shown in →figure 4.64. It uses a gas filled tube as noise source yielding a rather wide noise spectrum ranging from a few 10 Hz up to about 100 kHz. A portion of this spectrum exhibiting good flatness is extracted by a band-pass filter with a bandwidth of a few 100 Hz. The resulting signal is then down sampled by a chopper circuit running at 400 Hz. A final low-pass filter yields a noise signal with sufficiently flat spectral density between 0 Hz and 100 Hz. To ensure a stable mean amplitude, this signal is amplified by a variable gain amplifier. This, in turn, is controlled by a voltage derived from

the mean amplitude by means of a rectifier and an integrating amplifier introducing some time delay.[134]

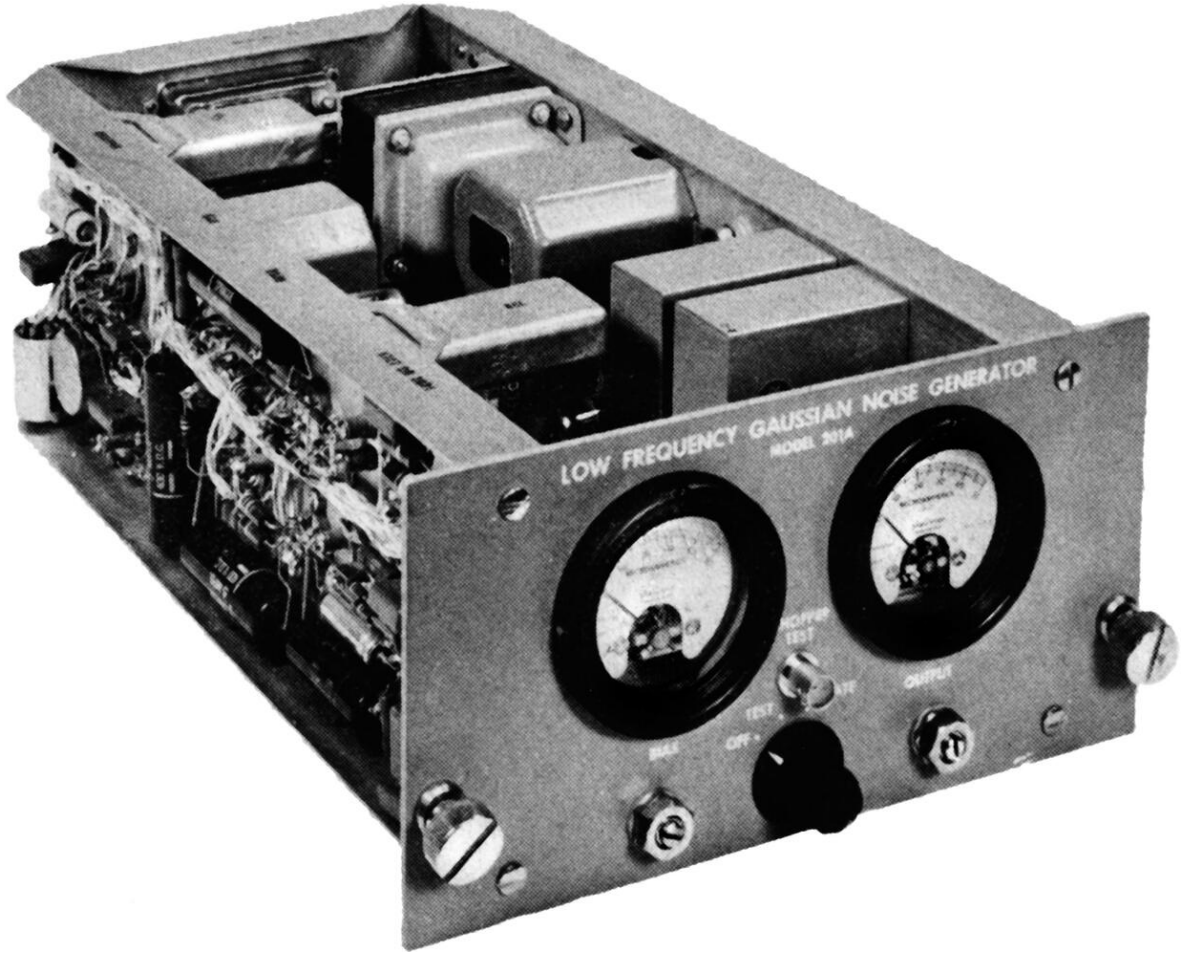


Figure 4.64. EAI PACE 201A low frequency Gaussian noise generator (see [→Howe 1961, p. 260])

An interesting approach for the generation of random signals with variable distribution of values using a white noise source as they can often be found in larger analog computers in conjunction with function generators is described in [→Solar 1977/1]. This publication gives practical examples for noise with exponential distribution, etc.

Another useful noise source are *pseudo-random sequence*s, i. e., sequences of numbers generated by some algorithm, thus

completely predictable sequences which “look like random”.[135] Ideally, such a sequence cannot be distinguished from a real random sequence, at least not easily.[136] The most simple such generators are based on recurrence formulas such as the *linear congruence generator* (*LKG*)

$$x_{i+1} = (ax_i + b) \bmod n$$

with parameters *a, b, n*, and an initial value x_0.[137] This idea of a feedback loop can also be implemented using linear shift-registers with multiple feedback paths involving *exclusive OR* (*XOR*) gates. Despite the central disadvantage of predictability of such pseudo-random generators, they offer a number of advantages, including ease of implementation, selectable sequence length, and repeatability. The resulting integer or bit stream can be converted into an analog output signal either by a DAC or a simple RC-combination with suitable time constant. A typical example of such a noise generator is the *HP 3722A*.[138] [→Wise et al. 1967] describe an interesting comparison of sinusoidal and pseudo-random noise forcing functions. The latter are generated using feedback shift-registers implemented on a digital control unit.[139] [→Kopacek et al. 1977] focus on the characteristics of a number of different pseudo-random signals and their suitability for the identification of linear systems.

4.13 Output devices

Since an analog electronic analog computer typically represents values by voltages or currents, typical classic output devices are strip-chart recorders as shown in →figure 4.65, *xy*-plotters (see →figure 4.33), and oscilloscopes.

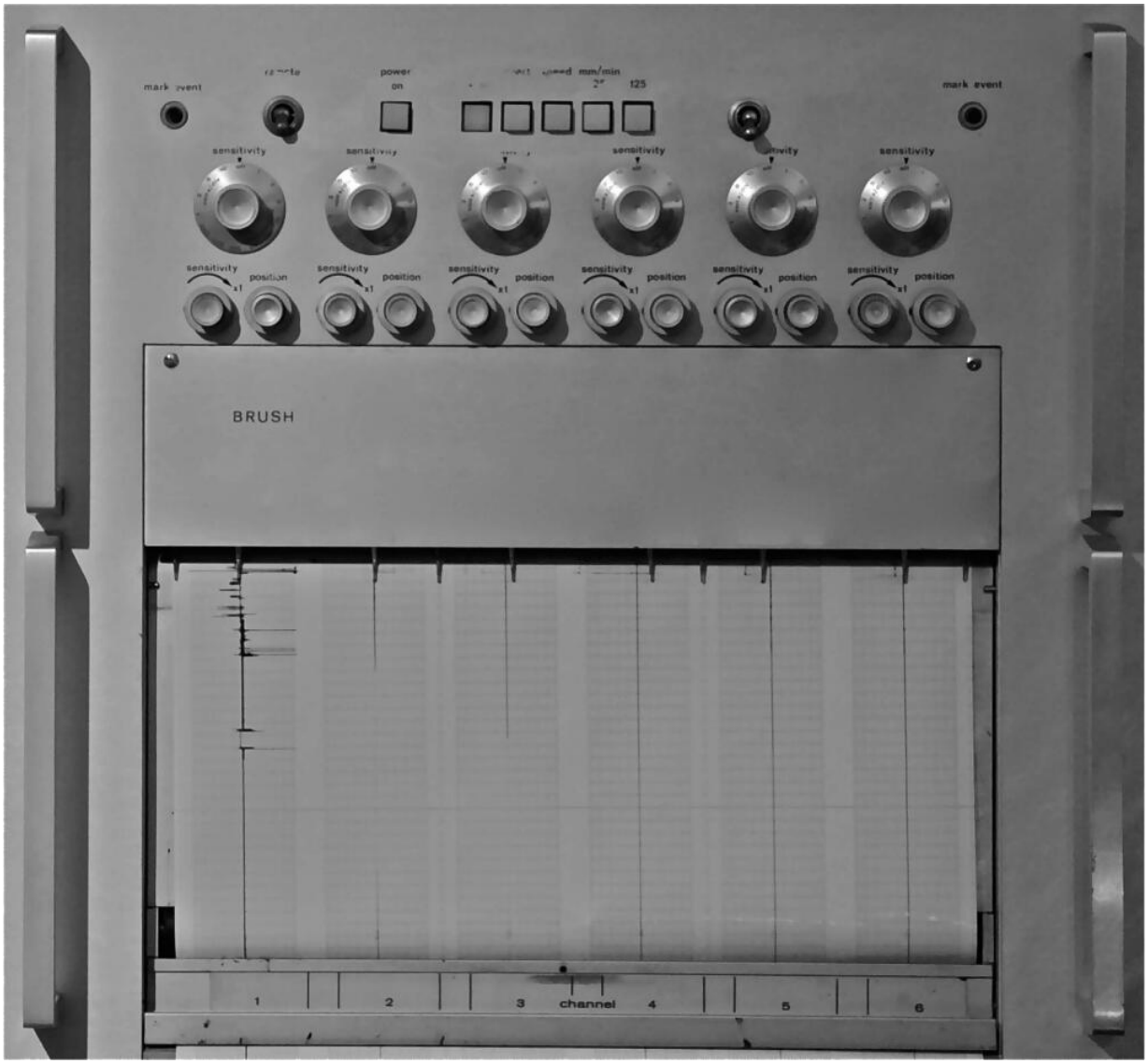


Figure 4.65. *Brush* six channel recorder

In cases where a graphical output of some solution is not required *digital voltmeters* (*DVM*s) are used to display values of variables in a simulation run with high precision. Large analog computers often feature a built-in digital voltmeter with an automatic selector.[140] To get a snapshot of the current state of a computer run, all integrators are typically placed into halt mode first. Then the output value of every computing element of interest is read out by the digital voltmeter and displayed or printed in decimal form. Additionally,

these values can be transmitted to an attached stored-program digital computer or can be stored on some digital medium.

→Figure 4.66 shows the display of a typical precision digital voltmeter – in this case the precision DVM of a Telefunken RA 770 analog computer. The left half of the display[141] shows the address of the computing element selected for readout. In this case it is `I000` denoting integrator number 000. The output value of this element is currently +0.9998. It should be noted that most of the digital voltmeters used in analog computers display their values scaled with respect to the machine unit of the system, so the above value corresponds to +9.998 V since the RA 770 is a ±10 V system.[142]

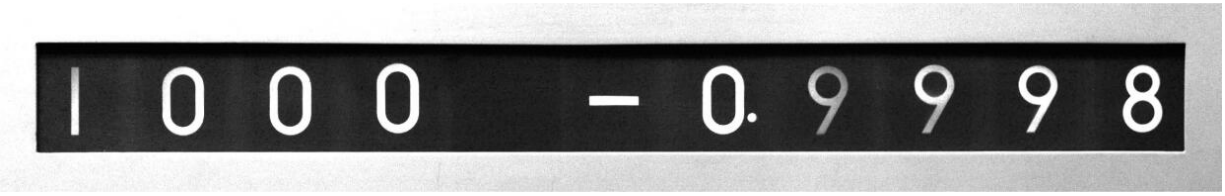


Figure 4.66. Readout of integrator `I000` on a Telefunken RA 770 analog computer

A typical stand-alone digital voltmeter is shown in →figure 4.67. This 3.5 digit instrument was built by Electronic Associates Inc. and was typically used in conjunction with small tabletop analog computers such as the TR-10 or TR-20. Its construction is quite remarkable as it consists of a frequency counter module and a DVM module. The instrument can be used as a frequency counter or as a digital voltmeter. In the latter mode the frequency counter becomes part of the DVM circuitry.



Figure 4.67. EAI 6200 digital voltmeter

The following chapter will now focus on the anatomy of a typical classic medium sized analog electronic analog computer, giving an impression of the arrangement of computing elements described in the preceding sections.

5 The anatomy of a classic analog computer

The system chosen for this section on the anatomy of classic analog computers is the medium sized *EAI-580*, built by Electronic Associates Inc. in the late 1960s and shown in →figure 5.1. Although labeled as an “ANALOG/HYBRID COMPUTING SYSTEM” by EAI’s marketing department, it is just a basic analog computer with provision to interface it to a digital computer, if required, in which case suitable external analog to digital and digital to analog converters (ADCs and DACs) must be added.[1]

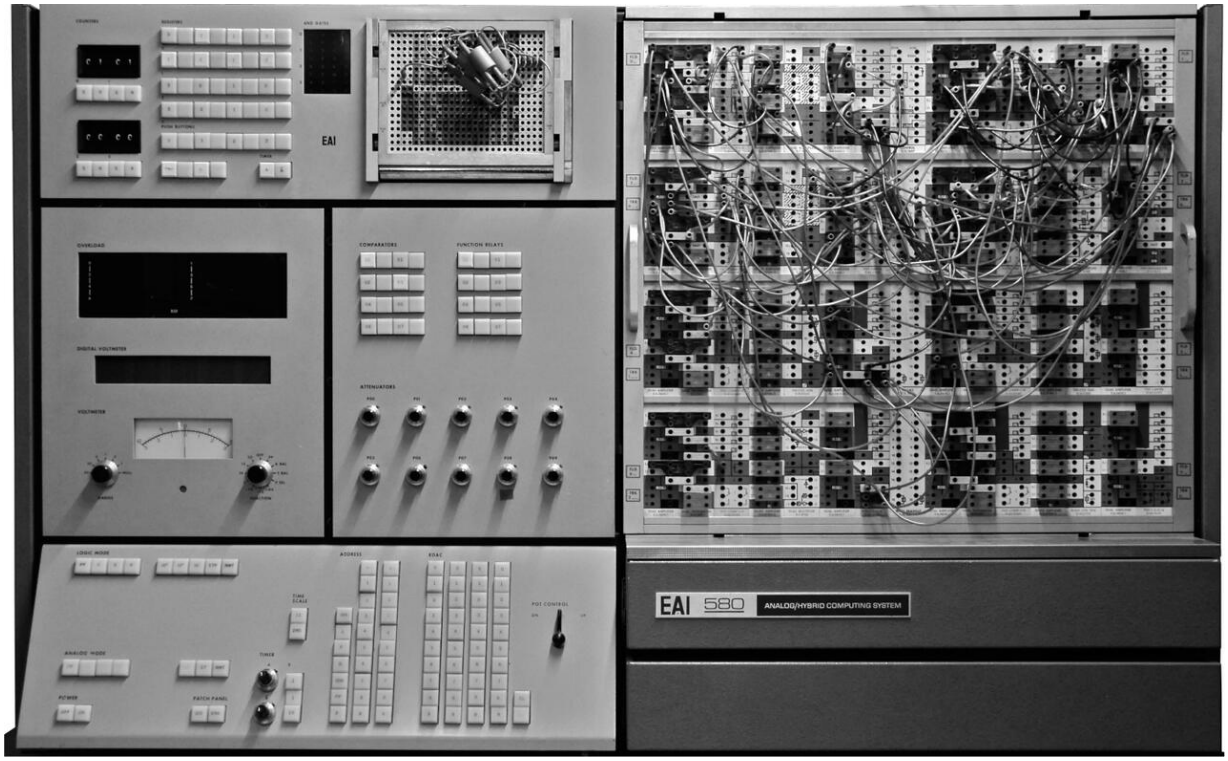


Figure 5.1. EAI 580 analog computer

5.1 Analog patch panel

The most prominent feature of any medium or large classic analog computer is its central *(analog) patch panel*. It contains hundreds, or even thousands, of jacks which connect to all inputs and outputs of the various computing elements of the system. A program is set up by means of *patch cables*, which are plugged into these jacks, to configure the computer to solve the problem. Patch cables for medium precision computers are typically simple unshielded cables with banana plugs on their ends. *Precision analog computers* often use shielded patch cables with special plugs and intricate shielded patch panels to achieve up to four decimal places of precision.

Since the process of patching an analog computer is quite time consuming, the patch panels of all but the smallest analog computers are exchangeable. Otherwise the times to switch from one program to another would be unreasonably high.

→ Figure 5.2 shows the analog patch panel of an EAI 580 analog computer. This panel is of the modular type, using insets, which represent the different computing elements. It contains four rows of computing elements each consisting of 15 inserts. Each of these has 2 columns and 12 rows of connectors. Such a computer typically contains the following types of computing elements:

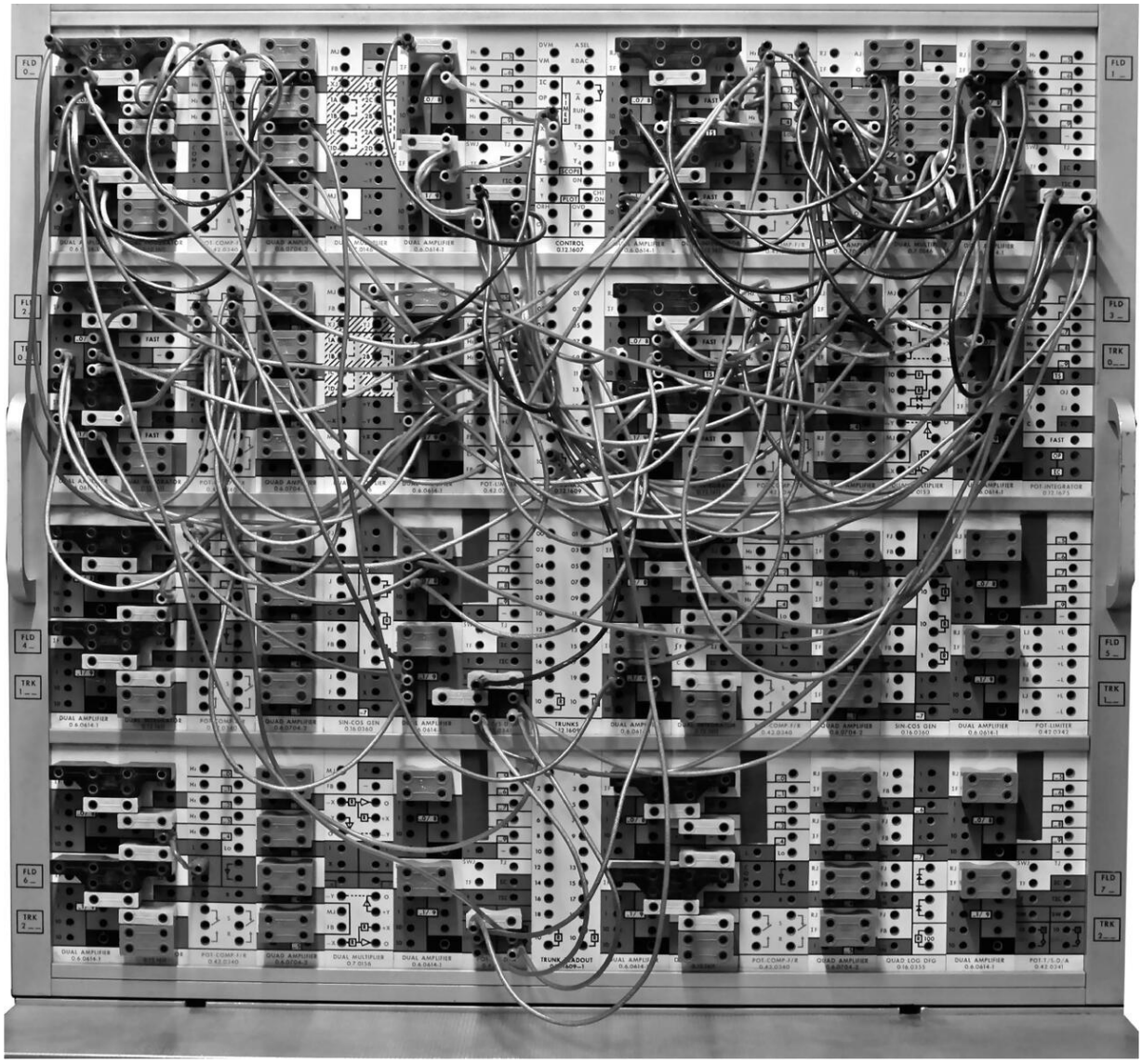


Figure 5.2. EAI 580 analog patch panel

- Dual operational amplifiers, usually paired with dual integrator circuits,[2]
- quadruple amplifiers,
- x^2-function generators, which are used to setup quarter square multipliers,
- coefficient potentiometers,
- comparators with electronic analog switches,

- function generators, and
- *trunk lines*,[3] etc.

5.2 Function generators

Located below the analog patch panel are two drawers, which each contain four diode function generators. Each of these function generators has ten diode segments with adjustable breakpoint and slope.[4] →Figure 5.3 shows the extended lower drawer.



Figure 5.3. EAI 580 function generators

As noted before, there are two basic variants of diode function generators: With and without adjustable breakpoints. The former implementation has the advantage that the breakpoints for a function approximation can be clustered in regions where the function to be implemented is “interesting” (i. e., changes rapidly in a certain region). This approach has the disadvantage of a tortuous setup procedure, as the breakpoints and slopes must be calculated and set manually. The second implementation variant, which is typically used in Telefunken analog computers, speeds up the setup considerably by omitting the variable breakpoints, but this can sacrifice precision of the function approximation. It is mainly a matter of personal preference which implementation is preferred.

5.3 Digital patch panel and controls

Located in the top half on the left of the EAI 580 is a small digital subsystem[5] consisting of a tiny patch panel, two thumb wheel switches to preset digital counters/dividers, and a number of switches which can be used to set or reset individual bits of these counters. These switches also display the current state of each associated bit. The remaining switches control individual *flip-flops*[6] and a timer. → Figure 5.4 shows this subsystem.

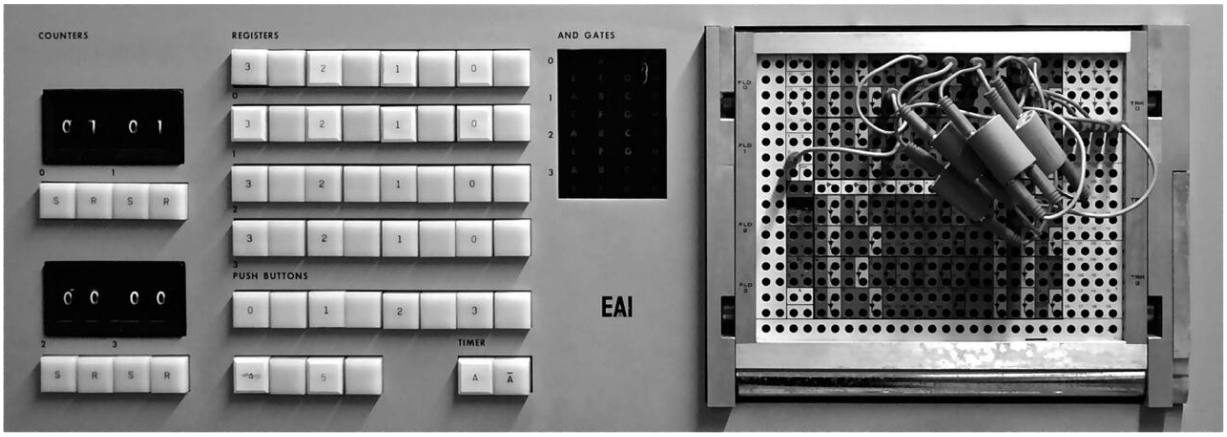


Figure 5.4. EAI 580 digital controls and patch panel

The various digital elements (simple logic gates, inverters, counters, and dividers) can be interconnected by means of patch cables plugged into the small removable digital patch panel. Although many applications of an analog computer do not require any digital elements at all, there are more complex tasks such as parameter optimization, etc., which benefit greatly from the flexibility offered by a digital control system. A typical example is the simulation of an automatic transmission for a car: The behavior of the gears, the torque converter, etc., can be simulated readily by the analog part of the computer. The control of these transmission parts, the gear changing decisions under varying loads, etc., is then implemented using the digital part of the computer.

The inputs to these digital elements are typically clock signals generated by a central timing unit, outputs of comparators used in the analog setup, manual switches, etc. Normally, the digital outputs are used to control electronic switches on the analog computer[7] or to control the operation of individual integrators or groups of integrators. Some tasks such as parameter optimization often require at least two groups of integrators, which operate in an alternating fashion: One

integrator group is halted and delivers initial conditions for a second group, which will perform some computation based on these values. This second group is then halted and can be used to update the first integrator group, etc. Alternatively, the first integrator group can compute a new set of initial conditions while the second group is still running.[8]

5.4 Readout

Located below the digital control unit is the overload display and readout system on the left and manual coefficient potentiometers and switches associated with comparators and function relays on the right. →Figure 5.5 shows these subsystems of the EAI 580 analog computer in detail.

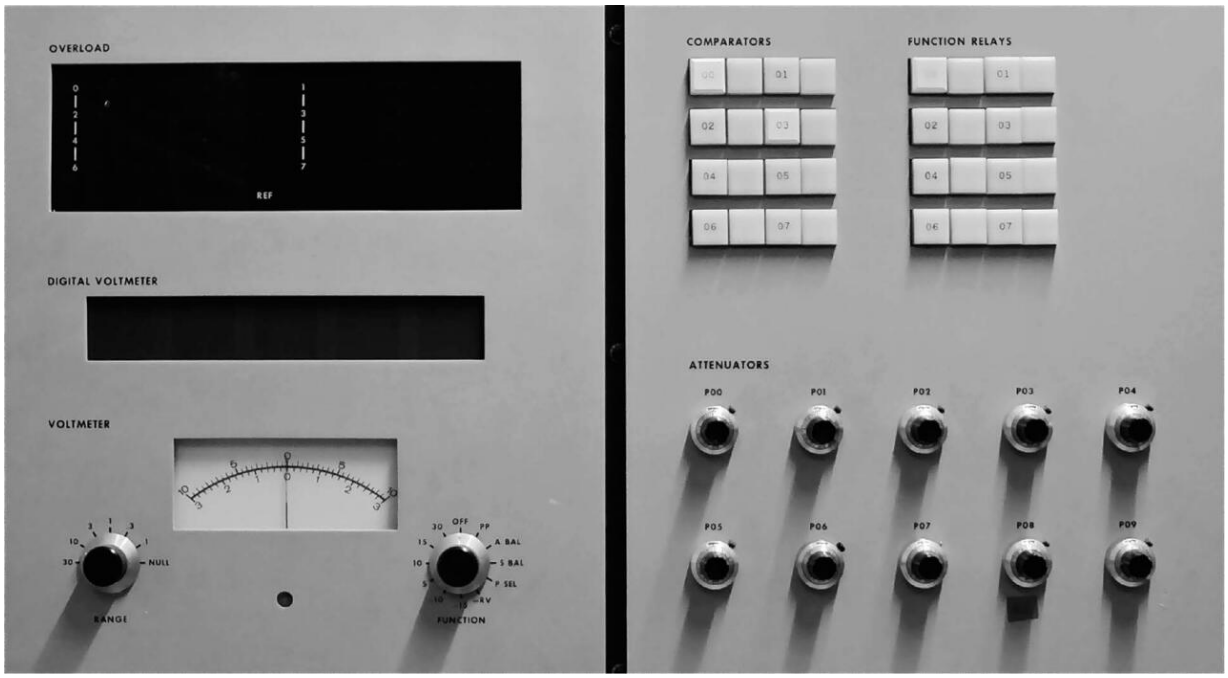


Figure 5.5. EAI 580 readout, overload display, manual potentiometers, and manual switches

The overload display on the upper left has one incandescent light bulb per amplifier in the system to display overload conditions. There are two basic types of overload that may occur in an operational amplifier, in both of which it will be unable to keep its output at the level required to satisfy the condition that the inverting summing junction of the amplifier must be maintained at 0 V. Taking a summer with two inputs e_1, e_2 and associated weights $a_1 = a_2 = 1$ as an example this error condition could result from the attempt to add two input voltages of +1 each. The output voltage of an amplifier cannot rise above or go below its positive and negative supply voltages. In this example the output, which should be −2, would be limited by the amplifier's output range to a lower value such as −1.4. Consequently, the current flowing through the feedback resistor can no longer cancel the input currents at the summing junction.

A similar problem would result from too high a load current at the output of an amplifier, in which case it could not maintain the proper output voltage. Connecting the output of an amplifier to ground or to one of the reference voltages would result in such an overload condition, as would be feeding too many computing elements from a single operational amplifier's output.

When chopper-stabilized amplifiers are used, the detection of such overload conditions is easy: In each case the inverting summing junction can no longer be maintained at 0 V. This will in turn yield an excessive error correction signal at the output of the stabilizing amplifier, which can then be used to light the corresponding overload indicator for this amplifier.

Since an overload condition normally renders a computation useless, it will in most cases also place the analog computer into halt mode. The programmer can then check the program to determine and remedy the reason for the overload.

Below the overload indicator panel is the display of a precision digital voltmeter with four digits of resolution. This is used to read out program variables.[9] Digital voltmeters like this can often be connected to a printer and a selector unit, which allows automatic readout of a number of computing elements.

On the bottom left is an analog voltmeter, which can also be used to determine the value of program variables as well as for a quick checkout of the computer. By means of the rotary switch on its right the various power supply voltages can be checked easily. In addition to this, the voltmeter is used for manual balancing of the operational amplifiers.[10]

The right half of this panel contains a couple of switches connected to the comparators and function relays of the analog computer. Below these are ten manual ten-turn coefficient potentiometers, called *attenuators* on this machine. The machine has many more coefficient potentiometers, which are set by servo-controlled motors, as it proved impractical to make users manually set up to several hundred coefficient potentiometers.

5.5 Control

The unit on the lower left of the computer is the central control panel, which is used to control the overall operation of the computer and for setting the servo coefficient potentiometers. →Figure 5.6 shows this control panel in detail.

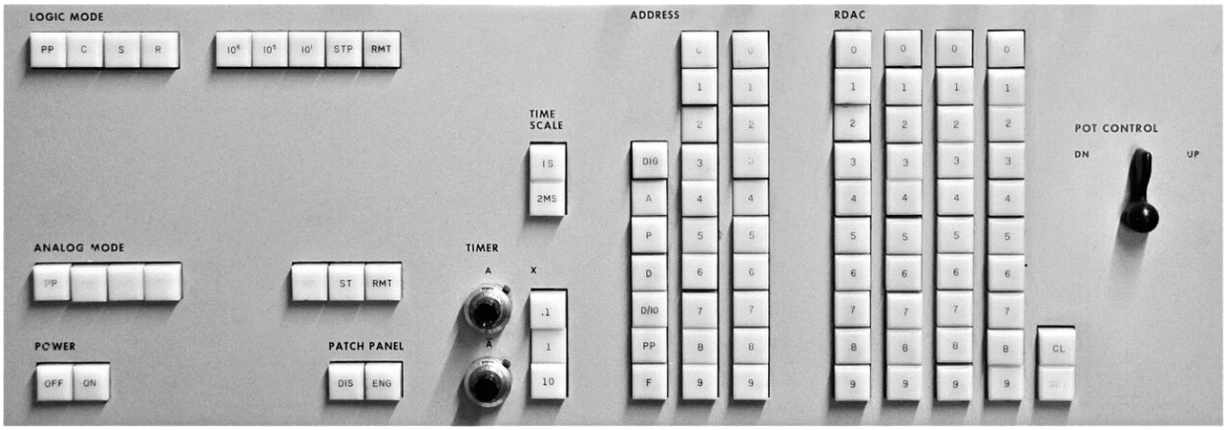


Figure 5.6. EAI 580 control panel

Located on the far left are, from bottom to top, the power switches, four control switches to select the mode of operation of the analog computing elements and similar switches for the digital part. The available analog operating modes are (from left to right[11]):

> **PP:** Short for *Program Panel*, this mode transfers the overall control of all integrators to the digital unit on the top left.
> **IC:** Place all integrators into initial condition mode.
> **HD:** All integrators are placed into halt mode.
> **OP:** Switch all integrators into operate mode.

The logic mode switches are used to

- transfer the control of the digital elements to the digital patch panel (PP),
- to clear (C),
- stop (S),
- and run (R) the digital circuit setup on the digital patch panel.

The switches labeled 10^6, 10^5, 10^1, and STP select the basic digital clock rate, one pulse per microsecond being the default cycle time, or place the digital subsystem into single step mode. The RMT switch places the digital unit into remote mode, which allows several EAI 580 systems to be coupled together or to couple an EAI 580 analog computer with a stored-program computer, basically forming a hybrid computer.[12]

The two switches labeled DIS and ENG are used to disengage and engage the large removable analog patch panel. Due to its many connections, it is locked into place by a motor driven mechanism, which is controlled by these two switches.

Above these two switches is a row of three buttons labeled SP, ST, and RMT. These place the machine into the potentiometer-set mode (SP), a check mode called *static test*[13] or into remote mode, which transfers the control of the analog portion to an external system such as a memory programmed computer.

Next to these are two switches, which are used to select the time base of the system. By default all integrators are set to a time-scale of 1 second, so that an input voltage corresponding to one machine unit will yield −1 at the output of the integrator after one second. Pressing the button labeled 2MS switches all integrators to a second set of integration capacitors with 1/500 of the capacity of the normal feedback capacitors, thus speeding up the computation by a factor of 500.

The two ten-turn potentiometers control timers for *repetitive operation*. In this mode a computation will be performed over and over again, cycling the integrators through their initial condition and operate modes automatically. The keys labeled .1, 1, and 10 set a global multiplicative factor for the timers controlling the repetitive operation of the system.

The next three columns of switches are labeled ADDRESS and allow the selection of any computing element of the analog

section of the EAI 580 for readout or setup in the case of servo potentiometers. Each element has an element type and a two digit decimal address. The output value of an addressed element is displayed on the digital voltmeter, together with a one letter function code denoting the type of the element.[14]

The next block of switches is labeled RDAC and forms a simple, yet precise resistor-based digital-analog converter. Its main purpose is to set servo controlled coefficient potentiometers: First, using the ADDRESS switches, a potentiometer to be set is selected. In the next step the desired value is keyed into the RDAC keys and the SET key is pressed. This causes the servo motor of the selected potentiometer to position its wiper to the desired value. In some cases it is desirable to control a servo potentiometer manually, which can be done with the big lever on the far right. Turning it to the left will cause the servo motor of the selected potentiometer to decrease the coefficient's value; turning it to the right increases its value.

5.6 Performing a computation

A computation on the EAI 580 analog computer is performed as follows:

1. The computer is switched on by pressing the ON button.
2. The programs prepared on the analog and digital panels have to be inserted into the machine. While the digital patch panel is engaged and disengaged manually by means of a lever, the analog patch panel is unlocked or locked by pressing the DIS or ENG switch on the control panel.
3. The various servo driven coefficient potentiometers and the manually controlled attenuators must be set.

4. If required for this computation, the diode function generators must be set up manually.
5. If required, initial conditions for the digital computing elements have to be set by pushing the switches associated with the counters, flip-flops, etc.
6. The time-scale is set if repetitive operation is desired.
7. The analog and digital subsystems of the analog computer can then be put into the required mode of operation. In the simplest case this will just involve setting the analog part of the machine into IC mode, followed by OP.

6 Some typical analog computers

So many companies made so many different models of analog computers that it is impossible to describe them in detail here.[1] The following sections describe some typical analog computers.

6.1 Telefunken RA 1

In the early 1950s Dr. Ernst Kettel, an engineer working for Telefunken, a world-renowned manufacturer of radio transmitters, receivers, RADAR sets, etc., started thinking about building an analog computer to do research in his main field of interest – communications equipment. Eventually, this project transformed Telefunken into a leading analog computer manufacturer in Germany and Europe. It seems reasonable to assume that Dr. Kettel met Helmut Hoelzer during his work in Peenemünde, where he developed radio control and telemetry systems for the rocket development projects. He had probably seen Hoelzer's general purpose analog computer in operation there, which would explain his general interest in this topic.

In 1953 actual work on this machine started with the help of Dr. Adolf Kley. The machine became known as the *RA 1*[2] and is shown in →figure 6.1.

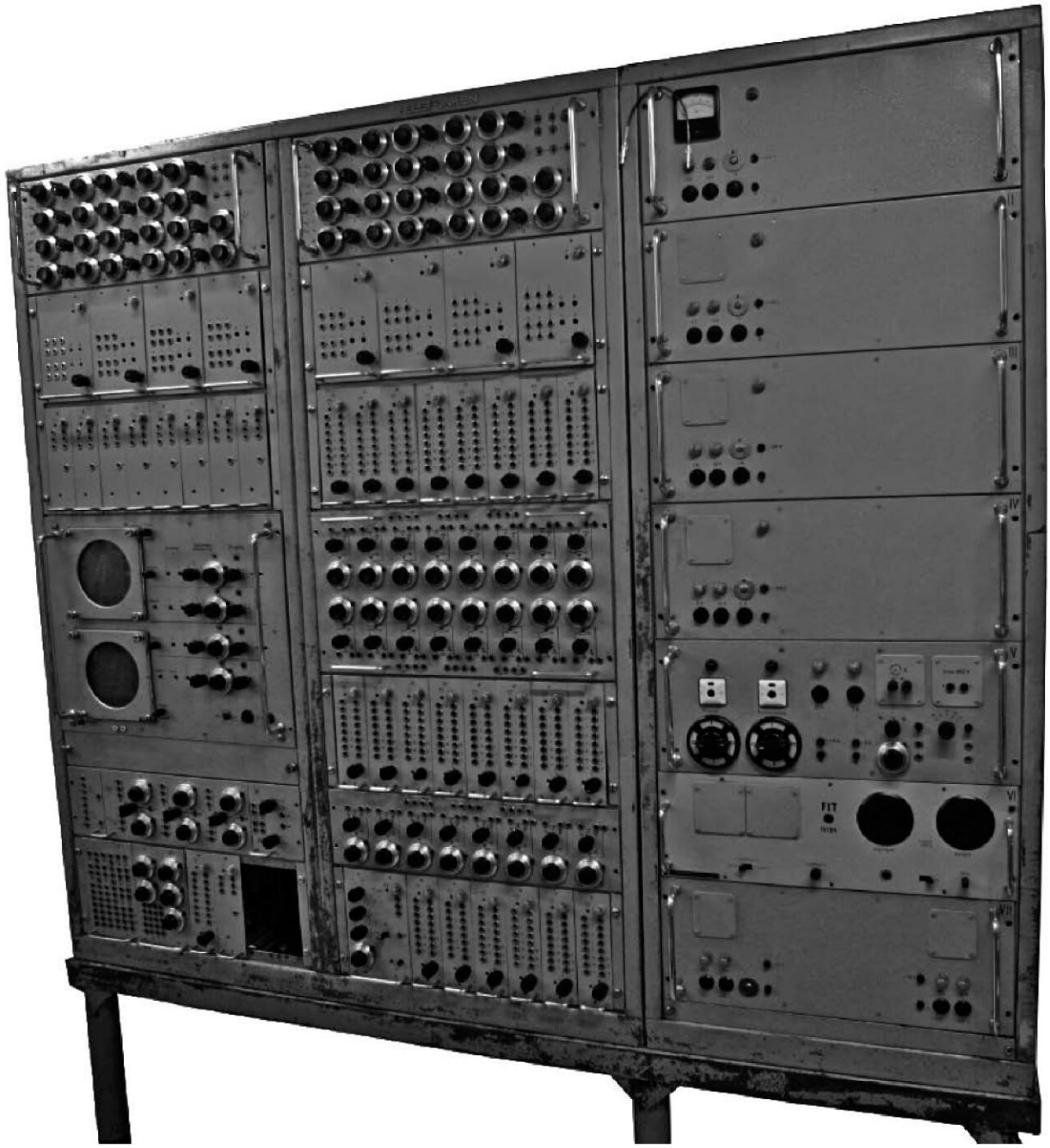


Figure 6.1. The Telefunken RA 1 as it is currently preserved in a private collection

[→ Feilmeier 1974, p. 18] describes the RA 1 as a typical system of the *heroic age* of analog computing, which spanned from about 1945 until 1955. The machine is of a completely modular design.

Three racks hold a variety of computing and support elements. The right-hand rack contains the power supplies delivering the highly precise machine voltages of ±100 V, the main supply voltages of ±200 V, a bias voltage of −450 V, etc.

The rack on the left holds the following subsystems from top to bottom:

- A vacuum diode function generator with 21 fixed breakpoints and 21 precision potentiometers and switches to set the slope of the polygon,
- four quarter square multipliers[3],
- eight summers,
- an oscilloscope of rather interesting design with two display tubes,
- a collection of limiters and the like,
- some *free diodes*,[4]
- relays to set up comparator circuits,
- five coefficient potentiometers,
- a selectable integrator/summer/amplifier,
- and another summer.

The rack in the middle contains (top to bottom)

- a second diode function generator,
- four quarter square multipliers,
- eight switchable integrators/summers/amplifiers,
- 16 coefficient potentiometers,
- another eight switchable integrators/summers/amplifiers,
- eight coefficient potentiometers and
- a drawer containing the central timing and control module as well as six switchable integrators/summers/amplifiers.

It is quite obvious that the RA 1 was built as a piece of laboratory equipment and not as the prototype of a product to be sold to

customers. All computing elements are interconnected with shielded patch cables, but there is no central patch panel, there is no means to simplify setting up coefficient potentiometers, there is no built-in voltmeter, etc.

Telefunken publicly displayed this machine at the Hannover fair 1957 where the promotion picture 6.2 was taken, showing the ease with which this machine could be operated.

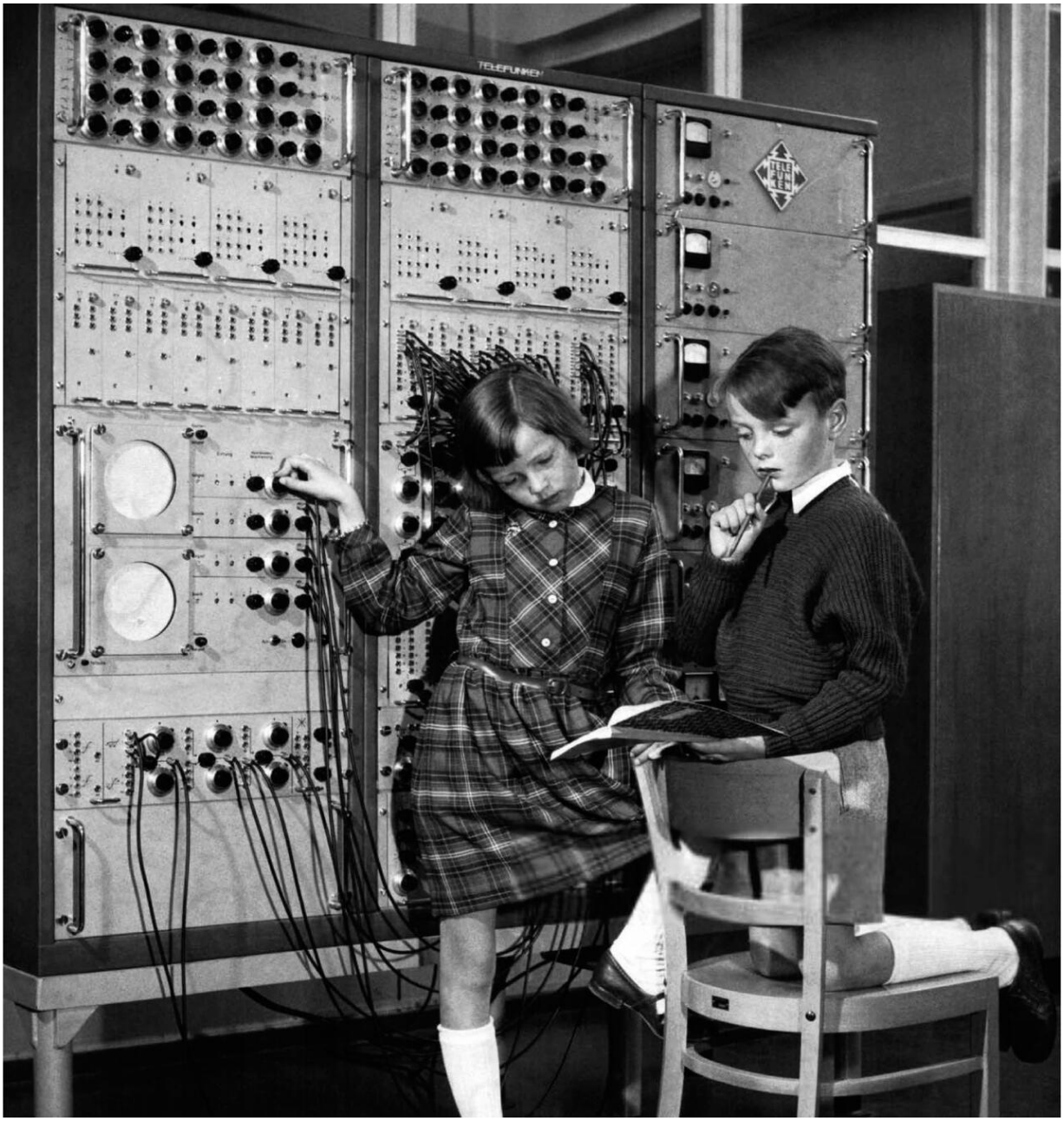


Figure 6.2. Promotional picture of the Telefunken RA 1 taken at the Hannover fair 1957

Following this display as well as laboratory visits, customers from industry expressed their interest in buying such a machine. According to various personal recollections, this was quite unexpected as the system was not intended to become a

product and should only demonstrate the capabilities of Telefunken's research departments. However, Telefunken agreed to build a batch of ten systems, which were slightly improved over this prototype machine.[5] A very large configuration of this production model, named *RA 463/2*[6] is shown in →figure 6.3. →Figure 6.4 shows a typical medium-scale installation.

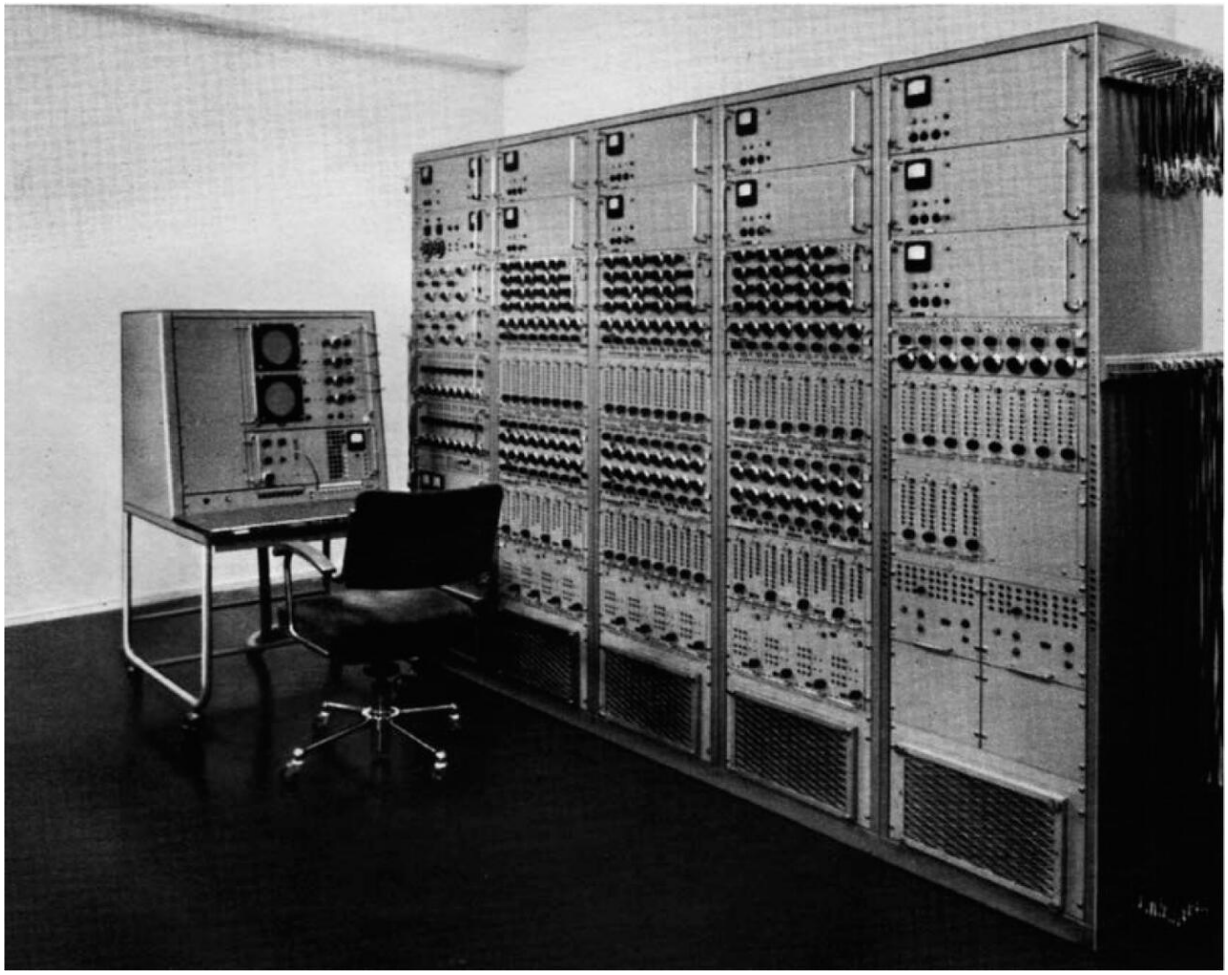


Figure 6.3. A large Telefunken RA 463/2 analog computer installation (see [→Ammon[a], p. 1])

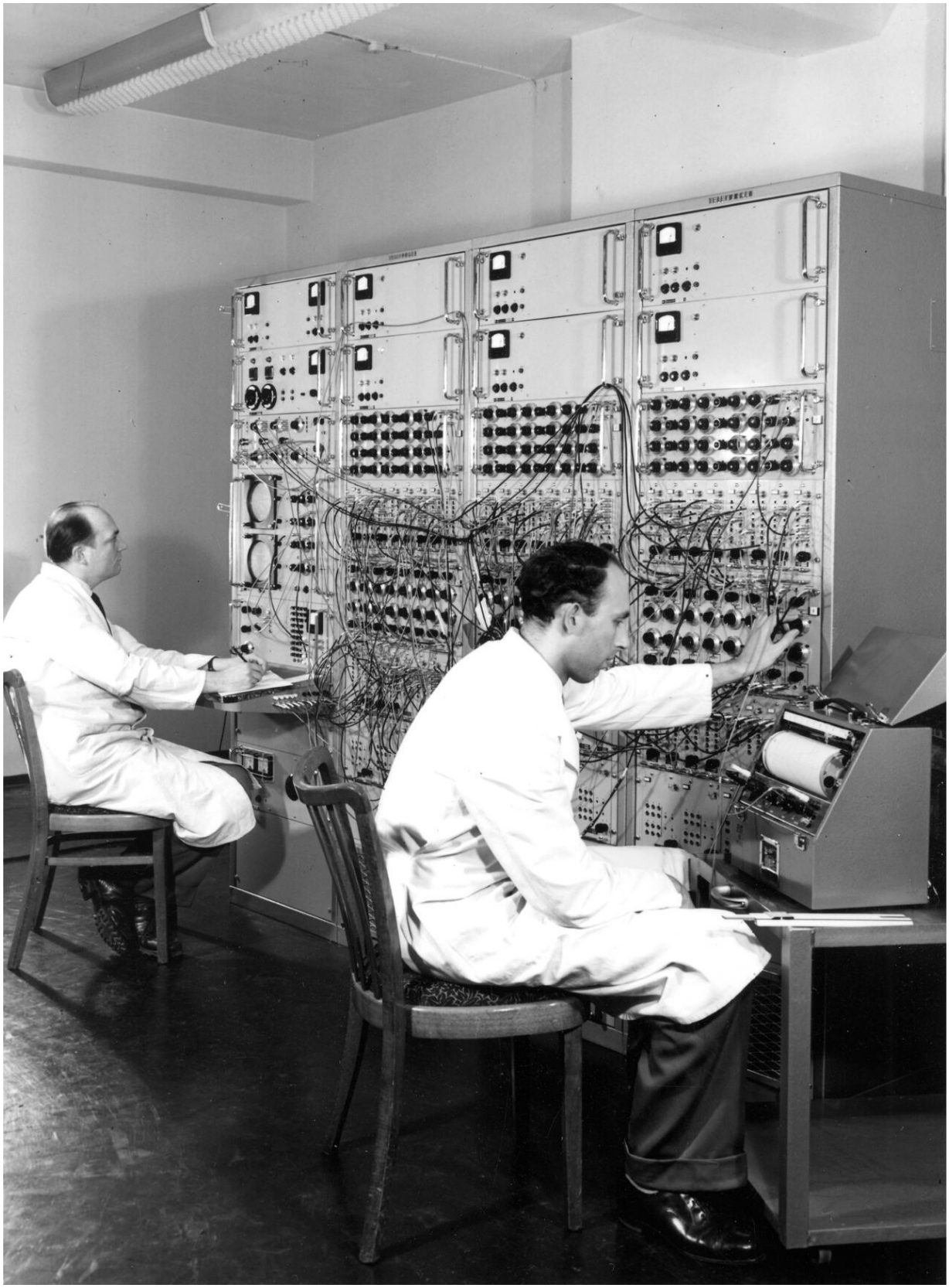


Figure 6.4. Another Telefunken RA 463/2 analog computer (sitting on the left is Prof. Dr. Otto Föllinger with Dr. Werner Ammon on the right)

A typical RA 463/2 system, consisting of only three racks with no operator console table, was priced at 158,000 DM in 1958 (see [→Telefunken 1958]) corresponding to about 410,000 US$ today. Since the first batch of these machines was sold before the computers were even built, additional systems were manufactured – reluctantly at first – and thus Telefunken entered the analog computer business.

6.2 GAP/R analog computers

The analog computer pioneer George A. Philbrick and founder of *GAP/R* (*George A. Philbrick Researches*) not only brought operational amplifiers into widespread commercial use in 1952 but also built and sold highly modular electronic analog computers.[7]

These computers are built from “black boxes”, which implemented various computing functions such as the K3-A adding component shown in →figure 4.21 in →section 4.2. These modules typically employed a number of operational amplifiers such as the K2-W[8] as well as the necessary passive elements to implement a certain function. These components were grouped into linear devices such as the K3-A adding component, the K3-C coefficient component, integrators, differentiators, etc. Nonlinear components included limiters, backlash, multipliers, square functions, square root, etc.

These systems were primarily intended for high-speed, repetitive operation.[9] A remarkable feature was the *Electronic Graph Paper*, which basically displayed a grid on an oscilloscope,

thus giving "*instantaneous and automatic calibration of both voltage and time on the oscilloscope screen itself*".[10]

→ Figure 6.5 shows a typical GAP/R analog computer installation from 1957. The large oscilloscope display in the middle is part of the electronic graph paper capability. The overall construction is highly modular and flexible.

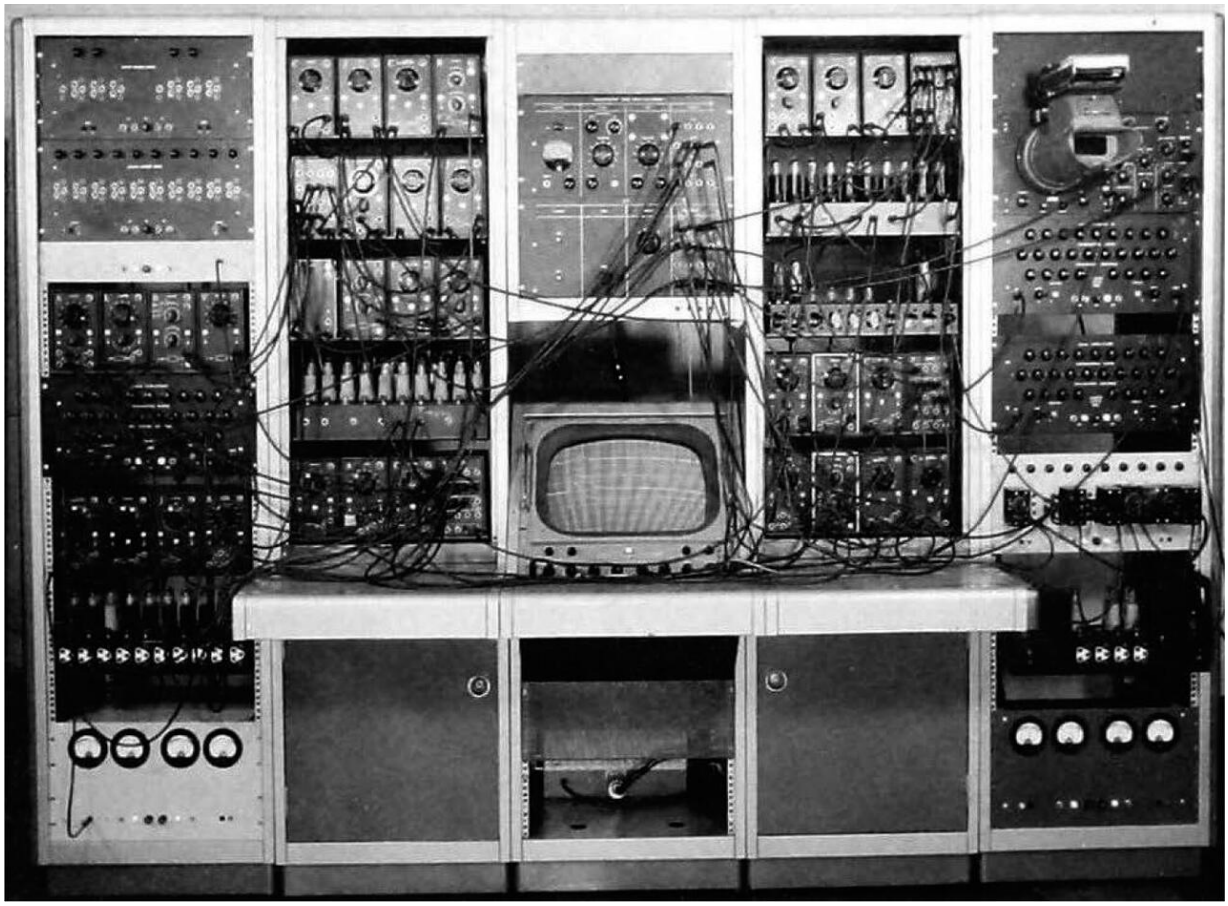


Figure 6.5. Typical GAP/R analog computer setup from 1957

6.3 EAI 231R

The Electronic Associates *EAI 231R* analog computer is one of the machines that became archetypal for analog computers during

the 1960s. It was introduced in 1959 and already had a long ancestry, which made this system far superior to the Telefunken RA 1 and its production model RA 463/2 with respect to its overall layout, the available computing elements, and their precision. The 231R and its predecessors were part of the *PACE* (short for *Precision Analog Computing Equipment*) series including the earlier 16-24A, 16-24D, 16-31R, and 16-131R models.

A typical EAI 231R analog computer is shown in →figure 6.6. This system became one of the most widely-used analog computers and systems of this type were still in use in the late 1970s. The system is dominated by the large patch panel on the right side containing 3450 plug sockets. These sockets as well as the patch cables are completely shielded, to minimize cross talk and noise injection. Below the patch panel is the operator control panel containing 20 manually operated coefficient potentiometers on the right and the timing control, analog voltmeter, and computing element selection circuitry on the left. The two large bays on the top of the system can hold either additional 80 manual coefficient potentiometers or servo potentiometers.[11] Located above the operator panel are a random noise generator (recognizable by its two circular analog instruments), a digital voltmeter, two servo multipliers, and a printer connected to the DVM.[12]

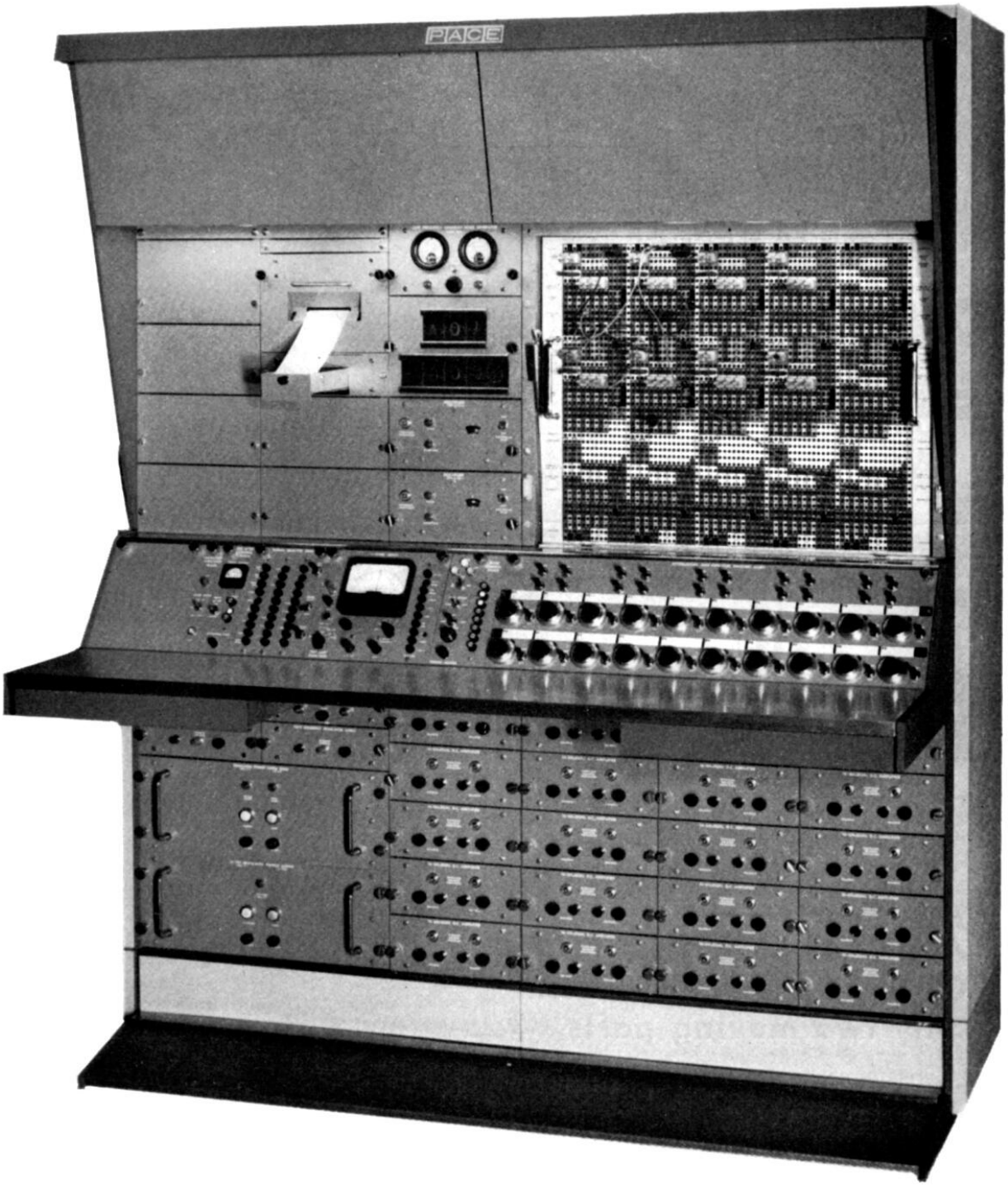


Figure 6.6. EAI 231R analog computer

The bays below the operator console contain 24 modules, each with four operational amplifiers and two power supplies. All

input and feedback resistors and capacitors are contained in a temperature controlled oven to minimize drift effects due to temperature variations. →Figure 6.7 shows the removable patch field. The system depicted has manual coefficient potentiometers installed in the top enclosures and an extension rack mounted to its right.

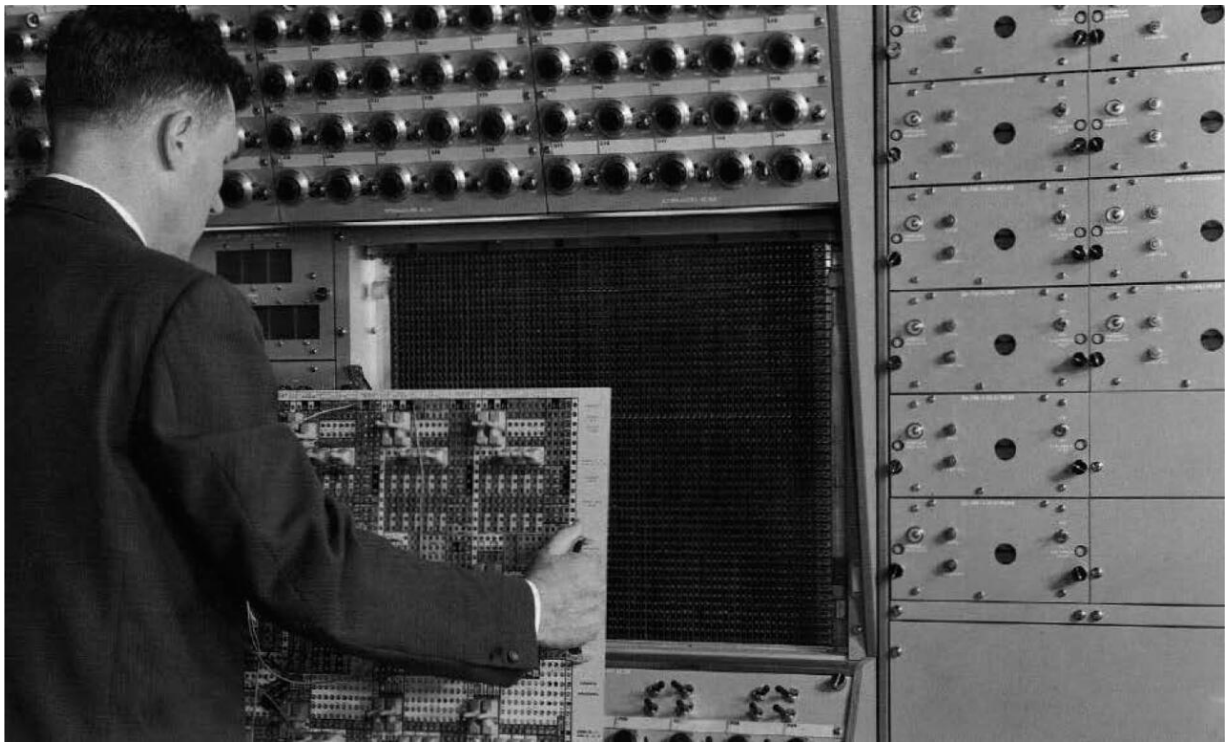


Figure 6.7. Removing the patch field from an EAI 231R computer (see [→EAI PACE 231R, p. 6])

A large EAI 231R installation is shown in →figure 6.8. It consists of two fully equipped EAI 231R analog computers, each with one extension rack to its left and one to its right. A large flatbed plotter can be seen in the left foreground. Visible in the middle of the picture is a smaller EAI *Variplotter* and on the right are two multi-channel recorders and an *ADIOS* (short for *Automatic Digital Input Output System*)[13] console, which can be used to

automatically set up servo potentiometers under paper tape control.

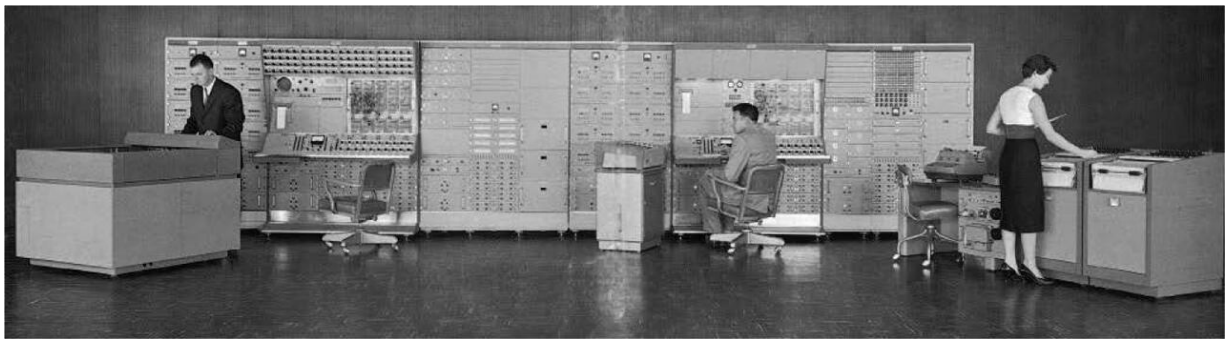


Figure 6.8. Installation of two EAI 231R with various plotters and an ADIOS console

The 231R was such a successful system that EAI developed an enhanced version, the *EAI 231RV*, that was put on the market in 1964. In contrast to the 231R, it contains a sophisticated digital control system that allows control of individual integrators and easy implementation of complex control sequences. In addition, the integrators featured additional time-scale factors. These systems found widespread use in nearly all branches of science and engineering but most prominently in the field of aerospace engineering. →Figure 6.9 shows a large EAI 231RV system with the main console in the middle, surrounded by additional computing elements on the racks to its left and right. Both expansion racks hold a number of additional quad operational amplifiers as well as a large number of diode function generators. The right rack also contains some servo multipliers and a vertically mounted *x/y* plotter.

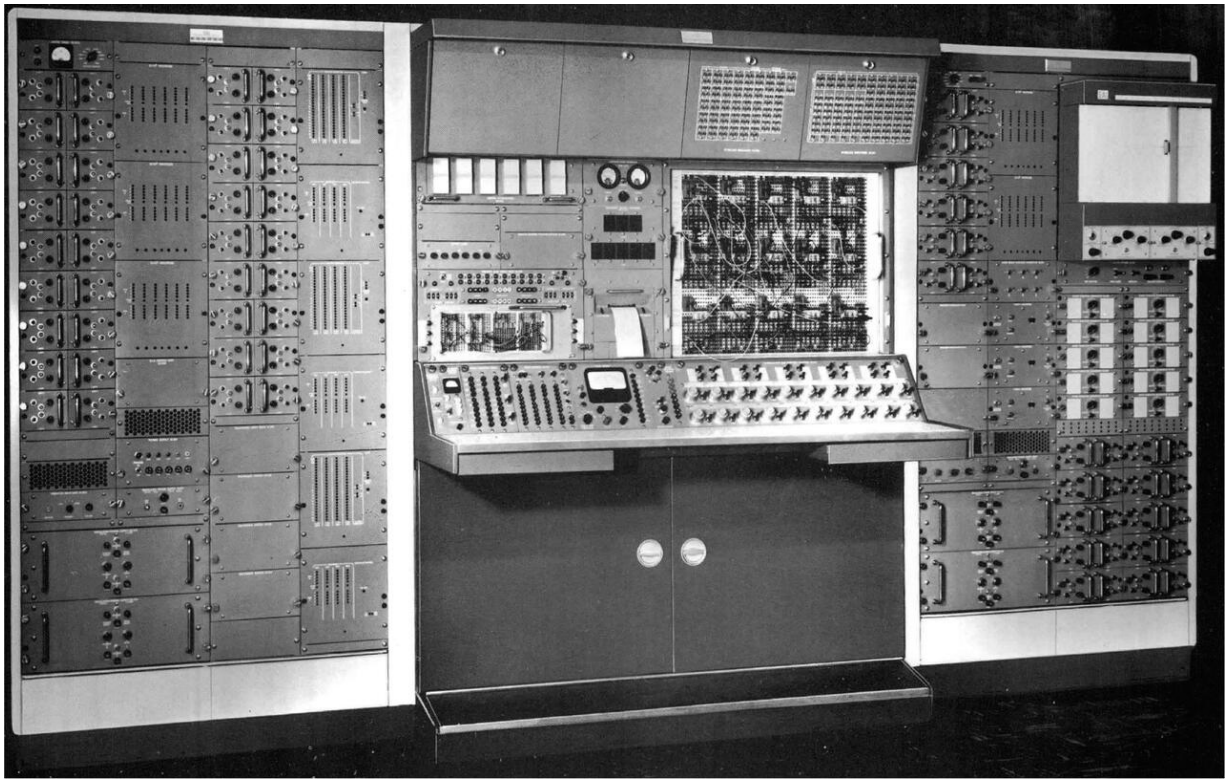


Figure 6.9. EAI 231RV system ([→EAI 1964/1, p. vi])

→Figure 6.10 shows a *HYDAC 2000* system, consisting of an EAI 231RV analog computer with several expansion racks and an eight channel recorder on the right. On the left a *DOS 350* digital console can be seen. This consists of a plethora of simple digital circuit elements such as individual gates, flip-flops, monostable multivibrators, etc., that can be interconnected by means of a central digital patch panel similar to that used in the analog computer. Using these elements, quite complex control circuits can be implemented.



Figure 6.10. EAI HYDAC-2000 system (see [→Elektronische Rechenanlagen 1964])

Such a HYDAC system, costing over one million Dollars,[14] was acquired by General Dynamics in 1964 for the development of the F-111 jet fighter.[15]

6.4 Early transistorized systems

Telefunken was the first company that successfully developed a fully transistorized analog computer. After the rather unexpected commercial success of the vacuum tube based RA 463/2, the company's management decided that the era of vacuum tubes was over – a pretty bold decision in the mid 1950s when transistors were still exotic devices with widely varying performance characteristics. A project to develop a transistorized high-performance operational amplifier with chopper-stabilization was started. (See →section 4.1.2 for a description of this amplifier. The much simpler amplifiers in the

RA 1 and its corresponding production model RA 463/2 had no such drift compensation. Accordingly, these machines were capable of repetitive operation only with a maximum integration time of 110 seconds. Longer runs would compromise the results too much due to drift effects.) This development was undertaken by Günter Meyer-Brötz, who just finished university,[16] and Hans Otto Goldmann.

A small tabletop analog computer was built as proof-of-concept. The resulting system is shown in →figure 6.11. It is clearly a direct descendant of the RA 1 and RA 463/2 – there is no central patch panel, all computing elements are housed in separate modules, and the machine features a built-in oscilloscope.

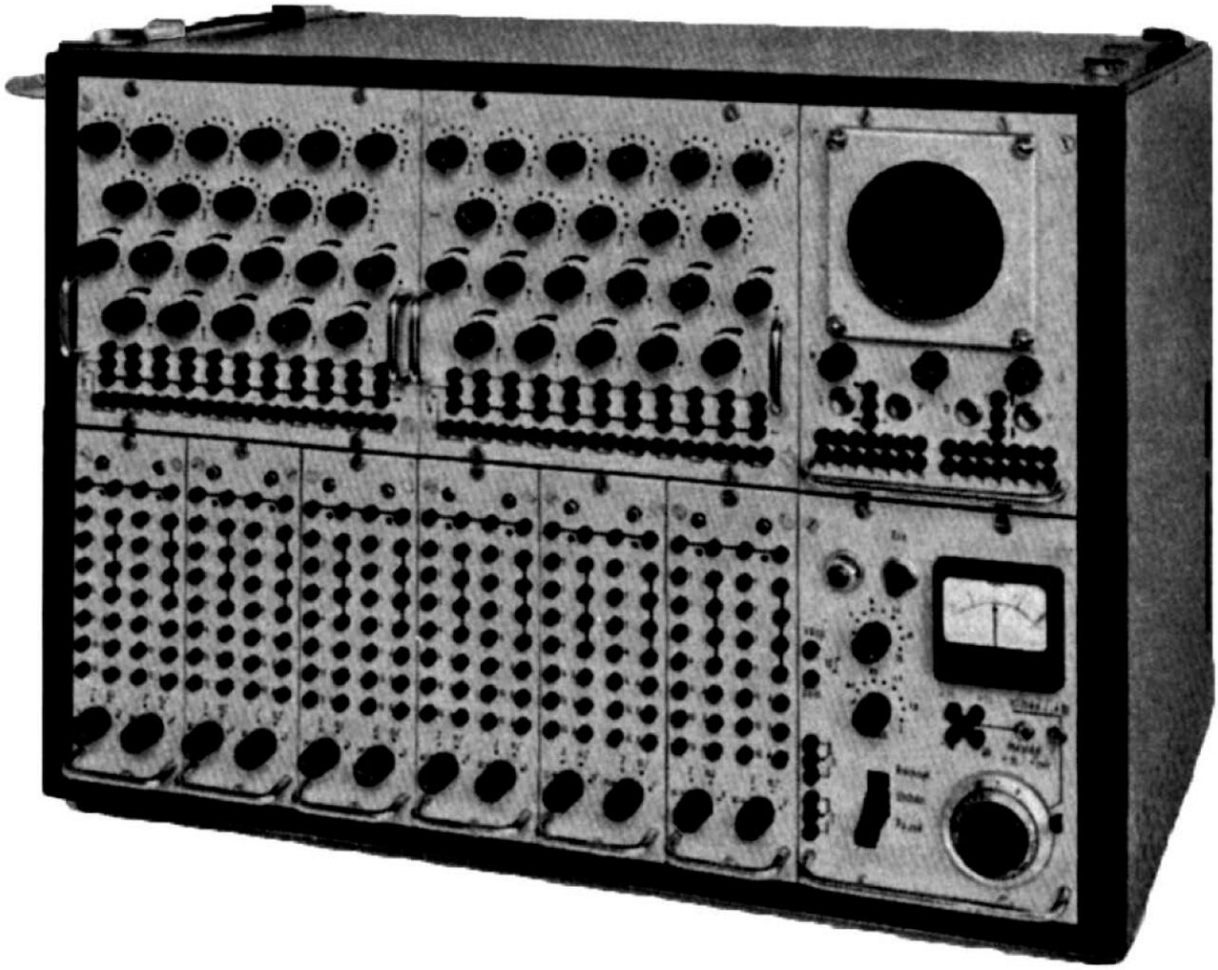


Figure 6.11. First fully transistorized tabletop analog computer made by Telefunken (see [→ Ernst 1960, p. 255])

→ Figure 6.12 shows one of the modules of this early transistorized prototype sitting in front of a comparable module built with traditional vacuum tubes.[17] Compared with similar modules from other vendors of this time, it is remarkably small with its chopper-stabilized operational amplifiers.

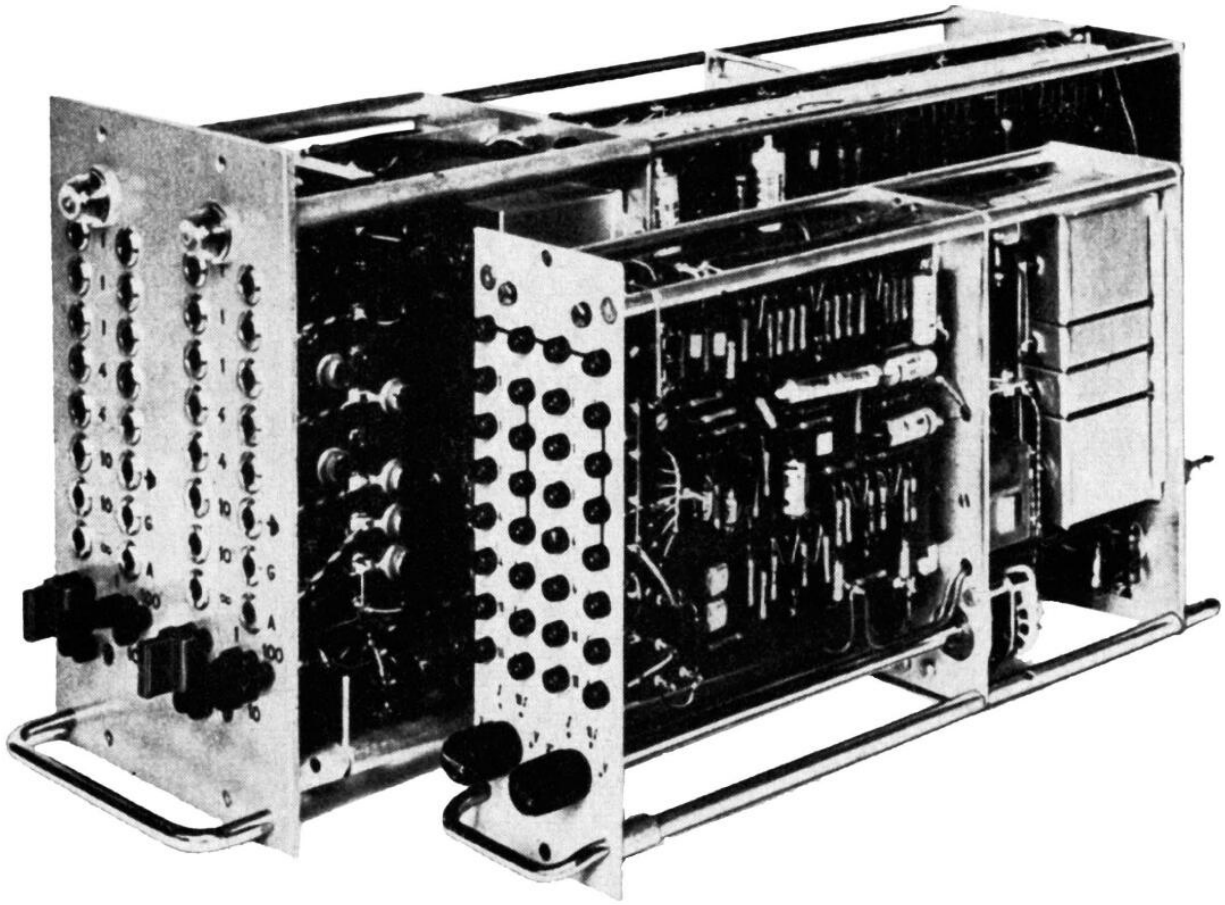


Figure 6.12. Transistorized dual integrator/summer/open amplifier module shown in front of a comparable module using vacuum tubes (see [→Mohr 1959])

It was decided to develop a line of transistorized analog computers based on this prototype. The first such machine was the *RAT 700*.[18] The operational amplifier used in this first fully transistorized analog computer has been shown in →chapter 4, →figure 4.16. The RAT 700 was first sold in 1959 and became the progenitor of a long and commercially successful series of analog tabletop computers built by Telefunken. →Figure 6.13 shows one of the early machines sold in 1959.[19]

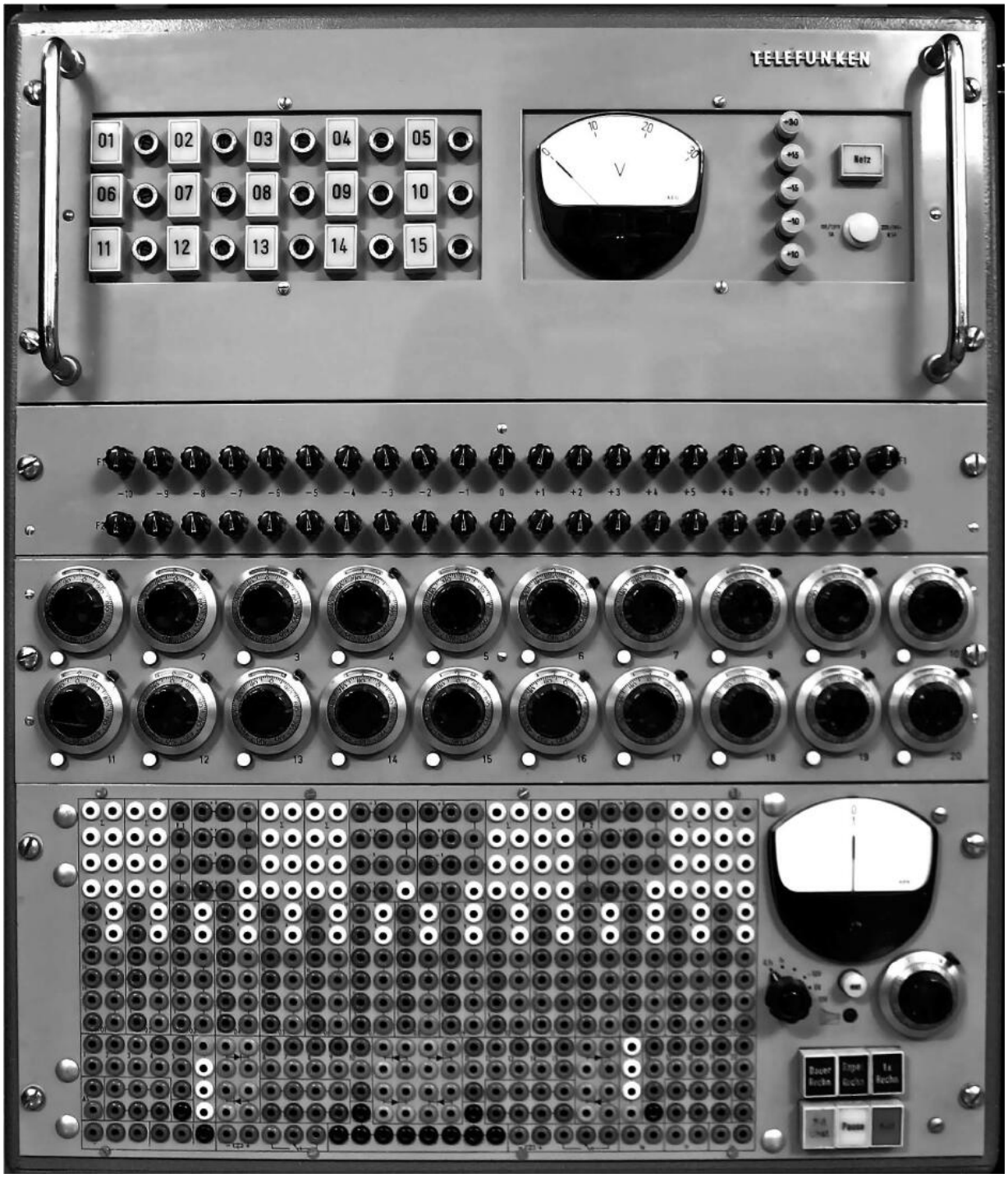


Figure 6.13. The RAT 700 tabletop analog computer

In contrast to the RA 1 and RA 463/2 this tabletop computer features a central patch panel[20] located in the lowest of four

chassis-mounted modules, which contains the input and feedback networks of the summers and integrators and additional computing elements such as free diodes, relays, etc. The operator control panel can be seen on the right-hand side of this module. The middle section of the machine holds two half-height drawers containing 20 precision coefficient potentiometers[21] and two diode function generators with fixed breakpoints. The top drawer holds the power supplies (±15 V, −25 V for relays, and 6 V for lamps, and the highly stabilized machine units of ±10 V) on the right and 15 chopper-stabilized precision operational amplifiers with their corresponding overload indicators on the left.

A fully expanded RAT 700 contains 19 operational amplifiers,[22] four quarter square multipliers, two diode function generators, 20 coefficient potentiometers, eight free diodes, and two relays that can be used to set up comparators in conjunction with a summer.

The RAT 700, which sold remarkably well, was in fact just a by-product of the development of a large precision analog computer that was completed in 1960 and introduced to the public at the Hannover trade show in the same year. This machine, the *RA 800*, shown in its basic configuration in →figure 6.14, was living proof that a precision analog computer with a machine unit of ±10 V could be built with transistors instead of the then ubiquitous vacuum tubes. Around this time, other companies started their own developments of transistorized analog computers to compete with the RA 800.

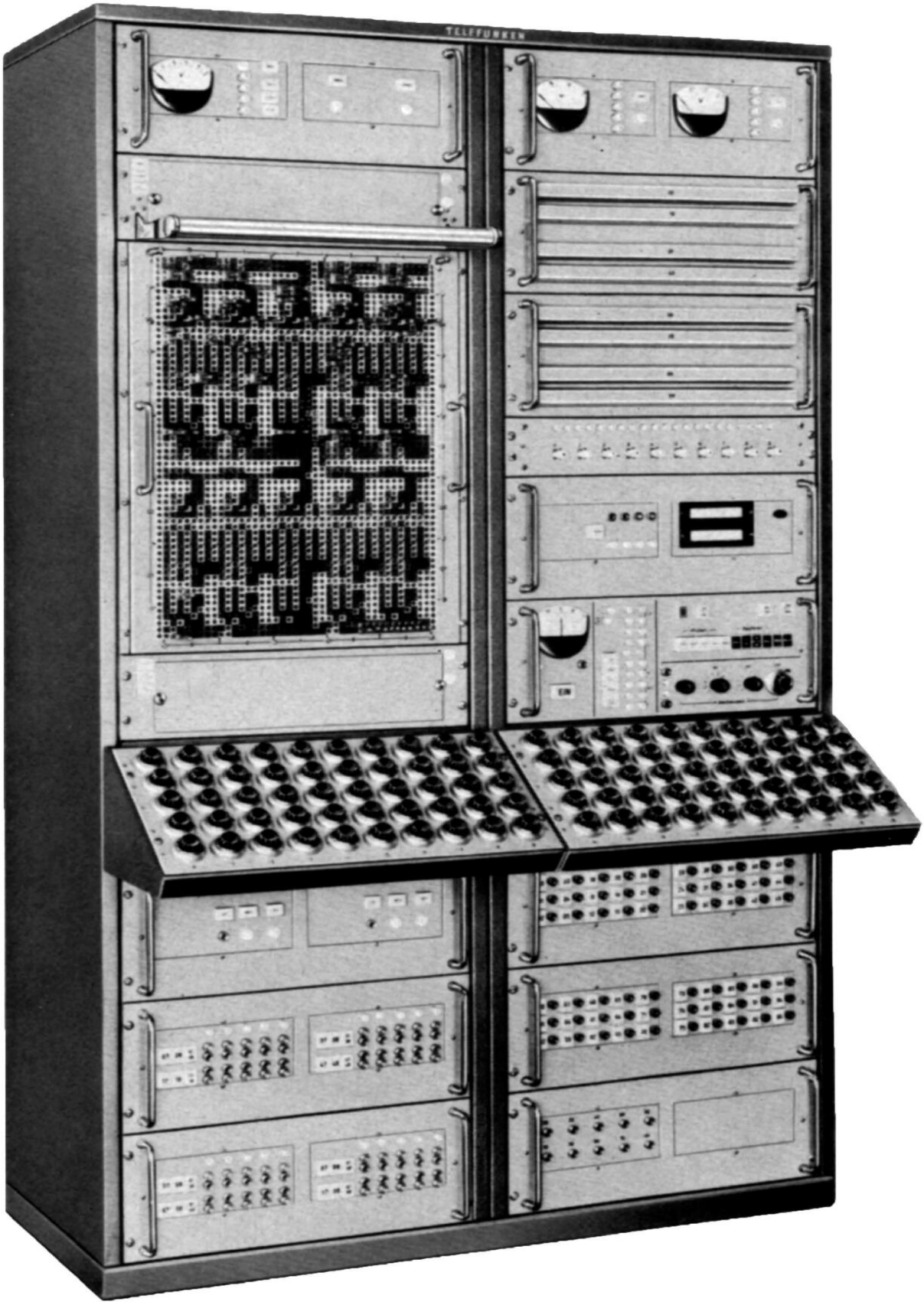

TELEFUNKEN

Figure 6.14. A basic RA 800 precision analog computer (see [→ MEYER-BRÖTZ 1960, p. 176])

The left rack contains, from top to bottom, a power supply, the central removable patch panel, 50 precision coefficient potentiometers, two servo resolvers and two drawers each with four time division multipliers. The right rack contains, top to bottom, a power supply, two drawers each containing four diode function generators, ten manual switches, a digital voltmeter, the central control panel with various timers, a compensation voltmeter and computing element readout selection logic, 50 additional potentiometers, two drawers each holding 30 operational amplifiers, and ten electronic comparators.

At the same time, EAI was developing a transistorized operational amplifier that would be the heart of its first tabletop analog computer, the *TR-10*, which was introduced to the market in 1960. → Figure 6.15 shows the front view of a TR-10. Instead of a central patch panel, it exposes its various components to the programmer.[23] Interestingly, this machine – like several other later systems made by EAI – does not offer integrators and summers as abstract computing components but requires the user to set up these elements explicitly by interconnecting operational amplifiers with suitable passive networks.[24]

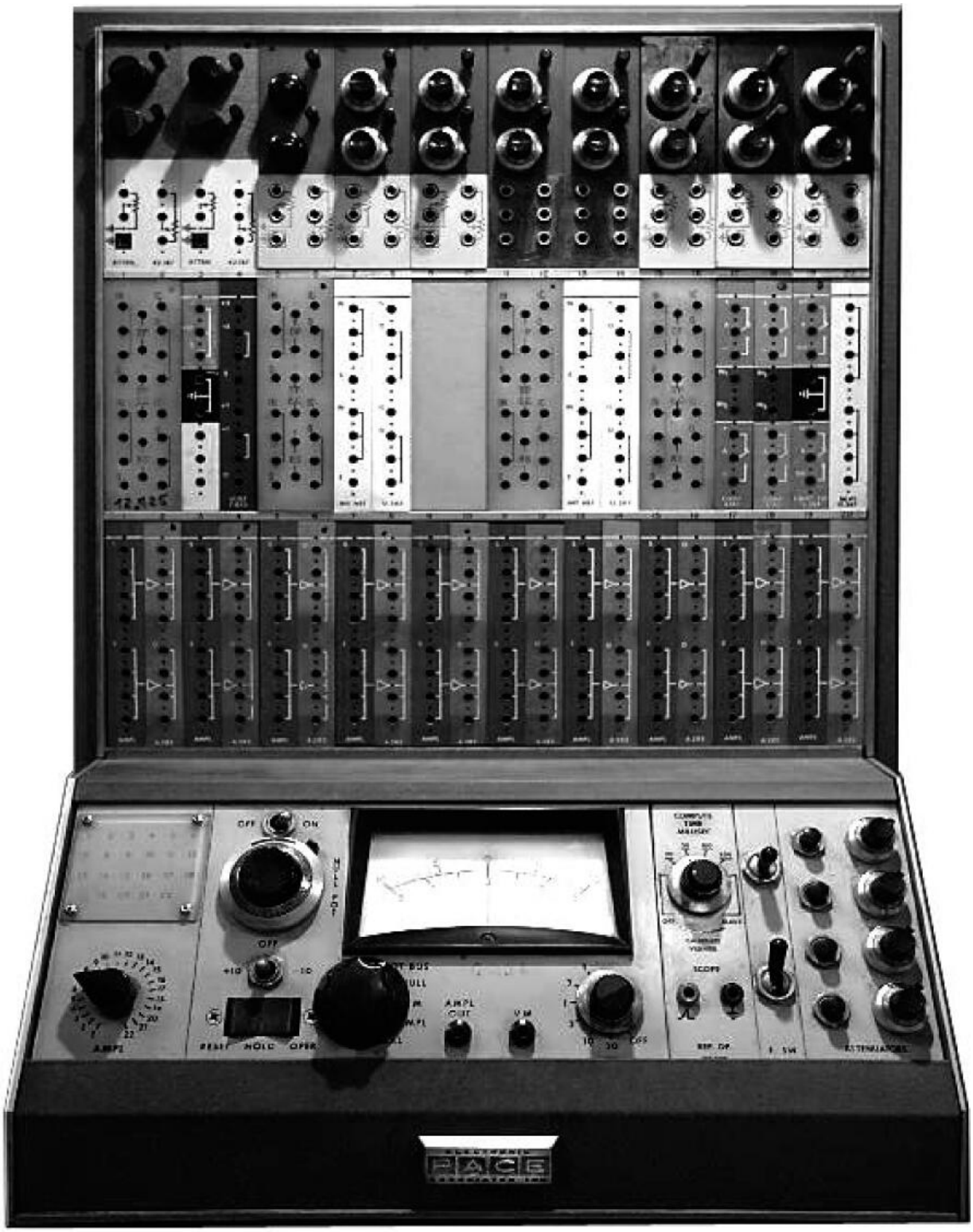


Figure 6.15. The EAI TR-10 tabletop analog computer

From top to bottom, this particular TR-10 contains 20 coefficient potentiometers, twelve computing networks to build summers, integrators, etc., ten dual operational amplifiers that have to be connected to the computing networks, and the control panel at the bottom.

6.5 Later analog computers

In the mid 1960s analog computers were increasingly often coupled with digital computers, forming *hybrid computers*.[25] Accordingly, most manufacturers started development of these systems by augmenting existing analog computers with the necessary hardware interfacing components such as analog-digital and digital-analog converters, etc. A typical example of this strategy is the EAI HYDAC 2000 system shown in →figure 6.10. Telefunken used a similar approach by evolving the RA 800 into the *RA 800H*, the *H* denoting its hybrid capability. In fact, this machine was basically still a RA 800 system but had a digital control system similar to the DOS 350 of the HYDAC-2000.

→Figure 6.16 shows a RA 800H system. The rack on the left holds the digital extension, the *DEX 802*.[26] This is basically just a collection of simple digital elements like logic gates, counters, and timers. In addition to this, there are drawers containing analog function generators, an electronic resolver, etc. Thanks to the seemingly simple DEX 802 the RA 800H was a much more versatile system than its RA 800 predecessor. It was possible to solve directly complex tasks such as parameter optimizations on the RA 800H.

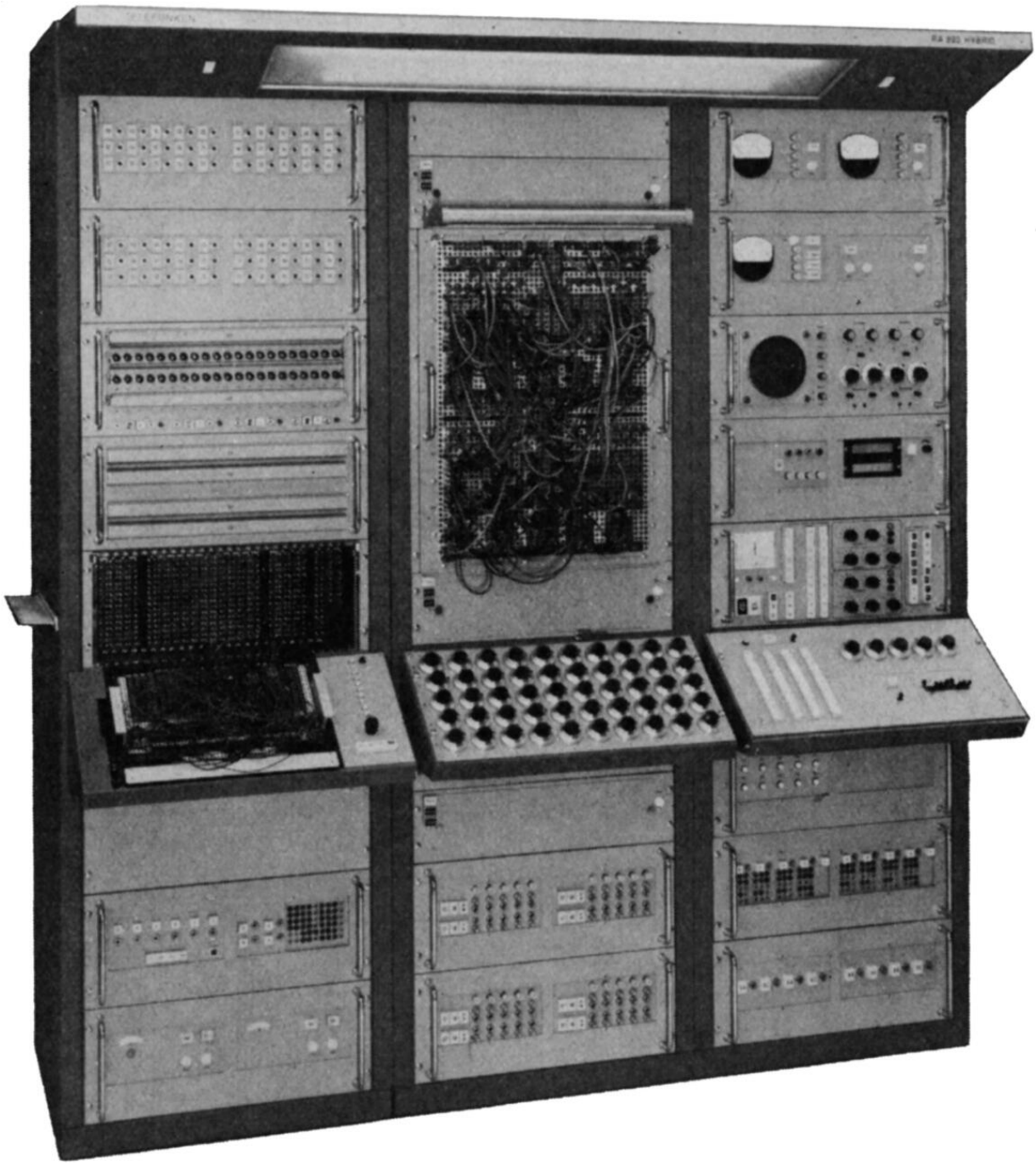


Figure 6.16. A RA 800H system (see [→ADLER 1968, p. 273])

The last Telefunken analog computer is the RA 770, unveiled in 1966, which was a significant improvement over the RA 800H. Shown in →figure 6.17, this system achieves an overall precision

of 10^{-4} thanks to improved operational amplifiers employing silicon transistors and fully electronic choppers.[27] Using integrated circuits in the control system and other improvements made the RA 770 a physically much smaller and yet more capable system, with a wider complement of computing elements, than the RA 800H.

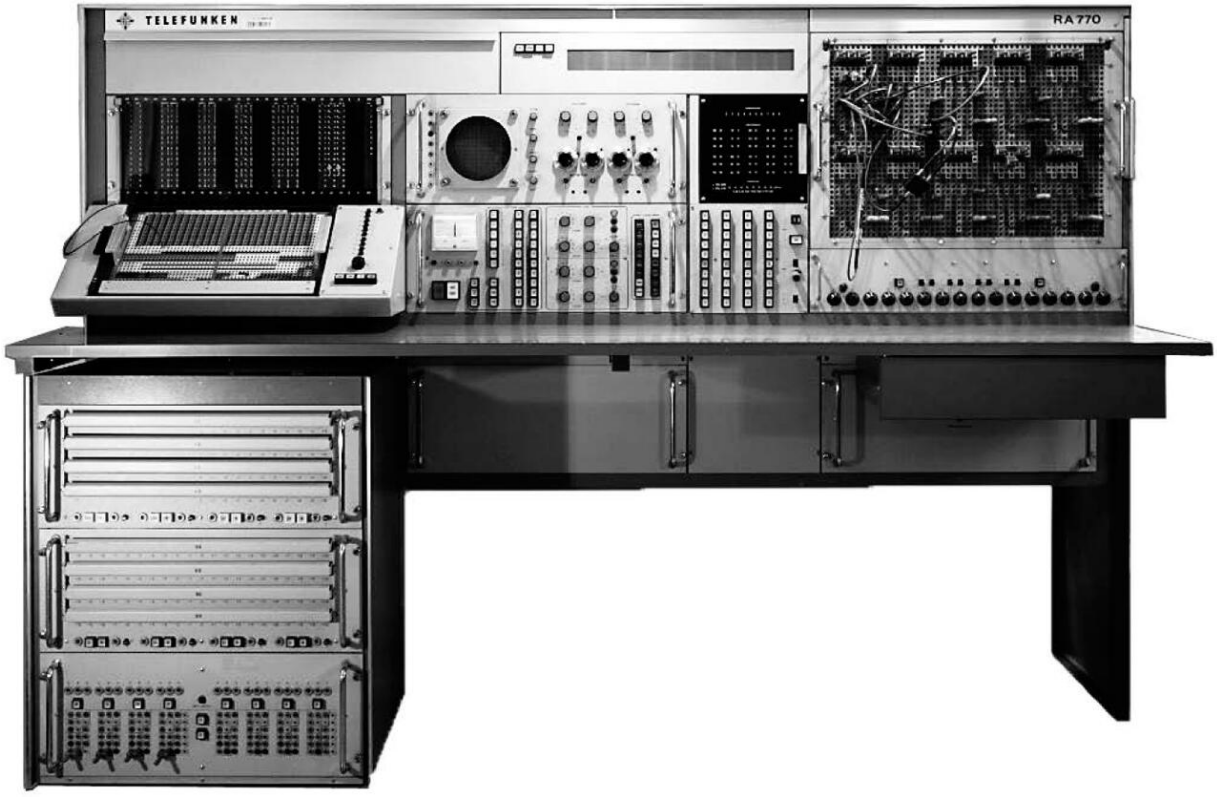


Figure 6.17. Telefunken RA 770 precision analog computer

Both the RA 770 and the RA 800H can also be coupled with a stored-program digital computer by means of a *hybrides Koppelwerk*[28] *HKW 860* to form a true hybrid computer. Unfortunately, Telefunken stopped the development of analog computers in the early 1970s with no successor to the venerable RA 770. Its analog computer branch went out of business in the

1970s while other manufacturers, especially EAI, were still bringing new and enhanced systems to the market.

An example of a late EAI system, a *PACER 500* hybrid computer, is shown in →figure 6.18.[29] From left to right, the console typewriter for the stored-program digital processor, the digital processor itself, the dominating analog patch field, the operator control panel, including a small digital patch field, and a small cathode ray tube display can be seen. Systems like these were in widespread use in many scientific and engineering branches well into the 1980s.

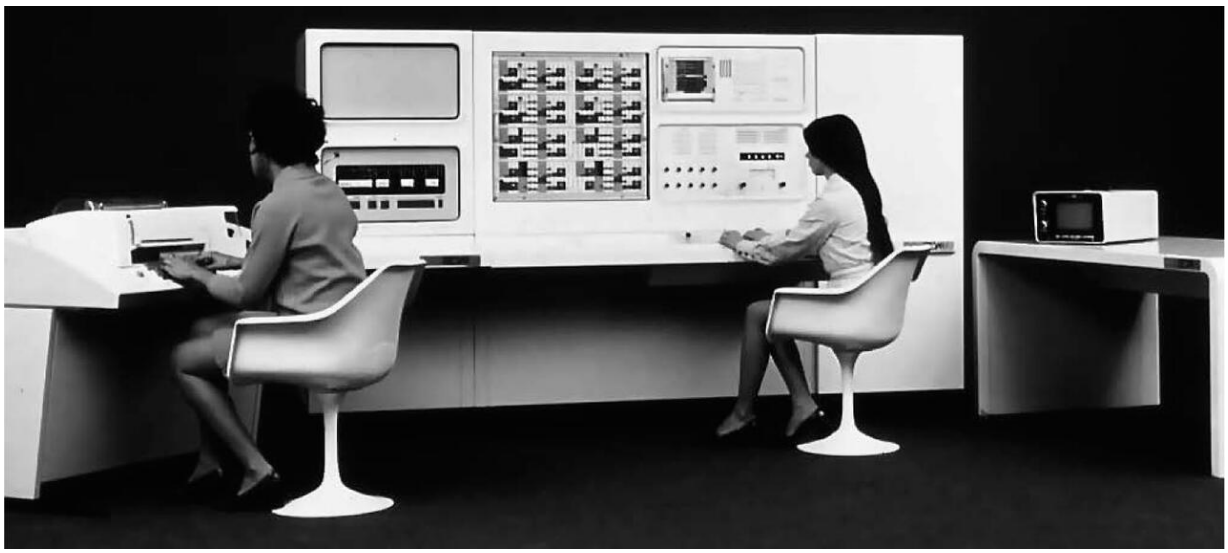


Figure 6.18. An EAI PACER 500 hybrid computer system

6.6 THE ANALOG THING

A noteworthy recent development[30] is *THE ANALOG THING* (*THAT*), a small-scale analog computer ideally suited for educational and hobbyist use.[31] Shown in →figure 6.19 this little analog computer contains five integrators, each with two time-scale factors, four summers, four sign-inverters, eight coefficient potentiometers,

two multipliers, two comparators with electronic switches, and several free elements such as diodes, capacitors, and resistor networks. It also supports manual operation as well as repetitive operation so that solutions can be displayed as steady figures on an oscilloscope.



Figure 6.19. Front view of THE ANALOG THING

One especially interesting feature of this system is that it is an open hardware project, so its schematics are publicly available[32] and can be used as a starting point for other developments. In fact, its circuitry is simple enough to serve as an example on how a basic analog computer can be built.[33]

6.7 LUCIDAC and REDAC

One of the most recent developments in modern hybrid computing is the *LUCIDAC* by German company anabrid GmbH, shown in →figure 6.20. This system is fully reconfigurable and contains eight integrators with $k_0 \in \{10^2, 10^4\}$, four multipliers, and 32 12 bit coefficients. Its architecture, described in detail in →section 11, is based on two crossbar matrices, one voltage and one current coupled. The latter matrix performs implicit summation of currents, thus making summers as explicit computing elements unnecessary. The system features eight analog inputs and eight analog outputs and can, thus, be used as part of a hardware-in-the-loop system as well as a stand-alone device. It is configured by `json` messages sent via an Ethernet port. Values represented as voltages are within the interval of ±2 V.



Figure 6.20. LUCIDAC – a reconfigurable desktop hybrid computer

7 Programming

Programming an analog computer differs substantially from the algorithmic approach used for stored-program digital computers. Programming an analog computer can be considered being easier and maybe even more “natural” than developing and implementing an algorithm on a digital computer. This is especially true for engineers, physicists, and mathematicians, since the analog computer is, due to its nature of forming an analog for a problem to be solved, closely related to their realms of expertise and their ways of thinking about complex dynamical systems.

From the perspective of a contemporary programmer, programming an analog computer will look like an arcane task, as it bears no resemblance at all to today’s ubiquitous algorithmic approach. Since analog computing is not part of the standard curriculum in a computer science class, programming such a machine is considered a curiosity while the much more abstract algorithmic approach has become natural.

The following sections are an introduction to analog computer programming and are based on practical examples of increasing complexity.[1] Since programming, be it analog or algorithmic, is to a certain degree an art form, becoming an expert analog computer programmer requires a lot of hands-on experience.

One of the biggest hurdles for aspiring analog computer programmers is to overcome the deeply ingrained algorithmic approach to programming, which has been the predominant approach to programming since the late 1970s.

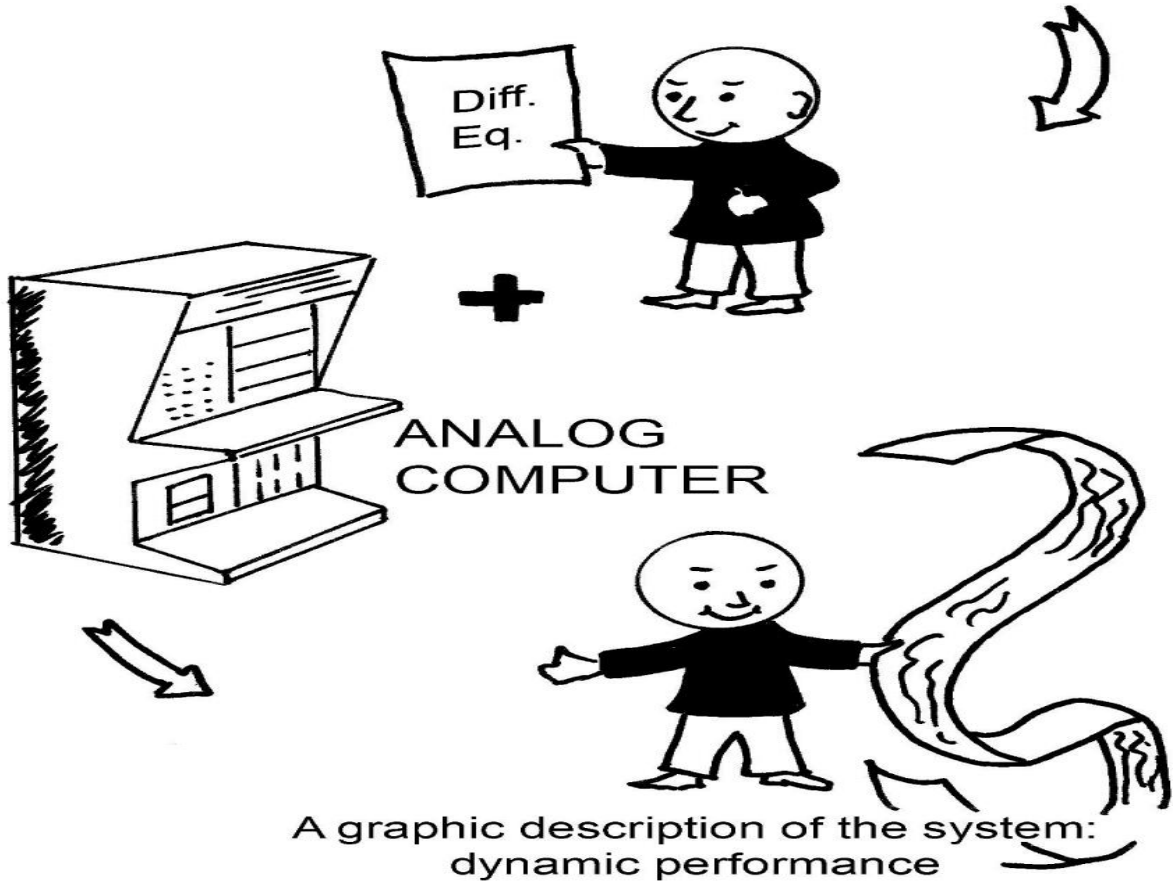
7.1 Basic approach

→Figure 7.1 shows the basic approach to programming an analog computer: First of all a thorough analysis of the system to be simulated is required, which typically results in a differential equation, or systems of coupled differential, or partial differential, equations.[2] Based on these equations a *program* for the analog computer is then devised. This program is a description of the connections required between the analog computer's various computing elements to form an electronic analog of the underlying problem. Since the variables within such a program are limited to the interval [−1, 1] a *scaling* or *normalization* step is typically necessary to ensure that no variable within the program will exceed the machine units.[3] This can often be quite tricky. Scaling a problem will not only change the values of coefficients but also introduce additional coefficients into the computer setup.

A PHYSICAL SYSTEM can be SIMULATED BY AN ANALOG COMPUTER



PHYSICAL LAWS



A graphic description of the system:
dynamic performance

Figure 7.1. Typical workflow in analog simulation (cf. [→ Truitt et al. 1960, p. 1-108])

Given sufficient initial conditions for the integrators in the computer setup, the system of which an *analog* has been implemented can now be simulated.

There are two main approaches to derive an analog computer program from a set of equations, either the *feedback technique*[4] developed by Lord Kelvin in 1875/1876 or the *substitution method*.[5] Both methods start with a set of differential equations of the general form

$$\frac{d^n}{dx^n}y(x) + a_{n-1}\frac{d^{n-1}}{dx^{n-1}}y(x) + \cdots + a_1\frac{d}{dx}y(x) + a_0 y(x) = f(x) \tag{7.1}$$

or shorter

$$y^{(n)}(x) + a_{n-1}y^{(n-1)}(x) + \cdots + a_1 y'(x) + a_0 y(x) = f(x)$$

with initial conditions

$$y(0), y^{(1)}(0), y^{(2)}(0), \ldots, y^{(n-1)}(0)$$

representing the system to be solved with the analog computer. Derivatives with respect to time *t* or the scaled *machine time* τ, such as $\frac{\mathrm{d}y}{\mathrm{d}t}$ or $\frac{\mathrm{d}y}{\mathrm{d}\tau}$, will be written here as $\dot{y}$.[6]

In an analog computer program differentials are typically avoided in favor of integrations since a (time) derivative of signal amplifies noise (a differentiator basically is a high-pass filter) while an integration filters noise out (low-pass filtering).[7]

7.2 Kelvin’s feedback technique

KELVIN's feedback technique, which has also been called the *classical differential analyzer technique*,[8] is based on the idea of a series of integrations starting with the highest derivative of a differential equation, successively yielding all lower derivatives necessary to represent the equation in question. As an example the following equation with initial conditions $\ddot{y}(0)$, $\dot{y}(0)$, and $y(0)$ is to be solved with an analog computer:

$$\dddot{y} + a_2\ddot{y} + a_1\dot{y} + a_0 y = 0 \tag{7.2}$$

First, the equation is rearranged so that its highest derivative is separated on the left-hand side of the equation sign:

$$\dddot{y} = -a_2\ddot{y} - a_1\dot{y} - a_0 y. \tag{7.3}$$

Using three integrators connected in series as shown in →figure 7.2, the lower derivatives $-\ddot{y}$, $\dot{y}$, and $-y$ can be computed.[9]

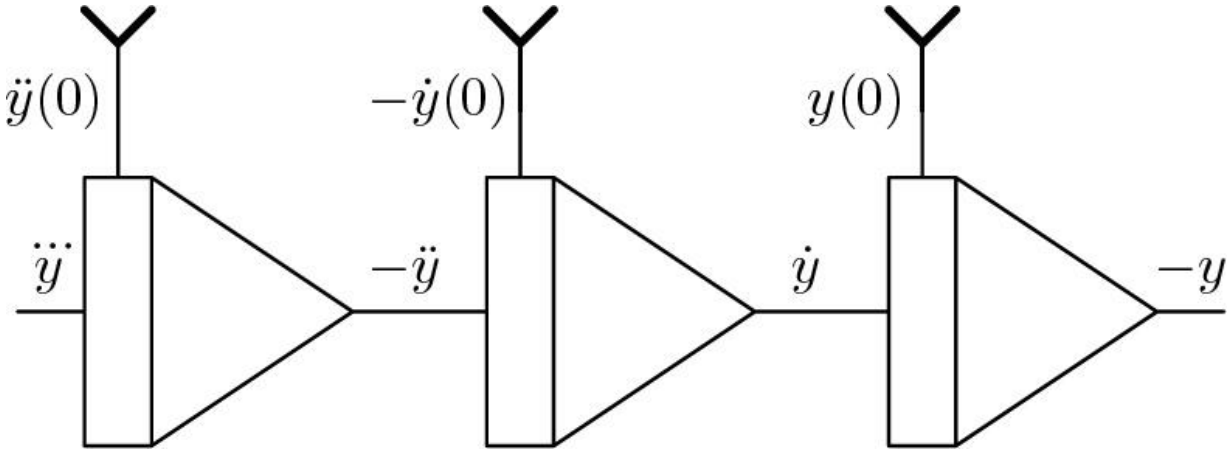


Figure 7.2. Kelvin's feedback technique, 1st step

Combining these derivatives, the right-hand side of equation (→7.3) can be obtained as shown in →figure 7.3. The coefficients a_0, a_1, and a_2 are implemented using coefficient potentiometers

while an additional summer is necessary to yield $-\dot{y}$. Since every integrator (and summer) performs a sign reversal, the three initial conditions have to be supplied with alternating signs in this setup, too.

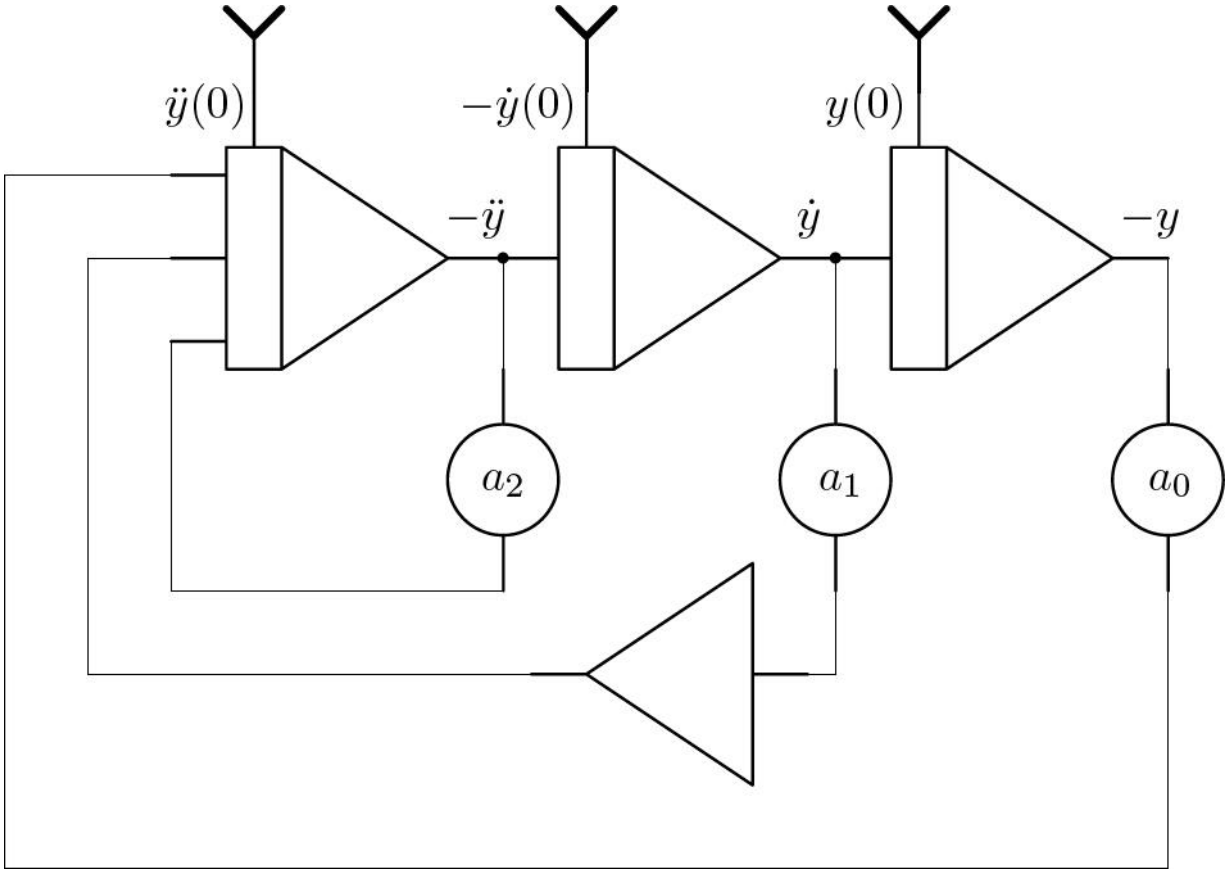


Figure 7.3. Kelvin's feedback technique, 2nd step

The computer setup derived by this simple method is now an analog of the system described by the differential equation (→7.2). Depending on its scaling, i. e., the coefficients a_0, a_1 and a_2 and the time-scaling factors of the integrators, the desired solution can be typically obtained in the same period of time the original system would behave. It is also possible to increase or decrease the time-scale factors yielding the solution faster or

slower than real-time. While many problems, especially those with hardware- or humans-in-the-loop, need to run in real-time and therefore do not require time scaling, other problems benefit considerably from stretching or compressing time.[10]

Summarizing the KELVIN feedback technique, an analog computer setup can be derived from a set of equations by executing the following steps:

1. Rearrange the equations so that the highest derivatives are separated on the left-hand side.
2. Generate all lower derivatives that are part of the equations by successive integrations.
3. Using these lower derivatives, the terms on the right-hand sides of the equations can be generated.
4. Since these right-hand sides are equal to the highest derivatives used in the first step, the feedback loops can now be closed by feeding these values into the inputs of the respective first integrator stages.

7.3 Substitution method

Application of the *substitution method*[11] is best shown by deriving a computer setup for solving a differential equation like[12]

$$\ddot{y} + a_1\dot{y} + a_0 y = f(t). \tag{7.4}$$

To simplify matters, the initial conditions $\dot{y}(0)$ and $y(0)$ are assumed to be zero.

The basic idea is to transform this differential equation into equations of first degree by repeated substitutions. Each substitution decreases the degree of the remaining differential equation by one. The substitution

$$\eta = \dot{y} + a_1 y \tag{7.5}$$

yields

$$\dot{\eta} = \ddot{y} + a_1 \dot{y} \tag{7.6}$$

as its first derivative with respect to time. Substituting (→7.6) into (→7.4) yields the inhomogeneous differential equation

$$\dot{\eta} + a_0 y = f(t). \tag{7.7}$$

Solving (→7.7) for its highest derivative yields

$$\dot{\eta} = f(t) - a_0 y.$$

which corresponds to the circuit shown in →figure 7.4. Now a second circuit representing equation (→7.5) has to be set up. Solving this equation for its highest derivative yields

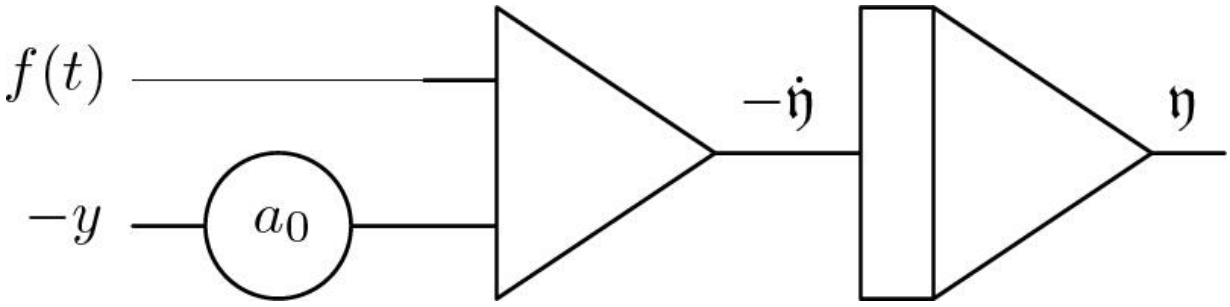


Figure 7.4. First sub-circuit derived by the method of substitution

$$\dot{y} = \eta - a_1 y,$$

which corresponds to the circuit shown in →figure 7.5. $\mathfrak{y}$ is readily available from the circuit shown in →figure 7.4.

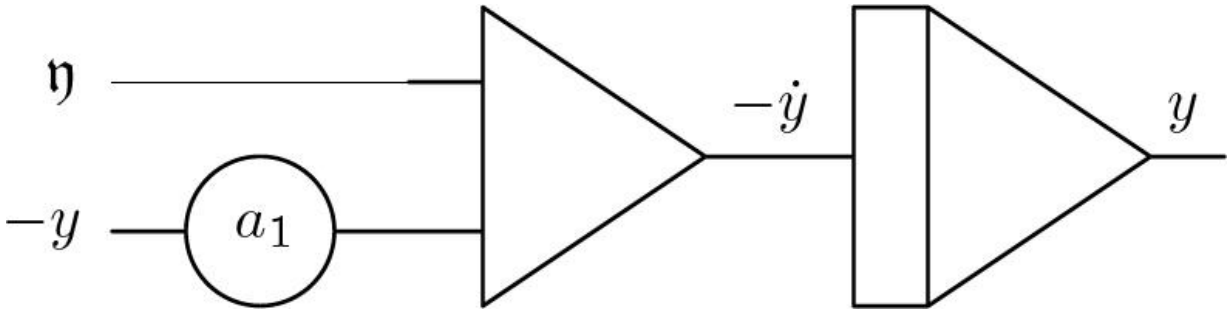


Figure 7.5. Second sub-circuit derived by the method of substitution

Combining these two sub-circuits in a straightforward fashion yields the setup shown in →figure 7.6. Realizing that every summer and integrator performs a sign inversion, this circuit can be simplified by removing two summers.[13] The resulting simplified setup is shown in →figure 7.7.

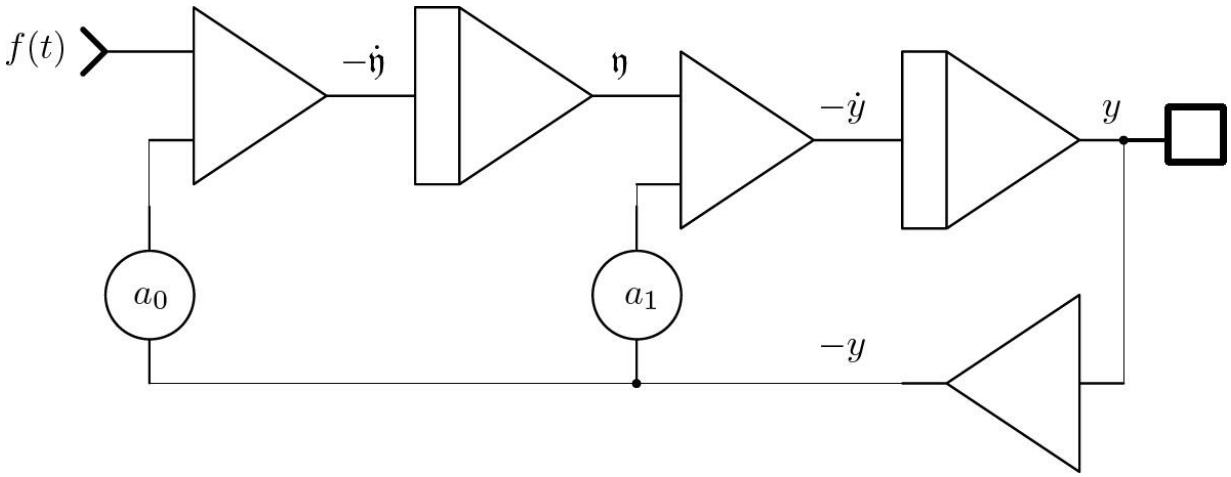


Figure 7.6. Non-optimized circuit derived by the substitution approach

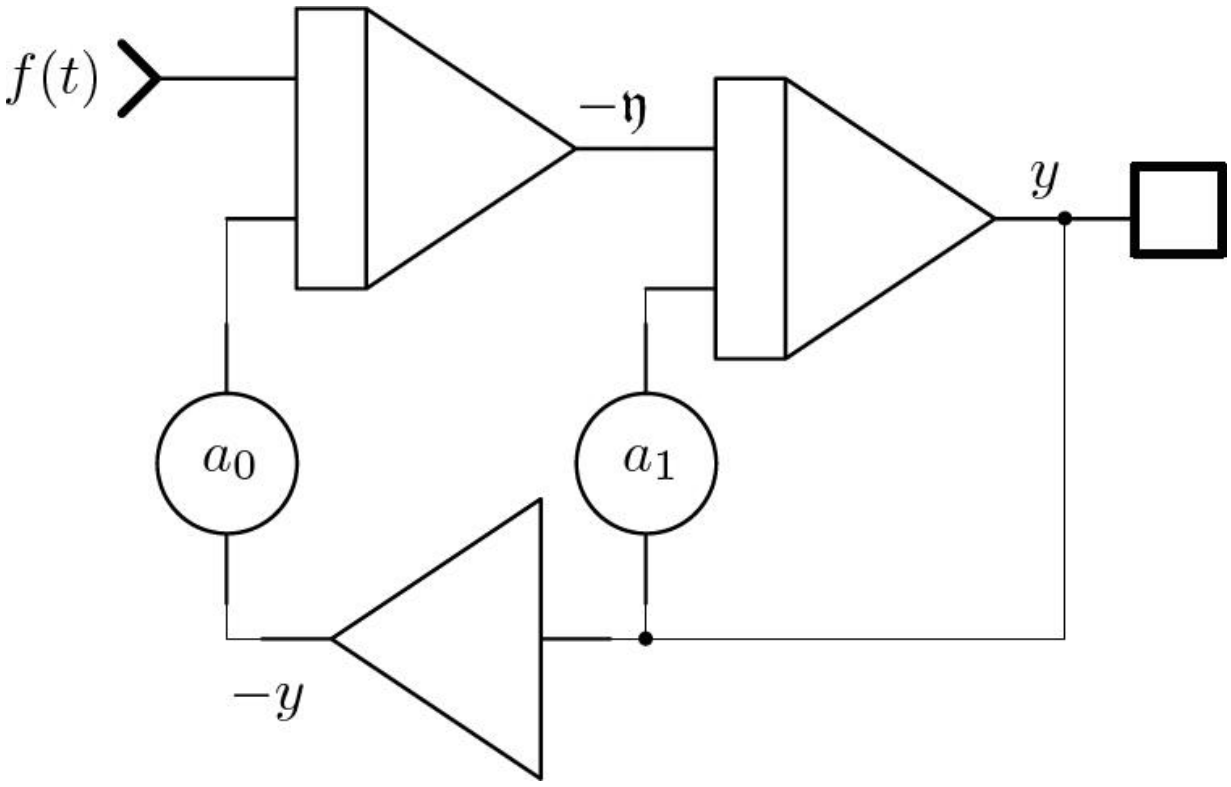


Figure 7.7. Optimized circuit

A major drawback of the substitution method is the complex treatment of non-zero initial conditions. While the initial condition $y(0)$ can be used directly for the last integrator in a program derived by this method, all other initial values have to be transformed according to the previously executed substitutions. So the initial value $\dot{y}(0)$ has to be transformed into an equivalent initial value

$$\dot{\eta}(0) = \dot{y}(0) + a_1 y(0).$$

$\mathfrak{y}$ can now be used to initialize the corresponding integrator. This initial value transformation makes the application of the substitution method quite cumbersome in real applications. Consequently, the following sections will use the simpler KELVIN method.

In general, both techniques, KELVIN's feedback method as well as the substitution method, can be applied to linear and nonlinear differential equations alike.[14]

7.4 Partial differential equations

Many problems are readily described by partial differential equations containing derivatives with respect to more than one variable.[15] Since an analog electronic analog computer can only use time as the free variable of integration,[16] partial differential equations (*PDE*s) are normally solved by either employing a quotient of differences approach or by separation of variables.[17]

7.4.1 Quotient of differences

The *quotient of differences* method is analogous to simple numeric algorithmic approaches. Derivatives with respect to another variable than time *t* are approximated as quotients thus discretizing the underlying problem.[18] The derivative

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x)-f(x)}{\Delta x}$$

is approximated by a quotient of differences:

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} \approx \frac{f(x+\Delta x)-f(x)}{\Delta x}$$

As simple as this method is, it usually requires a large number of computing elements, which – especially on classic analog computers – may outweigh the advantage of simplicity. →Figure 7.8 shows a simple one-dimensional heat-transfer problem being solved this way.[19]

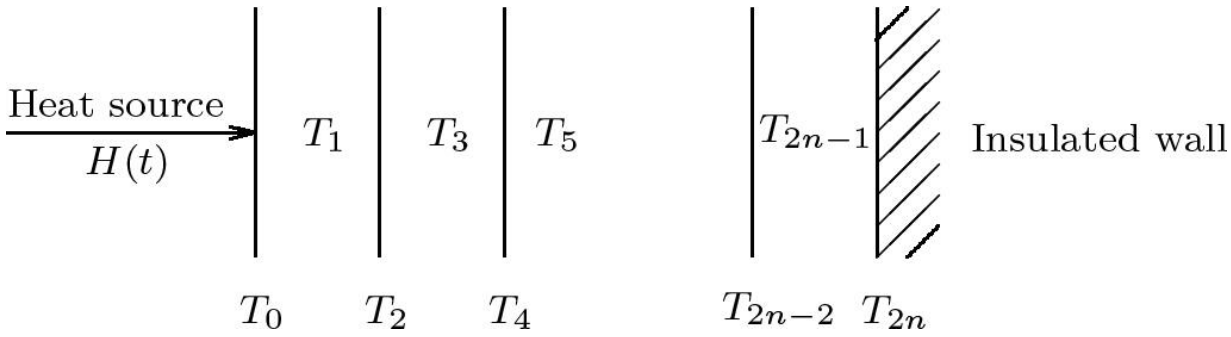


Figure 7.8. One-dimensional heat-transfer

A heat conducting rod is connected to a time dependent heat source $H(t)$ on its left end while the right end is placed against an insulating wall. This problem can be described by the PDE

$$\frac{\partial^2 T}{\partial x^2} = k\dot{T}$$

with $T = T(x, t)$ denoting the temperature at position x and time t. (Additional examples can be found in [→ Seyferth 1960, pp. 3-66 ff.], [→ Truitt et al. 1960], [→ Ameling 1963/1, pp. 284 ff.], [→ Sydow 1964, pp. 103 ff.], [→ MacKay et al. 1962, pp. 243 ff. and pp. 293 ff.], [→ Telefunken 1963/1], [→ Giloi et al. 1963, pp. 255 ff.], and [→ Ulmann 2023/1, pp. 180 ff.]. A more complex gas flow example is described in [→ Mahrenholtz 1968, pp. 143 ff.].)

The rod is now divided into slices of equal width as shown in →figure 7.8. Each of these slices is represented by a difference quotient. The heat source on the left and the insulator on the right side yield the following boundary conditions:

$$\dot{T}_0 = \frac{1}{k\Delta x} H(t) \tag{7.8}$$

$$\dot{T}_{2n} = \frac{1}{k\Delta x}(T_{2n-1} - T_{2n-2}). \tag{7.9}$$

The resulting quotient terms with indices $2m$ with $1 \leq m < n$ are

$$\frac{\partial^2 T_{2m}}{\partial x^2} \approx \frac{T_{2m-1} - 2T_{2m} + T_{2m+1}}{\frac{1}{4}(\Delta x)^2}\,. \tag{7.10}$$

with the interjacent elements being

$$T_{2m+1} = \frac{T_{2m} + T_{2m+2}}{2}\,.$$

T_{2m+1} represents the mean temperature on the edges of the corresponding slice in →figure 7.8. Starting with equation (→7.10), a system of differential equations involving only derivatives with respect to time can be derived. Assuming $n = 3$ and taking the boundary conditions (→7.8) and (→7.9) into account yields the following set of differential equations:

$$\begin{aligned}
\dot{T}_0 &= \frac{1}{k\Delta x} H(t) \\
T_1 &= \frac{T_0 + T_2}{2} \\
\dot{T}_2 &= \frac{4}{k(\Delta x)^2} (T_1 - 2T_2 + T_3) \\
T_3 &= \frac{T_2 + T_4}{2} \\
\dot{T}_4 &= \frac{4}{k(\Delta x)^2} (T_3 - 2T_4 + T_5) \\
T_5 &= \frac{T_4 + T_6}{2} \\
\dot{T}_6 &= \frac{1}{k\Delta x} (T_5 - T_4)
\end{aligned}$$

These equations can be readily transformed into the computer setup shown in →figure 7.9. The large number of computing elements is obvious.[20] This method has the advantage that the

coefficient potentiometers can be set up in groups, thereby simplifying the setup task significantly.

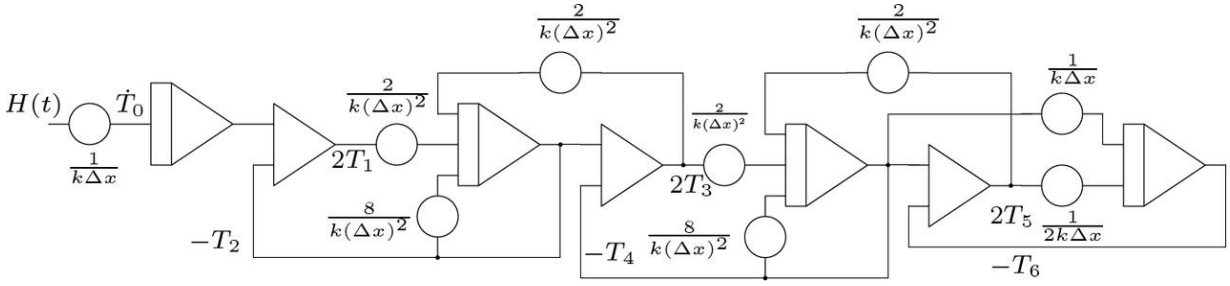


Figure 7.9. Circuit for one-dimensional heat-transfer

When the derivatives not depending on *t* are of higher degree, the method of difference quotients becomes rather cumbersome. In these cases, a different approach is often used, based on TAYLOR series approximations for the higher derivatives. Cf. [→AMELING 1964, pp. 270 ff.] and [→GILOI et al. 1963] for more information regarding this approach.

7.4.2 Separation of variables

Another method for the solution of partial differential equations with an analog computer is based on the separation of variables.[21] While the method described above discretizes all variables except *t*, the separation of variables allows all variables of a PDE to take on continuous values. As a result, computer setups based on the separation of variables often require substantially fewer computing elements. Nevertheless, this technique has a major drawback: It is much more complicated to derive a computer setup by using the separation approach than by the method of difference quotients.

As an example, the one-dimensional heat-transfer based on

$$\frac{\partial^2 T}{\partial x^2} = k\dot{T} \tag{7.11}$$

will be treated again using this method.[22] The task is the determination of the temperature *T* (*x, t*) at a certain position *x* and a certain time *t*. The separation of the variables *t* and *x* yields

$$T(x,t) = f_1(x) f_2(t). \tag{7.12}$$

From equations (→7.11) and (→7.12)

$$\frac{\mathrm{d}^2 f_1(x)}{\mathrm{d}x^2} f_2(t) = k\dot{f}_2(t) f_1(x)$$

follows. Separating *x* and *t* yields

$$\frac{\frac{\mathrm{d}^2 f_1(x)}{\mathrm{d}x^2}}{f_1(x)} = k\frac{\dot{f}_2(t)}{f_2(t)}.$$

Splitting this equation and setting both sides equal to −λ_n yields the eigenvalue problem

$$\frac{\frac{\mathrm{d}^2 f_1(x)}{\mathrm{d}x^2}}{f_1(x)} = -\lambda_n \text{and} \tag{7.13}$$

$$k\frac{\dot{f}_2(t)}{f_2(t)} = -\lambda_n. \tag{7.14}$$

Solving (→7.13) and (→7.14) for their highest derivatives results in

$$\frac{\mathrm{d}^2 f_1(x)}{\mathrm{d}x^2} = -\lambda_n f_1(x) \text{and} \tag{7.15}$$

$$\dot{f}_2(t) = -\frac{\lambda_n}{k} f_2(t), \tag{7.16}$$

which are readily transformed into the circuits shown in →figure 7.10 and →7.11 where the variable *x* is identified with *t*.

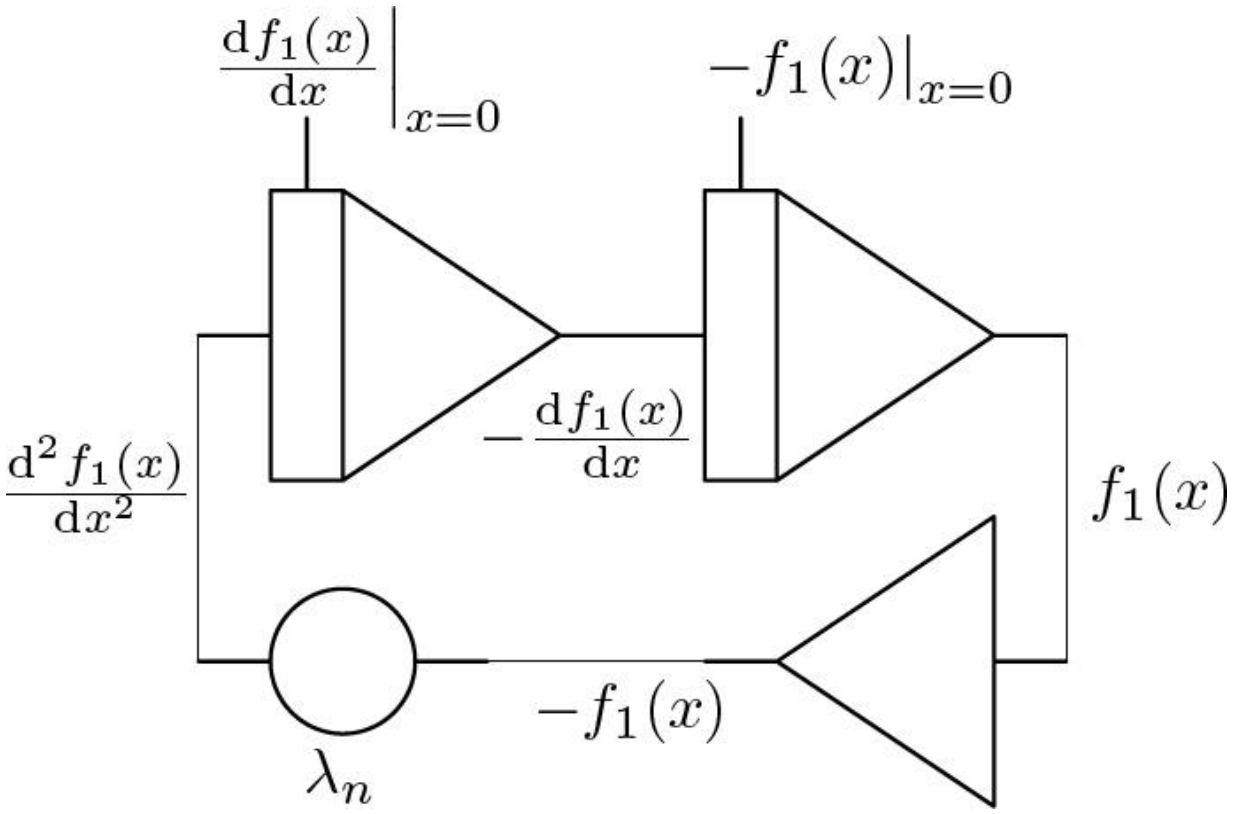


Figure 7.10. Circuit corresponding to (→7.15)

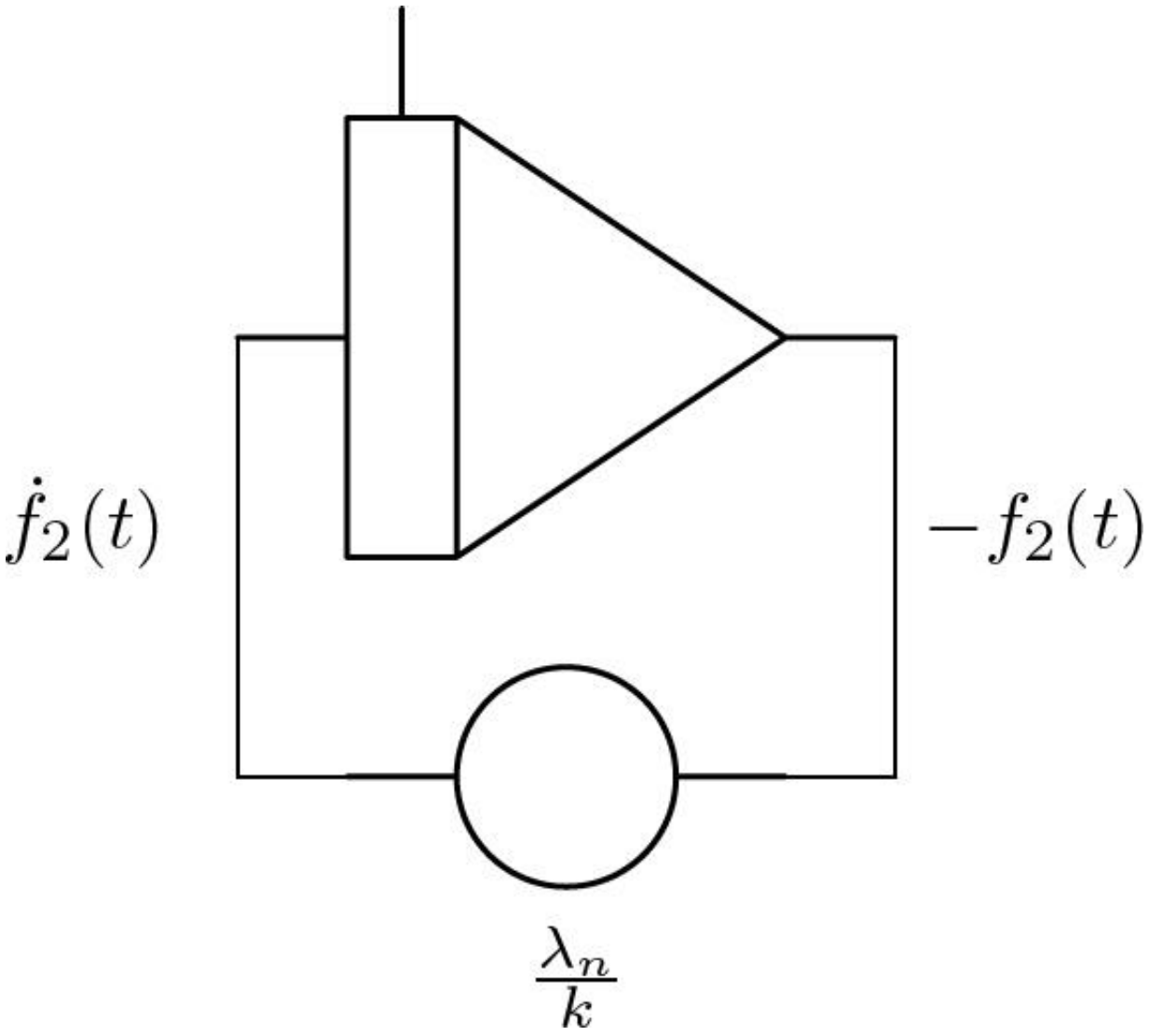


Figure 7.11. Circuit corresponding to (→7.16)

Initially, for $t = 0$, the rod is at temperature $T(0) \neq 0$. Then its ends located at $x = 0$ and $x = 1$ are forced to a temperature 0 so that $T(0, t) = T(1, t) = 0$. Using the circuit shown in →figure 7.11 suitable values λ_n are determined. This is normally done using a manual trial-and-error process by varying the coefficient potentiometer between successive runs of the analog computer with $0 < t \leq 1$.[23] This can be, of course, done automatically by an attached digital computer – one of the advantages of a hybrid computer setup.

With all λ_n determined, a linear combination of the eigenfunctions is sought that approximates $T(x, t)$ to the required degree of precision:[24]

$$T(x,t) = \sum_{i \in x} \lambda_i T_i \tag{7.17}$$

→ Figure 7.12 shows the mechanization of this equation. Using this setup it is either possible to hold t at a fixed value while x is running from 0 to 1 or vice versa. These two variables are controlled with the two switches labeled t_{start} and x_{start}.

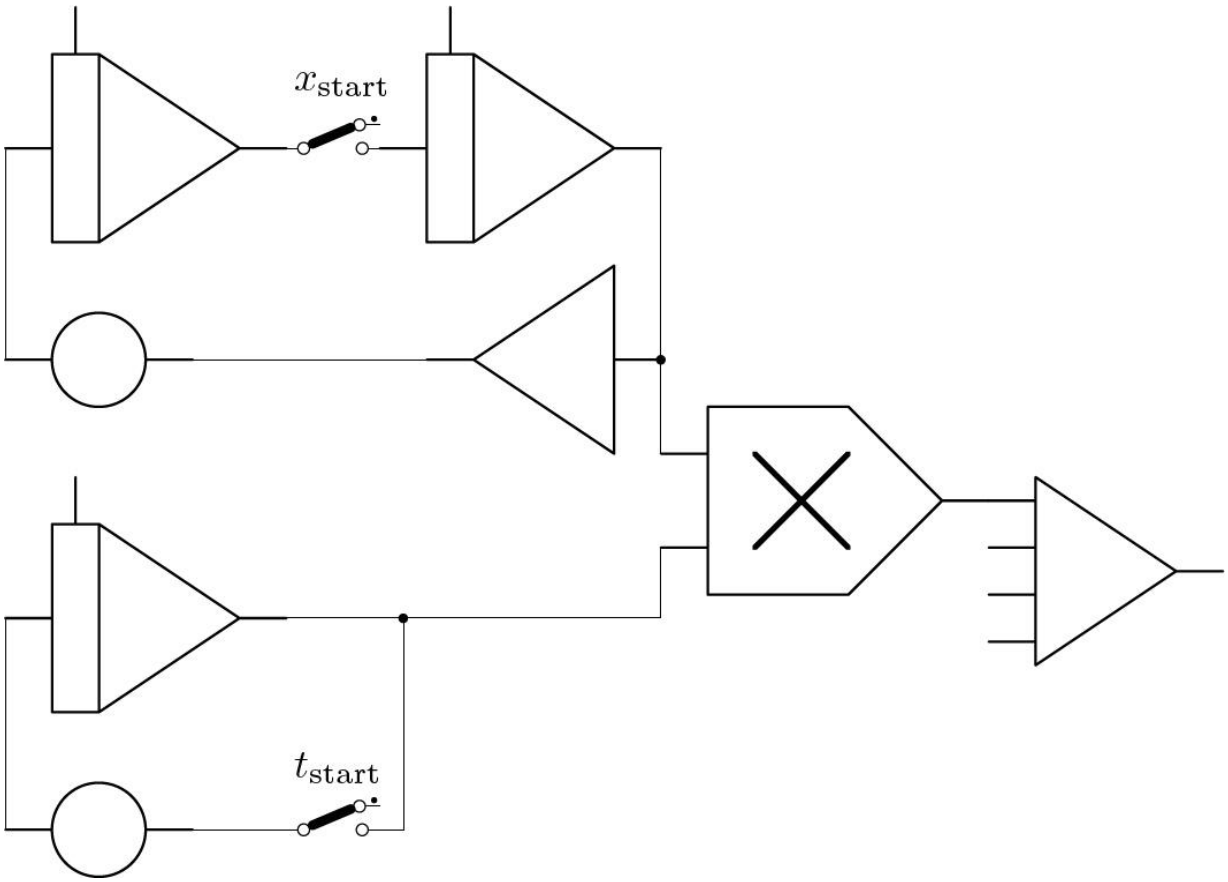


Figure 7.12. One-dimensional heat-transfer using separation of variables

7.5 Scaling

Deriving a circuit corresponding to a given set of differential equations is only one step towards the solution of a given problem using an analog electronic analog computer. Since all variables are bounded by ±1 machine unit, in most cases it is necessary to *scale* the variables in a computer setup. An unscaled variable v representing some property of the underlying problem is associated with a *machine variable* $\mathfrak{v}$, which must satisfy

$$-1 \leq \mathfrak{v} \leq 1 \tag{7.18}$$

at all times.

To achieve this, a set of scaling factors α_v has to be determined so that all $\mathfrak{v} = \alpha_v v$ satisfy (→ 7.18). Determining these α_v can be a time-consuming and difficult process since the overall behavior of the underlying differential equations has to be considered.[25] When $\max(|v|)$ is known with regard to a unit U then α_v is determined by[26]

$$\alpha_v = \frac{1\text{machine unit}}{\max(|v|)} \left[\frac{\text{V}}{\text{V}}\right].$$

Determining these α_v is further complicated by the technical requirement of using as much of the allowable interval [−1, 1] as possible in order to minimize errors due to inevitable inaccuracies of the actual computing elements.[27] Scaling a problem variable v with a scaling factor α_v yields a machine variable $\mathfrak{v}$, which is measured in Volts (or Amperes in case of a current-coupled computer). Machine variables are often denoted by square brackets in the literature. Considering a problem variable $\ddot{x}$ representing an acceleration, which can assume a

maximum value $\max(|\ddot{x}|) = 2 \cdot 9.81\text{m/s}^2$ and a machine unit of 10 V, results in a practical scaling factor $\alpha_{\ddot{x}} = 1/2$. The resulting machine variable $\alpha_{\ddot{x}}\ddot{x}$ is then denoted by $[\frac{\ddot{x}}{2}]$.

Time can be scaled, too. *Problem time*, sometimes also called *real-time*, is normally denoted by t while machine time is often represented as a dimensionless quantity τ. The time-scaling factor is denoted by λ, yielding $\tau = \lambda t$. Accordingly, the dimension of λ (as well as k_0 in the following) is s^{-1}. Since the integrators are the only computing elements depending on time, time-scaling is either done by changing the time-scale factors of the integrators[28] or by changing the input factors of the integrators. Often both scaling approaches are used together.

Integrators typically feature a number of time-scale factors k_0, which can take values of the form $k_0 = 10^n$ with $0 \leq n \leq N$. N can be as large as 6 in some high-speed analog computers. A time-scale factor $k_0 = 1$ means that the integral over a constant input value −1 will yield +1 at the integrator's output after one second. With $k_0 = 10^3$ this output value will be reached in one millisecond.

Any coefficient potentiometers at the inputs of an integrator will influence the overall time-scale factor based on k_0 of the respective integrator. → Figure 7.13 shows this effect with a coefficient α.

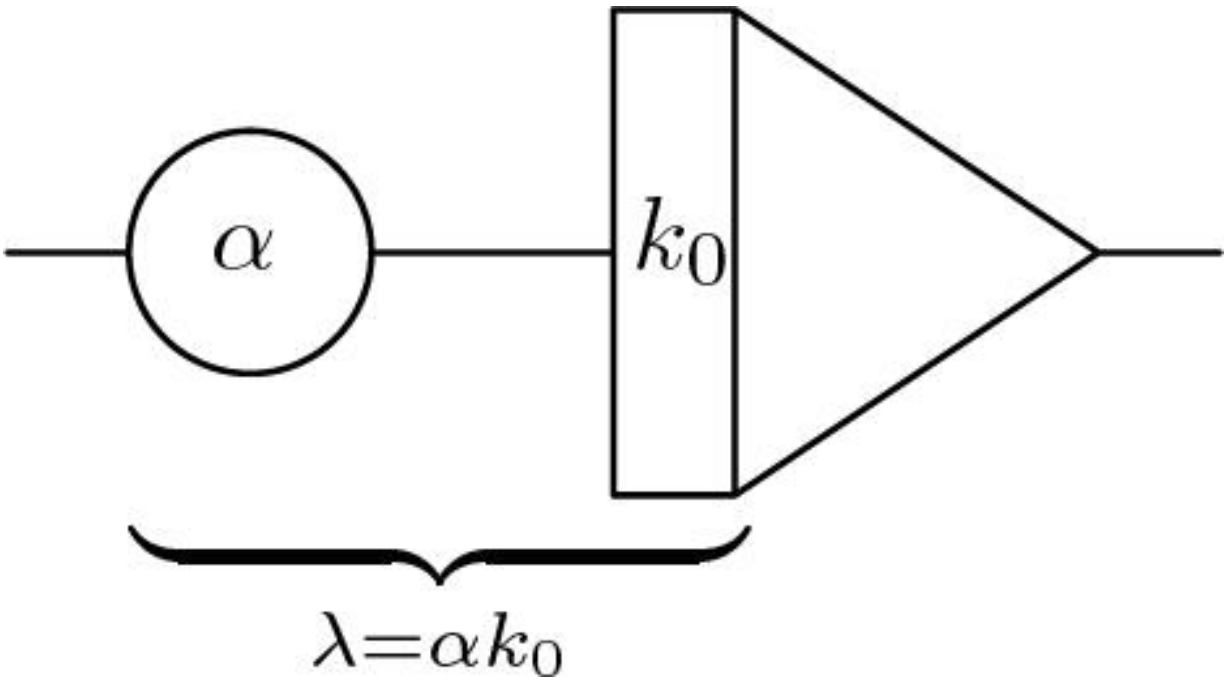


Figure 7.13. Time-scaling

By changing λ it is possible to run computations slower or faster than real-time. Large analog computers often feature a switch labeled “10×” that exchanges the feedback capacitors of all integrators by components with one-tenth of the normal capacity at the selected value k_0 thus speeding up a computer run by a factor of 10 at the press of single button.[29]

Many methods were devised to facilitate the scaling process, including numerical solutions of a problem by means of a stored-program digital computer to determine the maximum values of the problem variables involved.[30] Other approaches used hybrid computers[31] where the digital part generates scaling values for the analog part.[32]

8 Programming examples

Learning how to program an analog computer requires practice but is not too difficult. The following sections contain some programming examples, ranging from simple to difficult, all of which were implemented and run on real analog computers.

8.1 Solving $\ddot{y} + \omega^2 y = 0$

Generating a sine-/cosine signal pair, also called a *quadrature signal pair* on an analog computer, is a frequent requirement, since such signals can be used for various tasks such as filter design, research on vibrating mechanical systems, displaying figures on an oscilloscope, etc. On a stored-program digital computer polynomial approximations or even table lookups could be used to generate a function such as sin(ωt). On an analog computer a different approach is typically chosen: Solving a differential equation which yields the desired function as result:

$$\ddot{y} + \omega^2 y = 0 \tag{8.1}$$

with initial conditions

$$\begin{aligned} y(0) &= a\ \sin(\varphi) \quad \text{and} \\ \dot{y}(0) &= a\omega\ \cos(\varphi). \end{aligned}$$

Prior to applying Kelvin's feedback method, equation (→8.1) is solved for its highest derivative yielding

$$\ddot{y} = -\omega^2 y. \tag{8.2}$$

Since this differential equation is satisfied by any linear combinations of sine and cosine, it can be readily used to generate a sine signal, given suitable initial values. In this case, additional scaling of the problem variables is not necessary since max(| sin(ωt)|) = 1. →Figure 8.1 shows

the basic circuit based on equation (→8.2). This setup is still missing its initial values and would yield the null-function.[1]

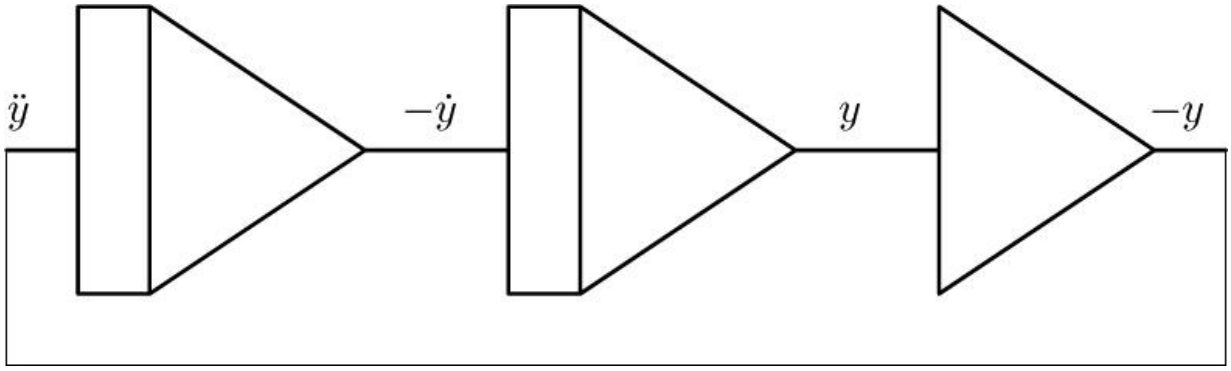


Figure 8.1. Basic circuit for $y = \sin(t)$

The frequency ω of the resulting sine signal influences the time-scaling process. →Figure 8.2 shows the time-scaled circuit. Using two coefficient potentiometers in conjunction with the time-constant k_0 of the associated integrators, $\omega = \alpha k_0$ can be set.[2] Thus, time is identified with τ in this setup. To generate a sine signal the second integrator is fed with an initial value of 0.

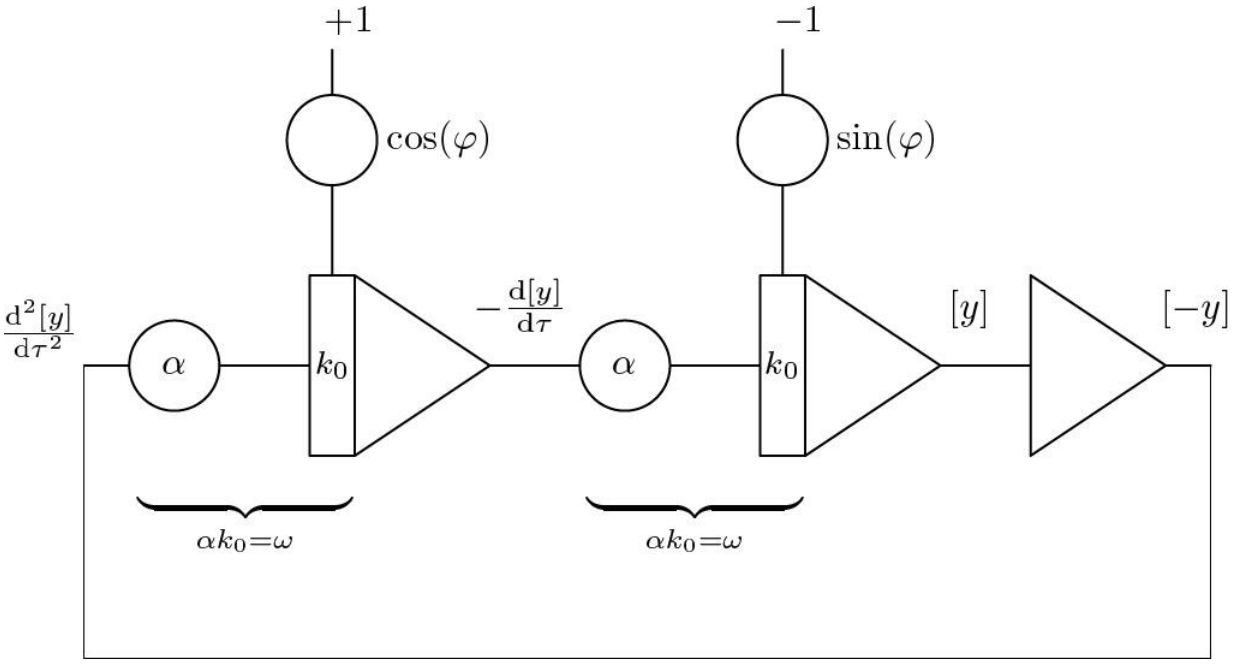


Figure 8.2. Basic circuit for $y = \sin(\omega t + \varphi)$

→ Figure 8.3 shows one period of the output signal [y] generated by this circuit. Using the time control system of the analog computer used, the integrators were placed into *operate* mode for the time it takes to generate one full wave. It should be noted that the output amplitude of a setup like this will not stay constant but instead decrease or increase over time. This is mainly due to little differences in the time-scaling factors k_0 of the integrators. If a sine/cosine signal with a stable amplitude is required over an extended period of time, additional measures have to be taken.[3]

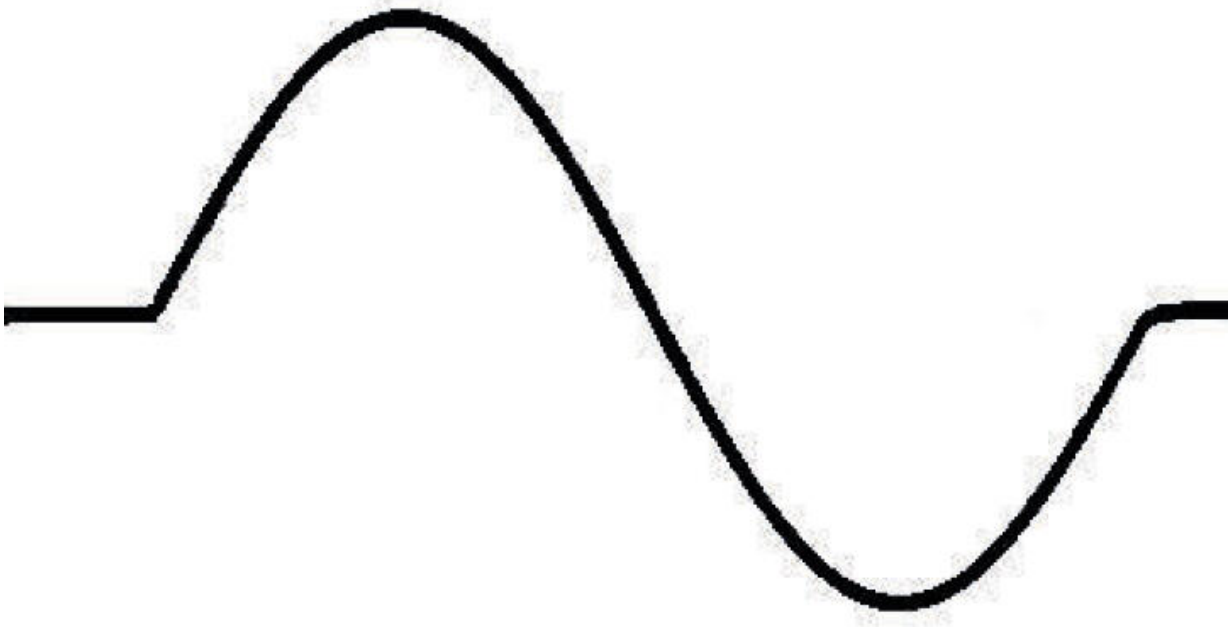


Figure 8.3. A single sine period generated by the circuit shown in →figure 8.2

8.2 Sweep generator

Since $\omega = \alpha k_0$ controls the output-signal's frequency, the circuit shown in →figure 8.2 can be easily extended to a sweep generator generating a sinusoidal signal with increasing frequency. Swept signals are frequently employed to analyze transmission systems, the response of vibrating systems, etc. Here, the two coefficient potentiometers in front of the two integrators are replaced by multipliers sharing a common factor.[4]

→Figure 8.4 shows the basic setup of such a sweep generator.[5] The frequency controlling signal is derived from integrating over a constant value α. To achieve a sufficient slow frequency change, two time-scale factors for the integrators involved are used. The integrator with α as its input has a time-scale $k_0 = 1$, while the two integrators forming the basic oscillator loop run with a time-scale ten times (or more) as fast ($k_0 = 10$). The patch panel setup for this sweep generator is shown in →figure 8.5. →Figure 8.6 shows the output of a typical sweep run with linearly increasing frequency.

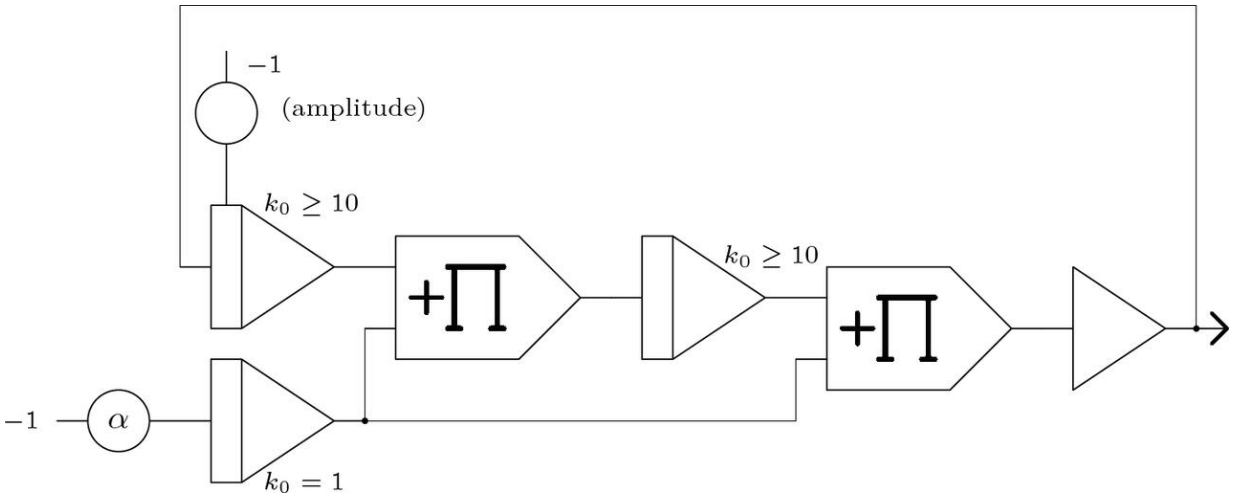


Figure 8.4. Basic circuit for a sweep generator

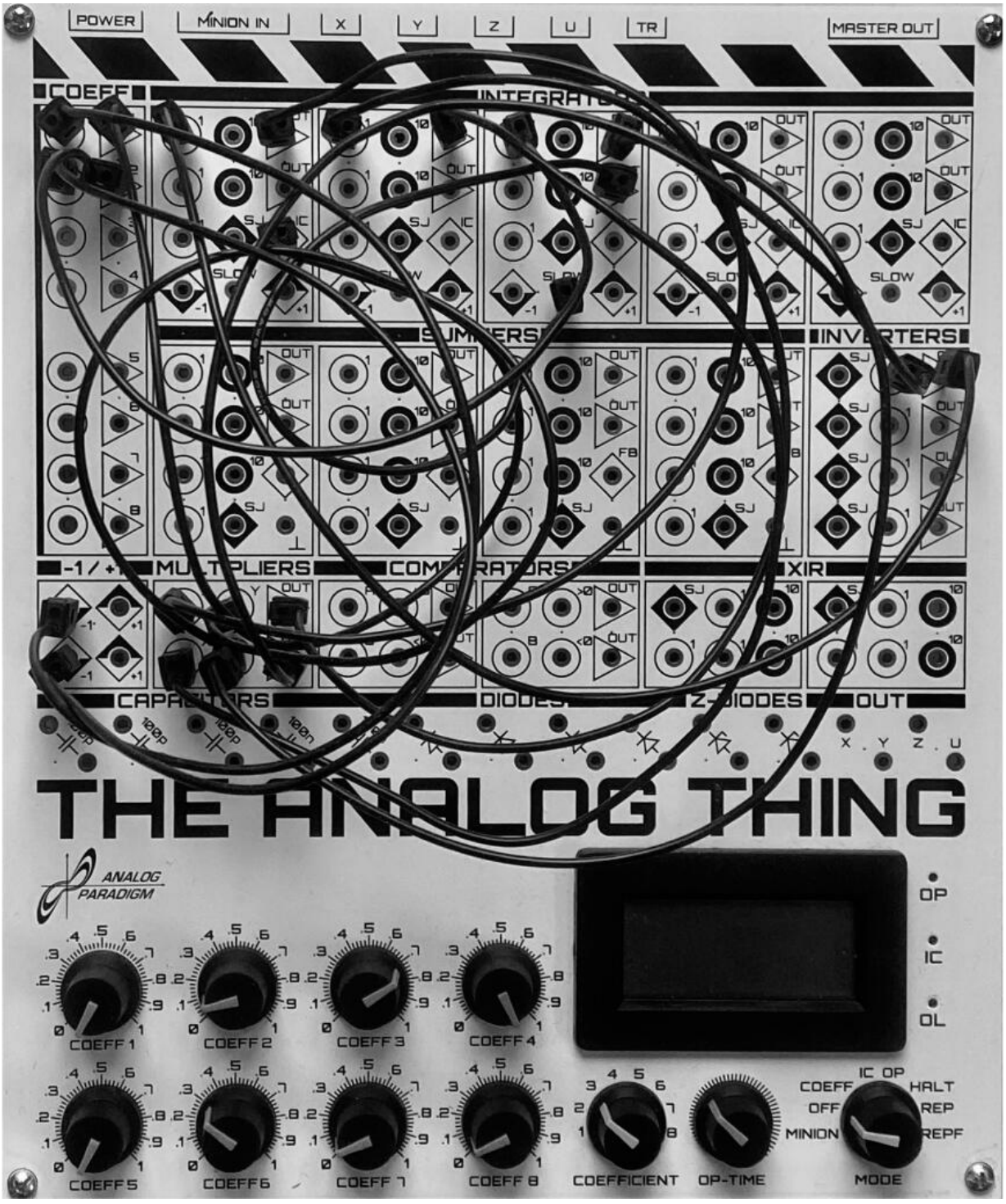


Figure 8.5. Patch panel setup for the sweep generator (THE ANALOG THING)

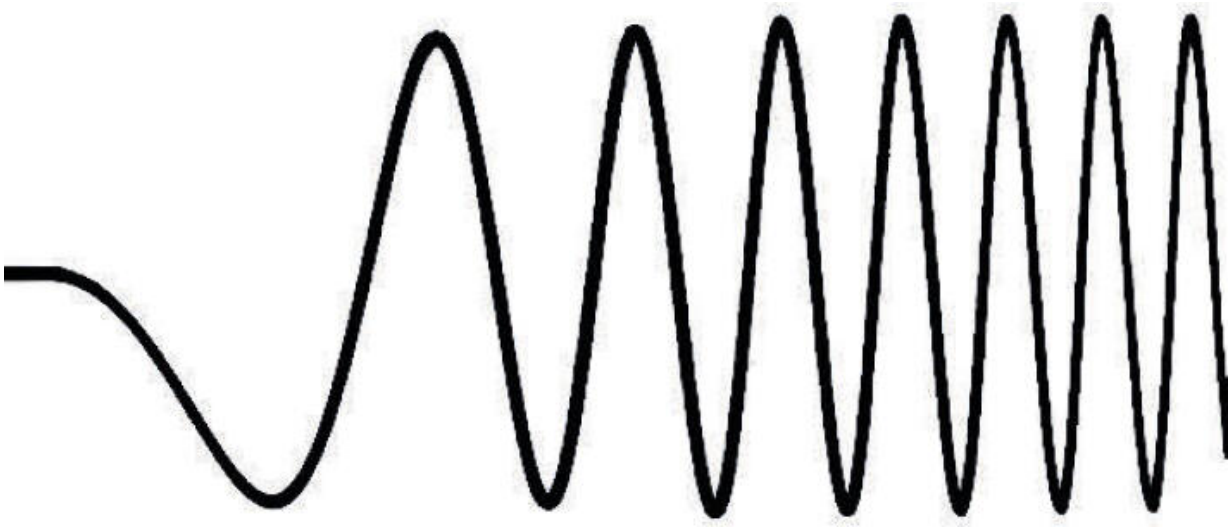
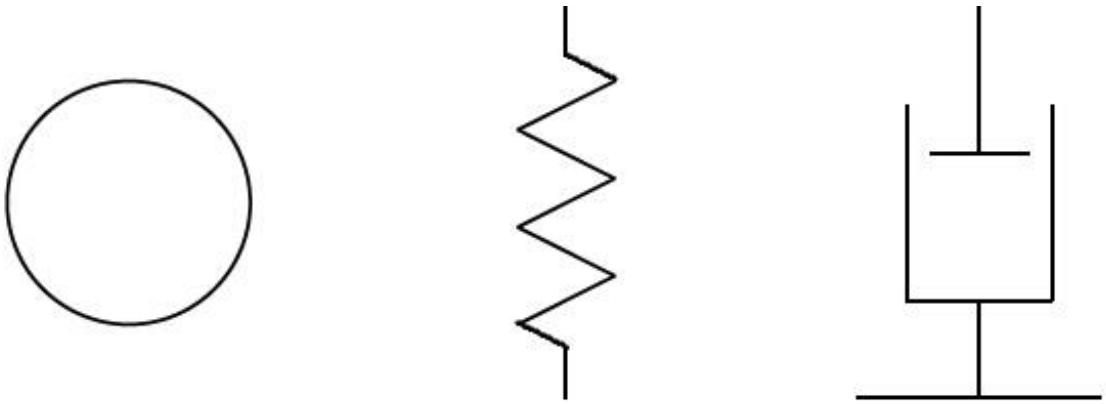


Figure 8.6. Sine oscillation with variable frequency

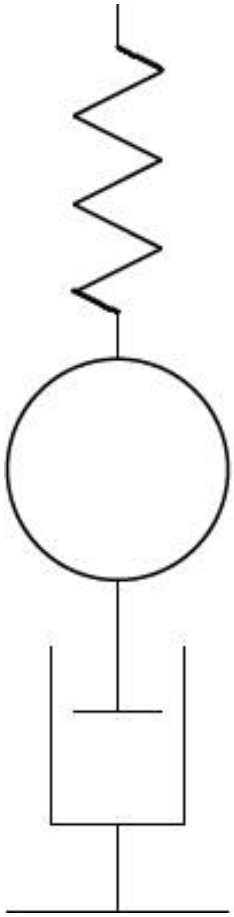
8.3 Mass-spring-damper system

The next problem is a bit more complex. A mass-spring-damper system consisting of the parts shown in →figure 8.7 is to be modeled and simulated with an analog computer. These parts are a mass *m* exerting a force *mÿ*, a spring with stiffness *s* resulting in a force *sy* and a damper with damping coefficient *d* exerting a force *dẏ*. Since the forces in a closed physical system add up to 0, the mass-spring-damper system shown in →figure 8.8 can be described by the following differential equation:



$$F_m = ma = m\ddot{y} \quad F_s = sy \qquad F_d = d\dot{y}$$

Figure 8.7. Parts of a mass-spring-damper system



$$F_m + F_d + F_s = 0$$

$$m\ddot{y} + d\dot{y} + sy = 0$$

Figure 8.8. Mass-spring-damper system

$$m\ddot{y} + d\dot{y} + sy = 0. \tag{8.3}$$

Rearranging so that the highest derivative is on the left side of the equal sign yields

$$\ddot{y} = \frac{-(d\dot{y}+sy)}{m} \tag{8.4}$$

as a starting point for applying the KELVIN method. → Figure 8.9 shows the first part of the program: Using an integrator, $\ddot{y}$ is integrated, yielding $-\dot{y}$ with an initial value of $\dot{y}(0)$.

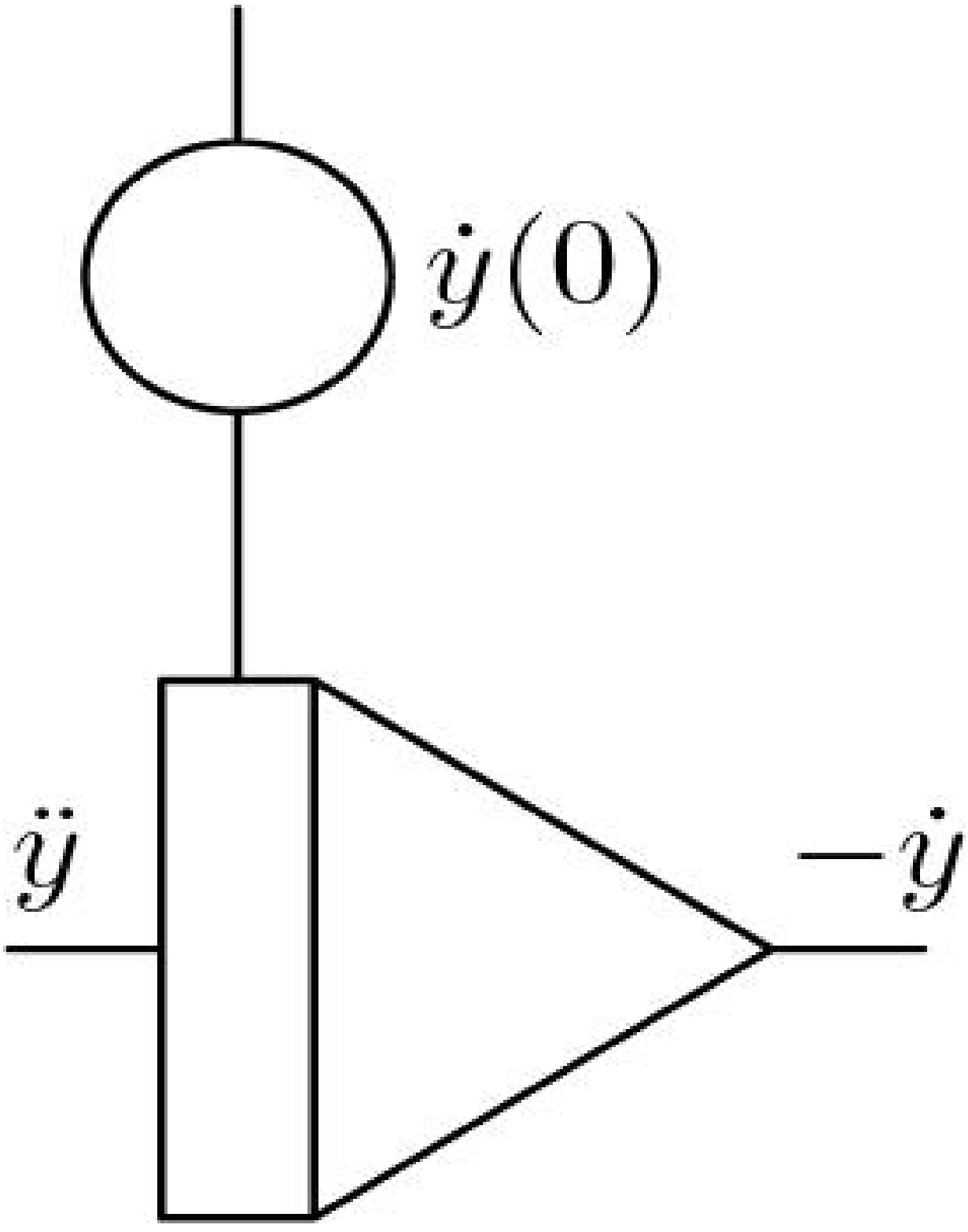


Figure 8.9. First integration step yielding $-\dot{y}$

Using a second integrator with initial value $-y(0)$ yields y, which can be multiplied by s using a coefficient potentiometer as shown in →figure 8.10. Adding an inverter and a second coefficient potentiometer to derive $d\dot{y}$ and summing this term and sy results in the circuit shown in →figure 8.11 yielding $-(d\dot{y} + sy)$. This can now be multiplied by $1/m$ with a coefficient potentiometer yielding the right side of equation (→8.4). Feeding this signal back into the input of the first integrator closes the feedback loop. The final circuit is shown in →figure 8.12.[6]

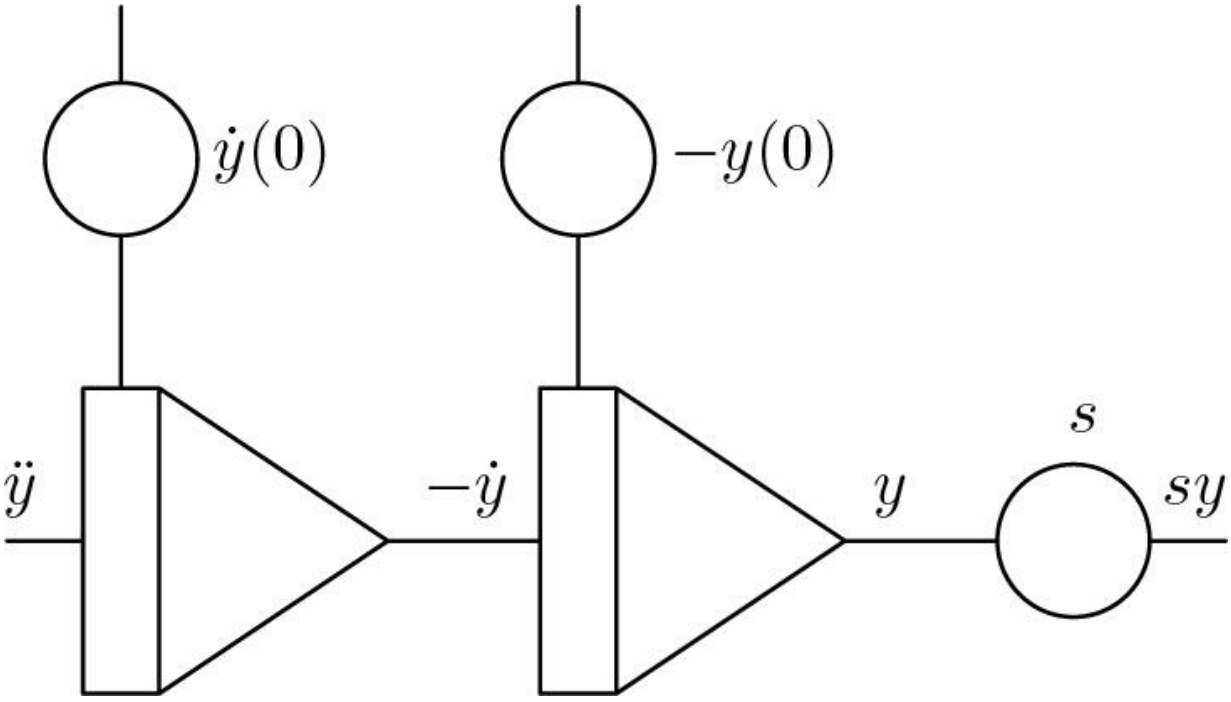


Figure 8.10. Second integration step yielding y

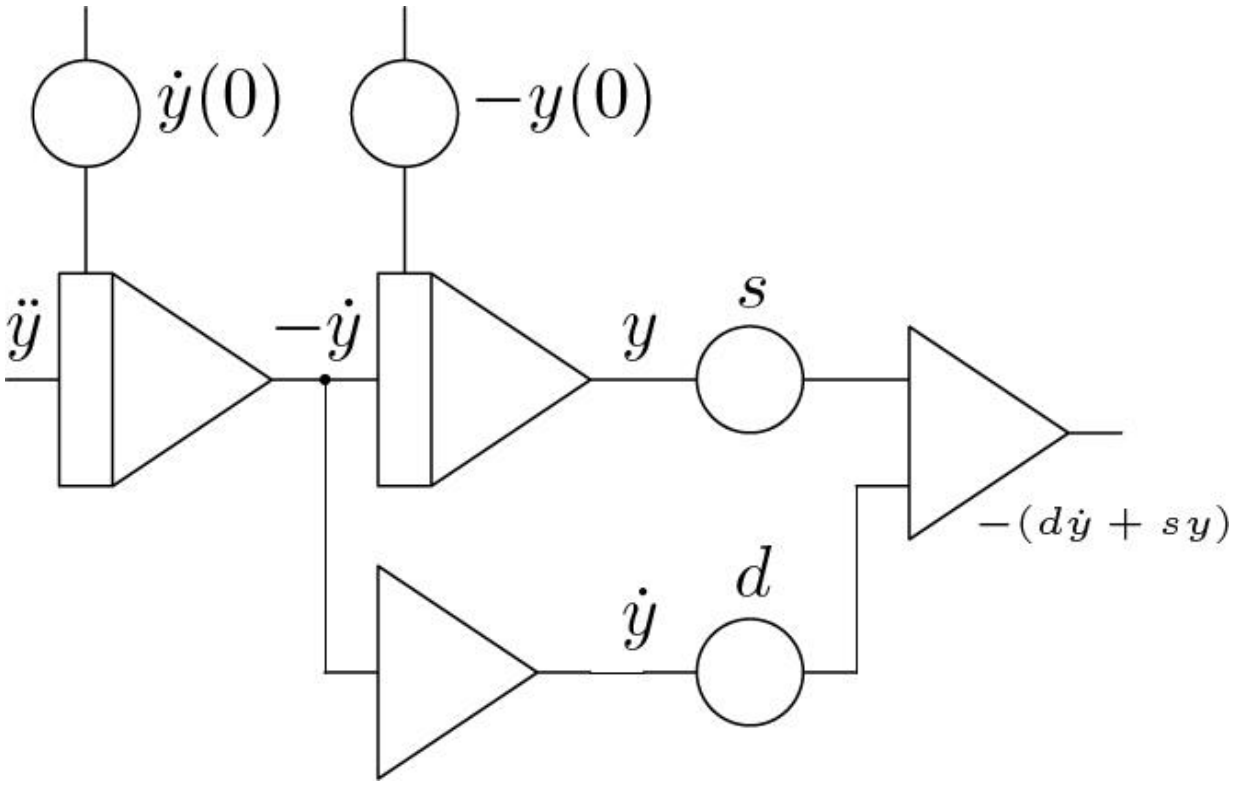


Figure 8.11. Computing –(*dẏ* + *sy*)

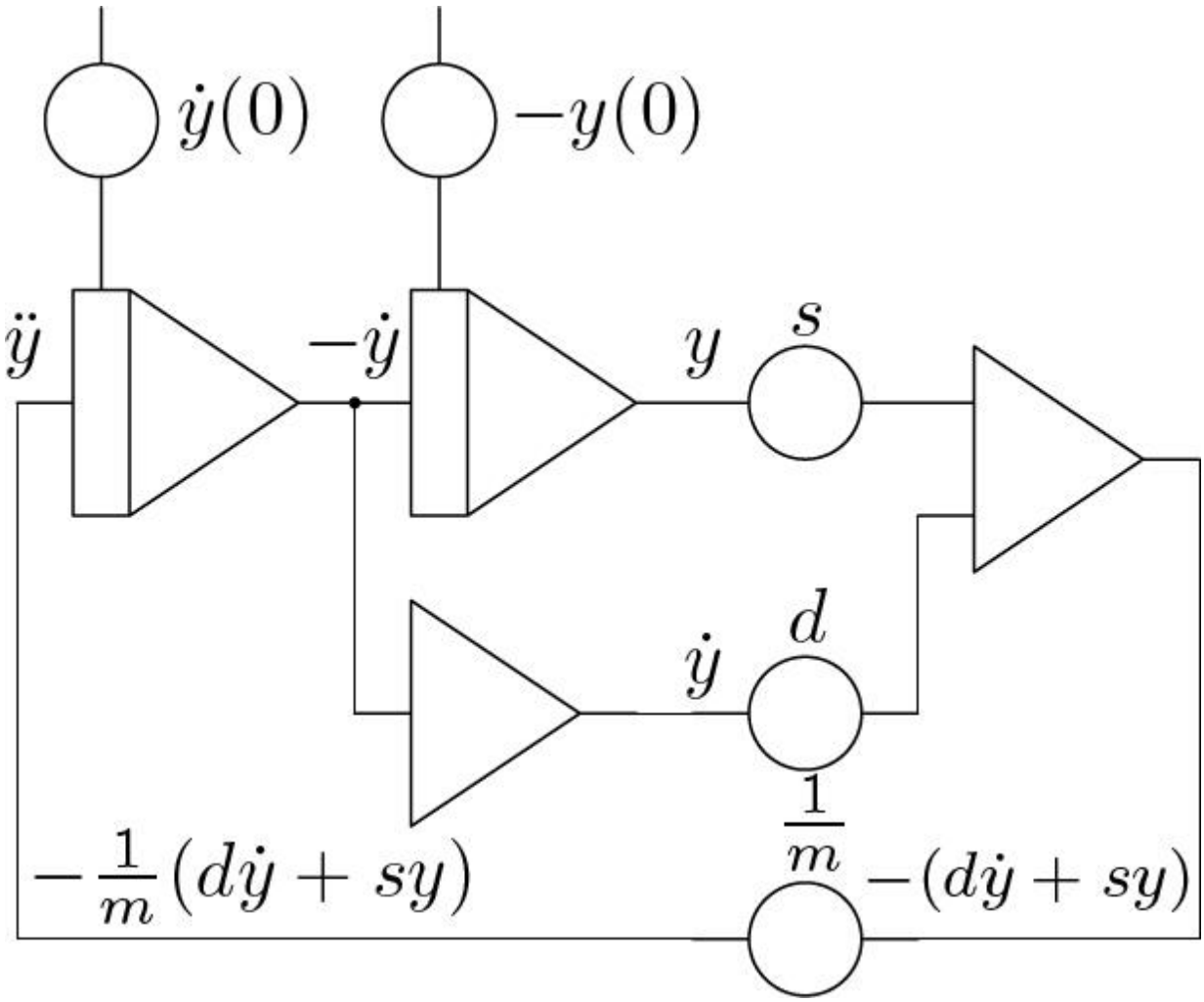


Figure 8.12. Final computer setup for the mass-spring-damper system

The overall setup for this program on a Telefunken RA 741 tabletop analog computer is shown in →figure 8.13. →Figures 8.14, →8.15, →8.16, and →8.17 show typical simulation results for various settings of the spring's stiffness and the damping coefficient with constant mass m = 1.[7] If the computer is running in repetitive mode and if the time-scale factors of the integrators are set to $k_0 = 10^2$ or $k_0 = 10^3$, a flicker-free image can be displayed on an oscilloscope screen. The effect of changing the coefficients *s, d*, and *m* can then be seen directly while turning the potentiometer dials.[8] These runs were performed on a non-scaled model and thus yielded only qualitative results.

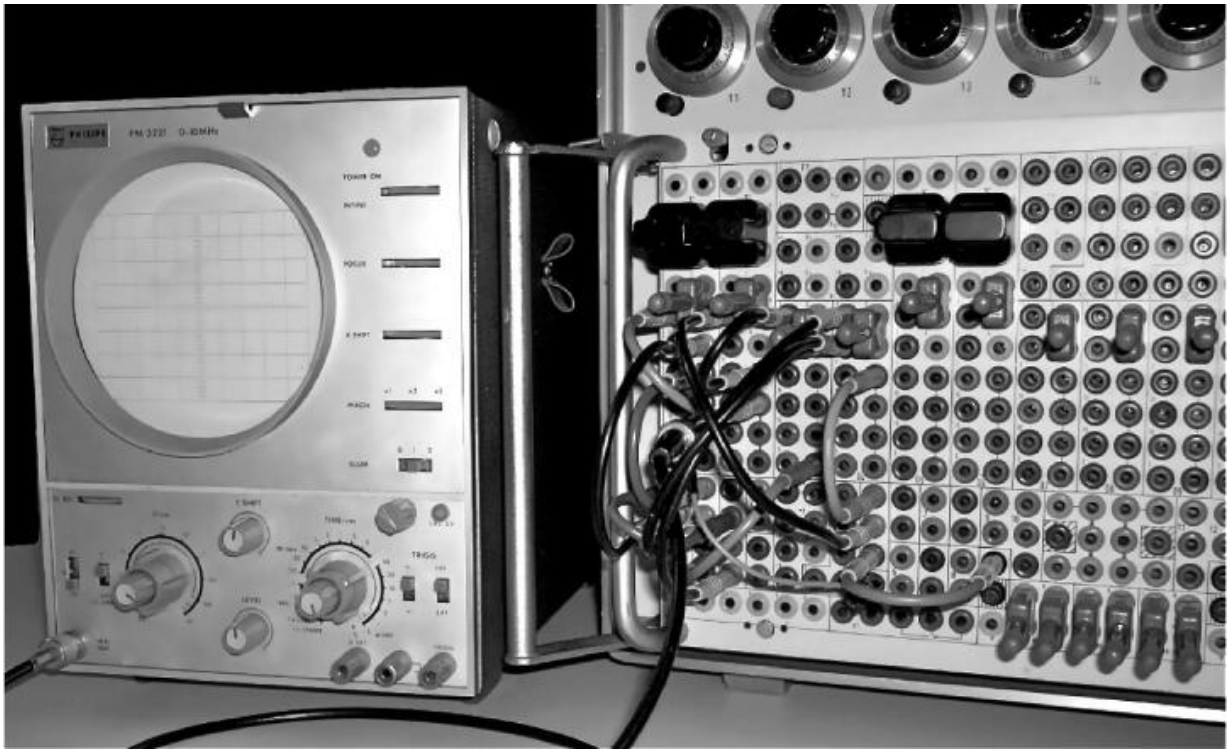


Figure 8.13. Setup of the mass-spring-damper simulation



Figure 8.14. $s = 0.2$, $d = 0.8$



Figure 8.15. $s = 0.6$, $d = 0.8$

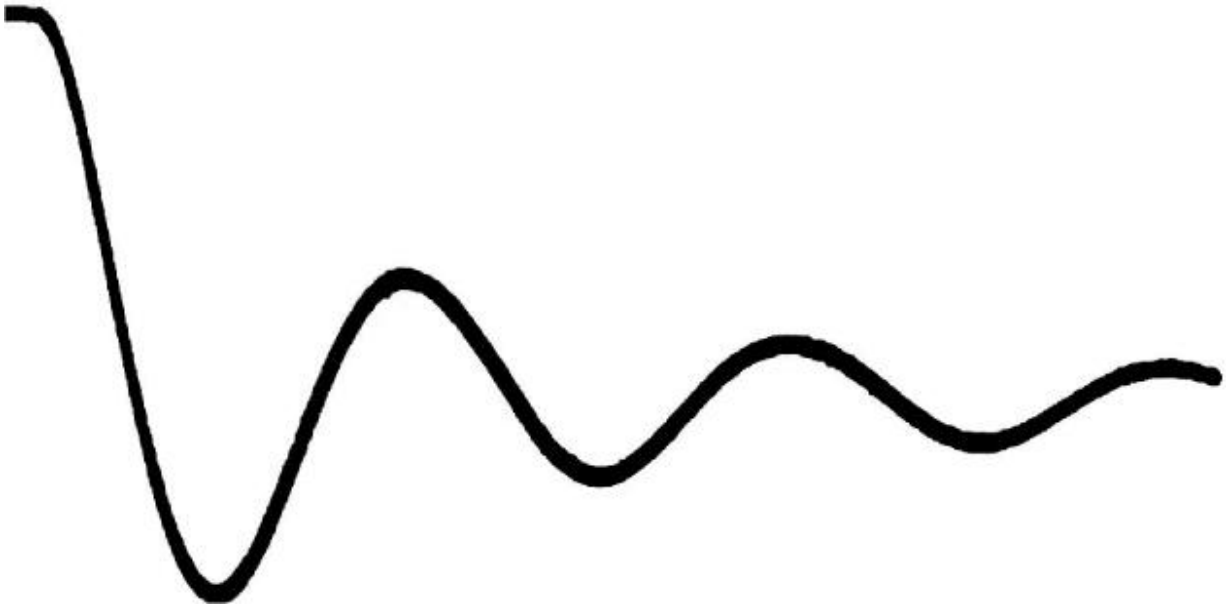


Figure 8.16. $s = 0.8$, $d = 0.6$



Figure 8.17. *s* = 0.8, *d* = 1

To derive quantitative results the following values are assumed: *m* = 1.5 kg, $s = 60\frac{\mathrm{kg}}{\mathrm{s}^2}$, $d = 3\frac{\mathrm{kg}}{\mathrm{s}}$, $\dot{y}(0) = 0\frac{\mathrm{m}}{\mathrm{s}}$, and *y*(0) = 0.1 m (initial displacement of the mass). Scaling the problem requires knowledge about the maximum values for all variables within the computer setup, *y*, *ẏ*, and *ÿ*. Good approximations for these can be derived from an undamped harmonic oscillator with a resonant frequency of

$$\omega = \sqrt{\frac{s}{m}} = \sqrt{\frac{60\frac{\mathrm{kg}}{\mathrm{s}^2}}{1.5\ \mathrm{kg}}} \approx 6.3\ \mathrm{s}^{-1}.$$

From *y* = *y*(0) cos(*ωt*) it follows that

$$\dot{y} = -y(0)\omega\ \sin(\omega t) \quad \text{and} \tag{8.5}$$

$$\ddot{y} = -y(0)\omega^2 \cos(\omega t). \tag{8.6}$$

Since the mass-spring-damper system in this example is a damped oscillator,

$$\max_{t>0} |y| \leq y(0)$$

holds yielding the following boundary values for $\dot{y}$ and $\ddot{y}$ based on equations (→ 8.5) and (→ 8.6):

$$\dot{y} < y(0)\omega \approx 0.63 \frac{\mathrm{m}}{\mathrm{s}}$$
$$\ddot{y} < y(0)\omega^2 \approx 4 \frac{\mathrm{m}}{\mathrm{s}^2}$$

Thus, a reasonable scaling for the machine variables would be $[10y]$, $[15\dot{y}]$, and $[\frac{5}{2}\ddot{y}]$ respectively. Time-scaling is not performed so that the simulation will yield results in real-time. Based on these scaled machine variables equation (→ 8.3) becomes

$$\left[\tfrac{5}{2}\ddot{y}\right] = -\tfrac{5}{2m}\left(\tfrac{d}{15}[15\dot{y}] + \tfrac{s}{10}[10y]\right).$$

To perform a scaled simulation run the coefficient potentiometers for the problem parameters *s, d* and 1/*m* must be set to [*s*/10], [*d*/15], and [5/2*m*].

After scaling a problem a *static check* should be performed, if the analog computer being used features this as a special mode of operation, to make sure that the patch panel setup is correct. The basic idea of such a test is simple: All integrators are switched into a mode where they act as simple summers. Then test signals, which are derived from a static solution of the problem's equations, are fed to special inputs of these integrators so that they appear at the respective outputs with inverted sign. The resulting values in the remaining circuit are then compared against this static solution, which has been prepared manually or with the help of a stored-program digital computer.

8.4 Predator and prey

While the mass-spring-damper system could be described by a single differential equation, the famous problem describing a simplified ecosystem containing predators and prey is based on two coupled differential equations. Systems like these were first studied in 1925 by Alfred James Lotka[9] and independently in 1926 by Vito Volterra.[10]

The resulting differential equations are known as Lotka-Volterra equations. Their interest was sparked by statistical data gathered by the

Hudson Bay Company in the years between 1850 and 1900. The data sets containing the numbers of rabbits and lynxes that were captured per year showed an interesting periodicity.[11]

Consider a closed-world ecosystem with unlimited food for the prey. This ecosystem can be described by the following two coupled differential equations with *l* and *r* representing the number of lynxes and rabbits respectively:

$$\dot{r} = \alpha_1 r - \alpha_2 r l \tag{8.7}$$

$$\dot{i} = -\beta_1 l + \beta_2 r l \tag{8.8}$$

Here, α_1 represents the rate of birth for rabbits, α_2 the decimation of rabbits due to lynxes catching rabbits,[12] β_1 describes the death-rate of lynxes due to natural causes, and β_2 models the increase of the lynx population due to food supply in the form of rabbits.[13]

The two coupled differential equations (→8.7) and (→8.8) can be directly transformed into the two sub-circuits shown in →figures 8.18 and →8.19 by using the KELVIN feedback technique.

