

---

# Analogrechnen

Prof. Dr. Bernd Ulmann

18. Februar 2014

# Einleitung

Ein Vortrag über Analogrechner?

Im 21. Jahrhundert?

Ist das nicht unzeitgemäß?

**NEIN! :-)**

Warum sollte man heutzutage über Analogrechner sprechen?  
Gründe gibt es viele, der wichtigste ist jedoch folgender:

Das Paradigma des Analogrechnens ermöglicht einen extrem feingranularen Parallelismus, der high-performance und/oder low-power Computing möglich macht.

Im Folgenden werden zunächst einige Grundbegriffe des Analogrechnens erläutert, bevor exemplarische Beispiele behandelt werden. Das Ende des Vortrages bildet ein Ausblick auf mögliche Anwendungen des Analogrechnens im 21. Jahrhundert.

# Grundlagen

Zunächst eine wichtige Feststellung:

Analogrechner sind **nicht** (zumindest nicht zwingend) dadurch gekennzeichnet, dass sie mit (quasi-)kontinuierlichen Werten wie Spannungen etc. arbeiten, obwohl dies immer wieder zu ihrer Definition herangezogen wird!

Das Wesen eines Analogrechners liegt in der Tatsache, dass er auf Basis eines *Modells*, eines *Analogons* arbeitet.

Es muss also zwischen analogelektronischen Analogrechnern und digitalen Analogrechnern (sogenannten *DDAs*, kurz für *Digital Differential Analyzer*) unterschieden werden.

Bei Analogiebildungen ist zwischen zwei grundlegenden Ausprägungen zu unterscheiden:

**Direkte Analogien:** Diese basieren auf gleichen oder zumindest ähnlichen physikalischen Effekten wie das zu untersuchende Problem.

**Indirekte Analogien:** Analogelektronische Analogrechner sind ein schönes Beispiel hierfür – alle zu untersuchenden Fragestellungen werden als elektronische Modellschaltungen abgebildet, ungeachtet des eigentlich zu modellierenden Effektes.

Eigentlich muss also nicht zwischen Analog- und Digitalrechner unterschieden werden, vielmehr verläuft die Trennlinie zwischen diesen beiden Klassen von Systemen:

- Speicherprogrammierte (Digital-)Rechner
- Analogrechner

Rechner der erstgenannten Gruppe haben eine feste Struktur und werden durch Vorgabe eines Algorithmus programmiert, während Rechner der zweiten Gruppe dadurch programmiert werden, dass ihre Struktur (Verschaltung) dem zu lösenden Problem angepasst wird (z.B. durch explizite Steckverbindungen bzw. heutzutage in Form von FPGAs oder FPAAs).

TRUITT bringt das schön auf den Punkt (siehe [TRUITT et al. 1960] [p. 1-40/41].):

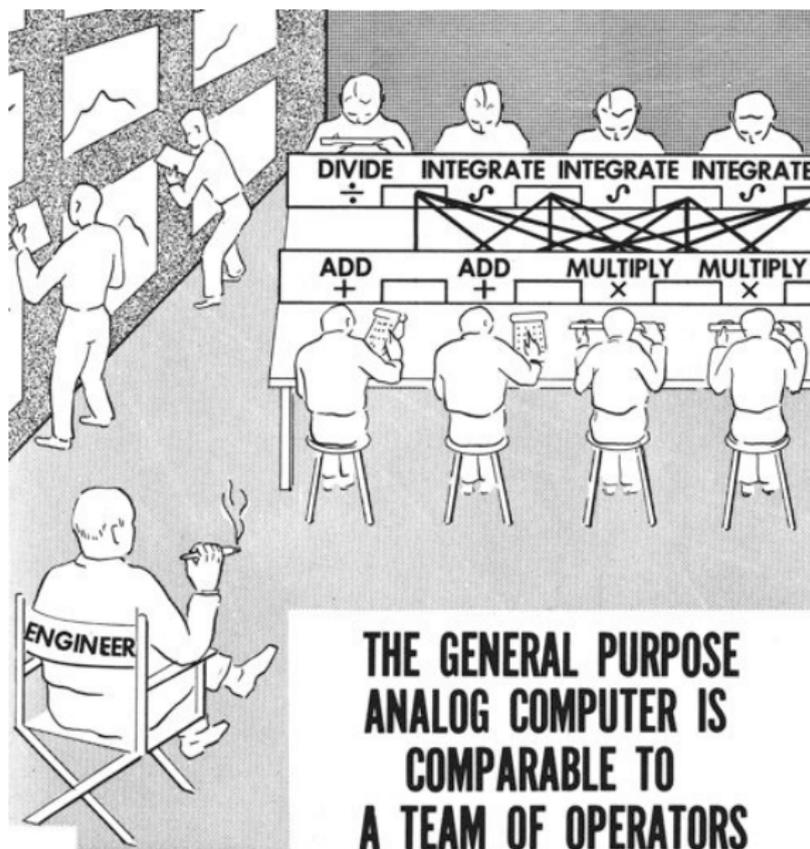
**A DIGITAL COMPUTER is equivalent to a very reliable,  
highly paid, exceptionally fast**