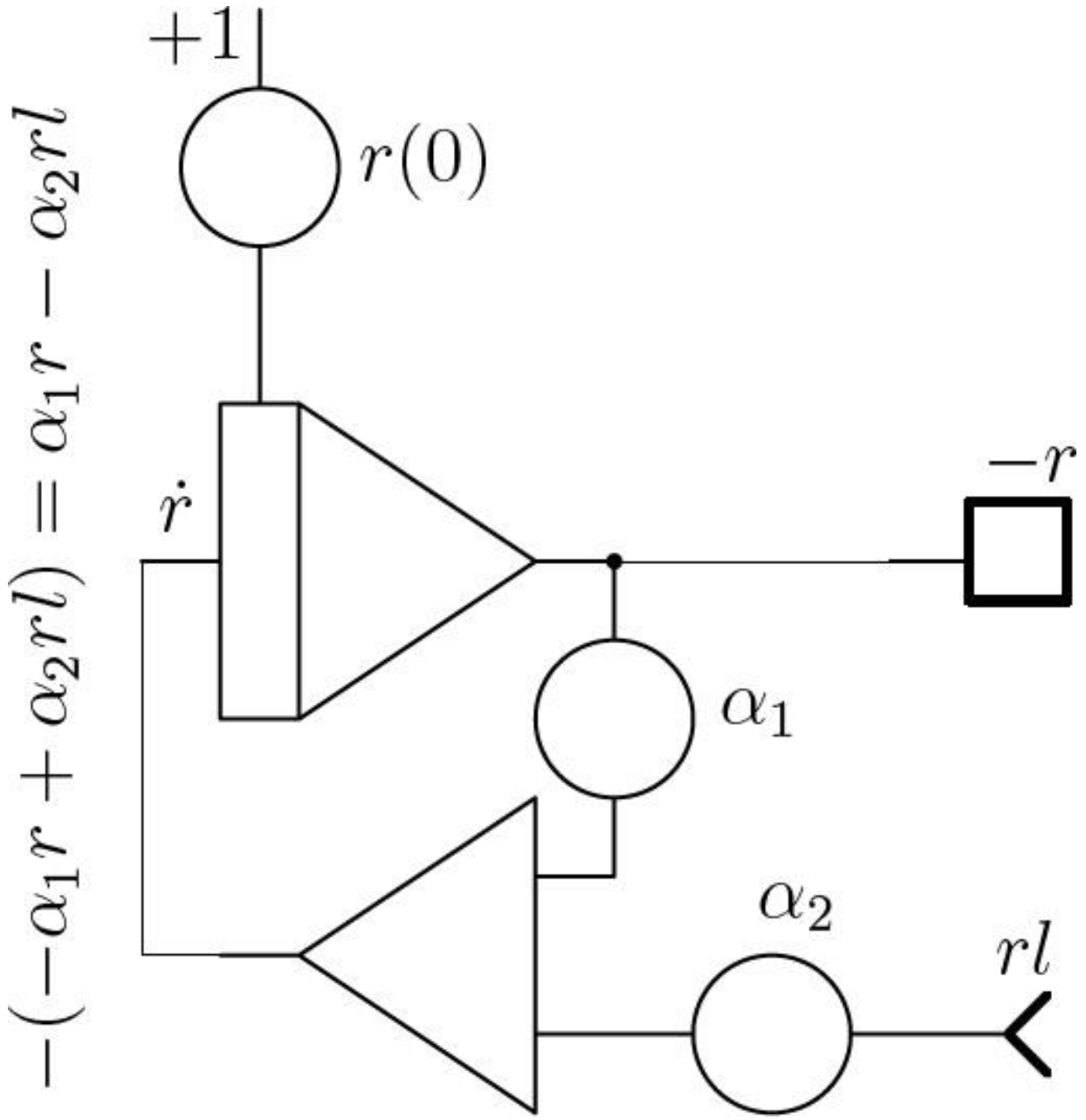


Figure 8.18. Computing –*r*

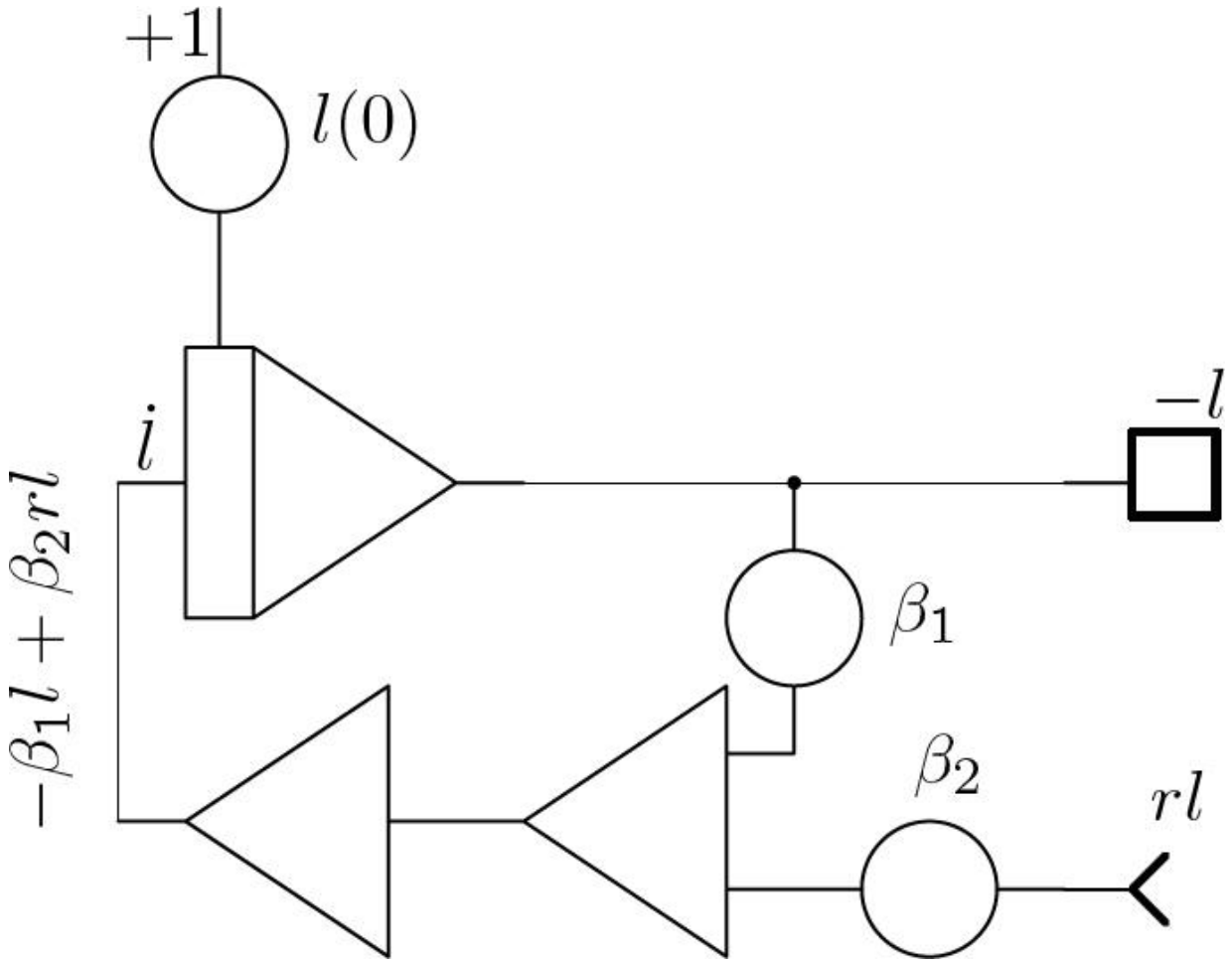


Figure 8.19. Computing –*l*

Without the common input *rl* (lynxes eating rabbits), these circuits would just model an ever faster increasing rabbit population[14] and a decreasing lynx population. Noting that the two summers connected in series in →figure 8.19 just act as a summer without sign-inversion readily yields the simplified circuit shown in →figure 8.20 saving both summers and thus not only simplifying the overall setup but also increasing the precision of the simulation.

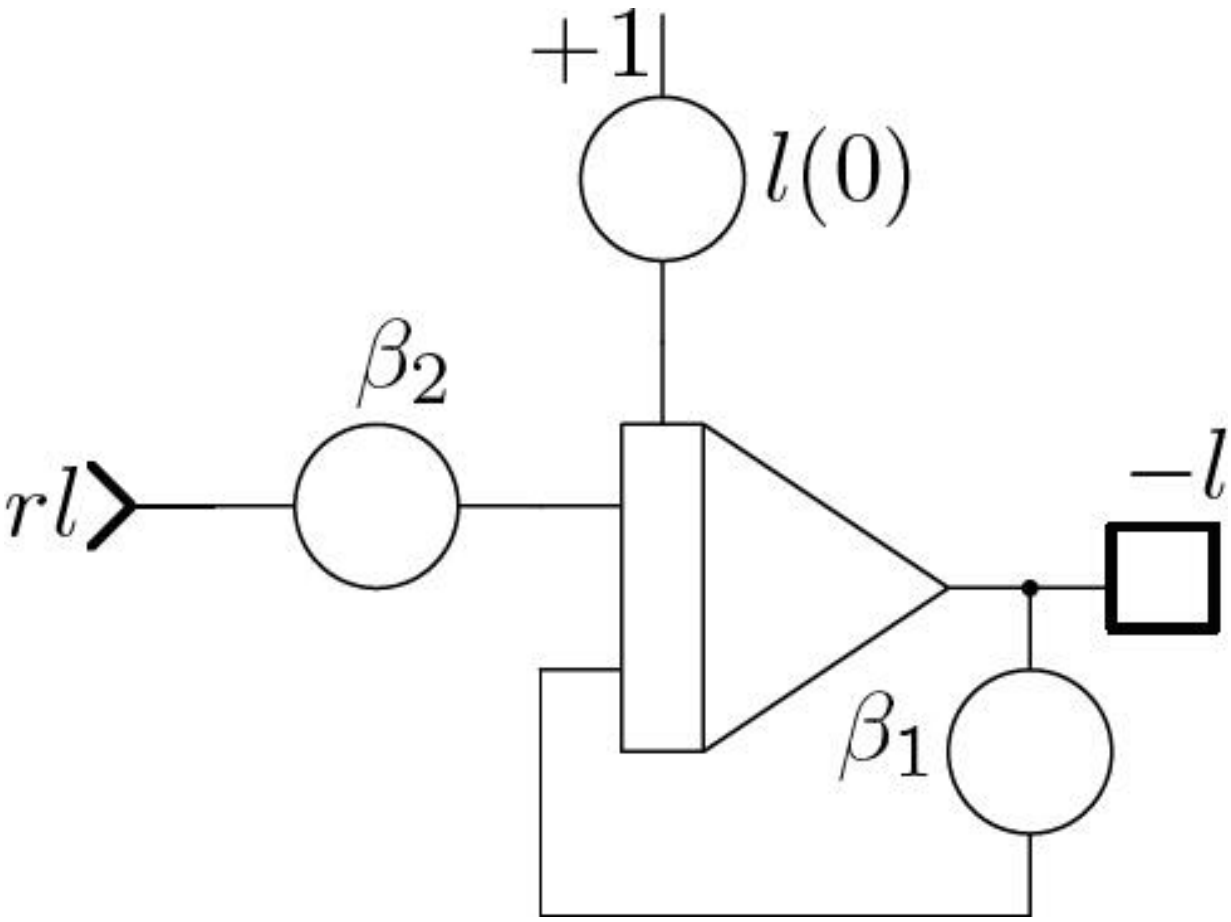


Figure 8.20. Improved circuit for −*l*

→ Figure 8.21 combines both sub-circuits by means of a multiplier yielding the shared input *rl*.

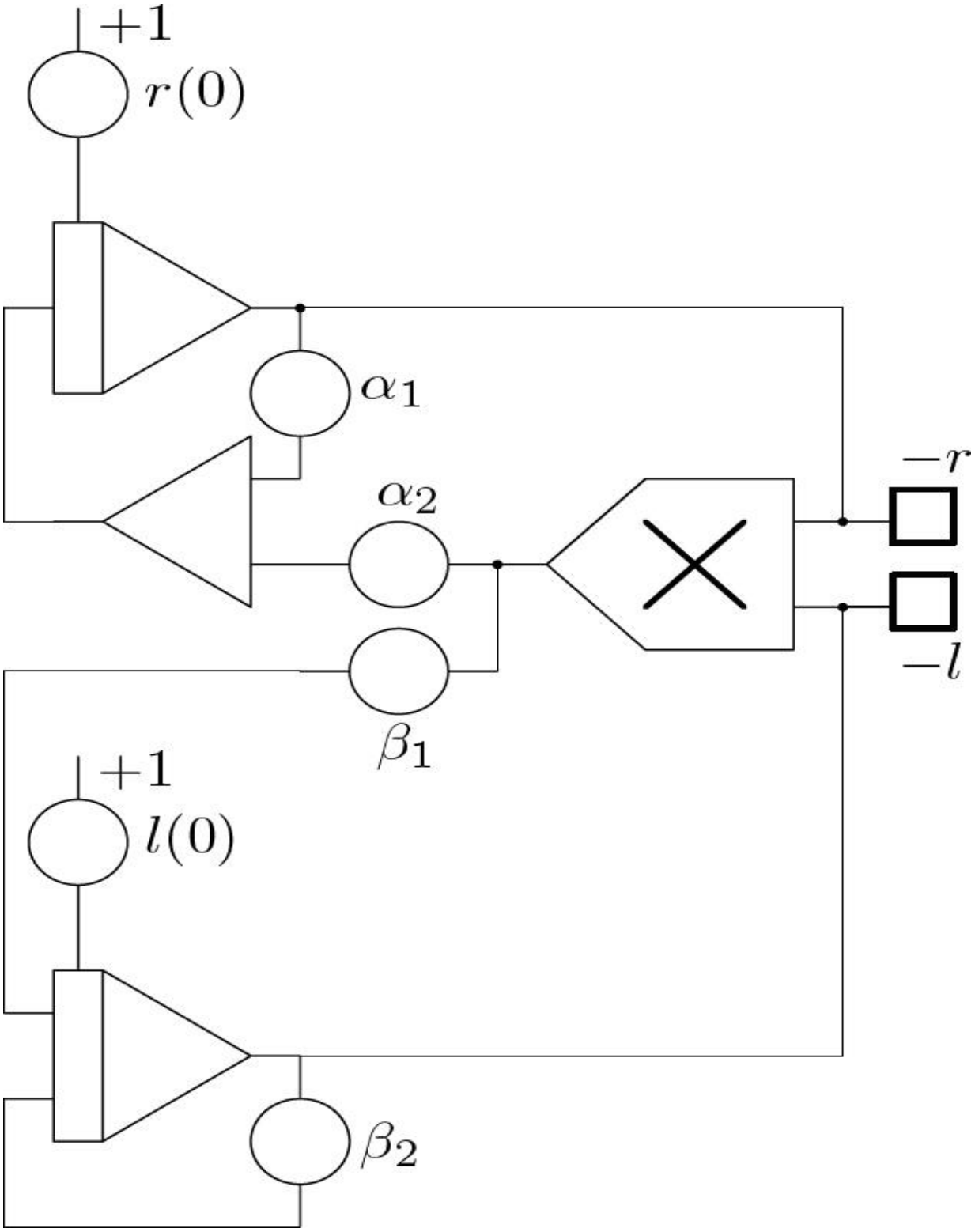


Figure 8.21. Completed predator-prey setup

The overall setup of the predator-prey simulation, consisting of a Nicolet 4094C digital storage oscilloscope on the left and a Telefunken RA 741 tabletop analog computer, is shown in →figure 8.22. Typical results showing an oscillatory behavior of the ecosystem are depicted in →figure 8.23. The all-important phase lag between rabbit and lynx populations is clearly visible.[15]

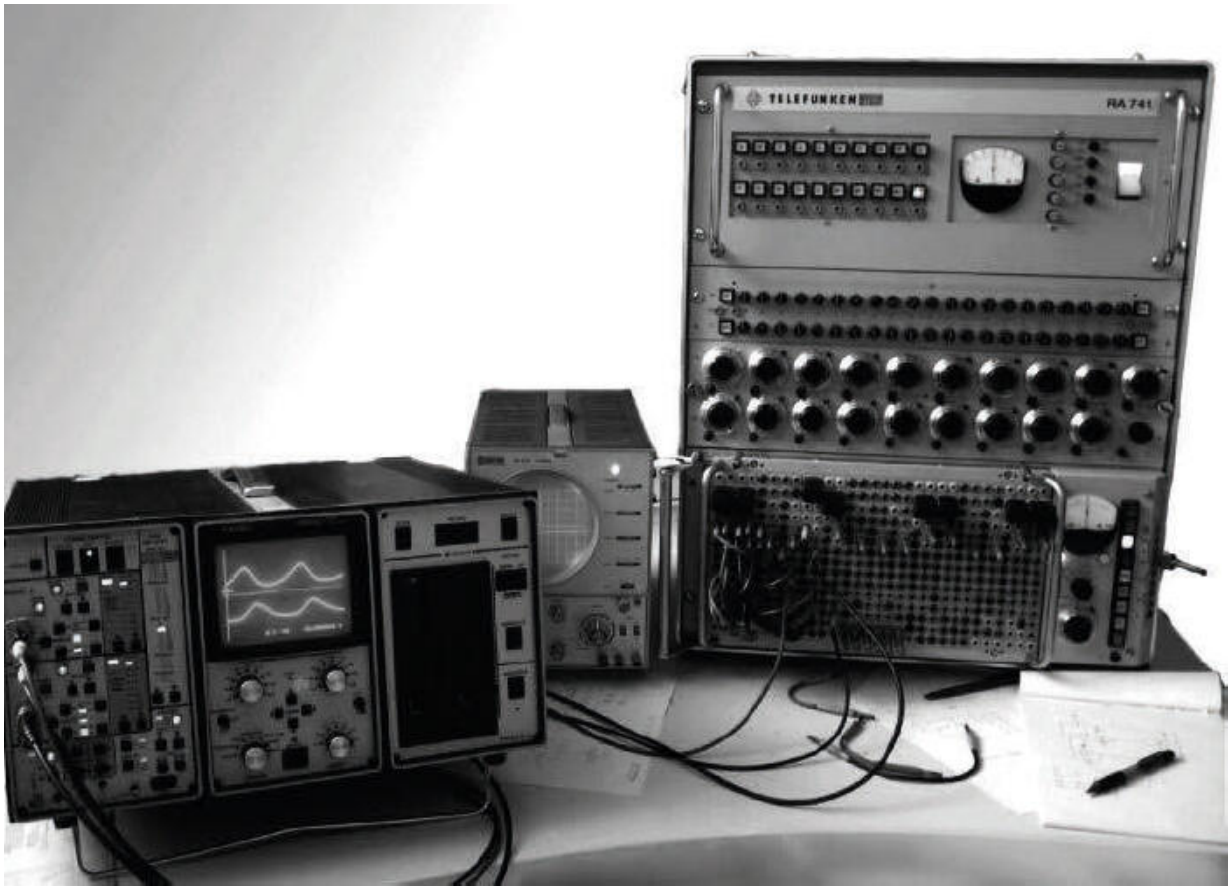


Figure 8.22. Setup for the predator-prey simulation

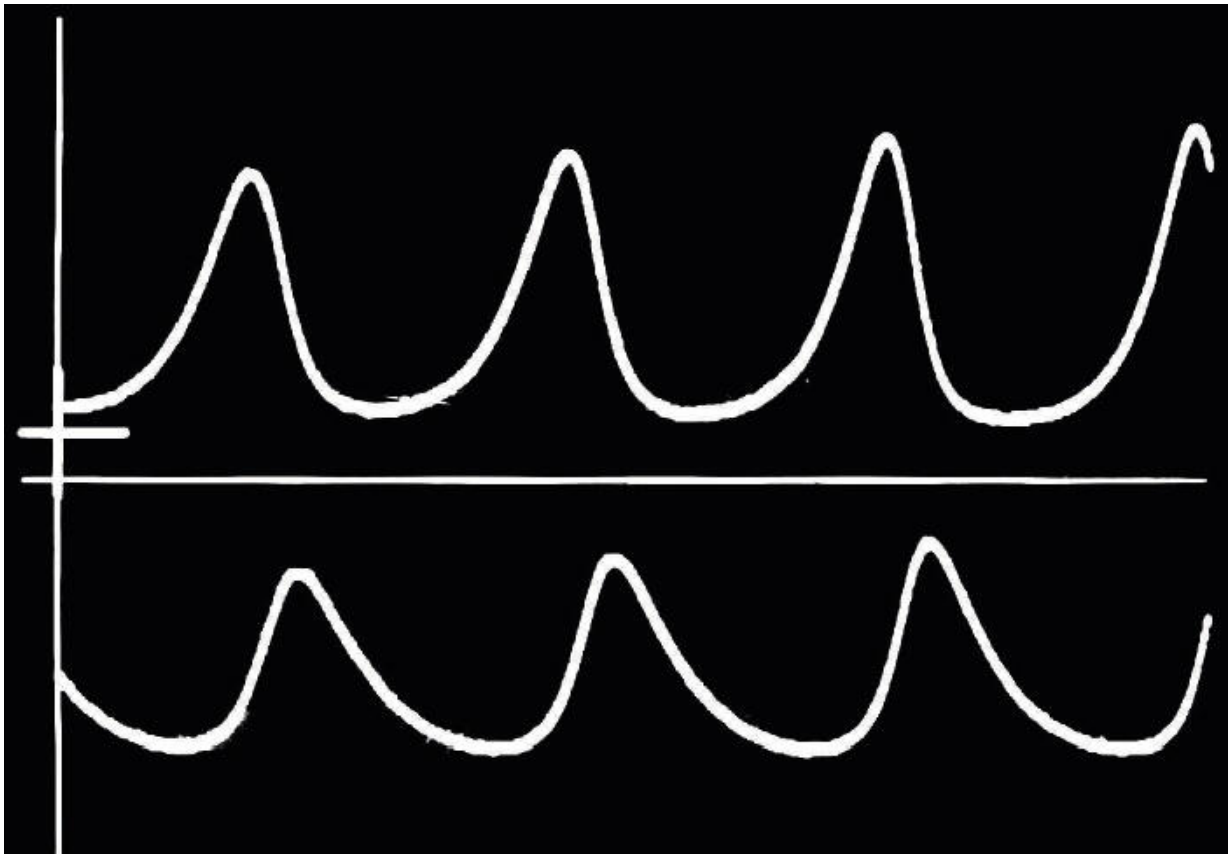


Figure 8.23. Predator-prey simulation results

Here, only a qualitative study of the simple ecosystem has been performed. Scaling a system of coupled nonlinear differential equations is more difficult than in the simple case of the mass-spring-damper system, so a qualitative study is outside the scope of this example.[16]

8.5 Simulation of an epidemic

This section describes the simulation of an epidemic based on the *SEIR* model. This model consists of four groups of individuals: Susceptible persons, i. e., persons who can be infected, exposed persons who had contact with an infected person, actually infected persons, and recovered persons who overcame the infection. This is a typical example of a compartmental model (*S, E, I*, and *R*). The following treatment is of a qualitative nature only, so no scaling has been done.[17]

The model is described by four coupled differential equations, each describing the change of the population in one compartment over time:

$$\dot{S} = -\beta SI \quad \dot{I} = \alpha E - \gamma I$$
$$\dot{E} = \beta SI - \alpha E \quad \dot{R} = \gamma I$$

β is the transmission rate, its reciprocal is the mean time between contacts of susceptible persons with infected ones. α represents the degree with which exposed persons become actually infected. γ is the recovery rate and describes how fast infected people recover (or die).

These equations can be readily transformed into the analog computer program shown in →figure 8.24. The integrator yielding S has an initial condition of −1, thus the simulation starts with the maximum number of susceptible persons. The initial condition $I(0)$ of the third integrator determines the number of persons which are initially infected.

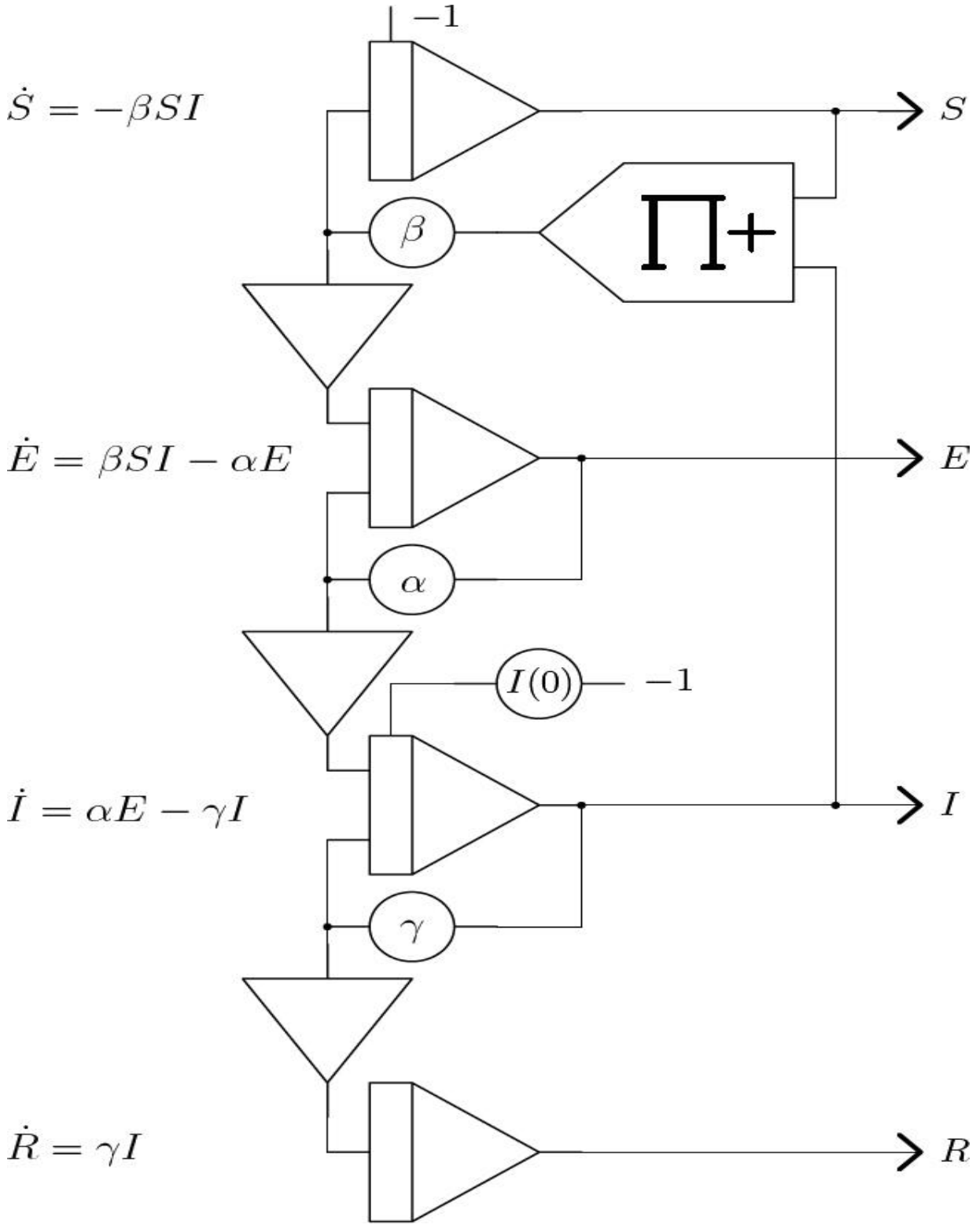


Figure 8.24. Program for the SEIR model

A typical simulation result is shown in →figure 8.25. The first curve to show a peak is the one describing the compartment of exposed persons, followed by the peak of infected persons. In this case *S* vanishes completely at the end of the simulation. With different parameters scenarios can be found where not every individual will be infected over time.

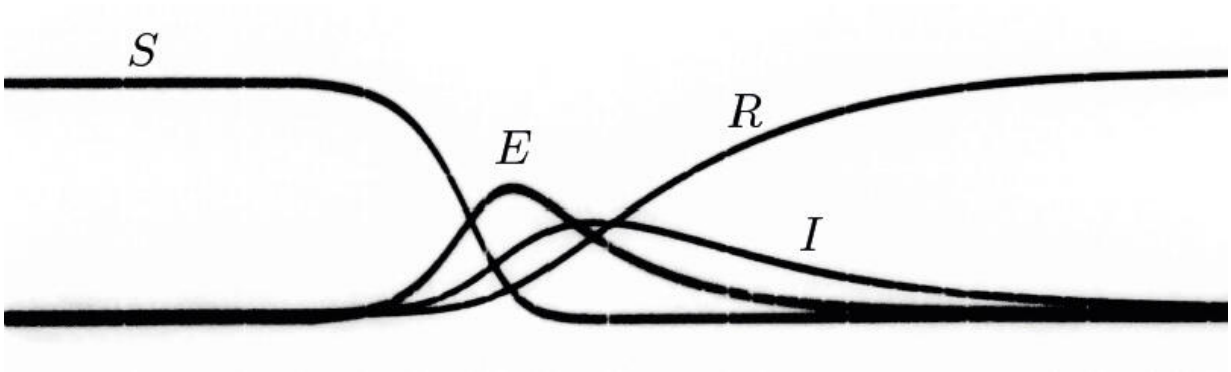


Figure 8.25. Typical SEIR simulation result

The setup of this model on THE ANALOG THING is shown in →figure 8.26. Running the analog computer in repetitive mode, a flicker-free picture can be obtained on an oscilloscope, which makes it possible to explore the effects of various settings for the coefficients *β, α*, and *γ*.



Figure 8.26. Setup of the SEIR model on THE ANALOG THING

8.6 Bouncing ball

The following section describes the simulation of a ball bouncing in a box on an analog computer with an oscilloscope as the display device. The setup follows closely an example published by Telefunken in the 1960s,[18] which was used as a marketing example to show the power of their new transistorized tabletop analog computers.[19]

→ Figure 8.27 shows the basic parts of the simulation. A ball is thrown into a rectangular box of dimension $[-1, 1] \times [-1, 1]$ with an initial velocity $v(0)$ and a given initial position $y(0)$. It is accelerated by gravitational force until it hits the floor of the box where it rebounds fully elastically, as it also does when it hits the left or right wall. The ball loses energy over time due to air-friction.

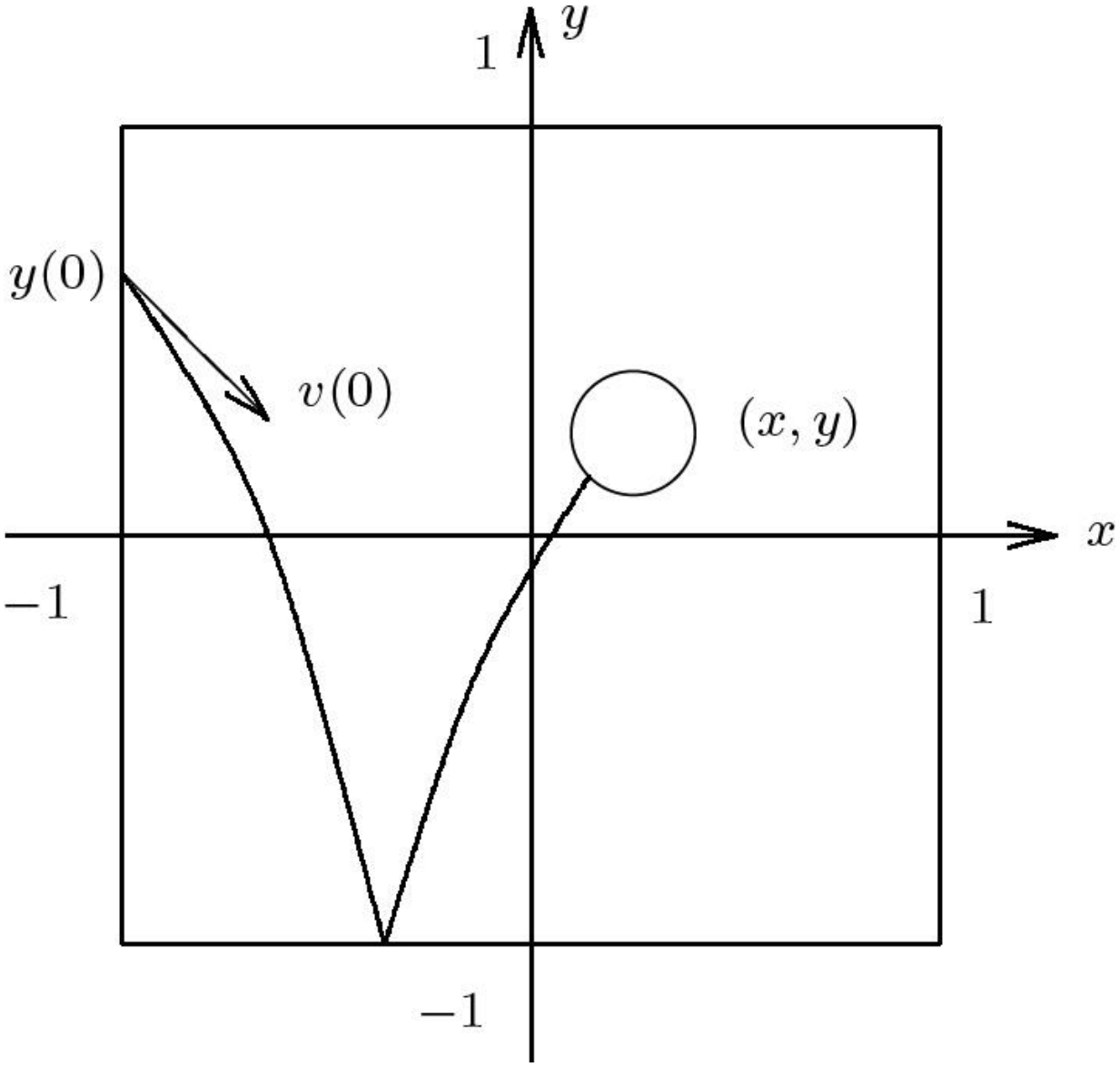


Figure 8.27. Movement of the bouncing ball

Since the *x*- and *y*-component of the ball position are independent from each other, they can be generated independently. →Figure 8.28 shows the qualitative *y*-position of the ball with respect to time. The basis for calculating *y* is the acceleration $\ddot{y}$ consisting of the gravitational force, air friction, and the acceleration caused by elastic collision of the ball with the bottom (or the ceiling) of the box. $\ddot{y}$ can be described by

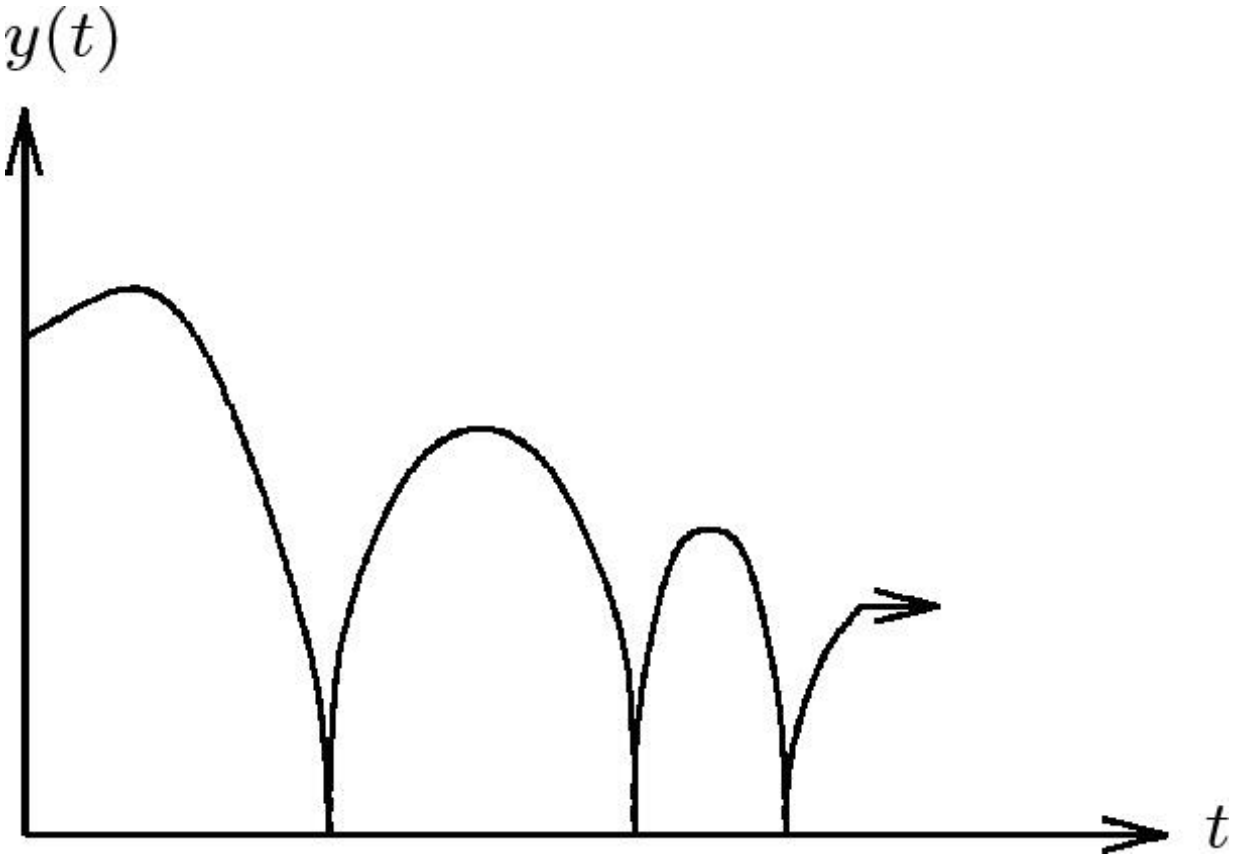


Figure 8.28. *y*-component of the bouncing ball

$$\ddot{y} = -g + d\dot{y} \begin{cases} +\frac{c}{m}(|y| - 1) & \text{if } y < -1 \\ -\frac{c}{m}(|y| - 1) & \text{if } y > 1. \end{cases}$$

Integrating twice over $\ddot{y}$ yields the ball velocity $\dot{y}$ and position y:

$$\dot{y} = \int_0^T \ddot{y}\,\mathrm{d}t + \dot{y}(0)$$
$$y = \int_0^T \dot{y}\,\mathrm{d}t + y(0)$$

The circuit yielding y is shown in →figure 8.29. The integrator on the left integrates over the sum of three terms: The gravitational force, which is set by the coefficient potentiometer labeled *g*, the rebound-acceleration, which comes from the inverter at the right side, and a damping signal that is proportional to the ball's velocity $\dot{y}$ with a factor *d*.

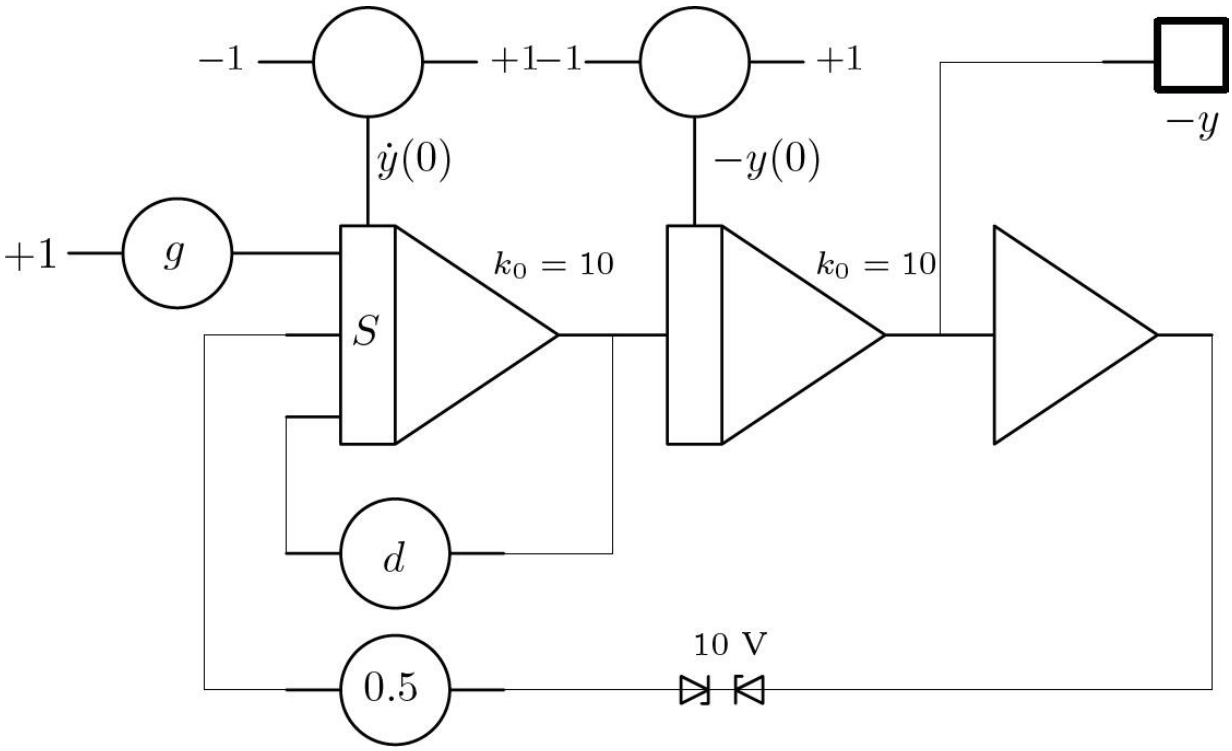


Figure 8.29. Computing the *y* position of the bouncing ball

There are several things to be noted about this circuit: First of all, the rebound-acceleration must by very high to yield a realistic behavior of the ball. Thus, an integrator input with a weight far exceeding the standard inputs of 1 and 10 is required. In this implementation this acceleration variable is fed directly to the summing junction through a coefficient potentiometer.[20] The acceleration signal representing the rebound of the ball has to be generated whenever the ball hits either the floor or the ceiling of the box.

Normally, two comparators would be used to implement this, but the small analog computer chosen in this example did not have enough comparators, so a different approach was chosen. The two ZENER diodes in the output line of the inverter have the effect that they pass a signal only when it exceeds their respective ZENER voltage. Both ZENER diodes are selected to yield a ZENER voltage of 10 Volts in each direction thus representing the floor and ceiling of the box.[21] The initial condition inputs of both integrators are connected to free coefficient

potentiometers thus allowing arbitrary initial values for the ball's *y* velocity and position.

Generating the *x*-position of the bouncing ball is even easier. It is assumed that its velocity in the *x*-direction is linearly decreasing while changing the direction every time the ball hits a wall. →Figure 8.30 shows the ball's *x*-position with respect to time while →figure 8.31 shows the corresponding computer setup.[22]

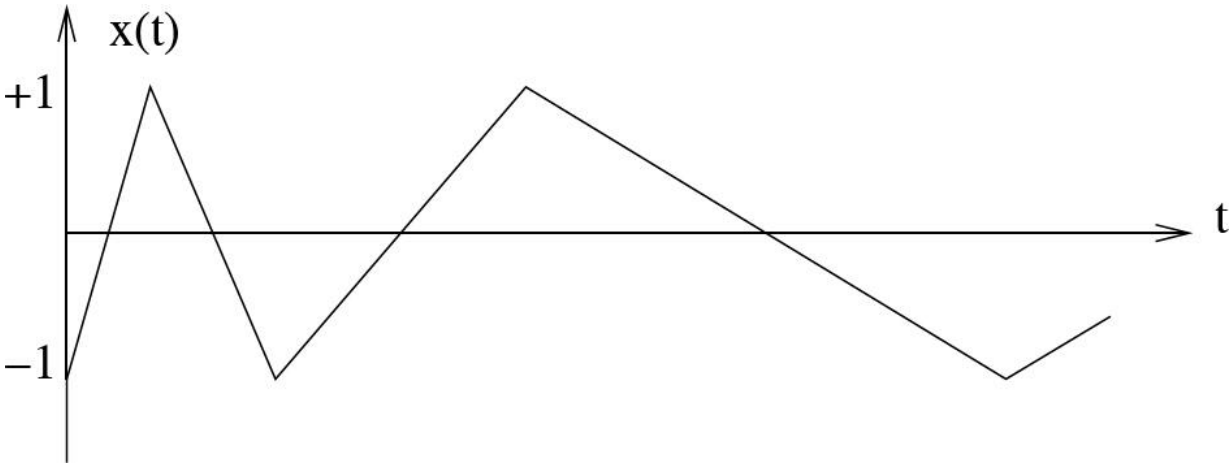


Figure 8.30. *x*-portion of the ball's movement

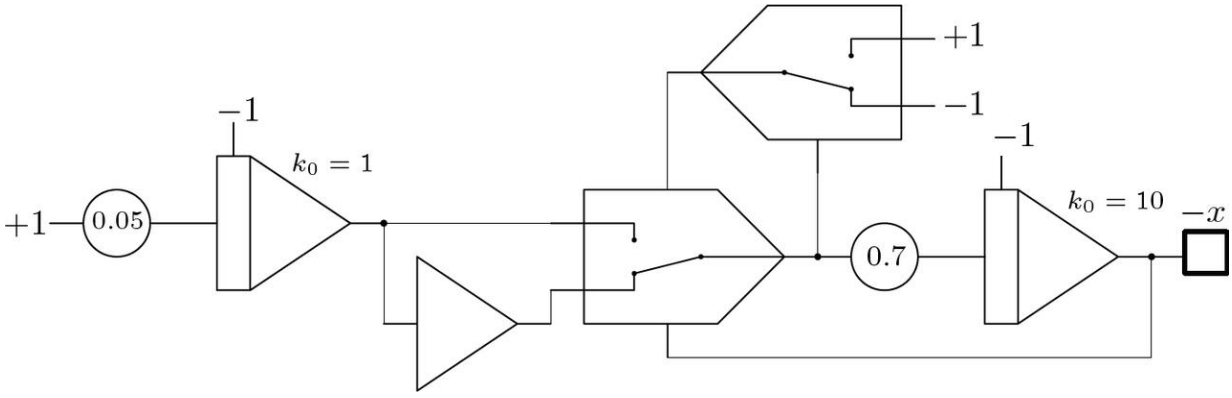


Figure 8.31. Computing the *x* position of the bouncing ball

The leftmost integrator, which generated the *x*-velocity, starts with an initial value of −1 and integrates over a small constant like 0.05 with an integration time-scale factor k_0 = 1, so the output of the integrator will

linearly decrease, starting at the value +1 representing the maximum velocity of the ball. Its output is thus

$$\dot{x}(t) = -\left(\int_0^T 0.05\mathrm{d}t - 1\right).$$

This velocity signal is then integrated by a second integrator with a time constant ten times faster to generate the *x*-position. The two comparators between the integrators detect the ball hitting the left or right wall of the box. The left comparator also performs the necessary sign-reversal of the velocity signal.

Now that both the *x*- and *y*-positions of the bouncing ball are known, a sine-/cosine quadrature generator circuit is required to generate a ball shape on an oscilloscope screen. Basically, this is done as described in →section 8.1, but it is a bit more involved here since the generated signal pair must have a high frequency and a stabilized amplitude to display a flicker-free figure on the screen. →Figure 8.32 shows the sine-/cosine generator circuit used.

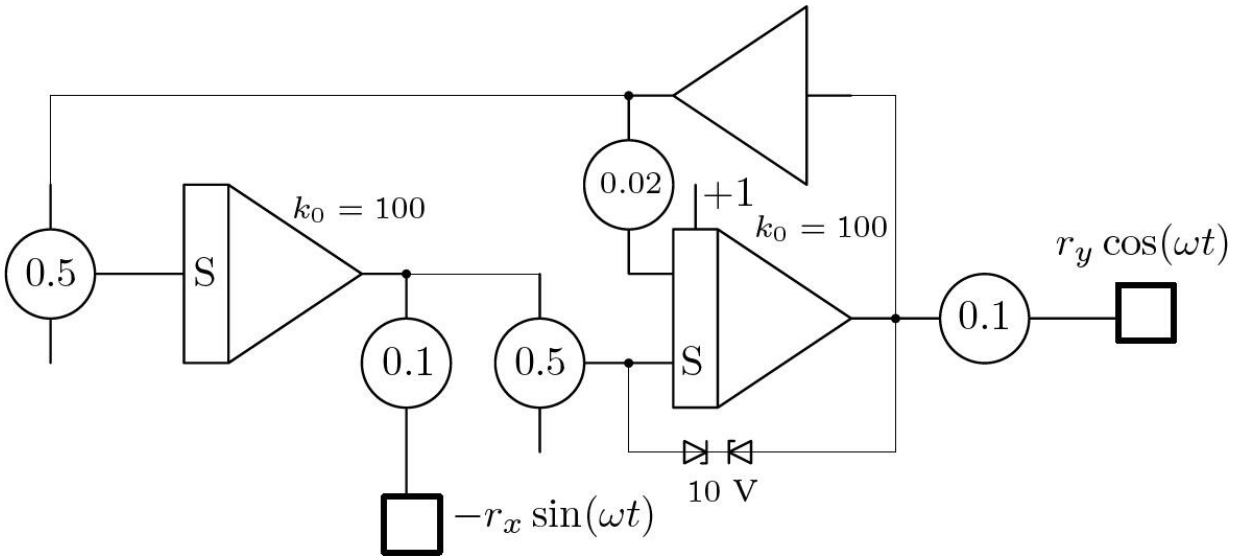


Figure 8.32. Generating a high-frequency sine-/cosine signal pair for displaying the ball

The basic structure of this circuit corresponds to that shown in →figure 8.2. Two integrators and an inverter form a loop effectively solving a

second order differential equation like (→8.1). To achieve a high value for ω, the integrators need large input weights, which is achieved by connecting free potentiometers to their respective summing junction inputs.

To ensure that the amplitude of the generated sine-/cosine signal pair does not decay, a positive feedback loop is set up between the output of the inverter and one input of the right integrator. As this loop only has to provide a small positive feedback, it is attenuated by a coefficient potentiometer set to a small value like 0.02. Without additional measures this would yield an ever increasing signal amplitude, which would drive the computing elements into overload. To counteract this effect, a pair of ZENER diodes is used as a negative feedback on the right integrator. These connect the integrator's output directly to its summing junction, effectively limiting its output signal to their ZENER voltage.[23]

Using two summers as shown in →figure 8.33, the position signals $-x$ and $-y$ of the ball and the attenuated sine-/cosine signals are combined yielding two output signals x^* and y^*, which are connected to the x- and y-inputs of the oscilloscope used to display the simulated bouncing ball.

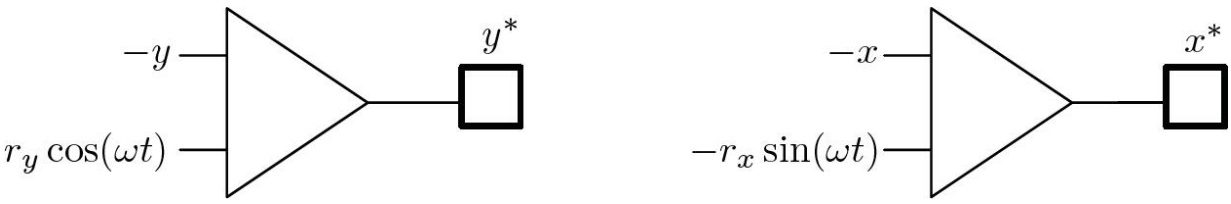


Figure 8.33. Generating the signals to display the ball's outline on the oscilloscope

→Figure 8.34 shows the setup of the bouncing ball simulation on a Dornier DO-80 analog computer, while →figure 8.35 shows a long-exposure shot of a simulation run.

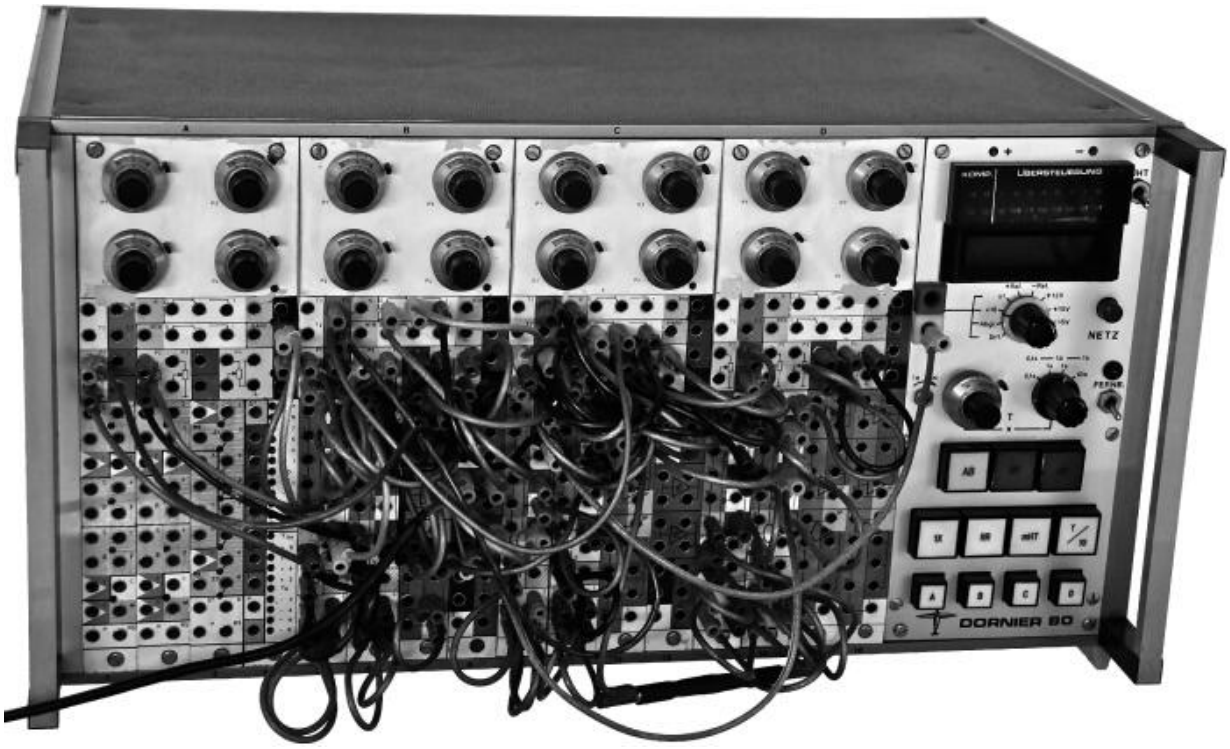


Figure 8.34. Setup of the bouncing ball simulation (Dornier 80)

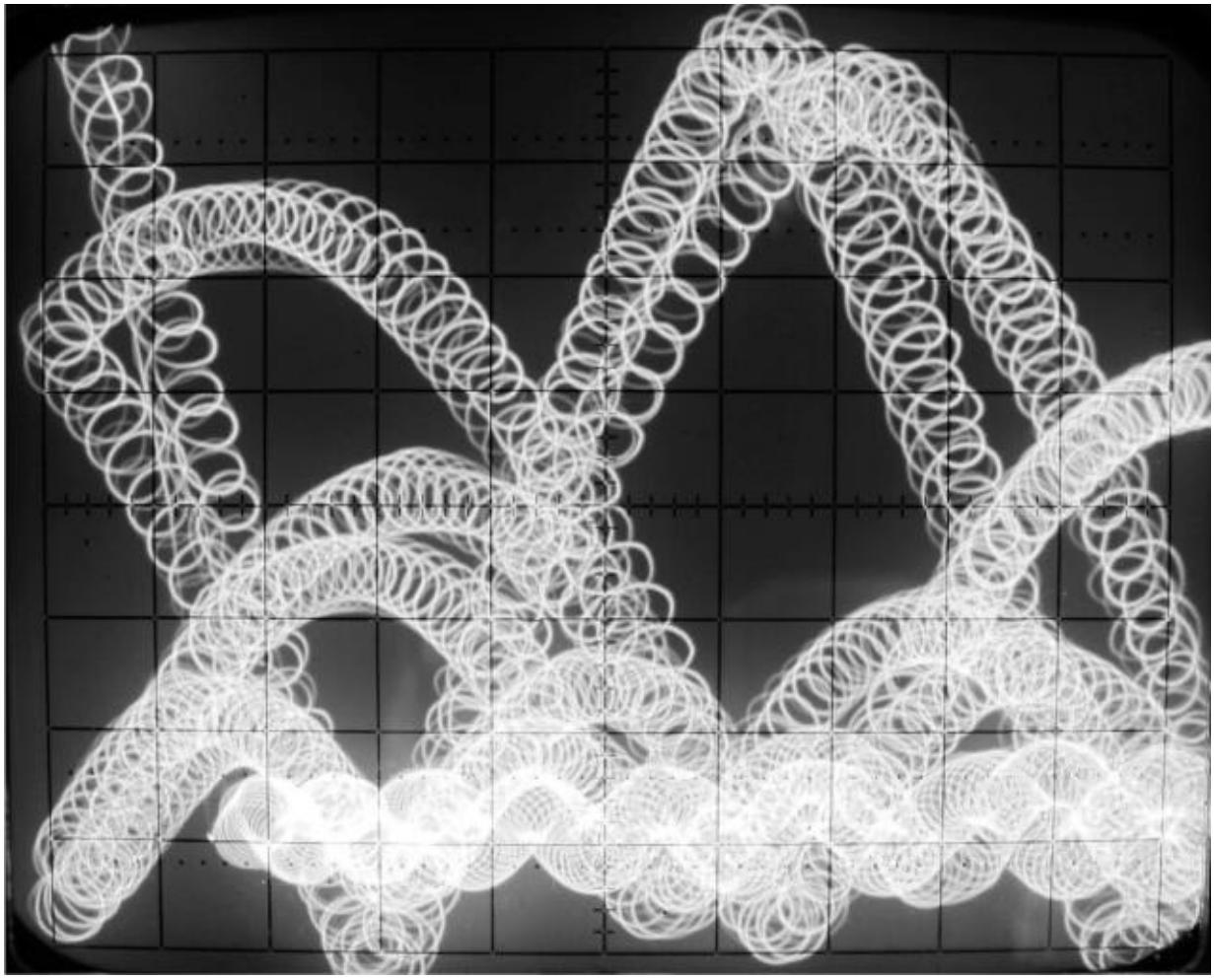


Figure 8.35. Time exposure shot of a bouncing ball simulation run (photo taken by TORE SINDING BEKKEDAL, reprinted with permission)

8.7 Car suspension

The simulation of vibrating mechanical systems was, and still is, of high commercial value since vibrations in typical transport systems, support structures, machines, etc., can cause negative effects ranging from making a train ride uncomfortable to dangerous situations when resonant frequencies are encountered causing even rigid structures to collapse.

In the late 1950s the simulation and analysis of the dynamic behavior of automobiles and railway vehicles gained a lot of interest and several manufacturers installed large analog computing facilities. The

rationale behind this is made clear by the following quotation from Robert H. Kohr:[24]

> "*However, the general trend toward heavier cars with softer tires and the increasing adoption of power steering and air suspensions calls for a complete dynamic analysis of the automobile with a view to gaining a basic understanding of the automobile's behaviour on the road*."

Realistic simulations of a car suspension system require up to seven degrees of freedom[25] and many function generators to model the nonlinear behavior of wheels, suspension springs, and other mechanical parts.[26] To make the analog model as accurate as possible, actual road profiles were traced and stored on analog tape units to be used as inputs for the simulations. Using multiple read heads, the time-delayed excitation of the front and back wheels could be easily implemented.

Simulations like these were also standard for the development of railway vehicles as this citation shows:[27]

> "*A railroad freight vehicle is a complex dynamic system consisting of numerous interrelated physical components [...] Comprehensive models for such a system will generally consist of a large number of non-linear simultaneous differential equations with complicated functional relationships between the variables to be integrated. The computer implementation of these models will require a significant amount of computer resource to run*."

Due to the complex structure of the bogies of such a vehicle, typical simulations featured between 17 and 23 degrees-of-freedom.[28] A 5 degree-of-freedom simulation of a single bogie is described in [→Malstrom et al. 1977], while [→Roedel 1955/2] covers the simulation of the dynamic behavior of railway vehicles in general. →Figure 8.36 shows the analog computer installation that was in use at the Pullman-Standard Car Manufacturing Company in the early 1950s. This installation was mainly used to simulate complete railway vehicles as well as individual bogies.

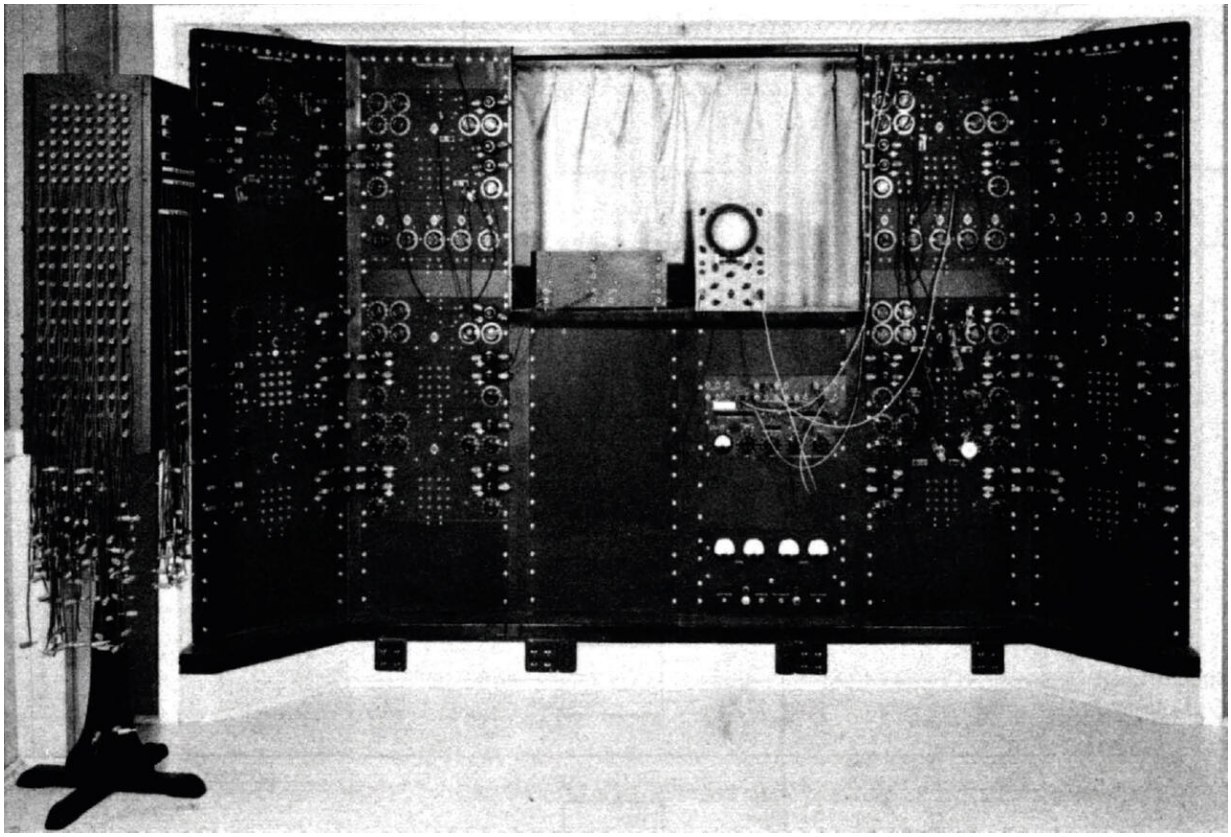


Figure 8.36. Analog computer installed at the Pullman-Standard Car Manufacturing Company in the early 1950s (see [→ROEDEL 1955/1, p. 42])

A much simpler setup was demonstrated in the early 1960s at an industry exhibition in Germany (Hannover fair). This setup, which employed a small Telefunken RAT 700 tabletop analog computer, turned out to be an eye-catcher and generated a lot of interest in these small analog computers.[29]

This setup is recreated in the following example. At the heart of the simulation is a simple coupled two-mass system with linear springs and a damper, as shown in →figure 8.37.[30] This mechanical system is described by the following two coupled differential equations

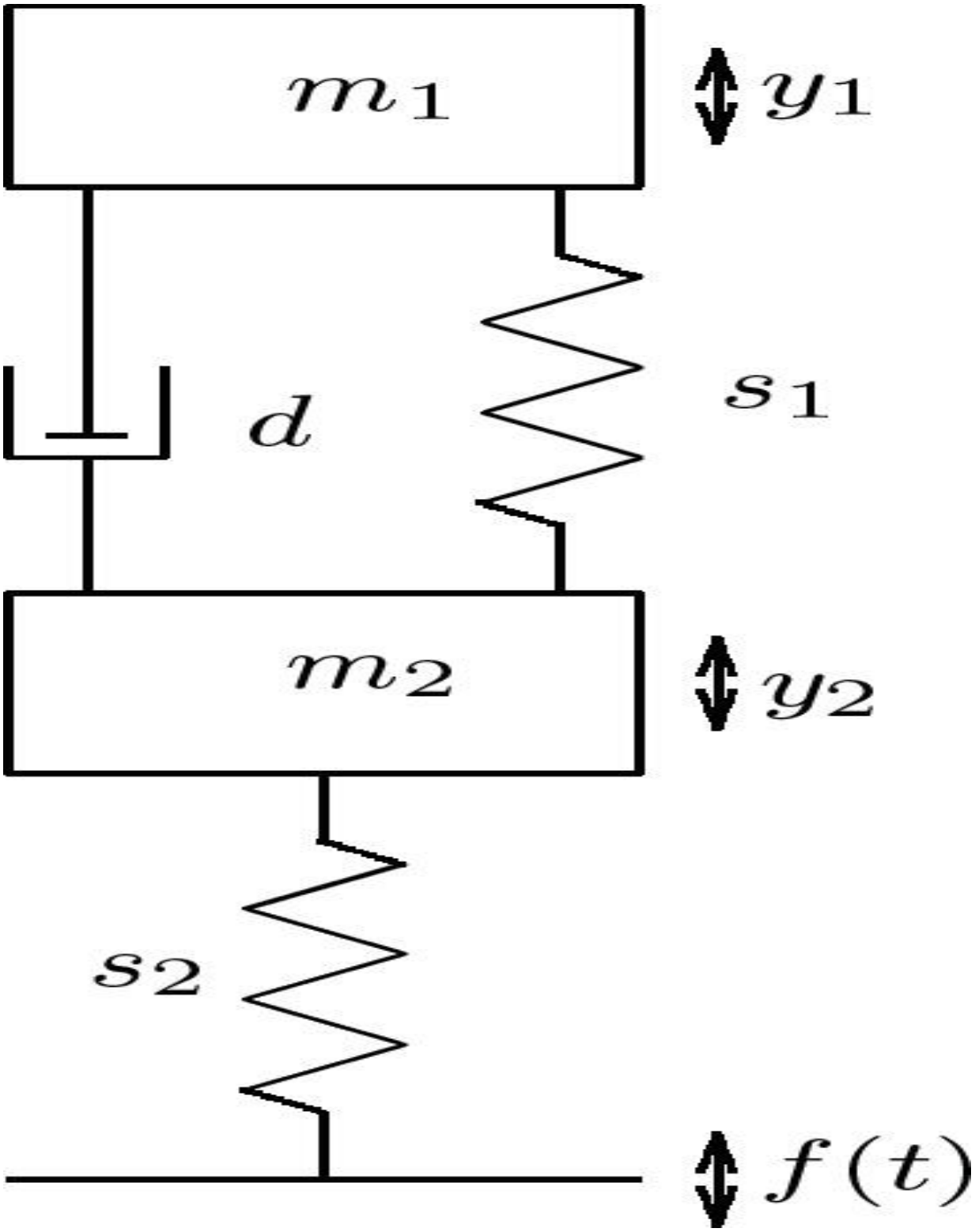


Figure 8.37. Two-mass system

$$0 = m_1\ddot{y}_1 + d(\dot{y}_1 - \dot{y}_2) + s_1(y_1 - y_2)$$
$$0 = m_2\ddot{y}_2 + d(\dot{y}_2 - \dot{y}_1) + s_1(y_2 - y_1) + s_2(y_2 - f(t)),$$

which yield

$$\ddot{y}_1 = -\left(\tfrac{d}{m_1}(\dot{y}_1 - \dot{y}_2) + \tfrac{s_1}{m_1}(y_1 - y_2)\right) \quad \text{and} \tag{8.9}$$

$$-\ddot{y}_2 = \tfrac{d}{m_2}(\dot{y}_2 - \dot{y}_1) + \tfrac{s_1}{m_2}(y_2 - y_1) + \tfrac{s_2}{m_2}(y_2 - f(t)) \tag{8.10}$$

as the starting point for applying the KELVIN method to derive a computer setup.

→ Figure 8.38 shows the circuit equivalent to equation (→ 8.9). This circuit is basically a damped oscillator consisting of two partial circuits. The oscillator itself consists of two integrators in series with a sign-inverting summer in the feedback path. A second feedback path containing two summers in series implements the damping effect $e^{-\alpha t}$ with α being defined implicitly by d/m_1.

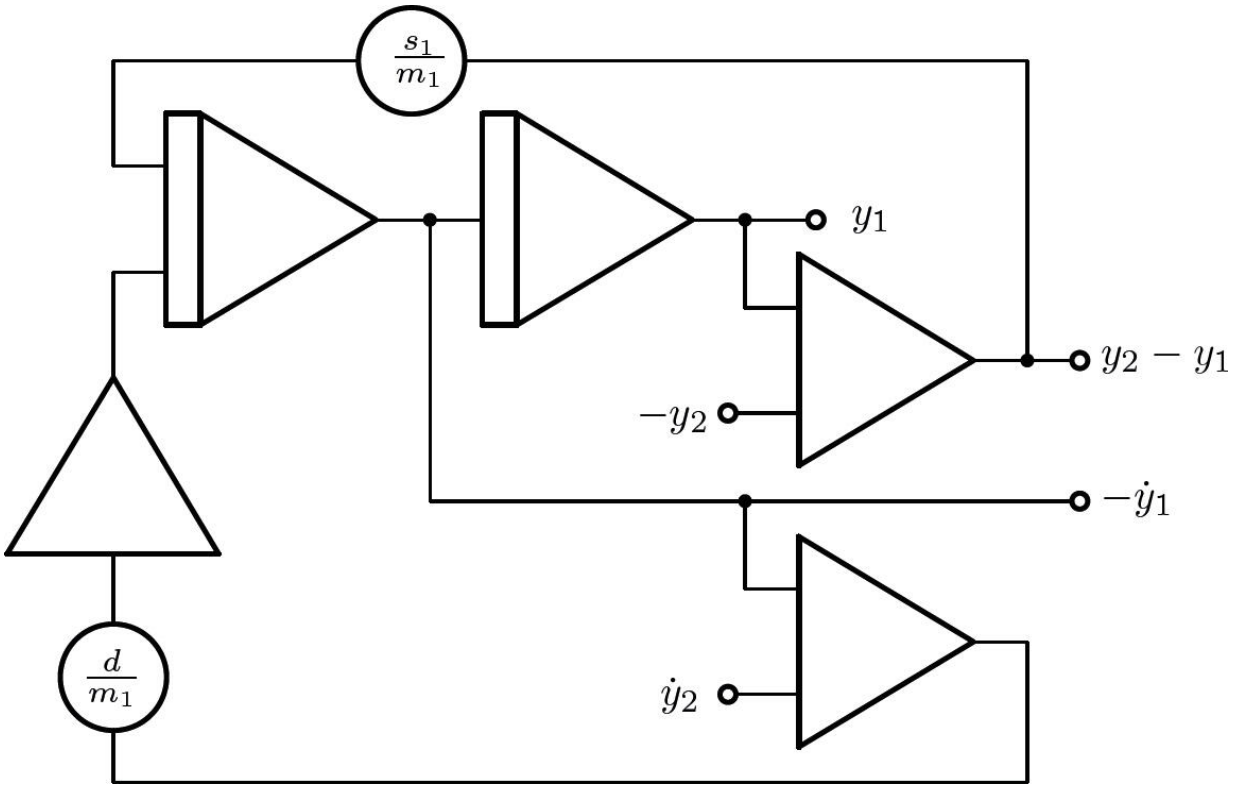


Figure 8.38. Circuit corresponding to equation (→8.9)

Equation (→8.10) is implemented by the circuit shown in →figure 8.39. It, too, is basically a damped oscillator. Both circuits are coupled through the variables y_2, $\dot{y}_2$, $-\dot{y}_1$, and $-s_1(y_2 - y_1)/m_1$. For an engineering simulation these two coupled circuits are already sufficient, but to replicate the historic exhibition setup a nice real-time display of a car bouncing up and down is required. The car is shown from the side, with the car's body representing m_1 and the wheels, which are not modeled separately, representing m_2.

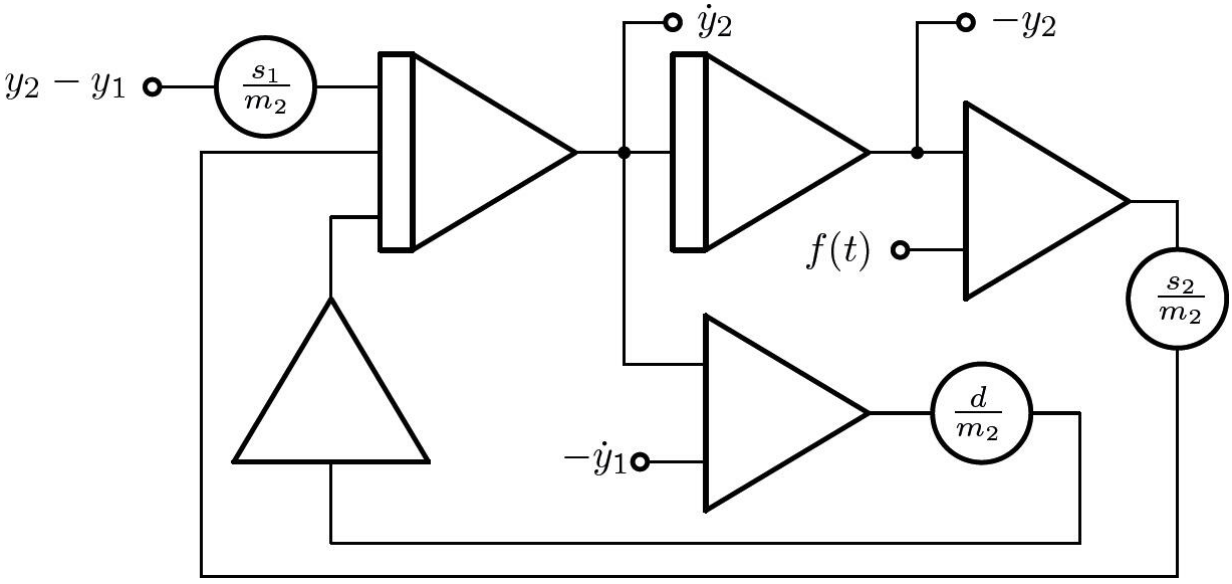


Figure 8.39. Circuit corresponding to equation (→8.10)

A high-frequency[31] sine-/cosine signal pair is an ideal basis for displaying closed figures on an oscilloscope screen. Consequently, the additional circuit shown in →figure 8.32 is required for this simulation. In addition to this, an electronic switch is necessary so that three individual figures (both wheels and the car frame) can be displayed in rapid succession on the oscilloscope. The wheels can be directly generated by feeding the *x*- and *y*-deflection channel of the oscilloscope with the sine-/cosine-pair plus the wheel displacement y_2. The car frame is also based on this signal pair with the cosine part suitably modified by means of a function generator. Adding its corresponding height signal, y_1 controls the *y* position at which it is displayed.

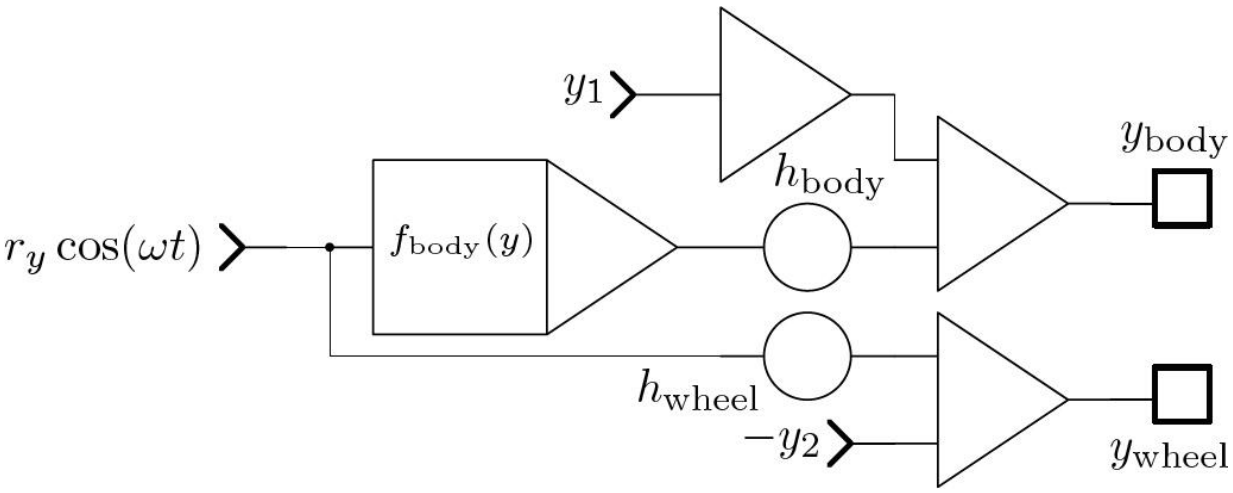


Figure 8.40. Generating the signals for the graphic representation of the two-mass system

The overall program setup on a Telefunken RA 770 precision analog computer is shown in →figure 8.41. In addition to the circuits described above, it also contains the three pairs of electronic switches to rapidly switch the oscilloscope's inputs between the three *x*/*y*-outputs generated by the simulation. Additionally, a sweep generator like the one shown in →figure 8.4 has been implemented to explore the dynamic response of the two-mass vibrating system to various excitation frequencies and amplitudes. Also, a second excitation source is available in this setup since this analog computer is equipped with a noise generator. A screenshot of the display generated by this setup is shown in →figure 8.42.

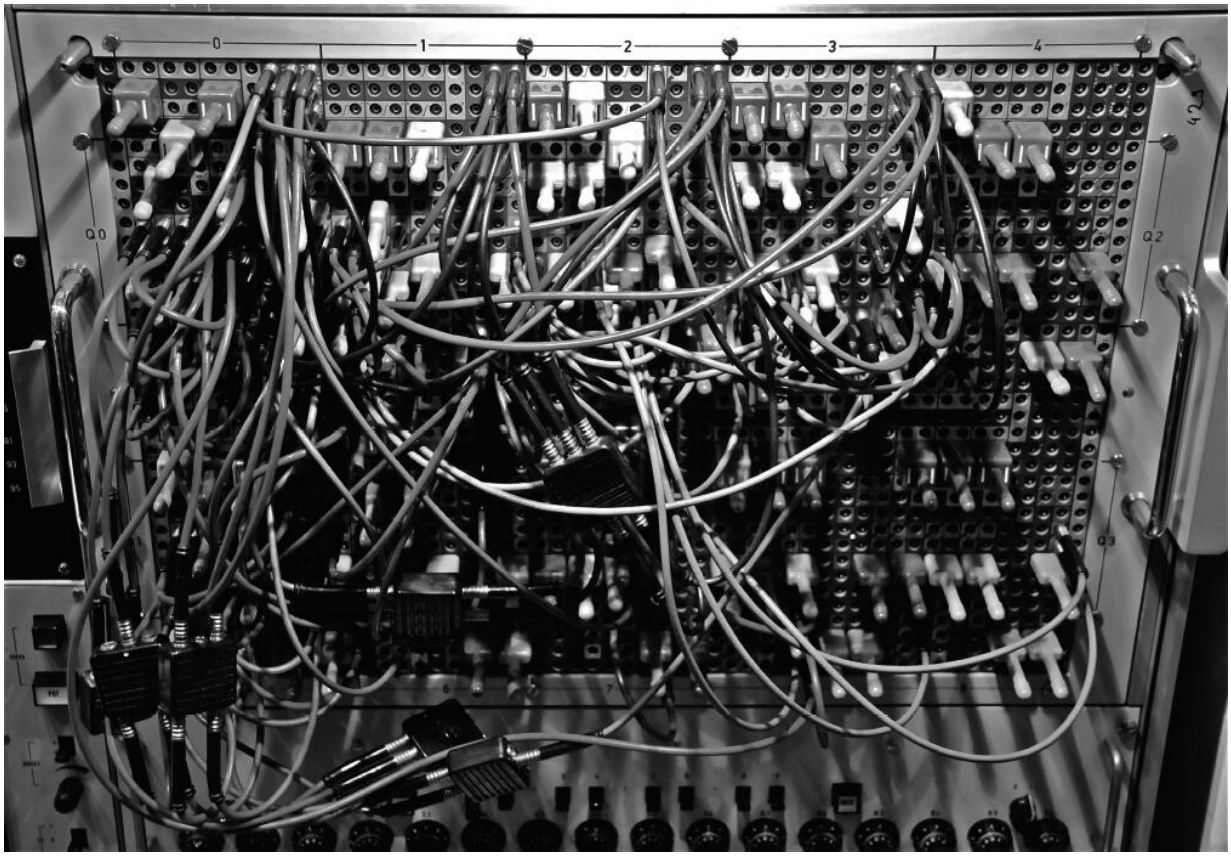


Figure 8.41. Setup of the car suspension simulation (Telefunken RA 770)

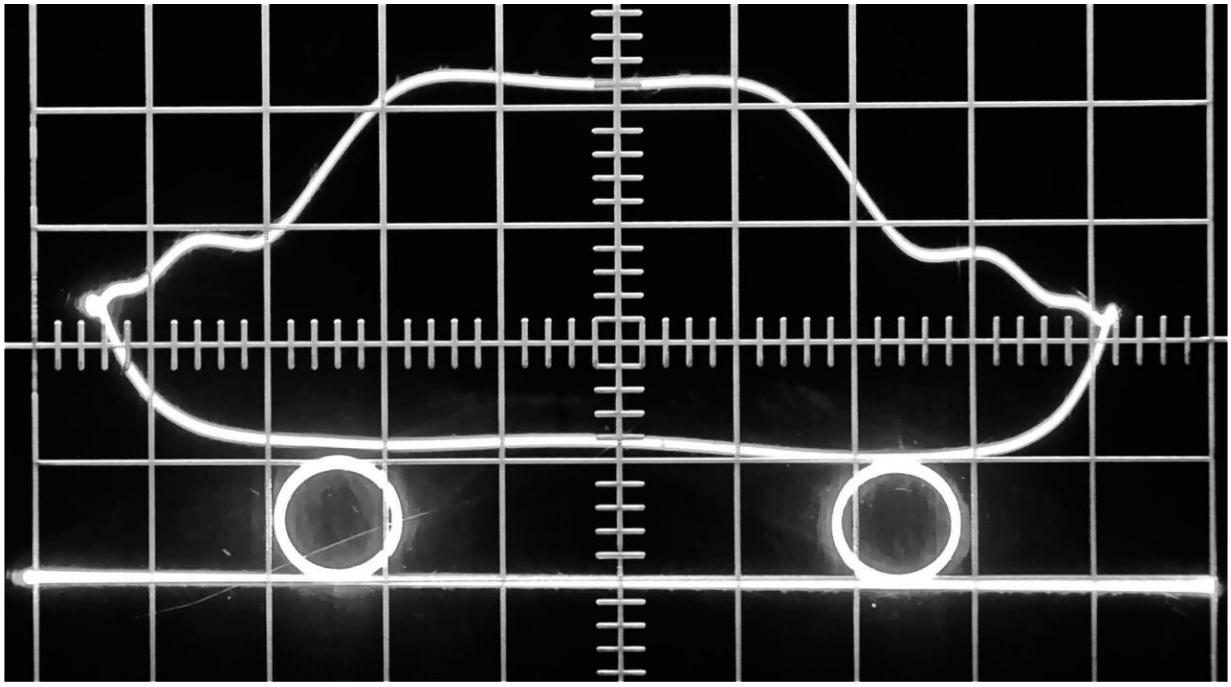


Figure 8.42. Display of the car suspension simulation

8.8 Lorenz attractor

A well-known chaotic system is the *Lorenz attractor*, discovered and published by Edward N. Lorenz[32] in 1963.[33] He devised a simplified model for atmospheric convection, which he analyzed on a simple digital computer, a Royal McBee *LGP-30*, and it turned out that this system exhibits a behavior nowadays called *chaotic*.[34] An analog computer is ideally suited to implement a chaotic system like this based on the defining differential equations[35]

$$\dot{x} = \sigma(y - x) \tag{8.11}$$

$$\dot{y} = x(\rho - z) - y \tag{8.12}$$

$$\dot{z} = xy - \beta z \tag{8.13}$$

with parameters σ = 10, ρ = 28, and $\beta = \frac{8}{3}$. A quick look into the literature or a numerical experiment show that $|x| < 20$, $|y| < 30$, and

$|z| < 50$. Accordingly, suitable scaling factors for these three variables would be $\lambda_x = \frac{1}{20}$, $\lambda_y = \frac{1}{30}$, and $\lambda_z = \frac{1}{50}$.

Rewriting equations (→8.11), (→8.12), and (→8.13) in integral form already resembles the analog computer setup:

$$\begin{aligned} x &= \int_0^T (\sigma y - \sigma x)\,\mathrm{d}t \\ y &= \int_0^T (\rho x - xz - y)\,\mathrm{d}t \\ z &= \int_0^T (xy - \beta z)\,\mathrm{d}t \end{aligned}$$

In the first step x is scaled by λ_x, yielding

$$x = \tfrac{1}{20}\int_0^T (\sigma y - 20\sigma x)\,\mathrm{d}t = \int_0^T \left(\tfrac{1}{20}\sigma y - \sigma x\right)\mathrm{d}t = \int_0^T \left(\tfrac{y}{2} - 10x\right)\mathrm{d}t \tag{8.14}$$

$$y = \int_0^T (20\rho x - 20xz - y)\,\mathrm{d}t \tag{8.15}$$

$$z = \int_0^T (20xy - \beta z)\,\mathrm{d}t. \tag{8.16}$$

The changes in (→8.15) and (→8.16) compensate for the downscaling of x.[36] Now y gets scaled with λ_y, yielding

$$\begin{aligned} x &= \int_0^T (15y - 10x)\,\mathrm{d}t \\ y &= \frac{1}{30}\int_0^T (20\rho x - 20xz - 30y)\,\mathrm{d}t = \int_0^T \left(\frac{2}{3}\rho x - \frac{2}{3}xz - y\right)\mathrm{d}t \\ z &= \int_0^T (20 \cdot 30xy - \beta z)\,\mathrm{d}t = \int_0^T (600xy - \beta z)\,\mathrm{d}t. \end{aligned}$$

Finally, z must be scaled with λ_z:[37]

$$\begin{aligned} y &\approx \int_0^T \left(\frac{2}{3}\rho x - 33.3xz - y\right) \mathrm{d}t \approx \int_0^T (18.6x - 33.3xz - y)\,\mathrm{d}t \\ z &= \frac{1}{50}\int_0^T (600xy - 50\beta z)\,\mathrm{d}t \approx \int_0^T (12xy - 2.667z)\,\mathrm{d}t. \end{aligned}$$

Since many of the coefficients are ≥ 10 all three equations will be downscaled again by a factor of $\frac{1}{10}$ to simplify the actual implementation on an analog computer. Since this downscaling affects all equations, it just causes an overall reduction in computation speed by a tenth, which typically can be compensated for by selecting a larger k_0 if required. The final equations for the Lorenz system are

$$x = \textstyle\int_0^T (1.5y - x)\,\mathrm{d}t \tag{8.17}$$

$$y = \textstyle\int_0^T (1.86x - 3.33xz - 0.1y)\,\mathrm{d}t \tag{8.18}$$

$$z = \textstyle\int_0^T (1.2xy - 0.2667z)\,\mathrm{d}t. \tag{8.19}$$

These equations can now be mechanized as shown in →figure 8.43. To display the famous phase space plot in →figure 8.45, the functions *x* and *z* are required. *x* is already available in the setup, but *z* requires an additional inverter since only −*z* is available in the setup. Taking advantage of the implicit sign inversions performed by summers and integrators, the circuit can be slightly changed as shown in →figure 8.44. Now, *x* and *z* are available without the requirement for an additional inverter.

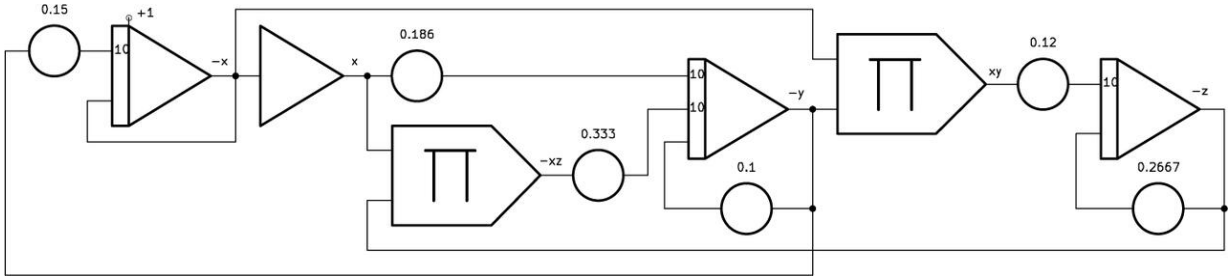


Figure 8.43. Direct implementation of equations (→8.17), (→8.18), and (→8.19)

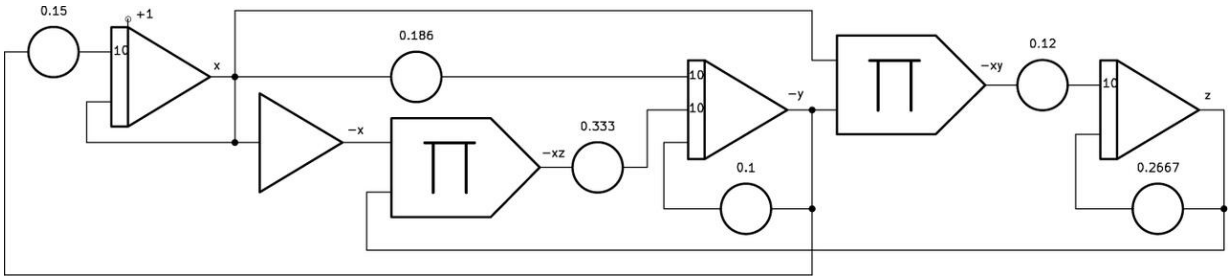


Figure 8.44. Slightly improved implementation of equations (→8.17), (→8.18), and (→8.19)

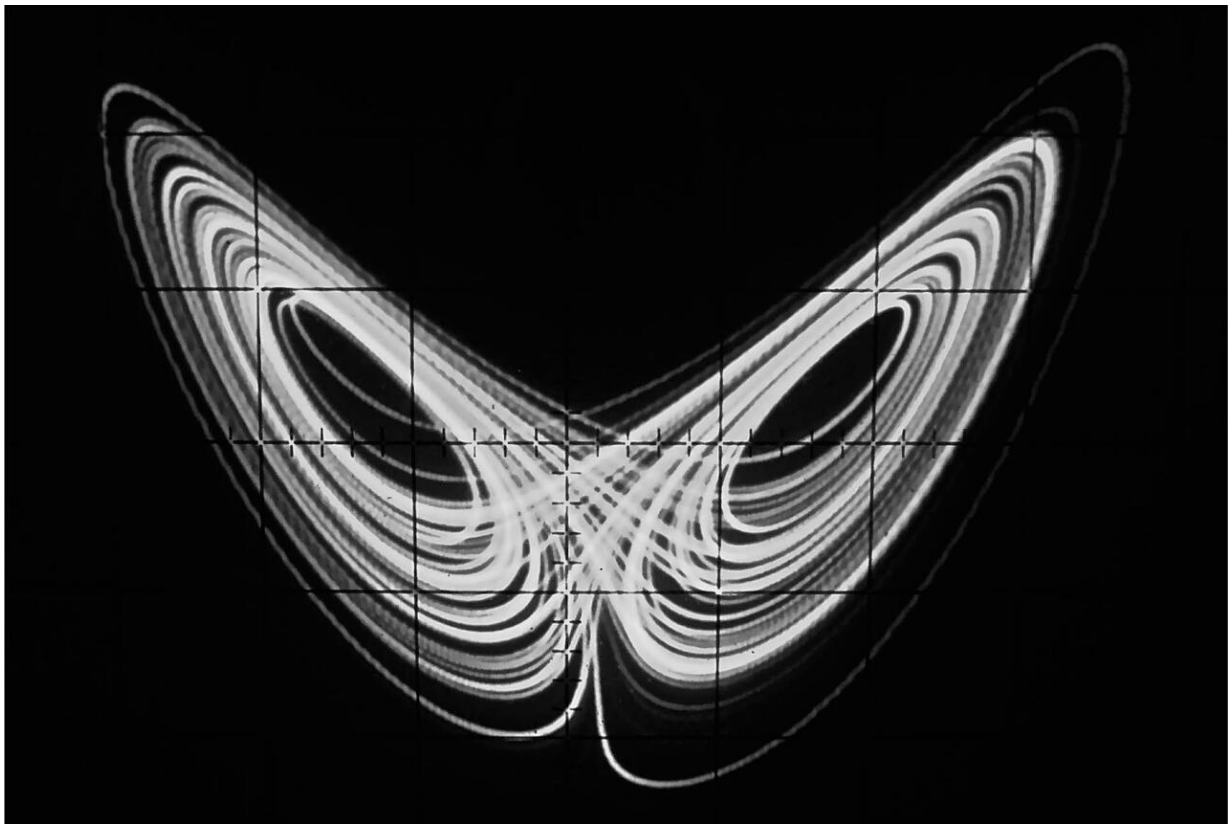


Figure 8.45. Screenshot showing a Lorenz attractor in the *x, z* phase space

8.9 Mathieu's equation

One of the author's favorite equations is Mathieu's differential equation

$$\ddot{y} + (a - 2q\,\cos(2t))y = 0, \tag{8.20}$$

with initial conditions

$$\begin{aligned} y(0) &= 1 \quad \text{and} \\ \dot{y}(0) &= 0, \end{aligned} \tag{8.21}$$

which was devised by Émile Léonard Mathieu[38] during his theoretical studies of vibrating surfaces such as drum skins.[39] Nowadays, this equation finds applications in quantum mechanics, optical systems, and many other areas.[40]

This equation is a good example of the scaling process in analog computer programming.[41] Without loss of generality,

$$a := 2q \quad \text{and} \qquad (8.22)$$
$$x := 1 - \cos(2t)$$

are defined, thus transforming (→8.20) into

$$\ddot{y} + axy = 0. \qquad (8.23)$$

The first task is now to generate the function *x*(*t*), which is done by devising an auxiliary differential equation yielding *x*(*t*) as a solution. Therefore, (→8.22) is differentiated twice yielding

$$\dot{x} = 2\ \sin(2t) \quad \text{and} \qquad (8.24)$$

$$\ddot{x} = 4\ \cos(2t). \qquad (8.25)$$

The resulting differential equation is then

$$\ddot{x} + 4x = 4. \qquad (8.26)$$

Using two integrations, the unscaled program would basically look like this:

$$\ddot{x} = 4 - 4x, \qquad (8.27)$$

$$\dot{x} = \int \ddot{x}\, \mathrm{d}t, \quad \text{and} \qquad (8.28)$$

$$x = \int \dot{x}\, \mathrm{d}t. \qquad (8.29)$$

According to (→8.22) 0 ≤ *x* ≤ 2, so (→8.29) must be scaled with a factor $\lambda_x = \frac{1}{2}$ to ensure that *x* does not exceed the machine interval.[42] This yields the scaled equation[43]

$$\hat{x} = \lambda_x \int \dot{x}\, \mathrm{d}t = \tfrac{1}{2} \int \dot{x}\, \mathrm{d}t. \qquad (8.30)$$

This scaling factor must now be compensated for in (→8.27):

$$\ddot{x} = 4 - \tfrac{1}{\lambda_x} 4\hat{x} = 4 - 8\hat{x}. \tag{8.31}$$

$-4 \le \ddot{x} \le 4$, according to (→8.25), so (→8.31) must be scaled with a factor $\lambda_{\ddot{x}} = \frac{1}{4}$ yielding

$$\widehat{\ddot{x}} = \lambda_{\ddot{x}} 4 - \lambda_{\ddot{x}} 8\widehat{x} = 1 - 2\widehat{x},$$

which must be compensated for in (→8.28):

$$\dot{x} = \tfrac{1}{\lambda_{\ddot{x}}} \int \widehat{\ddot{x}} \, \mathrm{d}t = 4 \int \widehat{\ddot{x}} \, \mathrm{d}t$$

According to (→8.24) this is bounded by $-2 \le \dddot{x} \le 2$ requiring a third scale factor $\lambda_{\dot{x}} = \frac{1}{2}$ yielding

$$\dot{x} = \lambda_{\dot{x}} 4 \int \widehat{\ddot{x}} \, \mathrm{d}t = 2 \int \widehat{\ddot{x}} \, \mathrm{d}t.$$

The factor $\lambda_{\dddot{x}}$ cancels out the factor $\lambda_{\dot{x}}$ in equation (→8.30). This results in the following set of scaled equations to implement the auxiliary differential equation:

$$\begin{aligned} \widehat{\ddot{x}} &= 1 - 2\widehat{x}, \\ \widehat{\dot{x}} &= 2 \int \widehat{\ddot{x}} \, \mathrm{d}t, \text{ and} \\ \widehat{x} &= \int \widehat{\dot{x}} \, \mathrm{d}t. \end{aligned}$$

This ensures that $\dot{x}$, $\ddot{x}$, and $\dddot{x}$ all stay within the machine interval [−1, 1]. The resulting analog computer setup generating x is shown in →figure 8.46.

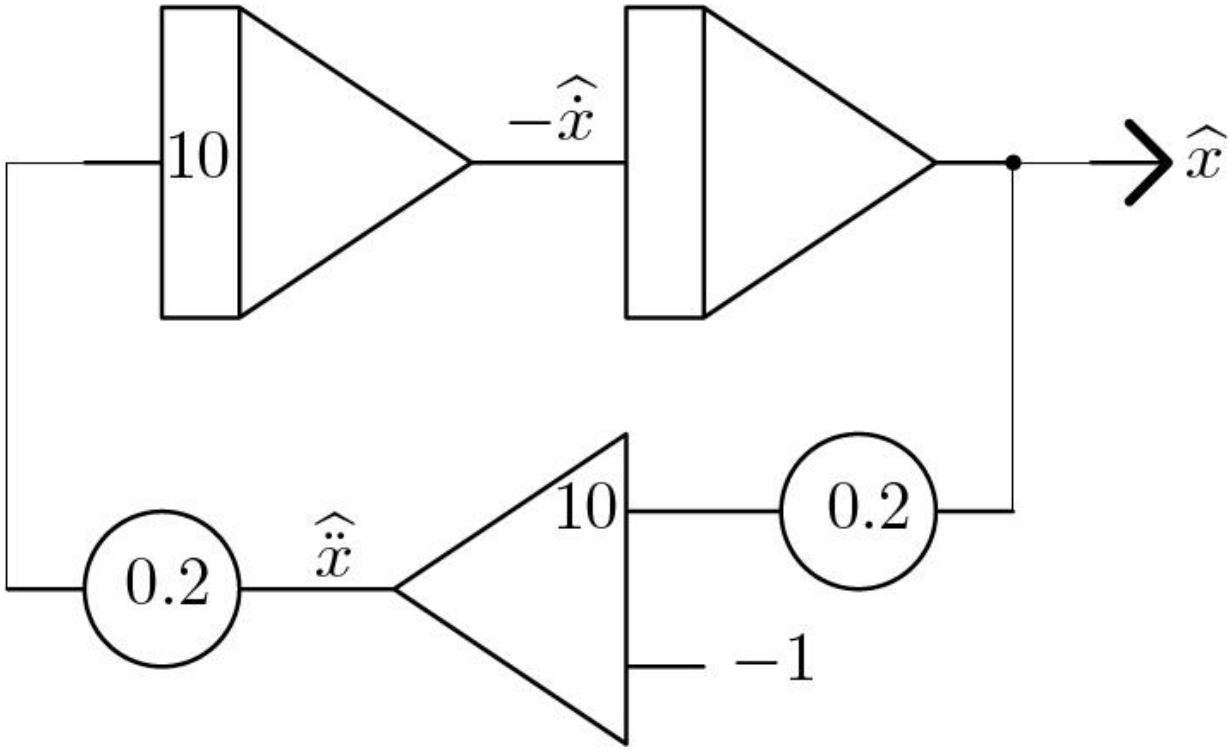


Figure 8.46. Scaled setup for equation (→8.26)

Now equation (→8.23) must be implemented. Since $0 \leq x \leq 2$, this can be reduced to[44]

$$\ddot{y} + 2ay = 0 \tag{8.32}$$

for scaling. This equation describes a harmonic oscillator, so it has a solution

$$y = y(0)\ \cos(\omega t) \tag{8.33}$$

with $y(0) = 1$ according to (→8.21). For this harmonic oscillator $\omega^2 = 2a$. Differentiating (→8.33) yields

$$\dot{y} = -y(0)\omega\ \sin(\omega t). \tag{8.34}$$

Based on (→8.32), a basic analog computer program can be set up according to

$$\begin{aligned} \dot{y} &= \int \ddot{y} \, \mathrm{d}t, \quad \text{and} \\ y &= \int \dot{y} \, \mathrm{d}t. \end{aligned} \tag{8.35}$$

$$\ddot{y} = -2ay. \tag{8.36}$$

Since MATHIEU's equation tends to instability, a scaling factor $\lambda_y = \frac{1}{5}$ is introduced into equation (→8.35), which must be compensated for in (→8.36):

$$\dot{y} = \int \ddot{y} \, \mathrm{d}t \tag{8.37}$$

$$\hat{y} = \lambda_y \int \dot{y} \, \mathrm{d}t = \tfrac{1}{5} \int \dot{y} \, \mathrm{d}t \tag{8.38}$$

$0 \leq |_{\dot{y}}| \leq y(0)\omega$ according to (→8.34). If $0 \leq a \leq 10$, then $\omega = \sqrt{20} \approx 5$. This, in conjunction with the safety scaling factor $\lambda_{\dot{y}}$, which was chosen to at least partially counteract the blow-up tendencies of the equation, yields the scaling factor $\lambda_{\dot{y}} = \frac{1}{25}$, which must be applied to (→8.37) and compensated for in (→8.38):

$$\begin{aligned} \dot{\hat{y}} &= \lambda_y \int \ddot{y} \, \mathrm{d}t = \frac{1}{25} \int \ddot{y} \, \mathrm{d}t \\ \hat{y} &= \frac{\lambda_y}{\lambda_{\hat{y}}} \int \dot{\hat{y}} \, \mathrm{d}t = t \int \dot{\hat{y}} \, \mathrm{d}t \\ \ddot{y} &= -10a\hat{y} \end{aligned} \tag{8.39}$$

The scale factors can now be regrouped, i. e., the factor 10 in front of a in (→8.39) will be moved into $\lambda_{\dot{y}}$. Scaling a down by another factor of 10 to simplify setting of this parameter introduces another factor of 10, which can be implemented by using an input with weight 10 at the integrator yielding $-\dot{y}$. Finally, including a multiplier in the loop of this program, yielding $\frac{a}{10}\widehat{x}y$, introduces another factor of 2 since $0 \leq \hat{x} \leq 1$ instead of $0 \leq x \leq 2$. The resulting program is shown in →figure 8.47. The overall implementation on THE ANALOG THING is shown in →figure 8.48.

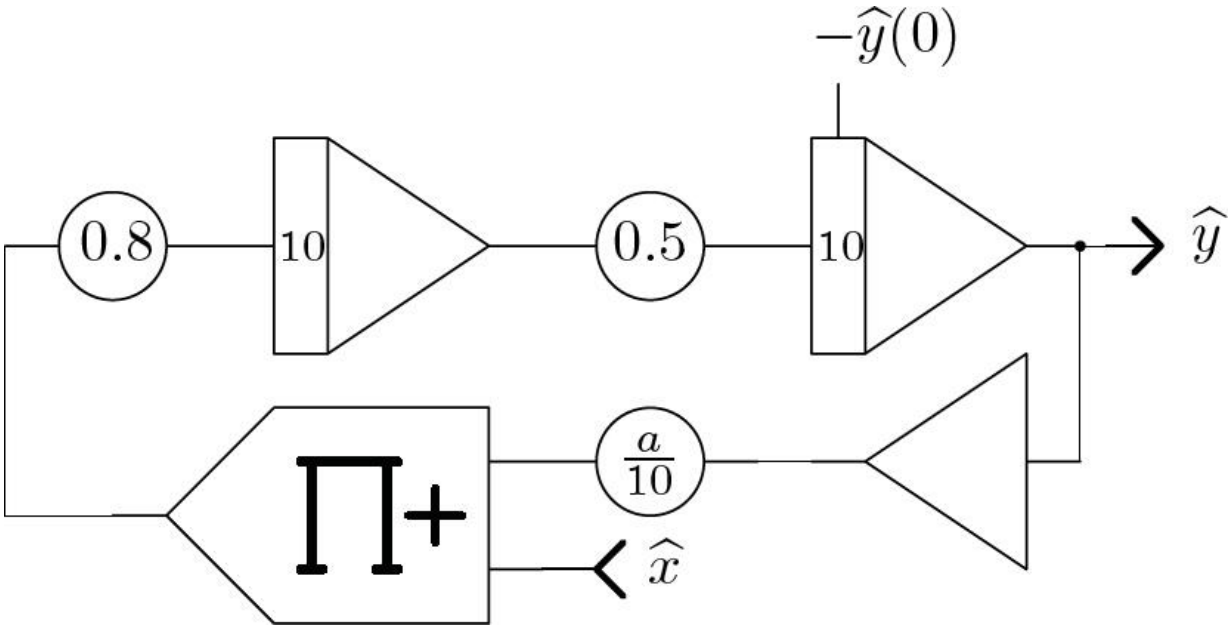


Figure 8.47. Second half of the implementation of MATHIEU's equation

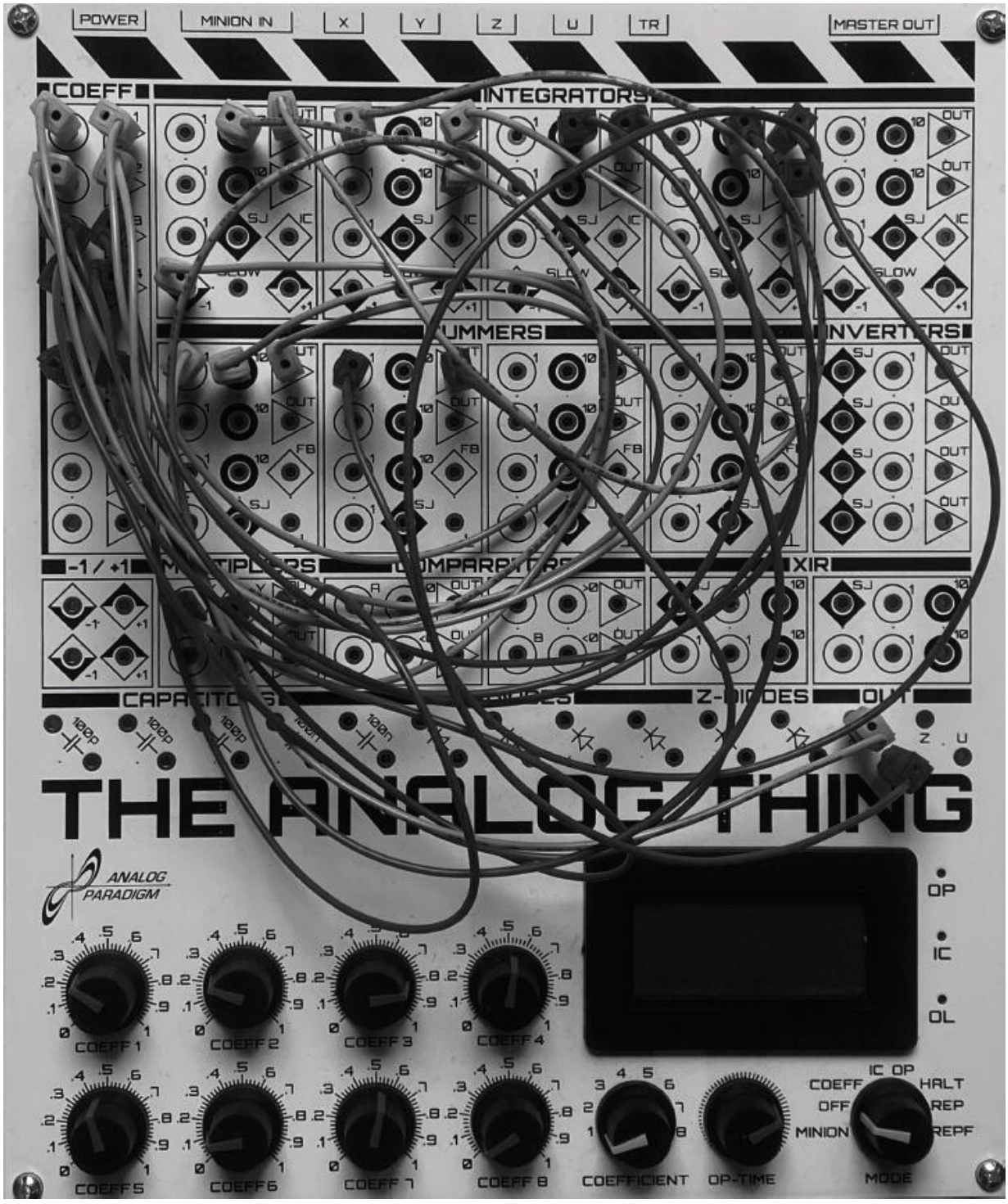


Figure 8.48. Setup of MATHIEU's equation on THE ANALOG THING

A typical solution for the MATHIEU equation is shown in →figure 8.49. This picture was taken with a digital oscilloscope and shows the first few

milliseconds of the solution.

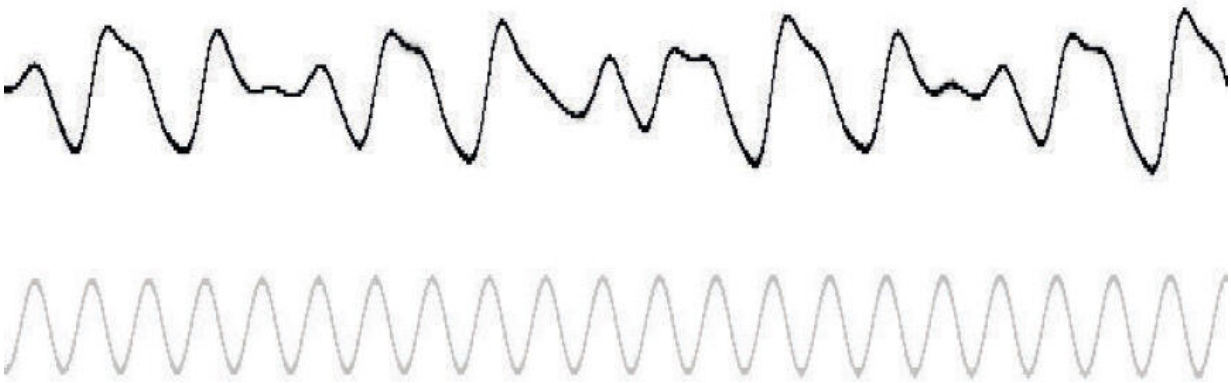


Figure 8.49. Typical behavior of the MATHIEU equation

8.10 Projection of rotating bodies

The following example shows how an analog computer operating in repetitive mode can be used to display the projection of three-dimensional rotating figures.[45] It generates a two-dimensional display of a rotating three-dimensional spiral, is based on [→Telefunken/3] and [→Hitachi 1967].

At first, the coordinates of a three-dimensional spiral must be generated. This is done by the modified sine-/cosine-generator shown in →figure 8.50 yielding

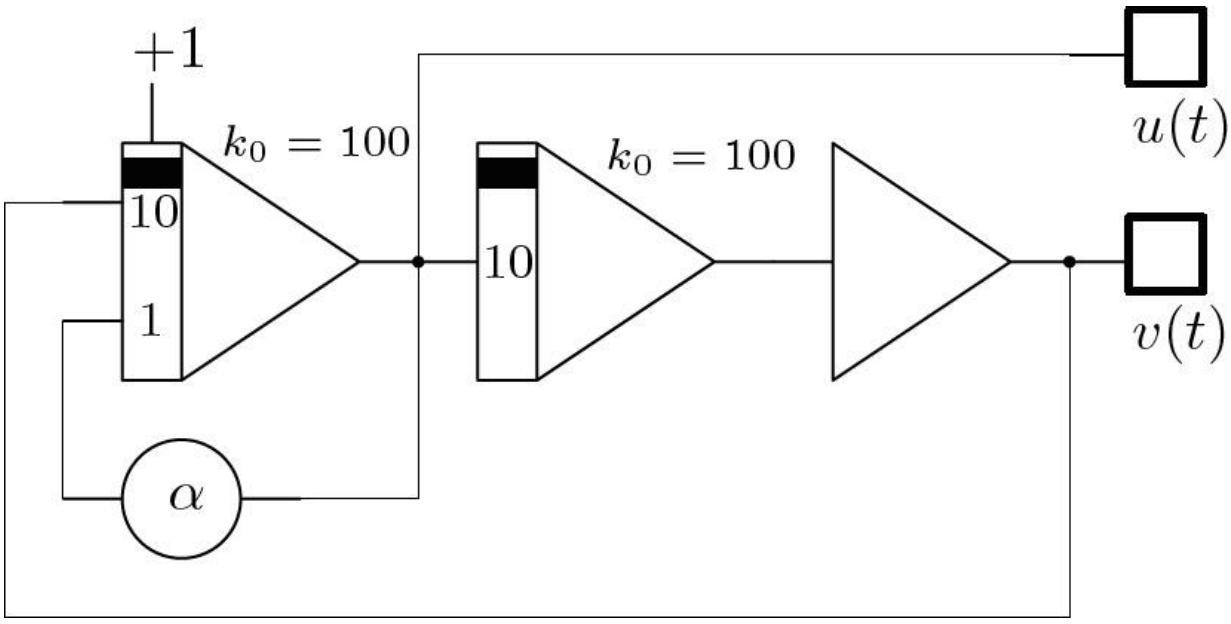


Figure 8.50. Generating *u*(*t*) and *v*(*t*) for the spiral

$$\begin{aligned} u(t) &= \cos(\omega_{\text{rep}} t) e^{-\alpha k_0 t} \text{ and} \\ v(t) &= -\sin(\omega_{\text{rep}} t) e^{-\alpha k_0 t}. \end{aligned}$$

The attenuating term $e^{-\alpha k_0 t}$ results from the negative feedback path of the leftmost integrator, containing the coefficient potentiometer labeled *α*. *u*(*t*) and *v*(*t*) only describe a two-dimensional spiral, so a third function *w*(*t*) is required for a three-dimensional figure.

This function is defined as

$$w\Big(t\Big) = -1 + \int_{t=0}^{t_{\text{rep}}} \beta k_0 \, \mathrm{d}t$$

with t_{rep} denoting the operation time during one repetition cycle of the analog computer. The corresponding setup is shown in →figure 8.51. It is to be noted that the three integrators shown in →figures 8.50 and →8.51 are marked with a black bar. This denotes that these integrators are part of an integrator-group that is controlled by an external clock generator.

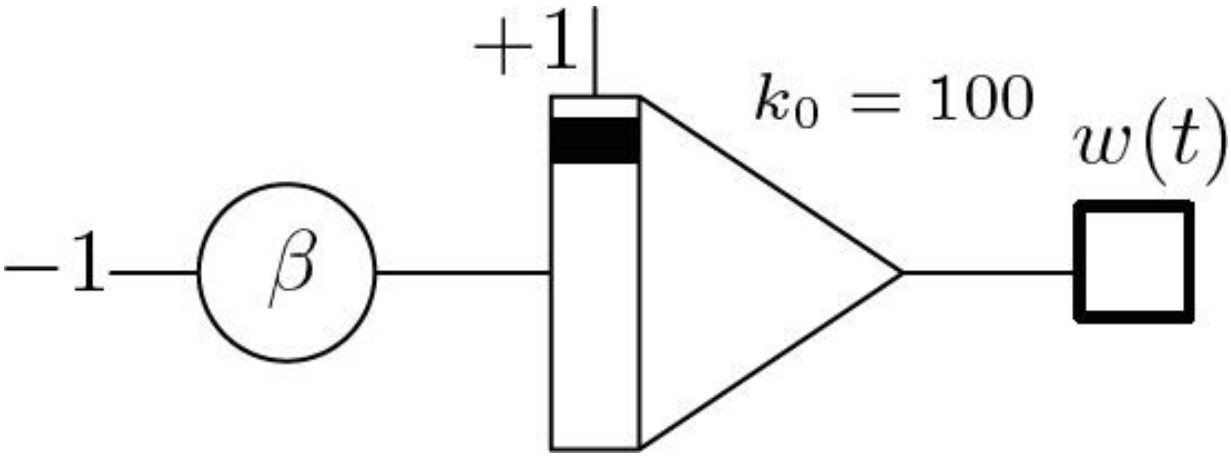


Figure 8.51. Generating $w(t)$ for the spiral

To generate a display of a slowly rotating three-dimensional spiral, two time-scales are necessary: A very short time-scale for the integrator group generating the spiral coordinates $u(t)$, $v(t)$, and $w(t)$, and a much slower time-scale for a second integrator group yielding a sine-/cosine-pair for the rotation of the figure. The three integrators denoted by the black bar have an effective $k_0 = 10^3$ due to $k_0 = 10^2$ being set at the integrators and using inputs with weight 10.

The second integrator group is run in continuous mode and is shown in →figure 8.52. It is a common sine-/cosine signal generator. Its output signals are fed to the circuit shown in →figure 8.53, which performs the actual rotation and projection of the three-dimensional figure. It implements the functions

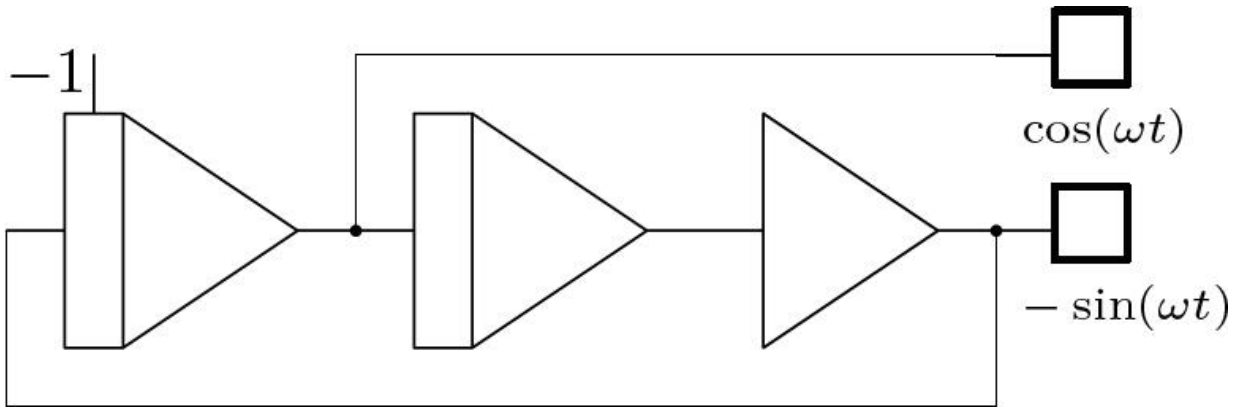


Figure 8.52. Generating a sine-/cosine signal pair for the rotation of the spiral

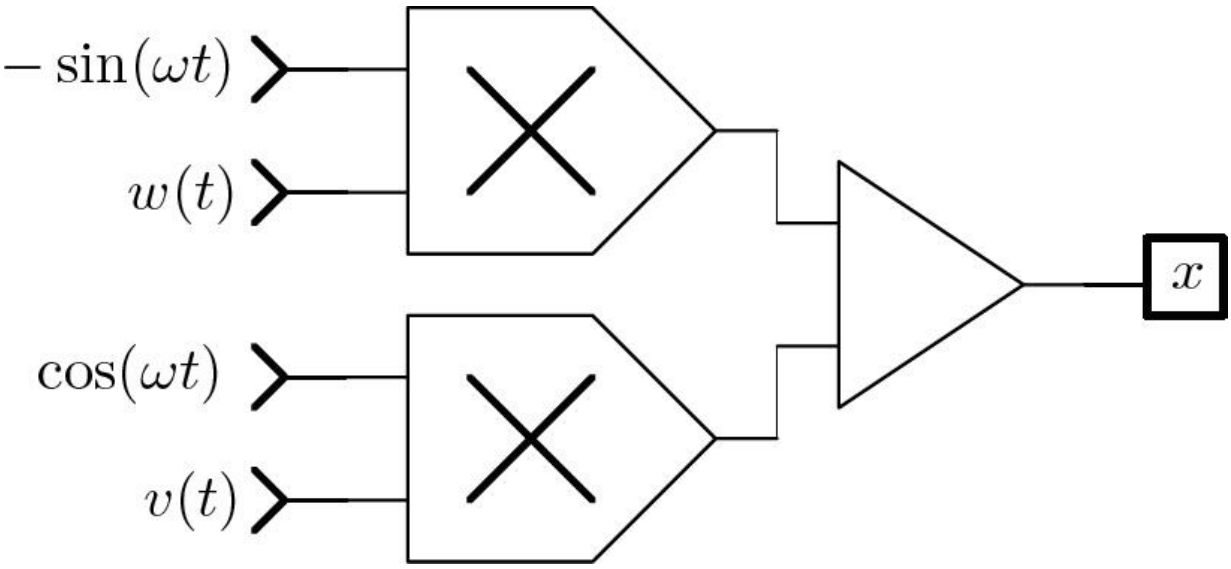


Figure 8.53. Rotation and projection

$$\begin{aligned} Y &= u(t) \quad \text{and} \\ X &= w(t)\ \sin(\omega t) - v(t)\ \cos(\omega t), \end{aligned}$$

which are then connected to the *x*- and *y*-inputs of an *x, y*-display.

Control of the first group of integrators is typically done by means of two control signals *OP* and *IC* controlling the three modes *operate, initial condition*, and *halt* of the integrators. →Figure 8.54 shows these two signals during repetitive operation of the analog computer. A short

period is required to reset the integrators to their respective initial conditions. This is followed by a longer period during which the actual computation takes place. These two modes of operation are repeated rapidly to generate a flicker-free display of the rotating spiral.

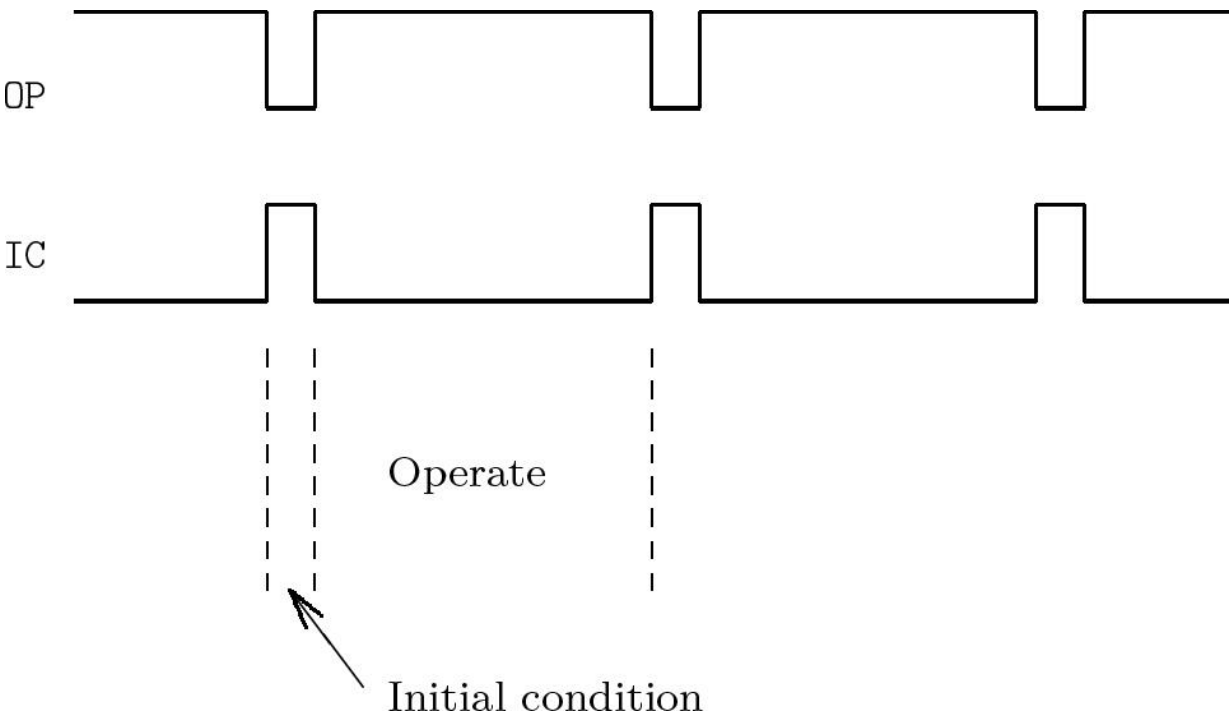


Figure 8.54. Control signals for the integrators operating in repetitive mode

→Figure 8.55 shows a suitable setup of the digital elements of an EAI 580 analog computer to generate these control signals.[46] A snapshot of the display of the rotating three-dimensional spiral is shown in →figure 8.56.

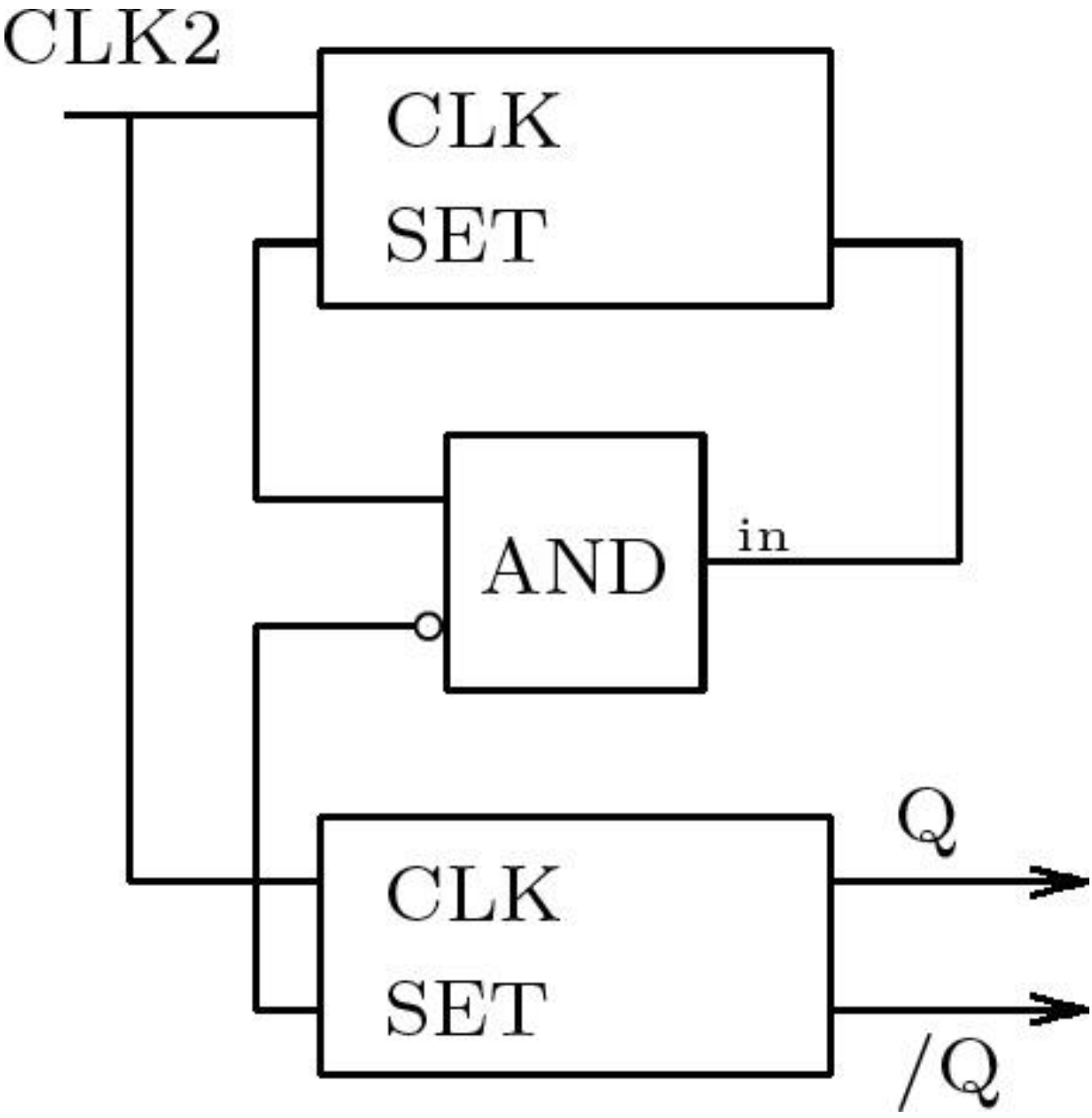


Figure 8.55. Generation of the control signals for the integrators operating in repetitive mode

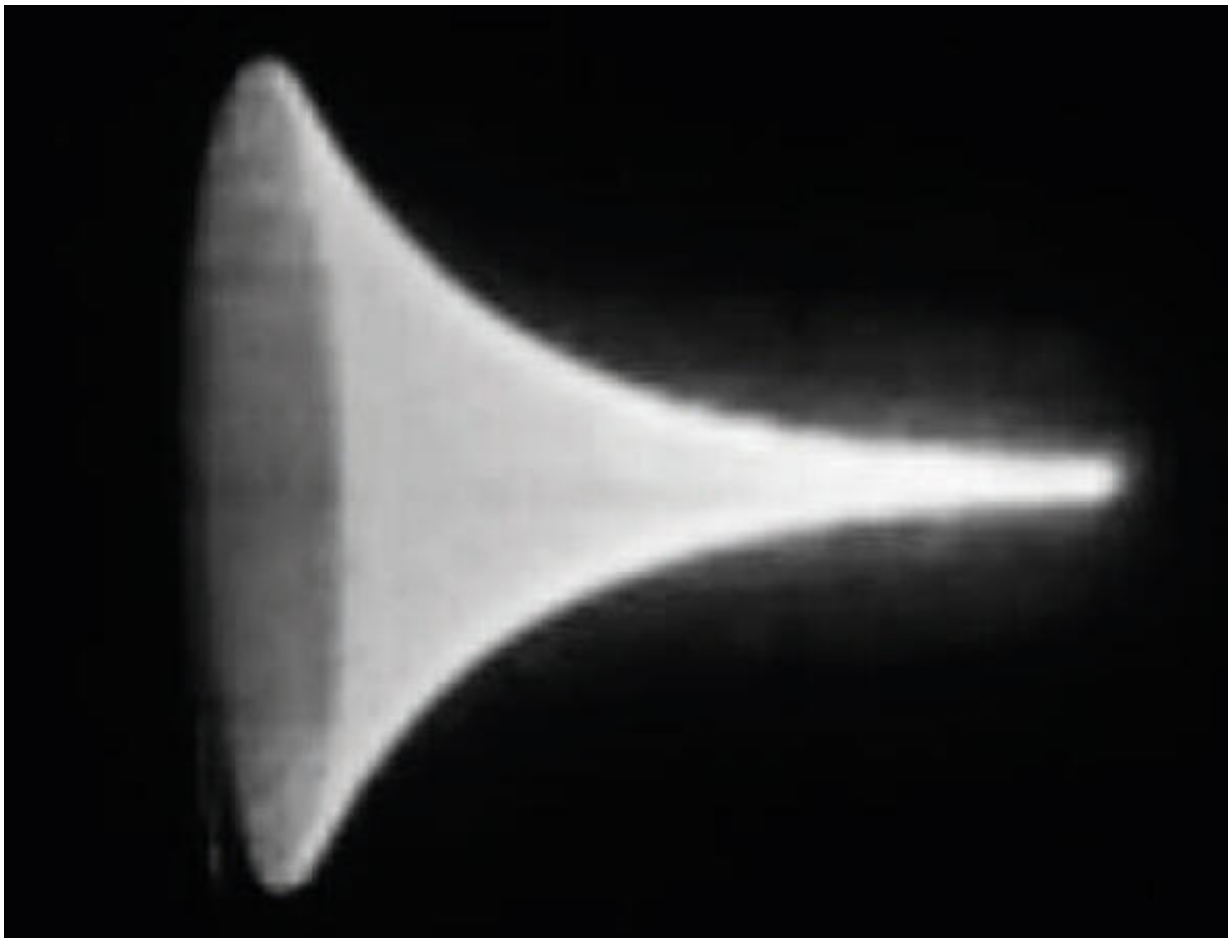


Figure 8.56. Oscilloscope screen capture of a rotating spiral

8.11 Conformal mapping

The next and most complex example covered in this chapter is based on an application note [→Telefunken/4, p. 123], [→Sydow 1964], and [→Reutter et al. 1968], and implements a *conformal mapping*, which is an angle-preserving mapping. Such mappings are described either explicitly by analytic functions of a complex variable such as

$$w = f(z) = u(x, y) + \mathrm{i}v(x, y) \quad \text{where} \quad z = x + \mathrm{i}y$$

or implicitly by the solution of suitable ordinary differential equations.[47] Implementing a conformal mapping on an analog computer is quite

straightforward. The complex variable is split into its real and imaginary parts which are handled explicitly as a two-dimensional vector.

A particularly interesting conformal mapping is

$$w(z) = (z - z_0) + \frac{\lambda^2}{z - z_0}, \tag{8.40}$$

which describes a *Joukowsky*[48] *airfoil*.[49] Applying this mapping to a unit circle yields an airfoil that can then be used to analyze the flow of a medium such as air around the structure. Implementing equation (→8.40) on an analog computer is based on the following two equations which describe the real and imaginary part separately:

$$u(x(t), y(t)) = (x(t) - x_0(t)) + \frac{\lambda^2 (x(t) - x_0(t))}{(x(t) - x_0(t))^2 + (y(t) - y_0(t))^2} \tag{8.41}$$

$$v(x(t), y(t)) = (y(t) - y_0(t)) - \frac{\lambda^2 (y(t) - y_0(t))}{(x(t) - x_0(t))^2 + (y(t) - y_0(t))^2} \tag{8.42}$$

The mechanization of equations (→8.41) and (→8.42) is shown in →figure 8.57. This function is required twice to map a unit-circle into an airfoil and to map the path of a particle flowing around a rotating cylinder into its corresponding path around this airfoil. Since the setup is quite convoluted, the circuit shown is applied to both tasks in an alternating fashion using analog switches.

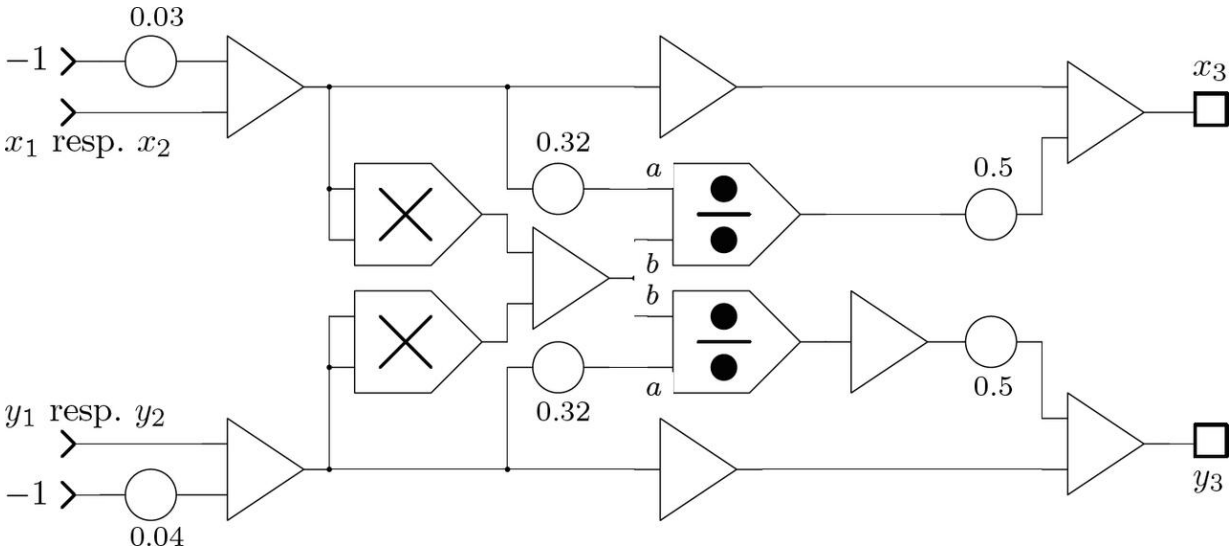


Figure 8.57. Transforming a unit circle into a Joukowsky air foil using a conformal mapping

The path of a particle flowing around a rotating cylinder is described by the complex velocity potential

$$f(z) = v_0 \left(e^{i\delta} z + \frac{e^{-i\delta} r^2}{z} \right) - i \frac{\Gamma}{2\pi} \ln(z),$$

which is realized by the circuit shown in →figure 8.58. The upper leftmost integrator generates the time varying *x*-coordinate of a particle, the path of which is determined by two parameters: the angle of attack, which is controlled by the term tan(*δ*), and the *y*-position of the particle with respect to the rotating cylinder. *y* is generated by the lower leftmost integrator, which is controlled separately from the remaining integrators of the computer setup.[50]

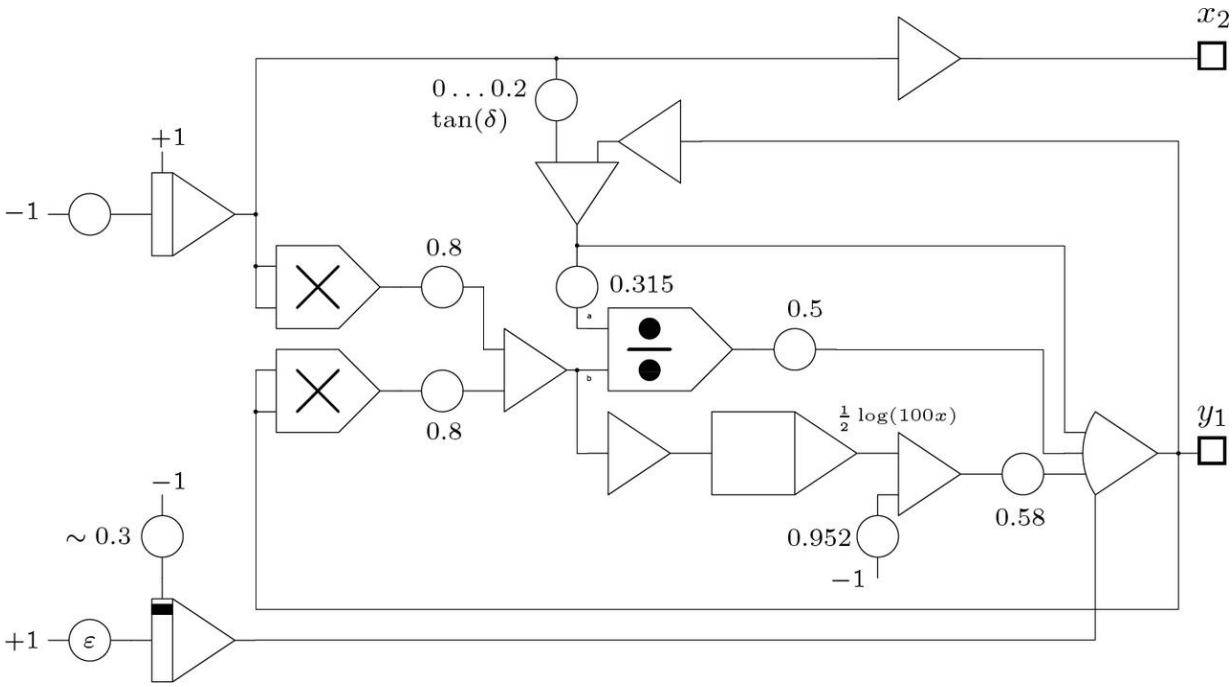


Figure 8.58. Generating the flow lines

The overall setup of this program on a Telefunken RA 770 precision analog computer is shown in →figure 8.59. →Figure 8.60 shows a single flow line around a JOUKOWKSY airfoil on top and a group of flow lines spaced equidistantly in the bottom half.

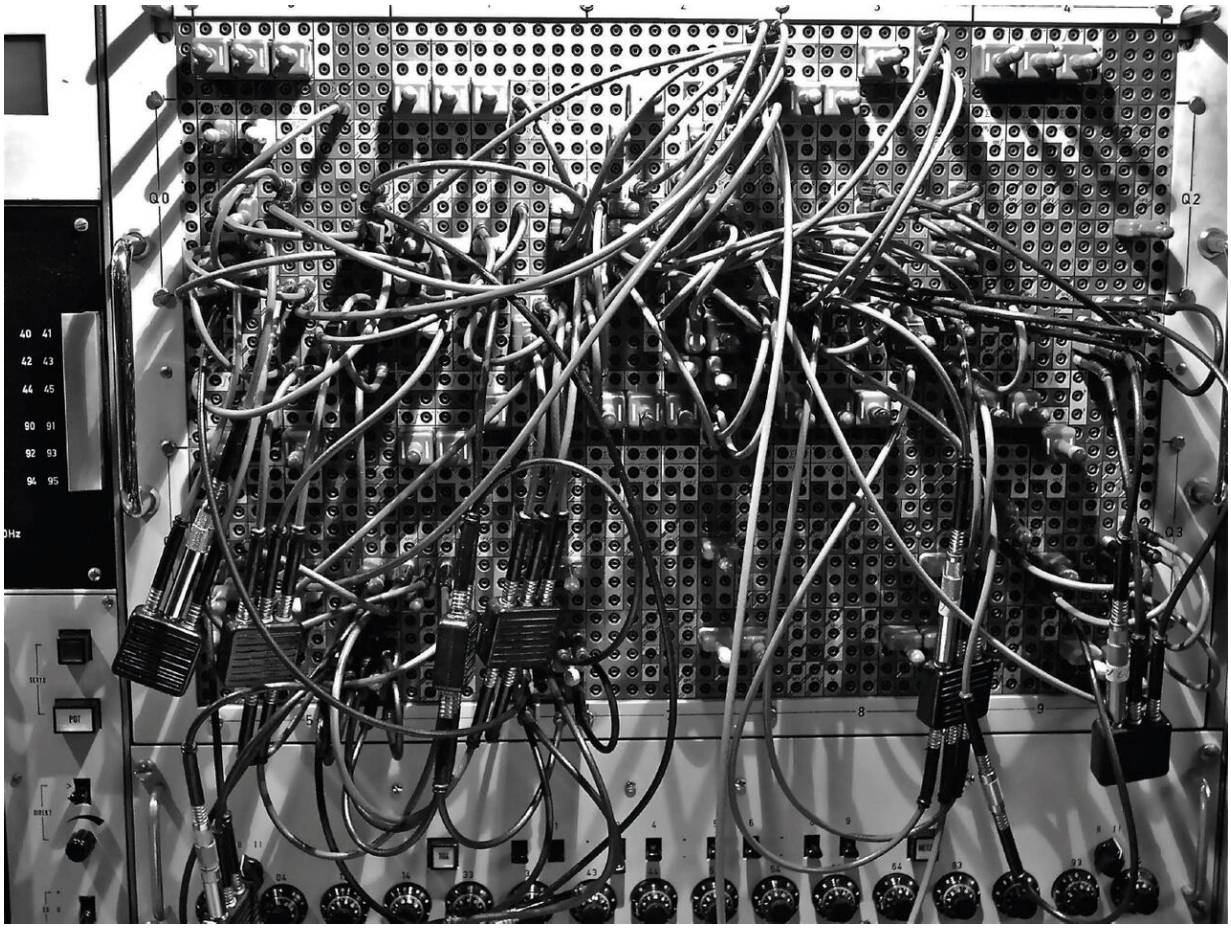


Figure 8.59. Implementation of the conformal mapping and the generation of flow lines

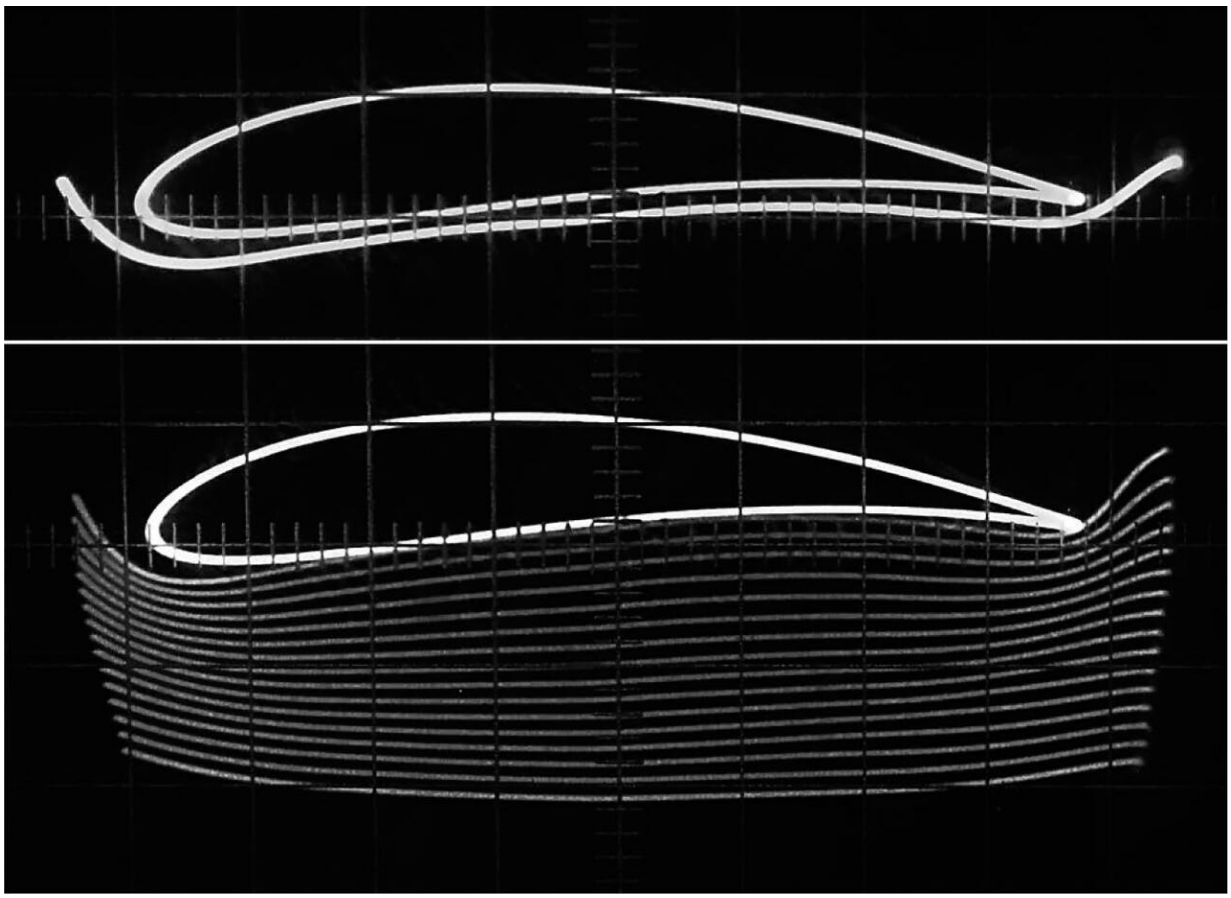


Figure 8.60. JOUKOWSKY air foil with flow lines

8.12 Hindmarsh-Rose neuron model

In the early 20th century mathematical models of the complex behavior of biological neurons became more and more realistic.[51] One of the most simple and basic such models is the *(leaky) integrate-and-fire* model described by LOUIS LAPICQUE in 1907. A much more sophisticated model was developed in the early 1960s by RICHARD FITZHUGH and J. NAGUMO and is described by

$$\begin{aligned} \dot{v} &= v - \frac{v^3}{3} - w + I_{\text{ext}} \text{and} \\ \dot{w} &= v + a - bw. \end{aligned}$$

This model is still quite simple and basically equivalent to the VAN DER POL equation

$$\ddot{y} + \mu(y^2 - 1)\dot{y} + y = 0,$$

which was devised by BALTHASAR VAN DER POL in 1920 as a result of his pioneering work on vacuum tubes and oscillator circuits. Accordingly, the FITZHUGH-NAGUMO model is basically a relaxation oscillator[52] controlled by an external stimulus I_{ext}.

A more interesting model is was developed by HINDMARSH and ROSE.[53] It consists of three coupled differential equations

$$\dot{x} = -ax^3 + bx^2 + y - z + I_{\text{ext}} \tag{8.43}$$

$$\dot{y} = dx^2 + c - y \tag{8.44}$$

$$\dot{z} = r\big(s(x - x_r) - z\big) \tag{8.45}$$

with the parameters a = 1, b = 3, c = 1, d = 5, r = 10^{-3}, s = 4, $x_r = -\frac{8}{5}$, and initial conditions of value 2 for all of the three required integrators.

Scaling this system is a great exercise and is described in detail below. A quick numeric integration yields the solution shown in →figure 8.61. Obviously, the system must be scaled in order to be implemented on an analog computer since the variables *x, y*, and *z* all exceed the standard machine unit interval of [−1, 1] by far.[54]

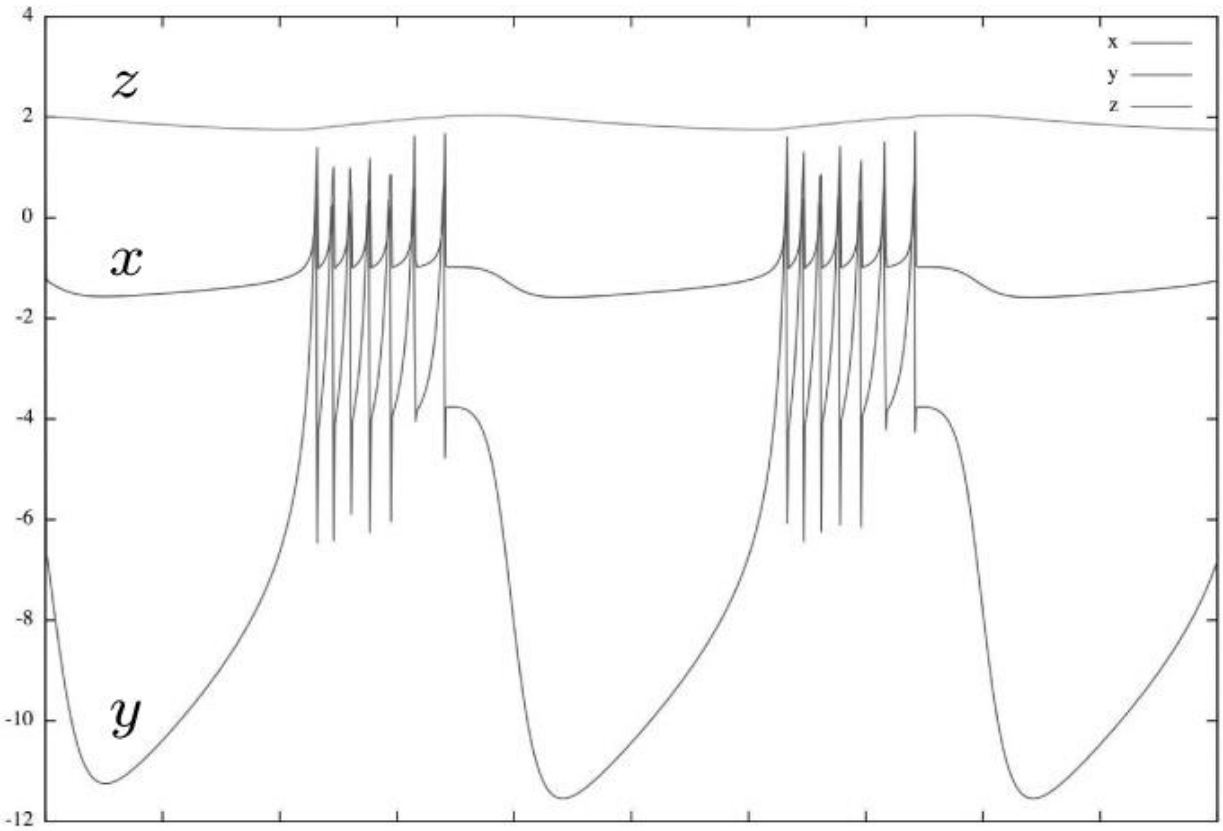


Figure 8.61. Unscaled HINDMARSH-Rose model

First, *x* is scaled by $\lambda_x = \frac{1}{2}$:[55]

$$\begin{aligned}
\dot{x} &= \frac{1}{2}(-8ax^3 + 4bx^2 + y - z + I_{\text{ext}}) \\
&= -4ax^3 + 2bx^2 + \frac{y}{2} - \frac{z}{2} + I_{\text{ext}} \\
\dot{y} &= -4dx^2 + c - y \\
\dot{z} &= r(s(2x - x_r) - z)
\end{aligned}$$

I_{ext} has also been scaled by λ_x and is now conveniently 1. The behavior of the system after this first scaling step is shown in →figure 8.62. *x* is now well within [−1, 1].

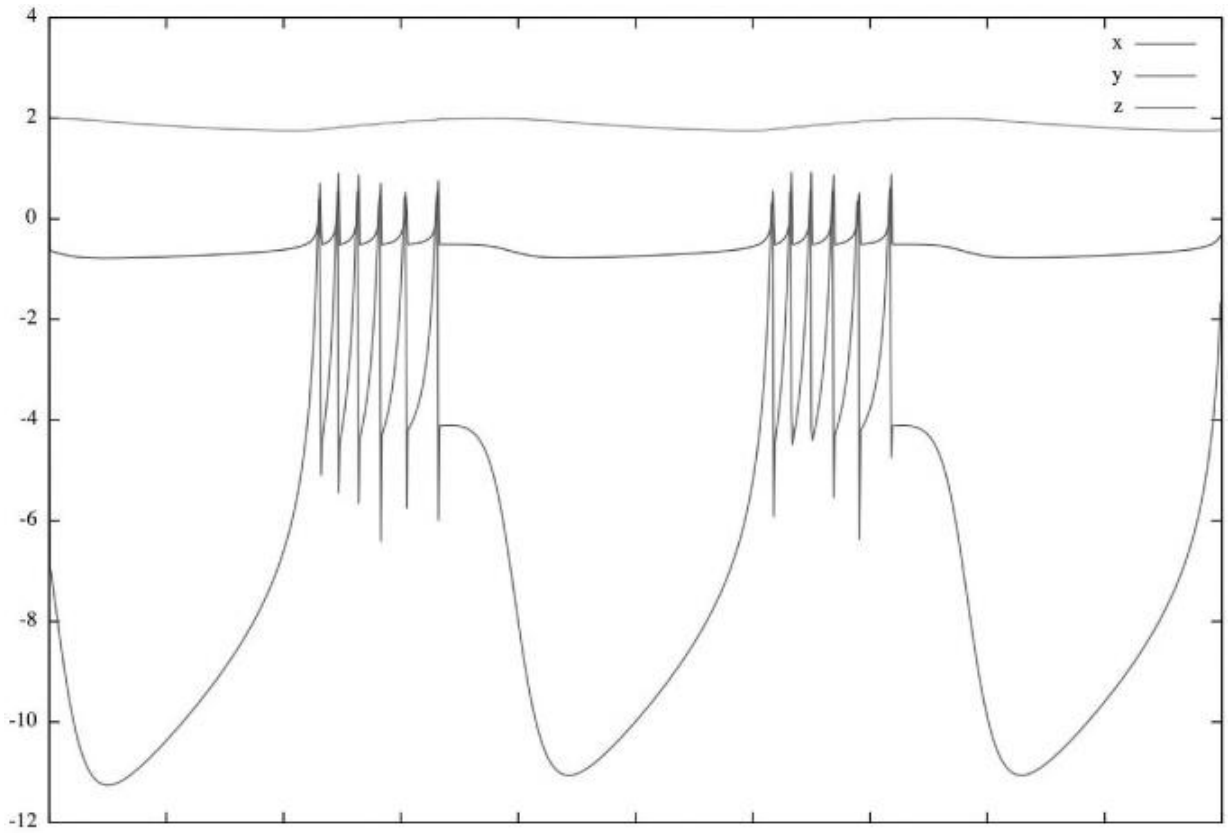


Figure 8.62. HINDMARSH-ROSE model after the first scaling step

z can also be scaled with a factor $\lambda_z = \frac{1}{2}$ yielding the following system of equations:

$$\begin{aligned}
\dot{x} &= -4ax^3 + 2bx^2 + \frac{y}{2} - z + I_{\text{ext}} \\
\dot{y} &= -4dx^2 + c - y \\
\dot{z} &= \frac{1}{2}(r\left(s(2x - x_r) - 2z\right)) \\
&= r\left(s\left(x - \frac{x_r}{2}\right) - z\right)
\end{aligned}$$

The output of this system is shown in →figure 8.63. All that is now left to do is scaling *y* with a “nice” value such as $\lambda_y = \frac{1}{15}$:

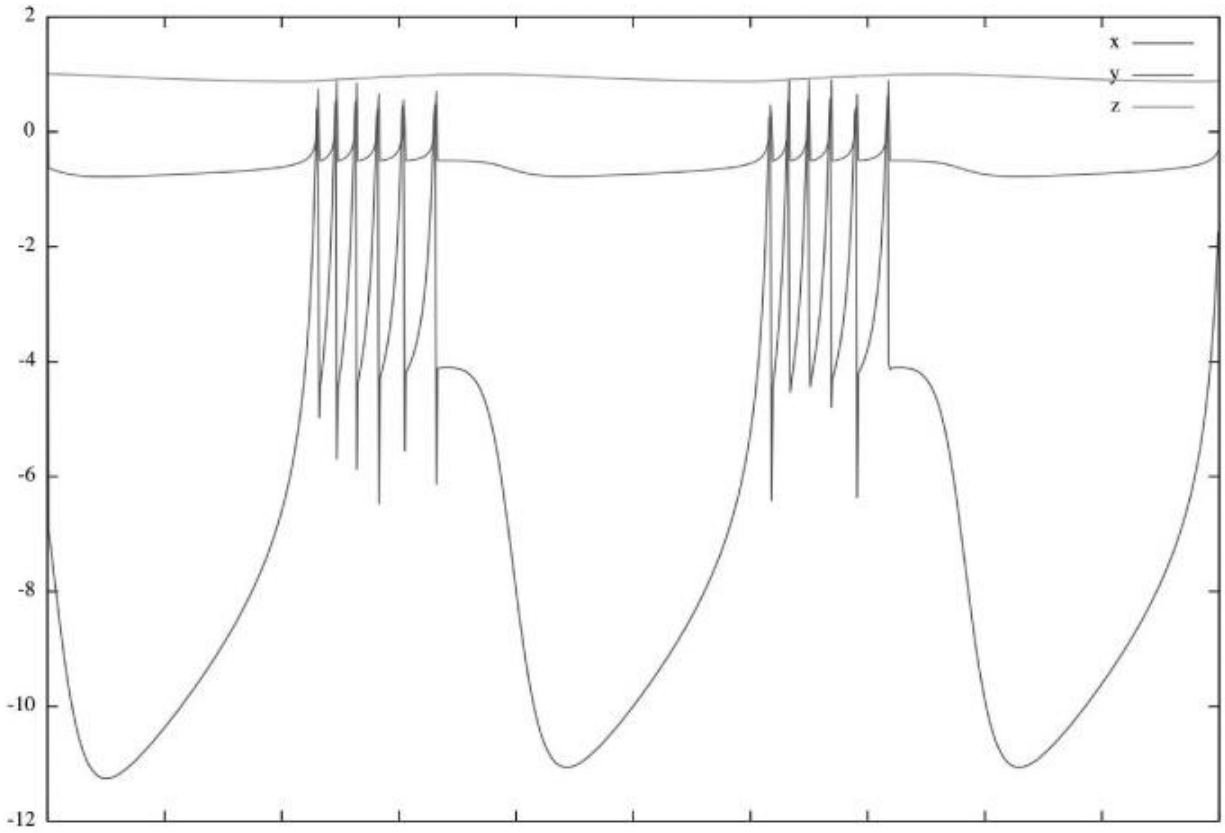


Figure 8.63. HINDMARSH-ROSE model after the second scaling step

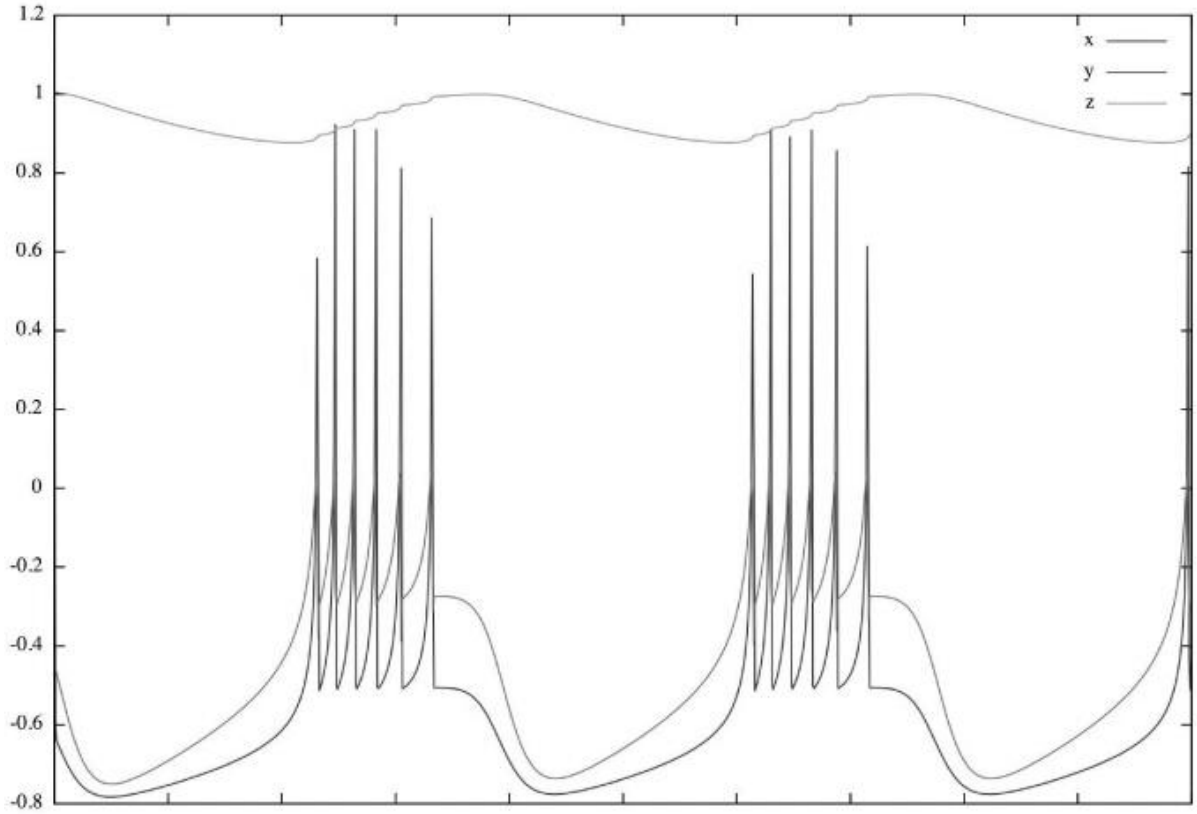


Figure 8.64. Final scaled system

$$\begin{aligned}
\dot{x} &= -4ax^3 + 2bx^2 + 7.5y - z + I_{\text{ext}} \\
\dot{y} &= \frac{1}{15}(-4dx^2 + c - 15y) \\
&= -0.2667dx^2 + \frac{c}{15} - y \\
\dot{z} &= r\left(s\left(x - \frac{x_r}{2}\right) - z\right)
\end{aligned}$$

The resulting initial condition of the *y* integrator is now so small that it can be safely ignored and set to 0.

The scaled parameters are a = 1, which can be omitted, $\frac{b}{\lambda_x} = 6$, $\lambda_y c \approx 0.0667$, and $0.2667d \approx 1.336$. Multiplying out the equation for $\dot{z}$ yields

$$\dot{z} = r\left(s\left(x - \tfrac{x_r}{2}\right) - z\right) = rsx - \tfrac{rsx_r}{2} - rz.$$

Taking all constants into account results in

$$\dot{z} = 0.004x + 0.0032 - \tfrac{z}{1000}$$

yielding the following system of scaled equations:

$$\begin{aligned}
\dot{x} &= -4x^3 + 6x^2 + 7.5y - z + I_{\text{ext}} \\
\dot{y} &= -1.336x^2 + 0.0667 - y \\
\dot{z} &= 0.004x + 0.0032 - \frac{z}{1000}
\end{aligned}$$

Using different time-scale factors of the integrators, the very small coefficients in the last equation can be scaled up to more convenient values. Without loss of generality, let $k_{0x} = k_{0y} = 10^4$ and $k_{0z} = 10^2$,[56] yielding the final set of equations:

$$\begin{aligned}
\dot{x} &= -4x^3 + 6x^2 + 7.5y - z + I_{\text{ext}} & k_{0_x} &= 10^4, \text{IC} = +1 \\
\dot{y} &= -1.336x^2 + 0.0667 - y & k_{0_y} &= 10^4, \text{IC} = -1 \\
\dot{z} &= 0.4x + 0.32 - 0.1z & k_{0_z} &= 10^2, \text{IC} = +1
\end{aligned}$$

The resulting analog computer program is shown in →figure 8.65. Using several of these spiking and bursting neurons, more complex circuits such as flip-flops, oscillators, etc., can be set up.[57]

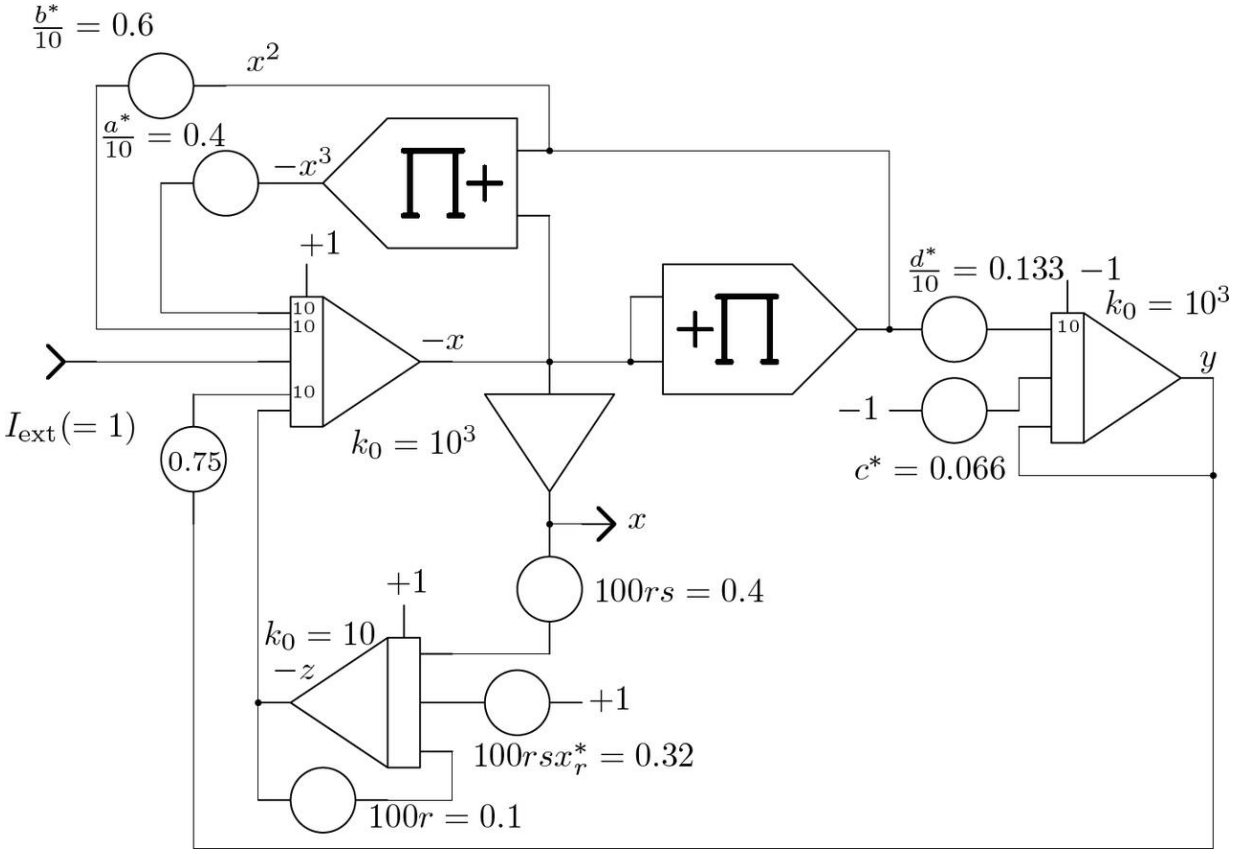


Figure 8.65. Scaled analog computer setup for the HINDMARSH-ROSE model

9 Digital control units

Only in the most simple applications a simple mode control switch selecting the mode of operation manually (IC, OP, HALT) is sufficient. Typically, at least some simple means for controlling repetitive operation,[1] i. e., rapidly cycling the analog computer between IC and OP states, is required. This allows one to obtain a more or less flicker free display on an oscilloscope while observing the effects of manually varied parameters on the solution of a problem. THE ANALOG THING is a good example for such a simple system.[2] In some cases a variant of this, *single run*, switching the computer to IC, OP, and HALT, is required as well. This can be typically achieved by slightly modifying the repetitive mode control logic.

However, there is a plethora of applications which benefit from or even require more sophisticated control logic. A typical application might involve two sub-circuits, one preparing initial conditions for the other, requiring some form of *iterative mode* control where two groups of integrators are cycled through IC, OP, HALT in an alternating pattern.

Accordingly, most medium to large scale analog computers feature rather versatile control systems like the *DBG* (*Digital-Bediengerät*) used in Telefunken's large scale analog computers RA 800H and RA 770,[3] shown in →figure 9.1. At the heart of the DBG is a quartz oscillator serving as a stable timing reference from which all control signals are derived. On the far right of the front panel are two columns of switches for standard manual mode control, including additional modes of operation such as *static checking* and *dynamic checking* for testing a program setup as well as the dynamic behavior of the integrators of the

machine. In addition to this the DBG also supports repetitive operation, single run, and iterative run.

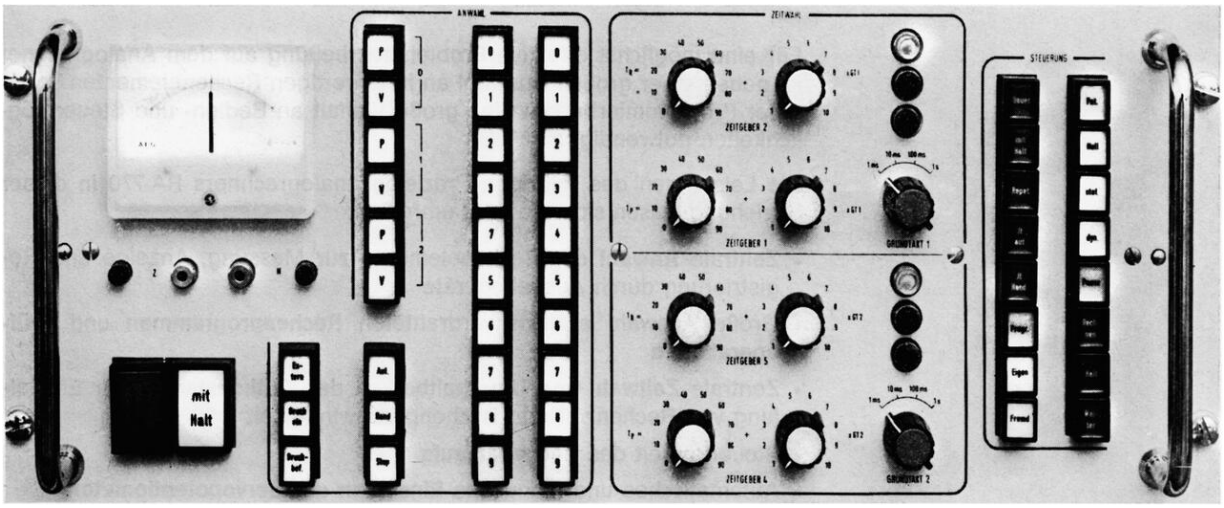


Figure 9.1. Digital control system used in Telefunken's RA 800H and RA 770 analog and hybrid computers (see [→Telefunken/8, p. 20])

To the left of these manual control switches are two groups of timing control switches. In the case of single run or repetitive operation only the upper group is employed. Using two sets of switches each, the times for IC and OP can be set with millisecond resolution. In the case of iterative operation where two groups of computing elements work in an alternating fashion both of these time control groups are used, each controlling one group of integrators.

The left half of the control unit contains three columns of switches for selecting individual computing elements for readout.[4] The output of a selected element is connected to a central digital voltmeter which can also control an external printer for tabulating values either continuously or after each OP cycle when running in repetitive mode. On the far left is a central

voltmeter for balancing the operational amplifiers in the system as well as two large buttons controlling the behavior of the system in case of an overload. Overloads can either be ignored or trigger the halt of the current OP cycle.

As complicated as such control units are, they only allow for a certain set of operating modes and leave no room for individual control schemes which would require some form of configurable control logic. In cases where a full hybrid computer setup was not feasible, many systems offered simple, yet more flexible digital control units such as the one shown in →figure 9.2 used in EAI 231R-V analog computers. Apart from several timers controlling IC and OP times, it features a dedicated small patch panel, the *mode logic patch panel* (*MLPP*), with associated logic elements such as AND gates, etc. It allows for individual control of a number of integrators, simplifying the setup of integrators as *track-and-store* devices,[5] controlling chains of such track-and-store integrators creating time delay circuits, etc.[6]

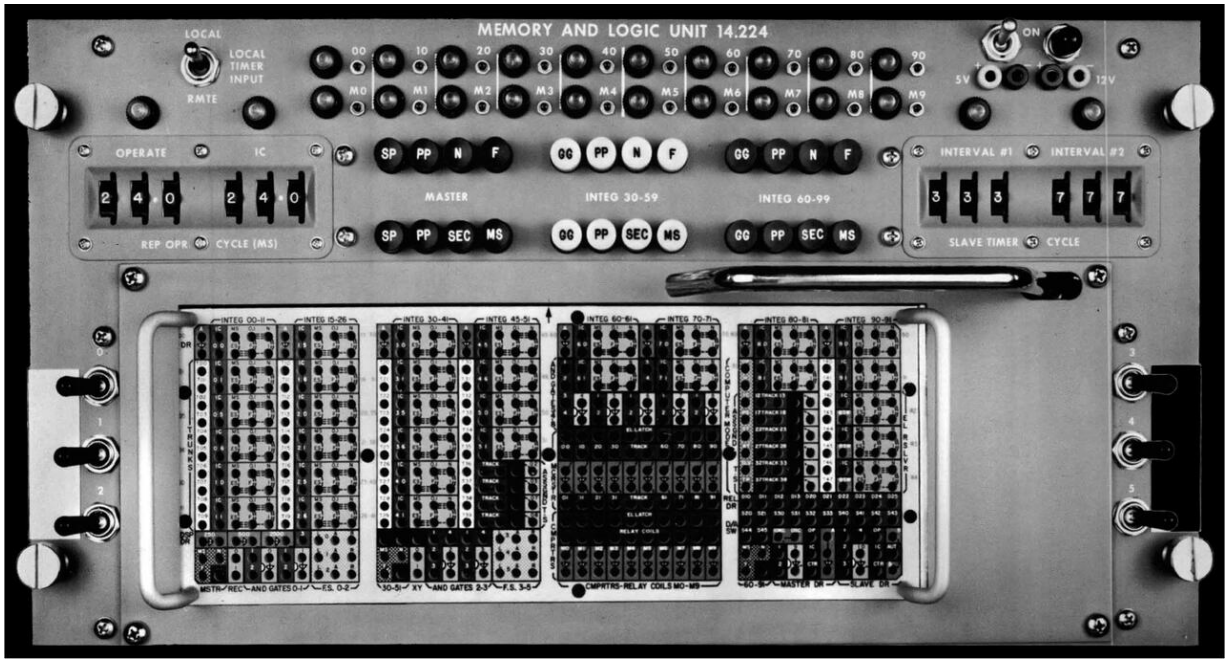


Figure 9.2. Analog memory and logic system control unit as used in the EAI 231R-V analog computer (see [→EAI 1964/1, p. 12])

When Telefunken unveiled the first commercial transistorized analog computer, the RAT 700, in 1959, it was such a small system that it did not even feature a removable patch panel, much to the annoyance of many customers. Accordingly, an upgrade kit, consisting of more than 400 spring loaded connectors with 4 mm banana plugs, some rather complex mounting hardware, and removable patch panels was offered. Not much later customers demanded some upgrade option offering more control over the analog computer's operation. Telefunken responded by offering the *DEX 100* and *DEX 102* (*Digital-Experimentiergerät*), which were marketed as a digital experimental kit with the option of being attached to a Telefunken tabletop analog computer. →Figure 9.3 shows a DEX 102, which looks quite like an analog computer at first glance. It is dominated by a (removable) patch panel and contains a

number of flip-flops, amplifiers,[7] logic inverters, *monostable multivibrators*,[8] logic gates (consisting of passive diode networks), a clock generator, several switches, and relay drivers.[9] As a control unit, it is incredibly basic and leaves every detail of a control setup to the programmer.

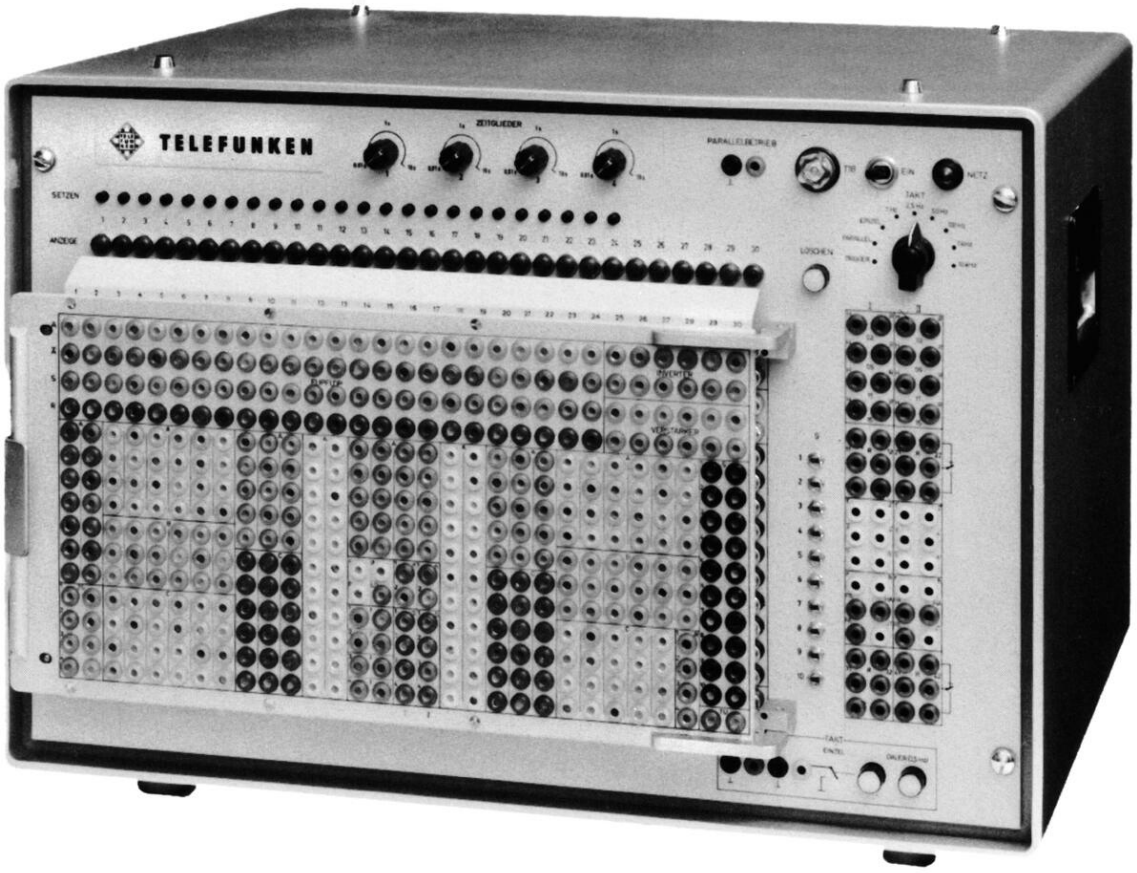


Figure 9.3. Digital extension (“Digitalzusatz”) DEX 102 for Telefunken table top analog computers (see [→Telefunken/9, p. 15])

A wonderful example application using a DEX 100 for a rather complex control scheme is described in [→Telefunken/5], which

deals with a simulation of the historic RUTHERFORD[10] experiment – alpha particles being scattered on a thin gold foil – on an analog computer. Basically, an alpha particle approaching an atomic nucleus is subject to a force

$$F = \frac{1}{4\pi\varepsilon_0}\frac{q_\alpha q_k}{r^2},$$

where q_α and q_k denote the charges of the alpha particle and that of the target nucleus. Simplifying this setup by restricting it to two dimensions yields the two components

$$\ddot{x} = a - \frac{x}{r^3} \quad \text{and}$$
$$\ddot{y} = a - \frac{y}{r^3}$$

with a suitably chosen constant of proportionality *a*. Integrating twice over $\ddot{x}$ and $\ddot{y}$ respectively yields a coordinate pair (*x*, *y*) describing the position of the alpha particle. The resulting circuit is shown in →figure 9.4.[11] It should be noted that instead of an explicit division by r^3 a function generator yielding $1/r^3$ is used.

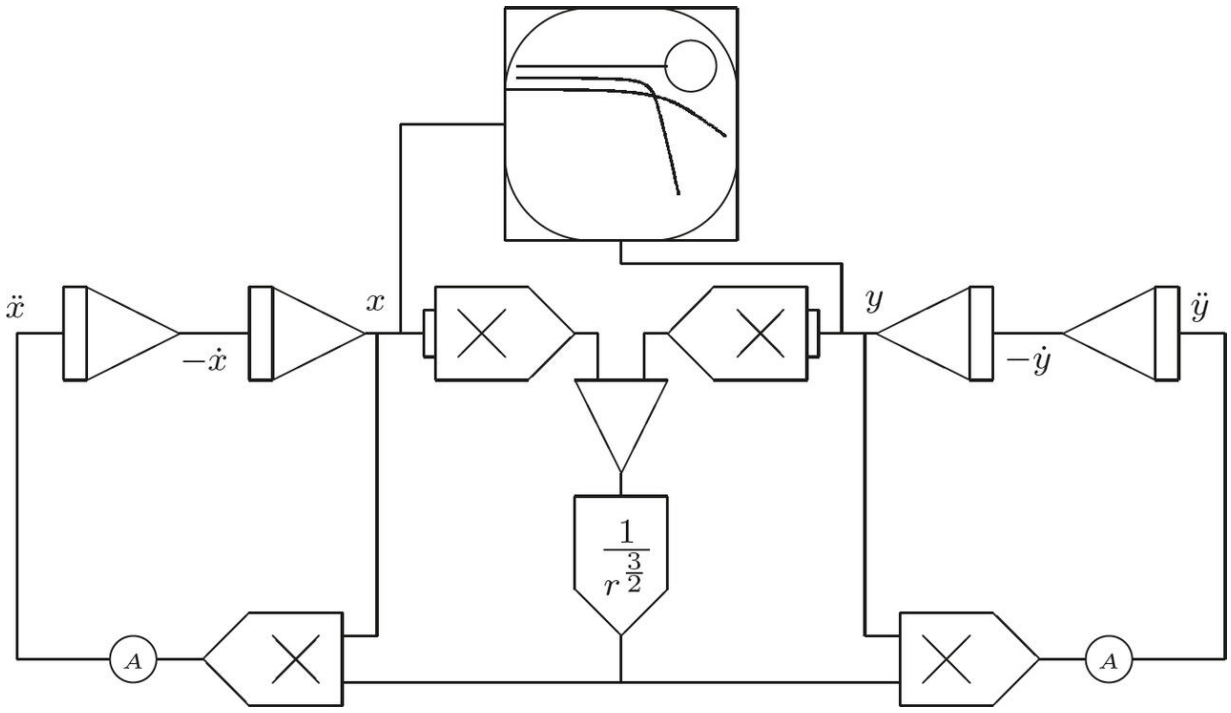


Figure 9.4. Basic circuit for simulating an alpha particle trajectory (see [→Telefunken/5, p. 2])

To generate a picture showing various trajectories on an oscilloscope some external control is required. The central parameter of such a simulation is the $y(0)$ input of the second integrator from the right yielding y, i. e., the initial vertical position of the α particle. →Figure 9.5 shows the detailed analog computer setup. It consists of three parts: On the lower right is a simple sine/cosine generator used to display a circle on the oscilloscope denoting the atom's nucleus. Since the overall mode of operation of the setup is repetitive, there is no need for an explicit amplitude stabilization as the integrators will be reset prior to each consecutive run. The upper two thirds of the schematic contain the program for the actual simulation of an α particle's path hurtling towards the nucleus at a given initial vertical position y_0. This value, in turn, is generated by the two-integrator-circuit shown on the bottom right. These generate a

step function under mode control by a DEX 100, which is used as initial condition for the integrator yielding $+y$ shown in the lower left. The integrator marked with a black rectangle is controlled independently from the other integrators. This creates a family of curves starting at different heights, which is adumbrated in the oscilloscope screen depicted →figure 9.4.

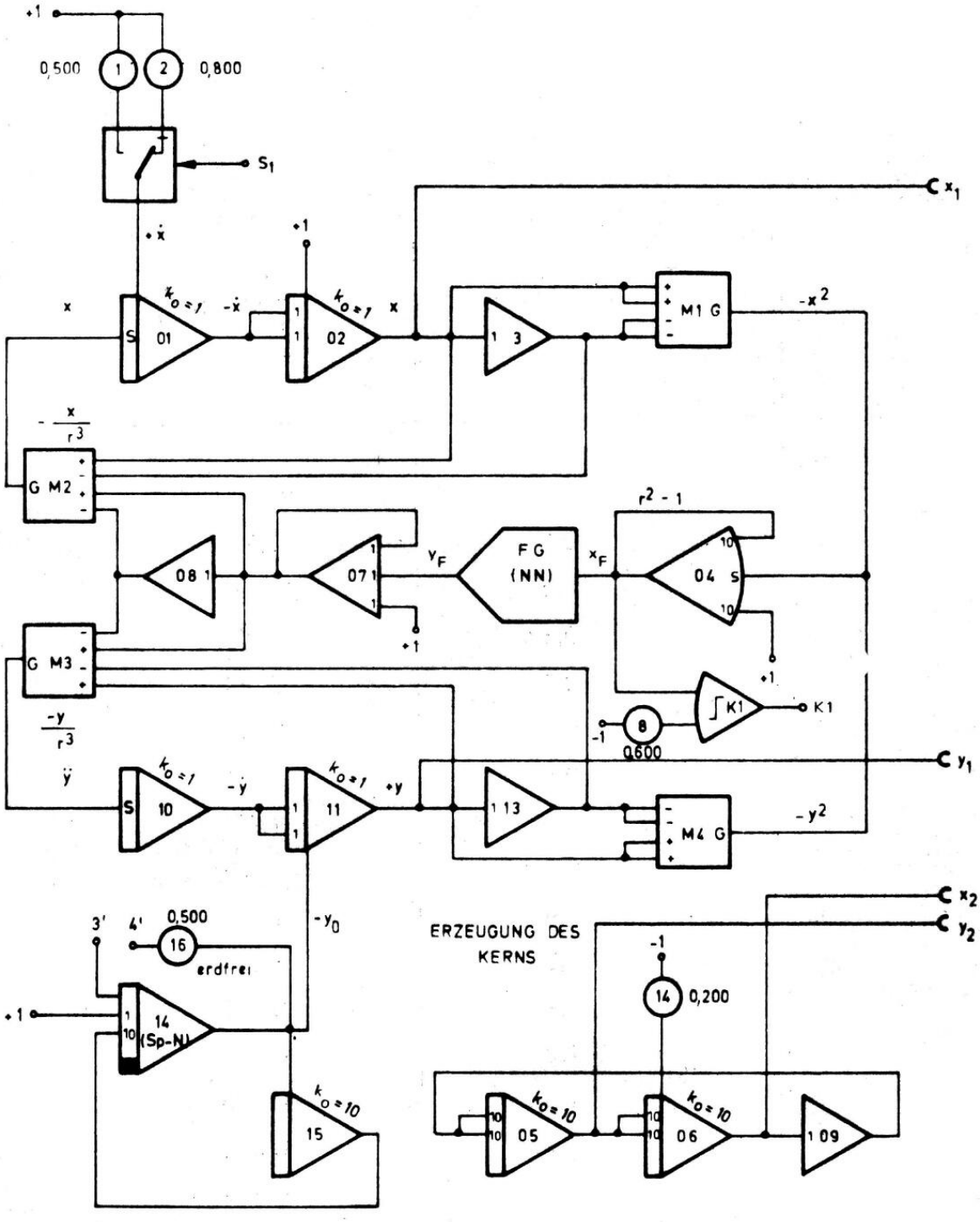


Figure 9.5. Analog portion of the α-particle simulation

The corresponding DEX 100 setup is shown in →figure 9.6. It controls the overall system operation by means of the two relays R12 and R22 as well as the generation of the step function by the cyclically operated relay R32. One computer run takes about five seconds to complete (the integrators performing the actual simulation are set to $k_0 = 1$) so that a screenshot requires a very long film exposure time.[12]

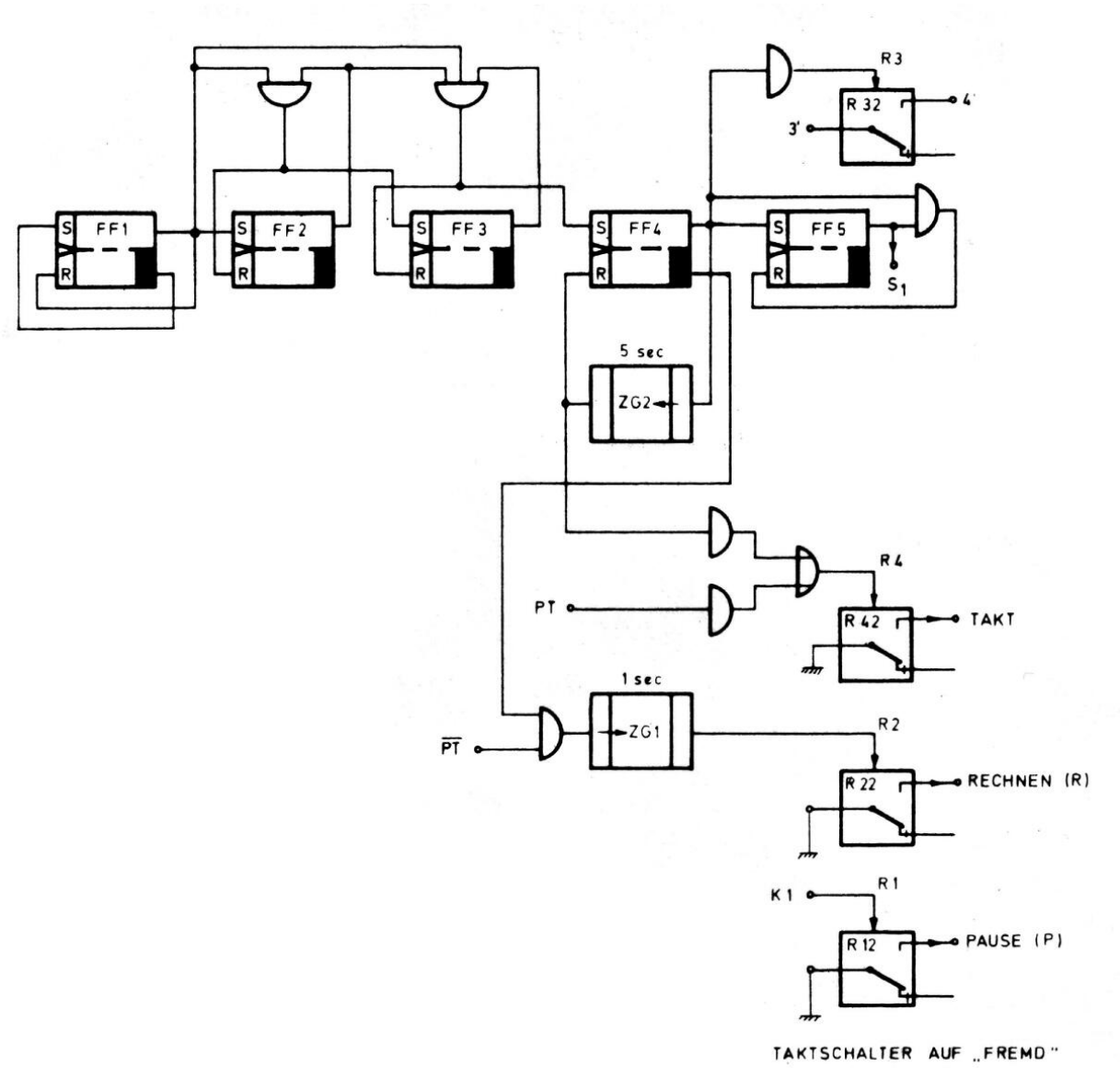


Figure 9.6. Digital portion of the α-particle simulation

The overall setup is shown in →figure 9.7. The Telefunken *RAT 741* analog computer is on the right with a DEX 100 on the left. On top of the DEX 100 sits a Telefunken *OMS 800* dual channel *x/y*-oscilloscope.[13] The RAT 741 and the controlling DEX 100 are linked by a control cable on the rear.[14]

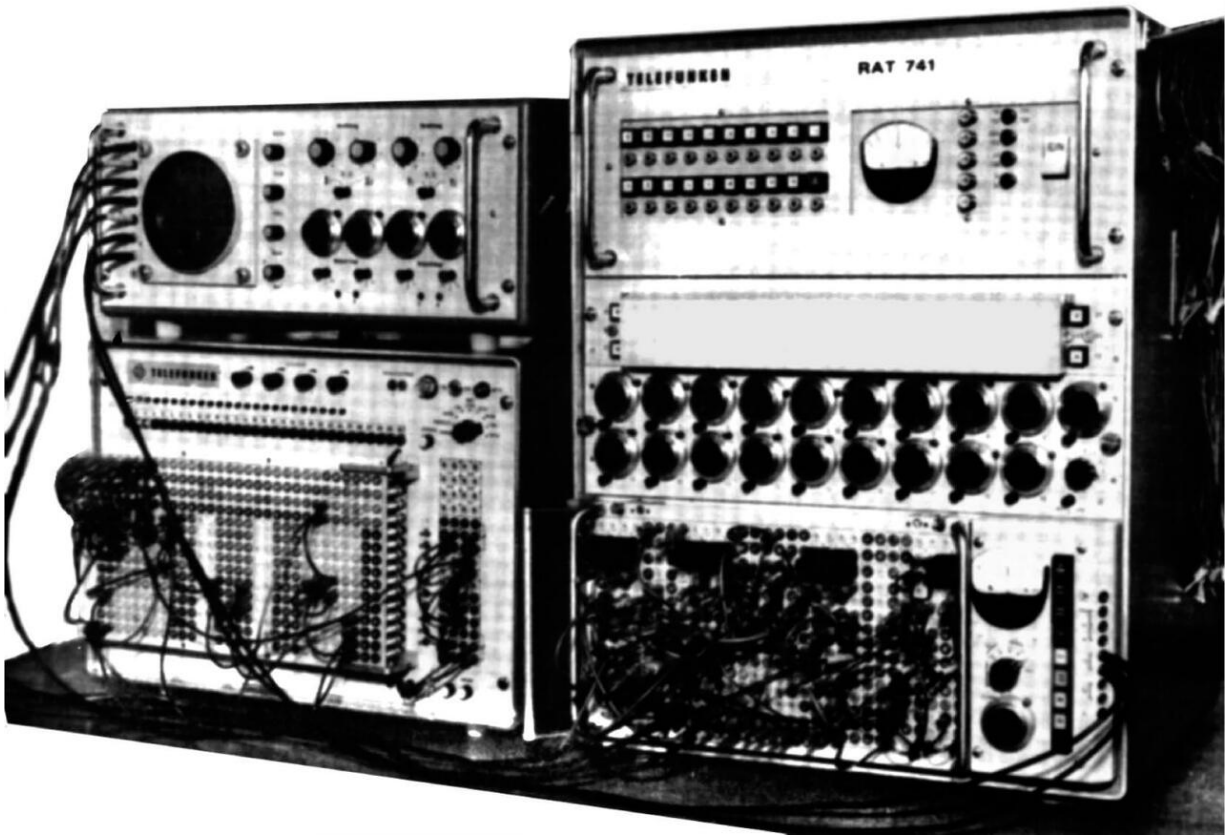


Figure 9.7. Setup of the particle trajectory simulation (see [→Telefunken/5, p. 1])

Programming something like a DEX 100 is even more tedious than programming an analog computer since it only contains the most basic logic elements. Even simple tasks like setting up a counter require a major effort. However, it quickly became clear that the flexibility of such a digital control unit allowed

customers to tackle complex problems without having to resort to a much more expensive hybrid computer setup. Accordingly, the concept was later extended to the sophisticated Telefunken *DEX 802* used in Telefunken's large analog computers RA 800H and the RA 770. →Figure 9.8 shows a DEX 802 as part of a RA 770 system. Like the small scale DEX 100 it features a removable patch panel on top of which a card cage holding up to 24 logic cards as well as two mandatory cards for comparator control and a stepping switch is located. Each of these logic cards implements a certain type of function ranging from simple AND/OR/NOR gates to counters, decoders, monostable multivibrators, etc. Thanks to the fully modular construction of the DEX 802, its functionality can be tailored to meet customer requirements. The DEX 802 can be used even in a hybrid computer configuration allowing for faster reaction to events within a simulation setup than possible by the attached digital computer which is limited by its interrupt response time.

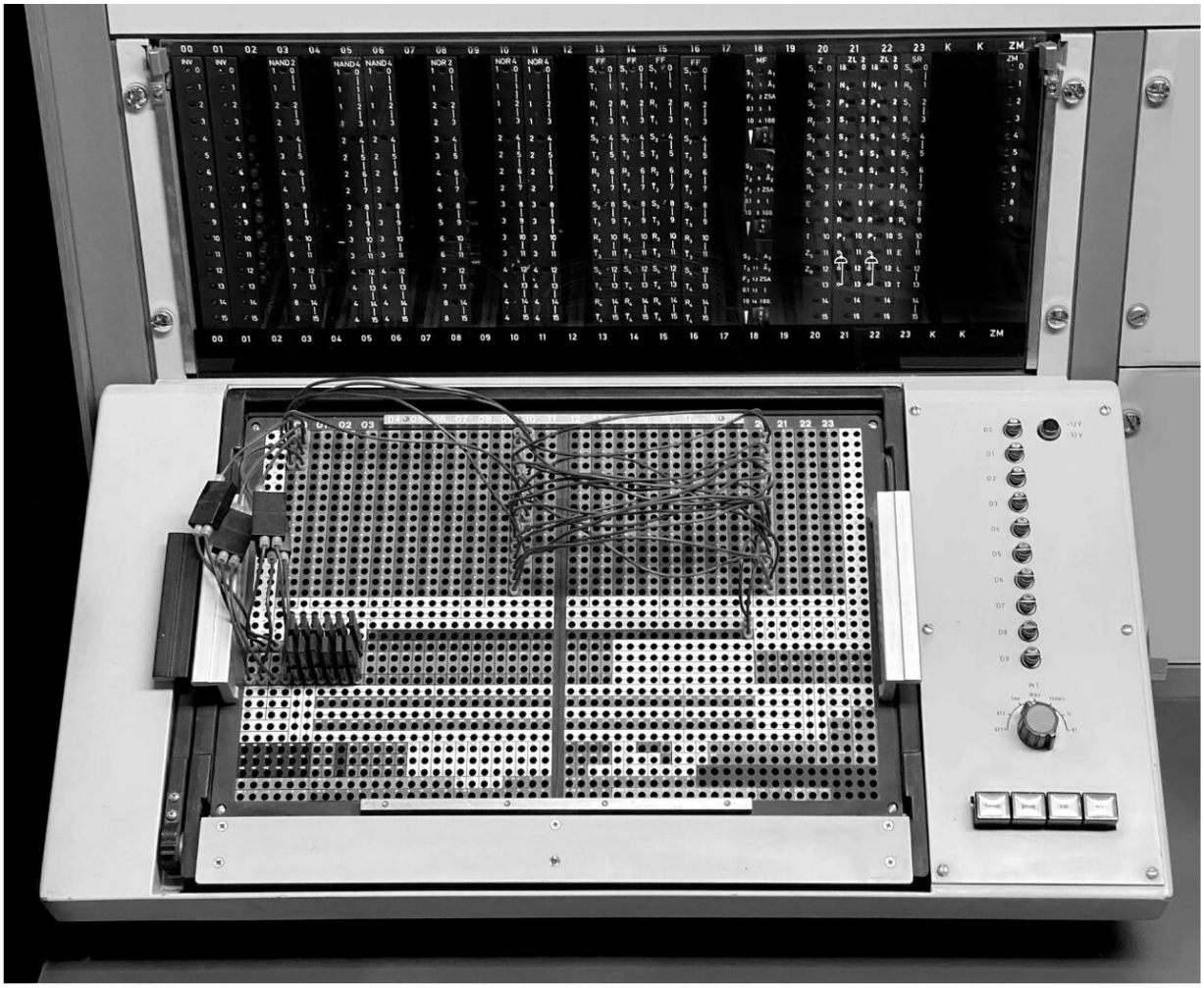


Figure 9.8. Large digital extension DEX 802 for Telefunken's RA 800H and RA 770 analog computers

Probably the most advanced implementation of such a digital control unit is the *Parallel Logic Unit* (*PLU*) being a central part of the *EAI SIMSTAR* system.[15] Being part of a hybrid computer setup, this unit is fully reconfigurable under control of the attached digital computer and features 160 input signals like comparator outputs, etc., which can be fed to either a sequential or combinatory logic circuit yielding up to 320 digital output signals. The system implements about 4000 four-input logic gates with cycle times of ≤ 2μs.[16]

Apart from controlling complex computer setups, such digital control units can also be used to implement pseudo-random generators by implementing shift registers with XOR feedback paths, etc. An interesting example for this application is described in [→Wise et al. 1967].

10 Hybrid computers

In the mid-1950s it became clear that analog computers had some drawbacks that could not be easily overcome by analog electronic means. One of the main challenges was the cumbersome process of setting up function generators and the generation of functions of more than one variable – a task often required in aerospace applications or whenever functions are defined by measurement data and not analytically. The classic stored-program digital computer conversely excels at tasks like these. Thus, the idea of coupling analog computers with digital computers, forming *hybrid computers*, was born.

10.1 Systems

An early account of this rationale is given in [→ Waltman 2000, p. 69] describing the application of analog computers for the simulation of the *X-15* aircraft:

> "*The thought of building up another set of function generators like those already in use was probably considered, but not by me or any other X-15 simulation programmers. We had had enough of those fuses and dinky pots. The idea of using a digital computer to do this job was unanimously and immediately accepted. No discussion was needed. We were going hybrid*."

To couple an analog computer with a digital computer, analog-digital-converters (ADCs) and digital-analog-converters (DACs) are required. These devices translate analog voltages into digital signals and vice versa. In addition to that, typical hybrid computers allow control of the integrators by the digital computer. They also typically feature digital interface lines, which can be used to trigger interrupts on the digital system by

comparators on the analog side and to control electronic or relay switches on the analog system from the digital computer.
→Figure 10.1 shows the structure of an EAI 690 hybrid computer system.

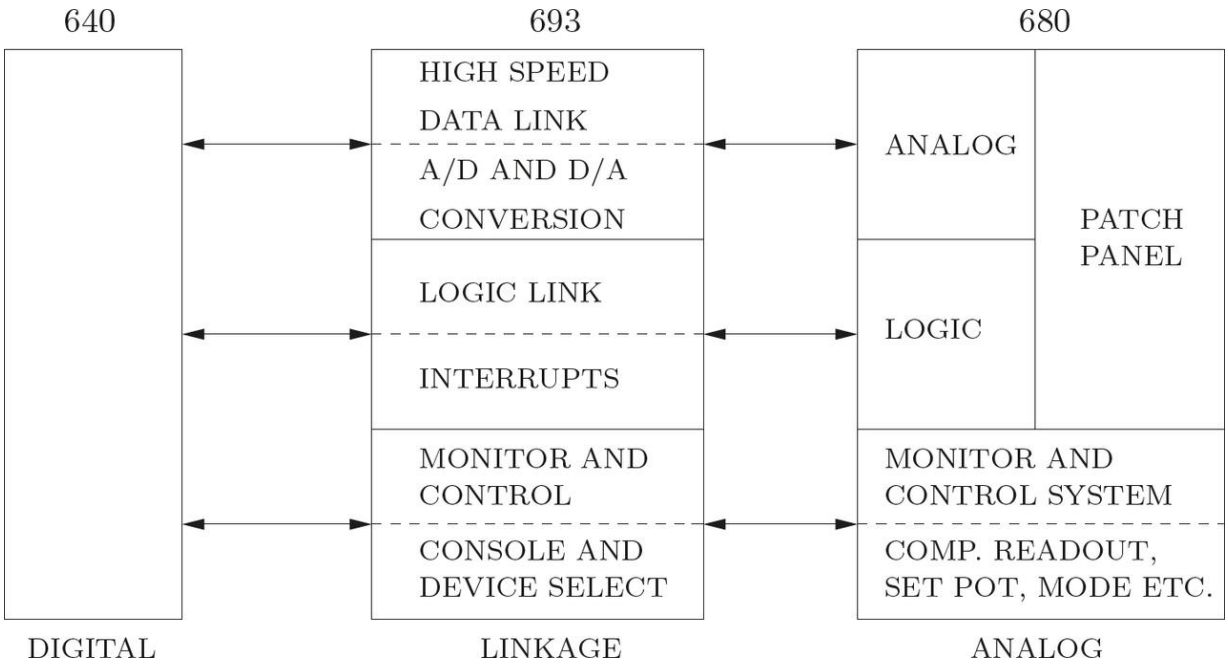


Figure 10.1. Block diagram of the EAI 690 hybrid computer system consisting of an EAI 640 digital computer, the EAI 693 linkage system, and an EAI 680 analog computer (cf. [→EAI 690])

One of the first hybrid controllers was the *ADDAVERTER* shown in →figures 10.2 and →10.3.[1] This system was developed and built by *Space Technology Laboratories*[2] in 1956.[3] The system shown features 15 ADC- and 10 DAC-channels[4] with a precision of ±0.1%, matching well the precision of large analog computers of that time.

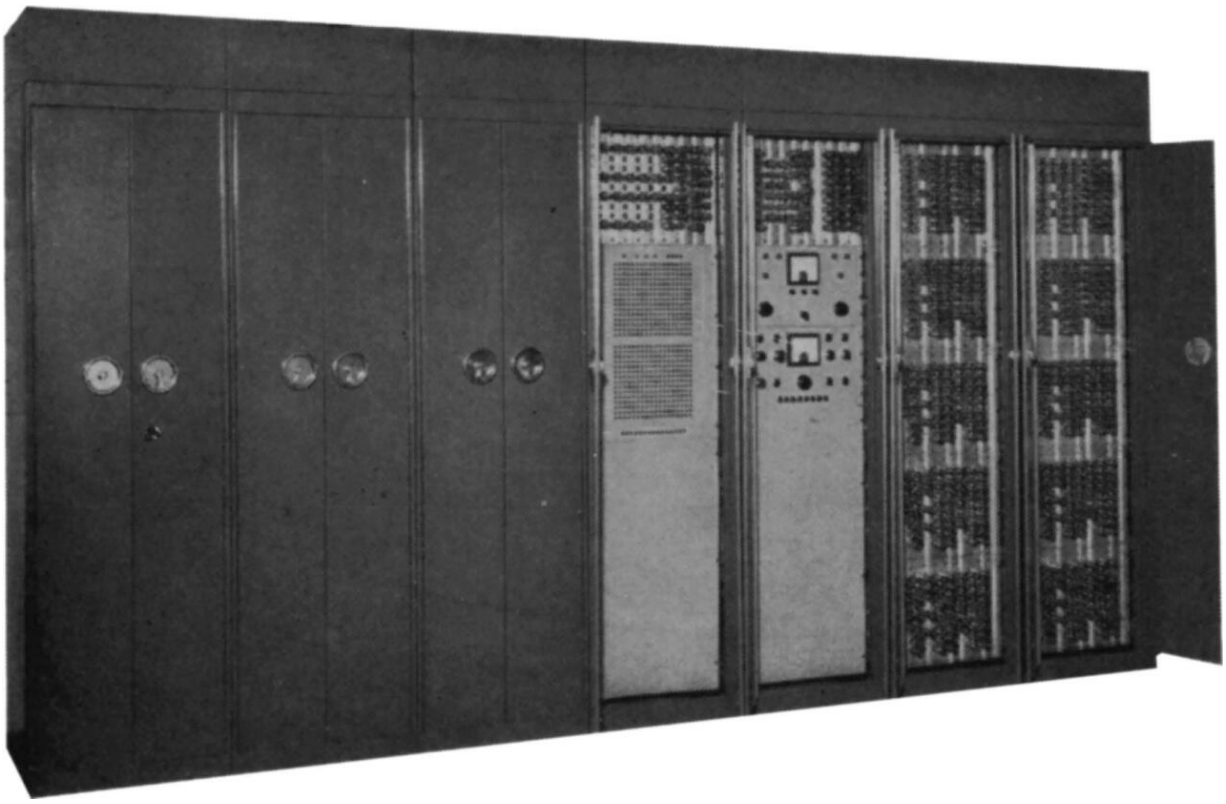


Figure 10.2. Space Technology Laboratories ADDAVERTER (see [→ McLeod et al. 1957, p. 1129])



Figure 10.3. Large ADDAVERTER installation

This development was the result of previous research efforts by Ramo-Wooldridge Corporation and *Convair Astronautics*. Both companies were working on simulations of intercontinental ballistic missiles, which involved very large analog computer systems.[5] A typical ADDAVERTER system, also called “computer-converter” cost $200,000 USD in 1957.[6] The requirements of these simulations are summarized by [→ McLeod et al. 1957, p. 1127] as follows:

> “*It was imperative that the simulation be done in real time to allow inclusion of weapon system hardware, and to conserve operating time. This ruled out an all-digital simulation because the large number of individual computations could not be made fast enough, and because analog equipment would be necessary to connect to some of the weapon system components which were to be included. An all-analog simulation was ruled out by accuracy requirements, particularly with respect to the navigational problem. Clearly a combined simulation was necessary to fulfill the requirements […]*”

The system that was finally put into operation at Convair Astronautics consisted of an IBM 704 digital computer, an ADDAVERTER, and a large EAI analog computer. The overall cost for this system was 2.3 million US$ for the IBM 701, 200,000 US$ for the ADDAVERTER, and 1.6 million US$ for the analog computer[7] – very substantial sums at that time.[8] A similar system is described in [→ Burns et al. 1961] where an IBM 704 is coupled with a REAC system.

Although this system was never used for the proposed task, which had been solved by conventional means in the meantime,[9] it served as a demonstrator of the efficiency and power of hybrid computers.[10] Subsequently the market demanded more and more hybrid systems, which was aided by the development of fully transistorized precision analog computers using a machine unit of ±100 V.[11] One of the first such analog computers was developed by *Comcor* in 1964 and offered the same precision as previous vacuum tube based machines but at lower cost, lower power consumption, and longer maintenance intervals.[12]

A typical 1960s hybrid system is shown in → figure 10.4. This system consists of an SDS-9300 digital computer[13] visible on the left and a Comcor *CI-5000*[14] analog computer on the right with a multi-channel recorder in front of it.



Figure 10.4. Hybrid computer installation at the Department for Electrical Engineering of the Naval Postgraduate School in the late 1960s (with permission of Robert Limes)

Suitable digital computers for a hybrid computer setup must be fast enough to keep up with the analog system in a simulation.[15] Particularly, simulations of nuclear reactors showed that the analog computer could easily outperform the digital system and was consequently slowed down considerably in a hybrid setup. In some cases a hybrid computer approach was only about three to ten times faster than a purely digital solution.[16]

Apart from this obvious idea of coupling analog and stored-program digital computers, other schemes were devised as well. In [→ Karplus et al. 1972], an analog co-processor consisting of a complex resistor-network is described, aiding a digital computer in the solution of parabolic partial differential equations. A rather unique if not to say strange concept is described in [→ Connelly 1963]: The idea is to extend a classic digital computer

with a *pulsed-analog* system consisting of summers, multipliers, absolute value functions, logarithm functions, and exponential functions. These are fed with values from the digital computer using DACs and perform some computation, the output of which is stored, and then fed back to the digital computer. Using this technique, the analog computing elements are time-shared between various parts of the digital main program.

Another interesting idea is the implementation of hybrid number systems in which values are represented by a combination of continuous signals like voltages and sequences of bits, thus forming a generalized form of floating point numbers consisting of a mantissa part and an exponent. Here the analog part of a number interpolates between two values that can be described exactly by a binary sequence.[17]

In 2005 Glenn Edward Russell Cowan and Ning Guo in 2016 developed two reconfigurable analog computers in the form of *VLSI* (*Very Large Scale Integrated circuit*) integrated circuits.[18] Although no products resulted from these academic developments, it was demonstrated that analog computers can still be used as co-processors to speed up digital computers considerably for certain types of application.[19]

10.2 Programming

Programming the analog half of a hybrid computer is not different from classic analog computer programming. The only difference are some additional input/output and computing elements such as ADCs, DACs, and function generators. Programming the digital half of such a setup is more challenging, as it has to work closely coupled with the very fast analog computer. This typically demands (very) low interrupt latencies, reentrant code, etc. As requirements like these are

typical for systems used for process automation tasks, programming paradigms from this area can at least partially be applied to hybrid computers.

Basically, two modes of operation must be distinguished in a hybrid computer system:

> ***Alternating operation***: In this mode the analog and digital subsystems work in an alternating fashion. Typically, the digital computer sets coefficients on the analog computer, which is then run for some time to yield a solution. At the end of or even during this analog computer run data is collected by the digital computer, which can then be used to derive an improved parameter set for the analog computer, etc. From a programming perspective, this mode of operation is quite simple since timing is not critical.[20] Furthermore, the digital computer has full control of the analog computer.
> ***Simultaneous operation***: This mode of operation is by far more complicated. Here both computers operate in a closely coupled fashion and the digital computer must respond to interrupts generated by the analog computer, it must generate functions of more than one variable, it must read values in real time, perform calculations, and generate input signals for the analog system.

Historically two main approaches for programming the digital part of a hybrid computer have been pursued. The first is based on extending traditional programming languages such as *FORTRAN* or *ALGOL* with additional language features to support the control of the various interfacing devices including fast interrupt routines.[21] The second approach is to implement all required functionality in libraries not changing any language features at all.

A good overview of the various software subsystems included in the EAI 8900 hybrid computer system is given in [→ Bekey et al. 1968, p. 181]. Apart from an extended FORTRAN dialect, a special purpose language, *HYTRAN*, was also available as well as a powerful macro assembler, which was typically used to implement highly time-critical routines. These systems also had the advantage that specialized diagnostic routines were available to automatically check the digital computer as well as the analog computer. Typical tests included *rate checks* and *static checks*.[22] A rate check tests the time-scale factors of the integrators by applying a known constant input signal for a precisely determined amount of time.

A complex, yet instructive example of the application of a hybrid computer system to an optimization task can be found in [→ Witsenhausen 1962]. This paper gives a thorough description not only of the analog computer setup but also describes the necessary control routines written for the digital processor. [→ Giloi et al. 1969] describe a complex and flexible programming system for hybrid computers consisting of a "*h*ybrid-procedure *a*ugmented *r*eal-*t*ime FOR*TRAN*" (*HARTRAN*), a "*hy*brid system con*trol*" (*HYTROL*), the "*a*nalog *c*omputer and *i*nterface *d*iagnostics" (*ACID*), tools for "*stat*ic *test*s" of the analog computer setup (*STATEST*), and hybrid library programs.

10.3 ML training example

A modern hybrid controller implementation is shown in →figure 10.5. This particular device allows an Analog Paradigm *Model-1* analog computer to be coupled with a digital computer via a USB[23] connection. At the heart of the hybrid controller is a variant of an *Arduino MEGA-2650* controller, which handles the communication with the digital computer as well as control of

the analog computer. This module also contains eight *digital potentiometers* with 10 bit resolution, eight digital input, and eight digital output lines.

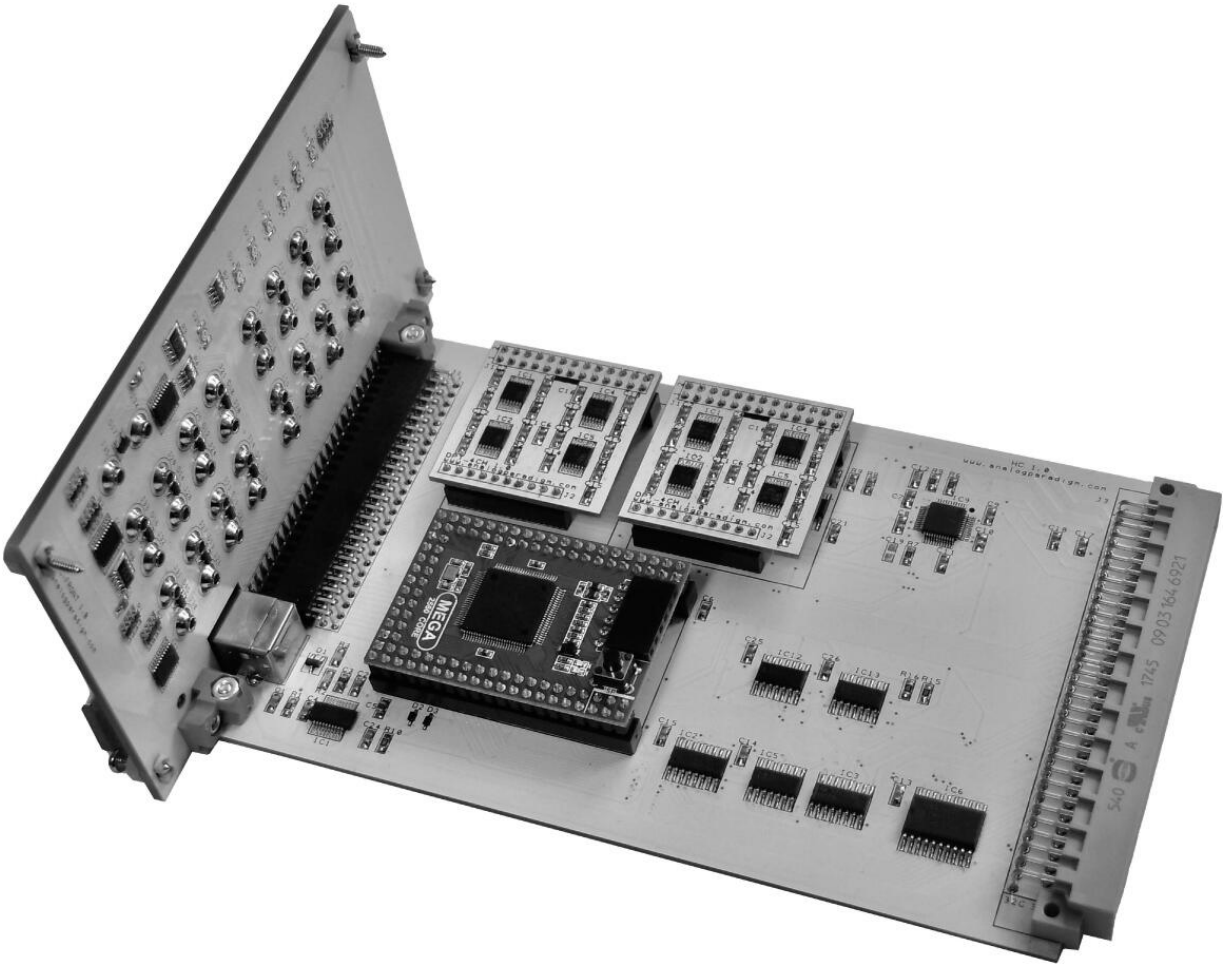


Figure 10.5. Modern USB-based hybrid controller for an Analog Paradigm Model-1 analog computer

An nice example of a hybrid computer employing simultaneous operation based on this module is described in [→ Holzer et al. 2021]. The analog part of the setup is programmed to simulate the behavior of an inverted pendulum mounted on a cart, which can be moved along the *x*-axis under control of the digital

computer. The digital computer runs a reinforcement learning system, which is trained to balance the inverted pendulum by moving the cart in little steps.

The analog computer program simulating the behavior of the inverted pendulum is shown in →figure 10.6. Its derivation, based on the EULER-LAGRANGE-equations, is described in detail in [→HOLZER et al. 2021]. Input to this program is $\ddot{x}$, the cart's acceleration in x-direction. Using two digital output lines `D0` and `D1`, the digital computer can push the cart to the left or to the right in small increments using the circuit shown in →figure 10.7.[24] The output signals x and y can be used to display the inverted pendulum on an x, y-display.

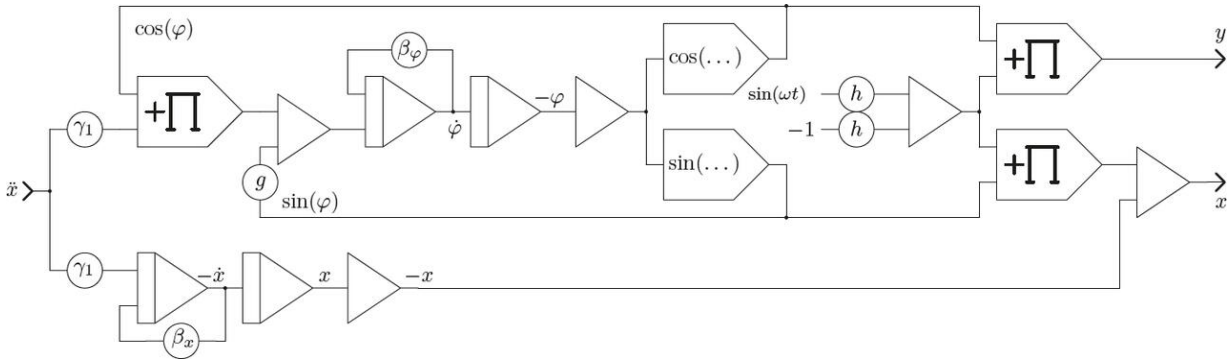


Figure 10.6. Analog computer program simulating an inverted pendulum mounted on a moving cart with one degree of freedom

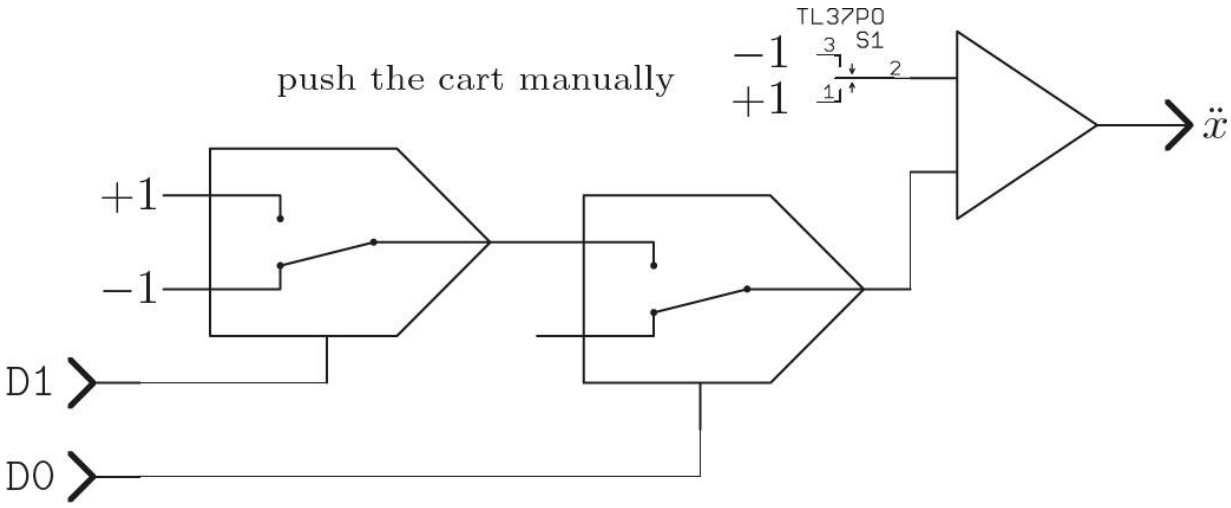


Figure 10.7. Control circuit for the cart's motion

The reinforcement learning system implemented on the attached digital computer uses a *Q-learning*[25] approach to learn how to balance the inverted pendulum. At the start of a simulation/training run, the cart is positioned in the middle of the simulated x-space with the pendulum in an upright position. The digital computer continuously reads the pendulums angle φ from the analog simulation and tries to keep the pendulum in this position. Whenever the pendulums angle exceeds a predetermined value or when the cart hits the left or right boundaries, the simulation is restarted.

After a few dozen runs, the digital computer showed first signs of actively balancing the pendulum, which evolved into near perfect control after several hundred runs. After this, even manual interventions, i. e., perturbing the cart by applying little pushes to the left or right, were easily corrected by the digital controller.[26]

10.4 Stochastic ODE on a hybrid computer

A simple example of alternating operation in a hybrid computer setup is the solution of a stochastic differential equation.[27] This class of equations is not only interesting as such but is also of great interest in financial mathematics and related fields.[28] The equation under consideration here is the BLACK[29]-SCHOLES[30] stochastic differential equation[31] (*SDE*)

$$\dot{y} = \lambda y + \sigma y W\,. \tag{10.1}$$

Here, W denotes a WIENER process, a stochastic process in continuous time, which was first described by NORBERT WIENER.[32] Such processes are the basis of *random walks*.[33] λ represents a *drift parameter,* σ denotes a volatility, and the initial condition of (→ 10.1) is $y(0)$.[34]

The WIENER process can be implemented by an electronic noise generator, while λ and σ are chosen so that a particular behavior of the equation results.[35] The corresponding analog computer setup is shown in → figure 10.8. The loop consisting of the integrator and the inverter on the right implements the $\dot{y} = \lambda y$ part of (→ 10.1). The output of the integrator is multiplied with the noise input signal, the product being fed into the integrator.[36] The upper frequency limit of the noise generator should match the integrator's k_0. Experiments have shown that a maximum noise frequency of 100 kHz works well with $k_0 = 10^3 \cdot 10$.

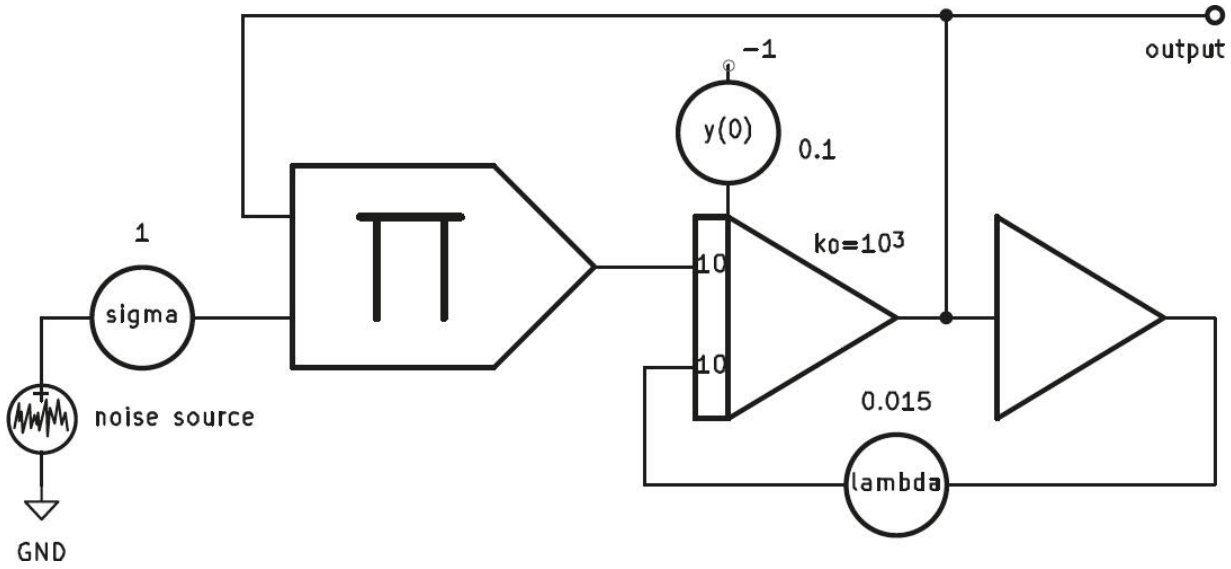


Figure 10.8. Analog computer setup for equation →10.1

Being a stochastic problem, the behavior of the system becomes apparent only when doing statistics over a large number of runs. Here, the digital part of the hybrid computer setup cycles the analog computer through IC and OP mode and reads the value of the integrator at the end of each OP period. These values are then sent by a serial line protocol to a digital computer, which collects and groups these into bins of equal width. A typical result plot is shown in →figure 10.9.

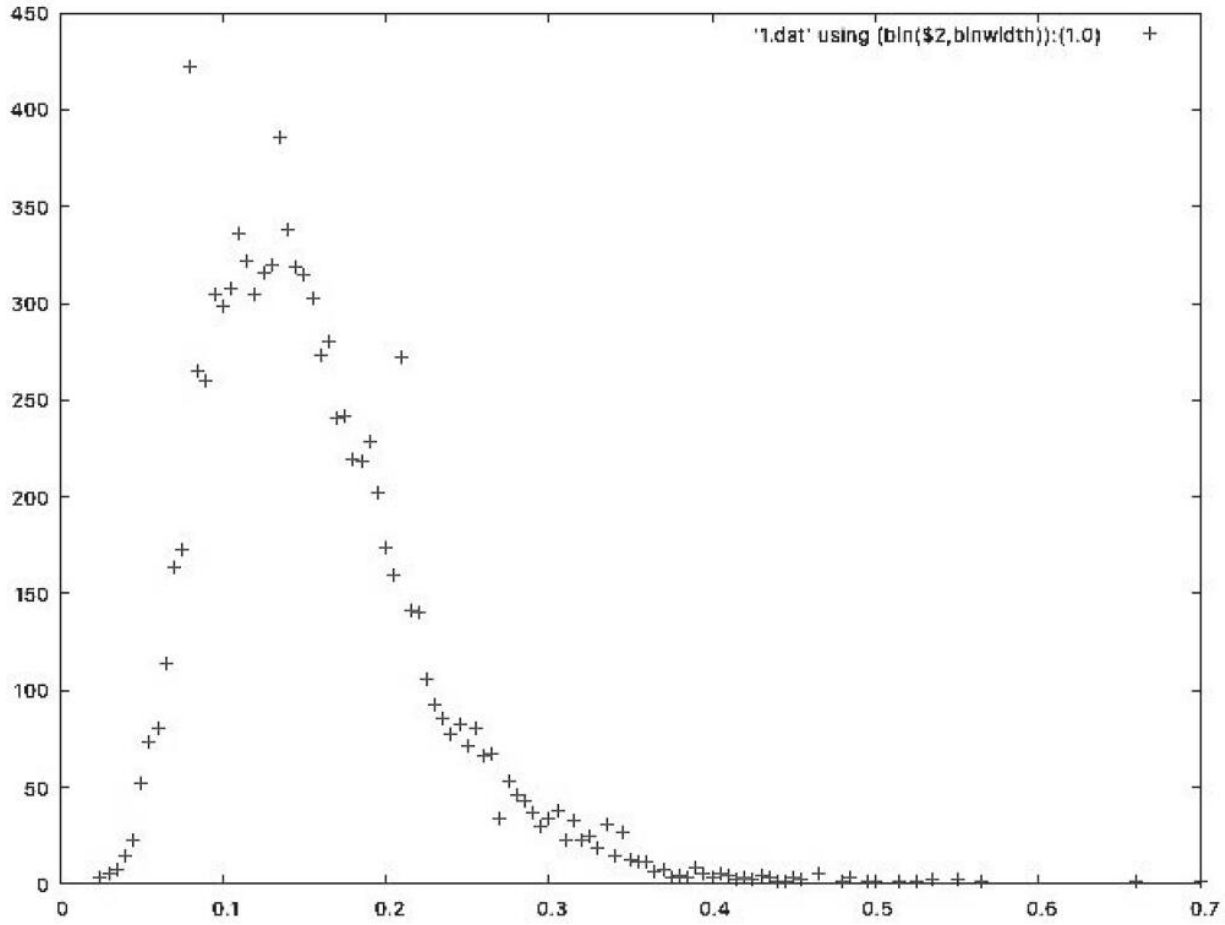


Figure 10.9. Solution for equation →10.1 obtained by the hybrid computer approach

The problem was programmed on THE ANALOG THING connected to an *Arduino* Mega 2560 microcontroller.[37] The overall control program is shown in →figure 10.10. Using the `THAThc` library, the analog computer can be controlled with simple method calls. The `setup` routine initializes a serial connection which is used to send data to an attached computer via USB and instantiates a `THAThc` singleton which is used in the central LOOP.

```
1  #include "THAThc.h"
2  #define BAUD_RATE 250000
3
4  void setup() {
5    Serial.begin(BAUD_RATE);
6    THAThc.begin();
7  }
8
9  void loop() {
10   THAThc.enable();                  // Configure the hybrid controller
11   THAThc.set_ic_time(1);            // IC time is one millisecond
12   THAThc.set_op_time(2);            // OP time is two milliseconds
13   THAThc.set_channels(1);           // Only one ADC channel is used
14
15   // Perform repetitive runs and sample result at the end of each run
16   for (unsigned long i = 0; i < 10000; i++) {
17     THAThc.single_run();            // Perform a single IC/OP cycle
18     THAThc.block();                 // Block until this cycle finished
19
20     float result;
21     THAThc.sample_adc(&result);     // Read the result
22     Serial.print(String(i) + "\t"); // Print number of run
23     Serial.print(result, 3);        // and the result
24     Serial.print("\n");
25   }
26
27   for( ; ; delay(1000));            // Stop
28 }
```

Figure 10.10. Control program

Here, the hybrid controller is activated. Then the duration of the IC and OP period of a single run are defined as one and two milliseconds, respectively. The microcontroller is then configured to use one analog input channel. The FOR-loop spanning lines 16 to 25 performs 10000 IC/OP cycles, reading the result at the end of each OP period,[38] which is then sent though the serial line interface. These values are read by an attached digital computer,

written to a file and processed for display using `gnuplot`[39] with the plotting commands shown in →figure 10.11.

```
1  binwidth=.01
2  bin(x,width)=width*floor(x/width)
3  plot '2.dat' using (bin($2,binwidth)):(1.0) smooth freq
```

Figure 10.11. `gnuplot` commands to generate a bin plot

11 Autopatch systems

One of the main drawbacks of classic analog computers is the tedious and often error prone programming process. Many realistic problems require hundreds, in some cases thousands of patch cables, making even small changes in an existing program or switching from one problem to another a major undertaking. Removable patch panels as usually found in larger analog computers simplify things quite a lot but are still clumsy, and changing from one problem to another still involves setting dozens or hundreds of potentiometers, function generators, etc., manually.

This was especially pronounced in the days of mechanical differential analyzers as BUSH notes:[1]

> "*It was a disadvantage [...] that the assembly of a problem by means of an array of shafts, gears, and couplings required a substantial amount of time, frequently one or two man-days. Once the assembly was completed solutions came swiftly, and after a few days of operation the 'set-up' time per solution became tolerably small and even negligible. The situation was unsatisfactory [...]. [...] a machine of this type could not render effective aid to the user if the number of solutions required was so small that the time required to assemble the machine was comparable with the time required for numerical integration*."

When the setup time is about as high as the time required to obtain a numerical solution, even with the crude, mostly manual methods of the early 20th century, the benefit of such a machine would not be worthwhile. Accordingly, the ROCKEFELLER differential analyzer[2] became the first reconfigurable electromechanic differential analyzer. It used a crossbar switch similar to those found in telephone exchanges controlled by three types of punched paper tape, A-, B-, and C-tapes controlling the crossbar

switch, the values of coefficients within a setup, and the initial conditions of the integrators, respectively.[3] This worked very well:[4]

> "*The punched tape merely controls the operation of an automatic switching system which makes the servomechanism connections*.[5] *For the largest problem so far put on the machine, the entire assembly was done in 3 to 5 minutes – less than the time required for a single solution*.[6]"

This gives rise to the question why later analog computers used large, clumsy, and cumbersome patch panels instead of a central crossbar switch that proved so beneficial in this early differential analyzer. Getting rid of the patch panel as such would have been a major asset for analog computers and, in conjunction with automatic means for setting coefficients and initial conditions, could at least have delayed their demise.

There are a number of reasons why the idea of a central crossbar switch did not catch on: First of all, it does not scale well, at least not without having a hierarchy of switch matrices. Having n computing elements with one output each and m inputs in total would require a crossbar with $n \times m$ switching elements. An early electromechanical machine like the ROCKEFELLER differential analyzer had sufficiently small n and m making the straightforward approach of using a single-stage crossbar structure feasible despite the rather large size of electromechanical crossbar switches as used in telephone exchanges. The sheer size of this setup required long connecting cables, thus severely limiting the bandwidth of signals within such a machine. Again, this was not a problem with electromechanical computing elements having bandwidths of a few Hz at most but would have been utterly impractical with later electronic high-speed analog computers having bandwidths sometimes in excess of several 100 kHz and more.

After a long hiatus in this area of research, EAI conducted a study in 1961 showing that an autopatch system, i. e. a reconfigurable interconnect for an EAI 231R analog computer would require between $5 \cdot 10^4$ and 10^5 individually controlled switches – far from being practical at the time. However, the advantages such a system would offer over the traditional patch panel were significant enough to justify further research. One of the most comprehensive systematic studies on reconfigurable analog computers was done in the late 1960s and is described in [→Hannauer 1968]. One of the major requirements was to keep the number of switches at around 10^4 without causing too many restrictions regarding configuration flexibility. Of course, an *all-to-all* interconnect would be ideal, but this is, even today, not within reach due to the quadratic growth of the switching matrices. Accordingly, some restrictions with respect to configuration flexibility are unavoidable.

In this study coefficient potentiometers were no longer treated as individual computing elements but were assigned to the inputs of computing elements such as summers, integrators, multipliers, etc. Typically, this required more potentiometers than a traditional setup but lowered the number of configurable connections substantially, helping to keep the number of crossbar switches within reasonable limits. In addition to that, computing elements were grouped together so that the degree of connectivity within a group was higher than the degree of connectivity between groups, further reducing the number of switches required but at the cost of reduced flexibility.

A program for such a reconfigurable analog computer is a bitstream containing the configuration of the interconnect system, the coefficients within a particular setup, and the respective initial conditions. Typically, it will be generated by a more or less sophisticated *compiler* running on a digital computer.[7]

Another central idea in [→HANNAUER 1968] is that of switch matrices being characterized by their *expansion factor* $e = \frac{m}{n}$ with *n* and *m* representing the number of respective outputs and inputs of such a matrix. In the case *e* < 1 the matrix is called a *concentrator* as it has fewer outputs than inputs, while *e* > 1 corresponds to an *expander*. Based on this idea, a three stage structure as shown in →figure 11.1 was suggested. This quite closely resembles a classic telephone exchange which has to be able to serve any customer connected to it but not all at the same time, having a certain capacity based on the average number of simultaneous connections for a given area.[8]

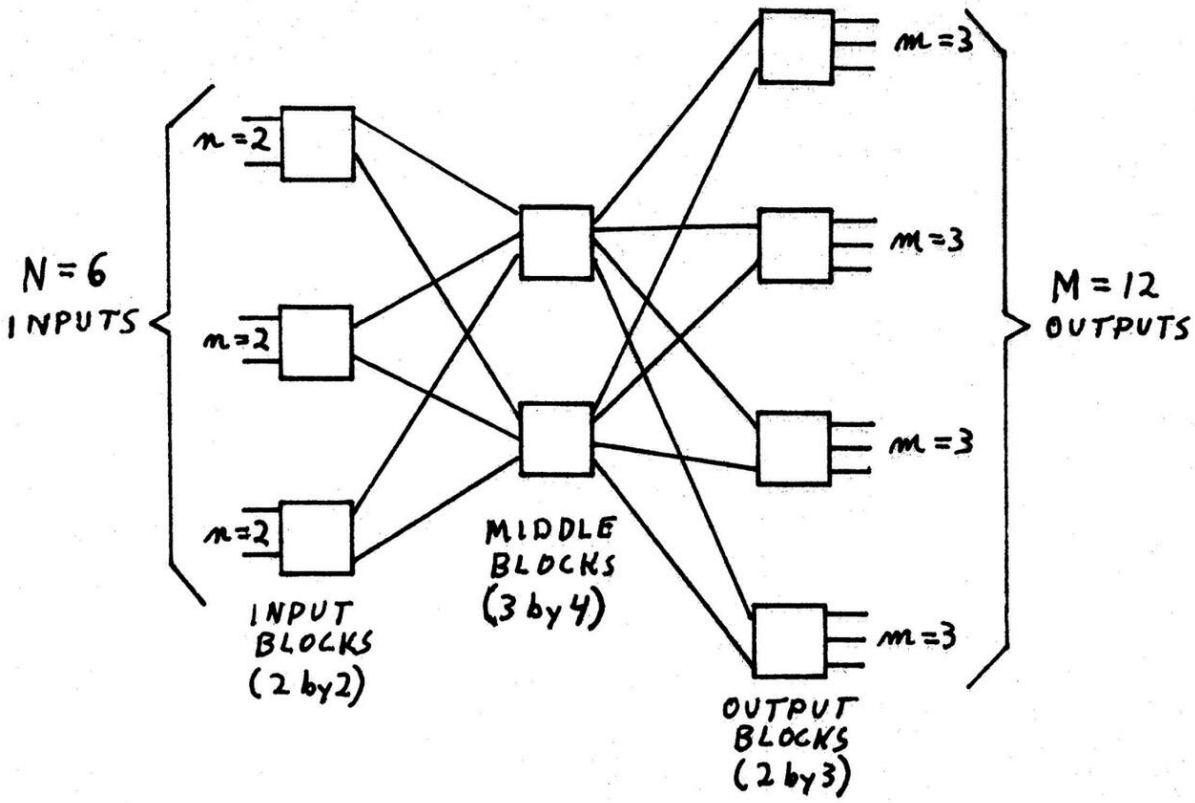


Figure 11.1. Three stage coupling matrix structure (see [→HANNAUER 1968, p. 4-2])

[→ HANNAUER 1968, pp. 6-20 ff.] continues to compare the number of switches required in a naïve rectangular matrix with that of the proposed three-stage design based on a hypothetical analog computer system with 150 inputs and 372 outputs.[9] The single matrix approach would require $150 \cdot 372 = 55800$ individual switches, while the concentrator-expander setup would only need 19 concentrator input blocks, 14 “middle blocks”, and 28 expander output blocks totaling 14602 switches for a reasonable degree of flexibility. This is still a rather large number but well within reach even with relay technology of the 1960s.

However, this approach does not allow for an all-to-all connectivity. In fact a certain *blocking penalty* has to be taken into account. This quantifies the sparsity of the interconnect structure in a system. In classic telephone exchanges this problem can usually be tolerated since it rarely arises and parties willing to communicate typically can postpone their calls if the system is blocked. In an analog computer setup this problem is much more severe as it precludes certain programs to be implemented at all. However, a study conducted at NASA showed that this three-stage design would have allowed all of their benchmark problems to be implemented.[10]

Based on these promising results, EAI developed an autopatch expansion for their 680 analog computer from 1975 to 1980,[11] thus forming the base for the EAI *SIMSTAR* system, which was announced in 1983.[12] It used a three-stage switch matrix structure with twenty 16 × 20 input blocks for a total of 320 input signals, twenty 20 × 32 middle blocks, and 32 22 × 16 output blocks[13] yielding 512 output signals. This required a total of 30464 switches instead of 163840 that would be required in case of a single 320 × 512 matrix. At the heart of these crossbar matrices were *μ*PD22100 CMOS 4 × 4 crossbar switches[14] superseding the previous reed-relay based implementations.

Using these integrated circuits the whole 320×512 interconnect structure of SIMSTAR required only 46 printed circuit boards (*PCB*s). However, this system as a whole was still quite big, occupying five large racks.

In only three years EAI received 15 orders for SIMSTAR systems from a variety of customers and for an even larger variety of application areas including nuclear power plant controls, missile studies, flight systems, etc.,[15] see →table 11.1.[16]

Table 11.1 EAI SIMSTAR orders as of 1986 ([→KACZALA 1986])

Order	date	Units	Customer	Application area
Nov	1983	1	Westinghouse, USA	nuclear power plant controls
Dec	1983	1	MICOM (US Army)	missile studies
Feb	1984	2	MICOM (US Army)	missile studies
May	1984	1	TU Vienna, Austria	robotics, control
Aug	1984	1	NCSC (US Navy)	torpedo studies
Dec	1984	1	Eastman Kodak, USA	controls of motors
Feb	1985	3	Martin Marietta, USA	missile studies
Mar	1985	2	N.N., USA	—
May	1985	1	Nissan Aerospace, Japan	missile control studies
Jul	1985	2	WPAF Inst. of Technology, USA	flight system

As an example, the system ordered by the Technical University Vienna, *TU Wien*, in 1984 underwent acceptance tests at the EAI factory in December 1984 and subsequently was delivered and installed at the university where it was finally accepted on 19.03.1985. It contained 28 integrators which could also be used as track/store elements, 18 multiply/divide/square root elements, ten summers with variable gain, 18 summers with

fixed inputs, twelve comparators, 32 trunk inputs, and 64 buffered trunk outputs, connecting the system with external equipment. A noteworthy feature of this system were variable limiters for all integrators, multipliers, and the 18 summers.[17]

The SIMSTAR system was an astonishing machine.[18] Apart from classic analog computing elements it featured an automatic noise/oscillation detection system, an intricate high-quality grounding system, deglitcher circuits to suppress transients during fast switching processes, a programmable parallel logic processing unit (PLU), an integrated 32 bit digital processor with an 80 MB hard disk,[19] a high-precision readout system, operational amplifiers with a bandwidth of 60 MHz, and an autobalance system to counteract drift effects. The FORTRAN based programming environment was called *STARTRAN*.[20]

[→ Kleinert 1985] describes five benchmark problems executed on the SIMSTAR at TU Wien. The first of these problems is a differential equation of order 22, $y^{(22)} = -y$. The solution obtained by the SIMSTAR is very close to the expected sine function (with suitable initial conditions), proving that the phase shift caused by the three-stage switch matrix is basically negligible. The second example, a nonlinear system of tenth degree, yields a near perfect solution on the SIMSTAR, while a numeric solution of the same problem fails to reach the correct result. The third problem consists of a chain of 22 integrators with zero input which will eventually diverge due to unavoidable imperfections of the analog computing elements. However, even in this case the SIMSTAR system outperformed the numeric approach with respect to stability. The fourth problem, a linear differential equation of order 11 also yielded a better result on the SIMSTAR compared with the chosen numerical approach. The final task, a highly oscillatory problem from laser physics,

yields a correct analog solution while the numeric solution does not even converge.

A more recent development is described in [→Cowan 2005]. Here, a reconfigurable analog computer in CMOS technology is developed, containing 80 integrators, 80 variable gain amplifiers, which can be also used as multipliers, 16 logarithms, 16 exponential functions, 64 nonlinear computing elements, sign, and absolute value functions, etc.[21] These computing elements are grouped into *macroblocks* as shown in →figure 11.2. Here, the connectivity is highest within a macroblock and sparser between individual macroblocks.[22]

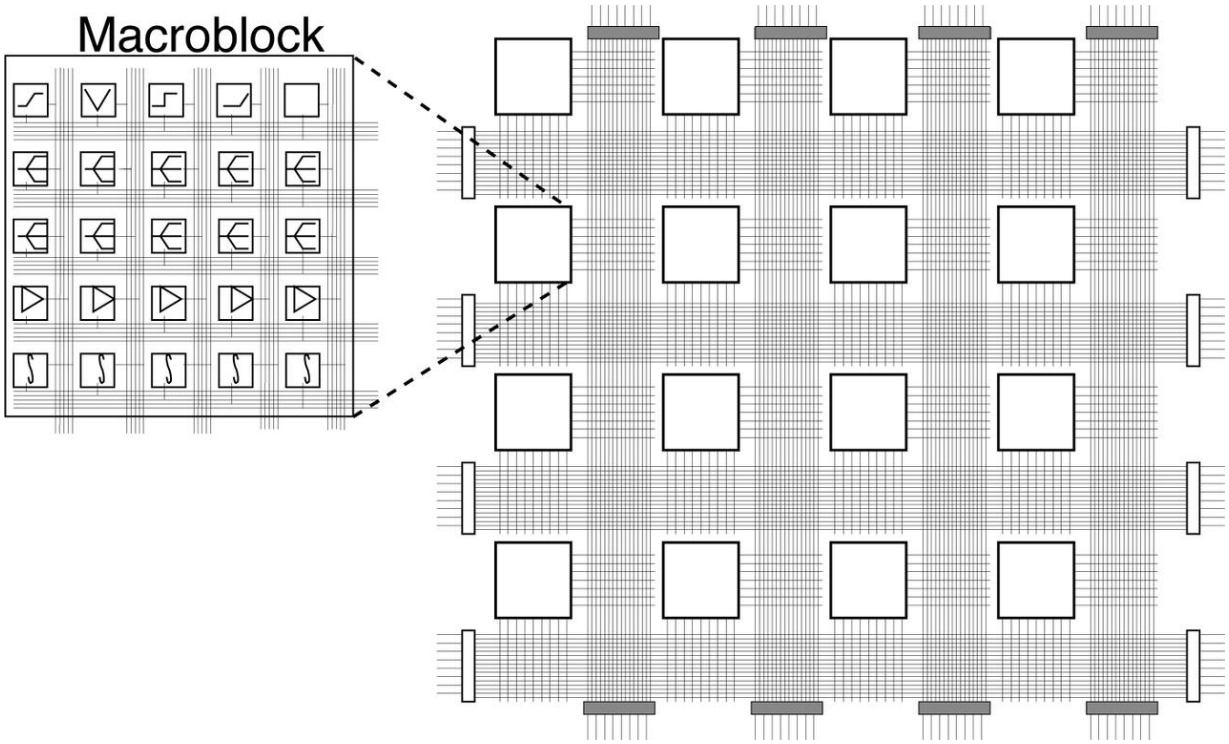


Figure 11.2. Global structure of Cowan's integrated analog computer

Another recent development are the *LUCIDAC* and *REDAC* systems developed by anabrid GmbH. These hybrid computer

systems employ a hierarchy of crossbar switches using voltage as well as current coupling, thus taking advantage of both coupling techniques in a single system. →Figure 11.3 gives a slightly simplified overview of the basic architecture of a LUCIDAC system.

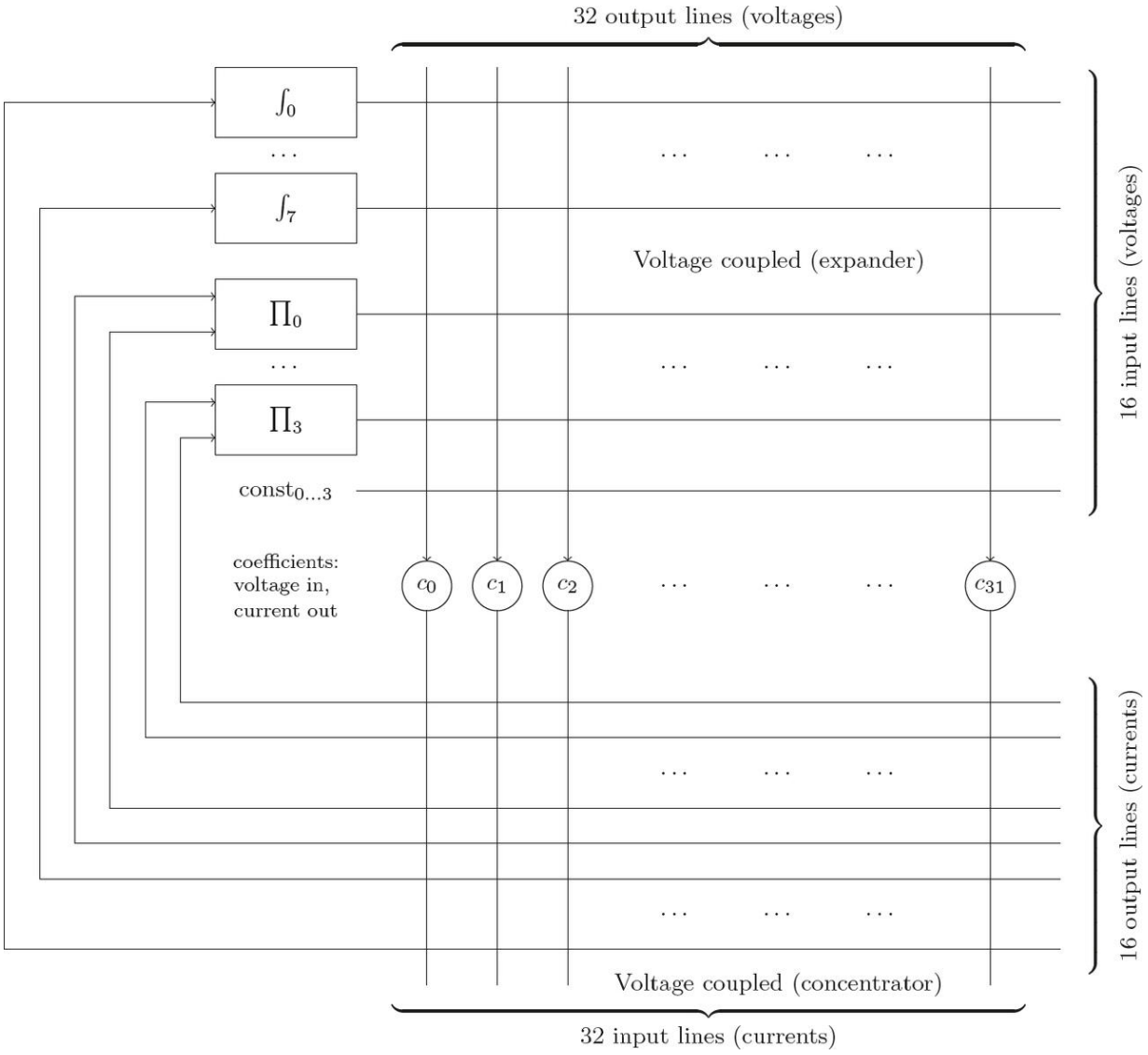


Figure 11.3. Crossbar structure of anabrid's LUCIDAC

A single LUCIDAC contains eight integrators with selectable $k_0 \in \{10^2, 10^4\}$, four multipliers, and 32 coefficient elements.[23] The widely spaced k_0 values are compensated for by coefficients spanning the interval $[-10, 10]$. This wide range of coefficient values also simplifies scaling of problems substantially. Each of the integrators and multipliers has a voltage output which is fed to a voltage coupled crossbar, the *U-block*. This crossbar has 16 input and 32 output lines, thus it is an expander. The switch matrix features built-in output drivers, so a single voltage input signal can be distributed to many output signals shown as vertical lines in the figure simultaneously. Each of these output lines is connected to a dedicated coefficient element with 12 bits of resolution. These coefficient elements yield an output current which is fed to a second crossbar, the *I-Block*. This crossbar has 32 inputs and 16 outputs and thus acts as a concentrator. It is just a matrix of switches with no output buffers allowing the implicit summation of multiple input currents into a single output line, feeding the computing elements. Thus, computing elements can have a variable number of inputs, each with an associated coefficient, without requiring any summers as explicit computing elements, greatly reducing the number of crossbar elements required. The main limitation of this approach is the sparsity dictated by the 32 coefficient elements, limiting the number of active connections within a LUCIDAC setup to 32.

This implicit summation characteristic of the LUCIDAC and REDAC systems requires a slight deviation from the standard notation for analog programs where all inputs to a computing element are just joined at a common summing junction instead of feeding individual inputs. →Figure 11.4 depicts the setup for a chaotic Lorenz system.[24]

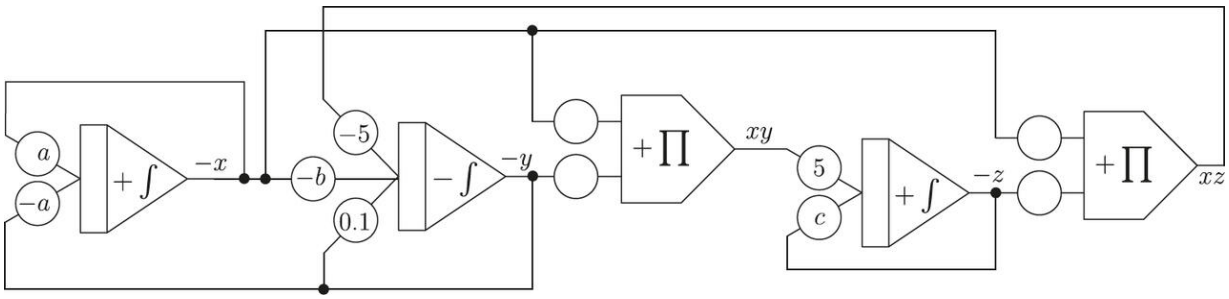


Figure 11.4. LORENZ system for a voltage/current coupled LUCIDAC system

In the most simple case, a LUCIDAC system is programmed using a low level Python library LUCIPY as shown in →figure 11.5.[25] Using this library, a circuit is instantiated with `l = Circuit()`. It offers a number of methods to allocate individual computing elements, in this case three integrators and two multipliers. The connections between these elements are then defined with a number of CONNECT calls. Using the PROBE method, output values are connected to up to eight ADCs for digitization. This readout system supports continuous digitization as well as readout of only the last value in a computer run depending on the problem.

```
from lucipy import LUCIDAC, Circuit

a = 1.0
b = 2.8
c = 2.666 / 10

l = Circuit()                    # Create a circuit

mx = l.int(ic = 0.1)             # Allocate an integrator with
my = l.int()                     # initial condition and two further
mz = l.int()                     # integrators
xz = l.mul()                     # Allocate two multipliers
xy = l.mul()

l.connect(mx, xz.a)              # Product -x * -z = xz
l.connect(mz, xz.b)
l.connect(mx, xy.a)              # Product -x * -y = xy
l.connect(my, xy.b)
l.connect(my, mx, weight = -a)
l.connect(mx, mx, weight = a)
l.connect(mx, my, weight = -b)
l.connect(xz, my, weight = -5)
l.connect(my, my, weight = 0.1)
l.connect(xy, mz, weight = 5)
l.connect(mz, mz, weight = c)

l.probe(mx, front_port=0)        # Connect an oscilloscope to
l.probe(my, front_port=1)        # ports 0, 1, and 2 for display
l.probe(mz, front_port=2)

hc = LUCIDAC()
hc.set_circuit(l)                # Configure the system
hc.set_op_time(unlimited=True)   # Continuous OP time
hc.start_run()                   # Run
```

Figure 11.5. Low level Python program implementing the LORENZ system on a LUCIDAC

A much more abstract and user-friendly way of describing a LUCIDAC setup is shown in →figure 11.6. This source code is compiled into a corresponding analog computer setup by an anabrid developed compiler which also supports automatic scaling of problems, further simplifying the programming process.

```
1   fn X(t);
2   fn Y(t);
3   fn Z(t);
4
5   let diff[X, t] = 1.8 * Y - X;
6   let diff[Y, t] = 1.56 * X * (1 - 2.678 * Z) - 0.1 * Y;
7   let diff[Z, t] = 1.5 * X * Y - 0.2667 * Z;
8
9   let X(t: 0) = 0.1;
10  let Y(t: 0) = 0.0;
11  let Z(t: 0) = 0.0;
12
13  plot(x: X(t), y: Y(t));
14
15  out X(t);
16  out Y(t);
```

Figure 11.6. High-level description of the LORENZ system for a LUCIDAC computer

12 Digital differential analyzers

The essence of an analog computer is that it forms an analog of the problem to be solved. Although most analog computers are based on analog electronic implementations of their respective computing elements working in continuous time and using continuous voltages or currents to represent values, there is no need to restrict analog computers to this particular type of implementation.

A *digital differential analyzer* (*DDA*) is basically an analog computer but one working in discrete time with values represented as serial or parallel bitstreams.[1] The earliest developments of DDAs date back to the late 1940s when *Northrop* started the development of a cruise-missile like system.[2]

Representing values as bit sequences makes it possible to achieve much higher precision than with a traditional analog computer, and it is even possible to represent values as floating point numbers, thus eliminating the need for scaling altogether, if the complexity of the implementation is acceptable. Using logic gates as the foundation also eliminates any problems caused by drift or aging of components. Nevertheless, these advantages come at a price: Computation is no longer continuous, neither with respect to the value representation nor regarding time. Furthermore, the energy efficiency of such machines is not as high as that of machines using analog electronic computing elements.[3]

Basically, there are two different approaches to implement a DDA. A straight-forward *parallel* DDA[4] consists of a whole complement of typical analog computing elements, which can be

interconnected by means of a traditional patch panel or the like. An example for this class of machines is *TRICE*, which is described in more detail in →section 12.4.4. Machines like this exhibit the same amount of fine-grained parallelism as a traditional analog computer, thus easily outperforming classic stored-program digital computers. Nevertheless, this high computational power comes at a very high cost since the number of computing elements is dictated by the complexity of the problems to be solved, with large problems requiring a vast number of computing elements.

The second basic approach is based on a time-multiplexed use of a few – in some cases only one – central computing elements. Like a simple stored-program digital computer this *sequential* DDA has one or a few central *arithmetic/logic unit*s (*ALU*s), which are fed with data under control of a sequencing unit that in turn is controlled by a machine-readable description of the connections between the virtual computing elements. This second approach is much cheaper to implement due to its reduced complexity and the lower number of computing elements required. In addition to that, it scales quite well since large programs just require more time to solution whereas they could not be implemented on a parallel DDA if it did not contain enough computing elements. The major disadvantage of sequential DDAs is their relatively low computational power.

12.1 Basic computing elements

The following section describes some basic DDA computing elements regardless of how they are implemented.[5]

12.1.1 Integrators

As with an analog computer, the central element of a DDA is the integrator, which is most easily implemented based on an *accumulator*. Although simple accumulation of values will not be sufficient for more demanding computations, this basic approach led to the term *incremental computer* for a DDA.[6] An accumulator-based integrator has the distinct advantage that not just time, but every variable can serve as the free variable of integration so that integrals like

$$\int_{y_0}^{y_1} f(x)\mathrm{d}x$$

can be treated directly.

The basic structure of a simple DDA integrator is shown in →figure 12.1. In contrast to an analog electronic integrator there are two inputs $(\Delta Y)_i$ and $(\Delta X)_i$. These variables are often represented not as absolute values but as incremental values, which simplifies the implementation of the DDA and reduces the number of signal lines required to interconnect computing elements. These incremental values are often restricted to the set {−1; 0; 1}.[7] $(\Delta Y)_i$ represents the change of the integrand, while $(\Delta X)_i$ is the change of the variable of integration and corresponds to d*x*.

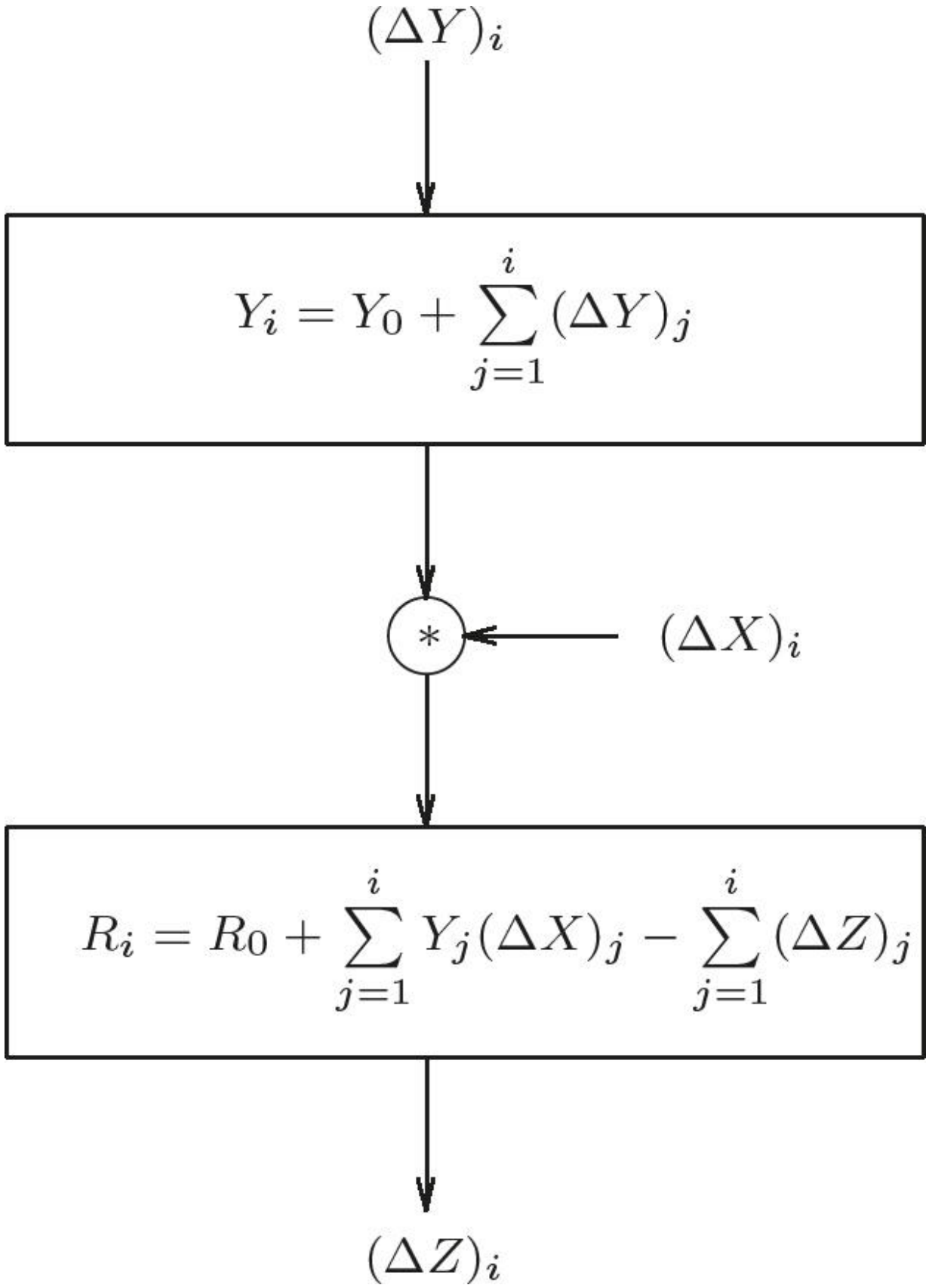

$(\Delta Y)_i$
$Y_i = Y_0 + \sum_{j=1}^{i} (\Delta Y)_j$
$*$
$(\Delta X)_i$
$R_i = R_0 + \sum_{j=1}^{i} Y_j(\Delta X)_j - \sum_{j=1}^{i} (\Delta Z)_j$
$(\Delta Z)_i$

Figure 12.1. Basic integrator of a DDA (cf. [→Bendix 1954, p. 1])

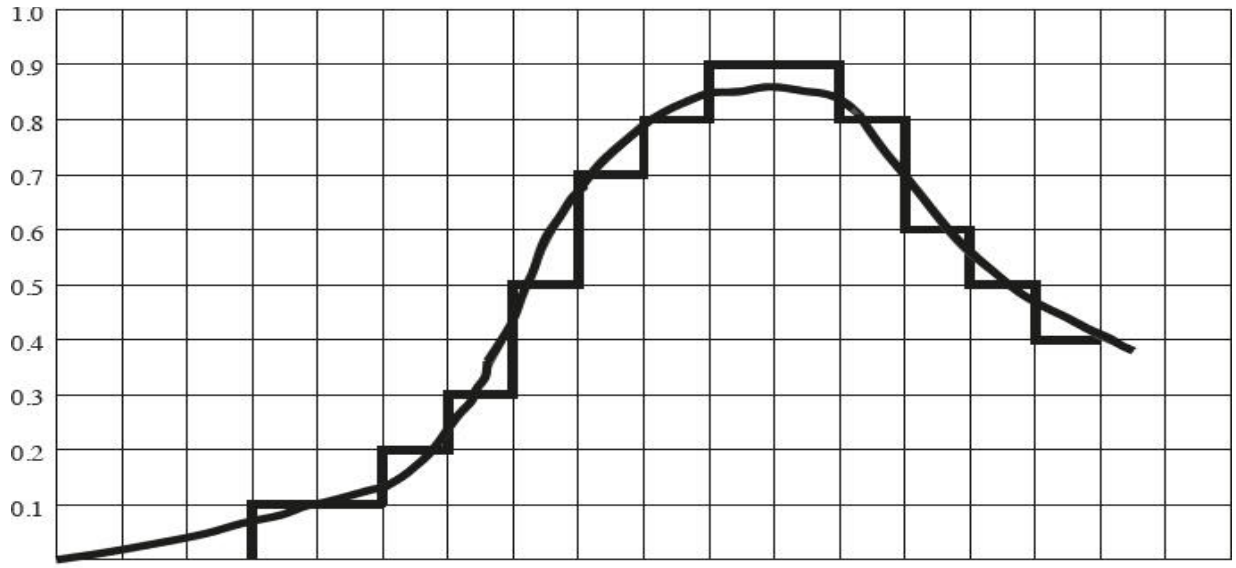
At the heart of an integrator like this are two accumulators: One accumulates the series of $(\Delta Y)_i$ values at its input and yields

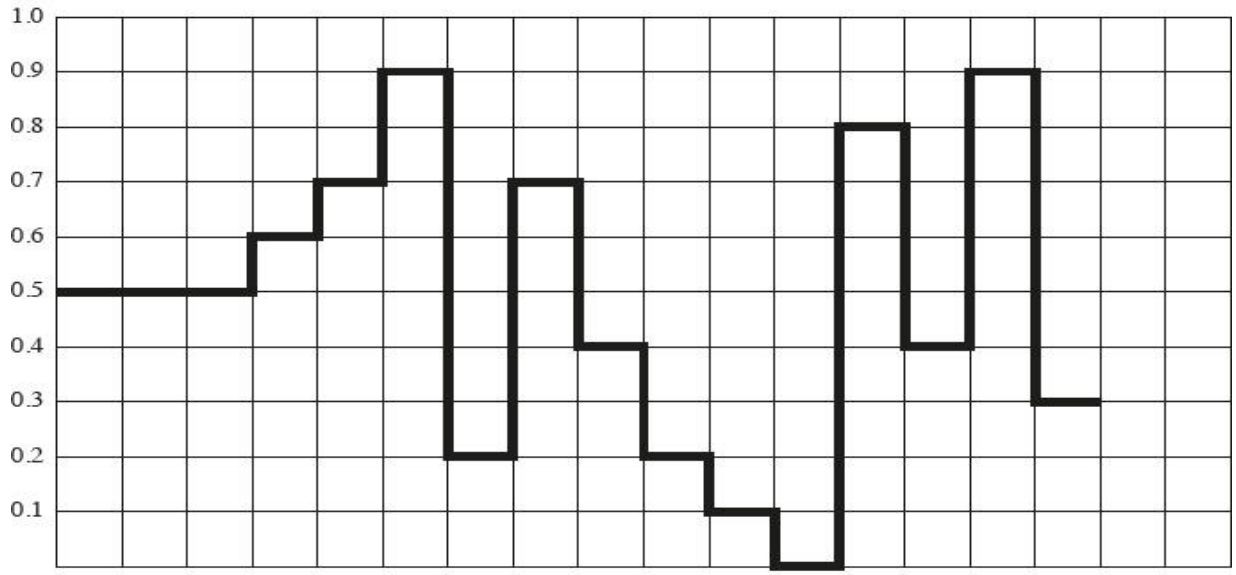
$$Y_i = Y_0 + \sum_{j=1}^{i} (\Delta Y)_j$$

where Y_0 is an initial value. The resulting Y_i is then multiplied by the incremental input $(\Delta X)_i$. Since these incremental values are restricted to $(\Delta X)_i \in \{-1; 0; 1\}$ this multiplication step is reduced to either addition or subtraction. The result is then accumulated in the second accumulator yielding

$$R_i = R_0 + \sum_{j=1}^{i} Y_j (\Delta X)_j - \sum_{j=1}^{i} (\Delta Z)_i.$$

The output of this integrator is not R_i but instead an incremental value, $(\Delta Z)_i$ representing the over- or underflow of this second accumulator stage. $(\Delta Z)_i$ can now be used as an input signal for other computing elements of the DDA. The behavior of such an integrator is depicted in →figure 12.2: The input function and its representation as a sequence of increments and decrements $(\Delta Y)_i$ is shown in the upper graph. The output of the second accumulator is shown in the middle graph. Accumulator overflows generate a corresponding incremental output signal $(\Delta Z)_i$ representing the result of the integration, which is shown in the bottom graph.


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

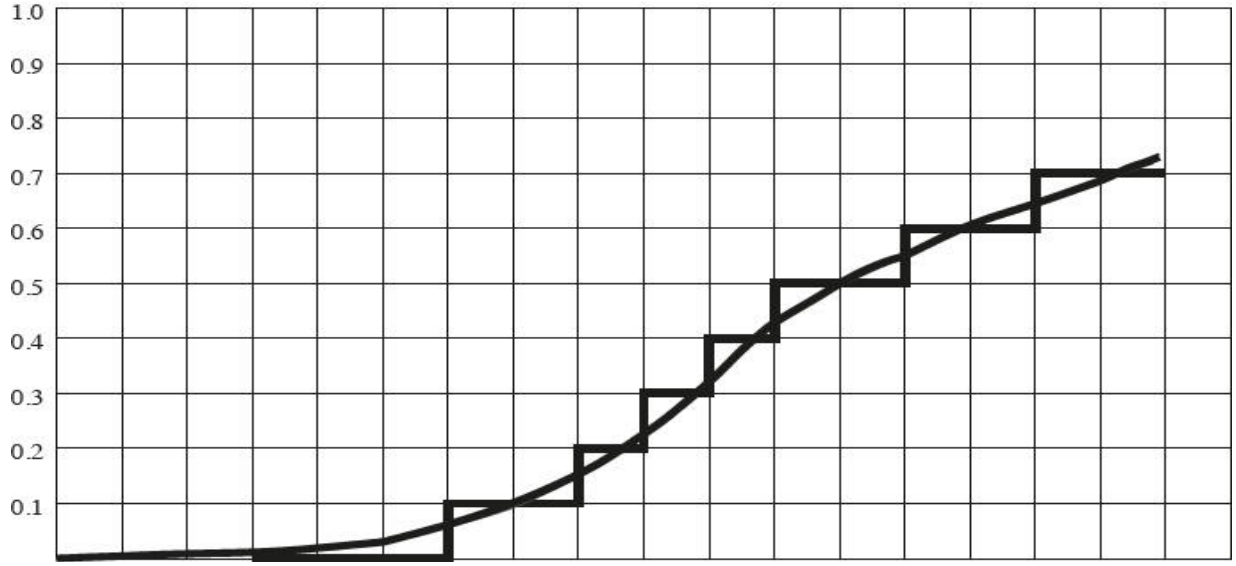

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

Figure 12.2. Operation of a DDA integrator (cf. [→Michels 1954, p. 4])

Although the use of incremental values simplifies the implementation of a DDA and thus saves cost, this representation limits the maximum rate of change of variable values in a computation. To overcome this restriction, additional time-scaling of the computer setup may be required, which may result either in non-real-time operation or very high clock rates of the DDA.

The symbol of a DDA integrator is shown in →figure 12.3. A typical setup of a DDA yielding the integral over a function is shown in →figure 12.4. The first integrator is fed with the incremental input values d*X* and d*Y* yielding an incremental output signal d*Z*, which is accumulated in a second integrator resulting in the actual value of the integration. The missing second incremental input of this integrator is assumed to be +1.[8,9]



Figure 12.3. Symbol of a DDA integrator

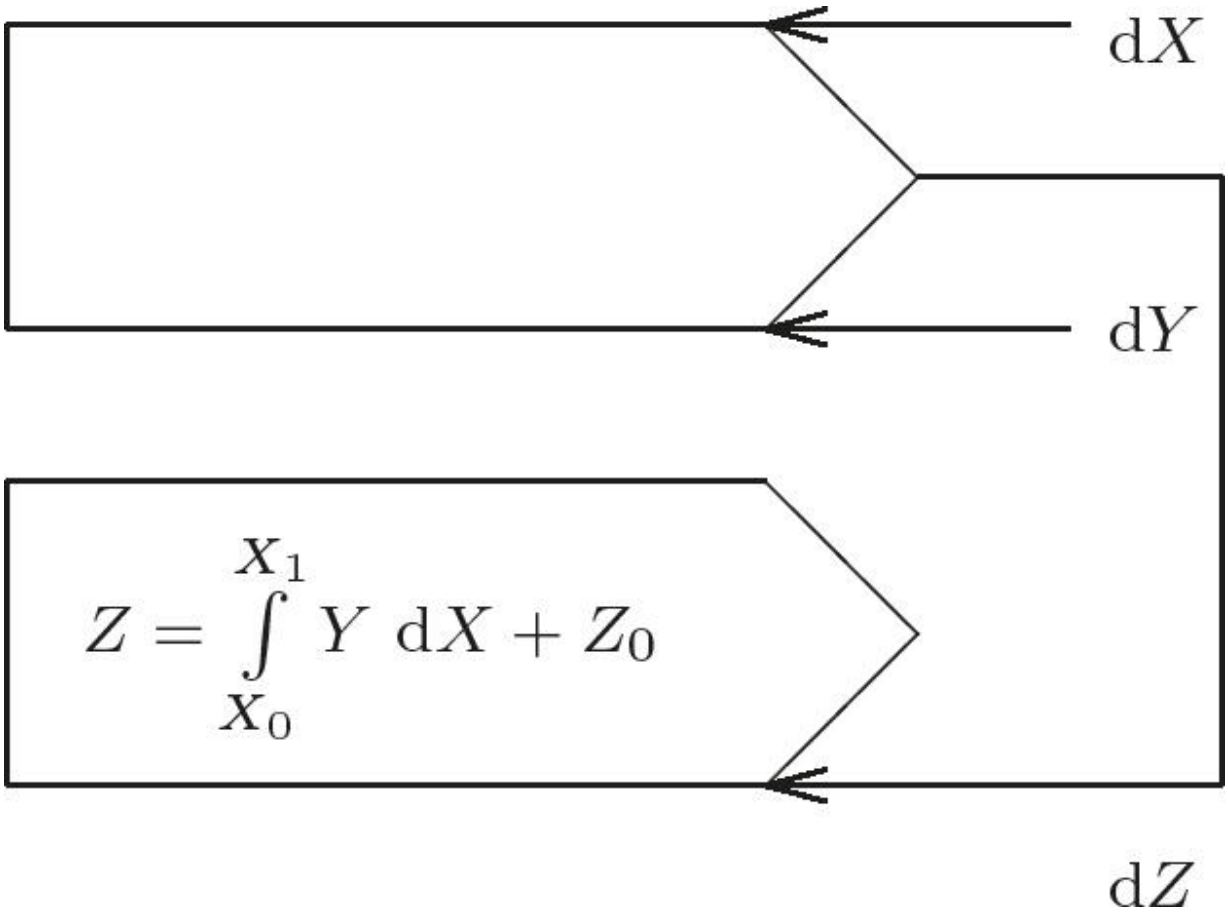


Figure 12.4. Explicit computation of a simple integral (see [→Bendix 1954, p. 7])

Integrators in a DDA have an even more central role than those of a traditional analog computer. While the latter uses free or open operational amplifiers to create implicit functions by means of a function generator in the feedback path, DDAs use specially configured integrators for this purpose. These are called *Servos*.[10]

12.1.2 Servos

As its name implies a *servo* is a computing element that is typically used to minimize some error term in a feedback loop

arrangement (this is then called a *servo loop*), which can be used to generate implicit functions. Typical servos are based on DDA integrators but feature only one incremental input $(\Delta Y)_i$ and one accumulator. The output $(\Delta Z)_i$ of this element is defined as

$$(\Delta Z)_i = \begin{cases} +1, & \text{if } Y_i > 0, \\ 0, & \text{if } Y_i = 0, \\ -1, & \text{if } Y_i < 0. \end{cases} \tag{12.1}$$

→Figure 12.5 shows the symbol of a servo element. Some implementations allow for an additional initial value input, which has been omitted here. Using such an initial value, the threshold level of the servo can be set to an arbitrary value ≠ 0 making it possible to use the servo as a generalized decision element.

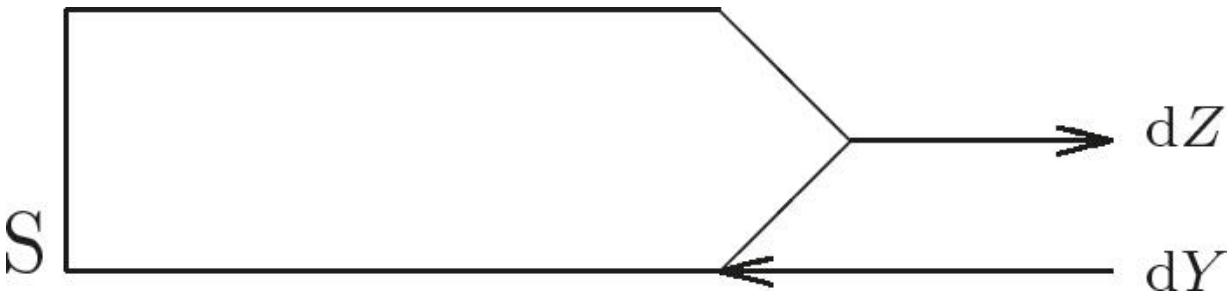


Figure 12.5. Symbol of a DDA servo element

→Figure 12.6 shows a simple application example of a servo element: Using a function generator, an implicit function based on the condition $F(U, V) = 0$ is generated.[11] The input of the servo is the incremental output signal $dF(U, V)$ of the function generator circuit. The output signal of the servo, defined by (→12.1), is in turn used as one input to the function generator while its second input is fed by dV from some external source.

The servo tries to drive the output of the function generator to 0 thus yielding the desired implicit function.

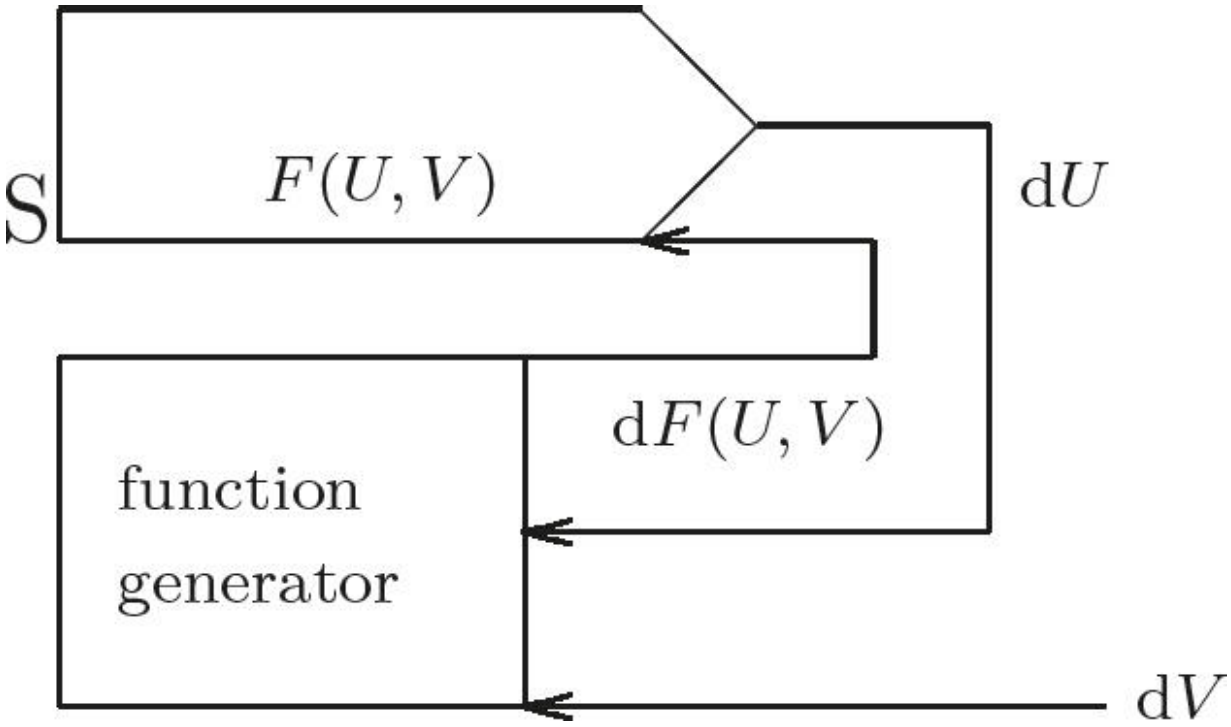


Figure 12.6. Application of a servo element (see [→Bendix 1954, p. 18])

12.1.3 Summers

Typical incremental DDAs, like the *Bendix D-12* described in →section 12.4.2 reduce the addition of two incremental values d*U* = d*V* + d*W* to solving the equation

$$V + W - U = 0. \tag{12.2}$$

A summer can thus be realized by using a modified servo element featuring more than one incremental input. →Figure 12.7 shows the symbol of a DDA summer having two inputs d*W*, d*V*, an output d*U*, which also feeds a third input after being sign

reversed. According to equation (→12.2) d*U* represents the desired sum.

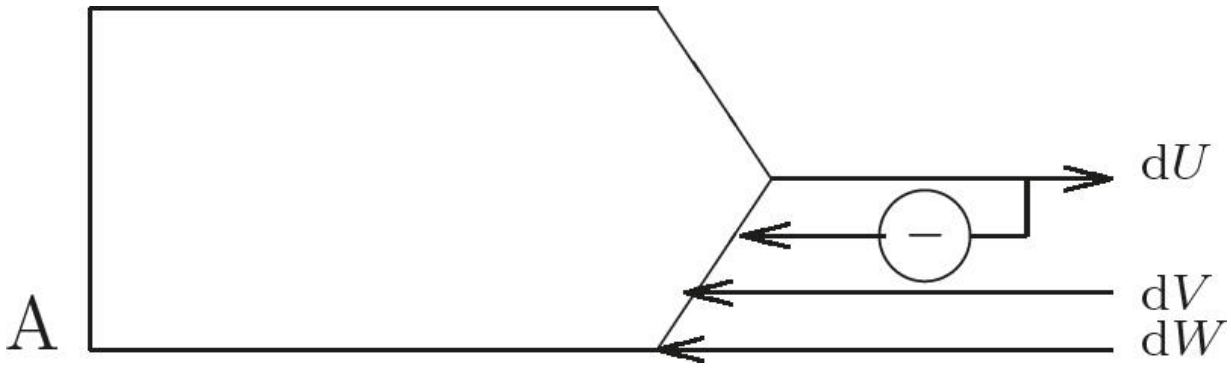


Figure 12.7. Symbol of a DDA summer

12.1.4 Additional elements

In addition to integrators, servos, and summers, typical incremental DDAs often feature circuit elements like *output multipliers*, which multiply incremental values by fixed rates thus acting as coefficient units,[12] function generators, which are often based on table-lookup techniques, etc.

Since integration can be performed with respect to any variable and not just time *t*, multiplication in a DDA is most often based on two integrations as described by equation (→2.1) in →section 2.5.4. An example of this is given in the following section.

12.2 Programming examples

→Figure 12.8 shows a DDA-setup to generate an exponential function, which is based on a single integrator with its output fed back to its incremental d*Y* -input. Its incremental d*X*-input is fed

with the incremental values representing the desired exponent value.

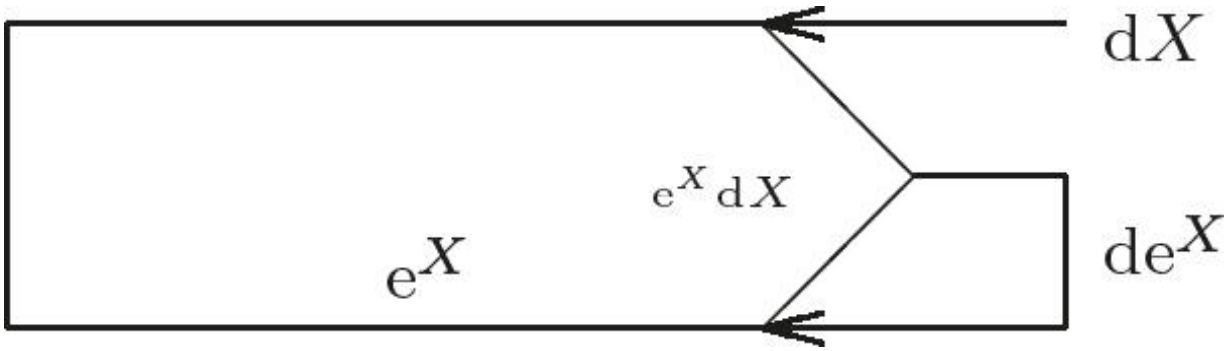


Figure 12.8. Computing e^x with a DDA (cf. [→Bendix 1954, p. 9])

This function can be used in conjunction with a servo element to implicitly generate a logarithm function as shown in →figure 12.9. The servo at the top has two inputs: The incremental input value −d*V* determining the argument of the logarithm function and a second input, which is fed with the output of an exponential function generator as described above. At the output of the servo a signal representing d(ln *V*) is available. This is due to the fact that the servo tries to ensure $F(U, V) = e^U - V = 0$ holds, so $e^U = V$ yielding $dU = d \ln V$.

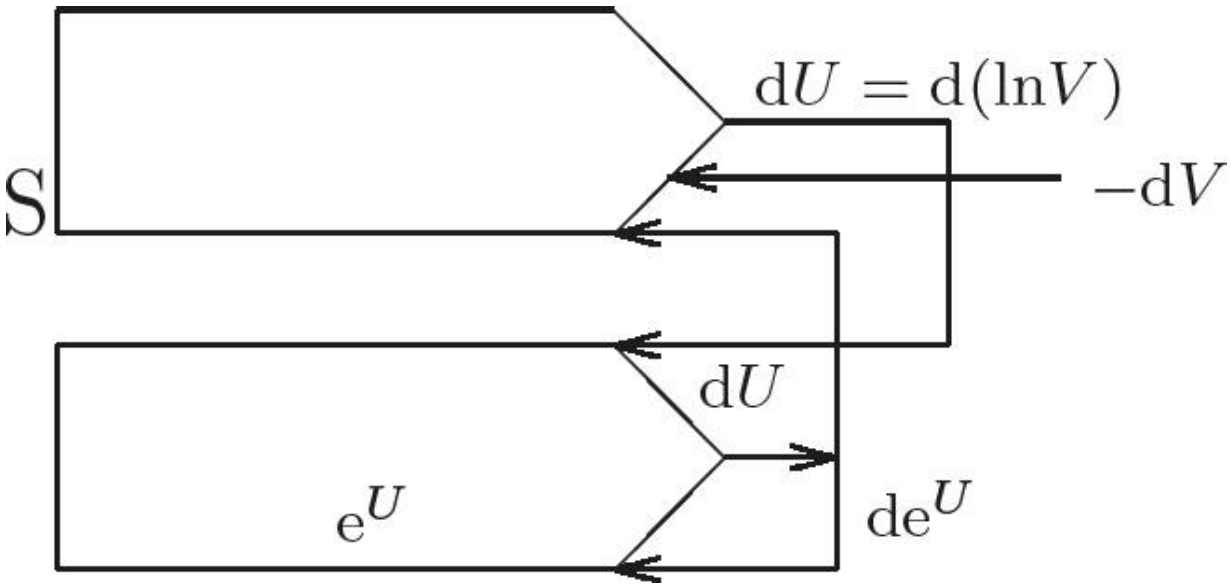


Figure 12.9. Implicitly generating a logarithm function (see [→Bendix 1954, p. 19])

Generating trigonometric functions can be done equally simply by solving an auxiliary differential equation of the form $\ddot{y} + \omega^2 y = 0$ as shown in →figure 12.10. In contrast to a traditional analog computer implementation the sign reversal operation does not require an additional inverter since it can be done directly using incremental values.

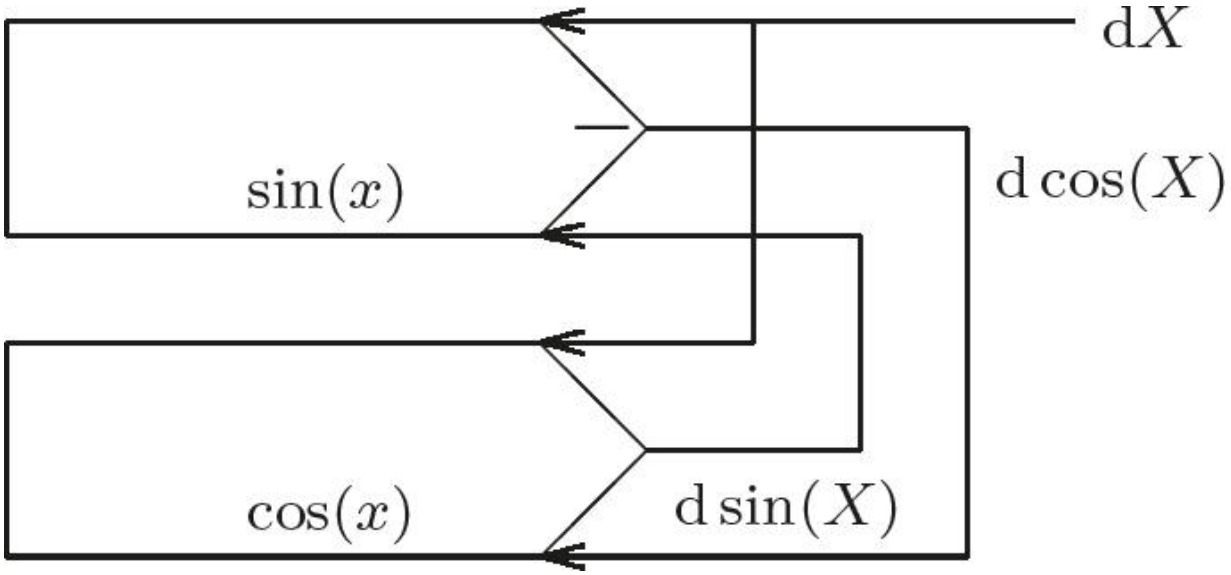


Figure 12.10. Generating a sine-/cosine signal pair (cf. [→Bendix 1954, p. 10])

→Figure 12.11 shows a typical DDA setup to perform the multiplication of two incremental variables d*V* and d*U*. As already described in →section 2.5.4, the product rule can be used to perform a multiplication of two values if integration is not limited to time being the only free variable. The two integrators shown on top of the figure perform the actual multiplication of the incremental input values yielding an incremental output that is fed into a third integrator yielding the desired product.

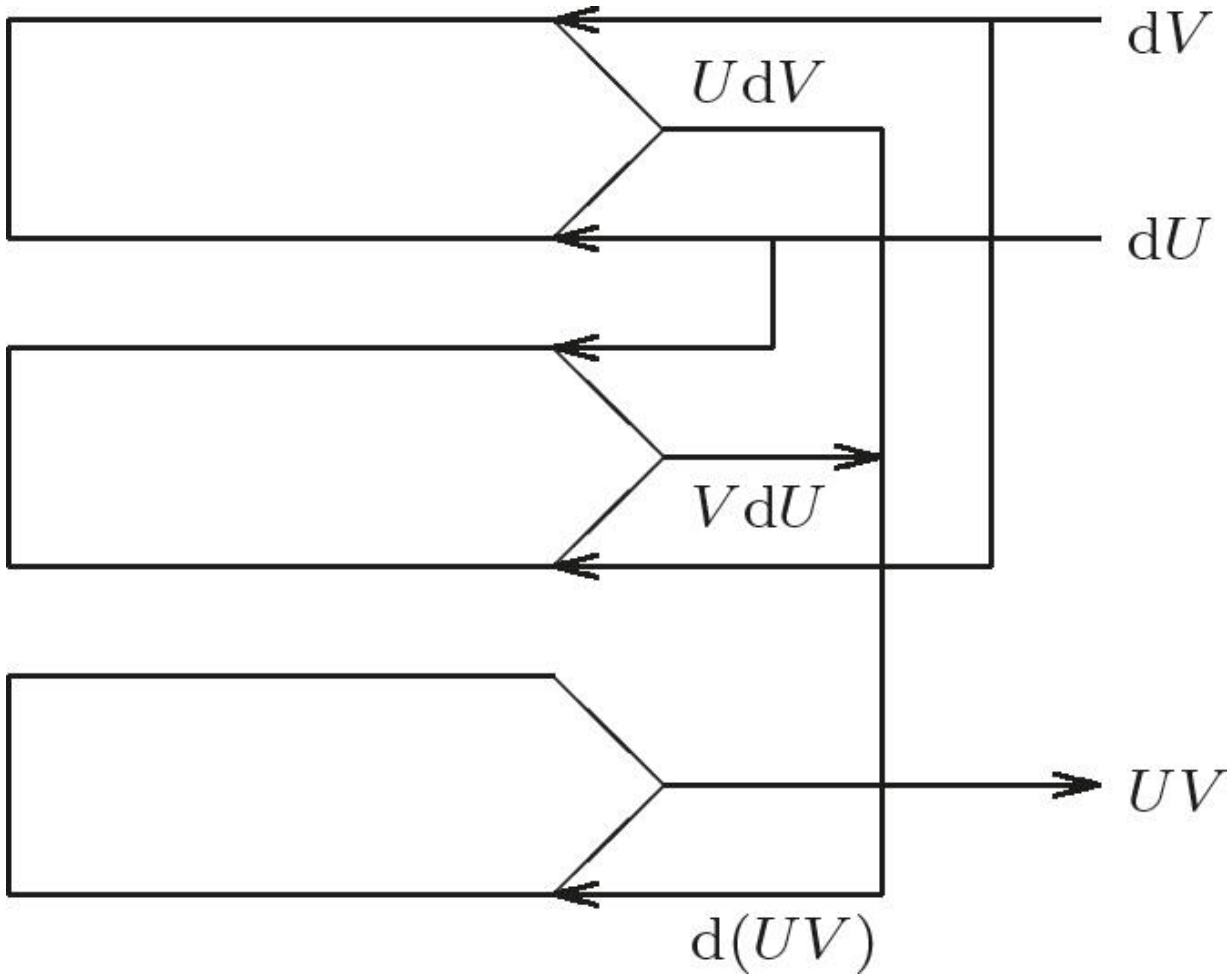


Figure 12.11. Multiplying two incremental values using two coupled integrators (see [→ Bendix 1954, pp. 10 f.])

Additional programming examples of varying complexity for DDAs can be found in [→ Bendix 1954], [→ Palevsky 1962], and [→ Forbes 1972], the latter focusing on the solution partial differential equations by means of DDAs.

12.3 Problems

The advantages of DDAs, principally their ability to use any variable as the free variable of integration, their immunity to drift and aging effects of components as well as the advantage

of using basically arbitrarily long bit sequences to achieve any degree of precision required, come at a price.

A simple bit-serial incremental DDA working with values −1 and +1, and featuring a word-length of *n* bit exhibits a basic time-constant of

$$t = \frac{2n}{f}$$

with *f* denoting the systems clock frequency. This limits the real-time capabilities of such a basic DDA considerably.[13] Making *n* smaller would yield a faster response time but would also diminish the precision of the machine, so increasing *f* is often the easiest way to speed up an incremental DDA without sacrificing precision. Another approach is to allow incremental values of a larger domain – ideally floating point numbers could be used throughout a DDA.[14]

One source of error is the integration scheme used in a particular DDA. Using just simple accumulators implementing the rectangle rule is not sufficient for more demanding applications. Other techniques, implementing at least a trapezoidal rule, yield better results but complicate the hardware implementation considerably. In addition to that, a DDA cannot speed up a calculation as easily as a traditional analog computer where the integrator capacitors can be changed to speed up or slow down a calculation.

The limited precision and speed of incremental DDAs make them unsuited to many applications as shown by Fred Lesh in [→ McLeod et al. 1958/3, pp. 488 f.]. In this study a problem involving the computation of rocket trajectories is treated on a DDA as well as on a traditional analog computer. It turned out that the analog computer took only 30 seconds for a solution while the DDA was running for 3 minutes, 30 minutes or up to 5 hours depending on the integration step size. While the analog

computer yielded results with an error of only about 2%, compared to the exact solution, it took the DDA 30 minutes to product a solution with an error of 20%.

12.4 Systems

Over the years many different DDAs have been implemented. All of these are of the incremental type, with many of them being based on sequential operation. Most of these systems are programmed by specifying the connections between the computing elements in binary form on some storage system with *CORSAIR*, which uses a traditional patch panel, being an exception. Only one relevant system, *TRICE*, was of the parallel type, using many individual computing elements and a patch panel.

12.4.1 MADDIDA

Shortly after the end of World War II, in March 1946, *Northrop* started the development of a subsonic cruise missile designated *MX-775A*, which was later designated *SSM-A-3 Snark*.[15] This system should be able to hit a target at a distance of up to 5,000 miles with a precision of 200 yards – much better than the German vengeance weapons V1 and V2.[16] To achieve this level of accuracy, a new guidance system, based on astronavigation, was developed. Using a built-in telescope, azimuth, and elevation angles for sighting predetermined stars were determined to get a navigational fix. This system was then coupled with an inertial guidance system.[17]

While Helmut Hoelzer used an analog electronic analog computer in the guidance system of the A4 rocket, the long time-of-flight and the required high precision of Snark required a

digital approach to minimize computational errors. Floyd Steele[18] developed the idea of implementing an analog computer using only digital elements, which was called *DIDA*, short for *DIgital Differential Analyzer*.

Donald E. Eckdahl,[19] Richard Sprague,[20] and Floyd Steele[21] headed a team that developed a laboratory prototype of a DIDA, which was called *MADDIDA*.[22] This machine, actually built by *Hewlett Packard*,[23] was based on a magnetic drum like those pioneered in sophisticated radar systems of World War II. The tracks of this drum housed bit-serial accumulators, one for each of the integrators of the machine, as well as data representing the interconnection of the computing elements. The actual computations were performed on a single, central ALU that was time-shared between the various computing elements used in any given setup. This remarkable prototype, which only used 53 vacuum tubes and 904 diodes and was built in "*less than 600 man hours*",[24] is shown in →figure 12.12.[25]

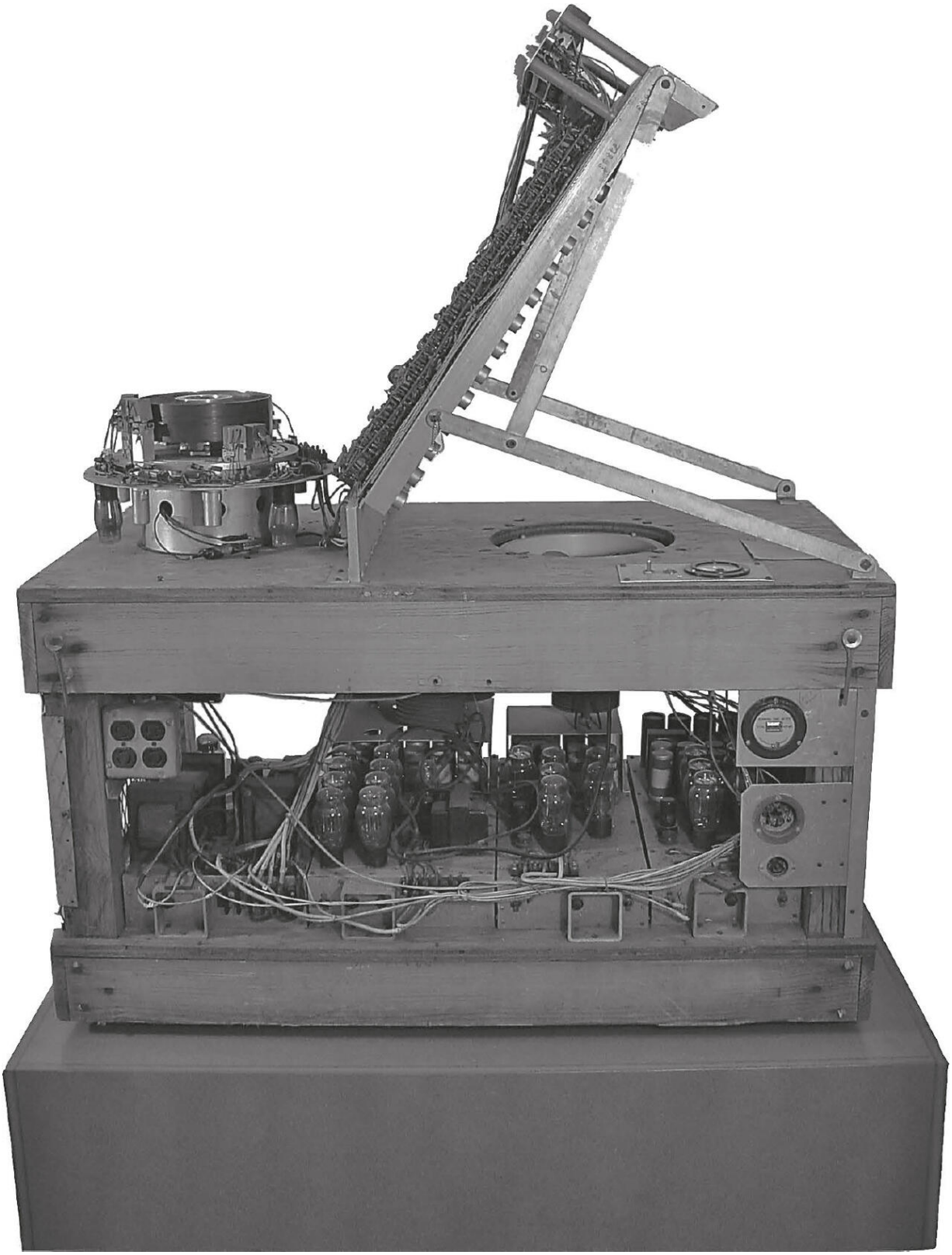


Figure 12.12. MADDIDA prototype (with permission of the *Computer History Museum*, Dag Spicer)

Clearly visible on the left is the small magnetic drum with its read/write heads and the amplifier tubes hanging down. The heart of the machine, the digital logic elements implementing the bit-serial ALU with its surrounding control logic is visible on the right in its raised service position.

In 1950 MADDIDA was demonstrated to John von Neumann,[26] who was excited to see the system computing Bessel functions and characterized it as "*a most remarkable and promising instrument*".[27] Although no actual guidance system for Snark resulted from the work on MADDIDA,[28] it became the ancestor of most DDAs to follow. A direct successor was built in a small production batch and used mostly for aerospace research. → Figure 12.13 shows two of these systems.[29]

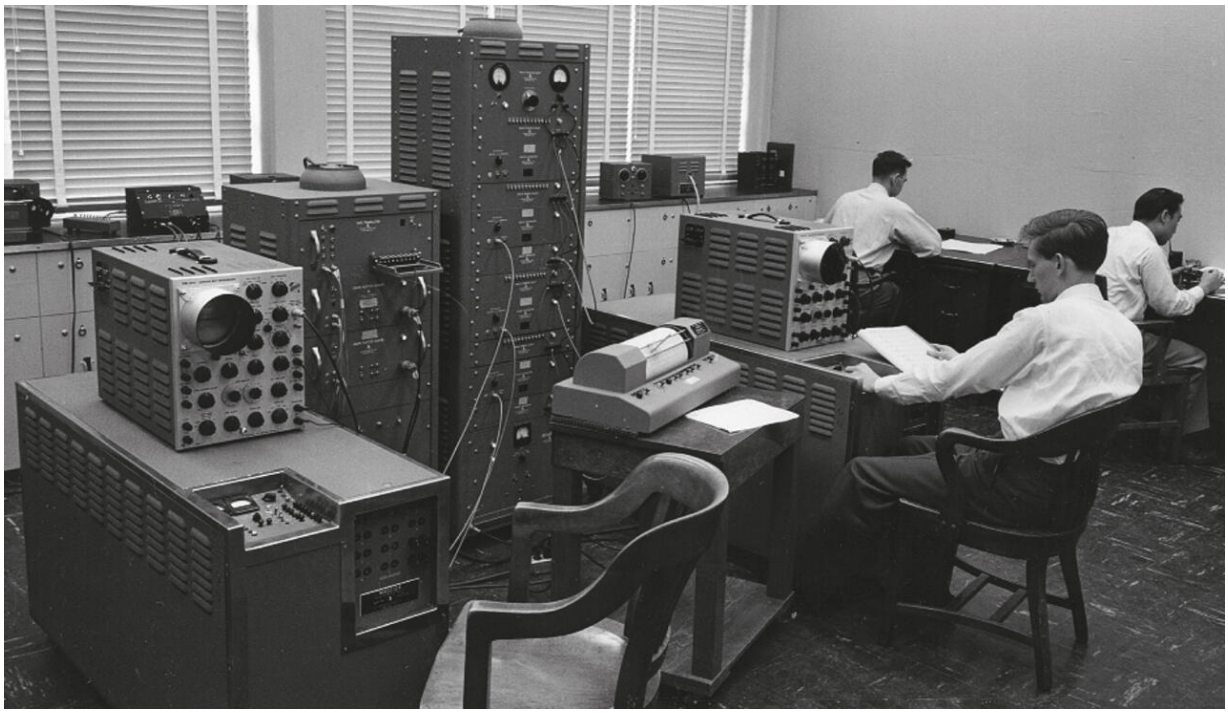


Figure 12.13. Two MADDIDA systems in use at the Navy Electronics Laboratory (NEL, file number E1278)

12.4.2 Bendix D-12

In the early 1950s the *Bendix Corporation* developed and marketed the *D-12*, a DDA quite similar to MADDIDA. →Figure 12.14 shows the DDA on the right with its operator desk on the left. On this desk, from left to right, are the control console, an *xy*-plotter, and a console typewriter.[30]



Figure 12.14. Bendix D-12 DDA (cf. [→Bendix, p. 2])

The D-12, operating on a serial stream of decimal values represented by four bits each,[31] is based on a magnetic drum holding the integrator registers as well as the necessary information about signal sources and destinations. The system can work in two modes with either 30 integrators operating at 200 iteration steps per second or with 60 integrators running at half this speed.[32] Apart from the simple integration scheme described in →section 12.1.1, the D-12 integrators can also work based on

$$\frac{(\Delta Y)_i + (\Delta Y)_{i+1}}{2}$$

yielding better results than those obtained by the rectangle rule.[33]

While MADDIDA was basically a stand-alone system, the D-12 could be coupled directly to a Bendix *G-15D* stored-program digital computer as shown in →figure 12.15. On the left the G-15D stored-program digital computer can be seen with the DDA on its right. The enclosure housing the DDA in this configuration is considerably smaller than that of the stand-alone system since the G-15D and the D-12 share a common magnetic drum, which is part of the G-15D stored-program computer, thus saving a considerable amount of hardware on the DDA part of the machine.



Figure 12.15. Bendix D-12 coupled with a Bendix G-15D stored-program computer (see [→KLEIN et al. 1957, p. 1105])

A typical program for the D-12 is shown in →figure 12.16. The four integrators, labeled (4) to (8), solve the VAN DER POL differential equation[34]

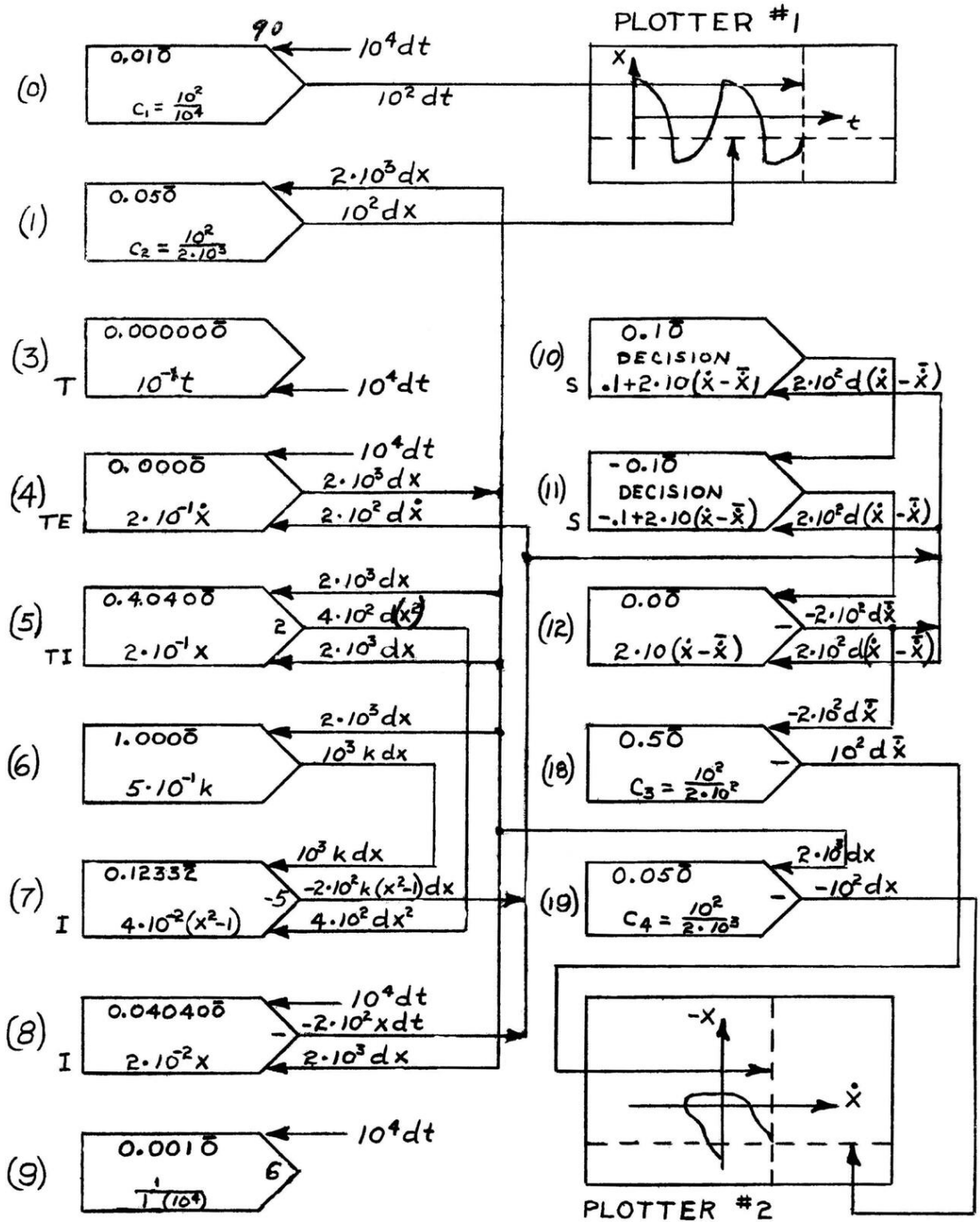


Figure 12.16. DDA setup for solving $\ddot{x} + k(x^2 + 1)\dot{x} + x = 0$ (see [→Bendix 1954, p. 65])

$$\ddot{x} - k(x^2 - 1)\dot{x} + x = 0.$$

Integrator (9) controls the printout of the values generated by the integrators (3), (4), and (5) on the console typewriter. The integrators (0) and (1) generate the value-pair ($x(t)$, t) that is used to control plotter #1. A phase-space plot is generated on plotter #2 based on the signal pair (≤(t), $x(t)$), which is generated by the integrators (18) and (19). The remaining three integrators (10), (11), and (12) perform some post-processing of the signals required for the phase-space plot.

This DDA setup is implemented by the program listing shown in →figure 12.17. Each line corresponds to one computing element, the number of which is specified in columns two and three. The various initial values used for the integrators is specified by the digits in columns six to 14.[35] Columns 16 to 20 specify integrator control:

```
          1111111111222222222233333333334444444445
12345678901234567890123456789012345678901234567890
----------------------------------------------------
 00   0010       11111 090
 01   0050       11111 04
 03   0000000    11211       90
 04   00000      31211 90    07 08
 05   040400     21212 04    04
 06   10000      11111 04
 07   012332     211-5 06    05
 08   0040400    211-1 90    05
 09   00010      11116 90
 10   010        91111       07 08 12
 11  -000        91111 10    07 08 12
 12   000        111-1 11    07 08 12
 18   050        111-1 12
 19   0050       111-1 04
```

Figure 12.17. D-12 example program (cf. [→Bendix 1954, p. 69])

Column 16: This column controls the mode of operation for the integrator. The values “1” to “4” select integrator-mode with one out of four different integration schemes, while a value of “5” switches the integrator into summing mode and “6” specifies a servo.
Column 17: Possible values are “1” or “2”, selecting normal or automatic reset mode of the integrator.
Column 18: Integrators flagged with a value of “1” in this column are running in normal (silent) mode while a value of

“2” causes the contents of the second accumulator register to be printed on the console printer after each iteration.
Column 19: This can be used to select an automatic sign-reversal for the incremental output signal of an integrator. “1” specifies normal operation while “-” causes a change of sign.
Column 20: Using this column a constant output multiplication factor of 1, 2, or 5 can be selected. The special value “6” selects the corresponding integrator as control circuit for the typewriter.[36]

Columns 22 and 23 specify the source of the d*X* input while columns 25 and 26 select the source of the initial value of the integrator. Up to 8 addresses of source elements for the d*Y* inputs can be selected by values in the columns starting at 28.[37] The *UNIVAC incremental computer*, see [→Remington 1956], a system developed for the *Operational Flight Trainer* described by [→Gray 1958] and *STARDAC*, see [→Milan-Kamski 1969], are similar sequential incremental DDAs, to name just a few.

12.4.3 CORSAIR

An interesting variant of such a sequential incremental DDA is the *CORSAIR* system,[38] developed at the *Royal Aircraft Establishment*, Farnborough, starting in the late 1950s. This system is remarkable in several respects: It is fully transistorized, uses core memory,[39] and is programmed by a patch panel similar to those found on traditional analog computers. The name CORSAIR consists of three parts: “COR” representing the core memory, “S” for the surface barrier transistors used in construction, and “AIR” for its aircraft association. →Figure 12.18 shows the overall system. On the left is a plotter, the DDA is in the middle of the picture with its power supply on the right.

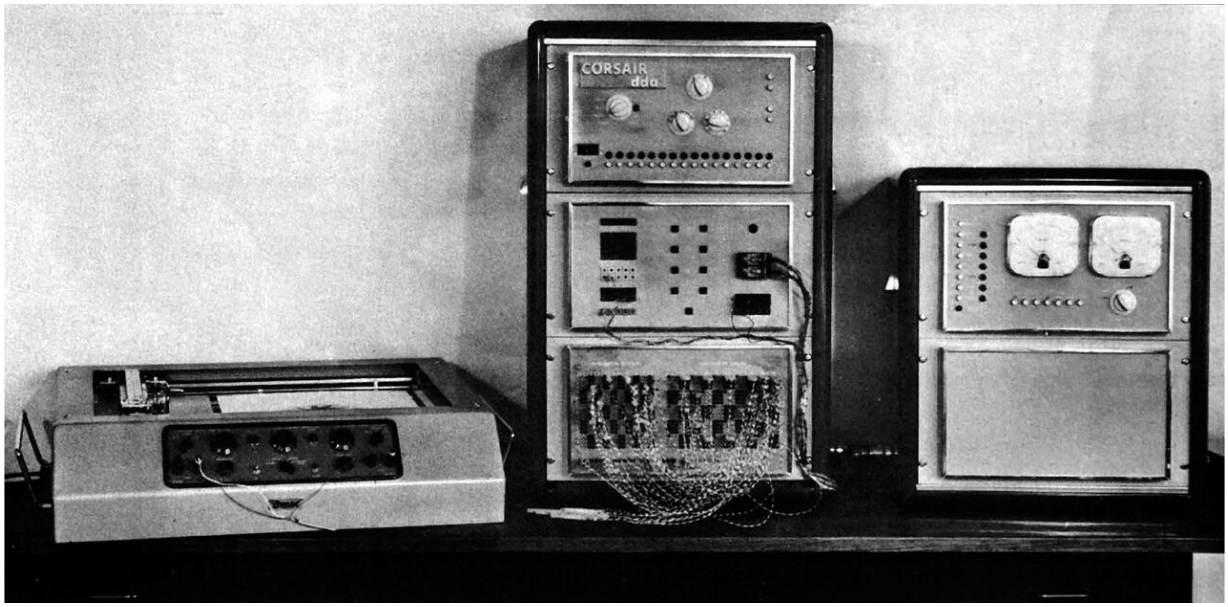


Figure 12.18. Typical CORSAIR installation (see [→Owen et al. 1960, p. 33])

The system features a 500 kHz bit rate resulting in 500 integrations steps per second.[40] The core memory consists of a 50 × 32 matrix of tiny ferrite cores, capable of storing fifty value pairs, each associated with one computing element. Each connection made on the central patch panel consists of two wires, one for each of the two incremental bit-serial signals.[41]

12.4.4 TRICE

A radically different and far more complex and costly implementation of a DDA was developed by *Packard Bell* in 1958. In contrast to the systems described above, which operate sequentially, this machine, named *TRICE*, was the first parallel DDA. This required a fully transistorized implementation allowing a high packing density of the circuitry while minimizing energy consumption and thus simplifying the problem of cooling the densely packed electronics.[42]

→Figure 12.19 shows a fully expanded TRICE system. From left to right it consists of four racks housing the various computing elements, a wide rack containing the central patch panel and the operator console, a rack with a *PB-250* bit-serial stored-program digital computer, which controls the overall system, loads initial values into integrators, etc. Next to this is a rack containing a paper-tape reader/puncher and various ADCs and DACs allowing TRICE to be coupled with a traditional analog computer or to feed it with data from analog signal sources. The two racks on the right contain additional computing elements.

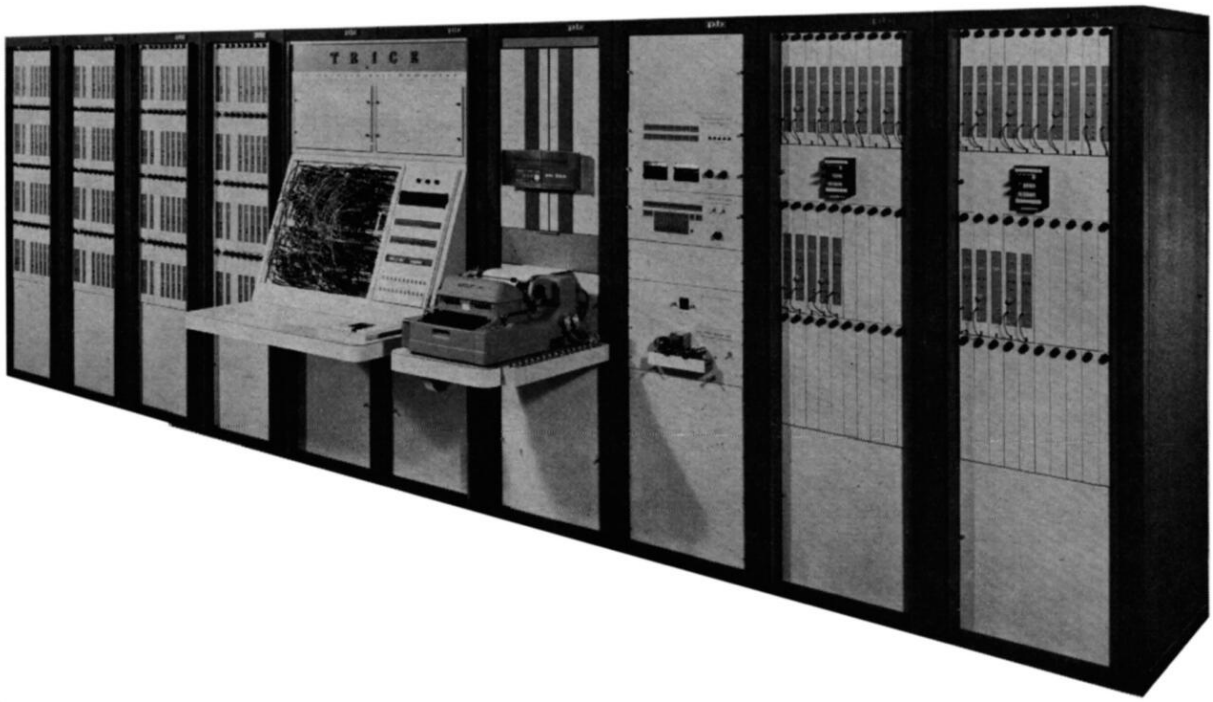


Figure 12.19. Fully expanded TRICE system (cf. [→AMELING 1963/2, p. 30])

TRICE ran at a clock rate of 3 MHz, which was quite remarkable for such a large and early transistorized system. Its word length is 30 bits[43] while values are represented in binary and not in

excess-three code decimal form as in the D-12, allowing a maximum iteration frequency of 10^5 s^{-1}.[44] Due to the completely parallel operation of TRICE, repetitive operation is possible with a frequency of up to 100 Hz, thus challenging contemporary analog computers. TRICE has been described as the "*most advanced DDA that was ever built*."[45]

TRICE contains summers, integrators with only one d*X* and d*Y* input,[46] special summing-integrators, constant value multipliers, variable multipliers, servos, and a variety of ADCs and DACs. A typical TRICE module, a multiplier, is shown in →figure 12.20. This multiplier consists of two dedicated integrators implementing the product rule technique described earlier.[47]



Figure 12.20. TRICE multiplier (cf. [→ Ameling 1963/2, p. 30])

A typical application of TRICE is described in [→ Ameling 1963/2, pp. 40 f.]: A complex flight-dynamics problem is solved on a TRICE system coupled to a large analog computer. While TRICE performs those parts of the simulation requiring utmost precision such as trajectory calculations and the like, the analog computer implements operations requiring extreme high speed but won't introduce errors into operations such as coordinate transformations, etc. The TRICE system used in this particular simulation contained 42 integrators, 11 multipliers, 5 constant value multipliers, 5 servos, 6 ADCs, and 13 DACs, while the

analog computer featured 200 operational amplifiers, 14 multipliers, 5 function generators, and 20 servos for coordinate transformation and rotation.

Due to the large amount of hardware necessary to implement a system like TRICE, only a few of these systems were eventually built. According to Tom Graves three TRICE installations were used at NASA's flight simulation laboratory in Houston. While much simpler systems such as the Bendix D-12 were not successful in the long run due to their inherently slow speed of calculation, TRICE, too, wasn't a market success due to its complexity and associated cost.

The legacy of DDAs could be found long after the demise of these marvelous machines in line drawing algorithms for digital computers and in *CNC*[48] machines, where DDA techniques were used to perform surface and line interpolation for controlling cutter paths.

13 Stochastic computing

In a series of lectures held at the California Institute of Technology (Caltech) from January 4th to January 15th 1952, John von Neumann wondered how one could build "*reliable organisms from unreliable components*". He advocated treating (unavoidable) errors in computer elements by thermodynamical methods instead of the classic techniques typically used in digital computers. [→ Manohar 2015, p. 119] describes this approach aptly as "*embracing, not competing the device physics*".

One idea to introduce more robustness in digital circuitry was the *multiple line trick*.[1] Here, instead of a single line transferring a bit, n lines in parallel are grouped, each transmitting the same value of 0 or 1. If at least Δn of these lines are set to 1 the overall value is assumed to be 1 by the receiver, 0 otherwise. Despite the obvious shortcomings of such an arrangement with respect to its complexity, it could easily tolerate the failure of multiple lines in a group without disabling the overall digital circuit it is part of.[2]

Later in this lecture series von Neumann discussed different ways of representing and processing information in a fault tolerant way by taking biological neural networks as an antetype. He notes that[3]

> "*message pulse trains seem to convey meaning by certain analogic traits (within the pulse notation – i. e., this seems to be a mixed, part digital, part analog system), like the time density of the pulses in one line, correlations of the pulse time series between different lines in a bundle, etc.*"

Obviously, small errors such as transients or glitches in a system employing this type of value representation would only

superficially impair the operation of the overall system.[4]

In 1967 Brian R. Gaines[5] elaborated on the idea of using a stochastic bit sequence for representing values, for which the term *stochastic computing* was coined.[6] The *unipolar stochastic number* is a simple example for such a value representation. Given a machine unit $m \in \mathbb{R}^+$ the two states of a bit can be interpreted as 0 and m respectively. Accordingly, a continuous variable $x \in [0, m]$ can then be represented as the arithmetic mean of a sequence of a stochastic bitstream as shown in the following two examples:[7]

$$01010101\ldots \triangleq \frac{m}{2}$$
$$001001001\ldots \triangleq \frac{m}{3} \text{ etc.}$$

Ideally, the bit sequence would not show any regularity but instead be of (pseudo-)random[8] nature with the probability of a 1 in the bitstream being

$$P_1(x) = \frac{x}{m}.$$

This idea can be easily extended to *bipolar stochastic numbers* by defining

$$P_1(x) = \frac{1}{2}\left(1 + \frac{x}{m}\right).$$

An alternating sequence of 0 and 1 would then represent $x = 0$, all zeros would correspond to $x = -m$ and all ones to $x = m$.[9]

Based on such a number representation scheme, basic operations can then be implemented using logic gates to build an actual computer.[10] Maybe the most fascinating operation is multiplication, which can be implemented by a single two-input *AND gate*,[11] which combines the two stochastic bitstreams

representing the numbers to be multiplied. [→ POPPELBAUM 1979, p. 1] notes

> "*To multiply two numbers with 7 bit (i. e., 1% accuracy a 2000 transistor microprocessor needs about 10 clock-periods. A stochastic computer gives the same accuracy (within one standard deviation) in* 10^4 *clock periods – but it only uses a 2 transistor AND for the operation.*"[12]

It is interesting to note that although the multiplication of two independent stochastic bitstreams is simple, squaring a value is a bit more complicated since an AND gate fed with the same bitstream applied to both of its inputs just yields an identical bitstream at its output. To implement a squaring function one input of the AND gate is fed with a delayed version of the bitstream representing the value to be squared. This delay can be as short as one bit-time and implemented as a simple flip-flop circuit.[13]

Addition and subtraction are a bit more involved. Two bitstreams are added by yielding an output of 1 if a 1 is received at either input and storing a 1 when both inputs happen to by 1 until a free *slot* is available in the output sequence (i. e., until the next 0 in the output bitstream). Subtraction can be implemented similarly by subtracting a subtrahend 1 from the next 1 in the minuend stochastic bitstream.

Even an integrator is not that difficult to implement, as it just requires an up/down counter and some means to generate a stochastic output bitstream with $P_1(x)$ corresponding to its current counter value; this can be implemented using a comparator and a noise source.[14]

Apart from the very simple implementation of basic operations, stochastic computers have the characteristic trait that a rough approximation of some solution for a problem can be quickly obtained while computing times rapidly increase

when higher precision is required, a behavior called *progressive precision*.[15] In short, a stochastic computer features an extremely simple hardware implementation but at the price of low computational bandwidth and low precision.[16]

→ Figure 13.1 shows the *RASCEL*[17] computer, developed by John William Esch in the late 1960s.[18] This system represents values as fractions with numerator and denominator consisting of independent stochastic bitstreams. It contains a number of computing elements arranged in the tree-like structure visible on top of the computer. Each such element can be configured to perform either squaring, addition, subtraction, multiplication, or division. RASCEL could yield results spanning three to four orders of magnitude with an accuracy of 1% and a slew rate of 100 ms.

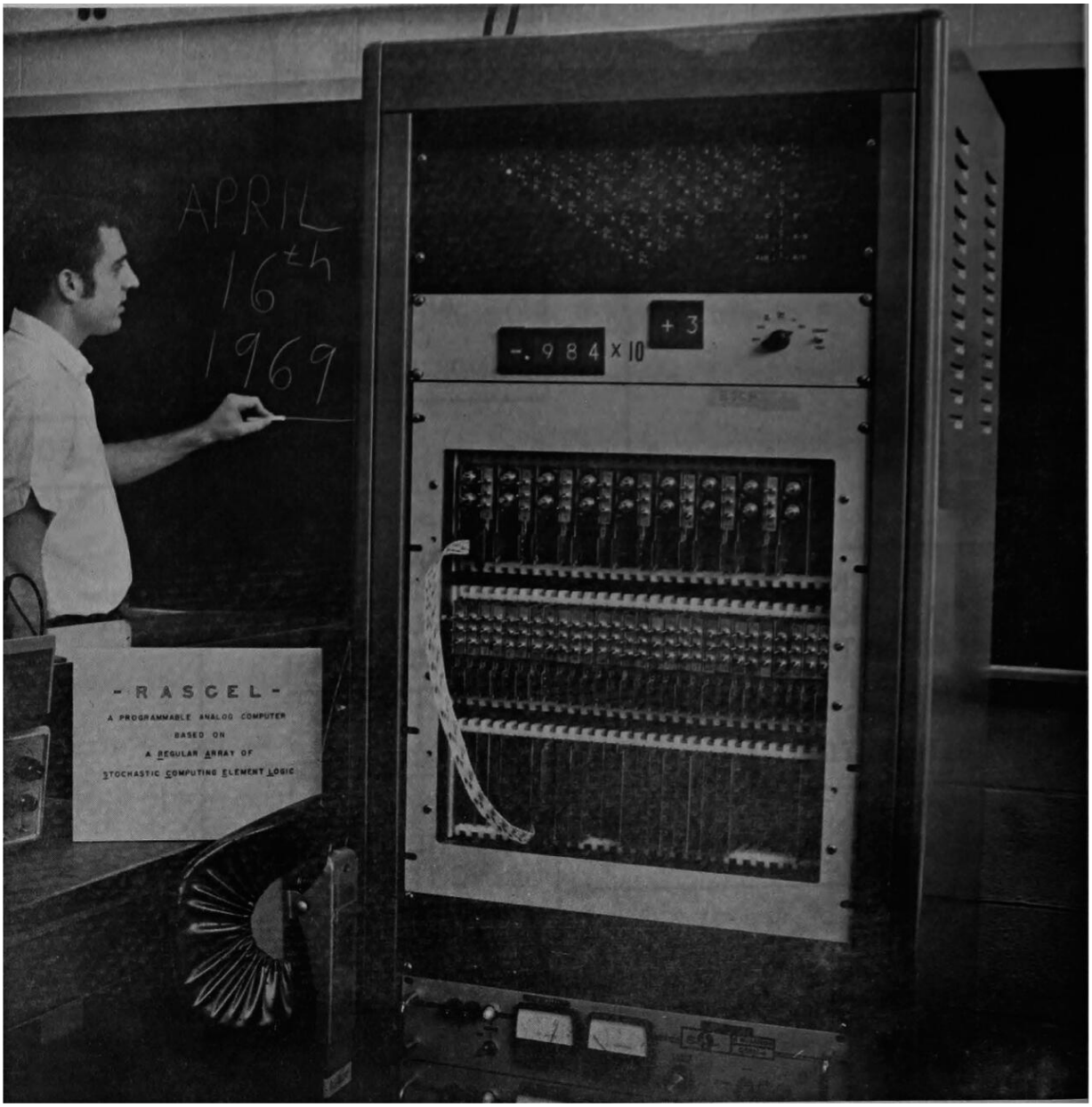


Figure 13.1. The RASCEL system in 1969 (John William Esch on the left), see [→ Esch 1969, p. 44]

The main advantage of stochastic computers is the simplicity of the individual computing elements, which are not only quite tolerant to jitter or skew in the various bitstreams but are also robust with respect to noise. Unfortunately, this is more than outweighed by several disadvantages:[19] First of all the precision

is very limited. 2^n different states of a variable could be represented in standard binary form by *n* bits. A stochastic computer would require a bitstream of 2^{2n} bits to represent a value with the same precision, which not only becomes quickly impractical but also very slow given the serial nature of such a bitstream.[20] Another disadvantage is the fact that it is not easy to generate a multitude of truly independent stochastic bitstreams, which is essential for a stochastic computer.[21]

This is reflected by the fact that the first, and at the same time last, symposium on stochastic computing took place in 1978 in Toulouse, France. Nevertheless, stochastic computing might experience a revival in the 21st century as it is suitable for implementing *artificial neural networks* (*ANN*s).[22]

As a side note it should be mentioned that it is possible to build analog computers using pulse position coded values, which is not too dissimilar to stochastic computers and spiking neural networks.[23]

14 Simulation of analog computers

Clearly everyone simulating an analog computer on a stored program digital computer is in a state of sin[1] as the main advantages of an analog computer (its extremely high degree of parallelism, speed, and energy efficiency) are forsaken by simulating its computing elements in a sequential fashion on a digital computer.

Nevertheless, simulating an analog computer on a digital computer can be justified in certain cases – such as education, the scaling of differential equations, or the treatment of problems involving variables of such vastly different magnitudes that scaling the problem for an analog computer would at least be very cumbersome if not next to impossible. It is also an option when there is no access to an analog computer at all or to one of appropriate size for a certain problem.

A typical example of such a problem is the classic simulation of bubble formation in the cooling loop of a nuclear reactor. The time constants involved differ by up to eight orders of magnitude while the radius of a bubble changes by five to six orders of magnitude. All of this in conjunction with variables to the fourth power makes the problem unsuitable for a classic analog computer.[2]

A problem like this could be solved quite efficiently on a parallel DDA such as TRICE, but machines like this were rare while digital computers became increasingly common in the early 1960s. Consequently, the idea of simulating an analog computer was self-evident. [→ Selfridge 1955] describes a system allowing "*a digital computer to operate as a differential analyzer*".[3] The digital computer used in this work was the *defense calculator*,

later called the *IBM 701*,[4] one of the earliest commercially available digital computers. Although this system was incredibly slow from today's perspective, it gave rise in the 1960s to the development of about 30 such simulation systems implemented in various programming languages on a variety of digital computers.[5]

One of the first practical systems was introduced in 1963 and became known as *MIDAS*.[6] Typical for the 1960s, MIDAS was operated in batch mode. A simulation run had to be prepared as a stack of punch cards, which were fed to the digital computer executing the simulation program and returning the results on fan-fold paper – not a very convenient or appealing mode of operation, particularly for people accustomed to the high degree of interactivity offered by analog computers.

In 1964 *PACTOLUS*[7] was introduced for the *IBM 1620*, a very small digital computer.[8] Since the 1620 was typically only used by a single user, it allowed for some degree of interactivity, which greatly helped the case of simulating analog computers. Based on experiences gained with this setup other systems allowing for increased degrees of interactivity were developed in due course. Typical examples are *DSL-90, MIMIC*, and most importantly *CSMP*, which is described in more detail in →section 14.3.[9]

14.1 Basics

The overall structure of a typical simulation system is shown in →figure 14.1. The model description is first analyzed and then translated into a suitable intermediate representation such as an abstract syntax tree. This step is followed by a sorting run, which is necessary since the analog computing elements are simulated in a sequential fashion, abandoning the full parallelism of an

analog computer. The order in which these individual steps are performed has a great impact on the result.[10]

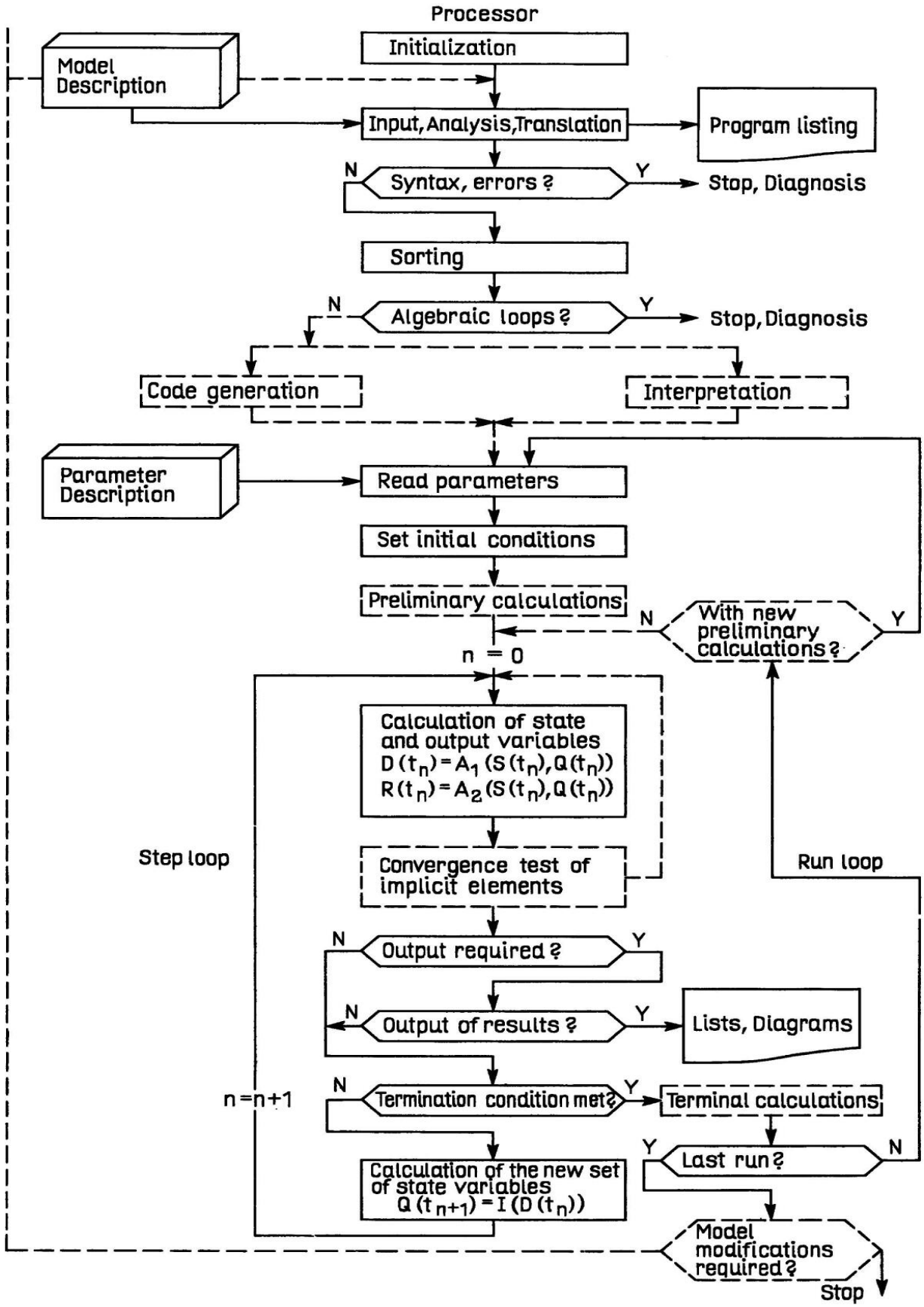

Processor
Model Description
Initialization
Input, Analysis, Translation
Program listing
N
Syntax, errors ?
Y
Stop, Diagnosis
Sorting
N
Algebraic loops ?
Y
Stop, Diagnosis
Code generation
Interpretation
Parameter Description
Read parameters
Set initial conditions
Preliminary calculations
N
With new preliminary calculations ?
Y
n = 0
Calculation of state and output variables
D (t n) = A1 (S (t n), Q (t n))
R (t n) = A2 (S (t n), Q (t n))
Step loop
Convergence test of implicit elements
Run loop
N
Output required ?
Y
N
Output of results ?
Y
Lists, Diagrams
n = n+1
N
Termination condition met?
Y
Terminal calculations
Y
Last run ?
N
Calculation of the new set of state variables
Q (t n+1) = I (D (t n))
Model modifications required ?
Stop

Figure 14.1. Flow diagram of a typical digital analog computer simulation system (see [→ GILOI 1975, p. 99])

One advantage of systems like this over a DDA is the possibility of using sophisticated numerical integration schemes, such as linear multistep methods, thus increasing the accuracy of the results and often having shorter run times.[11]

14.2 DDA programming system for the IBM 7074

An interesting system was developed by GEORGE JOSEPH FARRIS in the first half of the 1960s at the Institute for Chemical Engineering at Iowa State University of Science and Technology[12] for the *IBM 7070/7074* computer series.[13] The rationale behind this development was described as follows:

> "*In any problems arising in engineering, particularly in the simulation of processes and equipment, it may be faster, simpler, and more meaningful to use an analog method of solution than to reduce the problem to its most compact mathematical form and write a digital computer program for solving the resulting equations.*"[14]

This simulator, called *DIAN* as it combined some of the typical advantages of stored program *di*gital and *an*alog computers, eliminated the need to scale the problem equations by employing a floating point value representation. It, too, was run in batch mode with every component of the simulated analog computer with its associated inputs and outputs represented by a single punch card.

The central algorithm for performing the integration steps is interesting as it is not a traditional RUNGE-KUTTA method, which

was also considered but turned out to require more computing time while not yielding substantially better results. The integration in DIAN is bootstrapped with a simple EULER iteration step

$$y_1 = y_0 + y'_0 \Delta x,$$

which is followed by a single trapezoidal step

$$y_2 = y_1 + \left(\tfrac{3y'_1 - y'_0}{2} \right) \Delta x.$$

From then on explicit GREGORY-NEWTON steps

$$y_{n+1} = y_n + \left(\tfrac{23y'_n}{12} - \tfrac{4}{3y'_{n-1}} + \tfrac{5}{12y'_{n-2}} \right) \Delta x$$

are computed.

The DIAN system is demonstrated in [→ FARRIS 1964] with an initial value problem and a boundary problem, both simulating a stirred tank reactor described by

$$\dot{C} = a_1 - b_1 C - k_1 C e^{-\frac{E}{RT}} \text{ and} \tag{14.1}$$

$$\dot{T} = a_2 - b_2 T + k_2 C e^{-\frac{E}{RT}} . \tag{14.2}$$

These equations for concentration and temperature would be implemented on an analog computer by the program shown in →figure 14.2. This setup can be described by a set of just 21 punch cards as input for DIAN.[15] The simulation was run for integration limits of 0 and 3000 seconds at 10 second intervals with intermediate results for temperature and concentration being printed every 300 seconds of simulation time. This required a total of 30 seconds of actual run time on the IBM 7074 system.

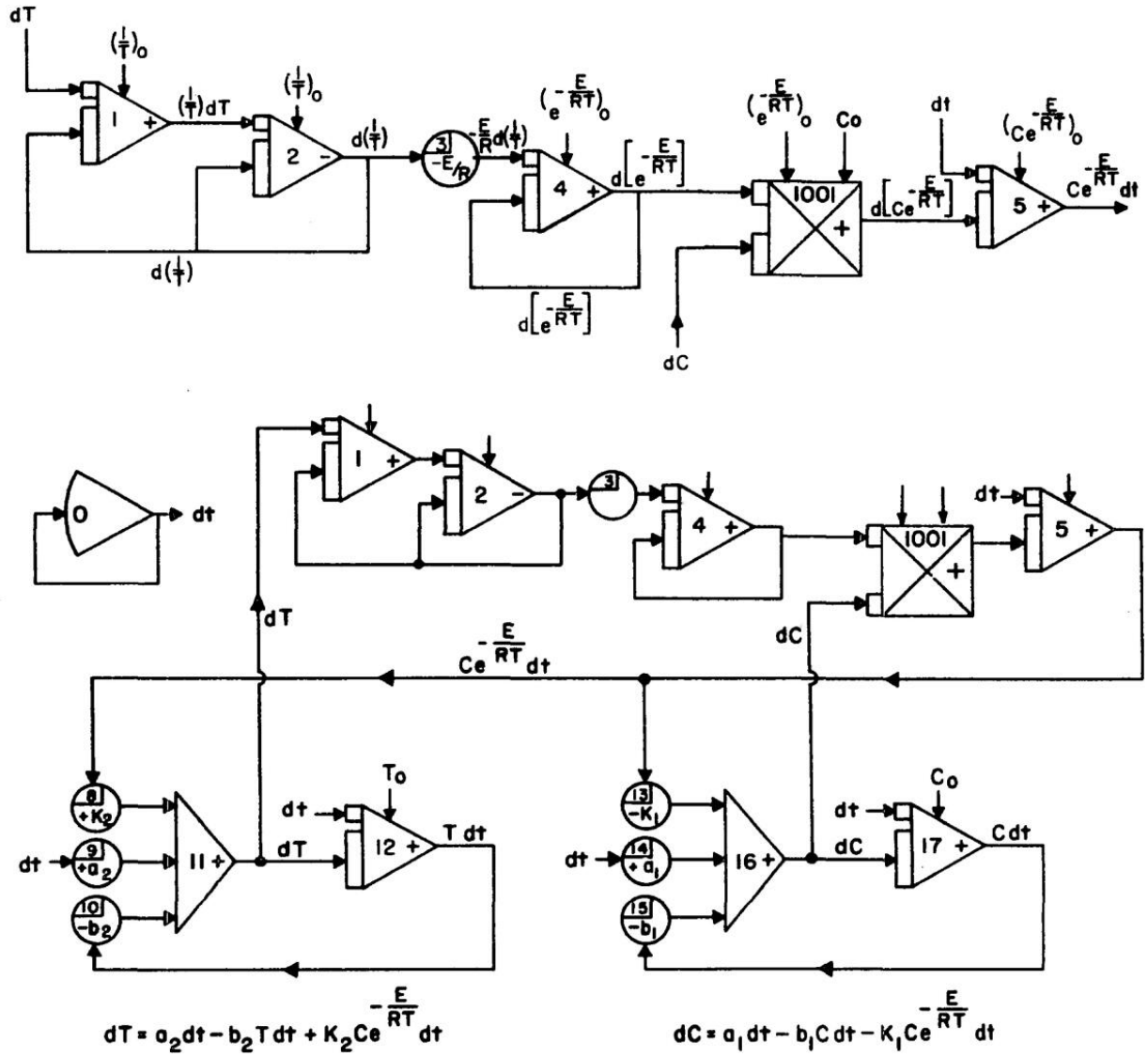


Figure 14.2. Simulated analog computer setup for equations (→14.1) and (→14.2) (see [→FARRIS 1964, pp. 23 f.])

Similar systems, also developed in the first half of the 1960s, were the *DIGITAL ANALOG SIMULATOR* (*DAS*)[16] running on a much larger *IBM 7094* and *DYSAC*, a "Digitally Simulated Analog Computer" running on a *Control Data 1604* computer. This system was used for example in studies on automatic landing systems for aircraft.[17]

14.3 CSMP

CSMP, the *Continuous System Modeling Program* is one of the best known and most influential early analog computer simulators.[18] A CSMP program consists of three sections of code:

> **INITIAL:** This (optional) section contains everything necessary prior to an actual simulation run. This includes initial conditions for integrators as well as static scaling options, etc.
> **DYNAMIC:** This section contains the description of the actual simulation circuit, which is then iteratively solved by the digital computer.
> **TERMINAL:** This section is also optional and allows to specify certain actions such as output of values, plotting of functions, etc., after a simulation run has been completed.

The following simple example problem shows the actual process of using the CSMP system. A mass-spring-damper system described by

$$m\ddot{y} + d\dot{y} + sy = 0$$

is to be simulated. First, the program variables are defined:

$$
\begin{aligned}
\mathrm{MASS} &= m \\
\mathrm{POS} &= y \\
\mathrm{POS0} &= y_0 \\
\mathrm{D} &= d \\
\mathrm{S} &= s \\
\mathrm{VEL} &= \mathrm{P\dot{O}S} \\
\mathrm{ACC} &= \mathrm{V\dot{E}L}
\end{aligned}
$$

Using the initial conditions $y(0) = 1$, $\dot{y}(0) = 0$, and $m = 1.5$, $d = 4$, $s = 150$ yields the CSMP program shown in →figure 14.3, consisting only of an (implicit) `DYNAMIC` section.[19] Output is controlled by the `TIMER` statement in line 8.[20] A typical simulation output listing the values for `TIME`, `POS`, `VEL`, and `ACC` is shown in →figure 14.4. At its heart are the two integrations

```
          CONSTANT MASS = 1.5, D = 4.0
          CONSTANT S = 150.0, POSO = 1.0
          ACC = (-S * POS - D * VEL) / MASS
          POS = INTGRL(POSO, VEL)
          VEL = INTGRL(0.0, ACC)
          PRINT POS, VEL, ACC
TITLE     MASS-SPRING-DAMPER SIMULATION
          TIMER FINTIM = 2.0, PRDEL = 0.05
END
STOP
ENDJOB
```

Figure 14.3. Simulating a mass-spring-damper system with CSMP

```
      SIMULATION OF A MASS-SPRING-DAMPER SYSTEM

TIME          POS           VEL           ACC
0.0           1.0000E 00    0.0          -1.0000E 02
5.0000E-02    8.8283E-01   -4.4884E 00   -7.6313E 01
1.0000E-01    5.7793E-01   -7.3878E 00   -3.8093E 01
1.7863E-01   -8.2319E 00    4.0884E 00
2.0000E-01   -2.1177E-01   -7.0839E 00    4.0068E 01
2.5000E-01   -5.0491E-01   -4.4556E 00    6.2373E 01
3.0000E-01   -6.4574E-01   -1.1340E 00    6.7598E 01
3.5000E-01   -6.2098E-01    2.0328E 00    5.6677E 01
4.0000E-01   ...           ...           ...
```

Figure 14.4. Typical CSMP simulation result (see [→ SPECKHART et al. 1976, p. 17])

$$\mathrm{POS} = \int_0^T \mathrm{VEL} dt + \mathrm{POSO} \text{ and}$$
$$\mathrm{VEL} = \int_0^T \mathrm{ACC} dt.$$

A more complex example, the simulation of a motorized cable reel, is described in [→Brennan et al. 1967]. The system to be simulated is shown in →figure 14.5. It consists of a motor-driven shaft connected to a reel. The velocity of the cable taken up on the reel or rolling off is measured by a tachometer feeding a motor control system, which is also connected to a manual unwind/reverse control switch and the objective is to control the speed of the cable. →Figure 14.6 shows a classic analog computer setup for this system.

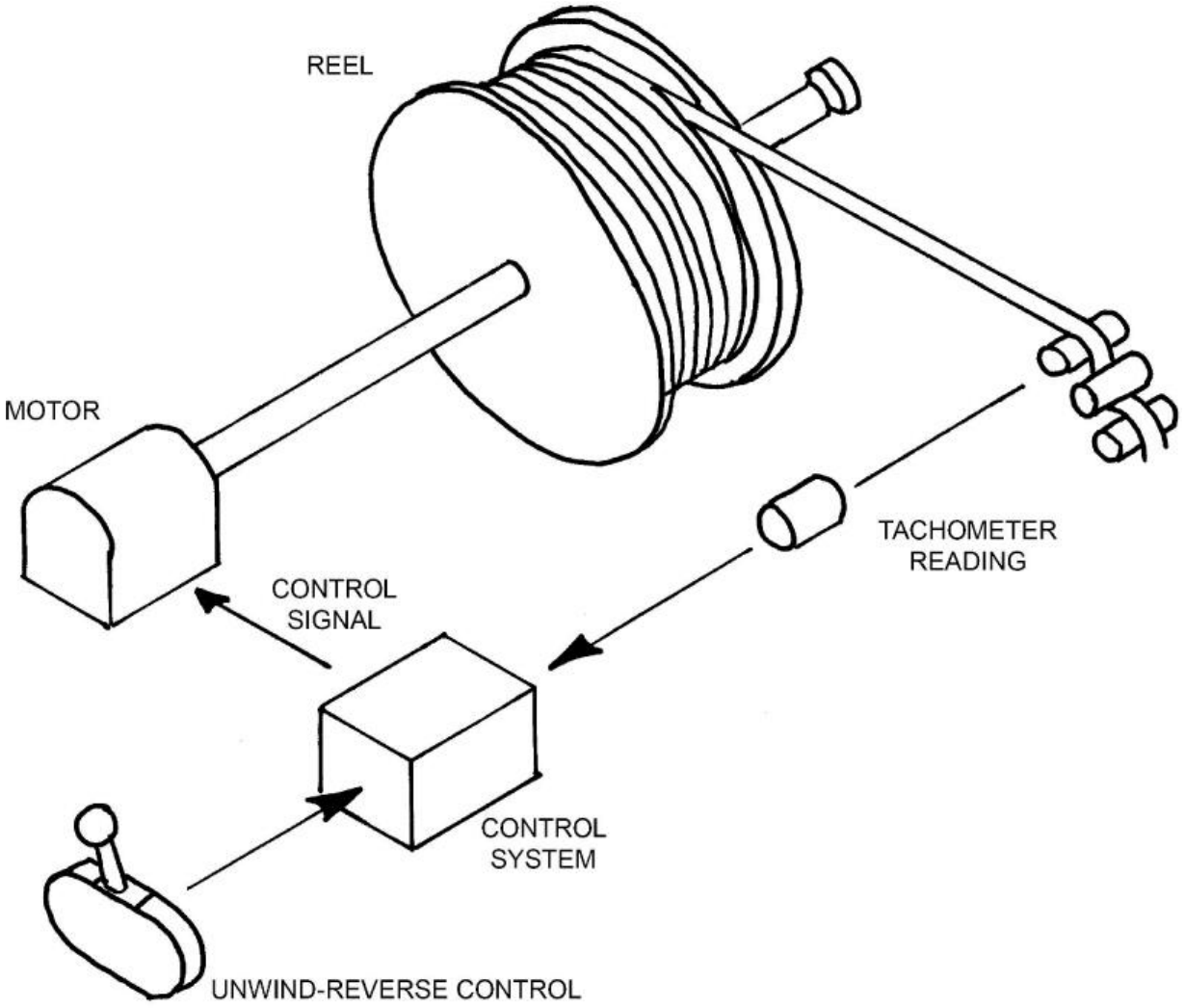


Figure 14.5. Motorized cable reel system

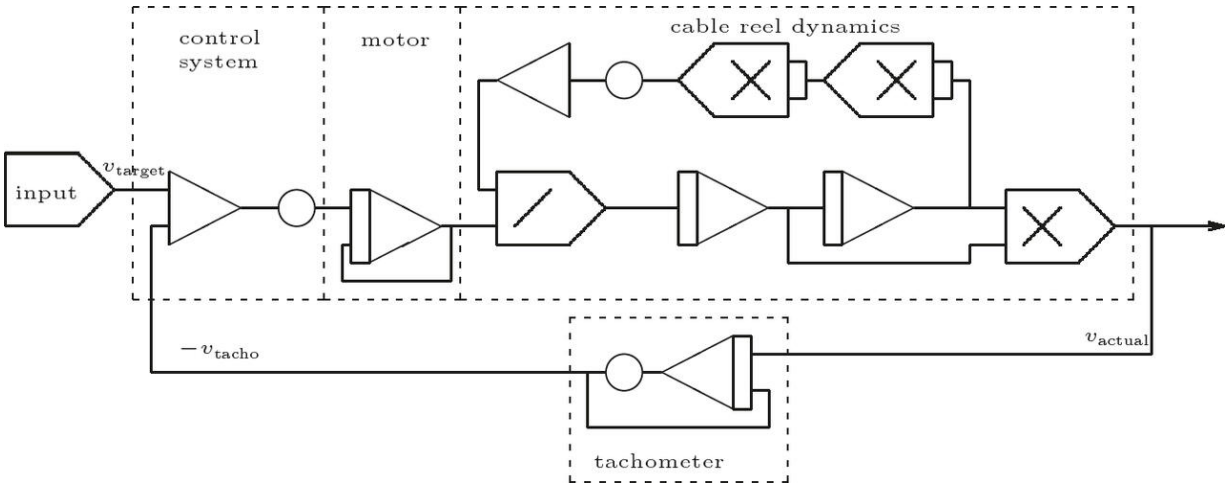


Figure 14.6. Analog computer setup for the cable reel problem (see [→BRENNAN et al. 1967, p. 251])

The corresponding CSMP program is shown in →figure 14.7. As a simple list of numerical values would not give the required insight into complex systems like this, CSMP could also print graphs, as shown in →figure 14.8.

```
TITLE  CABLE REEL SIMULATION
*
INIT
          K1  = (D ** 2) / (2.0 * PI * W)
PARAM      D  = 0.1, W = 2.0
CONST     PI  = 3.14159
PARAM   RFULL = 4.0, REMPTY = 2.0
*
DYNAM
           I = 18.5 * (R ** 4) - 221.0
       TH2DOT = TORQUE / I
       TH1DOT = INTGRL(0.0, TH2DOT)
            R = INTGRL(RFULL, (-K1 * TH1DOT))
        ERROR = VDESIR - VM
*
PARAM VDESIR = 50.0
       CONTL = GAIN * ERROR
        GAIN = 0.5
      TORQUE = 500.0 * DUMMY
       DUMMY = REALPL(0.0, 1.0, CONTL)
        VACT = R * TH1DOT
          VM = REALPL(0.0, 0.5, VACT)
*
FINISH     R = 2.0
TIMER   DELT = .05, FINTIM = 20.0
        PRDEL = 0.5, OUTDEL = 0.5
*
PRINT  VACT, VM, ERROR, CONTL, TORQUE, R, I
PRTPLT VACT
*
LABEL  PRELIM. TEST OF SYSTEM STAB. (G = 0.5)
METHOD RECT
END
*
PARAM   GAIN = 1.5
RESET  LABEL
LABEL  PRELIM. TEST OF SYSTEM STAB. (G = 1.5)
END
STOP
ENDJOB
```

Figure 14.7. Cable reel simulation program (see [→ Brennan et al. 1967, p. 260])

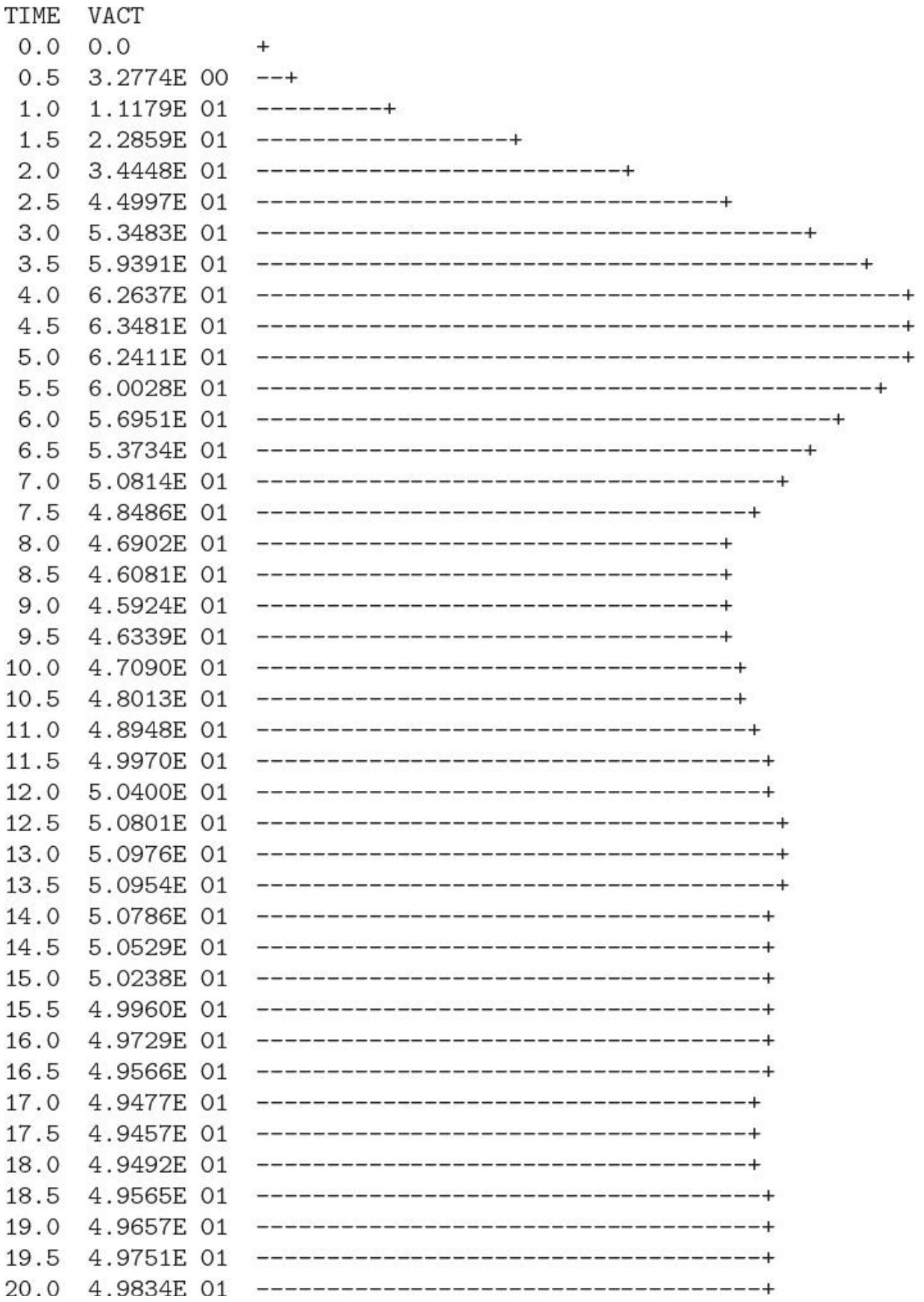

TIME VACT
0.0 0.0
0.5 3.2774E 00
1.0 1.1179E 01
1.5 2.2859E 01
2.0 3.4448E 01
2.5 4.4997E 01
3.0 5.3483E 01
3.5 5.9391E 01
4.0 6.2637E 01
4.5 6.3481E 01
5.0 6.2411E 01
5.5 6.0028E 01
6.0 5.6951E 01
6.5 5.3734E 01
7.0 5.0814E 01
7.5 4.8486E 01
8.0 4.6902E 01
8.5 4.6081E 01
9.0 4.5924E 01
9.5 4.6339E 01
10.0 4.7090E 01
10.5 4.8013E 01
11.0 4.8948E 01
11.5 4.9970E 01
12.0 5.0400E 01
12.5 5.0801E 01
13.0 5.0976E 01
13.5 5.0954E 01
14.0 5.0786E 01
14.5 5.0529E 01
15.0 5.0238E 01
15.5 4.9960E 01
16.0 4.9729E 01
16.5 4.9566E 01
17.0 4.9477E 01
17.5 4.9457E 01
18.0 4.9492E 01
18.5 4.9565E 01
19.0 4.9657E 01
19.5 4.9751E 01
20.0 4.9834E 01

Figure 14.8. Cable reel simulation result (see [→SPECKHART et al. 1976, p. 17])

14.4 DARE

A number of interesting simulation systems are described in [→KORN et al. 1983], the first being a collection of FORTRAN IV subroutines and functions to facilitate the implementation of simulation programs. In addition to this two stand-alone systems, *DARE P* (*D*ifferential *A*nalyzer *Re*placement, *P*ortable) and *DARE/ELEVEN*, are described. DARE P is basically a higher level language for system simulation intended to run in batch mode, while DARE/ELEVEN runs on a *PDP-11*[21] system and supports an incredibly high degree of interactivity for its time.

The batch DARE P system consists of three parts, a compiler, a runtime environment, and an output system capable of printing (tabulating) or plotting results. A typical DARE P program computing the impulse response of

$$\ddot{y} + D\dot{y} + y = 1$$

is shown in →figure 14.9.[22]

```
1         $D1
2        *
3        *    STEP RESPONSE OF DAMPED SECOND-ORDER SYSTEM
4        *
5             Y.=YDOT
6             YDOT.=-D*YDOT-Y+1.0
7    END
8             TMAX=50.,D=0.1
9    END
10            LIST,Y,YDOT
11            PLOT,Y
12   END
```

Figure 14.9. DARE P example program (see [→Korn et al. 1983, p. 158])

In contrast to this rather typical batch system DARE/ELEVEN offered a high degree of interactivity using not only an interactive command line interface but also a dedicated graphic display system with a variety of manual controls connected to the PDP-11 system as shown in →figure 14.10. On the right is a text based terminal with attached alphanumeric keyboard, which is mainly used to edit DARE/ELEVEN programs and to communicate with the underlying operating system. On the left is a graphic display with specialized keyboard for interactive control of simulation runs. This not only allows for control of the overall mode of operation such as RESET or RUN but also makes changes to parameters possible during a simulation run, thus resembling the manual potentiometers of a classic analog computer. Later extensions to this system also included a joystick to control values within a simulation.[23]

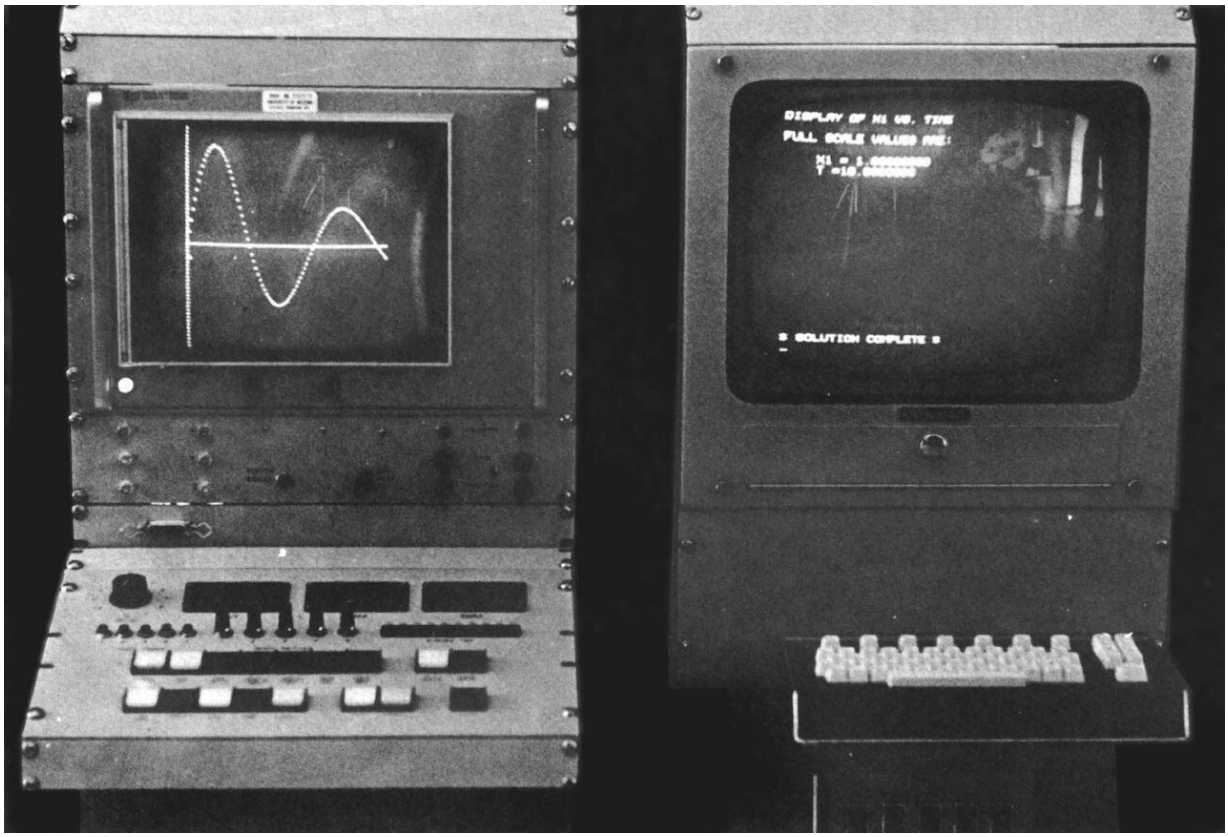


Figure 14.10. An early implementation of DARE/ELEVEN (see [→ Korn et al. 1983, p. 188])

14.5 Modern approaches

In the early 1980s simulation systems were developed for the rapidly growing personal computer market. [→ Titchener et al. 1983] describe an early example which was aimed at applications in computational chemistry. In 1985 Beukeboom et al. described another personal computer based simulation system.[24] This influenced the development of more modern tools such as *MATLAB* featuring (among many others) a plugin called *Simulink*. This contains a graphical block diagramming tool, which allows systems to be described quickly by using what are essentially classic analog computer programming symbols and techniques.

→Figure 14.11 shows a mass-spring-damper simulation in MATLAB using Simulink.

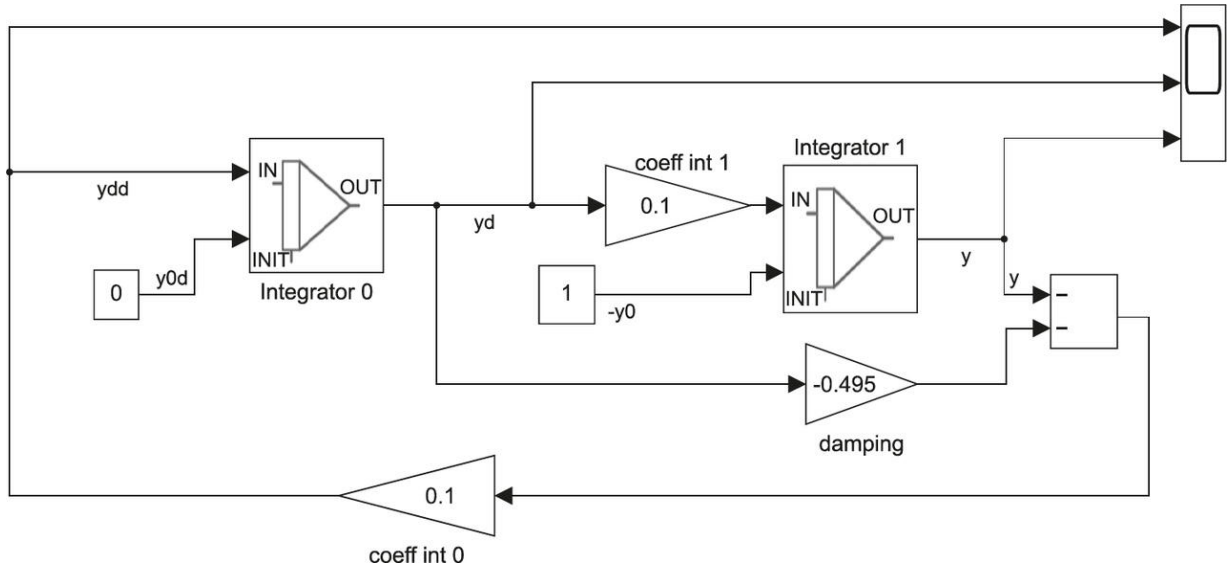


Figure 14.11. Simulation of a mass-spring-damper system in MATLAB

A more recent development is the open source system PYBRID.[25] It can be used as an analog computer simulator as well as for generating configuration setup data for reconfigurable analog computers in a hybrid computer configuration.

15 Applications

The following sections give an overview of typical analog computer applications. Due to the constrained space and the multitude and complexity of areas in which analog computers were used, these descriptions will be quite cursory. As Ernst Kettel puts it, there is next to no scientific field where an analog computer could not be applied successfully.[1]

15.1 Mathematics

Although analog computers do not excel when it comes to accuracy,[2] there are many areas in mathematics where they can be applied with great benefit.

15.1.1 Differential equations

Ordinary differential equations (*ODE*s) are obviously ideally suited for treatment on an analog computer. The simplest case of *linear ordinary differential equations*[3] poses no special problems and can be handled directly with the Kelvin feedback technique or the substitution method described in →sections 7.2 and →7.3. The equations of interest are of the general form

$$\sum_{i=0}^{n} a_i \frac{\mathrm{d}^i y}{\mathrm{d}y^i} = f(t).$$

If the coefficients a_i are constant, the treatment of such equations and systems thereof with an analog computer is straightforward. If the a_i are variable, the additional problem of generating these coefficients during a simulation run arises. In most cases the a_i will be generated as solutions of auxiliary differential equations. Nevertheless, the scaling process is often difficult in the case of variable coefficients.

Nonlinear differential equations are also challenging with respect to the scaling process. The fact that the structure of the computer setup often depends on the selected scaling factors complicates their solution as rescaling often not only requires the change of coefficients but also a change of the computer setup. A wealth of information about these challenges can be found in [→Giloi et al. 1963, pp. 221 ff.] and [→MacKay et al. 1962, pp. 212 ff.]. The application of hybrid computers in this area is covered in detail in [→Valisalo et al.].

Boundary value problems are problems in which boundary values for a differential equation at $t = 0$ and $t = t_{max}$ are given and these must be satisfied by a solution. These can often be solved easily with an analog computer[4] running in repetitive mode by applying a trial-and-error procedure.[5] Starting with the boundary value at $t = 0$ the equations are solved repeatedly with a human operator or the digital computer part of a hybrid computer system varying the parameters until the second boundary value condition is satisfied by a certain set of parameters.[6]

The treatment of partial differential equations has already been described cursorily in →section 7.4. More information can be found in [→Heinhold et al. 1969, pp. 183–197], [→MacKay et al. 1962, pp. 243 ff., pp. 293 ff.], and [→Howe 1962]. An interesting method for generating a flicker-free display of the solutions of partial differential equations on an oscilloscope is described in [→Ameling 1962/1], while [→Feilmeier 1974, pp. 215 ff.] deals with the application of hybrid computers for solving this class of equations. [→Bekey et al. 1971] describe hybrid computer techniques for solving nonlinear parabolic partial differential equations while [→Hsu et al. 1968] show a kind of time-sharing technique for reusing analog computer components within a single large-scale problem.

Another interesting technique, developed by Lawrence Wainwright, uses a magnetic tape for storing analog data and is described in [→Jackson 1960, pp. 586 f.]. [→Ashley et al. 1968] show the application of the classic separation of variables approach for solving partial differential equations by means of a hybrid computer setup.

There are also approaches to solve PDEs by stochastic approaches such as random walks based on the Feynman[7]-Kac[8] formula linking parabolic partial differential equations to stochastic processes, typically

Wiener processes, i. e. Brownian[9] motion. This makes it possible to get a solution for a PDE at some point of interest without having to compute the values of its neighbors in a discretized grid.[10] A simple example can be found in [→ Sawhney et al. 2020] and [→ Sawhney et al. 2022]. All that is required are *n* uncorrelated noise sources for an *n*-dimensional PDE, a number of integrators to implement the actual random walk and some means of detecting boundaries.[11][12]

15.1.2 Multiple integrals

Multiple integrals are a bit nasty to implement on an analog computer. A straightforward approach for a two-dimensional domain would be to have one integrator running with very high k_0, integrating over slices of that region with a second integrator running much slower, and controlling the slicing process, thus approximating the actual integral

$$I = \int_a^b \int_{c(x)}^{d(x)} f(x, y) \, \mathrm{d}y \, \mathrm{d}x$$

by

$$I \approx \sum_{i=0}^{n} \int_{c(x_i)}^{d(x_i)} f(x, y) \, \mathrm{d}y.$$

This approach can be implemented quite easily on an analog computer, ideally with the support of a digital control unit controlling the generation of a step function for slicing the function domain. This would also be a suitable approach on a hybrid computer. [→ Hausner 1963] describes an interesting approach using a sinusoidal or triangular signal delimited by the domain to be integrated over to “scan” *f*(*x*, *y*). Such approaches can also benefit from analog storage elements such as provided by the *DYSTAC*[13] (*Dynamic Storage Analog Computer*), see [→ Andrews 1960/1] and [→ Andrews 1960/2]. [→ Kobelt[a]] describes a similar problem from chemical engineering where several groups of integrators are controlled individually, basically implementing an interactive process.

15.1.3 Integral equations

In general, an analog computer is not particularly well suited for treating integral equations. However, due to their importance in fields like electrical and communications engineering, etc., many approaches were developed to solve this class of equations with an analog computer. Integral equations are of the form[14]

$$f(x) + \int_0^a K(t,x)u(t)\,\mathrm{d}t = 0 \quad \text{or} \tag{15.1}$$

$$f(x) + \int_0^a K(t,x)u(t)\,\mathrm{d}t = u(x). \tag{15.2}$$

These equations are called FREDHOLM[15] equations of type 1 or type 2, respectively. $u(x)$ is the unknown, while $f(x)$ and $K(t, x)$, the *kernel*, are known. The representation of this kernel is the main challenge on an analog computer since functions of more than one variable are typically difficult to implement. A hybrid computer can simplify this considerably. In cases where no hybrid computer is available, x has often been restricted to a constant value $\bar{x}$ so that a simple function generator using only one argument variable can be used to implement $K(t, \bar{x})$.

Using proper time-scaling, the upper boundary of the integral in equation (→ 15.1) or (→ 15.2) can be identified with computer time τ. The result of a computer run is then compared with either 0 or $u(x)$, depending on the type of equation to be solved. Based on the deviation from the desired result, the parameters defining $u(x)$ can be varied manually or under program control. So basically solving integral equations with an analog or hybrid computer boils down to a trial-and-error technique.

Restricting x to a constant value is often not sufficient – in these cases x is discretized appropriately, using function generators to implement $K(t, x_i)$ for a number of such constant values x_i. Running the analog computer in repetitive operation and using electronic switches to switch between these function generators depending on the actual index i, it is possible to display a family of curves on an oscilloscope.[16] More information about solving integral equations with analog and hybrid computers can be found in [→ SYDOW 1964, pp. 112 ff.], [→ MACKAY et al. 1962, pp. 317 ff., pp. 331 ff., pp. 352 ff.], and [→ CHAN 1969].

It should be noted that the idea of using analog computers to treat integral equations is quite old. Norbert Wiener[17] suggested a *product integraph*,[18] which was later developed and built by Vannevar Bush,[19] F. G. Kear, H. L. Hazen, H. R. Stewart, and F. D. Gage at the Massachusetts Institute of Technology and is shown in →figure 15.1.[20] This mechanical device could solve the following equation where $f(t)$ is a known function, and $\Psi(x)$ is the unknown:[21]



Figure 15.1. The product integraph (see [→Popular Mechanics 1928, p. 15])

$$\Psi(x) = \int_a^x f(t)\Psi(t)\,\mathrm{d}t.$$

This required two operators to manually follow given curves for $f(t)$ and $\Phi(t)$ (similarly to the setup shown in →figure 2.23 on page 43), a tedious and not very precise process, which typically took about four minutes for a run performed at “normal speed”.[22] The result was produced by a watt-hour meter performing the required multiplication based on one of the functions represented a time-varying voltage and one by a current.

The problem of manual curve followers was solved with the subsequent development of the *cinema integraph*. It could evaluate expressions such as

$$\int_0^\infty f(t)\sin(nt)\,\mathrm{d}t \text{ or } \int_0^\infty f(t)\cos(nt)\,\mathrm{d}t$$

as well as the more general form

$$\int_0^T f(t)g(t)\,\mathrm{d}t.$$

The two functions under the integral were represented by the optical density of photographic films. These were arranged in such a way that light from a suitable and constant light source had to pass both films after which it was measured by a photocell. This rendered the manual curve following process obsolete and allowed much shorter solution times.[23]

[→Bekey et al. 1967] describe a hybrid computer approach for solving integration equations using the von Neumann and Fisher iteration method.

15.1.4 Roots of polynomials

Determining the real or complex roots of polynomials

$$P_n(x) = \sum_{i=0}^{n} a_i x^i.$$

with an analog computer is a straightforward process:[24] $P_n(x)$ is generated either by explicit multiplication generating the x^i, or by repeated integration yielding x, $-\frac{1}{2}x^2$, $\frac{1}{6}x^3$, etc. Since analog computers typically have more integrators than multipliers available and since multipliers are expensive and multiplication is not as precise an

operation as integration, the latter approach is usually preferred.[25] This proved especially useful in the case of LEGENDRE-, CHEBYSHEV-, or HERMITE-polynomials.[26]

Given a computer setup generating the required polynomial, the analog computer can then vary *x* in the interval of interest. Using a comparator, the zeros of the polynomial can be detected and the machine can be automatically placed into halt mode to read out the corresponding value of *x*.

15.1.5 Orthogonal functions

An important problem in control engineering applications and other fields is the approximation of a function $f(t)$ as a sum of orthogonal functions $\varphi_i(t)$:[27]

$$f(t) \sim \sum_{i=1}^{n} a_i \phi_i(t)$$

Normally, the square error term

$$\int_I \left(f(t) - \sum_{i=1}^{n} a_i \phi_i(t) \right)^2 w(t)\,\mathrm{d}t$$

is to be minimized over an interval I with $w(t)$ denoting a weighting function. This is satisfied if the $\varphi_i(t)$ form an orthogonal system and the coefficients a_i are defined by

$$a_i = \tfrac{1}{c} \int_I f(t)\phi_i(t)\,\mathrm{d}t \tag{15.3}$$

where

$$\int_I \varphi_i\,(t)\varphi_k\,(t)\mathrm{d}t = \begin{cases} c \neq 0 & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases}$$

holds. Using an analog computer, the coefficients a_i can be determined based on (→15.3).

15.1.6 Algebraic functions of complex variables

Beginning in the 1950s, the use of LAPLACE transforms and *pole-zero plots* became commonplace in all branches of control theory, driven not least by the upcoming *space race*. The behavior of control systems represented by transfer functions can be analyzed by this technique in the complex plane. Since manual calculations involving complex numbers are quite tedious and classic analog computers were more focused on DEQs and PDEs in the real domain, special machines such as the one described in [→MORGAN 1954] were developed. It allows the evaluation of functions of the form

$$F = Ks^{n_0} \sum_i (s - s_i)^{n_i},$$

where *F, K, s*, and s_i are complex values represented as magnitude $|F|$ and phase $\angle F$ using resistive sheets to derive magnitude and phase values for given parameters. This early machine is shown in →figure 15.2. At the left are the power supply and a meter unit connected to either of the aforementioned conductive sheets by means of the coordinate frame seen on the right. This contains the connections to the conductive sheet under evaluation feeding various voltages to the sheet and reading out voltage potentials at selected points.

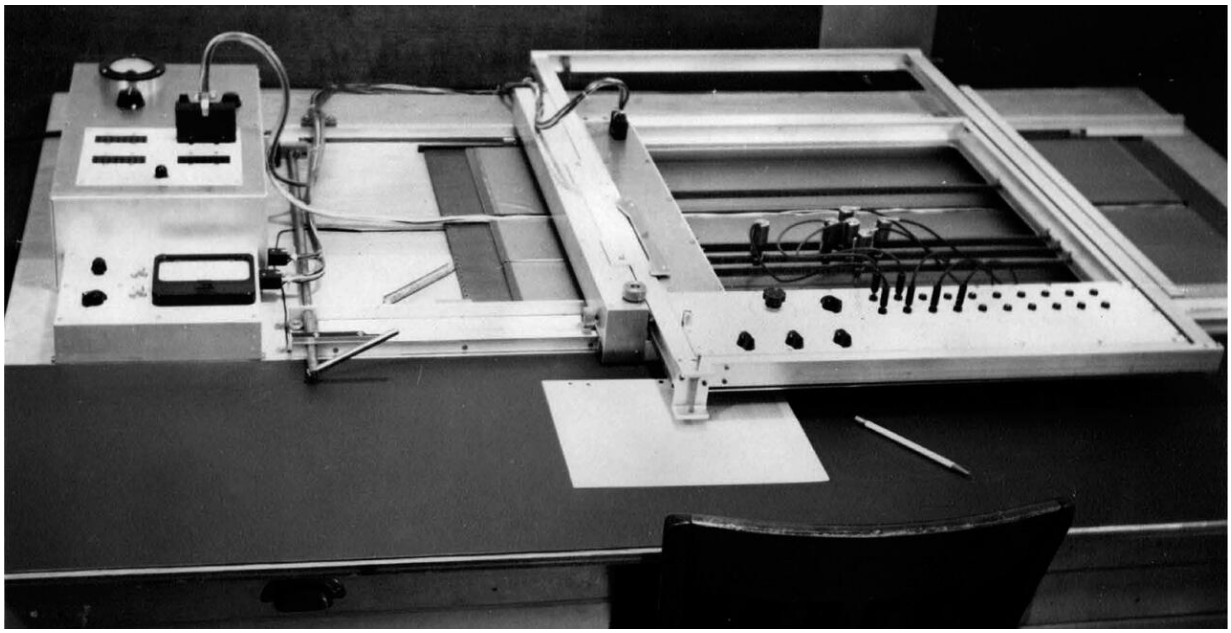


Figure 15.2. Complex variable algebraic function computer (see [→ MORGAN 1954, p. 85])

This machine was developed further and later commercialized as the *ESIAC* (*Electro Scientific Industries Analog Computer*).[28] This production model also featured two conductive sheets for magnitude and phase computations but both could be read simultaneously instead of sequentially as in the early system shown before.[29] →Figure 15.3 shows the overall system on the left. It resembles a desk with a central control panel and two subunits, each containing one resistive sheet, on the left and right. The right half of the picture shows a detailed view of the control panel with the conductive sheets exposed. Each sheet can be probed by some manually adjustable pickups visible on the far left and right.

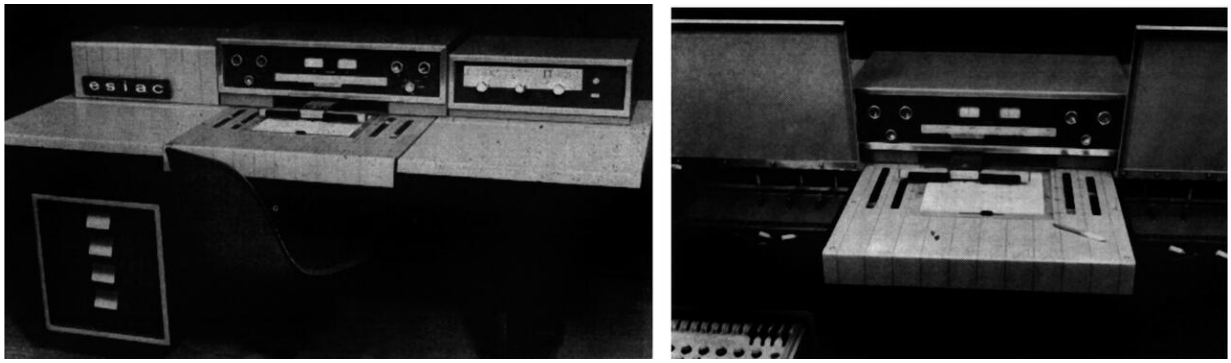


Figure 15.3. ESIAC system and detail of the conductive sheet pickups (see [→Morgan 1961/2, p. 525])

An interesting application example for this production system can be found in [→Arpasi et al. 1967] and [→Hart et al. 1967] where the control system of a nuclear powered rocket engine is treated. It was also used during the development of a general purpose airborne simulator system, see [→Kroll et al. 1966]. A recent description of this historic machine can be found in [→Pronk et al. 2021] and [→Pronk et al. 2022].

15.1.7 Linear algebra

Solving systems of linear equations is a task not inherently well suited for analog computers – a stored-program digital computer is often a better approach. However, the problem itself was relevant enough to justify the development of specialized machines, maybe the most prominent of which being the system developed under John "Bud" Benson Wilbur[30] shown in →figure 15.4. This unique machine was developed in the first half of the 1930s and solved systems of linear equations by a ten steel plates, each capable of rotating along a central axis. Nine of these plates represented one unknowns each while the tenth implemented the constants. The central summation was implemented by a steel tape, 200 meters in length, similar to the one used in Kelvin's tide predictor for summing the various harmonic terms. In fact, the theoretical foundations for this machine were laid in 1878 by William Thomson.[31]

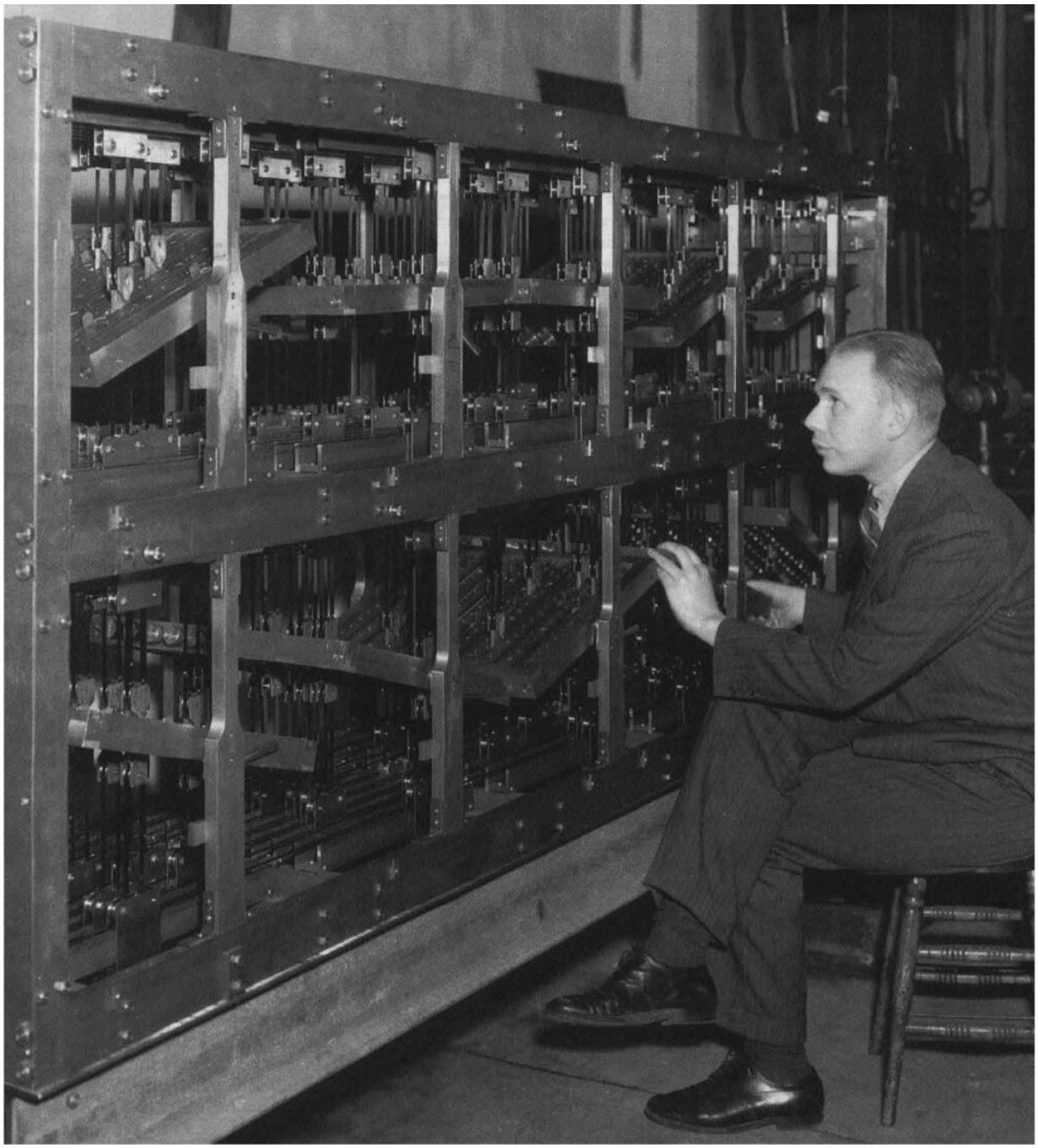


Figure 15.4. John B. Wilbur at his linear equation system solver ([→ Petzold 1997, p. 29])

The machine was built by the Singer Sewing Machine Company between 1934 and 1935 and was put into operation in 1936. Solving a system of

nine coupled equations required between one and three hours, yielding a solution of three decimal digits precision.[32]

In later years several techniques suitable for traditional analog computers were developed, which were typically employed when no digital computer was available. For a long time these approaches were of historic interest only but with the advent of artificial intelligence, especially *artificial neural networks*, such analog techniques might gain importance again as they offer a chance to drastically lower the required energy for training such networks using approaches like *backpropagation*.

A direct approach implementing the system of linear equations using only summers and coefficient potentiometers is unsuited for an analog computer since it results in *algebraic loops*.[33] Other techniques like the JACOBI[34] method, the GAUSS[35]-SEIDEL[36] method or relaxation approaches are better suited for an analog computer.[37]

As an example a system of linear equations $\mathbf{A}\vec{x} = \vec{b}$ is to be solved by the JACOBI method. First $\vec{x} = \mathbf{B}\vec{x} + \vec{k}$ is computed, with

$$\mathbf{B} = -\begin{pmatrix} 0 & \frac{a_{12}}{a_{11}} & \cdots & \frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & 0 & \cdots & \frac{a_{12}}{a_{22}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{n1}}{a_{nn}} & \frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{pmatrix} \quad \text{and} \quad \vec{k} = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}.$$

Typically, **B** and $\vec{k}$ are determined once, aided by a digital computer. Iteration steps of the form

$$\vec{x}_{n+1} = \mathbf{B}\vec{x}_n + \vec{k}$$

can then be performed on the analog computer.[38] This iterative approach requires pairs of integrators for each component of $\vec{x}$ being used as storage elements, requiring a complex control scheme.[39]

[→Mitra 1955] describes a technique for solving systems of linear equations on a specialized analog computer, which not only yields the desired solution but also the eigenvalue of the matrix with the largest absolute value.

A different approach transforms the problem of solving a system of linear equations to the problem of solving a corresponding system of differential equations by

$$\begin{aligned} \mathbf{A}\vec{x} - \vec{b} &= \vec{\epsilon} \\ \dot{\vec{x}} &= -\vec{\epsilon}, \end{aligned}$$

which converge to the desired solution for $t \to \infty$. This technique is treated in detail in [→Heinhold et al. 1969, pp. 148 ff.] and [→Ulmann et al. 2019], which not only examines convergence criteria but also contains practical examples.

15.1.8 Eigenvalues and -vectors

Determining eigenvalues and eigenvectors for a given matrix of degree n, $\mathbf{A} = (a_{ij})$, $i, j = 1, \ldots, n$, is a common task and can be reduced to solving a system of linear equations of the form

$$(\mathbf{A} - \lambda_i \mathbf{I}) \overrightarrow{x}_i = 0. \tag{15.4}$$

In the case of complex valued eigenvalues, a system of equations of degree $2n$ is required. In either case, the eigenvalues λ_i correspond to the roots of the characteristic polynomial $p_A(\lambda) = \det[\mathbf{A} - \lambda\mathbf{I}]$ with $\mathbf{I}$ denoting the identity matrix.[40]

Although this problem can be readily solved by applying the methods described above, the von Mises iteration is often employed, as described in [→Popović 1964]. This method works directly on the components of **A** and does not require an explicit characteristic polynomial to be derived.

[→Jennings 1962, pp. 31 ff.] gives an overview of various specialized analog computers suitable for eigenvalue and -vector problems before outlining a novel approach to this class of problems.

15.1.9 Fourier synthesis and analysis

While transformations from frequency domain into time domain and vice versa are performed nowadays on stored-program digital computers employing *Fast Fourier Transforms* (*FFT*) or *Wavelet Transforms* neither of these methods, nor adequately fast stored-program computers, were available until the 1970s. Therefore the decomposition of a signal into its harmonic components, specified by frequency, phase, and amplitude, was done regularly with analog computers[41] by a Fourier transform. Basically, every periodic function $f(x)$ with a period of l can be represented by

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos\left(\frac{2\pi k x}{l}\right) + b_k \sin\left(\frac{2\pi k x}{l}\right)\right) \tag{15.5}$$

under certain circumstances where the coefficients a_k and b_k are defined by

$$a_k = \frac{2}{l} \int_0^l f(x) \cos\left(\frac{2k\pi x}{l}\right) \mathrm{d}x \quad \text{and} \tag{15.6}$$

$$b_k = \frac{2}{l} \int_0^l f(x) \sin\left(\frac{2k\pi x}{l}\right) \mathrm{d}x. \tag{15.7}$$

Given a_k and b_k the reconstruction of a signal, the *Fourier synthesis*, is what harmonic synthesizers were built for.[42] On an electronic analog computer, the sine-/cosine-terms required can be implemented by an analog computer setup as shown in →figure 8.2 (page 201).

The inverse process, determining a_k and b_k for a given function $f(x)$, the *Fourier analysis*, is more involved. The direct approach, evaluating the integrals (→15.6) and (→15.7), requires a very exact time base as this determines the bounds of the integration operations.[43]

Another approach uses low-pass filters, each consisting of a single integrator with negative feedback in the simplest case, to determine the coefficients for a given $f(t)$.[44] This has the disadvantage that only one pair of a_k and b_k can be determined at a time. In addition to that, the low-pass filters typically have rather long time constants.

Another approach is based on a circuit like that shown in →figure 8.2 (page 202), which is subjected to a forcing function. The resonant frequency of this circuit allows it to determine a pair a_k and b_k directly.[45]

An interesting application involving tide calculations in conjunction with a control problem is described in [→Apter 1967] where a tidal power plant is simulated and optimized.

15.1.10 Random processes and Monte-Carlo simulations

The study of random processes is an important area of application for analog computers since many commercially interesting processes, ranging from chemical technology to the behavior of flight vehicles, nuclear reactors, the creation of steam bubbles in coolant loops, etc., are affected by external random events.[46]

Consequently, a suitable source of randomness (noise) is required. Typical random generators are either based on physical noise sources, see →section 4.12, or on shift-registers with feedback generating pseudorandom bit sequences, which can be converted to analog voltages by DACs. These can be readily implemented using digital elements most medium to large analog computers already offer. The main advantage of a pseudorandom source is the repeatability of simulation runs since it is possible to generate the same pseudorandom sequence over and over again. Its disadvantage is that controlling and guaranteeing the required degree of randomness can be challenging, depending on the sensitivity of the system under consideration.[47]

Using such noise sources *Monte-Carlo methods* can be employed to a wide variety of problems on an analog or hybrid computer.[48] [→Bekey et al. 1966] describe a hybrid computer technique for parameter optimization based on a random search. A fascinating approach for solving partial differential equations by Monte-Carlo methods on a hybrid computer is described in [→Little 1966].

The investigation of random processes often requires the extension of the classic discrete arithmetic mean to a time varying continuous variable. An interesting approach to this problem is the *exponentially mapped past* developed by Joseph Otterman and Robert Mario Fano.[49] Details of this technique can be found in [→Otterman 1960], [→Fano 1950], with

practical implementations described in [→EAI 1964/3], and [→ULMANN 2021/3].

15.1.11 Optimization and operational research

Optimization problems are of great commercial importance and were often solved by analog computers. In general, a linear optimization problem can be represented by a real-valued matrix **A** ∈ $\mathbb{R}^{m,n}$ with *m* < *n* and two vectors $\vec{b} \in \mathbb{R}^m$ and $\vec{x} \in \mathbb{R}^n$, component wise satisfying the condition

$$A\vec{x} \leq \vec{b}. \tag{15.8}$$

The goal of the optimization process is to determine a vector $\vec{x}$ that not only satisfies (→15.8) but also maximizes the scalar product $\vec{c}^{\mathrm{T}}\vec{x}$ with $\vec{c} \in \mathbb{R}^n$ representing the *value coefficients*.[50]

Very early approaches used direct analogies such as conductive sheets or contour map function generators[51] that were sampled by a stylus driven with an *xy*-plotter.[52] Another simple analog computer for optimization tasks is shown in 15.5. Its basic circuit, shown in →figure 15.6, is extremely simple: Each parameter of the optimization problem is represented by two potentiometers of which the first one represents the parameter itself and will be varied manually during the optimization process, while the second potentiometer is used to set a fixed weighting factor describing the underlying process. The currents flowing through these potentiometer groups are then summed and displayed on a meter.