**(300 to 10,000**

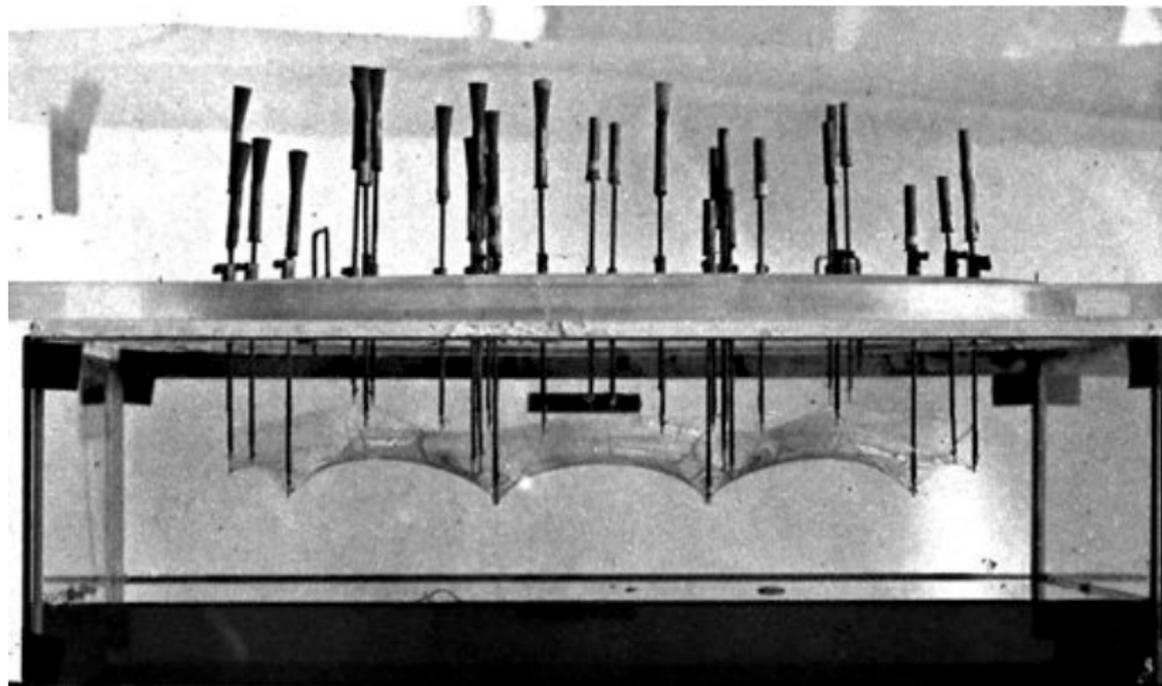
**operations/second)**

**DESK CALCULATOR  
OPERATOR**

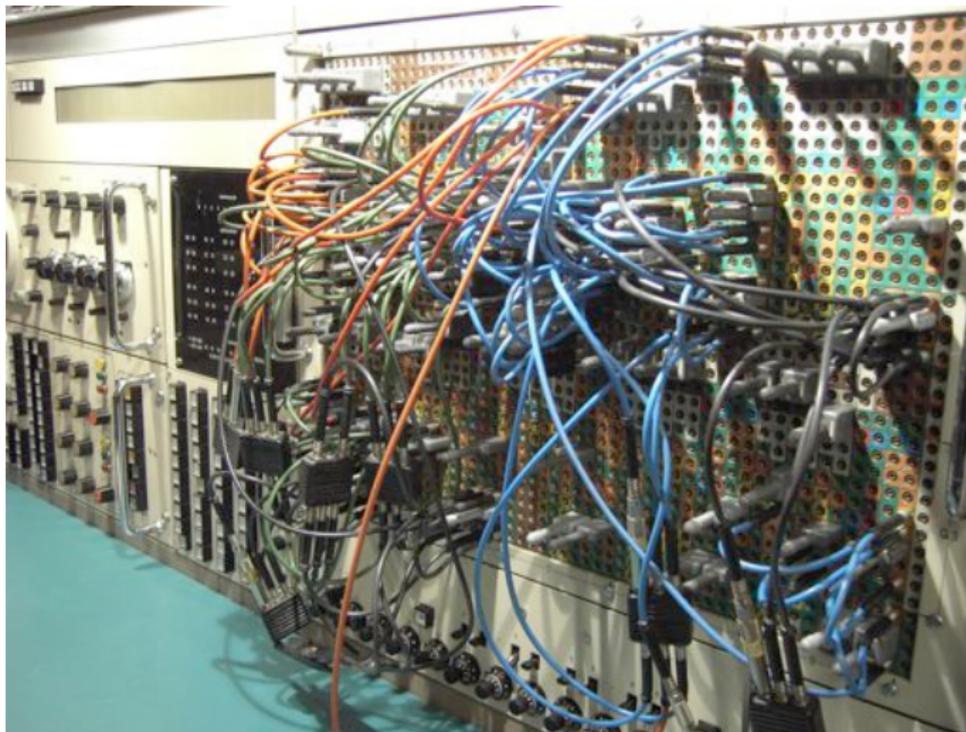




Ein typisches Beispiel für eine direkte Analogie ist die Bestimmung von Minimalflächen mit Hilfe von Seifenfilmen oder Geweben (siehe [DRESSLER 1972][p. 52]):



Ein Beispiel für eine indirekte Analogie ist das folgende Setup einer RA 770 zur Simulation der Umströmung eines JOUKOWSKI-Profiles:



Was sind typische Implementationsvarianten von Rechenelementen für Analogrechner?

- mechanische Rechenelemente,
- hydraulische Elemente,
- analogelektronische Rechenglieder sowie auch
- rein digitale Techniken.

Aus heutiger Sicht sind selbstverständlich analogelektronische Rechenelemente bzw. rein digitale Implementationen am interessantesten und erfolgversprechendsten.

- Vorteile:
- Extrem (!) geringe Leistungsaufnahme
  - Numerische Stabilität ist (eigentlich erstaunlicherweise) selten ein Problem, da Integration eine quasi *natürliche* Operation darstellt

- Nachteile:
- Beschränkte Genauigkeit (kann mitunter auch ein Vorteil sein) – maximal  $10^{-4}$  erreichbar
  - Rauschen, Drift (heutzutage kein großes Problem mehr)
  - Als freie Variable bezüglich der Integration steht nur die Zeit  $t$  zur Verfügung, was das Lösen partieller DGLs nur mit teils extremem Aufwand möglich werden lässt.
  - Funktionserzeugung schwer bis sehr schwer

- Vorteile:
- Beliebige Genauigkeit
  - Hervorragend für die Implementation mit FPGAs geeignet
  - Funktionsgenerierung ist einfach (Lookup-Tables)

- Nachteile:
- Deutlich höhere Leistungsaufnahme als vergleichbare analogelektronische Schaltungen
  - Wahl des geeignetsten Integrationsverfahrens nötig, numerische Stabilität wird wieder eine Frage

Typische Rechenelemente eines Analogrechners:

- Koeffizientenglieder (Spannungsteiler / Multiplizierer bei digitaler Implementation)
- Summierer
- Integrierer
- Multiplizierer (analogelektronische Implementation ist schwer)
- Funktionsgeber (einfach mit Lookup-Tables, schwer mit rein analogelektronischer Implementation)
- Entscheidungsglieder (Komparatoren) und Schalter

- Analogrechner werden grundsätzlich dadurch programmiert, dass die zur Verfügung stehenden Rechenelemente in geeigneter Weise miteinander verschaltet werden, um ein gewünschtes Modell herzustellen.
- Im Unterschied zu Digitalrechnern gibt es (zumindest bei reinen Analogrechnern) keine Möglichkeit eines Tradeoffs zwischen Zeit und Komplexität, wie er bei speicherprogrammierten Digitalrechnern möglich ist. D.h. erfordert ein Problem zu seiner Lösung mehr Rechenelemente als zur Verfügung stehen, kann es nicht gelöst werden.
- Alle Rechenelemente arbeiten vollständig parallel (keine explizite Synchronisation nötig, kein VON NEUMANN-Flaschenhals etc.).

# Geschichte

Der früheste Analogrechner ist vermutlich der *Mechanismus von Antikythera* (ca. 150 v. Chr.):



(Siehe [http://upload.wikimedia.org/wikipedia/commons/6/66/NAMA\\_Machine\\_d%27Anticythre-1.jpg](http://upload.wikimedia.org/wikipedia/commons/6/66/NAMA_Machine_d%27Anticythre-1.jpg))

Was konnte dieser Analogrechner leisten?<sup>1</sup>

- Es konnten siderische, synodische, drakonische und andere lunare Zyklen berechnet werden.
- Hierbei wurde bereits eine lunare Anomalie berücksichtigt (die aufgrund der Elliptizität der Mondbahn nicht konstante Geschwindigkeit).
- Es konnten Sonnen- und Mondfinsternisse berechnet werden – sogar Aussagen über die Sichtbarkeit solcher Ereignisse war möglich.
- ... vieles mehr – die Forschung konnte noch lange nicht alle Fragen zu diesem einzigartigen Artefakt klären.

Der Mechanismus von Antikythera ist ein schönes Beispiel für einen mechanischen Analogrechner.

---

<sup>1</sup>Siehe [McCARTHY 2009].

Fast 2000 Jahre später entwickelte Lord KELVIN die Idee, mit Hilfe mechanischer Rechenelemente Differentialgleichungen zu lösen, wobei er das bis heute verwendete KELVINSche *Rückführungsverfahren* entwickelte, das zum zentralen Programmierparadigma der Analogrechnerära wurde:<sup>2</sup>

*[...] it seems to me very remarkable that the general differential equation of the second order with variable coefficients may be rigorously, and in a single process solved by a machine.*

Leider entstand aus diesen Ideen kein praktisch und allgemein einsetzbarer Analogrechner.

---

<sup>2</sup>Siehe [THOMSON 1876].

Dennoch stellte Lord KELVIN 1876 den folgenden Gezeitenrechner fertig, der bereits 10 Partialtiden berücksichtigte:

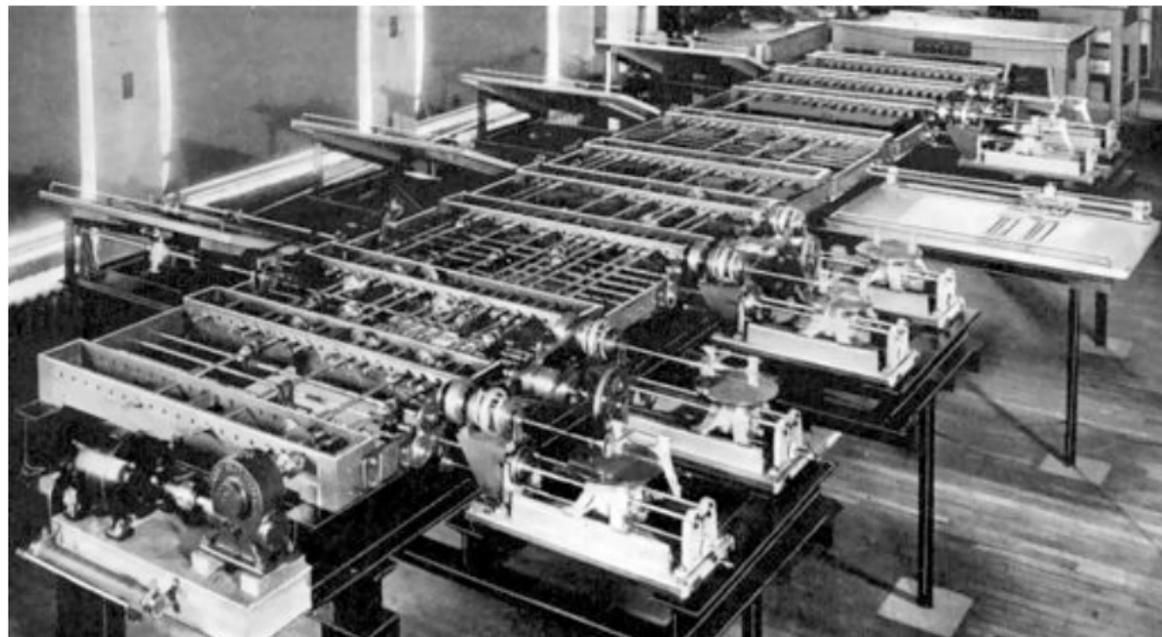


Ein weites Anwendungsfeld fanden mechanische Analogrechner in Form sogenannter *Feuerleitrechner* in den ersten beiden Weltkriegen. Hier gezeigt ist der *Torpedo Data Computer Mark 3* (nach [BORD44][S. 150]) Frontverkleidung:



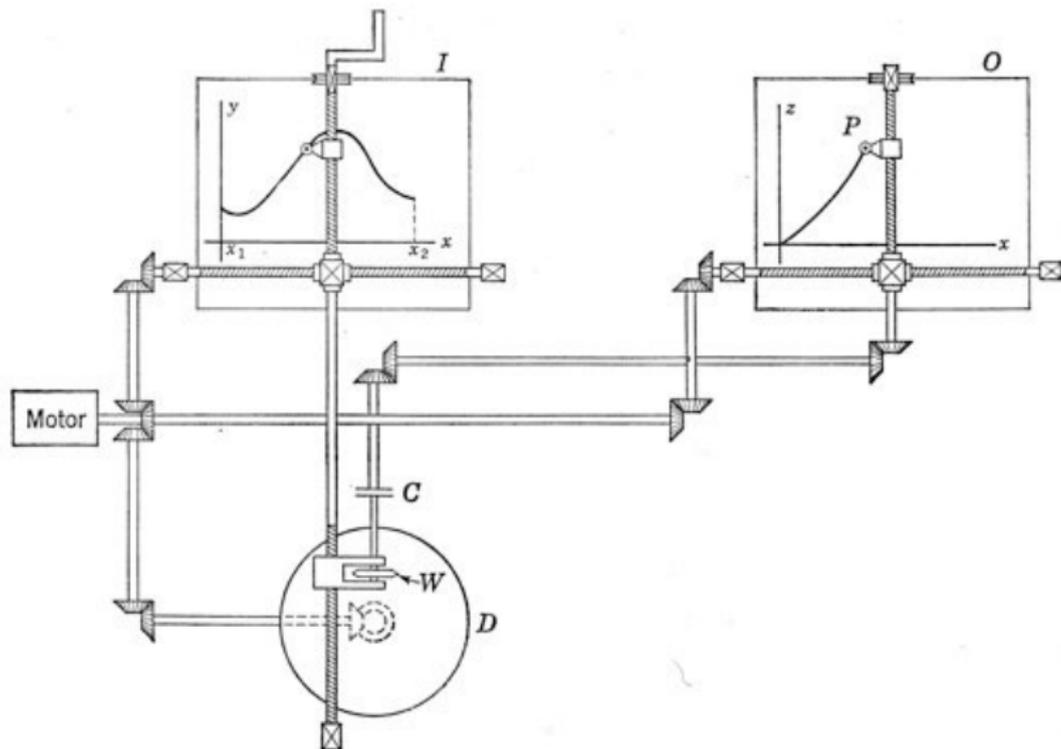
Deutlich allgemeiner einsetzbar, da rekonfigurierbar, waren sogenannte *Differentialanalysatoren*, die im Wesentlichen aus einer Reihe mechanischer Rechenelemente bestanden, die über Getriebestangen, später auch über Servos und Motoren miteinander verschaltet werden konnten.

Einen der ersten praktisch einsetzbaren mechanischen Differentialanalysatoren entwickelten VANNEVAR BUSH et al. am MIT in den späten 1920er Jahren:



(Abbildung aus [Meccano 1934/2][p. 443].)

Ein einfaches Einsatzbeispiel eines DA (ohne Rückkopplung) findet sich in [KARPLUS et al. 1958][S. 190]:



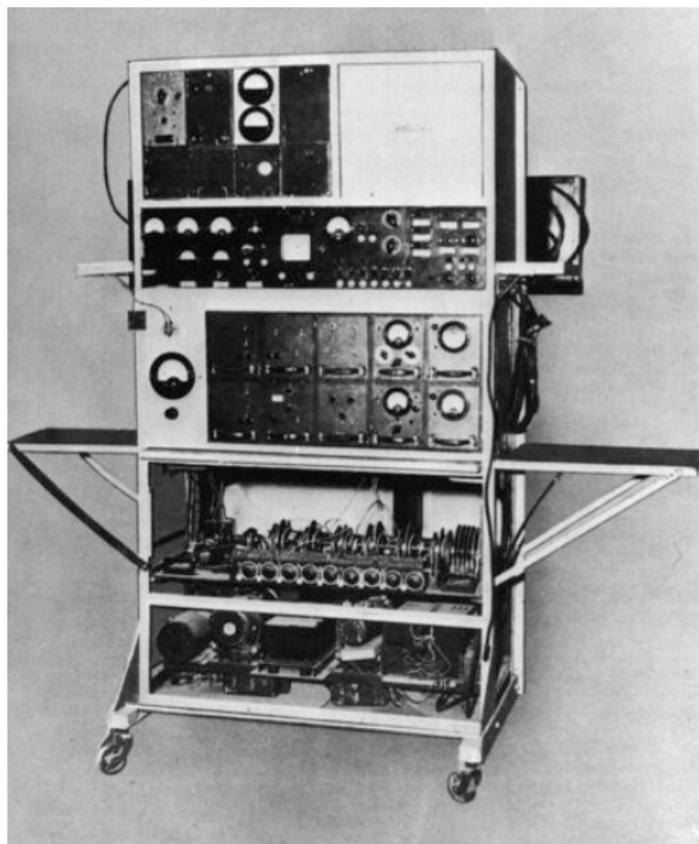
Neben dem Hauptvorteil ihrer Robustheit haben mechanische Analogrechner eigentlich nur Nachteile: Sie sind aufgrund der vergleichsweise großen bewegten Massen langsam, Genauigkeiten von  $10^{-3}$  und besser sind nur mit extremem Aufwand zu erzielen, sie sind extrem wartungsaufwändig und sie sind schlicht schwer.

In den frühen 1940er Jahren wurden unabhängig voneinander gleich zwei elektronische Analogrechner entwickelt, zu denen sich 1945 noch ein dritter gesellte:

- HELMUT HOELZERS *Mischgerät* zur Steuerung der Rakete A4 (ein Spezialanalogrechner) sowie ein von ihm entwickelter universeller Analogrechner, der bis in die 1950er Jahre hinein in den USA im Redstone Arsenal eingesetzt wurde,
- GEORGE A. PHILBRICKS *Polyphemus* zur Prozesssimulation in der chemischen Industrie und
- der repetierende Analogrechner von A. B. MACNEE (MIT).



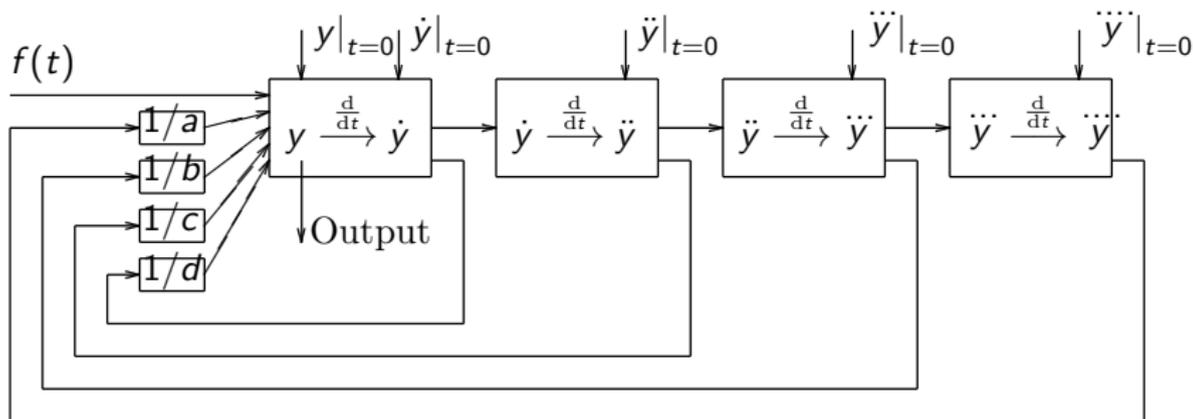
(Abbildung von ADRI DE KEIJZER mit freundlicher Genehmigung.)