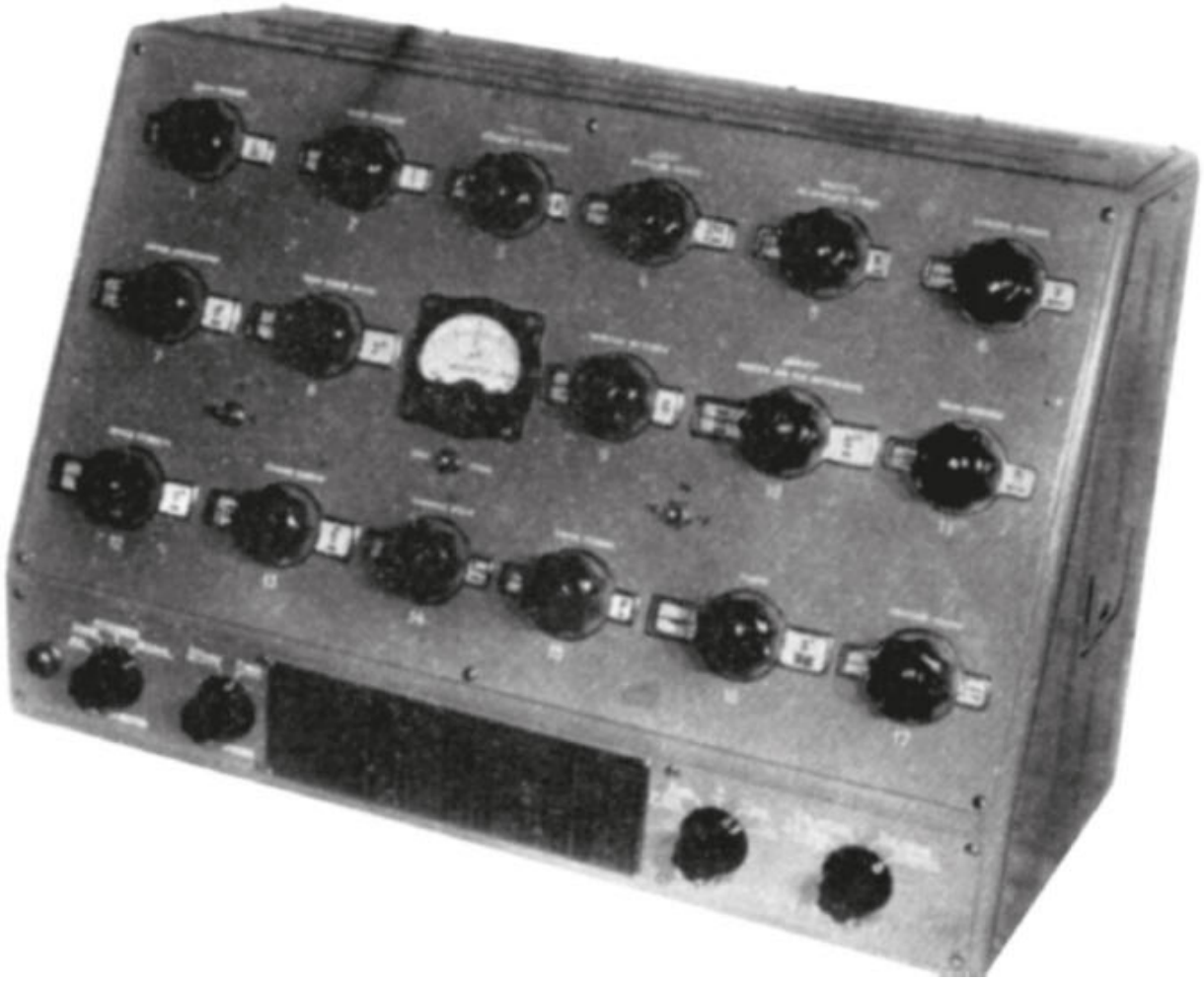


Figure 15.5. BPRR-2 computer used for the optimization of industrial processes (see [→ USHAKOV 1958/2, p. 1961])

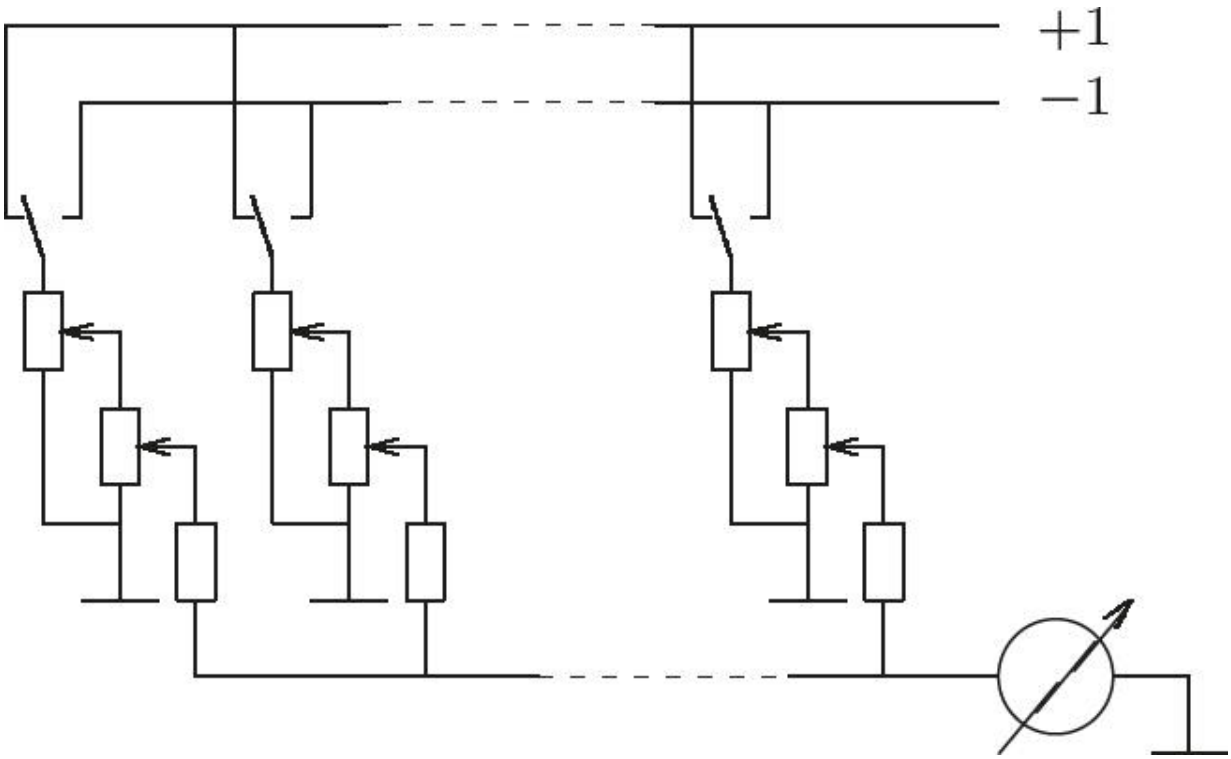


Figure 15.6. Basic circuit of the BPRR-2 (see [→ USHAKOV 1958/2, p. 1961])

[→ DENNIS 1958] describes techniques and specialized electric networks to

> "*minimise a function of several variables while satisfying some inequality constraints on other functions of the variables.*"

Other approaches to optimization with analog computers are based on Monte-Carlo techniques as mentioned above, or use gradient methods like *continuous steepest ascent/descent*.[53] Gradient methods are especially well suited for hybrid computers where the digital processor controls the ascent/descent-process performed by the attached analog computer.[54]

15.1.12 Pipeline networks

The simulation of gas, water, or ventilation pipeline networks is as commercially important as it is difficult from a practical point of view.[55] No modern city or even village could do without proper water, waste water, and gas pipeline networks. Every mine operation not only requires adequate ventilation facilities but also faces the danger of fires

and explosions releasing combustion gases and heat through the mine, something the ventilation network must be capable of coping with, protecting personnel.[56]

While voltages *U*, currents *I*, and resistances *R* are related by Ohm's[57] *U* = *IR* law, the Navier[58]-Stokes[59] equations describing fluid dynamics cannot be reduced to such simple relationships between flow rates and pressures. This has led to a variety of empirical equations allowing the calculation of pressures and flow in pipes, such as the Hazen[60]-Williams[61] equation

$$V = C\sqrt{RS}$$

with *V, C, R*, and *S* denoting the fluid velocity, a proportionality value (not a constant), the hydraulic radius, and the head loss per unit length, respectively. Another well known approximation is the Darcy[62]-Weisbach[63] equation

$$\Delta H = f\frac{LV^2}{2gD}$$

with Δ*H* being the head loss, *L* the pipe length, *V* the fluid velocity, *D* the diameter of the pipe and *f* a friction factor, which was determined empirically as

$$f = \alpha + \frac{\beta}{\sqrt{V}}$$

where *α* and *β* are two parameters depending on the pipe diameter and the characteristics of its wall.[64] Other noteworthy empirical equations in this field are the Manning[65] formula using exponents of 2/3 and 1/2, or the Darcy friction factor formulas. A common theme to all of them is that the pressure drop in a pipe is proportional to the flow rate to a power of roughly $1.8 \leq x \leq 2$.

Calculating the flow in a single pipe using such formulas is straightforward but calculating the flow in a network of connected pipes, like a municipal water distribution system, gets rapidly more challenging with increasing number of pipes since the underlying nonlinear coupled equations typically require some iterative approach. Furthermore, tasks

such as the extension of existing pipeline networks require hundreds or more simulation runs with different parameters and configurations to select the best configuration under given constraints, which is unfeasible without some means of simulation.

There exists an obvious analogy between the flow of fluids in a pipeline network and the flow of direct current in a resistive network, with equivalence between fluid flow rate and electrical current, pressures and voltages, as well as pipe friction and electrical resistance. Even KIRCHHOFF's laws hold since the sum of flows into a node is obviously zero and the sum of pressures round a loop is zero, too. →Figure 15.7 shows a side-by-side comparison of a pipeline network with an equivalent electrical circuit. The main problem remaining is that of the exponential relationship between pressures and flow rates. Thus, an electrical model cannot simply use linear resistors representing pipe friction since these have to be exponential functions of the flow rates.

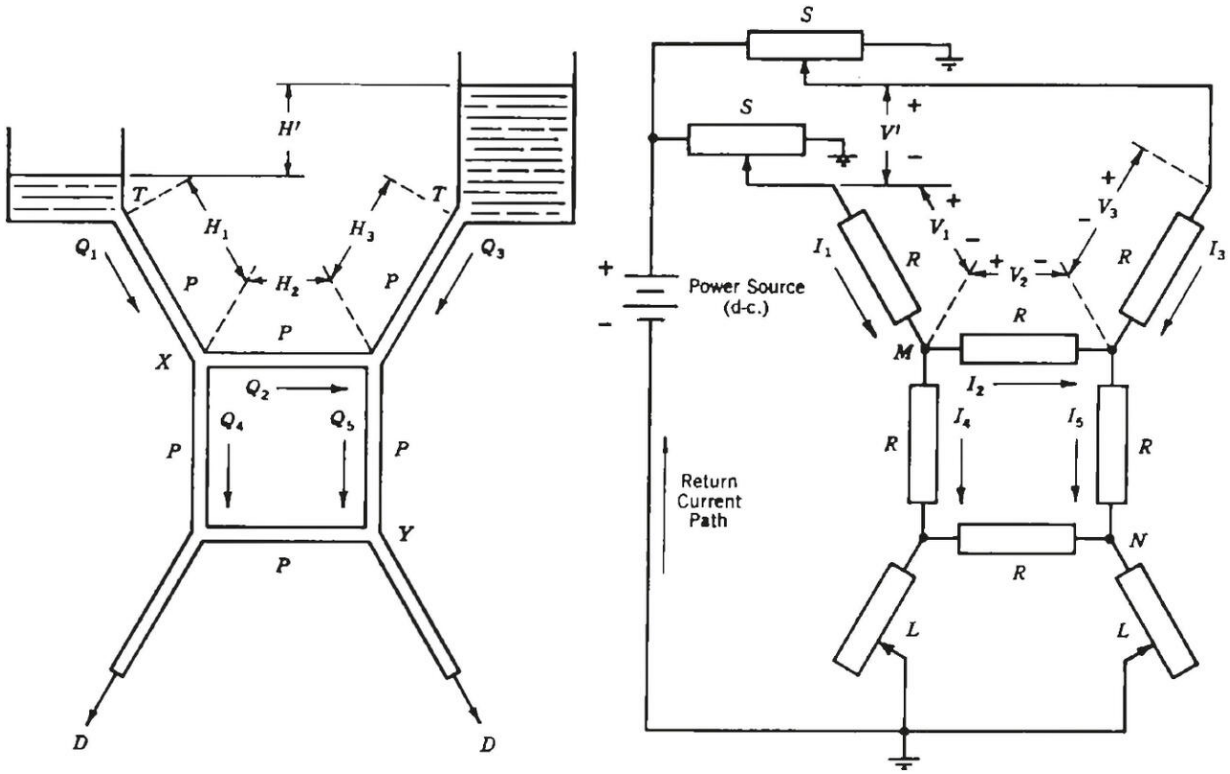


Figure 15.7. Comparison of a pipeline network with its electrical analog (see [→MCILROY 1950, p. 349])

Hazen, working on the purely linear MIT DC network analyzer, employed an iterative process: First, the hydraulic problem units were converted to Volts, Amperes, and Ohms. Based on this, a direct analog was set up on the network analyzer with initial resistance values based on the known pipe properties such as length, diameter, and roughness of the walls, as well as initial guesses for the flow rates in every branch of the network. The analyzer was then run with resulting currents and voltages being measured. Based on these values the resistances were changed, and the analyzer was run again. These two steps were repeated until the values reached stable values, representing the solution of the problem.

Malcolm Strong McIlroy[66] developed a special nonlinear resistive element based on a low voltage incandescent light bulb working in the range of 0 to 5 V with $U \propto I^{1.85}$ describing the voltage/current relationship.[67] This low operating voltage was deliberately chosen and results from the fact that many such resistors will be in series in a large pipeline network which would require dangerously high voltages if the individual operating voltage range was much higher.[68]

This exponent is a perfect match for the relation of pressure drop to flow rate in the Hazen-Williams equation for turbulent fluid flow. Accordingly, this new resistive element was named *Fluistor* and was manufactured spanning a range of resistances. →Figure 15.8 shows the McIlroy pipeline network analyzer based on these Fluistors. A large system contained up to 1000 Fluistors spanning a range of flow coefficients ranging from 0.058 to 500 with the required voltage sources, current sinks, measurement facilities, and a patch panel.[69] The system is described in [→Computers and Automation 1960, p. 15] as follows:

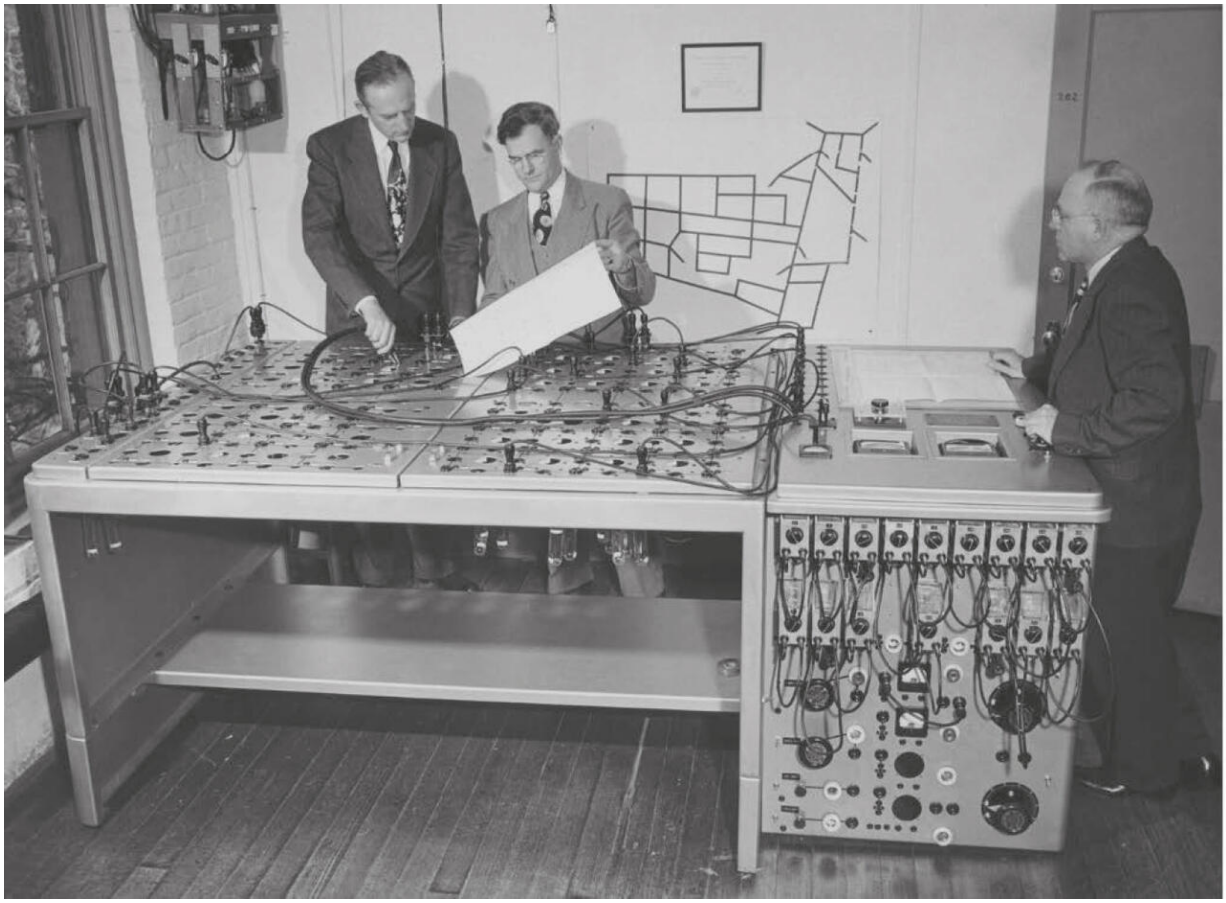


Figure 15.8. The McIlroy pipeline network analyzer, 1951 (Cornell University of Engineering photographs, #16-2-2378. Division of Rare and Manuscript Collections, Cornell University Library, with permission)

> "*The System will be used to simulate steady state conditions in a pipeline network. It is used by water and gas utilities to determine, in advance, a pipeline system's performance under projected conditions of use. The analog computer will also be used to train students in the operation of computing equipment.*"

The scaling process is quite noteworthy as problems were scaled so that the brightest Fluistor was glowing cherry red. Any brighter color required a rescaling process in order not to damage the fragile Fluistors.

Obviously, pipeline network simulators were typically large machines. →Figure 15.9 shows a German system developed by the company *Montan-Forschung Dr. Hans Ziller GmbH*, founded in the early 1950s. About 15 of these machines were built in the years until 1962 and some were in operation until the 1990s. Initially, these machines were

mostly used for the development of underground mine ventilation systems. Later applications involved the simulation of water distribution networks, gas pipeline systems, etc. These systems were not based on Fluistors but instead used a more elaborate nonlinear computing element, based on a servo circuit, the value range of which could be easily selected by switches.[70]

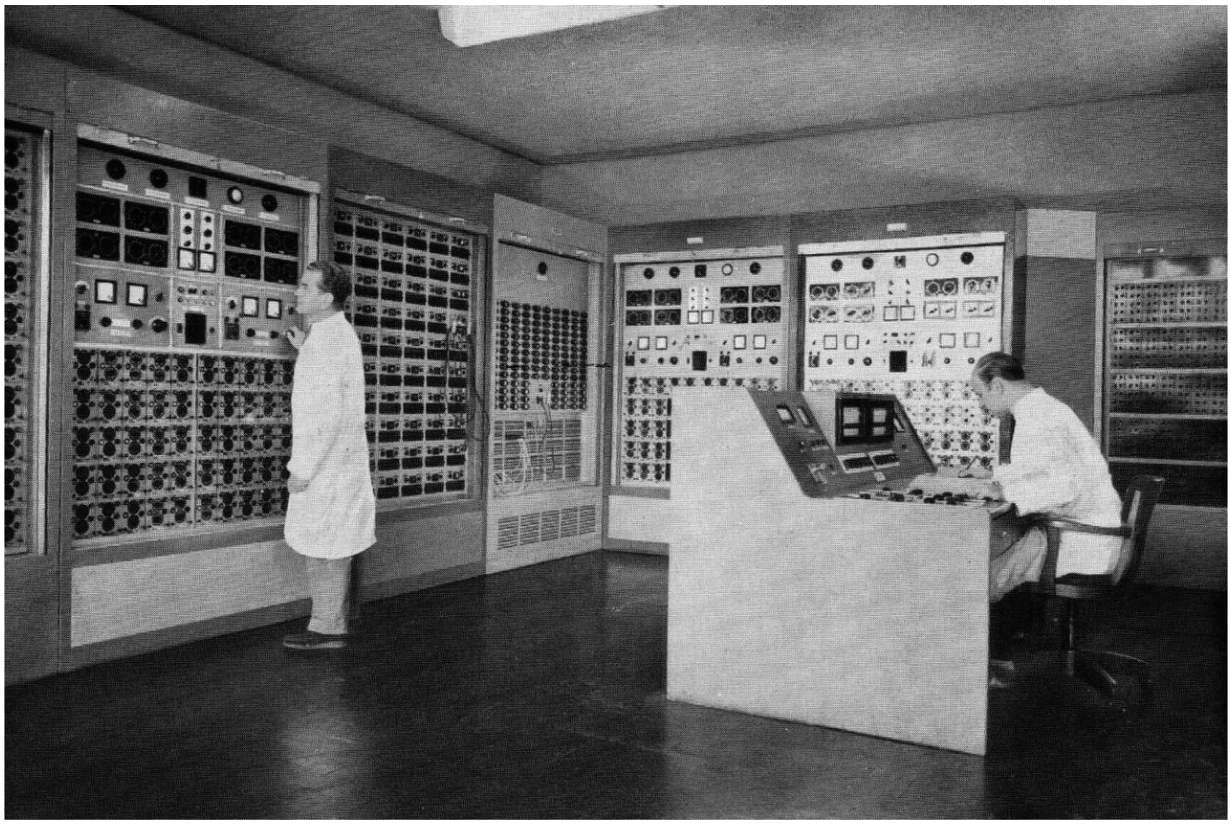


Figure 15.9. Large analog computer installation for the simulation of pipeline networks

Another large installation of such a system is shown in →figure 15.10. The walls are covered by racks holding the nonlinearities, typically in the four topmost rows, with voltage and current sources and sinks mounted below.[71]



Figure 15.10. Another large pipeline simulation computer (see [→WERNER 1968])

The *Southern Gas Association Analog*, another large and long lived pipeline network simulator, is described in detail in [→ASME 1990]. The first incarnation of this computer was put into operation in 1955 and was continuously expanded and modernized. It could simulate systems containing up to 4000 pipe sections and was in operation until 1990 in the form of a hybrid computer.

[→KLEINERT 1979/1] describes a complex hybrid computer simulation of gas flow through a long pipe using a time discretization approach for the solution of the underlying PDE. The simulation of the stationary laminar boundary layer in curved pipes is described in [→WEISS 1976]. This problem requires the solution of a system of 13 coupled differential equations of order 3.

15.1.13 Display of complex shapes

An intriguing application of analog computers is the display of multidimensional shapes on an oscilloscope. A simple example for such a display has been described in →section 8.10. →Figure 15.11 shows a

more complex example, the projection of a four-dimensional hyper-cube, a *tesseract*, generated by a high-speed analog computer.[72] In addition to this, even stereographic projections were implemented with analog computers.[73]

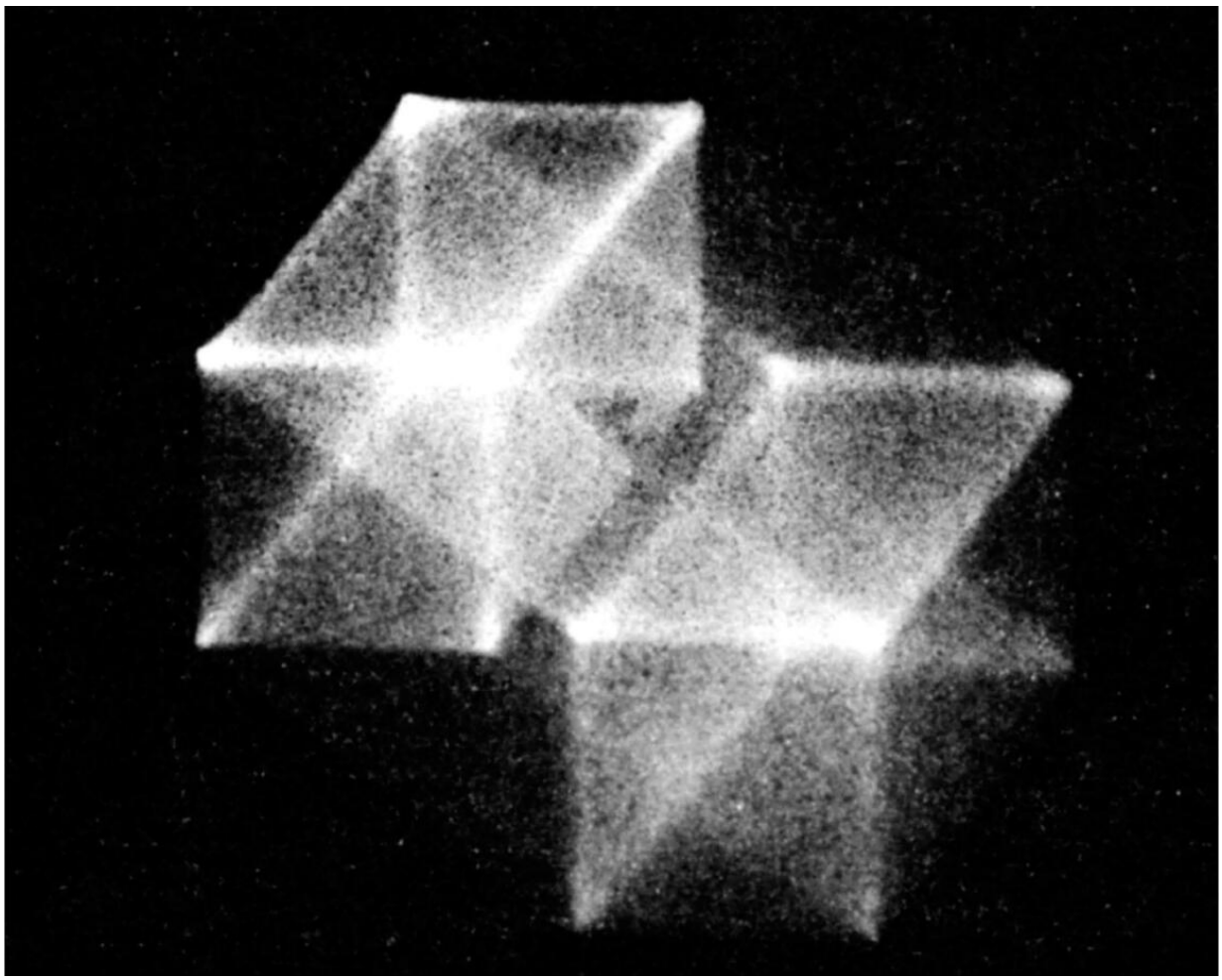


Figure 15.11. Display of a four-dimensional hyper-cube on an oscilloscope (see [→ MacKay et al. 1962, p. 137])

15.2 Physics

Analog computers are ideally suited for applications in physics due to their underlying principle of operation. The following sections describe some typical uses.

15.2.1 Orbit calculations

Many-body problems arising in astronomy and space flight are well suited to being solved on analog computers. For example, rendezvous simulations were of prime importance during the early days of space flight.[74] →Figure 15.12 shows chaotic orbits in a two-star system.

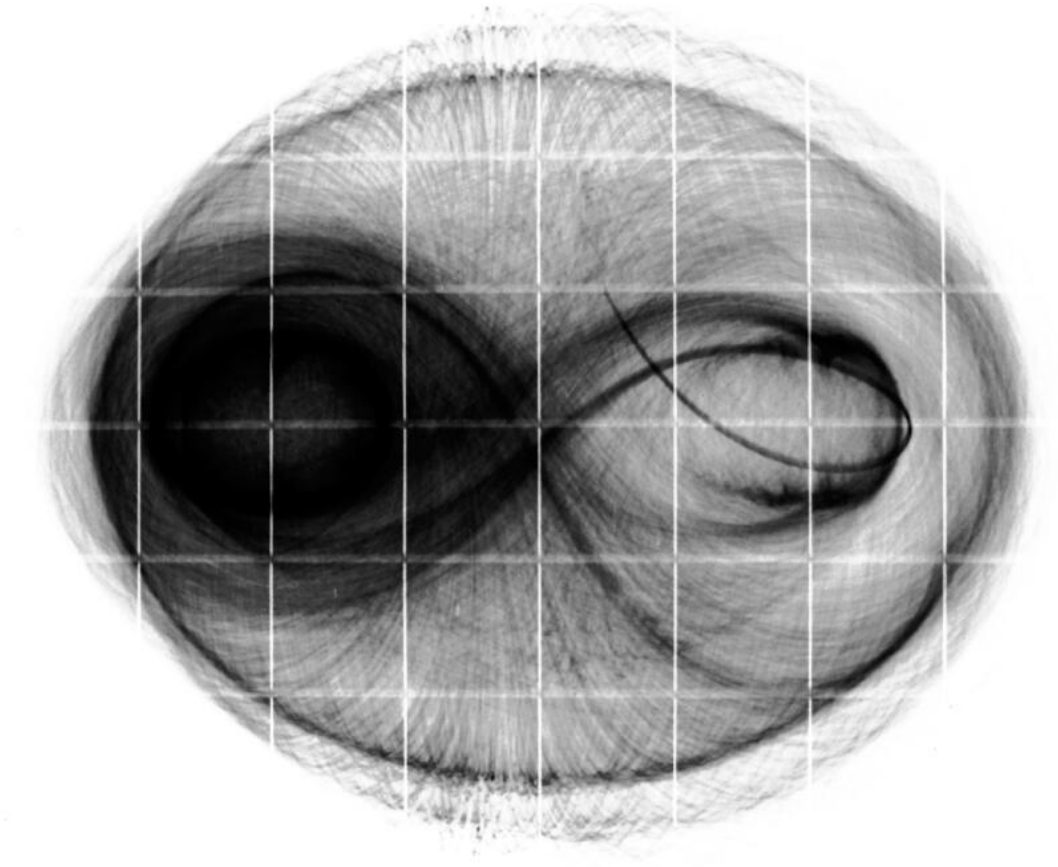


Figure 15.12. Chaotic orbits in a two-star system (see [→Ulmann 2017])

15.2.2 Particle trajectories and plasma physics

Two typical applications from nuclear physics are the calculation of particle trajectories in a magnetic field and beam control in particle accelerators.[75]

The following example shows the simulation of charged particle trajectories in a magnetic field – a problem typically arising during the

development and operation of particle accelerators. Large-scale simulations like this were performed at *DESY*,[76] using at first a large *EAI 231RV* analog computer, which was later replaced by a Telefunken RA 770 based hybrid computer system.[77]

A charged particle moving in a magnetic field is subject to a LORENTZ force[78]

$$\vec{F}_L = q(\vec{v} \times \vec{B})$$

with q representing the particle's charge, $\vec{v}$ its velocity, and $\vec{B}$ the magnetic flux density. Due to F_L being orthogonal to v, $|v|$ is preserved, but in a constant magnetic field, the particle is forced into a circular trajectory. This effect is shown schematically in →figure 15.13 where a particle is injected diagonally into a field of constant magnetic flux. →Figure 15.14 shows the analog computer setup for computing trajectories like this with a typical simulation output shown in →figure 15.15. Using a digital control system or the digital computer part of a hybrid computer, various trajectories, corresponding to different particle injection speeds $\dot{x}$, can be generated.

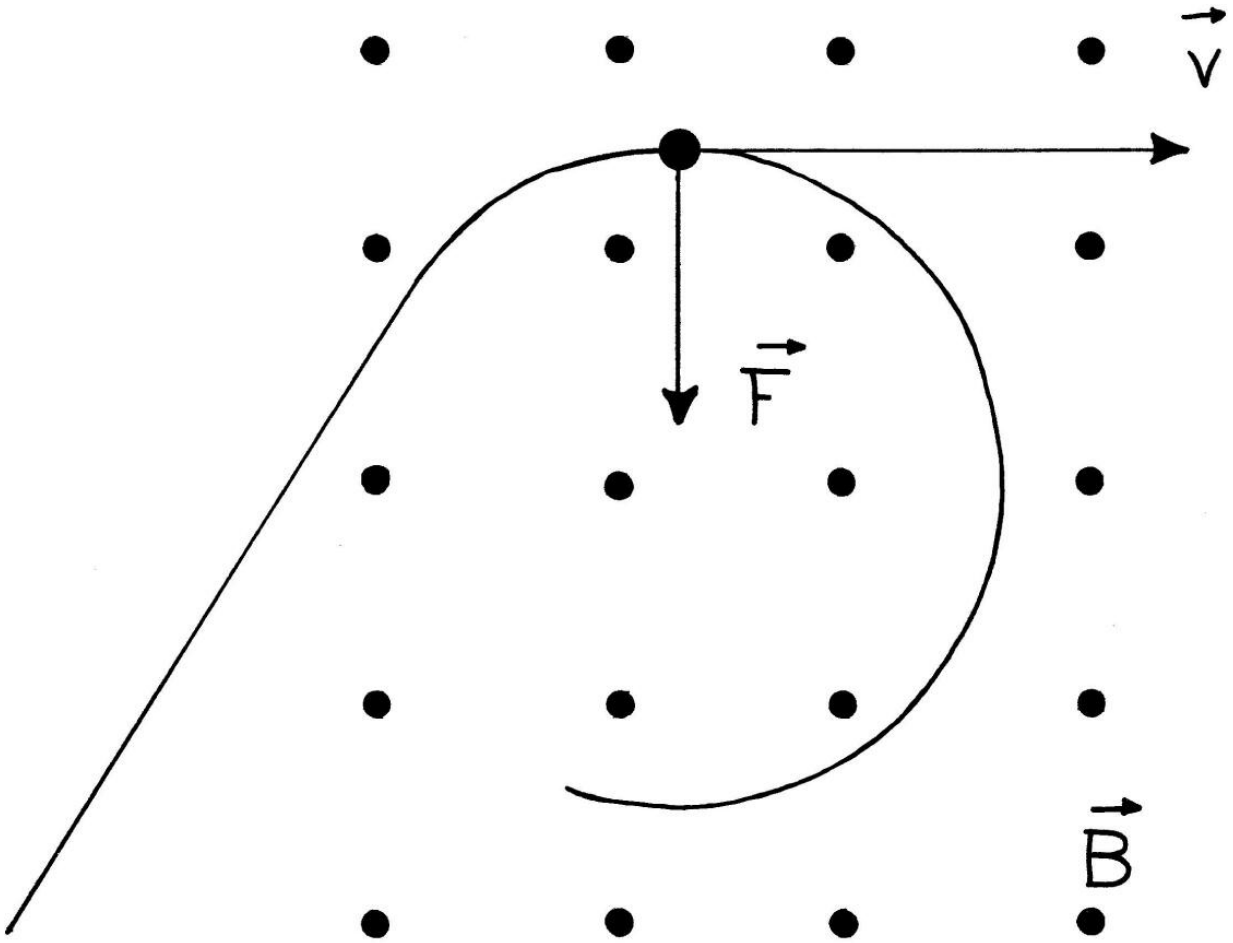


Figure 15.13. Trajectory of a charged particle in a magnetic field (see [→ BORCHARDT[b], p. 1])

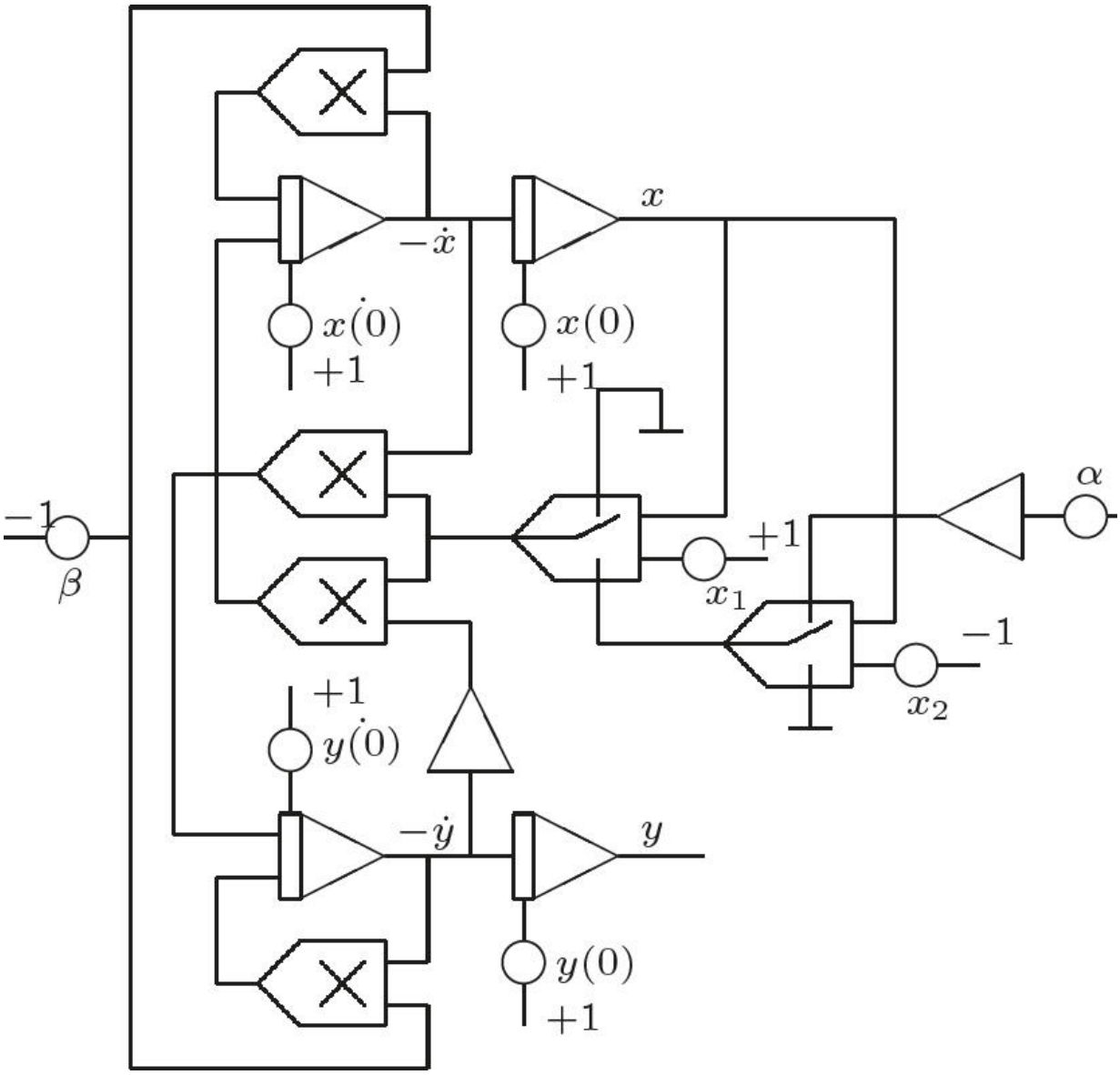


Figure 15.14. Computer setup for simulating the trajectory of a charged particle in a magnetic field (see [→ BORCHARDT[c], p. 6])

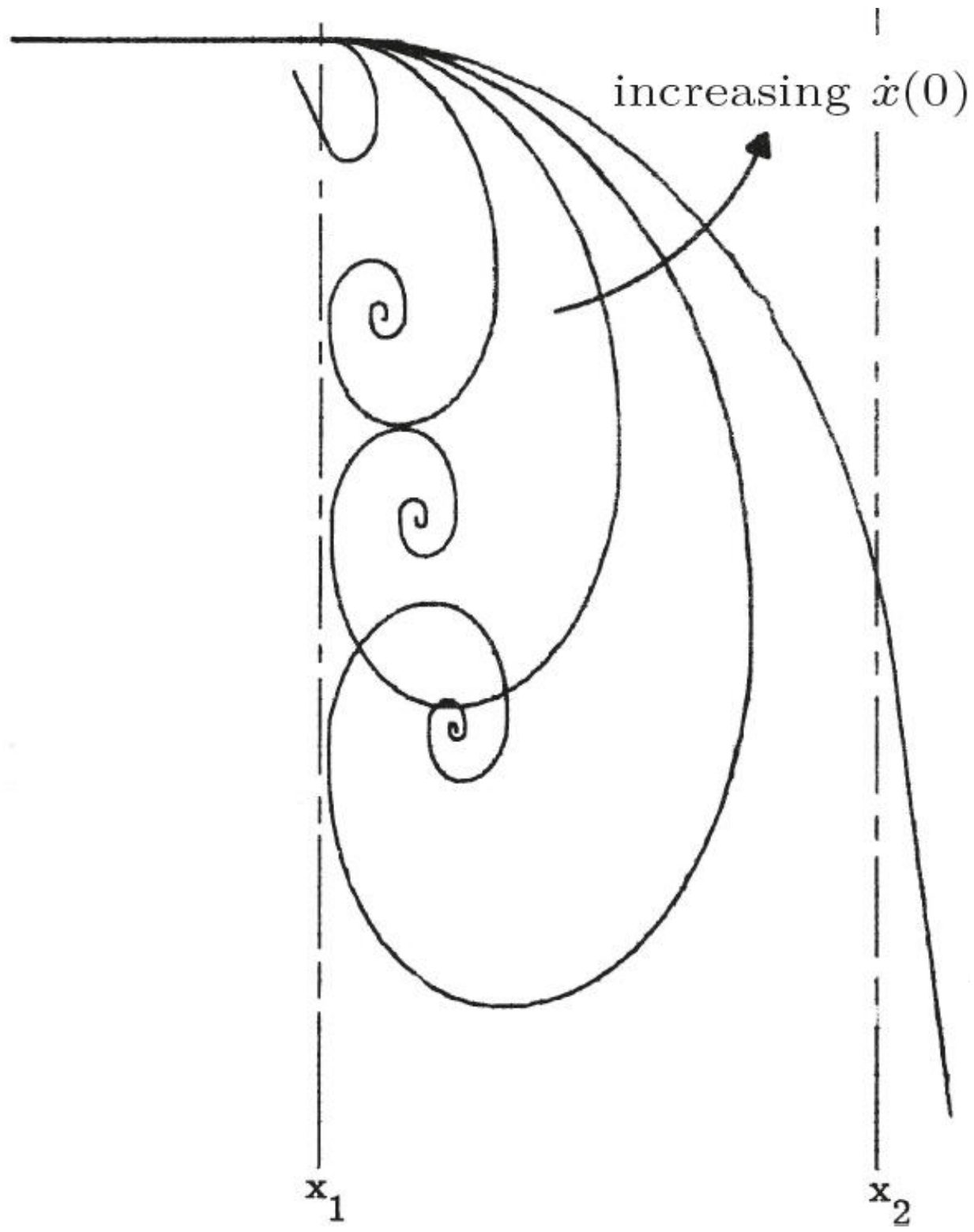


Figure 15.15. Various trajectories of charged particles in a magnetic field (see [→ BORCHARDT[d], p. 4])

[→Shen 1970] and [→Shen et al. 1970] describe the application of analog computers to problems in plasma physics.[79] Using only analog computer techniques several interesting observations can be made, which can be also observed in real plasma experiments:[80]

> *"[...] we observe: An 'anomalous drift' motion in nonstatic magnetic field, cyclotron resonance as well as subharmonic cyclotron resonance, [and the] violation of the commonly used definition for the adiabatic invariance of the magnetic moment in a nonstatic but spatially uniform magnetic field."*

An interesting problem, the simulation of the magnetic field created in a special purpose coil system for use in a neutron spin torquer on a hybrid computer is described in [→Rattay 1977]. Using digitally controlled function generators, the magnetic fields corresponding to various coil geometries are computed with the goal of implementing a certain magnetic field pattern.

15.2.3 Optics

A basic yet instructive application of analog computers in the field of optics is described in [→Rudnicki 1977]. This work describes the simulation of a laser interferometer based on a modulated Helium-Neon (HeNe) gas laser. Using two oscillator setups like the one shown in →figure 8.2, the laser beam as well as the modulation oscillator are modeled, thus simulating the interferometer on a wave basis.

15.2.4 Heat-transfer

The solution of heat-transfer problems invariably involves partial differential equations typically requiring a large number of computing elements when a discretization approach as shown in →section 7.4.1 is used. A great overview on heat transfer and fluid flow in general with a comprehensive bibliography can be found in the survey articles [→Nolan 1954/1], [→Nolan 1954/2], and [→Nolan 1955].

A practical application example for a complex heat-transfer simulation is given in [→James et al. 1971, pp. 193 ff.] where the design of heat sinks for radioisotope thermoelectric generators to be used in unmanned spacecraft is described. The temperature on one side of the

heat sink is determined solely by the decay heat of the radioisotopes. The variable parameter of the optimization problem is the thickness of the heat sink. This simulation was implemented by associating machine time τ with the position x on a heat sink cross section.

[→Swithenbank 1960] describes a special purpose analog computer for the simulation of regenerative heat exchangers.

A nice introductory example can be found in [→Telefunken 1963/1], which describes the simulation of the radial heat distribution in a loaded power cable using a finite difference approach to model the cable's geometry. [→Valisalo et al. 1982] describe a heat distribution analysis of irregularly shaped two-dimensional structures using a Monte-Carlo approach on a hybrid computer.

Two special purpose direct analog computers for the analysis of heat-transfer problems, the *Electronic Analog Frost Computor* (*EAFCOM*) and the *Heat Exchange Transient Analog Computer* (*HETAC*), are shown in →figures 15.16 and →15.17. The EAFCOM was used by the US Army Corps of Engineers to simulate soil freezing and associated effects.[81] It is a hardware implementation of a two-dimensional quotient of differences approach to heat conduction.[82] This grid consists of four different types of computing elements: *A*-elements are simple adders accepting values from their neighbor elements, *C*-elements are basically coefficient potentiometers, *J*-elements represent integrators, and *Z*-elements are used to model inert areas such as heat sources or sinks.

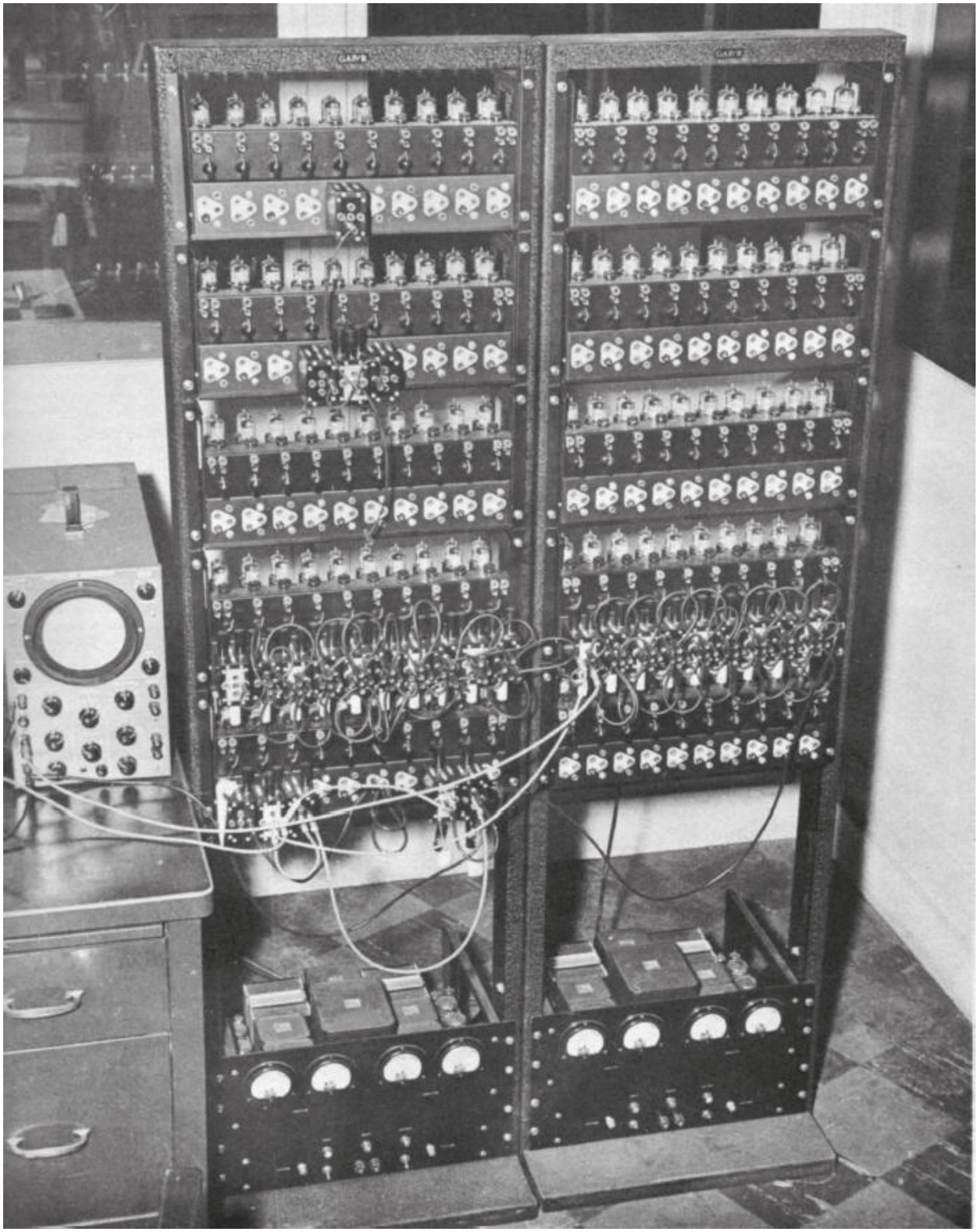


Figure 15.16. The *Electronic Analog Frost Computor*, US Army Corps of Engineers, New England Division (see [→ALDRICH et al. 1955, p. 259])

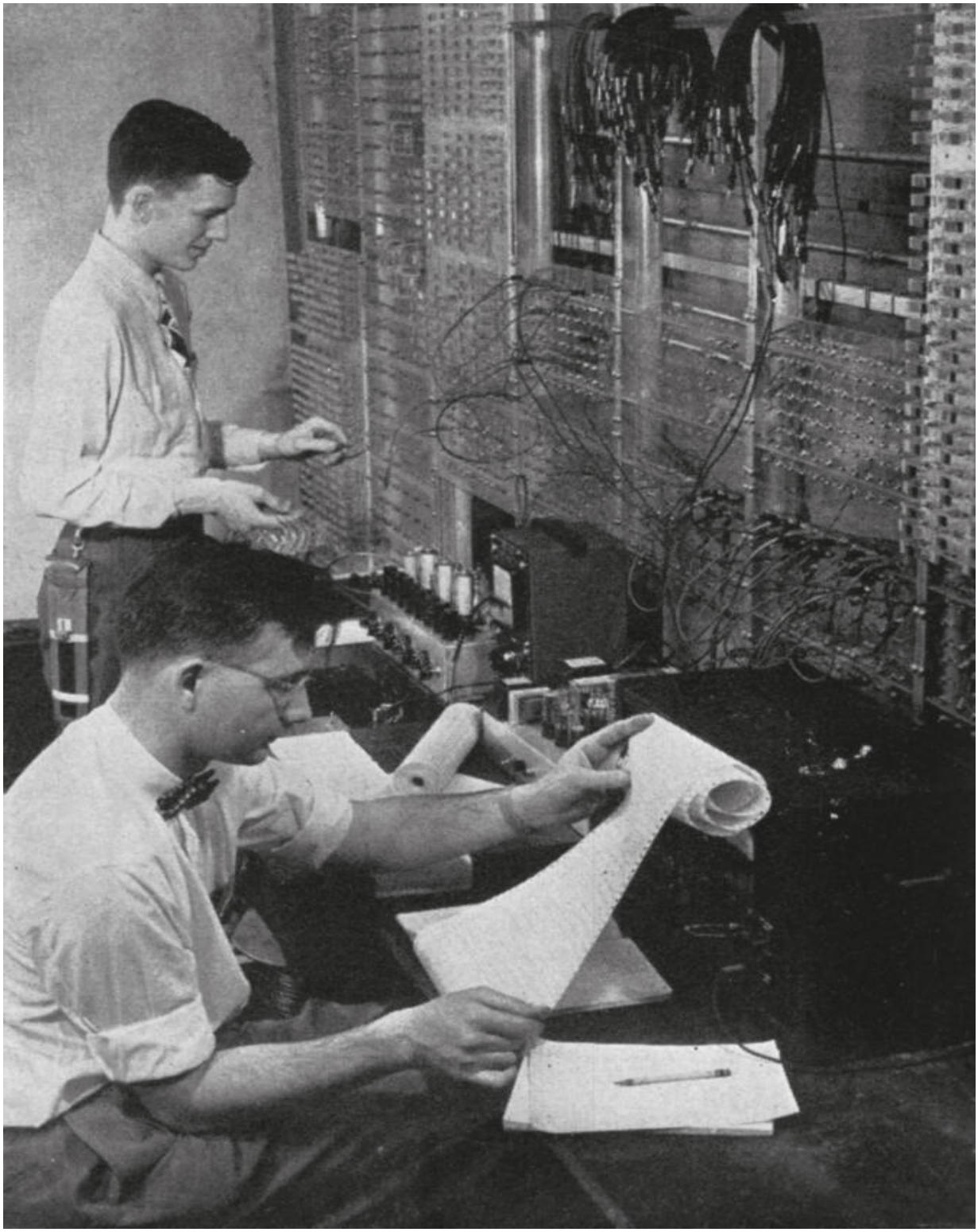


Figure 15.17. The *Heat Exchange Transient Analog Computer, HETAC* for short (see [→Instruments and Automation 1957/4])

One A-, J-, Z-, and two C-elements form a macrocell representing one layer of soil. These macrocells are connected not only with their direct neighbors but also with more remote elements representing thermal bridges or water flowing between layers.[83]

HETAC, developed at the *University of Virginia*, also is a two-dimensional implementation of a quotient of differences approach for solving the heat-transfer equation.

[→KERR 1978] and [→KERR 1980] describe analog computer models for the analysis of convection currents developing on inclined heat-exchanger surfaces. Using a small *EAI 380* analog computer solutions for this problem could be generated at a frequency of 500 Hz; this generated a flicker-free display of solution curves on an oscilloscope. Parameters of the simulation could be changed manually while observing their influence on the underlying system.

An interesting zone melting simulation is described in [→CARLSON et al. 1967, pp. 318 ff.]. Applications like this were essential for the rapid development of semiconductor technology, which required high-purity materials like germanium or silicon. These raw materials are usually purified using a zone melting technique, which depends on a variety of parameters, including the diameter of the germanium or silicon monocrystal, the width of the melting zone, the speed at which this zone moves through the crystal, etc. These parameters could be determined using analog computer simulations.

Similar techniques were also applied to the development of building structures. An analog computer setup simulating the "*one-dimensional heat gain through a flat, composite roof section into an air-conditioned space below*" is described in [→HEADRICK et al. 1969]. The simulation of heat and cooling loads using digital and analog approaches is detailed in [→BORDES et al. 1967].

A fascinating one-of-a-kind special purpose analog computer, the *Phytotron*, is shown in →figure 15.18. It was used to develop modern and energy efficient greenhouses and is described in detail in [→KOPPE et al. 1971] and [→EUSER 1966].

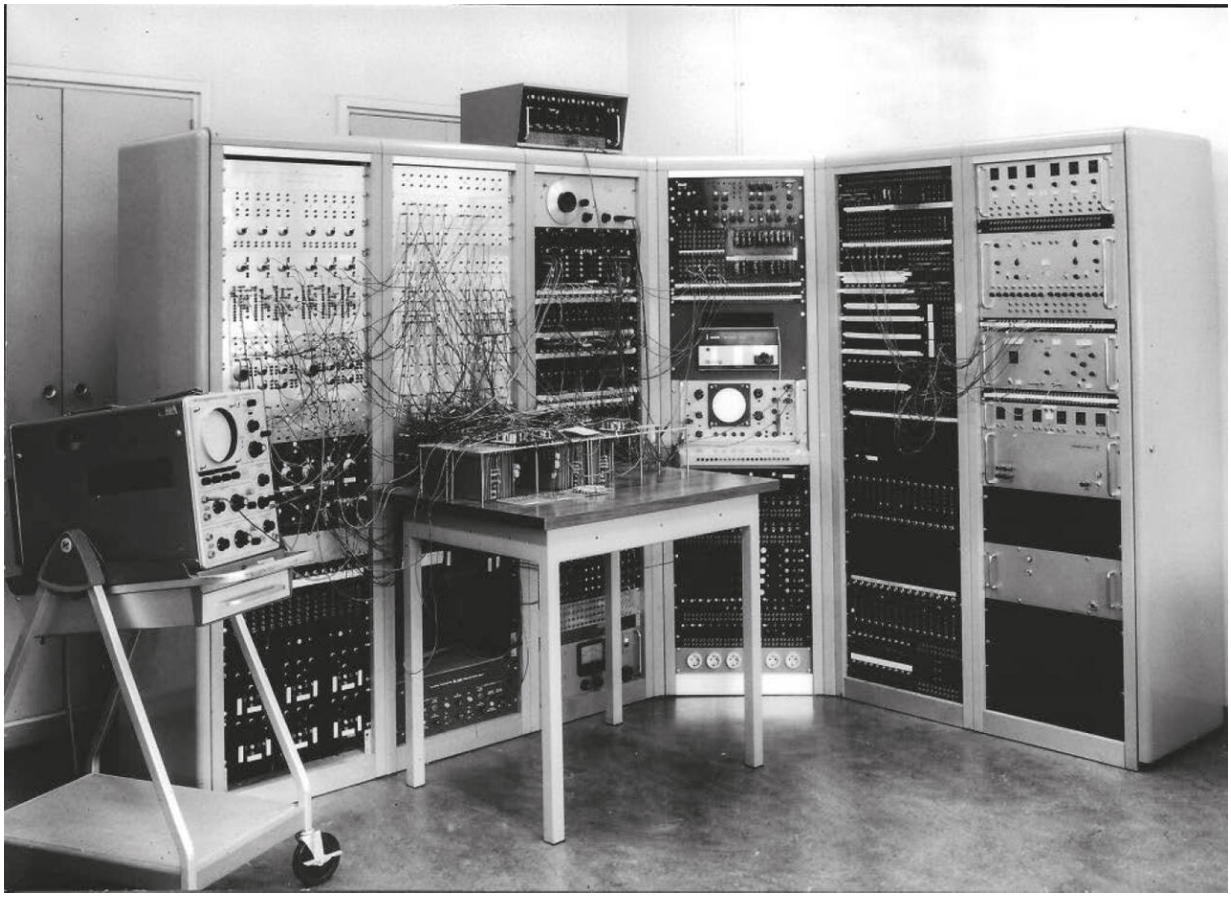


Figure 15.18. The Phytotron analog computer (picture by Bert Brouwer)

[→Bryant et al. 1966] describe a complex large-scale simulation of a co-current laminar double-pipe heat exchanger by means of a digitally controlled trial-and-error technique. The heat exchanger is described by two Sturm[84]-Liouville[85] differential equations, which are solved on the analog computer while the digital system performs the parameter space search. The application of finite Fourier transforms to analog computer simulations with emphasis on heat transfer problems is described in detail in [→Liban 1962/1].

Maybe one of the most arcane applications of analog computers to heat transfer and related problems is described in [→Zyl 1964]. This describes the application of analog computer to the solution of nonlinear partial equations describing the self-heating of fishmeals.[86]

Another interesting approach to solving the partial differential equations describing heat transfer and similar problems was pursued by

Vladimir Lukyanov who developed a hydraulic integrator in 1936. He was involved with the construction of the Troitsk-Orsk and Kartaly-Magnitnaya railways in the Soviet Union in the 1920s. Due do the extreme cold experienced in these areas, concrete was initially only poured during summer, however cracks continued to form in the structures, which posed a serious problem to the projects.[87]

To solve the heat transfer problem underlying these phenomena Lukyanov built a large hydraulic analog computer shown in →figure 15.19. This system was so successful that copies of it were used will into the 1970s in the Soviet Union. A detailed description of its construction and application can be found in [→Lukyanov 1939].[88]

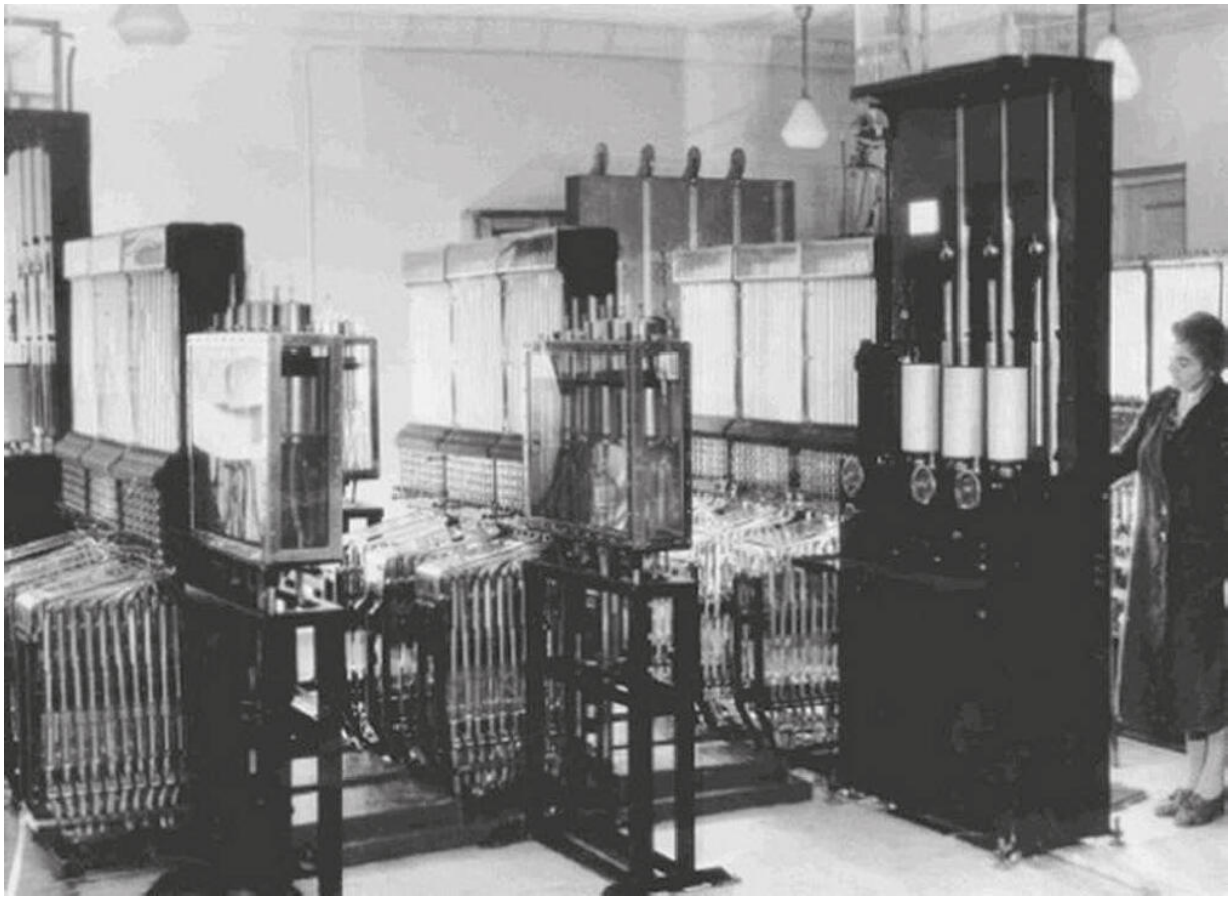


Figure 15.19. Hydraulic analog computer for solving heat transfer problems

An interesting point is made in [→HINTERGRÄBER 1967] where the simulation of a steam generator for a nuclear power plant is detailed. Instead of modelling the steam generator by a complex PDE it is simplified significantly and treated as a system consisting of two first order delays only. This is possible since the general behavior of the system is quite simple despite the underlying PDEs and dominated by the large masses of water, steam, and even the steel enclosure of the steam generator. Also nuclear power plant related is [→BONA 1967]. Here, a natural circulation drum boiler is modeled and implemented on an analog computer simulator system *APACHE*. . [→UNBEHAUEN 1967] not only deals with the simulation of a 100 t/h steam generator but also treats its behavior in conjunction with the surrounding control systems in a complex power plant setup. The resulting lumped parameter model is small enough to allow its implementation on a small Telefunken tabletop analog computer RAT 700.

Another interesting study can be found in [→HANUS 1967] where a 630 t/h steam generator is simulated using a rather unusual combination of active and passive analog computing elements including an RC-network consisting of 168 nodes with 64 buffer amplifiers for the treatment of the PDEs involved in the problem.

A study on optimal boiler control using PONTRYAGIN's[89] maximum principle using two coupled analog computers with a digital expansion unit is described in [→THOMAS et al. 1967]. This study also includes an experimental oil fueled boiler capable of producing 1.5 t/h of superheated steam coupled to and controlled by an EAI TR-48 analog computer. Of special interest is the behavior of the boiler under external large-scale disturbances. Even with a rather simple control system the overall system works quite well as long as the control system is reset regularly.

A rather classic approach to the simulation of a radioisotope thermoelectric generator (*RTG*) is described in [→LANDROT et al. 1967]. The heat transfer problem is solved by discretizing the underlying PDE along one spatial coordinate. Of special interest in this study is the comparison of heat transfer by conduction and heat transfer by radiation.

15.2.5 Fallout prediction and nuclear weapons effects

Surface testing of nuclear weapons required an accurate means of predicting radioactive fallout. A special purpose analog fallout predictor was developed in 1956 and is described in [→Skramstad 1957].[90] The system consisted primarily of a variety of function generators (*winds units*) and an elaborate scanning mechanism driving a two-dimensional oscilloscope display.[91] The underlying forecast problem was simplified considerably by ignoring vertical winds and assuming horizontal wind speeds and directions to be constant during the entire fallout period.[92]

The overall predictor and a typical fallout prediction result are shown in →figures 15.20 and →15.21. The brightness of the display corresponds to the expected intensity of fallout in the respective area. The structure of the fallout computer is shown in →figure 15.22.[93]

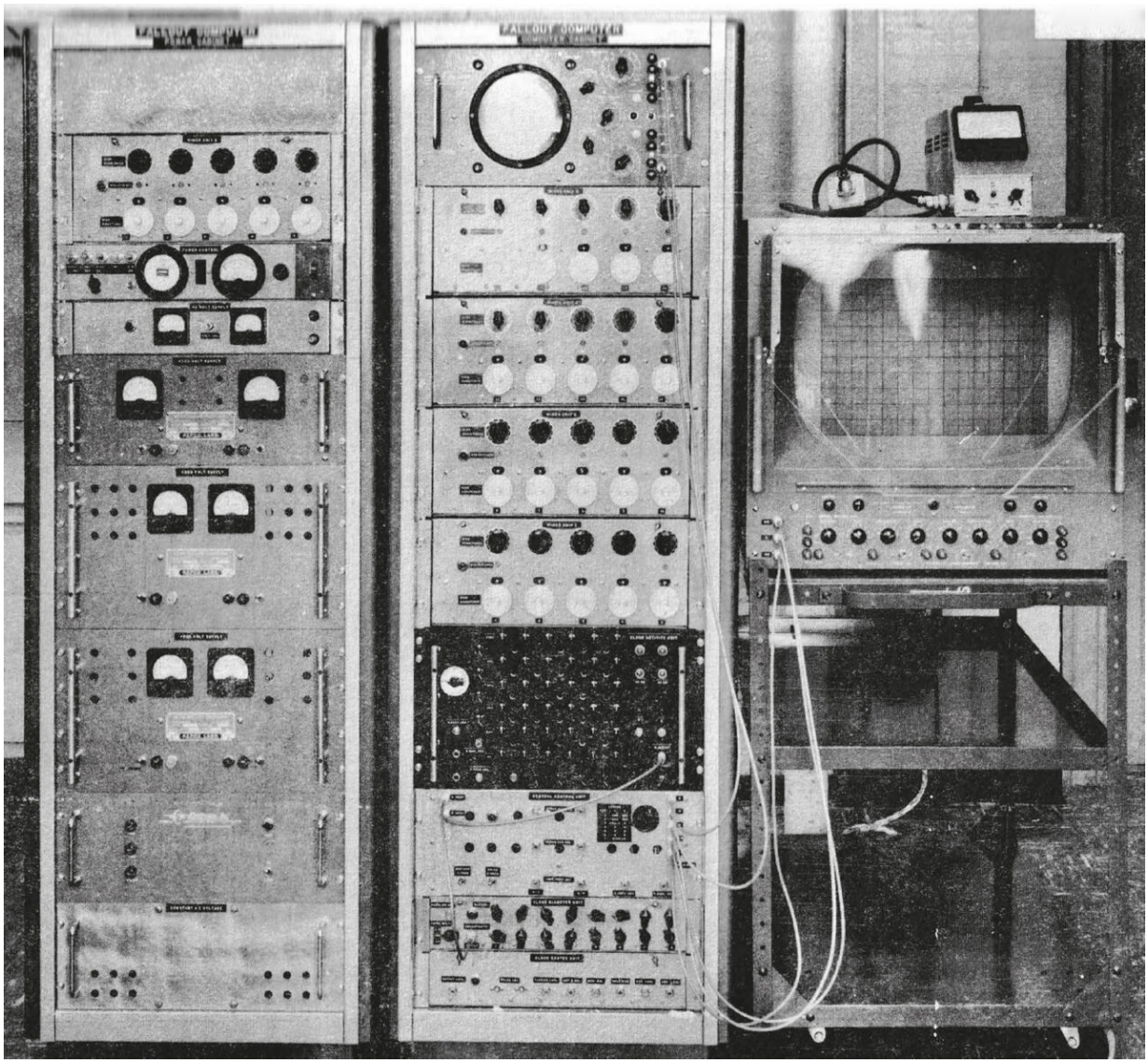


Figure 15.20. Fallout computer (see [→ Skramstad 1957, p. 103])

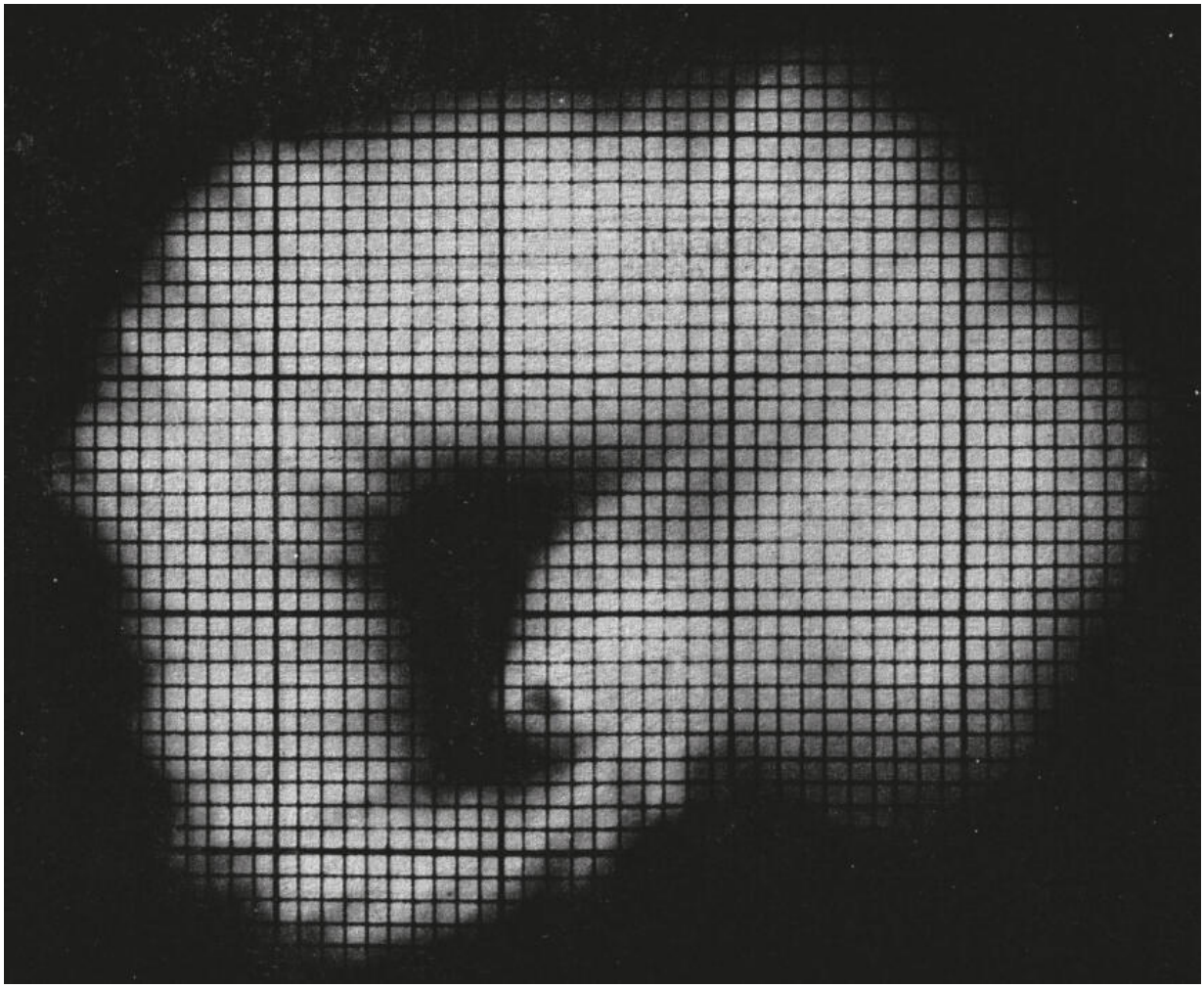


Figure 15.21. Typical predicted fallout pattern (see [→ SKRAMSTAD 1957, p. 102])

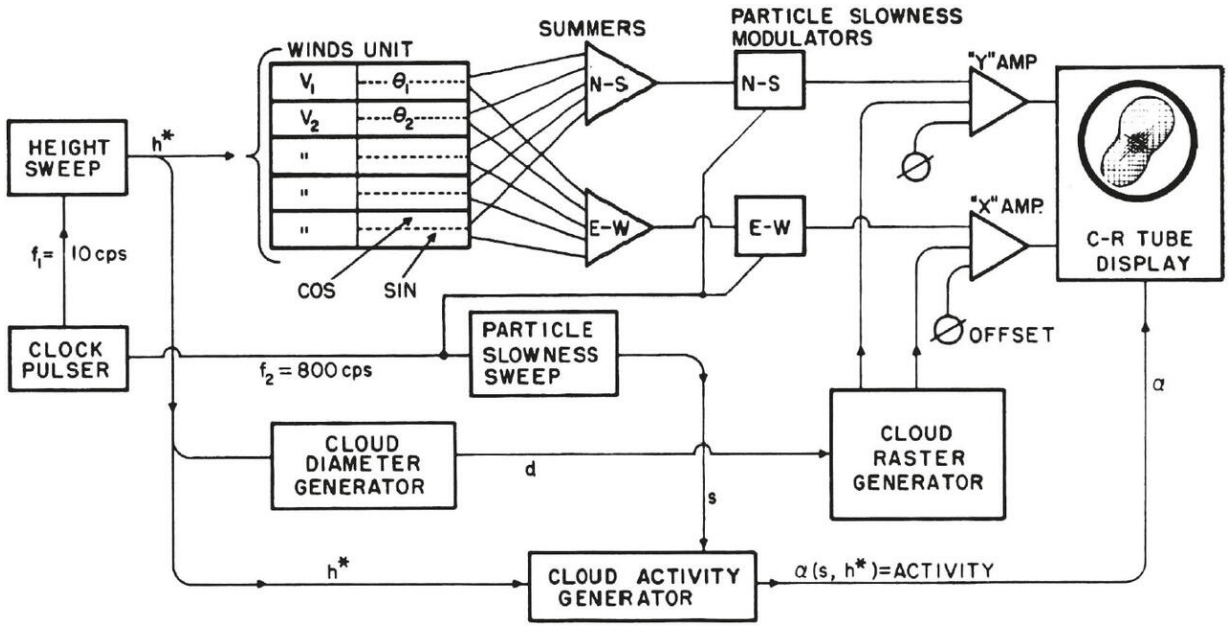


Figure 15.22. Block diagram of the fallout computer ([→SKRAMSTAD et al. 1961])

Another problem encountered during the Cold War was that of predicting the immediate effects of nuclear weapons. Simple means such as specialized circular slide rules like the one shown in →figure 2.4 were neither sufficiently accurate nor did they take more than just a few basic nuclear weapons effects into account.[94] More precise means of predicting the effects of nuclear weapons, of utmost importance for the planning of their actual use, required tedious manual calculations based on complicated charts, nomograms, etc.

Accordingly, a *Weapons Effects Display System* (*WEDS*) was developed to simplify and substantially speed up these tasks.[95] This system was intended to be used by weapons design strategists, military strategists, and military field commanders[96] and predicted a number of effects such as thermal flux, blast, radiation, etc.[97] At its heart were four special purpose analog computers, a *blast computer* predicting blast pressure and a damage index, a *radiation computer* predicting the total radiation, a *thermal computer* computing the overall thermal flux on a target, and a *map computer* controlling an intricate display system where a projection of a map would be overlaid with effect displays as shown in →figure 15.23.

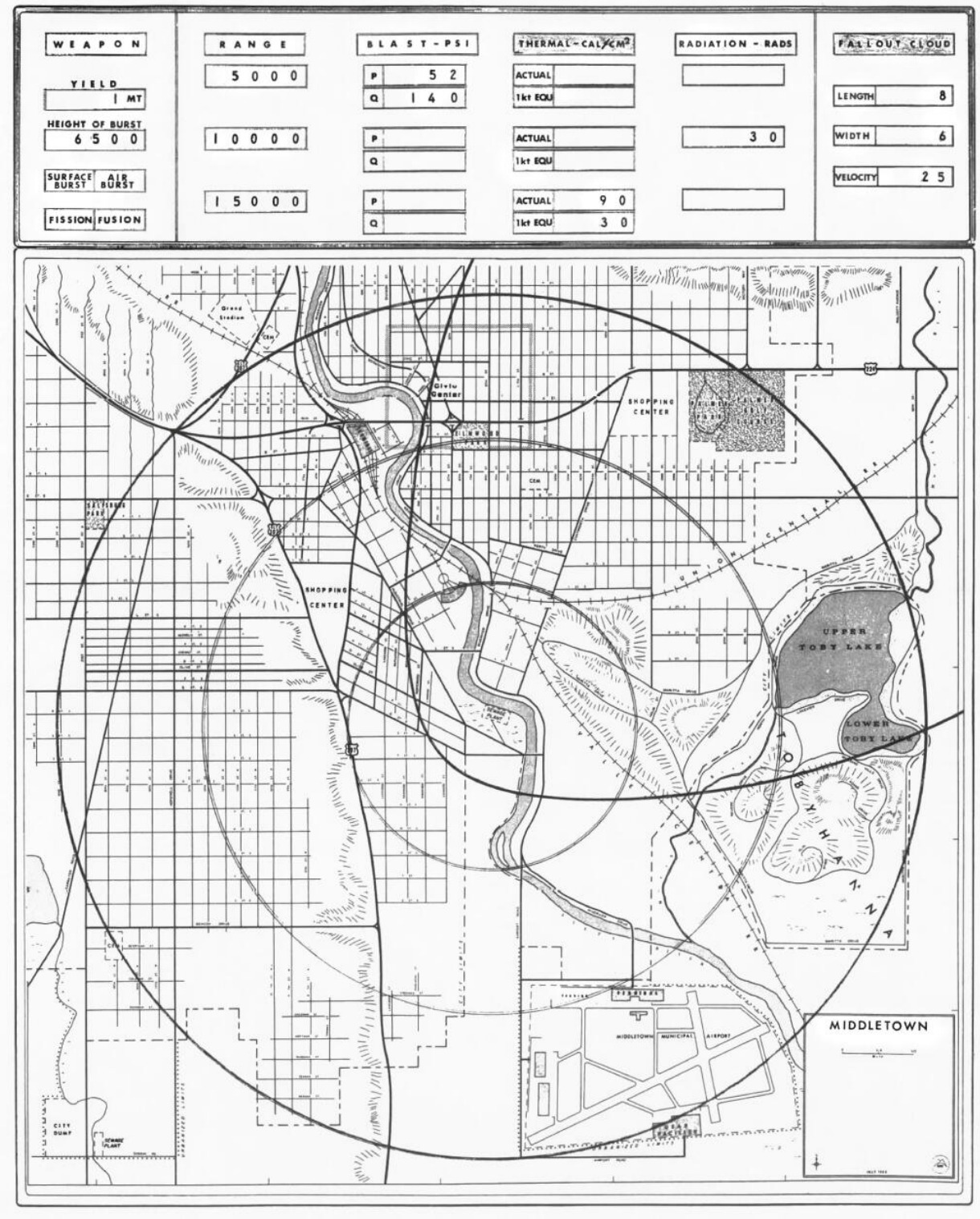


Figure 15.23. Weapons Effects Display System (see [→HENDERSON et al. 1965, p. 2-5])

The computation of these effects required a large number of logarithmic functions as well as fixed breakpoint diode function generators implementing functions for determining effects such as peak overpressure at ground level which are highly nonlinear. Some functions were implemented using servo driven barrel cams as described in →section 2.5.1.[98] User input was mainly done using linear potentiometers and manual switches.[99]

15.2.6 Semiconductor research

Analog computers were also used extensively in early semiconductor research. [→Apalovičová 1979] describes the simulation of the electric field in the depletion zone of a *Metal Oxide Semiconductor* (*MOS*) field effect transistor[100] in order to predict the behavior of different transistor structures. The solution of the underlying elliptic partial differential equations was done by a hybrid computer.[101]

[→Carlson et al. 1967, pp. 332 ff.] describe the analysis and simulation of the dynamic response of a *tunnel diode*.[102] An interesting aspect of this particular analog computer setup is that it contains an algebraic loop, which is stable due to a unique combination of circumstances.[103] Furthermore, it is a good example of a problem that happens too fast to be observed and analyzed in a real specimen, so an extremely slowed down machine time on an analog computer is required to gain the necessary insight.

[→Gummel et al. 1967] describe an interesting approach to simulating transistors by a “quasi analog computation”. The idea is to use a transistor as its own model, extended by a surrounding circuit consisting of an operational amplifier driving external capacitors simulating time-scaled carrier storage effects. This idea was further extended in [→Balaban et al. 1968]. →Figure 15.24 shows the model of a semiconductor diode based on this approach which is called the *breadboard method*. A real diode is used as a diode model for fast effects while a current sensing arrangement consisting of a resistor *r* and an operational amplifier with gain *A* control the current through a capacitor *C* implementing time-scaled carrier effects.

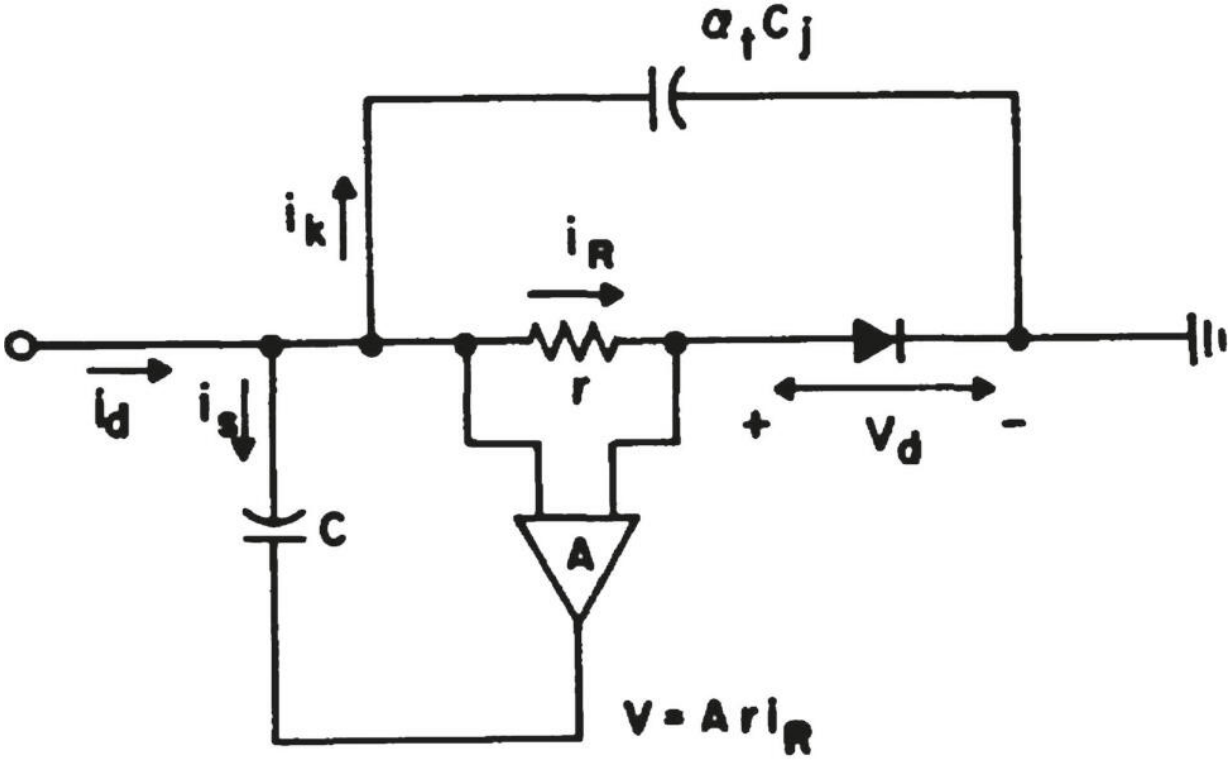


Figure 15.24. Breadboard method for the simulation of a semiconductor diode (see [→Balaban et al. 1968, p. 136])

A more abstract simulation setup for the simulation of a semiconductor diode is shown in →figure 15.25. The underlying idea is basically the same, using an actual diode which is so fast in comparison with scaled computer time that the slower effects can be implemented using analog computing elements. Using such an arrangement, time-scaling factors of up to 10^7 have been found to be practical. The paper also describes similar analog computer models for bipolar junction transistors.

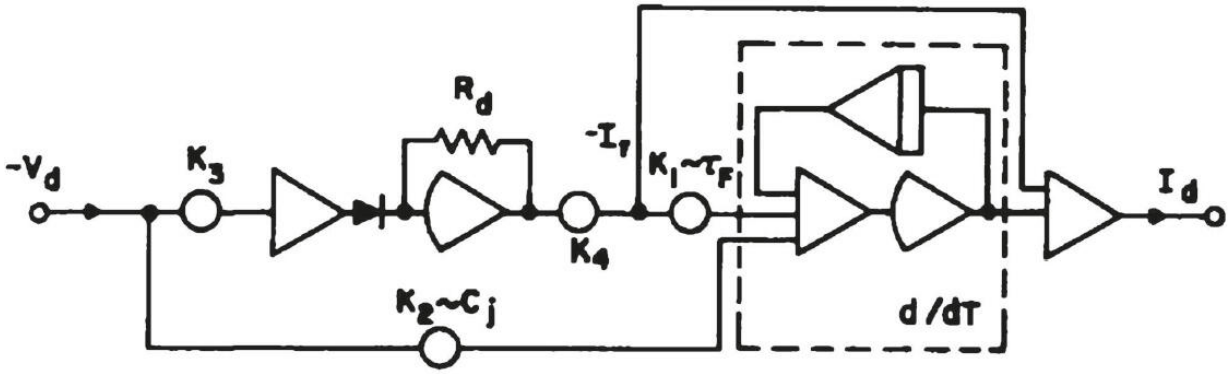


Figure 15.25. Analog computer model for a semiconductor diode (see [→ Balaban et al. 1968, p. 137])

Two such diode models can then be combined to create a realistic model of a bipolar junction transistor in a grounded emitter configuration as shown in →figure 15.26.

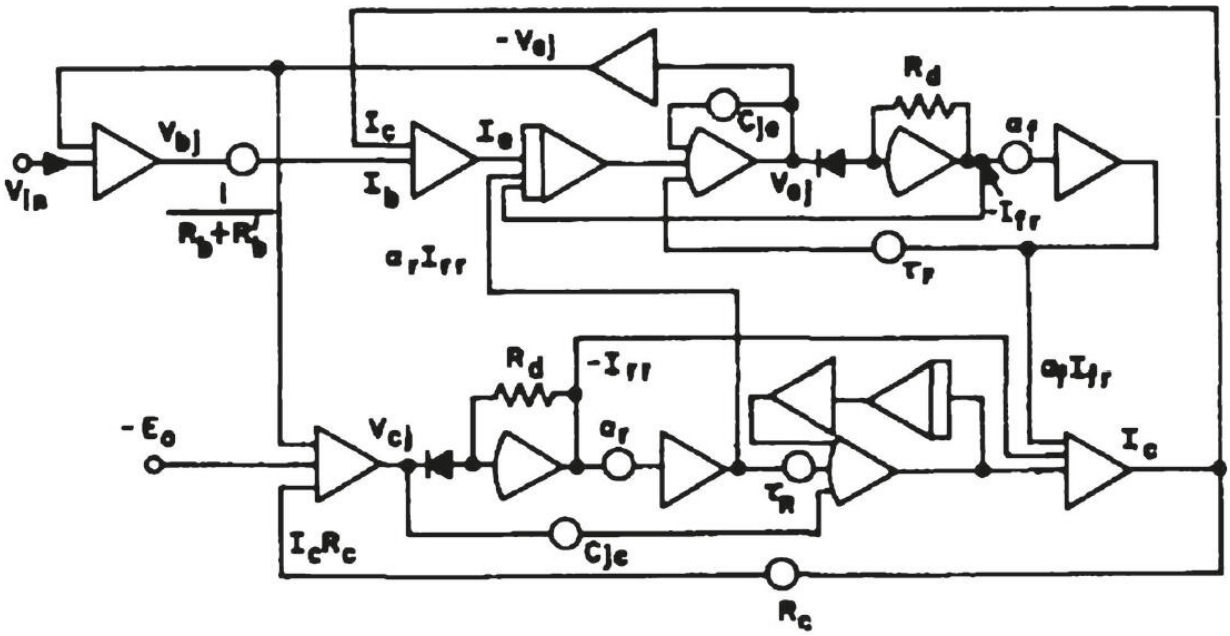


Figure 15.26. Analog computer model for a bipolar junction transistor in grounded emitter configuration (see [→ Balaban et al. 1968, p. 140])

The simulation of time dependent effects in semiconductor laser diodes using a hybrid computer is described in [→ Gründorfer 1974]. This study

aims at understanding effects such as *Q* switching[104] pulses of the laser after switching off the driving current, variable time delays between the application of power and the onset of the stimulated emission of light.

15.2.7 Ferromagnetic films

Ferromagnetic films were of great interest in the 1960s as prospective storage elements for digital computers, before static and dynamic semiconductor *Random Access Memory* (*RAM*) circuits eventually took over. An interesting research problem was the behavior of such ferromagnetic films under varying environmental conditions. [→Borsei et al. 1967] describe an analog computer approach to simulate the behavior of a ferromagnetic film subjected to mechanical stress.

15.3 Chemistry

The following sections describe two typical applications of analog computers in chemistry. For applications in chemical engineering refer to →section 15.13.

15.3.1 Reaction kinetics

Most chemical processes including enzymatically catalyzed processes in biochemistry[105] involve a variety of heavily interdependent reaction steps. Studies regarding such processes were mainly performed on analog computers well until the late 1970s/early 1980s because the underlying differential equations are ideally suited for this approach. A simple example problem is shown in →figure 15.27:[106] A substance *A* is to be converted into a product *D* involving an intermediate substance *B*, which in turn is partially transformed into *C* and vice versa. The rates at which these steps happen are k_1, k_2, k_3, and k_4.

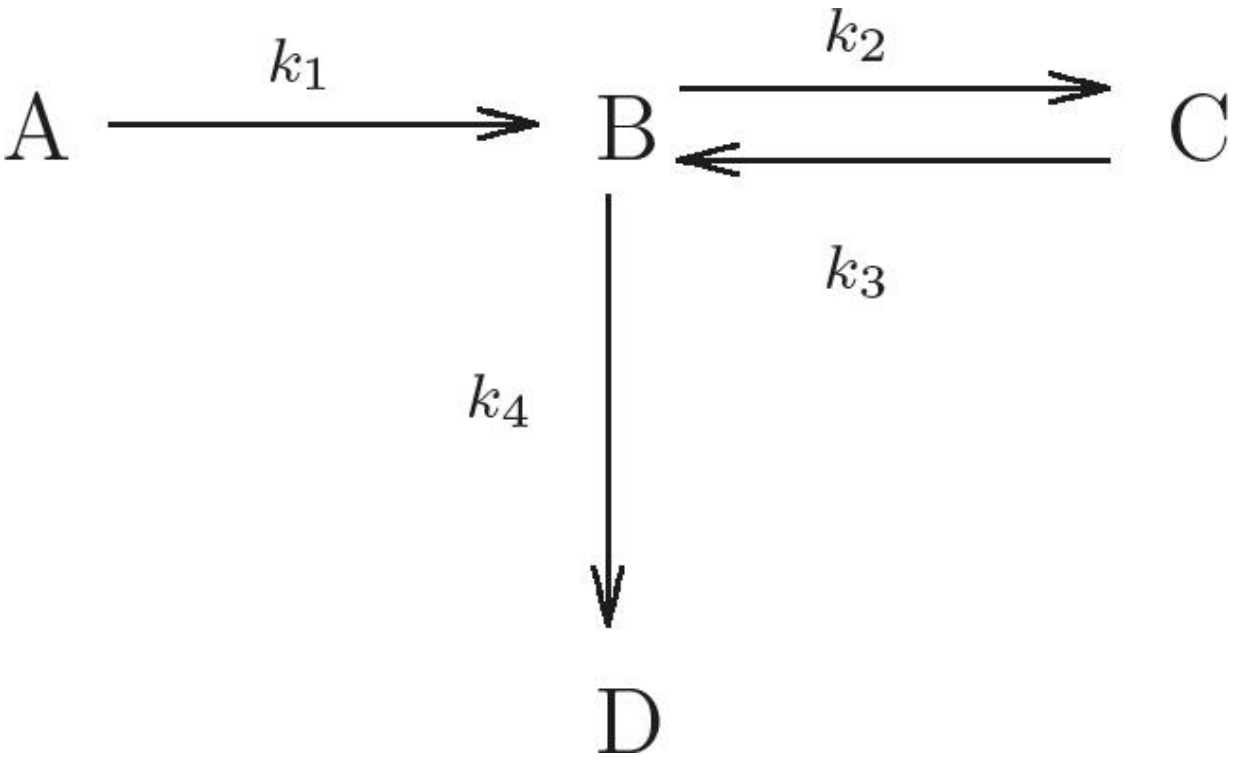


Figure 15.27. Simple reaction kinetics example involving four substances (see [→Dornier/2, p. 3])

Based on →figure 15.27 this reaction can be described by the following four coupled differential equations:

$$\begin{aligned}
\dot{A} &= -k_1 A \\
\dot{B} &= k_1 A - k_2 B - k_4 B + k_3 C \\
\dot{C} &= k_2 B - k_3 C \\
\dot{D} &= k_4 B
\end{aligned}$$

The result of a typical simulation run based on these equations with k_1 = k_2 = k_3 = k_4 = const. is shown in →figure 15.28. Starting with 100% of component *A*, the four parallel reactions set in, finally yielding the desired end product *D*. The corresponding simulation program is shown in →figure 15.29.[107]

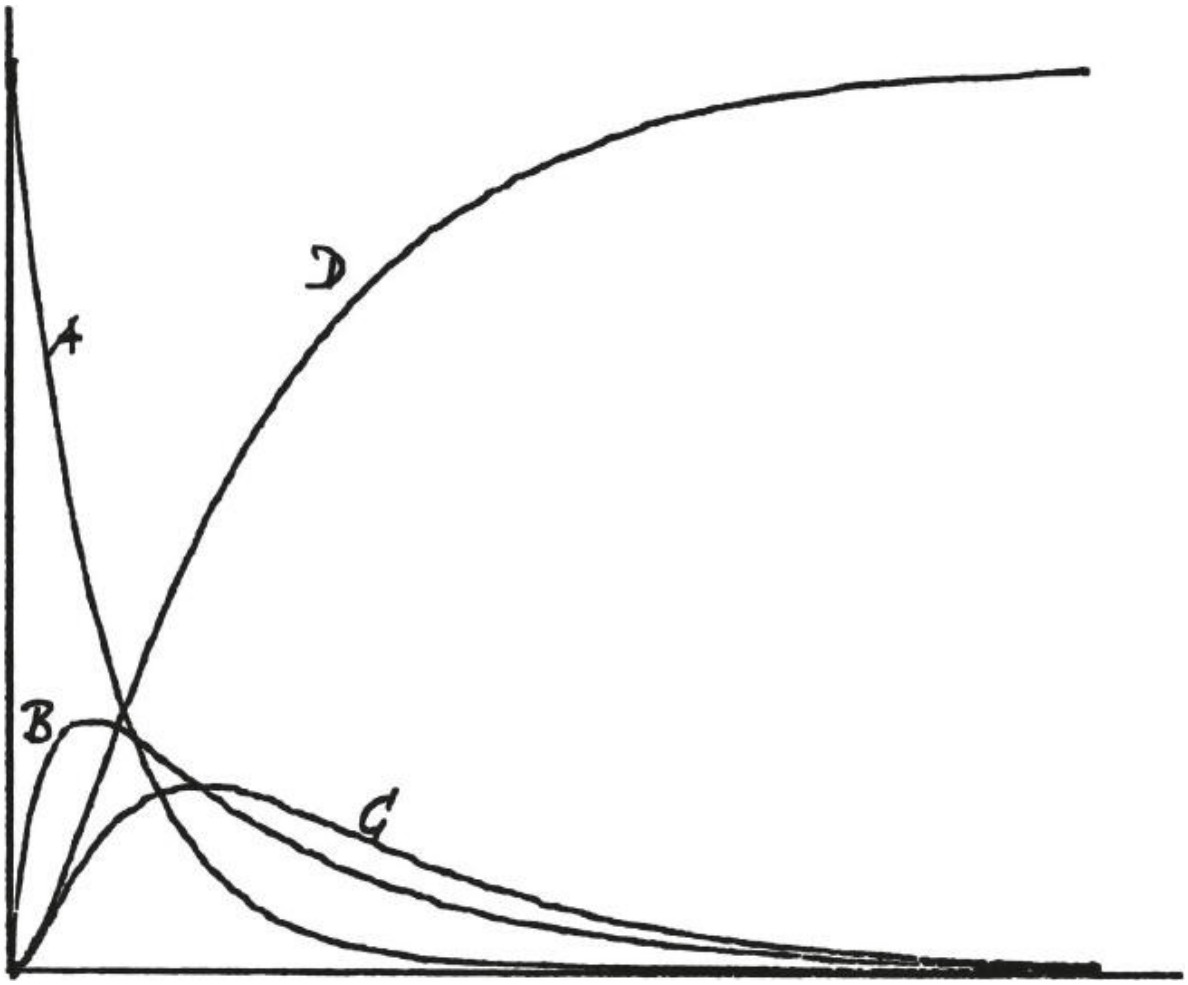


Figure 15.28. Reaction kinetics simulation result (see [→Dornier/2, p. 7])

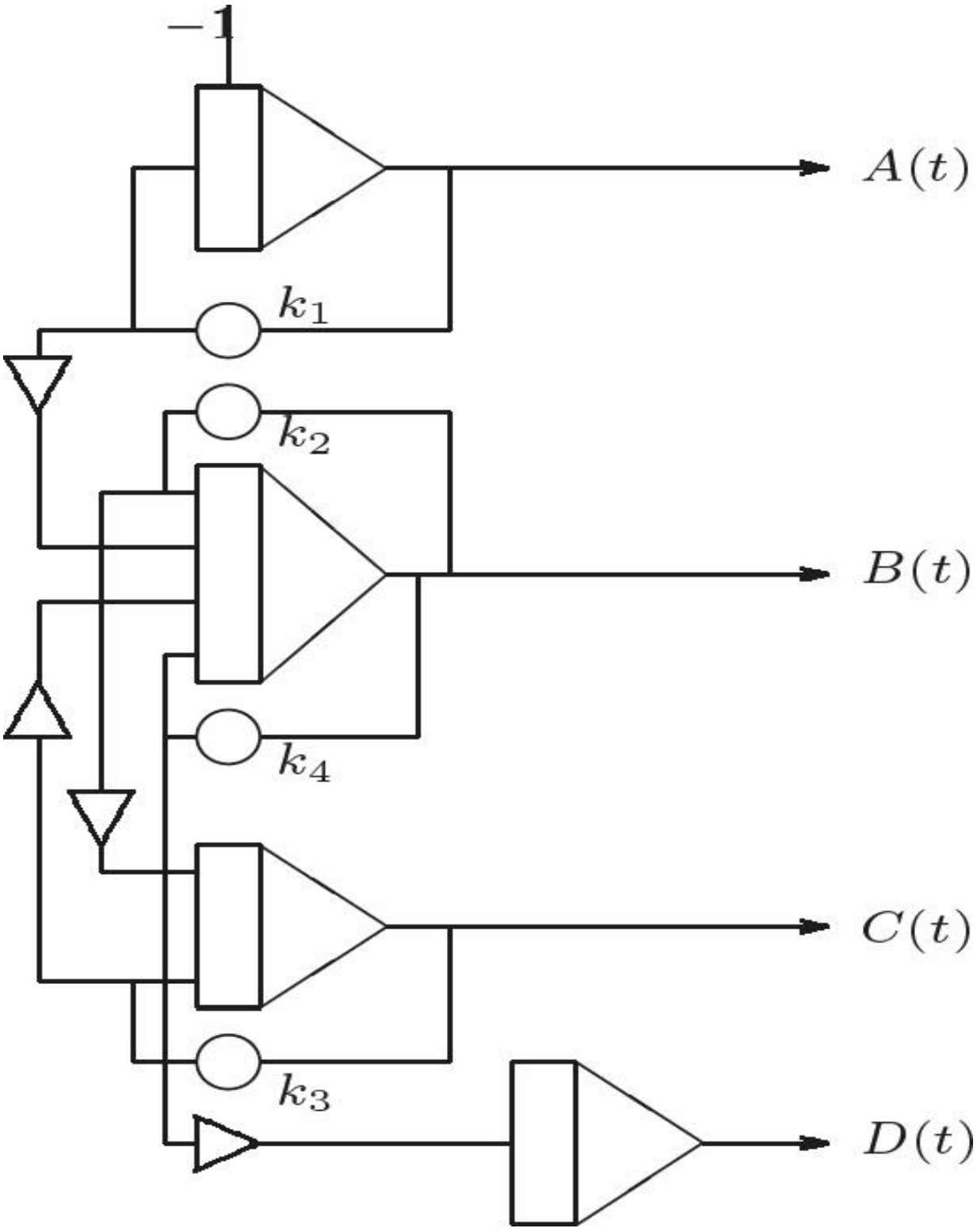


Figure 15.29. Reaction kinetics simulation program (see [→Dornier/2, p. 4])

Further interesting examples can be found in [→ Cheng[a]], where the simulation of polymerization processes using analog computers is described. Techniques to determine process parameters for the chlorination of methane are described in [→ Bassano et al. 1976], while [→ Ramirez 1976, pp. 74 ff.] describes the simulation of a large-scale benzene chlorination. [→ Köppel et al. 2022] is a recent work on the application of analog computers to molecular dynamics.

A large area of application for reaction kinetics studies can be found in biochemistry, medicine, etc., where enzymatically catalyzed reactions are of prime importance. A thorough introduction to this topic from the perspective of systems biology can be found in [→ Sauro 2019].

An interesting industrial application is described in [→ Sefcik et al. 1967] where the reaction kinetics of CO_2 with water steam over a ferrochromium catalyst in a gas shift reactor in a temperature range of 320–620°C at a pressure of 15 atm is modeled. An interesting aspect is that a digital computer is used for the preparation of the analog computer simulation. The simulation contains a number of algebraic equations which are typically not well suited for an analog computer but in this case are stable since rather slow servo multipliers are used at critical points within the loop stabilizing the overall setup. Eventually simulation results obtained by analog as well as digital means are compared and correspond to within 2%.

15.3.2 Quantum chemistry and mechanics

Direct analog computers proved very useful in the early days of quantum chemistry,[108] where solutions of the *Schrödinger*[109] *equation*, which are wave functions, are of prime interest.[110] These solutions determine the *electronic structure* of molecules, which in turn determine their respective properties.

In 1944, Gabriel Kron[111] suggested a direct analog electronic analog computer to model the behavior of the Schrödinger equation under different boundary conditions. A typical circuit derived for this purpose is shown in → figure 15.30.[112] Later, Hans Kuhn[113] developed an *acoustical analog computer* containing a multitude of coil springs mounted on a carrier. These springs were interconnected horizontally by leaf springs of equal moduli of resilience. This particular machine is described in detail

in [→ Preuss 1962, pp. 291 f.] and was used to simulate state transitions of π-electrons in complex shaped molecule chains. An easy treatment of the one-dimensional Schrödinger equation for educational purposes can be found in [→ Müller 1986] and [→ Hund 1982].[114]

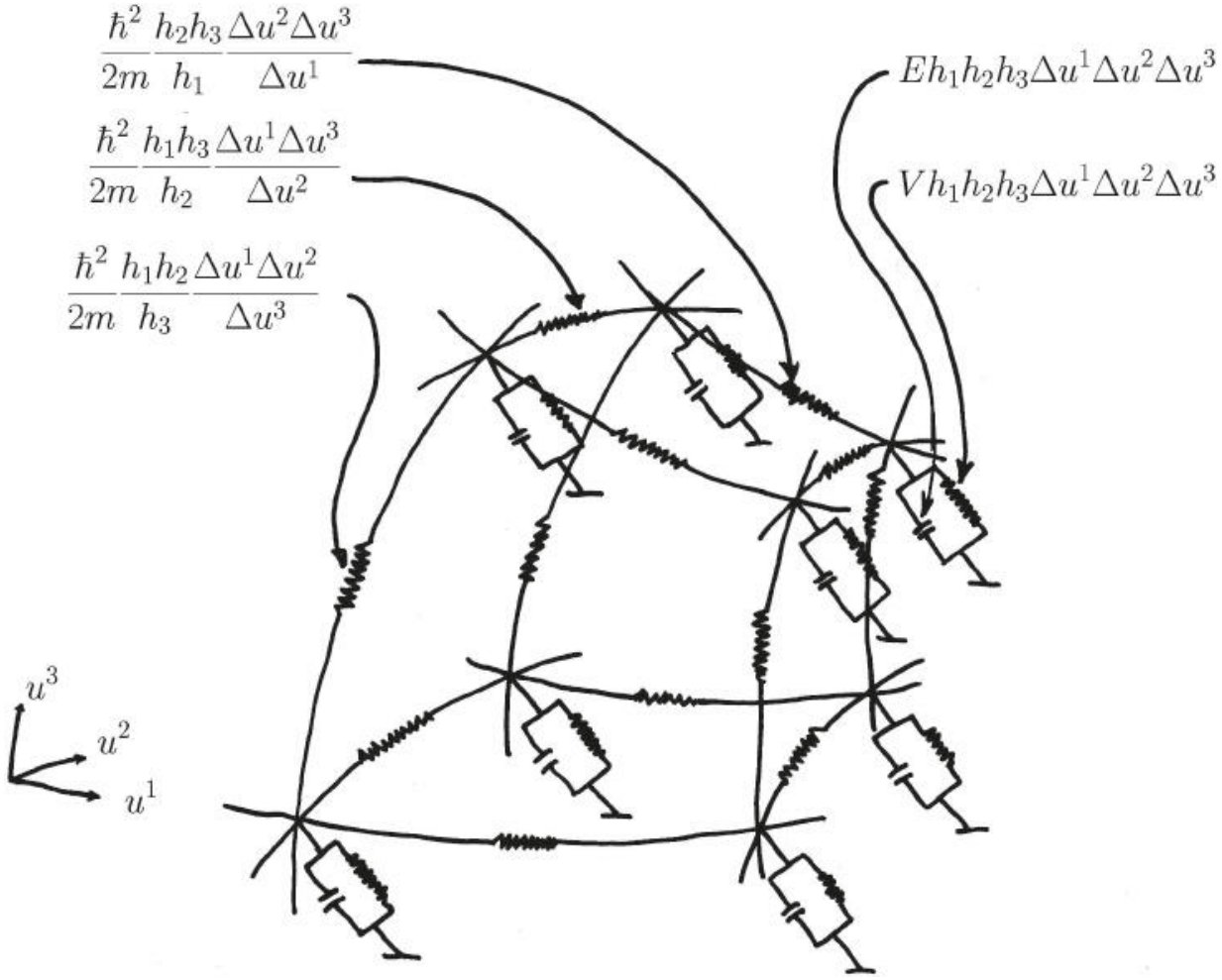


Figure 15.30. Analog model of Schrödinger's equation with three independent variables (see [→ Kron 1945/1, Fig. 7])

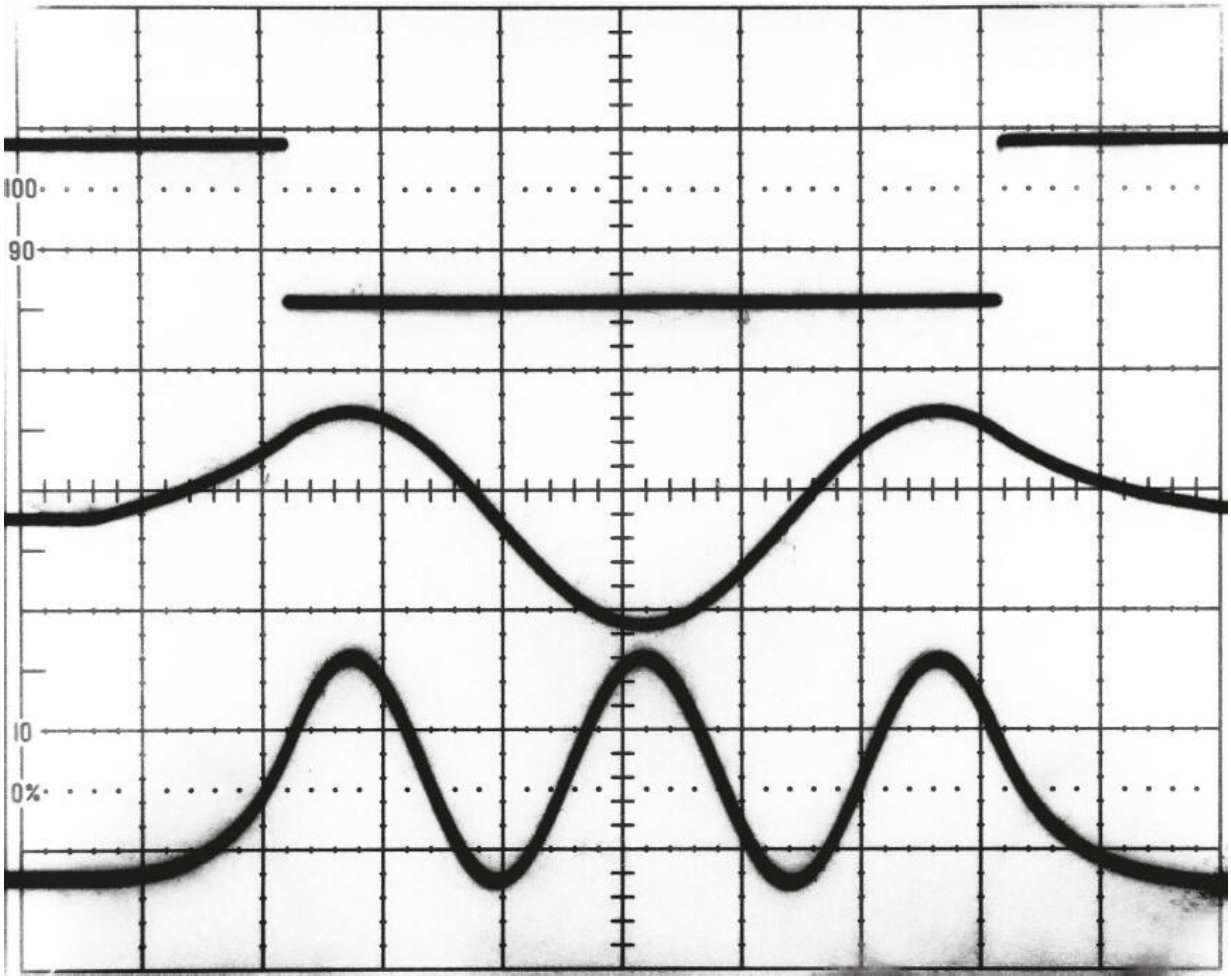


Figure 15.31. Typical solution for the one-dimensional SCHRÖDINGER equation

An interesting example, the simulation of a one-dimensional quantum mechanic transmission and reflection process, is described in [→HUMMER et al. 1976/1] and [→HUMMER et al. 1976/2]. It differs from the majority of PDE solutions on analog computers described so far as it is problem time that is discretized while the spatial coordinate *x* is identified with continuous machine time. This time discretization is implemented using a hybrid computer approach to store and playback a function.

15.4 Mechanics and engineering

Problems emerging from mechanics and engineering, ranging from vibrating coupled mass systems to the simulation of surge chambers and many more, are also well suited for treatment by analog computers.[115]

15.4.1 Vibrations

The treatment of vibrating systems is a typical application for an analog computer as the examples in →sections 8.3 and →8.7 demonstrated. Vibrations are highly undesirable in certain applications, ranging from mere inconveniences to real hazards, while they are a necessity in other areas, such as vibratory conveyors. Accordingly, the commercial impact of such studies was, and still is, high. Depending on the complexity of the underlying mechanical system, these problems can be quite challenging as [→Sankar et al. 1979, p. 11.] note:

> "*The control of vibration in mechanical systems is a serious and challenging design problem.*"

A notorious example for unwanted vibrations is the *Pogo effect* that was experienced in early large rockets, which used liquid fuels.[116] This effect was caused by pressure variations in the feed lines for the rocket engines which caused unsteady combustion. This resulted in varying accelerations of the rocket which in turn tended to amplify these pressure variations and so on. In extreme cases the resulting forces can reach levels capable of destroying the rocket or at least damaging sensitive payloads such as satellites or astronauts. Other examples involve the stability of tall buildings subjected to wind forces, earthquakes, etc. In the case of a vibrating conveyor the isolation of the vibrating machine parts from their mountings is a critical design parameter.

In general, machines can be considered as vibrating multi-mass systems with spring and damper elements coupling the various parts of the overall system. Treating systems like this with analog computers is described in detail in [→Dornier/3], [→Macduff et al. 1958], [→Tse et al. 1964], and [→Trubert 1968, pp. 416 ff.], while [→Jackson 1960] focuses on the analysis of vibrating truss structures. An interesting study of the

vibration of elastic components in a spin stabilized satellite based on an integro-differential equation using a successive approximation approach is described in [→Gorez 1967].

15.4.2 Shock absorbers

[→Sankar et al. 1979] describe an interesting optimization problem: A shock absorber is to be designed, which should impose minimal accelerations on the attached mass while simultaneously limiting the distance the damped mass travels under an excitation impulse. The simulations for determining the parameters of this system were performed on a hybrid computer. The task was considerably complicated by the necessity of implementing an accurate model of dry friction requiring many nonlinear computing elements.[117]

[→Mezencev et al. 1978] describe the development of an adaptive hydropneumatic anti-vibration device for marine propulsion systems using an analog computer.

15.4.3 Earthquake simulation

The analysis of the behavior of buildings excited by Love- and Rayleigh-waves resulting from an earthquake is of great importance, especially in countries with high seismic activity, such as Japan, where pioneering work in this area was done. [→Hitachi 200X, p. 17] describes the simulation of a multi-story building which is modeled as a multi-mass vibrating system where each floor is lumped together as a single mass. The walls act as spring-damper systems connecting adjacent floors.

A simple building with two floors is characterized by the weights of the floors, m_1 and m_2, the moduli of resilience s_1, s_2, and the damping coefficients d_1, d_2. With y_1 and y_2 denoting the horizontal displacements of the respective floors, the building dynamics are described by the following two coupled differential equations

$$\begin{aligned} m_1\ddot{y}_1 + d_1\dot{y}_1 + d_2\left(\dot{y}_1 - \dot{y}_2\right) + s_1 y_1 + s_2\left(y_1 - y_2\right) &= m_1\alpha\left(t\right) \text{ and} \\ m_1\ddot{y}_2 + d_2\left(\dot{y}_2 - \dot{y}_1\right) + s_2\left(y_1 - y_2\right) &= m_2\alpha\left(t\right) \end{aligned}$$

with $\alpha(t)$ representing the ground motion induced by the earthquake. In practical applications real seismic data stored on analog magnetic tapes was used to generate $\alpha(t)$.

Based on these equations, the program shown in →figure 15.32 can be derived.[118] The two function generators on the right-hand side are used to implement two hysteresis curves modeling the nonlinear stiffness of the building structures. The results of a typical simulation run are shown in →figure 15.33.[119]

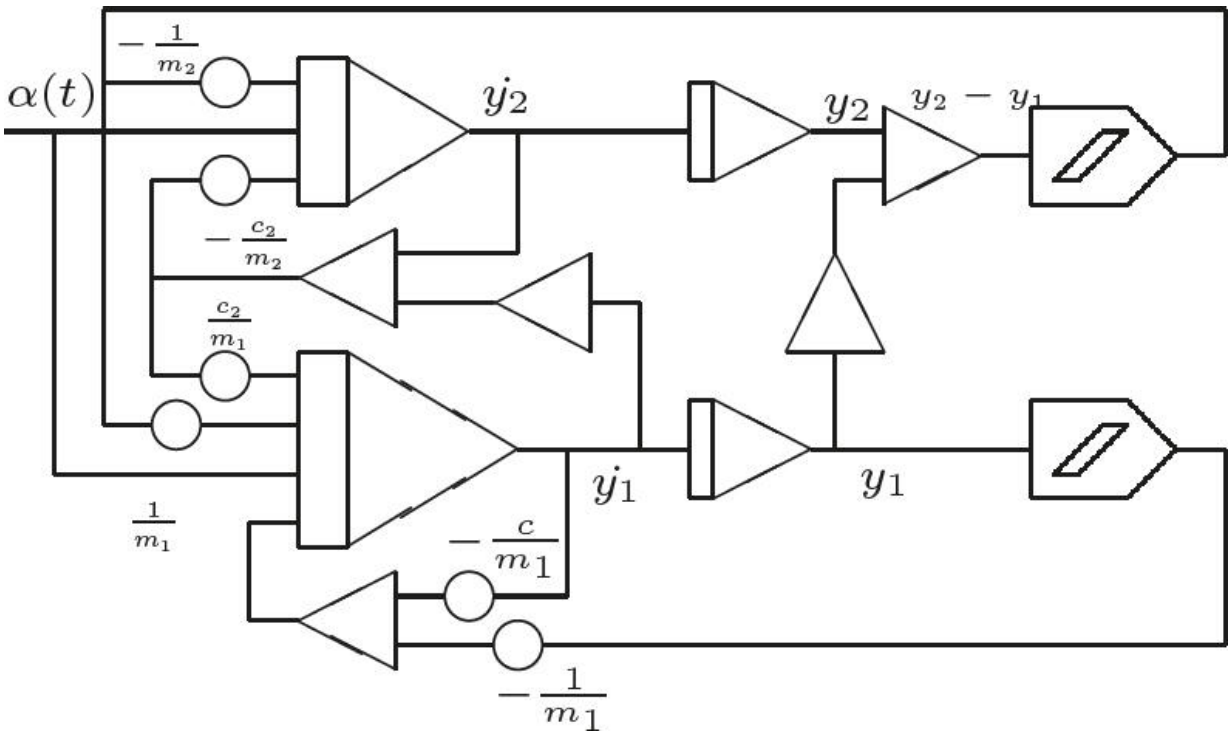


Figure 15.32. Computer setup to simulate a two-story building excited by horizontal ground movements (see [→Hitachi 200X, p. 17])

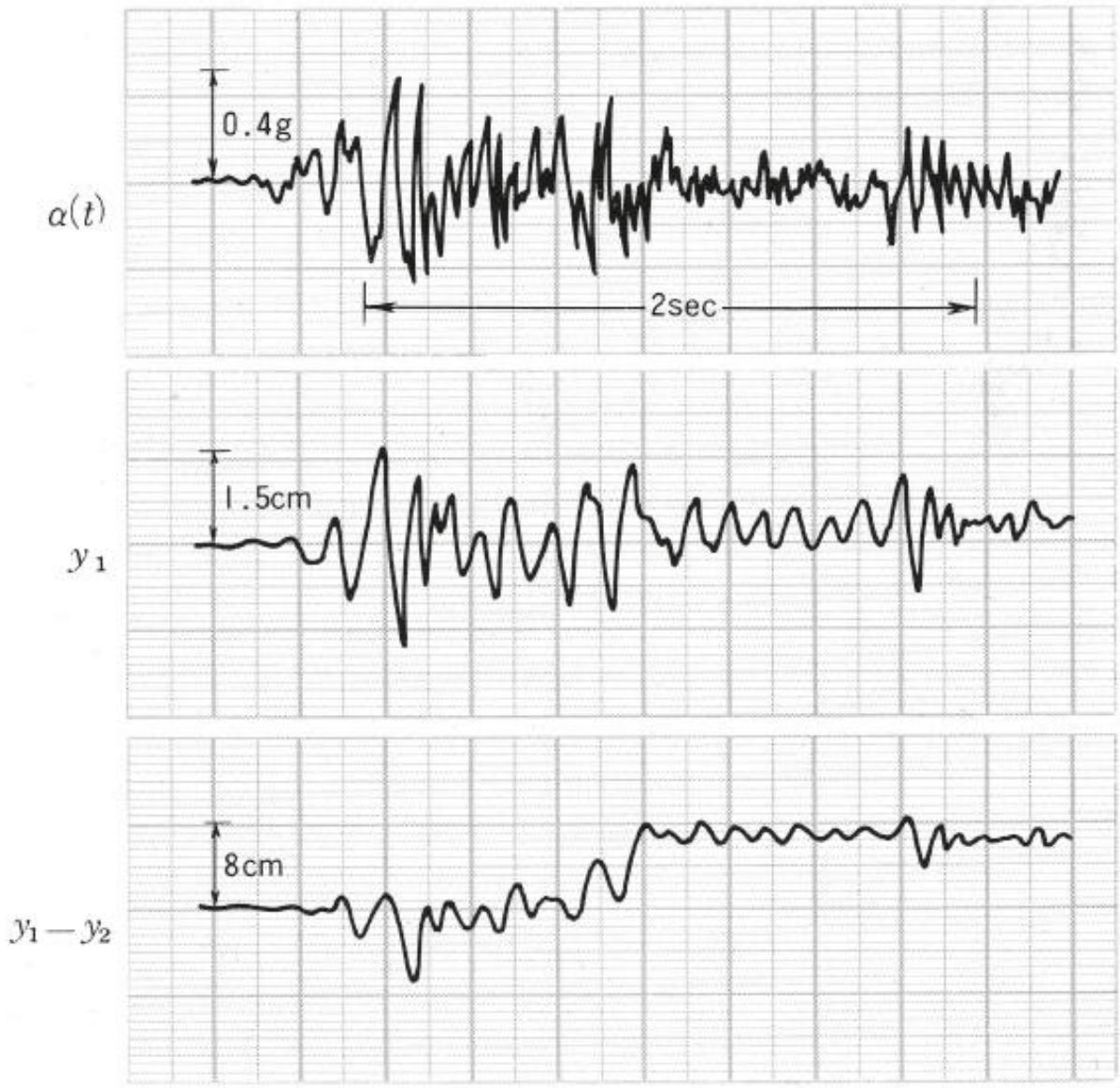


Figure 15.33. Results of an earthquake simulation (see [→Hitachi 200X, p. 17])

A special purpose analog computer, *SERAC*,[120] was developed by Hitachi in the late 1960s.[121] This system featured a photoelectric curve follower, which directly generated the excitation signal $\alpha(t)$ based on actual seismograms.

15.4.4 Bending beams and similar problems

Bending and vibrating beams typically leads to a boundary value problem where values of variables y_i within a system are determined at $t = 0$ by $y_i(0)$ and at some suitable end time T by values $y_i(T)$. A simple approach to this class of problems, the *shooting method*, is based on parameter variations in repeated analog computer runs until the desired results $y_i(T)$ are reached, starting from initial values $y_i(0)$. This works quite well for differential equations up to fourth order but typically breaks down for higher orders as the behavior gets increasingly unpredictable. [→ Yanowitch 1956] describes a different approach based on suitable linear combinations of solutions of DEQs using a REAC analog computer to quickly generate the underlying particular solutions. It also describes the simulation of coupled beam vibrations.

More complex shapes of heavy, uniformly thick strips of material, including heart-shaped loops, are treated in [→ Lippmann et al. 1961] using a Telefunken RA 463/2 analog computer. Two further problems described in this study are an elastic catenary and a thin, heavy, vertically hanging wire, exposed to horizontal winds. General techniques for treating large deflections of cantilevers on analog computers can be found in [→ Khwaja 1968] and [→ Khwaja 1971].

15.4.5 Rotating systems and gears

Rotating systems are of great importance in many areas, ranging from large positioning systems to centrifuges, jet engines, etc. These systems are often critical in the sense that human lives depend on their flawless operation.

Bearings are some of the most critical parts in rotating systems. Long shafts, common in turbines and similar machines, are prone to bending and vibrations, which subject the bearings to additional and sometimes critical loads. Consequently, many studies were performed to simulate the behavior of such systems.[122]

Gas and steam turbines often use hydrodynamic bearings since traditional bearings would fail rapidly in such applications due to disturbances of the oil film. The simulation of such hydrodynamic bearings on analog computers is described in [→ McLean et al. 1977] and [→ Rieger et al. 1974].

Some rotating systems achieve extremely high speeds. A turbomolecular pump with a rotor weight of 1.7 kg, described in [→ Fremerey[a], p. 33], operates at 51,600 revolutions per minute. Ultra centrifuges for the enrichment of uranium run at even higher speeds. Applications like these require magnetic bearings, of which active and passive variants exist. Passive magnetic bearings were developed in the 1960s and rely on permanent magnets, while active magnetic bearings use electromagnets controlled by a servo loop.[123] Analog computers were not only used to determine the basic parameters of bearing systems like these but were also employed to model the servo circuits for active magnetic bearing systems. Often the analog computer was used in *hardware in the loop* simulation setups where the analog computer is an active part of the system to be analyzed. In this case the analog computer was used to control the current energizing the electromagnets of a real magnetic bearing in operation. In such a setup the consequences of parameter variations can be directly explored using real hardware.

[→ Okah-Avae 1978] focuses on the effects on bearings caused by transverse cracks in turbine shafts. The goal of this work was to develop methods allowing an early detection of such damages. The simulation results were compared with actual data gathered from a damaged turbo generator and showed a high degree of correlation.

A complex simulation of the behavior of a hydraulic turbine driving an electric generator including the hydraulic part as well as the effects of the rotating parts under various external conditions such as load changes, etc., can be found in [→ Djordjevic 1967].

15.4.6 Compressors

A complex centrifugal compressor simulation is described in [→ Davis et al. 1974]. Interestingly, this simulation does not just model the macroscopic behavior of the compressor system itself but also the gas dynamics of the medium to be compressed. Another analog centrifugal compressor system simulation can be found in [→ Schultz et al. 1974].

A special purpose analog computer for the simulation of a reciprocating piston gas compressor system is described in patent [→ Hughes et al. 1960]. An interesting point is the treatment of pressure

pulses resulting from the piston operation in the compressor output. The system is intended to serve as the base for parameter studies in acoustic filter design and piping systems. The proposed system is not too dissimilar to the Anacom system although much smaller and tailored towards this special application.

A complex and comprehensive simulation system for a free piston gas generator is described in [→Plow 1959] using a *Boeing* electronic analog computer (*BEAC*). This implementation is incredibly detailed, even taking the process of heat removal from the compressor by means of delivery and intake processes into account.

15.4.7 Crank mechanisms and linkages

Many systems such as reciprocating pumps or piston compressors use a crank mechanism like a sliding crank to transform a rotary motion into a linear movement.[124] In the simplest case, this type of transformation can be achieved with a scotch yoke mechanism as shown in →figure 2.16 in →section 2.6. In most cases, more complex mechanisms are required since a scotch yoke develops high forces on the mechanism's bearings. Additionally, many commercial applications require special curve progressions of the linear movement based on the rotary input of the mechanism.[125]

[→Crossley 1963] and [→Crossley 1965] describe an interesting approach for the simulation of complex path generating linkages using inverse functions on an analog computer. →Figure 15.34 shows typical curves of motion obtained by the simulation of a four joint linkage on a Telefunken RAT 700 tabletop computer. A further example of a linkage simulation can be found in [→Mahrenholtz 1968, pp. 179 ff.].

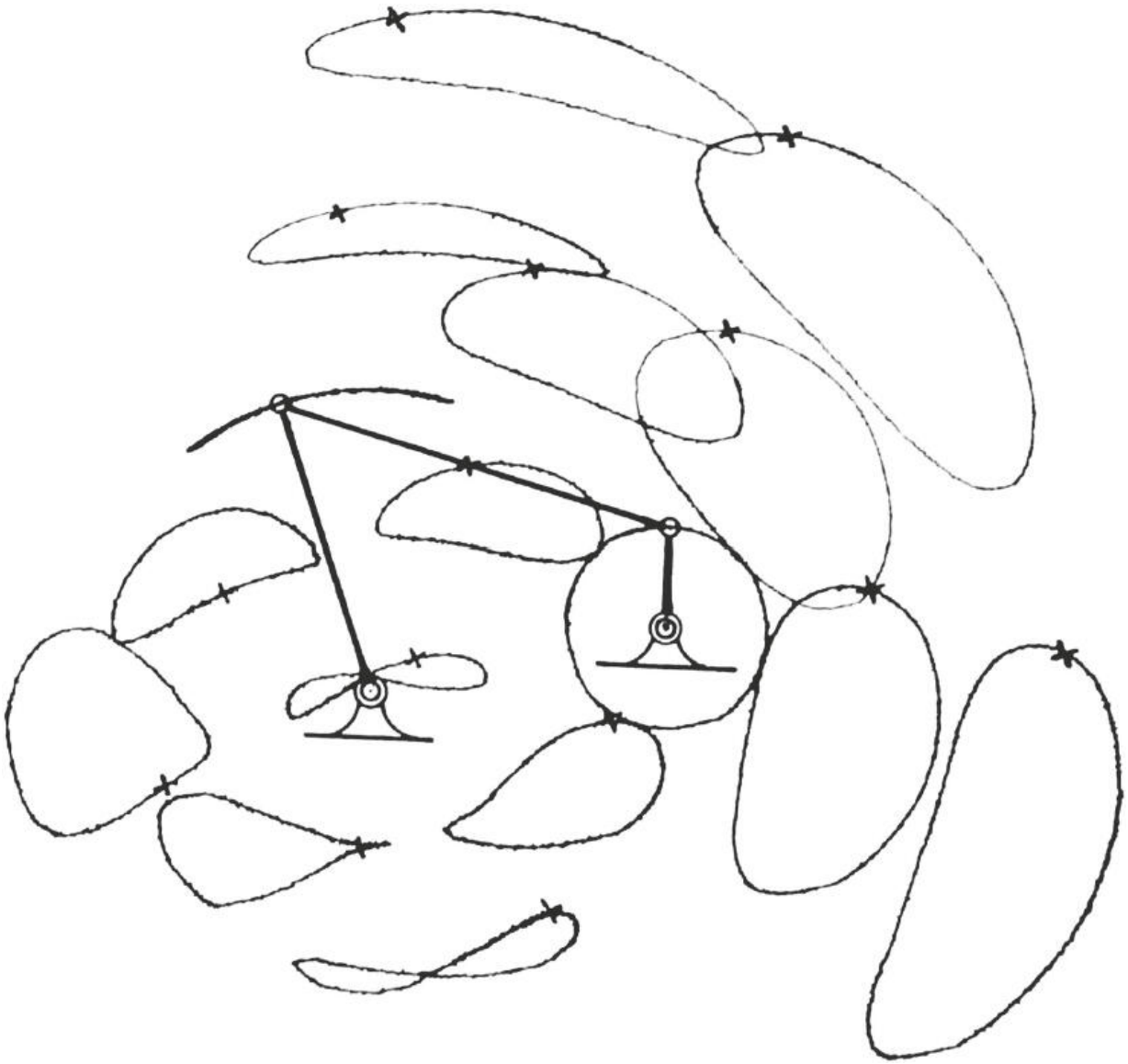


Figure 15.34. Curves of motion obtained through the simulation of a four joint linkage (see [→ Crossley 1963])

15.4.8 Non-destructive testing

Non-destructive testing is frequently cheaper and safer than testing systems to destruction. Typical application areas are listed by [→ Landauer 1975]:

> "*A speeding automobile goes out of control on a test track and crashes violently. A reactor vessel ruptures at a process plant when unstable conditions are reached. Or, a*

650-MW generator loses bearing lubricant and goes into uncontrollable vibration. Each of these occurrences represents destructive testing at its best [...] or worst."

In the majority of cases, hybrid computers were used in this context since often functions of more than one variable have to be generated. The analog computer was often also closely tied to the system being analyzed.[126]

15.4.9 Ductile deformation

In 1956 *General Electric* implemented a comprehensive analog computer based rolling mill simulation for the *Sharon Steel Corporation* to optimize operating and startup procedures for a newly planned rolling mill.[127] As a result of this extensive simulation, a *tune-up time*[128] of only one day, much lower than the seven to ten days normally required, was achieved.

Steel rolling processes can be described by the *Karman differential equation*. Its implementation and treatment on analog computers is described in [→ Hitachi 1968]. Another practical example can be found in [→ Golten et al. 1967] where a hybrid computer is used. The mathematical principles for the simulation of cold rolling are described in [→ Heideprim 1976]. A thorough treatment of the simulation of drag rolling processes can be found in [→ Mahrenholtz 1968, pp. 122 ff.].

[→ Darlington 1965] describes a complex analog simulation for a multistand cold rolling mill. An interesting aspect of this work is that the analog simulation is based on coefficients obtained by a digital computation.

The ideal velocity for roller drives can also be determined using analog computers as [→ Rohde 1977] and [→ Rohde et al. 1981] show. The hybrid computer system used in these simulations is described in detail in [→ Schloemann-Siemag 1978]. [→ Mahrenholtz 1968, pp. 117 ff.] focuses on determining optimal parameters for convex dies, while [→ Ovsyanko 1977] describes the analysis of plastic deformations on an analog computer. [→ Ramirez 1976, pp. 22 f.] covers the simulation of injection molding processes. The analysis of the deformation of steel plates subjected to the forces of an explosion by means of an analog computer is described in [→ James et al. 1971, pp. 175 ff.]. An interesting

investigation of the forces exerted by shearing processes is given in [→Dornier/6].[129]

15.4.10 Pneumatic and hydraulic systems

Hydraulic elements are of prime importance in many application areas, and consequently much research on these devices was done on analog and hybrid computers. [→Cohen 1971] describes the development of a pneumatic relay based on simulation runs on a hybrid computer, while more complex hydraulic switching elements are analyzed in [→Hannigan[a]]. General information about the treatment of hydraulic systems on analog and hybrid computers can be found in [→Sankar et al. 1980]. [→Nolan 1955] and [→Ameling 1962/2] focus on the simulation of liquid flows (with an emphasis on complex piping networks).

Large-scale hydraulic problems, such as flood simulations or the behavior of sewage systems, have been treated with indirect as well as with direct analog computers. Examples for such studies can be found in [→Hitachi 1969, pp. 1 ff.] and [→Paynter ed. 1955, pp. 239 ff.].

An interesting application is the simulation of *surge chambers*, reservoirs placed at the end of pipes to absorb shock waves generated by the closing of valves, etc. Simulating the generation, distribution and elimination of these shock waves is of prime importance as these can destroy even large and sturdy machines and piping systems.[130]

A nice example is given in [→Meissl 1960/1] and [→Meissl 1960/2] where the focus is on the water level in a *differential surge chamber* as described in [→Johnson 1915] and shown in →figure 15.35. Such a differential surge chamber can be described by the following system of differential equations[131]

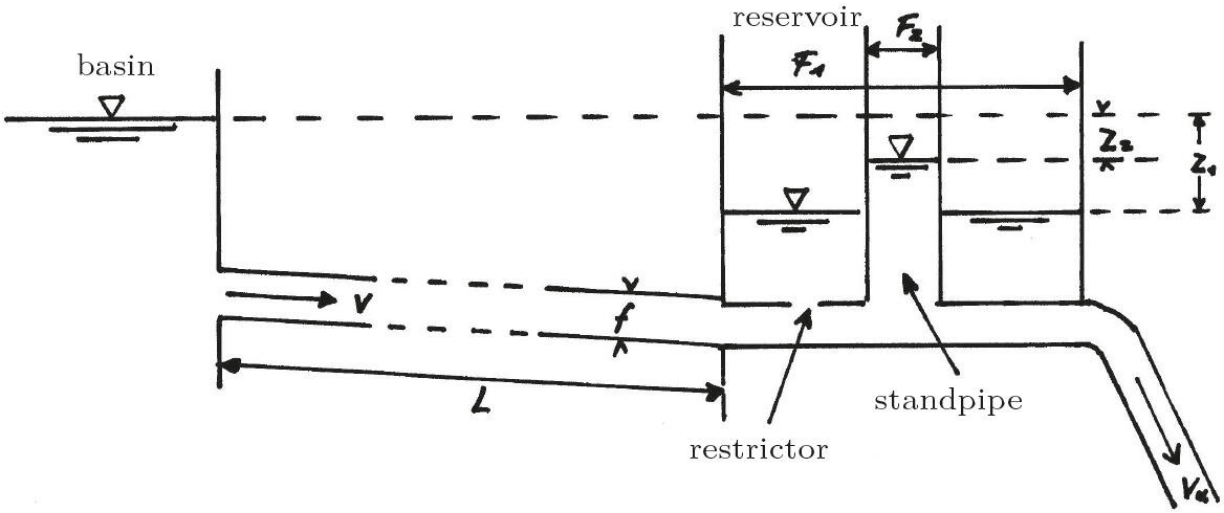


Figure 15.35. Surge chamber (see [→ Meissl 1960/2, p. 76])

$$\begin{aligned} z_2 - z_1 &= \varepsilon_1 \left(\frac{F_1}{t} \dot{z}_1 \right)^2 \operatorname{sgn}(\dot{z}_1), \\ F_1 \dot{z}_1 + F_2 \dot{z}_2 &= f(v_a - v), \text{ and} \\ \dot{v} &= \frac{g}{L}(z_2 - \varepsilon v |v|) \end{aligned}$$

with z_1 and z_2 denoting the respective water levels. The corresponding computer setup is shown in →figure 15.36. Typical simulation results for z_1 and z_2 of a 100 second run are shown in →figures 15.37 and →15.38.[132]

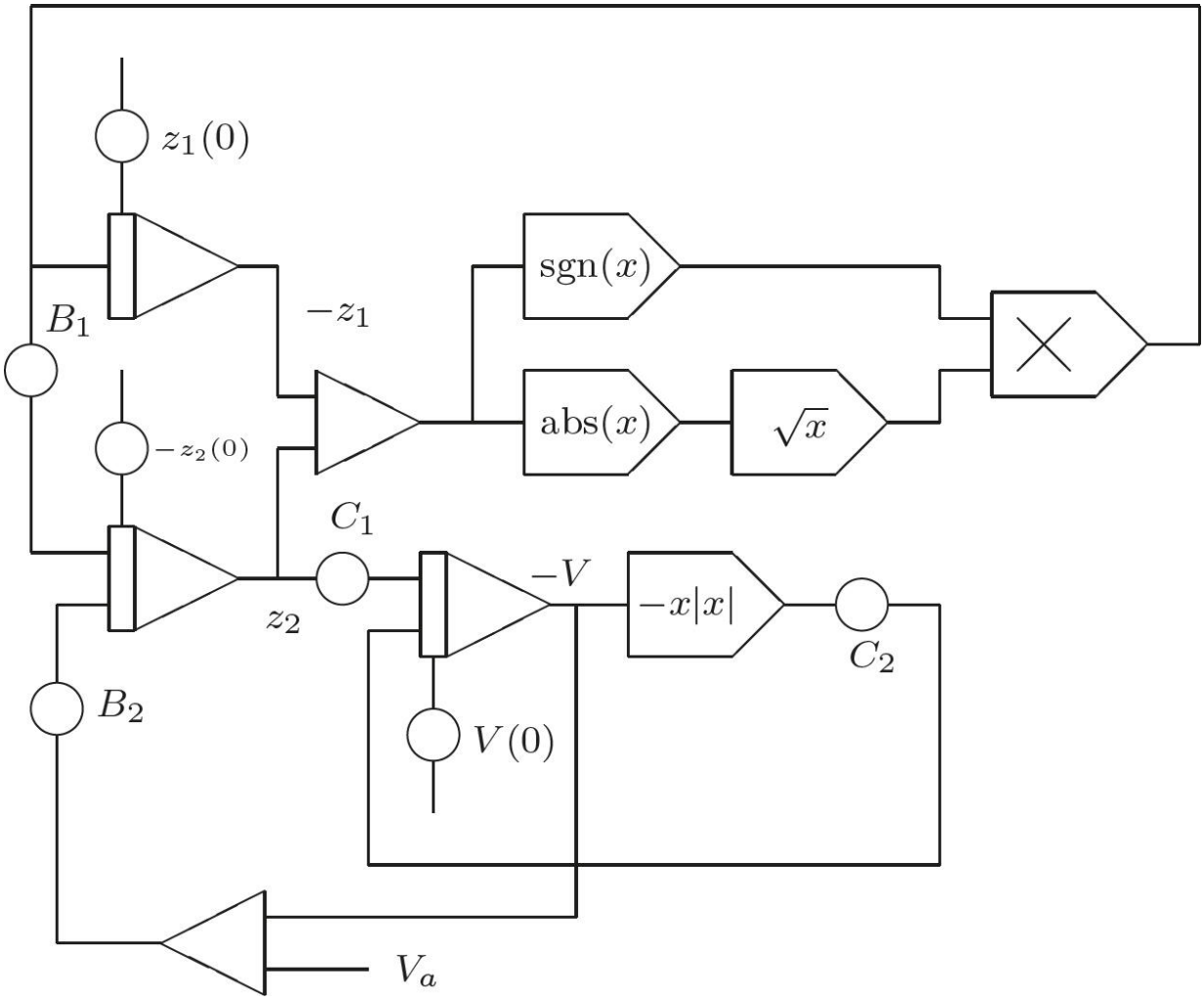


Figure 15.36. Program for the surge chamber simulation (see [→ MEISSL 1960/2, p. 76])

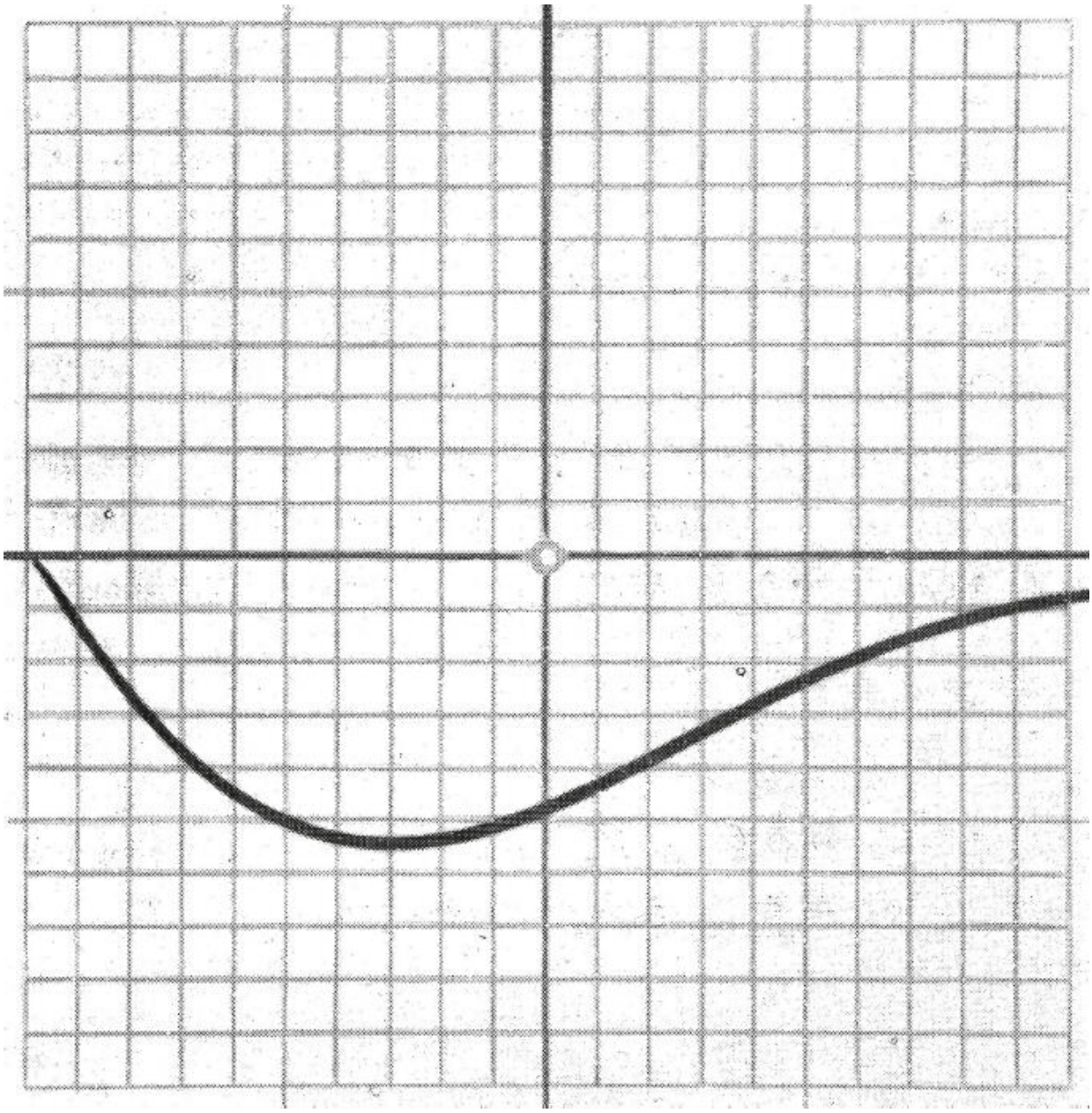


Figure 15.37. Water-level z_1 for $0 \leq t \leq 100$ s (see [→ MEISSL 1960/2, p. 77])

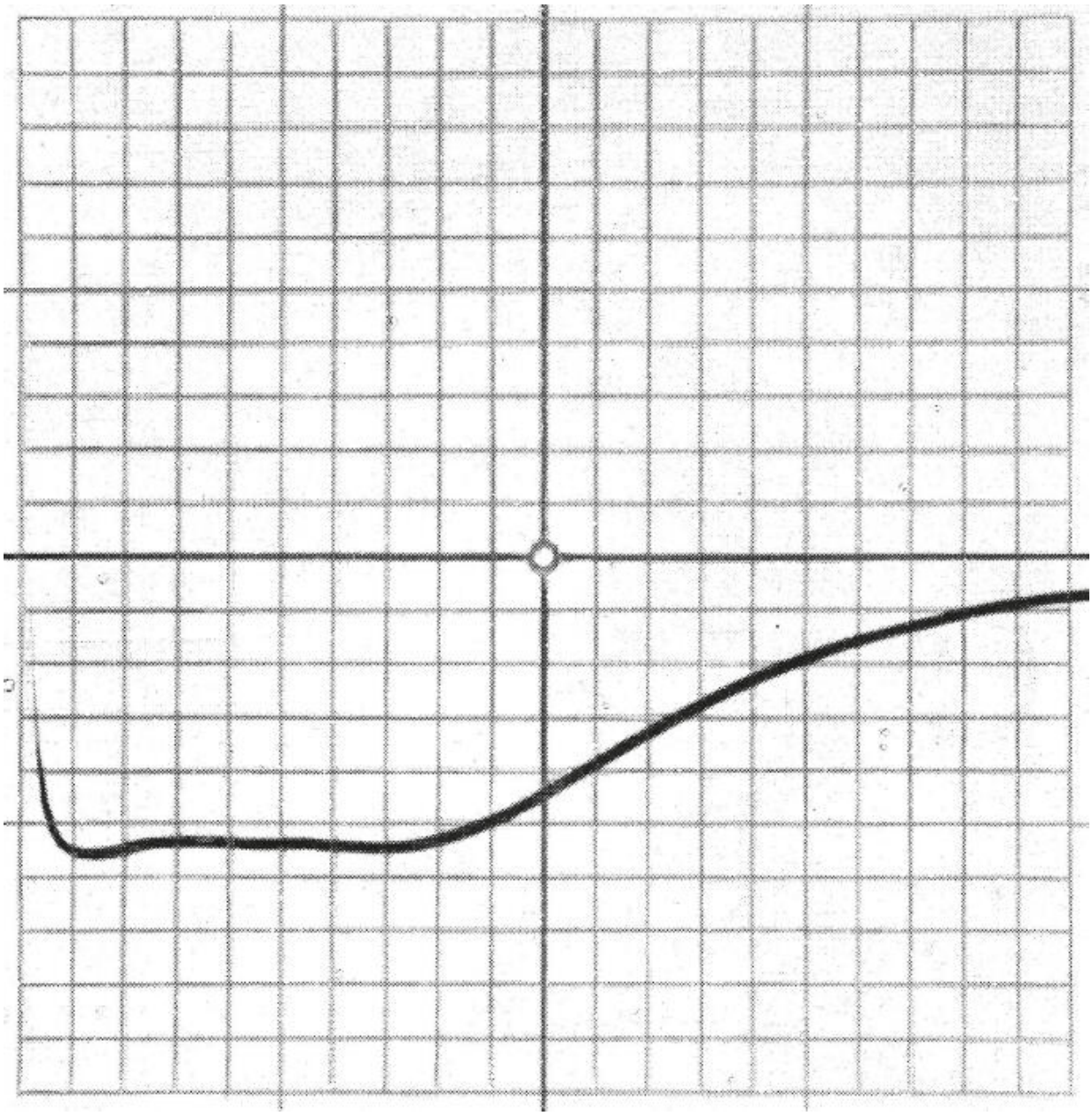


Figure 15.38. Water-level z_2 for $0 \le t \le 100$ s (see [→ Meissl 1960/2, p. 77])

The simulation of the behavior of the *Appalachian surge tank* operated by the *Tennessee Valley Authority* on an analog computer is described in [→ Jackson 1960, pp. 403 ff.]. General information about the use of analog computers for the analysis of water hammer problems can be found in [→ Paynter et al. 1955], [→ Paynter 1955/1], and [→ James et al. 1971, pp. 184 ff.]. The design of surge tanks for hydro stations is covered in [→ Ando

1971]. [→Sorondo et al.] describe the simulation of the piping system in a thermal power plant. The focus here is on the effects of the short run-up times of only 3.5 seconds the pumps are capable of. This placed extreme stresses on the piping network which had a total length of about four miles. Another comprehensive study of surge chambers used in water turbine systems can be found in [→Nicole 1967]. This study was performed on an EAI 231R analog computer and also used a digital simulator APACHE for the computation of steady state solutions.

The problem of sloshing in a steel converter is treated in [→Barth 1976, pp. 600 f.]. In this simulation, not only the sloshing itself is simulated but also its effect on the tilting control system for the converter.[133] In addition to a direct analog model consisting of a scaled down version of the converter, an indirect analog model was used for the simulations.

A very interesting one-of-a-kind analog computer was the Dutch *Deltar*, short for *Delta Getij Analogon Rekenmachine*[134] used from 1960 until 1984 at the *Deltawerken*[135] for the planning of dams, storm surge barriers, etc., which is shown in →figure 15.39. It was based on the seminal works of Johan van Veen,[136] see [→Veen 1947/1], [→Veen 1947/2], and [→Veen 1947/3].[137]



Figure 15.39. The Deltar analog computer (picture by Erik Verhagen)

15.4.11 Control of machine tools

Analog computers were also used for the development and implementation of machine tools. Two interesting analog interpolation systems for the control of turbine blade milling machines were developed at the *Lewis Flight Propulsion Laboratory* and are described in [→Johnson 1962]. One of the systems implements a cubic interpolator using three mechanical integrators to generate a polynomial of third degree thus interpolating a function specified by values given at equidistant sampling points. The other system was based on a direct analogy using a flexible steel ruler to implement a spline interpolation. This ruler was bent into position by a punch card controlled servo system controlling a number of linear actuators. The bent ruler was then used as

a model to control the milling process using a magnetic pickup in a servo loop.

15.4.12 Servo systems

Servo systems are central to nearly every technical system. Consequently the simulation of such systems using analog computers was of high importance and numerous studies deal with related topics. Servo systems are divided into *linear* and *nonlinear* systems. Linear servo systems use actuators operating in a continuous way while nonlinear servo systems often use actuators supporting only a limited number of states such as valve actuators for reaction control engines, etc. The simulation of such systems is described in [→McLeod 1962, pp. 92 ff.] and [→Korn et al. 1956], while [→McLeod et al. 1958/2, pp. 297 f.] treat instabilities of servo systems.

Another early, yet comprehensive study of the transient response of servomechanisms can be found in [→Tasini 1957]. In contrast to earlier works typically focused on the overall stability of servo systems this study develops criteria for optimum servo system parameters with respect to their transient response. The analog computer used in this study was purpose built at the electrical department of *The Rice Institute*, Houston, Texas. It was built from standard components, mainly GAP/R *K2-X*. Ten of these were used in conjunction with *K2-P* chopper amplifiers to implement ten chopper-stabilized integrators. In addition to this, ten summers and ten committed amplifiers for special function generators were used. Interestingly, four different multiplication techniques were examined with respect to their respective suitability for the servo simulation system, the most interesting of these schemes being a coincidence multiplier.[138]

[→Klittich 1974] describes the design and simulation of a positioning system for a large radio telescope antenna. This study was done during the design and development of the Effelsberg radio telescope in Germany. Positioning systems like this are quite complicated as they usually employ multiple motor drive systems for moving the large and heavy (the parabolic dish of this particular installation has a diameter of 100 m with an overall system weight of about 3 200 tons) rotating structure. In systems this large, the elastic properties of the drive train

become a problem and can cause underdamped parasitic oscillations. This study not only describes the simulation itself but also compares simulation results with actual measurements taken after completion of the installation in 1971.

A similar study is described in [→THUN[a]] where the attitude control system of another large radio telescope dish antenna is modeled. It covers effects such as the inertia of the antenna itself, wind load, achievable accuracy, as well as the generation of nonlinear antenna movements required for certain tracking tasks.

[→BEKEY et al. 1960] describe an interesting technique for generating BODE[139] and NYQUIST diagrams[140] by means of an analog computer.

15.5 Color matching

A very interesting and arcane application of analog computers was the task of *color matching*. The problem is to find the proportions of a number of pigments so that the resulting color matches a given color sample. This is not a simple task as in both cases of additive as well as subtractive mixing an under-determined system of equations must be solved. Color matching finds application in a variety of applications including blood tests, making false teeth, etc.

This problem can be solved by applying the *KUBELKA-MUNK theory*, developed by PAUL KUBELKA[141] and FRANZ MUNK[142] in 1931.[143] The first simple analog device for solving the resulting equations, which were previously solved by graphical techniques, was developed in 1955.[144] This approach is based on the absorption and scattering of light by particles and the substrate according to

$$\frac{K}{S} = \frac{(1-R)^2}{2R}$$

with *K* representing the absorption, *S* the scattering, and *R* being the reflectance. Applying this to pastel paints or dyed fabric, scattering due to the underlying white pigment, S_w, has to be taken into account yielding a system of linear equations with unknowns c_i:[145]

$$\left(\frac{K}{S}\right)_{\text{mixture}} = c_1 \frac{K_1}{S_2} + c_2 \frac{K_2}{S_w} + c_3 \frac{K_3}{S_w} + \frac{K_w}{S_w}$$

Based on this, Hugh R. Davidson[146] and Henry Hemmendinger[147] developed *COMIC* (*Colorant Mixture Computer*), a sophisticated specialized analog computer for color matching in 1958. This system, shown in →figure 15.40, was used extensively in various industries well into the 1970s with about 200 machines sold before it was eventually replaced by digital systems.[148] It reduced the time for a typical color matching task from hours to a maximum of 5 to 20 minutes. COMIC could solve 16 simultaneous equations with three to five unknowns.

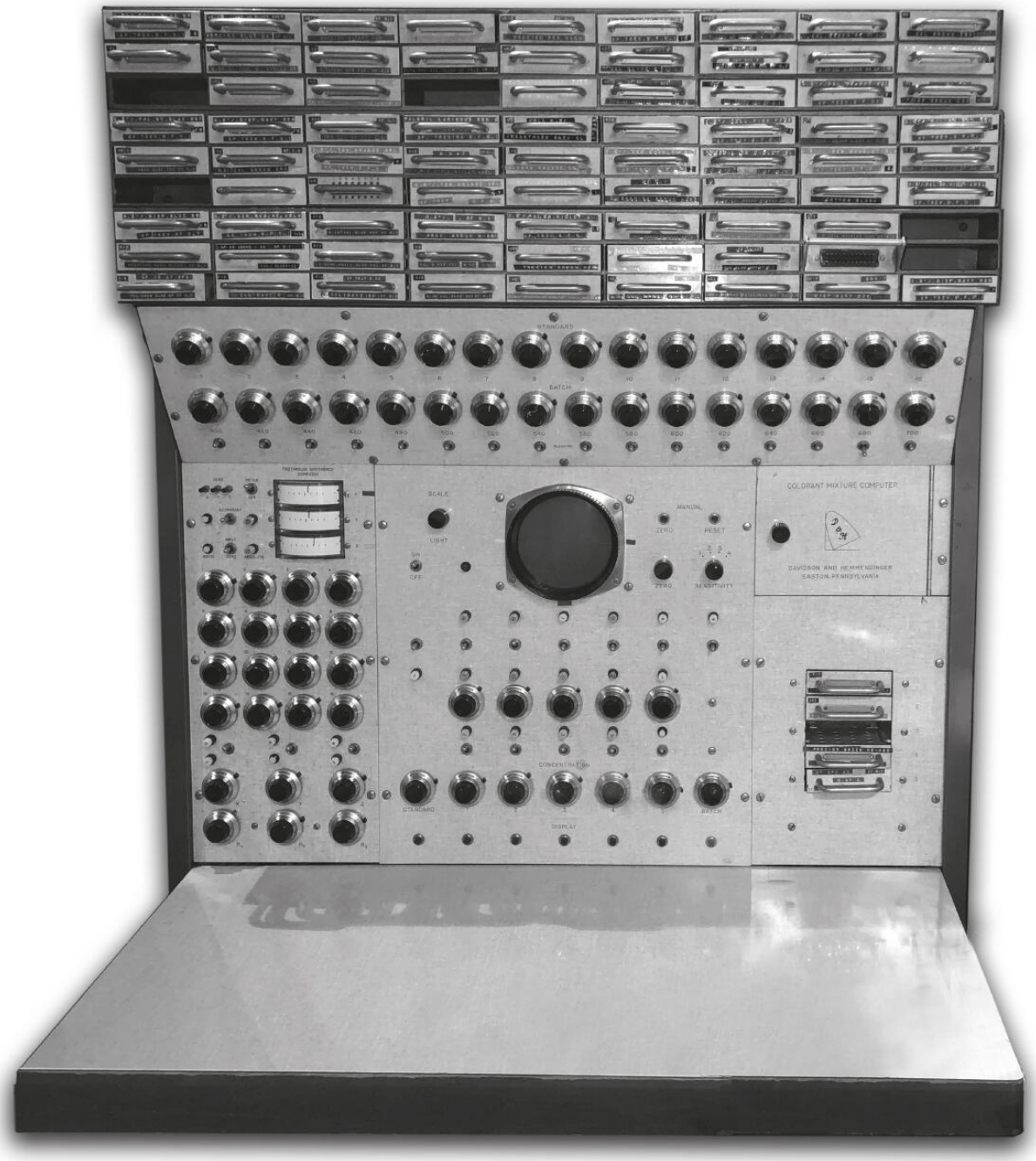


Figure 15.40. The Colorant Mixture Computer (reproduced with kind permission of the Society of Dyers and Colourist Textile Collection)

15.6 Nuclear technology

Although many problems related to nuclear reactors and the like are not well suited for an analog computer due to vastly different time-scales, analog and hybrid computers nevertheless played a vital role in the early decades of nuclear technology. These systems were used for basic research as well as training aids, etc. A lot of general information about the application of analog computers in nuclear technology can be found in [→ Morrison 1962], [→ Frisch 1971], and [→ Dagbjartsson et al. 1976].[149]

15.6.1 Research

Analog and hybrid computers were irreplaceable tools in fundamental nuclear technology research – not least because safety constraints often forbid performing experiments on actual reactors. Typical application examples are

- the analysis of neutron generation and distribution in reactor cores,[150]
- the consequences of changes in reactor geometry due to temperature variations and the generation of delayed neutrons,[151]
- heat-transfer problems in pressurized-water reactors,[152]
- xenon-poisoning,[153]
- control system development,[154] etc.

Most of these applications make heavy use of an analog computer's capability to scale machine time in order to speed up or slow down simulation runs, thus allowing the study of phenomena occurring too quickly in real time to be actually observable by changing k_0 of the integrators accordingly.

→ Figure 15.41 shows a program used to analyze the stability of a research reactor in Hanford (USA). Of special interest was the behavior of the reactor in the case of a rapid depressurization of the primary coolant loop occurring in conjunction with an increase in reactivity. The resulting power gain in output power spans three decades in a one second time frame, which is about the maximum that can be handled even on a precision analog computer without either having the variables exceeding the range of ±1 machine units or introducing exceedingly large errors.[155] Simulations like this, covering longer time intervals, often

have to be split into smaller steps, typically requiring rescaling between successive steps.

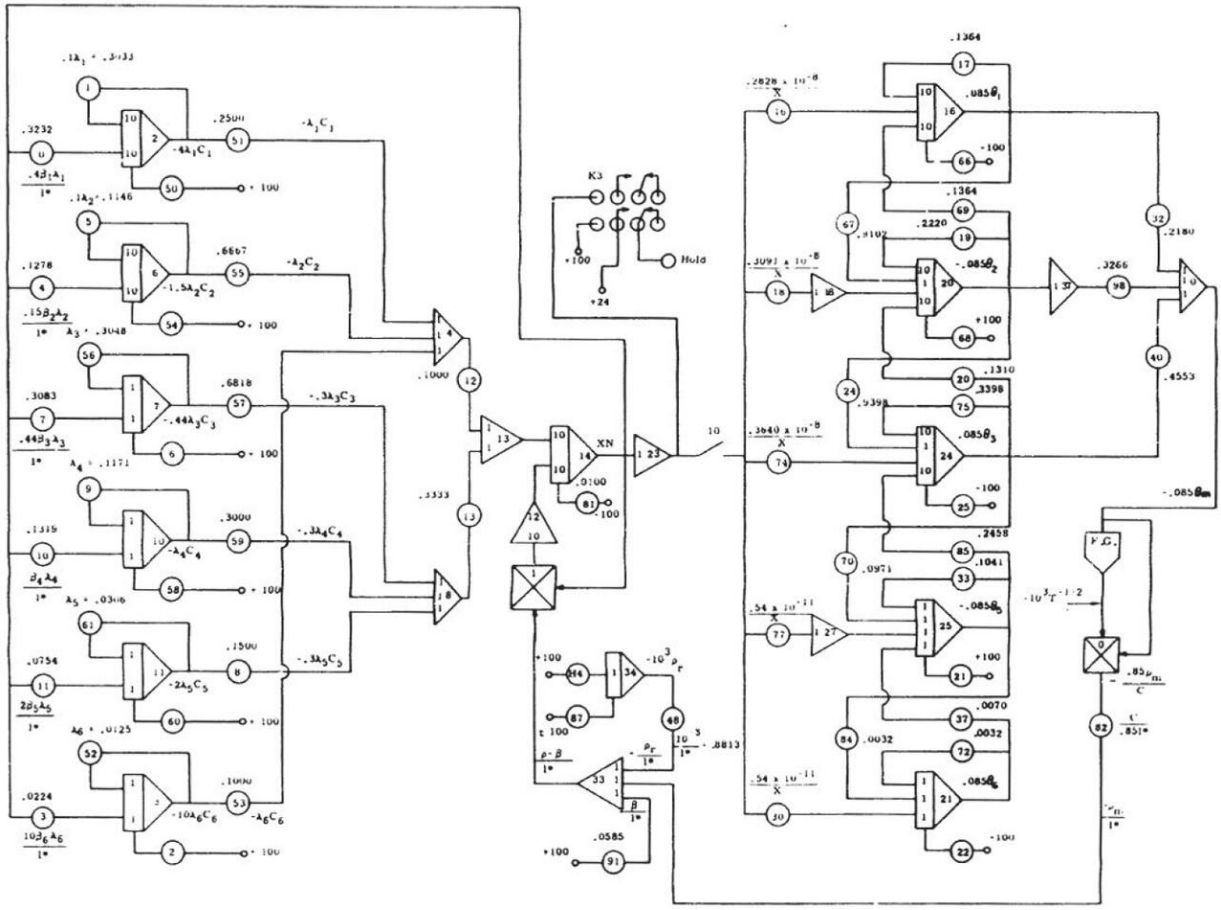


Figure 15.41. Nuclear reactor simulation (see [→JONES 1961, p. 17])

[→GALLAGHER et al. 1957] describe basic techniques for nuclear reactor studies, while [→LOTZ et al.] treat the simulation of a digital control system for a fast nuclear reactor on a Telefunken hybrid computer based on an RA 770 precision analog computer. [→JUST et al. 1962] describe a comprehensive simulation of a multiple-region reactor, while the simulation of steam pressurizing tanks by analog computers is treated in [→BOSLEY et al. 1956].

Another interesting simulation is described in [→HANSEN et al. 1959], where the dynamic behavior of a sodium cooled reactor is studied. The

simulation is very detailed, even taking hydraulic effects in the piping system, etc., into account.

[→ Petrarca 1967] describes an interesting approach to the simulation of coolant densities in a pressure-tube nuclear reactor taking void reactivity feedback into account. Since the void reactivity takes place on a much shorter time scale than changes in coolant density, an iterative approach is implemented. One circuit with a small k_0 runs continuously and computes the coolant density, while a second circuit runs 100 times faster in an iterative fashion. Coupling these two circuits requires some form of storage element which is also implemented by integrators. In addition to this the setup requires functions of the form $f(x, y) = x^y$ which are implemented in a rather clever way using servo multipliers.

15.6.2 Reactor and neutron kinetics

Neutron kinetics is a central aspect of reactor design and simulation. The chain reaction in a typical nuclear reactor is based on two groups of neutrons, the overwhelming majority of *prompt* neutrons, which are released basically in the instant when fission takes place, and a tiny fraction of *delayed* neutrons, released with some delay by the decay of unstable fission products.[156] This can be described by[157]

$$\dot{n} = \frac{n}{l^*}(\delta K - \beta) + \sum_n \lambda_i c_i + s \text{ and } \dot{c}_i = \frac{n\beta_i}{l^*} - \lambda_i c_i$$

with n and s representing the neutron *density* and an external neutron source.[158] l^* is the effective neutron *lifetime*, the c_i represent the precursors of the i-th group of delayed neutrons, and β_i is the *decay constant* of this i-th group. Most practical simulations use six of these delayed neutron groups.

Although these equations can be implemented in a straightforward fashion, this requires quite a lot of computing elements. Consequently, specialized computing elements, using a number of RC-combinations in the feedback path of an operational amplifier were used to model the neutron kinetics of certain nuclear fuels such as ^{235}U or ^{239}Pu. Practical examples of this approach are described in [→ Tyror et al. 1970] and [→ Ulmann 2021/2].

[→Bryant et al. 1962, pp. 18–26] describe a reactor kinetics simulation of a reactivity transient caused by the sudden removal of a control rod, which is then quenched by the intrinsic negative temperature coefficient of the reactor under consideration. A complex nuclear reactor reactivity simulation taking the spatial distribution of fuel elements and sources into account, is described in [→Crommelynckx et al. 1967]. From a technical point of view, the approach described here is interesting as it combines analog simulations of the control system and other reactor subsystems with a spatial system simulation using finite differences, implemented as arrays of resistors, including negative resistances.[159]

→Figure 15.42 shows a large analog computer system used at the Whetstone Laboratories of the English Electric, Babcock & Wilcox, Taylor Woodrow Atomic Power Group in England. It is based on a transistorized Saturn system.[160]

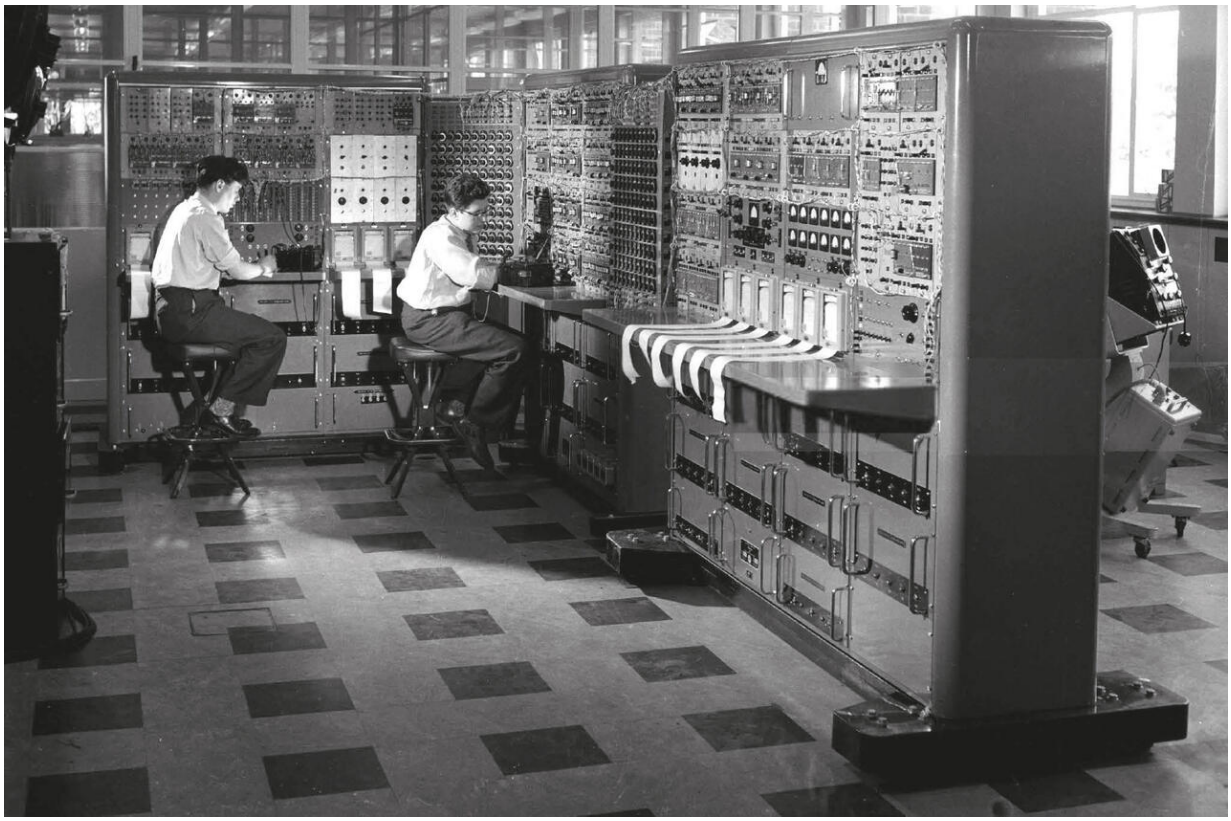


Figure 15.42. Saturn transistorized analog computer used to verify operation and safety of nuclear reactors and their associated control systems ([→Computers and Automation 1960, p. 16])

[→Giordano et al. 1967] give a comprehensive overview of the application of analog computers to the study of space-time dynamics of nuclear reactors. With increasing size of reactor cores the spatial and temporal variations of neutron density as well as reactor poisons such as ^{135}Xe within the core become more and more complicated.[161] Accordingly, a fine-grained model of these dynamics is required, involving a number of volume integrals which are difficult to implement on an analog computer. Using Green's theorem, these are transformed into simpler surface integrals, yet the resulting equations still contain dozens of terms and are quite demanding to implement.

15.6.3 Training

Training reactor operators is of prime importance and requires simulators to be as realistic as possible.[162] An early such simulator was built to train the operators of the nuclear reactor aboard the *N. S. Savannah*, the first nuclear powered merchant ship, which was launched on 27.07.1959. This simulator is described as follows in [→MTW 1964/2]:

> "*Through the use of an operator's control console identical to the one aboard the SAVANNAH, the two PACE 231R*[163] *General-Purpose Analog Computers [allow...], trainees [to] acquire operating experience just as if they were actually on board the ship.*
>
> *The analog computers, which are programmed to represent the complete reactor kinetics, as well as primary and secondary heat balances, activate all the recording instruments and dials on the control panel and respond to the student's manipulation of the operating controls just as the real reactor would.*"

In addition to the operator consoles, this simulator featured an additional console where instructors could introduce faults to be handled by the operators in training. A much simpler reactor simulator setup is shown in →figure 15.43. This particular system was based on a minor variant of the Telefunken RA 463/2 analog computer.[164]

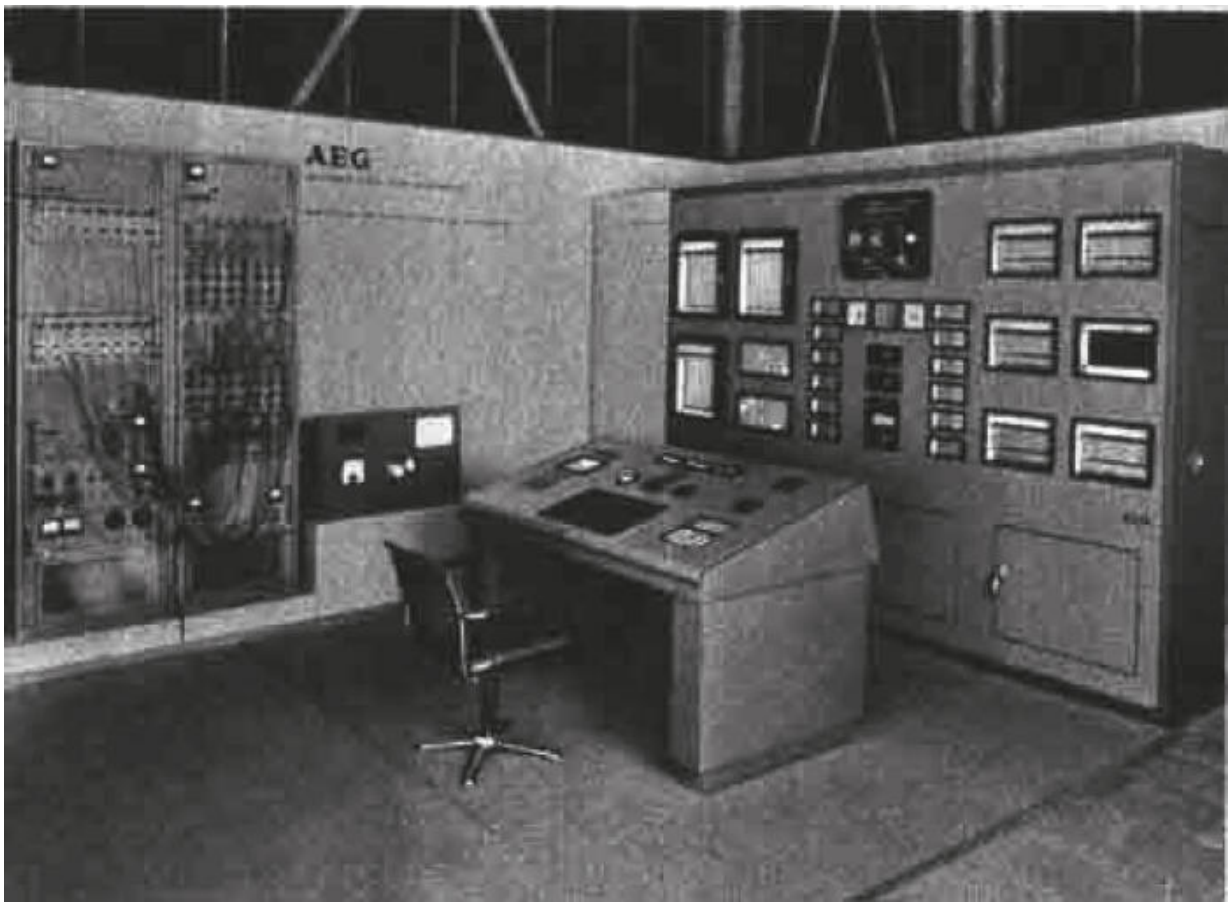


Figure 15.43. AEG reactor simulator based on a Telefunken RA 463/2 (see [→ GERWIN 1958])

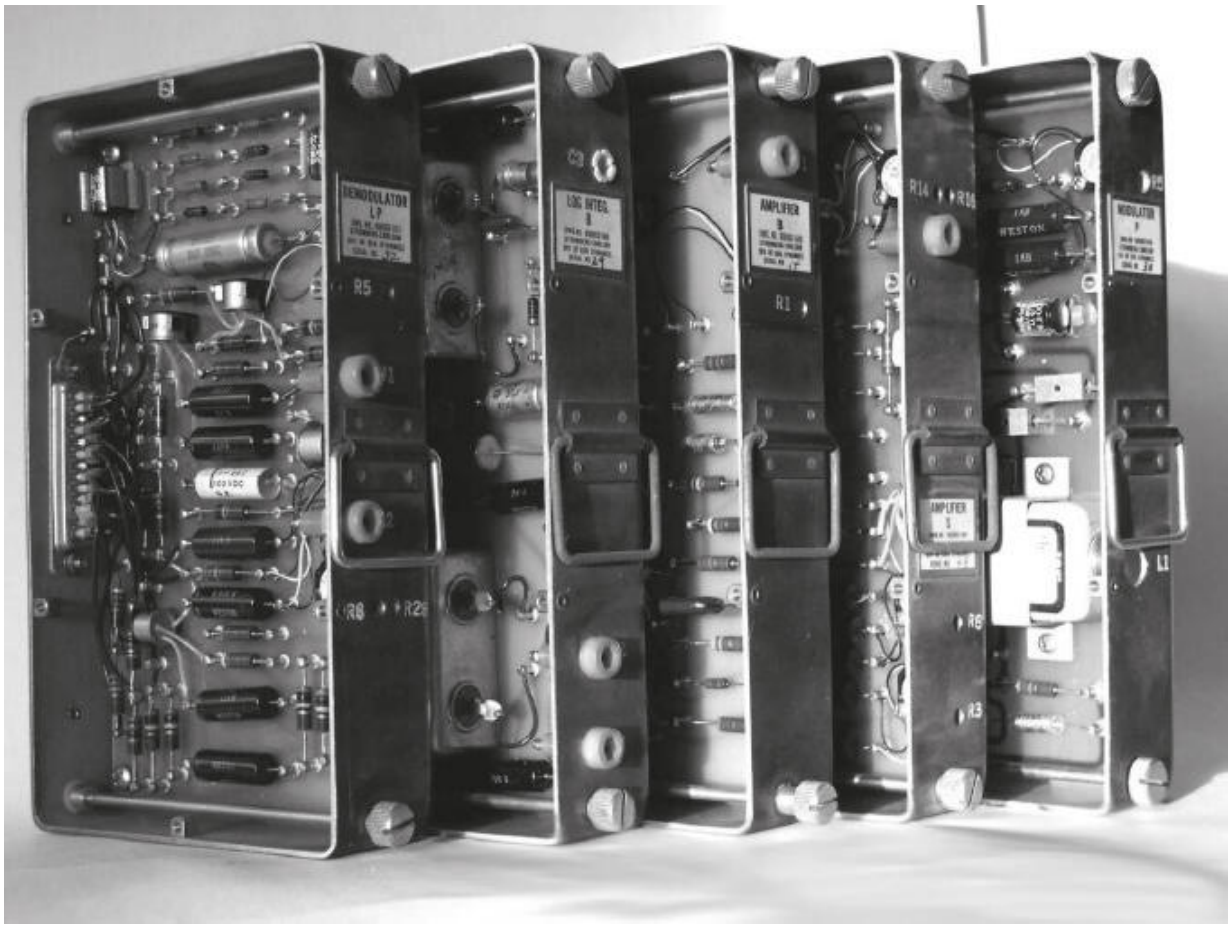


Figure 15.44. Typical reactor control analog computer modules (General Dynamics)

→Figure 15.45 shows the complex *ML-1* reactor simulator developed for the *Army Gas-Cooled Reactor Systems Problem* in the early 1960s, which is described in detail in [→Calamore et al. 1963]. The analog computer at its heart is an *Aerojet* system with several large consoles.

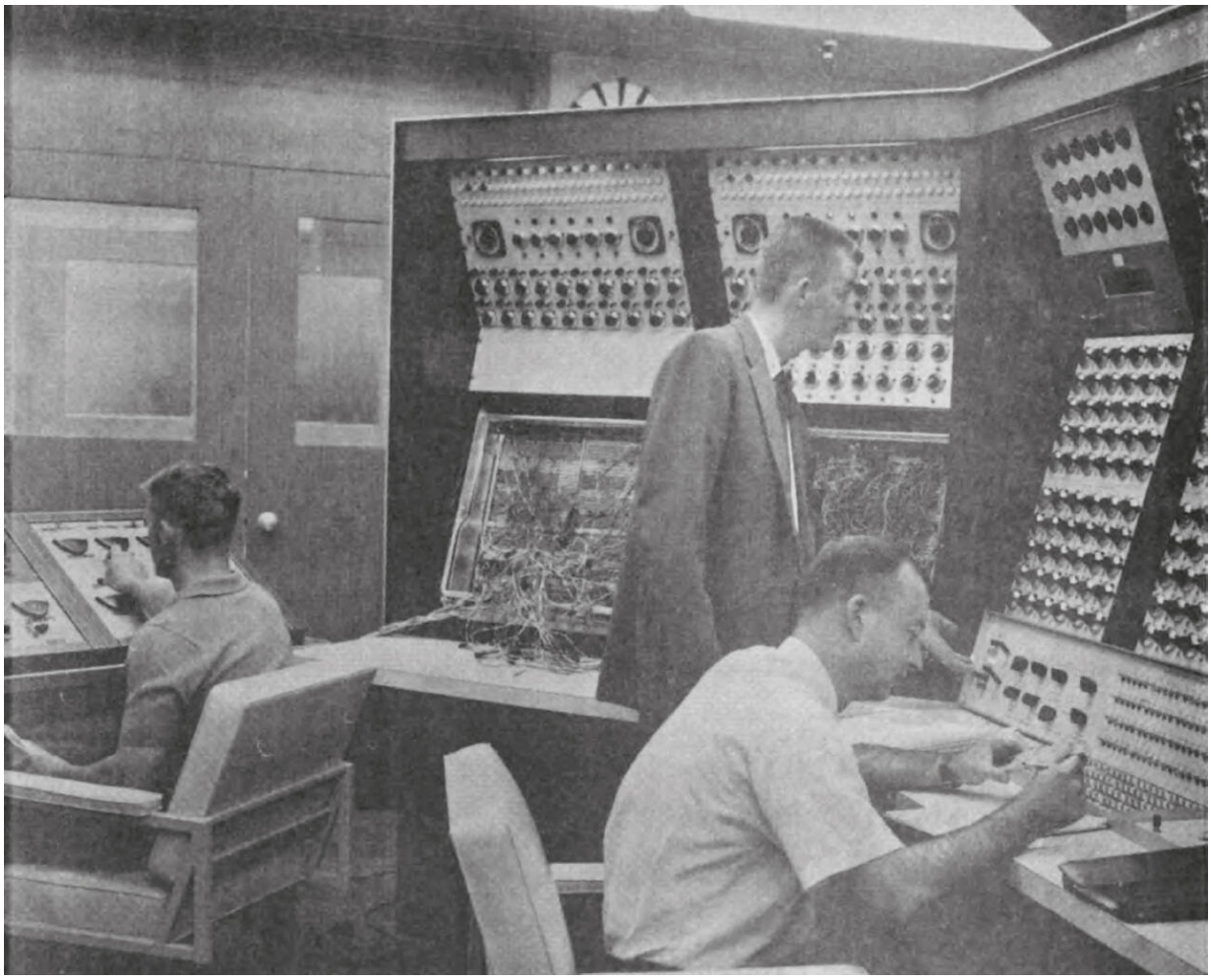


Figure 15.45. ML-1 reactor simulator – part of the reactor control panel can be seen on the left next to the Aerojet analog computer (see [→ Calamore et al. 1963, p. viii])

[→ Fenech et al. 1973] describe an interesting small-scale analog reactor simulator that is realistic enough to train operators, yet small enough to fit on a medium-scale EAI TR-48 analog computer. A complex large-scale analog computer based atomic power plant reactor is described in [→ Irwin et al. 1960]. This system was not only used as a training facility but also as a research tool for further reactor developments. It modeled one of three loops of the Enrico Fermi[165] Atomic Power Plant, a sodium cooled fast breeder reactor, including the reactor itself, a heat exchanger, and a steam generator, taking effects such as transport delays and steam system hydraulics into account.[166] The transport

delays were implemented using a capacitor storage device, a *delay wheel*, roughly resembling a bucket brigade device,[167] This simulator was initially used for control system studies and later adapted to serve as a training system for plant personnel.

15.6.4 Control

Analog computer elements were also used as building blocks for actual nuclear reactor control systems – a natural step, given that analog computers had played an important role during the development of early nuclear reactors. →Figure 15.44 shows some typical analog control elements. These particular modules were used in the control system of two research reactors installed at the University of Virginia.[168]

A comprehensive description of the simulation of the movement of a control rod in a nuclear reactor can be found in [→Bryant et al. 1962, pp. 6–12]. The rod under consideration is cushioned by a hydraulic system suppressing shocks from rapid stops.

[→Chamboredon et al. 1967] describe a hybrid computer study of a fast breeder reactor with respect to control questions. The implementation is quite interesting as it runs in discrete analog computer runs, each devoted to a different part of the overall system under control of the digital computer.

A very interesting cooling loop study for fast breeder reactors with a focus on accidents is described in [→Schönfeld et al. 1967]. The analog computer model is incredibly detailed, including neutron dynamics, effects of temperature changes on reactivity, transport delays, etc. The overall simulation requires two EAI 231RV analog computers with 42 multipliers, ten function generators, and 160 amplifiers in all. The two analog computer are complemented by an EAI DOS 350 digital expansion unit for the implementation of the required time delays using ADCs and DACs in conjunction with a sequential magneto strictive memory.

15.6.5 Enrichment

Uranium enrichment plays a central role in nuclear technology. The task of separating different isotopes from elements such as Uranium is

extremely difficult due to the incredibly tiny differences in mass on which the separation process is typically based. Classic approaches are separation by ultra centrifuges, by mass spectrometer techniques, and gaseous diffusion.[169] [→ Delarousse et al. 1962] describe a large complex special purpose analog computer for the simulation of gaseous diffusion plants based on a complex finite difference system.

15.6.6 Planning of nuclear power plants

An interesting optimization study for a nuclear power plant using an EAI *8900* hybrid computer system is described in [→ Kovacs 1968]. The problem has been solved in parallel on an IBM *360/75*, a powerful mainframe of its time (1968). The speed advantage of the hybrid computer approach over the digital simulation is impressive. Using CSMP on the mainframe, the actual computation took twenty times longer than problem time. It is then estimated that a complex optimization problem like this would require roughly 3000 individual computer runs costing the equivalent of about 6, 000, 000 USD at a typical cost of 600 USD per hour of mainframe time. The same study performed on the hybrid computer showed a speedup of a factor of 200 at roughly half the rental cost of the IBM mainframe giving an advantage of 400 : 1. Eventually the analog part of the hybrid computer was used to implement four different programs, one for the nuclear reactor itself, one for the boiler, one for the turbine section, and one for controllers including various plotters, strip-chart recorders, and graphical displays. The digital subsystem runs six different programs, ranging from automatic setup and checkout routines to a Simplex optimization program.

[→ Barzacchi et al. 1967] describe a highly nonlinear model of a direct cycle nuclear power plant to be used as early as the preliminary planning stages of a power plant. A first implementation using only linear models for the various parts of the overall plant proved to be too imprecise, not least with respect to safety problems.

The study of the operational behavior of a pressure water reactor can be found in [→ Orth 1967]. It turned out that the model had to be simplified as it would have required much more computing elements than available in the two Beckman analog computer (72 integrators, 60

summers, 42 multipliers, 12 function generators, 290 servo potentiometers, 38 manual potentiometers, and 48 manual switches).

An interesting approach to the solution of systems of equations using a digital computer to prepare the problem for an analog computer is described in detail in [→Reinhardt 1967]. Especially remarkable is the wide range of types of equation under consideration, including algebraic equations, linear and nonlinear DEQs, PDEs, as well as highly discontinuous functions, etc. The problems under consideration are so large that an analog computer with up to 300 amplifiers is required.

15.7 Biology and medicine

There are abundant opportunities for applying analog and hybrid computers in medicine and biology. The earliest attempts to analyze complex biological systems were performed by Britton Chance[170] using a mechanical differential analyzer in 1943 to investigate some aspects of enzyme kinetics.[171] The following sections focus on some later analog computer applications in medicine and biology.[172]

15.7.1 Ecosystems

A simple example of the simulation of a closed ecosystem has already been discussed in →section 8.4. A much more complex and interesting simulation is described in [→Rigas et al.] where a food chain spanning from phytoplankton to salmon in an aquatic ecosystem is modeled. This study focuses on external influences such as the effects of fertilizers like phosphates emerging from detergents washed into rivers and eventually the sea, seasonal effects due to incident solar radiation, etc. Interestingly, this study was initially performed on a stored-program digital computer[173] with a graphical display but was then transferred to a mid-range hybrid computer:[174]

> "*[...] The initial programming was laborious and the memory requirements were such that the use of the model was restricted to certain hours of the day. Eventually, rental on the CRT*[175] *was terminated and the interactive capability was lost. At about the same time, a [...] hybrid computer*[176] *became available and the development of an interactive hybrid computer model of the same system became a reasonable alternative.*"

It turned out that this hybrid approach outperformed its algorithmic predecessor and even the restricted precision offered by the hybrid computer had no perceptible influence on the results obtained in these simulation runs.

15.7.2 Metabolism and pharmacokinetics

The study of the metabolism of pharmaceuticals is highly important in modern medicine. The development of radioactive tracers made it possible to gather metabolic data from organisms, which formed the basis for extensive analog and hybrid simulations.[177]

A thorough introduction to the topic can be found in [→ Englert et al.] with a number of worked examples based on the *LADME* approach (*L*iberation, *A*bsorption, *D*istribution, *M*etabolism, *E*xcretion). Many realistic studies in pharmacology are based on *multi-compartment models* where substances are transported between various parts (compartments) of an organism during their metabolism. A number of examples of such studies can be found in [→ Habermehl et al. 1969] and [→ Habermehl et al. 1969/2]. [→ Knorre 1971, pp. 201 ff.] describes the simulation of the metabolism of Paracetamol in an organism.

An extremely complicated simulation of metabolic processes with a focus on the tricarboxylic acid cycle is described in [→ Kubicek et al. 1979]. The model consists of twelve coupled differential equations. An interesting aspect is the use of a hybrid computer where the digital subsystem performs an automatic rescaling process after parameter changes since the equations show highly nonlinear behavior and would otherwise require a tedious manual scaling process after every single parameter change.

Three much simpler pharmacological problems, the metabolism of Glycodiazin, intravenously administered substances, and repeated doses of some medication, are described in [→ Solar 1977/3]. All models are implemented on a small hybrid computer consisting of an EAI *180* analog computer coupled to a microprogrammed EAI *DATA-PACER* digital computer. This digital part mainly performs parameter variations and displays solution curves on a graphic terminal.

15.7.3 Cardiovascular systems

Multi-compartment models are also used to simulate cardiovascular systems, enabling lesions to be studied in detail on an analog or hybrid computer. A complex system simulation based on seven compartments is described in [→BENHAM et al. 1973]. Direct analogies involving intricate hydraulic systems were also used to model cardiovascular systems as shown in [→STEWART 1979, p. 43].

Another application area for analog computers was the preprocessing and analysis of ECG data. Operations such as filtering, peak detection, jitter analysis, etc., have been successfully implemented on analog computers.[178]

The following example shows how *respiratory arrhythmia*[179] has been studied on analog computers. The first such study is described in [→CLYNES 1960], while a later study can be found in [→ALBRECHT 1968]. →Picture 15.46 shows the program used in this study. Its purpose is to determine the parameters controlling the heartbeat function $a(t)$.

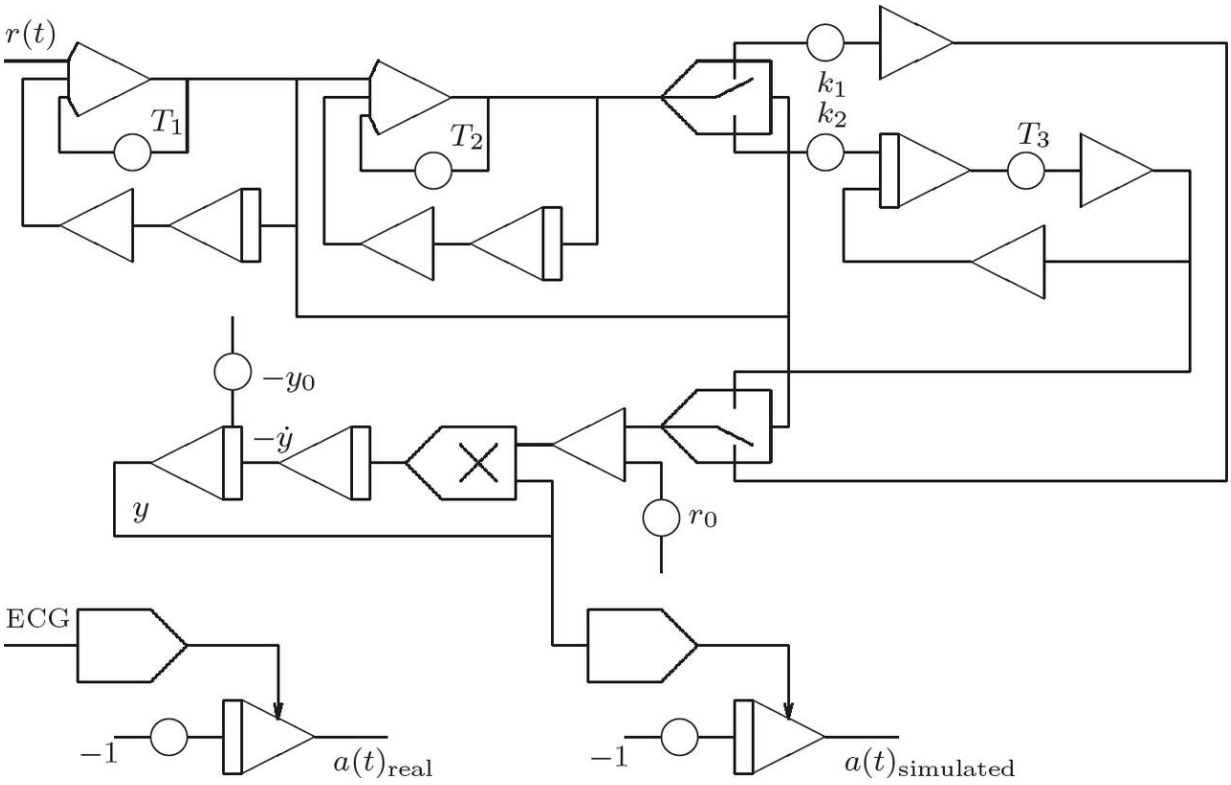


Figure 15.46. Simulation program for the respiratory arrhythmia (see [→ALBRECHT 1968, p. 2])

The integrator in the lower left of this figure is used to generate $a(t)_{\text{real}}$ based on real ECG data of a test subject.[180] This function can then be compared to a function $a(t)_{\text{simulated}}$ generated by the remaining circuit based on the respiratory activity $r(t)$ of the same test subject.

Based on this, the parameters of the simulator circuit can be tweaked until $a(t)_{\text{real}}$ and $a(t)_{\text{simulated}}$ are in good accordance. The resulting parameter set characterizes the behavior of the sinu-atrial node of the test subject and the effects the subject's vagus nerve has on this.

A very impressive and complex analog computer used in cardiovascular research is described in [→Pater et al. 1964]. This system models the entire human circulatory system, divided into two subsystems, one active circuit implementing the action of the heart and one passive circuit implementing the plethora of blood vessels. The heart subsystem is driven by a complex pulse generator allowing for control of parameters such as heart rate, systolic and diastolic duration, atrioventricular delay, etc. A heart rate of 60 beats per minute at rest in a human is represented by a 1 kHz signal in the analog computer simulation. Individual blood vessels are divided into 6 cm long sections, each implemented as a passive RLC-circuit. Peripheral blood vessels with very small diameters even forgo the inductances since resistance with respect to fluid flow is the dominant feature in this case.

→Figure 15.47 shows the overall system implementing the human circulatory system shown on the far left. The impressive fidelity of this simulation system is evident in →figure 15.48. Here, actual measurements done in vivo are shown on top, while the corresponding simulation results are shown below.

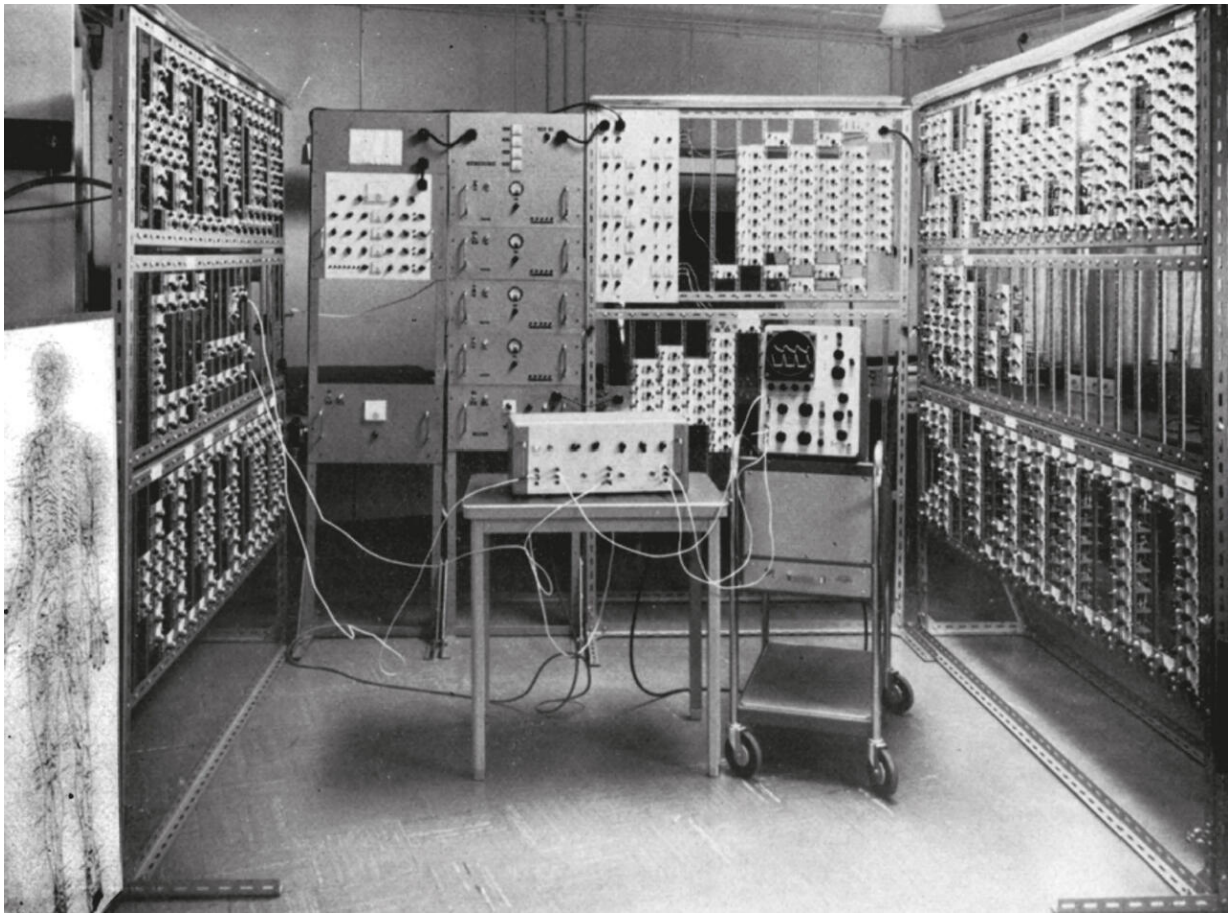


Figure 15.47. Overall view of the analog computer simulating the entire human circulatory system (see [→ Pater et al. 1964, p. 165])

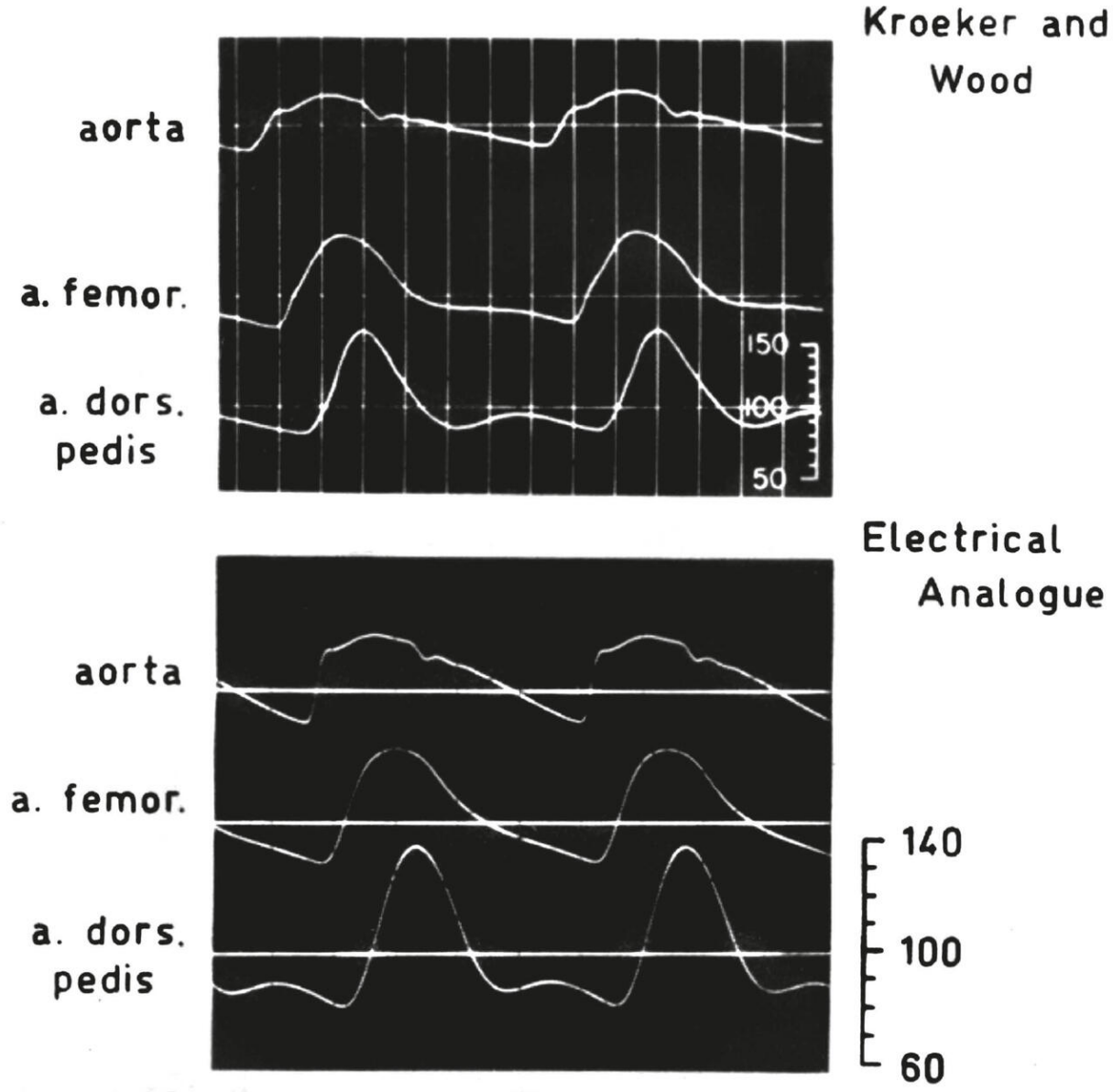


Figure 15.48. Actual measurements obtained in vivo (top), compared to simulation results (bottom) – the similarity is striking (see [→ PATER et al. 1964, Fig. 8])

The importance of analog computers in this particular field of research is also exemplified by the fact that the manufacturer Beckman even advertised their systems in a 1960s advert as "*The Analog Computer that Pumped Blood*".[181]

15.7.4 Closed loop control studies

A nice example of the simulation of closed loop control in organisms is featured in [→Stewart 1979, pp. 46 f.], which describes the regulation of CO_2 in the lung. The simulation of the adaptation process of the pupil to changes in environmental brightness is described in [→Giloi 1975, pp. 157 ff.], [→Künkel 1961], and [→Lunderstädt et al. 1981].[182]

A very impressive special purpose built analog computer is described in [→Fincham 1963]. The machine helped to investigate regulation mechanisms of blood volume and concentration of alkalines in blood plasma. The simulation consists of three subsystems: The *basic system* implementing processes of water, sodium, and potassium distribution in the body, the *regulatory system* modeling the endocrine system, the regulation of water, sodium, and potassium excretion, and, finally, the *injury-response system* modeling effects of large external perturbations such as the administration of substances, sudden and large blood losses, etc. The overall model is incredibly detailed, even taking interstital protein masses into account.

Scaling such a vastly complex system was a major undertaking and was done in a way that a single scaled problem was run for all sets of parameters, sometimes resulting in very small values "*close to the range in which hum pick-up and the noise voltages inherent in electronic apparatus become troublesome.*"[183] Time scaling was done so that two seconds of analog computer runtime correspond to one hour of real time. The main analog computer also required a variety of external devices to create stimuli such as the injection of hormones, sudden haemorrhages, etc. →Figures 15.49 and →15.50 give an impression of the overall system.

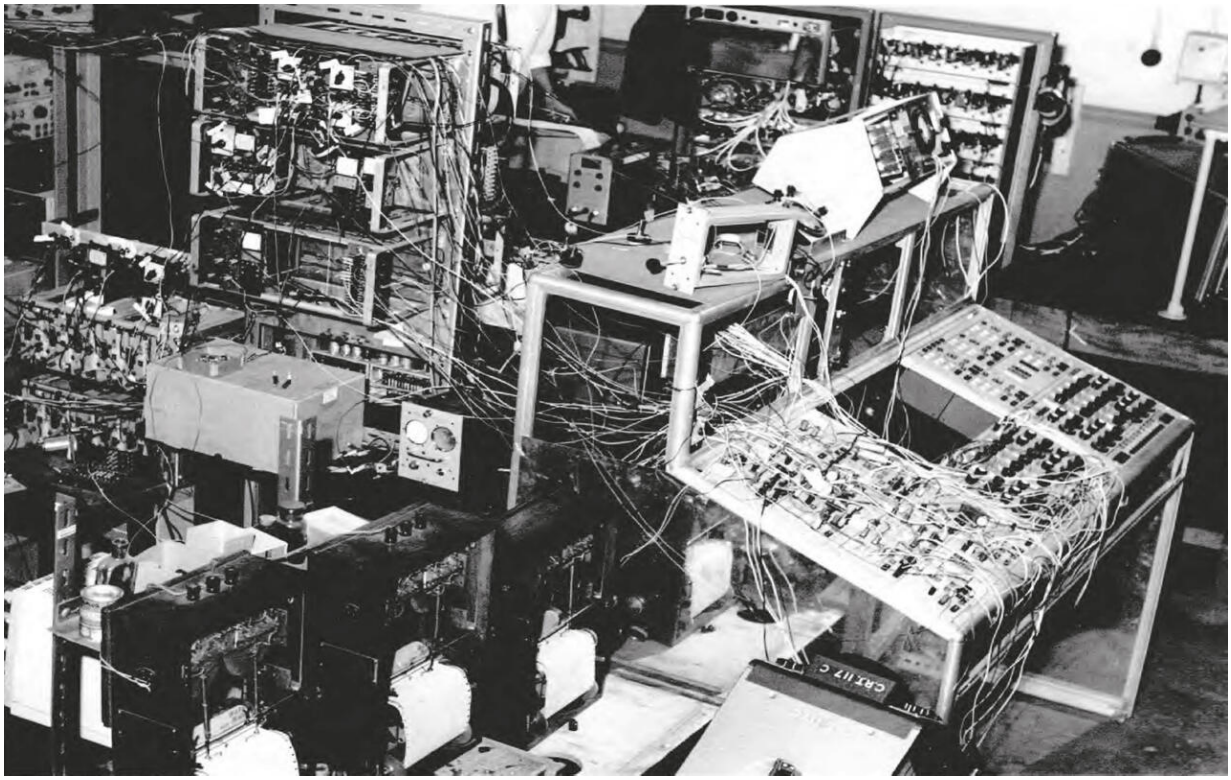


Figure 15.49. Part of FINCHAM's simulation system with the central control desk (see [→ FINCHAM 1963, p. 234])

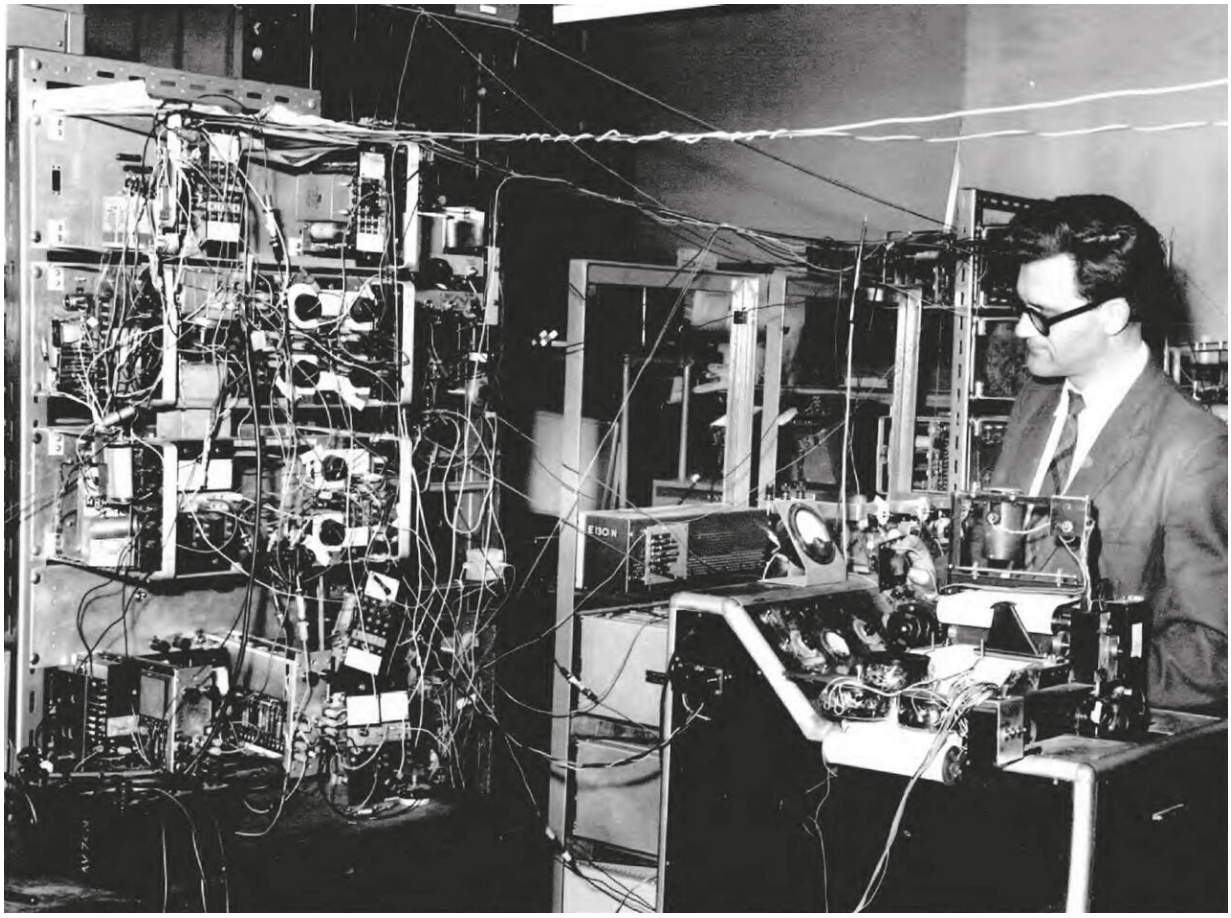


Figure 15.50. William Frank Fincham standing next to his machine (see [→ Fincham 1963, p. 236])

Another rather complex simulation of the pulmonary and renal acid-base regulation mechanisms is described in [→ Nedey et al. 1967]. The system consists of 40 coupled equations, requiring additional means for introducing time delays.

15.7.5 Neurophysiology

A central question in neurophysiology is how action potentials in nerve cells are generated and propagated. In 1952 Alan Lloyd Hodgkin[184] and Andrew Fielding Huxley[185] developed a mathematical model describing signal generation and propagation in squid giant axons.[186] This model is described by the following differential equation[187]

$$C\dot{V} = -g_{\mathrm{Na}} m^2 h \left(V - V_{\mathrm{Na}}\right) - g_{\mathrm{K}} n^4 \left(V - V_{\mathrm{K}}\right) - g_{\mathrm{L}} \left(V - V_{\mathrm{L}}\right) + I_a,$$

where g_i represents the conductivity caused by the potassium and sodium ions and the unavoidable leakage current in the axon, while $(V - V_i)$ represent differences to the equilibrium potentials V_i. The coefficients $0 \le m, h, n \le 1$ are determined by

$$\begin{aligned}
\dot{m} &= \alpha_m(V)(1 - m) - \beta_m(V)m, \\
\dot{h} &= \alpha_h(V)(1 - h) - \beta_h(V)h, \quad \text{and} \\
\dot{n} &= \alpha_n(V)(1 - n) - \beta_n(V)n.
\end{aligned}$$

where $\alpha_i(V)$ and $\beta_i(V)$ are determined through experiments. This model explains many experimental observations and formed the basis of later research by Richard FitzHugh[188] and Jin-ichi Nagumo,[189] which resulted in the following set of differential equations that was extensively studied using analog computers:[190]

$$\begin{aligned}
\dot{v} &= v(a - v)(v - 1) + w - I_a \\
\dot{w} &= bv - \gamma w
\end{aligned}$$

A more complex model for neural bursting and spiking devised by James L. Hindmarsh and R. M. Rose has already been described in →section 8.12.

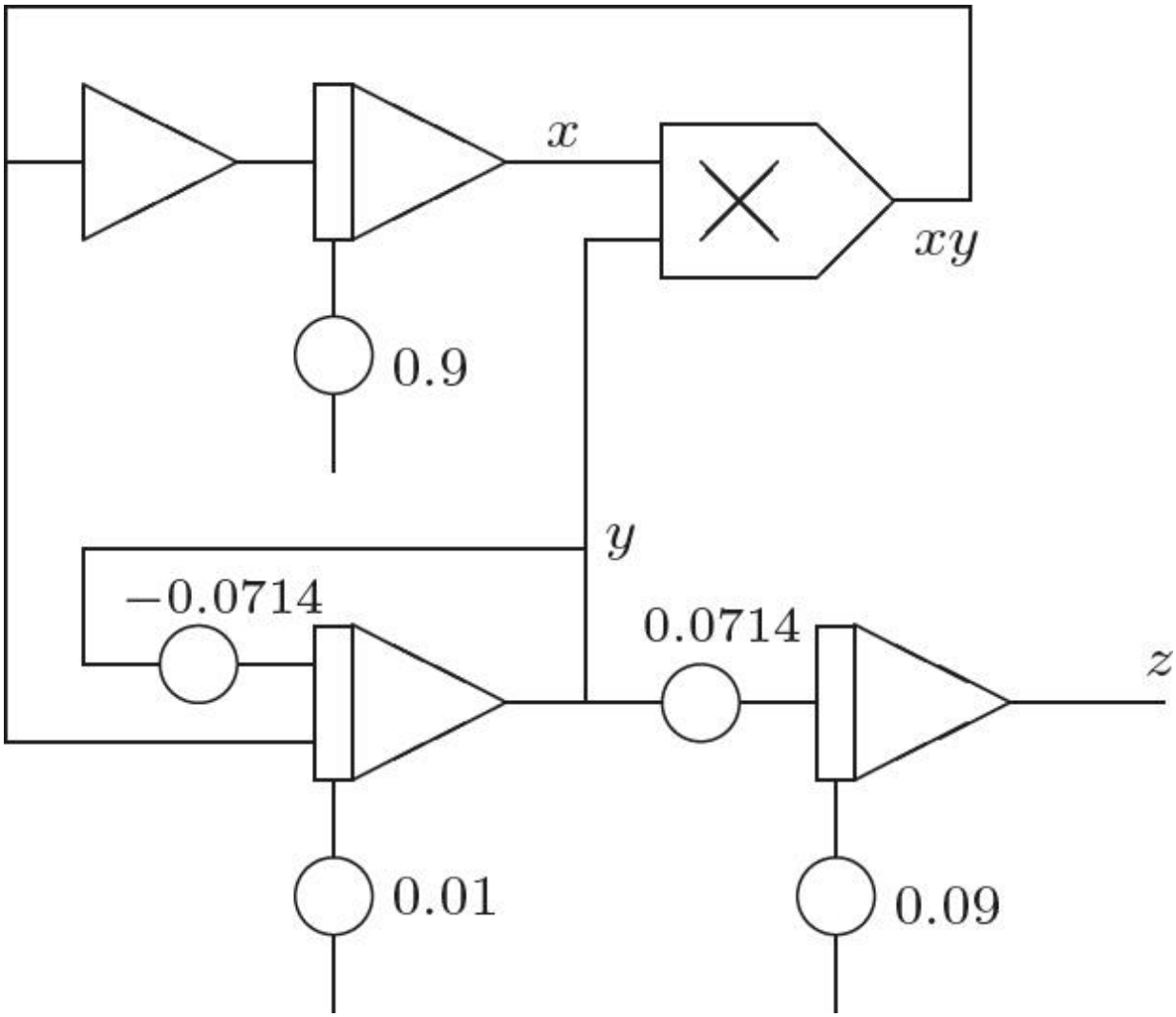


Figure 15.51. Setup for an infection spreading simulation (see [→Hitachi 200X, p. 14])

[→Damjanovic et al. 1967] describe the implementation of networks of *neuromimes*, i. e., analog equivalents of biological nerve cells. However, the resulting analog circuits are rather simple, using comparators to create spike-like signals. A study of various modes of signal transmission in biological neural systems can be found in [→Bourton et al. 1967].

Last but not least the *perceptron* should be mentioned in this context – an early hardware implementation of an *artificial neural network*[191] (*ANN*) which not only laid the foundations for today's ANNs but also inspired the development of another interesting, though now mostly forgotten novel electronic component, the *memistor*, a resistor with

memory, as described in [→WIDROW 1960].[192] This novel element was suggested to be used as a variable weight for the implementation of artificial neurons. The work then continues to describe the implementation of the *Adaline* (*adaptive linear*) computer based on these memistors.

15.7.6 Epidemiology

A simple example for the application of analog computers in epidemiological research is given in [→Hitachi 200X, p. 14], which describes a closed system containing a number of individuals, a few of whom are initially infected with a contagious disease. An infected individual can then infect other persons who are not already immune.[193] The output of a typical simulation run is shown in →figure 15.52.[194]

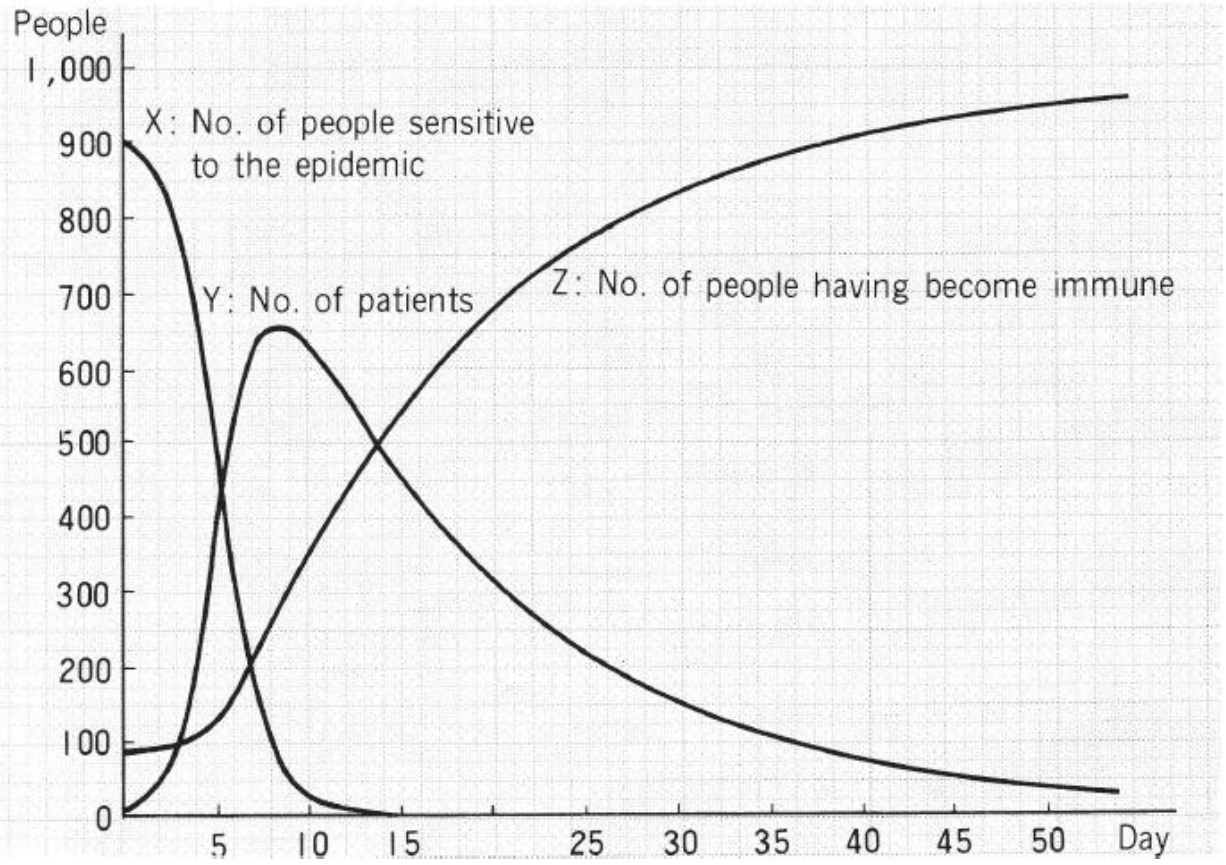


Figure 15.52. Results of a simulation run for the infection spreading simulation ([→Hitachi 200X, p. 14])

15.7.7 Aerospace medicine

Analog computers were also used extensively in aerospace medicine research. [→ Paynter ed. 1955] describes the analysis of the forces acting on a pilot bailing out with an ejection seat. For this study a special-purpose analog computer was developed to model the effects of accelerations which typically arise during ejection. The purpose of such *body dynamics studies* is to determine optimal acceleration curves that guarantee a quick and safe separation of the ejection seat from a failing plane while not harming its passenger.[195]

Analog computers were also used to control large-scale centrifuges used to determine the performance of pilots subjected to high and varying accelerations as would be encountered during emergency situations, rocket launches, or re-entry maneuvers. A highly complex centrifuge control system based on a large-scale analog computer installation is described in [→ Stone et al. 1965].

15.7.8 Locomotor systems

Analog computers were also used at the boundary between biology and mechanics in locomotor studies. [→ Tomovic et al. 1961] performed simulations to determine the degree of autarky in typical locomotor systems. Similar studies are described in [→ Bekey et al. 1968, pp. 417 ff.].[196]

15.7.9 Dosimetry and related fields

Analog computers were heavily used in nuclear medicine as an example from the *University of Chicago hospital* shows. Here a *Beckman EASE 2132* analog computer was used from the mid 1960s well into the 1970s to perform dosimetry calculations, a task required for cancer treatments. In addition to that, this particular machine was also used for basic research involving nuclear and non-nuclear applications.[197]

An interesting application of analog computers to the evaluation of radioisotope renograms is described in [→ Kathel et al. 1973]. A patient is injected some radioactive tracer substances with the aim of determining detailed information on the function of the kidneys. In the simplest case,

the concentration of the tracer is determined by external radiation measuring equipment, yielding a curve quickly rising after the initial administration of the tracer until a peak is reached. The curve then decays according to the excretion capacity of the kidney. These curves are then optically overlaid with a curve displayed on an oscilloscope connected to an analog computer. The computer operator then performs a manual parameter fitting process by changing parameters on the analog computer by means of manual coefficient potentiometers so that the differences between the measured curve and the one generated by the analog computer are minimized. The parameters obtained by this process are then used to mathematically describe the patient's kidney functions in detail.

[→W. Hofmann 1977] describes a hybrid computer model for the simulation of the dispersion of naturally occurring radioisotopes in the air of living and work spaces. Although the problem consisting of eleven individual compartments can be described by a system of coupled linear differential equations, the number of equations is so large that a digital approach was ruled out due to performance reasons. Scaling and parameter variation were done by the digital part of the hybrid computer used in this study.

In a follow-up study, [→Hofmann et al. 1977], the effects of age in humans on the overall dose by inhalation of naturally occurring radionuclides are simulated. For this, the respiratory system of a human is implemented as a compartment model with age depending geometries and physiological characteristics.

15.7.10 Training

A life like patient-simulator, *Sim One*, used for the training of physicians in anaesthesiology, is described in [→Computers and Automation 1967]. Developed by medical researchers at the University of Southern California's School of medicine, the system "interfaces" to the physicians by means of a highly realistic manikin resembling a patient prior to being anaesthetized for a surgical procedure. The manikin features a realistic mouth section, bronchial tubes, a heartbeat, a carotid, breathing action, etc. This highly complex system is controlled by an EAI 231R analog computer coupled with a Computer Control Company *DDP-24* 24 bit

digital computer. Using the digital computer, an instructor can change system parameters during a training session or hold the simulation temporarily to discuss problems or questions. Eventually, data gathered by the digital computer is printed documenting every step of the medical personnel and the resulting reaction of the analog patient simulation.

15.8 Geology and marine science

Analog computer applications in the fields of geology and marine sciences were often of great commercial value, especially with respect to prospecting tasks.

15.8.1 Oil and gas reservoirs

Oil and gas production relies heavily on hydraulic models of real oil and gas fields. These models are highly complex and require the solution of partial differential equations, typically with four variables. Since good models result in better revenues, several very large and complex direct analog computers have been built to study the effects caused by oil wells to the below-ground distribution of oil and gas.

An impressive example for such a direct analog computer is the *ZI-S* system, which was developed in 1958 at the *All-Union Scientific Research Oil-Gas Institute* in the Soviet Union.[198] This machine consisted of a three-dimensional grid of passive computing elements with about 20,000 vertices. In addition to this, it also contained a large number of operational amplifiers and other active components.[199] Such a large number of components was required to achieve a sufficiently high resolution of the model.

This ZI-S analog computer shown in →figure 15.53 was based on

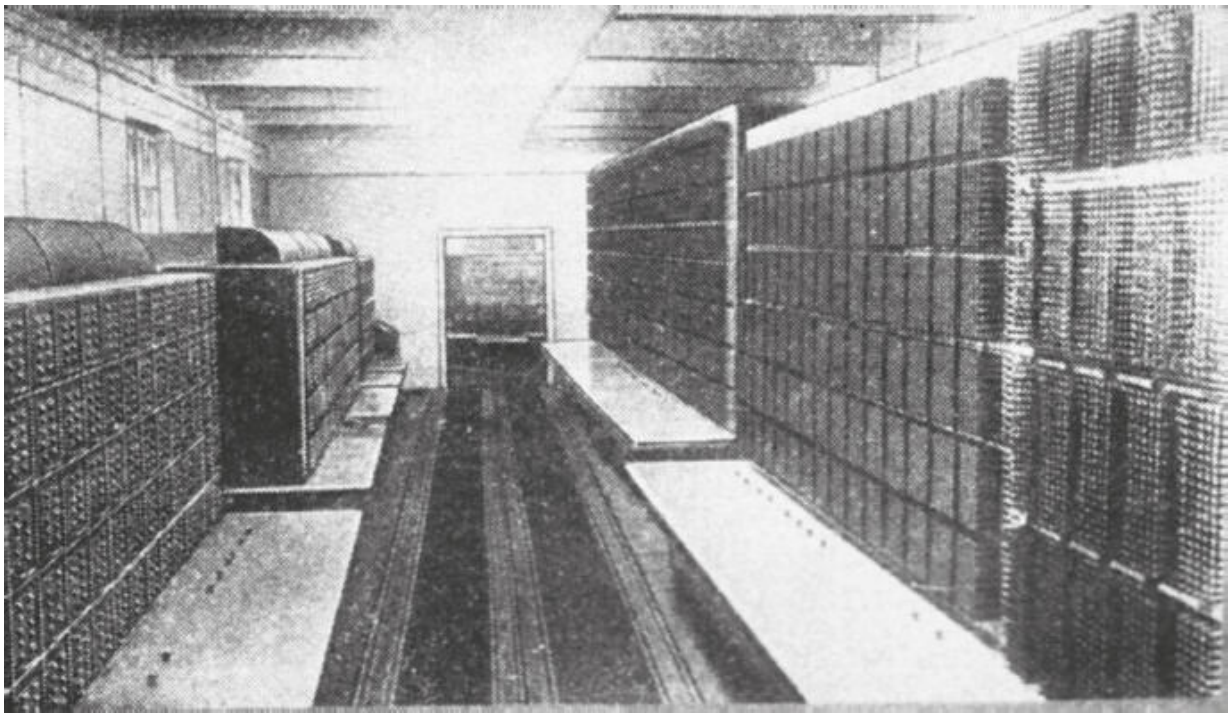


Figure 15.53. Overview of the ZI-S computer ([→USHAKOV 1958/1, p. 1812])

$$\frac{\partial}{\partial x}\left(A_1(x,y)\frac{\partial P}{\partial x}\right)+\frac{\partial}{\partial y}\left(A_2(x,y)\frac{\partial P}{\partial y}\right)=0$$

and

$$\frac{\partial}{\partial x}\left(A_1(x,y)\frac{\partial P}{\partial x}\right)+\frac{\partial}{\partial y}\left(A_2(x,y)\frac{\partial P}{\partial y}\right)=A(x,y)\frac{\partial P}{\partial t}$$

respectively, where *x* and *y* represent coordinates in a single layer of the oil or gas reservoir, while *P* and A_i denote pressures and hydraulic conductivities.

Using this system, reservoirs with a radius of up to 120 km were studied. These simulations considered up to 750 wells depleting the reservoir. The passive analog computer simulating the reservoir itself can be seen in the middle of →figure 15.54. The reservoir is modeled with a three-dimensional mesh of resistors and capacitors, which implement lag phenomena, etc. The surrounding modules implement the wells and the instrumentation of the simulation circuits.[200]

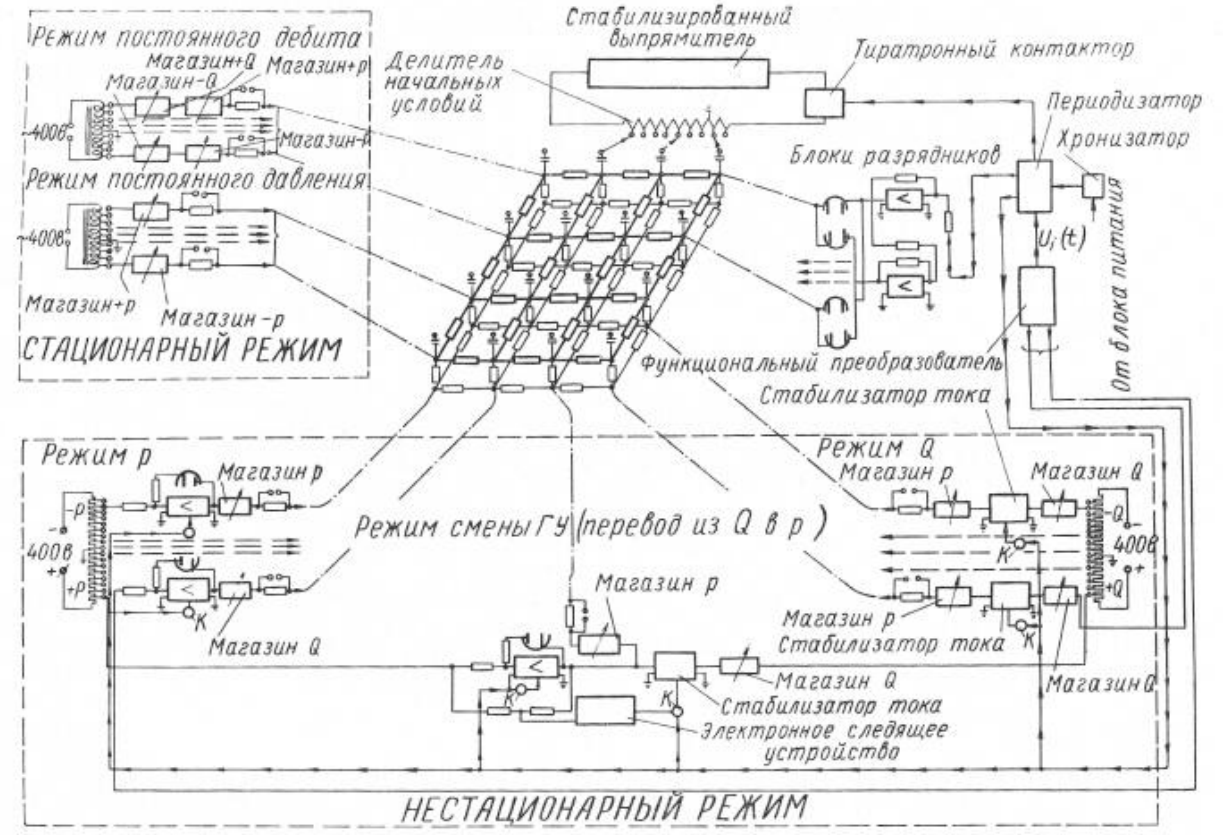


Figure 15.54. Structure of the ZI-S computer ([→ USHAKOV 1958/1, p. 1813])

A comprehensive account of the use of analog computers for modeling oil reservoirs at *British Petroleum* (*BP*) can be found in [→ CARE 2006/2]. The machine used as the base for this application was the *EMIAC II*, built by *Electrical and Musical Industries* (*EMI*), originally an entertainment company.[201] The final system consisted of nine large racks holding two EMIAC II systems each, complemented by additional equipment purpose built for reservoir simulations.

The application of analog computers by the California Department of Water Resources to the problem of injecting water into ground water basins in order to replenish this natural water storage system is described in [→ EAI 1961/2]. The first question was where to perform such water injections in order to yield an even distribution of this added fluid. The second question regarded suitable locations for water

injections to suppress or at least retard the ingress of salt water into the underground aquifer.

15.8.2 Seismology

Prospecting typically relies heavily on the analysis of seismograms to detect below-ground reservoirs. This task traditionally requires the fastest stored-program computers available and was thus done, or at least supported by, analog and hybrid computers in the years before the mid 1970s.

[→Evans 1959] describes a special-purpose analog computer developed and used to generate seismic weathering time corrections for refraction and reflection studies.[202] The implementation of this analog computer is quite similar to the BPRR-2 system described in →section 15.1.11. Indirect analog computers were used by [→Sutton et al. 1963] for filtering tasks as well as spectrum analyses on seismic data sets.[203]

15.8.3 Ray tracing

Acoustic ray tracing is a common task in marine applications using sonar data and of high commercial and especially military value. This task is quite complex due to varying salt concentrations, temperature differences between water layers, etc.

[→Light et al. 1966] describe a ray tracer based on a small EAI TR-10[204] analog computer. The task of tracing a single acoustic wave from a transmitter through a complex medium like open sea to a pickup involves the solution of a second degree differential equation. The analog computer used in this setup contained 17 operational amplifiers and a special function generator to model the structure of the ocean bed reflecting acoustic waves. A typical simulation run took about 15 minutes to complete. →Figures 15.55 and →15.56 show the analog computer setup as well as a typical simulation result.

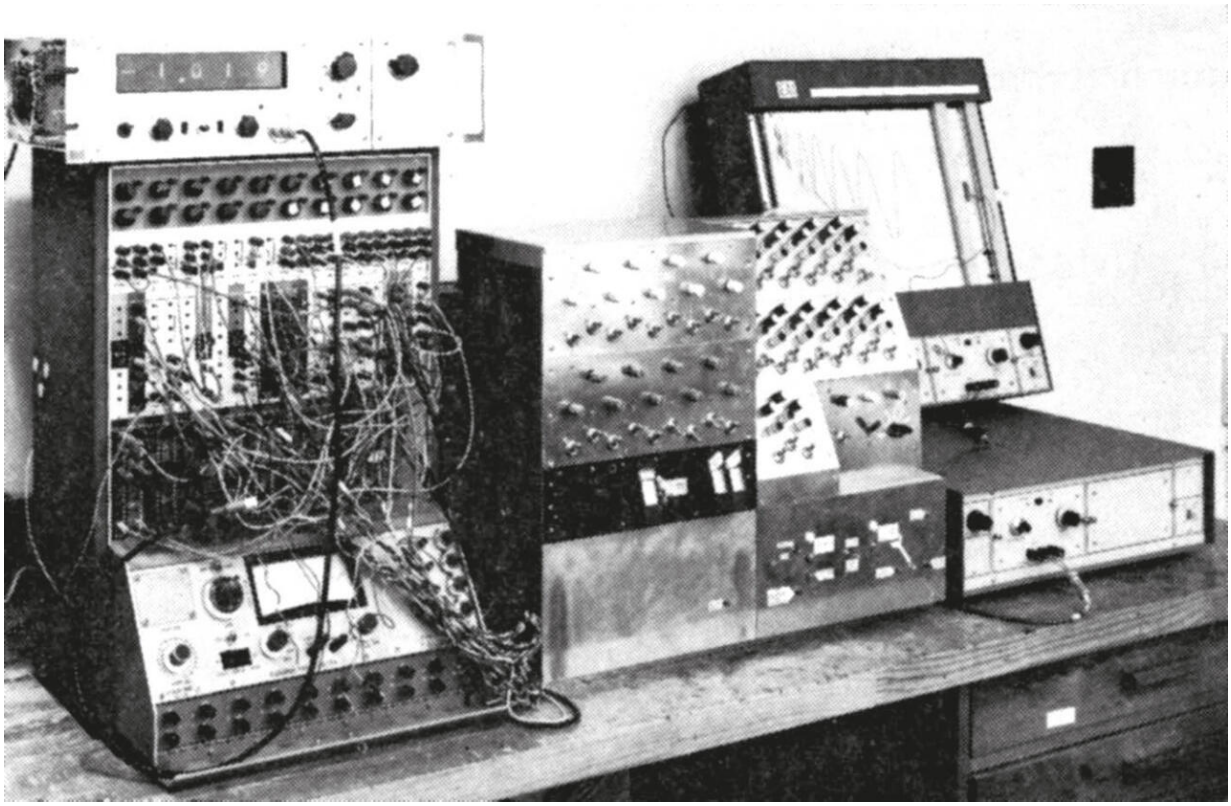


Figure 15.55. The ray tracer developed by LIGHT, BADGER, and BARNES (see [→ LIGHT et al. 1966, p. 724])

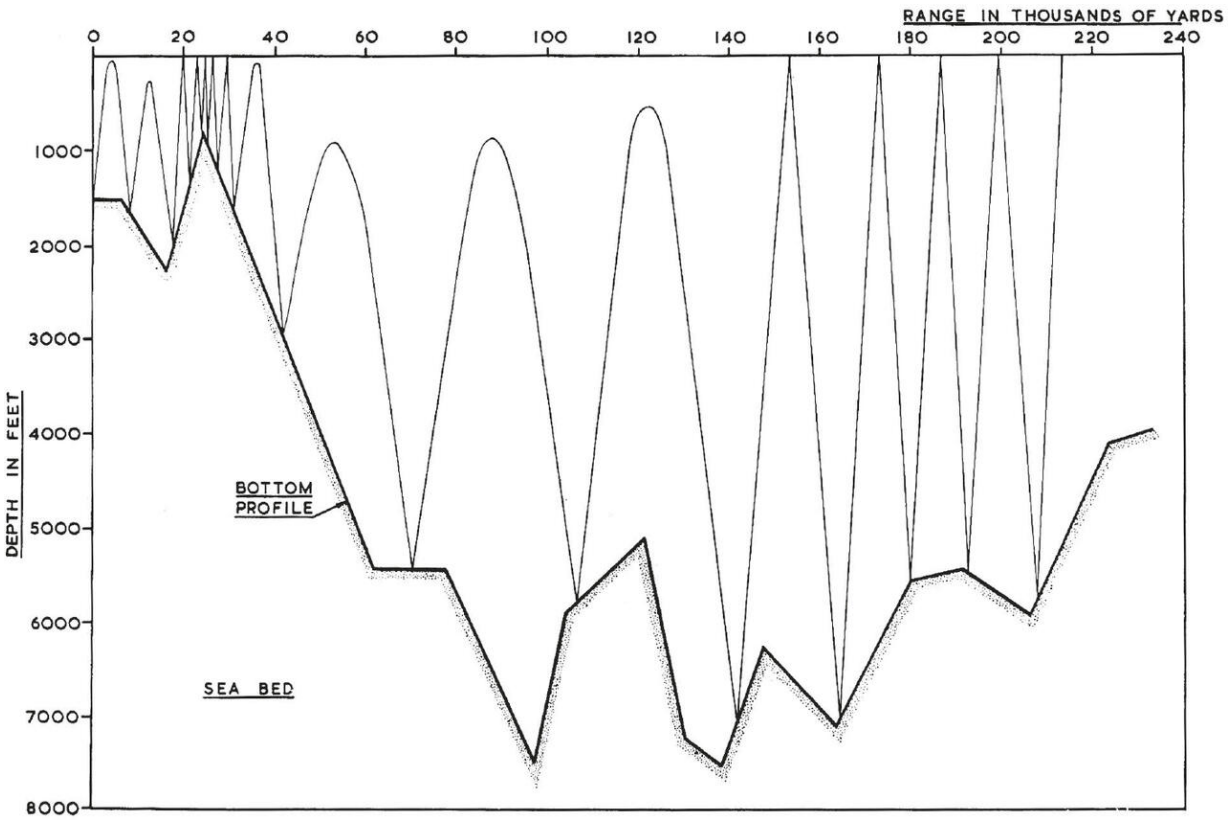


Figure 15.56. Result of a simulation run of the ray tracer (see [→ Light et al. 1966, p. 724])

15.9 Economics

Alban William Phillips,[205] a New Zealand economist, was one of the first people to build economic models using analog computers. Being a trained engineer, the idea of applying engineering principles to economic problems was an obvious approach. Consequently, in 1949 he developed and built *MONIAC*,[206] a hydraulic analog computer, whose structure is shown in →figure 15.57.[207]

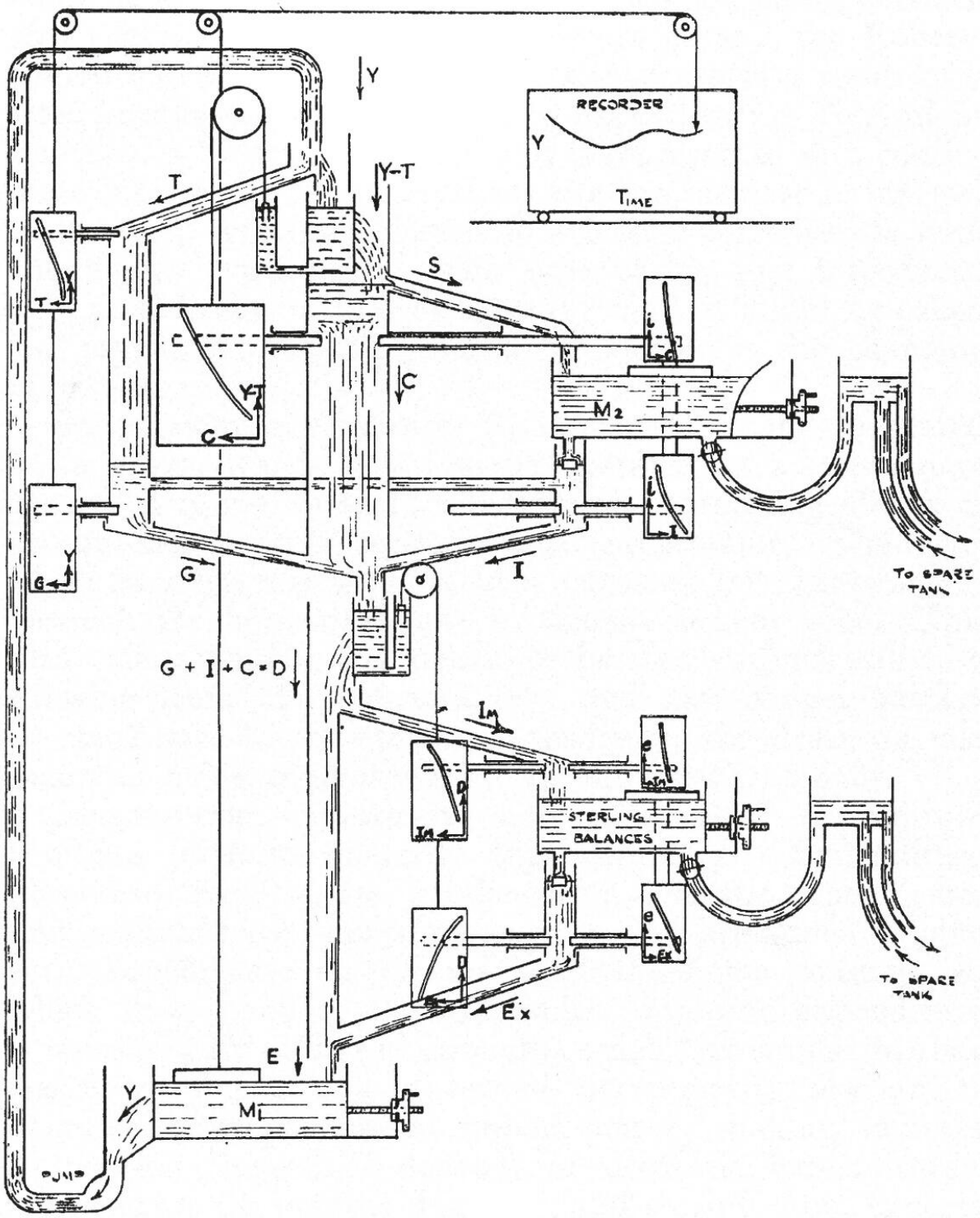


Figure 15.57. PHILLIPS' hydraulic simulator for economic dynamics (see [→ PHILLIPS 1950, p. 302])

This system, of which eventually ten were built, was affectionately known as the *financephalograph* and was not only used for early research in mathematical economics but also, due to its structural clearness, for teaching purposes. The flow of money in an economy is represented by colored water flowing through various hydraulic computing elements. MONIAC features nine control variables representing tax load, import/export subsidies, and the like.[208] Recent detailed descriptions of the overall system can be found in [→ Ryder 2009] and [→ Ryder et al. 2021].[209]

In later years, Philipps used an electronic analog computer at the *National Physical Laboratory* (*NPL*) as well as a stored-program digital computer for his research.[210] Interestingly, he still preferred the hydraulic MONIAC despite its shortcomings with respect to accuracy, which were more than outweighed by its high clarity. He characterized his models as being "*exposition[s] rather than accurate calculation[s]*".[211]

Analog computer models for the study of economic problems were still used in later years as a number of publications like [→ Hülsenberg et al. 1975], [→ Herschel 1961], [→ Kregeloh 1956], and [→ Jackson 1960, pp. 363 ff.] show. An electronic recreation of Phillip's MONIAC is described in [→ Foreman 2020] using operational amplifiers and SPICE as simulation tool.

An interesting little preprogrammed "personal analog computer" was the *ISEC 250*.[212] This machine was aimed at stock investment analysis and used an *ISEC index*, a weekly index table distributed by the manufacturer, as well as real-time inputs based on current market behavior entered through means of some potentiometers and switches, to give an "*instant evaluation of any programmed stock*".[213]

[→ Strotz et al. 1951], [→ Strotz et al. 1953], and [→ Engel et al. 1964] contain a wealth of practical examples for the application of analog computers to problems in economics such as the effects of marketing and production inertias, national income models, a nonlinear theory of the business cycle, the effects of policy decisions, and many more. The analog simulation of a colonial socio-economic system is described in detail in [→ Howard 1961].

A detailed and highly realistic simulation of a phosphate plant as a tool for management decisions is described in [→ Valisalo et al. 1967]. The

analog computer setup shown in →figure 15.58 is part of a hybrid system where the digital part is mainly used for automatic or user controlled parameter variations. The system can also be fed with noise signals to gather statistics of profits and inventories.

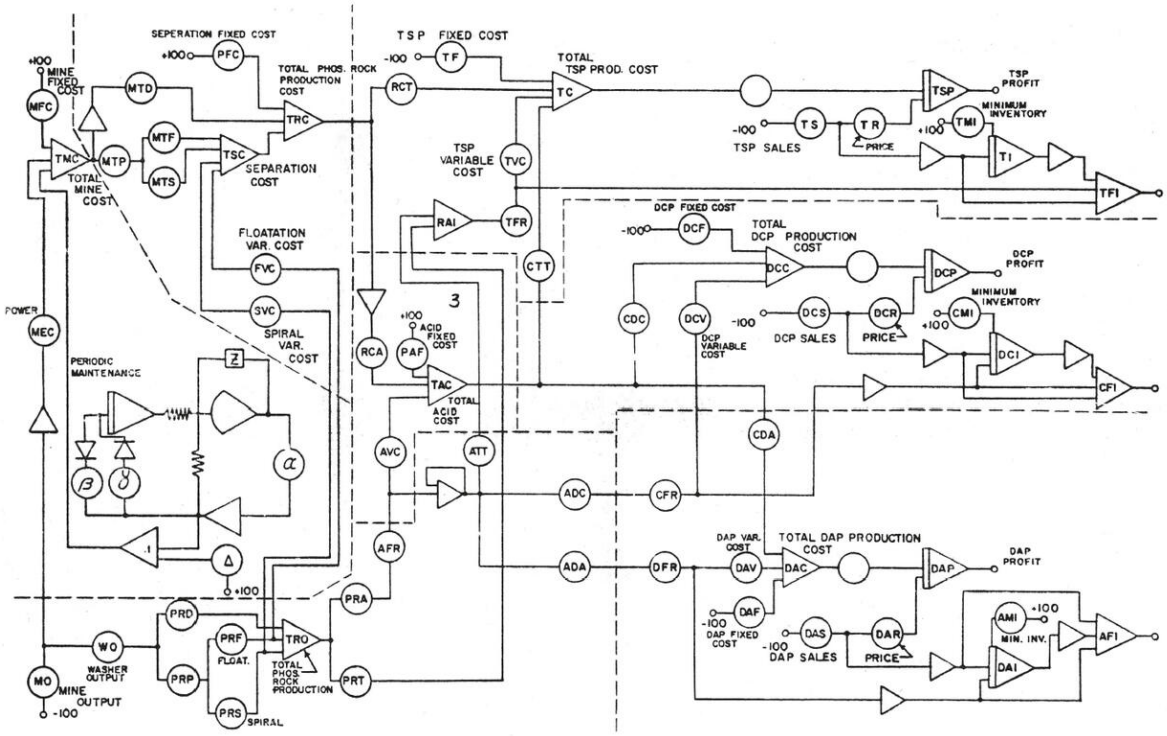


Figure 15.58. Analog computer setup for the phosphate plant simulation (see [→VALISALO et al. 1967, p. 866])

15.10 Power engineering

In the early 20th century, the field of power engineering developed a demand for computational power that could only be fulfilled by analog and later hybrid computers. As with nuclear reactors, many questions regarding power systems cannot be analyzed using real equipment due to the inherent dangers of such experiments.[214] Consequently, simulation techniques were, and still are, the only way to solve a number of vital problems in this field. The following sections describe typical simulation tasks.[215]

15.10.1 Generators

A study of the dynamic behavior of generators feeding large power grids containing only asynchronous machines as loads is described in [→Juslin 1981].[216] The main problem of such power grids is that asynchronous machines exhibit vastly different load characteristics depending on their state of operation. A starting asynchronous machine behaves like a transformer with its secondary shorted, an effect vanishing with increasing motor speed. This study's focus is on severe faults influencing large parts of the simulated power grid, which will result in oscillatory behavior that might be unstable.

[→Gilbert 1970] describes the simulation of a generator with a parasitically loaded speed controller to be used in aerospace applications. This model consists of the turbine driven alternator, a voltage regulator and exciter, a parasitic load speed controller, and the load itself.

The simulation of a three phase asynchronous generator taking the iron saturation into account is described in [→Kheireldin 1967]. The system is used for the evaluation of effects such as speed variations on the terminal voltages for various values of the exciting capacitors of the generator, etc.

An very detailed simulation of a synchronous machine connected to an ideal (infinite) bus through a transmission line is described in [→Anderson et al. 2003]. The seven coupled differential equations representing the machine are highly nonlinear, not only containing products but also various trigonometric terms resulting in a rather convoluted analog program as shown in →figure 15.59.

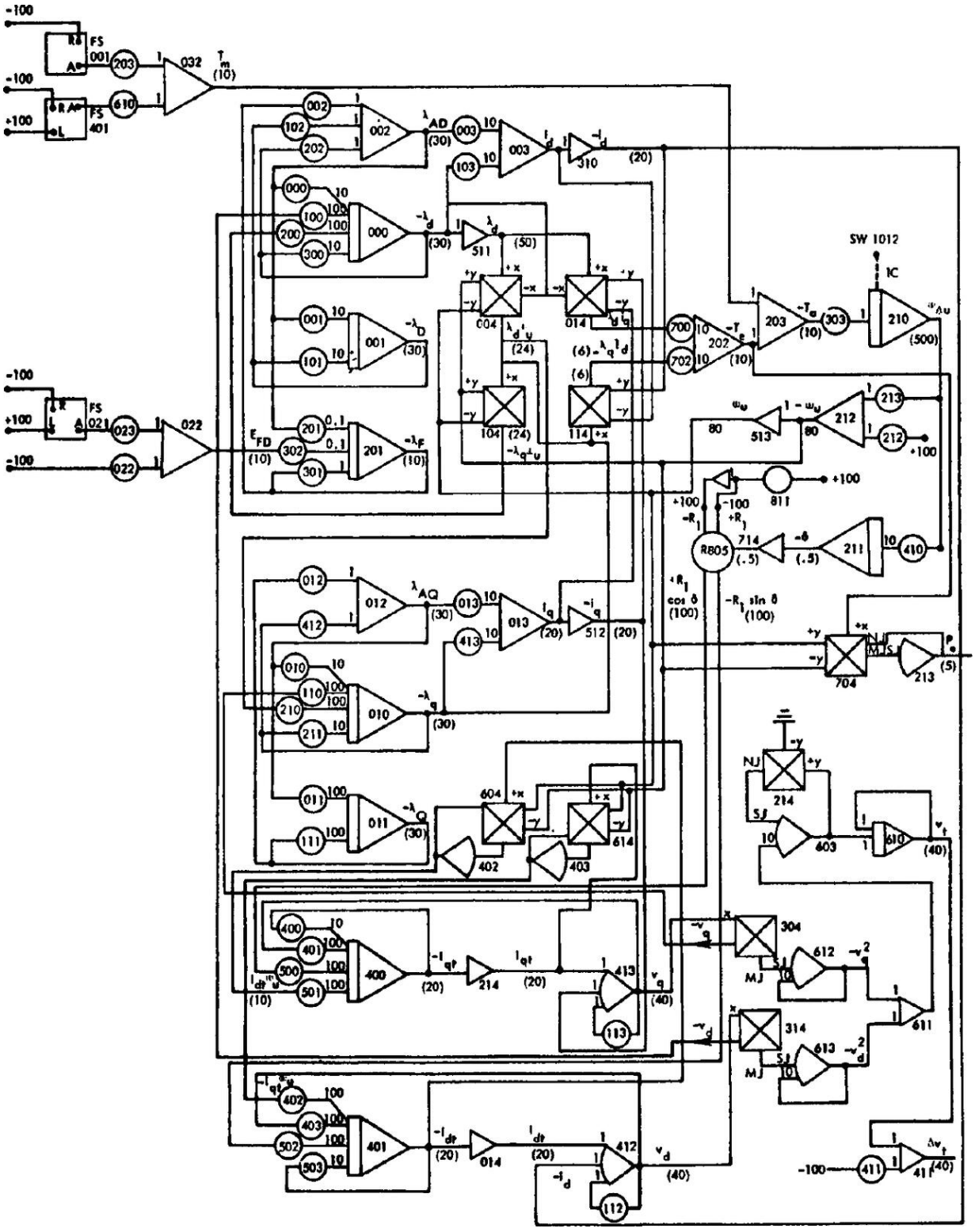


Figure 15.59. Analog computer setup for the simulation of a synchronous machine connected to an infinite bus by means of a transmission line (see [→ANDERSON et al. 2003, p. 175])

15.10.2 Transformers

Although a transformer might appear simple at first sight, its behavior is non-trivial to model and predict. Many studies into the dynamic characteristics of transformers have been conducted. [→Giloi 1960] and [→Telefunken 1963/2] cover the simulation of a transformer under mixed resistive/inductive or resistive/capacitive loads. In this study, which also takes the dynamic *B*-*H* magnetization curve[217] of the transformer into account, the primary and secondary currents are simulated.

15.10.3 Power inverters and rectifiers

Much more complex than this is the simulation of power inverters and (controlled) rectifiers. Devices like these are essential for high-voltage direct current systems, which were first developed in the early 20th century. Inverters are also commonly used to drive induction motors, etc.[218]

The analysis of a controlled three-phase rectifier by an analog computer model is described by [→Eyman et al. 1976], while similar studies can be found in [→Blum et al.], [→Tisdale[a]], and [→Nava-Segura et al.]. The latter focuses on fault situations such as shorted thyristors. [→Shore 1977] describes a hybrid computer based simulation as well as an online digital computer control setup for a high voltage direct current link.

The conversion of a direct current into alternating or three-phase current by power inverters has been analyzed in analog computer studies such as [→Krause 1970], where basic simulation techniques in this field are described, and [→Fornel et al. 1981], where a power inverter loaded by an asynchronous machine is modeled. [→Bellini et al. 1975] describe the analysis of an inverter driving electromechanical actuators, while [→Winarno 1982] focuses on the special requirements posed by large photovoltaic systems. The analog model of a complex high-voltage direct current system is described in [→Sadek 1976]. A comprehensive study of analog computer simulations of various inverter circuits is described in [→Gerecke 1967/1]. An interesting detail in this study is the implementation of the diode models which takes even the tiny reverse

currents into account by means of comparators switching between two parts of the characteristic curve of the simulated diode.

[→Gerecke et al. 1967] describe a complex study of thyristor controlled three-phase squirrel cage motors, the results of which are compared with actual measurements on a 4 kW motor and show good correspondence. Of special interest is the fact that several properties of the machine cannot be measured on a real system and are only available through simulation.

The application of an analog computer for the development of optimum speed control using the Pontryagin maximum principle is described in [→Bohren 1967]. Interesting is the linearization process used in this study. Since the overall system is highly nonlinear, linearization is applied on small scales, each spanning an interval of a certain operating point.

[→Borchardt et al. 1977] give a detailed description of the simulation of a high-power inverter system driving the magnets of a large synchrotron at DESY. This study was done with the synchrotron already in operation as the need for significant changes in the power supply system for the magnets arose during its operation. The system itself could not be used to take measurements during parameter changes since this would have shortened its beam-time. Thus, an extensive simulation was performed on a *HRS 860* Telefunken hybrid computer consisting of a RA 770 precision analog computer and an attached TR-86 stored-program digital computer. This model even took care of intrinsic effects of heavily loaded thyristors and other parts of the inverter system and proved to be so accurate that the conversion of the system was performed successfully mainly relying on the simulation results from this study.

Another interesting analog computer study regarding the behavior of inverters during severe fault scenarios is described by [→Borchardt et al. 1969]. The simulation focused on the development of suitable protection measures for large inverter systems. [→Borchardt et al. 1976] describe an analog computer model of a quadrupole magnet power supply used for ejecting the beam from a synchrotron.

15.10.4 Transmission lines

Long power lines are elaborate structures exhibiting a complex behavior with phenomena such as traveling waves, etc. A summary of typical hybrid computer approaches for modeling power lines can be found in [→Baun 1970]. A study with special emphasis on traveling wave phenomena is described by [→Thomas 1968]. An interesting aspect of simulations like these is that time delay units are required to accurately model the delays encountered in long power line systems. Magnetic tape units and capacitor wheels[219] have been used successfully for this.

15.10.5 Power grid simulation

The study of power grids is not only extremely complex but also highly important due to the dependency of our culture on the ubiquitous availability of electric energy. Since no analytic solutions are possible for complex systems like these, simulations based on analogies are a suitable approach to gain deeper insight into the intricate behavior of such systems. A good overview of the historical development of such simulation techniques can be found in [→Krause 1971], [→Krause 1974], and especially [→Tympas 2003].

Even the development of Pearl Street Station, the first commercial power station in the United States, conceived and built by Thomas Edison, proved to be so complicated that some means for computation was required:[220]

> "*[...] Francis Jehl [...] detailed how Edison thought of constructing a miniaturized network to be used as a computer of the actual network. [... He] remembered that the labor of constructing a miniature network of conductors, 'all in proportion, to show Mr Edison what he would have to install in New York City in connection with the Pearl Street Station' was 'gigantic'.*"

From these humble beginnings, power grids quickly grew in size and complexity, requiring far more sophisticated means for their simulation, yielding the first such special purpose analog computer, a *network analyzer*, conceived in 1925. The purpose of such analyzers is described by [→Nolan 1955, pp. 111 f.] as follows:

> "*Network analyzers have been used to solve quickly the many and various problems concerned with the operation of power systems. They are practical, adjustable miniature*

power systems. They can be used to analyze results during the progress of a system study and therefore play an active part in system planning as well as checking the performance of completed systems."

Early network analyzers were direct analog computers, modeling the power grid by means of discrete and mostly passive computing elements.[221] Typical analyzers were large machines featuring thousands of elements. →Figure 15.60 shows as an example the General Electric AC network simulator installed at the *American Gas and Electric Company* (*AG&E*)[222] in Columbus, Ohio, with Kenneth Raymond Geiser Sr.[223] sitting in front of the central control console.

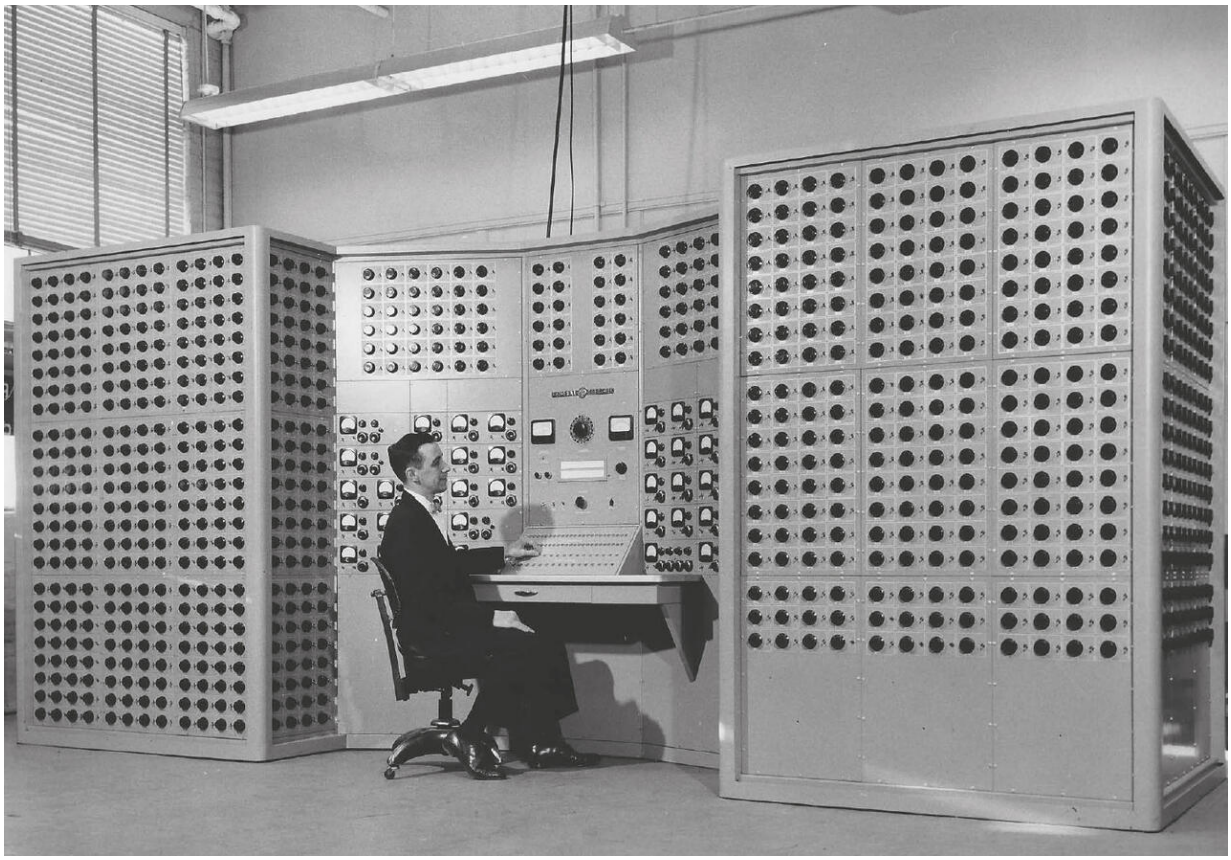


Figure 15.60. The General Electric AC network simulator

Another typical large scale system, the Westinhouse AC power network calculator, installed in Phillips Hall, is shown in →figure 15.61.[224]

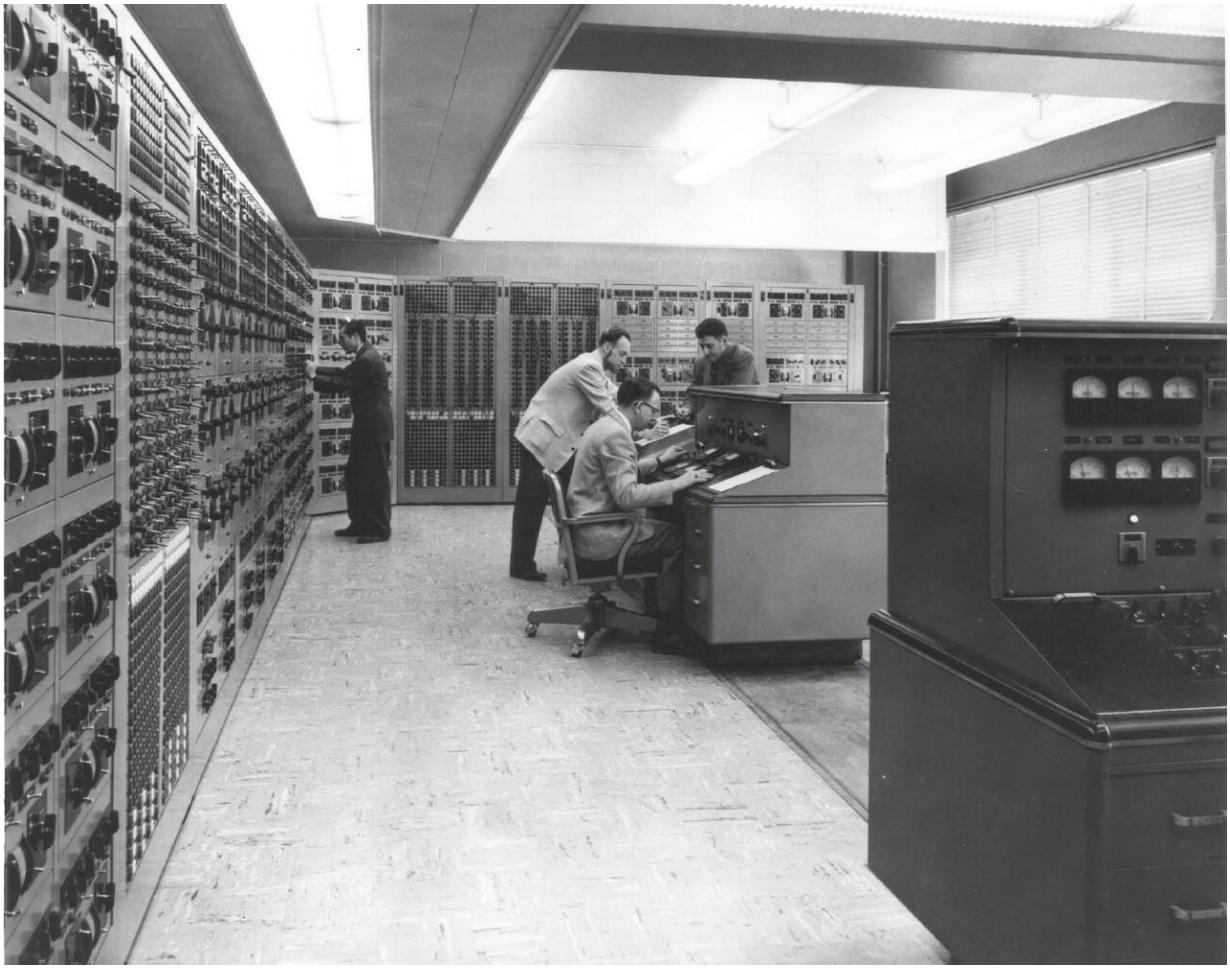


Figure 15.61. The Westinghouse AC power network calculator installed in Phillips Hall, ca. 1955

Later installations made extensive use of indirect analog computers such as the *Beckman EASE 1032* system shown in →figure 15.62, which was used by the *Bonneville Power Administration* for power grid simulations. A typical problem arising in such simulations on indirect analog computers are algebraic loops.[225] Basic techniques for avoiding such loops in power grid simulations are described in [→Giloi et al. 1963, pp. 326 ff.] and [→Giloi 1962].

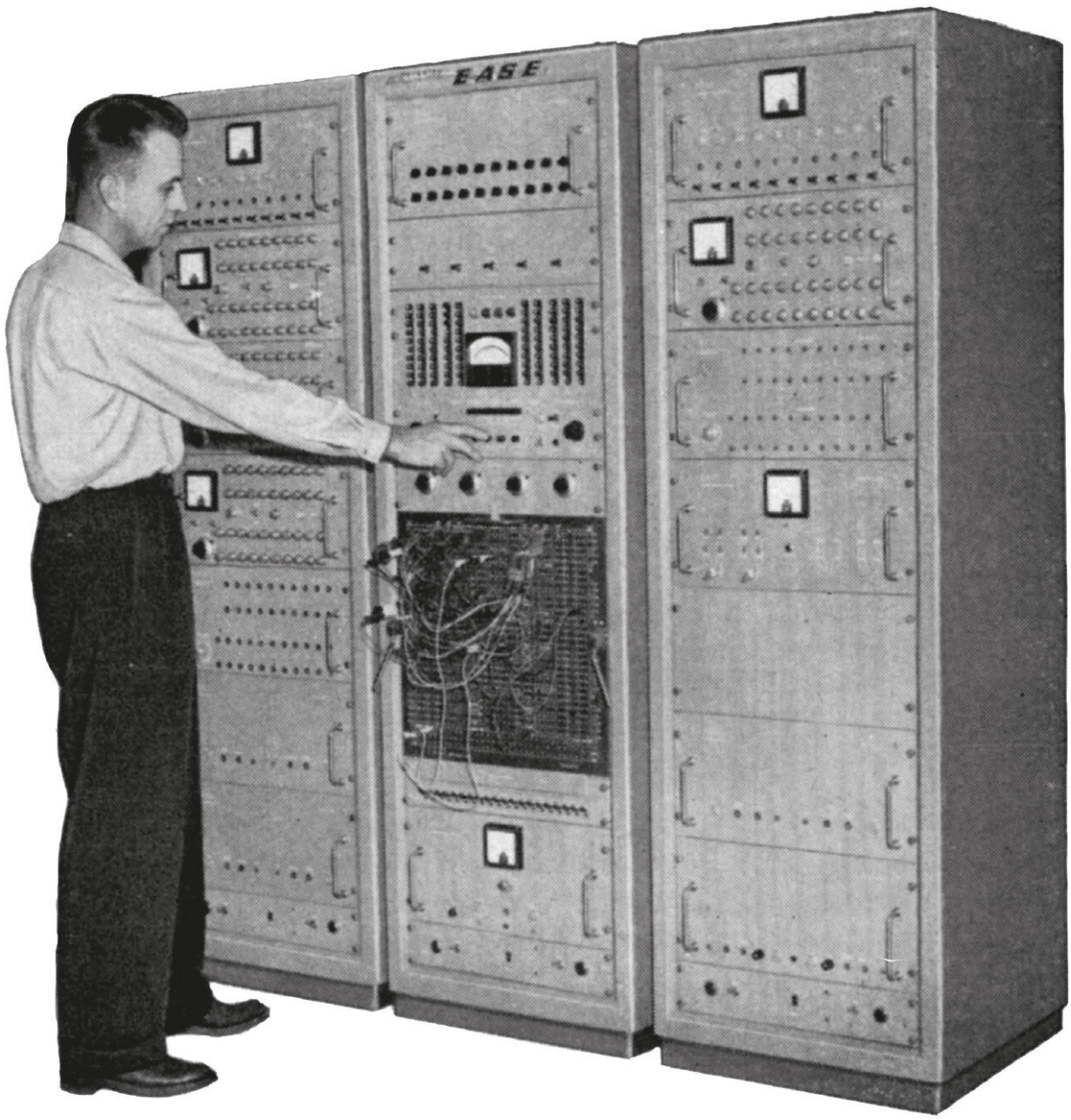


Figure 15.62. EASE 1032 analog computer used at the Bonneville Power Administration for power grid simulations (see [→Instruments and Automation 1957/2])

Another example of a power grid simulator can be found in [→Michaels[a]] and [→Michaels et al.] where a high-voltage distribution system consisting of five power stations and about a dozen transmission lines is simulated. This system was used to train switch room operators.

An even more complex system based on a hybrid computer is described in [→Enns et al. 1971]. This hybrid simulation was able to determine the state of a power grid containing 181 power lines from a given set of initial conditions in 20 seconds, while the same computation using only the digital computer part of the hybrid installation required 30 minutes of run time.

The effect of transients caused by breaking conductors, short-circuits, etc., are studied in [→Thomas et al. 1969]. These simulations also required time delay units, which were implemented with a capacitor-wheel containing 24 capacitors. The simulation of transients caused by lightning strikes in electric power transformation substations is described in detail in [→Hedin et al.].

The effects of capacitive and inductive crosstalk between high-voltage power lines have been studied by simulating a power grid consisting of one 500 kV section and two 250 kV sections on a large analog computer installation. These studies are described in [→Welle 1972] and [→Thomas et al.]. This analog computer contained 180 operational amplifiers, 350 coefficient potentiometers, and 24 time delay units – telltale signs of the complexity of the problem under consideration.

A large power grid simulator, the *Hybrid Computer Power Simulator* (*HCS*), was proposed in 1976 and is described in [→Janac 1976]. This system would have consisted of a special purpose analog computer containing specialized computing elements representing typical analogies found in such simulations. Instead of a traditional patch panel, an automatic patch panel, an *autopatch*, was proposed. Unfortunately, this machine was never built and autopatch systems did not prove commercially viable due to the extremely large number of switching elements required back then.[226]

15.10.6 Frequency control

While power grids are complicated dynamic systems already, things get even more complicated when multiple power grids are coupled together, as [→Paynter 1955/2, p. 229] notes:

> *"[...] the [...] problem is fundamentally complex due to the multiplicity of generally disparate machines and regulators coupled together by a large number of 'elastic' links."*

Early work on the analysis of coupled power grids using direct as well as indirect analog computers was begun in 1947 at MIT. [→KRAUSE et al. 1977] describe a study of shaft distortion in generators caused by misaligned grid frequencies when disjunct grids are coupled. Special topics regarding the synchronization and control of such coupled power grids are treated in [→SYDOW 1964, pp. 214 ff.].

15.10.7 Power station simulation

[→WHITESELL et al. 1969] describe an analog computer based training system for power station operators. It modeled the "*two separate, but tightly interconnected, steam and electric power generating stations at the Whiting refinery*". Learning how to control such a complex system is a daunting task and learning-by-doing is not an option in the demanding environment of a refinery. The impressive analog computer setup for this simulator is shown in →figure 15.63.

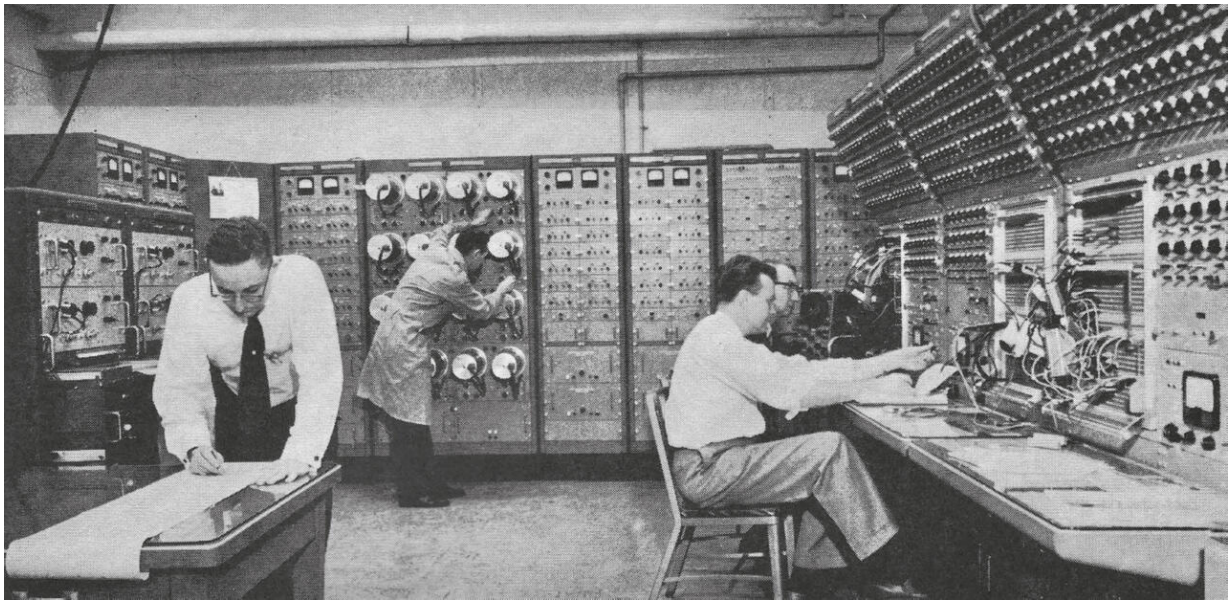


Figure 15.63. Power station simulator (cf. [→WHITESELL et al. 1969, p. 85])

The simulation of a hydroelectric power station on a hybrid computer is described in [→Kausel et al. 1985]. The study focuses on a comparison of this hybrid implementation with a purely digital model. One disadvantage of the analog computer approach was the number of time delays required for the simulation. However, the hybrid model was found to be more realistic than the digital simulation it was compared to. A short description of hybrid computer based simulations of turbines, etc., can be found in [→Minne 1986].

15.10.8 Dispatch computers

There are a lot of optimization problems ranging from scheduling questions to the supply of combustibles, etc., to be solved for running power systems efficiently. As early as in 1953, *Edwin L. Harder* submitted a patent for an *economic dispatch computer*, which solved a complex system of coupled equations in order to determine optimum dispatch strategies for power generating stations given their characteristics, location in the network, fuel prices, etc.[227] Although not marketed, this system was followed by later developments of *Electronic Dispatch Computers* (*EDC*), aimed at optimization tasks.[228] The savings made possible by the application of analog computing techniques were remarkable and justified the development of large analog computer installations:[229]

> "*Estimated annual fuel savings obtainable with EDCs may approach US$50 per megawatt installed. For a typical 1,000-megawatt system this saving can amount to US$30,000 to US$50,000 per year and thus may warrant an investment of US$250,000.*"

→Figure 15.64 shows an example of an electronic dispatch computer, the *Goodyear Electronic Differential Analyzer* (*GEDA*). This power dispatch computer saved US$ 200,000 annually (see [→Instruments and Automation 1957/3]).

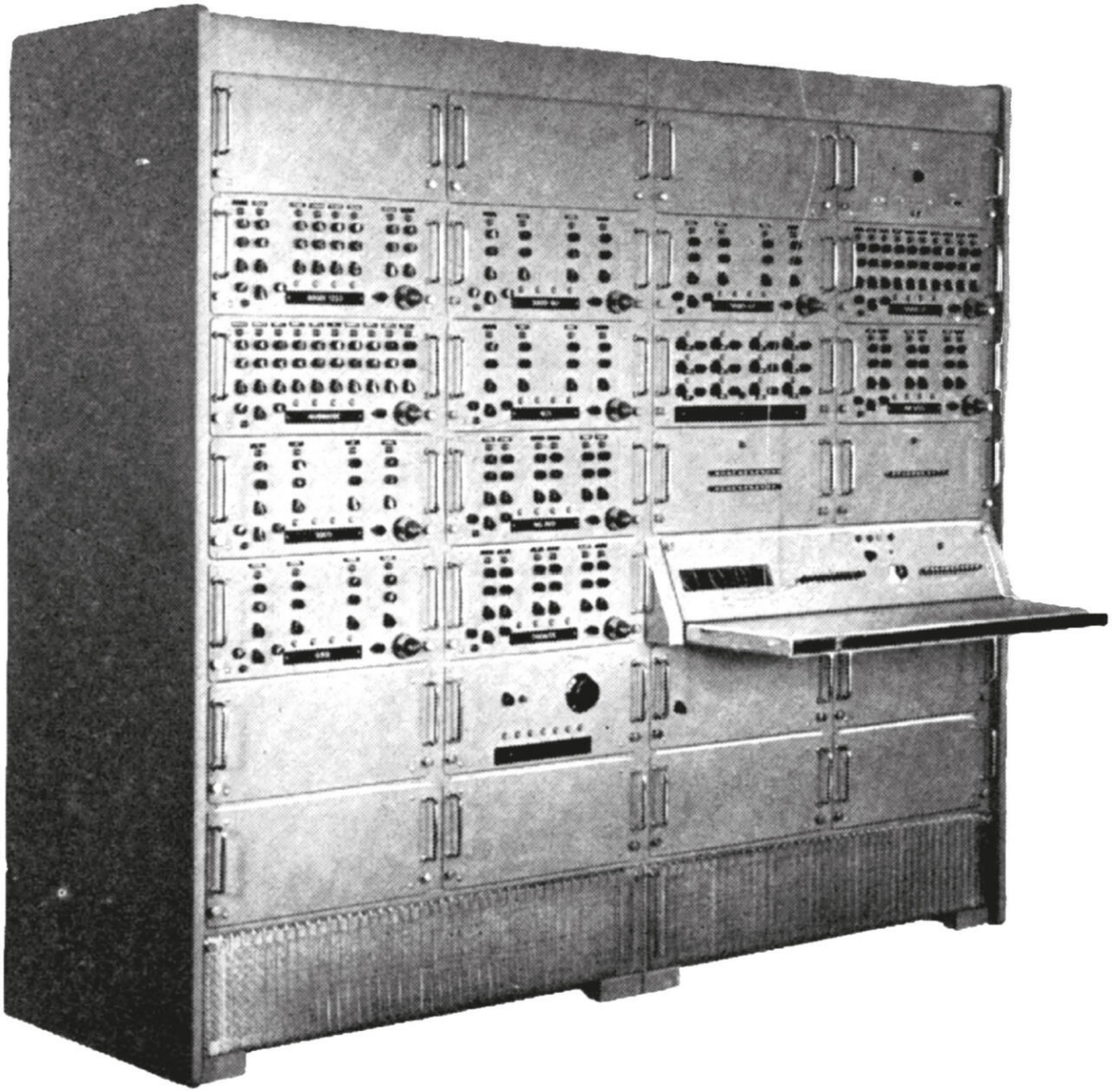


Figure 15.64. GEDA power dispatch computer (see [→Instruments and Automation 1957/3])

A not-too-complex optimization example regarding a thermal plant is described in [→Heinhold et al. 1969, pp. 224 ff.]. An analog computer study aimed at the computation of optimal schedules for a hydrothermal power plant is described by [→Perera 1969].[230]

An interesting hybrid system built by *Leeds & Northrup* and installed at the Detroit Edison Company is mentioned in [→Computers and

Automation 1964/2]. This particular system regulated power from 37 generating units supplying well over one million customers in real time.

15.11 Electronics and telecommunications

The rapid development of electronics and especially telecommunication technology since World War II resulted in an ever-increasing demand for component and circuit simulations. As [→ Kettel et al. 1967, p. 3][231] put it, the more complex electronic circuits get, the more important it is to gain insight in their respective behavior by mathematical models and analogies instead of building breadboard prototypes. The following sections give some examples of such work.

15.11.1 Circuit simulation

An interesting analysis of the behavior of a transistorized circuit subjected to short transients is described by [→ Ranfft et al. 1977]. This simulation makes extensive use of the ability of an analog computer to perform temporal expansions since the effects of interest, which are analyzed by the simulation, take only a few nanoseconds (10^{-9} s). This study strikingly shows the remarkable complexity of such circuit simulations. As [→ Ranfft et al. 1977, p. 76] note

> "*[s]tandard analog computers are normally not well suited for simulation of transistor circuits.*"

[→ Balaban[a], p. 771] remarks that

> "*a six to eight transistor circuit can be patched on a large analog computer.*"

Roland Ranfft and Hans-Martin Rein set out to develop a special-purpose analog computer aimed at the simulation of electronic circuits:[232]

> "*Its advantages, when compared to the usual digital computer simulation, are low costs, simple operation, and short simulation time, allowing a fast man-machine dialog. This simulation method has already been successfully used in the design of several high-speed integrated circuits.*"

The main advantage of this system was not only its high simulation speed but also the high degree of interactivity made possible by the analog computer. The projected cost for an implementation containing 20 transistor models, 12 SCHOTTKY[233] diodes, 24 resistors, and two current/voltage sources was only US$ 3,600. The same amount of money would have had to be spent to perform 100 to 200 transient analyses on a rented stored-program digital computer.

Another simulator, *HYPAC*, short for *Hybrid PACTOLUS*,[234] is described by [→ BALABAN[b]]. This system was basically a hybrid computer with a special setup on the analog part simulating basic electronic circuit elements. Using the digital computer of the hybrid computer setup, the analog computer's elements were time-shared between the various circuit elements of the circuit to be simulated. This made it possible to simulate circuits of nearly unlimited complexity given enough time.

An extensive simulation study of mixer circuits using a Telefunken RA 463/2 analog computer[235] can be found in [→ SIERCK 1963]. This study was unique at its time as it was the first all-embracing simulation of such circuits and showed that the results achieved by means of analog computers were meaningful and could be used for actual circuit design.

[→ BECK et al. 1970] describe a hybrid computer approach to circuit design with a number of practical examples.

15.11.2 Frequency response

Before spectrum analyzers were introduced as independent devices into laboratories, determining the frequency response of electronic circuits was a laborious task that could be simplified considerably by an analog computer. Depending on the complexity of the circuit to be analyzed and the frequency range of interest, the frequency/phase response of a prototype circuit could be measured and processed directly on an analog computer. It is also possible to simulate the device on the same analog or hybrid computer performing the processing of the data obtained. The advantage of the latter approach is the possibility to change the time-scaling, which is not possible when working with an actual prototype circuit.

Typical analog computer setups for performing a spectral analysis are described in [→ BARD 1965, pp. 212 f.] and [→ GILOI et al. 1963]. These

analog computer programs are often based on a maximum detector circuit,[236] which is basically a delay circuit. Using a sweep generator as shown in →section 8.2 the frequency response of a circuit, be it real or simulated, can then be determined.[237] Another example for such a technique can be found in [→Schüssler 1961] where a sampling process is used to generate a frequency response plot.

15.11.3 Filter design

The design of filter circuits is not too different from the problems described above. An example of such a simulation can be found in [→Larrowe 1966], which describes the design of a quadrature band-pass filter.[238] [→Wadel 1956] describes the simulation of digital [sic] filters on an analog computer.

A very complex example of filter design is given in [→Giloi 1961]. This study describes the development of a Wiener filter on an analog computer, which requires the solution of integral equations.[239] The technique used in this paper is to have the machine time running backwards, which was done by using an analog tape drive on which values were prerecorded in a preliminary step and played back in reverse direction for the actual determination of the filter parameters. A similar technique used to determine solutions of optimum control problems is described in [→Anderson et al. 1967].

15.11.4 Modulators and demodulators

Modulator and demodulator circuits were also developed by means of analog computer simulations. The design of phase detectors and frequency discriminators is described in [→Manske 1968]. In this study the circuit under development as well as the processing of measurement data has been implemented on an analog computer. The analog simulation of a frequency modulation based communication system is described in [→Hu 1972]. [→Kettel 1960, pp. 170 f.] describes the study of modulators and demodulators for telemetry applications on an analog computer.

A complex and comprehensive study of frequency multipliers using nonlinear iron core inductors can be found in [→Nafalski 1976]. An

interesting aspect of this study is the treatment of the behavior of such devices during transients such as power up.

15.11.5 Antenna and radar systems

Around 1950 J. A. Hammer developed an analog computer for the simulation and study of radar[240] antenna systems at the *Nederlands Radar Proefstation*, which is described in detail in [→ Hammer 1956], [→ Pronk 2019], and [→ Goldbohm 1993]. Radar antennas pose some special problems such as *sidelobes*,[241] which must be minimized to obtain a beam as narrow as possible. The antennas under consideration here were slot antennas with up to 41 slots. Calculating the resulting radiation pattern is a formidable task, which justified the construction of a special purpose analog computer. The *Hammer computer* as it became known used electronic as well as electromechanical computing elements effectively implementing a complex series analyzer. Computing the E-field of an antenna took 12 seconds on this machine while contemporary digital computers required up to 700 seconds for the same task.

[→ Mitchell 1960] describes an analog computer for the simulation of the directivity characteristics of complex antenna arrays. The electromechanical analog computer built for this purpose was capable of simulating arrays with 50 to 200 aerials.

[→ Bickel et al. 1957] describe a complex radar simulation system taking into account a variety of adverse effects such as *target scintillation*,[242] various fading effects, and receiver noise. Even more complex is the simulation of the radiation pattern of a rotating antenna operating in an inhomogeneous plasma described in [→ Fritz 1967]. This problem arises in the development of warhead re-entry bodies. The medium surrounding the antenna is simulated by a mesh of passive RLC networks.

15.12 Automation

There are many applications for analog and hybrid computers in the fields of industrial process measurement and control technologies. Obviously, the development of new processes and new control systems

can profit from analog computer studies. Analog computers were sometimes even used as integral parts of complex control systems as [→Kettel 1960, p. 165] noted:

> *"Not only can an analog computer serve as a computer to simulate control systems, but can also be part of a control system itself, especially if it is a small and portable system. As such it allows the optimization of the control system in place."*[243]

15.12.1 Data processing

An example for such an embedded analog computer is the *TRS-2* data acquisition and processing system shown in →figure 15.65. This system was developed and manufactured by *Rohde & Schwarz* and contained standard analog computer components manufactured by EAI for their TR-10 and TR-20 tabletop analog computers. The system could be equipped with standard analog computer modules as well as with user specific modules implementing programs for special applications.[244] Another interesting electromechanic computing element capable of processing up to five input values is described in [→ZSMR 1963]. At its heart is a beam balance driven by two moving coil actuators.

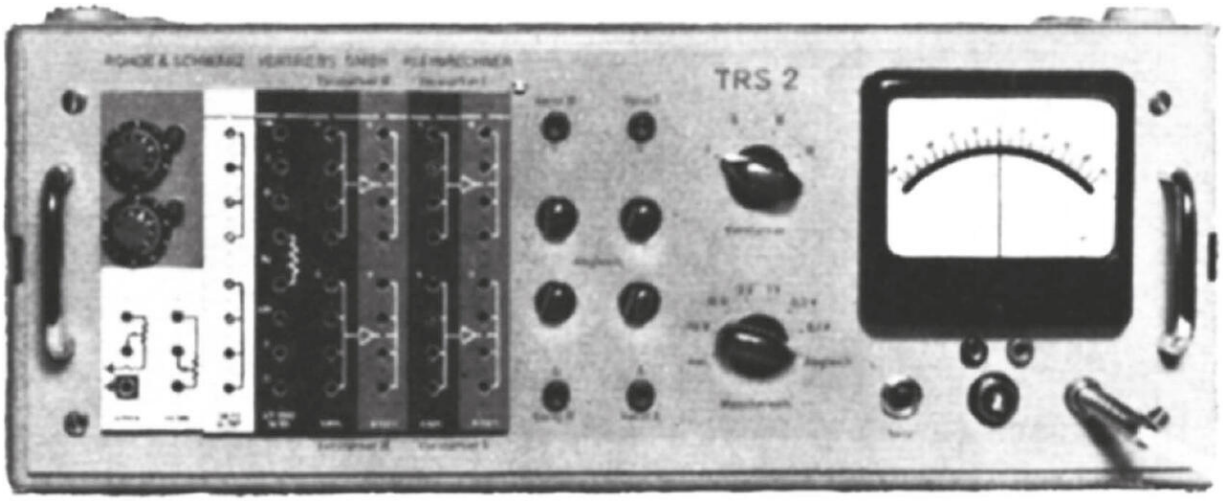


Figure 15.65. Data capturing and preprocessing system TRS-2 (Rohde & Schwarz, see [→Elektronische Rechenanlagen 1961])

15.12.2 Correlation analysis

[→Wierwille et al. 1968] describe the use of a hybrid computer for performing autocorrelation analyses on vibration data gathered during rocket tests. A fully digital implementation is also described and both methods are compared.

15.12.3 Closed loop control and servo systems

The earliest example of an analog computer used to analyze and demonstrate closed loop control systems is George A. Philbrick's Polyphemus, described in →section 3.2. Another early analog process simulation system, the *Electro-Analogue*, is described by [→Janssen et al. 1955]. This process and control simulator consisted of three largely independent subsystems: A model of the control circuit, a model of the process being controlled, and an output system containing an oscilloscope display, some panel meters, and recorders. An interesting aspect is the implementation of time delays, which are often required in control system simulations to model transport delays, etc. These time delay units were implemented as a string of capacitors and inductors with switch-selectable taps representing various delay times.

Further examples can be found in [→James et al. 1971, pp. 215 ff.] and [→Worley 1962], covering bang-bang servos, proportional controllers, etc. A comprehensive treatment of the simulation and development of governors is given in [→Ammon et al. 1959]. [→Lotz 1969, pp. 211 ff.] describes the simulation of a control system for a hydraulic system. [→Sydow 1964] describes a complex control system study for a paper machine. The emphasis of this study is on the control of 40 steam heated cylinders, which must run at precisely controlled speeds to avoid tearing the paper running through the machine. [→Koenig et al. 1955] describe the treatment of flyweight governors used in a hydroelectric power station by means of an analog computer. A nice and simple simulation of a liquid level control system as used in the chemical industry, using an analog computer, is described in [→Worly 1956].

The analysis of damped, nonlinear servo systems using an analog computer is described in detail by [→Caldwell et al. 1955]. Nonlinear servo systems are of great importance for practical applications but are

notoriously complex to model and analyze. Additional information on such systems can be found in [→Hurst[a]]. The determination of parameters for a nonlinear control system using an EAI 231R analog computer[245] is covered by [→Suryanarayanan et al. 1968].

15.12.4 Sampling systems

Many control systems are, in fact, sampling systems where input signals are sampled at discrete time intervals. These values are then used to control an underlying process. The simulation and analysis of these sampled data systems is difficult to perform on an analog computer since the sampling process has to be implemented by trick circuits such as bucket-brigade circuits consisting of chained integrators under individual control. Typically, hybrid computers were used to model sampling systems since their digital part is well suited to implement the sampling process, while the underlying process can be modeled on the analog system. Typical examples for this can be found in [→Schneider 1960] and [→Simons et al.].

15.12.5 Embedded systems

Analog computers were used early on as embedded components of complex control systems, as described in [→Korn et al. 1956, pp. 109 f.]:

> "*D-c analog representation of automatic control systems has proved to be a powerful aid in the design of prototype models and pilot plants and in the determination of starting procedures and of optimum controller settings after changes in raw materials or other conditions. But the utility of d-c analog techniques for automatic control applications is not restricted to computations of this type. Special-purpose d-c analog computers, which may often be conveniently assembled from the standard components of commercially available machines, can themselves serve as control-system elements in many applications suited to their characteristics.*"

An example for such modules has already been shown in →figure 15.44 in →section 15.6.4. Other examples include the computing elements developed by Telefunken for the RA 800 analog computer,[246] which were also sold as building blocks for industrial applications.[247]

The basics of analog process computers are covered in [→Ludwig et al. 1974], while [→Weitner 1955] describes the actual implementation of a

closed loop model. The design and implementation of a special purpose amplifier with a characteristic of $\sqrt[3]{x}$ is described in [→ KINZEL et al. 1962/1] and [→ KINZEL et al. 1962/2]. This amplifier was actually used to control injection parameters in large diesel engines.

15.13 Chemical process engineering

Analog computers were invaluable tools for process engineering applications and consequently most chemical companies set up dedicated analog computing centers to aid their development departments. The importance of analog computers in this area can be seen by the following quotation from [→ HOLST 1982, pp. 316 f.]:

> "*Analog computing capacity [at Foxboro] has increased from some 8-10 operational amplifiers in 1938 to more than 150 thirty years later.*"

→ Table 15.1 gives an overview of the analog computing capacity installed at some major chemical companies in 1961.[248]

Table 15.1 Analog computer installations for research in petrochemistry in 1961 (see [→ Carlson et al. 1967, p. 356])

Company	Year	Number of amplifiers	Manufacturer
Dow Chemical Co.			
Midland Division	1954	20	Beckman (Berkeley)
	1961	140	EAI
Texas Division	1956	30	Daystrom (Heath)
	1961	80	Philbrick
E. I. du Pont de Nemours & Co.			
Newark, Del.	1950	30	Beckman (Berkeley)
	1955	50	Beckman (Berkeley)
	1958	120	EAI
	1960	300	EAI
Experimental Station, Wilmington, Del.	1960	70	Computer Systems
Monsanto Chemical Co.			
St. Louis, Mo.	1957	116	EAI
	1958	24	EAI
	1959	88	EAI
Ohio Oil Co.			
Denver, Colorado	1957	56	EAI
Humble Oil & Refining Co.			
Baytown, Texas	1960	80	EAI
	1961	80	EAI
Baton Rouge, La.	1959	80	EAI
	1960	40	EAI
Esso Research & Engineering Co.			
Florham Park, N.J.	1959	40	EAI
	1959	40	EAI
	1960	80	EAI
	1960	80	EAI
American Oil Co.			
Whiting, Ind.	1955	?	EAI

Company	Year	Number of amplifiers	Manufacturer
	1957	168	EAI
Standard Oil Co.			
Cleveland, Ohio	1955	90	Beckman (Berkeley)
	1957	10	Beckman (Berkeley)
	1961	170	Beckman (Berkeley)
Union Carbide Olefins Co.			
South Charleston	1956	30	EAI
	1958	60	EAI
	1959	60	EAI
Thiokol Chemical Corp.			
Brigham City, Utah	1959	168	EAI
Phillips Petroleum Co.			
Bartlesville, Okla.	1959	80	EAI
	1960	80	EAI
Chemstrand Corp.			
Decatur, Ala.	1960	80	EAI
Shell Oil Co.			
Shell Chemical Corp.	1960	120	EAI
Development Corp.	1956	24	Goodyear
	1957	24	Goodyear
	1960	10	Donner Scientific
	1960	10	Donner Scientific
Hercules Powder Co.			
Wilmington, Del.	1960	44	Beckman (Berkeley)
Daystrom, Inc.			
La Jolla, Calif.	1960	100	Computer Systems

A wealth of information and examples regarding the use of analog computers in process engineering applications can be found in [→Wagner 1972], [→Worley 1962], [→Holst 1982], and [→Ramirez 1976]. A small, yet powerful, process simulation system based on an EAI *TR-48*

analog computer[249] was introduced at the *Systems Engineering Conference* in 1964 by EAI.

A highly complex simulation of a solvent recovery process is described by [→ Lewis 1958]. It requires 120 summers/integrators, 120 limiters, 48 summers, 28 servo multipliers, 16 photoformers, 300 coefficient potentiometers, and 186 external precision capacitors.[250] Another example for a complex process engineering simulation can be found in [→ Graefe et al. 1974] where a copper melting process is modeled, even taking the discontinuous flow of materials into account.

15.13.1 Mixing tanks, heat exchangers, evaporators, and distillation columns

A common and basic element in every chemical plant is the mixing tank, which can be simulated quite straightforwardly on an analog computer as described in [→ Ramirez 1976].

The simulation of heat exchangers and evaporators is more complex since these are described by partial differential equations.[251] The simulation of an evaporator on an EAI 590 hybrid computer system is described by [→ Oliver et al. 1974], while a more complex example involving a heat exchanger reactor can be found in [→ Carlson 1968].[252] [→ Chareton et al. 1967] describe a generic approach for the simulation of heat exchangers and a special purpose analog computer module which can be used as a subsystem in more complex simulation setups.

The very complex simulation of a Fischer-Tropsch[253] reactor for coal liquefaction is treated in [→ Govindarao 1975]. The main challenge in this type of simulation is the fact that gaseous, liquid, and solid phases of the reactants coexist simultaneously in the reactor system.[254] This particular simulation required 33 integrators, twelve sample and hold elements, 13 summers, ten inverters, 18 multipliers, 80 coefficient potentiometers, and a digital control system.

Even more complex are simulations of fractionating columns containing many column trays. A simulation involving three trays is described in [→ N. N. 1958/2], while [→ Williams et al. 1958] describe a more complex system with five trays.[255] The complexity of such simulations is caused by the complex mixing processes and phase

transitions taking place at the column trays. A six tray fractionating column simulation is described in [→Ramirez 1976, pp.28 ff.].[256]

15.13.2 Control processes and adaptive control

The implementation and analysis of an adaptive control system[257] is described in [→Powell[a]]. The system covered in this study operates in two timescales: A short timescale subsystem implementing a predictor and a slow running subsystem implementing the actual control loop using the values generated by the predictor as its inputs.

[→Hill et al. 1964] outline an interesting *learning control system*, which is distinguished from a more classic adaptive system by the fact that it contains memory and logic decision capabilities. The system described was implemented on a hybrid computer consisting of a GEDA analog and an *IBM 1620* digital computer with associated ADCs and DACs. The authors show that their approach surpasses an adaptive control system by subjecting the learning control system to pseudo-random noise and changing the environmental conditions quickly enough that "*purely adaptvie action (i. e. hill-climbing on the IP surface) could not optimize the system during the period of constant environment.*"[258]

A comprehensive description of project *CHANCE* (*Calculateur Hybride, Analogique et Numérique pour la Commande et l'Etude des processus industriels automatisés*) can be found in [→Peretz 1967]. The paper details a large number of application examples ranging from the mathematical modeling of boiler and turbine systems to complex control systems.

15.13.3 Parameter determination and optimization

Another important application area for analog computers was the determination and optimization of process control parameters. The complexity of such control problems often exceeds that of other optimization tasks and typically uses hybrid computers to implement gradient methods. Examples of these techniques and applications can be found in a variety of sources such as [→Kopacek et al. 1976], [→Kopacek et al. 1979], [→Kopacek 1983], [→Troch 1977], [→Woźniakowski 1977],

[→Gröbner 1961], [→Kramer 1968], [→Kramer[a]], and finally [→Jamshidi 1976].

[→Piceni et al. 1975] describe the evaluation of operating parameters for industrial processes in general. An interesting aspect of this study is the comparison of a priori and a posteriori values to determine a measure for the quality of the underlying model. Optimal process control by analog computer studies is the focus of [→Miura et al. 1967/1].[259] An interesting optimization technique restricting its parameters to binary values (0 and 1) is described in [→O'Grady 1967].

A very interesting optimization scheme based on random generators[260] is developed in [→Korn et al. 1970]. The basic idea is to perform random parameter variations and test their respective performance. If one of these variations results in a better overall performance, this parameter set is used as the base for the next variation cycle, otherwise the old parameters are restored.[261] Interestingly, it can be shown[262] that parameters determined by such a stochastic process are often superior to parameters that are the result of a deterministic process. [→Fricke et al. 1964] describe the use of a portable analog computer for process identification and control. The basic idea is to model the process on the analog computer and the perturb it with an impulse or a cosine signal. The process' input is then fed to an orthogonal model, the output of which is compared to the output of the process by means of an integrator and a squaring circuit.

The application of an analog computer to derive *operating guides* from process variables in an oil refinery is described in [→Taylor 1957]. The systems takes 101 process variables, such as gas and liquid flow rates, pressures, tank levels, oxygen percentage, etc., into account and derives high-level operating guide values like the carbon burning rate, catalyst-circulation rates and more. An interesting characteristic of the system used is that variables are not represented by DC voltages but instead by 60 Hz AC signals so that AC coupled computing elements can be used throughout the machine, thus eliminating drift problems.

15.13.4 Plant startup simulation

The startup of complex chemical plants is a very complex and error prone process. Consequently, analog computers were used to simulate

startup processes and to train operators. →Figure 15.66 shows an analog computer coupled with typical plant monitoring and control devices. This setup, described in detail in [→Lieber et al. 1969], allowed the study of complex startup processes in oil refineries and the training of operator personnel.

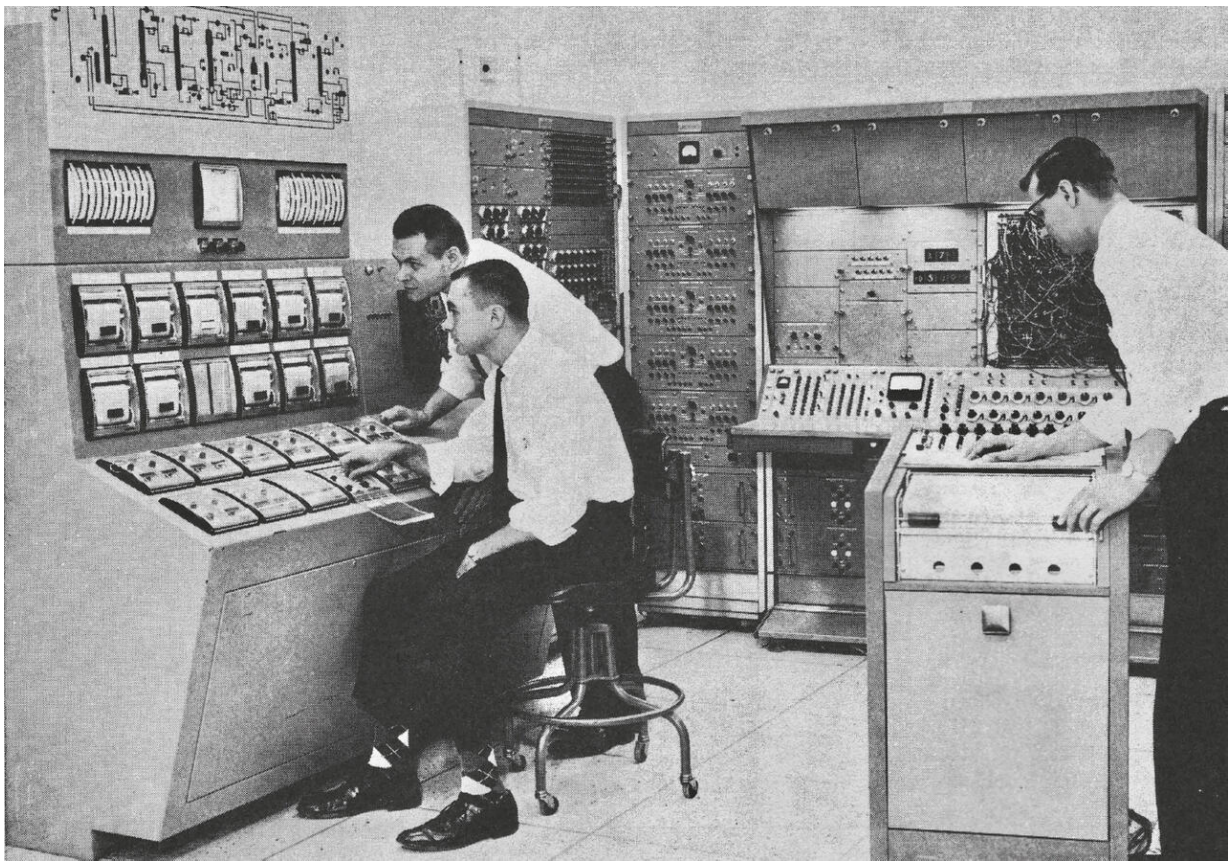


Figure 15.66. Process plant simulator ([→Lieber et al. 1969, p. 78])

15.14 Transport systems

Transport systems of all sorts can benefit from the application of analog and hybrid computers as the following sections demonstrate.

15.14.1 Automotive engineering

Based on the *Electrical Transients Analyzer*, built in 1936 at Westinghouse, GILBERT D. MACCANN, HARRY E. CRINER, and C. E. WARREN developed the *Mechanical Transients Analyzer* in 1945. This system basically extended the Electrical Transients Analyzer by an electronic model of mechanical systems and was used for all kinds of mechanical simulations involving transient perturbations, such as a car moving over a bumpy road.[263] →Figure 15.67 shows the application of this system to the development of a car's suspension system. In the upper half, the car and its basic mechanical equivalent are shown. The two sub-figures on the bottom show equivalent electronic circuits implemented on the analyzer for this study.

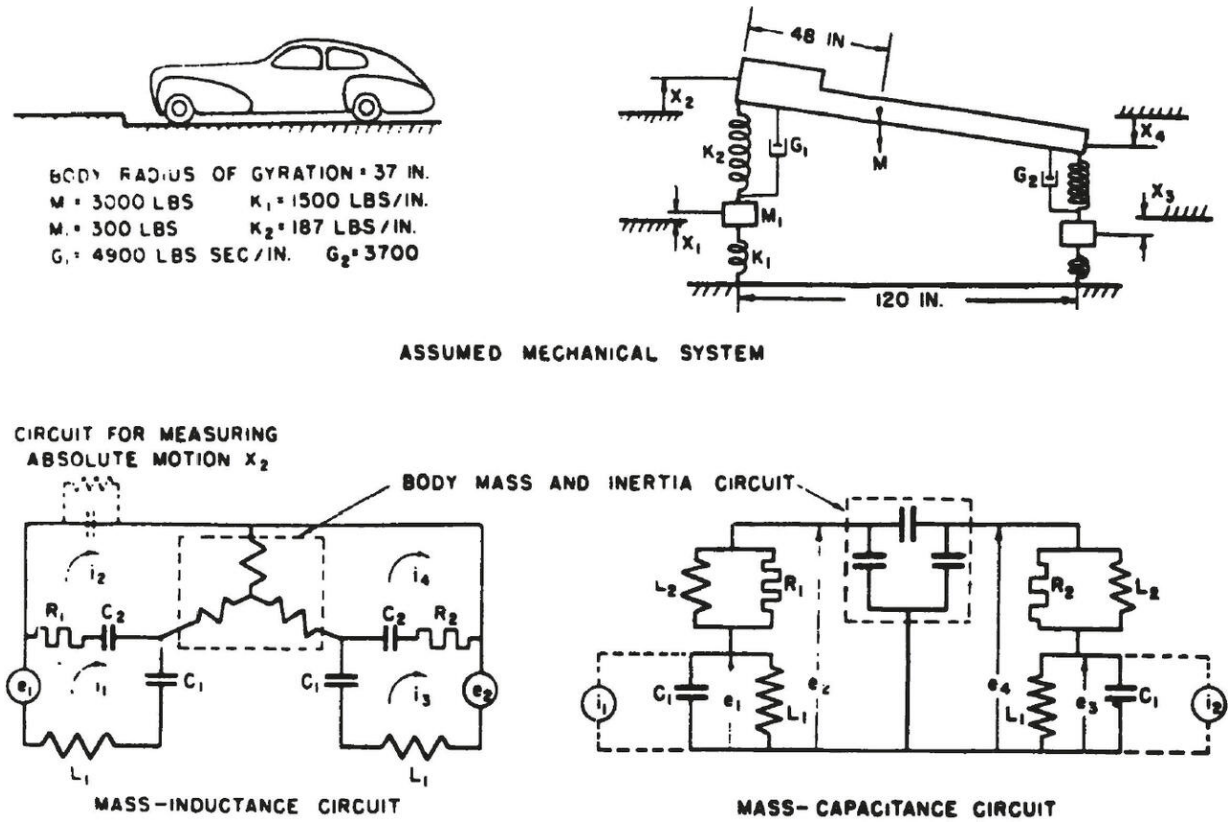


Figure 15.67. Application of the Mechanical Transients Analyzer to a suspension problem (see [→ASPRAY 1993, p. 42])

Beginning with these early developments, analog computers were a crucial part of the engineering departments of all major vehicle

manufacturers well into the 1980s. A simplified basic application was described in →section 8.7, where the dynamic behavior of a vehicle, modeled as a two-mass system, was simulated. A more realistic, but still simplified, three-mass simulation is described in [→Sydow 1964, pp. 245 ff.], while [→McLeod et al. 1958/6, p. 1992] cover the specific problems of the simulation of shock absorbers for heavy lorries and excavators.[264] [→Gumpert 1972] describes an analog computer simulation of self-excited vibrations in cars.[265]

[→Dornier/7] describes the application of an analog computer for data processing in a runout inspection. The ability to perform real-time spectral analysis using the analog computer is of particular importance in this type of application.

15.14.1.1 Steering systems

Steering systems are a central part of every car and analog computers were often used to simulate steering systems, giving the freedom to vary and optimize parameters prior to building actual prototype systems. A special purpose analog steering simulator developed by General Motors is shown in →figure 15.68. It allowed to analyze the dynamic behavior of a steering system with respect to yaw, roll, and slip moments. The importance of this system is described by [→McLeod et al. 1958/6, p. 1995] as follows:

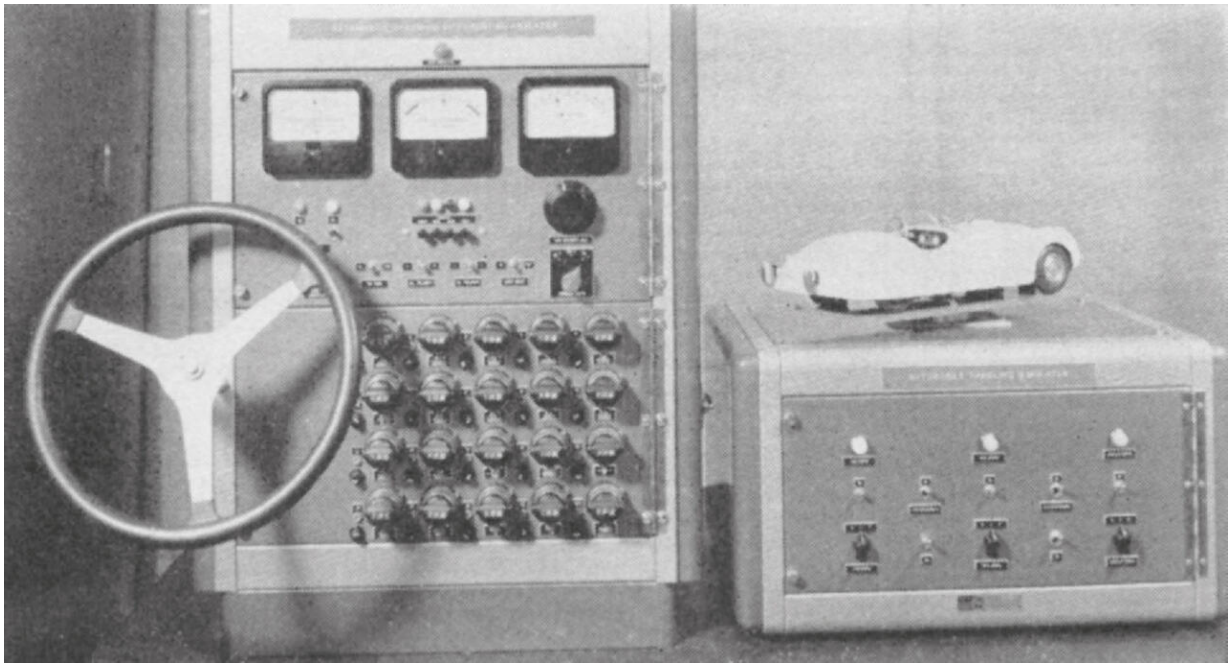


Figure 15.68. Steering simulator (see [→ McLeod et al. 1958/6, p. 1995])

> "*As a result of this work, an understanding has been gained of the effects of the various car parameters on the car's lateral response. In addition, a number of general observations have been made: Yaw and sideslip are strongly coupled, but there is only a weak coupling linking the roll to the yaw and sideslip. However, this weak coupling is often the reason that a particular automobile is stable or unstable, particularly in yaw.*"

[→ Zimdahl 1965] describes an automatic control system enabling cars to follow a buried cable by means of an inductive pickup.

15.14.1.2 Noise control

An extremely intricate and complex hybrid simulation study of the elastic mounting of an internal combustion engine within a car frame is described in [→ Filipsky et al. 1977]. Using a digital computer, eigenvalues and eigenvectors are computed, while the analog computer solves the coupled differential equations describing the three-dimensional oscillations of the engine block itself in six degrees of freedom. These values are then used by the digital computer to compute the overall forces acting on the motor mountings with the aim of determining optimum mount points for a production vehicle.

15.14.1.3 Transmissions

An interesting example of the application of analog computers in automotive engineering, namely the simulation of a four-speed automatic transmission, is described in [→Dornier/1]. This example is especially noteworthy due to the high number of comparators required to model the discrete gearshift points.[266]

The simplified motor driven car underlying this simulation is described by

$$m\ddot{x} = F_{\text{drive}} - F_{\text{drag}} - F_{\text{rolling drag}} - F_{\text{road slope}}.$$

To simplify matters further, $F_{\text{rolling drag}}$ and $F_{\text{road slope}}$ are assumed to be zero, yielding

$$m\ddot{x} - F_{\text{drive}} + F_{\text{drag}} = 0$$

or, in more detail

$$m\ddot{x} - \frac{\eta i J_k}{r} M\left(n\right) + CC_w Av^2 = 0,$$

with m = 1120 representing the car's mass, η = 0.9 the efficiency factor, i = 4.11 the axle ratio, r the wheel radius, J_k the four gear transmission ratios, $M(n) = 5.306 + 50n - 81.25n^2$ the rotation speed dependent moment of force, and Cc_wAv^2 the drag, approximated by $0.003421v^2$. The gear transmission ratios are defined as J_1 = 3.8346, J_2 = 2.0526, J_3 = 1.345, and J_4 = 1. The transmission performs an upshift every time the rotational speed of the motor reaches n = 6000 min^{-1} corresponding to the following velocities: v_1 = 41.4 km/h, v_2 = 78.8 km/h, and v_3 = 118 km/h.

Based on equation (→15.9) the program shown in →figure 15.69 can be derived. A typical simulation output is shown in →figure 15.70.

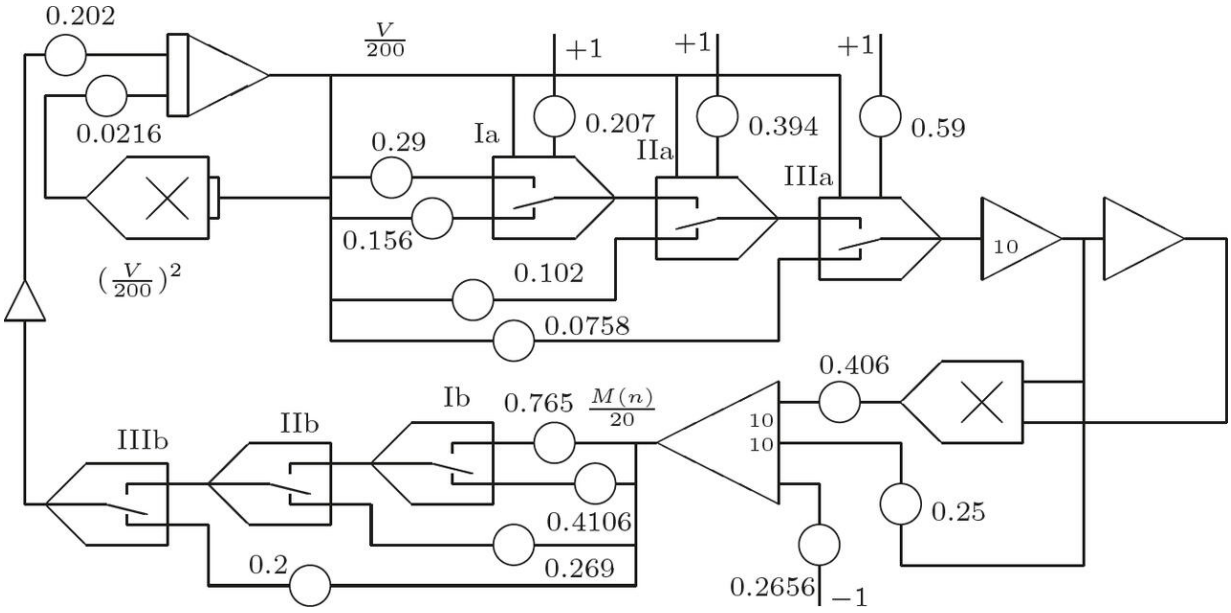


Figure 15.69. Program for the simulation of a four-speed automatic transmission (see [→Dornier/1, p. 5])

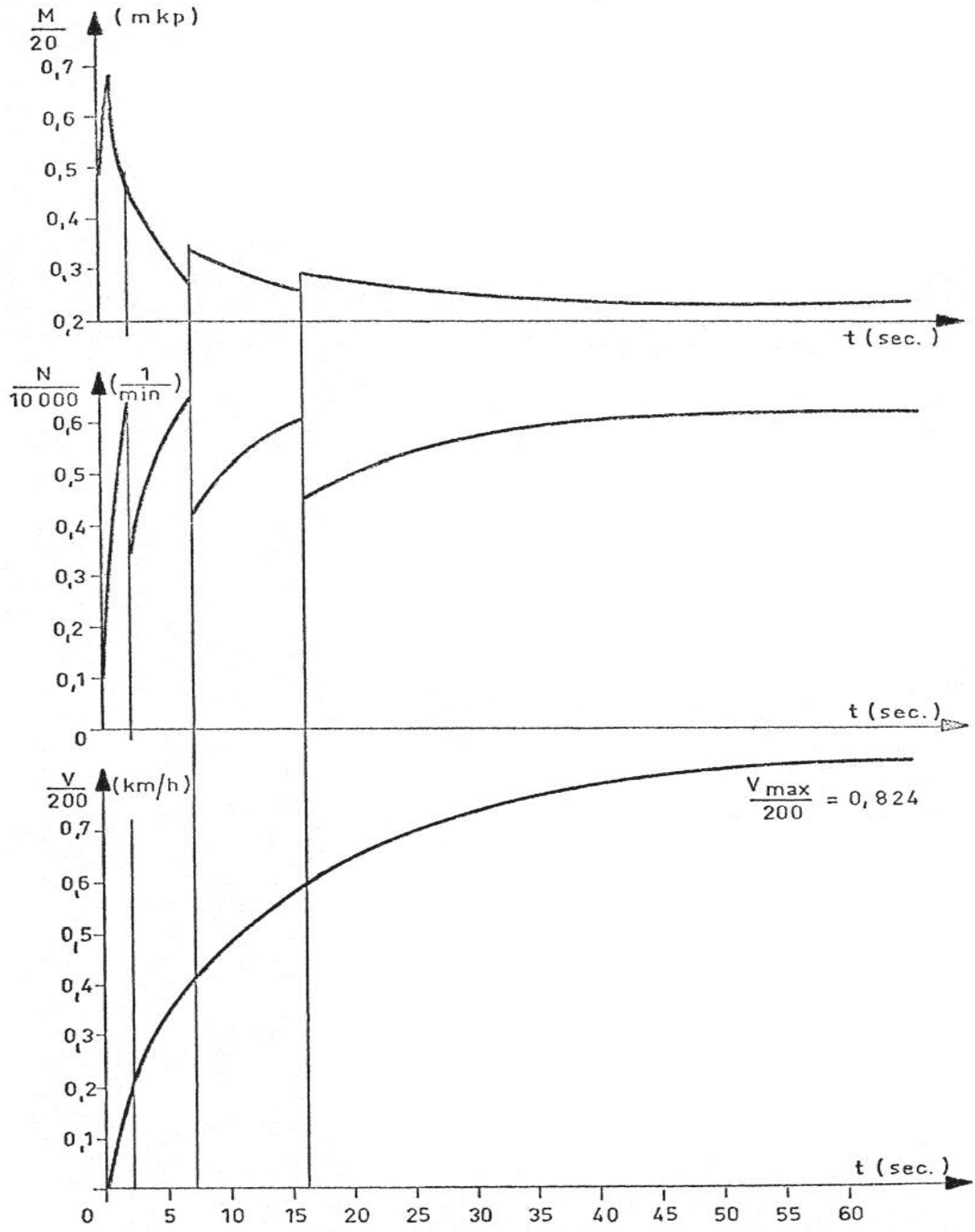


Figure 15.70. Results of a simulation run for the four-speed automatic transmission (see [→Dornier/1, p. 4])

[→KLITTICH 1966] contains a comprehensive treatment of the simulation of transmissions in general by means of analog computers.

[→FASOL 1986] describes a (for its time) novel and highly effective transmission gear for motor cars developed at the Institute for Gear Techniques at the Ruhr-Universität in Germany. This system was simulated on an EAI 2000 analog computer coupled with a DEC LSI 11/23 minicomputer system forming a rather powerful hybrid computer setup. Input to the system came from a foot operated pedal and the system also generated an audio output signal representing the noise generated by the motor of the car.

15.14.1.4 Ride simulation systems

The Westinghouse Mechanical Transient Analyzer[267] may be considered the first ride simulation system. From these humble beginnings, much more sophisticated systems were developed, such as the large-scale automobile ride simulation system described by [→KOHR 1960]. This system, built and used by the engineering mechanics department of General Motors, was capable of carrying two passengers in a car frame mounted on top of a hydraulic actuator system and could simulate the overall behavior of a car with respect to actual road “waves”, which were read from a magnetic tape unit and fed to an analog computer. This computer then determined the actual car motions by implementing a seven degree of freedom suspension system. It also controlled the hydraulic motion simulator. The overall system required only 82 operational amplifiers.

15.14.1.5 Traffic flow simulation

Interestingly, even dynamic systems with strong discrete properties such as traffic flows were analyzed using analog computers. A complex example can be found in [→LANDAUER 1974] where the simulation of a transportation system consisting of 40 stations, 600 vehicles, each of which able to transport up to twelve persons, ten crossroads, and 23 junctions is described. The goal of this simulation is the development of an optimal schedule ideally avoiding traffic jams or at least minimizing such effects. A much simpler simulation is described in [→JACKSON 1960,

pp. 371 ff.], which deals with a light signaling system and its parametrization.

[→Berger et al. 1979/2] describe a hybrid computer simulation of a metro network train based on differential equations for the train's velocity, energy, and travel time. The aim of the study is to determine energy optimal routes. It is shown that gravity is the most important effect resulting in railway lines having the steepest gradients at its start and end points.

15.14.2 Railway vehicles

Railway vehicle simulations are quite similar to those encountered in automotive applications. Notable exceptions include the simulation of *marshaling humps*[268] and the treatment of motor coach trains. [→Sydow 1964, pp. 238 ff.] describes the simulation of a direct-current railroad engine with emphasis on braking procedures. Two different braking techniques are analyzed: Braking by placing a resistive load in parallel to the motor acting as a generator and braking with recuperation of the energy dissipated by the deceleration.

[→Lotz 1970/2] describes an interesting application of an analog computer for fabrication quality control of motor vehicle wheels. One of the most important parameters here is the eccentricity of the wheel. An automated wheel test rig yields acceleration signals, which are then analyzed by performing a Fourier analysis. The results obtained by this are then used as the base for the actual eccentricity computation. All of these steps are done on an analog computer.

Another interesting study can be found in [→Gerecke 1967/2] where skidding effects in thyristor controlled railway locomotives are studied. The system under consideration contains four independent motors, each controlled by a thyristor inverter system. The friction between wheels and the track is modeled based on actual measurements. The aim of the simulation is to model the behavior of the locomotive from standstill to full speed on dry and wet tracks, optimizing the thyristor control for minimum slip. [→Fatton 1967] describes the design of an automatic control system for spring coupled vehicles with the goal of minimizing oscillatory behavior of the overall system. A similar problem, the position and velocity control of high-speed vehicles, is treated in [→Athans et al.

1967]. A very interesting analog computer model of urban train systems is described in [→Lehmann 1967]. The goal of the simulation is to find energy optimal operation modes as well as the determination of optimal time intervals between successive trains in automated signaling systems. This latter problem requires a rather complicated digital control unit setup implementing not only time delays between trains but also boolean conditions for various states of the overall train system.

15.14.3 Hovercrafts and maglevs

Analog and hybrid computer systems were also used in the development of hovercraft and maglevs. [→Leatherwood 1972] describes the simulation and analysis of a tracked hovercraft[269] with the goal of developing an active control system for the vehicle. [→Thaler et al. 1980] describe a complex hovercraft simulation system in which the actual simulation was implemented on three stored-program digital computers with a COMCOR CI-5000 analog computer being mainly used as an interface to the operators of the simulator.

The simulation of maglevs is more complex than that of hovercrafts although their basic principle is similar. Most of the additional complexity results from the possibility of coupling several cars together to form a train. Coupled systems like these cannot be treated analytically so that large analog computer installations were used in early maglev developments. An example can be found in [→Meisinger 1978] where a simplified one-mass maglev is modeled. This maglev hovers over a long track, which bends under the weight of the maglev.

15.14.4 Nautics

The development of ships and vessels also benefited from analog and hybrid computers as the following examples show.

15.14.4.1 Dynamic behavior

A complex simulation of the dynamic response of a ship subjected to waves is described in [→Maslo 1974]. Such studies are of high relevance with regard to loading and unloading operations as well as military

applications. Parameters of interest are the ship's geometry and dimensions with respect to the wavelengths encountered in the sea, etc. The partial differential equations involved in this type of problem were typically solved on hybrid computers where the discretization process was left to the stored-program digital computer. This particular simulation required 22 integrators, 42 summers, 28 inverters, 75 coefficient potentiometers, 30 comparators, 15 ADCs, and nine DACs, clearly showing the complexity of this type of problem.

[→Glumineau et al. 1982] describe another interesting study in which the dynamic response of an oil tanker is analyzed. The tanker is to be loaded and is anchored and subjected to wind and wave forces. Since the forces of interest in this system have most of their respective energies concentrated at frequencies of about 1/200 Hz,[270] the time-scaling ability of the analog computer is used to shorten the simulation time considerably.

15.14.4.2 Propulsion systems

The analysis of a complex turbine fuel system is described in [→Horling et al.]. This simulation was implemented on a hybrid computer, which was capable of yielding results in real-time. A comparative study implemented on a *Control Data CDC 6800* digital computer[271] only achieved 10 % of real-time while a third implementation on a *UNIVAC 1110* was timed at 1 % of real-time, clearly demonstrating the superiority of this hybrid simulation approach.

The simulation of a military vessel powered by twin diesel engines and a gas turbine is described in [→Thompson[a]]. Of special interest in this study is the behavior of the vessel while changing from one propulsion system to the other since the diesel engines are used for low and medium speed with the gas turbine only being used at high speeds. For a military vessel it must be guaranteed that there is no loss of thrust in all modes of operations and under all conditions and circumstances.

15.14.4.3 Ship simulation

While the simulations described above focus on specific aspects of ships' behaviors, some applications, especially the training of skippers and the

like, require complex and comprehensive full ship motion simulation systems.[272] A simple submarine simulation with only one degree of freedom is described in [→Jackson 1960, pp. 384 ff.].

The simulation of an assault boat *LCM-6*[273] is described in [→Kaplan[a]]. The focus of this study is on the dynamic behavior of the boat due to forces exerted by waves. The simulation of the waves is done on the digital portion of a hybrid computer while the analog computer is used to implement a six degree of freedom simulation of the ship.

[→Griswold et al. 1957] describe an early mechanical ship motion simulator

> "*intended to duplicate the essential features of the motions of a ship, so that these motions may be studied with reference to their influence on the handling, servicing, and launching characteristics of a missle or of any other device where these ship motions will affect performance.*"

Only limited degrees of freedom are taken into account in this simulator, namely roll, pitch, and heave motions. The actual system was built by *Chrysler Corporation Missile Operations* and featured a large platform capable of supporting a missile under test with a weight of up to 150 000 pounds, that could by hydraulically moved according to simulated (or actual measured) ship motions.

[→Kastner 1968/1] describes an electromechanical analog computer for the analysis of the motion of the sea. This motion is represented by an input voltage obtained by measurements on a research vessel. The analog computer then computed the Fourier coefficients of this signal and its power density spectrum.

In his Ph. D. thesis,[274] Kastner describes an analog computer setup for the simulation and analysis of a capsizing ship.

[→Takaishi 1965] describes the analog simulation of a rolling vessel with large, partially filled tanks for fluids. Special emphasis is put on the effects of nonlinear damping effects on the rolling of the vessel.

A intricate ship steering simulator is shown in →figure 15.71. The analog computer used in this setup is an EAI PACE system. Interesting are the two servo multipliers on top of the central slanted operating panel with attached motors. These were probably used to interface with the make-shift steering wheel, which in turn seems to have a synchro

mounted to its back. Also remarkable is the digital voltmeter, which also looks like an experimental device and a forerunner of the later DVMs used in the successor system EAI 231.

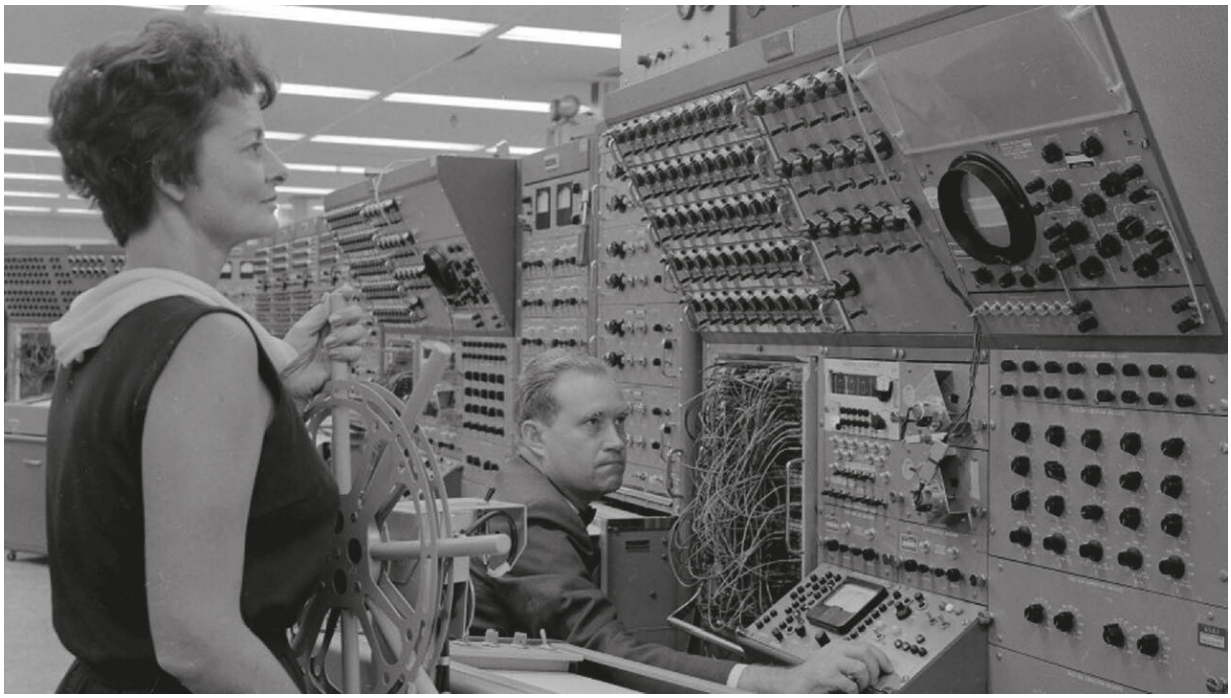


Figure 15.71. Ship steering simulator

[→Blumberg 1967] describes a "*real-time simulation facility [consisting] of an advanced two-man surface-ship bridge control center, with computer simulation of steering and speed-control systems, ship motions, and environmental conditions*" developed at the Marine Engineering Laboratory of Naval Ship Research and Development Center in Annapolis, Maryland. This incredibly complex system consists of four coupled EAI TR-48 analog computers with 232 amplifiers in total, connected to an EAI *DES-30* digital expansion system. The system also contained a *symbol generator* capable of generating basic symbols to be displayed at the bridge control center's CRTs. The simulation system not only takes things like propulsion controls and propulsion plant control, etc., into account but also implements thrust and torque coefficients of the propellers.

[→ Bilinski 1967] details a nonlinear ship model aimed at the development of a motion controller. This is based on data gathered by highly instrumented research vessels carrying accelerometers and pressure transducers.

15.14.4.4 Torpedo simulation

The development of torpedoes relied heavily on analog simulation techniques since tests involving real devices are extremely costly. Analog computers were used at the *Naval Undersea Center* for the simulation of torpedoes from the early 1950s onward.[275]

The complexity of this area of application is illustrated by the amount of analog computing equipment in use in the 1970s. All in all, two *EAI 8800* hybrid computers[276] each with 350 operational amplifiers, three EAI 231R analog computers[277], each with 250 operational amplifiers, and two stored-program digital computers, a UNIVAC 1110 and a UNIVAC 1230, were used. The first of these digital computers was coupled to 64 ADCs and 120 DACs while the second had 32 ADCs and 24 DACs.

This impressive installation allowed two torpedo simulations to be run simultaneously in real-time. These simulations also included models of the various seeker heads as well as the noise emission by the targets and the torpedoes.

15.15 Aeronautical engineering

Applications of analog and hybrid computers in the field of aeronautical engineering are manifold. Many of the driving forces for analog computer development actually came from this application area as the problems to be solved were extremely complex and money typically was not a limiting factor during the Cold War. Consequently, some of the first commercially available analog computers were developed and built by companies such as *Boeing, Goodyear, Short Brothers*, and *Reeves*, all strong players in aviation and space flight.

Simulations in aeronautical engineering and space flight often pose special requirements, which must be borne in mind when analog and hybrid computers are to be used. First of all many problems involve

variables with large domains, often too large to be handled by an analog computer directly.

To complicate things further, many simulations involve various coordinate transformations and coordinate system rotations, requiring either special analog computing elements such as resolvers[278] or many sine/cosine function generators, multipliers, etc.[279] In addition to this, many aerospace simulations also involve functions of more than one variable, which either requires a vast amount of analog hardware or a digital computer as part of a hybrid system.[280]

The importance of analog computers in these areas of application has been emphasized by [→Levine 1964, p. 2]:

> "*Ehricke*[281] *believes that the accelerated development of American missiles would not have been possible without [analog] computers.*"

→Figure 15.72[282] shows an early *REAC* 100[283] used by *NACA*[284] as early as 1949. For its time this machine was remarkably large and complex. Another indicator of the importance of analog computers for aerospace research is the sheer size of the often purpose built systems:



Figure 15.72. Very early REAC analog computer (most probably a REAC 100) used by NACA as early as 1949 (detailed information about these systems can be found in [→Reeves Instrument Corporation])

> "*The world's largest and most advanced general purpose analog computer; designed and built for the National Aeronautics and Space Administration, is [...] undergoing final inspection and check-out at this company's plant. Developed and built under a $1,510,000 contract, all units of the computer will be installed at NASA's research center at Langley Field, Va., this spring. The computer system will be used for analytical computation in a vast electronics calculation program for simulating future satellite and space vehicle programs. It occupies more than 1,800 square feet of floor space and comprises five separate PACE 231R computer consoles; these will be used separately or in slaved groups with a sixth unit already installed [...]*"[285]

Eventually, this system featured more than 1500 computing elements and measured 100 feet in length.[286] Another early study of aircraft

performance using an analog computer can be found in [→WADEL et al. 1955]. This study also notes that the number of integrators in such simulations is comparatively small while the number of multipliers and function generators required is exceptionally high.

The size of analog and hybrid computers continued to grow over the 1960s and 1970s as shown by the Lockheed hybrid system which is aptly called “a giant step” and became operational in 1967 consisted of a Control Data *6400* digital computer[287] coupled with four large Comcor *Ci-5000* analog computers via two Comcor *Intracom* linkages.[288] Each of these analog computers contained several hundred computing elements.

Two interesting sources are [→BARNETT 1963] and [→EAI 1964/2], the first giving an account of the NASA-AMES hybrid computer facilities and their application to problems in aeronautics, the latter detailing on the application of hybrid computers in aerospace engineering in general. The following sections show some typical application examples of analog computers in this field. Basic information on these topics can be found in [→BAUER 1962/1].

15.15.1 Landing gears

The development of landing gears for high-performance aircraft is an astonishingly complex and dangerous task if it is based on experiments with real flight hardware.[289] As early as 1953, analog computers were used to simulate the dynamic behavior of this aircraft subsystem at the *Wright Air Development Center*.

The first simulations of landing gear were performed on analog computers under the auspices of Professor WILLIAM J. MORELAND.[290] These were based on systems of differential equations with four degrees of freedom and 14 parameters. An interesting detail of these studies is that the shock absorbers used, and modeled, often had different dynamic behavior depending on the direction of movement of the piston, further complicating the simulation task.[291]

Since no hybrid computers were available in these early days, the necessary parameter variations were done manually, which nevertheless

quickly yielded feasible landing gear configurations as the following quote shows:[292]

> "*For the past three years all new landing gear designs have been evaluated according to Professor* Moreland*'s method, thus eliminating the dangers that accompany violent shimmy.*"[293]

[→Drake et al. 1963] describe the simulation of an aircraft landing gear on an analog computer and compares results from these simulations with data obtained from actual drop tests of a landing gear with very favorable results: "*[...] given an accurate functional knowledge of landing gears, accurate simulations of a given gear can be expected on the computer*".[294]

15.15.2 Aircraft arresting gear systems

Aircraft arresting gear systems aboard aircraft carriers are highly complex systems. They must withstand the immense forces exerted by aircraft being quickly decelerated. Further, they must also exhibit a force characteristic that prevents damage to aircraft structure during its deceleration. Consequently, control of the braking system used to tension the arresting cable is a difficult task that has been successfully modeled and analyzed using analog computers. Such a simulation, focusing on two types of aircraft, a jet fighter and a light bomber, is described in [→Carlson et al. 1967, pp. 296 ff.].

The opposite system, a take-off catapult for aircraft carriers, is described and analyzed in [→Addicott et al. 1968].

15.15.3 Jet engines

The simulation of a turbo jet engine with afterburner on an analog computer is described by [→Jackson 1960, pp. 426 ff.]. This study focuses on the design of a control system for the variable-geometry exhaust of this particular engine. The simulation requires function generators for functions of more than one variable, substantially complicating the computer setup.

A simplified jet engine simulation is covered in [→Schweizer 1976/1, pp. 422 ff.]. A study to test the compatibility of a particular jet engine and a special inlet is described in [→Costakis 1974]. A comprehensive treatment of the simulation of a turbojet engine can be found in [→Rockcastle et al. 1956]. Another jet engine simulator is described in [→Skramstad 1957, pp. 92 ff.].

15.15.4 Helicopters

The analysis of bending and torsion effects in helicopter rotor blades using an analog computer is described in [→McLeod et al. 1958/4, pp. 1222 f.]. Mechanical systems such as rotor blades are described by partial differential equations and are thus not well suited for direct treatment with an indirect analog computer. Accordingly, the system used in this study was a direct analog computer similar in its structure to the HETAC and EAFCOM.[295] It was described as follows:[296]

> "*The direct analog computer consists of an assemblage of passive electrical circuit elements (resistors, capacitors, inductors and transformers), amplifiers, signal generators and control equipment. [...] The electrical analogy for the bending of beams has great importance in the direct analog method of dynamic analysis for lifting surfaces, since many lifting surfaces can be replaced by lifting lines with bending and torsional flexibility. This is certainly true of the helicopter rotor blade which characteristically has a very large span-to-chord ratio.*"

The system consisted of about 100 inductors and 50 resistors and capacitors representing seven discrete sections of a rotor blade.

15.15.5 Flutter simulations

A complex direct analog computer for the "*study of supersonic flutter of elastic delta wings*" is described in [→Basin 1954]. Here, the pressures acting on various positions on the wing are derived as power series, which can be handled quite easily. From this the resulting lift is computed, forming the base for dividing the wing into small cells which are then represented by electrical analogs consisting of passive elements.[297]

[→Smith et al. 1959] describe a large and complex flutter simulator capable of up to six degrees of freedom, which is shown in →figure

15.73. A typical simulation result obtained for a problem with four degrees of freedom is shown in →figure 15.74. It shows the response amplitude "*as a function of excitation frequency at three different air-speed settings*.[298]

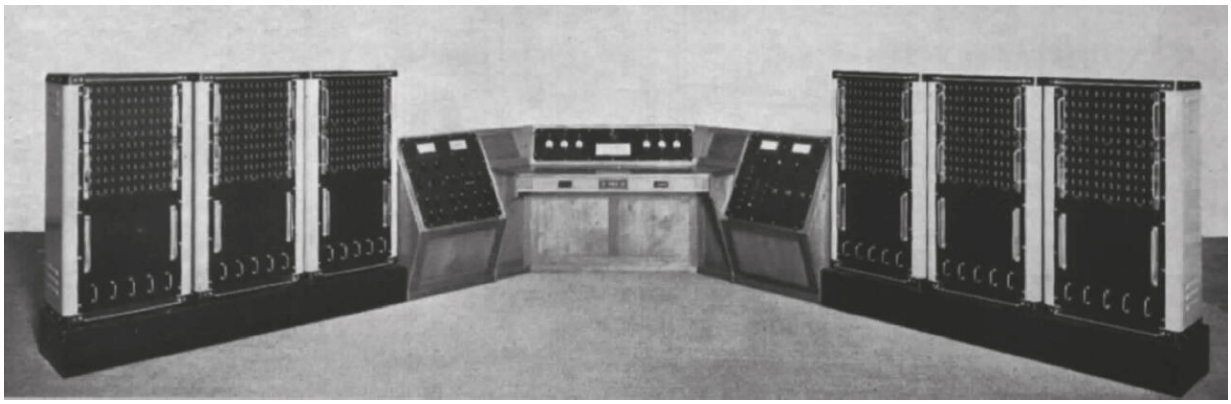


Figure 15.73. Front view of the R.A.E. electronic simulator for flutter investigations in six degrees of freedom (see [→Smith et al. 1959, p. 44])

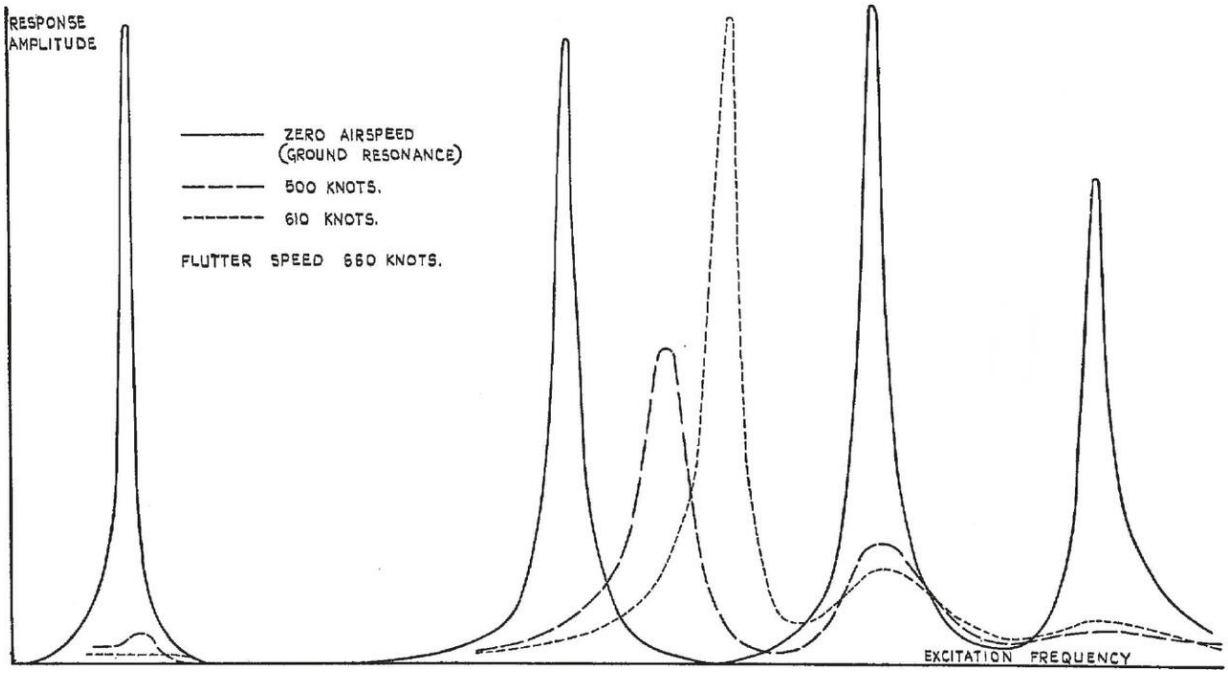


Figure 15.74. Typical simulation obtained by the R.A.E. flutter simulator [→Smith et al. 1959, p. 42])

Flutter simulations are complicated as this effect causes parts of an aircraft's structure to oscillate in a variety of modes while being coupled mechanically as well as by aerodynamic effects. This system was basically a large indirect analog computer – probably the first of its kind for this type of problems.

An even more complex successor of this system is described in [→Hicks 1968]. Vacuum tubes as well as transistors were used to implement this system, which was capable of simulating the behavior of up to twelve coupled linear differential equations of second order. The work described by [→Salvioni 1971], which deals with flutter of cylindrical panels used in aerospace applications, is especially interesting as not only the actual bending behavior of the panels is simulated but the analog computer is also used to create an intuitive projection of the three-dimensional structures on an oscilloscope using an intricate digital control system. →Figure 15.75 shows a typical screen shot. In addition to this the simulation was scaled to be run at two different time scales – one in real time, aiding the observation of snapping movements, the second running one hundred times faster, used for the graphic display of bending panels.

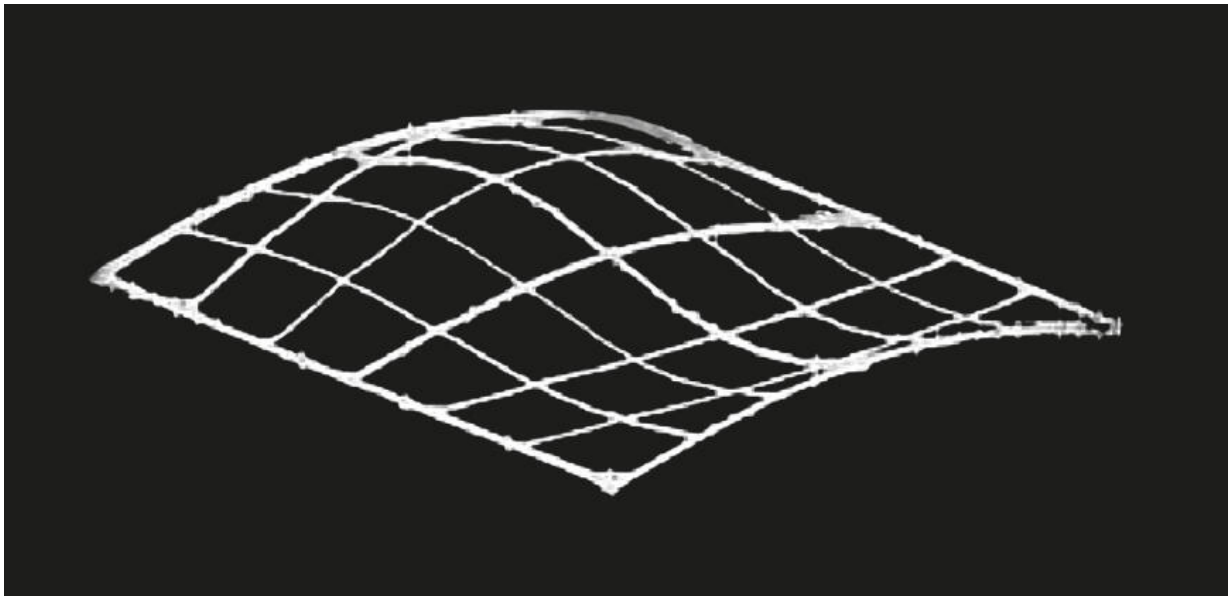


Figure 15.75. Typical output of the flutter simulation described in [→Salvioni 1971, p. 143]

15.15.6 Flight simulation

One of the most fascinating applications for analog computers are flight simulators, in which aircraft are modeled with typically five or six degrees of freedom. This allows the behavior of various aircraft configurations to be studied, without the necessity of building complex, costly, and sometimes outright dangerous prototypes risking the lives of test pilots.[299] Analog computer based flight simulation systems are characterized as follows by [→Korn et al. 1956, p. 119]:

> "*The operation of a flight simulator on the ground is vastly cheaper as well as safer than operation of actual aircraft.*"

Maybe the most impressive early flight simulation system was the *RCA Typhoon*, put into operation in 1951 and used for 17 years until 1968 when it was finally dismantled.[300] The system was used for a wide variety of aerospace projects including the famous (and in many respects unsurpassed) *X-15* project.[301] It contained 4000 vacuum tubes in 450 DC amplifiers, 20 servo multipliers, 80 integrators, and lots of additional equipment filling 43 racks and consuming 46 kW of power. An interesting element in this very early system were hybrid multipliers resembling today's multiplying DACs capable of multiplying an analog value with a value represented digitally. These early DACs used special relays capable of switching in 100μs without bouncing.[302] These DAC based multipliers could change from 0 to 1 incrementally in just one second.[303] Thus, the system was already a step towards hybrid computing, designed to yields a degree of flexibility and accuracy not obtainable from other systems of its time. It was the culmination of three years of research and development work under the guidance of *Arthur W. Vance*, with the resulting overall system priced at 3 million USD, about 40 million USD of today's money. According to [→Electronics 1951] some of the system's applications were the determination of optimum trajectories for radar-controlled guided missiles for the interception of attacking bombing targets.

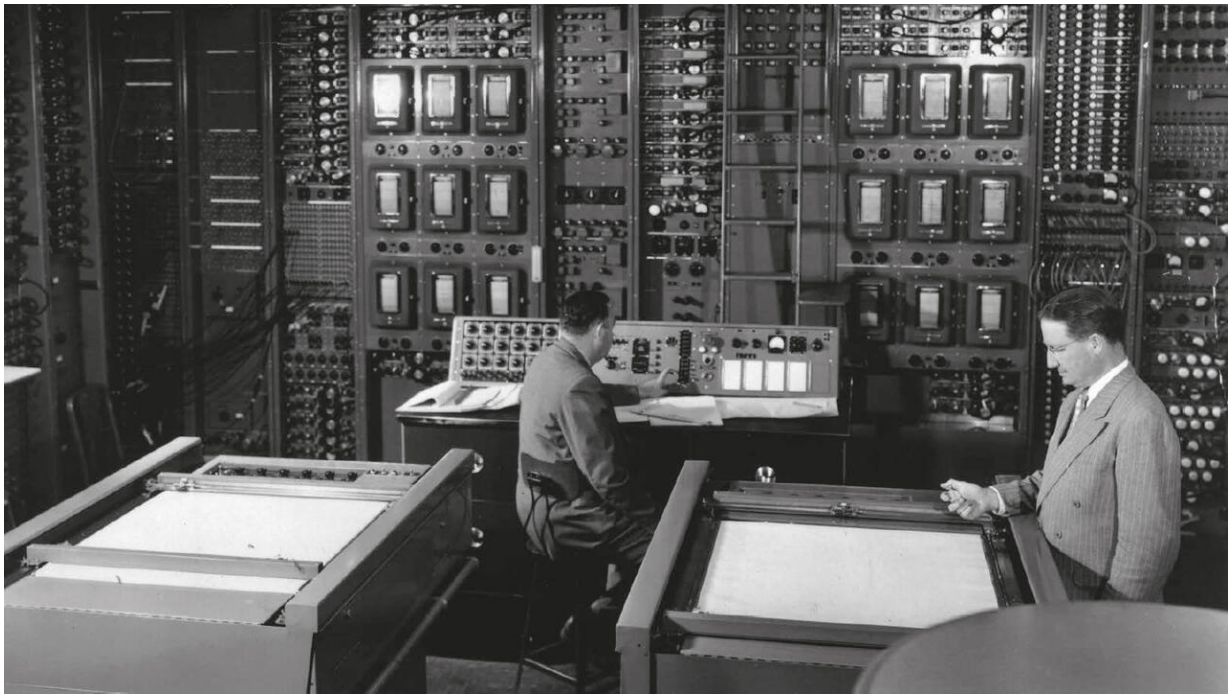


Figure 15.76. Partial view of the RCA Typhoon analog computer

Parallel to these developments, the *US-Air Force* acquired a GEDA analog computer[304] in the early 1950s, which would eventually become the first in a long series of increasingly complex air force flight simulation systems.[305]

A very simple flight simulation in the vertical plane, taking only the movements of the elevator into account, is described in [→Wass 1955, p. 39]. →Figure 15.77 shows the basic setup of this simulation. Here, u denotes the speed along the x-axis, Θ is the inclination, while q represents the angular rate with respect to the y-axis. Based on this, a simple simulation with the elevator angle η as its input can be derived from the following simplified set of equations:

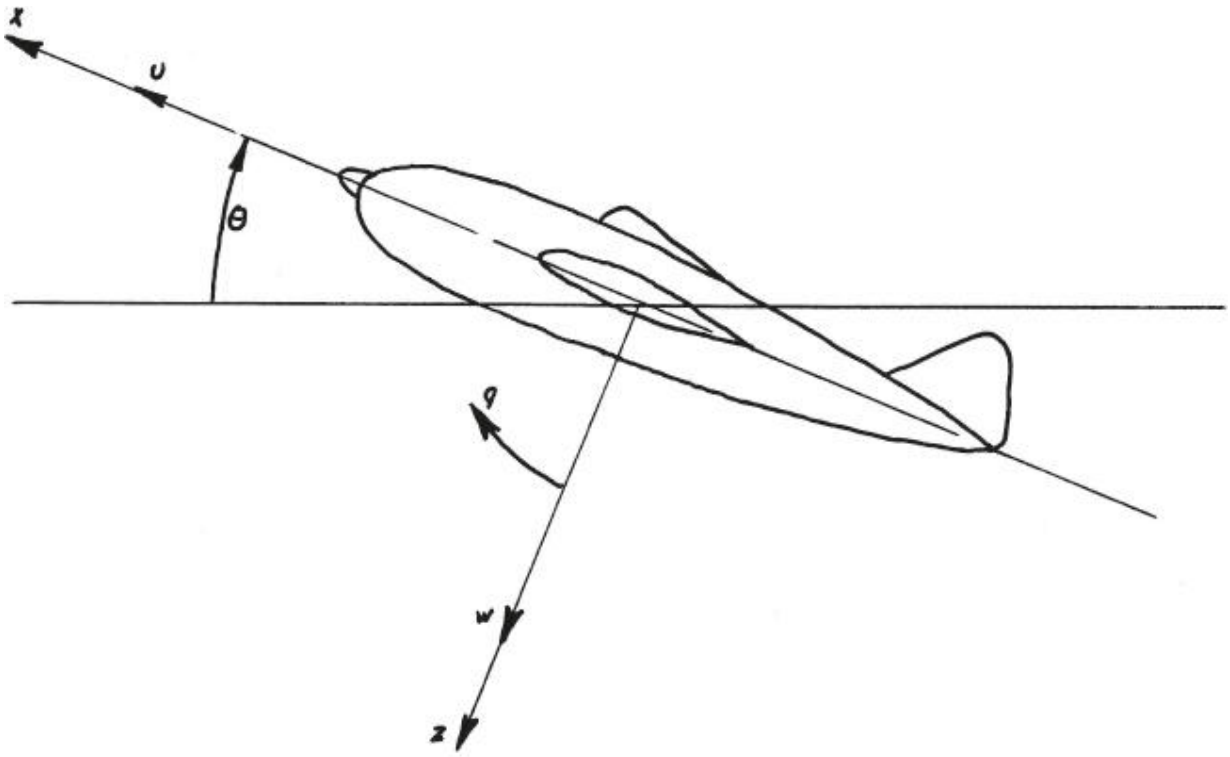


Figure 15.77. Longitudinal motion of an aircraft (see [→WASS 1955, p. 40])

$$\begin{aligned} m(\dot{u} + w(0)q) &= -mg\,\cos(\Theta(0))\Theta + uX_u + wX_x \\ m(\dot{w} + u(0)q) &= -mg\,\sin(\Theta(0))\Theta + uZ_u + wZ_w \\ B\dot{q} &= wM_w + qM_q + \eta M_\eta \end{aligned}$$

Θ(0) is the initial angle of inclination, Θ the change of inclination, $u(0)$ and $w(0)$ represent the initial speed components along the *x*- and *z*-axis. *B* is the moment of inertia, Z_u, Z_w and X_u, X_w represent the aerodynamic forces along the *z*- and *x*-axis per unit of *u* and *w*. Finally, M_w, M_q, and M_η are the moments about the *y*-axis per unit of *w, q*, and *η*.

Based on these equations, the program shown in →figure 15.78 can be derived. Using *η* as its input value, the movements of the aircraft in the vertical plane can be determined. This computer setup can now be extended to include a simple autopilot as shown in →figure 15.79. The autopilot shown will hold the line of flight constant.[306] The term for the required elevator setting to be determined by this simple autopilot is[307]

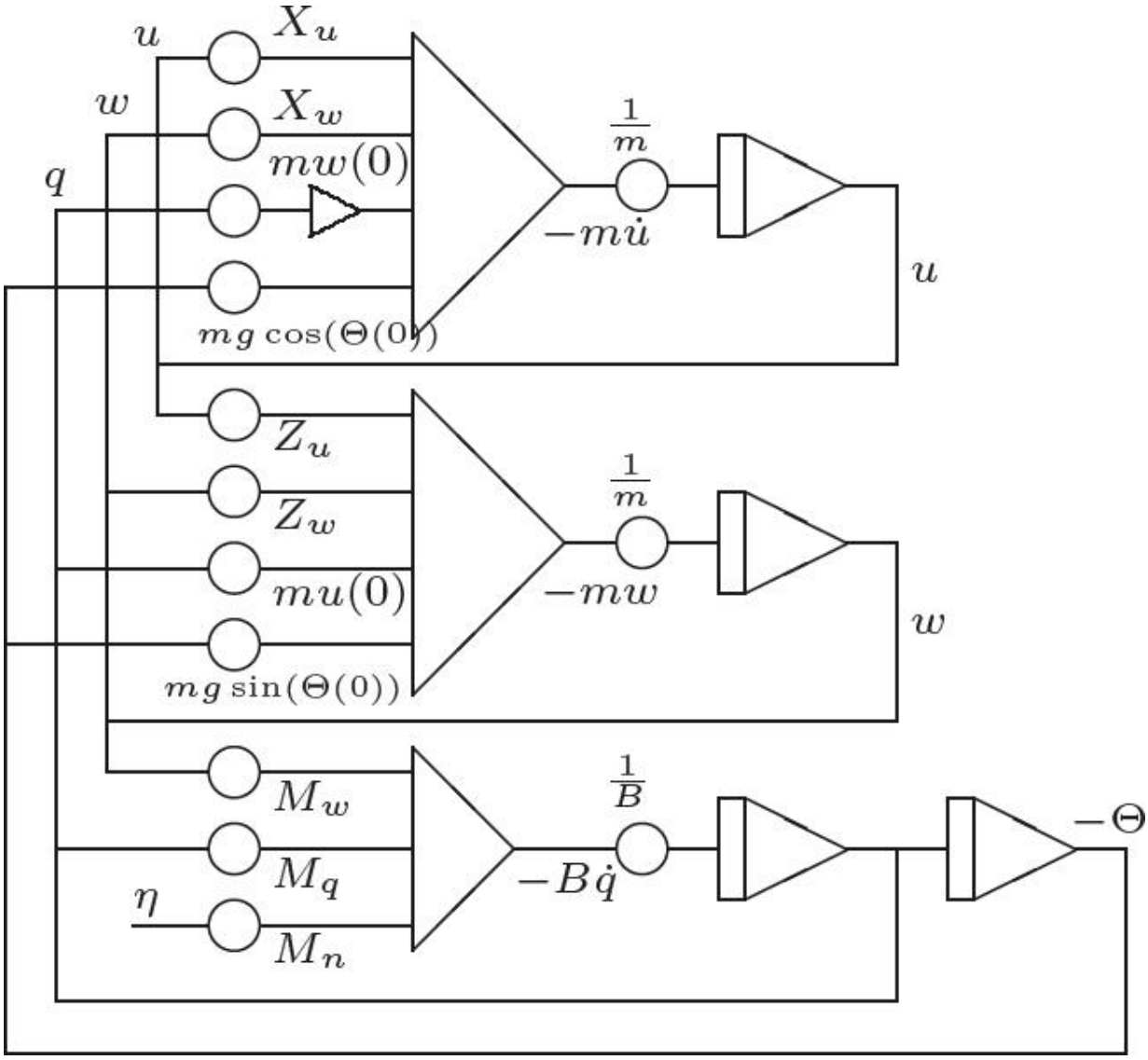


Figure 15.78. Simulation circuit for the longitudinal motion component of an aircraft (see [→WASS 1955, p. 40])

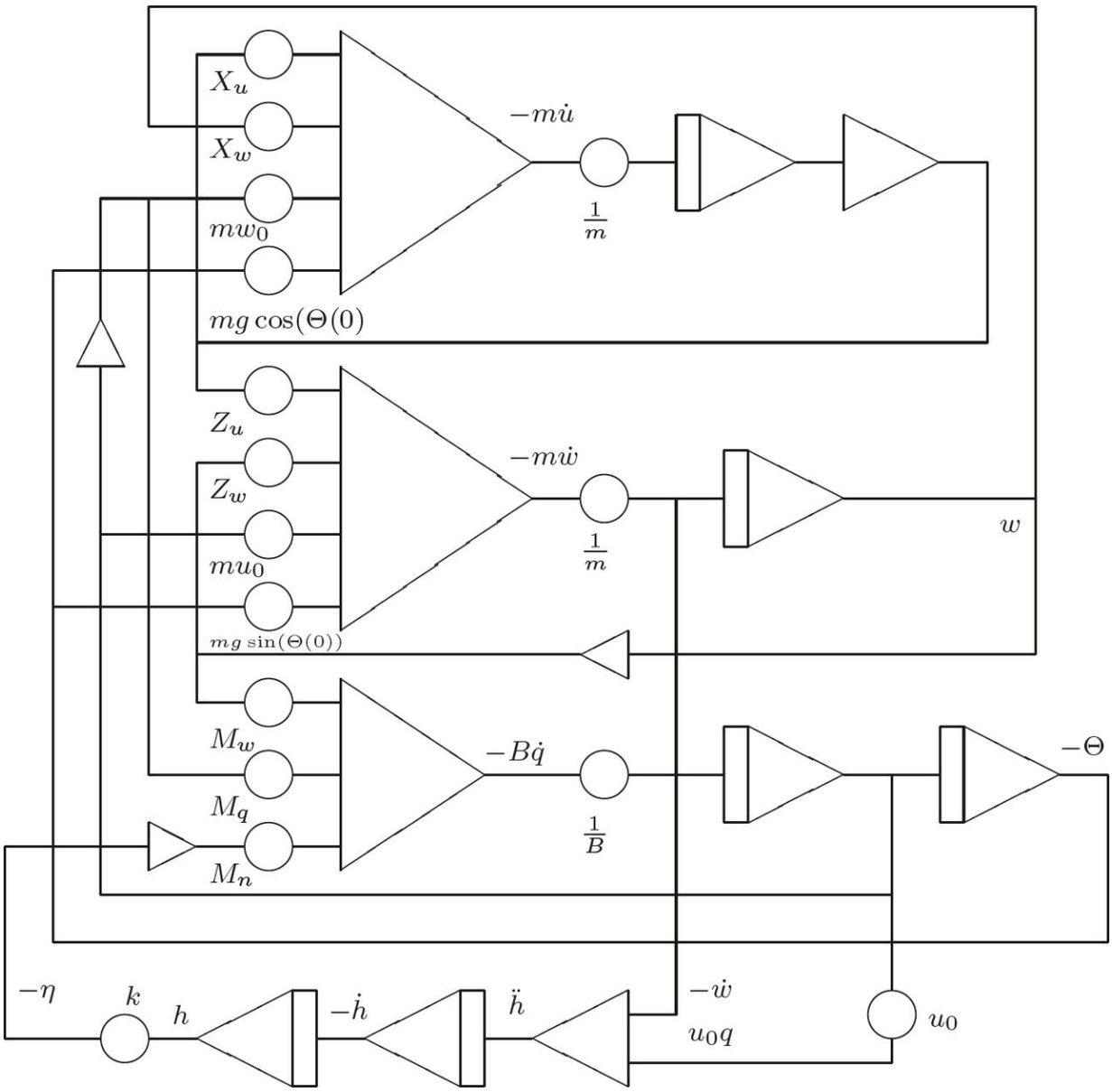


Figure 15.79. Longitudinal motion simulation with simple autopilot (see [→WASS 1955, p. 43])

$$\eta = k \iint (\dot{w} - u(0)q)\,\mathrm{d}t\,\mathrm{d}t.$$

The variables $\dot{w}$ and $u(0)q$ in this equation are readily available from the previous computer setup.

A much more complex simulator system is described in [→NOSKER 1957] – the *Dynamic Systems Synthesizer*. This system is especially noteworthy as it abandoned the classic patch panel, which was replaced

by a punch card configuration system shown in →figure 15.80, thereby allowing rapid reconfiguration of the system. It also featured a *trigonometric resolver*, which generates sin(φ) and cos(φ) with φ given as the quotient of two values x and y.

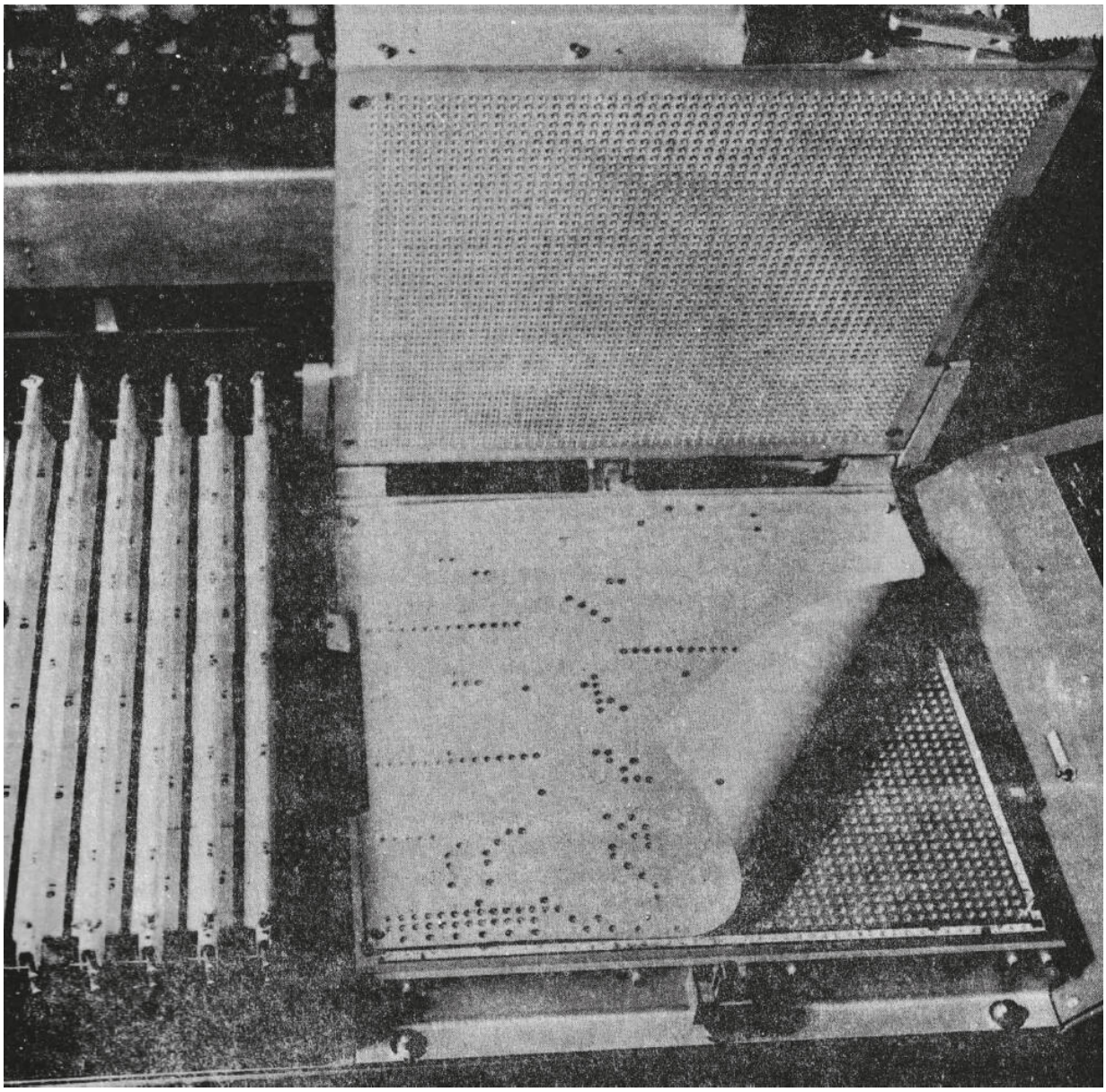


Figure 15.80. Punch card programming system of the Dynamic Systems Synthesizer (see [→Nosker 1957, 156])

Of course most of the analog and hybrid computer based flight simulators[308] were much more complex than these examples might suggest. One machine, *TRIDAC*,[309] is particularly interesting not least due to its sheer size and complexity. It was inaugurated on 08.10.1954 at the Royal Aircraft Establishment, Farnborough, and was described as follows in *The Times*:[310]

> "*He was introducing Tridac […] which is ten times larger than anything else of its kind in this country and one of the biggest computers in the world. It has been installed by the Ministry of Supply at a cost of about £750 000 and within its massive form – the equipment would fill six ordinary three-bedroom houses – there are mechanical computing elements which need 400-horse power to drive them. The total electricity consumption would light a small town. As an example of its capacity, in 20 seconds Tridac could achieve as much as 100 girls using calculating machines in eight hours.*"

→Figure 15.81 shows the basic structure of TRIDAC. In contrast to general purpose analog computers, most of TRIDAC's computing elements were grouped into building blocks performing basic operations required for flight simulation. These building blocks could then be configured and coupled together arbitrarily.[311]

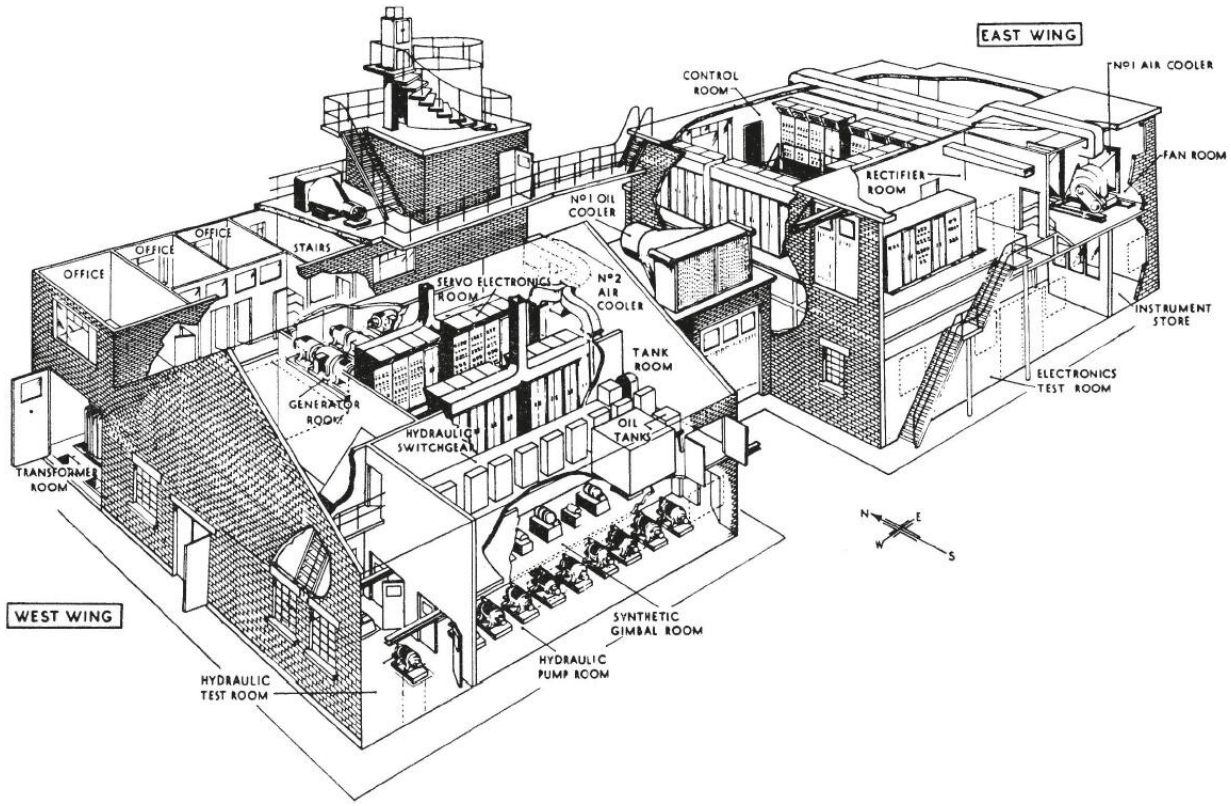


Figure 15.81. Structure of the TRIDAC simulator (see [→WASS 1955, Fig. 141])

The central element was the *aerodynamic unit* surrounded by blocks of integrators, which integrated rotational and translation movements of the simulated flight vehicle. There were also resolver blocks, which derived the necessary signals for an *aircraft position computer*. Units to simulate autopilots, the dynamic behavior of control surface motors, etc., were also available. All in all, TRIDAC contained about 650 chopper-stabilized operational amplifiers and a plethora of hydraulic function generators and resolvers.[312] It achieved an accuracy of 0.1%.[313] →Figure 15.82 shows the central control console of TRIDAC, making the size of the overall system palpable.

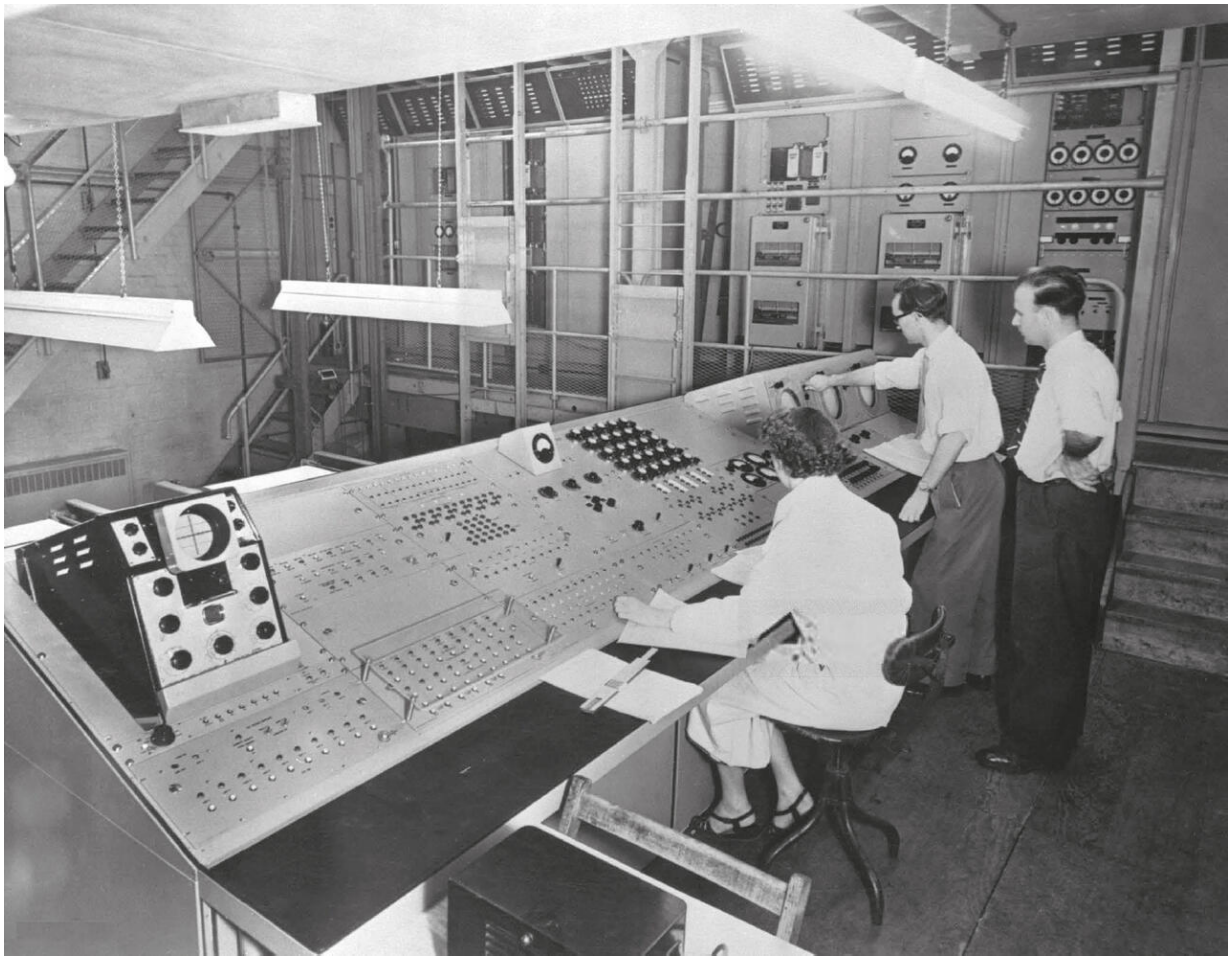


Figure 15.82. Control console of the TRIDAC system

Probably the first supersonic analog flight simulator is described briefly in [→MACDONALD 1955]. This system was demonstrated in March 1955 at the Nellis Air Force Base, Las Vegas, Nevada. It featured a full-fledged cockpit of an *F-100A* supersonic aircraft and was used as a training device for prospective air force pilots. An instructor could throw problems at the pilot and monitor his performance under stress, which could be enhanced by flash lights simulating lightning, and speakers for engine and flight noises.

LACE, the *Luton Analogue Computing Engine*, is described in detail in [→GOMPERTS et al. 1957] and [→JONES et al. 1957]. It was a general purpose analog computer, which found many applications in the field of flight simulation and was built by the *Guided Weapons Division* of the *English Electric Company*, Luton.

In cases where an analog or hybrid simulation required hardware-in-the-loop, *flight tables* were employed. The earliest such flight table was developed in Peenemünde (Germany) during the development of the A4 rocket.[314] The idea was to mount the gyro system of a rocket or aircraft system under investigation on a platform that could be tilted under control of an analog computer to simulate the movements of the rocket and thus evaluate the performance of the gyro and other guidance system components. While this first flight table, being used until the mid 1950s at the Redstone Arsenal during the development of the Redstone missile, only offered one degree of freedom, later systems could be moved in three axes.[315] A very complex system consisting of the 3-axis flight table and attached analog simulator for testing guidance and control equipment is described in [→Bendix 1957]. The flight table of this installation could even be positioned outdoors for “line-of-sight simulation of missile guidance systems”.

Not all problems required such large analog computers for their solution. →Figure 15.83 shows an inertia coupling simulation performed at the NACA *High-Speed Flight Station* in 1955 during the development of the *Bell X-2* plane. Richard E. Day,[316] one of the pioneers of analog flight simulation at NACA and later NASA, can be seen on the control stick, with two GEDA analog computers in the background. The high speed of analog computers made it possible to implement man-in-the-loop simulations where the dynamic response of a simulated aircraft could be experienced directly.



Figure 15.83. Inertia coupling simulation (Richard E. Day at the control stick, see [→Waltman 2000, p. 6])

TASS,[317] a large hybrid computer based simulation system, is shown in →figure 15.84. This system consisted of several EAI 231R analog

computers, a *DOS-350* digital subsystem,[318] and several digital computers such as an *EAI 8400* and a *DDP-24*, etc.[319]



Figure 15.84. The *Tactical Avionics System Simulator* (photo from Bruce Baker, reprinted with permission)

A central problem in flight simulation are the large value intervals of variables such as the height of the plane over ground. The precison of an analog computer alone is typically not sufficient to implement all flight regimes at once in a simulation. An airplane with a ceiling height of 10 km, represented by a voltage of 100 V, would land at a height represented by 10^{-4} V, requiring an analog value resolution of 10^{-6}, clearly impossible. Another problem is that realistic flight simulations typically involve functions of several variables – the lift coefficient, e. g.,

depends on nine arguments. This is clearly not possible with purely analog means of computation. Accordingly, hybrid computers are required where the analog computer simulates effects in, of, and around the air plane, while the digital system takes care of the overall trajectory from start to landing, function generation, etc. An example for this is described in [→Roderer et al. 1967]. Here, the hybrid computer simulation of the jet-propelled, vertical take-off and landing (*VTOL*) cargo aircraft *Do 31*[320] is outlined.

A typical hybrid computer setup for a flight simulation system used in a terrain avoidance guidance study is shown in →figure 15.85. The analog computer, shown on the left, implements the actual aircraft flight dynamics and the aircraft flight control system, while the digital subsystem, shown on the right, simulates the radar system, the terrain function generator with a range of 100 miles, the guidance computer, etc.[321]

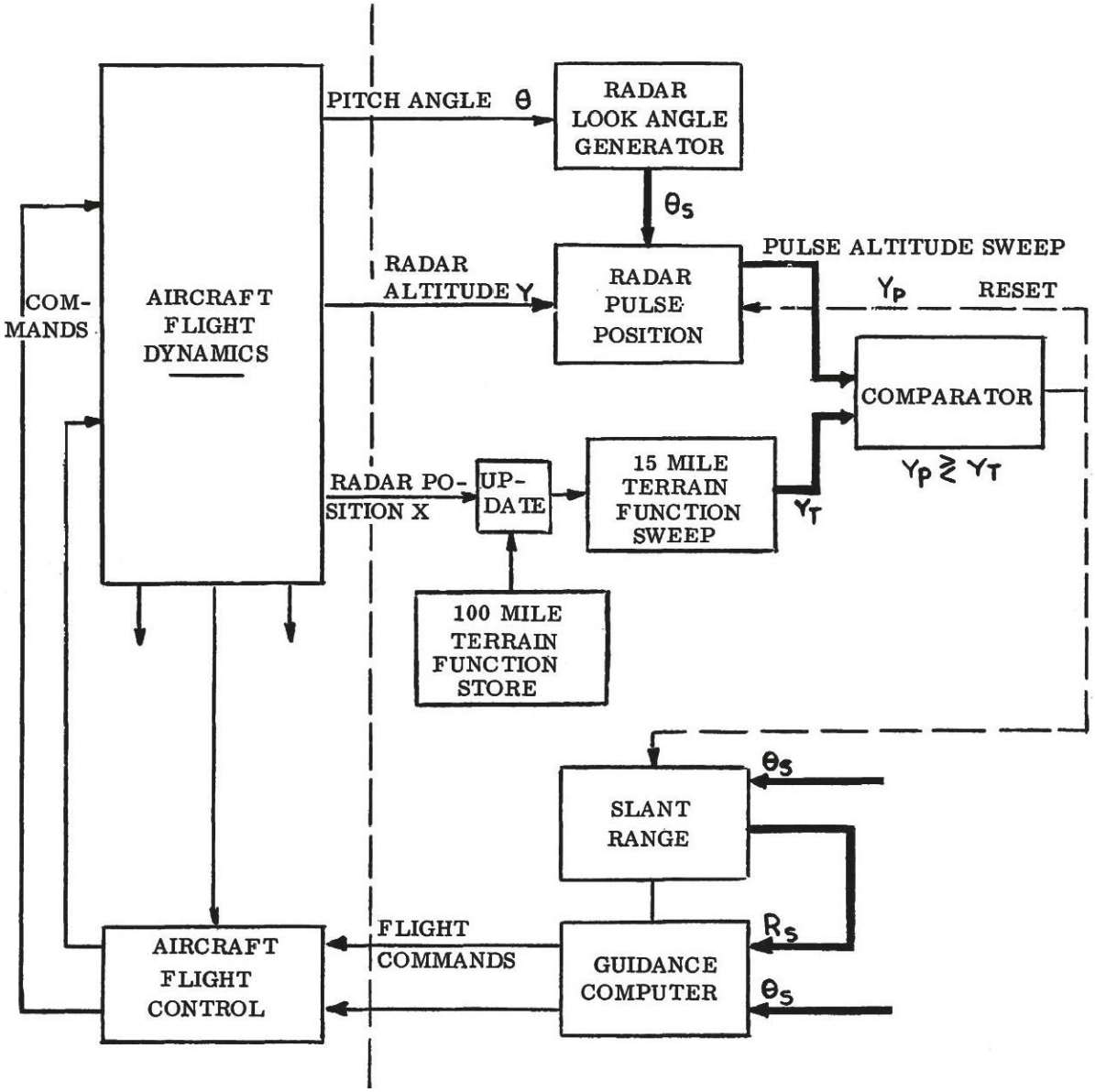


Figure 15.85. Hybrid computer setup for a terrain avoidance guidance study (see [→ ROGERS 1963, p. 14])

Another impressive analog computer system, a *Beckman EASE 2133*, was operated at the *MBB*[322] Aircraft Division and is shown in →figure 15.86. This system featured 520 operational amplifiers, 72 multipliers, 64 function generators, ten sine-/cosine-function generators, 24 limiters, 240 servo coefficient potentiometers, 112 manual coefficient potentiometers, and a variety of digital control elements.

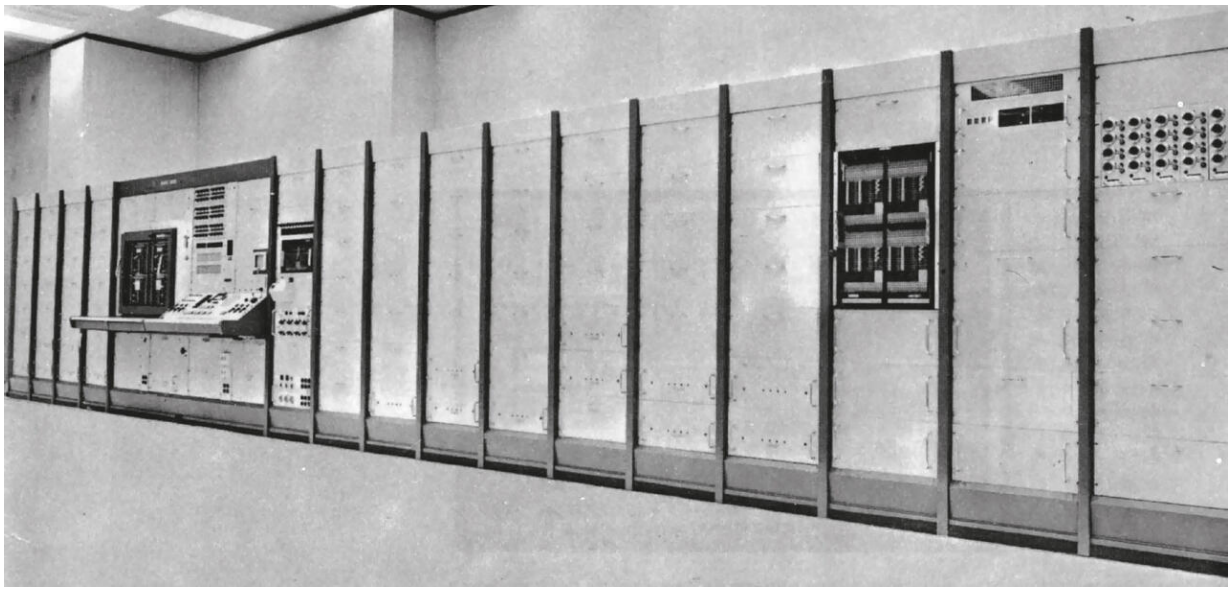


Figure 15.86. Beckman EASE 2133 analog computer installed at MBB Aircraft Division (see [→MBB])

The simulation of a complex aircraft such as the *VJ 101C-X2* vertical take-off plane required even larger analog computers. During the development of this particular plane, an analog computer installation containing more than 1 000 operational amplifiers was used at MBB.[323]

A more recent analog computer installation is shown in →figure 15.87. Although fully transistorized, the EAI 8800 analog computer (as the later PACER 700 systems) employed machine units of ±100 V which made them plug-in replacements for earlier vacuum tube based systems coupled to extensive external equipment such as airplane cockpits in simulator installations, etc.

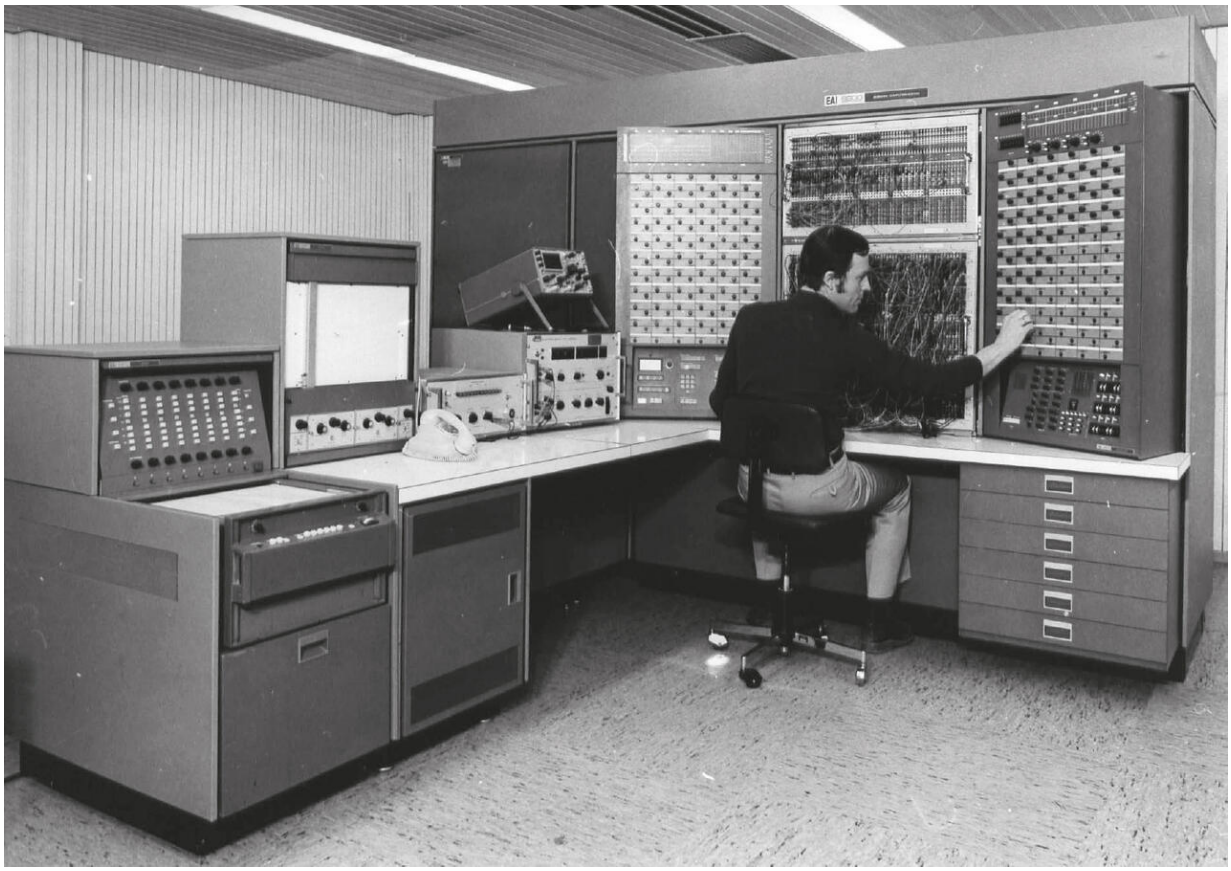


Figure 15.87. EAI 8800 analog computer system installed at the Dornier simulation center in the 1960s

The accuracy of early analog computer simulations was demonstrated in a striking, yet tragic, way in 1956 when test pilot MEL APT[324] died as the result of a stability problem of the X-2 plane that had been predicted previously by a simulation:[325]

> "*We showed him if he increased AOA*[326] *to about 5 degrees, he would start losing directional stability. He'd start this, and due to adverse aileron, he'd put in stick one way and the plane would yaw the other way […] We showed APT this, and he did it many times.*"

At that time, test pilots did not show much trust in the results of such simulations, which clearly were not "the right stuff" as RICHARD E. DAY remembers:[327]

> "*Well, the simulator was a new device that has never been used previously for training or flight planning. Most pilots had, in fact, expressed a certain amount of distrust in the device.*"

During Mel Apt's test flight, he encountered the critical region at Mach 3 where roll-coupling set in, as predicted by the analog model, and the X-2 entered a fatal spin. This fatal accident was then analyzed by the very same analog computer model.[328] As a result, flight simulation played an increasingly central role in the following years in aircraft development, as well as in pilot training. Eventually, Neil A. Armstrong[329] spent about 50 to 60 hours of simulator training for each of his flights with the *X-15*, while every real flight only lasted about ten minutes:[330]

> "*[…] our simulators in the space program were so much more sophisticated and accurate, and our preparation was so much more intense, that we convinced ourselves that the pilots could handle whatever situation we might encounter in flight.*"

A description of the simulation techniques used during the X-15 project can be found in [→ Day 1959]. [→ Mitchell et al. 1966] cover additional details of the simulation implementations. Analog computer studies regarding the X-15's reaction control system are described in [→ Stillwell 1956], while [→ Stillwell et al. 1958] deal with basic aspects of such simulations.

A central problem in the early days of analog flight simulation was how to display information in the cockpit. Simple, yet readily available, analog instruments proved to be insufficient, since pilots were, and still are, used to the special instruments found in real cockpits:[331]

> "*Presenting the pilot with the necessary flight instruments was a problem of major concern. Originally galvanometer-type meter movements were used. This type of display seemed satisfactory to the analysts and engineers with no actual flight experience. However, serious objections were raised when experienced test pilots were asked to evaluate the simulator's performance. Even though the necessary information was being presented to the pilot, the unrealistic instruments distracted the pilot so much that they impeded his efforts to truly evaluate the flight condition.*"

Consequently, adapters were required to interface the analog and hybrid computers at the heart of these simulations to traditional cockpit

instruments. Later systems used universal graphic display units as described in [→SCHWEIZER 1976/6, pp. 531 ff.].[332]

To create a realistic impression for the pilot, a flight simulator not only requires a cockpit, which is ideally mounted on a hexapod platform providing realistic movement, but also needs a realistic representation of the outside world. →Figure 15.88 shows the overall structure of the hybrid flight simulation system installed at *Martin Marietta's Simulation & Test Laboratory* (*STL*). The picture generation system is based on a large terrain model, which is traversed by a video camera system capable of moving in three axes.[333] This particular model measures 80 by 40 feet. A scale of 1200:1 was used for aircraft simulations while a scale of 225:1 was used for helicopter simulations.[334] →Figure 15.89 shows the terrain model in detail with a person standing on the far right giving an impression of its size.[335] The hybrid computer system used at Martin Marietta consisted of a three-CPU *Sigma-5* digital computer and six EAI 231RV analog computers with 1496 operational amplifiers in all.[336]

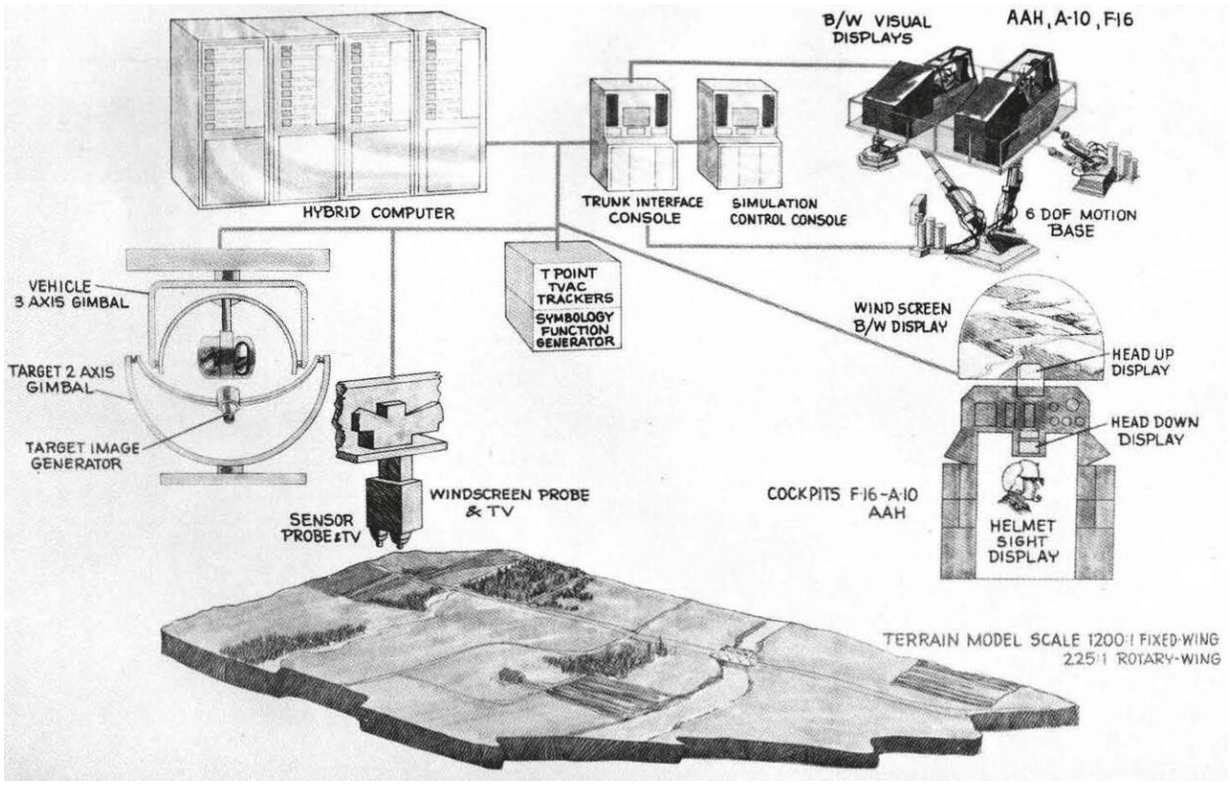


Figure 15.88. Flight simulation with terrain model (see [→ Baker 1978, p. 9])



Figure 15.89. Terrain model (see [→BAKER 1978, p. 10])

An interesting electronic terrain generator for use in flight simulator setups is described in [→General Electric 1959/1] and [→General Electric 1959/2]. →Figure 15.90 shows an example display and the overall pattern generator circuitry. A typical scene generated by such a system can be seen in →figure 15.91.

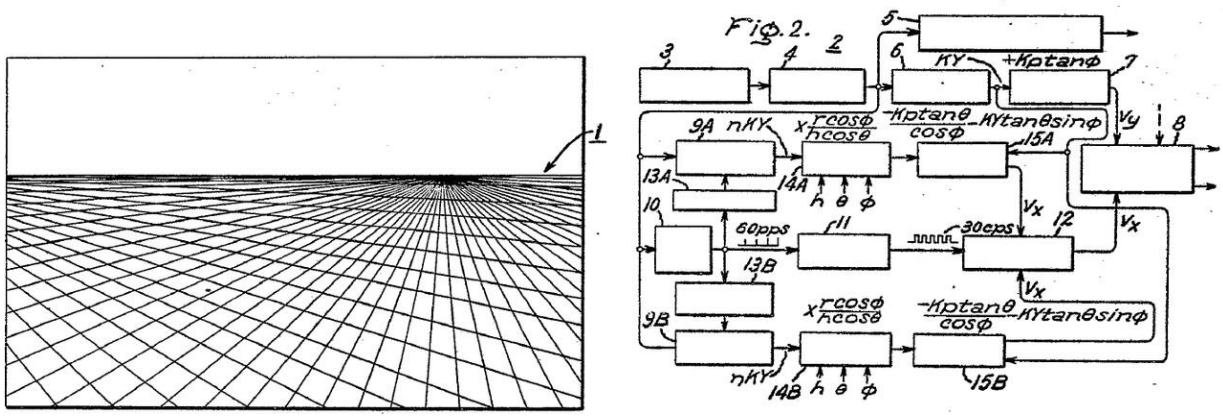


Figure 15.90. Terrain display generated by electronic circuits only (see [→General Electric 1959/1] and [→General Electric 1959/2])



Figure 15.91. Practical example of an electronic terrain simulation output

[→ Sherman et al. 1958] describe the simulation and analysis of the exit phase of flight of a hypersonic aircraft. A very simple analog simulation

of the flight of a paper glider is described in [→ Ulmann 2020/2], which is based on [→ Lanchester 1908] and [→ Simanca et al. 2002].

15.15.7 Airborne simulators

Even more fascinating than these ground-based flight simulators are *airborne simulators*, also known as *in-flight simulations*. Here, an analog or hybrid computer aboard an aircraft was used to actively change the behavior of the plane in a way that made the simulation of a wide variety of other aircraft possible.[337] Such simulators are also known as *variable-stability aircraft*.

A basic requirement for such a simulator is an aircraft with a fly-by-wire system. The signals generated by the pilot controls in the cockpit are then fed to the analog computer, which in turn generates the necessary output signals to control the aircraft's actuator systems. Normally two cockpits and two pilots were used with one set of controls wired directly to the plane's control motors, etc., while the second set of controls delivered the input signals to the analog computer.

One of the earliest such systems is the *General Purpose Airborne Simulator* (*GPAS*), which was based on a *Lockheed JetStar* aircraft. Development of this system began in 1960[338] and many studies were performed with this particular airborne simulator.[339]

Airborne simulators were not without risks as one of the GPAS programmers, Bob Kempel, remembers:[340]

> "*I remember the incident when we were airborne and we [Larry Caw and Bob Kempel] were looking at different feedback schemes. […] Well, as you know, signs were sometimes confusing. Fitzhugh Fulton*[341] *was the pilot. The sign on beta was wrong, and we ended up with a dynamically unstable airplane because of it. We turned on the system for Fitz to evaluate, and the airplane immediately began an oscillatory divergence! Larry and I were in the back hollering to Fitz to turn it off, but Fitz was intrigued with the thing so he wanted to watch it as it diverged or maybe just teach us a lesson. He finally punched the thing off and Larry and I sighed in relief. Larry changed the beta-input sign, and we proceeded with the test. […] The JetStar was a fun airplane to fly in, but I always had a feeling of impending doom or something else going wrong.*"

The vibrations were so violent that an observer in a chase plane worried about an impending crash: "*Stan told me […] he just looked out of the*

window to see where they would crash as he believed the wings would be torn off."[342]

15.15.8 Guidance and control

Analog computers and computing elements were frequently used as integral elements of guidance and control systems aboard aircraft and missiles, with the A4 rocket being the first rocket to be controlled by a fully electronic analog onboard computer.

An example of a mechanical analog subsystem is shown in →figure 15.92. It is a mechanical resolver module used in the *PHI-4* dead reckoning computer of the *Starfighter*. The *resolver assembly* basically contains a rotating ball, which splits a flight velocity signal into *x*- and *y*-components, both which are then fed to further subsystems of the onboard computer.

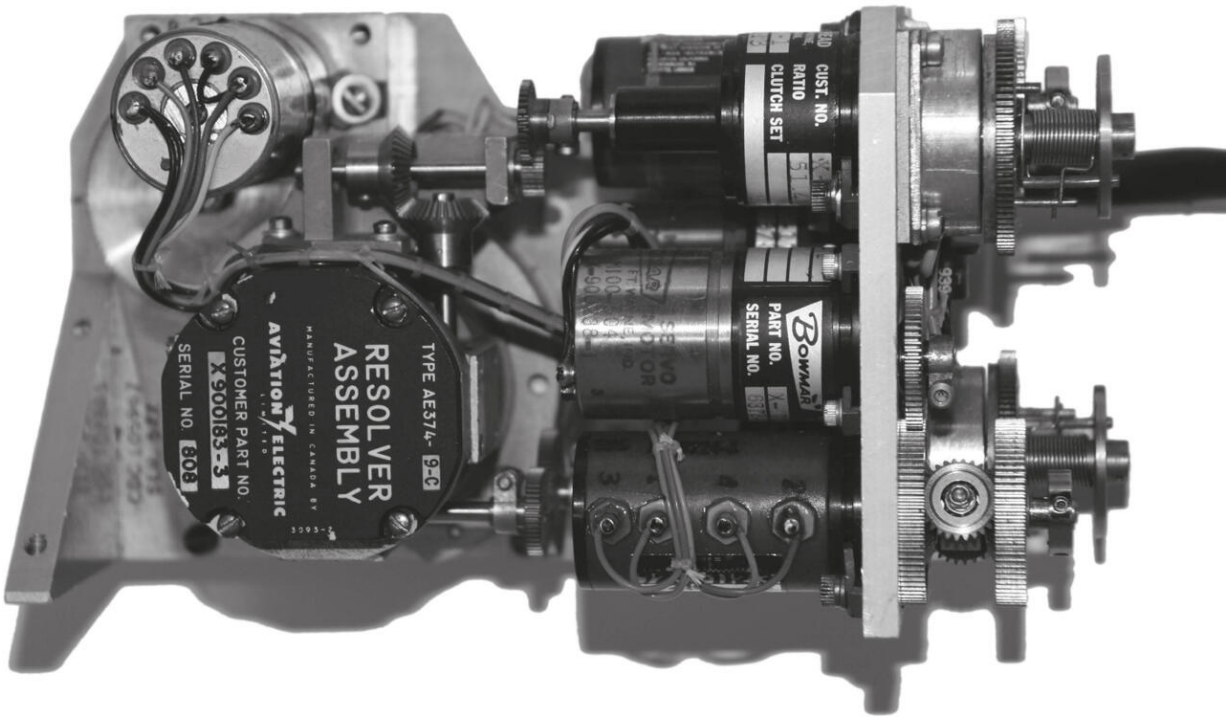


Figure 15.92. Resolver module of the PHI-4 navigation system that was used in the Starfighter

One of the most influential analog computer based missile guidance systems was developed for *Nike* missiles. Due to space constraints, a *command guidance system* was developed. At its heart was a large ground based analog computer fed with target data from various radar sources. The system then generated steering commands for the missile.[343] This approach not only made the missiles lighter and cheaper but also allowed for improvements of the guidance system without requiring modifications to already deployed missiles.