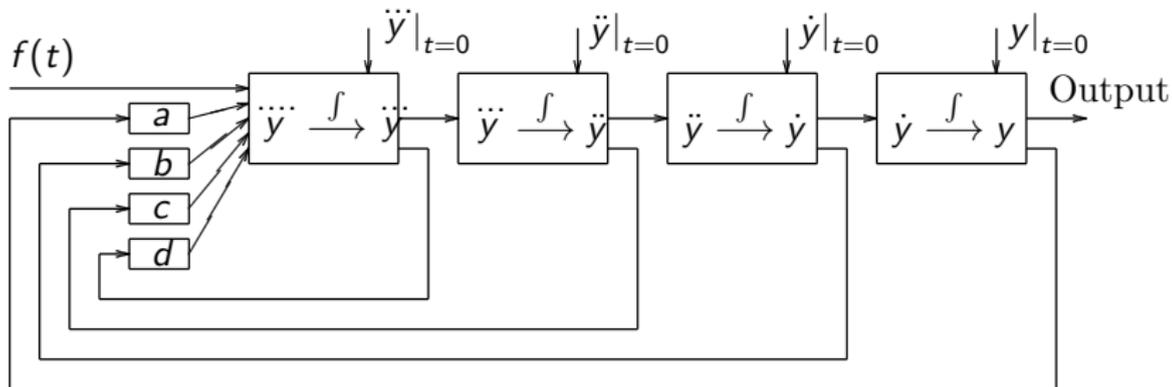
Lösung einer DGL vierten Grades

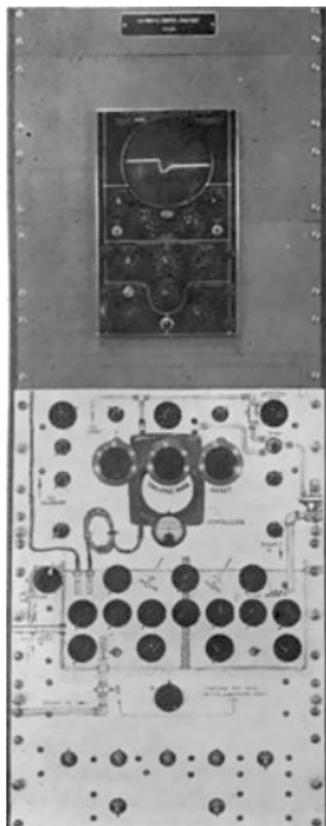
$$\ddot{\ddot{y}} + a\ddot{y} + b\dot{y} + cy + dy = f(t)$$

durch Differentiation (siehe [HOELZER 1946][fig. II, 2]):



Lösung mit Hilfe von Integrierern (welchen Vorteil hat das?):





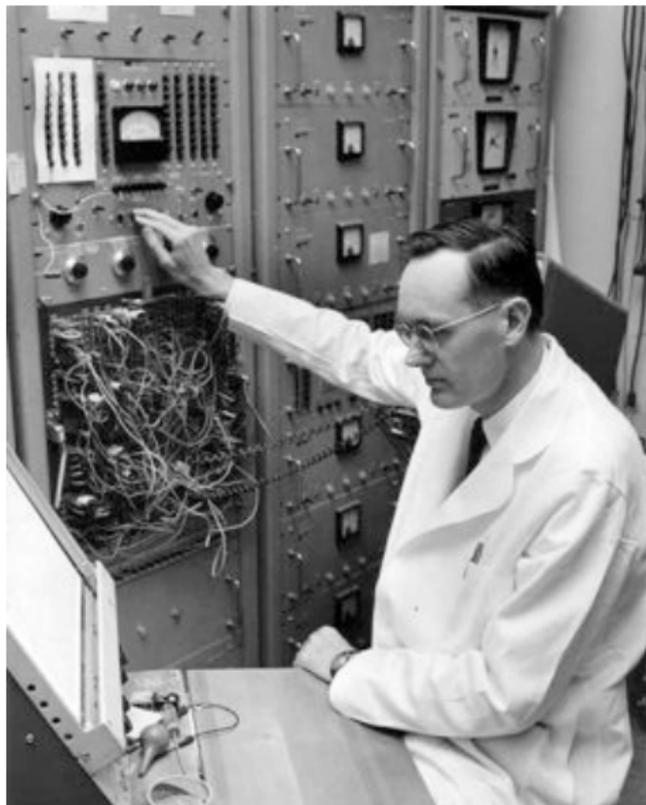
In den späten 1930er Jahren entwickelte PHILBRICK mit *Polyphemus* einen elektronischen Analogrechner zur Prozesssimulation (siehe [PHILBRICK 1948][p. 108]).

Dargestellt ist hier ein Setup zur Simulation eines Prozessmodells, bestehend aus einem zweistufigen Heizbad mit Dampf- und Kaltwasserzuführung.

Analogelektronische Analogrechner machten in den Jahren nach 1945 eine extrem stürmische Entwicklung durch und prägten Bereiche wie die Simulation in Luft- und Raumfahrt, die chemische Industrie, Medizin und Biologie, Mechanik, Nukleartechnik, Starkstromtechnik, Prozessautomation, Nachrichtentechnik, Automobilbau uvm. in unvergleichlichem Maße.

Die wichtigsten Hersteller waren Electronic Associates, Telefunken, Beckman, Systron Donner, Simulators Inc., Dornier, Goodyear uvm.

Die folgenden Folien zeigen exemplarisch einige typische kommerzielle Analogrechenanlagen.