The analog computer computed a course, which was significantly better than a plain pursuit curve, thus rendering evasive maneuvers of the target ineffective.[344] Such a control system requires some kind of memory to store past movements of the target and the missile. Remarkably, the Nike analog computer used differentiators with large time-scale factors to implement these memory functions.[345]

In contrast, the *Polaris* missile, whose development started in 1956, employed an onboard guidance system based on a DDA.[346] These missiles were launched from ballistic missile submarines, thus a self-contained onboard guidance system was required as no command guidance system could be implemented for a submerged submarine. The required high accuracy of the missile system further forced the development of a digital guidance system. Since the researchers at MIT's *Instrumentation Laboratory* mostly had an analog computer background, the decision to develop a DDA was obvious.[347]

The resulting bit-serial DDA featured a word length of 17 bit corresponding to a resolution of about 10^{-5} and twelve words of memory implemented as shift registers. The first incarnation of this DDA had a volume of 11,000 cm^3, contained about 500 NOR-gates[348] implemented with germanium transistors, and required about 80 W of electrical power. A later implementation based on silicon transistors and an improved module packaging technology only required a quarter of this volume and only about half of the electrical power.[349]

The trigonometric operations performed by this onboard computer for steering the Polaris missiles were based on *CORDIC*,[350] an algorithm developed by Jack E. Volder.[351] This algorithm was subsequently extensively used in early scientific pocket calculators such as the *HP-35*,

introduced by Hewlett-Packard in 1972. It is still widely used to compute trigonometric functions and logarithms on digital computers.

Another *High Speed Differential Analyzer* (HSDDA) for use in airborne systems was developed in the early 1960s by the Guidance and Control Systems Division of *Litton Systems, Inc.*, and is described in detail in [→Litton 1963].

15.15.9 Miscellaneous

The applications described in the preceding sections are not exhaustive – there were many more – sometimes quite arcane – areas of application of analog and hybrid computers in aerospace engineering. An example of this is a study on parachute stability using analog as well as digital computers, which is described in [→Ludwig 1966]. This paper focuses on the strongly nonlinear oscillations induced by parachute systems.

15.16 Rocketry

The development of rockets offered an equally fertile ground for the application of large analog and hybrid computer installations as the following sections show.

15.16.1 Rocket motor simulation

Liquid-propellant rocket motors exhibit an extremely complex dynamic behavior that is impossible to study using purely analytic methods. The particular value of (analog) computer simulations for the development of rocket motors has been described by [→Szuch et al. 1965, p. 2] as follows:

> "*A computer simulation, when properly used, can be a powerful tool in guiding an engine development program. It provides an easy and economical means for evaluating various design approaches and forewarns the designer of possible problem areas before costly hardware is developed and subsequently scrapped. Once a qualitative design is established, the system may be 'tuned' for optimum performance by varying the system parameters about their design values.*"

This study describes the simulation of the *M-1* LH_2/LOX-rocket motor,[352] which was intended to be used in the *NOVA* rocket and was eventually

superseded by the highly successful family of *Saturn* rockets. This study is quite remarkable since apart from the analog computer, a digital computer was used to study details of the dynamic behavior of the rocket motor that could not be treated directly on the analog computer due to its limited precision.

The analog part of this simulation required about 250 operational amplifiers, 50 multipliers, 20 variable, and five fixed diode function generators, demonstrating the complexity of the problem. It turned out that one of the most useful features of the analog computer was its ability to study short-lived effects during engine startup through time-scaling.

Another interesting source is [→Fox et al. 1969], which describes an analog computer study of low-frequency fluid dynamics encountered in nuclear rockets, while [→Hart et al. 1967] focus on frequency responses and transfer functions of such rockets. [→Norum et al. 1962] describe the simulation of particle trajectories in fluid flow governed by Laplace's equation with a detailed error analysis.

15.16.2 Rocket simulation

The requirements for the simulation and analysis of the dynamic behavior of rockets are quite similar to those encountered in aircraft flight simulations, although normally there is no pilot in the loop in a rocket simulation. The basics of such simulations are described in [→Jackson 1960, pp. 390 ff.]. The number of computing elements necessary for such simulations is as equally impressive as for aircraft flight simulators. An early rocket simulation performed on a *REAC* analog computer required 304 operational amplifiers, 369 coefficient potentiometers, 25 servo multipliers, eight resolvers, eight diode function generators, and two random signal generators.[353] The following quotation from [→Bilstein 2003, p. 72] emphasizes these vast hardware requirements:

> "*For modifications and installation of new equipment, MSFC spent over $ 2 000 000 after acquiring the site in the summer of 1962. The array of digital and analog computers for test, checkout, simulation and engineering studies made it one of the largest computer installations in the country.*"

The advantage of high speed operation of analog computers, compared with their contemporary program digital computer counterparts, has been noted by [→Biggs[a], p. 6]:

> "*Because of the random nature of some of the missile system inputs the prediction or extrapolation work may require a large number of runs. AGWAC should compute these runs in real time or something like real time, whereas fast digital machines at present available would be at least one hundred times slower.*"[354]

The setup of an early laboratory for three-dimensional guided missile simulation is described by [→Bauer 1953]. This development was the objective of *Project Cyclone* at Reeves Instrument Corporation. The simulation system was put into operation in 1952 and resulted in the development of the first analog computer, the REAC, in 1948 of which more than sixty were sold by 1951 at ≈ 40 US sites. 13 of these were part of the Cyclone simulation system, each containing seven integrators, seven summers, six inverters, and 23 coefficient potentiometers. In addition to this, 14 servo units containing a mix of servo multipliers and servo function generators, and three cabinets for additional devices like uncommitted operational amplifiers, limiters, etc., were used. Most of this installation is shown in →figure 15.93. It was described in the original press release as follows:[355]

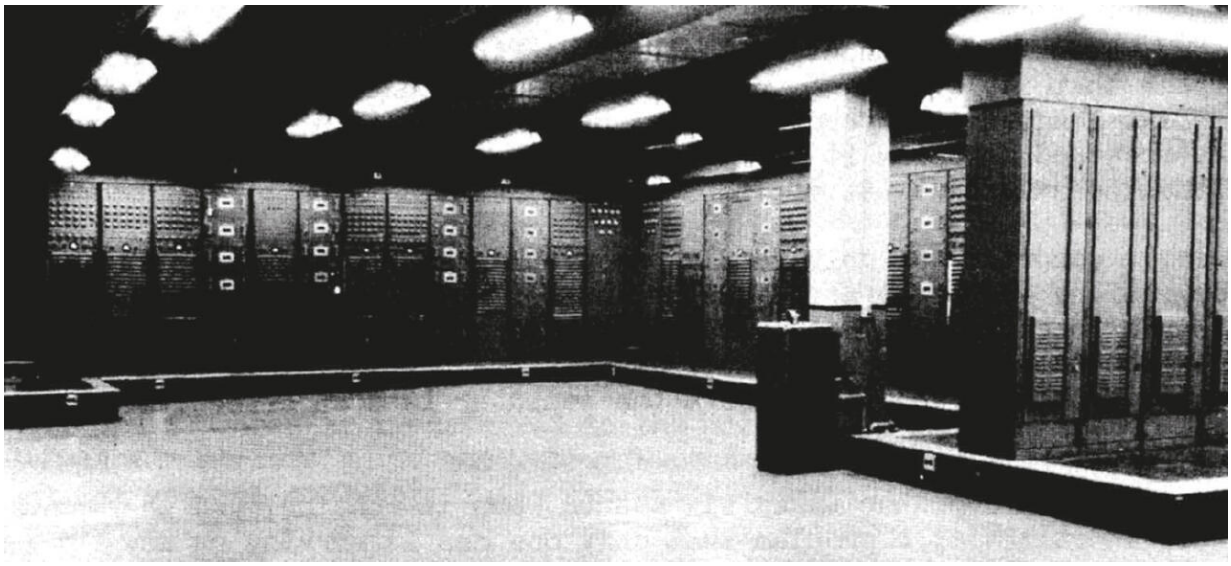


Figure 15.93. Laboratory for three-dimensional guided missile simulation (see [→ Bauer 1953, p. 195])

> "*The Navy revealed a guided missile 'launching site' right in the heart of New York city today – but the launchings are all on paper and nobody gets hurt. The hitherto top secret research program – 'Project Cyclone' – is centered in what has become known as 'the house of 91st street'. There is a sort of brainlike machine [which] is fed flight problems and comes up with mathematical computations. From these figures, scientists can tell what a certain type of guided missile can do in flight – even though the missile doesn't exist except on paper. With the paper work out of the way, exact specifications and plans are read and the missiles can be built. It's a big money-saver, the Navy said. It would cost millions to build the missiles, try them out and then correct their shortcomings. It would take a lot of time, too. The Reeves Instrument Corp., a subsidiary of Claude Neon, Inc., created the computing center in 1946 at the direction of the Navy. Capt. J. R. Ruhsenberger, a naval research expert, said project Cyclone has given the United States a big head start in the development of guided missiles. 'We are not tipping our hand to our enemies by unveiling project Cyclone', he told newsmen. 'We are showing the American taxpayer through you what has been going on these past 'peacetime' years.*"

A comprehensive description of typical large-scale problems simulated at project Cyclone can be found in [→ Bauer et al. 1957]. Some of the conclusions of this report were:[356]

> "*The feasibility of solving full, three-dimensional problems has been established. Accuracy has not been impaired in problems requiring as many as 300 amplifiers. The time spent in maintenance, trouble-shooting and checking is not disproportionate to the size of the problem.*"

Another enthusiastic review was given by [→Merems 1951] under the catchy title "*Project Cyclone May Foreshadow Trips to the Moon*", showing the importance and public perception of these early large scale forays into analog computation during the Cold War.

A comprehensive treatment of mathematical vs. physical simulations can be found in [→Hintze 1957]. [→Elfers 1957] gives details on the simulation of spacecraft, eventually leading to the first US satellites in orbit. Putting the first successful US satellite, *Explorer I*, into orbit required the attitude control system of the *Jupiter C* rocket to tilt the rocket gradually into the required injection attitude. [→Hosenthien et al. 1962, pp. 454 ff.] describe the attitude control system simulations performed for this project.

Part of a large-scale analog computer system for the simulation of the dynamic behavior of the *Saturn V* rocket is shown in →figure 15.94. This *General Purpose Simulator* (*GPS*) was used to implement a twelve degree of freedom simulation of the first stage of a Saturn V rocket, taking effects such as dynamic winds, bending of the rocket structure, and fuel sloshing into account.

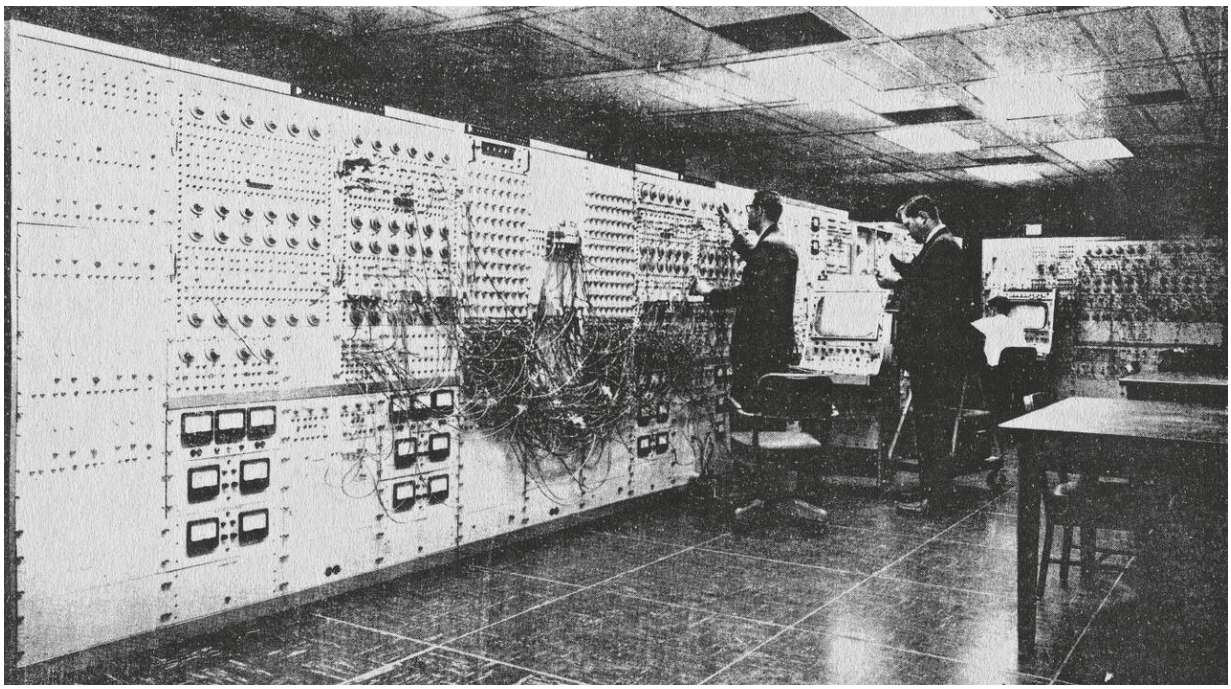


Figure 15.94. The GPS-system, used to simulate the Saturn V rocket's dynamic behavior ([→Teuber 1964, p. 26])

Using real wind data stored on analog tape drives, this analog computer could simulate the performance of a S-IC first stage up to 3 000 times faster than real-time. The system contained 50 integrators, 50 summers, 350 coefficient potentiometers, 20 quarter square multipliers, and 15 function generators, each containing an additional 70 operational amplifiers. To make simulations with stochastic inputs possible, the system also featured random generators, high- and low-pass filters, etc. It allowed 50 complete flight simulations per second, making it possible to generate a flicker-free display of the solutions on oscilloscopes.[357]

The simulation of a two-stage satellite launch vehicle on an analog computer is described in [→Jackson 1960, pp. 261 ff.]. The focus of this study is primarily on possible abort scenarios. The possibility of a piloted rocket flight was analyzed in a study described in [→Waltman 2000, pp. 34 f.]. A centrifuge was used in these simulations to exert gravitational forces of up to 14 G[358] on the pilot while studying his performance in controlling a simulated rocket in various adverse flight regimes.

[→Kraft et al. 2002, p. 209] remember that these simulations of the dynamic behavior of rockets and associated systems also resulted in frequent changes to the systems based on results of these simulations:

> "*When we first used the word* dynamic *to describe the simulators, it meant that they acted pretty much like the real thing. But in practice* dynamic *also applied to design because we discovered things in simulations that needed to be changed.*"

15.16.3 Real-time data analysis

Another interesting application of analog computers in the development of guided missiles is the real-time analysis of flight data transmitted in-flight by telemetry systems from a missile to a ground station. The need for such a system was recognized in the late 1940s which led to a sophisticated system consisting of two *UNIVAC 1103A* digital computers, a three-axis flight table, and several analog computers. This system and its purpose is described in [→Steinhoff et al. 1957]:[359]

> "*The function of the loop [consisting of the aforementioned digital and analog computers] is to perform the analysis of the missile behavior by comparison of the actual behavior with the simulated behavior. If one has been completely accurate in setting up his missile simulation then the performance of the actual missile and the simulated*

missile would be identical. This perfect match is highly unlikely, so the analysis task is that of making adjustments in the simulated missile so as to effect simulated performance matching with the actual performance [...] "

15.16.4 Spacecraft maneuvers

Although the limited precision of electronic analog computers places severe constraints on orbital simulations and similar calculations, analog simulations were nevertheless used to obtain at least rough estimates for orbital elements, etc. If the orbit radii are restricted to a small range and if the number of circuits is small (less then 10 e. g.), solutions obtained by analog computers are often sufficiently exact to gain valuable insight into spacecraft maneuvers.[360]

[→GLATZEL et al. 1967] compare a variety of digital and hybrid approaches for the simulation of the behavior of satellites. This includes hardware-in-the-loop approaches such as the use of actual damping rods, a scaled down satellite model in a chamber with *Helmholtz* coils for the simulation of earth's magnetic field, and many more.

Basic information on such simulations can be found in [→SCHWEIZER 1976/2], while [→Telefunken/7] describes the computation of the orbital elements for the early passive satellite *Echo-1*. A trajectory optimization for a Mars mission is described in [→GILOI 1975, pp. 161 ff.]. The much more complicated fine-positioning of a geostationary satellite is simulated on an analog computer and described in [→Telefunken/6]. →Figure 15.95 shows the simulation results of the final positioning phase of the satellite.

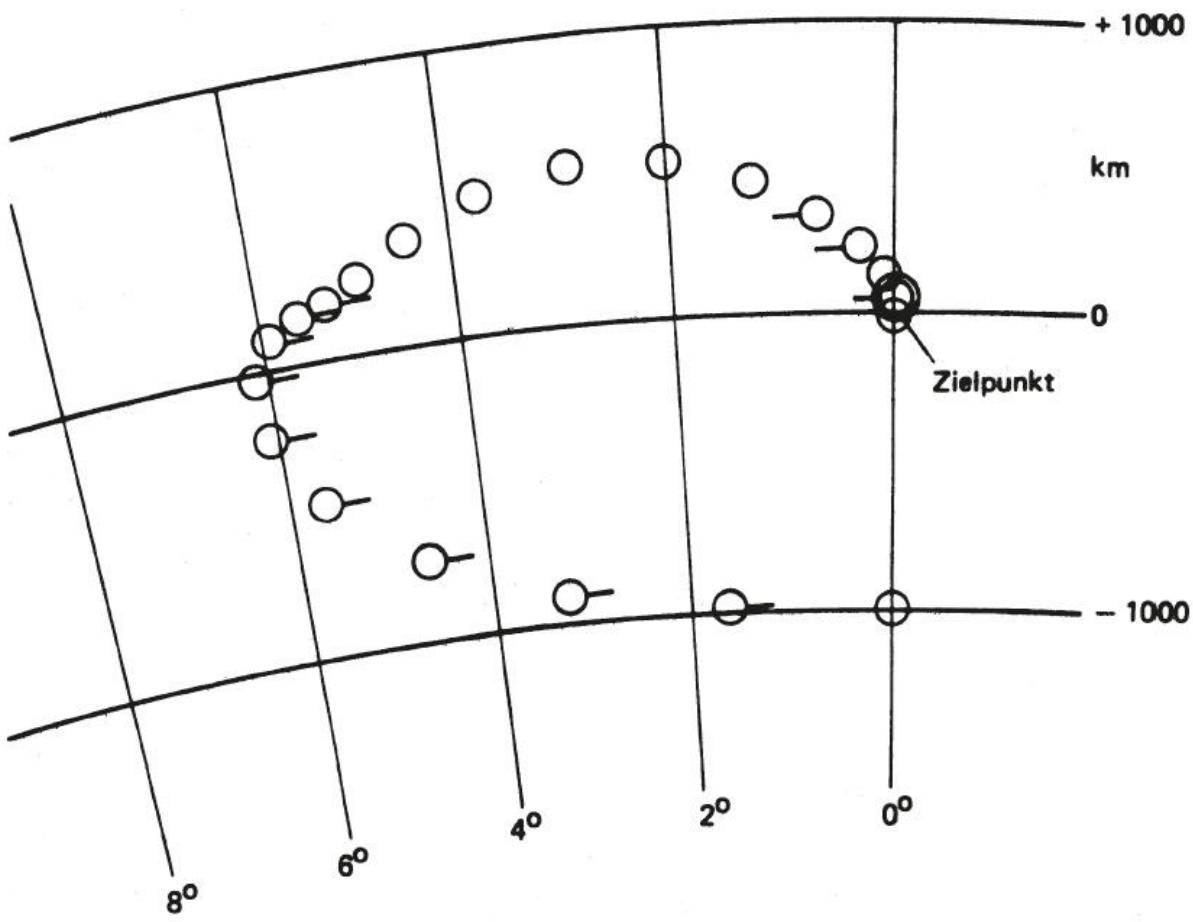


Figure 15.95. Positioning of a satellite into a geostationary orbit (see [→Telefunken/6])

[→Fogarty et al. 1963] describe an analog orbital flight simulation system. This approach is particularly noteworthy as the severe scaling problems due to the wide range of variables involved are solved by introducing a novel axis system. This also makes it possible to not only simulate the orbital flight regime but also re-entry maneuvers involving lifting vehicles. All in all five different coordinate systems are used, the earth centered inertial frame (*I-frame*), the navigational frame with respect to the earth's surface (*N-frame*), the body axes with their origin in the center of gravity of the satellite (*B-frame*), the *Euler* angles (*E-frame*), and, finally, the modified flight path (*H-frame*). The simulation of re-entry uses a number of function generators for air density, etc.

[→Reinel 1976] describes the analog simulation of the onboard attitude control system of the European communication satellite *Symphony*. This simulation required a flight table as well as a sun and earth simulator to generate proper input signals to the satellite's sensor elements, which were part of the simulation setup.

A comparative study regarding the application of a *reaction wheel* for attitude control of a satellite was performed on a hybrid computer as well as on several digital computers and is described in [→Grierson et al.]. While the hybrid computer was able to solve the equations in real-time, a *Cray-1* vector processor, the fastest digital computer of its day, only achieved a tenth of this speed.[361]

[→Krause 1964] describes the simulation of the dynamic behavior of a rotating space station with astronauts on board. To compensate for oscillations in roll direction induced by movements of the astronauts, control systems using flywheels or even "*corrective motion of the astronauts*"[362] were suggested and extensively simulated. The optimization of an attitude control servo loop for a satellite using magnetorquers is described in [→Bismut et al. 1967].

A comprehensive study of large angle attitude maneuvers of bodies in space can be found in [→Krause et al. 1963]. Attitude maneuvers like this are commonplace in rendezvous operations, preparation of re-entry of a spacecraft, soft landing on the moon, midcourse corrections, etc. This study also takes stability issues of an asymmetric spacecraft into account.

15.16.5 Mercury, Gemini, and Apollo

Analog and hybrid computers played a central role in the *Mercury, Gemini*, and *Apollo* space flight programs of the United States. Apart from applications like those described above, sophisticated training simulators were built to gain an understanding of the special requirements and problems posed by manned space vehicles. →Figure 15.96 shows such a simulation system, which was used in the early phases of the Mercury program to develop the control stick used in the Mercury capsules. A complex mission simulator for the Mercury project, based on an EAI 231R analog computer,[363] was developed at Bell Laboratories.

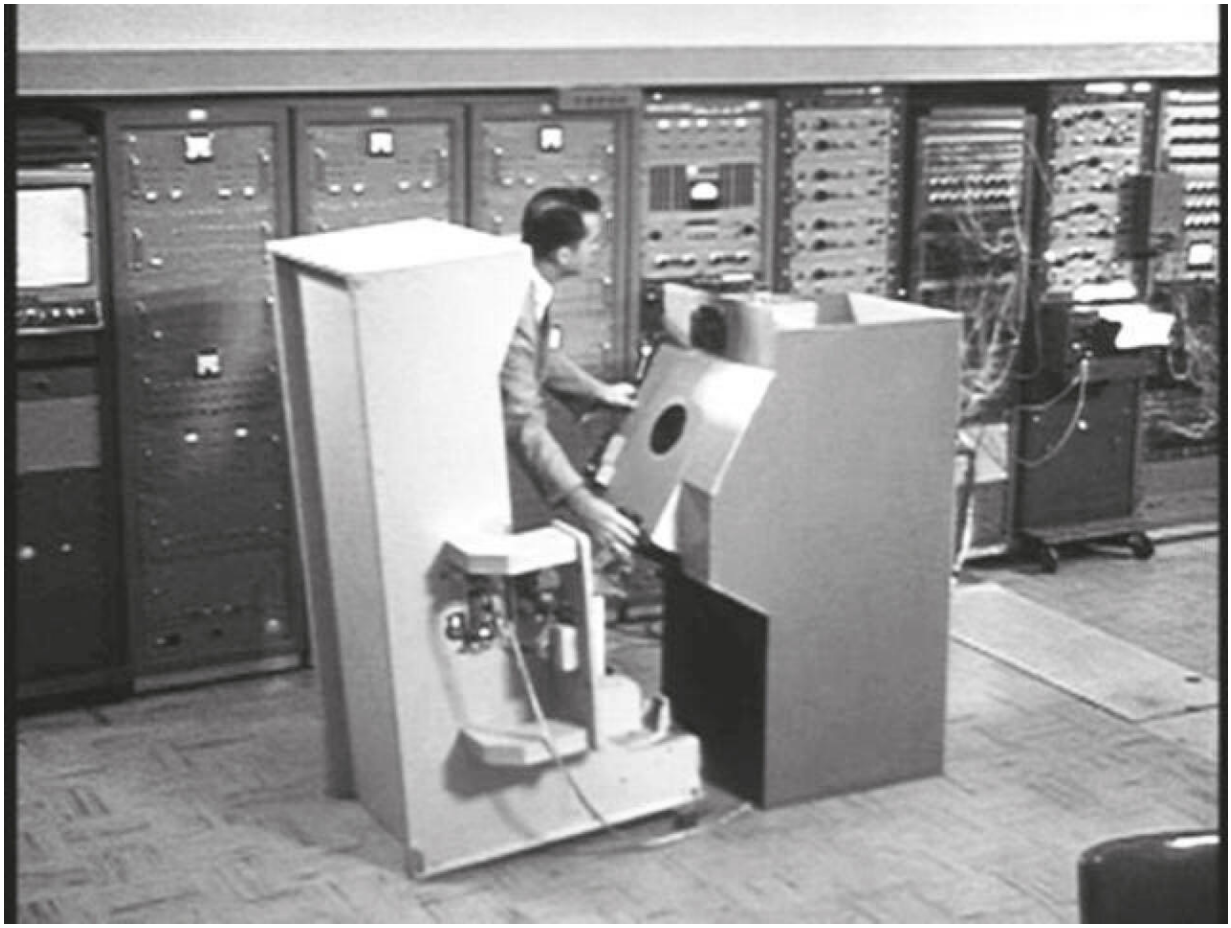


Figure 15.96. Analog computer setup used during the development of the Mercury control stick (author's archive)

→Figure 15.97 shows an EAI 16-31R analog computer used in a flight simulation study for project Mercury. Especially interesting is the hardware-in-the-loop approach using actual flight hardware visible in the foreground.

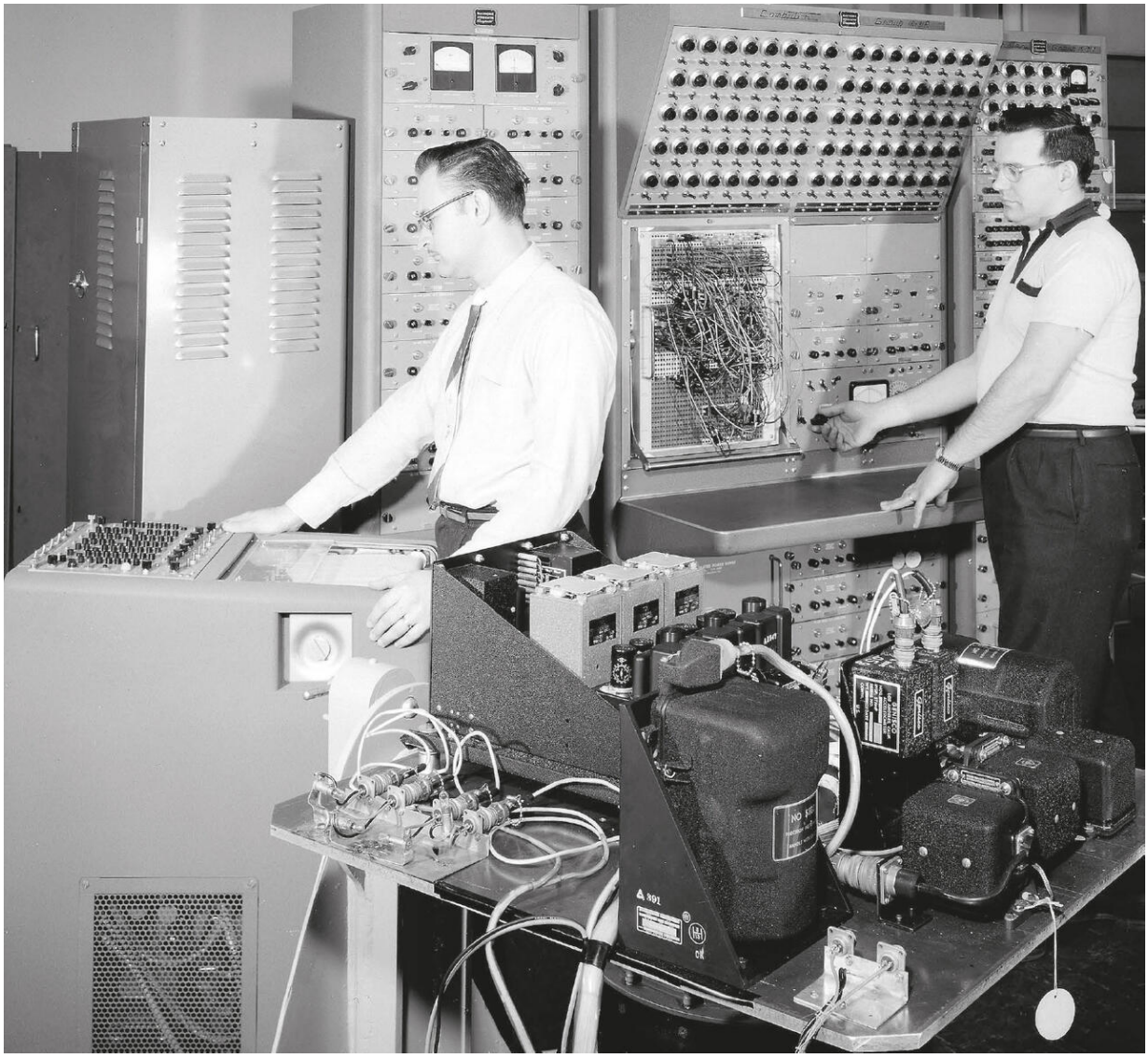


Figure 15.97. EAI 16-31R analog computer used at NASA Lewis Research Center in 1959 for simulating flight maneuvers during project Mercury, 1959

Following Project Mercury, one of the main goals of Project Gemini was to work out procedures for docking spacecraft in orbit, a central requirement for the subsequent Apollo moon landing project. Simple rendezvous simulations on analog computers were performed as early as in the late 1950s.[364] The importance of such simulations cannot be overestimated as [→Hansen 2005, p. 248] explains:

> "*Without extensive simulator time, it is doubtful that any astronaut could ever have been truly ready to perform a space rendezvous. 'Rendezvous simulation in Gemini was really quite good', Armstrong notes. 'We achieved fifty to sixty rendezvous simulations on the ground, about two-thirds of which were with some sort of emergency.'*"

The simulation of a pilot controlled rendezvous maneuver is described in detail in [→Brissenden et al. 1961]. This study not only focused on the performance of the pilot during the maneuver but also helped developing suitable display instruments for use during the piloted approach. In addition to this, boundary conditions of trajectory errors, which could make a successful rendezvous maneuver impossible were determined. A highly complex rendezvous simulation system was put into operation at the *TRW Systems Group*[365] in the 1960s. This system featured a TV camera motion unit similar to that used in the terrain generator described in →section 15.15.6.[366] [→Fox et al. 1963] describe a six degree-of-freedom feasibility study of a manned orbital docking system. This study, too, has its focus on pilot performance.

A comprehensive six-degrees-of-freedom simulation of Gemini re-entry is described in [→NASA 1963]. Here, pilots performed a number of simulated reentries using different modes of command input, rate command, and direct control, yielding recommendations for indicator and control systems. A comparable hybrid simulation system was developed by *McDonnell Aircraft Corporation* and *IBM*, which was used for re-entry simulations during Project Gemini.[367]

→Figure 15.98 shows an EAI 231R analog computer used for training purposes during project Apollo. Standing next to the analog computer is engineer Bill Summers,[368] with test pilot Jim O'Neil sitting in an early pressure suit in front of the system.



Figure 15.98. EAI 231R analog computer used for training purposes during the Apollo program (see [→ Splaver 1967])

A nice simulation of an energy optimal satellite rendezvous with a space station is described in [→ Breitenecker 1978/1]. The rendezvous itself consists of a primary approach under ground control and a final approach under local control. The resulting analog computer setup is quite large, requiring 19 integrators and a multitude of multipliers. A typical optimum trajectory resulting from this simulation is shown in →figure 15.99.

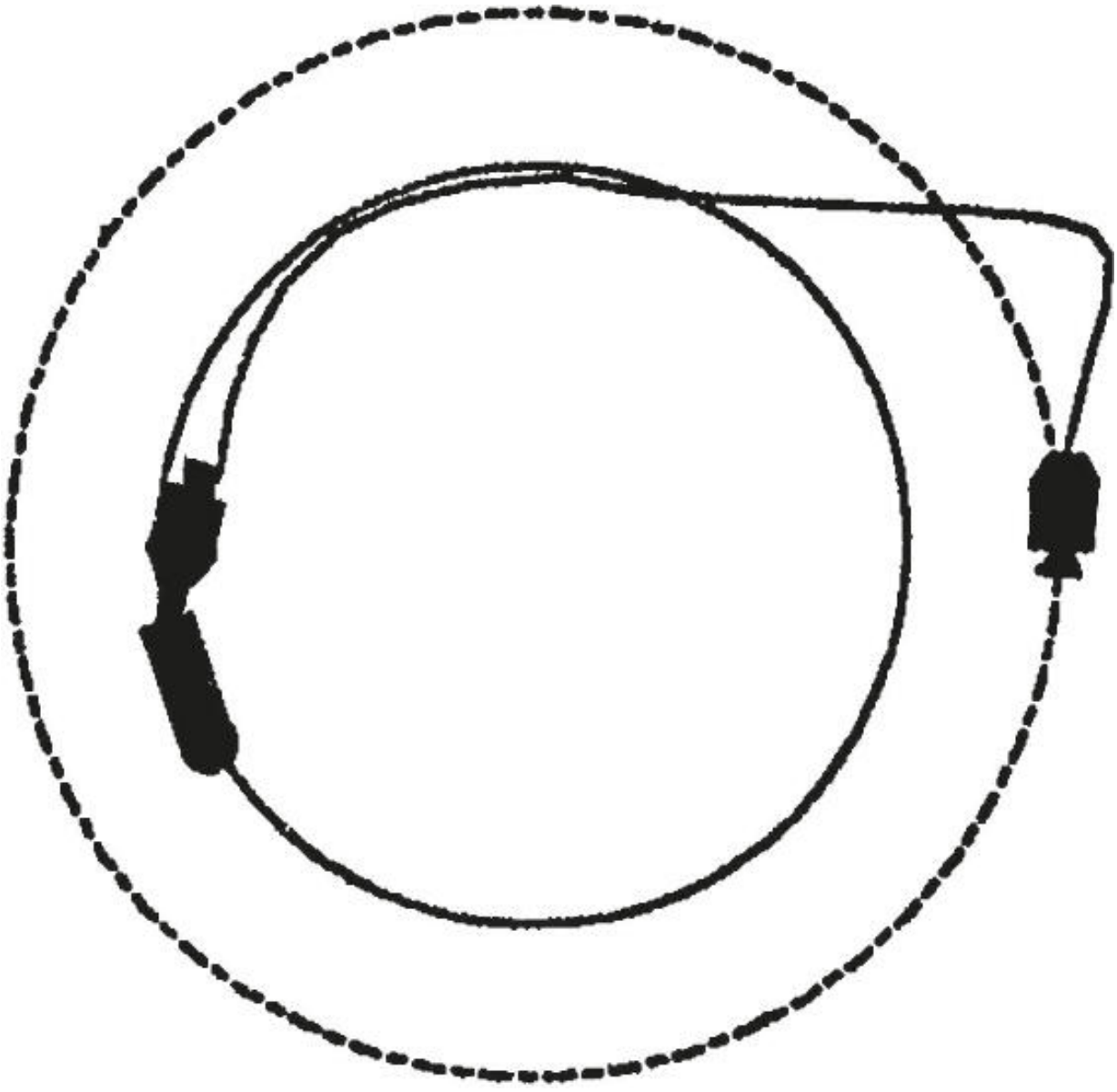


Figure 15.99. Energy optimized trajectory for a docking maneuver (see [→ Breitenecker 1978/1, p. 47])

An interesting in-flight simulator system was developed to train the pilots of the *Lunar Excursion Module* (*LEM*). This system, affectionately known as the *flying bedstead* due to its peculiar appearance, was in fact a flight vehicle mimicking the dynamic behavior of the LEM under the constraints of earth's gravitation. This was done by an onboard special purpose analog computer. The resulting system proved very valuable as a research and training device:[369]

> "*The notion of attacking the unique stability and control problems of a machine flying in the absence of an atmosphere, through an entirely different gravity field, 'That was a natural thing for us, because in-flight simulation was our thing at Edwards', Armstrong relates. 'We did lots and lots of in-flight simulations, trying to duplicate other vehicles, or duplicate trajectories, making something fly like something else.'*"

Development work of the LEM at Grumman started in January 1963. At first, purely analog and purely digital simulations were tried. Quickly, it became clear that a true hybrid computer setup instead of distinct systems would be required since the simulation complexity exceeded the capabilities of each one technology by far. A number of trajectory parameters would require a resolution of 10^{-5}, rendering a purely analog simulation impossible. On the other hand, the spacecraft and orbital dynamics proved to be much too complicated for a digital only approach. Eventually, a hybrid system consisting of an *IBM 7094-II* digital computer and three *Reeves 500* analog computers using an *Adage 770* computer link system containing 48 ADCs, 35 sample and hold amplifiers, and 55 DACs was employed. Typical studies performed on this large system were the actual moon landing process, the return from the moon, as well as a number of abort studies.[370]

15.17 Military applications

Although analog and hybrid computer based flight and rocket simulations were used extensively in a military context, too, there were also problems unique to military applications that were tackled using analog simulation techniques. An interesting example can be found in [→ EAI 1966, pp. 13 ff.], which describes the simulation of a setback leaf system used as a safety mechanism in a projectile fuse. This mechanism

arms the fuse only after a sufficient acceleration profile has been experienced by the projectile.

Studies of projectile trajectories are described in [→Korn et al. 1956, pp. 110 ff.] and [→Johnson 1963, pp. 175 ff.]. Of prime importance is the accurate representation of aerodynamic properties and forces acting on the projectile, often involving functions of more than one variable, which required a large number of computing elements.

The implementation of a simplified naval gunnery problem on an analog computer is covered by [→Wass 1955, pp. 57 ff.], while [→Demoyer 1980] describes a highly complex interactive anti-aircraft gun fire control simulation developed in the late 1970s.

The Cold War gave rise to the development of one of the earliest military computer games: *HUTSPIEL*. Developed in 1955, it is a

> "*theater-level war game [...] directed to the study of the effects on a defense of stabilized positions in Western Europe of various employments of tactical atomic weapons and conventional air support*,"[371]

implemented on several GEDA systems. Two combatants, *blue* representing NATO troops and *red* representing Soviet forces, face each other in HUTSPIEL. One minute of computer time corresponds to one day of battle action with a pause after each simulated day to give the opponents the chance to plan their next moves. The current states of blue and red troops is represented by about 40 individual variables each.

This remarkable system is described in detail in [→Clark et al. 1958].[372] →Figure 15.100 shows the overall setup during an actual run of the game.[373]

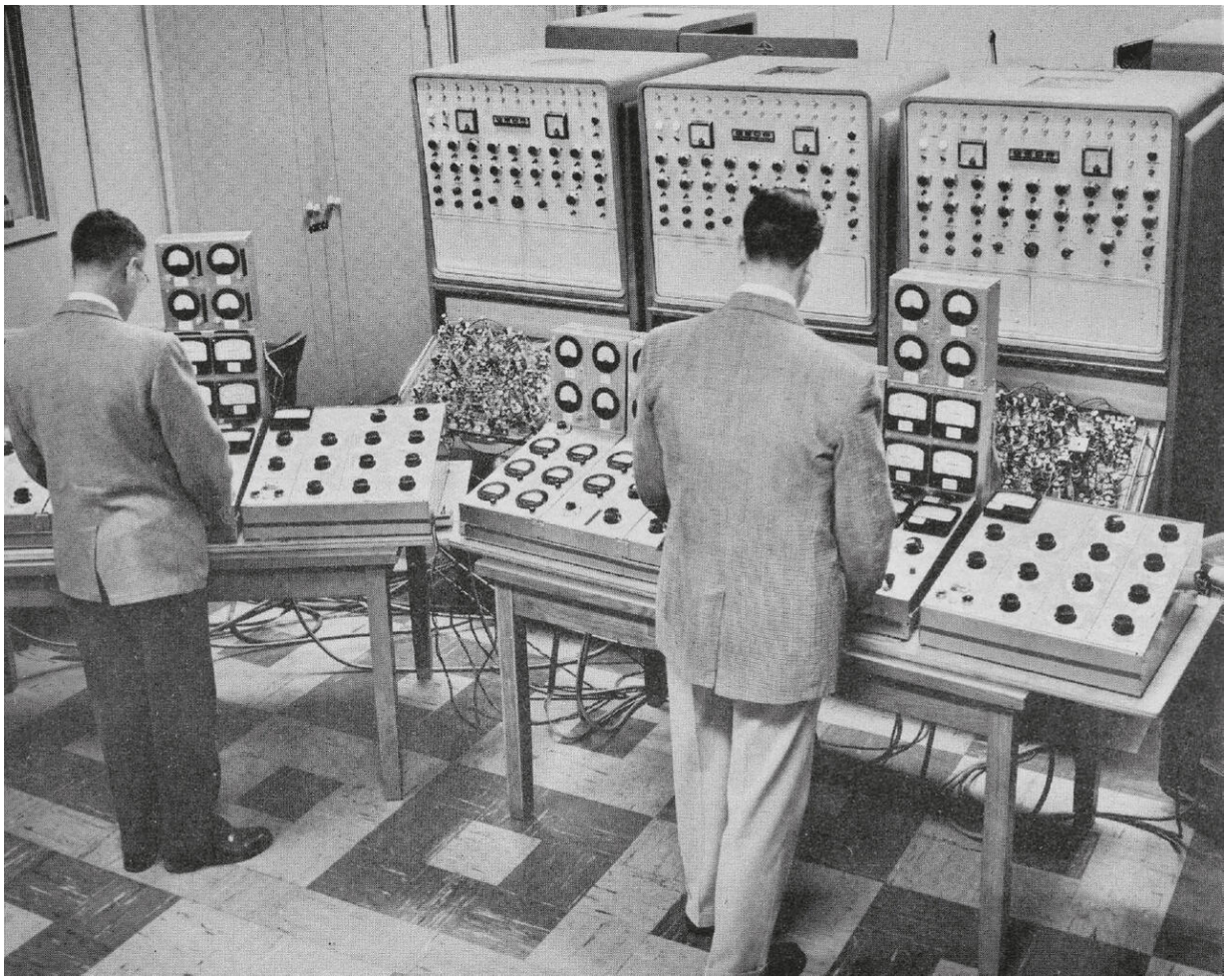


Figure 15.100. The HUTSPIEL system

15.18 Education

One of the main advantages of analog computers in education is their intuitive way of operation. Being mathematical machines at their heart, they avoid the additional abstraction layer of the algorithmic approach imposed by digital computers. This was first noted by Warren Weaver after the dismantling of the Rockefeller differential analyzer:[374]

> *"[I]t seems rather a pity not to have around such a place as MIT a really impressive Analogue computer; for there is vividness and directness of meaning of the electrical and mechanical processes involved [...] which can hardly fail, I would think, to have a very considerable educational value. A Digital Electronic computer is bound to be a somewhat abstract affair, in which the actual computational processes are fairly deeply submerged."*

A further advantage of analog computers is their unmatched degree of interactivity which allows students to gain insight into the behavior of complex dynamic systems in a very hands-on way.[375] →Figure 15.101 shows a small *Donner 30* analog computer in a typical educational setup. The problem under consideration is patched on the horizontal patch panel in the middle of the desk.[376]

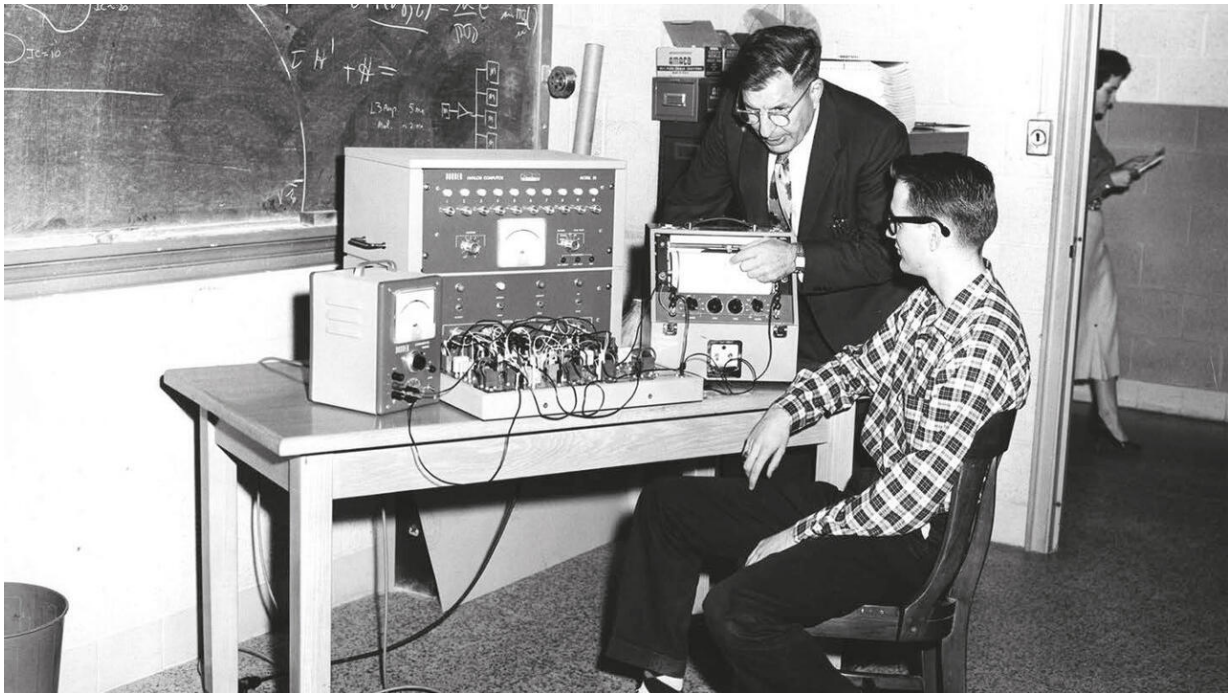


Figure 15.101. A vacuum tube based Donner 30 analog computer used in a typical educational setup (cf. [→NC STATE 2017, p. 1])

More general information about the application of analog computers in education can be found in [→Martin 1969] and [→Martin 1972]. Educational examples in mathematics are described in [→Dornier/8], [→Dornier/9], [→Rasfeld 1983], [→Blum 1982], and [→Dornier/10]. The simulation of mechanical systems is covered in [→Park et al. 1972] and [→Spiess 2005]. Educational applications in process control are the focus of [→Nise[a]] and [→Medkeff et al. 1955], while [→Tabbutt 1967] and [→Hamori 1972] describe some examples from chemistry. [→Müller 1986] contains a collection of physical problems and their solution by means of an analog computer. An interesting panel session on the application of

analog and hybrid computers in education can be found in [→Kovach 1967]. Another thorough exposition of analog computers in education is given in [→Stanley 1956] with a focus on the actual model building and transfer functions.

[→Tabbutt 1969] describes the plan for a time-shared analog computer to be used in science education. This system would have a number of graphical display terminals connected to it, each of which would serve an individual student who could experiment with parameter settings for a given science problem. This would make use of the high speed of an analog computer – the proposed system would be able to support up to 20 student terminals to be used at once without noticeable delay in the system's response. [→Martin 1970] describes the development of analog/hybrid terminals for use in a teaching environment.

A modern educational analog computer, *THE ANALOG THING*,[377] was introduced to the market in 2022 by anabrid GmbH, a German research company and manufacturer of analog computers. It comes with a booklet containing several introductory examples and is covered in more detail in →sections 6.6 and →A.[378]

15.19 Voice recognition

Already in the 1970s work on analog voice recognition was done. An analog program on an EAI 680 analog computer for feature extraction from voice signals is described in [→Gudonawitschius et al. 1977/1]. First, a pause detection is applied to the amplified microphone signal, so that individual words can be distinguished. Then voiced and unvoiced signal parts are discriminated. Based on this information alone, simple words can be easily distinguished.

[→Gudonawitschius et al. 1977/2] then describe and compare different feature extraction strategies for voice recognition using analog computers. Another approach, using a phase space representation of a voice signal $x(t)$ using pairs of time derivatives of $x(t)$ is described in [→Gudonawitschius et al. 1979]. Finally, a comparative study of various classification schemes is given in [→Gudonawitschius et al. 1979].

Today, the idea of analog voice recognition has gained some traction with the ubiquity of *always on devices* such as home control systems, etc. Using a digital recognition algorithm for the detection of a set of trigger words is quite energy intensive. Accordingly, interest in low-energy analog approaches for this problem has increased as described in [→Maxfield 2021].

15.20 Arts, entertainment, and music

Analog computers also found extensive applications in arts, entertainment, and music as shown in the following sections.

15.20.1 Arts

In the 1940s, Ben Laposky[379] developed the first analog computer solely intended for artistic purposes. In the late 1950s, the computer graphics pioneer John Whitney built an electromechanical analog computer from parts of an *M-5* antiaircraft director that had been used during World War II and was sold for scrap afterwards.[380]

Heinrich Heidersberger[381] was charged with the creation of a large wall painting for the School of Engineering in Wolfenbüttel (Germany).[382] His first ideas for this work were based on Lissajous figures.[383] These figures are described by

$$\begin{aligned} x &= a\ \sin(\alpha t + \phi) \text{and} \\ y &= b\ \cos(\beta t). \end{aligned}$$

Based on this idea, Heidersberger built a mechanical analog computer, the *Rhythmograph*, which could generate much more complex figures than plain Lissajous figures. →Figure 15.102 shows him working on the Rhythmograph. A typical picture generated by this setup can be seen in →figure 15.103. The device exposed photographic paper to a fine spot of light that was deflected by its swinging booms and mirrors. The Rhythmograph was itself a piece of art. After setting its parameters, which influenced the periods of the various mechanical oscillators, the

Rhythmograph was set into motion by electromagnets and then started to “paint” a picture by light.[384]



Figure 15.102. Heinrich Heidersberger working on the *Rhythmograph* (Archiv Nr. 9179/1, self-portrait, Wolfsburg 1962, reprinted with permission from Benjamin Heidersberger)

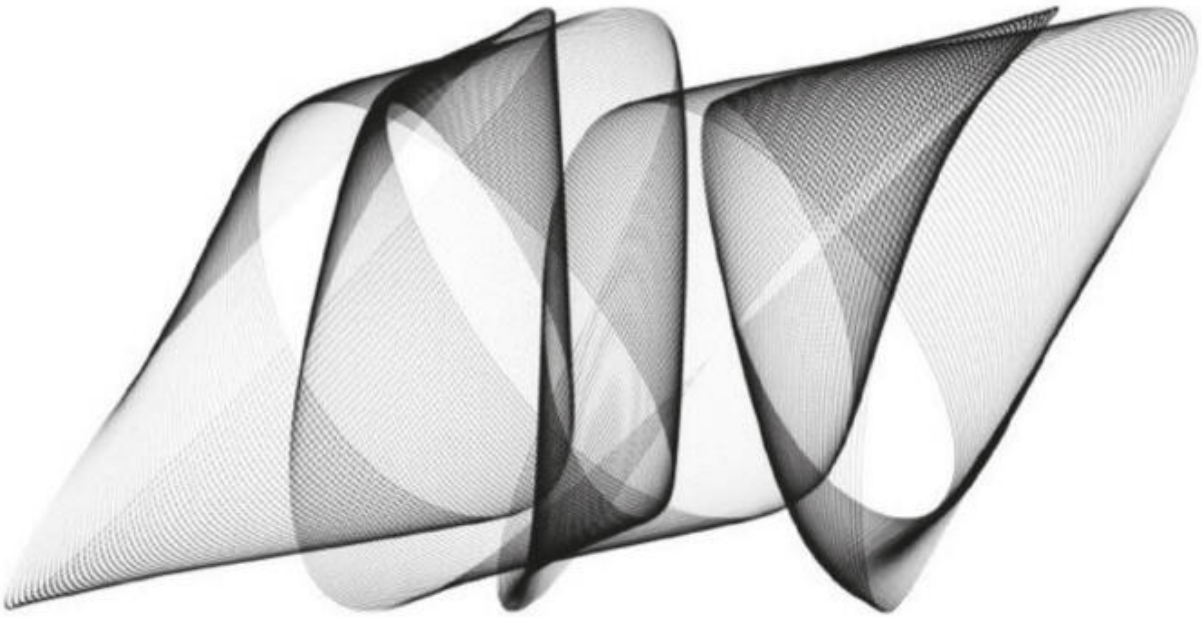


Figure 15.103. Rhythmogramm #229, *Arabeske*, 1950s/60s (reprinted with permission from Benjamin Heidersberger)

An electronic analog computer, built by Franz Raimann, was used in the early 1960s by the scientist, author, and artist Herbert W. Franke[385] to generate pictures such as the one shown in →figure 15.104 on an oscilloscope screen. In contrast to Heidersberger's Rhythmograph, this analog computer offered the high degree of interactivity that is characteristic of electronic analog computers, giving the artist more freedom to experiment with the mathematical parameters of an artwork. →Figure 15.105 shows Herbert W. Franke working with this analog computer.[386] Another example of Franke's analog computer art can be found in [→Computers and Automation 1975, p. 14].

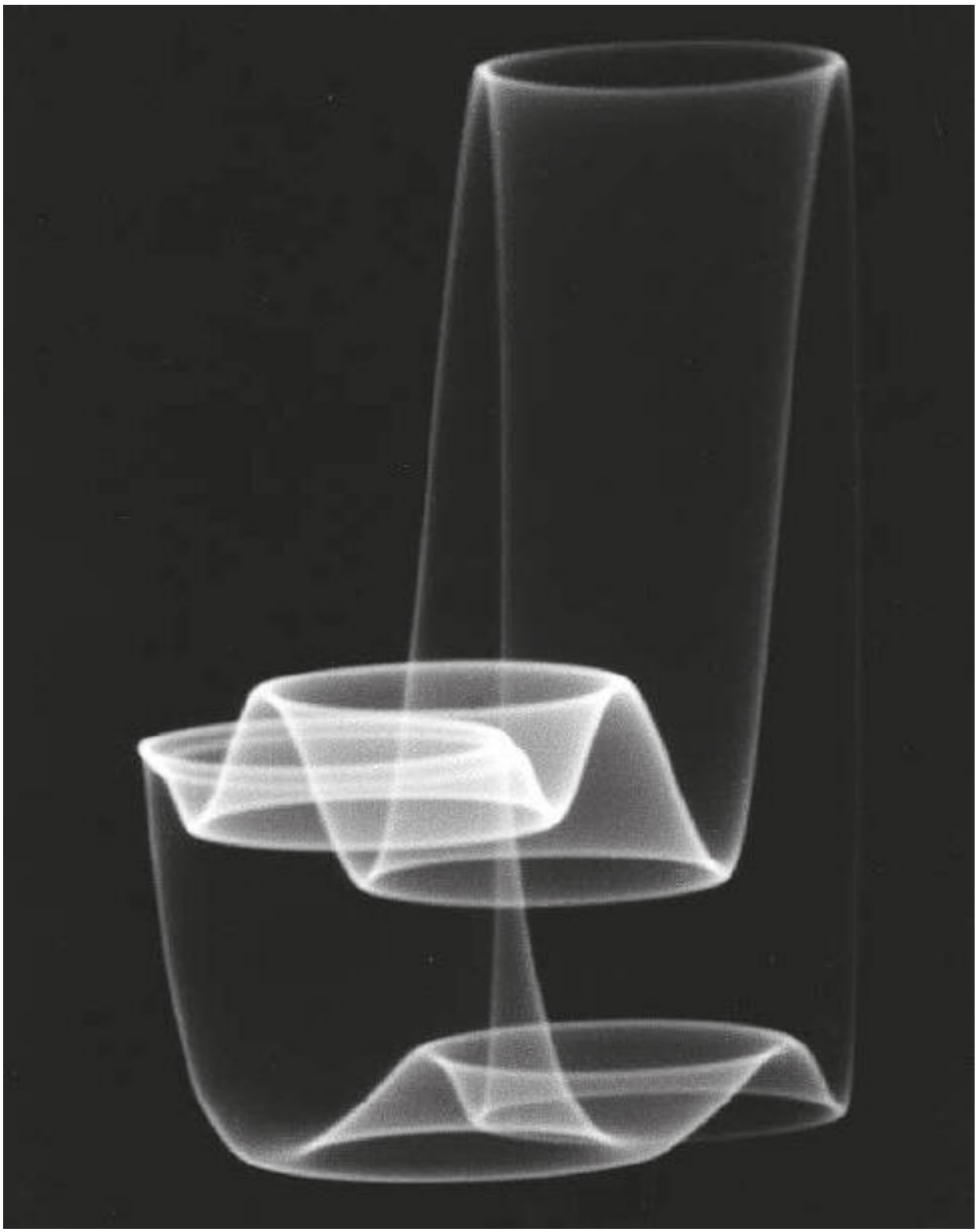


Figure 15.104. "Dance of the electrons" – an artwork of Herbert W. Franke, 1961/1962 (reprinted with permission of the artist)

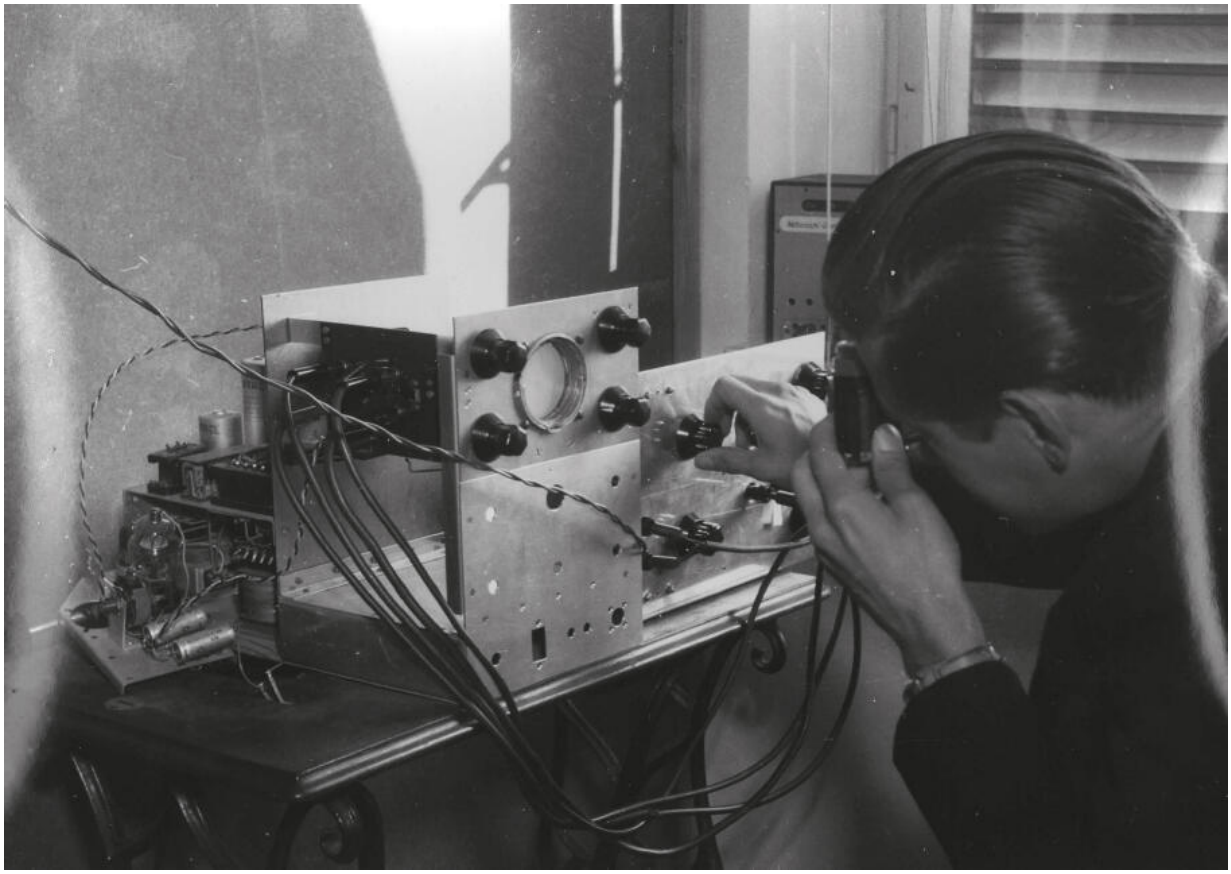


Figure 15.105. Herbert W. Franke with his special purpose analog computer (reprinted with permission of the artist)

In 1979, Benjamin Heidersberger[387] started the development of an electronic analog computer resembling the Rhythmograph. This machine contains three oscillators, which can be controlled separately with respect to frequency, phase, and damping. Additional modules implement the necessary control circuitry for repetitive operation, function generators, multipliers, etc. The machine is shown in →figure 15.106, while →figure 15.107 shows an example of a typical artwork generated with it.

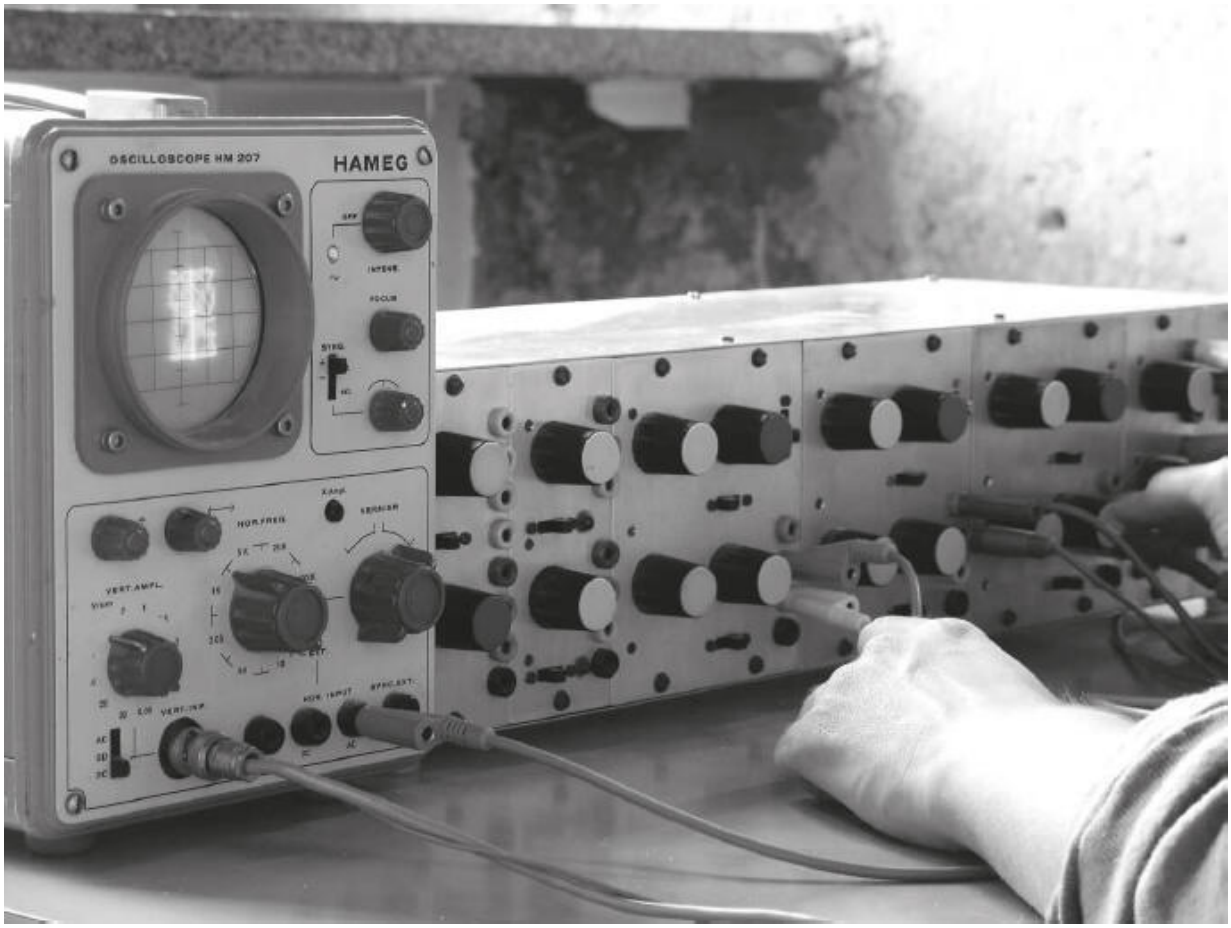


Figure 15.106. BENJAMIN HEIDERSBERGER's analog computer (reprinted with permission)

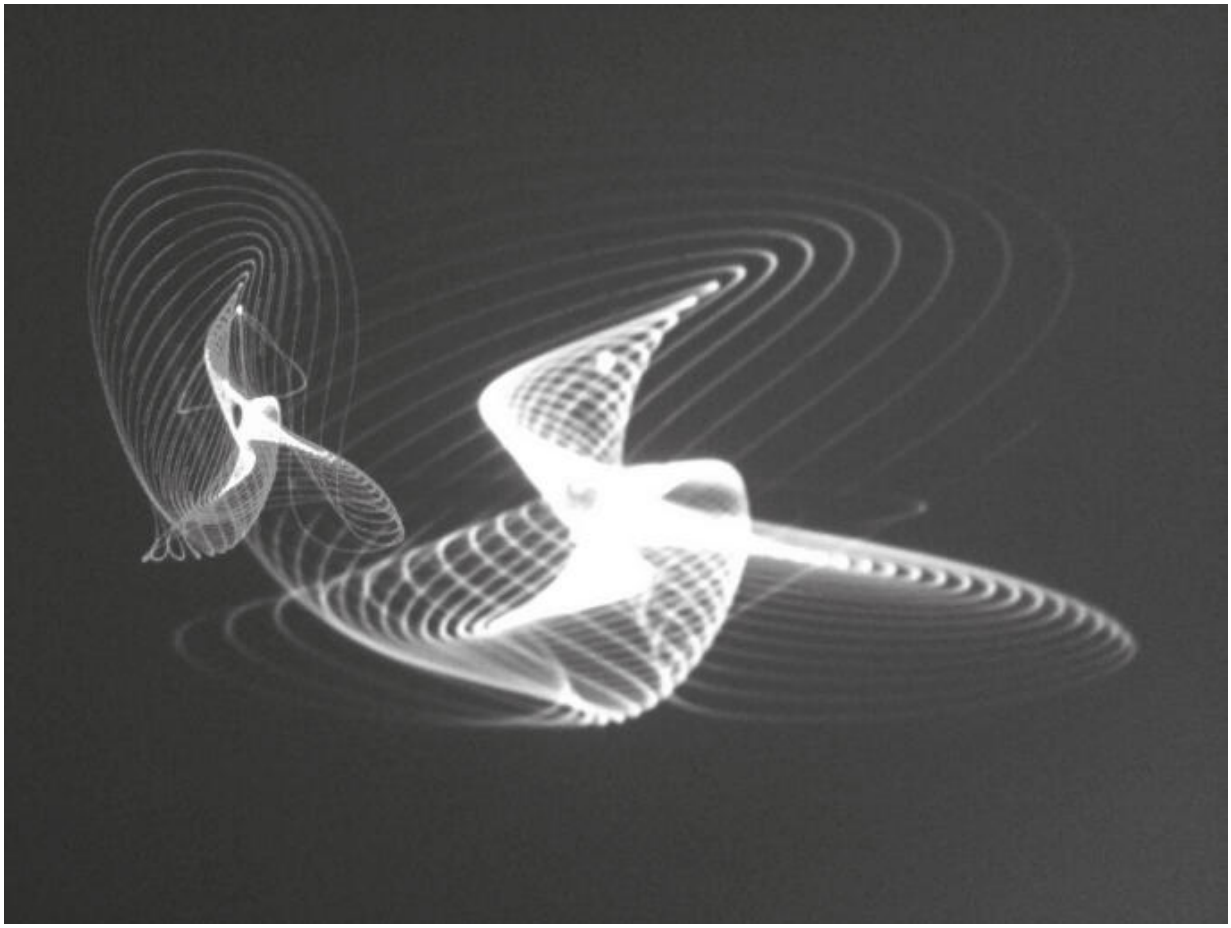


Figure 15.107. Graphic, generated with Benjamin Heidersberger's analog computer (reprinted with permission)

Two examples of analog computer art by Maughan S. Mason can also be found in [→Computers and Automation 1966, p. 9]. Two pictures generated on an analog computer by Otto Beckmann[388] are shown in [→Computers and Automation 1975, p. 12]. [→Rattay 1979] shows the artistic use of an analog computer, featuring the high repetition rate of pictures displayed on an oscilloscope facilitating parameter control by the artist.

15.20.2 Entertainment

Probably the first special purpose analog computer exclusively built for entertainment purposes was the *Cathode-Ray Tube Amusement Device* patented in 1948 and described as follows in [→Goldsmith et al. 1948]:

> "*This invention relates to a device with which a game can be played. The game is of such a character that it requires care and skill in playing it or operating the device with which the game is played. Skill can be increased with practice and the exercise of care contributes to success.*"

The goal of this game is to hit targets displayed on an oscilloscope screen by firing a beam, which can be controlled in its length and angle by the player. To generate a flicker-free picture, the analog computer operates in a high-speed repetitive mode. Clearly, this device was motivated by analog training devices for gunners developed during World War II. The patent specifications suggest the use of small, adhesive aircraft-shaped stickers on the oscilloscope tube to denote the targets.

Ten years later, William Higinbotham[389] independently developed a two-player game aptly called *Tennis for Two*, which was demonstrated during an open house presentation of the *Brookhaven National Laboratories*; previous events had shown that the public was not too interested in purely static displays as Higinbotham remembers:[390]

> "*I knew from past visitors days that people were not much interested in static exhibits, so for that year I came up with an idea for a hands-on display - a video tennis game. [...] It was wildly successful, and Higinbotham could tell from the crowd reaction that he had designed something very special. 'But if I had realized just how significant it was, I would have taken out a patent and the U.S. government would own it!' he said.*"

An analog electronic replica of Higinbotham's game, which was originally implemented on a small *Systron Donner* tabletop analog computer, has been developed by the *Museum of Electronic Games & Art*.[391]

A much more complex game based on an electronic analog computer is the "Golf Game Computing System" described by [→ Russel et al. 1971]. This golf simulator is characterized by the unusually high simulation fidelity. Even effects like the change from laminar to turbulent air flow around the ball, spin, and many more were taken into account for the trajectory computation.

A rather complex implementation of a Squash simulation can be found in [→ Breitenecker 1979]. The game takes place in a three-sided box with a single squash racket which can be controlled in its position and angle. Using a simple digital control program, a picture consisting of three parts, the cube walls, the racket, and the ball, is displayed with high

repetition rate on an oscilloscope screen. The racket is controlled by a two-axis joystick with one axis representing its position along the open side of the box ad the other axis representing its angle. An equally sophisticated and complicated simulation of a billiard game on a Telefunken RA 770 system is described in [→Lotz 1970/1].

[→Russell 1978] describes a simple homebrew analog computer implementing a moon landing simulation.[392]

Another fascinating application of analog computers was the animation of cartoons described in [→Miura et al. 1967/2]. →Figure 15.108 shows a rabbit driving a car, animated on a Hitachi *ALS-2000* analog computer running in repetitive mode with a repetition rate of 3 kHz.[393]

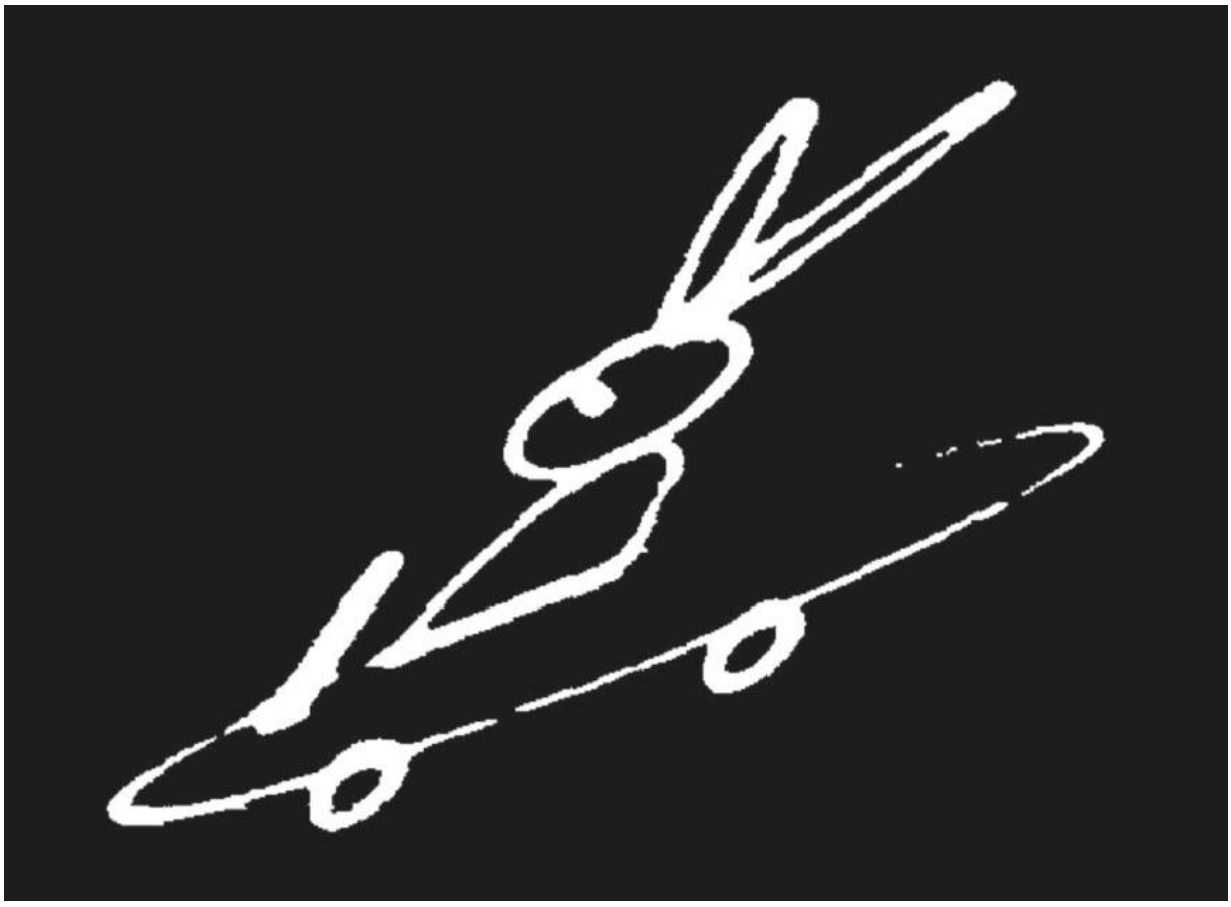


Figure 15.108. Cartoon still showing a rabbit driving a car ([→Miura et al. 1967/2, p. 143])

15.20.3 Music

Although analog synthesizers can be justifiably regarded as being special purpose analog computers, general purpose analog computers have been used rarely in this area.[394] Analog computers were used as early as in the late 1950s in France for *musique cybernetique*, where sequences of notes were generated by analog means which were then manually converted into scores.[395]

Another approach, *PACER MUSICALE*, is described in [→ GSANDTNER 1975]. This system is described as an interactive hybrid program for the analysis and synthesis of music. It uses a hybrid computer where the digital part generates frequency and duration values which are fed to the analog computer, where the actual generation of sounds is implemented. The tone color can be controlled either digitally or on the analog system. Using the analog computer for a frequency analysis, it is also possible to analyze a piece of music and transform it into a digital representation.

Also noteworthy is the work of Dutch composer HANS KULK, who has used *Hitachi-240* analog computers to create music since the late 1980s.[396] The high degree of interactivity offered by analog computers is exploited extensively in his work by using special input devices such as a three-dimensional manual controller, which enables the composer to become part of the composition and sound generation process by changing various parameters of the analog computer setup with simple hand movements. The advantages and the possibilities of using analog computers in musical applications has been summarized by KULK as follows:[397]

> "*One main conclusion is that the use of an analog [...] computer in analog sound synthesis is very effective, [yet] has been highly ignored by the electronic music community for long, but is slowly gaining interest as the concepts of analog technique become valuable tools in current developments. The 1970s dual Hitachi 240 set-up in this sound synthesis lab will continue to be an inspiring and open system with still many hours of fun to come.*"[398]

A fascinating simulation of the ring in process of a bell is described in [→ RATTAY 1978/2]. The actual bell is modeled as a double pendulum with the simplifying assumption that the motion of the bell itself is not

influenced by the ringer hitting the bell. The ring in process is manually controlled using a switch to apply a force to the bell rope. An especially nice aspect of this work is the display of the moving bell with its ringer on an oscilloscope screen as shown in →figure 15.109. The bell shape is based on a cosine function generated by the classic two-integrator circuit using a variable input weight controlled by a comparator, yielding the flat top part.

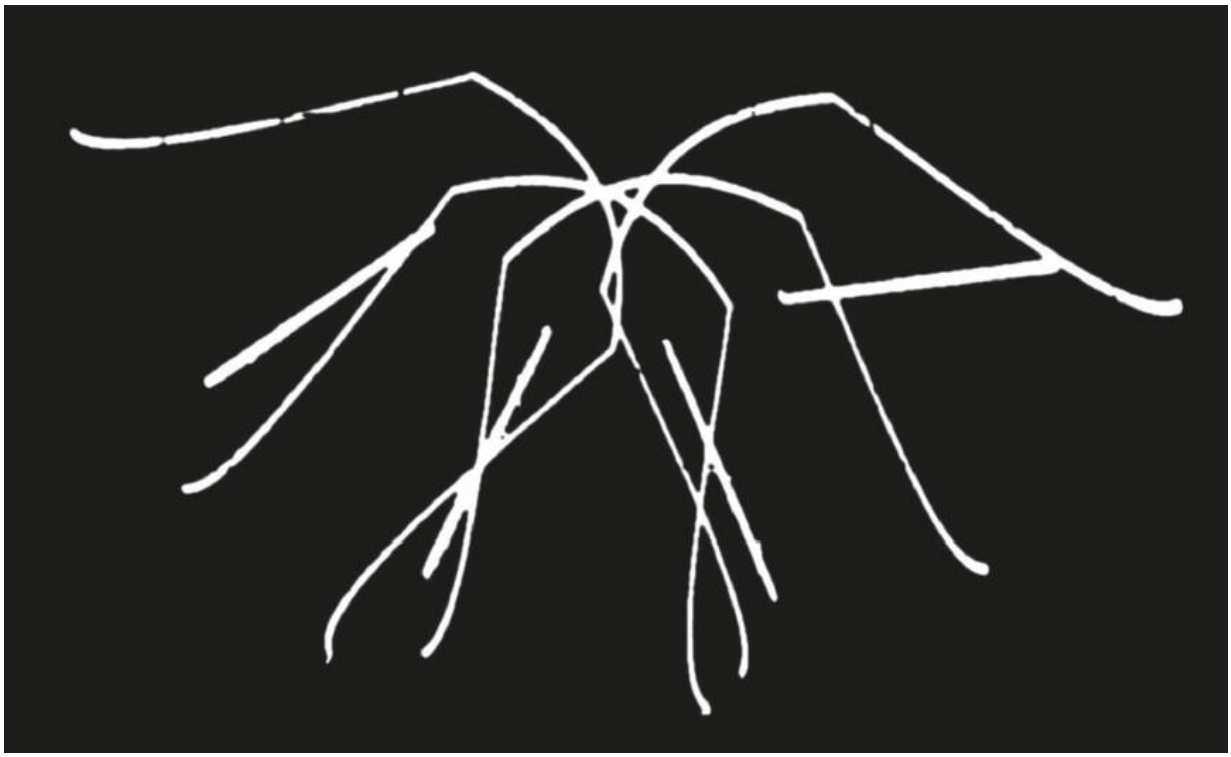


Figure 15.109. Simulation of a bell as a double pendulum (see [→RATTAY 1978/2, p. 32])

15.21 Analog computer centers

A typical phenomenon in the heyday of electronic analog and hybrid computers were computer centers set up by the major manufacturers of these systems such as Electronic Associates Inc., Beckman, etc. These computer centers served a dual purpose: First, potential customers were able to get hands-on experience with the latest machines and could solve practical problems in order to decide whether a particular analog

computer met their needs. Second, these computer centers offered analog computer time on a rental basis for customers who either had no analog computer installation at all or required additional computing power for the solution of certain problems.

The first analog computer center in Europe was set up in 1957 in Brussels by EAI.[399] →Figure 15.110 shows part of the EAI *Pace 96* analog computer which was installed in Brussels and contained 96 operational amplifiers, 120 coefficient potentiometers, 20 servo multipliers, four quarter square multipliers, five function generators, two digital voltmeters, a two-channel *xy*-plotter, two single channel *xy*-plotters, two six-channel recorders, and two electronic noise sources.[400] EAI's computer center in Brussels was always equipped with their latest systems. In 1964 a HYDAC 2000,[401] shown in →figure 6.10, was installed.[402]

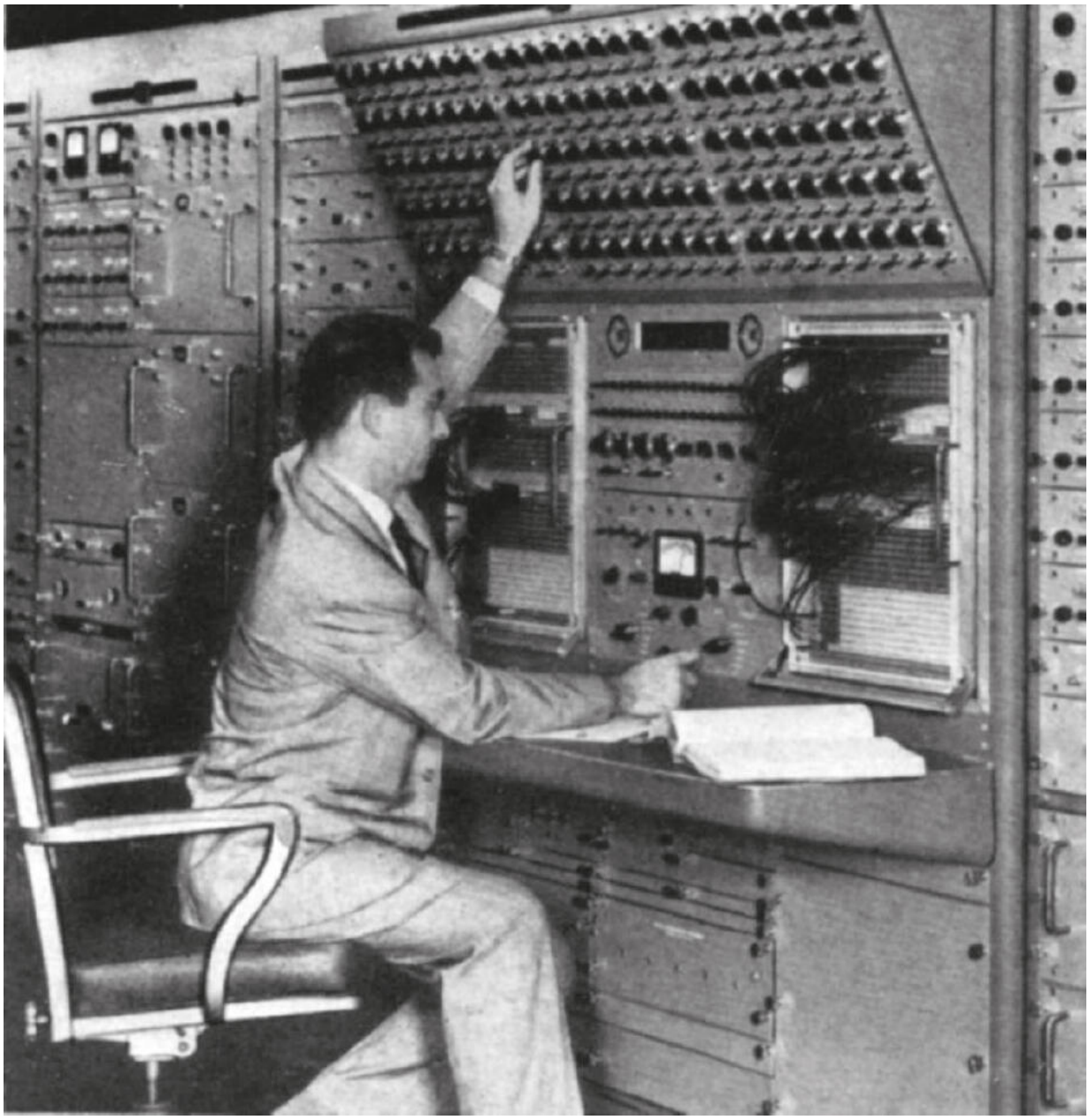


Figure 15.110. EAI Pace 96 analog computer, installed in Europe's first analog computer center in Brussels (see [→Elektronische Rundschau 1957])

→Figure 15.111 shows one of EAI's large US domestic analog computer centers, equipped with several fully equipped EAI 231RV analog computers and many additional components.

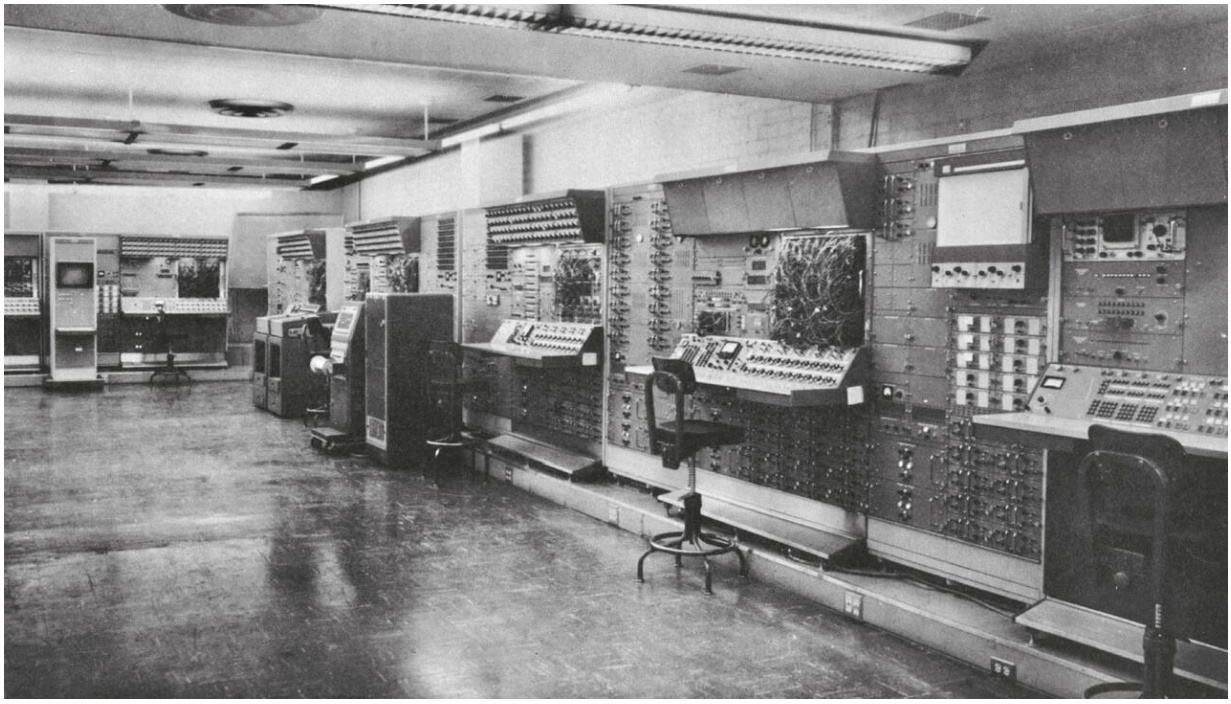


Figure 15.111. EAI computation center (see [→EAI 1964/2, p. iv])

One year later, Beckman opened their first analog computer center in Los Angeles, which mainly offered computer time for paying customers. →Figure 15.112 shows the EASE 1132, the first computer installed there. One year before its inauguration, this computer center was announced:[403]

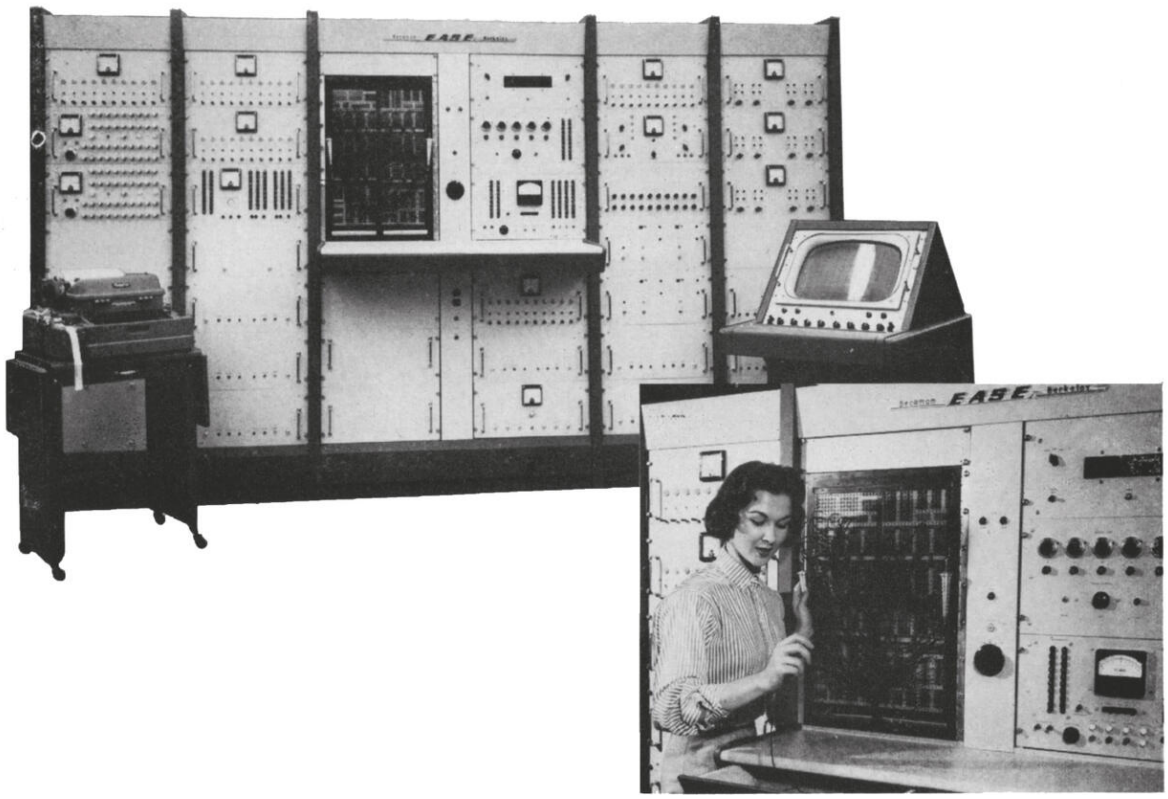


Figure 15.112. EASE 1132 analog computer, installed in the *Beckman/Berkeley Computation Center*, which opened on February 28, 1958 in Los Angeles (see [→Instruments and Automation 1957/1])

> "*It will be available to business, industry and educational institutions for solution of complex problems relating to aircraft, jet engines, guided missiles and industrial processes.*"

A detailed view of this machine is shown in →figure 15.113.



Figure 15.113. Detailed view of the system shown in →figure 15.112 (see [→Hussey et al., p. 5])

16 Analog and hybrid computing in the 21st century

The reasons for the decline of analog computing in the 1970s and 1980s are manifold. These marvelous machines were quickly displaced by the upcoming mini-, supermini-, and later microcomputers. Although these could not compete with analog computers in terms of sheer computational power, they quickly became cheaper due to mass production and their comparatively simple implementation, while analog computers continued to be expensive and delicate machines requiring high-precision components.

Another problem was that a classic analog computer typically had to be reserved on a single-user base for quite some time by a researcher or project group since it could not readily be time-shared between different tasks at once. On the other hand, time-sharing was supported even on rather small minicomputers of the early 1970s, outweighing their lack of computational power in many cases, especially from a commercial point of view.[1]

Changing an analog computer setup from one program to another also required a substantial amount of time. Even with removable patch panels, there are typically a lot of potentiometers and switches to be set, external devices to be configured, etc. This tedious reconfiguration process on classic analog computers was a major contributor to their eventual demise, see [→Tisdale 1981, p. 1]. In contrast to this, switching from one program to another on a 1970s minicomputer was a matter of some ten to hundred milliseconds. In addition to this, storing an analog computer program was quite costly as it

required at least a dedicated and expensive patch panel as well as many hundred patch cables and finally a storage location for the pre-patched panel. On a digital computer, program storage only required cheap storage media such as paper tapes, punch cards, magnetic tape, or a few sectors on a disk drive.[2]

Another inherent advantage of digital computers is their ability to allow a trade-off between solution-time and problem complexity. As long as memory size is not a limiting factor, problems of varying complexity can be solved with execution time typically increasing with problem complexity, typically described by the *Big-O-Notation*. Within certain limits this characteristic is an advantage over analog computers.[3] An analog computer, on the other hand, has to grow linearly ($O(n)$) with the complexity of the problem to be solved. If there are not enough computing elements for a certain problem, it cannot be solved on an analog computer. Given the technology of the 1970s and 1980s, this was a clear disadvantage of analog computers as filling more and more racks with expensive electronic equipment was just not practical – neither from a technological nor a commercial point of view.

However, with modern technology allowing the integration of analog computers with hundreds or thousands of computing elements on small integrated circuits, this disadvantage is nowadays more than outweighed by the fact that the solution time on an analog computer is basically constant, $O(1)$, i. e., it basically does not grow with increasing problem size.[4] This cannot be overestimated!

In the 1980s, classic analog computers were finally relegated to museum pieces and even the idea of analog, i. e., non-algorithmic computing vanished from school and university curricula.[5] The idea of setting up a typically electronic model to solve a problem not by executing a sequence of instructions but

through a measurement process became obsolete despite its advantages in terms of computational speed and energy efficiency over the digital approach.

Digital computers have become many orders of magnitude faster, smaller, and cheaper since the 1970s, but this progress with respect to computational power, integration density, and energy efficiency is slowing down considerably as physical limits are being reached in recent years. Integration densities are eventually limited by the size of atoms, not too far from today's 5nm and 2nm technology nodes. Maximum clock frequencies are limited by energy consumption, which grows super-linearly with clock frequency – in fact, clock frequencies sit pretty stable at roughly 3 GHz for the last quarter of a century now, and the speedup obtained with massively parallel digital computers is basically an asymptotic process as described by *Amdahl's*[6] *law*. Additionally, the overall energy consumption of today's information and communication sector (*ICT*) continues to soar, with computationally-intensive applications, such as artificial intelligence, being the main contributors.[7] Current super computers require several 10 MW of electric energy, sometimes even more, and there are already plans for super computer centers requiring a gigawatt of energy necessitating dedicated (nuclear) power plants. As impressive as this is from a purely technological point of view, it obviously does not scale well.

Modern digital processors also require multi-level cache structures to alleviate the exceedingly long access times to main memory. This, together with the requirement for ever increasingly complex control circuitry, results in only a comparatively small portion of the overall transistor functions on a digital processor being involved in actual computations. In contrast to this, an integrated circuit implementing an analog computer could use all of its silicon estate for actual

computations since there is no need for memory as such, no complex control system is required, etc.

These inherent problems of algorithmic computers already led to a massive interest in quantum computing and will also spark a renaissance of analog computing, complementing the promises of quantum computers. Analog computing will return in the form of co-processors as it perfectly complements digital computers in its characteristics, effectively creating modern hybrid computers. These modern incarnations of analog computers will be integrated circuits employing modern CMOS technology, will be reconfigurable in a matter of microseconds, will feature integrated ADCs and DACs, function generators, etc. The main advantages offered by such analog co-processors are

- their unmatched degree of parallelism and thus high computational power,[8]
- their high energy efficiency,[9]
- their lack of any algorithmic (mainly sequential) control whatsoever,
- the fact that there is no main memory, eliminating slow instruction and data fetches, further enhancing the energy efficiency of computing.
- Finally, analog computers are an ideal match for our analog world to which they not just interface, but of which they can become an actual part.

There are additional advantages such as the fact that analog computers do not suffer from many of the effects caused by a binary floating point number representation and numerical algorithms, namely integration schemes.[10] Some researchers even argue that phenomena such as chaotic systems or molecular dynamics can only be implemented by analog means due to the detrimental characteristics of floating point numbers,

mainly the uneven spacing of successive values.[11] [→ Coveney et al. 2025, p. 167] even state for the realm of molecular dynamics

> "*[o]nly the advent of analogue computers capable of solving such systems of equations will allow us to get to the bottom of these discrepancies as well as arrest the insatiable demand of digital computers for electrical energy.*"

Additionally, programming an analog computer is a much more natural process than developing a numerical algorithm for the solution of a problem. [→ Tisdale 1981, p. 3] notes aptly:

> "*Digital languages [...] obstruct the user's contact with the physical analogy. In other words, the digital programmer becomes preoccupied with the programming task and unfortunately loses sight of the analogy he hopes to create. Hybrid techniques go hand-in-hand with Laplace and Fourier expressions. The analogy is not only established, but the engineer's knowledge and skill grows quickly, giving rise to innovations and breakthroughs.*"

As early as 1975, Wolfgang Giloi[12] predicted that "*certainly [...] the analog method of hardware parallel processing*" will be preserved. Lee Albert Rubel was even more optimistic, as the following quote from a personal note to Jonathan W. Mills of the Indiana University shows:

> "*The future of analog computing is unlimited. As a visionary, I see it eventually displacing digital computing, especially, in the beginning, in partial differential equations and as a model in neurobiology. It will take some decades for this to be done. In the meantime, it is a very rich and challenging field of investigation, although (or maybe because) it is not in the current fashion.*"

[→ Guzdial et al. 2013] are also quite enthusiastic (p. 12):

> "*For exascale computing, reliability, resilience, numerical stability and confidence can be problematic when input uncertainties can propagate, and single and multiple bit upsets can disturb numerical representations. [...] Could analog computing play a role? Please note that I am not advocating a*

> *return to slide rules or pneumatic computing systems. Rather, I am suggesting we step back and remember that the evolution of technologies brings new opportunities to revisit old assumptions. Hybrid computing may be one possible way to address the challenges we face on the intersecting frontiers of device physics, computer architecture, and software."*

It should be noted that even parallel DDAs based on *FPGA*s (*Field Programmable Gate Array*s) will probably also find widespread application as co-processors/accelerators in the near future as they, too, exhibit the perfect parallelism of analog computers. Though they lack the high energy-efficiency of analog computers, this might be compensated for by the possibility of arbitrary precision in computations where this is needed.[13][14]

16.1 Challenges

In order to re-establish analog, and thus hybrid computing as an ubiquitous computational paradigm in the 21st century, a number of challenges must be addressed and overcome.

The first to mention is the need for high density integration using a standard semiconductor CMOS process. A lot of work has already been done in this area with analog computing element designs readily available for 65 nm and also 22 nm technology nodes. A noteworthy aspect is that analog computing elements in general require much fewer transistor functions than comparable digital units. Summing values with a precision of three to four decimal places in the analog domain just requires a number of resistors and an operational amplifier, totaling at a few dozen components to be integrated, while a similar digital unit, a *full-adder* requires nine NAND or NOR gates per bit. So even in the case of three decimal places of precision, corresponding to 10 bits of resolution, a digital adder would require 90 gates, each requiring about four transistors in CMOS technology resulting in a total of about 360 transistor functions.

Accordingly, modern analog computers on integrated circuits do not even require highly expensive 5 nm or 2 nm structures, but can be manufactured using mature and cheap processes such as 65 nm 22 nm instead. This makes it possible to fabricate such devices locally in regions like Germany, the EU, etc., and not just in the United States or Taiwan, reducing dependencies on other countries. All in all, the design and development of such integrated circuits is merely an engineering problem not requiring any fundamental research, in contrast to quantum computers, for example.

The next challenge is that of reconfigurability. The classic patch panel is clearly a thing of the past and belongs in a museum. Modern integrated circuit analog computers must be reconfigurable by an attached digital computer with configuration times of a few milliseconds at most, ideally much less. This idea is by no means new as shown in →chapter 11, and there are already viable implementations ranging from simple switched capacitor interconnects to hierarchical crossbar architectures, mixed voltage and current coupled switch matrices, etc.[15]

This capability requires a suitable compiler translating some high-level problem description, ideally just mathematical equations, into a semantically equivalent analog computer setup. There is already quite some work done on domain specific languages (*DSL*) such as *LEGNO*[16] or commercial developments by companies like anabrid GmbH.[17] Apart from a DSL with corresponding compiler, a true software ecosystem for modern hybrid computers is required. This not only requires libraries to interact with the analog co-processor but also new numerical algorithms taking maximum advantage of the analog subsystem. An interesting problem in this context are the long interrupt latencies exhibited by modern operating systems,

which can easily reach > 10μs, often exceeding typical solution times on a modern analog computer.[18]

This ecosystem will also need to provide means for automatic scaling.[19] A straightforward approach would be to run a roughly scaled system on the analog subsystem until an overload occurs. The attached digital computer could then rescale the problem and restart it on the analog computer. It could even read the values obtained so far and use these, properly rescaled, as initial conditions for the next scaling iteration, thus saving additional time.

Another requirement for a modern hybrid computer software stack are means for checkout[20] and calibration of the analog computer. Classic analog computers went to great lengths to minimize static and dynamic errors using high precision components. Using an automatic calibration process, high accuracies can be achieved even with low-precision computing elements, substantially decreasing the cost of implementation of modern analog computers.

16.2 Applications

Applications for modern reconfigurable analog computers are abundant – not just as co-processors but also in a variety of stand-alone applications.[21] The first application that comes to mind is surely that of *high performance computing*, (*HPC*) using a tightly integrated analog computer as a co-processor to speed up computations for certain classes of problems. In this role, the analog computer could generate approximate solutions for problems much faster and much more energy efficiently than digital approaches. If the precision of results obtained this way is not sufficient, modern numeric algorithms could use them as initial conditions and then increase the precision using some

iterative process as desired.[22] The general topic of benchmarking analog co-processors is described in [→Hasler et al. 2024].

Even the early idea of using direct analogies might have a future as shown by Mills and his co-workers in the early 2000s. They developed analog co-processors based on direct analogies, as described by Lee Albert Rubel.[23] These *Extended Analog Computers* (*EAC*s) are similar in their structure to the electrolytic tanks mentioned in →section 1.4 but use much more modern materials, such as polymer foils.[24] Co-processors like these are especially well suited for the solution of problems described by partial differential equations.

The high energy efficiency of analog techniques, often exceeding several decades over classic digital approaches, can not be overestimated in current times. An example for this can be found in [→Cowan et al. 2009, pp. 13 f.] where different approaches to solving a particular stochastic differential equation are compared with respect to the amount of energy consumed. The traditional algorithmic approach required about 1.2–40 J of energy when run on a typical microprocessor. The same computation running on a *DSP* (*Digital Signal Processor*) required only 0.04–0.4 J. Using a VLSI analog processor developed by Cowan,[25] the same task could be performed with only 0.008 J – quite a substantial saving. An application for this particular chip, featuring 416 analog functional blocks, is described in [→Freedman 2011]. Cowan writes:[26]

> "*This chip is controlled and programmed by a PC via a data acquisition card. This arrangement has been used to solve differential equations with acceptable accuracy, as much as 400× fast than a modern workstation. The utility of a VLSI analog computer has been demonstrated by solving stochastic differential equations, partial differential equations, and ordinary differential equations.*"

Although these figures are outdated by now, they serve as a good example of possible energy savings. A more recent treatment can be found in [→Köppel et al. 2021].

Their high energy efficiency makes modern analog co-processors ideally suited for medical applications where energy is at a premium. The idea of medical implants using energy harvesting instead of bulky accumulators is intriguing and in the future might affect many lives for the better. Even if powered by some chemical battery, using analog processing technologies will increase battery life-time substantially.

The same reasoning makes analog computers interesting for mobile devices, edge computing, smart sensors and actors, robotics,[27] as well as always-on-devices by performing highly energy efficient signal pre- and post-processing.[28] There are already FPAAs on the market aimed at such edge-applications.[29] An example for this is shown in →figure 16.1. The board uses an *Anadigm*/*Okika Devices* FPAA, which can be programmed using a Python *API* (*Application Programming Interface*) and used for audio applications.[30] These FPAAs can also be used as simple analog computers as shown in [→Moreno et al. 2020] and [→Beil 2021].

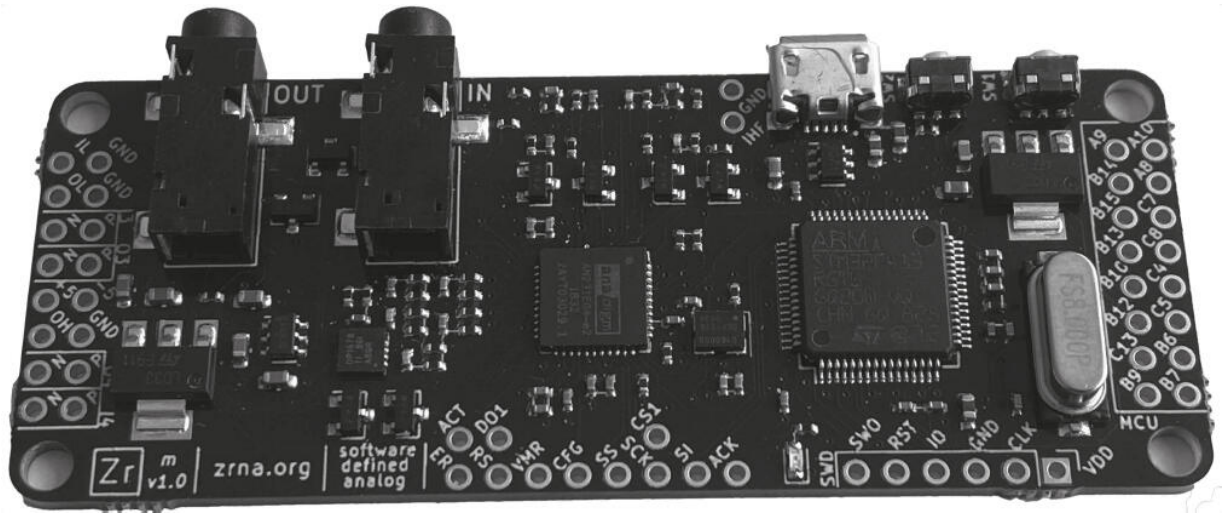


Figure 16.1. ZRNA FPAA board (see →zrna.org, retrieved 04.09.2022)

In always-on-devices, analog co-processors will be used for highly energy efficient and robust trigger word detection, saving substantial amounts of energy on a global scale while enhancing user experience.[31]

The ability of analog computers to adapt quite seamlessly to the physical world is an additional and central benefit in many applications.[32] [→SRC 2021, pp. 12 ff.] notes

> "*Fundamental breakthroughs in analog hardware are required to generate smarter world-machine interfaces that can sense, perceive and reason. [...] The physical world is inherently analog and the 'digital society' drives increasing demand for advanced analog electronics to enable interaction between the physical and 'cyber worlds'.*"

In high-volume applications, where reconfigurability is not required, statically interconnected highly integrated analog computers will be used, making them un-hackable. This is a massive asset in many application areas such as industrial control, medical systems, etc.

However, the most important application area for analog computers in the near future is surely that of artificial intelligence including topics such as *machine learning ML*, but most importantly artificial neural networks (ANNs) and *neuromorphic computing* in general. A simple ANN basically consists of layers of neurons, numbering in the millions and billions nowadays, each basically computing a scalar product of some input vector and a weight vector, and applying the result of this operation to some *activation function* such as a *rectified linear unit* (*ReLU*).[33]

Classic digital approaches implement these operations using linear algebraic operations on matrices and vectors, employing specialized digital co-processor architectures such as graphics cards, etc. Implementing the functions of such a simple artificial neuron on an analog computer is rather trivial – the input weights can be represented as (programmable) resistors, the summation is just done by connecting these input resistors to the summing junction of an operational amplifier, and the activation function can also be implemented easily by appropriate elements in the feedback path of the operational amplifier. The main advantage of this approach is its high energy efficiency, surpassing digital approaches by several decades, making it possible to perform inference and in some cases also the training process directly in customer devices, smart-sensors and actors, etc.[34] Analog AI accelerators in general are described in [→ Channamadhavuni et al. 2021]. A *long short-term memory* (*LSTM*) network design for analog computing is described in detail in [→ Zhao et al. 2019]. [→ Shrestha et al. 2022] contains a comprehensive survey on *neuromorphic computing* including analog computing approaches. A good overview of *brain inspired computing* in general can be found in [→ Parhi et al. 2020]. The application of memristors to these fields is covered in [→ Saleh et al. 2023].

Some authors even argue that true brain-like machines or even *artificial brains* are already well within reach using analog computing technologies. [→ Hasler 2017] argues that even back in 2013 a "*silicon version of the human cerebral cortex*" would be feasible using the semiconductor technologies already in production back then. Such a machine "*would take up less than a cubic meter of space and consume less than* 100 *watts, not too far from the human brain*". Using classic algorithmic digital approaches, something like this will clearly not be possible.[35] Some authors, like [→ Resconi et al. 2015], even argue that a brain, being an electric network, is in principle not comparable with a stored-program digital computer, thus precluding any way to actually implement brain-like functionality using digital computers. If this is true, analog computers would be the only way to go further in this field.

Taking all of this into account, it is safe to say that probably no technology of the past has such a great potential for future applications as the paradigm of analog computing. Quoting the inscription of James Earl Fraser's[36] statue "*Heritage*" at the Federal Triangle, Constitution Ave. & 9th, Washington, DC:

> "*The Heritage of the Past is the Seed that Brings Forth the Harvest of the Future.*"

A THE ANALOG THING explained

THE ANALOG THING is sufficiently simple in its design and construction to serve as an example on how to build a small-scale analog computer with modern electronics parts.[1]

→ Figures A.1 and → A.2 show the two schematics of the front panel containing the passive input resistor networks for the

various computing elements. Summer, inverter, and integrator inputs with a weight of 1 use one megaohm resistors. Accordingly, inputs with weight 10 use 100 kiloohm resistors. It should be noted that all patch panel jacks are connected to ground by means of transient-voltage-suppression diodes.[2] This is necessary since most of the active components, mainly the operational amplifiers but also the multipliers, etc., are sensitive with respect to electrostatic discharges.

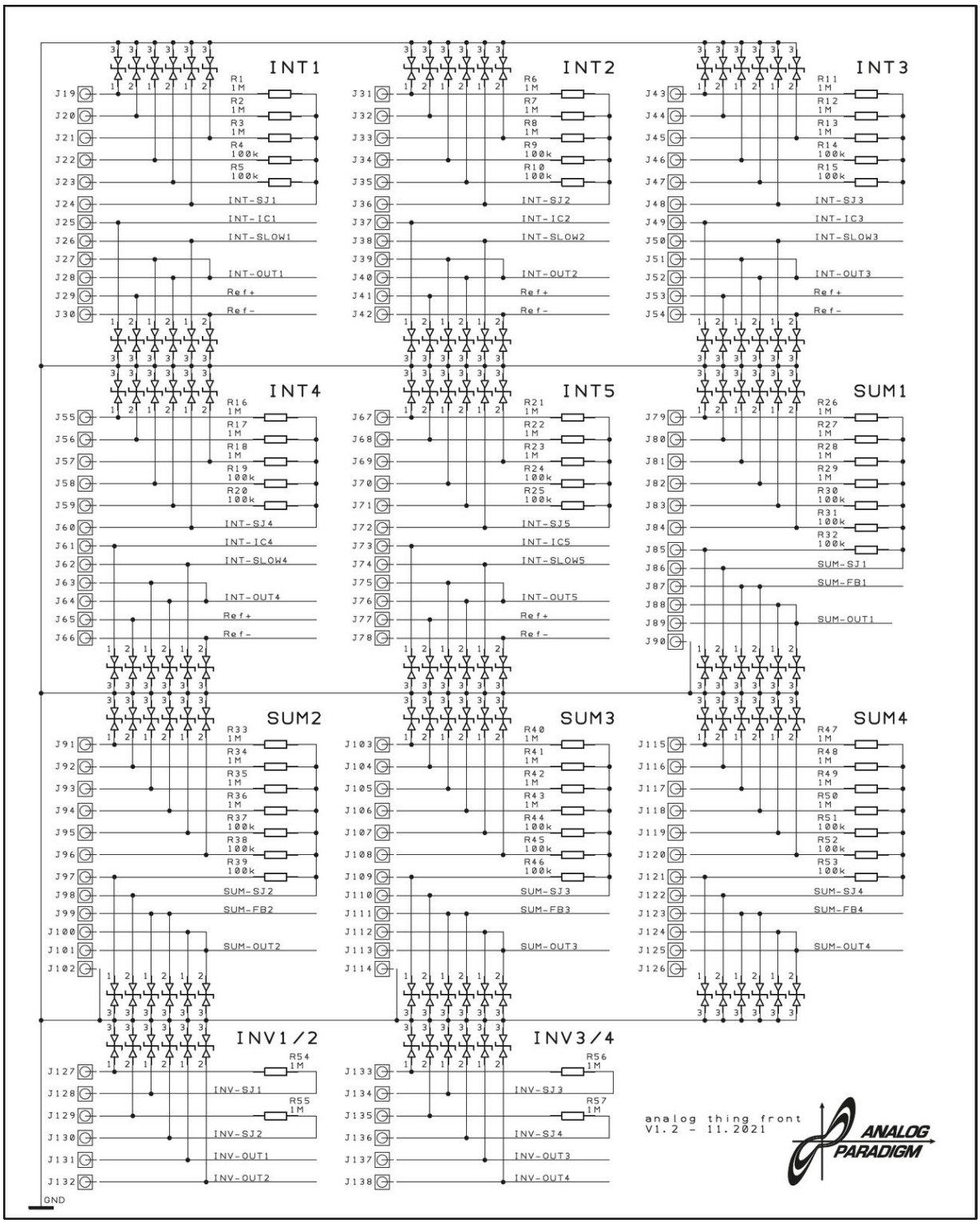


Figure A.1. First part of the front panel schematic

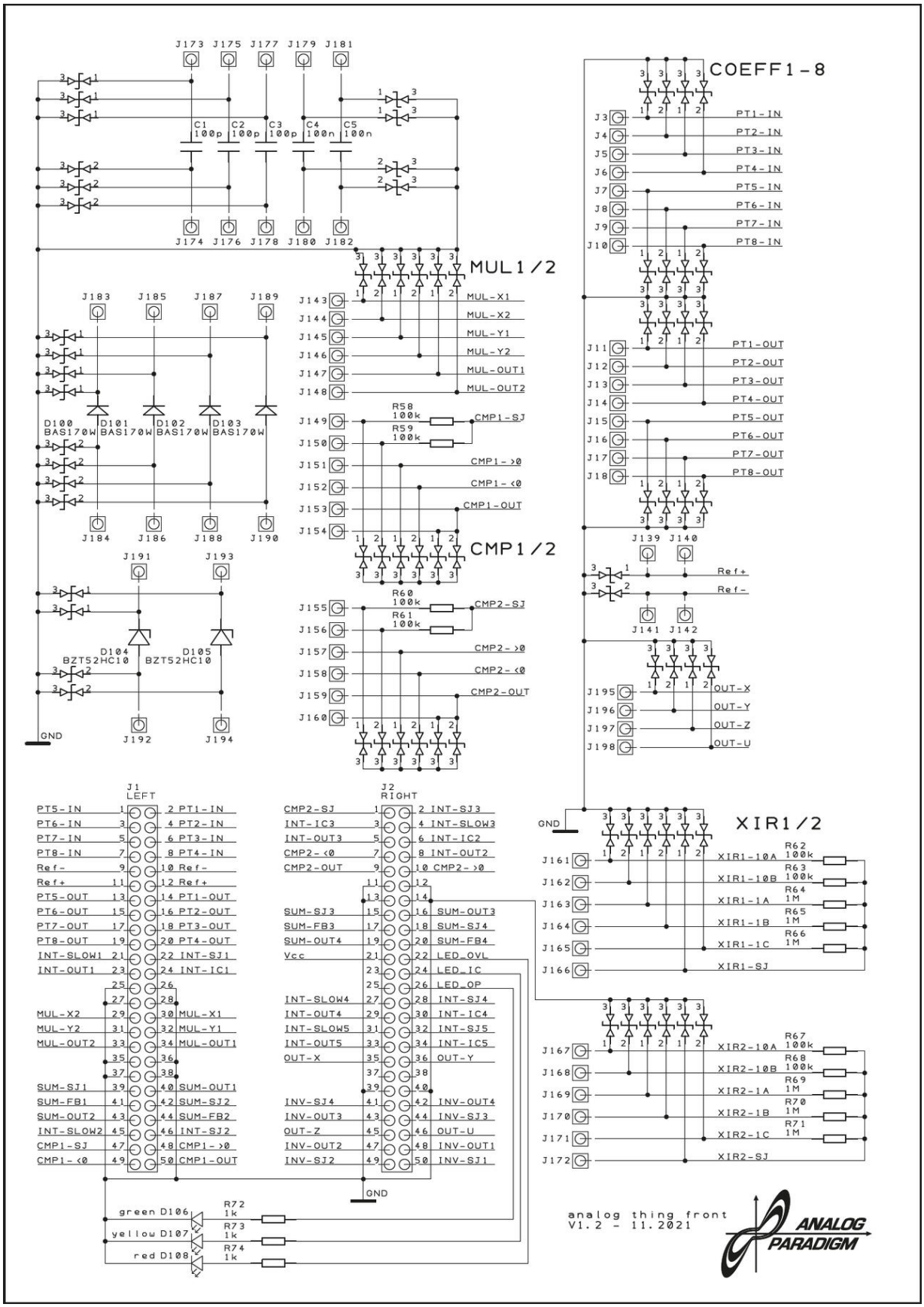

J173 J175 J177 J179 J181
C1 100p C2 100p C3 100p C4 100n C5 100n
J174 J176 J178 J180 J182
MUL1/2
J143 MUL-X1
J144 MUL-X2
J145 MUL-Y1
J146 MUL-Y2
J147 MUL-OUT1
J148 MUL-OUT2
J183 J185 J187 J189
D100 BAS170W D101 BAS170W D102 BAS170W D103 BAS170W
J184 J186 J188 J190
R58 100k
R59 100k
J149 CMP1-SJ
J150
J151 CMP1->0
J152 CMP1-<0
J153 CMP1-OUT
J154
CMP1/2
J191 J193
D104 BZT52HC10 D105 BZT52HC10
J192 J194
GND
R60 100k
R61 100k
J155 CMP2-SJ
J156
J157 CMP2->0
J158 CMP2-<0
J159 CMP2-OUT
J160
COEFF1-8
J3 PT1-IN
J4 PT2-IN
J5 PT3-IN
J6 PT4-IN
J7 PT5-IN
J8 PT6-IN
J9 PT7-IN
J10 PT8-IN
J11 PT1-OUT
J12 PT2-OUT
J13 PT3-OUT
J14 PT4-OUT
J15 PT5-OUT
J16 PT6-OUT
J17 PT7-OUT
J18 PT8-OUT
J139 J140
Ref+
Ref-
J141 J142
J195 OUT-X
J196 OUT-Y
J197 OUT-Z
J198 OUT-U
J1 LEFT
PT5-IN 1 2 PT1-IN
PT6-IN 3 4 PT2-IN
PT7-IN 5 6 PT3-IN
PT8-IN 7 8 PT4-IN
Ref- 9 10 Ref-
Ref+ 11 12 Ref+
PT5-OUT 13 14 PT1-OUT
PT6-OUT 15 16 PT2-OUT
PT7-OUT 17 18 PT3-OUT
PT8-OUT 19 20 PT4-OUT
INT-SLOW1 21 22 INT-SJ1
INT-OUT1 23 24 INT-IC1
25 26
27 28
MUL-X2 29 30 MUL-X1
MUL-Y2 31 32 MUL-Y1
MUL-OUT2 33 34 MUL-OUT1
35 36
37 38
SUM-SJ1 39 40 SUM-OUT1
SUM-FB1 41 42 SUM-SJ2
SUM-OUT2 43 44 SUM-FB2
INT-SLOW2 45 46 INT-SJ2
CMP1-SJ 47 48 CMP1->0
CMP1-<0 49 50 CMP1-OUT
J2 RIGHT
CMP2-SJ 1 2 INT-SJ3
INT-IC3 3 4 INT-SLOW3
INT-OUT3 5 6 INT-IC2
CMP2-<0 7 8 INT-OUT2
CMP2-OUT 9 10 CMP2->0
11 12
13 14
SUM-SJ3 15 16 SUM-OUT3
SUM-FB3 17 18 SUM-SJ4
SUM-OUT4 19 20 SUM-FB4
Vcc 21 22 LED_OVL
23 24 LED_IC
25 26 LED_OP
INT-SLOW4 27 28 INT-SJ4
INT-OUT4 29 30 INT-IC4
INT-SLOW5 31 32 INT-SJ5
INT-OUT5 33 34 INT-IC5
OUT-X 35 36 OUT-Y
37 38
39 40
INV-SJ4 41 42 INV-OUT4
INV-OUT3 43 44 INV-SJ3
OUT-Z 45 46 OUT-U
INV-OUT2 47 48 INV-OUT1
INV-SJ2 49 50 INV-SJ1
GND
green D106 R72 1k
yellow D107 R73 1k
red D108 R74 1k
GND
XIR1/2
J161 XIR1-10A R62 100k
J162 XIR1-10B R63 100k
J163 XIR1-1A R64 1M
J164 XIR1-1B R65 1M
J165 XIR1-1C R66 1M
J166 XIR1-SJ
J167 XIR2-10A R67 100k
J168 XIR2-10B R68 100k
J169 XIR2-1A R69 1M
J170 XIR2-1B R70 1M
J171 XIR2-1C R71 1M
J172 XIR2-SJ
analog thing front
V1.2 - 11.2021
ANALOG PARADIGM

Figure A.2. Second part of the front panel schematic

The four integrators, five summers, and four inverters of the THAT are shown in →figure A.3. The inverters consist of only one operational amplifier each with a fixed 1 megaohm feedback resistor. If should be noted that the inverting inputs of all of the operational amplifiers used in this configuration are connected to the summing junction not directly but via a 5k1 resistor.[3] This protects the operational amplifier in a case where someone might patch a machine unit of ±10 V to the summing junction connector on the front panel. Without this resistor, the input would typically draw excessive current due to internal protection diodes of the operational amplifier and thus thermally destroy the amplifier chip.

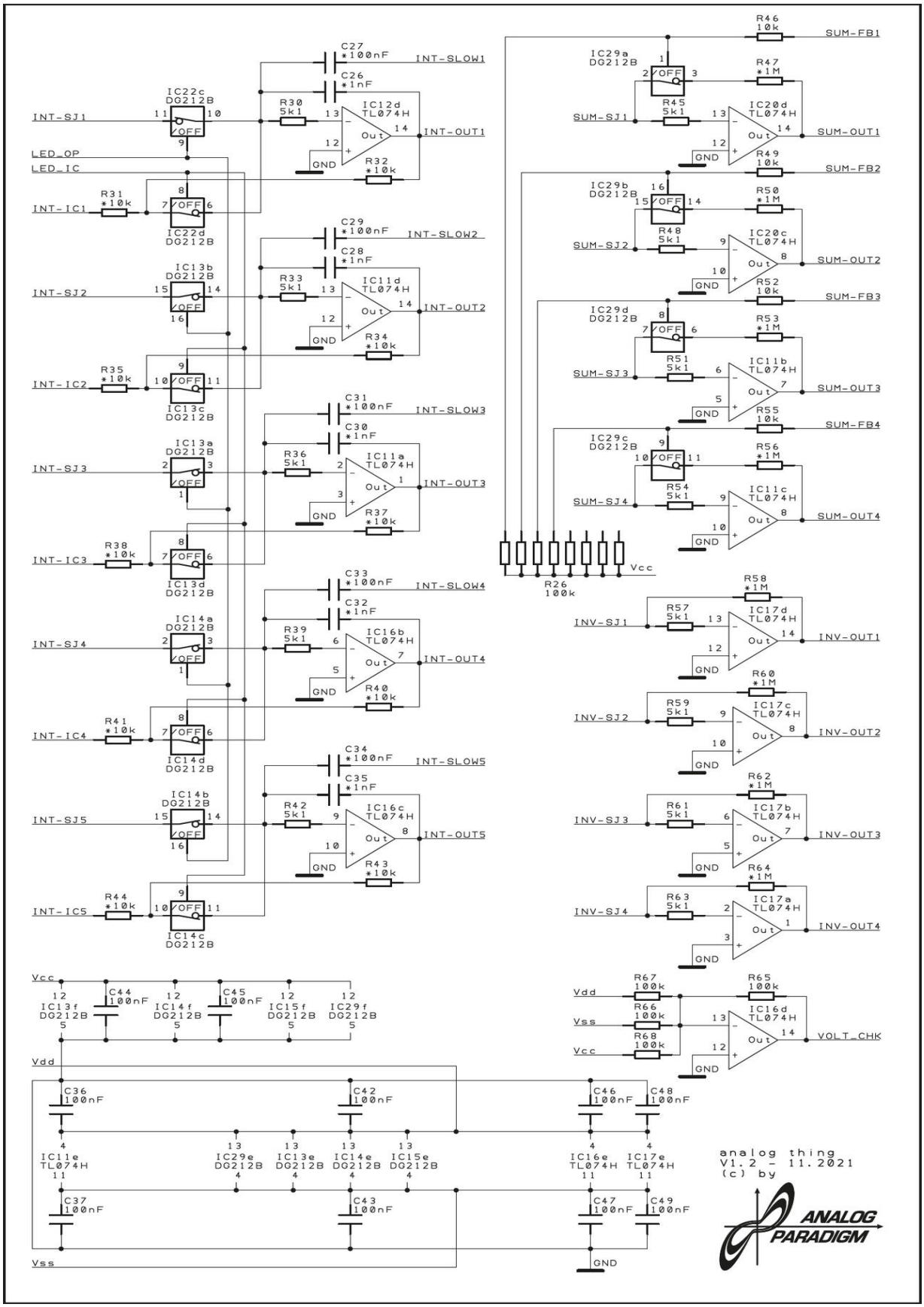

INT-SJ1
INT-SJ2
INT-SJ3
INT-SJ4
INT-SJ5
LED_OP
LED_IC
INT-IC1
INT-IC2
INT-IC3
INT-IC4
INT-IC5
IC22c DG212B
IC22d DG212B
IC13b DG212B
IC13c DG212B
IC13a DG212B
IC13d DG212B
IC14a DG212B
IC14d DG212B
IC14b DG212B
IC14c DG212B
IC12d TL074H
IC11d TL074H
IC11a TL074H
IC16b TL074H
IC16c TL074H
R30 5k1
R33 5k1
R36 5k1
R39 5k1
R42 5k1
R31 *10k
R35 *10k
R38 *10k
R41 *10k
R44 *10k
R32 *10k
R34 *10k
R37 *10k
R40 *10k
R43 *10k
C27 *100nF
C26 *1nF
C29 *100nF
C28 *1nF
C31 *100nF
C30 *1nF
C33 *100nF
C32 *1nF
C34 *100nF
C35 *1nF
INT-SLOW1
INT-SLOW2
INT-SLOW3
INT-SLOW4
INT-SLOW5
INT-OUT1
INT-OUT2
INT-OUT3
INT-OUT4
INT-OUT5
GND
OFF
Out
IC29a DG212B
IC29b DG212B
IC29d DG212B
IC29c DG212B
IC20d TL074H
IC20c TL074H
IC11b TL074H
IC11c TL074H
R45 5k1
R48 5k1
R51 5k1
R54 5k1
R46 10k
R47 *1M
R49 10k
R50 *1M
R52 10k
R53 *1M
R55 10k
R56 *1M
SUM-SJ1
SUM-SJ2
SUM-SJ3
SUM-SJ4
SUM-FB1
SUM-FB2
SUM-FB3
SUM-FB4
SUM-OUT1
SUM-OUT2
SUM-OUT3
SUM-OUT4
R26 100k
Vcc
IC17d TL074H
IC17c TL074H
IC17b TL074H
IC17a TL074H
R57 5k1
R59 5k1
R61 5k1
R63 5k1
R58 *1M
R60 *1M
R62 *1M
R64 *1M
INV-SJ1
INV-SJ2
INV-SJ3
INV-SJ4
INV-OUT1
INV-OUT2
INV-OUT3
INV-OUT4
IC13f DG212B
IC14f DG212B
IC15f DG212B
IC29f DG212B
C44 100nF
C45 100nF
Vdd
Vss
C36 100nF
C42 100nF
C46 100nF
C48 100nF
C37 100nF
C43 100nF
C47 100nF
C49 100nF
IC11e TL074H
IC29e DG212B
IC13e DG212B
IC14e DG212B
IC15e DG212B
IC16e TL074H
IC17e TL074H
R67 100k
R66 100k
R68 100k
R65 100k
IC16d TL074H
VOLT_CHK
analog thing
V1.2 - 11.2021
(c) by
ANALOG PARADIGM

Figure A.3. THAT summers, inverters, and integrators

The five summers basically employ the same structure as the inverters with the addition of an electronic switch that is normally closed. Grounding the FB (feedback) jack on the front panel will cause this switch to open and thus converts the summer into an open amplifier by removing the 1 megaohm feedback resistor from the feedback loop.

The integrators follow the basic circuit shown in →figure 4.24 on page 111. At the heart of each integrator is an operational amplifier with a fixed feedback capacitor of 1 nF[4] yielding a standard time-scaling factor $k_0 = 10^3$ with the 1 megaohm input resistors on the front panel. Connecting the integrator output with the SLOW jack parallels a 100 nF capacitor resulting in $k_0 \approx 10$.[5] Each integrator uses two electronic switches for mode control. Using the part numbers of integrator 1 on the top left of the schematic IC22c enables or disables the input jacks, while IC22d enables or disables the initial condition resistor network.

With the integrator in initial condition mode, IC22c is open and IC22d is closed. This disables all of the input jacks on the front panel and turns the operational amplifier basically into a summer with a 10k input resistor R31 connected to the IC jack and 10k resistor R32 in its feedback path. In this mode of operation, the summing junction at R30 will be held at GND potential while the output of the operational amplifier will be the negative of the voltage applied to the IC connection. Since the integration capacitor(s) are in parallel with R32, they will be precharged to this voltage. Since R31 and R32 are comparatively small with 10k each, the time for settling to the desired initial condition voltage is minimized.

In OP mode, IC22d is opened while IC22c is closed, thus the initial condition resistor network is deactivated (it is still

connected to the integrator's output and is a certain but negligible load) while the input connections on the front panel are now routed to the summing junction at the left-hand side of the protection resistor R30. Opening both switches sets the integrator to HALT mode where it holds its last value (apart from some unavoidable drift).

All of these switch pairs in the five integrators are connected to two control lines, `LED_OP` and `LED_IC`, which are generated by the central mode control unit.

→ Figure A.4 shows the circuitry for the two multipliers, two comparators, and the global overload detector. At the heart of the multipliers are two low cost analog multiplier ICs type AD633.[6] This integrated circuit typically requires a supply voltage of ±15 V while the THAT uses ±12 V. To achieve high accuracy despite this low supply voltage, a little trick is employed. The *x* and *y* input values are fed to a 2 : 1 voltage divider each before being connected to the multiplier. Since the output of the (still rather expensive) multiplier needs a buffer amplifier to protect the multiplier itself against patching errors (such as connecting a machine unit to its output), the output buffer has a gain of 4, thus compensating the downscaled input values.

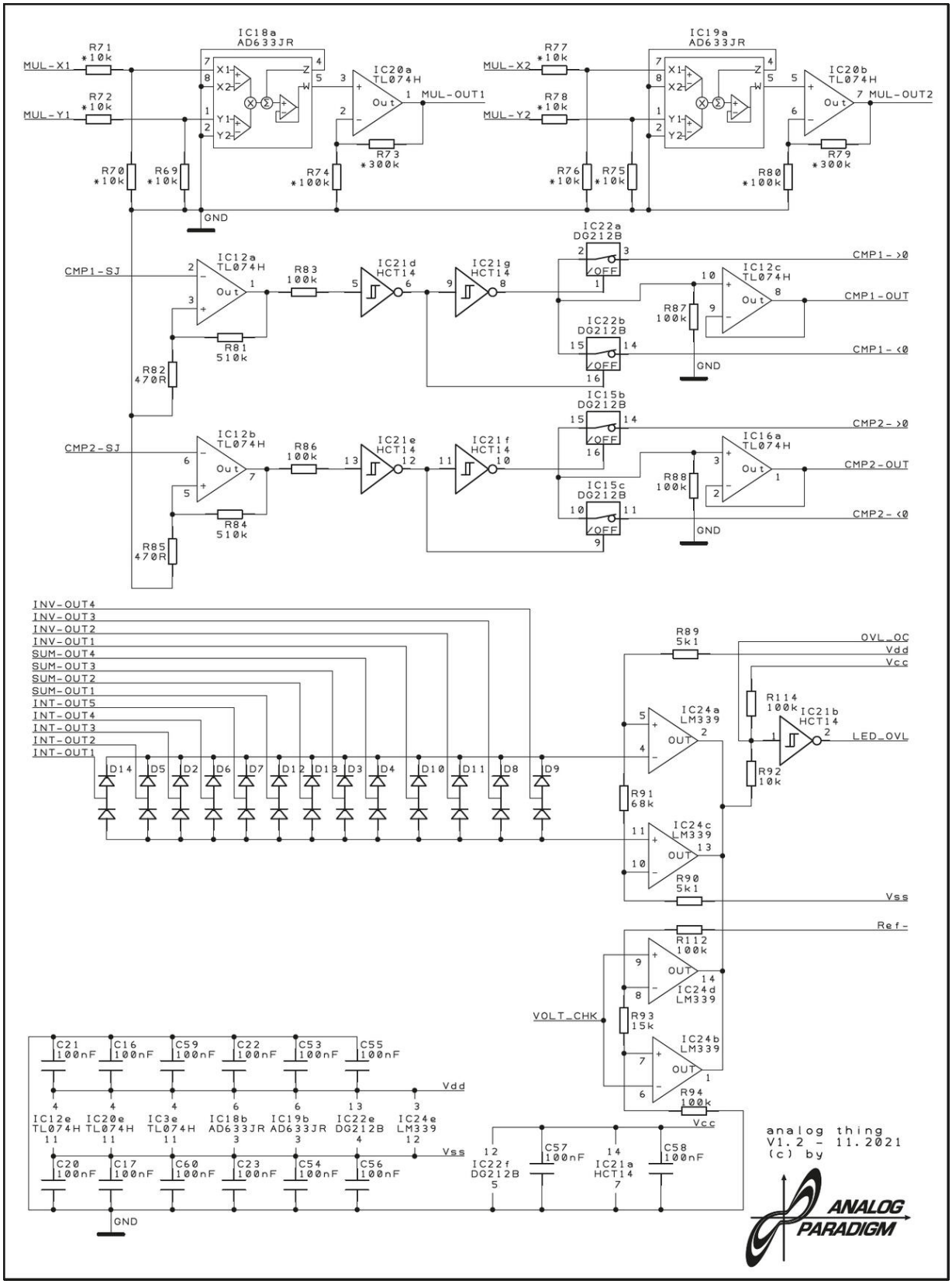

IC18a AD633JR
IC19a AD633JR
IC20a TL074H
IC20b TL074H
MUL-X1
MUL-Y1
MUL-X2
MUL-Y2
MUL-OUT1
MUL-OUT2
R71 *10k
R72 *10k
R77 *10k
R78 *10k
R70 *10k
R69 *10k
R76 *10k
R75 *10k
R73 *300k
R79 *300k
R74 *100k
R80 *100k
GND
IC12a TL074H
IC12b TL074H
IC12c TL074H
IC21d HCT14
IC21e HCT14
IC21f HCT14
IC21g HCT14
IC22a DG212B
IC22b DG212B
IC15b DG212B
IC15c DG212B
CMP1-SJ
CMP2-SJ
R83 100k
R86 100k
R87 100k
R88 100k
R81 510k
R84 510k
R82 470R
R85 470R
CMP1->0
CMP1-OUT
CMP1-<0
CMP2->0
CMP2-OUT
CMP2-<0
INV-OUT4
INV-OUT3
INV-OUT2
INV-OUT1
SUM-OUT4
SUM-OUT3
SUM-OUT2
SUM-OUT1
INT-OUT5
INT-OUT4
INT-OUT3
INT-OUT2
INT-OUT1
D14 D5 D2 D6 D7 D12 D13 D3 D4 D10 D11 D8 D9
R89 5k1
R90 5k1
R91 68k
R92 10k
R93 15k
R94 100k
R112 100k
R114 100k
IC24a LM339
IC24b LM339
IC24c LM339
IC24d LM339
IC21b HCT14
IC16a TL074H
OVL_OC
Vdd
Vcc
LED_OVL
Vss
Ref-
VOLT_CHK
C21 100nF
C16 100nF
C59 100nF
C22 100nF
C53 100nF
C55 100nF
C20 100nF
C17 100nF
C60 100nF
C23 100nF
C54 100nF
C56 100nF
C57 100nF
C58 100nF
IC12e TL074H
IC20e TL074H
IC3e TL074H
IC18b AD633JR
IC19b AD633JR
IC22e DG212B
IC24e LM339
IC22f DG212B
IC21a HCT14
analog thing
V1.2 - 11.2021
(c) by
ANALOG PARADIGM

Figure A.4. THAT multipliers, comparators, and overload detection

The comparators also use standard operational amplifiers instead of dedicated comparator integrated circuits to keep cost down. These work as inverting amplifiers in open loop configuration with some hysteresis added by a voltage divider connecting the non-inverting input to the amplifier's output. This is connected to two digital SCHMITT trigger inverter gates 74HCT14 generating a complementary control signal pair used for two electronic switches which thus basically form a SPDT switch. An additional operational amplifier serves as an output buffer and protects the switches from patching errors.

Larger analog computers typically have a dedicated overload detection circuit for every computing element capable of overloading (summers and integrators, mainly). To keep costs down, THE ANALOG THING has a global overload detector consisting of a window comparator (IC24a and IC24c) driving a SCHMITT trigger digital inverter gate (IC21b) controlling the central overload LED. This window comparator is connected to all outputs of the computing elements by means of thirteen decoupling diode pairs.

The circuitry for the eight coefficient potentiometers with associated digital readout voltmeter is shown in →figure A.5. Each coefficient is basically just a voltage divider built from a linear 100k potentiometer. Each of these potentiometers has an associated buffer amplifier which serves the dual purpose of protecting the potentiometers against patching errors and to eliminate load at the potentiometer wiper. To set these potentiometers with a precision of up to three decimal places, a small digital readout voltmeter is provided.

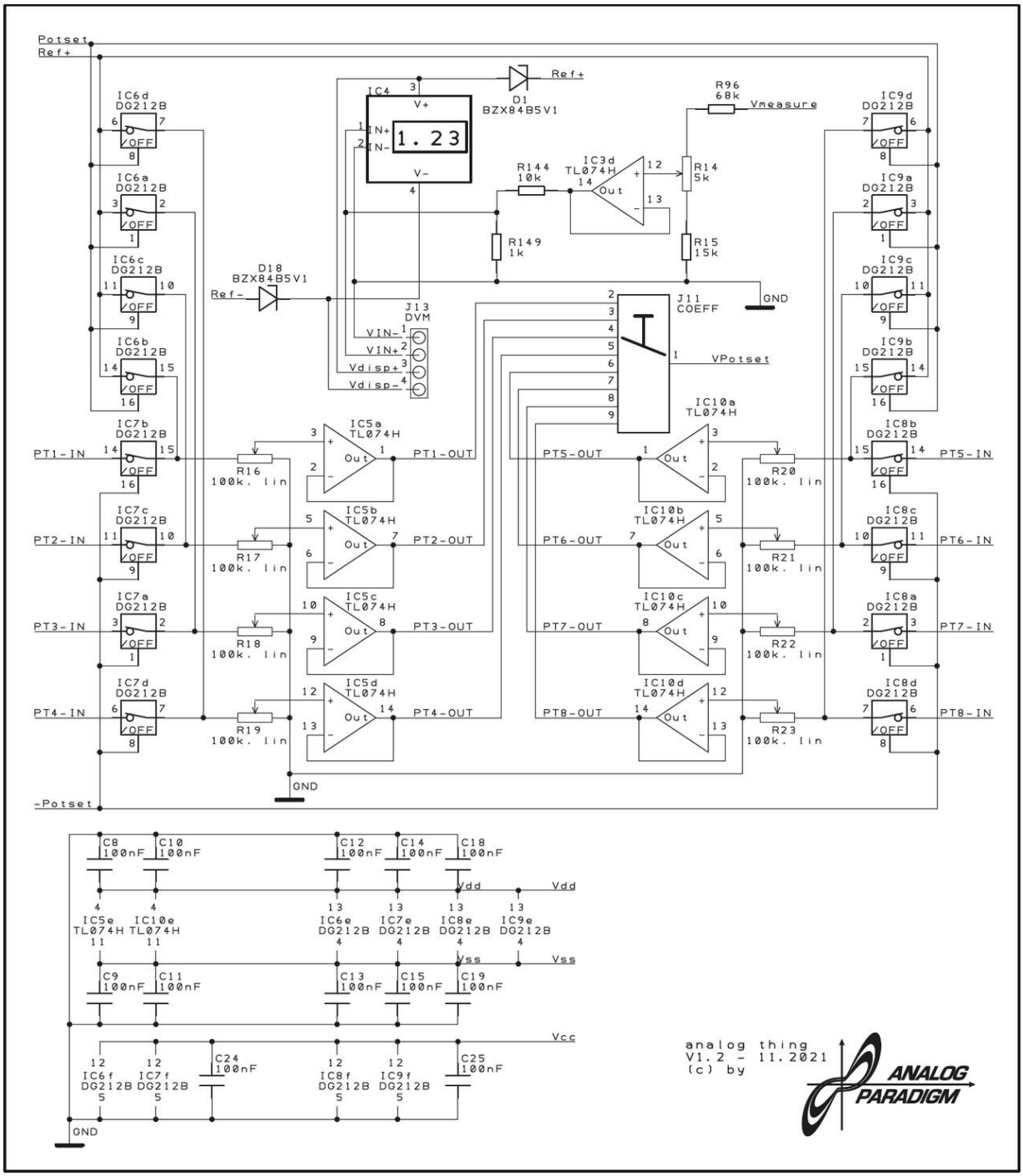


Figure A.5. THAT coefficient and digital readout voltmeter

To setup the coefficients, the analog computer is switched into COEFF mode. This disconnects the coefficient input connections

at the front panel and connects all potentiometer inputs to the positive machine unit of +10 V. This is done by sixteen electronic switches controlled by the digital signal `Potset`. The panel voltmeter typically requires a galvanically isolated power supply such as a battery. Here it is powered from the reference voltages `Ref-` and `Ref+`, using two Z-diodes to drop the voltage enough to not exceed the operational limits of the voltmeter. The signal to be measured, `Vmeasure`, is fed to a buffer amplifier by means of an adjustable voltage divider consisting of R96, R14, and R15. `Vmeasure` is connected to `VPotset` by an electronic switch shown in →figure A.7 when THE ANALOG THING is set to COEFF mode. When set to REP or REPF mode (repetitive operation), `Vmeasure` is connected to the output of the OP time control potentiometer while it shows the signal at the U jack on the front panel in OP mode.

→Figure A.6 shows the circuitry for the hybrid port while allowing remote control of THE ANALOG THING by means of an attached microcontroller or an equivalent device and the level shifters for readout. On the back of the computer are five RCA connectors labeled X, Y, Z, U, and TRIG. The first four are connected to the corresponding jacks on the front panel by means of ≈ 10 : 1 voltage dividers, thus making it possible to directly connect these outputs to the inputs of a sound card as a cheap drop in replacement for an oscilloscope. The TRIG output is connected to the `TRIG` signal, which is basically the OP control signal. This can be used to trigger some data capture or display device such as an oscilloscope with the analog computer running in repetitive mode.

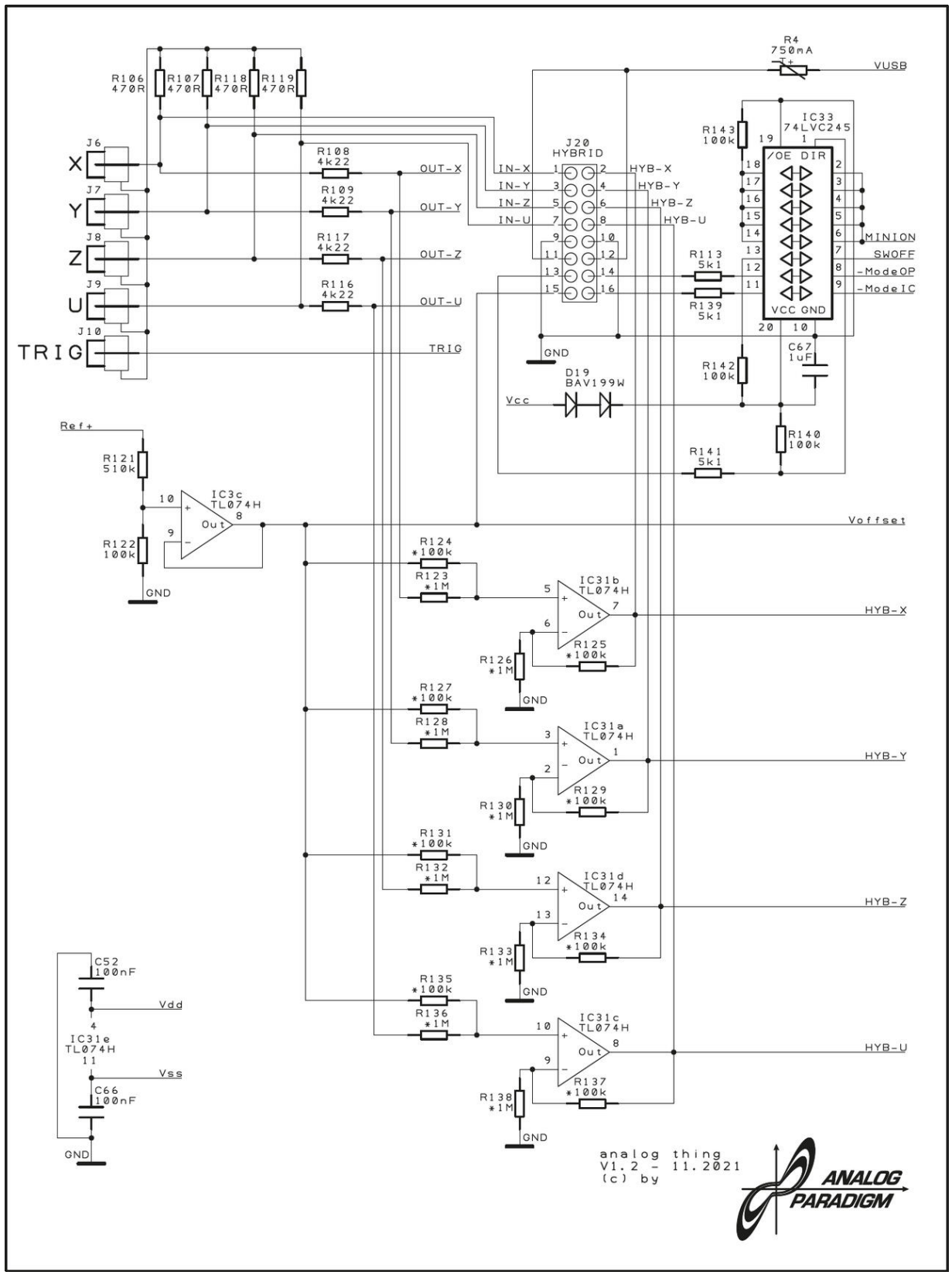

R106 470R
R107 470R
R118 470R
R119 470R
J6
X
J7
Y
J8
Z
J9
U
J10
TRIG
R108 4k22
R109 4k22
R117 4k22
R116 4k22
OUT-X
OUT-Y
OUT-Z
OUT-U
TRIG
J20 HYBRID
IN-X
IN-Y
IN-Z
IN-U
HYB-X
HYB-Y
HYB-Z
HYB-U
R4 750mA
VUSB
IC33 74LVC245
/OE DIR
VCC GND
R143 100k
R113 5k1
R139 5k1
R142 100k
C67 1uF
R140 100k
R141 5k1
MINION
SWOFF
-ModeOP
-ModeIC
GND
D19 BAV199W
Vcc
Ref+
R121 510k
R122 100k
IC3c TL074H
Voffset
R124 *100k
R123 *1M
R125 *100k
R126 *1M
IC31b TL074H
R127 *100k
R128 *1M
R129 *100k
R130 *1M
IC31a TL074H
R131 *100k
R132 *1M
R134 *100k
R133 *1M
IC31d TL074H
R135 *100k
R136 *1M
R137 *100k
R138 *1M
IC31c TL074H
C52 100nF
Vdd
IC31e TL074H
Vss
C66 100nF
analog thing
V1.2 - 11.2021
(c) by
ANALOG PARADIGM

Figure A.6. THAT hybrid port and level shifters

The hybrid port allows THE ANALOG THING to be controlled by three digital signals `ModeOP` (pin 14), `ModeIC` (pin 16), and an `enable` signal (pin 13). As long as `enable` is high, the analog computer can be controlled from its front panel. Pulling `enable` low places the analog computer under hybrid control. Now, `ModeOP` and `ModeIC` are controlled via the hybrid port.

Since most modern microcontrollers offer a number of on-chip analog/digital converters, the hybrid port also supports four analog readout signals `HYB-X`, `HYB-Y`, `HYB-Z`, and `HYB-U`. These correspond to X, Y, Z, and U but are level shifted to be compatible with analog/digital converters with an input range of 0 to 3.3 V. This is done using the four operational amplifiers IC31a, IC31b, IC31c, and IC31d, using an offset signal derived from `Ref+` via the amplifier IC3c.

The most complex part of THE ANALOG THING is the control logic and power supply shown in →figure A.7. The power supply unit is shown on top. Power is connected via the USB-C connector on the back of the computer. The 5 V are controlled by a high-side-switch BD2244G-M (IC30) which is enabled via pin 3 which is connected to a form of a wired-or to the mode switch. This switches the computer on in every switch position except OFF. R24 sets the current limit to about $19364 \cdot 33k^{-0.98}.^{7}$ These 5 V are directly used to power all of the digital logic.

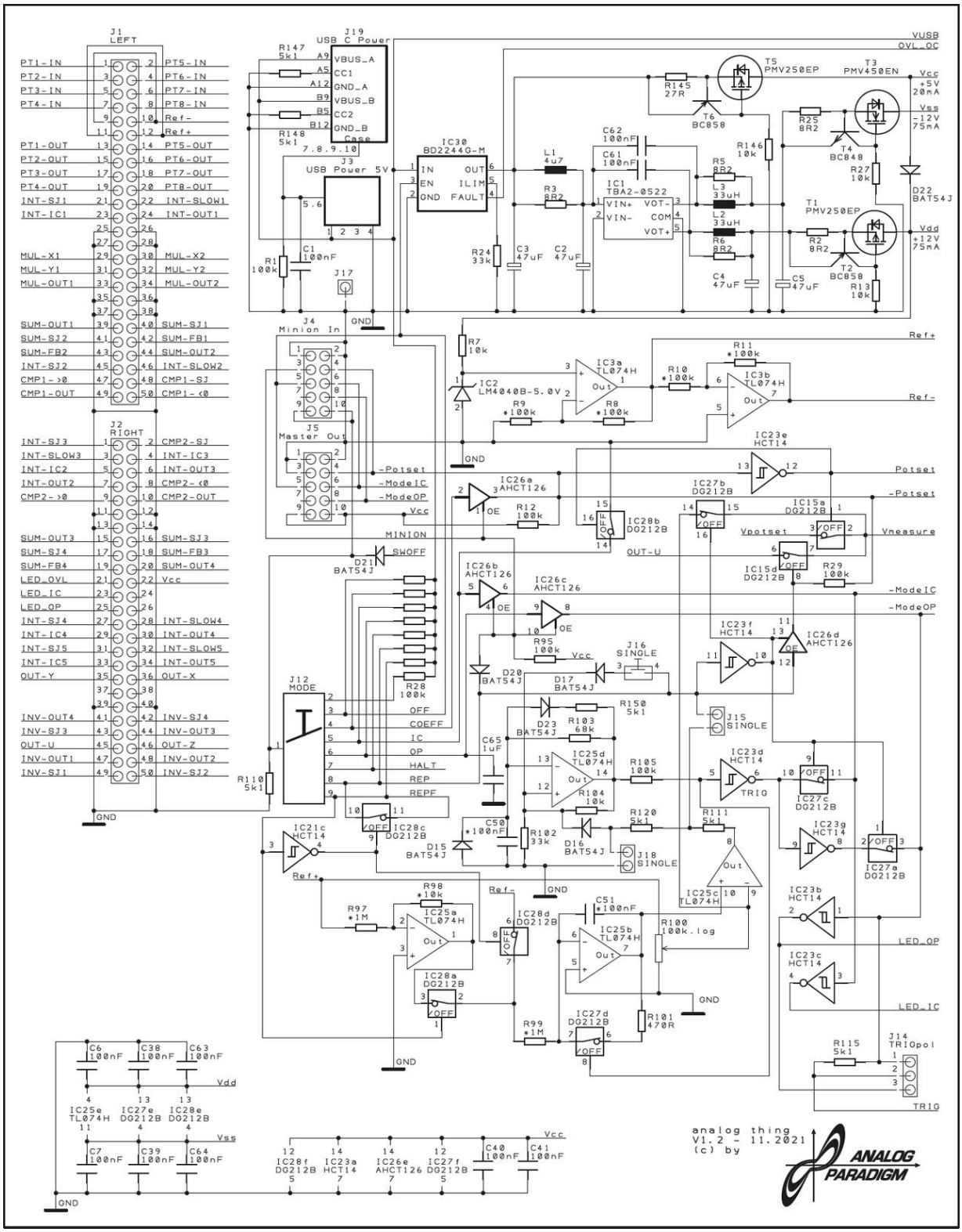


Figure A.7. THAT power supply and mode control circuitry

The analog support voltages of ±12 V are generated by a DC/DC converter (IC1). All three supply voltages are protected against short circuits by combinations of a bipolar transistor and a MOSFET.

The all-important reference voltages `Ref+` and `Ref−` are derived from a precision micropower shunt voltage reference type LM4040B-5.0V. The output of this reference is buffered by a non-inverting operational amplifier (IC3a) yielding `Ref+`, while `Ref-` is generated using an inverting amplifier IC3b.

In the operating mode MINION the system is controlled via its Minion In port. This, in conjunction with the corresponding Minion Out port on another THAT, allows several of these analog computers to be run in a daisy-chained fashion with one, the master, controlling the minions.

The current mode of operation is selected by switch J12. The modes OFF, COEFF, IC, OP, and HALT are self-explanatory. The two repetitive operation modes, REP and REPF, are more complicated. In both cases a ramp generator consisting of amplifier IC25b and an integration capacitor C51 determines the time spent in OP mode. The ramp output of this integrator is compared to the value set at the OP-TIME potentiometer R100.[8] Using a number of electronic switches, the integration capacitor can be reset, thus initiating the IC portion of the repetition cycle.[9]

Noteworthy in general are the decoupling capacitors located next to every integrated circuit. These are all 100nF ceramic capacitors. They are shown in the lower part of each of the schematics (except the front panel drawings).

© 2026 Walter de Gruyter GmbH, Berlin/Boston 2026 Walter de Gruyter GmbH, Berlin/Boston

Bibliography

[Abramovitch 2005] [Abramovitch 2005] Daniel Abramovitch, “Analog Computing in the Soviet Union”, in *IEEE Control Systems Magazine*, June 2005, pp. 52–62 →

[Achour et al. 2020] [Achour et al. 2020] Sara Acour, Martin Rinard, “Noise-Aware Dynamical System Compilation for Analog Devices with Legno”, in *ASPLOS'20*, March 16–20, 2020, →https://doi.org/10.1145/3373376.3378449, retrieved 31.08.2025 →

[Adamatzky et al. 2005] [Adamatzky et al. 2005] Andrew Adamatzky, Benjamin De Lacy Costello, Tetsuya Asai, *Reaction-Diffusion Computers*, Elsevier Science, 2005 →

[Addicott et al. 1968] [Addicott et al. 1968] E. W. Addicott, R. W. Jones, *Solution of the Catapult Take-off Performance Equations by an Analogue Method*, Ministry of Technology, Aeronautical Research Council, London, Her Majesty's Stationary Office, 1968 →

[Adler 1968] [Adler 1968] Helmut Adler, *Elektronische Analogrechner*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1968 →

[Admiralty 1943] [Admiralty 1943] N. N., *Handbook of the Admiralty Fire Control Clock Marks I and I**, Admiralty, S.W.1., Gunnery Branch, 1943 →

[Akers 1977] [Akers 1977] Lex A. Akers, “Simulation of semiconductor devices”, in *SIMULATION*, August 1977, pp. 33–41 →

[Alaghi et al. 2013] [Alaghi et al. 2013] Armin Alaghi, John P. Hayes, “Survey of Stochastic Computing”, in *ACM Transactions on Embedded Computing Systems*, Vol. 12, No. 2s, Article 92, May 2013 a, b

[Albrecht 1968] [Albrecht 1968] Peter Albrecht, “Über die Simulation der respiratorischen Arrhythmie auf einem Analogrechner”, Technische Mitteilungen AEG-Telefunken, *2. Beiheft Datenverarbeitung*, 1968, pp. 13–15 a, b

[Albrecht et al.] [Albrecht et al.] P. Albrecht, H. Lotz, “Einsatz der hybriden Präzisionsrechenanlage RA 770 zur automatischen Parameteroptimierung nach dem Gradientenverfahren”, AEG-Telefunken, AFA 005 0670, n. d. →

[Aldrich et al. 1955] [Aldrich et al. 1955] H. P. Aldrich, H. M. Paynter, “First Interim Report – Analytic Studies of Freezing and Thawing of Soils (for the Arctic Construction and Frost Effects Laboratory New England Division, Corps of Engineers)”, in [Paynter ed. 1955, pp. 247–260] a, b

[Ameling 1962/1] [Ameling 1962/1] Walter Werner Ameling, “Die Lösung partieller Differentialgleichungen und ihre Darstellungsmöglichkeiten auf dem elektronischen Analogrechner”, in *elektronische datenverarbeitung*, No. 5/1962, pp. 197–215 →

[Ameling 1962/2] [Ameling 1962/2] Walter Werner Ameling, “Der Einsatz des elektronischen Analogrechners zur Rohrnetzberechnung kompressibler und inkompressibler Stoffströme”, in *Elektronische Rechenanlagen*, 4 (1962), No. 3, pp. 109–116 →

[Ameling 1963/1] [Ameling 1963/1] Walter Werner Ameling, *Aufbau und Wirkungsweise elektronischer Analogrechner*, Vieweg Verlag, 1963 a, b, c

[Ameling 1963/2] [Ameling 1963/2] Walter Werner Ameling, “Aufbau und Arbeitsweise des Hybrid-Rechners TRICE”, in *Elektronische Rechenanlagen*, 5 (1963), No. 1, pp. 28–41 a, b, c, d

[Ameling 1964] [Ameling 1964] Walter Werner Ameling, “Die Entwicklung verbesserter Ersatzschaltungen mit Hilfe der Differenzmethode”, in *Elektronische Rechenanlagen*, 6 (1964), No. 1, pp. 35–41 →

[Ammon] [Ammon] W. Ammon, *Der elektronische Analogrechner und seine Verwendung in der Industrie*, AEG, n. d. →

[Ammon et al. 1959] [Ammon et al. 1959] W. Ammon, G. Schneider, “Beispiele zur Lösung technischer Probleme mit dem Analogrechner”, in *Elektronische Rechenanlagen*, 1 (1959), No. 1, pp. 29–34 →

[Analog Devices/1] [Analog Devices/1] Analog Devices, Inc., *Low Cost Analog Multiplier AD633*, →https://www.analog.com/media/en/technical-documentation/data-sheets/ad633.pdf, n. d., retrieved 20.03.2025 →

[Analog Devices/2] [Analog Devices/2] Analog Devices, Inc., *AD 639 Universal Trigonometric Function Converter*, →https://www.analog.com/media/en/technical-documentation/obsolete-data-sheets/ad639.pdf, n. d., retrieved 13.03.2022 →

[Analog Devices 2006] [Analog Devices 2006] Analog Devices Inc., *Op Amp Applications Handbook*, Elsevier Newnes, 2006

[Analog Devices 2008] [Analog Devices 2008] Analog Devices Inc., *Analog Multipliers*, MT-079 Tutorial, 2008 →

[Anders] [Anders] Udo Anders, “Early Ideas in the History of Quantum Chemistry”, in →http://www.quantum-chemistry-history.com, n. d., retrieved 24.03.2008 →

[Anderson et al. 1967] [Anderson et al. 1967] Max. D. Anderson, Someshwar C. Gupta, “Backward time analog computer solutions of optimum control problems”, in *AFIPS ’67 (Spring): Proceedings of the April 18–20, 1967, spring joint computer conference*, April 1967, pp. 133–139 →

[Anderson et al. 2003] [Anderson et al. 2003] P. M. Anderson, A. A. Fouad, *Power System Control and Stability*, second edition, IEEE Press Power Engineering Series, 2003 a, b, c

[Ando 1971] [Ando 1971] Joji Ando, “Simulation Study of Surge Tank System for Hydro Power Plant by Using Hybrid Electronic Computer”, in *The Second International JSME Symposium Fluid Machinery and Fluidics*, Tokyo, September 1971, pp. 259–147 →

[Angelo 1952] [Angelo 1952] E. J. Angelo, *An Electron-Beam Tube for Analog Multiplication*, Technical Report No. 249, October 27, 1952, Research Laboratory of Electronics, Massachusetts Institute of Technology →

[Andrews 1960/1] [Andrews 1960/1] Jack M. Andrews, “The Dynamic Storage Analog Computer – DYSTAC”, in *Instruments & Control Systems*, Vol. 33, September 1960, pp. 1540–1544 →

[Andrews 1960/2] [Andrews 1960/2] Jack M. Andrews, “Mathematical applications of the dynamic storage analog computer”, in *Western Joint IRE-AIEE-ACM Computer Conference*, May 3–5, 1960, pp. 119–131 →

[Apalovičová 1979] [Apalovičová 1979] R. Apalovičová, “Simulation of MOS Transistor Structure on Hybrid Computer Systems”, in Tagungsband *SIMULATION OF SYSTEMS ’79*, pp. 1013–1019 →

[Appleton et al. 1975] [Appleton et al. 1975] Jon H. Appleton, Ronald C. Perera, *The Development of and Practice of Electronic Music*, Prentice-Hall, New Jersey, 1975 →

[Appleyard et al. 1957] [Appleyard et al. 1957] Victor A. Appleyard, Frank P. Linaweaver Jr., “The McIlroy Fluid Analyzer in Water Works Practice”, in *Journal AWWA*, Volume 49, No. 1, January 1957, pp. 15–20 →

[Apter 1967] [Apter 1967] H. Apter, “Traitement par voie Analogique du «Probleme Maremotrice»”, in [IACM 1976, pp. 821–824] →

[Archibald 2014] [Archibald 2014] Herbert L. Archibald, “The Enigma of the 10-year Wildlife Population Cycle Solved? Evidence that the Periodicity and Regularity of the Cycle Are Driven by a Lunar Zeitgeber”, in *The Canadian Field-Naturalist*, Vol. 128, No. 4, October-December 2014, pp. 328–340 →

[Argyris et al. 1995] [Argyris et al. 1995] John Argyris, Gunter Faust, Maria Haase, *Die Erforschung des Chaos – Studienbuch für Naturwissenschaftler und Ingenieure*, Vieweg, 1995 →

[Armstrong et al. 1962] [Armstrong et al. 1962] N. A. Armstrong, E. C. Holleman, “A Review of In-Flight Simulation Pertinent to Piloted Space Vehicles”, AGARD Report 403, July 1962 →

[Arpasi et al. 1967] [Arpasi et al. 1967] Dale J. Arpasi, Clint E. Hart, *Controls Analysis of Nuclear Rocket Engine at Power Range Operating Conditions*, NASA technical note, NASA TN D-3978, 1967 →

[Arscott 1964] [Arscott 1964] F. M. Arscott, *Periodic Differential Equations – An Introduction to Mathieu, Lamé, and Allied Functions*, The MacMillan Company, New York, 1964 →

[Aschoff 1938] [Aschoff 1938] V. Aschoff, “Der Sternmodulator als Doppelgegentaktmodulator”, in *Telegraphen-Fernsprech-Funk- und Fernsehtechnik*, Bd. 27, No. 10, 1938, p. 379–383 →

[Ascoli 1947] [Ascoli 1947] Guido Ascoli, “Vedute sintetiche sugli strumenti integratori”, in *Rend. Sem. Mat. Fis. Milano*, 18:36, 1947

→

[Ashley et al. 1968] [Ashley et al. 1968] J. Robert Ashley, Thomas E. Bullock, “Hybrid computer integration of partial differential equations by use of an assumed sum separation of variables”, in *AFIPS ’68 (Fall, part 1): Proceedings of the December 9–11, 1968, fall joint computer conference, part 1*, December 1968, pp. 585–591 →

[Ashley et al. 1985] [Ashley et al. 1985] Holt Ashley, Marten Landahl, *Aerodynamics of Wings and Bodies*, Dover Publications, Inc., New York, 1985 (unabridged reprint from 1965) →

[Asmar et al. 2018] [Asmar et al. 2018] Nakhlé Asmar, Loukas Grafakos, *Complex Analysis with Applications*, Springer, 2018 →

[ASME 1990] [ASME 1990] N. N., *Southern Gas Association Analog – A National Historic Mechanical Engineering Landmark*, The American Society of Mechanial Engineers, 1990 →

[Aspray et al. 1990] [Aspray et al. 1990] William Aspray (ed.), *Computing before Computers*, Iowa State University Press / Ames, 1991 →

[Aspray et al. 1991] [Aspray et al. 1991] William Aspray, Ewdin L. Harder, *Oral-History: Edwin Harder*, →https://ethw.org/Oral-History:Edwin_Harder, retrieved 08.05.2025 a, b, c

[Aspray 1993] [Aspray 1993] William Aspray, “Edwin L. Harder and the Anacom: analog computing at Westinghouse”, in *IEEE Annals of the History of Computing*, Vol. 15, No. 2, pp. 35–52 a, b, c, d, e

[Athans et al. 1967] [Athans et al. 1967] Michael Athans, Willam S. Levine, Alexander H. Lewis, “A System for the Optimal and Suboptimal Position and Velocity Control for a String of High-Speed Vehicles”, in [IACM 1976, pp. 746–760] →

[Atherton 1955] [Atherton 1955] E. Atherton, “The relation of the reflectance of dyed fabrics to dye concentration and the

instrumental approach to colour matching", in *Journal of the Society of Dyers and Colorists*, 71, 1955, pp. 389-398 →

[Atherton 2004] [Atherton 2004] Derek P. Atherton, "Some reflections on analogue simulation and control engineering", in *Measurement+Control*, Vol. 37/10, December 2004, pp. 300–306 a, b

[Athreyas et al. 2018] [Athreyas et al. 2018] Nihar Athreyas, Wenhao Song, Blair Perot, Qiangfei Xia, Abbie Mathew, Jai Gupta, Dev Gupta, J. Joshua Yang, "Memristor-CMOS Analog Coprocessor for Acceleration of High-Performance Computing Applications", in ACM Journal on Emerging Technologies in Computing Systems, Volume 14, Issue 3, July 2018, Article No.: 38, pp. 1–30 →

[Atkinson 1955/1] [Atkinson 1955/1] Cyril Atkinson, "Polynomial Root Solving on the Electronic Differential Analyser (A Technique for Finding the Real and Complex Roots of a Polynomial Using an Electronic Differential Analyser)", in *Mathematical Tables and Other Aids to Computation*, Vol. 9, No. 52, Oct. 1955, pp. 139–143 →

[Atkinson 1955/2] [Atkinson 1955/2] Cyril P. Atkinson, "A Mechanical Analog Computer for Solving Linear Simultaneous Algebraic Equations", in *Computers and Automation*, 1955, Vol. 4, No. 3, pp. 12–15 →

[Aude et al. 1936] [Aude et al. 1936] Erich Aude, Ernst Reipert, *Gezeitenrechenmaschine*, Patentschrift Nr. 682836, Klasse 42m, Gruppe 36, A 78729 IX b/42 m, patentiert vom 6. März 1936 →

[Bader 1985] [Bader 1985] Heinz Bader, *Operationsverstärker – Grundlagen und Anwendungen*, Karamanolis Verlag, 1985 →

[Bärwolff 2021] [Bärwolff 2021] Günter Bärwolff, "Modelling of COVID-19 propagation with compartment models", in

Mathematische Semesterberichte, Springer, Band 68, Heft 2, Oktober 2021, pp. 181–219 a, b

[Bai et al. 2015] [Bai et al. 2015] Yu Bai, Mingjie Lin, “Energy-Efficient Discrete Signal Processing with Field Programmable Analog Arrays (FPAAs)”, in *FPGA ’15: Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, February 2015, pp. 84–93 →

[Baker 1978] [Baker 1978] Bruce Baker, “Martin-Marietta Aerospace Simulation & Test Laboratory”, handout for a talk delivered at a Simulation Councils Conference, San Francisco, 1978, author’s archive a, b, c, d, e

[Balaban] [Balaban] Philip Balaban, *HYPAC – A hybrid-computer circuit simulation program*, Bell Telephone Laboratories, Holmdel, New Jersey, n. d. a, b

[Balaban et al. 1968] [Balaban et al. 1968] Philip Balaban, John Logan, “Analog computer simulation of semiconductor circuits”, in *Spring Joing Computer Conference*, 1968, pp. 135–142 a, b, c, d

[Bard 1965] [Bard 1965] M. Bard, “Schaltung zur automatischen Aufzeichnung von Frequenzgängen mit Analogrechner und Koordinatenschreiber”, in *Elektronische Rechenanlagen*, 7 (1965), No. 1, pp. 29–33 →

[Barnett 1963] [Barnett 1963] Robert M. Barnett, *NASA-AMES Hybrid Computer Facilities and Their Application to Problems in Aeronautics*, Ames Research Center, Moffett Field, California, 1963 →

[Barth 1976] [Barth 1976] H.-J. Barth, “Simulation im Maschinenbau zur Festigkeitsermittlung und zur Untersuchung physikalischer Zusammenhänge”, in [Schöne 1976/1, pp. 581–602] →

[Barzacchi et al. 1967] [Barzacchi et al. 1967] Marco Barzacchi, Paolo Giordano, Agostino Mathis, “Development and Use of a Non-Linear Model for a Direct Cycle Nuclear Power Plant”, in [IACM 1976, pp. 1056–1064] →

[Bashe et al. 1986] [Bashe et al. 1986] Charles J. Bashe, Lyle R. Johnson, John H. Palmer, Emerson W. Pugh, *IBM’s Early Computers*, The MIT Press, 1986 a, b, c

[Basin 1954] [Basin 1954] Michael Abram Basin, *Electric Analog Computer Study of Supersonic Flutter of Elastic Delta Wings*, Thesis, California Institute of Technology, Pasadena, 1954 →

[Bassano et al. 1976] [Bassano et al. 1976] J. C. Bassano, Y. Lennon, J. Vignes, “An Identification of Parameters on a Hybrid-Computer”, in *Trans. IMACS*, Vol. XVIII, No. 1, Jan. 1976, pp. 3–7 →

[Bate et al. 1971] [Bate et al. 1971] Roger R. Bate, Donald D. Mueller, Jerry E. White, *Fundamentals of Astrodynamics*, Dover Publications, Inc., 1971 →

[Basu et al. 2010] [Basu et al. 2010] Arindam Basu, Stephen Brink, Craig Schlottmann, Shubha Ramakrishnan, Csaba Petre, Scott Koziol, Faik Baskaya, Christopher M. Twigg, Paul Hasler, “A Floating-Gate-Based Field-Programmable Analog Array”, in *IEEE Journal of Solid-State Circuits*, Vol. 45, No. 9, September 2010, pp. 1781–1794 →

[Bauer 1953] [Bauer 1953] Louis Bauer, “New Laboratory for Three-Dimensional Guided Missile Simulation”, in *AIEE-IRE ’53 (Western) Proceedings of the February 4–6, 1953, western computer conference*, pp. 187–195 a, b

[Bauer et al. 1956] [Bauer et al. 1956] Walter F. Bauer, George P. West, “A System for General-Purpose Analog-Digital Computation”, in *Proceedings of the 1956 11th ACM national meeting*, January 1956, pp. 79–82 →

[Bauer et al. 1957] [Bauer et al. 1957] Louis Bauer, A. Karen, B. Loveman, “Solution of Large Problems at Project Cyclone”, in [White Sands 1957, pp. 211-239] a, b

[Bauer 1962/1] [Bauer 1962/1] Louis Bauer, “Aircraft, Autopilot, and Missile Problems”, in [Huskey et al. 1962, pp. 5-49–5-64] a, b

[Bauer 1962/2] [Bauer 1962/2] Louis Bauer, “Partial System Tests and Flight Tables”, in [Huskey et al. 1962, pp. 5-64–5-71] →

[Baun 1970] [Baun 1970] P. J. Baun Jr., “Hybrid Computers: Valuable Aids in Transmission Studies”, in *Bell Laboratories Record*, June/July 1970, pp. 181–185 →

[Beck et al. 1958] [Beck et al. 1958] Robert M. Beck, Max Palevsky, “The DDA”, in *Instruments and Automation*, November 1958, pp. 1836–1837 →

[Beck et al. 1970] [Beck et al. 1970] Charles H. Beck, Ming H. Kuo, *Designer’s Manual for Circuit Design by Analog/Digital Techniques*, Systems Laboratory, School of Engineering, Tulane University, 1970 →

[Beckman Instruments Inc./1] [Beckman Instruments Inc./1] Beckman Instruments, Inc., *Preparing Problems for Solution on an Analog Computer*, n. d. →

[Beckman Instruments Inc./2] [Beckman Instruments Inc./2] Beckman Instruments, Inc., *Beckman – Analog Computing Systems*, n. d. →

[Beckman 1963] [Beckman 1963] N. N., *The Analog Computer that pumped blood*, advertisement in *Instrumente & Control Systems*, Job L-7110, Ad E-2, 1963 →

[Bedford et al. 1952] [Bedford et al. 1952] Leslie Herbert Bedford, John Bell, Eric Miles Langham, *Electrical Fire Control Calculating Apparatus*, United States Patent 2623692, Dec. 30, 1952 →

[Bedient et al. 1968] [Bedient et al. 1968] Conrad K. Bedient, Larry L. Dike, “The Lockheed hybrid system – A giant step”, in *Fall Joing Computer Conference*, 1968, pp. 663–682 →

[Beecham et al. 1965] [Beecham et al. 1965] L. J. Beecham, W. L. Walters, D. W. Partridge, *Proposal for an Integrated Wind Tunnel-Flight Dynamics Simulator System*, Ministry of Aviation, Aeronautical Research Council, 1965 →

[Behrendt 1965] [Behrendt 1965] E. Behrendt, “Wunschautos auf Knopfdruck”, in *hobby – Das Magazin der Technik*, Nr. 6/65, pp. 36–41 →

[Beil 2021] [Beil 2021] Martin Beil, *Entwicklung und Anwendung eines Analogcomputers mit FPAA Schaltkreisen*, Bachelorarbeit im Studiengang Elektrotechnik, B-TU Cottbus-Senftenberg, 2021 →

[Bekey 1960] [Bekey 1960] George A. Bekey, “Analog Simulation of Nerve Excitation”, in [Jackson 1960, pp. 436–444] →

[Bekey et al. 1960] [Bekey et al. 1960] G. A. Bekey, L. W. Neustadt, “Analog computer techniques for plotting Bode and Nyquist diagrams”, in *Western Joint IRE-AIEE-ACM Computer Conference*, May 3–5, 1960, pp. 165-172 →

[Bekey et al. 1966] [Bekey et al. 1966] George A. Bekey, M. H. Gran, A. E. Sabroff, A. Wong, “Parameter optimization by random search using hybrid computer techniques”, in *AFIPS ’66 (Fall): Proceedings of the November 7–10, 1966, fall joint computer conference*, November 1966, pp. 191–200 →

[Bekey et al. 1967] [Bekey et al. 1967] George A. Bekey, J. C. Maloney, R. Tomovic, “Solution of integral equations by hybrid computation”, in *AFIPS ’67 (Fall): Proceedings of the November 14–16, 1967, fall joint computer conference*, November 1967, pp. 143–148 →

[Bekey et al. 1968] [Bekey et al. 1968] George A. Bekey, Walter J. Karplus, *Hybrid Computation*, John Wiley & Sons, Inc., 1968 a, b, c, d, e, f, g

[Bekey et al. 1971] [Bekey et al. 1971] George A. Bekey, Man T. Ung, “On the hybrid computer solution of partial differential equations with two spatial dimensions”, in *AFIPS ’71 (Fall): Proceedings of the November 16–18, 1971, fall joint computer conference*, November 1971, pp. 401–410 →

[Belgaumkar 1967] [Belgaumkar 1967] B. M. Belgaumkar, “Computer-Aided Design and Fabrication of Cam”, in [IACM 1976, pp. 1092–1094] →

[Bell 1960] [Bell 1960] D. A. Bell, *Electric Noise – Fundamentals and Physical Mechanism*, D. van Nostrand Company, Ltd., 1960 →

[Bellini et al. 1975] [Bellini et al. 1975] Armando Bellini, Claudio Cerri, Alessandro de Carli, “A New Approach to the Simulation of Static Converter Drives”, in *Annales de l’Association internationale pour le Calcul analogique*, No. 1, Janvier 1975, pp. 3–7 →

[Bender et al. 1960] [Bender et al. 1960] R. R. Bender, D. T. Doody, P. N. Stoughton, “A Description of the IBM 7074 System”, in *IRE-AIEE-ACM ’60 (Eastern): Papers presented at the December 13-15, 1960, eastern joint IRE-AIEE-ACM computer conference*, December 1960, pp. 161-–171 →

[Bendix] [Bendix] N. N., *Bendix Computer – Digital Differential Analyzer D-12*, Bendix Computer, 5630 Arbor Vitae Street, Los Angeles 45, California, n. d. →

[Bendix 1954] [Bendix 1954] N. N., *Operation Manual – Digital Differential Analyzer – Model D12*, Bendix Computer, 5630 Arbor Vitae Stress, Los Angeles 45, California, Copy No. 2, April 1954 a, b, c, d, e, f, g, h, i, j, k, l, m

[Bendix 1957] [Bendix 1957] Bendix Computer Division, “A Three-Axis Simulator for Controlled Flight Test of Airborne Systems”, in *Computers and Automation*, 1957, Vol. 6, No. 5, p. 18 →

[Benecke et al. 1957] [Benecke et al. 1957] Th. Benecke, A. W. Quick (eds), *History of German Guided Missiles Development*, Verlag E. Appelhans & Co., 1957

[Beneking 1971] [Beneking 1971] Heinz Beneking, *Praxis des elektronischen Rauschens*, Bibliographisches Institut, 1971 →

[Benes 1965] [Benes 1965] V. E. Benes, *Mathematical Theory of Connecting Networks and Telephone Traffic*, Academic Press, New York, 1965 →

[Benham 1970] [Benham 1970] R. D. Benham (ed.), “Evaluation of Hybrid Computer Performance on a Cross Section of Scientific Problems”, in *AEC Research & Development Report*, BNWL-1278, UC-32, January 1970 →

[Benham et al. 1973] [Benham et al. 1973] R. D. Benham, G. R. Taylor, “A PDP 11 Study of the Physiological Simulation Benchmark Experiment”, in *DECUS PROCEEDINGS*, Fall 1973, pp. 83–88 →

[Bennett 1993] [Bennett 1993] S. Bennett, *A history of control engineering 1930–1955*, IEE Control Engineering Series 47, Peter Peregrinus Ltd., 1993 →

[Benyon 1961] [Benyon 1961] P. R. Benyon, “The Australian Guided Weapons Analogue Computer AGWAC”, Third International Conference on Analog Computation, Opatija, 4–9 September 1961 →

[Berger et al. 1979/1] [Berger et al. 1979/1] E. Berger, F. Berger, “Die Verwendung von digital setzbaren Funktionsgebern”, in [TU Wien 1979/1, pp. 22 ff.] →

[Berger et al. 1979/2] [Berger et al. 1979/2] E. Berger, E. Wittek, “Optimale Trassen für ein U-Bahn-Netz”, in [TU Wien 1979/2, pp. 21 ff.] →

[Berkeley et al. 1956] [Berkeley et al. 1956] Edmund Callis Berkeley, Lawrence Wainwright, *Computers – Their Operation And Applications*, Reinhold Publishing Corporation, New York, Chapman & Hall, Limited, London, 1956 a, b, c, d, e, f, g, h

[Berry et al. 1966] [Berry et al. 1966] Donald T. Berry, Dwain A. Deets, *Design, development, and utilization of a general purpose airborne simulator*, presented at the 28th Meeting of the AGARD Flight Mechanics Panel, Paris 10–11 May 1966 →

[Betz 1948] [Betz 1948] Albert Betz, *Konforme Abbildung*, Springer Verlag, 1948 →

[Beukeboom et al. 1985] [Beukeboom et al. 1985] J. J. A. J. Beukeboom, J. J. van Dixhoorn, J. W. Meerman, “Simulation of mixed bond graphs and block diagrams on personal computers using TUTSIM”, in *Journal of the Franklin Institute*, Volume 319, Issues 1–2, January–February 1985, pp. 257–267 →

[Bickel et al. 1957] [Bickel et al. 1957] Henry J. Bickel, Robert I. Bernstein, “A Realistic Radar Simulator”, in [White Sands 1957, pp. 199–210] →

[Biggs] [Biggs] A. G. Biggs, *Red Duster Acceptance Trials, Scientific Evaluation – A Mathematical Model of the Missile System Suitable for Analogue Computation*, Department of Supply, Australian Defence Scientific Service, Weapons Research Establishment, Report SAD 20, No. 8 J.S.T.U. D3, n. d. →

[Bild der Wissenschaft 1970] [Bild der Wissenschaft 1970] N. N., “Das Mathematische Kabinett”, in *Bild der Wissenschaft*, Mai 1970 →

[Bilinski 1967] [Bilinski 1967] Aleksander Bilinski, “Free Motion of Mass-Body (Ship) System Model with Application to the Synthesis of Motion Controller”, in [IACM 1976, pp. 1101–1109] →

[Billet 1965] [Billet 1965] Reinhard Billet, *Verdampfertechnik*, Bibliographisches Institut Mannheim, B.I.-Wissenschaftsverlag, 1965 →

[Bilstein 2003] [Bilstein 2003] Roger E. Bilstein, *Stages to Saturn – A Technological History of the Apollo/Saturn Launch Vehicles*, University Press of Florida, 2003 a, b, c

[Bismut et al. 1967] [Bismut et al. 1967] Marcel Bismut, Jacques Soubirou, “Optimilisation du Reglage d’une Boucle d’Asservissement Application au Controle d’Attitude d’un Satellite“, in [IACM 1976, pp. 910–915] →

[Black 1937] [Black 1937] Harold Stephen Black, *Wave Translation System*, United States Patent 2102671, Dec. 21, 1937 →

[Blum et al.] [Blum et al.] Alfons Blum, Manfred Glesner, “Macromodeling Procedures for the Computer Simulation of Power Electronics Circuits”, F.B. 12.2 Elektrotechnik, Universität des Saarlandes, n. d. →

[Blum 1982] [Blum 1982] Werner Blum, “Der Integraph im Analysisunterricht – Ein altes Gerät in neuer Verwendung”, in *ZDM Zentralblatt für Didaktik der Mathematik*, Jahrgang 14, 1982, Ernst Klett Verlag, pp. 25–30 →

[Blumberg 1967] [Blumberg 1967] Walter J. Blumberg, “The Use of Analog/Hybrid Computers in Real-Time Simulation and Study of Surface-Ship-Control Systems”, in [IACM 1976, pp. 774–783] →

[Boghosian et al. 1950] [Boghosian et al. 1950] W. H. Boghosian, S. Darlington, H. G. Och, *Artillery Director*, United States Patent 2493183, January 3, 1950 →

[Bohling et al. 1970] [Bohling et al. 1970] Dorothea M. Bohling, Lawrence A. O'Neill, "An Interactive Computer Approach to Tolerance Analysis", in *IEEE Transactions on Computers*, Vol. C-19, No. 1, January 1970, pp. 10–16 →

[Bohn 1960] [Bohn 1960] E. V. Bohn, "A Pulse Position Modulation Analog Computer", in *IRE Transactions on Electronic Computers*, June 1960, pp. 256–261 →

[Bohren 1967] [Bohren 1967] Eduard Bohren, "Zeitoptimale Drehzahlregelung eines Stromrichtermotors mittels zweier Stellgrößen", in [IACM 1976, pp. 1095–1100] →

[Bona 1967] [Bona 1967] Cesare Bona, "A Mathematical Model for the Simulation of a Heat-Exchanger with a Natural Circulation Drum Boiler", in [IACM 1976, pp. 988–1005] →

[Borchardt] [Borchardt] Inge Borchardt, *Demonstrationsbeispiel: Elektrisch geladenes Teilchen im Magnetfeld*, AEG-Telefunken, ADB 007, n. d. a, b, c, d

[Borchardt et al.] [Borchardt et al.] Inge Borchardt, P. Maier, F. Hultschig, *Simulation von Strahlführungssystemen auf dem hybriden Rechnersystem HRS 860*, AEG Telefunken, Datenverarbeitung, n. d. →

[Borchardt 1965] [Borchardt 1965] Inge Borchardt, *Berechnung von Teilchenbahnen in einem magnetischen Horn am Analogrechner*, DESY – H 11, Hamburg, 17.12.1965 →

[Borchardt et al. 1965] [Borchardt et al. 1965] Inge Borchardt, G. Ripken, *Zur Berechnung der Teilchenbahnen in einem Sextupolfeld*, Hamburg, DESY – H 5, Hamburg, 24.05.1965 →

[Borchardt 1966] [Borchardt 1966] Inge Borchardt, *Strahloptische Gleichungen und ihre Verwendung im Analogrechenprogramm*, DESY-Strahloptik, 01.04.1966 →

[Borchardt et al. 1969] [Borchardt et al. 1969] Inge Borchardt, P. Zajíček, *Wechselrichter-Schutzprobleme*, Interner Bericht, DESY K-69/3, November 1969 →

[Borchardt et al. 1976] [Borchardt et al. 1976] Inge Borchardt, M. Levy, *Endstufen-Simulation der Quadrupol-Stromversorgung für langsame Ejektion*, Interner Bericht, DESY K-76/01, März 1976 →

[Borchardt et al. 1977] [Borchardt et al. 1977] Inge Borchardt, M. Levy, J. Maass, *Untersuchungen über die Regelung des 200 Hz Wechselrichters für das flat-top-System des Synchrotrons mit der hybriden Rechenanlage HRS 860*, Interner Bericht, DESY R1-77/01, June 1977 →

[Bordes et al. 1967] [Bordes et al. 1967] H. J. Bordes, K. te Velde, "Die Berechnung von Heiz- und Kühllast mit digitalen und analogen Rechenmaschinen", in *Heizung – Lüftung – Haustechnik*, 18, 1967, Nr. 8, pp. 300–307 →

[Borsei et al. 1967] [Borsei et al. 1967] A. Borsei, G. Estrin, "An Analog Computer Study of the Dynamic Behaviour of Stressed Thin Ferromagnetic Films", in *Mathematics and Computers in Simulation*, Volume 9, Issue 2, April 1967, pp. 68–75 →

[Bosley et al. 1956] [Bosley et al. 1956] Donald B. Bosley, Roth S. Leddick, *Simulation of Steam Pressurizing Tank Transients by Analog Computer*, United States Naval Postgraduate School, Monterey, California, 1956 →

[Bournez et al. 2007] [Bournez et al. 2007] Olivier Bournez, Manuel L. Campagnolo, Daniel S. Graça, Emmanuel Hainry, "Polynomial differential equations compute all real computable functions on computable compact intervals", in *Journal of Complexity*, Volume 23, Issue 3, June 2007, pp. 317–335 →

[Bournez et al. 2012] [Bournez et al. 2012] Olivier Bournez, Manuel Campagnolo, "A Survey on Continuous Time Computations", in

[Cooper et al. 2008, pp. 383–423] →

[Bournez 2015] [Bournez 2015] Olivier Bournez, *Course Notes for MPRI Course 2.33.1: The General Purpose Analog Computer*, →https://wikimpri.dptinfo.ens-paris-saclay.fr/lib/exe/fetch.php?media=cours:upload:gpac.pdf, retrieved 07.07.2025 a, b

[Bournez et al. 2017/1] [Bournez et al. 2017/1] Olivier Bournez, Daniel Silva Graça, Amaury Pouly, "On the functions generated by the general purpose analog computer", in *Information and Computation*, Volume 257, December 2017, pp. 34–57 →

[Bournez et al. 2017/2] [Bournez et al. 2017/2] Olivier Bournez, Daniel Silva Graça, Amaury Pouly, "Polynomial Time Corresponds to Solutions of Polynomial Ordinary Differential Equations of Polynomial Length", in *Journal of the ACM*, Vol. 64, Issue 6, Article No. 38, pp. 1–76 →

[Bourton et al. 1967] [Bourton et al. 1967] M. Bourton, P. Vidal, "Simulation des Divers Modes de Transmission de l'Information en Biologie", in [IACM 1976, pp. 1132–1137] →

[Bradley et al. 1962] [Bradley et al. 1962] R. E. Bradley, J. F. Genna, "Design of a one-megacycle iteration rate DDA", in *AIEE-IRE '62 (Spring): Proceedings of the May 1–3, 1962, spring joint computer conference*, pp. 353–364 →

[Breitenecker 1978/1] [Breitenecker 1978/1] Felix Breitenecker, "Rendezvous im Weltall", in [TU Wien 1978/1, pp. 45 ff.] a, b

[Breitenecker 1978/2] [Breitenecker 1978/2] Felix Breitenecker, "Bildhafte Darstellung eines Laufkrans am Oszillographen", in [TU Wien 1978/2, pp. 33 ff.] →

[Breitenecker 1979] [Breitenecker 1979] Felix Breitenecker, "Squash – Einmanntennis am Analogrechner", in [TU Wien 1979/1, pp. 47 f.] →

[Brennan et al. 1964] [Brennan et al. 1964] R. D. Brennan, H. Sano, “PACTOLUS – A digital simulator program for the IBM 1620”, in *AFIPS Conference Proceedings, Fall Joint Computer Conference 26*, October 1964, pp. 299–312 →

[Brennan et al. 1967] [Brennan et al. 1967] R. D. Brennan, M. Y. Silberberg, “Two continuous system modeling programs”, in *IBM SYSTEMS JOURNAL*, Vol. 6, No. 4, 1967, S. 242–266 a, b, c, d

[Brey 1958] [Brey 1958] R. N. Brey Jr., “Reactor Control”, in *Instruments and Automation*, Vol. 31, April 1958, pp. 630–635 →

[Brissenden et al. 1961] [Brissenden et al. 1961] Roy F. Brissenden, Bert B. Burton, Edwin C. Foudriat, James B. Whitten, *Analog Simulation of a Pilot-Controlled Rendezvous*, Technical Note D-747, National Aeronautics and Space Administration Washington, April 1961 →

[Britain 2011] [Britain 2011] James E. Britain, “Electrical Engineering Hall of Fame: Harold S. Black”, in *Proceedings of the IEEE*, Vol. 99, No. 2, February 2011, pp. 351–353 →

[Brock 2017] [Brock 2017] David C. Brock, “The Shocking Truth Behind Arnold Nordsieck’s Differential Analyzer”, *IEEE Spectrum*, →https://spectrum.ieee.org/the-shocking-truth-behind-arnold-nordsiecks-differential-analyzer, retrieved 16.08.2025 →

[Bromley 1984] [Bromley 1984] Allan G. Bromley, *British Mechanical Gunnery Computers of World War II*, Technical Report 223, January 1984 a, b

[Brouwer 2007] [Brouwer 2007] Jens Brouwer, *Das FitzHugh-Nagumo Modell einer Nervenzelle*, Universität Hamburg, Department Mathematik, 20.8.2007 →

[Brown 1969] [Brown 1969] Frank M. Brown, “Comment on Canonical Programming of Nonlinear and Time-Varying Differential-Equations”, in *IEEE Transactions on Computers*, Vol. C-18, No. 6, June 1969, p. 566 →

[Brubaker et al. 1961] [Brubaker et al. 1961] Thomas A. Brubaker, Harry R. Eckes, “A digital control unit for a repetitive analog computer”, in *Wester Joint IRE-AIEE-ACM Computer Conference*, May 9–11, 1961, pp. 353-360 →

[Bryant et al. 1962] [Bryant et al. 1962] Lawrence T. Bryant, Marion J. Janicke, Louis C. Just, Alan L. Winiecki, *Engineering Applications of Analog Computers*, Argonne National Laboratory, ANL-6319 Rev., Revised October 1962 a, b, c, d

[Bryant et al. 1966] [Bryant et al. 1966] Lawrence T. Bryant, Lawrence W. Amiot, Ralph P. Stein, “A hybrid computer solution of the co-current flow heat exchanger Sturm-Liouville problem”, in *AFIPS ’66 (Fall): Proceedings of the November 7–10, 1966, fall joint computer conference*, November 1966, pp. 759–769 →

[Bückner] [Bückner] Hans Bückner, *Über Nachlaufsteuerungen*, n. d., author’s archive →

[Bückner 1950] [Bückner 1950] Hans Bückner, “Ein neuer Typ einer Integrieranlage zur Behandlung von Differentialgleichungen”, in *Archiv der Mathematik*, 1949/50, Volume 7, Issue 6, pp. 424–433 →

[Bückner 1953] [Bückner 1953] Hans Bückner, “Über die Entwicklung des Integromat”, in [Cremer 1953, pp. 1–16] →

[Bureau of Ordnance 1940] [Bureau of Ordnance 1940] N. N., *Basic Fire Control Mechanisms, OP1140*, September 1940 a, b

[Bureau of Ordnance 1944] [Bureau of Ordnance 1944] N. N., *Torpedo Data Computer, Mark 3, Mods. 5 to 12 inclusive*, June 1944 →

[Burns et al. 1961] [Burns et al. 1961] Arthur J. Burns, Richard E. Kopp, “Combined analog-digital simulation”, in *AFIPS ’61 (Eastern): Proceedings of the December 12–14, 1961, eastern joint computer*

conference: computers – key to total systems control, December 1961, pp. 114–123 →

[Burns 1965] [Burns 1965] Arthur J. Burns, "Hybrid Computing for Lunar Excursion Module Studies", in *Proceedings – Fall Joint Computer Conference*, 1965, pp. 893–901 →

[Bush 1912] [Bush 1912] Vannevar Bush, *Profile Tracer*, United States Patent 1048649, Dec. 31, 1912 →

[Bush et al. 1927] [Bush et al. 1927] Vannevar Bush, F. D. Gage, H. R. Stewart, "A Continuous Integraph", in *Journal of the Franklin Institute*, Vol. 203, No. 1, 1927, pp. 63–84 a, b, c

[Bush 1936] [Bush 1936] Vannevar Bush, "Instrumental Analysis", in *Bulletin of the American Mathematical Society*, 42(10), October 1936, pp. 649–669 a, b

[Bush et al. 1945] [Bush et al. 1945] Vannevar Bush, S. H. Caldwell, "A New Type of Differential Analyzer", in *Journal of The Franklin Institute – Devoted to Science and the Mechanic Arts*, Vol. 240, No. 4, October 1945, pp. 255–326 a, b, c, d, e

[Bywater 1973] [Bywater 1973] R. E. H. Bywater, *The Systems Organisation of Incremental Computers*, Thesis, Faculty of Engineering, Department of Electronic and Electrical Engineering of the University of Surrey, April 1973 →

[Cajori 1994] [Cajori 1994] Florian Cajori, *A History of the Logarithmic Slide Rule and Allied Instruments*, Astragal Press, 1994 →

[Calamore et al. 1963] [Calamore et al. 1963] D. V. Calamore, D. H. Crimmins, E. Tobey, *Army Gas-Cooled Reactor Systems Program – Analog Computer Studies of the ML-1 Power Plant*, Report No. AGN-TM-400, AEC Research and Development Report UC-32, 2nd printing, 1963 a, b

[Caldwell et al. 1955] [Caldwell et al. 1955] R. R. Caldwell, V. C. Rideout, “A Differential-Analyzer Study of Certain Nonlinearly Damped Servomechanisms”, in [Paynter ed. 1955, pp. 193–198] →

[Caltech 1949] [Caltech 1949] N. N., “The Electric Analog Computer”, in *Research in Progress*, April 1949, pp. 17–18 →

[Cameron et al. 1961] [Cameron et al. 1961] W. D. Cameron, R. E. Tiller, *Analog Program in Reactor Speed of Control Systems*, Technical Report, HW-69940, 1961 Jun. 14 →

[Camp et al. 1934] [Camp et al. 1934] Thomas R. Camp, H. L. Hazen, *Hydraulic Analysis of Water Distribution Systems by Means of an Electric Network Analyzer*, New England Water Works Association, V. 48, No. 4, December, 1934 →

[Campbell-Kelly et al. 2007] [Campbell-Kelly et al. 2007] M. Campbell-Kelly, Mary Goretti Croarken, R. Flood, E. Robson (eds.), *The History of Mathematical Tables – from Sumer to Spreadsheets*, Oxford University Press, 2007

[Campeau 1969] [Campeau 1969] Joseph O. Campeau, “The Block-Oriented Computer”, in *IEEE Transactions on Computers*, Vol. C-18, No. 8, August 1969, pp. 706–718 →

[Care 2006/1] [Care 2006/1] Charles Philip Care, “A Chronology of Analogue Computing”, in *The Rutherford Journal*, Volume 2, 2006–2007 →

[Care 2006/2] [Care 2006/2] Charles Philip Care, *Modelling Oil Reservoirs: Analog Computing at British Petroleum*, Computer Science Research Report 424, Department of Computer Science, University of Warwick, 2006 →

[Care 2008] [Care 2008] Charles Philip Care, *From analogy-making to modelling: the history of analog computing as a modelling*

technology, PhD. thesis, University of Warwick, Department of Computer Science, September 2008 →

[Carlson et al. 1967] [Carlson et al. 1967] Alan M. Carlson, George Hannauer, Thomas Carey, Peter J. Holsberg (eds.), *Handbook of Analog Computation*, 2nd edition, Electronic Associates, Inc., Princeton, New Jersey, 1967 a, b, c, d, e, f, g, h

[Carlson 1968] [Carlson 1968] Alan M. Carlson, "Hybrid Simulation of an Exchanger/Reactor Control System", in *EAI applications reference library*, 6.4.20h, October 1968 →

[Carson 1926] [Carson 1926] John R. Carson, "Wave propagation in overhead wires with ground return", in *The Bell System Technical Journal*, Volume 5, Issue 4, Oct. 1926, pp. 539–554 →

[Cass 2023] [Cass 2023] Stephen Cass, "Try This Brand New Analog Computer – The THAT machine is designed to encourage you to go beyond digital", in *IEEE Spectrum*, →https://spectrum.ieee.org/try-this-new-analog-computer, retrieved 16.08.2025 →

[Cavana et al. 2021] [Cavana et al. 2021] Robert Y. Cavana, Brian C. Dangerfield, Oleg V. Pavlov, Michael J. Radzicki, I. David Wheat (eds.), *Feedback Economics – Economic Modelling with System Dynamics*, Springer, 2021

[Celinski et al. 1964] [Celinski et al. 1964] O. Celinski, I. H. Rimawi, "Classification of Analog Multipliers", in *Instruments & Control Systems*, June 1964, pp. 149–156 a, b

[Celmer et al. 1970] [Celmer et al. 1970] John Celmer, Mary Rouland, *Automatic Analog Computer Scaling Using Digital Optimization Techniques*, National Aeronautics and Space Administration, Washington, D. C., March 1970 →

[Ceruzzi 1989] [Ceruzzi 1989] Paul E. Ceruzzi, *Beyond the Limits – Flight Enters the Computer Age*, The MIT Press, 1989 a, b, c, d

[Chamboredon et al. 1967] [Chamboredon et al. 1967] C. Chamboredon, M. Hery, B. Marion, “Etude de la Regulation d’une Centrale Nucleaire a Reacteur Rapide sur Calculateur Hybride”, in [IACM 1976, pp. 1065–1070] →

[Chan 1969] [Chan 1969] Shu-Kwan Chan, “The Serial Solution of the Diffusion Equation Using Nonstandard Hybrid Techniques”, in *IEEE Transactions on Computers*, Vol. C-18, No. 9, September 1969, pp. 786–799 →

[Chance et al. 1947] [Chance et al. 1947] Britton Chance, J. N. Thurston, P. L. Richman, “Some Designs and Applications for Packaged Amplifiers Using Subminiature Tubes”, in *The Review of Scientific Instruments*, Volume 18, Number 9, September, 1947, pp. 610–616 →

[Chance et al. 1949] [Chance et al. 1949] Britton Chance, Vernon Hughes, Edward F. MacNichol, David Sayre, Frederic C. Williams (ed.), *Waveforms*, McGraw-Hill Book Company, Inc., 1949 →

[Chandler et al. 1961] [Chandler et al. 1961] W. G. Chandler, P. L. Dandeno, A. F. Glimn, “Short-range economic operation of a combined thermal and hydroelectric power system”, in *AIEE Transactions*, Part 3, Vol. 80, May 1961, pp. 1219–1228 →

[Channamadhavuni et al. 2021] [Channamadhavuni et al. 2021] Shravya Channamadhavuni, Sven Thijssen, Sumit Kumar Jha, Rickard Ewetz, “Accelerating AI Applications using Analog In-Memory Coputing: Challenges and Opportunities”, in *GLSVLSI ’21: Proceedings of the 2021 on Great Lakes Symposium on VLSI*, June 2021, pp. 379–384 →

[Chareton et al. 1967] [Chareton et al. 1967] H. Chareton, M. Chaigne, “Simulateur D’Echangeur de Chaleur”, in [IACM 1976, pp. 812–820] →

[Charlesworth et al. 1974] [Charlesworth et al. 1974] A. S. Charlesworth, J. R. Fletcher, *Systematic Analogue Computer Programming*, Pitman Publishing, Second Edition, 1974 →

[Cheng] [Cheng] Shang-I Cheng, "Analog Simulation and Polymerization Kinetics", n. d., author's archive →

[Chrysler Missile Division] [Chrysler Missile Division] Chrysler Corporation Missile Division, *This is Redstone*, n. d. →

[Chuang et al. 1960] [Chuang et al. 1960] Kuei Chuang, Louis F. Kazda, Thomas Windeknecht, *Report of Project MICHIGAN: A Stochastic Method of Solving Partial Differential Equations using an Electronic Analog Computer*, University of Michigan, June 1960 →

[Ciardiello 2009] [Ciardiello 2009] Emilio Ciardiello, *Analog Computers: RCA Typhoon, 1951*, →https://www.radiomuseum.org/forum/analog_computers_rca_typhoon_1951.html, retrieved 18.02.2025 →

[Claes 1975] [Claes 1975] W. Claes, "Überblick über die Erfahrung des Brüsseler Rechenzentrums in biomedizinischen Problemen", in [TU Wien 1975/2, pp. 9 f.] →

[Clark et al. 1958] [Clark et al. 1958] Dorothy K. Clark, Archie N. Colby, Paul Iribe, Nicholas M. Smith, Lloyd D. Yates, *HUTSPIEL – a Theater War Game*, Operations Research Office, The Johns Hopkins University, 1958 a, b, c

[Clos 1953] [Clos 1953] Charles Clos, "A Study of Non-Blocking Switching Networks", in *Bell System Technical Journal*, March 1953, Vol. 32, pp. 406–423 →

[Clymer 1993] [Clymer 1993] A. Ben Clymer, "The Mechanical Analog Computers of Hannibal Ford and William Newell", in *IEEE Annals of the History of Computing*, Vol. 15, No. 2, 1993, pp. 19–34 a, b, c, d

[Clynes 1960] [Clynes 1960] Manfred Clynes, “Respiratory control of heart rate: laws derived from analog computer simulation”, in *IRE Transaction on Medical Electronics*, Jan. 1960, pp. 2–14 →

[Coffin 1882] [Coffin 1882] J. Coffin, *Averageometer or Instrument for Measuring the Average Breadth of Irregular Planes*, United States Patent 258993, June 6, 1882 →

[Cohen 1971] [Cohen 1971] E. M. Cohen, *Hybrid Simulation-Aided Design of a Pneumatic Relay*, Presented at the Regional Meeting, Eastern Simulation Councils Foxboro, Mass., September 23, 1971 →

[Colebrook et al. 1937] [Colebrook et al. 1937] C. F. Colebrook, C. M. White, “Experiments with fluid friction in roughened pipes”, in *Proceedings of the Royal Society A*, 161, 03. August 1937, pp. 367–381 →

[Collins et al. 2016] [Collins et al. 2016] Michelle Collins, Jennifer Hasler, Suma George, “An Open-Source Tool Set Enabling Analog-Digital-Software Co-Design”, in *Journal of Low Power Electronics and Applications*, 2016, 6, 3, doi: 10.3390/jlpea6010003 →

[Comcor] [Comcor] Comcor, *Analog/Hybrid – Ci-5000*, n. d. →

[Computer Systems Inc. 1960] [Computer Systems Inc. 1960] Computer Systems, Inc., “How to Solve Multi-Component Distillation Problems in 60 Seconds”, in *Instruments & Control Systems*, Vol. 33, September 1960, p. 1016 →

[Computers and Automation 1960] [Computers and Automation 1960] N. N., “1960 Pictorial Report on the Computer Field”, in *Computers and Automation*, 1960, Vol. 9, No. 12, pp. 13 ff. a, b, c

[Computers and Automation 1964/1] [Computers and Automation 1964/1] N. N., “Use of Analog Computers Promises

Advances in Eye Research", in *Computers and Automation*, 1964, Vol. 13, No. 10, p. 35 →

[Computers and Automation 1964/2] [Computers and Automation 1964/2] N. N., "Digital-Directed, Analog Computer Control System", in *Computers and Automation*, 1964, Vol. 13, No. 12, p. 35 →

[Computers and Automation 1966] [Computers and Automation 1966] N. N., "The Annual Computer Art Contest of 'Computers and Automation'", in *Computers and Automation*, 1966, Vol. 15, No. 8, pp. 8–13 →

[Computers and Automation 1967] [Computers and Automation 1967] N. N., "Computer-Controlled Anesthesiology Patient Simulator", in *Computers and Automation*, 1967, Vol. 16, No. 5, pp. 46–47 →

[Computers and Automation 1975] [Computers and Automation 1975] N. N., "Thirteens Annual Computer Art Exposition", in *Computers and Automation*, 1975, Vol. 24, No. 8, pp. 8–26 a, b

[Connelly 1963] [Connelly 1963] Mark E. Connelly, "Real-Time Analog-Digital Computation", in *SIMULATION*, Volume 1, Issue 1, pp. R29–R39 →

[Cooper 1961] [Cooper 1961] N. R. Cooper, "X-15 analog flight simulation program: systems development and pilot training", in *IRE-AIEE-ACM '61 (Western): Papers presented at the May 9–11, 1961, western joint IRE-AIEE-ACM computer conference*, May 1961, pp. 623–638 →

[Cooper et al. 2008] [Cooper et al. 2008] S. Barry Cooper, Benedikt Löwe, Andreas Sorbi (eds.), *New Computational Paradigms – Changing Conceptions of What is Computable*, Sprinter Nature, 2008

[Costakis 1974] [Costakis 1974] William G. Costakis, *Analog Computer Implementation of Four Instantaneous Distortion Indices*, National Aeronautics and Space Administration, Washington, D. C., March 1974 →

[Coulter] [Coulter] Doug Coulter, *Evolution of a Model Kit of Tools – a Brief History of a Computor-Builder in Terms of Ideas & Instruments*, → http://www.philbrickarchive.org/dc032_philbrick_history.htm, n. d., retrieved 10.04.2022 →

[Coveney 2024] [Coveney 2024] Peter V. Coveney, "Sharkovskii's theorem and the limits of digital computers for the simulation of chaotic dynamical systems", in *Journal of Computational Science*, 82, 2024, 102449 a, b

[Coveney et al. 2025] [Coveney et al. 2025] Peter V. Coveney, ShunZhou Wan, *Molecular Dynamics: Probability and Uncertainty*, Oxford Graduate Texts, 2025 →

[Cowan 2005] [Cowan 2005] Glenn Edward Russell Cowan, *A VLSI Analog Computer / Math Co-processor for a Digital Computer*, Columbia University, 2005 a, b, c, d, e

[Cowan et al. 2006] [Cowan et al. 2006] Glenn Edward Russell Cowan, Robert C. Melville, Yannis P. Tsividis, "A VLSI Analog Computer/Digital Computer Accelerator", in *IEEE Journal of Solid-State Circuits*, Vol. 41, No. 1, January 2006, pp. 42–53 →

[Cowan et al. 2009] [Cowan et al. 2009] Glenn Edward Russell Cowan, Y. Tsividis, *Analog and Digital Continuous-Time Computation and Signal Processing*, CMOSET 2009 →

[Craig 1953] [Craig 1953] Leonard J. Craig, "The Magnetic Amplifier as an Analog Computer Component", in *Proceedings of the I.R.E.*, October 1953, pp.1477–1482 →

[Cremer 1953] [Cremer 1953] Hubert Cremer ed., *Probleme der Entwicklung programmgesteuerter Rechengeräte und Integrieranlagen*, Rhein.-Westf. Technische Hochschule Aachen, Mathematisches Institut, Lehrstuhl C, 1953

[Croarken 1985] [Croarken 1985] Mary Goretti Croarken, *The Centralization of Scientific Computation in Britain 1925–1955*, dissertation, Department of Computer Science, University of Warwick, Coventry, England, December 1985 →

[Crommelynckx et al. 1967] [Crommelynckx et al. 1967] D. Crommelynckx, J. Devooght, G. Sylin, "Simulateur de Reactivite de Reacteur Nucleaire", in [IACM 1976, pp. 955–965] →

[Crossley 1963] [Crossley 1963] Frank Erskine Crossley, "Die Nachbildung eines mechanischen Kurbelgetriebes mittels eines elektronischen Analogrechners", in *Feinwerktechnik*, Jahrgang 1967, Juni 1963, Heft 6 a, b

[Crossley 1965] [Crossley 1965] Frank Erskine Crossley, "Geometric computing: Analogue-simulation of a linkage", in *International Journal of Mechanical Sciences*, Volume 7, Issue 9, September 1965, pp. 595–601 →

[Cutler 1965] [Cutler 1965] A. E. Cutler, "A new colour-matching computer", in *Journal of the Society of Dyers and Colourists*, 81, 1965, pp. 601–608 →

[Czajkowski 1956/1] [Czajkowski 1956/1] Z. Czajkowski, "Electronic Methods of Analogue Multiplication (Part 1)", in *Electronic Engineering*, July 1956, pp. 283–287 →

[Czajkowski 1956/2] [Czajkowski 1956/2] Z. Czajkowski, "Electronic Methods of Analogue Multiplication (Part 2)", in *Electronic Engineering*, August 1956, pp. 352–355 a, b

[Dagbjartsson et al. 1976] [Dagbjartsson et al. 1976] S. Dagbjartsson, D. Emendörfer, "Simulation in der Kernenergietechnik", in [Schöne

1976/1, pp. 298–340] →

[Damjanovic et al. 1967] [Damjanovic et al. 1967] Z. Damjanovic, M. Maric, N. Parezanovic, P. Pejovic, “Investigation of Nets Composed of dad-Neuromimes”, in [IACM 1976, pp. 1138–1144] →

[Darlington 1965] [Darlington 1965] R. F. Darlington, *Automatic Control of Multistand Cold Rolling Mill*, doctoral thesis, Mechanical Engineering Department, Imperial College of Science and Technology, London, 1965 →

[Dassow 2015] [Dassow 2015] Achim Dassow, “Geschichte der nichtlinearen Röhren Raytheon QK-256 und QK-329”, →https://www.radiomuseum.org/tubes/tube_qk329.html, retrieved 22.05.2022 →

[Datamation 1957] [Datamation 1957] Datamation, *The Magazine of DATAmation, Research & Engineering*, October 1957 →

[Davidson 1953] [Davidson 1953] John David Davidson, *Application of magnetic amplifiers to electronic computers*, master’s thesis, United States Naval Postgraduate School, 1953 →

[Davis et al. 1955] [Davis et al. 1955] B. E. Davis, I. H. Swift, “Analogue Computers Using Magnetic Amplifiers”, in *Electrical Engineering*, Vol. 74, Issue 2, February 1955, p. 158 →

[Davis et al. 1974] [Davis et al. 1974] Frank T. Davis, Armando B. Corripio, “Dynamic Simulation of Variable Speed Centrifugal Compressors”, ISA CPD 74105, 1974 →

[Day 1959] [Day 1959] Richard E. Day, *Training Considerations During the X-15 Development*, paper presented to the Training Advisory Committee of the National Security Industrial Association, Los Angeles, California, November 17, 1959 →

[Deken 1984] [Deken 1984] Joseph Deken, *Computerbilder, Kreativität und Technik*, Birkhäuser Verlag, 1984 →

[Dehmel 1949] [Dehmel 1949] Richard C. Dehmel, *Aircraft Trainer for Aerial Gunners*, United States Patent 2471315, May 24, 1949 →

[Dehmel 1954] [Dehmel 1954] Richard C. Dehmel, *Flight Training Apparatus for Computing Flight Conditions and Simulating Reaction of Forces on Pilots*, United States Patent 2687580, Aug. 31, 1954 →

[Delarousse et al. 1962] [Delarousse et al. 1962] P. Delarousse, C. Trouve, R Jacques, "Etude Sur Simulateur Des Regimes Transitoires Des Concentrations Dans Une Installation De Diffusion Gazeuse", in *Service des Etudes sur la Séparation des Isotopes de l'Uranium*, Rapport C. E. A. no. 2010 →

[Demler 2018] [Demler 2018] Mike Demler, "Mythic Multiplies in a Flash – Analog In-Memory Computing Eliminates DRAM Read/Write Cycles", in *Microprocessor Report*, August 2018, pp. 1–3 →

[Demoyer 1980] [Demoyer 1980] R. DeMoyer Jr., "Interactive Anti-Aircraft Gun Fire Control Simulation: An Introduction to Hybrid Computation", in *EAI Product Information Bulletin*, February 18, 1980, Bulletin No. 023 →

[Dennis 1958] [Dennis 1958] Jack Bonnell Dennis, *Mathematical Programming and Electrical Networks*, dissertation, Massachusetts Institute of Technology, August, 1958 →

[Department of the Army 1956] [Department of the Army 1956] Department of the Army, *NIKE I SYSTEMS, NIKE I COMPUTER (U)*, Department of the Army Technical Manual TM 9-5000-3, April 1956 →

[Dern et al. 1957] [Dern et al. 1957] Herbert Dern, John O'Connor, Reinhold P. Vogel, "Low Frequency Cumulative Distribution Analyzer", in [White Sands 1957, pp. 171–197] →

[Dewdney 1988/1] [Dewdney 1988/1] A. K. Dewdney, "Rechnen mit Spaghetti – Wie der Spaghetti-Computer und andere kuriose

Analoggeräte Probleme im Handumdrehen lösen, an welchen selbst die größten Digitalrechner scheitern.”, in *Computer-Kurzweil*, Spektrum der Wissenschaft, 1988, pp. 198–203 →

[Dewdney 1988/2] [Dewdney 1988/2] A. K. Dewdney, “Kuriose Analog-Computer – Eine neue Kollektion von Analogrechnern für Heimwerker und eine vertiefte Diskussion ihrer Stärken und Schwächen im Vergleich zu Digitalrechnern.”, in *Computer-Kurzweil*, Spektrum der Wissenschaft, 1988, pp. 204–210 →

[Deziel 1966] [Deziel 1966] D. P. Deziel, *Applications of Analogue Computers in Operational Research*, Thesis, Imperial College, June 1966 →

[Dhen 1959] [Dhen 1959] Walter Dhen, *Entwurf und Aufbau eines repetierenden Analogrechners, unter besonderer Berücksichtigung der Zusammenhänge zwischen den Rechenfehlern und den Regelkreiseigenschaften in elektronischen Rechengeräten*, Fakultät für Mathematik und Physik der Technischen Hochschule Darmstadt, 1959 →

[Dhen 1960] [Dhen 1960] Walter Dhen, “Entwurf und Aufbau eines repetierenden Analogrechners unter besonderer Berücksichtigung der Zusammenhänge zwischen den Rechenfehlern und den Regelkreiseigenschaften in elektronischen Rechengeräten”, in *Nachrichtentechnische Fachberichte*, Band 17, Vieweg & Sohn, Braunschweig, 1960 →

[Dick et al. 1967] [Dick et al. 1967] D. E. Dick, H. J. Wertz, “Analog and Digital Computation of Fourier Series and Integrals”, in *IEEE Transactions on Electronic Computers*, Vol. EC-16, No. 1, February 1967, pp. 8–13 →

[Dixon 1952] [Dixon 1952] William Joseph Dixon, *An Analysis of Errors in the Electric-Analog Computer*, Ph.D. thesis, California Institute of Technology, Pasadena, California, 1952 a, b, c, d

[Djordjevic 1967] [Djordjevic 1967] Ivan Djordjevik, “The Non-Linear Mathematical Model and Analog Simulation of Hydraulic Turbine Systems”, in [IACM 1976, pp. 796–800] →

[Dodd 1969] [Dodd 1969] K. N. Dodd, *Analogue Computers*, The English Universities Press Ltd., 1969 a, b

[Donan 1950] [Donan 1950] J. Donan, *MADDIDA (Preliminary Report)*, Northrop Aircraft, Inc., Project MX-775, Report No. GM-545, 26 May 1950 →

[Donner] [Donner] Donner, *Donner’s 3500 Analog Computer*, n. d. →

[Doore et al. 2014] [Doore et al. 2014] Karen Doore, Paul Fishwick, “Prototyping an analog computing represetation of predator prey dynamics”, in *WSC ’14: Proceedings of the 2014 Winter Simulation Conference*, December 2014, pp. 3561–3571a →

[Dornier/1] [Dornier/1] N. N., *Beschleunigung eines PKW mit automatischem Getriebe (vereinfacht)*, Dornier, n. d. a, b, c

[Dornier/2] [Dornier/2] N. N., *Simulation reaktionskinetischer Probleme auf dem Analogrechner*, Dornier, n. d. a, b, c, d

[Dornier/3] [Dornier/3] N. N., *Der Analogrechner als Hilfsmittel bei der Untersuchung des Schwingungsverhaltens von Mehrmassensystemen*, Dornier, n. d. →

[Dornier/4] [Dornier/4] N. N., *Feder-Masse-System mit trockener Reibung*, Dornier, n. d. →

[Dornier/5] [Dornier/5] N. N., *Kurbeltrieb*, Dornier, n. d. →

[Dornier/6] [Dornier/6] N. N., *Ermittlung der Scherarbeit bei Materialprüfungen mit Hilfe eines Analogrechners*, Dornier, n. d. →

[Dornier/7] [Dornier/7] N. N., *Rundlaufprüfungen von Rädern mit Hilfe eines Analogrechners*, Dornier, n. d. →

[Dornier/8] [Dornier/8] N. N., *Anwendung von Analogrechnern bei der Ausbildung im Rahmen der Oberschulen – Kurvendiskussion*, Dornier, n. d. →

[Dornier/9] [Dornier/9] N. N., *Anwendung von Analogrechnern bei der Ausbildung im Rahmen der Oberschulen – Approximation von trigonometrischen Funktionen durch Reihenentwicklungen*, Dornier, n. d. →

[Dornier/10] [Dornier/10] N. N., *Anwendung von Analogrechnern bei der Ausbildung im Rahmen der Oberschulen – Parameterdarstellung von geschlossenen Kurven*, Dornier, n. d. →

[Drake et al. 1963] [Drake et al. 1963] D. W. Drake, H. W. Foster, "Airplane landing gear performance solutions with an electronic analog computer", in *Proceedings of the Western Computer Conference*, February 4–6, 1963, pp. 86–97 a, b

[Duffin 1981] [Duffin 1981] R. J. Duffin, "Rubel's universal differential equation", in *Proceedings of the National Academy of Sciences*, Vol. 78, No. 8, August 1981, pp. 4661–4662 →

[Dungan 2005] [Dungan 2005] Tracy Dwayne Dungan, *V-2 – A Combat History of the First Ballistic Missile*, Westholme Publishing, 2005 →

[EAI 1961/1] [EAI 1961/1] N. N., "$1.5 Million Computer for the National Aeronautics and Space Administration", in *Computers and Automation*, 1961, Vol. 10, No. 5, p. 1B →

[EAI 1961/2] [EAI 1961/2] Electronics Associates, Inc., "Analog Computers Solving Water Shortage Problems by Locating Strategic Points to Inject Water into Natural Underground Basins", in *Computers and Automation*, 1961, Vol. 10, No. 8, p. 3B →

[EAI 1964/1] [EAI 1964/1] N. N., *231R-V Analog Computer Information Manual*, Electronic Associated, Inc., 1964 a, b, c

[EAI 1964/2] [EAI 1964/2] N. N., *Seminar on Hybrid Computation as applied to the Aerospace Field*, Electronic Associates, Inc., Long Branch, New Jersey, 1964 a, b

[EAI 1964/3] [EAI 1964/3] N. N., *Continuous Data Analysis with Analog Computers Using Statistical and Regression Techniques*, EAI Applications Reference Library 1.3.2a, 1964, →http://bitsavers.org/pdf/eai/applicationsLibrary/1.3.2a_Continuous_Data_Analysis_with_Analog_Computers_Using_Statistical_and_Regression_Techniques_1964.pdf, retrieved 20.06.2022 →

[EAI 1964/4] [EAI 1964/4] N. N., *Solution of Mathieu's Equation on the Analog Computer*, EAI Applications Reference Library, 7.7.4a, 1964 →

[EAI 1965] [EAI 1965] N. N., *Investigation of Heat Transfer by Conduction*, EAI Applications Reference Library, 7.3.8a, 1965 →

[EAI 1966] [EAI 1966] N. N., *Primer on Analog Computations and Examples for EAI-180 Series of Computers*, Electronic Associates, Inc., Bulletin No. 957051, 1966 →

[EAI 1986] [EAI 1986] N. N., *EAI User's Group Newsletter*, 14th edition, April 1986 →

[EAI 690] [EAI 690] N. N., *690 Hybrid Computing System – Reference Handbook*, Electronic Associates, Inc., Long Branch, New Jersey, n. d. →

[EAI PACE 231R] [EAI PACE 231R] EAI, *PACE 231R analog computer*, Electronic Associates, Inc., Long Branch, New Jersey, Bulletin No. AC 6007, n. d. →

[EAI TR-10] [EAI TR-10] N. N., *PACE TR-10 Transistorized Analog Computer – operator's handbook*, Electronic Associates, Inc., Long Branch, New Jersey, n. d. →

[Eck 1954] [Eck 1954] Bruno Eck, *Technische Strömungslehre*, Springer-Verlag, Berlin/Göttingen/Heidelberg, 1954, 4. verbesserte Auflage →

[Eckdahl et al. 2003] [Eckdahl et al. 2003] Donald E. Eckdahl, Irving S. Reed, Hrant Harold Sarkissian, "West Coast Contributions to the Development of the General-Purpose Computer: Building Maddida and the Founding of Computer Research Corporation", in *IEEE Annals*, January-March 2003, Vol. 25, No. 1, pp. 4–33 →

[Eckert 1997] [Eckert 1997] Michael Eckert, "„Planck vorsichtig und abstrakt, ich etwas draufgängerisch" – Planck und Sommerfeld oder: Wie „wissenschaftliche Revolutionäre" denken", in *Kultur & Technik – Zeitschrift des Deutschen Museums*, C. H. Beck, München, 4/1997, pp. 38–45 →

[Eckert 2013] [Eckert 2013] Michael Eckert, *Arnold Sommerfeld – Science, Life and Turbulent Times 1968–1951*, Springer, 2013 →

[Edwards 1954] [Edwards 1954] C. M Edwards, "Survey of Analog Multiplication Schemes", in *Journal of the ACM*, Vol. 1, Issue 1, Jan. 1954, pp. 27–35 →

[Edwards 1955] [Edwards 1955] C. M. Edwards, "A New Approach to Grounding in DC Analog Computers", in *Wester Joint Computer Conference*, 1955, pp. 23–26 →

[Eggers 1954] [Eggers 1954] K. Eggers, *Über die Integrieranlage 'Integromat' des Instituts für angewandte Mathematik der Universität Hamburg*, Schriftenreihe Schiffbau, 3, Februar 1954 →

[Ehricke 1960] [Ehricke 1960] Krafft A. Ehricke, *Space Flight – Environment and Celestial Mechanics*, D. Van Nostrand Company, Inc., Princeton, N. J., 1960 →

[Eisenberg] [Eisenberg] W. Eisenberg, *Der junge Arnold Sommerfeld und die Integriermaschine*,

→https://www.synatech.de/Klassisch/PDF/Eisenberg_Integriermaschine.pdf, retrieved 23.02.2025 →

[Electronics 1951] [Electronics 1951] N. N., “Guided Missile Computer – The Front Cover”, in *Electronics*, April 1951, p. 138 →

[Elektronische Rechenanlagen 1960] [Elektronische Rechenanlagen 1960] N. N., “Funktionsgruppen für die Analogrechentechnik”, in *Elektronische Rechenanlagen*, 2 (1960), No. 1, pp. 43–44 →

[Elektronische Rechenanlagen 1961] [Elektronische Rechenanlagen 1961] N. N., “Messen – Datenverarbeiten – Auswerten”, advertisement in *Elektronische Rechenanlagen*, 3 (1961), No. 1, p. 44 a, b

[Elektronische Rechenanlagen 1964] [Elektronische Rechenanlagen 1964] N. N., “Electronic Associates, Inc., Europäisches Rechenzentrum für Analog- und Hybridrechentechnik”, in *Elektronische Rechenanlagen*, 6 (1964), No. 4, p. 214 →

[Elektronische Rundschau 1957] [Elektronische Rundschau 1957] N. N., “Eröffnung des ersten europäischen Analog-Rechenzentrums”, in *Elektronische Rundschau*, August 1957, Vol. 11, No. 8, p. 253 →

[Elfers 1957] [Elfers 1957] W. A. Elfers, “Simulation Analysis Technique”, in [White Sands 1957, pp. 35–46] →

[Elshoff et al. 1970] [Elshoff et al. 1970] J. L. Elshoff, P. T. Hulina, “The binary floating point digital differential analyzer”, in AFIPS ’70 (Fall) Proceedings of the November 17–19, 1970, fall joint computer conference, pp. 369–376 →

[Elton et al. 1942] [Elton et al. 1942] Charles Elton, Mary Nicholson, “The Ten-Year Cycle in Numbers of the Lynx in Canada”, in *Journal of Animal Ecology*, Vol. 11, No. 2, Nov. 1942, pp. 215-244 →

[Emblay] [Emblay] Ronald W. Embley, *The Technology behind SIMSTAR, an all-new Simulation Multiprocessor*, Electronic Associates, Inc., West Long Branch, New Jersey, USA, n. d. a, b

[Engel et al. 1964] [Engel et al. 1964] Alfred Engel, John J. Kennedy, "Modelling dynamic economic problems on the analog computer", in *International Journal of Computer Mathematics*, 1(1), DOI: 10.1080/00207167508803020, pp. 289–311 a, b

[Engelsman 1992] [Engelsman 1992] W. P. den Engelsman, *Shimmy, an Introduction*, in *DCT rapporten*, Vol. 1992.130, Technische Universiteit Eindhoven, 1992 →

[Englert et al.] [Englert et al.] R. Englert, M. Siess, *Versuch 7: Pharmakokinetik – Erläuterung der Grundbegriffe, Bestimmung pharmakokinetischer Größen, Therapeutische Anwendung pharmakokinetischer Daten, Simulierung von pharmakokinetischen Vorgängen im Analogrechner*, n. d., author's collection →

[Enns et al. 1971] [Enns et al. 1971] Mark Enns, Theo C. Giras, Norman R. Carlson, "Load Flows by Hybrid Computation for Power System Operation", in *IEEE Transactions on Power Apparatus and Systems*, Volume PAS-90, Issus 6, November 1971, pp. 2540–2547 →

[Epstein 2024] [Epstein 2024] Katherine C. Epstein, *Analog Superpowers – How Twentieth-Century Technology Theft Built the National Security State*, The University of Chicago Press, 2024 →

[Erismann 1962] [Erismann 1962] Theodor Erismann, "Digitale Integrieranlagen und semidigitale Methoden", in [Hoffmann 1962, pp. 160–211] a, b, c, d, e, f, g

[Ernst 1960] [Ernst 1960] Dietrich Ernst, *Elektronische Analogrechner – Wirkungsweise und Anwendung*, R. Oldenbourg Verlag München, 1960 →

[Esch 1969] [Esch 1969] John William Esch, *RASCEL – A Programmable Analog Computer Based on a Regular Array of Stochastic Computing Element Logic*, Department of Computer Science, University of Illinois, Urbana, Illinois, Report No. 332, June, 1969 a, b, c

[Eterman 1960] [Eterman 1960] I. I. Eterman, *Analogue Computers*, Pergamon Press, 1960 →

[Euler 1768] [Euler 1768] Leonhardi Euleri, *Institutiones Calculi Integralis*, Volumen Primum, Petropoly, →https://books.google.de/books?id=Vg8OAAAAQAAJ, 1768, retrieved 14.06.2025 →

[Euser 1966] [Euser 1966] Ir. P. Euser, "Elektrisch analogon voor klimaattechnische berekeningen", in *Dr Raadgevend-Ingenieur*, Jaargang 8, no. 10, 1966, pp. 3–7 →

[Evans 1959] [Evans 1959] William T. Evans, *Analog Computer to Determine Seismic Weathering Time Corrections*, United States Patent 2884194, April 28, 1959 →

[Evans et al. 1966] [Evans et al. 1966] D. C. Evans et al., *Digital Differential Analyzer in Conjunction with a General Purpose Computer*, United States Patent US 3274376, Sept. 20, 1966 →

[Evans 2013] [Evans 2013] Lawrence C. Evans, *An Introduction to Stochastic Differential Equations*, American Mathematical Society, 2013 →

[Everyday Science and Mechanics 1932] [Everyday Science and Mechanics 1932] N. N., "Mechanical SUPER-BRAINS – Calculations in Higher Mathematics Performed by Complex Machinery", in *Everyday Science and Mechanics*, June 1932, pp. 625, 678 →

[Eyman et al. 1976] [Eyman et al. 1976] Earl D. Eyman, Yevgeny V. Kolchev, "Universal Analog Computer Model for Three-Phase

Controlled Rectifier Bridges", in *IEEE PES Winter Meeting & Tesla Symposium*, New York, N.Y., January 25–30, 1976, pp. 1136–1144 →

[Fagen 1978] [Fagen 1978] M. D. Fagen (ed.), *A History of Engineering and Science in the Bell System – National Service in War and Peace (1925–1975)*, Bell Telephone Laboratories, Inc., First Printing, 1978 a, b, c, d, e

[Falk 1958] [Falk 1958] Harold Charles Falk, *An Electronic Fluid Flow Analyzer for Pipeline Networks*, Master of Science thesis, South Dakota State College of Agriculture and Mechanic Arts, May 1958 →

[Fano 1950] [Fano 1950] Robert Mario Fano, "Short-Time Autocorrelation Functions and Power Spectra", in *The Journal of the Acoustical Society of America*, Volume 22, Number 5, pp. 546–550 →

[Farris 1964] [Farris 1964] George Joseph Farris, *A Digital Differential Analyzer Programming System for the I.B.M. 7074 Computer*, dissertation, Iowa State University of Science and Technology, Ames, Iowa, 1964 a, b, c, d, e

[Farshchi 2016] [Farshchi 2016] Shahin Farshchi, "How Analog and Neuromorphic Chips Will Rule the Robotic Age – AI is going to make analog computing popular again", in *IEEE Spectrum*, →https://spectrum.ieee.org/analog-and-neuromorphic-chips-will-rule-robotic-age, retrieved 16.08.2025 →

[Fasol 1986] [Fasol 1986] Karl-Heinz Fasol, "Modelling and Simulation of a Controlled Motor-Vehicle Drive", in [EAI 1986, pp. 30 ff.] →

[Fatton 1967] [Fatton 1967] Reneé Fatton, "Regulated Drive of a Train of Spring-Coupled Vehicles", in [IACM 1976, pp. 740–745] →

[Feilmeier 1974] [Feilmeier 1974] Manfred Feilmeier, *Hybridrechnen*, International Series of Numerical Mathematics, Vol. 2, Birkhäuser Verlag, 1974 a, b, c, d, e, f, g, h, i, j

[Fenech et al. 1973] [Fenech et al. 1973] Henri Fenech, Jay Bane, "A nuclear reactor analog simulation for undergraduate nuclear engineering education", in *Simulation*, Volume 20, Issue 4, April 1974, pp. 127–135 a, b

[Fifer 1961] [Fifer 1961] Stanley Fifer, "Analogue Computation – Theory, Techniques and Applications", Vol. III, McGraw-Hill Book Company, Inc., 1961 a, b

[Filipsky et al. 1977] [Filipsky et al. 1977] H. Filipsky, H. Aibel, Rattay, "Die Optimierung der elastischen Lagerung eines Verbrennungsmotors mit Hilfe der Hybridrechenanlage an der TU-Wien", in [TU Wien 1977/2, pp. 30 ff.] →

[Fincham 1963] [Fincham 1963] William Frank Finchman, *A study of the Mechanism Regulating the Composition of the Human Blood by means of an Electrical Analogue Computer*, doctoral thesis, Imperial College of Science and Technology, April 1963 a, b, c, d

[Fischer 2022] [Fischer 2022] Thomas Fischer, *THE ANALOG THING – First Steps*, →https://the-analog-thing.org/THAT_First_Steps.pdf, retrieved 07.08.2022 a, b

[Fogarty et al. 1963] [Fogarty et al. 1963] L. E. Fogarty, R. M. Howe, "Analog Computer Solution of the Orbital Flight Equations", in *Simulation*, Fall 1963, pp. R-41–R-45 →

[Foote et al. 2007] [Foote et al. 2007] Robert L. Foote, Ed Sandifer, "Area Without Integration: Make Your Own Planimeter", in *Hands on History – A Resource for Teaching Mathematics*, Amy Shell-Gellash, ed., The Mathematical Association of America (Incorporated), 2007 a, b

[Forbes 1955] [Forbes 1955] George F. Forbes, “The Digital Differential Analyzer”, in *Computers and Automation*, 1955, Vol. 4, No. 1, pp. 8–10 →

[Forbes 1957] [Forbes 1957] Georges F. Forbes, *Digital Differential Analyzers*, Fourth Edition, 1957 →

[Forbes 1972] [Forbes 1972] George Forbes, “The simulation of partial differential equations on the digital differential analyzer”, in ACM ’72 Proceedings of the ACM annual conference, Volume 2, pp. 860–866 a, b

[Foreman 2020] [Foreman 2020] Mark R.StJ Foreman, “Using op-amps to model the Phillips economic computer in SPICE”, in *RadCom Plus*, Volume 5, Issue 1, Autumn 2020, pp. 34–44 →

[Fornel et al. 1981] [Fornel et al. 1981] B. de Fornel, H. C. Hapiot, J. M. Farines, J. Hector, “Hybrid Simulation of a Current Fed Asynchronous Machine”, in *Mathematics and Computers in Simulation*, XXIII (1981), pp. 253–261 →

[Fortune 1952] [Fortune 1952] N. N., “The Moniac – ‘Economics in thirty fascinating minutes’”, in *Fortune*, March 1952, pp. 101–102 →

[Fox et al. 1963] [Fox et al. 1963] J. C. Fox, T. G. Windeknecht, “Six degree-of-freedom simulation of a manned orbital docking system”, in *AFIPS ’63 (Spring): Proceedings of the May 21–23, 1963, spring joint computer conference*, May 1963, pp. 91–104 →

[Fox et al. 1969] [Fox et al. 1969] Harry W. Fox, Ronald J. Blaha, *An Analog Computer Study of the Low-Frequency Dynamics of two Nuclear-Rocket Cold-Flow Engine Systems*, National Aeronautics and Space Administration, Washington, D. C., July 1969 →

[François-Lavet et al. 2018] [François-Lavet et al. 2018] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, Joelle Pineau, “An Introduction to Deep Reinforcement Learning”,

in *Foundation and Trends in Machine Learning*, Vol. 11, No. 3–4, pp. 219–354 →

[Franklin et al. 1969] [Franklin et al. 1969] M. A. Franklin, J. C. Strauss, “A hybrid computer programming system”, in *Fall Joint Computer Conference*, 1969, pp. 275–285 →

[Freedman 2011] [Freedman 2011] Immanuel Freedman, *System for forecasting outcomes of clinical trials*, United States Patent US 2011/0238317 A1, Sep. 29, 2011 →

[Freeth 2008] [Freeth 2008] Tony Freeth, *The Antikythera Mechanism – Decoding an Ancient Greek Mystery*, Whipple Museum of the History of Science, University of Cambridge, 2008 a, b, c

[Freeth et al. 2021] [Freeth et al. 2021] Tony Freeth, David Higgon, Aris Dacanalis, Lindsay MacDonald, Myrto Georgakopoulou, Adam Wojcik, “A Model of the Cosmos in the ancient Greek Antikythera Mechanism”, in *Scientific Reports*, nature portfolio, 2021, 11:5821, →https://doi.org/10.1038/s41598-021-84310-w →

[Freeth 2022] [Freeth 2022] Tony Freeth, “Wonder of the Ancient World”, in *Scientific American*, 326, 1, January 2022, pp. 24-33 →

[Fremerey] [Fremerey] Johan K. Fremerey, *Permanentmagnetische Lager*, Forschungszentrum Jülich, Institut für Grenzflächenforschung und Vakuumphysik, 0B30-A30, n. d. a, b

[Fremerey 1978] [Fremerey 1978] Johan K. Fremerey, Karl Boden, “Active permanent magnet suspensions for scientific instruments”, in *Journal of Physics E – Scientific Instruments*, February 1978, Vol. 11, No. 2, pp. 106–113 →

[Fricke et al. 1964] [Fricke et al. 1964] L. H. Fricke, R. A. Walsh, “The Use of a Portable Analog Computer for Process Identification, Calculation and Control”, in *Proceedings – Fall Joint Computer Conference*, 1964, pp. 685–693 →

[Friedman 2008] [Friedman 2008] Norman Friedman, *Naval Firepower – Battleship Guns and Gunnery in the Dreadnought Era*, Seaforth Publishing, 2008 →

[Fries 1991] [Fries 1991] Sylvia Doughty Fries, *NASA Engineers and the Age of Apollo*, The NASA History Series, NASA, 1992 →

[Frisch 1968] [Frisch 1968] Willi Frisch, *Stabilitätsprobleme bei dampfgekühlten schnellen Reaktoren*, Dissertation, Universität Karlsruhe, 1968 →

[Frisch et al. 1969] [Frisch et al. 1969] Willi Frisch, G. Wilhelmi, *Dynamische Simulatoren in der Reaktorentwicklung – ein Vergleich*, Gesellschaft für Kernforschung mbH, Karlsruhe, Januar 1969, 8/69-1 a, b

[Frisch 1971] [Frisch 1971] Willi Frisch, *Analogrechnen in der Kernreaktorrechnik*, G. Braun Karlsruhe, 1971 a, b

[Fritz 1967] [Fritz 1967] J. Fritz, "Simulation Analogique du Rayonnement Electromagnetique d'une Antenne de Revolution dans la Vide Puis dans un Plasma a l'Aide d'un Reseau", in [IACM 1976, pp. 916–931] →

[Gaines 1967] [Gaines 1967] Brian R. Gaines, "Stochastic Computing", in *AFIPS '67 (Spring): Proceedings of the April 18–20, 1967, spring joint computer conference*, April 1967, pp. 149–156 a, b, c

[Gaines 2019] [Gaines 2019] Brian R. Gaines, "Origins of Stochastic Computing", in [Gross et al. 2019, pp. 13–38] →

[Gallagher et al. 1957] [Gallagher et al. 1957] John M. Gallagher, William W. Seifert, "Simulation Techniques for Nuclear Reactor Studies", in [White Sands 1957, pp. 69–89] a, b

[GAP/R EAC] [GAP/R EAC] N. N., *GAP/R Electronic Analog Computors*, George A. Philbrick Researches, Inc., n. d. a, b

[GAP/R Evolution] [GAP/R Evolution] N. N., *A Brief History of a Computor-Builder in Terms of Ideas & Instrument – Evolution of a* Model *Kit of Tools*, George A. Philbrick Researches, Inc., n. d. →

[GAP/R K2W] [GAP/R K2W] N. N., *Model K2-W Operational Amplifier*, George A. Philbrick Researches, Inc., n. d. →

[GAP/R 1959] [GAP/R 1959] N. N., *Squaring, Rooting, and the Douglas Quadratron*, GAP/R Application Brief, George A. Philbrick Researches, Inc., No. D1, December 1, 1959 →

[Gaskill et al. 1963] [Gaskill et al. 1963] R. A. Gaskill, J. W. Harris, A. L. McKnight, “DAS – a Digital Analog Simulator”, in *Proceedings – Spring Joint Computer Conference*, 1963, pp. 83–90 →

[Gaudet et al. 2019] [Gaudet et al. 2019] Vincent C. Gaudet, Warren J. Gross, Kenneth C. Smith, “Introduction to Stochastic Computing”, in [Gross et al. 2019, pp. 1–11] →

[General Electric 1959/1] [General Electric 1959/1] General Electric, *Improvements in Electronic Terrain Analogue Simulator for Aircraft*, UK patent 909,140, January 2, 1959 a, b

[General Electric 1959/2] [General Electric 1959/2] General Electric, *Improvements in Electronic Terrain Analogue Simulator for Aircraft*, UK patent 964,427, December 31, 1959 a, b

[GEO SPACE] [GEO SPACE] GEO SPACE, *DYSTAC SS-100 – Solid State Analog and Hybrid Computation at* ±100 *Volts*, n. d. →

[Gerecke 1967/1] [Gerecke 1967/1] Eduard Gerecke, “Darstellung von Stromrichterschaltungen auf dem Analogrechner”, in [IACM 1976, pp. 697–708] →

[Gerecke 1967/2] [Gerecke 1967/2] Eduard Gerecke, “Schleudern bei Thyristorlokomotiven”, in [IACM 1976, pp. 729–739] →

[Gerecke et al. 1967] [Gerecke et al. 1967] Eduard Gerecke, Thomas Müller, “Untersuchung von Thyristor-gesteuerten-Drehstrom-

Kaefiglaeufer-Motoren mit dem Analogrechner", in [IACM 1976, pp. 709–718] →

[Gerwin 1958] [Gerwin 1958] Robert Gerwin, "Atom-Strom für deutsche Städte", in *Hobby – Das Magazin der Technik*, Nr. 9, September 1958 →

[Gilbert 1968/1] [Gilbert 1968/1] Barrie Gilbert, "A New Wide-Band Amplifier Technique", in *IEEE Journal of Solid-State Circuits*, Vol. SC-3, No. 4, December 1968, pp. 353–365 →

[Gilbert 1968/2] [Gilbert 1968/2] Barrie Gilbert, "A Precise Four-Quadrant Multiplier with Subnanosecond Response", in *IEEE Journal of Solid-State Circuits*, Vol. SC-3, No. 4, December 1968, pp. 365–373 →

[Gilbert 1970] [Gilbert 1970] Leonard J. Gilbert, *Analog Computer Simulation of a Parasitically Loaded Rotating Electrical Power Generating System*, National Aeronautics and Space Administration, Washington, D. C., November 1970 →

[Gilbert 1982] [Gilbert 1982] Barrie Gilbert, "A monolithic microsystem for analog synthesis of trigonometrix functions and their inverses", in *IEEE Journal of Solid-State Circuits*, Volume 17, Issue 6, Dec. 1981, pp. 1179–1191 →

[Giloi 1960] [Giloi 1960] W. Giloi, "Behandlung von Transformatorproblemen mit dem Analogrechner", in *Telefunken Zeitung*, Vol. 33 (1960), No. 129, p. 50 →

[Giloi 1961] [Giloi 1961] W. Giloi, "Ein Verfahren zur Berechnung von Optimalfiltern auf dem Analogrechner", in *Elektronische Rechenanlagen*, 3 (1961), No. 2, pp. 61–65 →

[Giloi 1962] [Giloi 1962] W. Giloi, "Über die Behandlung elektrischer und mechanischer Netzwerke auf dem Analogrechner", in *Elektronische Rechenanlagen*, 4 (1962), No. 1, pp. 27–35 →

[Giloi 1963] [Giloi 1963] W. Giloi, “Hybride Rechenanlagen – ein neues Konzept”, in *Elektronische Rechenanlagen*, 5 (1963), No. 6, pp. 262–269 →

[Giloi et al. 1963] [Giloi et al. 1963] Wolfgang Giloi, Rudolf Lauber, *Analogrechnen*, Springer-Verlag, 1963 a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t

[Giloi et al. 1968] [Giloi et al. 1968] Wolfang Giloi, Reinhard Fritz, “Computer aided design of quarter-square multipliers”, in *Simulation*, September 1968, pp. 125–131 →

[Giloi et al. 1969] [Giloi et al. 1969] Wolfgang Giloi, Dieter Beckert, Hans C. Liebig, “A flexible standard programming system for hybrid computation”, in *Spring Joint Computer Conference*, 1969, pp. 827–841 →

[Giloi 1975] [Giloi 1975] W. K. Giloi, *Principles of Continuous System Simulation*, B. G. Teubner, Stuttgart, 1975 a, b, c, d, e, f

[Giloi et al.] [Giloi et al.] W. Giloi, R. Herschel, *Rechenanleitung für Analogrechner*, Telefunken-Fachbuch, AFB 001, n. d. →

[Gilpin 1973] [Gilpin 1973] Michael E. Gilpin, “Do Hares Eat Lynx?”, in *The American Naturalist*, Vol. 107, No. 957, pp. 727–730 →

[Giordano et al. 1967] [Giordano et al. 1967] Paolo Giordano, Agostino Mathis, Bruno Rimini, “Analog Methods for Studying the Space-Time Dynamics of Nuclear Reactors”, in [IACM 1976, pp. 966–979] →

[Glaschick 2023] [Glaschick 2023] Rainer Glaschick, “Claude Shannon und das Bit”, in *Frankfurter Allgemeine Zeitung*, 15.01.2023 →

[Glatzel et al. 1967] [Glatzel et al. 1967] K. Glatzel, G. Schweizer, K. Stopfkuchen, “Die Anwendung Digitaler und Hybrider Simulation in der Satellitentechnik”, in [IACM 1976, pp. 932–946] →

[Gleiser 1980] [Gleiser 1980] Molly Gleiser, “Analog Inventor”, in *DATAMATION*, October 1980, pp. 141–143 →

[Glumineau et al. 1982] [Glumineau et al. 1982] A. Glumineau, R. Mezencev, “Hybrid Simulation of a Tanker Moored at a Single Point Subjected to Effects of Wind, Current and Waves”, in *10th IMACS World Congress on System Simulation and Scientific Computation*, 1982, pp. 98–100 →

[Goldberg 1951] [Goldberg 1951] Edwin A. Goldberg, “Step Multiplier in Guided Missile Computer”, in *ELECTRONICS*, August, 1951, pp. 120–124 →

[Goldberg et al. 1954] [Goldberg et al. 1954] Edwin A. Goldberg, Jules Lehmann, *Stabilized Direct Current Amplifier*, United States Patent 2684999, July 27, 1954 a, b

[Goldbohm 1993] [Goldbohm 1993] E. Goldbohm, “Near field measurements and the synthesis of linear arrays with prescribed radiation patterns: a survey of some early work at Christian Huygens Laboratory”, in *IEEE Antennas and Propagation Magazine*, Volume 35, Issue 3, June 1993, pp. 26–34 →

[Goldman 1965] [Goldman 1965] Mark W. Goldman, “Design of a High Speed DDA”, in *AFIPS ’65 (Fall, part I) Proceedings of the November 20–December 1, 1965, fall joint computer conference*, part I, pp. 929–949 →

[Goldsmith et al. 1948] [Goldsmith et al. 1948] Thomas T. Goldsmith, Estle Ray Mann, *Cathode-Ray Tube Amusement Device*, United States Patent 2455992, Dec. 14, 1948 →

[Golten et al. 1967] [Golten et al. 1967] J. W. Golten, David Rees, “The Use of Hybrid Computing in the Analysis of Steel Rolling”, in *IEEE Transactions on Electronic Computers*, Vol. EC-16, No. 6, December 1967, pp. 717–722 →

[Gomperts et al. 1957] [Gomperts et al. 1957] R. J. Gomperts, D. W. Righton, “LACE (The Luton Analogue Computing Engine (Part 1)”, in *Electronic Engineering*, Vol. 29, July 1957, pp. 306–312 →

[Gorez 1967] [Gorez 1967] R. Gorez, “Vibrations de Parties Elastiques d’une Sonde Spatiale Stabilisee par Rotation”, in [IACM 1976, pp. 907–909] →

[Gottzein] [Gottzein] Eveline Gottzein, *Hybridsimulatoren unerläßlich für die Systemführung*, Sonderdruck, *Bölkow-WMD/SIAT*, Report 3, n. d. →

[Govindarao 1975] [Govindarao 1975] V. M. H. Govindarao, “Analog Simulation of an Isothermal Semibatch Bubble-Column Slurry Reactor”, in *Annales de l’Association internationale pour le Calcul analogique*, No. 2, Avril 1975, pp. 69–78 →

[Graça et al. 2003] [Graça et al. 2003] Daniel Silva Graça, José Félix Costa, “Analog computer and recursive functions over the reals”, in *Journal of Complexity*, 19, 2003, pp. 644–664 →

[Graça et al. 2011] [Graça et al. 2011] Daniel Silva Graça, Manuel L. Campagnolo, Jorge Buescu, *Computability with Polynomial Differential Equations*, → https://sqigmath.tecnico.ulisboa.pt/pub/GracaDS/06-GCB-Poly.pdf, retrieved 07.07.2025 →

[Gracon et al. 1970] [Gracon et al. 1970] T. J. Gracon, J. C. Strauss, “Design of automatic patching systems for analog computers”, in *AFIPS ’70 (Spring): Proceedings of the May 5–7, 1970, spring joint computer conference*, May 1970, pp. 31–38 →

[Graefe et al. 1974] [Graefe et al. 1974] P. W. U. Graefe, Leo K. Nenonen, “Simulation of combined discrete and continuous systems on a hybrid computer”, in *SIMULATION*, May 1974, pp. 129–137 →

[Grandi et al. 1951] [Grandi et al. 1951] Louis L. Grandi, Don Lebell, “Analog Computers Solve Complex Problems”, in *Radio & Television News*, 11, 1951, pp. 70–71… 138–140 →

[Gray 1948] [Gray 1948] John W. Gray, “Direct-Coupled Amplifiers”, in *Vacuum Tube Amplifiers*, Massachusetts Institute of Technology, Radiation Laboratory Series, pp. 409–495 →

[Gray et al. 1955] [Gray et al. 1955] John W. Gray, Duncan MacRae, *Bombing Computer*, United States Patent 2711856, June 28, 1955 →

[Gray 1958] [Gray 1958] H. J. Gray Jr., “Digital Computer Solution of Differential Equations in Real Time”, in IRE-ACM-AIEE ’58 (Western) Proceedings of the May 6–8, 1958, western joint computer conference: contrasts in computers, pp. 87–92 →

[Green 1959] [Green 1959] F. P. Green, *The ORNL Four-Linear-Element Package Analog Computer (FLEPACK) Instruction Manual*, Oak Ridge National Laboratory, ORNL-2727, TID-4500, Oct. 13 1959 →

[Grier 1998] [Grier 1998] David Alan Grier, “The Math Tables Project of the Work Projects Administration: The Reluctant Start of the Computing Era”, in *IEEE Annals of the History of Computing*, Vol. 20, No. 3, July-Sept. 1998, pp. 33–50 →

[Grier 2007] [Grier 2007] David Alan Grier, “Table making for the relief of labour”, in [Campbell-Kelly et al. 2007, pp. 265–292] →

[Grierson et al.] [Grierson et al.] W. O. Grierson, D. B. Lipski, N. O. Tiffany, *Simulation Tools: Where can we go?*, n. d., author’s archive →

[Grimwood et al. 1962] [Grimwood et al. 1962] James M. Grimwood, Frances Strowd, *History of the Jupiter Missile System*, U.S. Army Ordnance Missile Command, 27 July 1962 →

[Griswold et al. 1957] [Griswold et al. 1957] James A. Griswold, Herbert V. Loomis, “Ship Motion Simulator”, in [White Sands 1957, pp. 11–19] →

[Gröbner 1961] [Gröbner 1961] W. Gröbner, “Steuerungsprobleme mit Optimalbedingung”, in *mtw – Zeitschrift für moderne Rechentechnik und Automation*, 2/61, pp. 62–64 →

[Gronbach et al. 2012] [Gronbach et al. 2012] Thomas Müller-Gronbach, Erich Nowak, Klaus Ritter, *Monte Carlo-Algorithmen*, Springer Verlag, 2012 →

[Gross et al. 2019] [Gross et al. 2019] Warren J. Gross, Vincent C. Gaudet (eds.), *Stochastic Computing: Techniques and Applications*, Springer Nature Switzerland, 2019

[Grubert 1980] [Grubert 1980] Michael Grubert, “Ein Analogrechner für Lehr- und Versuchszwecke”, in *Elektronik*, Heft 9, 1980, pp. 73–79 →

[Gründorfer 1974] [Gründorfer 1974] S. Gründorfer, “Simulation von Einschwingvorgängen in Halbleiter-LASERn”, in [TU Wien 1974, pp. 30 f.] →

[Gsandtner 1975] [Gsandtner 1975] Anton Gsandtner, “PACER MUSICALE”, in [TU Wien 1975/1, pp. 22 f.] →

[Gudonawitschius et al. 1977/1] [Gudonawitschius et al. 1977/1] Romualdas Gudonawitschius, Irmgard Husinsky, “Ein Vergleich einiger Verfahren der Merkmalbestimmung für Spracherkennung”, in [TU Wien 1977/2, pp. 47 ff.] →

[Gudonawitschius et al. 1977/2] [Gudonawitschius et al. 1977/2] Romualdas Gudonawitschius, Irmgard Husinsky, “Ein Vergleich einiger Verfahren der Merkmalbestimmung für die Spracherkennung”, in [TU Wien 1977/3, pp. 57 ff.] →

[Gudonawitschius et al. 1979] [Gudonawitschius et al. 1979] Romualdas Gudonawitschius, Irmgard Husinsky, “Klassifikation von akustischen Signalen durch Phasenbildverfahren”, in [TU Wien 1979/1, pp. 40 ff] →

[Gudonawitschius et al. 1979] [Gudonawitschius et al. 1979] Romualdas Gudonawitschius, Irmgard Husinsky, “Vergleich verschiedener Klassifizierer für die automatische Erkennung von akustischen Signalen”, in [TU Wien 1979/2, pp. 31 ff.] →

[Guibert et al. 1949] [Guibert et al. 1949] A. G. Guibert, E. Janssen, W. M. Robbins, *Determination of Rate, Area, and Distribution of Impingement of Waterdrops on Various Airfoils from Trajectories Obtaines on the Differential Analyzer*, National Advisory Committee for Aeronautics, Washington, Research Memorandum No. 9A05, February 16, 1949 →

[Gul’ko 1961] [Gul’ko 1961] F. B. Gul’ko, “Thyrite Multiplier with an Increased Passband”, translated from *Avtomatika i Telemekhanika*, Vol. 22, No. 12, December, 1961, pp. 1649–1655 →

[Gummel et al. 1967] [Gummel et al. 1967] H. K. Gummel, B. T. Murphy, “Circuit analysis by quasi-analog computation”, in *Proceedings of the IEEE*, Volume 55, Issue 10, October 1967, pp. 1758–1760 →

[Gumpert 1972] [Gumpert 1972] W. Gumpert, “Analogberechnung selbsterregter Schwingungen beim Anfahren von Fahrzeugen”, in *Kraftfahrzeugtechnik*, 2/72, pp. 46–48 →

[Gundlach 1955] [Gundlach 1955] F. W. Gundlach, “A new electronic-beam multiplier with an electrostatic hyperbolic field”, in *Actes. J. Internat. Calcul Analog Analogique*, Brüssel, 1955, pp. 101–103 →

[Guo et al. 2016] [Guo et al. 2016] Nign Guo, Yipeng Huang, Tao Mai, Sharvil Patil, Chi Cao, Mingoo Seok, Simha Sethumadhavan, Yannis

Tsividis, “Energy-Efficient Hybrid Analog/Digital Approximate Computation in Continuous Time”, in *IEEE Journal of Solid-State Circuits*, Vol. 51, No. 7, July 2016, pp. 1514–1524 →

[Guo 2017] [Guo 2017] Ning Guo, *Investigation of Energy-Efficient Hybrid Analog/Digital Approximate Computation in Continuous Time*, Columbia University, 2017 a, b

[Guzdial et al. 2013] [Guzdial et al. 2013] Mark Guzdial, Daniel Reed, “Securing the Future of Computer Science; Reconsidering Analog Computing”, in *Communications of the ACM*, April 2013, Vol. 56, No. 4, pp. 11–12 →

[Habermehl et al. 1969] [Habermehl et al. 1969] A. Habermehl, E. H. Graul, *Analog Computer Investigations with a Mathematical Model for the Regulation Mechanism of Calcium Metabolism*, National Aeronautics and Space Administration, Washington, D. C., April 1969 a, b

[Habermehl et al. 1969/2] [Habermehl et al. 1969/2] A. Habermehl, E. H. Graul, H. Wolter, *Computer in der Nuklearmedizin – Die Anwendung des Analogrechners zur Untersuchung biologischer Systeme und Prozesse*, Dr. Alfred Hüthig Verlag Heidelberg, 1969 →

[Haeberlin et al. 2011] [Haeberlin et al. 2011] Barbara Haeberlin, Stefan Drechsler, *Wie funktioniert ein Kegel-Reibrad-Planimeter?*, RST 22, Worms, 22.10.2011, →http://www.rechenschieber.org/haedre2011.pdf, retrieved 22.11.2012 →

[Hall et al. 1969] [Hall et al. 1969] Carroll R. Hall, Stephen J. Kahne, “Automated Scaling for Hybrid Computers”, in *IEEE Transactions on Computers*, Vol. C-18, No. 5, May 1969, pp. 416–423 →

[Hall 1996] [Hall 1996] Eldon C. Hall, *Journey to the Moon: The History of the Apollo Guidance Computer*, American Institute of

Aeronautics and Astronautics, Inc., 1996 →

[Hammack et al. 2014] [Hammack et al. 2014] Bill Hammack, Steve Kranz, Bruce Carpenter, *Albert Michelson's Harmonic Analyzer – A Visual Tour of a Nineteenth Century Machine that Performs Fourier Analysis*, Articulate Noise Books, 2014 →

[Hammer 1956] [Hammer 1956] J. A. Hammer, "Evaluation of Diffraction Problems in Aerial Technology by Means of an Analyser for Computer and Fourier Series", in *Principes fondamentaux: Équations de Maxwell, Principe de Huygens et Théory de la diffraction en hyper-fréquences*, 1956 →

[Hamori 1972] [Hamori 1972] Eugene Hamori, "Use of the Analog Computer in Teaching Relaxation Kinetics", in *Journal of Chemical Education*, Volume 49, Number 1, January 1972, pp. 39–43 →

[Hannauer 1968] [Hannauer 1968] George Hannauer, *Stored Program Concept for Analog Computers*, final report, EAI project 320009, NASA order NAS8-21228 a, b, c, d

[Hannigan] [Hannigan] Frank J. Hannigan, *Hybrid Computer Simulation of Fluidic Devices*, Electronic Associates, Inc., n. d. →

[Hansen et al. 1959] [Hansen et al. 1959] P. D. Hansen, J. H. Eaton, *Control and Dynamics Performance of a Sodium Cooled Reactor Power System*, MICROTECH RESEARCH COMPANY, Massachusetts, Report No. 171, December 28., 1959 →

[Hansen 2005] [Hansen 2005] James R. Hansen, *First Man – The Life of Neil A. Armstrong*, Simon & Schuster UK, 2005 a, b, c, d, e

[Hanus 1967] [Hanus 1967] B. Hanus, "Untersuchungen der Regelung eines Durchflusskessels 200 MW an einem Analogrechner", in [IACM 1976, pp. 849–854] →

[Harder et al. 1950] [Harder et al. 1950] Edwin L. Harder, J. T. Carleton, "New Techniques on the Anacom – Electric Analog

Computer”, in *Transactions of the American Institute of Electrical Engineers*, Vol. 69, No. 1, Jan. 1950, pp. 547–556 →

[Harder 1952] [Harder 1952] Edwin L. Harder, “Some Engineering Problems Requiring Automatic Computation”, in *Proceedings of the 1952 ACM national meeting (Pittsburgh)*, 02 May 1950, pp. 85–90 →

[Harnett et al. 1963] [Harnett et al. 1963] R. T. Harnett, J. F. Sansom, L. M. Warshawsky, “MIDAS... An analog approach to digital computation”, in *SIMULATION*, September 1963, S. 17–43 →

[Hart et al. 1967] [Hart et al. 1967] Clint E. Hart, Dale J. Arpasi, *Frequency Response and Transfer Functions of a Nuclear Rocket Engine System obtained from Analog Computer Simulation*, National Aeronautics and Space Administration, Washington, D. C., May 1967 a, b

[Hartmann et al. 1961] [Hartmann et al. 1961] John Hartmann, Granino A. Korn, Richard L. Maybach, “Low-Cost Triangle-Integration Multipliers for Analog Computers”, in Annales de l’Association international pour le Calcul analogique, No. 4, Octobre 1961, pp. 167–172 →

[Hasler 2017] [Hasler 2017] Jennifer Hasler, “We Could Build an Artificial Brain Right Now”, in *IEEE Spectrum*, →https://spectrum.ieee.org/we-could-build-an-artificial-brain-right-now, retrieved 16.08.2025 →

[Hasler 2019] [Hasler 2019] Jennifer Hasler, “Large-Scale Field-Programmable Analog Arrays”, in *Proceedings of the IEEE*, Volume 108, Issue 8, August 2020, pp. 1283–1302 →

[Hasler et al. 2020] [Hasler et al. 2020] Jennifer Hasler, Aishwarya Natarajan, “An Open-Source ToolSet for FPAA Design”, in *WOSET*

Workshop, 2020, → https://woset-workshop.github.io/PDFs/2020/a18.pdf, retrieved 28.02.2025 →

[Hasler et al. 2024] [Hasler et al. 2024] Jennifer Hasler, Cong Hao, "Programmable Analog System Benchmarks Leading to Efficient Analog Computation Synthesis", in *ACM Transactions on Reconfigurable Technology and Systems*, Vol. 17, No. 1, Article 12, January 2024 a, b

[Haug 1960] [Haug 1960] Albert Haug, "Funktionsgeneratoren und Funktionsspeicher der Formen $y = f(x)$ und $z = f(x, y)$", in *Nachrichtentechnische Fachberichte*, Band 17, Vieweg & Sohn, Braunschweig, 1960 a, b, c, d, e, f, g, h, i, j

[Hausner 1963] [Hausner 1963] Arthur Hausner, "Multiple Integrals on a Non-Repetivie Analog Computer", in *Proceedings – Spring Joint Computer Conference*, 1963, pp. 205–212 →

[Havil 2019] [Havil 2019] Julian Havil, *Curves for the Mathematically Curious – an Anthology of the Unpredictable, Historical, Beautiful, and Romantic*, Princeton University Press, 2019 →

[Hazen et al. 1940] [Hazen et al. 1940] H. L. Hazen, G. S. Brown, Walter R. Hedeman, "The cinema integraph – A machine for evaluating a parametric product integral", in *Journal of the Frankfurt Institute – Engineering and Applied Mathematics*, 1. Aug. 1940, pp. 183–205 →

[Headrick et al. 1969] [Headrick et al. 1969] J. B. Headrick, D. P. Jordan, "Analog Computer Simulation of the Heat Gain Through a Flat Composite Roof Section", in *ASHRAE Transactions*, Vol. 75, 1969, pp. 21–33 →

[Hedeman 1941] [Hedeman 1941] Walter R. Hedeman, "The Cinema Integraph in Interreflection Problems", April 1941, → https://doi.org/10.1002/sapm1941201402, retrieved 22.02.2022 →

[Hedin et al.] [Hedin et al.] Ronald A. Hedin, Kenneth W. Priest, *Progress in Hybrid Simulation of Power Systems*, n. d., author's archive →

[Heideprim 1976] [Heideprim 1976] J. Heideprim, "Modelle und Simulation von Produktionsprozessen in der Stahlindustrie", in [Schöne 1976/1, pp. 206–254] →

[Heidersberger] [Heidersberger] Heinrich Heidersberger, *Rhythmogramme*, CARGO Verlag, n. d. →

[Heimann et al. 2021] [Heimann et al. 2021] Lars Heimann, Bernd Ulmann, Sven Köppel, *Highly Linear Multiplier*, European Patent EP2021085622, filed 14.12.2021 →

[Heinhold 1959] [Heinhold 1959] J. Heinhold, "Konforme Abbildung mittels elektronischer Analogrechner", in *mtw – Zeitschrift für moderne Rechentechnik und Automation*, 1/59, pp. 44–48 →

[Heinhold et al. 1969] [Heinhold et al. 1969] Josef Heinhold, Ulrich Kulisch, *Analogrechnen – Eine Einführung*, Bibliographisches Institut Mannheim/Wien/Zürich, B.I.-Wissenschaftsverlag, 1969 a, b, c, d, e, f, g, h, i

[Heller et al. 1976] [Heller et al. 1976] Rainer Heller, Carl W. Malstrom, E. Harry Law, "Hybrid Simulation of Rail Vehicle Lateral Dynamics", presented at *The 1976 Summer Computer Simulation Conference*, Washington, D. C., July 12–14, 1976 a, b

[Hemmendinger 2014] [Hemmendinger 2014] David Hemmendinger, "COMIC: An Analog Computer in the Colorant Industry", in *IEEE Annals of the History of Computing*, Vol. 36, Issue 3, July–Sept. 2014, pp. 4–18 a, b

[Henderson et al. 1965] [Henderson et al. 1965] R. L. Henderson, J. C. Campbell, H. F. Gaydos, S. G. Kester, *Final Report Weapons Effects Display System Study*, SDC, Technical Memorandum, TM-2622/000/01, 1965 a, b, c

[Henrici 1894] [Henrici 1894] O. Henrici, “Report on Planimeters”, in *Report of the Sixty-Fourth Meeting of the British Association for the Advancement of Science*, London: John Murray, Albemarle Street, 1894 a, b, c, d

[Henze 2018] [Henze 2018] Norbert Henze, *Irrfahrten – Fazination der Random Walks*, Springer Spektrum, 2018 →

[Herschel 1957] [Herschel 1957] R. Herschel, “Elektronische Rechenmaschinen ohne Einmaleins”, in *Funkschau*, 1957, Heft 10, pp. 10–11 →

[Herschel 1961] [Herschel 1961] R. Herschel, “Automatische Optimisatoren”, in *Elektronische Rechenanlagen*, 3 (1961), No. 1, pp. 30–36 →

[Herschel 1962] [Herschel 1962] R. Herschel, “Analogrechenschaltungen für die Entwicklungskoeffizienten nach Orthogonalfunktionen”, in *Elektronische Rechenanlagen*, 3 (1961), No. 5, pp. 212–217 →

[Herschel 1966] [Herschel 1966] R. Herschel, “Zur Programmierung von hybriden Rechenanlagen in ALGOL”, in *Telefunken Zeitung*, Vol. 39 (1966), No. 1, pp. 100–109 →

[Herzog et al. 2018] [Herzog et al. 2018] Benedict Herzog, Luis Gerhorst, Bernhard Heinloth, Stefan Reif, Timo Hönig, Wolfgang Schröder-Preikschat, “INTspect: Interrupt Latencies in the Linux Kernel”, in *2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC)*, 2018, pp. 83–90 →

[Hicks 1968] [Hicks 1968] W. D. Hicks, *An Electronic Analogue Computer Representing Twelve Coupled Linear Differential Equations*, Ministry of Technology, Aeronautical Research Council, London, Her Majesty’s Stationary Office, 1968 →

[Higgins et al. 1982] [Higgins et al. 1982] W. H. C. Higgins, B. D. Holbrook, J. W. Emling, “Defense Research at Bell Laboratories”, in

Annals of the History of Computing, Volume 4, Number 3, July 1982, pp. 218–236 →

[Hill et al. 1964] [Hill et al. 1964] J. D. Hill, G. J. McMurty, K. S. Fu, "A Computer-Simulated On-Line Experiment in Learning Control Systems", in *Proceedings – Spring Joint Computer Conference*, 1964, pp. 315–325 a, b

[Hindmarsh et al. 1982] [Hindmarsh et al. 1982] J. L. Hindmarsh, R. M. Rose, "A model of the nerve impulse using two first-order differential equations", in *Nature*, Vol. 296, 11 March 1982, pp. 162–164 →

[Hindmarsh et al. 1984] [Hindmarsh et al. 1984] J. L. Hindmarsh, R. M. Rose, "A model of neuronal bursting using three coupled first order differential equations", in *Prov. R. Soc. Lond.*, B 221, 87–102 (1984) →

[Hintergräber 1967] [Hintergräber 1967] Max Hintergräber, "Analogrechneruntersuchungen über das Zeitverhalten eines Dampferzeugers mit Zwangsdurchlauf für Gasgekühlte Reaktoren", in [IACM 1976, pp. 1011–1029] →

[Hintze 1957] [Hintze 1957] Guenther Hintze, "Mathematical vs. Physical Simulation", in [White Sands 1957, pp. 3–9] →

[Hitachi 1967] [Hitachi 1967] N. N., "Double Integral – Calculation of Volume of Cone", in *Technical Information Series No. 3*, Hitachi Electronics, Ltd., 1967 →

[Hitachi 1968] [Hitachi 1968] N. N., "Analysis of Rolling Theory by Analog Computer – Karman's Differential Equation", in *Technical Information Series No. 8*, Hitachi Electronics, Ltd., 1968 →

[Hitachi 1969] [Hitachi 1969] N. N., "Introduction to Simulator – Part 1", in *Technical Information Series No. 9*, Hitachi Electronics, Ltd., 1969 a, b

[Hitachi 1970] [Hitachi 1970] N. N., *Hitachi – Analog/Hybrid Computer 220/2240 – Operator's Manual for Digital Logics*, Hitachi, 1970 →

[Hitachi 200X] [Hitachi 200X] N. N., *Hitachi Analog Hybrid Computer – Hitachi-200X*, Hitachi Electronics, Ltd., n. d. a, b, c, d, e, f, g

[Hitachi 505E] [Hitachi 505E] N. N., *HITACHI 505E analog/hybrid computer*, Hitachi Electronics, Ltd., n. d. →

[Hobby 1969] [Hobby 1969] N. N., "Psychedelic – Explosion der Farben", in *hobby – Das Magazin der Technik*, Nr. 15/69, p. 36–45 →

[Hoelzer 1946/1] [Hoelzer 1946/1] Helmut Hoelzer, *Anwendung elektrischer Netzwerke zur Lösung von Differentialgleichungen*, Dissertation TH Darmstadt, 1946 a, b, c, d, e, f, g, h

[Hoelzer 1946/2] [Hoelzer 1946/2] Helmut Hoelzer, *Anwendung elektrischer Netzwerke zur Stabilisierung von Regelvorgängen und zur Lösung von Differentialgleichungen, gezeigt an der Stabilisierung des Fluges einer selbst- bzw. ferngesteuerten Großrakete*, Dissertation TH Darmstadt, 1946 →

[Hoelzer 1976] [Hoelzer 1976] Helmut Hoelzer, "Guidance and Control Symposium", in *The Eagle has Returned: Proceedings of the Dedication Conference of the International Space Hall of Fame*, 1976, pp. 301–316 →

[Hoelzer 1992] [Hoelzer 1992] Helmut Hoelzer, *50 Jahre Analogcomputer*, Rede anlässlich des fünfzigsten Jubiläums des elektronischen Analogrechners im Senatssaal in Berlin, 12.05.1992, Manuskript aus dem Archiv der Familie Hoelzer-Beck a, b, c

[Hoffert 1964] [Hoffert 1964] Fritz Hoffert, "The Dornier DO 31 Jet-Lift Concept, A Light Military Transport with VTOL Capability",

in *SAE Technical Paper* 640229, 1964, →https://doi.org/10.4271/640229, retrieved 29.08.2025 →

[Hoffman 1979] [Hoffman 1979] Dale T. Hoffman, “Smart Soap Bubbles Can Do Calculus”, in *The Mathematics Teacher*, Vol. 72, No. 5, Mai 1979, pp. 377–385 →

[Hoffmann 1962] [Hoffmann 1962] Walter Hoffmann (ed.), *Digitale Informationswandler*, Friedrich Vieweg & Sohn, 1962

[Hoffmann 2006] [Hoffmann 2006] Justin Hoffmann / Kunstverein Wolfsburg (Hg.), *Der Traum von der Zeichenmaschine – Heinrich Heidersbergers Rhythmogramme und die Computergrafik ihrer Zeit*, Kunstverein Wolfsburg, Kataloge #1/2006 a, b

[W. Hofmann 1977] [W. Hofmann 1977] W. Hofmann, “Modell zur Simulation des Verhaltens natürlich radioaktiver Nuklide in der Luft von Wohn- und Arbeitsräumen mit Hilfe eines Hybridrechners”, in [TU Wien 1977/1, pp. 19 ff.] →

[Hofmann et al. 1977] [Hofmann et al. 1977] W. Hofmann, F. Steinhäusler, Hummer, “Modell zur Simulation der Altersabhängigkeit der Dosis im Atemtrakt bei der Inhalation natürlich radioaktiver Nuklide (Teil 1)”, in [TU Wien 1977/2, pp. 37 ff.] →

[Holst 1982] [Holst 1982] Per A. Holst, “George A. Philbrick and Polyphemus – The First Electronic Training Simulator”, in *Annals of the History of Computing*, Volume 4, Number 2, April 1982 a, b, c, d, e, f, g, h, i, j, k, l

[Holst 1996] [Holst 1996] Per A. Holst, “Svein Rosseland and the Oslo Analyzer”, in *IEEE Annals*, Winter 1996, Vol. 18, No. 4, pp. 16–26 →

[Holt et al. 1995] [Holt et al. 1995] Frederick B. Holt, Dziem D. Nguyen, *Pseudorandom Stochastic Data Processing*, United States Patent 9412587, May 2, 1995 →

[Holzer et al. 2021] [Holzer et al. 2021] Mirko Holzer, Bernd Ulmann, “Hybrid computer approach to train a machine learning system”, in *Handbook in Unconventional Computing*, ed. Andrew Adamatzky, World Scientific Publishers, 2021 a, b

[Hopcroft et al. 1979] [Hopcroft et al. 1979] J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979 →

[Horling et al.] [Horling et al.] James E. Horling, Esmat Mahmout, Frank J. Hannigan, *Hardware-in-the-Loop Simulation for Evaluating Turbine Engine Fuel System Components*, n. d., author’s archive →

[Horsman et al. 2014] [Horsman et al. 2014] Dominic Horsman, Susan Stepney, Rob C. Wanger, Viv Kendon, “When does a physical system compute?”, in *Proceedings of the Royal Society A*, Volume 470, Issue 2169, 08 September 2014, →https://doi.org/10.1098/rspa.2014.0182, retrieved 02.03.2025 →

[Hosenthien et al. 1962] [Hosenthien et al. 1962] H. H. Hosenthien, J. Boehm, “Flight Simulation of Rockets and Spacecraft”, in *From Peenemünde to Outer Space – Commemorating the Fiftieth Birthday of Wernher von Braun*, March 23, 1962, pp. 437–469 a, b, c, d, e

[Howard et al. 1953] [Howard et al. 1953] R. C. Howard, C. J. Savant, R. S. Neiswander, “Linear-to-Logarithmic Voltage Converter”, in *Electronics*, July, 1953, pp. 156–157 →

[Howard 1961] [Howard 1961] Warren Dee Howard, “The computer simulation of a colonial socio-economic system”, in *IRE-AIEE-ACM ’61 (Western): Papers presented at the May 9–11, 1961, western joint IRE-AIEE-ACM computer conference*, May 1961, pp. 613–622 →

[Howe 1953] [Howe 1953] C. E. Howe, *A Tabletop Electronic Differential Analyzer*, Department of Aeronautical Engeineering, University of Michigan, AIR-6, October 1953 →

[Howe et al. 1953] [Howe et al. 1953] Robert M. Howe, V. S. Haneman, “The solution of partial differential equations by difference methods using the electronic differential analyzer”, in *AIEE-IRE ’53 (Western) Proceedings of the February 4–6, 1953, western computer conference*, pp. 208–226 →

[Howe 1957] [Howe 1957] Robert M. Howe, “Coordinate Systems and Methods of Coordinate Transformations for Dimensional Flight Equations”, in [White Sands 1957, pp. 49–66] →

[Howe 1961] [Howe 1961] Robert M. Howe, *Design Fundamentals of Analog Computer Components*, D. van Nostrand Company, Inc., 1961 a, b, c, d, e, f

[Howe 1962] [Howe 1962] Robert M. Howe, “Solution of Partial Differential Equations”, in [Huskey et al. 1962, pp. 5-110–5-132] a, b

[Howe et al. 1970] [Howe et al. 1970] Robert M. Howe, Richard A. Moran, Thomas D. Berge, “Time sharing of hybrid computers using electronic patching”, in *AFIPS ’70 (Fall): Proceedings of the November 17–19, fall joint computer conference*, November 1970, pp. 377–386 →

[Howe et al. 1975] [Howe et al. 1975] Robert M. Howe, Aldric Saucier, “A new fourth generation of hybrid computer systems”, in *AFIPS ’75: Proceedings of the May 19–22, 1975, national computer conference and exposition*, May 1975, pp. 861–866 →

[Howe 2005] [Howe 2005] Robert M. Howe, “Analog Computers in Academia and Industry”, in *IEEE Control Systems Magazine*, June 2005, pp. 37–43 →

[HP 3722A] [HP 3722A] N. N., *Model 3722A Noise Generator – Operating and Service Manual*, Hewlett-Packard, 1971 →

[Hsiao et al. 2019] [Hsiao et al. 2019] Hsuan Hsiao, Jason Anderson, Yuko Hara-Azumi, “Generating Stochastic Bitstreams”, in [Gross et

al. 2019, pp. 137–152] →

[Hsu et al. 1968] [Hsu et al. 1968] Stephen K. T. Hsu, Robert M. Howe, “Preliminary investigation of a hybrid method for solving partial differential equations”, in *AFIPS ’68 (Fall, part 1): Proceedings of the December 9–11, 1968, fall joint computer conference, part 1*, December 1968, pp. 601–609 →

[Hu 1972] [Hu 1972] Richard H. Hu, “Analog Computer Simulation of an FM Communication System”, American Society for Engineering Education, Annual Conference, June 19–22, 1972 →

[Hu et al. 2024] [Hu et al. 2024] Jinhai Hu, Zhongyi Zhang, Cong Sheng Leow, Wang Ling Goh, Yuan Gao, “Late Breaking Results: Circuit-Algorithm Co-design for Learnable Audio Analog Front-End”, in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, 2024, Article No. 347, pp. 1–2 →

[Huang et al. 2016] [Huang et al. 2016] Yipeng Huang, Ning Guo, Mingoo Seok, Yannis Tsividis, Simha Sethumadhavan, “Evaluation of an Analog Accelerator for Linear Algebra”, in *43rd Annual Interational Symposium on Computer Architecture*, IEEE, 2016, pp. 570–582 →

[Huang et al. 2017] [Huang et al. 2017] Yipeng Huang, Ning Guo, Mingoo Seok, Yannis Tsividis, Kyle Mandli, “Hybrid Analog-Digital Solutions of Nonlinear Partial Differential Equations”, in *MICRO-50 ’17: Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, October 2017, pp: 665–678 →

[Huang 2018] [Huang 2018] Yipeng Huang, *Hybrid Analog-Digital Co-Processing for Scientific Computation*, Ph.D. thesis, Columbia University, 2018 →

[Huang et al. 2018] [Huang et al. 2018] Yipeng Huang, Ning Guo, Simha Sethumadhavan, Mingoo Seok, Yannis Tsividis, “A Case Study in

Analog Co-Processing for Solving Stochastic Differential Equations", 23rd DSP 2018, Shanghai, China →

[Hudson et al. 1945] [Hudson et al. 1945] C. S. Hudson, I. A. Mossop, *Flight Control Amplifier of the German Long Range Rocket*, Royal Aircraft Establishment, Farnborough, E. A. No. 228-22, Report No. El. 1365, March 1945 a, b, c

[Hugo Sachs Elektronik K. G.] [Hugo Sachs Elektronik K. G.] N. N., *HSE-Analog-Computer – Eine Neuentwicklung für die biomedizinische Forschung*, n. d. →

[Hülsenberg et al. 1975] [Hülsenberg et al. 1975] Frieder Hülsenberg, Uwe Kiessling, Hartmut Schönborn, *Beziehung zwischen Produktion, Lagerhaltung und Marktrealisation – dargestellt an einem Analogie-Rechenmodell für den Analogrechner MEDA T*, VEB Deutscher Verlag für Grundstoffindustrie, 1975 →

[Hütte 1926] [Hütte 1926] Akademischer Verein Hütte, e.V. in Berlin (ed.), *Hütte – Des Ingenieurs Taschenbuch*, 25. neubearbeitete Auflage, II. Band, Berlin 1926, Verlag von Wilhelm Ernst & Sohn a, b

[Huggins 1958] [Huggins 1958] Phyllis Huggins, "A Computer Case History: Miniature Complex Cams", in *Computers and Automation*, 1958, Vol. 7, No. 4, p. 10 ff. a, b

[Hughes et al. 1960] [Hughes et al. 1960] John V. Hughes, William L. Rollwitz, *Gas Pumping System Analog*, United States Patent 2951638, Sept. 6, 1960 →

[Hume et al. 2005] [Hume et al. 2005] Ted Hume, Bob Koppany (ed.), *The Oughtred Society Slide Rule Reference Manual*, Striking Impressions, Los Angeles, California, First Edition, 2005 →

[Hummer et al. 1976/1] [Hummer et al. 1976/1] H. Hummer, H. Novotny, "Zeitliche Entwicklung eines eindimensionalen quantenmechanischen Transmissions- und Reflexionsvorganges

am Beispiel eines Gauss’schen Wellenpaketes und eines Potentials beliebiger Form (Teil 1)”, in [TU Wien 1976/1, pp. 35 ff.] →

[Hummer et al. 1976/2] [Hummer et al. 1976/2] H. Hummer, H. Novotny, “Zeitliche Entwicklung eines eindimensionalen quantenmechanischen Transmissions- und Reflexionsvorganges am Beispiel eines Gauss’schen Wellenpaketes und eines Potentials beliebiger Form (Teil 2)”, in [TU Wien 1976/2, pp. 36 ff.] →

[Hund 1982] [Hund 1982] Martin Hund, *Der Analogrechner – Beschreibung mit 150 Programmbeispielen aus Mathematik, Regelungstechnik und Physik*, Leybold-Heraeus GmbH, 2. Auflage, 1982 →

[Hurley et al. 1963] [Hurley et al. 1963] J. R. Hurley, J. J. Skiles, “DYSAC: A digitally simulated analog computer”, in *Proceedings of the Spring Joint Computer Conference*, 1963, pp. 69–82 →

[Hurst] [Hurst] Charles J. Hurst, *Computer Simulation in a Mechanical Engineering Laboratory Program*, n. d., author’s archive →

[Husinsky 1976] [Husinsky 1976] W. Husinsky, “Lösung der Dichtematrix eines Atoms für monochromatische Anregung mit Hilfe eines hybriden Analogrechners”, in [TU Wien 1976/3, pp. 35 ff.] →

[Husinsky et al. 1985] [Husinsky et al. 1985] Irmgard Husinsky, Wolfgang Kleinert, “Die SIMSTAR Technologie”, in [TU Wien 1985, pp. 5 ff.] a, b, c

[Huskey et al. 1962] [Huskey et al. 1962] Harry D. Huskey, Granino A. Korn, *Computer Handbook*, McGraw-Hill Book Company, Inc., 1962

[Hussey et al.] [Hussey et al.] Joseph L. Hussey, C. D. Finnell, *EASE – The New 1100 Series – a Discussion of Design Theory and Operating Techniques*, Beckman, Berkeley Division, n. d. →

[Hyatt et al. 1968] [Hyatt et al. 1968] Gilbert P. Hyatt, Gene Ohlberg, "Electrically alterable digital differential analyzer", in *AFIPS '68 (Spring) Proceedings of the April 30–May 2, 1968 spring joint computer conference*, pp. 161–169 →

[Hydrocarbon Processing 1959] [Hydrocarbon Processing 1959] Hydrocarbon Processing (Reprint), *Analog Computers Handbook*, Gulf Publishing Company, 1969

[Hyman 1982] [Hyman 1982] Anthony Hyman, *Charles Babbage – Pioneer of the Computer*, Princeton University Press, 1982 →

[IACM 1976] [IACM 1976] N. N., *Fifth International Analogue Computation Meetings*, Lausanne, 28-VIII – 2-IX-1967, Preceedings, Vol. II, Applications

[IAEA 1992] [IAEA 1992] N. N., *INSAG-7 – The Chernobly Accident: Updating of INSAG-1*, International Atomic Energy Agency, safety series No. 75-INSAG-7, 1992 →

[IAEA 2004] [IAEA 2004] N. N., *Use of control room simulators for training of nuclear power plant personnel*, International Atomic Energy Agency, technical document 1411, September 2004 →

[Ifrah 2010] [Ifrah 2010] Georges Ifrah, *Universalgeschichte der Zahlen*, Tolkemitt Verlag bei Zweitausendeins, 2010 →

[Instruments and Automation 1957/1] [Instruments and Automation 1957/1] N. N., "Berkeley opens its new computer facility", in *Instruments and Automation*, February 1957, p. 288 →

[Instruments and Automation 1957/2] [Instruments and Automation 1957/2] N. N., "Bonneville Power Administration

Solves Swing Equations with EASE”, in *Instruments and Automation*, March 1957, p. 498 →

[Instruments and Automation 1957/3] [Instruments and Automation 1957/3] N. N., “New GEDA Power Dispatch Computer”, in *Instruments and Automation*, Vol. 30, February 1957, p. 179 a, b

[Instruments and Automation 1957/4] [Instruments and Automation 1957/4] N. N., “University Research Instrumentation”, in *Instruments and Automation*, June 1957, p. 1120 a, b

[Instruments and Automation 1957/5] [Instruments and Automation 1957/5] N. N., “New Data Handling Centers”, in *Instruments and Automation*, April 1957, p. 608

[Instruments and Automation 1958] [Instruments and Automation 1958] N. N., “Computer Designed Rolling Mill”, in *Instruments and Automation*, Vol. 31, February 1958, p. 283 →

[N. N. 1958/2] [N. N. 1958/2] N. N., “Distillation-Column Dynamic Characteristics”, in *Instruments and Automation*, Vol. 31, August 1958, pp. 1357–1359 →

[Irwin et al. 1960] [Irwin et al. 1960] Samuel N. Irwin, Robert R. Kley, “Analog computer serves as both systems analysis tool and operator training facility for Enrico Fermi Atomic Power Plant”, in *IRE-AIEE-ACM ’60 (Western): Papers presented at the May 3–5, 1960, western joint IRE-AIEE-ACM computer conference*, May 1960, pp. 301–313 →

[isec/1] [isec/1] N. N., *Instruction and Operating Manual for your isec 250 Model C and D*, isec, Princeton, New Jersey, n. d. a, b

[isec/2] [isec/2] N. N., *Programming the ISEC 250*, isec, Princeton New Jersey, n. d. →

[Izhikevich 2007] [Izhikevich 2007] Eugene M. Izhikevich, *Dynamical systems in neuroscience: the geometry of excitability and bursting*, MIT Press, 2007 a, b

[Jackson 1960] [Jackson 1960] Albert S. Jackson, *Analog Computation*, McGraw-Hill Book Company, Inc., 1960 a, b, c, d, e, f, g, h, i, j, k, l, m, n, o

[James et al. 1971] [James et al. 1971] M. L. James, G. M. Smith, J. C. Wolford, *Analog Computer Simulation of Engineering Systems*, Intext Educational Publishers, 1971, 3rd edition a, b, c, d, e

[Jamshidi 1976] [Jamshidi 1976] M. Jamshidi, "Optimization of some Dynamic Industrial Control Processes by Analog Simulation", in *Trans. IMACS*, Vol. XVIII, No. 2, April 1976, pp. 93–100 →

[Janac 1976] [Janac 1976] Karel Janac, *Control of Large Power Systems Based on Situation Recognition and High Speed Simulation*, presented at 9th-Hawaii International Conference on System Sciences, January 6–8, 1976, University of Hawaii, author's archive →

[Janssen et al. 1955] [Janssen et al. 1955] J. M. L. Janssen, L. Ensing, "The Electro-Analogue, an Apparatus for Studying Regulating Systems", in [Paynter ed. 1955, pp. 147–161] →

[Jenkins et al. 2003] [Jenkins et al. 2003] Dennis R. Jenkins, Tony R. Landis, *Hypersonic – The Story of the North American X-15*, Specialty Press, 2003 →

[Jennings 1962] [Jennings 1962] Jack R. Jennings, *An Analog Eigenvalue Technique*, doctoral thesis, University of Michigan, 1962 →

[Jezierski 2000] [Jezierski 2000] Dieter von Jezierski, *Slide Rules – A Journey Through Three Centuries*, Astragal Press, Mendham, New Jersey, 2000 a, b

[Jia et al. 2019] [Jia et al. 2019] Xiaotao Jia, You Wang, Zhe Huang, Yue Zhang, Jianlei Yang, Yuanzhuo Qu, Bruce F. Cockburn, Jie Han, Weisheng Zhao, “Spintronic Solutions for Stochastic Computing”, in [Gross et al. 2019, pp. 165–183] →

[Jönck et al. 2003] [Jönck et al. 2003] Uwe Jönck, Florian Prill, “Das Lorenz-System”, in *Seminar über gewöhnliche Differentialgleichungen*, Universität Hamburg, 2003, → https://www.math.uni-hamburg.de/home/lauterbach/scripts/seminar03/prill.pdf, retrieved 18.05.2022 →

[Johnson 1915] [Johnson 1915] R. D. Johnson, “The Differential Surge Tank”, in *Transactions of the American Society of Civil Engineers*, Vol. LXXVIII, 1915 →

[Johnson et al. 1961] [Johnson et al. 1961] R. S. Johnson, F. R. Williamson, R. D. Loftin, *Development of New Methods and Applications of Analog Computation*, George C. Marshall Space Flight Center, Huntsville, Alabama, GIT/EES Report A588/P1, 12 January 1962 →

[Johnson 1962] [Johnson 1962] E. Calvin Johnson, “Computers and Control”, in [Huskey et al. 1962, pp. 21-62 ff.] →

[Johnson 1963] [Johnson 1963] Clarence L. Johnson, *Analog Computer Techniques*, McGraw-Hill Book Company, Inc., Second Edition, 1963 a, b

[Jones et al. 1957] [Jones et al. 1957] J. C. Jones, D. Readshaw, “LACE (The Luton Analogue Computing Engine (Part 2)”, in *Electronic Engineering*, Vol. 29, August 1957, pp. 380–385 →

[Jones 1961] [Jones 1961] E. D. Jones, *Power Excursion in a Hanford Reactor Due to a Positive Reactivity Ramp*, HW-71119, Hanford Atomic Products Corporation, Richland, Washington, September 20, 1961 a, b

[Jung 2006] [Jung 2006] Walt Jung, *Op Amp History*, in [Analog Devices 2006], pp. 765–829 a, b, c, d, e, f, g

[Juslin 1981] [Juslin 1981] Kaj Juslin, *Hybrid Computer Model for Synchronous and Asynchronous Motor Interaction Studies*, Scandinavian Simulation Society annual meeting, 18–20th May, 1981 at Royal Institute of Technology, Stockholm →

[Just et al. 1962] [Just et al. 1962] L. C. Just, C. N. Kelber, N. F. Morehouse, *An Analog Computer Model of a Multiple-Region Reactor*, Argonne National Laboratory, ANL-6482, February 1962 →

[Kaczala 1986] [Kaczala 1986] F. Kaczala, "SIMSTAR Status Report", in [EAI 1986, p. 9] →

[Kahne 1968] [Kahne 1968] Stephen J. Kahne, "Sensitivity-Function Calculation in Linear Systems Using Time-Shared Analog Integration", in *IEEE Transactions on Computers*, Vol. C-17, No. 4, April, 1968, pp. 375–279 →

[Kalinin et al. 2023] [Kalinin et al. 2023] Kirill P. Kalinin, George Mourgias-Alexandris, Hitesh Ballani, Natlia G. Berloff, James Clegg, Daniel Cletheroe, Christos Gkantsidis, Istvan Haller, Vassily Lyutsarev, Francesca Parmigiani, Lucinda Pickup, Antony Rowstron, *Analog Iterative Machine (AIM): using light to solve quadratic optimization problems with mixed variables*, →https://doi.org/10.48550/arXiv.2304.12594, retrieved 18.02.2025 →

[Kalinin et al. 2025] [Kalinin et al. 2025] Kirill P. Kalinin, Jannes Gladrow, Jiaqi Chu, James H. Clegg, Daniel Cletheroa, Douglas J. Kelly, Babak Rahmani, Grace Brennan, Buru Canakci, Fabian Falck, Michael Hansen, Jim Kleewein, Heiner Kremer, Greo O'Shea, Lucinda Pickup, Saravn Rajmohan, Ant Rowstron, Victor Ruhle, Lee Braine, Shirang Khedeka, Natalie G. Berloff, Christos Gkantsidis, Francesca Parmigiani, Hitesh Ballani, "Analog optical computer for AI inference and

combinatorial optimization", in *Natur*, 03.09.2025, →https://www.nature.com/articles/s41586-025-09430-z, retrieved 08.09.2025 →

[Kaplan] [Kaplan] Paul Kaplan, *A Mathematical Model for Assault Boat Motions in Waves*, Oceanics Inc., Plainview, New York, n. d., author's archive →

[Karplus 1958] [Karplus 1958] Walter J. Karplus, *Analog Simulation – Solution of Field Problems*, McGraw-Hill Book Company, Inc., 1958 →

[Karplus et al. 1958] [Karplus et al. 1958] Walter J. Karplus, Walter W. Soroka, *Analog Methods – Computation and Simulation*, McGraw-Hill Book Company, Inc., 1958 a, b, c, d, e

[Karplus et al. 1972] [Karplus et al. 1972] Walter J. Karplus, Richard A. Russell, "Increasing Digital Computer Efficiency with the Aid of Error-Correcting Analog Subroutines", in *IEEE Transactions on Computers*, Vol. C-20, No. 8, August 1972, pp. 831–837 a, b

[Kasper 1955] [Kasper 1955] Joseph Emil Kasper, *Construction and application of a mechanical differential analyzer*, Thesis, State University of Iowa, February 1955 →

[Kastner 1968/1] [Kastner 1968/1] Sigismund Kastner, "Analyse von natürlichem Modellseegang", in *Schriftenreihe Schiffbau*, Technische Universtität Hamburg-Harburg, 223, 1968 →

[Kastner 1968/2] [Kastner 1968/2] Sigismund Kastner, *Das Kentern von Schiffen in unregelmässiger längs laufender See*, Technische Universtität Hamburg-Harburg, 249, 1968 →

[Kathel et al. 1973] [Kathel et al. 1973] B. L. Kathel, "The Radioisotope Renogram and its Analogue Computer Simulation", in *Progress in pediatric surgery*, 5, 173, pp. 61–112 →

[Kaufmann et al. 1955] [Kaufmann et al. 1955] M. G. Kaufmann, R. E. Gardner, “An Electronic Slide Rule”, in *Radio & Television News*, December 1955, pp 58 ff. →

[Kausel et al. 1985] [Kausel et al. 1985] W. Kausel, H. Stallbaumer, H. Silberbauer, “Simulation eines Kraftwerksmodells am Hybridrechner”, in [TU Wien 1985, p. 46 ff.] →

[Kayraklioglu 2020] [Kayraklioglu 2020] Engin Kayraklioglu, Jeff Anderson, Hamid Reza Imani, “Software Stack for an Analog Mesh Computer: The Case of a Nanophotonic PDE Accelerator”, in *Proceedings of the 17th ACM Interational Conference on Computing Frontiers*, May 2020, pp. 241–244 →

[Keefe 1955] [Keefe 1955] T. J. Keefe, “Truer Input Functions in Electronic Analog Computers”, in *Computers and Automation*, 1955, Vol. 4, No. 1, pp. 27–30 →

[Keleş et al. 2009] [Keleş et al. 2009] Sinem Keleş, H. Hakan Kuntman, “Four Quadrant FGMOS Multiplier”, Conference: *Electrical and Electronics Engineering (ELECO)*, 2009, pp. II-45–II-48 →

[Kella 1967] [Kella 1967] J. Kella, “A Note on the Accuracy of Digital Differential Analyzers”, in *IEEE Transactions on Electronic Computers*, Vol. EC-16, No. 2, April 1967, p. 230 →

[Kella et al. 1968] [Kella et al. 1968] J. Kella, A. Shani, “On the Reversibility of Computations in a Digital Differential Analyzer”, in *IEEE Transactions on Computers*, Vol. C-17, No. 3, 1968, pp. 283–284 →

[Kenneally et al. 1966] [Kenneally et al. 1966] W. J. Kenneally, E. E. L. Mitchell, I. Hay, G. Bolton, “Hybrid simulation of a helicopter”, in *AFIPS ’66 (Spring): Proceedings of the April 26–28, 1966, Spring joint computer conference*, April 1966, pp. 347-354 →

[Kennedy 1962] [Kennedy 1962] Jerome D. Kennedy Sr., “Representation of Time Delays”, in [Huskey et al. 1962, pp. 6-3–6-

16] a, b, c

[Kerr 1978] [Kerr 1978] C. N. Kerr, *Use of the Analog/Hybrid Computer in Boundary Layer and Convection Studies*, American Society for Engineering Education, 86th Annual Conference, University of British Columbia, June 19–22, 1978 →

[Kerr 1980] [Kerr 1980] C. N. Kerr, "Analog Solution of Free Convection Mass Transfer From Downward-Facing Horizontal Plates", in *Int. J. Heat Mass Transfer*, Vol. 23, 1980, pp. 247–249 →

[Kettel 1960] [Kettel 1960] E. Kettel, "Die Anwendungsmöglichkeiten der Analogrechentechnik in Meßtechnik und Nachrichtenverarbeitung", in *Telefunken Zeitung*, Vol. 33 (September 1960), No. 129, pp. 164–171 a, b, c

[Kettel et al. 1967] [Kettel et al. 1967] E. Kettel, A. Kley, H. Mangold, R. Maunz, G. Meyer-Brötz, H. Ohnsorge, J. Schürmann, "Der Einsatz elektronischer Rechner für Aufgaben der nachrichtentechnischen Systemforschung", in *Telefunken Zeitung*, Vol. 40 (1967), No. 1/2, pp. 3–9 →

[Khan et al. 2024] [Khan et al. 2024] Asif Ali Khan, Joaão C. de Lima, Hamid Farzaneh, Jeronimo Castrillon, *The Landscale of Compute-near-memory and Compute-in-memory: A Reserch and Commercial Overview*, →https://doi.org/10.48550/arXiv.2401.14428, retrieved 02.03.2025 →

[Kheireldin 1967] [Kheireldin 1967] Abdel-Fattah Kheireldin, "Simulation of a Three Phase Self-Excited Asynchronous Generator on an Analog Computer Considering Iron Saturation", in [IACM 1976, pp. 784–795] →

[Khwaja 1968] [Khwaja 1968] I. Khwaja, "Large deflexions of cantilevers by the electronic analogue computer", in *International Journal of Mechanical Sciences*, Vol. 10, Issue 11, November 1968, pp. 881–885 →

[Khwaja 1971] [Khwaja 1971] I. Khwaja, “Large deflexions of thin elastic bars by the electronic analogue computer”, in *International Journal of Mechanical Sciences*, Vol. 13, Issue 4, April 1971, pp. 323–330 →

[Kidd et al. 1961] [Kidd et al. 1961] E. A. Kidd, G. Bull, R. P. Harper Jr., “In-Flight-Simulation – Theory and Application”, AGARD Report 368, April 1961 →

[Killeen 1953] [Killeen 1953] John Killeen, *Application and Operation of the UCRL Differential Analyzer*, University of California, Radiation Laboratory, UCRL-2239, Contract No. w-7405-eng-48 →

[Kingery 1960] [Kingery 1960] Donald Samuel Kingery, *Introduction to Mine Ventilating Principles and Practices*, Bulletin 589, Bureau of Mines, United States Government Printing Office, Washington, 1960 →

[Kinzel et al. 1962/1] [Kinzel et al. 1962/1] B. Kinzel, L. Sengewitz, “Radizierender Verstärker, insbesondere zur Verarbeitung von Einspritzwerten bei Dieselmotoren”, in *Elektronische Rundschau*, Januar 1962, Vol. 16, No. 1, pp. 21–23 →

[Kinzel et al. 1962/2] [Kinzel et al. 1962/2] B. Kinzel, L. Sengewitz, “Erweiterter radizierender Verstärker”, in *Elektronische Rundschau*, Mai 1962, Vol. 16, No. 5, p. 223 →

[Klein et al. 1957] [Klein et al. 1957] Martin L. Klein, Frank K. Williams, Harry C. Morgan, “Digital Differential Analyzers”, in *Instruments and Automation*, June 1957, pp. 1105–1109 a, b

[Klein 1965] [Klein 1965] Manfred Klein, *Eine Hyperbelfeldröhre für die Multiplikation in elektronischen Analogrechnern*, Dissertation, Technische Universität Berlin, 1965 →

[Kleinert 1978] [Kleinert 1978] Wolfgang Kleinert, “Arbeitsweise und Verwendung der REP-OP Timer an der Hybridrechenanlage”, in [TU Wien 1978/2, pp. 15 ff.] →

[Kleinert 1979/1] [Kleinert 1979/1] Wolfgang Kleinert,“Gasfluss durch ein langes Rohr”, in [TU Wien 1979/1, pp. 31 ff.] →

[Kleinert 1979/2] [Kleinert 1979/2] Wolfgang Kleinert, “Analoges Time-Sharing nichtlinearer Elemente”, in [TU Wien 1979/2, pp. 14 ff.] a, b

[Kleinert 1985] [Kleinert 1985] Wolfgang Kleinert, “SIMSTAR Leistungstests”, in [TU Wien 1985, pp. 18 ff] →

[Kleinwächter 1938] [Kleinwächter 1938] Hans Kleinwächter, “Anwendung der Braunschen Röhre für die Auflösung von Differentialgleichungen auf elektrischem Wege”, in *Archiv für Elektrotechnik*, 1938, pp. 118–120 →

[Kley et al. 1966] [Kley et al. 1966] A. Kley, E. Heim, “Ein elektronischer Koordinatenwandler”, in *Telefunkenzeitung*, Vol. 39, No. 1, 1966, pp. 60–65 →

[Kline 1993] [Kline 1993] Ronald Kline, “Harold Black and the Negative-Feedback Amplifier”, in *IEEE Control Systems*, August 1993, pp. 82–85 →

[Klipsch 1981] [Klipsch 1981] Paul W. Klipsch, “In Memoriam: Paul G. A. H. Voigt”, in *J. Audio Eng. Soc.*, Vol. 29, No. 4, 1981 April, p. 308 →

[Klittich 1966] [Klittich 1966] Manfred Klittich, “Über die Nachbildung von Getrieben auf dem Analogrechner”, in *Elektronische Rechenanlagen*, 8, Heft 3, 1966, pp. 125–130 →

[Klittich 1974] [Klittich 1974] Manfred Klittich, “Entwurf und Inbetriebnahme des Nachführ- und Positioniersystems für das Radioteleskop Effelsberg”, in *IFAC Symposium*, Düsseldorf, 1974, pp. 301–316 →

[Knorre 1971] [Knorre 1971] Wolfgang A. Knorre, *Analogcomputer in Biologie und Medizin – Einführung in die dynamische Analyse*

biologischer Systeme, VEB Gustav Fischer Verlag Jena, 1971 a, b, c

[Knuth 1998] [Knuth 1998] Donald Ervin Knuth, *The Art of Computer Programming, Volume 2, Seminumerical Algorithms*, 3rd edition, Addison-Wesley, 1998 →

[Kobelt] [Kobelt] D. Kobelt, *Fortgeschrittene Analogrechnertechnik*, Farbwerke Hoechst AG, Frankfurt (Main)-Höchst, n. d. →

[Koenig 1950] [Koenig 1950] Eldo C. Koenig, "An Electric Analogue Computer Using the Photo Cell as a Nonlinear Element", in *AIEE Transactions*, Vol. 69, 1950, pp. 1606–1611 →

[Koenig et al. 1955] [Koenig et al. 1955] Eldo C. Koenig, William C. Schultz, "How to Select Governor Parameters with Analog Computers", in [Paynter ed. 1955, pp. 237–238] →

[Köppel et al. 2021] [Köppel et al. 2021] Sven Köppel, Bernd Ulmann, Lars Heimann, Dirk Killat, "Using analog computers in today's largest computational challenges", in *Advances in Radio Science*, 19, 2021, pp. 105–116, →https://doi.org/10.5194/ars-19-105-2021 a, b

[Köppel et al. 2022] [Köppel et al. 2022] Sven Köppel, Alexandra Krause, Bernd Ulmann, "Analog Computing for Molecular Dynamics", in *International Journal of Unconventional Computing*, Vol. 17, 2022, pp. 259–282 →

[Köppel 2024] [Köppel 2024] Sven Köppel, *LUCIDAC handbook*, →https://anabrid.dev/docs/lucidac-user-docs.pdf, retrieved 13.03.2025 →

[Kohr 1960] [Kohr 1960] Robert H. Kohr, "Real-Time Automobile Ride Simulation", in *IRE-AIEE-ACM '60 (Western): Papers presented at the May 3-5, 1960, western joint IRE-AIEE-ACM computer conference*, May 1960, pp. 285–300 →

[Kopacek et al. 1976] [Kopacek et al. 1976] P. Kopacek, J. Weninger, “Experimentelle Überprüfung eines Identifikationsverfahrens für lineare zeitinvariante Systeme auf dem Hybridrechner”, in [TU Wien 1976/3, pp. 39 ff.] →

[Kopacek et al. 1977] [Kopacek et al. 1977] P. Kopacek, P. Nowotny, “Pseudozufällige Binärsignale und ihre Eignung zur Identifikation linearer Eingröszensysteme”, in [TU Wien 1977/3, pp. 51 ff.] →

[Kopacek et al. 1979] [Kopacek et al. 1979] P. Kopacek, H. Traxler, “Experimentelle Überprüfung eines Parameterschätzverfahrens für Wasserkraftmaschinen und Pumpen”, in [TU Wien 1979/1, pp. 37 ff.] →

[Kopacek 1983] [Kopacek 1983] P. Kopacek, “Testing Various Identification Algorithms for Control Systems with Stochastically Varying Parameters by a Hybrid Computer”, in *10th IMACS World Congress on System Simulation and Scientific Computation*, 1983, pp. 69–71 →

[Kolms et al. 2020] [Kolms et al. 2020] Thore Kolms, Andreas Waldner, Christine Lang, Philipp Grothe, Jan Haase, “Analog implementation of arithmetic operations on real memristors”, in *SCOPES ’20: Proceedings of the 23th Interational Workshop on Software and Compilers for Embedded Systems*, May 2020, pp. 54–57 →

[Koppe et al. 1971] [Koppe et al. 1971] Ir. R. Koppe, Ir. P. Euser, “Analoge simulatie van warmteen waterdamp-transport in klimaatkamers voor planten en in de daarbij toegepaste luchtbehandelingsinstallaties”, in *TNO-nieuws*, 1971, 26, pp. 497–513 →

[Korn et al. 1954] [Korn et al. 1954] Granino A. Korn, Theresa M. Korn, “Relay Time-Division Multiplier”, in *The Review of Scientific*

Instruments, Vol. 25, No. 10, October 1954, pp. 977–982 a, b

[Korn et al. 1956] [Korn et al. 1956] Granino A. Korn, Theresa M. Korn, *Electronic Analog Computers (D-c Analog Computers)*, McGraw-Hill Book Company, Inc., 1956 a, b, c, d, e, f, g, h, i

[Korn 1962] [Korn 1962] Granino A. Korn, "Electronic Function Generators, Switching Circuits and Random-Noise Generators", in [Huskey et al. 1962, pp. 3-62– 3-84] a, b, c, d, e, f, g, h

[Korn et al. 1964] [Korn et al. 1964] Granino A. Korn, Theresa M. Korn, *Electronic Analog and Hybrid Computers*, McGraw-Hill Book Company, 1964 a, b, c, d, e

[Korn 1966] [Korn 1966] Granino A. Korn, *Random-Process Simulation and Measurements*, McGraw-Hill Book Company, 1966 a, b

[Korn et al. 1970] [Korn et al. 1970] Granino A. Korn, H. Kosako, "A Proposed Hybrid-Computer Method for Functional Optimization", in *IEEE Transactions on Computers*, Vol. C-19, No. 2, February 1970, pp. 149–153 →

[Korn et al. 1983] [Korn et al. 1983] Granino A. Korn, John V. Wait, *Digitale Simulation kontinuierlicher Systeme*, R. Oldenbourg Verlag München Wien, 1983 a, b, c, d, e

[Korn 2005] [Korn 2005] Granino A. Korn, "Continuous-System Simulation and Analog Computers – From op-amp design to aerospace applications", in *IEEE Control Systems Magazine*, June 2005, pp. 44–51 →

[Kovach 1952] [Kovach 1952] Ladis D. Kovach, "Solution of Difference Equations", in [Huskey et al. 1962, pp. 6-52–6-56] →

[Kovach et al. 1960] [Kovach et al. 1960] K. D. Kovach, W. Comley, "A New, Solid-State, Nonlinear Analog Component", in *IRE*

Transactions on Electronic Computers, December 1960, pp. 496–503 →

[Kovach et al. 1962] [Kovach et al. 1962] Ladis D. Kovach, H. F. Meissinger, “Solution of Algebraic Equations, Linear Programming, and Parameter Optimization”, in [Huskey et al. 1962, pp. 5-133–5-154] a, b

[Kovach 1967] [Kovach 1967] Ladis D. Kovach, “Analog & hybrid computers in education: a panel session”, in Spring Joint Computer Conference, 1967, pp. 487–489 →

[Kovacs 1968] [Kovacs 1968] Joseph J. Kovacs, “Optimization of a Nuclear Power Plant by Hybrid Computer”, in *Computers and Automation*, 1968, Vol. 17, No. 7, pp. 26–31 →

[Kraft et al. 2002] [Kraft et al. 2002] Chris Kraft, James L. Schefter, *Flight – My Life in Mission Control*, First Plume Printing, March 2002 →

[Krämer 1989] [Krämer 1989] K. Krämer, “Das Modell. Antoni Gaudis Hängemodell und seine Rekonstruktion. Neue Erkenntnisse zum Entwurf für die Kirche der Colonia Güell”, in *Mitteilungen des Instituts für leichte Flächentragwerke*, Universität Stuttgart, Nr. 34, 1989 →

[Kramer] [Kramer] H. Kramer, *Parameteroptimierung mit einem hybriden Analogrechner an einem Beispiel aus der chemischen Reaktionskinetik*, AEG-Telefunken, AFA 003 0570, n. d. →

[Kramer 1968] [Kramer 1968] H. Kramer, “Optimierung eines Regelkreises mit Tischanalogrechner und Digitalzusatz”, in *elektronische datenverarbeitung*, 1968, 6, pp. 293–297 →

[Krause et al. 1963] [Krause et al. 1963] Walter P. Krause, Frank L. Vinz, *Study of Large Angle Attitude Maneuvers of Bodies in Space*, George C. Marshall Space Flight Center, Huntsville, Alabama, Report IN-M-COMP-S-63-3, April 9, 1963 →

[Krause 1964] [Krause 1964] Walter P. Krause, *Dynamics of a Rotating Space Station*, Research & Development Applications Division, Computation Laboratory, George C. Marshall Space Flight Center, Report R-COMP-RS-64-1, February 1964 a, b

[Krause 1970] [Krause 1970] Paul C. Krause, “Applications of analog and hybrid computers in electric power research”, in *SIMULATION*, August 1970, pp. 73–79 →

[Krause 1971] [Krause 1971] Paul C. Krause, *Hybrid Computation Techniques Applied to Power Systems Simulation*, Purdue University, School of Electrical Engineering, November, 1971 →

[Krause 1974] [Krause 1974] Paul C. Krause, “Applications of Analog and Hybrid Computation in Electric Power System Analysis”, in *Proceedings of the IEEE*, Vol. 62, No. 7, July 1974, pp. 994–1009 →

[Krause et al. 1977] [Krause et al. 1977] P. C. Krause, W. C. Hollopeter, D. M. Triezenberg, “Sharp Torques During Out-Of-Phase Synchronization”, in *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-96, No. 4, July/August 1977, pp. 1318–1323 →

[Krause 2006] [Krause 2006] Christine Krause, *Die Entwicklung der Analogrechentechnik in Thüringen und Sachsen*, 3. Greifswalder Symposium zur Entwicklung der Rechentechnik und 12. Internationales Treffen für Rechenschieber- und Rechenmaschinensammler, 2006 →

[Kregeloh 1956] [Kregeloh 1956] H. Kregeloh, “Analogrechner und ihre Anwendung auf ein volkswirtschaftliches Modell”, in *Mathematical Methods of Operations Research*, Springer-Verlag, Vol. 1, No. 1, Dezember 1956, pp. 97–106 →

[Kroll et al. 1966] [Kroll et al. 1966] John Kroll, Rudy H. Arendt, Francis E. Pritchard, *Development of a General Purpose Airborne Simulator*, NASA contractor report, CR-641, 1966 a, b

[Kron 1945/1] [Kron 1945/1] Gabriel Kron, “Electric Circuit Models of the Schrödinger Equation”, in *Physical Review*, Vol. 67, No. 1/2, January 1 and 15, 1945, pp. 39–43 a, b

[Kron 1945/2] [Kron 1945/2] Gabriel Kron, “Numerical Solution of Ordinary and Partial Differential Equations by Means of Equivalent Circuits”, in *Journal of Applied Physics*, 16, 1945, pp. 172–186 →

[Krug 2006] [Krug 2006] Wilfried Krug, “Entwicklung und Anwendung der Analog-, Hybrid- und digitalen Rechentechnik zur Modellierung, Simulation und Optimierung technischer Prozesse und Systeme in der DDR”, in *Informatik in der DDR – eine Bilanz*, 2006, pp. 556–567 →

[Kubelka et al. 1931] [Kubelka et al. 1931] Paul Kubelka, Franz Munk, “Ein Beitrag zur Optik der Farbanstriche”, in *Zeitschrift für technische Physik*, 12, 1931, pp. 593–601 →

[Kubicek et al. 1979] [Kubicek et al. 1979] C. P. Kubicek, Irmgard Husinsky, D. Solar, “Analoge Simulation von Stoffwechselabläufen an Hand von Stoffwechselfluxgrößen”, in [TU Wien 1979/1, pp. 43 ff.] →

[Kuehn 2015] [Kuehn 2015] Christian Kuehn, *Multiple Time Scale Dynamics*, Springer, 2015 →

[Künkel 1961] [Künkel 1961] H. Künkel, “Beitrag zu einer regeltheoretischen Analyse der Pupillenreflexdynamik”, in *Kybernetik*, 1, 1961, pp. 69–75 →

[Kuhn 1996] [Kuhn 1996] Thomas Samuel Kuhn, *The Structure of Scientific Revolutions*, Universty of Chicago Press, 1996 →

[Laage et al. 1995] [Laage et al. 1995] Linneas W. Laage, Rudolf E. Greuer, William H. Pomroy, *MFIRE Users Manual*, Version 2.20, August 1995, U. S. Bureau of Mines, Twin Cities Research Center, Minneapolis, MN a, b

[Lanchester 1908] [Lanchester 1908] Frederick William Lanchester, *Aerial Flight: Aerodonetics*, London, Constable, 1908 →

[Land] [Land] Bruce Land, *DDA on FPGA – a modern Analog Computer*, →https://instruct1.cit.cornell.edu/Courses/ece576/DDA/index.htm, n. d., retrieved 03.03.2013 →

[Landauer 1974] [Landauer 1974] Paul Landauer, "Personal Rapid Transit (PRT) System Design by Hybrid Computation", in *EAI Scientific Computation Report*, No. SCR 74-17, November 11, 1974 →

[Landauer 1975] [Landauer 1975] Paul Landauer, "Non-Destructive Destructive Testing", in *Industrial Research*, March 1975 a, b

[Landauer 1983] [Landauer 1983] Paul Landauer, "SIMSTAR – an attached multiprocessor for dynamic system engineering", in *Informatik Fachberichte: First European Simulation Congress ESC 83*, Springer-Verlag, 1983 →

[Landauer 1986/1] [Landauer 1986/1] Paul Landauer, "Family of Digital Arithmetic Processors in SIMSTAR", in [EAI 1986, p. 11] →

[Landauer 1986/2] [Landauer 1986/2] Paul Landauer, "SIMSTAR Nonlinear Oscillator", in [EAI 1986, pp. 17 f.] →

[Landauer 1986/3] [Landauer 1986/3] Paul Landauer, "SIMSTAR Simulation of a Satellite Attitude Controller", in [EAI 1986, p. 19] →

[Landrot et al. 1967] [Landrot et al. 1967] J.-P. Landrot, D. List, "Simulation de Convertisseurs Thermoïoniques", in [IACM 1976, pp. 1079–1086] →

[Lang 2000] [Lang 2000] George Fox Lang, "Analog was not a Computer Trademark! Why Would Anyone Write About Analog

Computers In Year 2000?", in *Sound and Vibration*, August 2000, pp. 16–24 →

[Lange 2006] [Lange 2006] Thomas H. Lange, *Peenemünde – Analyse einer Technologieentwicklung im Dritten Reich*, Reihe *Technikgeschichte in Einzeldarstellungen*, VDI-Verlag, GmbH, Düsseldorf 2006 a, b, c

[Lange et al. 2020] [Lange et al. 2020] Steffen Lange, Johanna Pohl, Tilman Santarius, "Digitalization and energy consumption. Does ICT reduce energy demand?", in *Ecological Economics*, 176, 2020 →

[Langheld 1979/1] [Langheld 1979/1] Erwin Landheld, "Praxis der stochastischen Rechentechnik", in *Elektronik*, 25, 13. Dezember 1979, pp. 43–48 →

[Langheld 1979/2] [Langheld 1979/2] Erwin Landheld, "Praxis der stochastischen Rechentechnik", in *Elektronik*, 26, 27. Dezember 1979, pp. 39–42 →

[Laning et al. 1956] [Laning et al. 1956] J. Halcombe Laning, Richard H. Battin, *Random Processes in Automatic Control*, Mcgraw-Hill Book Company, Inc., 1956 a, b

[Larrowe 1955] [Larrowe 1955] Vernon L. Larrowe, "Direct Simulation – Bypasses Mathematics, Simplifies Analysis", in [Paynter ed. 1955, p. 127–133] →

[Larrowe 1966] [Larrowe 1966] Vernon L. Larrowe, "Band-Pass Quadrature Filters", in *IEEE Transactions on Electronic Computers*, Vol. EC-15, No. 5, October 1966, pp. 726–731 →

[Leatherwood 1972] [Leatherwood 1972] Jack D. Leatherwood, "Analog Analysis of a Tracked Air-Cushion Vehicle", in *INSTRUMENTS and CONTROL SYSTEMS*, April 1972, pp. 81–86 →

[Lechner 1966] [Lechner 1966] Thomas J. Lechner, *FLUERICS – 19. Low-pressure Amplifier and Analog Computer*, Harry Diamond Laboratories, April 1966 →

[Lehmann 1967] [Lehmann 1967] S. Lehmann, "Einsatz Analoger Rechengeräte in der Entwicklung von Automatisierungssystemen für Nahverkehrs-Schnellbahnen", in [IACM 1976, pp. 761–773] a, b

[Leise 2007] [Leise 2007] Tanya Leise, "As the Planimeter's Wheel Turns: Planimeter Proofs for Calculus Class", in *College Mathematics Journal*, January 2007 →

[Lemaitre et al. 1936/1] [Lemaitre et al. 1936/1] G. Lemaitre, M. S. Vallarta, "On the Geomagnetic Analysis of Cosmic Radiation", in *Physical Review*, Volume 49, 1936, pp. 719–726 →

[Lemaitre et al. 1936/2] [Lemaitre et al. 1936/2] G. Lemaitre, M. S. Vallarta, "On the Allowed Cone of Cosmic Radiation", in *Physical Review*, Volume 50, 1936, pp. 493–504 →

[Levine 1964] [Levine 1964] Leon Levine, *Methods for Solving Engineering Problems Using Analog Computers*, McGraw-Hill Book Company, 1964 a, b

[Lewin 1972] [Lewin 1972] John Ernest Lewin, *Area Measurement*, United States Patent 3652842, Mar. 28, 1972 →

[Lewis 1958] [Lewis 1958] Lloyd G. Lewis, "Simulation of a Solvent Recovery Process", in *Instruments and Automation*, Vol. 31, April 1958, pp. 644–647 →

[Liban 1962/1] [Liban 1962/1] Eric Liban, "The application of finite fourier transforms to analog computer simulations", in *AIEE-IRE '62 (Spring): Proceedings of the May 1–3, 1962, spring joint computer conference*, May 1962, pp. 255-265 →

[Liban 1962/2] [Liban 1962/2] Eric Liban, “The application of finite fourier transforms to analog computer simulations”, in *Proceedings of the spring joint computer conference*, May 1–3, 1962, pp. 255–265 →

[Librascope 1957] [Librascope 1957] N. N., “Ball/Disc Integrator”, in *Instruments and Automation*, April 1957, p. 769 →

[Lieber et al. 1969] [Lieber et al. 1969] R. E. Lieber, T. R. Herndon, “Analog Simulation Spells Safe Startups”, in [Hydrocarbon Processing 1959, pp. 78–82] a, b

[Lienhard 1969] [Lienhard 1969] K. Lienhard, “Planung und Berechnung von Rohrnetzen”, in *Gas – Wasser – Abwasser*, 49. Jahrgang, 1969, Nr. 9 →

[Light et al. 1966] [Light et al. 1966] L. Light, J. Badger, D. Barnes, “An Automatic Acoustic Ray Tracing Computer”, in *IEEE Transactions on Electronic Computers*, Vol. EC-15, No. 5, October, 1966, pp. 719–725 a, b, c

[Light 1975] [Light 1975] Leon Henry Light, *Apparatus for Integration and Averaging*, United States Patent 3906190, Sept. 16, 1975 →

[Lilamand 1956] [Lilamand 1956] M. Lejet Lilamand, “A Time-Division Multiplier” in *IRE Transactions on Electronic Computers*, March 1956, pp. 26–34 →

[Linear Technology] [Linear Technology] Linear Technology, *LTC6943 – Micropower, Dual Precision Instrumentation Switched Capacitor Building Block*, Linear Technology Corporation, n. d. →

[Lippmann et al. 1961] [Lippmann et al. 1961] H. Lippmann, O. Mahrenholtz, W. Johnson, “Thin Heavy Elastic Strips at Large Deflexions”, in International Journal of Mechanical Sciences, Vol. 2, Issue 4,. April 1961, pp. 294–310 →

[Little 1966] [Little 1966] Warren D. Little, “Hybrid computer solutions of partial differential equations by Monte Carlo methods”, in *AFIPS ’66 (Fall): Proceedings of the November 7–10, 1966, fall joint computer conference*, November 1966, pp. 181–190 →

[Litton 1963] [Litton 1963] N. N., *Digital Differential Analyzer*, Technical Documentation Report ASD-TDR-63-158, June 1963 →

[Lockheed 1961] [Lockheed 1961] Lockheed Missiles and Space Division, “Simulating both Digital and Analog in Studies of the Polaris’ Missile Guidance”, in *Computers and Automation*, 1961, Vol. 10, No. 8, p. 16B →

[Lorenz 1963] [Lorenz 1963] Edward Norton Lorenz, “Deterministic Nonperiodic Flow”, in *Journal of the Atmospheric Sciences*, Vol. 20, March 1963, pp. 130–141 →

[Lott 2021] [Lott 2021] Chris Lott, *The Modding, Restoration, and Demise of a $3M Analog Computer*, →https://hackaday.com/2021/02/09/the-modding-restoration-and-demise-of-a-3m-analog-computer/, retrieved 28.06.2025 →

[Lotz 1969] [Lotz 1969] Hermann Lotz, “Einsatz des Analogrechners in der Regelungs- und Steuerungstechnik”, in *Steuerungstechnik*, 2 (1969) 11, pp. 430–435 →

[Lotz 1970/1] [Lotz 1970/1] Hermann Lotz, “Programmiertes Spielen auf einem Hybriden Analogrechner: Billardsimulator”, in *Datenverarbeitung*, AEG-TELEFUNKEN, 1970, 2, pp. 64–67 →

[Lotz 1970/2] [Lotz 1970/2] Hermann Lotz, “Fertigungskontrolle mit Hilfe analoger Meßwertverarbeitung bei der Rundlaufabweichung bei Kraftfahrzeugrädern”, in *Datenverarbeitung*, AEG-TELEFUNKEN, 1970, 2, pp. 68–71 →

[Lotz et al.] [Lotz et al.] Hermann Lotz, G. Ramsauer, *Hybride Simulation einer direkten digitalen Regelung auf dem Hybriden*

Rechnersystem HRS 860 am Beispiel eines dampfgekühlten schnellen Reaktors, AEG-TELEFUNKEN, Datenverarbeitung, n. d. →

[Lovell et al. 1946] [Lovell et al. 1946] Clarence A. Lovell, David B. Parkinson, Bruce T. Weber, *Electrical Computing System*, United States Patent 2404387, July 23, 1946 a, b, c, d

[Loveman 1962] [Loveman 1962] Bernard D. Loveman, "Computer Servomechanisms and Servo Resolvers", in [Huskey et al. 1962, pp. 3-1–3-40] a, b, c

[Lowe] [Lowe] William Lowe, *NUC Simulation Facility*, Naval Undersea Center, San Diego, California, n. d., author's archive →

[Ludwig 1966] [Ludwig 1966] R. Ludwig, *Stability Research on Parachutes using Digital and Analog Computers*, National Aeronautics and Space Administration, Washington, D. C., November 1966 →

[Ludwig et al. 1974] [Ludwig et al. 1974] Manfred Ludwig, Klaus Kaplick, *Elektronische Analogrechner und Prozeßrechnereinsatz*, Reihe *Programmierung und Nutzung von Rechenanlagen, Teil 6*, Verlag "Die Wirtschaft", Berlin, 1974 →

[Lukes 1967] [Lukes 1967] Jaroslav H. Lukes, "Oscillographic Examination of the Operation of Function Generators", in *IEEE Transactions on Electronic Computers*, Vol. EC-16, No. 2, April 1967, pp. 133–139 →

[Lukyanov 1939] [Lukyanov 1939] Vladimir S. Lukyanov, *Hydraulic Apparatur for Engineering Computations*, Translation of Russian literatures, Corps of Engineers, U. S. Army, Permafrost Investigations, Arctiv Construction and Frost Effects Laboratory, June 1955 →

[Lunderstädt et al. 1981] [Lunderstädt et al. 1981] R. Lunderstädt, W. Menssen, *Regelmechanismus der menschlichen Pupille – Stabilität und Simulation*, Dornier System GmbH, 1981 →

[Lyon 1950] [Lyon 1950] G. Lyon, “Some Experience with a British A.C. Network Analyser”, in *Proceedings of the IEE – Part II: Power Engineering*, Volume 97, Issue 60, pp.697–714 a, b

[Macdonald 1955] [Macdonald 1955] Neil Macdonald, “First Supersonic Flight Simulator”, in *Computers and Automation*, 1955, Vol. 4, No. 5, p. 40 →

[Macduff et al. 1958] [Macduff et al. 1958] John N. Macduff, John R. Curreri, *Vibration Control*, McGraw-Hill Book Company, Inc., 1958 a, b

[MacKay et al. 1962] [MacKay et al. 1962] Donald M. MacKay, Michael E. Fisher, *Analogue Computing at Ultra-High Speed*, John Wiley & Sons Inc., 1962 a, b, c, d, e, f, g, h, i

[MacNeal et al. 1955] [MacNeal et al. 1955] Richard H. MacNeal, G. D. McCann, “Ideal Transformers in the Synthesis of Analog Computer Circuits”, in *Western Joint Computer Conference*, 1955, pp. 16–23 →

[MacNeal 2002] [MacNeal 2002] Richard H. MacNeal, *Richard H. MacNeal (b. 1923) Interviewed by Shirley K. Cohen*, in *Archives California Institute of Technology*, Pasadena, California, January 23, 2002, →https://oralhistories.library.caltech.edu/262/1/MacNeal%20OHO.pdf, retrieved 28.02.2021 →

[Macnee 1948] [Macnee 1948] A. B. Macnee, *An Electronic Differential Analyzer*, Technical Report No. 90, December 16, 1948, Research Laboratory of Electronics, Massachusetts Institute of Technology a, b, c, d

[Macnee 1949] [Macnee 1949] A. B. Macnee, “An Electronic Differential Analyzer”, in *Proceedings of the I.R.E. – Waves and Electrons Section*, November 1949, pp. 1315–1324 →

[Macnee 1953] [Macnee 1953] A. B. Macnee, “A High Speed Product Integrator”, in *The Review of Scientific Instruments*, Volume 24, Number 3, March 1953, pp. 207–211 →

[Mahrenholtz 1968] [Mahrenholtz 1968] O. Mahrenholtz, *Analogrechnen in Maschinenbau und Mechanik*, Bibliographisches Institut, Mannheim/Zürich, 1968 a, b, c, d, e, f, g

[Maley 2023] [Maley 2023] Corey J. Maley, “Analogue Computation and Representation”, in *The British Journal for the Philosophy of Science*, Volume 74, Number 3, September 2023, pp. 739–769 →

[Malstrom et al. 1977] [Malstrom et al. 1977] Carl W. Malstrom, Rainer Heller, Mohammad S. Khan, “Hybrid Computation – an Advanced Computation Tool for Simulating the Nonlinear Dynamic Response of Railroad Vehicles”, Pre-Publication Copy of Submission to the Post Conference Proceedings, *Advanced Techniques in Track/Train Dynamics and Design Conference*, Chicago, Illinois, September 27 and 28, 1977 →

[Manohar 2015] [Manohar 2015] Rajit Manohar, “Comparing Stochastic and Deterministic Computing”, in *IEEE Computer Architecture Letters*, Vol. 14, No. 2, July-Dec. 2015, pp. 119–122 a, b

[Manske 1968] [Manske 1968] R. A. Manske, “Computer Simulation of Narrowband Systems”, in *IEEE Transactions on Computers*, Vol. C-17, No. 4, April, 1968, pp. 301–308 →

[Markson 1958] [Markson 1958] A. A. Markson, “Introduction to Reactor Physics”, in *Instruments and Automation*, Vol. 31, April 1958, pp. 616–623 →

[Marquitz et al. 1968] [Marquitz et al. 1968] W. T. Marquitz, Y. Tokad, “On Improving the Analog Computer Solutions of Linear Systems”, in *IEEE Transactions on Computers*, Vol. C-17, No. 3, March, 1968, pp. 268–270 →

[Marsh 1961] [Marsh 1961] Charles J. Marsh, “Do it Yourself Analog Computer Components”, in *Computers and Automation*, 1961, Vol. 10, No. 3, p. 12B →

[Marsh 2024] [Marsh 2024] Allison Marsh, “Lord Kelvin and His Analog Computer – This tide-predicting machine was one of many advances he made to maritime tech”, in *IEEE Spectrum*, 02.06.2024, →https://spectrum.ieee.org/tide-predictions, retrieved 23.02.2025 →

[Marsocci 1956] [Marsocci 1956] Velio A. Marsocci, “An Error Analysis of Electronic Analog Computers”, in *IRE Transactions on Electronic Computers*, December 1956, pp. 207–212 →

[Martin 1969] [Martin 1969] George J. Martin, “Hybrid Computation in the Engineering College”, in *Engineering Education*, January 1969, pp. 395–400 →

[Martin 1970] [Martin 1970] Donald C. Martin, “Development of analog/hybrid terminals for teaching system dynamics”, in *AFIPS ’70 (Fall): Proceedings of the November 17–19, 1970, fall joint computer conference*, November 1970, pp. 241–249 →

[Martin 1972] [Martin 1972] George J. Martin, “Analog and Hybrid Simulation in Science Education”, in *Educational Technology*, April, 1972, pp. 62–63 →

[Maslo 1974] [Maslo 1974] Ronald M. Maslo, “Dynamic Response of a Ship in Waves”, in EAI Scientific Computation Report, No. 74-14, September 20, 1974 →

[Mason et al. 1967] [Mason et al. 1967] Peter Mason, Philip H. Price, “The Design of an Economical Real Time Simulator of a Two Phase Heat Exchanger”, in [IACM 1976, pp. 1006–1010] →

[Massen 1977] [Massen 1977] Robert Massen, *Stochastische Rechentechnik – Eine Einführung in die Informationsverarbeitung mit zufälligen Pulsfolgen*, Carl Hanser Verlag, 1977 →

[Master et al. 1955] [Master et al. 1955] R. C. Master, R. L. Merrill, B. H. List, “Analogous Systems in Engineering Design”, in [Paynter ed. 1955, p. 134–145] →

[Maxfield 2020] [Maxfield 2020] Max Maxfield, “A Brave New World of Analog Artificial Neural Networks (AANNs)”, in *Electronic Engineering Journal*, →https://www.eejournal.com/article/a-brave-new-world-of-analog-artificial-neural-networks-aanns/, retrieved 30.08.2025 a, b

[Maxfield 2021] [Maxfield 2021] Max Maxfield, “Aspinity’s Awesome Analog Artificial Neural Networks (AANNs)”, in *Electronic Engineering Journal*, →https://www.eejournal.com/article/aspinitys-awesome-analog-artificial-neural-networks-aanns/, retrieved 30.08.2025 →

[MBB] [MBB] N. N., *MBB Simulation*, Firmenschrift Messerschmitt Bölkow Blohm GmbH, Unternehmensbereich Flugzeuge, n. d. →

[McCallum] [McCallum] I. R. McCallum, *Horses for Courses: The Mathematical Modelling Requirements of Maritime Simulators*, n. d., author’s archive →

[McCann 1949] [McCann 1949] G. D. McCann, “The California Institute of Technology Electric Analog Computer”, in *Mathematical Tables and Other Aids to Computation*, Vol. 3, No. 28, Oct. 1949, pp. 501–513 a, b, c, d

[McCann et al. 1949] [McCann et al. 1949] G. D. McCann, C. H. Wilts, B. N. Locanthi, “Application of the California Institute of Technology Electric Analog Computer to Nonlinear Mechanics and Servomechanisms”, in *AIEE Transactions*, Volume 68, 1949, pp. 652–660 →

[McCann 1959] [McCann 1959] G. D. McCann, “Caltech’s Computing Center”, in *Engineering and Science*, February 1959, pp. 24–34 →

[McCarthy 2009] [McCarthy 2009] Jerry McCarthy, *Der Mechanismus von Antikythera*, 15. Internationales Treffen der Rechenschiebersammlung und 4. Symposium zur Entwicklunssg der Rechentechnik, Ernst Moritz Arndt Universität Greifswald, 2009 →

[McDonal 1956] [McDonal 1956] Frank J. McDonal, *Wave Analysis*, United States Patent 2752092, June 26, 1956 →

[McFadden et al. 1958] [McFadden et al. 1958] Norman M. McFadden, Frank A. Pauli, Donovan R. Heinle, *A Flight Study of Longitudinal-Control-System Dynamic Characteristics by the Use of a Variable-Control-System Airplane*, NACA RM A57L10, 1958 →

[McGhee et al. 1970] [McGhee et al. 1970] Robert B. McGhee, Ragnar N. Nilsen, "The Extended Resolution Digital Differential Analyzer: A New Computing Structure for Solving Differential Equations", in *IEEE Transactions on Computers*, Vol. C-19, No. 1, January 1970, pp. 1–9 a, b

[McIlroy 1950] [McIlroy 1950] Malcolm S. McIlroy, "Direct-Reading Electric Analyzer for Pipeline Networks", in *Journal AWWA*, Volume 42, Issue 4, Aprils 1950, pp. 347–366 a, b, c

[McLachlan 1947] [McLachlan 1947] N. W. McLachlan, *Theory and Applications of Mathieu Functions*, Oxford at the Clarendon Press, 1947 →

[McLean et al. 1977] [McLean et al. 1977] L. J. McLean, E. J. Hahn, "Simulation of the Transient Behaviour of a Rigid Rotor in Squeeze Film Supported Journal Bearings", 2nd AINSE Engineering Conference, 1977 →

[McLeod et al. 1957] [McLeod et al. 1957] John H. McLeod, Robert M. Leger, "Combined Analog and Digital Systems – Why, When, and How", in *Instruments and Automation*, June 1957, pp. 1126–1130 a, b, c

[McLeod et al. 1958/1] [McLeod et al. 1958/1] John H. McLeod, Suzette McLeod, “The Simulation Council Newsletter”, in *Instruments and Automation*, Vol. 31, January 1958, pp. 119–124 →

[McLeod et al. 1958/2] [McLeod et al. 1958/2] John H. McLeod, Suzette McLeod, “The Simulation Council Newsletter”, in *Instruments and Automation*, Vol. 31, February 1958, pp. 297–300 →

[McLeod et al. 1958/3] [McLeod et al. 1958/3] John H. McLeod, Suzette McLeod, “The Simulation Council Newsletter”, in *Instruments and Automation*, Vol. 31, March 1958, pp. 487–491 →

[McLeod et al. 1958/4] [McLeod et al. 1958/4] John H. McLeod, Suzette McLeod, “The Simulation Council Newsletter”, in *Instruments and Automation*, Vol. 31, July 1958, pp. 1219–1225 a, b, c

[McLeod et al. 1958/5] [McLeod et al. 1958/5] John H. McLeod, Suzette McLeod, “The Simulation Council Newsletter”, in *Instruments and Automation*, Vol. 31, August 1958, S. 1385–1390 →

[McLeod et al. 1958/6] [McLeod et al. 1958/6] John H. McLeod, Suzette McLeod, “The Simulation Council Newsletter”, in *Instruments and Automation*, Vol. 31, December 1958, pp. 1991–1997 a, b, c, d, e, f, g

[McLeod 1962] [McLeod 1962] John H. McLeod, “Electronic-Analog-Computer Techniques for the Design of Servo Systems”, in [Huskey et al. 1962, pp. 5-35 ff.] →

[Meccano 1934/1] [Meccano 1934/1] N. N., “Meccano Aids Scientific Research”, in *Meccano Magazine*, Vol. XIX, No. 6, June, 1934, p. 441 →

[Meccano 1934/2] [Meccano 1934/2] N. N., “Machine Solves Mathematical Problems – A Wonderful Meccano Mechanism”, in

Meccano Magazine, Vol. XIX, No. 6, June, 1934, pp. 442–444 a, b, c

[Medkeff et al. 1955] [Medkeff et al. 1955] R. J. Medkeff, H. Matthews, “Solving process-control problems by ANALOG COMPUTER”, in [Paynter ed. 1955, pp. 164–166] →

[Meilander 1957] [Meilander 1957] Willard C. Meilander, “A New Method of Verifying Analog Computer Problems and Performances”, in *1957 Western Computer Proceedings*, 1957, pp. 138–142 →

[Meisinger 1978] [Meisinger 1978] Reinhold Meisinger, “Analog Simulation of Magnetically Levitated Vehicles on Flexible Guideways”, in *Simulation of Control-Systems*, I. Troch (ed.), North-Holland Publishing Company, 1978, pp. 207–214 →

[Meissl 1960/1] [Meissl 1960/1] P. Meissl, “Behandlung von Wasserschloßaufgaben mit Hilfe eines elektronischen Analogrechners, Teil 1”, in *mtw – Zeitschrift für moderne Rechentechnik und Automation*, 1/60, pp. 9–13 →

[Meissl 1960/2] [Meissl 1960/2] P. Meissl, “Behandlung von Wasserschloßaufgaben mit Hilfe eines elektronischen Analogrechners, Teil 2”, in *mtw – Zeitschrift für moderne Rechentechnik und Automation*, 2/60, pp. 74–77 a, b, c, d, e, f

[Menzel et al. 2021] [Menzel et al. 2021] Johannes Menzel, Christian Plessl, Tobias Kenter, “The Strong Scaling Advantage of FPGAs in HPC for N-body Simulations”, in *ACM Transactions on Reconfigurable Technology and Systems*, Vol. 15, No. 1, pp. 10:1–10:30 →

[Merems 1951] [Merems 1951] Bernard H. Merems, “The House on 91st Street – Project Cyclone May Foreshadow Trips to the Moon”, in *Our Navy*, 1951, p. 4 →

[Meyer-Brötz 1960] [Meyer-Brötz 1960] G. Meyer-Brötz, “RA 800 – Ein transistorisierter Präzisions-Analogrechner”, in *Telefunken*

Zeitung, Vol. 33 (September 1960), No. 129, pp. 171–182 →

[Meyer-Brötz 1962] [Meyer-Brötz 1962] G. Meyer-Brötz, “Die Messung von Kenngrößen stochastischer Prozesse mit dem elektronischen Analogrechner”, in *Elektronische Rechenanlagen*, 4 (1962), No. 3, pp. 103–108 →

[Meyer-Brötz et al. 1966] [Meyer-Brötz et al. 1966] G. Meyer-Brötz, E. Heim, “Ein breitbandiger Operationsverstärker mit Silizium-Transistoren”, in *Telefunken Zeitung*, Vol. 39 (1966), No. 1, pp. 16–32 a, b

[Meyer zur Capellen 1949] [Meyer zur Capellen 1949] W. Meyer zur Capellen, *Mathematische Instrumente*, Akademische Verlagsgesellschaft Geest & Portig K.-G., Leipzig 1949 a, b

[Mezencev et al. 1978] [Mezencev et al. 1978] R. Mezencev, R. Lepeix, “Hybrid Simulation of a Non Linear Hydro Pneumatic Damper for Ships”, in *Simulation of Control Systems*, I. Troch (ed.), North-Holland Publishing Company, 1978, pp. 135–137 →

[Michaels] [Michaels] Lawrence H. Michaels, *The AC/Hybrid Power System Simulator and its Role in System Security*, n. d., author’s archive →

[Michaels et al.] [Michaels et al.] Lawrence H. Michaels, William Tessmer, John Muller, *The On-Line Power System Simulator*, Electronic Associates, Inc., n. d., author’s archive →

[Michaels et al. 1971] [Michaels et al. 1971] G. C. Michaels, V. Gourishankar, “Hybrid Computer Solution of Optimal Control Problems”, in *IEEE Transactions on Computers*, Vol. C-20, No. 2, February 1971, pp. 209–211 →

[Michels 1954] [Michels 1954] Lowell S. Michels, *Description of BENDIX D-12 DIGITAL DIFFERENTIAL ANALYZER*, Bendix Computer Division, Bendix Aviation Corporation, 5630 Arbor Vitae Street, Los Angeles 45, California, March 13, 1954 a, b, c

[Michelson et al. 1898] [Michelson et al. 1898] Albert Abraham Michelson, Samuel Wesley Stratton, “A New Harmonic Analyzer”, in *American Journal of Science*, 25, 1898, pp. 1–13 →

[Milan-Kamski 1969] [Milan-Kamski 1969] W. J. Milan-Kamski, “A High-Accuracy, Real-Time Digital Computer for Use in Continuous Control Systems”, in *1959 Proceedings of the Western Joint Computer Conference*, pp. 197–201 →

[Miller et al. 1954] [Miller et al. 1954] Joseph A. Miller, Aaron S. Soltes, Ronald E. Scott, *A Wide-Band Function Multiplier*, Computer Laboratory, Electronics Research Directorate, Air Force, Cambridge Research Center, Cambridge Massachusetts, December 1954 →

[Miller et al. 1955] [Miller et al. 1955] Joseph A. Miller, Aaron S. Soltes, Ronald E. Scott, “Wide-Band Analog Function Multiplier”, in *Electronics*, February, 1955, pp. 160–163 a, b

[Miller 2011] [Miller 2011] David Philip Miller, “The Mysterious Case of James Watt’s ’1785 Steam Indicator’: Forgery or Folklore in the History of an Instrument?”, in *Int. J. for the History of Eng. & Tech.*, Vol. 81, No. 1, January, 2011, pp. 129–150 →

[Mills/1] [Mills/1] Jonathan W. Mills, *The Architecture of an Extended Analog Computer Core*, Computer Science Department, Indiana University, n. d. →

[Mills/2] [Mills/2] Jonathan W. Mills, *Polymer Processors*, Computer Science Department, Indiana University, →http://www.cs.indiana.edu/pub/techreports/TR580.pdf, n. d., retrieved 03.03.2013 →

[Mills 1995] [Mills 1995] Jonathan W. Mills, *The continuous retina: Image processing with a single-sensor artificial neural field network*, Computer Science Department, Indiana University, technical report 443, November 13, 1995 →

[Mills et al. 2006] [Mills et al. 2006] Jonathan W. Mills, Bryce Himebaugh, Brian Kopecky, Matt Parker, Craig Shue, Chris Weilemann, "'Empty Space' Computers: The Evolution of an Unconventional Supercomputer", in *CF06*, May 3–8, 2006, Ischia, Italy →

[Mindell 1995] [Mindell 1995] David A. Mindell, "Automation's Finest Hour: Bell Labs and Automatic Control in World War II", in *IEEE Control Systems*, December 1995, pp. 72–78 a, b

[Mindell 2000] [Mindell 2000] David A. Mindell, "Opening Black's Box – Rethinking Feedback's Myth of Origin" in *Technology and Culture*, July 2000, Vol. 41, pp. 405–434 →

[Mindell 2002] [Mindell 2002] David A. Mindell, *Between Human and Machine*, The Johns Hopkins University Press, 2002 →

[Minne 1986] [Minne 1986] Michael Minne, "Analog Simulation brings Quick Solutions of Control Problems", in [EAI 1986, pp. 27 f.] →

[Minogue et al. 1960] [Minogue et al. 1960] Lorne E. Minogue, Cameron G. McIntyre, *Development of a Time-Shared Analog Computer*, Master's thesis, United States Naval Postgraduate School, Monterey, California, 1960 →

[Mitchell 1960] [Mitchell 1960] G. Mitchell, "An Analogue Computer for Investigating the Directivity Characteristics of Complex Arrays of Unit Aerials", in *The Post Office Electrical Engineers' Journal*, Vol. 52, Part 4, January 1960, pp. 246–250 →

[Mitchell et al. 1966] [Mitchell et al. 1966] E. E. L. Mitchell, J. B. Mawson, J. Bulger, "A Generalized Hybrid Simulation for an Aerospace Vehicle", in *IEEE Transactions on Electronic Computers*, Vol. EC-15, No. 3, June 1966, pp. 304–313 →

[Mitra 1955] [Mitra 1955] Samarendra Kumar Mitra, "Electrical Analog Computing Machine for Solving Linear Equations and

Related Problems”, in *The Review of Scientific Instruments*, Volume 26, Number 5, May, 1955, pp. 453–457 →

[Miura et al. 1967/1] [Miura et al. 1967/1] Takeo Miura, Junji Tsuda, Junzo Iwata, “Hybrid Computer Solution of Optimal Control Problems by the Maximum Principle”, in *IEEE Transactions on Electronic Computers*, Vol. EC-16, No. 5, October 1967, pp. 666–670 →

[Miura et al. 1967/2] [Miura et al. 1967/2] Takeo Miura, Junzo Iwata, Junji Tsuda, “An application of hybrid curve generation – cartoon animation by electronic computers”, in *AFIPS – spring joint computer conference*, April 1967, pp. 141–148 a, b

[Mohr 1959] [Mohr 1959] Otto Mohr (ed.), *Steuerungen und Regelungen elektrischer Antriebe*, VDE-Verlag GmbH, Berlin, 1959 →

[Montan-Forschung] [Montan-Forschung] N. N., *Rechengeräte für Verbundnetze*, Montan-Forschung, Düsseldorf, n. d. →

[Montan Forschung 1960] [Montan Forschung 1960] N. N., *Analog Rechengeräte WMK*, Technische Information, M 163, 10.10.1960 →

[Moore 1962] [Moore 1962] Robert C. Moore, *Studying Fluid Systems with the McIlroy Fluid Network Analyzer*, The Standard Electric Time Company, Springfield, Mass., 1962 →

[Moreland 1954] [Moreland 1954] William J. Moreland, “The Story of Shimmy”, in *Journal of the Aeronautical Sciences*, Volume 21, Number 12, December 1954, pp. 793 ff. →

[Moreno et al. 2020] [Moreno et al. 2020] Daniel Garcia Moreno, Alberto A. Del Barrio, Guillermo Botella Juan, “Simulating and deploying analog arithmetic circuits on FPAAs”, in *SummerSim ’20: Proceedings of the 2020 Summer Simulation Conference*, July 2020, Article No.: 19, pp. 1–12 →

[Morgan 1954] [Morgan 1954] Merle Loren Morgan, *A Computer for Algebraic Functions of a Complex Variable*, Ph.D. thesis, California Institute of Technology, 1954 a, b

[Morgan 1960] [Morgan 1960] Merle Loren Morgan, "A New Computer for Algebraic Functions of a Complex Variable", in *IFAC Proceedings Volumes*, Volume 1, Issue 1, August 1960, pp. 1222-1229 →

[Morgan 1961/1] [Morgan 1961/1] Merle Loren Morgan, "Algebraic Function Calculations Using Potential Analog Pairs", in *Proceedings of the IRE*, January 1961, pp. 276–282 →

[Morgan 1961/2] [Morgan 1961/2] Merle Loren Morgan, "Design of the ESIAC Algebraic Computer", in *IRE Transactions on Electronic Computers*, September 1961, pp. 524–529 a, b

[Morgan et al. 1961] [Morgan et al. 1961] Merle Loren Morgan, J. C. Looney, "Design of the ESIAC Algebraic Computer", in *IRE Transactions on Electronic Computers*, September 1961, pp. 524–529 →

[Morrill 1962] [Morrill 1962] Charles D. Morrill, "Electronic Multipliers and Related Topics", in [Huskey et al. 1962, pp. 3-40–3-62] a, b, c, d

[Morrison 1962] [Morrison 1962] E. Morrison, "Nuclear-Reactor Simulation", in [Huskey et al. 1962, pp. 5-87–5-93] a, b

[Morrison 2007] [Morrison 2007] James E. Morrison, *The Astrolabe*, Janus Publishing, 2007 →

[Morton 1966] [Morton 1966] R. R. A. Morton, "A simple d.c. to 10 Mc/s analogue multiplier", in *Journal of Scientific Instruments*, Vol. 43, 1966, pp. 165–168 →

[MTW 1964/1] [MTW 1964/1] N. N., "EAI awarded contract for Hybrid Computing System", in *mtw – Zeitschrift für moderne*

Rechentechnik und Automation, 4/64, p. 175 →

[MTW 1964/2] [MTW 1964/2] N. N., “Nuclear Power Plant of N. S. Savannah simulated by Analog Computers”, in *mtw – Zeitschrift für moderne Rechentechnik und Automation*, 3/64, p. 129 →

[Müller 1956] [Müller 1956] Paul Friedrich Müller, “Die Integrieranlage des Rheinisch-Westfälischen Instituts für Instrumentelle Mathematik in Bonn”, in *Forschungsberichte des Wirtschafts- und Verkehrsministeriums Nordrhein-Westfalen*, Springer Fachmedien Wiesbaden GmbH, 1956 a, b

[Müller 1957] [Müller 1957] O. Müller, “The Control System of the V-2”, in [Benecke et al. 1957, pp. 80–101], 1957 →

[Müller 1986] [Müller 1986] Heribert Müller, “Simulation und Lösung physikalischer Probleme mit dem Analogrechner”, in *Praxis der Naturwissenschaften, Physik*, Aulis Verlag, Heft 3/35, 15. April 1986, pp. 21–25 →

[murhpy 1960] [murhpy 1960] Walter A. Murphy, “Fire Control Computers – Their Development”, in *Computers and Automation*, 1960, Vol. 9, No. 8, pp. 14–18 →

[Muskhelishvili 1953] [Muskhelishvili 1953] N. I. Muskhelishvili, *Singular Integral Equations*, P. Noordhoff N. V., Groningen-Holland, 1953 →

[Mycka 2005] [Mycka 2005] Jerzy Mycka, “Analog computation beyond the Turing limit”, in *Applied Mathematics and Computation*, 178 (2006), Elsevier, pp. 103–117 →

[Mynall 1947] [Mynall 1947] D. J. Mynall, “Electronic Function Generator”, in *Nature*, No. 4048, May 31, 1947, p. 743 →

[Nafalski 1976] [Nafalski 1976] Andrzej Nafalski, “Simulierung eines statischen Frequenzvervielfachers mit nichtlinearem Eisenkern am Analogrechner”, in [TU Wien 1976/3, pp. 32 ff.] →

[Nalley 1969] [Nalley 1969] Donald Nalley, *Z Transform and the Use of the Digital Differential Analyzer as a Peripheral Device to a General Purpose Computer*, NASA Technical Memorandum, NASA TM X-53866, August 12, 1969 →

[NASA 1963] [NASA 1963] National Aeronautics and Space Administration, *Six-Degrees-of-Freedom Gemini Reentry Simulation*, NASA Project Gemini Working Paper No. 5003, September 12, 1963 →

[NASA 1964] [NASA 1964] National Aeronautics and Space Administration, *The Mercury-Redstone Project*, Saturn/Apollo Systems Office, December 1964 →

[NASA 1968] [NASA 1968] National Aeronautics and Space Administration, *Saturn V Flight Manual – SA 503*, 1. November 1968 →

[Nava-Segura et al.] [Nava-Segura et al.] A. Nava-Segura, L. L. Freris, *Hybrid Computer Simulation of DC Transmission Systems*, n. d., author's archive →

[Naylor et al. 1966] [Naylor et al. 1966] Thomas H. Naylor, Joseph L. Balintfy, Donald S. Burdick, Kang Chu, *Computer Simulation Techniques*, John Wiley & Sons, Inc., New York, London, Sydney, 1966 →

[NC STATE 2017] [NC STATE 2017] N. N., *NC STATE Engineering*, Fall/Winter 2017 →

[Nebeker 1994] [Nebeker 1994] Frederik Nebeker, "Calculating Power: Edwin L. Harder and Analog Computing in the Electric Power Industry", in *Sparks of Genius: Portraits of Electrical Engineering Excellence*, IEEE Press, 1994, pp. 159–200 a, b, c, d, e, f, g, h, i, j, k, l, m

[NEC 1987] [NEC 1987] NEC, *MOS Integrated Circuit μPD22100, μPD22148, Crosspoint Switch with Control Memory*, Data Sheet,

NEC Corporation, 1987 →

[Nedey et al. 1967] [Nedey et al. 1967] R. Nedey, R. Trichet, M. Duvelleroy, M. Fourquet, H. Apter, Y. Ploton, “Simulation de l’Equilibre Acido-Basique de l’Organisme”, in [IACM 1976, pp. 1124–1131] →

[Neufeld et al. 1989] [Neufeld et al. 1989] Michael Neufeld, Helmut Hoelzer, *Oral Interview*, →https://www.si.edu/media/NASM/NASM-NASM_AudioIt-000003012DOCS-000001.pdf, retrieved 27.06.2025 →

[Neufeld 2007] [Neufeld 2007] Michael J. Neufeld, *Von Braun – Dreamer of Space, Engineer of War*, Borzoi Book, Alfred A. Knopf, 2007 →

[Neumann 1956] [Neumann 1956] John von Neumann, “Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components”, in *Automata Studies*, ed. Claude Shannon, Princeton University Press, 1956 a, b, c

[Newland 2005] [Newland 2005] D. E. Newland, *An Introduction to Random Vibrations, Spectral & Wavelet Analysis*, 3rd edition, reprint, Dover Publications →

[Nicole 1967] [Nicole 1967] Liles Nicole, “Exploitation Systematique de Cheminees D’Equilibre et de Turbines d’un Ensemble Hydraulique sur Calculateur Analogique”, in [IACM 1976, pp. 801–811] →

[Nise] [Nise] Norman S. Nise, *Analog Computer Experiments for Undergraduate Courses in Network Analysis and Automatic Controls*, Vol. II, No. 2, n. d., author’s archive →

[Nix 1965] [Nix 1965] Siegfried H. Nix, “Problematik und Methodik bei der Analyse von Gasrohrnetzen”, in *Rohre – Rohrleitungsbau – Rohrleitungstransport*, Heft 5, Oktober 1965, pp. 255–271 →

[N. N.] [N. N.] N. N., “Scientist Perfects Machine to Solve Hardest Mathematical Problems Easily”, n. d., author’s collection →

[N. N. 1945] [N. N. 1945] N. N., *Das Gerät A4 Baureihe B, Teil III, Gerätebeschreibung V2*, OKH/Wa A/Wa Prüf, Anlage zu Bb.Nr 19/45 gK, 1.2.1945 4/64, p. 175 a, b, c

[N. N. 2006] [N. N. 2006] N. N., *Video Games – Did They Begin at Brookhaven?*, → http://www.osti.gov/accomplishments/videogame.html, retrieved 20.11.2006 →

[Nolan 1954/1] [Nolan 1954/1] John E. Nolan, “Analog Computers and their Application to Heat Transfer and Fluid Flow – Part 1”, in *Computers and Automation*, 1954, Vol. 3., No. 9, pp. 20–23 →

[Nolan 1954/2] [Nolan 1954/2] John E. Nolan, “Analog Computers and their Application to Heat Transfer and Fluid Flow – Part 2”, in *Computers and Automation*, 1954, Vol. 3., No. 10, pp. 24–34 →

[Nolan 1955] [Nolan 1955] John E. Nolan, “Analog Computers and their Application to Heat Transfer and Fluid Flow – Part 3”, in *Computers and Automation*, 1955, Vol. 4., No. 1, pp. 31–35 a, b

[Nolan 1955] [Nolan 1955] John E. Nolan, “Analog Computers and their Application to Heat Transfer and Fluid Flow – Part 1, 2, 3”, in [Paynter ed. 1955, pp. 109–126] →

[Nordsieck] [Nordsieck] Nordsieck, *The Nordsieck Computer – a precision low cost differential analyzer*, n. d. →

[Noronha] [Noronha] Leo G. Noronha, *The Benefits of Analog Computation and Simulation in the Electrical Supply Industry*, n. d., author’s archive →

[Northrop 1950] [Northrop 1950] N. N., *MADDIDA Digital Differential Analyzer*, Northrop Aircraft, Inc., Brochure No. 38, December, 1950 →

[Norum et al. 1962] [Norum et al. 1962] Vance D. Norum, Marvin Adelberg, Robert L. Farrenkopf, “Analog simulation of particle trajectories in fluid flow”, in *AIEE-IRE ’62 (Spring): Proceedings of the May 1–3, 1962, spring joint computer conference*, May 1962, pp. 235–254 →

[Nosker 1957] [Nosker 1957] Paul Nosker, “Dynamic Systems Synthesizer”, in [White Sands 1957, pp. 143–168] a, b

[Ochs et al. 2021] [Ochs et al. 2021] Karlheinz Ochs, Sebastian Jenderny, “An equivalent electrical circuit for the Hindmarsh-Rose model”, in *International Journal of Circuit Theory and Applications*, 49, 2021, pp. 3526–3539 →

[O’Grady 1966] [O’Grady 1966] Emmett Pearse O’Grady, “A Hybrid-Code Differential Analyzer”, in *Annales de l’Association internationale pour le Calcul analogique*, No. 1, Janvier 1966, pp. 13–21 →

[O’Grady 1967] [O’Grady 1967] Emmett Pearse O’Grady, “Correlation Method for Computing Sensitivity Functions on a High-Speed Iterative Analog Computer”, in *IEEE Transactions on Electronic Computers*, Vol. EC-16, No. 2, April 1967, pp. 140–146 →

[Oliver et al. 1974] [Oliver et al. 1974] W. Kent Oliver, Dale E. Seborg, D. Grant Gisher, “Hybrid Simulation of a Computer-Controlled Evaporator”, in *SIMULATION*, September 1974, pp. 77–84 →

[Olson 1943] [Olson 1943] Harry F. Olson, *Dynamical Analogies*, D. van Nostrand Company, Inc., 1943 →

[Okah-Avae 1978] [Okah-Avae 1978] B. E. Okah-Avae, “Analogue computer simulation of a rotor system containing a transverse crack”, in *SIMULATION*, December 1978, pp. 193–198 →

[Onizawa et al. 2019] [Onizawa et al. 2019] Naoya Onizawa, Warren J. Gross, Takahiro Hanyu, “Brain-Inspired Computing”, in [Gross et al.

2019, pp. 185–199] →

[Onsemi 2005] [Onsemi 2005] N. N., *TVS/Zener Theory and Design Considerations*, Handbook, HBD854/D, Rev. 0, ON Semiconductor, June 2005 →

[Oodrys 1953] [Oodrys 1953] Arthur Alexander Oodrys, *The ANACOM – an Electrical Analogue Computer*, Master's Thesis, University of Southern California, 1953 a, b, c, d, e, f, g, h

[Orth 1967] [Orth 1967] Karlheinz Orth, "Untersuchung des Betriebsverhaltens eines grossen Druckwasser-Leistungsreaktors", in [IACM 1976, pp. 1043–1055] →

[Ott 1964] [Ott 1964] A. Ott, "Zur Bestimmung des Korrelationskoeffizienten zweier Funktionen mit dem Analogrechner", in *Elektronische Rechenanlagen*, 6 (1964), No. 3, pp. 144–148 →

[Otterman 1960] [Otterman 1960] Joseph Otterman, "The Properties and Methods for Computation of Exponentially-Mapped-Past Statistical Variables", in *IRE Transactions on Automatic Control*, Volume: AC-5, Issue: 1, 1. Jan. 1960, pp. 11–17, DOI: 10.1109/TAC.1960.6429289 →

[Ovsyanko 1977] [Ovsyanko 1977] V. M. Ovsyanko, "The Theory of Synthesis of Electronic Circuits of Linear and Non-linear Object of Structural Mechanics and Applied Elasticity Theory", in *14. seminář MEDA ANALOGOVÁ A HYBRIDNÍ VÝPOČETNÍ TECHNIKA*, Praha 1977, pp. 25–29 →

[Owen et al. 1960] [Owen et al. 1960] P. L. Owen, M. F. Partridge, T. R. H. Sizer, *CORSAIR, a Digital Differential Analyser*, Royal Aircraft Establishment (Farnborough), Technical Note No. I.A.P. 1123, December, 1960 a, b, c

[Owens 1986] [Owens 1986] Larry Owens, "Vannevar Bush and the Differential Analyzer: The Text and Context of an Early

Computer", in *Technology and Culture*, Vol. 27, No. 1, Jan. 1986, pp. 63–95 a, b, c, d, e

[Packard Bell] [Packard Bell] N. N., *The HYCOMP Hybrid Analog/Digital Computing System*, Packard-Bell Computer, n. d. →

[Palevsky 1962] [Palevsky 1962] Max Palevsky, "The Digital Differential Analyzer", in [Huskey et al. 1962, pp. 19-14–19-74] →

[Palevsky 1973] [Palevsky 1973] Max Palevsky, *Computer Oral History Collection*, Smithsonian National Museum of American History, February 15, 1973, →https://www.si.edu/media/NMAH/NMAH-AC0196_pale730215.pdf, retrieved 02.03.2025 →

[Palm 2014] [Palm 2014] Michael Palm, *Historische Integratoren der Firma A. Ott – Anschauliche Darstellung der Funktionsweise und Animation*, Wissenschaftliche Hausarbeit, Darmstadt, 2014 →

[Panasonic 3007] [Panasonic 3007] N. N., *MN3007 – 1024-stage low noise BBD* →

[Parhi et al. 2020] [Parhi et al. 2020] Keshab K. Parhi, Nanda A. Unnikrishnan, "Brain-Inspired Computing: Models and Architectures", in *IEEE Open Journal of Circuits and Systems*, Volume 1, 2020, pp. 165–204 →

[Park et al. 1972] [Park et al. 1972] William H. Park, James C. Wambold, "Teaching Digital and Hybrid Simulation of Mechanical Systems at the Graduate Level", Delivered during the Joint ACES/ASEE Session No. 3540 at the 1972 Annual Meeting of the American Society for Engineering Education at Texas Tech, Lubbock, Texas, June 19–22, 1972 →

[Park et al. 2024] [Park et al. 2024] Juseong Park, Boseok Kim, Hyojin Sung, "NavCim: Comprehensive Design Space Exploration for Analog Computing-in-Memory Architectures", in *PACT'24*, October 14–16, Long Beach, CA, USA, pp. 168–182 →

[Paschkis et al. 1968] [Paschkis et al. 1968] Victor Paschkis, Frederick L. Ryder, *Direct Analog Computers*, Interscience Publishers, 1968 a, b, c

[Pater et al. 1964] [Pater et al. 1964] L. de Pater, Jw. van den Berg, "An Electrical Analogue of the Entire Human Circulatory System", in *Medical Electronics and Biological Engineering*, Vol. 2, 1964, pp. 161–166 a, b, c

[Patowary 2019] [Patowary 2019] Kaushik Patowary, *Vladimir Lukyanov's Water Computer*, → https://www.amusingplanet.com/2019/12/vladimir-lukyanovs-water-computer.html, retrieved 06.05.2025 →

[Payne 1988] [Payne 1988] Peter R. Payne, "An Analog Computer which Determines Human Tolerance to Acceleration", in *39th Annual Astronautical Congress of the International Astronautical Federation*, Bangalore, 8-15 Oct. 1988, pp. 271–300 →

[Paynter ed. 1955] [Paynter ed. 1955] Henry M. Paynter (ed.), *A Palimpsest on the Electronic Analog Art*, printed by Geo. A. Philbrick Researches Inc., AD 1955 a, b

[Paynter et al. 1955] [Paynter et al. 1955] Henry M. Paynter, J. M. Asce, "Surge and Water Hammer Problems", in [Paynter ed. 1955, pp. 217–223] →

[Paynter 1955/1] [Paynter 1955/1] Henry M. Paynter, "Methods and Results from M.I.T. Studies in Unsteady Flow", in [Paynter ed. 1955, pp. 224–228] →

[Paynter 1955/2] [Paynter 1955/2] Henry M. Paynter, "A Discussion by H. M. Paynter of AIEE Paper 53 – 172", in [Paynter ed. 1955, pp. 229–232] →

[Paynter] [Paynter] Henry M. Paynter, "A Retrospective on Early Analysis and Simulation of Freeze and Thaw Dynamics", cf.

→http://www.me.utexas.edu/~lotario/paynter/hmp/PAYNTER_Permafrost.pdf, n. d., retrieved 04.12.2008 →

[Pease 2003] [Pease 2003] Bob Pease, "What's All This K2-W Stuff, Anyhow?", in *electronic design*, January 2003, →http://electronicdesign.com/article/analog-and-mixed-signal/what-s-all-this-k2-w-stuff-anyhow-2530, retrieved 12.12.2012 a, b

[Peenemünde] [Peenemünde] N. N., *The Story of Peenemünde or what might have been – Peenemünde East through the eyes of 500 detained at Garmish*, n. d. a, b

[Perera 1969] [Perera 1969] K. K. Y. Wije Perera, "Optimum generating schedule for a hydro-thermal power system / an analog computer solution to the short-range problem", in *SIMULATION*, April 1969, pp. 191–199 →

[Peretz 1967] [Peretz 1967] R. Peretz, "Projet CHANCE", in [IACM 1976, pp. 834–848] →

[Petrarca 1967] [Petrarca 1967] Sergio Petrarca, "An Analog Iterative Model for the Calculation of Coolant Density in Pressure-Tube Nuclear Boiler Channels", in [IACM 1976, pp. 980–987] →

[Petzold 1992] [Petzold 1992] Hartmut Petzold, *Moderne Rechenkünstler – Die Industrialisierung der Rechentechnik in Deutschland*, Verlag C. H. Beck, 1992 a, b, c, d

[Petzold 1997] [Petzold 1997] Harmut Petzold, "Der Rechenriese – Die Maschine von John B. Wilbur", in *Kultur & Technik – Zeitschrift des Deutschen Museums*, C. H. Beck, München, 4/1997, pp. 28–29 a, b

[Pfaltzgraff 1969] [Pfaltzgraff 1969] David J. Pfaltzgraff, "Analog Simulation of the Bouncing-Ball Problem", in *American Journal of Physics*, Volume 37, Number 10, October 1969, pp. 1008–1013 →

[Pfeiffer 1959] [Pfeiffer 1959] Paul E. Pfeiffer, “A Four-Quadrant Multiplier Using Triangular Waves, Diodes, Resistors, and Operational Amplifiers”, in *IRE Transactions on Electronic Computers*, June 1959, pp. 222–227 →

[Phalanx 2020] [Phalanx 2020] N. N., “Highlighting Women in Operations Research and Their Achievements”, in *Phalanx*, Vol. 53, No. 4, 2020, pp. 16–19 →

[Philbrick 1948] [Philbrick 1948] George A. Philbrick, “Designing Industrial Controllers by Analog”, in *Electronics*, June, 1948, pp. 108–111 →

[Phillips 1950] [Phillips 1950] A. W. Phillips, “Mechanical Models in Economic Dynamics”, in *Economica*, New Series, Vol. 17, No. 67, Aug. 1950, pp. 283–305 a, b

[Piceni et al. 1975] [Piceni et al. 1975] Hans A. L. Piceni, Pieter Eykhoff, “The Use of Hybrid Computers for System-Parameter Estimation”, in *Annales de l’Association internationale pour le Calcul analogique*, No. 1, Janvier 1975, pp. 9–22 →

[Pierre 1986] [Pierre 1986] Donald A. Pierre, *Optimization Theory with Applications*, Dover Publications, Inc., New York, 1986 a, b, c

[Pirrello et al. 1971] [Pirrello et al. 1971] C. J. Pirrello, R. D. Hardin, J. P. Capellupo, W. D. Harrison, *An Inventory of Aeronautical Ground Research Facilities – Volume IV – Engineering Flight Simulation Facilities*, National Aeronautics and Space Administration, Washington, D. C., November 1971, NASA CR-1877 →

[Plow 1959] [Plow 1959] Arthur E. Plow, *Electronic analog of a free piston gas generator*, U. S. Naval Postgraduate School, Monterey, California, 1959 →

[Pol et al. 1928] [Pol et al. 1928] Balthasar van der Pol, J. van der Mark, “The Heartbeat considered as a Relaxation Oscillation, and

an Electrical Model of the Heart", in *Phil. Mag.* 7, 1928, pp. 763–775 →

[Popović 1964] [Popović 1964] D. P. Popović, "Die Automatisierung des von Mieses'schen Iterationsverfahrens auf dem Analogrechner", in *mtw – Zeitschrift für moderne Rechentechnik und Automation*, 3/64, pp. 104–110 →

[Poppelbaum et al. 1967] [Poppelbaum et al. 1967] W. C. Poppelbaum, C. Afuso, J. W. Esch, "Stochastic computing elements and systems", in *Proceedings of the AFIPS Fall Joint Computer Conference*, pp. 635–644 →

[Poppelbaum 1968] [Poppelbaum 1968] W. C. Poppelbaum, "Annual Report Part B, Pattern Processing and Memory Research", in *Annual Report Computer Systems Research*, September 1, 1967 to August 31, 1968 →

[Poppelbaum 1979] [Poppelbaum 1979] W. C. Poppelbaum, *Burst Processing*, Final Report for the Navy for Contract N000014-75-C-0982, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1979 a, b, c, d

[Popular Mechanics 1928] [Popular Mechanics 1928] N. N., "Machine Solves Problems too Deep for Brain", in *Popular Mechanics*, 1, 1928 →

[Popular Mechanics 1950] [Popular Mechanics 1950] N. N., "It's Small But Smart, This 'Suitcase Brain'", in *Popular Mechanics*, 8, 1950 →

[Porter 2003] [Porter 2003] Arthur Porter, "Building the Manchester differential analyzers: A personal reflection", in *IEEE Annals of the History of Computing*, Vol. 25, No. 2, April–June 2003, pp. 86–92 →

[Pour-El 1974] [Pour-El 1974] Marian Boykan Pour-El, "Abstract computability and its relation to the general purpose analog

computer (some connections between logic, differential equations and analog computers)", in *Transactions of the American Mathematical Society*, Volume 199, 1974, pp. 1–28 →

[Powell] [Powell] Fred O. Powell, "Analog simulation of an adaptive two-time-scale control system", Advanced Electronic Systems Research Department, Bell Aerospace Division of Textron, Buffalo, New York 14240, n. d. →

[Press et al. 2001] [Press et al. 2001] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, *Numerical Recipes in Fortran 77 – The Art of Scientific Computing, Volume 1 of Fortran Numerical Recipes*, Cambridge University Press, Second Edition, 2001 a, b

[Preuss 1962] [Preuss 1962] Heinzwerner Preuss, *Grundriss der Quantenchemie*, Bibliographisches Institut, Mannheim, 1962 a, b

[Preuss 1965] [Preuss 1965] Heinzwerner Preuss, *Quantentheoretische Chemie*, Bibliographisches Institut, Mannheim, 1965 →

[de Solla Price 1974] [de Solla Price 1974] Derek John de Solla Price, "Gears from the Greeks: The Antikythera Mechanism – a Calendar Computer from ca. 80 B. C.", in *Transactions of the American Philosophical Society*, Volume 64, Part 7, 1974 →

[Pronk 2019] [Pronk 2019] Kees Pronk, "De Rekenmachine van Hammer", in *Jubileum Magazine 50 jaar Studieverzameling*, Delft University of Technology, 2019, pp. 37–39 →

[Pronk et al. 2021] [Pronk et al. 2021] Kees Pronk, Piet Trimp, *Lunchlezing Studieverzameling*, 4. October 2021 →

[Pronk et al. 2022] [Pronk et al. 2022] Kees Pronk, Piet Trimp, "The Esiac-10 algebraic computer – A theoretical decomposition by the Historic Collection", in *MAXWELL*, November 2022, pp.6-11 →

[Puchta 1996] [Puchta 1996] Susann Puchta, “On the Role of Mathematics and Mathematical Knowledge in the Invention of Vannevar Bush’s Early Analog Computers”, in *IEEE Annals of the History of Computing*, Vol. 18, No. 4, 1996, pp. 49–59 →

[Püttmann 2014] [Püttmann 2014] Thomas Püttmann, “Der Seilcomputer Kelvin”, in *Mitteilungen der Deutschen Mathematiker-Vereinigung*, Volume 22, Issue 4, pp. 222-225 →

[Radio und Fernsehen 1961] [Radio und Fernsehen 1961] N. N., “Moderne sowjetische Analogierechenmaschinen”, in *radio und fernsehen*, 12, 1961, pp. 379–380 →

[Ragazzini et al. 1947] [Ragazzini et al. 1947] John R. Ragazzini, Robert H. Randall, Frederick A. Russel, “Analysis of Problems in Dynamics by Electronic Circuits”, in *Proceedings of the I.R.E.*, Vol. 35, May 1947, pp. 444 ff. →

[Ramirez 1976] [Ramirez 1976] W. Fred Ramirez, *Process Simulation*, D. C. Heath and Company, 1976 a, b, c, d, e

[Randery 1964] [Randery 1964] Vijay K. Randery, “Study of a Parametron on an Analog Computer”, in *IEEE Transactions on Electronic Computers*, October, 1964, pp. 612–614 →

[Ranfft et al. 1977] [Ranfft et al. 1977] Roland Ranfft, Hans-Martin Rein, “Analog simulation of bipolar-transistor circuits”, in *SIMULATION*, September 1977, pp. 75–78 a, b, c

[Rasfeld 1983] [Rasfeld 1983] Peter Rasfeld, “Zur Darstellung und Untersuchung von Funktionen im Mathematikunterricht mittels elektronischer Analogrechner”, in *Praxis der Mathematik*, Vol. 25, Issue 11, pp. 325–333 →

[Ratheiser] [Ratheiser] Ludwig Ratheiser, *Rundfunkröhren – Eigenschaften und Anwendung*, Regelien’s Verlag, Berlin, n. d. →

[Rationalisierungskuratorium 1957]
[Rationalisierungskuratorium 1957] Rationalisierungskuratorium der Deutschen Wirtschaft (Hg.), *Automatisierung*, Carl Hanser Verlag, München, 1957 a, b

[Rattay 1976] [Rattay 1976] F. Rattay, “Bewegungssimulation eines Fussballs”, in [TU Wien 1976/1, pp. 33–34] →

[Rattay 1977] [Rattay 1977] F. Rattay, “Magnetfeldberechnung für einen adiabatischen Neutronenspindreher”, in [TU Wien 1977/3, pp. 22 ff.] →

[Rattay 1978/1] [Rattay 1978/1] F. Rattay, “Analoge Simulation der Bewegung eines Autos bei der Fahrt über einen Knüppelweg mit analoger Fahrzeugdarstellung”, in [TU Wien 1978/1, pp. 41 ff.] →

[Rattay 1978/2] [Rattay 1978/2] F. Rattay, “Analoge Simulation der mechanischen Vorgänge beim Läuten einer Glocke”, in [TU Wien 1978/2, pp. 30 ff.] a, b

[Rattay 1979] [Rattay 1979] F. Rattay, “Schmiedeeisenkunst mit dem Hybridrechner”, in [TU Wien 1979/1, pp. 49] →

[Ratz 1967] [Ratz 1967] Alfred G. Ratz, “Analog Computation of Fourier Series and Integrals”, in *IEEE Transactions on Electronic Computers*, Vol. EC-16, No. 4, August 1967, p. 515 →

[Rechberger 1959] [Rechberger 1959] H. Rechberger, “Zweite internationale Tagung für Analogierechentechnik”, in *mtw – Zeitschrift für moderne Rechentechnik und Automation*, 1/59, pp. 18–19 →

[Redheffer 1953] [Redheffer 1953] Raymond M. Redheffer, *Computing Machine*, United States Patent 2656102, Oct. 20, 1953 →

[Reeves Instrument Corporation] [Reeves Instrument Corporation] N. N., *Instruction Book for Reeves Electronic Analogue*

Computer C-101 Mod 5, Reeves Instrument Corporation, New York 28, N. Y., n. d. →

[Reif 2007] [Reif 2007] John H. Reif, "Mechanical Computation: its Computational Complexity and Technologies", in *Encyclopedia of Complexity and Systems Science*, 2007, =→https://api.semanticscholar.org/CorpusID:13484042, retrieved 02.05.2025 →

[Reihing 1959] [Reihing 1959] John V. Reihing, "A Time-Sharing Analog Computer", in *IRE-AIEE-ACM '59 (Western): Papers presented at the March 3–5, 1959, western joint computer conference*, March 1959, pp. 341–349 →

[Reinel 1976] [Reinel 1976] K. Reinel, "Bewegungssimulatoren für Raumfahrt-Lageregelungssysteme", in [Schöne 1976/1, pp. 464–475] →

[Reinhardt 1967] [Reinhardt 1967] A. Reinhardt, "Bearbeitung grosser Gleichungssysteme der Reaktortechnik auf dem Analogrechner", in [IACM 1976, pp. 1030–1042] a, b

[Reisig 1999] [Reisig 1999] Gerhard H. R. Reisig, *Raketenforschung in Deutschland – Wie die Menschen das All eroberten*, Wissenschaft und Technik Verlag, Berlin, 1999 →

[Remington 1956] [Remington 1956] Remington Rand Univac, *Increment Computer Logic And Programming*, Bomber Weapons Defense Computer Study, Final Engineering Report, Volume 4, October 1956 →

[Resconi et al. 2015] [Resconi et al. 2015] G. Resconi, I. Licata, "Computation by Intention and Electronic Image of the Brain", in *Journal of Applied & Computational Mathematics*, 4:4, 2015, DOI: 10.4172/2168-9679-1000232, →https://www.hilarispublisher.com/open-access/computation-

by-intention-and-electronic-image-of-the-brain-2168-9679-1000232.pdf, retrieved 31.08.2025 →

[Reutter et al. 1968] [Reutter et al. 1968] Fritz Reutter, Hermann Josef Neukirchen, Dietmar Sommer, *Herstellung konformer Abbildungen mit Hilfe des Analogrechners – Praktische Behandlung der Umströmung zweifach zusammenhängender Gebiete*, Forschungsberichte des Landes Nordrhein-Westfalen, Westdeutscher Verlag, Köln und Opladen, 1968 →

[Richards 1968] [Richards 1968] Jesse C. Richards, "Automatic checkout of a large hybrid computer system", in *Fall Joint Computer Conference*, 1968, pp. 987–996 →

[Riedel et al. 2019] [Riedel et al. 2019] Marc Riedel, Weikang Qian, "Synthesis of Polynomial Functions", in [Gross et al. 2019, pp. 103–120] →

[Riedel 2019] [Riedel 2019] Marc Riedel, "Deterministic Approaches to Bitstream Computing", in [Gross et al. 2019, pp. 121–136] a, b

[Rieger et al. 1974] [Rieger et al. 1974] Neville F. Rieger, Charles H. Thomas Jr., "Some Recent Computer Studies on the Stability of Rotors in Fluid-Film Bearings", Rochester Institute of Technology, Mechanical Engineering Department, July, 1974 a, b

[Rideout 1962] [Rideout 1962] Vincent C. Rideout, "Random-Process Studies", in [Huskey et al. 1962, pp. 5-94–5-110] a, b, c

[Rigas et al.] [Rigas et al.] Harriett B. Rigas, Andrew M. Juraszek, *Some Approaches to the Design of a Model for an Aquatic Ecosystem*, n. d., author's archive a, b, c

[Robinson] [Robinson] Tim Robinson, *Torque amplifiers in Meccano*, →http://www.meccano.us/differential_analyzers/robinson_da/torque_amplifiers.pdf, n. d., retrieved 26.07.2005 a, b

[Robinson 2005] [Robinson 2005] Tim Robinson, “The Meccano Set Computers”, in *IEEE Control Systems Magazine*, June 2005, pp. 74–83 →

[Robinson 2008] [Robinson 2008] Tim Robinson, *Oral History of Arthur Porter*, Computer History Museum, Recorded March 8, 2008, Advance, North Carolina, →https://archive.computerhistory.org/resources/access/text/2015/06/102658245-05-01-acc.pdf, retrieved 19.02.2021 →

[Rockcastle et al. 1956] [Rockcastle et al. 1956] C. H. Rockcastle, A. J. Yates, *Simulation of a Turbojet Engine on a Standard Electronic Analogue Computer for the Purpose of Control Study*, Thesis R664, 1956 →

[Roderer et al. 1967] [Roderer et al. 1967] H. Roderer, W. Schnitzler, H. Seelmann, “Die Anwendung eines Hybridrechners bei der Simulation Aerodynamischer Flugkörper”, in [IACM 1976, pp. 947–954] →

[Rodríguez 1991] [Rodríguez 1991] José Miguel Olazaran Rodríguez, *A Historical Sociology of Neural Network Research*, Ph.D. thesis, University of Edinburgh, 1991 →

[Roedel 1955/1] [Roedel 1955/1] Jerry Roedel, “History and Nature of Analog Computors”, in [Paynter ed. 1955, pp. 27–47] →

[Roedel 1955/2] [Roedel 1955/2] Jerry Roedel, “Application of an Analog Computer to Design Problems for Transportation Equipment”, in [Paynter ed. 1955, pp. 199–215] →

[Röpke et al. 1969] [Röpke et al. 1969] Horst Röpke, Jürgen Riemann, *Analogcomputer in Chemie und Biologie*, Springer-Verlag, 1969 →

[Rössler 2005] [Rössler 2005] Eberhard Rössler, *Die Torpedos der deutschen U-Boote*, Verlag E. S. Mittler & Sohn GmbH, 2005 →

[Roffman et al. 1965] [Roffman et al. 1965] Gary L. Roffman, Silas Katz, *FLUERICS – 15. Experimental Flueric Analog Computation*, Harry Diamond Laboratories, 10 June 1965 →

[Rogers 1963] [Rogers 1963] Alan E. Rogers, "Hybrid Computation: What and Why?", in *Computers and Automation*, 1963, Vol. 12, No. 10, pp. 10–17 a, b, c, d

[Rohde 1977] [Rohde 1977] Wolfgang H. Rohde, *Beurteilung und Optimierung von Maschinensystemen in der Entwurfsphase – Dargestellt am Beispiel eines drehzahlgesteuerten Walzwerksantriebes*, Dissertation an der Technischen Universität Clausthal, 1977 →

[Rohde et al. 1981] [Rohde et al. 1981] Wolfgang H. Rohde, Jürgen Stelbrink, "Auslegung und konstruktive Gestaltung von Antriebssystemen schwerer Walzwerke", in *Stahl und Eisen*, No. 13/14/1981, pp. 164–173 →

[Rosenblatt 1958] [Rosenblatt 1958] Frank Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain", in *Psychological Review*, 65(6), 1958, pp. 386–408 →

[Rosko 1968] [Rosko 1968] Joseph S. Rosko, "Comments on 'Hybrid Computer Solution of Optimal Control Problems by the Maximum Principle'", *IEEE Transactions on Computers*, Vol. C-17, No. 9, September 1968, p. 899 →

[Rubel 1981] [Rubel 1981] Lee A. Rubel, "A Universal Differential Equation", in *Bulletin of the American Mathematical Society*, Volume 4, Number 3, May 1981, pp. 345–349 →

[Rubel 1989] [Rubel 1989] Lee A. Rubel, "Digital Simulation of Analog Computation and Church's Thesis", in *The Journal of Symbolic Logic*, Volume 54, Number 3, September 1989, pp. 1011–1017 →

[Rubel 1993] [Rubel 1993] Lee A. Rubel, “The Extended Analog Computer”, in *Advances in Applied Mathematics*, 14, 1993, pp. 39–50 →

[Ruby 1996] [Ruby 1996] Lawrence Ruby, “Applications of the Mathieu equation”, in American Journal of Physics, 64 (1), January 1996, pp. 39-44 →

[Rudnicki 1977] [Rudnicki 1977] Mieczyslaw Rudnicki, “Analogrechner MEDA in der Lasertechnik”, in *14. seminář MEDA ANALOGOVÁ A HYBRIDNÍ VÝPOČETNÍ TECHNIKA*, Praha 1977, pp. 53–55 →

[Russell 1962] [Russell 1962] Paul E. Russell, “Repetitive Analog Computers”, in [Huskey et al. 1962, pp. 6-17–6-25] a, b

[Russel et al. 1971] [Russel et al. 1971] Jack A. Russel, Bradford J. Baldwin, *Golf Game Computing System*, United States Patent 3598976, August 10, 1971 →

[Russell 1978] [Russell 1978] A. Russell, “Moon Landing Game”, in *Practical Electronics*, November 1978, pp. 1138–1142 →

[Ryder 2009] [Ryder 2009] William H. Ryder, “A System Dynamics View of the Phillips Machine”, The 27th Interational Conference of the Systems Dynamics Society, July 26–30, 2009, →https://proceedings.systemdynamics.org/2009/proceed/papers/P1038.pdf, retrieved 01.08.2022 →

[Ryder et al. 2021] [Ryder et al. 2021] William H. Ryder, Robert Y. Cavana, “A System Dynamics Translation of the Phillips Machine”, in [Cavana et al. 2021, pp. 97–134] →

[Sadek 1976] [Sadek 1976] K. Sadek, “Nachbildung einer Hochspannungs-Gleichstrom-Übertragung”, in [Schöne 1976/1, pp. 360–388] →

[Saleh et al. 2023] [Saleh et al. 2023] Saad Saleh, Boris Koldehofe, “The Future is Analog: Energy-Efficient Cognitive Network Functions over Memristor-Based Analog Computations”, in *HotNets'23*, November 28-29, Cambridge, Massachusetts, pp. 254–262 →

[Salvioni 1971] [Salvioni 1971] Luigi Salvioni, “Some Aspects of Flutter of Cylindrical Panels”, in *Meccanica*, September 1971, pp. 139–146 a, b, c

[Sankar et al. 1979] [Sankar et al. 1979] Seshadri Sankar, David R. Hargreaves, “Hybrid computer optimization of a class of impact absorbers”, in *SIMULATION*, July 1979, pp. 11–18 a, b

[Sankar et al. 1980] [Sankar et al. 1980] Seshadri Sankar, J. V. Svoboda, “Hybrid Computer in the Optimal Design of Hydro-Mechanical Systems”, in *Mathematics and Computers in Simulation*, XXII (1980), pp. 353–367 →

[Santen 1966] [Santen 1966] N. van Santen, “Enkele bouwkundige toepassingen van elektrische weerstandsmodellen”, in *TNO-Nieuws*, 21, 1966, pp. 192–196 →

[Sarpeshkar 1998] [Sarpeshkar 1998] Rahul Sarpeshkar, “Analog Versus Digital: Extrapolating from Electronics to Neurobiology”, in *Neural Computation*, 10, 1998, pp. 1601–1638 →

[Sauer et al. 1944] [Sauer et al. 1944] R. Sauer, H. Pösch, “Zur Theorie der Integriermaschine für gewöhnliche Differentialgleichungen”, in *Zeitschrift für angewandte Mathematik und Mechanik*, Band 24, Nr. 2, 1944, pp. 63–70 →

[Sauer] [Sauer] Albrecht Sauer, *Gezeiten – Ein Ausstellungsführer des Deutschen Schifffahrtsmuseums*, Deutsches Schifffahrtsmuseum, n. d. →

[Sauro 2019] [Sauro 2019] Herbert M. Sauro, *Enzyme Kinetics for Systems Biology*, Ambrosius Publishing, 2nd edition, 2019 a, b

[Savant et al. 1954] [Savant et al. 1954] C. J. Savant, R. C. Howard, “Multiplier for Analog Computers”, in *Electronics*, September, 1954, pp. 144–147 →

[Savet 1962] [Savet 1962] Paul Savet, “Heat-Transfer Computing Elements”, in [Huskey et al. 1962, pp. 8-18–8-22] →

[Sawhney et al. 2020] [Sawhney et al. 2020] Rohan Sawhney, Keenan Crane, “Monte Carlo Geometry Processing: A Grid-Free Approach to PDE-Based Methods on Volumetric Domains”, *ACM Trans. Graph.*, Vol 38, No. 4, Article 1, July 2020, pp. 1-1–1-18 →

[Sawhney et al. 2022] [Sawhney et al. 2022] Rohan Sawhney, Dario Seyb, Wojciech Jarosz, Keenan Crane, “Grid-Free Monte Carlo for PDEs with Spatially Varying Coefficients”, *ACM Trans. Graph.*, Vol. 41, No. 4, Article 53, July 2022, pp. 53-1–53-17 →

[Schloemann-Siemag 1978] [Schloemann-Siemag 1978] N. N., *Simulationstechnik im Schwermaschinenbau – Einsatz einer Analogrechenanlage für die Untersuchung und Berechnung von Maschinensystemen*, Sonderdruck der Schloemann-Siemag AG, 2/12.78 →

[Schlottmann et al. 2012] [Schlottmann et al. 2012] Craig R. Schlottmann, Samuel Shapero, Stephen Nease, Paul Hasler, “A Digitally Enhanced Dynamically Reconfigurable Analog Platform for Low-Power Signal Processing”, in *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 9, September 2012, pp. 2174–2184 →

[Schmid 1965] [Schmid 1965] Hermann Schmid, “Sequential analog-digital computer”, in *Proceedings of the November 30–December 1, 1965, Fall Joint Computer Conference*, Part 1,. pp. 915–928 →

[Schmidt 1956] [Schmidt 1956] W. Schmidt, “Die Hyperbelfeldröhre, eine Elektronenstrahlröhre zum Multiplizieren in Analogie-Rechengeräten”, in *Zeitschrift für angewandte Physik*, VIII. Band, Heft 2, 1956, pp. 69–75 →

[Schneider 1960] [Schneider 1960] G. Schneider, “Über die Nachbildung und Untersuchung von Abtastsystemen auf einem elektrischen Analogrechner”, in *Elektronische Rechenanlagen*, 2 (1960), No. 1, pp. 31–37 →

[Schöne 1976/1] [Schöne 1976/1] Armin Schöne, *Simulation Technischer Systeme*, Band 2, Carl Hanser Verlag München Wien, 1976

[Schöne 1976/2] [Schöne 1976/2] A. Schöne, “Modelle von Wärmetauschern”, in [Schöne 1976/1, pp. 7–27] →

[Schönfeld et al. 1967] [Schönfeld et al. 1967] E. Schönfeld, W. Merk, “Unfalldynamische Untersuchungen an Kühlkreisläufen schneller Brutreaktoren”, in [IACM 1976, pp. 1071–1078] →

[Schröder] [Schröder] H.-P. Schröder, *Programmsystem zur automatischen Normierung*, Technische Hochschule Ilmenau, Sektion Mathematik, Rechentechnik und Ökonomische Kybernetik, n. d. →

[Schüssler 1961] [Schüssler 1961] W. Schüssler, “Messung des Frequenzverhaltens linearer Schaltungen am Analogrechner”, in *Elektronische Rundschau*, No. 10, 1961, pp. 471–477 →

[Schultz et al.] [Schultz et al.] H. W. Schultz, J. F. Calvert, E. L. Buell, “The Photoformer in ANACOM Calculations”, n. d., author’s archive →

[Schultz et al. 1974] [Schultz et al. 1974] Harold M. Schultz, Roger K. Miyasaki, Thomas B. Liem, Richard A. Stanley, “Analog Simulation of Compressor Systems”, ISA CPD 74106, 1974 →

[Schwarz 1971] [Schwarz 1971] Wolfgang Schwarz, *Analogprogrammierung – Theorie und Praxis des Programmierens für Analogrechner*, VEB Fachbuchverlag Leipzig, 1. Ed., 1971 a, b, c, d

[Schwarze 1972] [Schwarze 1972] K. Schwarze, “Automatisches Skalieren und statischer Test mit dem hybriden Interpreter HOI”, in *Angewandte Informatik*, 3/72, pp. 127–138 →

[Schweizer 1976/1] [Schweizer 1976/1] G. Schweizer, “Beispiele für die Simulation von Luftfahrzeugen”, in [Schöne 1976/1, pp. 414–426] →

[Schweizer 1976/2] [Schweizer 1976/2] G. Schweizer, “Das mathematische Modell für die Echtzeitsimulation von Erdsatelliten”, in [Schöne 1976/1, pp. 434–453] →

[Schweizer 1976/3] [Schweizer 1976/3] G. Schweizer, “Das Systemglied „Mensch“ in der Simulation”, in [Schöne 1976/1, pp. 555–580] →

[Schweizer 1976/4] [Schweizer 1976/4] G. Schweizer, “Der Aufbau der Hybridsimulation für das Strahlflugzeug”, in [Schöne 1976/1, pp. 520–526] →

[Schweizer 1976/5] [Schweizer 1976/5] G. Schweizer, “Die Aufbereitung der Gleichungen des mathematischen Modells eines Flugzeugs zur Simulation auf dem Analogrechner”, in [Schöne 1976/1, pp. 515–519] →

[Schweizer 1976/6] [Schweizer 1976/6] G. Schweizer, “Die Sichtsimulation”, in [Schöne 1976/1, pp. 526–533] →

[Schweizer 1976/7] [Schweizer 1976/7] G. Schweizer, “Simulationsprobleme aus der Luft- und Raumfahrt”, in [Schöne 1976/1, pp. 389–398] a, b

[SDS 1969] [SDS 1969] N. N., *SDS 9300 Computer Reference Manual*, Scientific Data Systems, July 1969 →

[Scott 1958] [Scott 1958] Clyde C. Scott, “Power Reactor Control”, in *Instruments and Automation*, Vol. 31, April 1958, pp. 636–637 →

[Seay 1964] [Seay 1964] Albert Seay, “The Computer of Music and the Computer”, in *Computers and Automation*, 1964, Vol. 13, No. 8, pp. 16–18 →

[Sefcik et al. 1967] [Sefcik et al. 1967] J. Sefcik, J. Hrdlicka, “Simulation of Temperature and Concentration Profiles for Gas Shift Reactor”, in [IACM 1976, pp. 878-881] →

[Selfridge 1955] [Selfridge 1955] R. G. Selfridge, “Coding a general-purpose digital computer to operate as a differential analyzer”, in *Proceedings of the Western Joint Computer Conference*, The Institute of Radio Engineers, New York, 1955, S. 82–84 →

[Selvam et al. 1995] [Selvam et al. 1995] C. Selvam, V. Jagadeesh Kumar, “A Simple and Inexpensive Implementation of Time Division Multiplier for two Quadrant Operation”, in *IETE Technical Review*, Vol. 12, No. 1, January-February 1995, pp. 33–35 →

[Senoner 2025] [Senoner 2025] Mathias Senoner, “Wichtiger Hinweis: Nicht alles, was nicht digital ist, ist zwangsläufig analog!”, in *Berliner Zeitung*, →https://www.berliner-zeitung.de/open-source/vieles-was-als-analog-bezeichnet-wird-ist-es-gar-nicht-li.2303594, retrieved 02.06.2025 →

[Seyferth 1960] [Seyferth 1960] H. Seyferth, “Über die Behandlung partieller Differentialgleichungen auf dem elektronischen Analogrechner”, in *Elektronische Rechenanlagen*, 2 (1960), No. 2, p. 85–92 →

[Shannon 1941] [Shannon 1941] Claude E. Shannon, “Mathematical Theory of the Differential Analyzer”, in *Journal of Mathematics and Physics*, April 1941, pp. 337–354 →

[Sheingold 1978] [Sheingold 1978] D. H. Sheingold (ed.), *Multiplier Application Guide*, Analog Devices, 1978 a, b, c

[Shen 1970] [Shen 1970] Kuei Shen, *Analog Computer Simulation in Plasma Physics*, Thesis, University of Iowa, Electrical Engineering, 1970 a, b

[Shen et al. 1970] [Shen et al. 1970] Kuei Shen, E. D. Alton, H C. S. Hsuan, “Analog Computer Simulation in Plasma Physics”, in *American Journal of Physics*, 38, 1970, pp. 1133–1135 →

[Sherman et al. 1958] [Sherman et al. 1958] Windsor L. Sherman, Stanley Faber, James B. Witten, *Study of Exit Phase of Flight of a Very High Altitude Hypersonic Airplane by Means of a Pilot-Controlled Analog Computer*, Langley Aeronautical Laboratory, 1958 →

[Shileiko 1964] [Shileiko 1964] A. V. Shileiko, *Digital Differential Analysers*, Pergamon Press, The Macmillan Company, New York, 1964 →

[Shore 1977] [Shore 1977] Nigel Leslie Shore, *Hybrid Computer Simulation and On-Line Digital Computer Control of D. C. Link*, Thesis, Faculty of Engineering, University of London, Imperial College of Science and Technology, Department of Electrical Engineering, 1977 →

[Shrestha et al. 2022] [Shrestha et al. 2022] Amar Shrestha, Haowen Fang, Zaidao Mei, Daniel Patrick Rider, Qing Wu, Qinru Qiu, “A Survey on Neuromorphic Computing: Models and Hardware”, in *IEEE Circuits and Systems Magazine*, second quarter 2022, pp. 2–35 →

[Shute 1956] [Shute 1956] Nevil Shute, *Slide Tule – the Autobiography of an Engineer*, Readers Union, William Heinemann, London, 1956 →

[Sierck 1963] [Sierck 1963] Joachim Sierck, *Untersuchung der nach dem Rückmischprinzip aufgebauten Frequenzteiler-Schaltungen unter besonderer Berücksichtigung der anomalen Mischrückkopplung*, Dissertation an der Fakultät für Elektrotechnik der RWTH Aachen →

[Siegelmann 1995] [Siegelmann 1995] Hava T. Siegelmann, “Computation Beyond the Turing Limit”, in *Science*, Vol. 268, 28 April 1995, pp. 545–548 →

[Siegelmann 1999] [Siegelmann 1999] Hava T. Siegelmann, *Neural Networks and Analog Computation – Beyond the Turing Limit*, Birkhäuser Boston, Basel, Berlin, 1999 →

[Simanca et al. 2002] [Simanca et al. 2002] Santiago R. Simanca, Scott Sutherland, *Notes for MAT 331 – Mathematical Problem Solving with Computers*, The University at Stony Brook, →https://www.math.stonybrook.edu/~scott/Book331/331book.pdf, retrieved 20.06.2022 →

[Simons et al.] [Simons et al.] Fred O. Simons, Richard C. Harden, Sam J. Monte, *Perfected Analog/Hybrid Simulations of all Classes of Sampled-Data Systems*, n. d., author’s archive →

[Single et al. 1961] [Single et al. 1961] Charles H. Single, Edward M. Billinghurst, “Optimization of analog computer linear system dynamic characteristics”, in *Western Joint IRE-AIEE-ACM Computer Conference*, May 9–1, 1961, pp. 315–339 →

[Skramstad 1957] [Skramstad 1957] Harold K. Skramstad, “Some Simulation Problems Under Study at the National Bureau of Standards”, in [White Sands 1957, pp. 91–107] a, b, c, d

[Skramstad 1959] [Skramstad 1959] Harold K. Skramstad, “A Combined Analog-Digital Differential Analyzer”, in *Proc. EJCC*, Volume 16, December 1959, pp. 94–101 a, b

[Skramstad et al. 1961] [Skramstad et al. 1961] Harold K. Skramstad, Julius H. Wright, *Electronic Analogue Computers for Radioactive Fallout Prediction*, United States Patent 2998193, Aug. 29, 1961 a, b

[Smith et al. 1959] [Smith et al. 1959] F. Smith, W. D. T. Hicks, *The R. A. E. Electronic Simulator for Flutter Investigations in Six Degrees of*

Freedom or Less, Ministry of Technology, Aeronautical Research Council, London, Her Majesty's Stationary Office, 1959 a, b, c, d

[Snow 1930] [Snow 1930] L. T. Snow, *Planimeter*, United States Patent 718166, January 13, 1930 →

[Solar 1977/1] [Solar 1977/1] D. Solar, "Erzeugung stochastischer Signale beliebiger Verteilung am Analogrechner mit Hilfe von Rauschgeneratoren und Funktionsgebern", in [TU Wien 1977/1, pp. 25 ff.] →

[Solar 1977/2] [Solar 1977/2] D. Solar, "PACER 600A – AutoPatch", in [TU Wien 1977/3, pp. 4 ff.] →

[Solar 1977/3] [Solar 1977/3] D. Solar, "Einsatzmöglichkeiten eines Hybridsystems in Biologie und Medizin", in [TU Wien 1977/3, pp. 36 ff.] →

[Solar 1978] [Solar 1978] D. Solar, "HYSMPK – Programmpaket zur digitalen Unterstützung von Analog- und Hybridsimulationen mit automatischer On-Line-Skalierung", in [TU Wien 1978/1, pp. 3 ff.] →

[Somers 1980] [Somers 1980] Eric Somers, "Computer graphics for television", in *Video Systems*, June 1980, pp. 11–21 →

[Somerville 1950] [Somerville 1950] Alexander Somerville, "A Beam-Type Tube That Multiplies", in *Proc. Natl. Elec. Conf.*, 6, 1950, pp. 145–154 →

[Soni et al. 2017] [Soni et al. 2017] Jimmy Soni, Rob Goodman, *A Mind at Play – How Claude Shannon invented the Information Age*, Simon & Schuster Paperbacks, 2017 a, b

[Soroka 1962] [Soroka 1962] Walter W. Soroka, "Mechanical Analog Computers", in [Huskey et al. 1962, pp. 8-2–8-16] →

[Sorondo et al.] [Sorondo et al.] Victor J. Sorondo, George D. Wilson, *Hybrid Computer Simulation of a Circulating Water System*, n. d.,

author's archive →

[Soudack 1968] [Soudack 1968] A. C. Soudack, "Canonical Programming of Nonlinear and Time-Varying Differential Equations", in *IEEE Transactions on Computers*, Vol. C-17, No. 4, April, 1968, p. 402 →

[Spearman et al. 1955] [Spearman et al. 1955] F. R. F. Spearman, A. V. Hemingway, R. W. Hynes, "TRIDAC, a large analogue computing machine", in *Proceedings of the IEE – Part B: Radio and Electronic Engineering*, Volume 103, Issue 9, Oct. 1955, pp. 375–390 →

[Speckhart et al. 1976] [Speckhart et al. 1976] Frank H. Speckhart, Walter L. Green, *A Guide to Using CSMP – The Continuous System Modeling Program – A Program for Simulating Physical Systems*, Prentice-Hall, Inc., 1976 a, b, c, d, e, f

[Speek 2017] [Speek 2017] J. H. R. Speek, *Robot Mathematician Solves Nine Simultaneous Equations*, Bachelor's Project Mathematics, University of Groningen, 2017 →

[Spiess 1992] [Spiess 1992] Ray Spiess, "The Comdyna GP-6 Analog Computer – Twenty Five Years... and still counting", in *Simulation*, Volume 59, Issue 5, pp. 323–325 →

[Spiess 2005] [Spiess 2005] Ray Spiess, "The Comdyna GP-6 Analog Computer", in *IEEE Control Systems Magazine*, June 2005, pp. 68–73 a, b

[Splaver 1967] [Splaver 1967] Sarah Splaver, *Some Day I'll be an Aerospace Engineer*, Hawthorn, 1978 →