RICHARD FITZHUGH löst die Hodgkin-Huxley-Gleichungen, mit deren Hilfe neuronale Aktionspotentiale und Entladungsvorgänge beschrieben werden, mit Hilfe eines Beckman Analogrechners:<sup>3</sup>

<sup>3</sup>Source: Izhikevich E. M., FitzHugh R.

EAI 231R-Installation am „Deutsches Elektronensynchrotron“, Hamburg:<sup>4</sup>.



<sup>4</sup> Bild: Inge Borchardt

EAI-Installation des DLR (1970er Jahre):<sup>5</sup>

<sup>5</sup> Bild: Dr. Jessika Wichner, DLR, Signatur FF-557.

EAI Pacer (1970er Jahre):





RICHARD DAY führt eine Trägheitskopplungssimulation durch:<sup>6</sup>

---

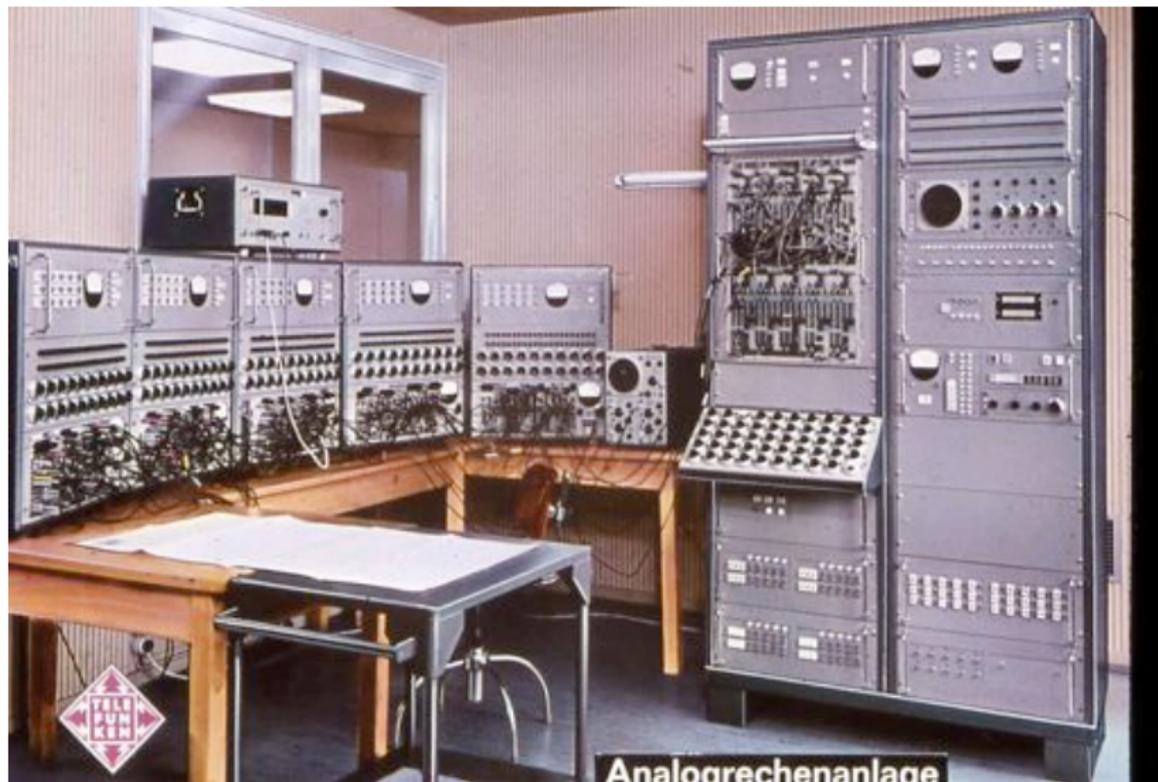
<sup>6</sup> NASA photo E-1841.



WINKLER, Ingenieur des „Labor Walter“ an einem Telefunken RAT 700 Tischanalogrechner.<sup>7</sup>

<sup>7</sup> Siehe [GERWIN 1964][p. 84].

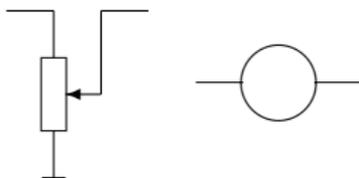
Fünf RAT 700 und eine minimal ausgebaute RA 800:



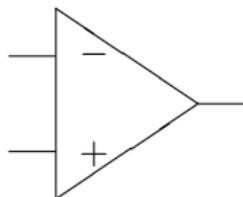
# Beispiele

Für die folgenden einführenden Beispiele werden einige typische Rechenelemente von Analogrechnern und ihre Schaltsymbole benötigt.

Das einfachste ist das *Koeffizientenpotentiometer*, das die Multiplikation eines Wertes mit einem Koeffizienten  $0 \leq a \leq 1$  ermöglicht:



Das wichtigste aktive Element eines elektronischen Analogrechners ist der sogenannte *Operationsverstärker*:

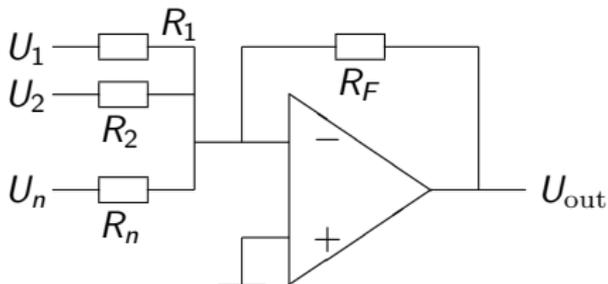


Ein solcher (idealisierter) Operationsverstärker verfügt über zwei Eingänge, von denen einer invertiert, und stellt an seinem Ausgang die mit einem Verstärkungsfaktor  $V = \infty$  multiplizierte Summe der beiden Eingangssignale zur Verfügung.

Natürlich ist ein unendlicher Verstärkungsfaktor nicht realisierbar – um die folgenden Ausführungen jedoch einfach zu halten, wird im folgenden stets von idealen Verstärkern ausgegangen, die zudem keine Drift und keinen Eingangsstrom aufweisen.

Mit Hilfe von Operationsverstärkern können auf einfache Art und Weise Summierer und Integrierer aufgebaut werden, welche zu den wichtigsten Rechenelementen überhaupt zählen.