[Sprott 2016] [Sprott 2016] Julien Clinton Sprott, *Elegant Chaos – Algebraically Simple Chaotic Flows*, World Scientific, 2016 →

[SRC 2021] [SRC 2021] N. N., *Decadal Plan for Semiconductors – Full Report*, Semiconductor Research Corporation, 2021,

→https://www.src.org/about/decadal-plan/decadal-plan-full-report.pdf, retrieved 31.08.2025 a, b

[Stanley 1956] [Stanley 1956] Paul E. Stanley, “Computers and Engineering Education”, in *Computers and Automation*, 1956, Vol. 5, No. 2, pp. 10–12 →

[Starnick 1976] [Starnick 1976] J. Starnick, “Simulation chemischer Reaktoren”, in [Schöne 1976/1, pp. 51–205] →

[Stata 1968] [Stata 1968] Ray Stata, “Transconductance Analog Multiplier”, in *Instruments and Control Systems*, November 1968, p. 115 →

[Steele 1973] [Steele 1973] Floyd Steele, *Computer Oral History Collection*, Smithsonian National Museum of American History, January 16, 1973, →https://www.si.edu/media/NMAH/NMAH-AC0196_steele730116.pdf, retrieved 02.03.2025 →

[Stein et al. 1959] [Stein et al. 1959] Marvin L. Stein, J. Rose, D. B. Parker, “A Compiler with an Analog-Oriented Input Language”, in *IRE-AIEE-ACM ‘59 (Western): Papers presented at the March 3–5, 1959, western joint computer conference*, March 1959, pp. 92–102 →

[Stein et al. 1970] [Stein et al. 1970] Marvin L. Stein, E. James Mundstock, „Sorting Implicit Outputs in Digital Simulation“, in *IEEE Transactions on Computers*, Vol. C-19, No. 9, September 1970, S. 844–847 →

[Steinhoff et al. 1957] [Steinhoff et al. 1957] E. A. Steinhoff, M. C. Green, “Real-Time Flight Performance Analysis”, in [White Sands 1957, pp. 241-245] a, b

[Stepanow 1956] [Stepanow 1956] W. W. Stepanow, *Lehrbuch der Differentialgleichungen*, VEB Deutscher Verlag der Wissenschaften Berlin, 1956 →

[Stewart 1979] [Stewart 1979] Peter A. Stewart, “The Analog Computer as a Physiology Adjunct”, in *The Physiologist*, 1979, Issue 1, pp. 43–47 a, b

[Stieber 1967] [Stieber 1967] Manfred Stieber, “Nachbildung des Verhaltens Umrichtergespeister Asynchronmotoren auf dem Analogrechner”, in [IACM 1976, pp. 719–728] →

[Stillwell 1956] [Stillwell 1956] Wendell H. Stillwell, “Studies of Reaction Controls”, in *Control Studies, Part B, Studies of Reaction Controls*, NASA, Document ID 19930092438, 1956 →

[Stillwell et al. 1958] [Stillwell et al. 1958] Wendell H. Stillwell, Hubert M. Drake, “Simulator Studies of Jet Reaction Controls for Use at High Altitude”, NACA Research Memorandum, September 26, 1958 →

[Stine 2014] [Stine 2014] Kyle Stine, “The Coupling of Cinematics and Kinematics”, in *Grey Room*, 56, Summer 2014, pp 34–57 →

[Stone et al. 1965] [Stone et al. 1965] John Stone, Ka-Cheung Taui, Eugene Pack, “Computer Control Study for a Manned Centrifuge”, NASA Technical Report, F-B2300-1, June 1965 →

[Strotz et al. 1951] [Strotz et al. 1951] R. H. Strotz, J. F. Calvert, N. F. Morehouse, “Analogue Computing Techniques Applied to Economics”, *Transactions of the American Institute of Electrical Engineers*, 70(1), DOI: 10.1109/T-AIEE.1951.5060443, pp. 557–563 →

[Strotz et al. 1953] [Strotz et al. 1953] R. H. Strotz, J. C. McAnulty, J. B. Naines, “Goodwin’s Nonlinear Theory of the Business Cycle: An Electro-Analog Solution”, in *Econometrica*, 21(3), DOI: 10.2307/1905446, pp. 390–411 →

[Stubbs et al. 1954] [Stubbs et al. 1954] G. S. Stubbs, C. H. Single, *Transport Delay Simulation Circuits*, Westinghouse, Atomic Power Division, 1954 →

[Suryanarayanan et al. 1968] [Suryanarayanan et al. 1968] K. L. Suryanarayanan, A. C. Soudack, “Analog Computer Automatic Parameter Optimization of Nonlinear Control Systems with Specific Inputs”, in *IEEE Transactions on Computers*, Vol. C-17, No. 8, August, 1968, pp. 782–788 →

[Sutton et al. 1963] [Sutton et al. 1963] George H. Sutton, Paul W. Pomeroy, “Analog Analyses of Seismograms Recorded on Magnetic Tape”, in *Journal of Geophysical Research*, Vol. 68, No. 9, May 1, 1963, pp. 2791–2815 →

[Sutton et al. 2018] [Sutton et al. 2018] Richard S. Sutton, Andrew G. Barto, *Reinforcement Learning: An Introduction*, second edition, The MIT Press, 2018 →

[Svoboda 1948] [Svoboda 1948] Antonin Svoboda, *Computing Mechanisms and Linkages*, McGraw-Hill Book Company, Inc., 1948 a, b, c, d, e, f, g, h, i, j

[Swade 1995] [Swade 1995] Doron Swade, “The Phillips Economic Computer”, in *Resurrection – The Bulletin of the Computer Conservation Society*, Issue Number 12, Summer 1995, pp. 11–18 a, b, c, d

[Swartzel 1946] [Swartzel 1946] Karl D. Swartzel, *Summing Amplifier*, United States Patent 2401779, June 11, 1946 →

[Swithenbank 1960] [Swithenbank 1960] J. Swithenbank, *Design of an Electrical Analogue Computer to Simulate Regenerative Heat Exchangers*, Mechanical Engineering Research Laboratories, Report No. SCS 12, McGill University, Montreal, February 1960 →

[Sydow 1964] [Sydow 1964] Achim Sydow, *Programmierungstechnik für elektronische Analogrechner*, VEB Verlag Technik Berlin, 1964 a, b, c, d, e, f, g, h, i, j, k, l, m, n

[Szalai 1971] [Szalai 1971] Kenneth J. Szalai, “Validation of a general purpose airborne simulator for simulation of large transport

aircraft handling qualities", NASA TN D-6431, October 1971 →

[Szuch et al. 1965] [Szuch et al. 1965] John R. Szuch, Leon M. Wenzel, Robert J. Baumbick, *Investigation of the Starting Characteristics of the M-1 Rocket Engine Using the Analog Computer*, NASA Technical Note TN D-3136, December 1965 →

[T-2 1946] [T-2 1946] N. N., *Bibliography on German Guided Missiles*, Technical Intelligence T-2 AMC, Headquarters Air Materiel Command, Wright Field, Dayton, Ohio, 1946 →

[Tabbutt 1967] [Tabbutt 1967] Frederick D. Tabbutt, "The Use of Analog Computers for Teaching Chemistry", in *Journal of Chemical Education*, Volume 44, Number 2, February 1967, pp. 64–69 →

[Tabbutt 1969] [Tabbutt 1969] Frederick D. Tabbutt, "A New Concept in Graphic Computer Instruction", in *Reed College Science Journal*, Spring 1969, Volume 1, Number 1, pp. 50–53 →

[Tacker et al.] [Tacker et al.] Edgar C. Tacker, Thomas D. Linton, *Hybrid Simulation of an Optimal Stochastic Control System*, Louisiana State University, n. d., author's archive →

[Takaishi 1965] [Takaishi 1965] Yoshifumi Takaishi, *Untersuchungen über den Einfluß der Tanklage und nichtlinearer Dämpfungseffekte auf die Wirkung eines Schlingertanks*, in *Schriftenreihe Schiffbau*, Technische Universität Hamburg-Harburg, 155, 1965 →

[Tasini 1957] [Tasini 1957] Betsalel Tasini, *An Investigation of a Criterion for the Optimum Transient Response of Servomechanisms*, master thesis, The Rice Institute, Houston, Texas, May 1957 a, b

[Taylor 1957] [Taylor 1957] C. H. Taylor, Jr., "A Coordinated Data-Processing System and Analog Computer to Determine Refinery-Process Operating Guides", in *Proceedings of the Eastern Computer Conference*, December 9–13, 1957, pp. 34–39 →

[Telefunken 1958] [Telefunken 1958] N. N., *Elektronischer Analogrechner RA 463/2*, Telefunken, AH 5.2 Apr. 58 a, b

[Telefunken/1] [Telefunken/1] N. N., *Demonstrationsbeispiel Nr. 5, Ball im Kasten*, AEG Telefunken, n. d. →

[Telefunken/2] [Telefunken/2] N. N., *Schwingungsberechnung eines Zwei-Massen-Systems*, AEG-Telefunken, n. d. →

[Telefunken/3] [Telefunken/3] N. N., *Perspektivische Darstellung von Rechenergebnissen mit Hilfe eines Analogrechners*, AEG-TELEFUNKEN Datenverarbeitung, n. d. a, b

[Telefunken/4] [Telefunken/4] N. N., *Darstellung von Tragflügeln und ihren Stromlinien mit einem Analogrechner*, AEG Telefunken, n. d. →

[Telefunken/5] [Telefunken/5] N. N., *Kepler und die Atomphysik – Der Beschuß eines Atomkerns mit Alphateilchen auf einem Tischanalogrechner*, Demonstrationsbeispiel 3, AEG-Telefunken, ADB 003 0570, n. d. a, b, c, d

[Telefunken/6] [Telefunken/6] N. N., *Steuermanöver eines Satelliten*, AEG-Telefunken, AB 009/10 70, n. d. a, b, c

[Telefunken/7] [Telefunken/7] N. N., *Echo I – Simulation der Umlaufbahn auf dem Analogrechner*, AEG-Telefunken, ADB 004 0770, n. d. →

[Telefunken/8] [Telefunken/8] N. N., *Hybrider Präzisionsanalogrechner RA 770 – Kurzbeschreibung*, Telefunken, n. d. a, b

[Telefunken/9] [Telefunken/9] N. N., *Tischanalogrechner und Zusatzgeräte – Kurzbeschreibung*, Telefunken, n. d. →

[Telefunken/10] [Telefunken/10] N. N., *Digitalzusatz DEX 100 – Beschreibung*, Telefunken, n. d. →

[Telefunken 1958] [Telefunken 1958] N. N., *Angebot Nr. 557/0010 der Telefunken GmbH an die Technische Hochschule München*, 1958, author's archive

[Telefunken 1963/1] [Telefunken 1963/1] N. N., *Anwendungsbeispiele für Analogrechner – Wärmeleitung*, Telefunken, 15. Oktober 1963 a, b

[Telefunken 1963/2] [Telefunken 1963/2] N. N., *Anwendungsbeispiele für Analogrechner – Transformator*, Telefunken, 15. Oktober 1963 →

[Telefunken 1966] [Telefunken 1966] N. N., *Demonstrationsbeispiele für Analogrechner RAT 740, Beispiel 2, Einfache Darstellung einer Planetenbahn*, Mitteilungen der Fachabteilung Analogrechner, TELEFUNKEN, 10.1.66 →

[Teuber 1964] [Teuber 1964] D. L. Teuber, *Nachbildung der Saturn V-Rakete auf elektronischen Analogrechnern*, Tagungsberichte Hermann Oberth-Gesellschaft, 13. Raketen- und Raumfahrttagung vom 25.–28. Juni 1964 in Darmstadt a, b

[Thaler et al. 1980] [Thaler et al. 1980] G. J. Thaler, T. S. Nelson III, A. Gerba, "Real Time, Man-Interfaced Motion Analysis of a 3000 Ton Surface Effect Ship", in *Summer Computer Simulation Conference*, 1980, pp. 593–598 →

[The Times 1954] [The Times 1954] N. N., "The Royal Aircraft Establishment's analogue computer… ", in *The Times*, October 8, 1954 →

[Thomas 1968] [Thomas 1968] Charles H. Thomas, "Transport Time-Delay Simulation for Transmission Line Representation", in *IEEE Transactions on Computers*, Vol. C-17, No. 3, March, 1968, pp. 205–214 →

[Thomas et al.] [Thomas et al.] Charles H. Thomas, A. E. Kilgour, D. H. Welle, T. A. Karnowski, *Transient Performance Study of a Parallel HV*

and EHV Transmission System, n. d., author's archive →

[Thomas et al. 1967] [Thomas et al. 1967] Y. Thomas, J. F. le Corre, H. Apter, "Application du Principe du Maximum á la Determination sur Calculateur Analogique d'une Commande Optimale de Chaudiere", in [IACM 1976, pp. 855–862] →

[Thomas et al. 1969] [Thomas et al. 1969] Charles H. Thomas, Ronald A. Hedin, "Switching Surges on Transmission Lines Studied by Differential Analyzer Simulation", in *IEEE Transactions on Power Apparatur and Systems*, Vol. PAS-88, No. 5, May 1969 a, b

[Thompson] [Thompson] R. V. Thompson, *Application of Hybrid Computer Simulation Techniques to Warship Propulsion Machinery Systems Design*, n. d., author's archive →

[Thomson 1876] [Thomson 1876] Sir William Thomson, "Mechanical integration of linear differential equations of the second order with variable coefficients", in *Proceedings of the Royal Society*, Volume 24, No. 167, 1876, pp. 269–270, →

[Thomson 1878] [Thomson 1878] Sir William Thomson, "Harmonic Analyzer", in *Proceedings of the Royal Society of London*, Vol. 27, 1878, pp. 371–373 a, b

[Thomson 1882] [Thomson 1882] Sir William Thomson, "Tides", *Evening Lecture To The British Association At The Southhampton Meeting*, source: →http://www.fordham.edu/halsall/mod/1882kelvin-tides.html, retrieved 24.11.2012 →

[Thomson 1911] [Thomson 1911] Sir William Thomson, "The tidal gauge, tidal harmonic analyser, and tide predicter", in *Kelvin, Mathematical and Physical Papers*, Volume VI, Cambridge 1911, pp. 272–305 →

[Thomson 1912] [Thomson 1912] James Thomson, *Collected Papers in Physics and Engineering – selected and arragned with unpublished*

material and annotations by Sir Joseph Larmor and James Thomson, Cambridge at the University Press, 1912 →

[Thun] [Thun] H. J. von Thun, *Simulation einer lagegeregelten Radiostern-Antenne auf dem hybriden Analogrechner*, BBC-Mannheim, Zentrale Entwicklung für Elektronik, Abteilung Systemtechnik, n. d. →

[Thwaites ed. 1987] [Thwaites ed. 1987] Bryan Thwaites (ed.), *Incompressible Aerodynamics – An Account of the Theory and Observation of the Steady Flow of Incompressible Fluid past Aerofoils, Wings, and Other Bodies*, Dover Publications, Inc., New York, 1987 →

[Tisdale] [Tisdale] Henry F. Tisdale, *How a Modern Analog Computer Duplicates Real-World Behaviour of a Thyristor-Controlled Rectifier Bridge*, Electronic Associates, Inc., n. d., author's archive →

[Tisdale 1981] [Tisdale 1981] Henry F. Tisdale, "Hybrid Computers Retaining Favor with Controls & Design Engineers", in *EAI OPENERS*, Product Information Bulletin #048 – May, 1981 a, b, c

[Titchener et al. 1983] [Titchener et al. 1983] M. R. Titchener, R. M. Stimpfle, "Digital Emulation of Analog Computer Techniques for the Solution of Kinetic Systems", in *Journal of Computational Chemistry*, Vol. 4, No. 1, 1983, pp. 58–67 →

[Tomayko 1985] [Tomayko 1985] James E. Tomayko, "Helmut Hoelzer's Fully Electronic Analog Computer", in *Annals of the History of Computing*, Volume 7, Number 3, July 1985, pp. 227–240 a, b, c, d, e

[Tomayko 2000] [Tomayko 2000] James E. Tomayko, *Computers Take Flight – a History of NASA's Pioneering Digital Fly-By-Wire Project*, NASA SP-2000-4224, 2000 a, b

[Tomovic et al. 1961] [Tomovic et al. 1961] R. Tomovic, W. J. Karplus, "Land Locomotion-Simulation and Control", in *Proc. 3rd AICA*

Conference on Analog Computation, Opatija, Yugoslavia, 1961, pp. 385–390 →

[Travers et al. 1930] [Travers et al. 1930] H. A. Travers, W. W. Parker, “An Alternating-Current Calculating Board”, in *Electric Journal*, Vol. 27, May 1930, pp. 266–270 a, b

[Trenkle 1982] [Trenkle 1982] Fritz Trenkle, *Die deutschen Funklenkverfahren bis 1945*, AEG-Telefunken Aktiengesellschaft, 1982, Anlagentechnik, Geschäftsbereich Hochfrequenztechnik →

[Troch 1977] [Troch 1977] Inge Troch, “Eine neue Methode der Parameteroptimierung mit Anwendung auf Randwertaufgaben”, in *14. seminář MEDA ANALOGOVÁ A HYBRIDNÍ VÝPOČETNÍ TECHNIKA*, Praha 1977, pp. 121–139 →

[Trubert 1968] [Trubert 1968] Marc R. Trubert, *Use of Analog Computer for the Equalization of Electromagnetic Shakers in Transient Testing*, Jet Propulsion Laboratory, January 1, 1968 →

[Truitt et al. 1960] [Truitt et al. 1960] Thos. D. Truitt, A. E. Rogers, *Basics of Analog Computers*, John F. Rider Publisher, Inc., New York, December 1960 a, b, c, d, e

[Tse et al. 1964] [Tse et al. 1964] Francis S. Tse, Ivan E. Morse, Rolland T. Hinkle, *Mechanical Vibrations*, Allyn and Bacon, Inc., Boston, Second Printing, August 1964 a, b

[Tucker 2002] [Tucker 2002] Warwick Tucker, “A Rigorous ODE Solver and Smale’s 14th Problem”, in *Foundations of Computational Mathematics* (2002) 2, pp. 53–117 →

[Turing 1937] [Turing 1937] Alan Mathison Turing, “On Computable Numbers, with an Application to the Entscheidungsproblem”, in *Proceedings of the London Mathematical Society*, Nr. 1, 1937, pp. 2–42 →

[Turner 1960] [Turner 1960] Robert M. Turner, “On the reduction of error in certain analog computer calculations by the use of constraint equations”, in *IRE-AIEE-ACM ’60 (Western)*, 1960, pp. 173–180 →

[TU Wien 1974] [TU Wien 1974] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 1, Oktober 1974

[TU Wien 1975/1] [TU Wien 1975/1] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 2, Februar 1975

[TU Wien 1975/2] [TU Wien 1975/2] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 3, Mai 1975

[TU Wien 1976/1] [TU Wien 1976/1] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 5, März 1976

[TU Wien 1976/2] [TU Wien 1976/2] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 6, Juni 1976

[TU Wien 1976/3] [TU Wien 1976/3] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 7, Oktober 1976 →

[TU Wien 1977/1] [TU Wien 1977/1] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des

Rechenzentrums der Technischen Universität Wien, Nummer 8, Februar 1977

[TU Wien 1977/2] [TU Wien 1977/2] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 9, Juni 1977

[TU Wien 1977/3] [TU Wien 1977/3] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 10, November 1977 →

[TU Wien 1978/1] [TU Wien 1978/1] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 11, April 1978

[TU Wien 1978/2] [TU Wien 1978/2] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 12, November 1978

[TU Wien 1979/1] [TU Wien 1979/1] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 13, April 1979

[TU Wien 1979/2] [TU Wien 1979/2] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 14, Dezember 1979

[TU Wien 1985] [TU Wien 1985] TU Wien, *Interface*, herausgegeben von der Abt. Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien, Nummer 22, Juni 1985 →

[Tympas 2003] [Tympas 2003] Aristotle Tympas, 'Perpetually Laborious: Computing Electric Power Transmission Before The Electronic Computer", in *IRSH* 48, 2003, Supplement, pp. 73–95, DOI: 10.1017/S0020859003001275 a, b, c

[Tyror et al. 1970] [Tyror et al. 1970] J. G. Tyror, R. I Vaughan, *An Introduction to the Neutron Kinetics of Nuclear Power Reactors*, Pergamon Press, 1970 →

[Ulmann 2014] [Ulmann 2014] Bernd Ulmann, *AN/FSQ-7: the computer that shaped the Cold War*, DeGrouyter / Oldenbourg, 2014 →

[Ulmann 2016] [Ulmann 2016] Bernd Ulmann, *The Lorenz-attractor*, Analog Computer Applications, →https://analogparadigm.com/downloads/alpaca_2.pdf, retrieved 14.01.2021 →

[Ulmann 2017] [Ulmann 2017] Bernd Ulmann, *Celestial mechanics: Three-body problem*, Analog Computer Applications, →https://analogparadigm.com/downloads/alpaca_11.pdf, retrieved 14.01.2021 →

[Ulmann 2019] [Ulmann 2019] Bernd Ulmann, *Solving the Schrödinger equation*, Analog Computer Applications, →https://analogparadigm.com/downloads/alpaca_22.pdf, retrieved 14.01.2021 →

[Ulmann et al. 2019] [Ulmann et al. 2019] Bernd Ulmann, Dirk Killat, "Solving systems of linear equations on analog computers", IEEE Xplore, Kleinheubach Conference, 2019, pp. 1–4 →

[Ulmann 2020/1] [Ulmann 2020/1] Bernd Ulmann, *A passive network for solving the two-dimensional heat equation*, Analog Computer Applications, →https://analogparadigm.com/downloads/alpaca_25.pdf, retrieved 10.08.2022 →

[Ulmann 2020/2] [Ulmann 2020/2] Bernd Ulmann, *Simulating the flight of a glider*, Analog Computer Applications, →https://analogparadigm.com/downloads/alpaca_26.pdf, retrieved 14.01.2021 →

[Ulmann 2021/1] [Ulmann 2021/1] Bernd Ulmann, *The Hindmarsh-Rose model of neuronal bursting*, Analog Computer Applications, →https://analogparadigm.com/downloads/alpaca_28.pdf, retrieved 14.01.2021 →

[Ulmann 2021/2] [Ulmann 2021/2] Bernd Ulmann, *Neutron kinetics*, Analog Computer Applications, →https://analogparadigm.com/downloads/alpaca_30.pdf, retrieved 14.01.2021 →

[Ulmann 2021/3] [Ulmann 2021/3] Bernd Ulmann, *The exponentially-mapped-past approach*, Analog Computer Applications, →https://analogparadigm.com/downloads/alpaca_32.pdf, retrieved 14.01.2021 →

[Ulmann 2023/1] [Ulmann 2023/1] Bernd Ulmann, *Analog and Hybrid Computer Programming*, 2nd edition, DeGruyter, 2023 a, b, c, d, e, f

[Ulmann 2023/2] [Ulmann 2023/2] Bernd Ulmann, *Experiments with Hindmarsh-Rose neurons*, Analog Computer Applications, →https://analogparadigm.com/downloads/alpaca_41.pdf, retrieved 31.03.2025 →

[Ulmann 2024] [Ulmann 2024] Bernd Ulmann, *Stochastic DEQs and hybrid computing*, Analog Computer Applications, →https://analogparadigm.com/downloads/alpaca_50.pdf, retrieved 02.08.2025 →

[Unbehauen 1967] [Unbehauen 1967] Heinz Unbehauen, “Nachbildung der Frischdampftemperaturregelung eines Dampferzeugers am

Analogrechner unter Berücksichtigung der Koppelung mit anderen Regelkreisen", in [IACM 1976, pp. 825–833] →

[University Bulletin 1978] [University Bulletin 1978] N. N. "Pioneer Computer Goes To Washington", in *University Bulletin*, Volume 26, Number 13, January 23, 1978, p. 1 →

[US Airforce 1957] [US Airforce 1957] N. N., *Handbook – Operation and Service Instructions, Simulator Trainer C-118A Aircraft*, T.O. 43D3-3-9-1, US Air Force, 30 August 1957 →

[US Army 1956] [US Army 1956] N. N., *Nike I Systems – Nike I Computer, SAM Problem Analysis, Servo Loop Elements and Power Distribution*, TM9-5000-13, Department of the Army, May 1956 a, b

[Ushakov 1958/1] [Ushakov 1958/1] V. B. Ushakov, "Soviet Trends in Computers for Control of Manufacturing Processes", in *Instruments and Automation*, November 1958, pp. 1810–1813 a, b, c, d

[Ushakov 1958/2] [Ushakov 1958/2] V. B. Ushakov, "Soviet Trends in Computers for Control of Manufacturing Processes", in *Instruments and Automation*, December 1958, pp. 1960–1961 a, b, c

[U.S.NRC] [U.S.NRC] N. N., *Fermi – Unit 1*, →https://www.nrc.gov/info-finder/decommissioning/power-reactor/enrico-fermi-atomic-power-plant-unit-1.html, retrieved 22.08.2025 →

[Valentin 2003] [Valentin 2003] Franz Valentin, *Hydraulik II – Angewandte Hydromechanik*, Skript des Lehrstuhles für Hydraulik und Gewässerkunde der Technischen Universität München, Oktober 2003 →

[Valisalo et al. 1967] [Valisalo et al. 1967] E. Valisalo, J. Lima, "A Hybrid Computer Simulation of the Economics of a Phosphate-

Chemical Plant", in [IACM 1976, pp. 863–877] a, b

[Valisalo et al. 1982] [Valisalo et al. 1982] P. E. Valisalo, D. Bergquist, V. McGrew, *A Hybrid Computer Algorithm for Temperature Distribution Analysis of Irregular Two Dimensional Shapes*, in *10th IMACS World Congress on System Simulation and Scientific Computation, 1982*, pp. 17–19 →

[Valisalo et al.] [Valisalo et al.] P. E. Valisalo, J. K. LeGro, "Hybrid Computer Utilization for System Optimization", author's archive →

[Veen 1947/1] [Veen 1947/1] Johan van Veen, "Analogy between Tides and A. C. Electricity (No. I)", in *The Engineer*, Nov. 28, 1947, pp. 498–500 →

[Veen 1947/2] [Veen 1947/2] Johan van Veen, "Analogy between Tides and A. C. Electricity (No. II)", in *The Engineer*, Dec. 5, 1947, pp. 520–521 →

[Veen 1947/3] [Veen 1947/3] Johan van Veen, "Analogy between Tides and A. C. Electricity (No. III)", in *The Engineer*, Dec. 12, 1947, pp. 544–545 →

[Vichik 2015] [Vichik 2015] Sergey Vichik, *Quadratic and linear optimization with analog circuits*, dissertation, University of California, Berkeley, 2015 →

[Vichnevetsky 1969] [Vichnevetsky 1969] Robert Vichnevetsky, "Use of Functional Approximation Methods in the Computer Solution of Initial Value Partial Differential Equation Problems", in *IEEE Transactions on Computers*, Vol. C-18, No. 6, June 1969, pp. 499–512 →

[Vocolides 1960] [Vocolides 1960] J. Vocolides, "Über die Behandlung linearer algebraischer Gleichungssysteme mit Analogrechnern", in *Elektronische Rechenanlagen*, 2 (1960), No. 3, pp. 136–141 →

[Vogel 1977] [Vogel 1977] Fritz Vogel, *Ein elektronischer Koordinatenwandler ohne Diodennetzwerke und seine Anwendung bei der Meßwertverarbeitung mechanischer Größen*, Dissertation, Technische Universität Wien, September 1977 →

[Volder 1959] [Volder 1959] Jack E. Volder, "The CORDIC Trigonometric Computing Technique", in *IRE Trans. Electron. Comput.*, EC-8, 1959, pp. 330–334 →

[Volynskii et al. 1965] [Volynskii et al. 1965] B. A. Volynskii, V. Ye. Bukhman, *Analogues for the Solution of Boundary-Value Problems*, Pergamon Press, 1965 →

[Wadel et al. 1955] [Wadel et al. 1955] Louis B. Wadel, C. C. Wan, "Aircraft Performance Studied on an Electronic Analog Computer", in *Western Joint Computer Conference*, 1955, pp. 78–82 →

[Wadel 1956] [Wadel 1956] Louis B. Wadel, "Simulation of Digital Filters on an Electronic Analog Computer", in *Journal of the ACM*, Volume 3, Issue 1, Jan. 1956, pp. 16–21 →

[Wagner 1972] [Wagner 1972] Manfred Wagner, *Analogrechner in der Verfahrenstechnik*, VEB Deutscher Verlag für Grundstoffindustrie, Leipzig, 1972 →

[Wait 1963] [Wait 1963] John V. Wait, "A hybrid analog-digital differential analyzer system", in *AFIPS '63 (Fall): Proceedings of the November 12–14, 1963, fall joint computer conference*, November 1963, pp. 277–298 →

[Waldschmidt 1970] [Waldschmidt 1970] Klaus Waldschmidt, *Hochgenaue Hybride Komponenten für Nichtlineare Rechenoperationen*, Dissertation, Universität Berlin, 1970 →

[Walther et al. 1949] [Walther et al. 1949] Alwin Walther, Hans-Joachim Dreyer, "Die Integrieranlage IPM-Ott für gewöhnliche

Differentialgleichungen", in *Die Naturwissenschaften*, No. 7, 1949, pp. 199–206 a, b, c

[Waltman 2000] [Waltman 2000] Gene L. Waltman, *Black Magic and Gremlins: Analog Flight Simulation at NASA's Flight Research Center*, NASA History Division, Monographs in Aerospace History, Number 20, 2000 a, b, c, d, e, f, g, h, i, j, k

[Warschawski 1945] [Warschawski 1945] S. E. Warschawski, "On Theodorsen's Method of Conformal Mapping of Nearly Circular Regions", in *Quarterly of Applied Mathematics*, April, 1945, Vol. 3, No. 1, pp. 12–28 →

[Washburn 1962] [Washburn 1962] R. P. Washburn, "Economic-Dispatch Computers for Power Systems", in [Huskey et al. 1962, pp. 5-155–5-160] a, b

[Wass 1955] [Wass 1955] C. A. A. Wass, *Introduction to Electronic Analogue Computers*, London, Pergamon Press Ltd., 1955 a, b, c, d, e, f, g, h, i

[Wauve 1962] [Wauve 1962] Armand Van Wauve, *Automatic Analog Computer Control by Means of Punched Cards – "CRESSIDA I"*, National Aeronautics and Space Administration, Washington, D. C., April 1962 →

[Welle 1972] [Welle 1972] D. H. Welle, R. A. Hedin, R. W. Weishaupt, C. H. Thomas, "Switching Surges on Parallel HV and EHV Untransposed Transmission Lines Studied by Analog Simulation", in *IEEE Transactions on Power Apparatus and Systems*, Volume PAS-91, Issue 1, Januar 1972, pp. 180–189 →

[Weinberg et al. 1958] [Weinberg et al. 1958] Alvin M. Weinberg, Eugene P. Wigner, *The Physical Theory of Neutron Chain Reactors*, The University of Chicago Press, 1958 →

[Weiss 1976] [Weiss 1976] H. H. Weiss, "Die stationäre laminare inkompressible Grenzschicht im Einlauf schwach gekrümmter

Rohre beliebigen Querschnitts", in [TU Wien 1976/2, pp. 34 f.] →

[Weitner 1955] [Weitner 1955] G. Weitner, "Grundschaltungen elektronischer Regler mit Rückführung", in *Elektronische Rundschau*, September 1955, Vol. 9, No. 9, pp. 320–323 →

[Werner 1968] [Werner 1968] Ernst Werner, "Guter Rat vom Netzmodell – Analogrechner überprüft erneut das Gasnetz von Wels", in *Energiewirtschaftliche Tagesfragen – Zeitschrift für die Elektrizitäts- und Gasversorgung*, Heft Nr. 3, März 1968, pp. 125–126 →

[Werrell 1985] [Werrell 1985] Kenneth P. Werrell, *The Evolution of the Cruise Missile*, Air University Press, Maxwell Air Force Base, Alabama, September 1985 →

[White 1966] [White 1966] M. E. White, "An Analog Computer Technique for Solving a Class of Nonlinear Ordinary Differential Equations", in *IEEE Transactions on Electronic Computers*, Vol. EC-15, No. 2, April 1966, pp. 157–163 →

[White Sands 1957] [White Sands 1957] White Sands Proving Ground, *Proceedings of First Flight Simulation Symposium, November 1956*, WSPG Special Report 9, September 1957

[Whitesell et al. 1969] [Whitesell et al. 1969] L. G. Whitesell, E. H. Bowles, "Train Power Station Operators by Analog", in [Hydrocarbon Processing 1959, pp. 84–86] a, b

[Widrow 1960] [Widrow 1960] B. Widrow, *An Adaptive "Adaline" Neuron Using Chemical "Memistors"*, technical report no. 1553-2, Solid-State Electronics Laboratory, Stanford University, California, October 17, 1960 →

[Wierwille et al. 1968] [Wierwille et al. 1968] Walter W. Wierwille, James R. Knight, "Off-Line Correlation Analysis of Nonstationary Signals", in *IEEE Transactions on Computers*, Vol. C-17, No. 5, May, 1968, pp. 525–536 →

[Wilbur 1936] [Wilbur 1936] John B. Wilbur, “The Mechanical Solution of Simultaneous Equations”, in *Journal of the Franklink Institute*, Volume 222, Issue 6, December 1936, pp. 715–724 →

[Williams et al. 1958] [Williams et al. 1958] Theodore J. Williams, R. Curtis Johnson, Arthur Rose, “Computers in the Process Industries”, in *Instruments and Automation*, Vol. 31, January 1958, pp. 90–94 →

[Willers 1943] [Willers 1943] Friedrich Adolf Willers, *Mathematische Instrumente*, Verlag von R. Oldenbourg, München und Berlin 1943 a, b, c, d, e, f, g, h

[Wilson 1966] [Wilson 1966] A. R. Wilson, *Preliminary Design Specifications for the Compact Weapon Effects Display System*, EG&G Inc., EGG-1183-1257, Report No. L-803, 15 August 1966 a, b, c

[Winarno 1982] [Winarno 1982] H. Winarno, J. Jalade, J. P. Gouyon, “Hybrid Simulation of a Microprocessor Controlled Multi-Convertor”, 10th IMACS World Congress on System Simulation and Scientific Computation, 1982, pp. 23–25 →

[Winkler 1961] [Winkler 1961] Helmut Winkler, *Elektronische Analogieanlagen*, Akademie-Verlag Berlin, 1961 a, b

[Winstead 2019] [Winstead 2019] Chris Winstead, “Tutorial on Stochastic Computing”, in [Gross et al. 2019, pp.39–76] a, b, c, d

[Wise et al. 1967] [Wise et al. 1967] H. E. Wise, R. J. Price, D. W. T. Rippin, “A Comparison of the use of Pseudo-Random Binary Sequences and Frequency Response Methods for the Determination of the Dynamics of a Pilot Scale Water Gas Shift Reactor“, in [IACM 1976, pp. 882–893] a, b

[Witsenhausen 1962] [Witsenhausen 1962] Hans S. Witsenhausen, “Hybrid techniques applied to optimization problems”, in *AIEE-*

IRE, ’62 (Spring), Proceedings of the May 1–3, 1962, spring joint computer conference, pp. 377–392 →

[Woods 2008] [Woods 2008] W. David Woods, *How Apollo Flew to the Moon*, Springer, Praxis Publishing Ltd., 2008 →

[Worly 1956] [Worly 1956] Charles W. Worley, “Simulation of a Liquid Level Control System with an Analog Computer”, in *Computers and Automation*, 1957, Vol. 6, No. 2, pp. 30–31 →

[Worley 1962] [Worley 1962] Charles W. Worley, “Process-Control Applications”, in [Huskey et al. 1962, pp. 5-71–5-86] a, b, c

[Woźniakowski 1977] [Woźniakowski 1977] Miroslav Woźniakowski, “Hybrid and Digital Simulation and Optimization of Dynamic Systems”, in *14. seminář MEDA ANALOGOVÁ A HYBRIDNÍ VÝPOČETNÍ TECHNIKA*, Praha 1977, pp. 159–163 →

[Wright et al. 1957] [Wright et al. 1957] J. H. Wright, L. Taback, H. K. Skramstad, “A High-Speed Computer for Predicting Radioactive Fallout”, in *Journal of Research of the National Bureau of Standards*, Vol. 58, No. 2, February 1957, Research Paper 2740, pp. 101–109 →

[Wright 1957] [Wright 1957] J. Howard Wright, “Fallout Computer – some Technical Corrections”, in *Computers and Automation*, 1957, Vol. 6, No. 10, p. 23 →

[Yanowitch 1956] [Yanowitch 1956] M. Yanowitch, “The Solution of Boundary Value Problems on a REAC Analog Computer”, in *Computers and Automation*, 1957, Vol. 6, No. 2, pp. 26–29 →

[Zachary 1999] [Zachary 1999] G. Pascal Zachary, *Endless Frontier – Vannevar Bush, Engineer of the American Century*, The MIT Press, 1999 a, b, c, d, e, f

[Zaroodny 1955] [Zaroodny 1955] S. J. Zaroodny, *An Elementary Review of the Mathieu-Hill Equation of Real Variable Based on*

Numerical Solutions, Department of the Army, Project No. 503-03-001, Aberdeen Proving Ground, Maryland, April 1955 →

[Zazula et al. 2013] [Zazula et al. 2013] Adam Zazula, Dariusz Myszor, Oleg Antemijczuk, Krzysztof A. Cyran, "Flight Simulators – from Electromechanical Analogue Computers to Modern Laboratory of Flying", in *Advances in Science and Technology*, Volume 7, Issue 17, March 2013, pp. 51–55 →

[Zhang 2015] [Zhang 2015] Wei-Bin Zhang, *Differential Equations, Bifurcations, and Chaos in Economics*, World Scientific Publishing Company, 2015 →

[Zhao et al. 2019] [Zhao et al. 2019] Zhou Zhao, Ashok Srivastava, Lu Peng, Qing Chen, "Long Short-Ter Memory Network Design for Analog Computing", in *ACM Journal on Emerging Technologies in Computing Systems*, Volume 15, Issue 1, January 2019, Article No.: 13, pp. 1–27 →

[Zheng et al. 2024] [Zheng et al. 2024] Qilin Zheng, Ziru Li, Jonathan Ku, Yitu Wang, Brady Taylor, Deliang Fan, Yiran Chen, "Improving the Efficiency of In-Memory-Computing Macro with a Hybrid Analog-Digital Computing Mode for Lossless Neural Network Inference", in *DAC '24: Proceedings of the 61st ACM/IEEE Design Automation Conference*, Article No. 313, pp. 1–6 →

[Zimdahl 1965] [Zimdahl 1965] W. Zimdahl, "Führungsverhalten des vierrädrigen Straßenfahrzeugs bei Regelung des Kurses auf festgelegter Bahn", in *Regelungstechnik – Zeitschrift für Steuern, Regeln und Automatisieren*, 13. Jahrgang 1965, Heft 5, pp. 221–226 →

[Zingg 1989] [Zingg 1989] David W. Zingg, *Low Mach Number Euler Computations*, NASA Technical Memorandum 102205 →

[Zoberbier 1968] [Zoberbier 1968] W. Zoberbier, "Die Funktionsgleichungen des digitalen Integrators", in *Elektronische*

Rechenanlagen, 10 (1968), Heft 5, pp. 234–242 →

[Zorpette 1989] [Zorpette 1989] Glenn Zorpette, “Parkinson’s gun director”, in *IEEE Spectrum*, April 1989, p. 43 a, b

[ZSMR 1963] [ZSMR 1963] N. N., “Elektromechanisches Rechengerät des UEB-Systems”, in *Zsmr*, 6, H. 2, 1963, pp. 51–52 →

[Zuse Z80 1961] [Zuse Z80 1961] N. N., “ZUSE Z 80 – Ein lochendes und druckendes Transistorzählwerk”, in *mtw – Zeitschrift für moderne Rechentechnik und Automation*, 1/61, p. 33 →

[Zyl 1964] [Zyl 1964] L. L. van Zyl, *An Electronic Analogue Computer for the Solution of Non-Linear Partial Differential Equations Encountered in the Study of the Self-Heating of Fishmeal*, Thesis, Faculty of Engineering, University of Cape Town, 1964 →

© 2026 Walter de Gruyter GmbH, Berlin/Boston 2026 Walter de Gruyter GmbH, Berlin/Boston

Index

A4, 1, 2, 3
AC Calculating Board, 1
accumulator, 1
ACID, 1
acoustic analog computer, 1
activation function, 1
Adage 770, 1
Adaline, 1
adaptive control, 1
ADC, 1, 2, 3, 4
ADDAVERTER, 1
adding component, 1
additive cell, 1
ADIOS, 1
aerodynamic unit, 1
Aerojet, 1
aeronautical engineering, 1
aerospace medicine, 1
AG&E, 1
AHCS, 1
AI, 1, 2, 3
AIM, 1
air
– – density, 1
– – rudder, 1
airborne simulator, 1
aircraft

– – arresting gear system, 1
– – position computer, 1
Aldrin, Buzz, 1
algebraic loop, 1, 2
ALGOL, 1, 2
all-to-all, 1
All-Union Scientific Research Oil-Gas Institute, 1
ALS-2000, 1
alternating operation, 1
ALU, 1
always on device, 1
Amdahl's law, 1
Amdahl, Gene Myron, 1
American Gas and Electric Company, 1
Ammon, Werner, 1
amplifier
– – negative-feedback, 1
– – open, 1
– – summing, 1
Amsler-Laffon, Jacob, 1
anabrid, 1, 2, 3, 4184,
– – LUCIDAC, 5
– – Model-1, 5
– – REDAC, 5
– – THAT, 5, 6
– – THE ANALOG THING, 5, 6
Anacom, 1, 2
Anacom II, 1
Anacom III, 1
Anadigm, 1
analog, 1, 23,
– – computer, 3, 45,
– – anatomy, 5

– – centers, 5
– – chances, 5
– – electronic, 5
– – future, 5
– – mechanical, 5
– – precision, 5
Analog Iterative Machine, 1
Analog Paradigm, 1
analog-digital-converter, 1, 2, 3, 4
analogy
– – direct, 1
– – indirect, 1
Analysis Laboratory, 1
analytical engine, 1
AND, 1
angle of attack, 1
ANN, 1, 2
Antikythera mechanism, 1
AOA, 1
APACHE, 1, 2
API, 1
Apollo, 1
Appalachian surge tank, 1
Application Programming Interface, 1
applications, 1313
– – aeronautical engineering, 2423
– – airborne simulator, 3
– – aircraft arresting gear systems, 3
– – flight simulation, 3
– – guidance and control, 3
– – helicopter, 3
– – jet engines, 3
– – landing gears, 3

– – nike, 3
– – parachutes, 3
– – polaris, 3
– – rotor blade, 3
– – aerospace engineering
– – Apollo, 2
– – Gemini, 2
– – Mercury, 2
– – rocket motor simulation, 2
– – rocket simulation, 2
– – spacecraft maneuvers, 2
– – arts, 2
– – automation, 2404
– – closed loop control, 3
– – correlation analysis, 3
– – data processing, 3
– – embedded systems, 3
– – sampling system, 3
– – servo systems, 3
– – biology, medicine, 2373
– – aerospace, 3
– – cardiovascular systems, 3
– – CO_2 regulation, 3
– – ecosystems, 3
– – epidemiology, 3
– – locomotor systems, 3
– – metabolism, 3
– – neurophysiology, 3
– – pupil regulation, 3
– – chemistry, 2348
– – quantum chemistry, 3
– – reaction kinetics, 3
– – economics, 2

– – education, 2
– – electronics, telecommunication, 2401
– – circuit simulation, 3
– – demodulator, 3
– – filter design, 3
– – frequency response, 3
– – modulator, 3
– – engineering, 2
– – entertainment, 2
– – geology, 2384
– – ray tracing, 3
– – resources, 3
– – seismology, 3
– – mathematics, 2313
– – differential equations, 3
– – Eigenvalues, Eigenvectors, 3
– – – Fourier synthesis and analysis, 3
– – integral equations, 3
– – linear algebra, 3
– – multidimensional shapes, 3
– – optimization, 3
– – orthogonal functions, 3
– – random process, 3
– – systems of linear equations, 3
– – zeros of polynomials, 3
– – mechanics, 2351
– – bearings, 3
– – compressors, 3
– – crank mechanisms, 3
– – ductile deformation, 3
– – earthquake simulation, 3
– – hydraulic systems, 3
– – machine tool control, 3

– – non-destructive testing, 3
– – pneumatic systems, 3
– – rotating system, 3
– – servo system, 3
– – shock absorber, 3
– – vibrations, 3
– – military, 2
– – music, 2
– – nuclear technology, 2364
– – control, 3
– – research, 3
– – training, 3
– – physics, 2334
– – ferromagnetic films, 3
– – heat-transfer, 3
– – optics, 3
– – orbit calculation, 3
– – particle trajectory, 3
– – semiconductor research, 3
– – power engineering, 2390
– – frequency control, 3
– – generators, 3
– – power grid simulation, 3
– – power inverter, 3
– – power stations, 3
– – rectifier, 3
– – transformers, 3
– – transmission lines, 3
– – process engineering, 2408
– – adaptive control, 3
– – distillation column, 3
– – evaporator, 3
– – heat exchanger, 3

– – mixing tank, 3
– – optimization, 3
– – parameter determination, 3
– – rocketry, 2
– – transport systems, 2413
– – automotive engineering, 3
– – dynamic behavior, 3
– – hovercraft, 3
– – maglev, 3
– – marshaling hump, 3
– – motor coach simulation, 3
– – nautics, 3
– – propulsion system, 3
– – railway vehicles, 3
– – ride simulation systems, 3
– – ship simulation, 3
– – steering system, 3
– – torpedo simulation, 3
– – traffic flow simulation, 3
– – transmissions, 3
Applied Dynamics, 1, 2, 391,
– – AD-10, 4
– – AD-2-64-PBC, 4
– – AD/FOUR, 4
approximation
– – Padé, 1
– – Stubbs-Single, 1
Arduino, 1263
– – MEGA-2650, 2
arithmetic/logic unit, 1
Arma Corporation, 1
Armstrong, Neil A., 1, 2, 3
artifical neural network, 1

artificial
– – brain, 1
– – intelligence, 1, 2, 3, 4
– – neural network, 1, 2
arts, 1
Asea Brown Boveri, 1
astrolabe, 115
– – planispheric, 2
attenuator, 1
attractor, 1
Automatic Digital Input Output System, 1
automatic level recorder, 1
automation, 1
automotive engineering, 1
Autonetics, 1
autopatch, 1

Bückner, Hans F., 1
Babbage, Charles, 1
backpropagation, 1
barrel cam, 1
BEAC, 1
bearings, 1
Beckman
– – EASE 2132, 1
– – EASE 1032, 1
– – EASE 1132, 1
– – EASE 2133, 1
Beckmann, Otto, 1
Bell Laboratories, 1, 2, 3, 4
Bell X-2, 1
Bendix
– – Corporation, 1

– – D-12, 1
– – G-15, 1
– – G-15D, 1
Bessel function, 1
bevel-gear differential, 1
binary digit, 1
biology, 1
bipolar stochastic number, 1
bit, 1
Black, Fischer, 1
Black, Harold Stephen, 1
blast computer, 1
blocking penalty, 1
Blumlein, Alan, 1
Bode diagram, 1
Bode, Hendrik Wade, 1
body, 118
– – dynamics, 2
Boeing, 1, 2
Bonneville Power Administration, 1
bouncing ball, 1
boundary value problem, 1, 2
Bournez, Olivier, 1
Boykan Pour-El, Marian, 1
BP, 1
brain inspired computing, 1
Bratt, J. B., 1
Braun, Wernher von, 1
Briggs, Henry, 1
British Petroleum, 1
broadside, 1
Brookhaven National Laboratories, 1
Brouwer, Bert, 1

BTL, 1
bucket brigade, 1
burst processing, 1
bus rod, 1
Bush, Vannevar, 1, 2

Calculating Board
  – – AC, 1
  – – DC, 1
California Institute of Technology, 1, 2
Caltech, 1
Caltech computer, 1
cam
  – – squaring, 1
  – – three-dimensional, 1
camoid, 1
capacitor
  – – switched, 1
capacitor wheel, 1
car suspension, 1
cardiovascular system, 1
Cathode Ray Tube, 1
Caw, Larry, 1
CDC
  – – 1604, 1
  – – 6800, 1
  – – 7600, 1
cell
  – – additive, 1
  – – linear, 1
Center for Analysis, 1
center slide, 1
CHANCE, 1

Chance, Britton, 1
check
– – dynamic, 1
– – rate, 1
– – static, 1, 2
chemistry, 1
chopper, 199
– – stabilization, 2
Chrysler Corporation Missile Operations, 1
CI-5000, 1
cinema integraph, 1
circle test, 1
circuit simulation, 1
closed loop control, 1
CMOS, 1
CNC, 1
CO_2 regulation, 1
coefficient potentiometer, 1
coincidence multiplier, 1
cold war, 1, 2
Colónia Güell, 1
color matching, 1
Colorant Mixture Computer, 1
Comcor, 1
Comdyna, 1461
– – GP-6, 2
COMIC, 1
command guidance system, 1
comparator, 1142
– – electronic, 2
– – relay, 2
compiler, 1
compressors, 1

computer, 1, 23,
– – analog, 3, 4
– – blast, 3
– – Caltech, 3
– – control, 3
– – digital, 3, 4
– – incremental, 3
– – map, 3
– – radiation, 3
– – thermal, 3
Computer History Museum, 1
computing, 1
concentrator, 1
conformal mapping, 1
continuous steepest ascent/descent, 1
contour map, 1
control computer, 1
Control Data, 1, 2
control of machine tools, 1
Convair Astronautics, 1, 2
Coordinate Rotation Digital Computer, 1
coordination equations, 1
CORDIC, 1
correlation analysis, 1
CORSAIR, 1
Coveney, Peter, 1
Cowan, Glenn Edward Russel, 1
crank mechanisms, 1
Cray-1, 1
crossbar switch, 1
crossed-fields multiplier, 1
CRT, 1, 2
CSMP, 1

CSS Virginia, 1
cursor, 1
curve follower, 1

D-12, 1
DAC, 1, 2, 3, 4, 5
Daimler Benz, 1
Darcy, Henry Philibert, 1
DARE P, 1
DARE/ELEVEN, 1
DAS, 1
data processing, 1
Davidson, Hugh R., 1
Day, Richard E., 1, 2
DBG, 1
DC Calculating Board, 1
DCFG, 1
DDA, 1, 28,
– – parallel, 3
– – sequential, 3
– – simultaneous, 3
DDP-24, 1, 2
de Solla Price, Derek, 1
DEC, 1
decay constant, 1
defense calculator, 1
delay wheel, 1
delayed neutron, 1
Deltar, 1
Deltawerken, 1
demodulator, 1
DEQ, 1
DESY, 1, 2

Deutsches Elektronensynchrotron, 1, 2
DIAN, 1
DIDA, 1
difference engine, 1
differential, 1
differential analyzer, 137
– – electromechanical, 2
differential equation, 1, 22,
– – partial, 3
differential gear, 1
differential surge chamber, 1
digit
– – binary, 1
digital
– – ANALOG SIMULATOR, 1
– – computer, 1, 2
– – differential analyzer, 1, 2
– – signal processor, 1
– – voltmeter, 1, 2
digital-analog converter, 1
digital-analog-converter, 1, 2, 3, 4
digitally controlled function generator, 1
diode
– – – ZENER, 1
– – function generator, 1
direct analogy, 1, 2
distillation column, 1
Donner
– – 30, 1
– – 3500, 1
Dornier
– – Do 31, 1
– – Do 80, 1

double-ball integrator, 1
Douglas Aircraft Company, 1
drift
– – parameter, 1
– – stabilization, 1
DSL, 1
DSL-90, 1
DSP, 1
ductile deformation, 1
duplex, 1
DVM, 1, 2
dynamic
– – behavior, 1
– – check, 1
Dynamic System Synthesizer, 1
dynamometer multiplier, 1
DYSAC, 1
DYSTAC, 1

E6-B, 1
EAC, 1, 2
EAFCOM, 1, 2
EAI, 1, 2, 3, 4101,
– – 16-31R, 5
– – 16-6H, 5
– – 231R, 5, 6, 7, 8, 9, 10, 11
– – 231RV, 5, 6, 7, 8
– – ADIOS, 5
– – DATA-PACER, 5
– – DES-30, 5
– – DOS 350, 5, 6
– – DOS-350, 5
– – HYDAC 2000, 5

– – PACE, 5
– – PACE 16-31R, 5
– – PACE 201A, 5
– – Pace 96, 5
– – PACER 500, 5
– – SIMSTAR, 5, 6, 7
– – time delay, 5
– – TR-10, 5, 6
– – TR-48, 5, 6, 7, 8, 9
– – Variplotter, 5
– – VDFG, 5
– – 180, 5
– – 380, 5
– – 580, 5, 6, 7
– – 590, 5
– – 640, 5
– – 680, 5
– – 690, 5, 6
– – 693, 5
– – 2000, 5
– – 6200, 5
– – 8400, 5
– – 8800, 5
– – 8900, 5, 6
earthquake simulation, 1
Echo-1, 1
Eckdahl, Donald E., 1
economic dispatch computer, 1
economics, 1
ecosystems, 1
EDC, 1
Edison, Thomas, 1
Edmunds, Mike, 1

education, 1
Effelsberg, 1
Ehricke, Krafft Arnold, 1
Eigenvalue, 1
Eigenvector, 1
ELARD, 1
Electrical Transients Analyzer, 1
Electro-Analogue, 1
electrolytic tank, 1, 2, 3
electromechanical differential analyzer, 1
electron-beam multiplier, 1
Electronic Analog Frost Computor, 1
Electronic Associates, Inc., 1, 2
electronic comparator, 1
Electronic Dispatch Computer, 1
Electronic Graph Paper, 1
electronic slide rule, 1
electronic structure, 1
electronics, 1
embedded system, 1
EMI, 1
EMIAC, 1
end
– – brace, 1
– – bracket, 1
engineering, 1
English Electric Company, 1
ENIAC, 1
enrichment, 1
entertainment, 1
epidemic, 1
epidemiology, 1
Esaki, Leo, 1

Esch, John William, 1
ESIAC, 1
Euler, Leonhard, 1
evaporator, 1
examples
– – $\ddot{y} - \omega^2 y = 0$, 1
– – bouncing ball, 1
– – car suspension, 1
– – conformal mapping
– – epidemic, 1
– – examples
– – – Hindmarsh-Rose, 1
– – examples
– – – Lorenz attractor, 1
– – Lotka-Volterra equations, 1
– – mass-spring-damper, 1
– – mass-spring-damper system, 1
– – examples
– – – Mathieu equation, 1
– – predator and prey, 1
– – projection of rotating body, 1
– – sweep generator, 1
Examples, 1
Excess-three, 1
exclusive OR, 1
exhaust rudder, 1
expander, 1
expansion factor, 1
Explorer I, 1, 2
exponentially mapped past, 1
extended analog computer, 1, 2

F-100A, 1

Faber Castell
– – 2/83, 1
false, 1
Fano, Robert Mario, 1
Farris, George Joseph, 1
Fast Fourier Transformation, 1
feedback
– – negative, 1
– – technique, 1
feedforward GPAC, 1
Fermi, Enrico, 1
ferromagnetic films, 1
Feynman, Richard, 1
FF-GPAC, 1
FFT, 1
FGMOS, 1
Field Programmable Gate Array, 1
filter design, 1
financephalograph, 1
fire control system
– – electronic, 1
– – mechanical, 1
Fischer, 1
FitzHugh, Richard, 1
FLEPAC, 1
flight
– – computer, 1
– – control computer, 1
– – simulation, 1
– – table, 1
flip-flop, 1, 2, 3, 4
fluid dynamics, 1
Fluistor, 1

fly-by-wire, 1
flying bedstead, 1
FORTRAN, 1
four-quadrant operation, 1
Fourier
– – analysis, 1, 2
– – synthesis, 1, 2
– – transform, 1, 2
Fourier, Jean-Baptiste-Joseph, 1
Foxboro, 1
FPAA, 1
FPGA, 1
Franke, Herbert W., 1
Fraser, James Earl, 1
Fredholm, Ivar, 1
frequency
– – control, 1
– – response, 1
friction-wheel, 121
– – integrator, 2
frontlash unit, 1
full-adder, 1
Fulton, Fitzhugh L., 1
function
– – – Mathieu, 1
– – – Bessel, 1
– – activation, 1
– – fitter, 1
– – generator, 1117
– – curve follower, 2
– – diode, 2
– – photoformer, 2
– – polygon, 2

– – orthogonal, 1

G-15D, 1
Gage, F. D., 1
gain
– – open loop, 1
Gaines, Brian R., 1
GAP/R, 1, 273,
– – K2-P, 3
– – K2-W, 3
– – K2-X, 3, 4
gate
– – logic, 1
Gauß, Carl Friedrich, 1
gearbox
– – helical, 1
GEDA, 1, 2
Geiser, Kenneth Raymond, 1
Gemini, 1
General
– – Dynamics, 1
– – Electric, 1
– – Motors, 1, 2, 3
– – Purpose
– – Airborne Simulator, 1
– – Analog Computer, 1
– – Simulator, 1
generators, 1
genetic programming, 1
geology, 1
George A. Philbrick Researches, 1, 2
Gilbert cell multiplier, 1
Gilbert, Barrie, 1

Giloi, Wolfgang, 1
Goddard, Robert, 1
Goldberg, Edwin A., 1, 2, 3
Goldmann, Hans Otto, 1, 2
Golf Game Computing System, 1
Gonnella, Tito, 1
Goodyear, 1
Goodyear Electronic Differential Analyzer, 1
GP-6, 1
GPAC, 1, 212,
– – feedforward, 3
GPAS, 1
GPS, 1
Graves, Tom, 1
Green theorem, 1
Green's theorem, 1
guidance and control, 1
Guided Weapons Division, 1
Gundlach, Friedrich-Wilhelm, 1
Guo, Ning, 1

Hall effect multiplier, 1
halt, 1, 2
Hammer, J. A., 1
Hannover fair, 1
Harder, Edwin L., 1, 2
hardware in the loop, 1
harmonic
– – analyzer, 1, 2
– – synthesizer, 1
harmonics, 1
HARTRAN, 1
Hartree, Douglas, 1

hatchet planimeter, 1
Hazen, Allen, 1
Hazen, H. L., 1
HCS, 1
Heat Exchange Transient Analog Computer, 1
heat exchanger, 1
heat-transfer, 1
heat-transfer multiplier, 1
Heidersberger, Benjamin, 1
Heidersberger, Heinrich, 1
helical gearbox, 1
helicopter, 1
Hemmendinger, Henry, 1
Hermann, Johann Martin, 1, 2
Hermes, 168
– – A-1, 2
HETAC, 1, 2
Hewlett Packard, 1
Hewlett-Packard, 1, 2
high
– – performance computing, 1
– – speed differential analyzer, 1
– – speed flight station, 1
Higinbotham, William, 1
Hindmarsh, James L., 1
Hindmarsh-Rose neuron model, 1
Hitachi-240, 1
HKW 860, 1
Hodgkin, Alan Lloyd, 1
Hoelzer, Helmut, 1
hovercraft, 1
HP, 185
– – 3722A, 2

– – 35, 2, 3
HPC, 1
HRS 860, 1
HSDDA, 1
HSE Analog Computer, 1
Hudson Bay Company, 1
hump marshaling, 1
HUTSPIEL, 1
HUXLEY, ANDREW FIELDING, 1
hybrid
– – computer, 1
– – computer power simulator, 1
– – PACTOLUS, 1
hybrides Koppelwerk, 1
HYDAC 2000, 1
hydraulic system, 1
HYPAC, 1
hyperbolic field tube, 1
hypercomputation, 1
HYTRAN, 1
HYTROL, 1

I-Block, 1
IBM, 1456
– – 360/67, 2
– – 360/75, 2
– – 7070/7074, 2
– – 7094-II, 2
– – 701, 2
– – 704, 2
– – 1620, 2, 3
– – 7094, 2
IC, 1

ICBM, 1
ICT, 1
in memory computing, 1
in-flight simulation, 1
increment, 1
incremental
– – computer, 1, 2
– – value, 1
indicator
– – diagram, 1
– – steam engine, 1
indirect analogy, 1
initial condition, 1, 2
input table, 1
Instrumentation Laboratory, 1
integral equations, 1
integraph
– – cinema, 1
– – product, 1
integrate-and-fire, 1
integrator, 1, 2, 327,
– – double-ball, 4
– – friction-wheel, 4, 5
– – mechanical, 4
– – step, 4
Integrieranlage IPM-Ott, 1
integromat, 1
intercontinental ballistic missile, 1
interposer, 1
inverse functions, 1
ISEC 1, 2
Isograph, 1
iterative mode, 1

Jacobi, Carl Gustav Jakob, 1
jerk, 1
jet engines, 1
Jet Propulsion Laboratory, 1
JetStar, 1
jolt, 1
Joukowsky, 1
JPL, 1
Julie, Loebe, 1
Jupiter, 168
– – C, 2

Kac, Mark, 1
KACO, 1
Kastner, Sigismund, 1
Kear, F. G., 1
Kelvin, Lord, 1
Kempel, Bob, 1
kernel, 1
Kettel, Ernst, 1, 2
Kirchhoff, Gustav Robert, 1
Kleinwächter, Hans, 1
Koenig, Eldo, 1
Koppelwerk
– – hybrides, 1
Kreiselgeräte GmbH, 1
Kron, Gabriel, 1
Kubelka, Paul, 1
Kubelka-Munk theory, 1
Kuhn, Hans, 1
Kuhn, Thomas Samuel, 1
Kulk, Hans, 1

Lämmle, 1
LACE, 1
LADME, 1
Landing Craft Mechanized, 1
landing gears, 1
Lapique, Louis, 1
Laplace, Pierre-Simon, 1
Laposky, Ben, 1
laser diode, 1
Law
– – – Amdahl's, 1
LCM-6, 1
learning control system, 1
LEGNO, 1
LEM, 1
LEV-3, 1
Lewis Flight Propulsion Laboratory, 1
LGP-30, 1
limiter, 1
linear
– – algebra, 1
– – cell, 1
– – ordinary differential equation, 1
– – planimeter, 1
– – servo system, 1
linear congruence generator, 1
Liouvill, Joseph, 1
Lissajous, Jules Antoine, 1
Litton Systems, 1
LKG, 1
Lockheed JetStar, 1
locomotor system, 1
locus curve, 1

LODE, 1
logarithmic multiplier, 1
logic gate, 1
long short-term memory, 1
long-tailed pair, 1, 2
loop
– – algebraic, 1, 2
– – servo, 1
Lord Kelvin, 1
Lorentz force, 1
Lorenz attractor, 1
Lorenz, Edward N., 1
Lotka, Alfred James, 1
Lotka-Volterra equations, 1
Lovell, Clarence A., 1
LSTM, 1
LUCIDAC, 1
lucipy, 1
Lukyanov, Vladimir, 1
Lunar Excursion Module, 1
Lunar Vehicle Digital Computer, 1
Luton Analogue Computing Engine, 1
LVDC, 1

M-1, 1
M-33, 1
M-5, 1
M-9, 1
machine
– – learning, 1
– – time, 1
– – tool control, 1
– – unit, 1

– – variable, 1
Macnee, A. B., 1
macroblock, 1
MADDIDA, 1, 2
Mader-Ott, 1
maglev, 1
magnetic bearing, 1
mainlobe, 1
majority voting, 1
Makoto, Shiga, 1
Manning, Robert, 1
map computer, 1
mark-space multiplier, 1
marshaling hump, 1
Mason, Maughan S., 1
mass-spring-damper, 1
mass-spring-damper system, 1
Massachusetts Institute of Technology, 1, 2
Mathematical Tables Project, 1
mathematics, 1
Mathieu equation, 1, 2
Mathieu functions, 1
Mathieu, Émile Léonard, 1
MATLAB, 1
MBB Aircraft Division, 1
McCann, Gilbert D., 1
McDonnell Aircraft Corporation, 1
McIlroy, Malcolm Strong, 1
Mean Time between Failure, 1
mechanical
– – analog computer, 1
– – integrator, 1
– – multiplier, 1

– – transients analyzer, 1
mechanics, 1
mechanism
– – crank, 1
medicine, 1
Mel Apt, 1, 2
memistor, 1
memristor, 1, 2, 3
Mercury, 1
metabolism, 1
Metal Oxide Semiconductor, 1
Meyer-Brötz, Günter, 1, 2
Michelson, Albert Abraham, 1
MIDAS, 1
Mike Boat, 1
military applications, 1
Mills, Jonathan W., 1
MIMIC, 1
Minden, 1
Minispace, 1
Mischgerät, 1, 2
Mises, Richard von, 1
missile
– – Jupiter, 1
– – Redstone, 1, 2, 3
MIT, 1, 2, 338,
– – Instrumentation Laboratory, 4
mixing tank, 1
ML, 1
ML-1, 1
MLPP, 1
mode logic patch panel, 1
model, 1

modulation multiplier, 1
modulator, 1
Monetary National Income Automatic Computer, 1
MONIAC, 1
monostable multivibrator, 1
Monte-Carlo, 1, 2, 3
Moreland, William J., 1
MOSFET, 1
motor coach simulation, 1
MTBF, 1
multi-compartment model, 1
multidimensional shape, 1
multiplication, 1
multiplier, 1, 230,
– – – Hall effect, 3
– – coincidence, 3
– – crossed-fields, 3
– – dynamometer, 3
– – electron-beam, 3
– – Gilbert cell, 3
– – heat-transfer, 3
– – hyperbolic field, 3
– – logarithmic, 3
– – mark-space, 3
– – mechanical, 3
– – modulation, 3
– – pulsed-attenuator, 3
– – quarter square, 3
– – servo, 3
– – strain-gauge, 3
– – time division, 3
Munk, Franz, 1
Museum of Electronic Games & Art, 1

music, 1
musique cybernetique, 1
muzzle velocity, 1
MX-775A, 1

N. S. Savannah, 1
NACA, 1, 2
Nagumo, Jin-ichi, 1
NAND, 1
Napier, John, 1
NASA, 1, 2
National Physical Laboratory, 1, 2
nautics, 1
Naval Undersea Center, 1
Navier, Claude Louis Marie Henri, 1
Nederlands Radar Proefstation, 1
negative
– – feedback, 1
– – feedback amplifier, 1
network analyzer, 1
Neumann, John von, 1, 2
neuromime, 1
neuromorphic computing, 1, 2
neurophysiology, 1
neutron
– – delayed, 1
– – density, 1
– – flux density, 1
– – kinetics, 1
– – lifetime, 1
– – prompt, 1
Nike, 1, 2
noise generator, 1

non-destructive testing, 1
nonlinear servo system, 1
Nordsieck computer, 1
Nordsieck, Arnold Theodore, 1
normalization, 1
North American Aviation, 1
Northrop, 1, 2
NOVA, 1
NPL, 1, 2
nuclear
– – technology, 1
– – weapon effect computer, 1
NUKEMAP, 1
Nyquist diagram, 1
Nyquist, Harry, 1

O'Neil, Jim, 1
ODE, 1, 2
Ohm, Georg Simon, 1
Okika Devices, 1
OP, 1
open
– – amplifier, 1
– – loop gain, 1
operate, 1, 2
operational amplifier, 1, 2, 376,
– – drift stabilization, 4
– – Nike, 4
operational flight trainer, 1
Oppikofer, Johannes, 1
optics, 1
optimization, 1, 2
OR, 1296

– – exclusive, 2
orbit calculation, 1
ordinary differential equation, 1
orthogonal function, 1
oscillator
– – relaxation, 1
oscilloscope, 1
Oslo analyzer, 1, 2
Ottermann, Joseph, 1
Oughtred, William, 1
output
– – device, 1
– – multiplier, 1

PACER MUSICALE, 1
Packard Bell, 1291
– – PB-250, 2
PACTOLUS, 1, 2
Padé approximation, 1
Padé, Henri, 1
parachutes, 1
parallel DDA, 1
Parallel Logic Unit, 1
parameter determination, 1
parametron, 1
Parkinson, David B., 1
partial
– – differential equation, 1
– – feedback technique, 1
– – tide, 1
particle trajectory, 1
Pascal, Blaise, 1
patch

– – cable, 1
– – panel, 1
PCB, 1
PDE, 1
PDP-11, 1
Pearl Street Station, 1
Peenemünde, 1, 2
PG, 1
PHI-4, 1
Philbrick, George A., 1, 2
Phillips, Alban Williams, 1
photo cell, 1
photoformer, 1, 2
physics, 1334
– – plasma, 2
Phytotron, 1
planimeter, 119
– – hatchet, 2
– – linear, 2
– – polar, 2
– – Prytz, 2
– – wheel-and-cone, 2
planispheric astrolabe, 1
plasma physics, 1
PLU, 1
pneumatic system, 1
Pogo effect, 1
Pol, Balthasar van der, 1
polar planimeter, 1
Polaris, 1
pole, 122
– – arm, 2
– – weight, 2

pole-zero plot, 1
polydromial, 1
polygon generator, 1
polynomial
– – zeros of, 1
Polyphemus, 1
Pontryagin, 1
Porter, Arthur, 1
Pot Set, 1
potentiometer
– – coefficient, 1
– – digital, 1
– – set, 1
– – tapped, 1
power
– – engineering, 1
– – grid simulation, 1
– – inverter, 1
– – stations, 1
power ground, 1
Pratchett, Terry, 1
precision analog computer, 1
Precision Analog Computing Equipment, 1
predator and prey, 1
pressurized-water reactor, 1
problem time, 1
process engineering, 1
product integraph, 1
profile tracer, 1
program, 1
programming, 1187
– – examples, 2
progressive precision, 1

Project Cyclone, 1
projection of rotating body, 1
prompt neutron, 1
propulsion system, 1
Prytz, Holger, 1
pseudo-random sequence, 1
Pullman, 1
pulsed-analog, 1
pulsed-attenuator multiplier, 1
pupil regulation, 1
pybrid, 1

Q-learning, 1
QK-256, 1
QK-329, 1
quadratron, 1
quadrature
– – band-pass, 1
– – signal pair, 1
quantum chemistry, 1
quarter square multiplier, 1
quotient of differences, 1

R100, 1
radar, 1
radiation computer, 1
radio telescope, 1
Ragazzini, John, 1
railway vehicle, 1
Raimann, Franz, 1
RAM, 1, 2
Ramo-Wooldridge Corporation, 1, 2
random

– – access memory, 1, 2
– – noise generator, 1
– – process, 1
random walk, 1
Ranfft, Roland, 1
RASCEL, 1
rate check, 1
ray tracing, 1
Raytheon, 1
REAC, 1, 2
reaction
– – kinetics, 1
– – wheel, 1
reaction-diffusion chemical computer, 1
real-time, 1
rectified linear unit, 1
rectifier, 1
Redstone, 1, 2, 358,
– – Arsenal, 4, 5
Reeves, 1423
– – Instrument Corporation, 2, 3
– – 2, 3
regulation
– – CO_2, 1
– – pupil, 1
Rein, Hans-Martin, 1
relaxation oscillator, 1
relay comparator, 1
ReLU, 1
repetitive operation, 1, 2
resolver, 1
resources, 1
respiratory arrhythmia, 1

Rhythmograph, 1
ride simulation system, 1
ring modulator, 1
Robinson, Tim, 1
Rockefeller differential analyzer, 1
Rockefeller Foundation, 1
rocket
– – A4, 1, 2
– – motor simulation, 1
– – NOVA, 1
– – Saturn, 1, 2
– – simulation, 1
– – V2, 1
rocketry, 1
Rohde & Schwarz, 1
Rose, R. M., 1
Rosseland, Svein, 1
rotating system, 1
rotor blade, 1
Royal Aircraft Establishment, 1
RTG, 1
Rubel, Lee Albert, 1, 2, 3
rudder
– – air, 1
– – exhaust, 1
Rutherford, Ernest, 1

SAH, 1
sample-and-hold, 1
sampling system, 1
Saturn, 1, 2
Saturn V, 1
scale model, 1

scaling, 1, 2
Schickard, Wilhelm, 1
Schmitt trigger, 1
Scholes, Myron, 1
Schoppe & Faeser, 1
Schottky diode, 1
Schottky, Walter Hans, 1
Schrödinger equation, 1
Schrödinger, Erwin, 1
scotch yoke, 1, 2, 3
SCR-584, 1
SDE, 1
Sedlacek, Josef Adalbert, 1
Seidel, Philipp Ludwig von, 1
Seilcomputer Kelvin, 1
SEIR model, 1, 2
seismology, 1
semiconductor research, 1
separation of variables, 1
sequential DDA, 1
SERAC, 1
servo, 1279
– – loop, 2
– – multiplier, 2
– – system, 2, 3362,
– – linear, 4
– – nonlinear, 4
setback leaf system, 1
SG, 1
Shannon, Claude Elwood, 1
shape
– – multidimensional, 1
Sharon Steel Corporation, 1

shimmy, 1
ship simulation, 1
shock absorber, 1, 2
shooting method, 1
Short Brothers, 1
sidelobe, 1
Siegelmann, Hava T., 1
Sigma-5, 1
signal ground, 1
silicon carbide, 1
Sim One, 1
simplex, 1
Simulag Unit, 1
simulation, 16
– – airborne, 2
– – circuit, 2
– – earthquake, 2
– – epidemic, 2
– – flight, 2
– – in-flight, 2
– – motor coach, 2
– – power grid, 2
– – rocket, 2
– – rocket motor, 2
– – ship, 2
– – torpedo, 2
– – traffic flow, 2
simulator, 1
Simulink, 1
simultaneous
– – DDA, 1
– – operation, 1
single run, 1

Single, C. H., 1
Sirutor, 1
slide rule, 117
– – body, 2
– – center slide, 2
– – cursor, 2
– – duplex, 2
– – electronic, 2
– – end
– – brace, 2
– – bracket, 2
– – simplex, 2
– – stator, 2
– – stock, 2
slipstick, 1
slot, 1
SNR, 1
Solartron, 1102
– – Minispace, 2
Sommerfeld, Arnold, 1
Southern, John, 1
space race, 1
Space Technology Laboratories, 1
spacecraft maneuvers, 1
SPEDAC, 1
spider block, 1
Spiess, Ray, 1
Sprague, Richard, 1
squaring cam, 1
Squash, 1
SSM-A-3 Snark, 1
Stahlröhre, 1
STARDAC, 1

Starfighter, 1
STARTRAN, 1
STATEST, 1
static
– – check, 1, 2, 3
– – test, 1
stator, 1
steam engine indicator, 1
Steele, Floyd, 1
steering system, 1
Steinhoff, Ernst, 1
step integrator, 1
Steuding, Hermann, 1
Stieltjes integral, 1
STL, 1
stochastic
– – computing, 1
– – number
– – bipolar, 1
– – unipolar, 1
stock, 1
Stokes, George Gabriel, 1
strain-gauge multiplier, 1
Stratton, Samuel Wesley, 1
strip-chart recorder, 1
Strong Earthquake Response Analog Computer, 1
Stubbs, G. S., 1
Stubbs-Single approximation, 1
Stufenintegrator, 1
Sturm, Charles-François, 1
substitution method, 1
summer, 1, 2
Summers, Bill, 1

summing amplifier, 1
surge
– – chamber, 1358
– – differential, 2
– – tank
– – Appalachian, 1
Swartzel, Karl D., 1
sweep generator, 1
switch
– – crossbar, 1
switched capacitor, 1
symbol generator, 1
Symphony, 1
system
– – cardiovascular, 1
– – embedded, 1
– – hydraulic, 1
– – locomotor, 1
– – pneumatic, 1
– – propulsion, 1
– – rotating, 1
– – sampling, 1
– – servo, 1
– – steering, 1
– – transport, 1
Systems Engineering Conference, 1
systems of linear equations, 1
Systron Donner, 1

T-10, 1
T-15, 1
Tactical Avionics System Simulator, 1
tank

– – electrolytic, 1
tapped potentiometer, 1
target scintillation, 1
TASS, 1
Tatudiro, Sasaki, 1
Teichmann, Alfred, 1
telecommunication, 1
Teledyne, 1
Teledyne-Philbrick, 1
Telefunken, 1, 2, 3, 452,
– – DEX 100, 5
– – DEX 102, 5
– – DEX 802, 5, 6
– – HKW 860, 5
– – HRS 860, 5
– – OMS 800, 5
– – RA 1, 5
– – RA 463/2, 5, 6, 7, 8
– – RA 741, 5, 6
– – RA 770, 5, 6, 7, 8, 9, 10
– – RA 800, 5, 6
– – RA 800H, 5
– – RAT 700, 5, 6
– – RAT 741, 5
Tellegen, Bernardus Dominicus Hubertus, 1
Tennessee Valley Authority, 1
Tennis for Two, 1
tesseract, 1
THE ANALOG THING, 1
The Rice Institute, 1
Theodorsen, 1
thermal computer, 1
Thomson, James, 1

Thomson, Margaret, 1
Thomson, William, 1
three-dimensional
– – analogue computer, 1
– – cam, 1
thyrite, 1
tide, 131
– – partial, 2
time
– – delay, 1
– – division multiplier, 1
– – problem, 1
– – real, 1
time-scale factor, 1
torpedo
– – data computer mark 1, 2
– – simulation, 1
torque amplifier, 1
TR-5, 1
tracer arm, 1
track-and-store, 1
traffic flow simulation, 1
transconductance technique, 1
transformer, 1
translinear circuit, 1
transmission lines, 1
transmissions, 1
transport systems, 1
Trefftz, Erich, 1
TRICE, 1
TRIDAC, 1
Tropsch, 1
TRS-2, 1

true, 1
trunk line, 1
TRW Systems Group, 1
TU Wien, 1
tune-up time, 1
tunnel diode, 1
Turing
– – complete, 1
– – machine, 1
Turing, Alan Mathison, 1
Typhoon, 1

U-block, 1
unipolar stochastic number, 1
UNIVAC
– – 1103A, 1
– – incremental computer, 1
– – 1110, 1, 2
– – 1230, 1
Universal-Integriergerät, 1
University of
– – Chicago hospital, 1
– – Virginia, 1
US-Air Force, 1
USS Monitor, 1

V1, 1
V2, 1
value coefficient, 1
van Vark, Tatjana Joëlle, 1
Vance, Arthur W., 1
variable-stability aircraft, 1
Variplotter, 1

varistor, 1, 2
VDFG, 1
Veen, Johan van, 1
Very Large Scale Integrated Circuit, 1
vibrations, 1
vibrator, 1
VJ 101C-X2, 1
VLSI, 1
Voigt, Paul, 1
Volterra, Vito, 1
VTOL, 1

Wainwright, Lawrence, 1
war
– – cold, 1, 2
– – of currents, 1
– – second world-, 1
Wasserfall, 1
water hammer, 1
Watt, James, 1
wavelet transform, 1
Weapons Effects Display System, 1
weathering layer, 1
WEDS, 1
Weinel, Ernst, 1
Weisbach, Julius Ludwig, 1
Westinghouse Electric Corporation, 1
Westinghouse, George, 1
wheel-and-cone planimeter, 1
Wiener, Norbert, 1, 2, 3
Wilbur, John “Bud” Benson, 1
Williams, Gardner Stewart, 1
windage, 1

winds unit, 1
Works Progress Administration, 1
WPA, 1
Wright Air Development Center, 1

X-15, 1, 2
Xenon-poisoning, 1
XOR, 1

Z3, 1
Zener diode, 1
zeros of polynomials, 1
Zhukovsky, Nikolay Yegorovich, 1
ZI-S, 1
Zuse, Konrad, 1

Notes

1 The 2nd edition had 858 bibliography entries.

2 … and Pietri: 

3 See →https://www.kicad.org, retrieved 02.06.2025.

1 John H. McLeod, Suzette McLeod, “The Simulation Council Newsletter”, in *Instruments and Automation*, Vol. 31, March 1958, p. 488.

1 See [→Lang 2000].

2 The notion of an *analog electronic analog computer* may look like a pleonasm, which it is not, as the following section will show.

3 A much more in-depth treatment of analog and hybrid computer programming can be found in [→Ulmann 2023/1].

4 Imagine doing division using Roman numerals… A wonderful account of the historic development of number systems can be found in [→Ifrah 2010].

5 There is a even a book devoted to this topic: [→Aspray et al. 1990].

6 This is due to the famous Swiss mathematician Leonhard Euler, 15.04.1707–18.09.1783, who published it in 1768, see [→Euler 1768]. This algorithm yields an approximate solution for differential equations by performing many successive multiply-and-add operations.

7 Jean-Baptiste Joseph Fourier, 21.03.1768–16.05.1830

8 26.12.1791–18.10.1871, a comprehensive description of his life and times can be found in [→Hyman 1982].

9 A rather philosophical treatment of this very idea can be found in [→Maley 2023].

10 See →section 2.2.

11 22.04.1592–23.10.1635

12 19.06.1623–19.08.1662

13 The *National Advisory Committee for Aeronautics*, the predecessor organization of *NASA*, the *National Aeronautics and Space Administration*.

14 [→Grier 1998] gives a comprehensive overview of the project including some actual worksheets for things like Mathieu function tables, etc. See also [→Grier 2007].

15 See [→Huggins 1958]. This is one of the occurrences where the construction of an analog computer, at parts of it, required a digital computer.

16 Today these numbers are normally represented by *binary digits, bits* for short.

17 [→Tse et al. 1964, p. 333] characterize these analogs or analogies as follows: "*The term 'analogy' is defined to mean similarity of relation without identity*." A good description of the term "analog" and its use in Germany with a focus on the model property can be found in [→Senoner 2025].

18 The path from analogy-making to modeling of a problem is treated comprehensively by [→Care 2008].

19 See [→Charlesworth et al. 1974, p. xi].

20 Cf. →sections 12 and →15.15.8.

21 It will be shown that DDAs are more capable machines than traditional analog computers since they can deal well with the highly important class of partial differential equations, which analog computers can do only with considerable difficulty and often only by means of discretization.

22 See [→Bild der Wissenschaft 1970] for examples.

23 The roof structure of the Olympic stadium in Munich was modeled to a large extent using curtain net lace as well as soap bubbles for determining the structure of single roof tiles. It should be noted here that *scale models* and direct analogies are different things, though. A scale model is not based on an analogy as it is basically the same thing as the problem under consideration.

24 More detailed information about this basic class of analogs can be found in [→Jackson 1960, pp. 319 ff.],

[→Paschkis et al. 1968], [→Master et al. 1955], [→Larrowe 1955] and [→Karplus 1958].

25 25.06.1852–10.06.1926

26 See [→Krämer 1989] for more details on this particular model. This approach was by no means new even in Gaudí's time. See [→Havil 2019, pp. 173 ff.] for a mathematical and historic perspective.

27 One of the earliest publications on the use of electronic analogs to simulate mechanical and acoustical systems was [→Olson 1943].

28 Direct analogs can also be employed in cases where no complete mathematical description of the problem to be solved exists – this may be caused by a principle lack of understanding or by the sheer complexity of the underlying problem. So in some cases direct analogs may even be employed today with success.

29 This does not hold true for digital differential analyzers, cf. →section 12.

30 Some clever error reduction methods for analog computers are described in [→Turner 1960]. [→Marsocci 1956] and [→Meilander 1957] give insights into error analysis and related problems.

31 See →https://en.wikipedia.org/wiki/Computing, retrieved 07.07.2025.

32 An in-depth treatment of the question "When does a physical system actually compute?" is given in [→Horsman et al. 2014]. The requirement of

programmability basically rules out fixed control systems to be considered computers. I. e., something like the Mischgerät, see →section 3.1.1, does not actually count as an analog computer. However, it is included in the chapters below due to its central role in the development of early electronic analog computers.

33 Alan Mathison Turing, 23.06.1912–07.06.1954

34 30.04.1916–24.02.2001, a comprehensive biography can be found in [→Soni et al. 2017] while [→Glaschick 2023] gives a more technically inclined account.

35 See [→Shannon 1941].

36 Multiplication is omitted as it can be implemented using two coupled integrators on a differential analyzer.

37 More details on the GPAC can be found in [→Bournez 2015]. [→Bournez et al. 2012] give an overview over continuous time computations.

38 See also [→Bournez 2015].

39 A Turing machine is a theoretical device, controlled by an infinitely long tape, which can be read and written by the machine. It executes simple operations under control of this tape, and can be shown to being capable of computing every computable function (see [→Turing 1937] and [→Hopcroft et al. 1979]).

40 29.04.1928–10.06.2009

41 See [→Pour-El 1974].

42 This does not mean that it is impossible to compute approximate functions in certain intervals for such functions.

43 01.12.1928–25.03.1995

44 See [→Rubel 1981], [→Duffin 1981].

45 This is a particularly interesting topic as it gives rise to the question of the influence of unavoidable noise in analog circuits to their computational power, not to mention noise explicitly introduced into some analog computer setup, i. e., in the case of treating stochastic differential equations.

46 See [→Rubel 1989].

47 See [→Bournez et al. 2017/2], [→Bournez et al. 2007], and [→Bournez et al. 2017/1].

48 30.10.1958–

49 See [→Coveney 2024].

50 23.08.1964–

51 However, as Olivier Bournez pointed out (personal communication), there are three things to consider: Reality, physics, and mathematical models. It is unclear if there is a total correspondence between all pairs of these, though. In the case of mathematical models more than offered by an abstract Turing machine can be computed. The question remains about the correspondence to physical implementations of such models.

1 Picture taken by Tilemahos Efthimiadis, protected by the *Creative Commons Attribution 2.0 Generic license*.

2 22.01.1922–03.09.1983

3 See [→ Freeth 2008, p. 7].

4 Cf. →http://www.antikythera-mechanism.gr/.

5 See [→ Freeth et al. 2021] and [→ Freeth 2022].

6 Prior to this discovery differential gears were thought to have been invented in medieval times.

7 See [→ Freeth 2008, p. 9].

8 There are still arguments whether the mechanism also featured indicators for the display of planet positions.

9 A wealth of information about this device may be found in [→ de Solla Price 1974], [→ Freeth 2008] and [→ McCarthy 2009].

10 Also known as a *slipstick*.

11 Additional precision is achieved by this type of slide rule by wrapping extended scales along a helical path. The downside of this is that such slide rules typically feature only two scales.

12 1550–03.04.1617

13 February 1561–26.01.1630

14 05.03.1574–30.06.1660

15 A comprehensive history of the slide rule may be found in a number of sources such as [→Cajori 1994] and [→Jezierski 2000]. A great introduction to the application and use of slide rules is given in [→Hume et al. 2005].

16 If all scales are just on one side of the body, a slide rule is called *simplex*. If scales are found on both sides of the body, it is a *duplex* slide rule. In this case the cursor is also used to transfer partial results from one side of the body to the other.

17 Early pocket calculators such as the Hewlett Packard HP-35 or some models made by Texas Instruments like the SR-10, etc., were explicitly marketed as *electronic slide rules*. The fixed number format featured by early HP calculators that was often set to display only 2 or 4 decimal places also was a reverence for the slide rule.

18 24.02.1785–02.02.1836

19 Cf. [→Jezierski 2000, p. 16].

20 In fact Buzz Aldrin (20.01.1930–) carried a Picket slide rule on the Apollo-11 mission, which was sold on September 20th, 2007 for $77,675.

21 [→Kaufmann et al. 1955] describe some interesting electronic circuits, most of which requiring only passive components such as potentiometers and resistors, to implement an *electronic slide rule*.

22 Apart from the ease of use, such specialized slide rules have the advantage of not requiring any electrical power or the like for their operation.

23 See [→ Hütte 1926, pp. 380 f.].

24 The first of these devices was invented by James Watt's[25] assistant John Southern[26] around 1796 (cf. [→ Miller 2011]).

27 1785–1841

28 Cf. [→ Henrici 1894, p. 505].

29 1794–1867

30 [→ Haeberlin et al. 2011] describe this instrument. See also [→ Henrici 1894, p. 500].

31 15.09.1782–21.04.1864

32 11.11.1823–03.01.1912

33 [→ Foote et al. 2007] show how to build a simple polar planimeter.

34 One notable exception is the *Prytz planimeter* (also known due to its physical shape as a *hatchet planimeter*) that was developed by the Danish mathematician and cavalry officer Holger Prytz (who published under the pseudonym "Z") around 1875. His instrument has no moving parts whatsoever but is of very limited precision. A good description of the principle of operation of this instrument can be found in [→ Foote et al. 2007, pp. 82 ff.]. Explaining why it works is by no means simple as [→ Bush 1936, p. 657] points out: "*[The] hatchet planimeter of Prytz, which works surprisingly well, and causes so much involvement when one is called upon to explain why*".

35 Cf. [→Asmar et al. 2018, pp. 177 ff].

36 Explanations of the operation of planimeters with different chains of reasoning can be found in [→Henrici 1894, pp. 179 ff.], [→Meyer zur Capellen 1949], and [→Leise 2007]. Some patents describing interesting planimeter designs are [→Coffin 1882], [→Snow 1930], etc.

37 This wheel is extremely delicate – its metal wheel tread should never be touched by hand since it is engraved with microscopic rills, which are vital for the overall precision of the instrument and are easily damaged or cluttered with dirt.

38 Typical errors caused by a wheel with an axis not being perfectly horizontal can thus be compensated to a certain degree.

39 [→Palm 2014] gives a thorough treatment of the planimeters and other integrators made by A. Ott.

40 These differ from polar planimeters in so far as the tracer arm is not free to rotate around the elbow connected to the pole arm but is guided in a strictly linear fashion, which is normally implemented by a two-wheel carriage that is dragged behind by the tracer arm while tracing the curve.

41 Although this instrument was not a financial success due to its complexity, its pickup mechanism anticipated the basic techniques used for today's touch screens and the like.

42 Information about mechanical computing elements can be found in [→Svoboda 1948], [→Meyer zur Capellen 1949], [→Willers 1943], and [→Bureau of Ordnance 1940].

43 This particular advantage of mechanical analog computers is also exhibited by digital analog computers, see →chapter 12, which makes these devices quite interesting for future computer architectures.

44 [→Belgaumkar 1967] describes an interesting approach to the design of such cams. A similar approach to the design of complex cams used in avionics systems is described in [→Huggins 1958].

45 Cf. [→Svoboda 1948, p. 22].

46 Three-dimensional cams like this have also been called *camoids*, cf. [→Svoboda 1948, p. 23].

47 See →section 4.5.8.

48 Such building blocks are also called *additive* or *linear* cells (cf. [→Svoboda 1948, p. 6]).

49 The central element carrying the bevel gears is called *spider block*.

50 Far from being trivial is the implementation of integrators due to the necessary high precision in order to minimize the accumulation of errors within a simulation.

51 See [→Petzold 1992, p. 26].

52 A similar mechanism was developed later by Tito Gonnella and used as the basis for his planimeter.

53 This particular machine featured twelve integrators of this type and was used until 1954, see →section 2.8.

54 On the left side a *torque amplifier* and a *frontlash unit* can be seen. These are typical devices in a mechanical differential analyzer but are outside the scope of this book. Refer to [→Willers 1943, pp. 236 ff.], [→Robinson[a]], or [→Fifer 1961, p. 672] for more information about these units.

55 16.02.1822–08.05.1892

56 26.06.1824–17.12.1907

57 See [→Thomson 1912, pp. 452 ff.].

58 Such double-ball integrators were used well into the second half of the 20th century. This was mainly due to their high precision – [→Librascope 1957] mentions a repeatability of 10^{-4} which is quite remarkable for a mechanical device – as well as to their robustness. Accordingly, such integrators were used quite often in aerospace applications like dead reckoning computers, etc.

59 This *classical differential analyzer technique*, as [→Korn et al. 1964, p. 1-5] put it, is described in more detail in →section 7.2.

60 20.40.1887–10.07.1960

61 Cf. [→Petzold 1992, p. 33].

62 See [→Walther et al. 1949, p. 200].

63 This requires that the integrators are not restricted with respect to the variable of integration. Only mechanical analog computers and DDAs (cf. →section 12) fulfill this requirement. Analog electronic analog computers can only use time as the variable of integration, so this class of machines requires different multiplication schemes.

64 For these machines the phrase "*substitute brass for brain*" was coined (see [→Zachary 1999, p. 49] and [→Everyday Science and Mechanics 1932]).

65 Following the death of his first wife, Margaret Thomson nee Crum on June 17, 1870, his interest for seafaring increased and he bought a 126 ton schooner, the Lalla Rookh, which in turn sparked his interest in tide prediction.

66 See [→Thomson 1882, Part I].

67 These effects are called *partial tides*.

68 Lord Kelvin's machine is on display at the Science Museum, London. More details on it can be found in [→Marsh 2024].

69 Later machines generated even more partial tides. The *United States Coast and Geodetic tide-predicting machine No. 2* that was completed in 1910 generated 37 harmonic terms. [→Aude et al. 1936] describe a machine that took 62 harmonics into account (this machine was in operation in Hamburg, Germany, until 1968 and is now on display at the *Deutsches Museum* in Munich).

70 Cf. [→ Sauer[a]].

71 Cf. [→ Berkeley et al. 1956, pp. 135 ff.].

72 A similar device, the high precision *Isograph*, which contained 10 sine/cosine-units was built at the Bell Telephone Laboratories in 1937: "*In the Isograph, gears were fitted to the bearings with an accuracy of one ten-thousandth of an inch for play and concentricity*." (see [→ Karplus et al. 1958, p. 243]).

73 See [→ Berkeley et al. 1956, pp. 132 ff.], [→ Willers 1943, pp. 168 ff.] and [→ McDonal 1956], which give a detailed description of a practical harmonic analyzer.

74 See [→ Thomson 1878] and [→ Berkeley et al. 1956, p. 133].

75 See [→ Thomson 1878].

76 19.12.1852–09.05.1931

77 18.07.1861–18.10.1931

78 See [→ Michelson et al. 1898].

79 [→ Hammack et al. 2014] describe this remarkable machine in incredible detail. Another harmonic analyzer was developed by Arnold Sommerfeld, based on the direct integration of products of the form $f(t)\sin(nt)$ and $f(t)\cos(nt)$. This machine was demonstrated in May 1891 for the first time and is described in [→ Eisenberg[a]], [→ Eckert 1997], and [→ Eckert 2013, pp. 1–21].

80 See [→ Willers 1943, pp. 178 ff.]. Tatjana Joëlle van Vark has built outstanding modern implementations of harmonic

analyzers and synthesizers (among other mathematical instruments) displaying an incredible degree of scientific craftsmanship, see →http://www.tatjavanvark.nl, retrieved 07.06.2025.

81 See [→Berkeley et al. 1956, p. 163].

82 The typical range of fire during the historic encounter between the *USS Monitor* and the *CSS Virginia* on March 9, 1862 was only about 100 meters (see [→Clymer 1993, p. 21]).

83 See [→Bromley 1984, p. 2].

84 At the battle of Jutland in 1916, the hit rate was less than 5% even though both navies were using mechanical fire control systems.

85 Details of the German torpedo control system can be found in [→Rössler 2005, pp. 79 ff.], including a schematic diagram of a typical analog lead-lag computer.

86 *TDC Mark-3* for short.

87 See [→Clymer 1993, p. 28].

88 11.03.1890–30.06.1974

89 As early as 1912 he submitted a patent describing a *profile tracer* that allowed to trace and plot a ground profile (see [→Bush 1912]). This device contained two mechanical integrators in series that took the displacement of a pendulum mass as input thus yielding

a profile trace by integrating twice over the acceleration of the mass.

90 Cf. [→Dodd 1969, p. 5] and [→Zachary 1999, p. 49].

91 See [→Zachary 1999, p. 51]. [→Puchta 1996] gives a thorough overview of his early analog computers.

92 Its construction cost about US$ 25,000.

93 See [→Willers 1943, pp. 236 ff.], [→Robinson[b]], and [→Fifer 1961, p. 672]. [→Bush 1936] gives a great account of "*Instrumental Analysis*" including details on these machines.

94 More information about this particular machine can be found in [→Meccano 1934/2], [→Zachary 1999], and [→Gleiser 1980]. A rather simple but still useful mechanical differential analyzer is described in [→Kasper 1955].

95 See [→Zachary 1999, p. 51].

96 See [→Macnee 1948, Sec. I].

97 See [→Owens 1986, p. 67].

98 It was soon called *thinking device, mechanical brain*, or even *man-made brain* by the awed press (see [→Zachary 1999, p. 51]).

99 27.03.1897–12.02.1958

100 See [→Robinson 2008] for an oral history of Arthur Porter. A great survey of Meccano set computers can be found

in [→Robinson 2005].

101 See [→Meccano 1934/1] and [→Meccano 1934/2].

102 A detailed account on these machines and scientific computation in general in Britain can be found in [→Porter 2003] and [→Croarken 1985].

103 This machine is now part of the collection of the *Museum of Transport and Technology*, Auckland, New Zealand: →http://www.motat.org.nz/explore/objects/differential-analyser, retrieved 30.11.2012.

104 Cf. [→Holst 1982] and [→Holst 1996].

105 18.03.1894–19.01.1985

106 A bit of trivia: When Germany occupied Norway during World War II, Rosseland removed the integrator wheels from the machine and buried them safely packaged behind the institute to make sure that his differential analyzer would not support Germany's war efforts.

107 See [→Eterman 1960, pp. 39 ff.].

108 See [→Walther et al. 1949].

109 The *Rockefeller Foundation* initially funded this machine in 1935.

110 See [→Owens 1986] for details on that particular machine.

111 See [→Sauer et al. 1944].

112 See [→University Bulletin 1978]. A noteworthy study performed on this machine is described in [→Guibert et al. 1949].

113 This machine was primarily intended to be used in education, see [→Krause 2006].

114 The problem of determining the area under a curve is, in fact, so simple and could be solved by employing a planimeter (see →section 2.4) so that such a setup would never have been found in a practical differential analyzer installation.

115 See [→Lemaitre et al. 1936/1] and [→Lemaitre et al. 1936/2].

116 See [→Soni et al. 2017, p. 30].

117 There is an incredibly realistic simulator for such differential analyzers allowing one to setup actual problems: →https://differentialanalyser.github.io, retrieved 20.05.2025.

118 Another such electromechanical differential analyzer was built in Germany from 1939 to 1945. This particular machine contained four integrators, six function tables with optical sampling, and two multipliers. This machine was used successfully and commercially well into the late 1950s. More information can be found in [→Walther et al. 1949].

119 See [→Owens 1986, p. 78]. This capability will be of central importance for highly integrated analog computers for the 21st century – see →chapter 16.1.

120 Details on this machine can be found in [→Bush et al. 1945, pp. 275 ff.].

121 See [→Berkeley et al. 1956, p. 117].

122 See →https://www.computerhistory.org/revolution/analog-computers/3/143, retrieved 02.02.2022.

123 [→Mindell 2002] contains additional information on this machine.

124 These function tables were based on input tables and allowed to trace a curve given an *x* coordinate as input yielding the *y* value of the curve at this particular point.

125 This last large Minden system was used by the German company *Siemens-Schuckert* until 1971 (see [→Petzold 1992, p. 53]). Most parts of the machine are now part of the collection of the Deutsches Museum in Munich.

126 At the time of this writing it is quite unclear what purpose this inverse mode of operation should have served.

127 The underlying servo system is described in [→Bückner[a]].

128 Arnold Theodore Nordsieck, 05.01.1911–09.01.1971

129 This machine was notorious for giving its programmers electric shocks: "*The plugging operation is unfortunately accompanied by an electric shock hazard (though hardly a dangerous one) [...]*", see [→Brock 2017].

130 22.11.1912–20.07.2002

131 See [→Bückner 1950], [→Bückner 1953], and [→Erismann 1962, pp. 190 ff.].

132 See →section 12.

133 See [→Eggers 1954].

134 Here and in the following a dot over a variable denotes its derivative with respect to time, i. e., $\dot{y} = \frac{\mathrm{d}y}{\mathrm{d}t}$. Accordingly, two dots represent the second time derivative, etc.

1 In 1938 Hans Kleinwächter (1915–26.10.1997) proposed an electronic analog computer based on special cathode ray tubes, see [→Kleinwächter 1938]. Unfortunately this early foray into electronic analog computing seems never to have been pursued further.

2 27.02.1912–19.08.1996; a highly interesting interview with Dr. *Hoelzer* can be found in [→Neufeld et al. 1989].

3 See [→Hoelzer 1992, p. 4]: "*[Es gab] in der ganzen Fliegerei nicht ein einziges Gerät […], welches die absolute Geschwindigkeit eines Flugzeuges […] gegenüber der Erde messen kann. Aha, dachte ich, das ist ja ganz einfach, man nimmt die Beschleunigung, die man ja messen kann, integriert sie und* – voilà! – *hier ist die Geschwindigkeit.*"

4 See [→Lange 2006], [→Reisig 1999], and [→Dungan 2005] for detailed information about the A4 rocket. The abbreviation "A4", short for "aggregate 4", denoted the rockets used during the extremely tedious and long

development process, while the actual weapon system became known as the “V2”, German for “Vergeltungswaffe 2”, “vengeance weapon 2”.

5 After the war Telefunken played a major role in the development of analog computers in the European market.

6 23.03.1912–16.06.1977

7 11.02.1908–02.12.1987

8 German for *guidance, steering, and onboard instrumentation*.

9 An incredible wealth of bibliographic information on the development of German guided missiles can be found in [→T-2 1946].

10 See [→Tomayko 1985, p. 230]. One of the resulting radio guidance systems was named *Hawaii I* and was built by Telefunken (see [→Lange 2006, pp. 163 ff.]).

11 A dot over a variable denotes its derivative with respect to time.

12 I. e., there is no active element such as an operational amplifier in the circuit.

13 A circuit consisting of a resistor and a capacitor.

14 The name stems from the fact that the central four diodes are often arranged in a ring-like structure.

15 See [→Aschoff 1938] and [→Chance et al. 1949, p. 379].

16 Short for *Siemens-Rundfunk-Detektor*.

17 See [→Tomayko 1985, p. 230].

18 05.10.1881–10.08.1945

19 [→Bate et al. 1971] give a thorough description of guidance and control of ballistic missiles in general.

20 In today's parlance this would be a P-controller, using only a proportional feedback path.

21 See [→Tomayko 1985, p. 232].

22 The control computer used in the *Jupiter* rocket was still called "mixing computer" (see [→Grimwood et al. 1962, p. 152].

23 A much more in-depth treatment of the Mischgerät from the British perspective can be found in [→Hudson et al. 1945]. Hoelzer, too, wrote in great detail about this device in the second (secret) part of this Ph. D. thesis (see [→Hoelzer 1946/2]) and in [→Hoelzer 1976]. Another in-depth treatment can be found in [→Müller 1957].

24 Short for *Umformer*, German for a motor-generator unit.

25 Using simple RC circuits directly to generate time derivatives and integrations yields quite imprecise results. Since the gyro based control scheme for the A4 rocket only required linear terms and first and second derivatives, this was not a problem as long as the controller itself was operating in a stable way. More precise computing circuits as used in general purpose

analog computers would have been too costly, too large and too complicated for the intended use of the rocket.

26 See [→NASA 1964, p. 1-5, p. 4-19]. A detailed description of the guidance and control system of the Redstone rocket can be found in [→Chrysler Missile Division, pp. IV-1–IV-32].

27 See [→NASA 1968, pp. 7-15 ff.].

28 See [→Hosenthien et al. 1962, pp. 449 ff.].

29 See [→Tomayko 1985, p. 234].

30 See [→Hoelzer 1992, pp. 13 f.].

31 "*Mein Chef kam ins Labor, sah [den] elektronische[n] Drahtverhau und sagte nur: „Hoelzer, hören Sie doch endlich auf mit dieser elektrischen Spielerei und kümmern Sie sich um Ihre Aufgabe. Ab morgen ist das alles weg, verstanden?" Ich sagte das einzige, was man in solcher Situation sagen konnte: „Jawohl, Herr Doktor." Am nächsten Morgen war alles weg und zwar war es jetzt in einem kleinen Raum ohne Fenster […] Aber mir schien es auch wichtig, ein Gerät zu schaffen, welches, wie man damals dachte, in der Hauptsache für die Entwicklung von Raketensteuerungen von nicht zu überbietender Wichtigkeit war. […] Als alles funktionierte, wurde mir dann vergeben*."

32 See [→Petzold 1992, p. 57].

33 In 1949 Edwin A. Goldberg applied for a patent that described a practical scheme of eliminating the drift effects in DC amplifiers. It was this invention that made

the cumbersome AC coupled amplifier approach with the ring modulators, etc., superfluous and the subsequent rapid development of analog computers possible (see →section 4.1.2).

34 The symbols in the schematic differ from those used today and are those used in HOELZER's dissertation.

35 The abbreviation *gl* denotes *glätten*, German for smoothing.

36 Differentiators are nowadays normally avoided in electronic analog computers since they naturally tend to increase the noise of a signal excessively. HOELZER alleviated this problem by the inclusion of this low pass filter.

37 A lot of information on that can be found in [→Peenemünde]. The system evolved into the *Hermes A-1* under the development of General Electric in 1935.

38 Explicit summers were not necessary since the modulated signals used to transmit values from one unit to another could simply be fed into the multiple transformer coupled inputs of the computing elements. These series connected transformers (see →figs. 3.9 and →3.10) then implicitly performed the summing operation.

39 See [→HOELZER 1946/1].

40 Sign reversals are not taken into account in this and the following circuits.

41 See →section 7.2.

42 Later analog computers avoided differentiators wherever possible, but this required some substantial technological advances, most notably very high gain drift stabilized DC amplifiers and high precision computing capacitors.

43 See [→Hosenthien et al. 1962] for a detailed description of this particular system as well as its successors.

44 In 1993 Helmut Hoelzer started building a replica of this first analog computer, which was completed in 1995. This machine is now part of the collection of the *Deutsches Technikmuseum Berlin*.

45 This is the machine shown in →figure 3.17.

46 See [→Hosenthien et al. 1962, p. 454] for a description of the simulation of injection attitude control for Explorer I and [→Tomayko 2000, p. 236] for general information on this system.

47 Details on the construction of these tubes can be found in [→Ratheiser[a], pp. 29 ff.].

48 05.01.1913–01.12.1974

49 Founded as *Industrial Instrument Company* in 1908, Foxboro (now owned by Schneider Electric) still is one of the leading brands of industrial control and measurement systems.

50 The term simulator is defined by [→Holst 1982, p. 144] as follows: "*A simulator is a fixed (to a large degree) structure embodying one unique model. […] A simulator's purpose is specific: to provide the accurate realization of its*

model for various parameters, stimuli, and operator interactions of interest to its users."

51 Cf. [→Holst 1982, p. 143].

52 Cf. [→Holst 1982, p. 149].

53 Cf. [→Holst 1982, p. 152].

54 See [→Holst 1982, pp. 153 f.].

55 See [→GAP/R Evolution].

56 Cf. [→Holst 1982, pp. 155 f.].

57 See →section 4.1.

58 See →section 4.1.

59 See [→Clymer 1993, p. 30].

60 In [→Fagen 1978, p. 135] Parkinson describes a lucid dream that led him to this development. See also [→Zorpette 1989].

61 It is remarkable that "*[n]either Parkinson nor Lovell had any experience in fire control; they did not even know of the existence of mechanical gun directors*" as [→Fagen 1978, p. 137] states.

62 See [→Mindell 1995, p. 73].

63 See [→Lovell et al. 1946].

64 See →section 4.1.

65 See →section 4.2.

66 These could also be called *inverters*.

67 See [→Higgins et al. 1982, p. 225].

68 See [→Fagen 1978, pp. X f.].

69 Eventually more than 3000 M-9 gun directors were built and deployed.

70 Cf. [→Fagen 1978, p. 148].

71 See [→Zorpette 1989].

72 [→Boghosian et al. 1950] give an impression of the complexity of these fire control computers.

73 Another system that performed all calculations in Cartesian coordinates is described in [→Bedford et al. 1952].

74 See [→Dehmel 1949].

75 See [→Dehmel 1954].

76 See [→Macnee 1948, p. 1].

77 Even not too low compared to our modern day stored-program digital computers in many cases.

78 This function generator was based on optical feedback as described in →section 4.5.3.

79 See →section 4.6.2.

80 See →section 8.9. A very interesting numerical treatment of the MATHIEU equation on the ENIAC done in 1948 is described in [→ZAROODNY 1955].

81 6.10.1846–12.03.1914

82 In the early days of electrification there were two competing systems: one using DC currents, pioneered by *Thomas Edison* (11.02.1847–18.10.1931), and one based on AC currents, introduced by *Westinghouse*.

83 See [→TRAVERS et al. 1930] and [→ASPRAY 1993, p. 38].

84 See [→TYMPAS 2003] for details on such calculating boards as well as earlier methods. [→LYON 1950] gives a wealth of examples based on a similar British system.

85 See [→NEBEKER 1994, p. 165].

86 See [→TRAVERS et al. 1930].

87 12.01.1912–09.04.2003

88 See [→NEBEKER 1994, pp. 179 ff.].

89 28.04.1905–24.05.2004

90 A comprehensive description of Anacom, its design and implementation, can be found in [→OODRYS 1953]. HARDER's long and highly interesting and influential life is described in detail in [→ASPRAY et al. 1991], [→ASPRAY 1993], and [→NEBEKER 1994].

91 See [→NEBEKER 1994, pp. 179 ff.].

92 See [→Nebeker 1994, p. 197] and [→Oodrys 1953, pp. 16 f.].

93 [→Grandi et al. 1951] give a wonderful pictorial overview of the various analog computers installed at Caltech in 1951.

94 A short description of this system can be found in [→McCann 1959, p. 26].

95 See [→Nebeker 1994, pp. 184].

96 See [→Nebeker 1994, p. 186].

97 See [→Aspray et al. 1991].

98 See →section 10 for details on this type of machine.

99 See [→Nebeker 1994, pp. 181 ff.] and [→Aspray 1993, pp. 43 f.]. The latter source shows a diagram of the structure of Anacom III.

100 See [→Nebeker 1994, p. 191].

101 An interesting study of ideal transformers and their application in analog computers can be found in [→MacNeal et al. 1955].

102 See [→Dixon 1952, pp. 44 f.] and [→Oodrys 1953, pp. 28 ff.].

103 One can imagine how bulky these inductors were with their massive iron cores.

104 See [→Oodrys 1953, pp. 36 ff.].

105 See [→Caltech 1949]. A wealth of information on the application of such passive analog computers can be found in [→Paschkis et al. 1968] and [→Volynskii et al. 1965].

106 See [→Oodrys 1953, pp. 50 ff.].

107 See →section 4.5.6.

108 See [→Schultz et al.], [→Oodrys 1953, pp. 44 ff.], and →section 4.5.3.

109 See [→Nebeker 1994, p. 185].

110 See [→Nebeker 1994, pp. ff. 185].

111 See [→Harder et al. 1950]. This source also contains a wonderful description of the overall system.

112 See [→Nebeker 1994, pp. ff. 185].

113 This could either be a diode function generator or a photoformer, denoted by a rhombus sign.

114 See [→Dixon 1952, pp. 48 ff.].

115 See [→MacNeal 2002, pp. 5 ff.].

1 A comprehensive treatment of various error sources and characteristics of passive components in analog computers can be found in [→Single et al. 1961].

2 Most classic analog computers use voltages except machines made by Applied Dynamics, which used currents. Modern implementations of analog computers

on integrated circuits will mainly use currents to represent values.

3 These two voltage ranges are not the only ones to be found in actual machines – some tube based circuits work with values in the range of ±50 V and some transistorized analog computers even allow values between +100 and −100 V while others work with voltages ranging from −5 V to +5 V.

4 See →section 4.1.2.

5 A comprehensive account of the history of operational amplifiers can be found in [→Jung 2006], which is very worthwhile to read.

6 1912–22.11.1988

7 See [→Ragazzini et al. 1947]. It should be noted that the developments on which this paper is based were mostly done by Loebe Julie, (10.12.1920–07.06.2015) who was working under the auspices of George A. Philbrick at Bell Labs (cf. [→Jung 2006, p. 779]).

8 14.04.1898–11.12.1983

9 A detailed account of Black's work is given in [→Kline 1993] and [→Mindell 2000].

10 It should be noted that Black was not the only developer working on negative-feedback schemes – others included Paul Voigt, 09.12.1901–09.02.1981, (in memoriam [→Klipsch 1981]), Alan Blumlein, 29.06.1903–07.06.1942, and Bernardus Dominicus Hubertus Tellegen, 24.06.1900–30.08.1990 (cf. [→Jung 2006, p. 767]).

11 Gustav Robert Kirchhoff, 12.03.1824–17.10.1887

12 1 nA = 10^{-9} A.

13 Magnetic amplifiers were suggested for application in analog computer but never really caught on and will, thus, not be treated in the following. More details on this type of amplifier can be found in [→ Craig 1953], [→ Davis et al. 1955], and especially [→ Davidson 1953].

14 See →section 15.15.8.

15 The schematic of the M-9 operational amplifier can be found in [→ Jung 2006, p. 781].

16 See [→ Pease 2003].

17 As [→ Pease 2003] notes, this is not the first “modern” operational amplifier – there were earlier developments, but none could match the K2-W in terms of practicability and commercial success.

18 Not bad for an operational amplifier – even from today’s perspective.

19 This configuration is called a *long-tailed pair*.

20 Nowadays, a *Zener diode* would be used instead of a neon tube. In early specimens of the K2-W the level shifter consisted of a *thyrite* device – today this would be called a *varistor*. These are made from silicon carbide and were not only used as level shifters in the early days of electronic analog computers but also as function generators in applications where precision was not of

prime importance. Today such devices are commonly used as surge arrestors.

21 The schematic of the K2-X operational amplifier can be found in [→Russell 1962, p. 6-20].

22 See [→Goldberg et al. 1954].

23 These relays are also called *vibrators*, especially when used in power supplies for battery powered mobile vacuum tube devices such as car radios, etc.

24 These choppers were typically driven by a 60 Hz or a 400 Hz signal resulting in a very characteristic audible hum emitted by such amplifiers.

25 The output voltage of the drift compensation stage is often also used to detect amplifier overload. If an amplifier is overloaded, it cannot maintain the necessary feedback signal to keep the summing junction at ground potential, which is reflected by an excessively high output voltage from the AC amplifier, which can be easily detected. Such overloads may result from improper setup of the analog computer typically due to scaling errors regarding the underlying equations. Since an overloaded amplifier will yield an erroneous result, overload conditions are normally indicated visually or acoustically. Ideally a computation run is halted in case of an overload so that its cause can be determined by checking which amplifier went into overload.

26 Open loop gain is the gain exhibited by an amplifier without any external negative feedback path in place.

27 See [→Jung 2006, p. 780].

28 *Precision Analog Computing Equipment*

29 Details on grounding problems in analog computers, which can be quite intricate, can be found in [→Edwards 1955].

30 Of course, this high gain quickly falls off with increasing input signal frequency.

31 See [→Howe 1961, p. 113].

32 [→Meyer-Brötz et al. 1966] gives a detailed description of a follow-up development of this amplifier based on more modern silicon transistors.

33 Mean Time between Failure

34 The single integrated circuit, a LM709 operational amplifier, forms the AC coupled amplifier of this circuit.

35 19.06.1907–23.04.1998

36 Cf. [→Jung 2006, pp. 777 f.].

37 Many analog computers allow direct connections to the summing junction of the operational amplifiers used in summers and integrators. Typical applications include (biased) diodes for amplitude limiting and little capacitors between the output of an amplifier and its summing junction to suppress parasitic oscillations, which can occur in setups like square rooting, etc.

38 This is a source of confusion for most beginners in analog computer programming.

39 If an input's weight differs from 1, it is written next to the input in the triangle of the simplified summer symbol shown in →figure 4.20. If no weight is noted, the corresponding input is weighted with 1.

40 Some analog computers from the 1950s employed a simpler circuit allowing the integration capacitor to be discharged only prior to switching the integrator into OP mode. Some circuits used neon tubes to short circuit the integration capacitors. Little neon tubes were enclosed in a coil that could be energized by a RF signal, which in turn caused the tube to conduct. Since this does not provide any means to preset an integrator to $e(0)$ this actual initial condition value had to be introduced later in the computer setup by a summer or the like. A schematic of a Philbrick K2-J integrator with a neon reset tube triggered by an RF generator can be found in [→Russell 1962, p. 6-24].

41 Due to small leakage currents in the capacitor and even the tiniest drift effects the output of the integrator will slowly change during halt mode. Accordingly, this mode should not be active for extended periods of time.

42 These are called *time-scale factors*, since they allow faster or slower operation of integrators with respect to integration time.

43 The last term in equation (→4.7) even requires a sawtooth signal for interpolation – the overall circuitry is quite complex.

44 Input and output of a coefficient potentiometer are not explicitly denoted in this representation form although they cannot be used interchangeably. In fact, connecting the output of a coefficient potentiometer to the output of a computing element can and most often will damage the potentiometer in a classic analog computer, so care should be taken.

45 A linear potentiometer is assumed here.

46 Later and more modern implementations used buffer amplifiers for the coefficient potentiometers as shown below.

47 In some cases load-correction graphs were used to give the required offset of α for some typical loads.

48 A compensation voltmeter basically consists of a reference voltage source, i. e., a precision potentiometer connected as a voltage divider between ground and +1 machine unit, and a null voltmeter connected between the output of this voltage divider and an input connection. Adjusting one of the inputs so that the null voltmeter is no longer deflected guarantees that there is no load acting on either the input source or the voltage divider generating the comparison voltage, thus values can be set very precisely with little effort.

49 In addition, an output buffer ensures that a potentiometer cannot be damaged by feeding a voltage into its output.

50 Cf. [→ BADER 1985, pp. 76 f.].

51 See →section 10.

52 These are basically multiplying *digital-analog-converters* (*DAC*).

53 This was one of the problems which led to the development of *hybrid computers* (see →section 10). Using a stored-program digital computer coupled to an analog computer, the digital system can be used to generate functions by suitable algorithms or table lookups.

54 An approach using an analog computer to generate input functions for another analog computer to process, the *function fitter*, is described in [→Keefe 1955].

55 See [→Loveman 1962, pp. 3-36 ff.] for more information about tapped potentiometers.

56 See [→US Airforce 1957].

57 See [→Erismann 1962, pp. 204 ff.].

58 An early account of this technique can be found in [→Mynall 1947].

59 Cf. [→Macnee 1948, Sec. IIIE], [→Haug 1960, p. A3], and [→Korn 1962, pp. 3-68 ff.] for more detailed information. Even a Williams-tube like construction was once employed. This proved useful in repetitive operation of high-speed analog computers when the result of one run had to be used as the input for the next run, see [→Haug 1960, p. A3].

60 See [→Korn 1962, pp. 3-73 ff.].

61 Cf. [→Korn 1962, p. 3-75].

62 See [→GAP/R 1959].

63 See [→Kovach et al. 1960] for details.

64 Cf. [→Korn et al. 1964, pp. 6-14 f.] for more details.

65 17.10.1919 (11.10.1919 in other sources)–11.07.2014

66 This prize is not to be confused with the *Nobel Prize*. It is awarded by the *American Society of Civil Engineers* for outstanding work in the fields of engineering.

67 See [→Koenig 1950].

68 More information can be found in [→Korn 1962, pp. 3-66 ff.] and [→Haug 1960, p. A4] for example.

69 See [→Gilbert 1982] and [→Analog Devices/2].

70 At least as long as there is no change of sign and the feedback function is reasonably smooth – in most cases it is advisable to add a small capacitor, about several ten pF, between the output of the amplifier and its summing junction to stabilize the circuit.

71 An operational amplifier with no (implicit) feedback path is called open amplifier.

72 Some basic information about this topic can be found in [→Korn 1962, pp. 3-78 ff.] and [→Haug 1960, pp. A9 f.].

73 Cf. →figure 2.7.

74 See →section 2.5.1.

75 See [→ Gallagher et al. 1957, p. 87].

76 See [→ Berger et al. 1979/1].

77 A good example of this is described in [→ TU Wien 1976/3, pp. 27 ff.].

78 In the simplest case such an SAH element can be implemented using an integrator which is cycled between IC and HALT mode.

79 A comprehensive classification of analog multipliers can be found in [→ Celinski et al. 1964].

80 A good overview of early multiplication techniques can be found in [→ Czajkowski 1956/1] and [→ Czajkowski 1956/2].

81 Typically, these potentiometers feature a grounded middle tap, thus enforcing that $0 \cdot y$ is as close to the value zero as possible regardless of any unavoidable little nonlinearities in the potentiometer itself.

82 Servo multipliers typically feature dedicated power supplies for the actual servo circuit to minimize noise coupling into the precision supplies of the analog computer.

83 Cf. [→ Haug 1960, pp. A6 f.].

84 Alternatively, a shield dividing the tube screen into two equal halves with two photodetectors has also been used successfully.

85 02.02.1912–27.01.1994

86 See [→Gundlach 1955], [→Schmidt 1956], and [→Klein 1965].

87 See [→Haug 1960, p. A7].

88 This tube is not too dissimilar to the crossed-fields electron-beam multiplier described before but without the external deflection coil which has been replaced by the hyperbolic deflection plates.

89 This multiplier was part of a large analog computer, *ELARD*, developed and built at the Technical University of Darmstadt by Walter Dhen, see [→Dhen 1959].

90 This specimen is part of the *Informatiksammlung* of the *Darmstadt University of Applied Sciences*.

91 Unfortunately, Somerville was unable to perform dynamic tests due to a tube failure (see [→Angelo 1952, p. 3]).

92 See [→Korn et al. 1964, pp. 7-10 ff.]. Other terms for this device are *mark-space multiplier* or *pulsed-attenuator multiplier*.

93 This is the ratio of the mark- and space-times of the signal.

94 An early and very detailed circuit description of such a time division multiplier can be found in [→Lilamand 1956]. A modern implementation might use a switched capacitor building block as described in [→Linear Technology, p. 9].

95 Using modern integrated circuit analog switches, very low cost multipliers can be implemented using this basic approach.

96 Cf. [→Morrill 1962, pp. 3-41 f.].

97 The function generator card shown in →figure 4.39 is actually part of a quarter square multiplier.

98 An interesting variation of this multiplication technique has been used by Hitachi in the 505E analog computer. Its multipliers are based on

$$xy = \frac{1}{2}\left(\frac{(x+y-1)^2}{2} - \frac{(x-y-1)^2}{2}\right) + y,$$

see [→Hitachi 505E, p. 16].

99 See [→Dassow 2015], [→Miller et al. 1954], and [→Miller et al. 1955].

100 [→Celinski et al. 1964, p. 152] report up to 5 kHz bandwidth for these multipliers.

101 [→Gul'ko 1961] describes an interesting multiplier circuit using such varistors with increased passband.

102 See [→Husinsky et al. 1985, pp. 15 ff.].

103 See [→Morrill 1962, p. 3-40].

104 Cf. [→Morrill 1962, p. 3-41].

105 See [→Savet 1962].

106 Cf. [→Haug 1960, p. A7].

107 [→Edwards 1954] contains a comprehensive overview of various historic multiplication schemes used in analog computers.

108 See also →section 13.

109 See [→Stata 1968].

110 This work continues to describe a large number of high precision hybrid computing elements including function generators, integrators, and coordinate resolvers.

111 See [→Sheingold 1978, p. 37]. This booklet also contains a wealth of highly interesting application examples.

112 05.06.1937–30.01.2020

113 See [→Gilbert 1968/1], [→Gilbert 1968/2], and [→Analog Devices 2008]. A similar multiplier capable of operation up to 10 MHz is described in [→Morton 1966].

114 A detailed description of this circuit can be found in [→Sheingold 1978, pp. 38 ff.].

115 The multiplier is also often denoted with a “Π” instead of the “×” symbol.

116 Cf. [→Giloi et al. 1963, p. 80]. The summing junction is denoted by *S*.

117 Usually it is $R_1 = R_2$.

118 Of course these bias sources are not actual batteries in a real implementation – they can be readily implemented by either using coefficient potentiometers

introducing a bias voltage or the two diodes can be altogether replaced by Zener diodes.

119 Sometimes this simple circuit is insufficient since the slope of the limited portion of the signal is not zero. See [→Giloi et al. 1963, pp. 207 ff.] and [→Korn 1962, pp. 3-62 ff.] for implementation variants yielding more precise results.

120 See →chapter 3.3.

121 [→Carlson et al. 1967, pp. 58 ff.], [→Giloi et al. 1963, pp. 95 ff.], [→Loveman 1962, pp. 3-2 ff.], [→Morrill 1962, pp. 3-56 ff.], [→Kley et al. 1966], and [→Vogel 1977] contain details and implementation examples.

122 During start-up time, $t < \Delta t$, the output is usually zero.

123 A typical example used in electronic music applications is the MN 3007 integrated circuit, see [→Panasonic 3007].

124 [→Kennedy 1962, pp. 6-8 ff.].

125 Apart from questions regarding the sampling period and the resolution of the ADC and DAC.

126 For more details on this unit see [→Howe 1961, pp. 226 ff.] and [→Kennedy 1962, pp. 6-10 ff.] for general information on tape delay units.

127 Cf. →section 4.5.2.

128 See [→Carlson et al. 1967, p. 225].

129 17.12.1863–09.06.1953

130 Pierre-Simon Laplace, 23.04.1749–05.03.1827

131 See [→Rideout 1962, pp. 12-5 f.], [→Korn et al. 1964], [→Giloi et al. 1963, pp. 357 ff.], [→Laning et al. 1956], and [→Korn 1966] for application examples and programming techniques.

132 A classic, yet comprehensive treatment of noise in electronic circuits can be found in [→Bell 1960] and [→Beneking 1971].

133 See [→Korn 1962, pp. 81 ff.].

134 See [→Howe 1961, pp. 221 ff.] for details.

135 An incredibly detailed treatment of such algorithms can be found in [→Knuth 1998, pp. 1– 193]. A comprehensive treatment of pseudo-random processes is given in [→Newland 2005, pp. 167-188].

136 With full knowledge of the parameters and internal state of such a pseudo-random generator, its output sequence can of course always be predicted.

137 It should be noted that this pseudo-random generator should not be used for any serious work as it has severe drawbacks regarding the apparent randomness of its output values.

138 See [→HP 3722A].

139 See →section 9.

140 This selector was often based on stepping relays in early analog computers, which were then replaced by readout relays located at every computing element. These readout relays could then connect the output of an individual computing element to a central readout bus which is connected to the input of the central digital voltmeter.

141 This particular display is based on the projection of digits onto a screen, i. e., every place of the display contains up to ten incandescent bulbs. Between each bulb and the screen of that particular digit a stencil with some symbol or digit is placed. Powering one of these tiny light bulbs will then project the desired shape.

142 Early implementations of this digital voltmeter used relay logic to decode the binary result to drive the incandescent lamp displays. This route was taken since relays were cheaper than comparable transistor logic in the first half of the 1960s. The sound of such a decoder during operation is truly impressive.

1 See →section 10.

2 In contrast to other systems, the EAI 580 requires the programmer to setup integrators by explicitly interconnecting operational amplifiers with dedicated integrator circuits. This approach has the distinct advantage of giving maximum freedom to the programmer. On the other hand, this approach breaks with the idea of a mathematical machine as some knowledge of the inner workings of the system is required in order to program it.

3 A trunk line is a direct connection between a patch panel connector and a connector on the back of the machine where input/output-devices may be connected in a convenient way without further cluttering the analog patch panel.

4 Cf. →figure 4.38.

5 See →chapter 9 for details regarding digital control units.

6 A flip-flop is a circuit exhibiting two stable states which can be used to store a single bit of information. In its simplest form it can be switched between states by means of set/reset control inputs.

7 Using such switches differing gear ratios can be selected in the automatic transmission example.

8 Some machines like the Telefunken RA 770 feature an intricate control and timing system that allows operating modes like this to be implemented without having to resort to individually patched digital control elements.

9 Normally the computer is placed into halt mode for readout.

10 In some cases the drift of an amplifier is so excessive that the chopper-stabilization stage cannot compensate it completely. Consequently, each amplifier has a potentiometer for manual coarse zeroing.

11 The key labels have faded away due to the age and usage of the particular machine depicted.

12 See →section 10.

13 This mode is quite similar to the initial condition mode, but instead of initial conditions static check voltages can be applied to integrators allowing a check of a program without actually running it.

14 “A” denotes an amplifier, “P” a potentiometer, etc.

1 In fact, many influential systems, such as the EAI 580, the EAI 680, the large systems EAI 7800, and EAI 8800, as well as many vendors like Beckman, COMCOR, Dornier, Simulators Inc., Goodyear, Solartron, etc., are not described. This should not imply that their machines are inferior to those described here.

2 *RA 1* is the abbreviation for *Rechner, Analog* number one. “Rechner” is the German term for computer. Telefunken followed a simple scheme for naming its analog computers, which were all denoted by *RA* followed by a number. Later tabletop machines were labeled with an additional *T* to distinguish these from the larger systems.

3 Of these eight units, seven use dual diode vacuum tubes EAA 91 (equivalent to the 6AL5) for the square functions while one module is of a clearly experimental nature: This left most multiplier uses unlabeled germanium diodes instead of vacuum tubes.

4 Many analog computers contain *free* or *uncommitted* diodes, which are very useful to limit the output of computing elements, which can be used to implement functions such as absolute value, etc.

5 A contemporary description of this machine with potential application can be found in [→Herschel 1957].

6 See [→Telefunken 1958].

7 A short history of these developments in the years from 1938 to 1957 can be found in [→Coulter[a]].

8 See →section 4.1.1.

9 See [→GAP/R EAC, p. 3].

10 See [→GAP/R EAC, p. 3].

11 The system shown here is equipped with servo potentiometers. These could be set automatically from an attached setup system or a digital computer.

12 Short for *Digital Voltmeter*.

13 Some background information about ADIOS can be found in [→Wauve 1962].

14 Adjusted for inflation this corresponds to roughly 9.2 million USD in 2022.

15 Cf. [→MTW 1964/1].

16 Prof. Dr. Meyer-Brötz told the author that he was hired without having any real experience in the field of amplifier development because he did not know that it was considered impossible to build a fully transistorized precision operational amplifier. Nearly everybody in the 1950s considered transistors to be well suited for

switching operation but not up to the task for precision analog computer applications.

17 Interestingly, this vacuum tube module looks as if it might be from a RA 463/2, but it is not as it is much more densely packed. As of now it is unclear to which Telefunken system this module belongs – it might very well have been a prototype, too.

18 Short for *analog tabletop computer* (German *Rechner Analog Tisch*).

19 Later models had a different color scheme and a slightly different enclosure.

20 Even a removable patch panel was offered as an option due to customers requesting some way to quickly change the setup from one program to another.

21 To protect these expensive potentiometers against damage caused by patching errors, a small incandescent bulb is placed in series with the wiper. Under normal circumstances only a tiny current flows through the wiper and the small resistance of the bulb in its cold state does not introduce any noticeable error into the computation. In case of a erroneous connection resulting in excessive current flowing through the potentiometer’s wiper, the bulb lights up and limits the current to a value small enough so that it will not damage the potentiometer.

22 Eight of these operational amplifiers are committed to switchable integrators/summers, seven are part of

dedicated summers while the remaining four are needed for the multipliers and function generators.

23 A removable patch panel was later offered as an option.

24 This is in sharp contrast to other systems such as all of the Telefunken made machines, which hide the inner workings of their computing elements and only offer mathematical operations as the basic elements for a computer setup.

25 See →section 10.

26 Short for “Digital-Experimentierzusatz”, which translates roughly to “digital extension for experiments”. See →section 9.

27 These amplifiers, described in [→Meyer-Brötz et al. 1966], and shown in →figure 4.17, →chapter 4, have an open-loop gain of $3 \cdot 10^8$.

28 German for *hybrid coupler*.

29 This system was based on the earlier EAI 580 analog computer, now equipped with the necessary interface electronics for attaching a digital computer.

30 This computer was introduced in 2021. There were similar small scale systems aimed at the educational market much earlier such as the *Donner 3500* (see [→Donner]), the table top differential analyzer developed and described by C. E. Howe in 1953 (see [→Howe 1953]), or the really small *FLEPAC* (see [→Green 1959]).

31 See →https://the-analog-thing.org and [→Cass 2023].

32 See →https://github.com/anabrid/the-analog-thing, retrieved 19.03.2025.

33 See →appendix A.

1 A much more comprehensive account of analog (and hybrid, for that matter) computer programming can be found in [→Ulmann 2023/1].

2 Of course, this first step is also required for developing an algorithmic solution for a given problem.

3 The value representation within an analog computer is thus similar to fixed point binary arithmetic or the mantissa part of a floating point number.

4 Cf. [→Schwarz 1971, pp. 25/165 ff.].

5 See [→Schwarz 1971, pp. 168 ff.].

6 It should be noted that an electronic analog computer can only integrate with respect to time, while classic mechanical or electromechanical differential analyzers as well as their digital counterparts (see →chapter 12) do not have this restriction. [→Johnson et al. 1961, pp. 2–13] describe an electronic generalized integrator.

7 See →section 3.1.2.

8 See [→Korn et al. 1964, p. 1-5].

9 The alternating signs are caused by the sign-changing nature of practical integrators.

10 Typical examples are the simulation of processes such as neutron dynamics in a nuclear reactor or problems in population dynamics.

11 Also known as *partial feedback method*.

12 Cf. [→Schwarz 1971, pp. 168 ff.].

13 Since every operation causes some inevitable errors due to noise or drift, analog computer programs should always use the minimum number of computing elements.

14 In the case of nonlinear differential equations the handling of initial conditions can be quite tricky. Sometimes special computer setups are used. Cf. [→Soudack 1968] and [→Brown 1969] for more information. Some special classes of nonlinear differential equations can be tackled with a method different from those described here, see [→White 1966].

15 Most often derivatives with respect to time *t* and Cartesian coordinates (*x, y, z*) or spherical coordinates (*t, φ, θ*) are encountered.

16 Using a multiplier, an integration can be performed with respect to another variable than time according to $\int_0^t y(t)\,\mathrm{d}(x(t)) = \int_0^t y(t)\frac{\mathrm{d}x(t)}{\mathrm{d}t}\,\mathrm{d}t = \int_0^t y(t)\dot{x}(t)\,\mathrm{d}t,$, which is basically a Stieltjes integral. The required function $\mathscr{e}$ is typically available in an analog computer setup and thus poses no additional overhead.

17 See [→Howe et al. 1953], [→Howe 1962], or [→Forbes 1972].

18 In some cases a problem variable other than t is identified with machine time τ. In this case all derivatives with respect to t must be discretized accordingly.

19 [→ Bryant et al. 1962, pp. 37–48] and [→ EAI 1965] describe a very similar problem.

20 It was not uncommon for some classic computer setups, especially in the area of chemical engineering or aerospace technology, to require hundreds of integrators and summers to model heat transfer problems with a reasonably fine discretization of space. [→ Reihing 1959] describes an interesting time-sharing approach allowing to map the same set of analog computing elements to small sections of a large grid over and over again, thus trading time against discretization granularity.

21 See [→ Stepanow 1956, pp. 12 ff.]. It should be noted that this approach is just one possible ansatz assuming a particular form of the solution.

22 Similar examples can be found in [→ Sydow 1964, pp. 107 f.] and [→ Ameling 1963/1, pp. 280 ff.].

23 Cf. [→ Ameling 1963/1, p. 281].

24 T_i denotes the i^{th} eigenfunction, while $\mathscr{J}$ denotes a suitable index set.

25 This is an even more complex problem in the case of nonlinear differential equations.

26 Units are noted in square brackets.

27 In particular, classic analog electronic multipliers tended to exhibit large errors for small products.

28 This is done by selecting appropriate capacitors in the feedback path of the integrators.

29 Some systems even allow this button to be depressed during a computer run without the need of restarting the computation.

30 A system using digital optimization techniques to generate scaling factors for analog computers is described in [→Celmer et al. 1970]. [→Schwarze 1972] describes a hybrid computer approach to automatic scaling. A modern implementation of such a tool can be found in →https://github.com/bernd-ulmann/DEQscaler.

31 Cf. →chapter 10.

32 See [→Hall et al. 1969] and [→Giloi 1975, pp. 129 ff.].

1 Over time inevitable errors due to drift and noise would build up forcing the circuit to eventually oscillate even with all initial conditions set to 0.

2 Some machines, like the Telefunken RA 770, even have mechanically coupled coefficient potentiometers to facilitate setups like this where two potentiometers must be set in lock step.

3 See →section 8.6.

4 In applications like this where two or more multiplicands are to be multiplied with a (slowly changing) multiplier, a servo multiplier or a time division multiplier could be used. If one of these multipliers is inverting the sign of its output value, the sign inverting summer in →figure 8.4 is not required. In a classic analog computer the sign-inversion can be easily achieved with a quarter square multiplier. Since these multipliers require both input signals with both polarities, $\pm x$ and $\pm y$, the sign of the result can be inverted by interchanging the connections to one of these signal pairs.

5 It should be noted that this circuit can be used to generate the functions $-\sin(\varphi)$ and $\cos(\varphi)$ if the first derivative $\dot{\varphi}$ is available as input. In this case, $\dot{\varphi}$ is fed to the two multipliers directly instead of the output signal of the integrator on the lower left.

6 This setup can be further simplified by connecting the output of the leftmost integrator directly to the coefficient potentiometer *d/m* and connecting its output to a second input of the integrator yielding $-\dot{y}$, effectively saving one summer.

7 These still frames were taken from the screen of a Nicolet 4094C digital storage oscilloscope.

8 This high degree of interactivity is even today of incredible value in education as well as in research as it allows a user to rapidly get a feeling for complex dynamical systems.

9 02.03.1880–05.12.1949

10 03.05.1860–11.10.1940

11 See [→Elton et al. 1942] and [→Gilpin 1973] for more details. [→Archibald 2014] contains some newer insights into longer periodic population cycles.

12 The *capture cross section* so to speak.

13 The food supply for rabbits is assumed to be unlimited.

14 Natural death for rabbits as an explicit parameter has been omitted from the model.

15 A recent hydraulic analog computer for modeling predator prey dynamics is described in [→Doore et al. 2014].

16 An example for scaling such a system of coupled differential equations can be found in [→Schwarz 1971, pp. 369 ff.].

17 A comprehensive treatment of this model can be found in [→Bärwolff 2021].

18 Cf. [→Telefunken/1] and [→Pfaltzgraff 1969] for a much more complex but also more precise treatment of the same problem. Other vendors, such as Heathkit, also used bouncing ball simulations for marketing, although these these were usually even simpler circuits than the one described here. Such a simplified simulation can be found in [→Winkler 1961, pp. 199 ff.]. A rather complex simulation of the movement of a soccer ball on a hybrid computer can be found in [→Rattay 1976].

19 A real-time simulation like this would have required a very expensive and large stored-program digital computer in those days.

20 A better alternative would be to use a free (not grounded) coefficient potentiometer.

21 A small error is introduced since one of the Zener diodes always exhibits its forward voltage, but this is negligible in a demonstration setup like this.

22 It should be noted that this graph is only a rough approximation – the velocity in *x* direction actually decreases and does not stay constant between collisions.

23 A circuit like this or variations thereof are often employed when a high-frequency sine-/cosine pair is required which does not need to be spectrally clean. “High-frequency” is a bit of a hyperbole – a frequency of a couple kHz is sufficient for a flicker-free display. Due to the combination of positive and negative feedback, harmonics are introduced, but for driving a display their amplitude is small enough not to cause any visible distortions.

24 Vehicle Dynamics Section, Engineering Mechanics Department of the *General Motors* Research Staff at Warren, Michigan, cf. [→McLeod et al. 1958/6, p. 1994].

25 Cf. [→McLeod et al. 1958/6, p. 1994].

26 A thorough introduction to the mathematics of vibrating multi-mass systems can be found in [→Macduff et al.

1958, pp. 193 ff.] and [→Telefunken/2].

27 See [→Heller et al. 1976, p. 2].

28 Cf. [→Heller et al. 1976, p. 39].

29 This particular setup is described in [→Behrendt 1965]. A much more realistic and complex simulation of a car running over a bumpy road is described in [→Rattay 1978/1].

30 A more detailed description of such a two-mass system, including scaling, can be found in [→Giloi et al., pp. 48 ff.], [→EAI TR-10, pp. 44 ff.], and [→Carlson et al. 1967, p. 91].

31 Several kHz.

32 23.05.1917–16.04.2008

33 See [→Lorenz 1963].

34 See [→Jönck et al. 2003] for a comprehensive description of the system. Recent work on this is described by [→Tucker 2002]. Here proof is given that the attractor is robust, "*it persists under small perturbations of the coefficients in the underlying differential equations*". It is also proven that the Lorenz equations "*support a strange attractor*", which was conjectured by Lorenz as early as 1963.

35 [→Sprott 2016] describes a wealth of chaotic systems, most of which can be easily implemented on analog computers. [→Argyris et al. 1995] give a good overview of chaotic systems and their mathematical treatment.

[→ KUEHN 2015] also contains a wealth of information about such systems. The scaling process applied here follows [→ ULMANN 2016].

36 Every variable that is scaled down (or up) must be scaled up (or down) in every place it is used to compensate for the scaling factor introduced. Typically most of these up/down scalings cancel out eventually.

37 Constants are now coalesced and shortened to three or four decimal places, thus ≈ instead of = is used.

38 15.05.1835–19.10.1890

39 Solutions of this differential equation are called *MATHIEU functions*. [→ MCLACHLAN 1947] contains a comprehensive analytical treatment of this equation. [→ ARSCOTT 1964, pp. 26 ff.] is also read worthy in this respect.

40 See [→ RUBY 1996]. [→ RANDERY 1964] shows the application of MATHIEU's equation to the simulation of a parametron.

41 This section basically follows [→ EAI 1964/4].

42 One might think about shifting x down by subtracting 1, but this would complicate the remaining parts of the program.

43 A hat over a variable typically denotes a scaled machine variable.

44 The fact that $0 \leq \hat{x} \leq 1$ will be taken care of later.

45 [→MacKay et al. 1962, p. 137] describe the projection of a four-dimensional hypercube on an oscilloscope by means of an analog computer.

46 The gate labeled *AND* is unusual as it is not used as an and-gate but as a signal driver with two outputs, one inverted and one non-inverted.

47 Cf. [→Heinhold 1959, pp. 46 ff.] for more information on this technique.

48 Nikolay Yegorovich Joukowsky, 17.01.1847–17.03.1921

49 A detailed description of this type of airfoil can be found in [→Thwaites ed. 1987, pp. 112 ff.] and [→Eck 1954, pp. 237 f.]. Applying suitable changes to equation (→8.40), more complex and realistic airfoils can be described. [→Ashley et al. 1985, pp. 52 ff.] describe two such enhanced mappings based on the Mises (Richard von Mises, 19.04.1883–14.07.1953), the Kármán-Trefftz transformation (Theodore von Kármán, 11.05.1881–07.05.1963, Erich Trefftz, 21.02.1888–21.01.1937), and the Theodorsen mapping (Theodore Theodorsen, 08.01.1897–05.11.1978). See [→Warschawski 1945] for details. Numerical approaches for the generation of such airfoils and the simulation of particle flows are described in [→Zingg 1989]. A comprehensive treatment of the Joukowsky airfoil can be found in [→Betz 1948, pp. 191 ff.].

50 This is, as before, denoted by a black bar at the top of the integrator symbol. The simple digital control circuit is set up on the digital control subsystem of the RA 770 analog computer and is not shown here. This control

system also generates the signals used to switch the circuit for the actual conformal mapping between both of the inputs, in an alternating fashion.

51 A thorough and exhaustive treatment of such models can be found in [→Izhikevich 2007]. [→Ochs et al. 2021] describe a recent electronic implementation of this model.

52 In contrast to a harmonic oscillator which is typically based on an amplifier with suitable feedback, a relaxation oscillator switches abruptly between charge and discharge mode when running in resonance mode, thus yielding non-harmonic output signals.

53 See [→Hindmarsh et al. 1982] and [→Hindmarsh et al. 1984], [→Izhikevich 2007, pp. 123 ff.], and [→Ulmann 2021/1].

54 Using a numeric simulation looks a bit like cheating but simplifies things considerably in the following. Using an analog computer only, one can also scale problems by running the unscaled problem until it reaches overload for the first time and then manually rescale all affected values, rerun the program until the next overload is reached, etc. This is tedious but works well in practice. It should be noted that using a hybrid computer setup this scaling approach can be rather easily mechanized.

55 Scaled variables are not denoted by a hat or star in the following as this would clutter the formulas considerable.

56 The important thing here is that the two k_0 values differ by a factor of 100, so on THE ANALOG THING $k_{0x} = k_{0y} = 10^3$ (fast) and $k_{0z} = 10$ (slow) will work perfectly.

57 See [→ULMANN 2023/2].

1 A comprehensive account of the repetitive mode control logic used in the EAI 680 analog computer can be found in [→KLEINERT 1978]. [→BRUBAKER et al. 1961] describe the design and implementation of a complex control unit for a repetitive analog computer.

2 See →appendix A.

3 See [→Telefunken/8, pp. 19 ff.].

4 Up to 200 computing elements can be selected in a single RA 770 system. However, it is possible to couple three RA 770 together for a total of up to 600 computing elements.

5 In this application an integrator is used as a storage element switching it between IC and HALT mode.

6 See [→EAI 1964/1, pp. 13 ff.].

7 These are no operational amplifiers but simple buffer stages for digital signals.

8 A monostable multivibrator is a circuit exhibiting one stable state it will return to after a certain amount of time following some trigger signal.

9 See [→Telefunken/10] for more details and schematics of the DEX 100.

10 Ernest Rutherford, 30.08.1871–19.10.1937

11 A more detailed setup can be found in [→Telefunken/5, p. 3].

12 Similar but much simpler examples are described in →sections 8.10 and →8.11. A number of complex examples using a digital control unit are described in [→Hitachi 1970]. [→Salvioni 1971] also describes a complex example using a digital control unit in conjunction with an analog computer to generate the shape of fluttering panels on an oscilloscope screen. [→Lehmann 1967] describes a system for the simulation of complex metropolitan train networks where a digital control unit is used to implement a traffic control system while the analog computer simulates the dynamic behavior of trains.

13 This special purpose oscilloscope has no built-in time base but features two separate *x/y*-inputs thus allowing the display of two *x/y*-figures at once using an internal chopper circuit.

14 It should be noted that all but the most early Telefunken tabletop analog computers had provisions for a DEX 100 or DEX 102. If no such extension is used, a special plug must be used on the analog computer's rear to connect all integrators to the built-in mode control system.

15 See 11.

16 See [→Husinsky et al. 1985, pp. 8 f.] for more details on the PLU.

1 Credits: University of Southern California, Libraries, and California Historical Society.

2 A subsidiary of *Ramo-Wooldridge Corporation*.

3 See [→ Bauer et al. 1956].

4 A fully expanded system had up to 15 DAC channels.

5 Cf. [→ Bekey et al. 1968, p. 154].

6 See [→ Datamation 1957, pp. 24 ff.].

7 See [→ McLeod et al. 1957, p. 1130].

8 Adjusted for inflation, 4.1 million USD in 1957 is equivalent to about 42 million USD in 2022.

9 [→ Bekey et al. 1968, pp. 154 f.] note that "*[i]n the time that elapsed between the original specification of the hybrid system and the delivery and acceptance of the conversion equipment, it was established that the guidance and control problems associated with missile flight are not closely coupled and can be studied separately. Consequently the basic problem for which the hybrid computing system was designed, vanished*."

10 A collection of typical problems that could be solved by hybrid computers is given in [→ Benham 1970].

11 These large machine units were commonplace in aerospace technology and it was imperative for any analog computer being coupled with simulator and flight hardware to support this voltage range.

12 See [→Bekey et al. 1968, p. 155].

13 See [→SDS 1969].

14 See [→Comcor] for details on the machine.

15 This is even today not an easy task, mostly due to high interrupt latencies on many modern digital computers.

16 See [→Frisch et al. 1969, p. 36].

17 Information about this can be found in [→Giloi 1963, pp. 268 f.] and [→Skramstad 1959].

18 See [→Cowan 2005] and [→Guo 2017].

19 See also →section 16.

20 Nevertheless, setting coefficients and reading data from the analog computer by means of the digital computer can take a considerable amount of time, thus slowing down the overall time to solution.

21 A description of an extended ALGOL system can be found in [→Herschel 1966] while a more general description of this approach is given in [→Feilmeier 1974, pp. 133 ff.].

22 See →section 5.5.

23 *Universal Serial Bus*

24 The manual input was used during this study to actively unbalance the pendulum and observe the reaction of the trained control system to this perturbation.

25 See [→ Sutton et al. 2018] or [→ François-Lavet et al. 2018] for more details on Q-learning.

26 This experiment is also described in →https://www.youtube.com/watch?v=jDGLh8YWvNE, retrieved 06.06.2022.

27 See [→ Evans 2013] for details on this type of equations.

28 This section is based on [→ Huang 2018], [→ Huang et al. 2018], and [→ Ulmann 2024].

29 Fischer Black, 11.01.1938–30.08.1995

30 Myron Scholes, 01.007.1941–

31 See [→ Gronbach et al. 2012].

32 26.11.1894–18.03.1964

33 See [→ Henze 2018] for details.

34 Without the stochastic part $\sigma y W$ the solution would be a simple exponential function.

35 Particularly this means no overloads and good coverage of the output interval $[y(0), 1]$.

36 In order for this to work $y(0) \neq 0$ must be satisfied.

37 The software used on the microcontroller can be found at →https://github.com/anabrid/THAThc, retrieved 19.03.2025.

38 The hybrid controller automatically switches the analog computer into HALT mode at the and of an OP period.

39 See →http://www.gnuplot.info, retrieved 02.08.2025.

1 See [→Bush et al. 1945, p. 277].

2 See →section 2.8, page 44.

3 See [→Bush et al. 1945, p. 277].

4 Ibid. Another interesting characteristic of this machine is the fact that output was not restricted to plotters but could also be obtained in the form of tabulated decimal data. Probably a first for its time.

5 It should be noted that although the computing elements were mechanical in nature, they were connected electronically by means of capacitively working “angle indicators” controlling servomotors (see [→Bush et al. 1945, pp. 278]).

6 It should be noted that tape preparation was done manually using a special tape punch unit.

7 In general, a compiler is a computer program translating code written in one programming language into another programming language. Typically, compilers translate a high-level language into machine language which can be executed by a stored-program digital computer. In the context of analog computers, the input will be some high-level problem description generating the required configuration bitstream for the analog computer.

8 See [→Clos 1953] and [→Benes 1965] for details.

9 This size roughly corresponds to an EAI 680 analog computer system.

10 An interesting programming environment capable of deriving an analog computer setup including a powerful allocation strategy for selection and placement of computing elements is described in [→ Franklin et al. 1969].

11 A detailed description of this system can be found in [→ Solar 1977/2].

12 See [→ EAI 1986, pp. 9 ff.]. The system is described in great detail in [→ Landauer 1983] and [→ Emblay[a]].

13 The two additional inputs of each of the output blocks were connected to +1 and GND for easy introduction of constants into a program and to facilitate static and dynamic tests of the underlying analog computer system.

14 See [→ NEC 1987].

15 [→ Landauer 1986/2] describes a nonlinear oscillator study and [→ Landauer 1986/3] the simulation of a satellite attitude control system on a SIMSTAR system.

16 It is unclear if the two last orders were actually fulfilled.

17 See [→ TU Wien 1985, pp. 3 f.].

18 Details can be found in [→ Emblay[b]] and [→ Husinsky et al. 1985].

19 This was a member of the *CONCEPT 32* minicomputer family built by *Gould*.

20 See [→ Landauer 1986/1].

21 Values are represented as currents in this design, thus no summers are required since summation can be done by connecting multiple current sources to a single computing element input.

22 An even more recent development using a 65 nm CMOS process is described in [→ Guo 2017]. Developments like these gave rise to *field programmable analog arrays* (*FPAA*s) such as those described in [→ Hasler 2019] and [→ Hasler et al. 2024] with corresponding programming environments, see [→ Hasler et al. 2020] and [→ Collins et al. 2016]. Some of these devices use *switched capacitors* to connect the computing elements. This replaces a crossbar structure with a number of capacitors which are used as basic sample and hold elements storing the output value of one computing element and then applying that voltage to another element in a time-shared fashion. This greatly reduces the complexity of the device itself but at the cost of massively decreased bandwidth since the capacitors being switched in and out of the circuit basically implement low-pass filters.

23 The REDAC system is based on the same architectural idea but contains up to ≈ 1000 integrators, ≈ 500 multipliers, ≈ 4000 coefficients, dozens of function generators, etc.

24 See →section 8.8.

25 See [→Köppel 2024, pp. 25 f.].

1 [→Michels 1954, p. 2] defines a DDA as "*an electronic computer which solves differential equations by numerical integration*.". This definition is debatable as it would also hold true for a classic stored-program digital computer and does not take the central feature of setting up an analog to solve a problem into account.

2 See →section 12.4.1. [→Erismann 1962] gives a comprehensive overview of DDAs.

3 It can be shown that analog computation is more energy efficient due to physical reasons for a *SNR* (*signal to noise ratio*) of up to 60 dB (see [→Sarpeshkar 1998, p. 1615]).

4 Also known as *simultaneous* DDA, see [→Owen et al. 1960, p. 6].

5 For further reading see [→Shileiko 1964], [→Bywater 1973], [→Forbes 1957, pp. 215 ff.], [→Forbes 1955], [→Winkler 1961], [→Beck et al. 1958], [→Klein et al. 1957, p. 1105], [→Goldman 1965], and [→Jackson 1960, pp. 578 ff.].

6 See [→McLeod et al. 1958/4, p. 1223].

7 Since two signal lines are necessary to transmit such values, some DDA implementations restrict the domain of these incremental values to {−1; 1}, thus simplifying the hardware implementation but requiring constant values to be represented by an alternating sequence of −1 and 1 increments. See [→Michels 1954, p. 19].

8 A comprehensive study of this type of integrator can be found in [→Zoberbier 1968].

9 This second integrator is not normally required when all values within a DDA setup are represented by incremental values.

10 See [→Bendix 1954, pp. 16 ff.].

11 Whenever possible $F(U, V)$ should be generated by solving a suitable differential equation.

12 See [→Bendix 1954, pp. 21 ff.].

13 See [→Campeau 1969, p. 711].

14 A detailed analysis of DDA specific questions regarding the precision of computations can be found in [→Kella 1967] and [→Kella et al. 1968]. Optimal ratios of the length of the integrator accumulator registers and incremental values are investigated in [→McGhee et al. 1970].

15 See [→Werrell 1985, pp. 82 ff.]. The system was named after Lewis Carrols's (27.01.1932–14.01.1898) *The Hunting of the Snark*. The follow-up cruise missile was aptly named *Boojum* following the last line of this poem.

16 The V2 was admittedly no cruise missile. Its engineering version was known as *A4*.

17 See [→Ceruzzi 1989, pp. 22 ff.].

18 28.06.1918–23.09.1995; a long interview with Floyd Steele can be found at [→Steele 1973].

19 29.04.1924–23.07.2001

20 27.08.1921–27.01.1996

21 See [→Ceruzzi 1989, p. 25].

22 Short for *MAgnetic Drum DIgital Differential Analyzer*. Due to reliability problems of this prototype, it was often pronounced as *mad Ida*.

23 See [→Ceruzzi 1989, p. 25].

24 See [→Donan 1950, p. 1].

25 See also [→Popular Mechanics 1950].

26 28.12.1930–08.02.1957

27 Cf. [→Eckdahl et al. 2003].

28 Snark finally employed an analog electronic analog computer for its guidance, see [→Ceruzzi 1989, p. 25]. Nevertheless, a DDA was developed for the *Polaris* guidance system, see →section 15.15.8.

29 A thorough description of these production systems can be found in [→Northrop 1950]. [→Palevsky 1973] contains some personal memories regarding these early DDA systems.

30 See [→Evans et al. 1966] for an in-depth description of the internal structure of the D-12.

31 Excess-three coded decimals.

32 This is a good example of the ability of a sequential DDA to trade speed against problem complexity.

33 See [→Bendix 1954, pp. 11 ff.].

34 This equation describes an oscillator with amplitude stabilization due to the term $x^2 - 1$. For $x < 1$ this becomes negative, thus increasing the oscillator's amplitude and decreasing it for $x > 1$. See [→Pol et al. 1928] for more information on this topic.

35 All values are represented as normalized fixed-point values with one decimal digit before and 5 after the implicit decimal point.

36 See bottom left of →figure 12.16.

37 Each address occupies two columns and successive input addresses are separated by a space character.

38 See [→Owen et al. 1960] for details on this system.

39 See [→Ulmann 2014, pp. 40 ff.] for more details on core memory and its development.

40 Two and a half times faster than the much larger Bendix D-12.

41 [→Beecham et al. 1965] describe a practical application of the CORSAIR system for the kinematic simulation and wind tunnel control in a wind tunnel based flight dynamics simulator.

42 See [→Ameling 1963/2], [→Rechberger 1959], and [→Packard Bell].

43 The internal registers of the basic computing elements are implemented as magnetostrictive delay lines, so word length and clock frequency are fixed.

44 This results in an iteration cycle time of 10μs. A much faster DDA, the *SPEDAC* (*Solid-state Parallel, Expandable, Differential Analyzer Computer*), achieving a 1 MHz iteration rate, is described in [→ Bradley et al. 1962].

45 See [→ Giloi 1975, p. 23].

46 If more inputs are required for an integrator, a summer in front of the integrator is necessary.

47 See [→ Jackson 1960, pp. 585 f.].

48 *Computerized Numerical Control*

1 Cf. [→ Neumann 1956, pp. 63 f.].

2 This idea eventually led to modern majority voting systems used in fault tolerant digital systems.

3 Cf. [→ Neumann 1956, p. 91].

4 See also [→ Neumann 1956, pp. 95].

5 1938–

6 See [→ Gaines 1967]. A comprehensive survey of stochastic computing can be found in [→ Alaghi et al. 2013], while [→ Gaudet et al. 2019] and [→ Gaines 2019] give an overview of its historic roots.

7 See [→ Winstead 2019, pp. 41 ff.] for more details on stochastic numbers in general. This work also contains a

very readable introduction to stochastic computing.

8 From an implementation perspective, stochastically independent pseudorandom bit sequences are very desirable as they can be generated without having to worry about the stability of its characteristics. [→ Holt et al. 1995] describe an interesting approach to such pseudorandom generators suitable for stochastic computers.

9 There are more ways to interpret a stochastic number such as *likelihood ratio stochastic numbers* and *log likelihood ratio stochastic numbers*, see [→ Winstead 2019, pp. 42 f].

10 See [→ Poppelbaum 1979, p. 1].

11 A *logic gate* implements a certain basic boolean operation such as *AND* (the output of such a gate is *true* if and only if all of its inputs are true), *OR* (its output is true if at least one of its inputs is true, so this is aptly called an *inclusive OR*), negation (the output is true if the input is *false* and vice versa), etc.

12 Division is considerably more involved, see [→ Poppelbaum 1979, p. 24], but still much simpler than in a classic digital computer.

13 See [→ Winstead 2019, pp. 55 f.].

14 See [→ Gaines 1967, pp. 152 f.] for more details.

15 See [→ Alaghi et al. 2013, p. 92:9].

16 See [→ Poppelbaum 1968, p. 22]. [→ Poppelbaum 1979] also describes a technique called *burst processing*, which lies between a classic binary value representation and stochastic computing requiring only about 10% of the circuitry of a weighted binary value representation. More detailed information on the theory and basics of stochastic computing can be found in [→ Gaines 1967], [→ Esch 1969], and [→ Massen 1977]. [→ Riedel et al. 2019, pp. 104 ff.] analyze what can be computed by stochastic computers in general. Example circuits for stochastic computers can be found in [→ Langheld 1979/1], [→ Langheld 1979/2], and [→ Poppelbaum et al. 1967].

17 Short for *Regular Array of Stochastic Computing Element Logic*.

18 See [→ Esch 1969].

19 See [→ Manohar 2015].

20 See [→ Riedel 2019, p. 122].

21 [→ Hsiao et al. 2019] describe various techniques for generating stochastic bitstreams, [→ Jia et al. 2019, pp. 166 ff.] describe an interesting approach to generating such bitstreams based on magnetic tunnel junctions. [→ Riedel 2019] details on deterministic approaches to stochastic computing.

22 See [→ Winstead 2019, pp. 69 ff.] and [→ Onizawa et al. 2019].

23 See [→ Bohn 1960].

1 John von Neumann used this wording when writing about generating (pseudo-)random numbers using digital computers.

2 See [→Frisch et al. 1969, pp. 19 f.].

3 A similar system is described by [→Stein et al. 1959].

4 See [→Bashe et al. 1986, pp. 135 ff.].

5 See [→Brennan et al. 1967, p. 243].

6 See [→Harnett et al. 1963].

7 See [→Brennan et al. 1964].

8 See [→Bashe et al. 1986, pp. 508 ff.].

9 [→Korn 2005] describes such early developments from the late 1960s onward. Another good overview is given in [→Naylor et al. 1966, pp. 239 ff.].

10 Cf. [→Giloi 1975, pp. 35 f.]. Typical algorithms for this are described in [→Stein et al. 1970], a description of the implementation used for CSMP can be found in [→Speckhart et al. 1976, pp. 12 ff.].

11 See [→Press et al. 2001, pp. 123 ff.] for details on such methods in general.

12 See [→Farris 1964].

13 See [→Bashe et al. 1986, p. 578] and [→Bender et al. 1960] for details on this early transistorized digital computer system.

14 See [→ Farris 1964, p. 1].

15 See [→ Farris 1964, pp. 71 f.].

16 See [→ Gaskill et al. 1963].

17 See [→ Hurley et al. 1963].

18 See [→ Speckhart et al. 1976] for a thorough description and introduction to CSMP.

19 Cf. [→ Speckhart et al. 1976, p. 15].

20 `PRDEL` controls the time interval between two print operations. If omitted, its default is $\frac{\text{FINTIM}}{100}$, see [→ Speckhart et al. 1976, pp. 18/24].

21 The PDP-11 family is one of the most successful 16 bit minicomputer architectures of all times. It consists of dozens of models ranging from small scale single board systems to large departmental servers. Machines of this type were built by *Digital Equipment Corporation* (*DEC*) from 1970 well into the 1990s.

22 A much more complicated example, the escape of a pilot using an ejection seat, can be found in [→ Korn et al. 1983, pp. 177 ff.].

23 See [→ Korn et al. 1983, pp. 201 f.].

24 See [→ Beukeboom et al. 1985].

25 See →https://pypi.org/project/pybrid-computing/, retrieved 03.07.2022.

1 Cf. [→Kettel 1960, p. 165].

2 There are ways to improve accuracy of solutions obtained by analog computers by using a hybrid computer approach.

3 *LODE* for short.

4 See [→Korn et al. 1956, pp. 142 ff.] for example.

5 See [→Giloi et al. 1963, pp. 249 ff.] and [→Heinhold et al. 1969, pp. 108 ff.].

6 See [→Heinhold et al. 1969, pp. 111 ff.] and [→Feilmeier 1974, pp. 202 ff.]. More detailed information about iterative approaches can be found in [→Heinhold et al. 1969, pp. 117 ff.], [→Feilmeier 1974, pp. 197 ff.], and [→Giloi et al. 1963, pp. 253 ff.]. Special approximation techniques for solving boundary value problems for partial differential equations are described in [→Vichnevetsky 1969].

7 Richard Feynman, 11.05.1918–15.02.1988

8 Mark Kac, 03.08.1914–26.10.1984

9 Robert Brown, 21.12.1773–10.06.1858

10 See [→Chuang et al. 1960] for example.

11 A practical example can also be found in [→Ulmann 2023/1, pp. 238 ff.].

12 Another interesting approach to the solution of PDEs on analog computers is described in [→Liban 1962/2]. It is

based on a finite variant of the Fourier transform suitable for analog computer implementation.

13 See [→GEO SPACE] for details.

14 A classic comprehensive introduction to integral equations in general can be found in [→Muskhelishvili 1953].

15 Ivar Fredholm, 07.04.1866–17.08.1927

16 This requires at least three independently controlled integrator groups, which in turn requires a complex control circuit.

17 26.11.1894–18.03.1964

18 See [→Bennett 1993, pp. 104 ff.].

19 See [→Bush et al. 1927].

20 See [→N. N.] for a short contemporary account of this machine.

21 See [→Bush et al. 1927, p. 82 ff.].

22 See [→Bush et al. 1927, p. 75].

23 The cinema integraph is described in much more detail in [→Hedeman 1941] and [→Hazen et al. 1940]. Further information on integraphs can be found in [→Stine 2014] and [→Macnee 1953].

24 See [→Heinhold et al. 1969, pp. 173 ff.] and [→Atkinson 1955/1].

25 If servo or time division multipliers are available, the powers of *x* can be generated in a straightforward way.

26 See [→Heinhold et al. 1969, pp. 174].

27 See [→Herschel 1962].

28 See [→Morgan 1960].

29 See [→Morgan 1961/1], [→Morgan 1961/2], and [→Morgan et al. 1961] for details on the system.

30 25.07.1904–01.07.1996

31 See [→Püttmann 2014]. Here, a small-scale model of such a machine is described called "*Seilcomputer Kelvin*".

32 See [→Wilbur 1936], [→Petzold 1997], and especially [→Speek 2017] for more details on that particular machine. Another system, closely based on Wilbur's machine was built at Tokyo Imperial University's Aeronautical Research Institute by Sasaki Tatudiro, Shiga Makoto et al. and might have been operational around 1944 (see →http://museum.ipsj.or.jp/en/computer/dawn/0057.html, retrieved 20.05.2025). Another specialized system for this task was developed in the 1950s and is described in [→Atkinson 1955/2].

33 An algebraic loop is a positive feedback path involving only summers, i. e., a loop involving an odd number of sign-inverting summers. Such a circuit is inherently unstable and must be avoided at any cost.

34 Carl Gustav Jacob Jacobi, 10.12.1803–18.02.1851

35 Carl Friedrich Gauss, 30.04.1777–23.02.1855

36 Philipp Ludwig von Seidel, 24.10.1821–13.08.1896

37 More detailed information about these methods can be found in [→ Heinhold et al. 1969, pp. 135–152], [→ Vocolides 1960], [→ Giloi et al. 1963, pp. 152 ff.], [→ Marquitz et al. 1968], [→ Kovach et al. 1962], [→ Jackson 1960, pp. 332 ff.], and [→ MacKay et al. 1962, pp. 192 ff.]. The effects of parameter variations are analyzed in [→ Kahne 1968].

38 The classic spectral radius convergence criterion must be satisfied for this to work.

39 Using the same idea, but starting with an identity matrix, matrix inversions can also be performed by an analog computer.

40 Cf. [→ Press et al. 2001, pp. 368 ff.].

41 See [→ Dick et al. 1967], [→ Giloi et al. 1963, pp. 306 ff.], [→ Kovach 1952], and [→ Ratz 1967].

42 See →section 2.6.

43 Cf. [→ Giloi et al. 1963, pp. 307 f.].

44 See [→ Giloi et al. 1963, pp. 308 ff.].

45 See [→ Giloi et al. 1963, pp. 309 ff.].

46 Cf. primarily [→ Laning et al. 1956], and [→ Korn 1966] but also [→ Giloi et al. 1963, pp. 357 ff.] and [→ Jackson 1960, p. 367].

47 Detailed information on these topics can be found in [→ Giloi et al. 1963, pp. 358 ff.], [→ Meyer-Brötz 1962], [→ Ott 1964] (computation of correlation coefficients), [→ Sydow 1964, pp. 250 ff.] (quality control), [→ Bohling et al. 1970], [→ Rideout 1962, pp. 140 ff.], [→ Korn et al. 1956], and [→ Feilmeier 1974, pp. 275 ff.]. [→ Dern et al. 1957] describe an early distribution analyzer.

48 See [→ Rideout 1962, pp. 238 ff.] and [→ Feilmeier 1974].

49 11.11.1917–13.07.2016

50 See [→ Pierre 1986, pp. 193 ff.] for general information on optimization problems. Basic techniques for tackling this class of problems with analog computers are described in [→ Korn et al. 1956, pp. 147 ff.], [→ Kovach et al. 1962], and [→ Deziel 1966]. Recent work on analog computers for optimization are described in [→ Vichik 2015].

51 See →section 4.5.8.

52 Cf. [→ Pierre 1986, pp. 251 ff.].

53 See [→ Pierre 1986, pp. 296 ff.], [→ Levine 1964, pp. 217 ff.], and [→ Albrecht et al.].

54 Cf. [→ Feilmeier 1974, pp. 243 ff.].

55 The author is greatly indebted to Dr. Chris Giles for many fruitful discussions and especially his support regarding this section.

56 A historical overview of mine ventilation system simulations can be found in [→ Laage et al. 1995, pp. 9

ff.], while a comprehensive introduction to the general problem of mine ventilation can be found in [→ Kingery 1960].

57 Georg Simon Ohm, 16.03.1789–06.07.1854

58 Claude Louis Marie Henri Navier, 10.02.1785–21.08.1836

59 George Gabriel Stokes, 13.08.1819–01.02.1903

60 Allen Hazen, 28.08.1869–26.07.1930

61 Gardner Stewart Williams, 1866–1931

62 Henry Philibert Darcy, 10.06.1803–03.01.1858

63 Julius Ludwig Weisbach, 10.08.1806–24.02.1871

64 An early study of fluid friction in pipes with rough walls can be found in [→ Colebrook et al. 1937].

65 Robert Manning, 22.10.1816–09.12.1897

66 28.08.1902–04.03.1956

67 See [→ Laage et al. 1995, pp. 9 f.].

68 Typically, these resistive elements should not be operated at more than 4.6 V for extended periods, otherwise their lifetime would be substantially degraded (see [→ McIlroy 1950, p. 354]).

69 See [→ McIlroy 1950] for details. [→ Appleyard et al. 1957] and [→ Moore 1962] describe practical applications of the McIlroy pipeline network analyzer. A smaller system using curve follower function generators inspired by

this analyzer is described in [→Falk 1958]. [→Camp et al. 1934] describe the analysis of water distribution systems by means of an early network analyzer.

70 See [→Montan Forschung 1960] for a detailed description of the system and its components.

71 See [→Montan-Forschung] for details. [→Lienhard 1969] and [→Nix 1965] describe the application of such systems to the design of pipeline networks.

72 See [→Telefunken/3], [→Lukes 1967, pp. 129 ff.], and [→MacKay et al. 1962] for general information on this topic.

73 [→MacKay et al. 1962, pp. 134 ff.].

74 A simple example describing the simulation of a planet's orbit can be found in [→Telefunken 1966].

75 One example of this is described in →section 9 page 245.

76 Short for *Deutsches Elektronensynchrotron*.

77 [→Borchardt et al.], [→Borchardt 1965], [→Borchardt 1966], and [→Borchardt et al. 1965] contain more information about these simulations and the special techniques developed.

78 See [→Borchardt[a]].

79 The systems used here were an EAI TR-48 and an *Applied Dynamics AD-2-64-PBC*.

80 See [→Shen 1970, pp. 4 f.].

81 See [→Aldrich et al. 1955].

82 A much smaller and simpler, but nevertheless similar, passive circuit is described in [→Ulmann 2020/1]. A recent development, a nanophotonic analog mesh computer, is described in [→Kayraklioglu 2020].

83 A detailed description of these direct analogies and their application to frost and thaw problems can be found in [→Paynter[a]].

84 Charles-François Sturm, 29.09.1803–18.12.1855

85 Joseph Liouville, 24.03.1809–08.09.1882

86 This work was done at the department of Electrical Engineering and the Fishing Industry Research Institute of the University of Cape Town.

87 See [→Patowary 2019].

88 It should be noted that hydraulic analog computers were also developed elsewhere, see [→Roffman et al. 1965] and [→Lechner 1966].

89 Lev Pontryagin, 03.09.1908–03.05.1988

90 Interestingly no direct analogies have been employed.

91 These function generators are quite interesting as they generate functions of two variables.

92 Obviously, such simplifications were common in the past and dated back to earlier manual fallout prediction

techniques.

93 More details on this system can be found in [→Wright et al. 1957], [→Wright 1957], and the patent [→Skramstad et al. 1961] which contains a detailed description of the circuitry of the overall system.

94 The use of such specialized slide rules was mainly restricted to civil defense groups.

95 See [→Henderson et al. 1965] and [→Wilson 1966] for more details.

96 See [→Henderson et al. 1965, p. 1-2].

97 See [→Wilson 1966, p. 5].

98 A comprehensive list of functions implemented can be found in [→Wilson 1966, p. 36 ff.].

99 A modern, online version of such a system is *NUKEMAP*, →https://nuclearsecrecy.com/nukemap/, retrieved 18.05.2025.

100 *MOSFET*

101 Numerical methods for the simulation of semiconductor junctions and field effect transistors can be found in [→Akers 1977].

102 Also known as *Esaki (Leo Esaki, 12.03.1925–) diode*.

103 Normally algebraic loops are to be avoided at any cost due to their inherent instability.

104 In *Q* switching the optical resonator of a laser contains a variable attenuator. This directly controls the *Q* factor of the resonator which can be used to generate short but extremely powerful laser pulses.

105 See [→Sauro 2019] for an in-depth treatment of enzyme kinetics.

106 See [→Dornier/2].

107 Further introductory examples can be found in [→Ulmann 2023/1, pp. 110 ff.].

108 [→Anders[a]] gives a good description of the historical development of quantum chemistry.

109 Erwin Schrödinger, 12.08.1887–04.01.1961

110 See [→Preuss 1962] and [→Preuss 1965] for the basics of quantum chemistry.

111 01.12.1901–25.10.1968

112 See [→Kron 1945/2] and [→Kron 1945/1].

113 05.12.1919–25.11.2012

114 [→Ulmann 2019] shows a more recent implementation.

115 A wealth of information on these subjects as well as example programs can be found in [→Mahrenholtz 1968].

116 See [→Bilstein 2003, pp. 360 ff.] and [→Woods 2008, pp. 83 ff.].

117 Cf. [→Dornier/4].

118 The analog computer used in this example, a Hitachi 200X, features summers and integrators each providing normal and inverted outputs, thus simplifying computer setup considerably, which is reflected in the program shown.

119 A similar simulation of a three-story building is described in [→James et al. 1971, pp. 168 ff.].

120 Short for *Strong Earthquake Response Analog Computer*.

121 See [→Hitachi 1969, pp. 6 ff.].

122 See [→Rieger et al. 1974].

123 See [→Fremerey[b]] and [→Fremerey 1978].

124 Cf. [→Hütte 1926, pp. 82 ff.] for basic information on crank mechanisms and [→Mahrenholtz 1968, pp. 129 ff.] or [→Mahrenholtz 1968, pp. 79 ff.] for examples of analog computer simulations of such systems.

125 A simple example can be found in [→Dornier/5].

126 See [→Landauer 1975] for more details.

127 See [→Instruments and Automation 1958].

128 Tune-up time described the time span elapsing after parameter changes in a complex industrial installation until normal operation is reached again.

129 This last example is a rare case in which an explicit differentiation of a variable is necessary.

130 See [→Valentin 2003, p. 135].

131 See [→Meissl 1960/2, p. 76].

132 These simulations were performed on a Telefunken RA 463/2 analog computer, see →section 6.1. It should be noted that this particular analog computer only allowed for a maximum of 110 seconds of computing time as its operational amplifiers were not chopper-stabilized and integration errors due to drift effects became excessive after about 100 seconds.

133 Given the high density of molten steel the resulting forces are quite considerable.

134 Delta tide analog computer.

135 The Deltawerken are a complex of large-scale construction projects in the Netherlands aimed at the protection against storm surges, floods, etc.

136 21.12.1893–09.12.1959

137 [→Santen 1966] describes resistor networks for modeling water systems in general.

138 See →section 4.6.8.

139 A Bode diagram consists of two function graphs – one showing the amplitude response and the other showing the phase shift of a system with the *x*-axis representing frequency. It was developed by Hendrik Wade Bode (24.12.1905–21.06.1982).

140 A Nyquist diagram shows the *locus curve* of a control system. It was conceived by Harry Nyquist (07.02.1889–04.04.1976).

141 17.04.1900–1956

142 29.04.1900–24.02.1964

143 See [→ Kubelka et al. 1931].

144 See [→ Atherton 1955] and [→ Hemmendinger 2014].

145 See [→ Hemmendinger 2014, p. 6].

146 11.05.1918–02.04.2010

147 01.04.1915–16.08.2003

148 A hybrid computer for the same task, the Redifon *Redi-colour* was developed in 1965, see [→ Cutler 1965].

149 The necessary basics of reactor physics are covered in [→ Markson 1958].

150 See [→ Morrison 1962]. The simulations described there make extensive use of complex passive feedback networks to model the neutron flux densities, etc.

151 Cf. [→ Sydow 1964, pp. 225 f.].

152 [→ Sydow 1964, pp. 230 ff.].

153 See [→ Frisch 1971, pp. 72 ff.], [→ Sydow 1964, pp. 233 f.], and [→ Bryant et al. 1962, pp. 49–51].

154 See [→Sydow 1964, pp. 234 ff.], [→Frisch 1968], [→Brey 1958], and [→Scott 1958].

155 See [→Jones 1961].

156 The basics of neutron kinetics are covered in [→Weinberg et al. 1958].

157 See [→Fenech et al. 1973, p. 128].

158 Such external neutron sources are typically used during the startup process of a nuclear reactor.

159 A device exhibiting negative resistance shows an increasing voltage across its terminals when the current through the device decreases. Gas discharges or Tunnel diodes exhibit such a behavior, though negative resistances are typically implemented using operational amplifier circuits.

160 See [→Computers and Automation 1960, p. 16].

161 Not the least problem of the Chernobyl RBMK reactor was its sheer size and insufficient instrumentation which made it a rather dangerous and hard to control system (see [→IAEA 1992, p. 6]).

162 A modern and detailed approach to the use of control room simulators for training nuclear power plant operators can be found in [→IAEA 2004].

163 See →section 6.3.

164 See →section 6.1.

165 20.09.1901–28.11.1954

166 This reactor, also known as *Fermi 1* experienced a partial fuel meltdown in October 1966 (see [→U.S.NRC])

167 See →section 4.11.

168 Modules like these were, of course, also developed based on analog computer studies, see [→Cameron et al. 1961].

169 Nowadays, ultra centrifuges are the technology of choice in Uranium enrichment.

170 24.07.1913–16.11.2010

171 See [→Knorre 1971, p. 107].

172 A wealth of information on these topics can be found in [→Knorre 1971] and [→Röpke et al. 1969]. This area of research was important enough to justify the development of special-purpose analog computers such as the *HSE Analog Computer*, see [→Hugo Sachs Elektronik K. G.].

173 An IBM 360/67.

174 See [→Rigas et al., p. 95].

175 Short for *Cathode Ray Tube* – in this case an IBM 2250 display was used.

176 An EAI 690 hybrid computer, see [→Rigas et al., p. 95].

177 Cf. [→Habermehl et al. 1969].

178 These techniques have also been used to process EEG data, see [→Feilmeier 1974, p. 30].

179 This term describes the effect that the interval Δt between two successive heartbeats is decreased during inhaling and is increased during exhaling.

180 The integrator's mode of operation is controlled by a signal derived from the ECG signal. Normally the R wave of the signal is used for this purpose. Every occurrence of this R spike resets the integrator, which yields a ramp function at its output linearly increasing with the time elapsing between two successive heartbeats.

181 See [→Beckman 1963]. More information on Beckman analog computers can be found in [→Beckman Instruments Inc./2]. Beckman also published a great introduction to analog computing, see [→Beckman Instruments Inc./1].

182 A somewhat strange application of an analog computer in eye research is described in [→Computers and Automation 1964/1].

183 See [→Fincham 1963, p. 60].

184 05.02.1914–20.12.1998

185 22.11.1917–30.05.2012

186 These giant axons were often used in early neurophysiology research due to their size reaching up to 10 cm in length and up to 1 mm in diameter.

187 Cf. [→Brouwer 2007, p. 3].

188 30.03.1922–21.11.2007

189 1929–10.03.1999

190 More information regarding these analog computer studies can be found in [→Bekey 1960].

191 See [→Rosenblatt 1958] and [→Rodríguez 1991].

192 The similarity of the idea to modern day *memristors* is quite remarkable.

193 The analog computer used in this study was a →Hitachi 200X featuring non-inverting integrators and summers, which has to be taken into account for the program shown.

194 A more realistic approach to simulating the spreading of contagious diseases is the *SEIR*-model (see →section 8.5), taking susceptible, exposed, infected, and recovered persons into account. This can be easily implemented on an analog computer. The basics of this model are described in [→Bärwolff 2021].

195 See [→Payne 1988, p. 272]. The parameter of interest in such studies is the first derivative of the acceleration experienced by a body, the *jerk* or *jolt*.

196 Based on these a mechanical quadruped was developed.

197 Greg Parkhouse, personal communication to the author.

198 A short overview of analog computer developments in the soviet union can be found in [→Radio und

Fernsehen 1961]. These often differed quite substantially from their western counterparts. An overview of analog computer developments in the *German Democratic Republic* (*GDR*) is given in [→ Krug 2006].

199 See [→ Ushakov 1958/1].

200 The rather unique developments of analog computing in the Soviet Union are described in detail in [→ Abramovitch 2005].

201 This machine also found widespread application in many other fields including nuclear technology, etc.

202 This correction takes the properties of a low-velocity zone, a *weathering layer*, into account.

203 This study also contains a comparison of spectral data generated by an analog computer approach with data generated on a digital computer.

204 Cf. →section 6.4.

205 18.11.1914–04.03.1975

206 Short for *Monetary National Income Automatic Computer*.

207 See [→ Phillips 1950], [→ Fortune 1952], [→ Swade 1995], and [→ Care 2006/1].

208 MONIAC also inspired some caricatures like that shown in [→ Swade 1995, p. 16] as well as the analog computer sitting in the basement of the bank in Terry Pratchett's (28.04.1948–12.03.2015) novel *Making Money*.

209 [→Engel et al. 1964] and [→Zhang 2015] give a comprehensive introduction to the description of economic systems by means of differential equations.

210 See [→Swade 1995, p. 17].

211 See [→Swade 1995, p. 17].

212 See [→isec/1] and [→isec/2].

213 See [→isec/1, p. 33].

214 See [→Noronha[a]].

215 [→Anderson et al. 2003] is a great resource regarding power systems, their control, and stability questions in general.

216 These constraints are not unrealistic in a mostly industrial environment.

217 This magnetization curve is implemented by a diode function generator. It should be noted that this study contains a good example of the implementation of implicit functions.

218 A complex model of an inverter driven asynchronous machine is described in [→Stieber 1967].

219 Cf. →section 4.11.

220 See [→Tympas 2003, p. 73].

221 [→Lyon 1950] contains a wealth of information based on experiences with a British AC network analyzer.

222 Now known as the *American Electric Power Company, Inc.*, (*AEP*).

223 16.10.1913–28.08.1985

224 A wonderful reminiscence of this particular system can be found in [→Lott 2021].

225 Direct analog computers can tolerate such loops due to their lack of amplifying stages coupling the passive computing elements.

226 See →section 11.

227 See [→Nebeker 1994, p. 172].

228 Details can be found in [→Washburn 1962].

229 See [→Washburn 1962, p. 5-155].

230 This study is based on the *coordination equations* developed by Chandler, Dandeno, Glimn, and Kirchmayer, see [→Chandler et al. 1961].

231 This work contains a good overview of typical applications of analog and hybrid computers in these fields.

232 See [→Ranfft et al. 1977, p. 76].

233 Walter Hans Schottky, 23.07.1886–04.03.1976

234 See →chapter 14 for more information about *PACTOLUS*.

235 See →section 6.1.

236 Cf. [→Sydow 1964, pp. 265 f.].

237 The methods described in →section 15.1.9 can also be used in this context.

238 Such a filter basically consists of two separate filters yielding a signal pair with a phase difference of $\pi/2$.

239 See →section 15.1.3.

240 Short for *radio detection and ranging*.

241 Sidelobes are local maxima of emission of an antenna neighboring the desired global maximum, the *mainlobe*.

242 This term describes the effect that the amplitude of the signal reflected by a target depends heavily on its attitude and can change quickly.

243 "*Ein vorhandener Analogrechner ist nicht nur ein Rechengerät, das z. B. die Berechnung eines Regelsystems erlaubt, sondern es kann ebensogut, vor allem wenn es klein und transportabel ist, als elektrischer Regler in ein Regelsystem eingefügt werden, damit in Verbindung mit der echten Regelstrecke der Regler optimal dimensioniert werden kann.*"

244 Cf. [→Elektronische Rechenanlagen 1961]. The system shown in →figure 15.65 contains from left to right two coefficient potentiometers, a couple of fixed resistors, and two chopper-stabilized dual operational amplifiers. A similar device, the *TR-5*, is described in [→Marsh 1961]. It consists of a mounting unit capable of holding up to six analog computing elements.

245 See →section 6.3.

246 Cf. →section 6.4.

247 See [→Elektronische Rechenanlagen 1960].

248 Equally large installations were used in other countries including the Soviet Union (see [→Ushakov 1958/1] and [→Ushakov 1958/2]).

249 This medium-scale analog computer contains 48 operational amplifiers.

250 Such complex simulations were the exception.

251 Basic information about this problem can be found in [→Billet 1965].

252 Cf. [→Schöne 1976/2] for a general presentation of the mathematical modeling of heat exchangers.

253 Franz Fischer, 19.03.1877–01.12.1947, and Hans Tropsch, 07.10.1889–08.10.1935

254 The basics of the simulation of chemical reactor systems are described in [→Starnick 1976]. A generalized real-time simulator for two-phase heat exchangers is treated in [→Mason et al. 1967].

255 A similar simulation is described in [→Worley 1962, pp. 5-80 ff.].

256 More realistic simulations involving many more column trays are usually out of reach of classic analog computer installations. Using hybrid computers, a resource-

sharing approach can be implemented, thus allowing the treatment of problems with large numbers of trays. The advertisement [→Computer Systems Inc. 1960] shows how the DYSTAC analog computer with its analog storage elements can be used to simulate distillation columns without needing a full-fledged hybrid computer.

257 The development of adaptive control systems began in the 1960s.

258 See [→Hill et al. 1964, p. 318].

259 Annotations to this can be found in [→Rosko 1968], while similar studies are described by [→Feilmeier 1974, pp. 259 ff.] and [→Michaels et al. 1971].

260 See →section 4.12.

261 This method can be seen as a precursor to genetic programming.

262 See [→Tacker et al.].

263 See [→Aspray 1993, pp. 40 ff.].

264 Similar studies were performed for railway vehicles, too.

265 In this study a soviet made MN-7 analog computer was used.

266 A similar, although more complex, simulation of an automatic transmission was implemented at *Daimler Benz* using a Telefunken RA 800 (see →section 6.4). This analog computer system was also used to develop one

of the first anti-lock braking systems (Bernd Acker, personal communication, 23.08.2007).

267 See →section 15.14.1.

268 Such a simulation is described in [→Giloi et al. 1963, pp. 45 ff.].

269 Although this sounds like a contradiction, such vehicles were explored since they promised very low frictional losses and offered better lateral stability than many traditional railway systems.

270 Higher frequencies can be mostly neglected due to the long time-constant of the oil tanker itself.

271 This system was later known as the *CDC 7600*.

272 The basic requirements for such systems are described in [→McCallum[a]].

273 Short for *Landing Craft Mechanized*, colloquially also known as *Mike Boat*.

274 See [→Kastner 1968/2].

275 Cf. [→Lowe[a]].

276 These fully transistorized machines used a machine unit of ±100 V, very unusual for transistorized analog computers, and were mainly used in applications where precision and compatibility with older external equipment was of prime importance.

277 See →section 6.3.

278 See →section 4.10.

279 [→Howe 1957] details on coordinate systems and transformations suitable for flight simulator systems.

280 This is one of the reasons why the idea of hybrid computers initially emerged at companies such as *Ramo-Wooldridge Corporation* and *Convair Astronautics*, cf. →section 10.

281 Krafft Arnold Ehricke, 24.03.1917–11.12.1984, see [→Ehricke 1960].

282 See →https://www.nasa.gov/centers/ames/multimedia/images/2010/iotw/reeves.html, retrieved 07.08.2022

283 This first *Reeves Electronic Analog Computer* system was developed in 1947 by *Reeves Instrument Corporation*.

284 NACA; the *National Advisory Committee for Aeronautics* was established on March 3rd, 1915, to direct research and development in the field of aerospace technology. In 1958, the *National Aeronautics and Space Act* was passed by congress as a reaction to the fear of the United States falling behind the Soviet Union from a technology perspective. This became the foundation of *NASA* into which NACA was incorporated subsequently.

285 See [→EAI 1961/1].

286 [→Gottzein[a]] also makes a point stating that hybrid computers are irreplaceable in the aerospace industry.

287 This machine was already a “super computer” in its time.

288 See [→Bedient et al. 1968]. →Figure 10.4 (page 257) shows such a Comcor system on the right.

289 A good example for this is the landing gear of the *Me 262* jet, which was plagued by a variety of problems, all of which could, and often did, result in fatal accidents.

290 See [→McLeod et al. 1958/6, p. 1995] and [→Moreland 1954].

291 Cf. [→Johnson 1963, p. 223].

292 See [→McLeod et al. 1958/6, p. 1996].

293 See [→Engelsman 1992] for a comprehensive description of shimmy. In general, *shimmy* describes movement of some mechanical system by self excitation. This is an especially important effect in aircraft landing gear design.

294 See [→Drake et al. 1963, p. 91].

295 See →section 15.2.4.

296 Cf. [→McLeod et al. 1958/4, p. 1222].

297 Flutter effects in aircraft became a major problem early in World War II. In 1940 Konrad Zuse (22.06.1910–18.12.1995) was approached by Alfred Teichmann (24.11.1902–21.09.1971), one of the leading German airframe specialists who realized that a relay based digital computer could be successfully applied to flutter

problems. This led to the development of Zuse's *Z3* computer.

298 See [→Smith et al. 1959, p. 12].

299 The basics of such flight simulation systems are described in [→Bauer 1962/1], [→Wass 1955, pp. 39 ff.], and [→Korn et al. 1956, pp. 115 ff.]. A short historical overview of flight simulators in general can be found in [→Zazula et al. 2013].

300 See [→Atherton 2004].

301 See [→Jenkins et al. 2003] for a comprehensive description of the X-15.

302 These relays were probably reed relays with mercury wetted contacts.

303 See [→Ciardiello 2009] for more information on Typhoon.

304 See →section 15.10.8 and [→Hansen 2005, p. 147].

305 The development of analog flight simulation at NASA's flight research center is described in [→Waltman 2000]. Typical analog computer application examples at *Autonetics* (this was a division of *North American Aviation* that primarily developed avionics systems and inertial navigation systems) are described in [→McLeod et al. 1958/1, p. 122].

306 A more detailed description of the simulation of autopilots on analog computers can be found in [→Korn et al. 1956, pp. 128 ff.].

307 See [→Wass 1955, p. 43].

308 A good introduction to the mathematical modeling of aircraft and missiles can be found in [→Schweizer 1976/5], while [→Schweizer 1976/4] describes the implementation of a flight simulation system on a hybrid computer.

309 Short for *Three-Dimensional Analogue Computer*. See [→Wass 1955, p. 213] for additional information.

310 See [→The Times 1954].

311 A detailed account of the system, its construction and structure, is given in [→Spearman et al. 1955].

312 Just the hydraulic subsystem for TRIDAC required 400 hp (about 300 kW) while the overall TRIDAC system was rated at 650 kW.

313 See [→Atherton 2004, p. 301].

314 See [→Lange 2006, pp. 249 ff.], [→Tomayko 1985, p. 233], and [→Hoelzer 1992, p. 18].

315 A plethora of information about flight tables can be found in [→Bauer 1962/2].

316 His obituary, which is very worth reading, can be found at →https://www.nasa.gov/centers/dryden/news/X-Press/stories/2004/073004/new_day.html, retrieved 23.02.2022.

317 Short for *Tactical Avionics System Simulator*.

318 An EAI 231R coupled with a DOS-350 forms a HYDAC 2000 computer system, see →section 6.3.

319 A detailed description of the system can be found in [→Kenneally et al. 1966].

320 See [→Hoffert 1964] for details on the Do 31.

321 See [→Rogers 1963] for details on this particular problem.

322 MBB, *Messerschmitt-Bölkow-Blohm*, was one of the largest German aerospace companies.

323 A list of analog and hybrid computer installations in use for aeronautical research in 1971 can be found in [→Pirrello et al. 1971].

324 09.04.1924–27.09.1956

325 See [→Waltman 2000, p. 138] and [→Hansen 2005, p. 148].

326 Short for *angle of attack*.

327 See [→Waltman 2000, p. 138].

328 See [→Waltman 2000, p. 140].

329 05.08.1930–25.08.2012

330 See [→Hansen 2005, p. 148]. The analog X-15 flight simulation system is described in [→Cooper 1961].

331 See [→McLeod et al. 1958/5, pp. 1387 f.].

332 An interesting study of the impact of instrumentation on a pilot's performance can be found in [→ Schweizer 1976/3, pp. 561 ff.].

333 The basics of such picture generation systems are described in [→ Schweizer 1976/7, pp. 394 ff.].

334 See [→ Baker 1978].

335 Bruce Baker remarks on this model (see [→ Baker 1978, p. 2]): "*Topography is rolling hills modeled after West Germany.*"

336 See [→ Baker 1978, pp. 5 f.].

337 Cf. [→ Schweizer 1976/7, pp. 396 f.].

338 See [→ Waltman 2000, pp. 59 ff.], [→ Berry et al. 1966], and especially [→ Kroll et al. 1966] for a detailed description of this incredible system. More information about this type of simulator can be found in [→ Armstrong et al. 1962], [→ McFadden et al. 1958], and [→ Kidd et al. 1961].

339 [→ Szalai 1971] describes such a study in which the JetStar was validated for the simulation of the handling qualities of large transport aircraft.

340 See [→ Waltman 2000, p. 64].

341 06.06.1925–04.02.2015

342 See [→ Waltman 2000, p. 64].

343 Cf. [→ Department of the Army 1956].

344 See [→ US Army 1956, p. 4].

345 See [→US Army 1956, p. 3]: "*[T]he computer needs a memory. […] It must receive and remember position data for 4 seconds until it knows exactly the direction and speed of the motion involved. The memory of the computer lies in its differentiating circuits.*"

346 See →section 12.

347 See [→Hall 1996, pp. 38 ff.].

348 A NOR-gate is an OR-gate with an inverter connected to its output.

349 A quick summary of the Polaris simulation program can be found in [→Lockheed 1961].

350 Short for *Coordinate Rotation Digital Computer*.

351 See [→Volder 1959].

352 LH_2 denotes liquid hydrogen, *LOX* is liquid oxygen.

353 See [→Jackson 1960, p. 393].

354 AGWAC contained more than 400 operational amplifiers, 20 fully electronic multipliers, six servo multipliers (each having about 20 ganged potentiometers), and some resolvers as well as other special computing elements (see [→Benyon 1961]).

355 Associated Press, July 26th, 1950.

356 See [→Bauer et al. 1957, p. 239].

357 Cf. [→Teuber 1964].

358 14 times of earth's gravitation.

359 See [→Steinhoff et al. 1957, p. 243].

360 See [→Telefunken/6, p. 1].

361 The only other digital computer capable of competing with the Cray-1 solving problems like this was an *AD-10*, developed by *Applied Dynamics*. This system was optimized for the solution of problems typically solved on analog computers.

362 See [→Krause 1964, p. 39].

363 See [→Fagen 1978, p. 569].

364 See [→Waltman 2000, p. 107].

365 *Thompson-Ramo-Wooldridge*

366 See [→Bekey et al. 1968, p. 392].

367 See [→Bekey et al. 1968, p. 394].

368 A comprehensive history of NASA engineers with a focus on project Apollo can be found in [→Fries 1991].

369 See [→Hansen 2005, p. 314].

370 See [→Burns 1965].

371 See [→Clark et al. 1958, p. 1].

372 More background information can be found in →https://if50.substack.com/p/hutspiel-and-dr-dorothy-

k-clark, retrieved 28.08.2022. More information on Dr. Dorothy Kneeland Clark can be found in [→Phalanx 2020].

373 See [→Clark et al. 1958, p. 16].

374 See [→Owens 1986, p. 66].

375 An early analog computer aimed at engineering education is the *Comdyna GP-6*, developed in the late 1960s by Ray Spiess, see [→Spiess 1992] and [→Spiess 2005].

376 A great account of the application of analog computers in academia and industry is given in [→Howe 2005].

377 See →section 6.6.

378 See [→Fischer 2022].

379 30.09.1914–2000

380 See [→Somers 1980] and →http://en.wikipedia.org/wiki/John_Whitney_(animator), retrieved 03.03.2013.

381 10.06.1906–14.07.2006

382 See [→Heidersberger[a], p. 7] and [→Hoffmann 2006, p. 35].

383 These are named after Jules Antoine Lissajous (04.03.1822–24.06.1880).

384 More information about this can be found in [→Hoffmann 2006].

385 14.05.1927–16.07.2022

386 See [→Hobby 1969, p. 43] and [→Deken 1984] for more examples and background information.

387 Son of Heinrich Heidersberger.

388 05.05.1908–13.02.1997

389 25.10.1920–10.11.1994

390 See [→N. N. 2006].

391 See →http://www.m-e-g-a.org/de/research-education/research/t42-tennis-for-two/, retrieved 03.03.2013.

392 A similar example can be found in [→Fischer 2022, p.12].

393 Despite the high complexity of analog computer setups for the generation of such pictures, many simulation setups used similar approaches to give a human operator a direct feedback of a manually controlled mechanical system, etc. A good example is the analog generation of the picture of a moving overhead crane described in [→Breitenecker 1978/2].

394 This might change due to THE ANALOG THING, see →section 6.6, that has been adopted in the electronic music scene.

395 See [→Seay 1964, p. 17].

396 This early work was stimulated by [→Appleton et al. 1975].

397 A more recent discussion on the application of analog computers in music, involving Hainbach, Hans Kulk, and Bernd Ulmann, has been recorded on 30.08.2020 and can be viewed at →https://www.youtube.com/watch?v=bgyzeyatS-0, retrieved 07.08.2022.

398 Personal communication to the author.

399 This was, in fact, the world's third analog computer center – two earlier computer centers were setup in Princeton (New Jersey) and Los Angeles (California).

400 This model was a precursor of the highly successful model EAI 231R.

401 See →section 6.3.

402 The EAI Brussels computer center did quite a lot of biomedical work, see [→Claes 1975].

403 See [→Instruments and Automation 1957/5].

1 There were approaches to implement time-sharing of analog computer components to alleviate this problem at least a bit, see [→TU Wien 1977/3, pp. 25 ff.], [→Kleinert 1979/2], and [→Minogue et al. 1960]. This could have at least reduced the price of analog computers, but without solving the actual issue.

2 See [→Reinhardt 1967].

3 Of course, this argument breaks down when solution times get unbearably large such as years, decades, millennia, or even exceeding the life-time of the universe.

4 Ideas such as the sequential analog computer described in [→Schmid 1965], of course, lack this central characteristic.

5 This rather sudden change in technology and computational paradigm is a near-perfect example for a *paradigm shift* as described by Thomas Samuel Kuhn (18.07.1922–17.06.1996) in his seminal work [→Kuhn 1996].

6 Gene Myron Amdahl, 16.11.1922–10.11.2015

7 See [→Lange et al. 2020].

8 Even direct analogies may be used in certain cases to great advantage, see below. Some simple, yet impressive examples of such problems can be found in [→Dewdney 1988/1], [→Dewdney 1988/2], and [→Hoffman 1979].

9 See [→Guo et al. 2016]. Cf. [→Köppel et al. 2021] for this and the preceding topic.

10 See [→Tisdale 1981, p. 2].

11 See [→Coveney 2024].

12 01.10.1930–31.05.2009

13 See [→Menzel et al. 2021]. [→Nalley 1969] suggested a small DDA as a co-processor to speed up *Z*-transformations as early as 1969. [→Hyatt et al. 1968] describe an *electrically alterable DDA*, and [→Elshoff et al. 1970] developed a DDA working on binary floating point arithmetic. [→McGhee et al. 1970] describe a DDA

working on very long integers, an attempt to minimize the computational problems caused by floating point numbers. Practical implementations of DDAs on FPGAs are described in [→Land[a]].

14 Another interesting development are hybrid value representations for modern analog computers. This could solve the problem of problem scaling, thereby simplifying programming considerably (see [→O'Grady 1966], [→Skramstad 1959], and [→Wait 1963]).

15 Another interesting automatically reconfigurable analog computer, not mentioned before, is the *US Army Material Command Advanced Hybrid Computer System* (*AHCS*), which could be accessed through distributed terminals connected to two digital computers, which were in turn tightly coupled with a central analog computer. The switching matrices required more than 10^5 switches and were connected to an Applied Dynamics *AD/FOUR* analog computer (see [→Howe et al. 1970], [→Howe et al. 1975], and [→Gracon et al. 1970]).

16 See [→Achour et al. 2020].

17 →https://anabrid.com, retrieved 31.08.2025.

18 See [→Herzog et al. 2018].

19 See [→Schröder[a]] and [→Solar 1978]. [→Husinsky 1976] describes a problem in laser spectroscopy which would not have been possible to solve in the analog domain without an automatic scaling system.

20 See [→Richards 1968].

21 [→SRC 2021, pp. 12 ff.] contains a wealth of examples and background information on this particular topic.

22 See [→Cowan 2005, pp. 182 ff.]. A similar approach is described by [→Karplus et al. 1972]. [→Huang et al. 2016] and [→Huang et al. 2017] focus on the application of analog computers as linear algebra accelerators for *HPC* in general. An interesting memristor based analog coprocessor for HPC is described in [→Athreyas et al. 2018]. See also [→Kolms et al. 2020]. An interesting analog computer, the *Analog Iterative Machine*, (*AIM*) aimed at optimization problems is described in [→Kalinin et al. 2023]. Interestingly, this particular machine also found its way into AI applications, see [→Kalinin et al. 2025].

23 01.12.1928–25.03.1995

24 See [→Mills/1], [→Mills et al. 2006], [→Mills 1995], and [→Mills/2] for more information.

25 See [→Cowan 2005] and [→Cowan et al. 2006].

26 See [→Cowan 2005].

27 The area of robotics is especially well suited for the application of analog computing technologies since it requires a lot of sensor data processing, actor control, and trajectory planning, see [→Farshchi 2016].

28 See [→Bai et al. 2015].

29 See [→Basu et al. 2010] for the description of a floating-gate based FPAA.

30 See →https://zrna.org, retrieved 04.09.2022.

31 See [→Hu et al. 2024] and [→Maxfield 2020].

32 An example for a reconfigurable analog system for low-power signal processing is described in [→Schlottmann et al. 2012].

33 $\mathrm{ReLU}(x) = x^+ = \max(0, x)$

34 See [→Maxfield 2020]. This idea is often called *in memory computing* with matrices of (programmable) resistors being the "memory". See [→Demler 2018], [→Khan et al. 2024], [→Park et al. 2024], and [→Zheng et al. 2024].

35 As a side note, it is not uncommon in the digital world to stack two or three chips such as CPU and memory, etc., on top of each other by means of some suitable *interposers*. This allows for a high degree of connectivity between these chips, but the height of the stack is rather limited due to the amount of heat that must be removed from the chips. Using highly energy efficient CMOS implementations of analog computers, much larger stacks could be built, allowing for an connection density unheard of today, rivaling that of biological systems such as a brain.

36 04.11.1876–11.10.1953

1 The system was designed by Karl-Heinz Dahlmann at anabrid GmbH. An even simpler system which might also serve as a starting point for hobbyist developments is described in detail in [→Grubert 1980].

2 See [→Onsemi 2005, pp. 63 ff.] for details.

3 Short for 5100 ohm.

4 10^{-9} F.

5 The small error resulting from just paralleling both capacitors can be neglected in a small analog computer like THE ANALOG THING.

6 See [→Analog Devices/1].

7 See →https://fscdn.rohm.com/en/products/databook/datasheet/ic/power/power_switch/bd2244g-m-e.pdf, p. 18, retrieved 20.03.2025.

8 This potentiometer has a logarithmic characteristic to facilitate setting small values.

9 Jumpers J15, J18, and switch J16 are optional and can be used to convert the repetitive run mode to single run.