Ein solcher (invertierender) Summierer hat folgende Prinzipschaltung:



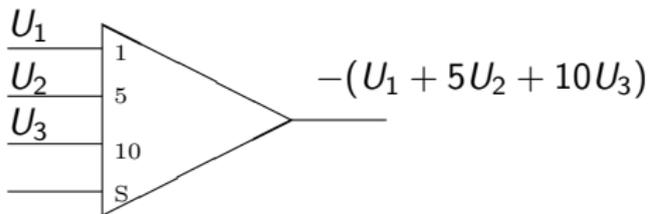
Für einen idealen Operationsverstärker gilt hiermit

$$\sum_{i=1}^n \frac{U_i}{R_i} = -\frac{U_{out}}{R_F}.$$

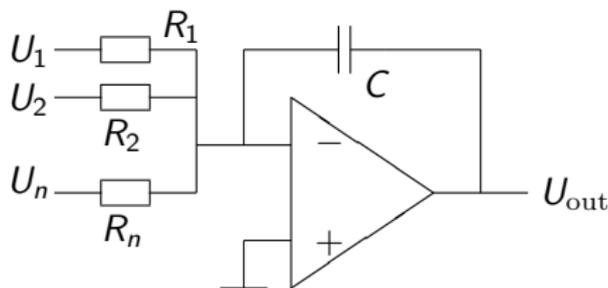
Insgesamt berechnet ein solcher Summierer also

$$-U_{\text{out}} = \sum_{i=1}^n a_i U_i$$

mit  $a_i = \frac{R_F}{R_i}$ . In der Regel wird er durch folgendes Symbol dargestellt:



Die ebenfalls auf einem (idealisierten) Operationsverstärker als aktivem Element beruhende Prinzipschaltung eines Integrierers zeigt folgende Abbildung:



Im Unterschied zu einem Summierer liegt hier eine Kapazität im Rückführungszweig des Verstärkers.

Aus dem durch den Kondensator fließenden Strom  $I_C = C\dot{U}_{\text{out}}$  folgt entsprechend

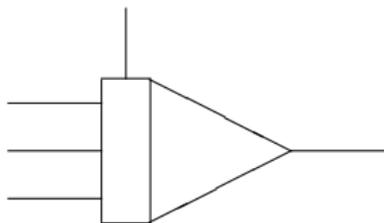
$$\sum_{i=1}^n \frac{U_i}{R_i} = -C\dot{U}_{\text{out}}.$$

Mit  $a_i$  wie zuvor ergibt sich damit letztlich

$$-U_{\text{out}} = \int_0^t \sum_{i=1}^n a_i U_i(t) dt + U(0)$$

für die Operation des Integrierers. Man beachte, dass nur nach der Zeit  $t$  integriert werden kann!

Ein solcher Integrierer wird in der Regel durch folgendes Symbol dargestellt:



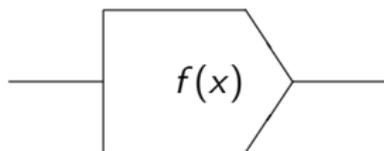
Er verfügt über mehrere Betriebsarten (vereinfacht):

**Anfangswert:** Der Kondensator wird auf einen Anfangswert  $U(0)$  geladen.

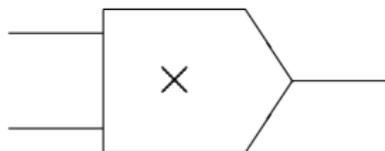
**Rechnen:** Die Integration läuft.

**Halt:** Die Integration ist unterbrochen, am Ausgang des Integrierers steht konstant der letzte Ausgangswert zur Verfügung.

Viele Aufgabenstellungen erfordern die Darstellung vorgegebener Funktionen. Hierfür stehen sogenannte *Funktionsgeber* zur Verfügung, für deren Implementation eine Vielzahl unterschiedlicher Varianten entwickelt wurden. Das allgemeine Symbol eines solchen Funktionsgebers hat folgende Form:

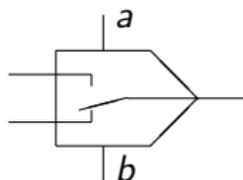


Die Multiplikation zweier Spannungen ist eine deutlich aufwändigere Operation als beispielsweise die Integration. Im Laufe der Zeit wurden eine Vielzahl unterschiedlicher Techniken zur Durchführung von Multiplikationen vorgeschlagen und umgesetzt – das allgemeine Symbol eines Multiplizierers besitzt jedoch in fast allen Fällen folgende Gestalt:



Mit Hilfe eines Tricks lassen sich mit einem Multiplizierer unter Zuhilfenahme eines sogenannten *offenen Verstärkers* auch Divisionen und Wurzeln berechnen, so dass hierfür in der Regel keine eigenen Rechenelemente zur Verfügung stehen.

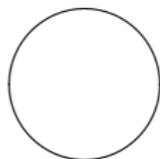
Einige Problemstellungen benötigen zu ihrer Behandlung Sprungfunktionen oder eine Möglichkeit, zwischen zwei Werten auswählen zu können etc. Hierzu bieten die meisten elektronischen Analogrechner sogenannte *Komparatoren* an, bei denen es sich um gesteuerte Schalter handelt. Das Symbol eines solchen Komparators zeigt folgende Abbildung:



Ist die Summe der beiden Eingangsspannungen  $a$  und  $b$  größer 0, so nimmt der Schalter die obere Position ein, ist sie kleiner gleich 0, ist die untere Position aktiv.

# Masse-Feder-Dämpfer-System

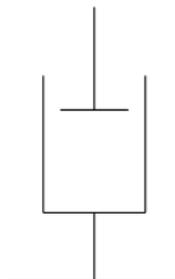
Als erstes Beispiel soll ein einfaches Masse-Feder-Dämpfer-System, bestehend aus folgenden Komponenten, simuliert werden:



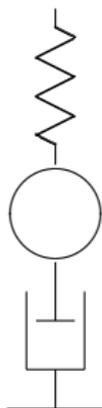
$$F_m = ma = m\ddot{y}$$



$$F_s = sy$$



$$F_d = d\dot{y}$$



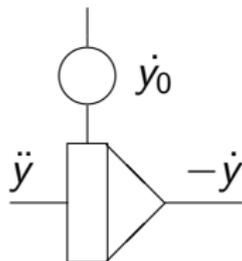
$$F_m + F_d + F_s = 0$$

$$m\ddot{y} + d\dot{y} + sy = 0$$

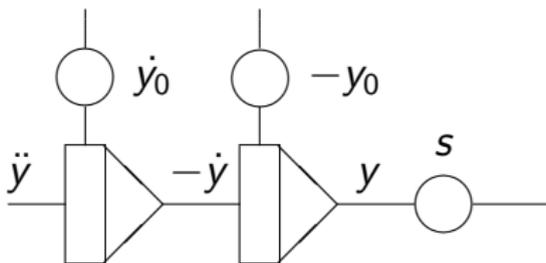
Es gilt also

$$\ddot{y} = \frac{-(d\dot{y} + sy)}{m}.$$

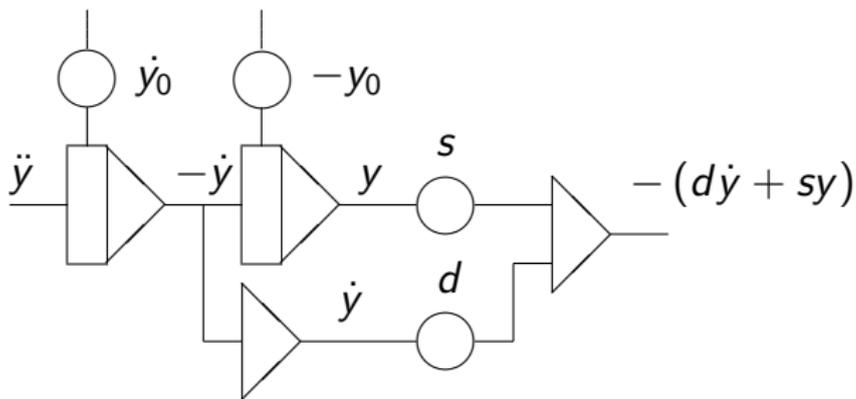
Eine erste Integration liefert:

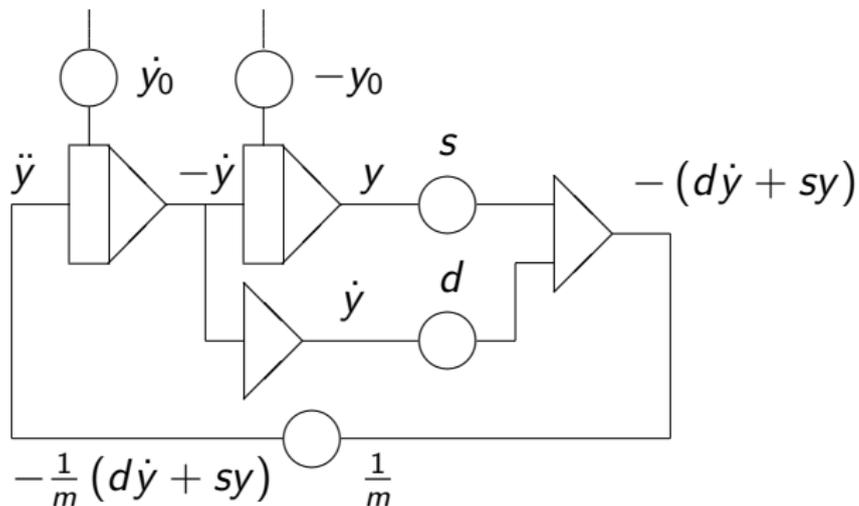


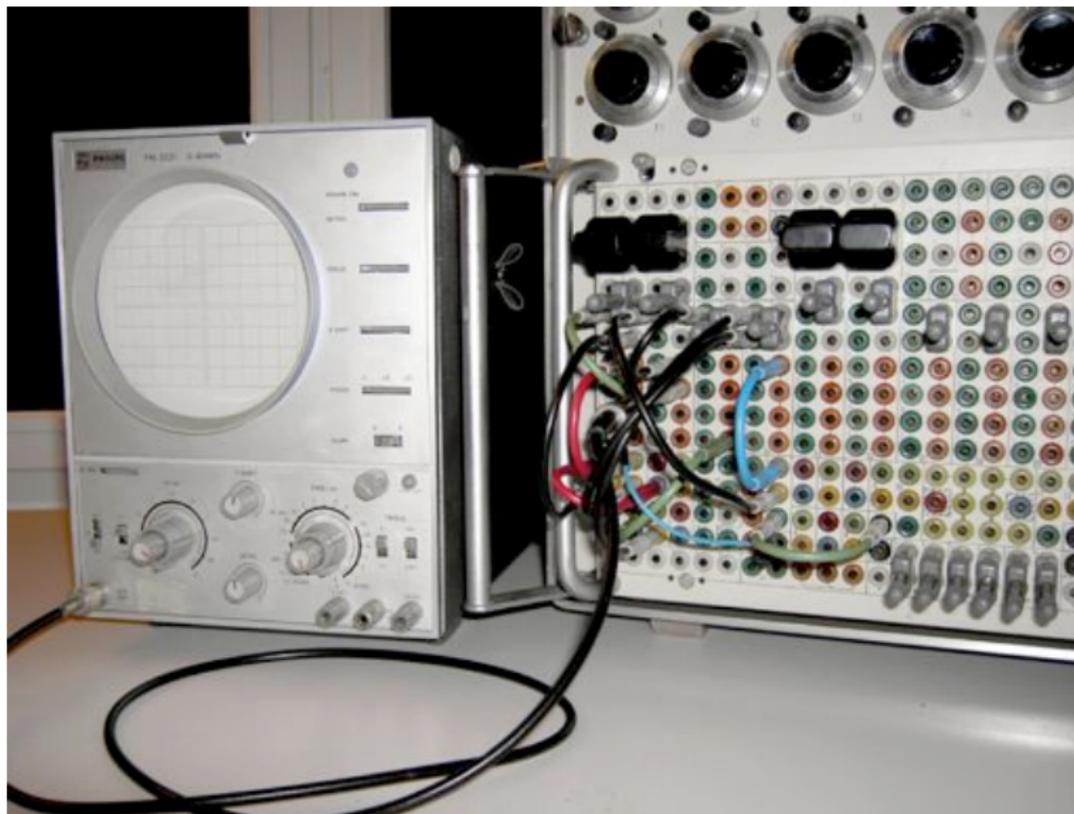
Nach einer weiteren Integration ergibt sich:



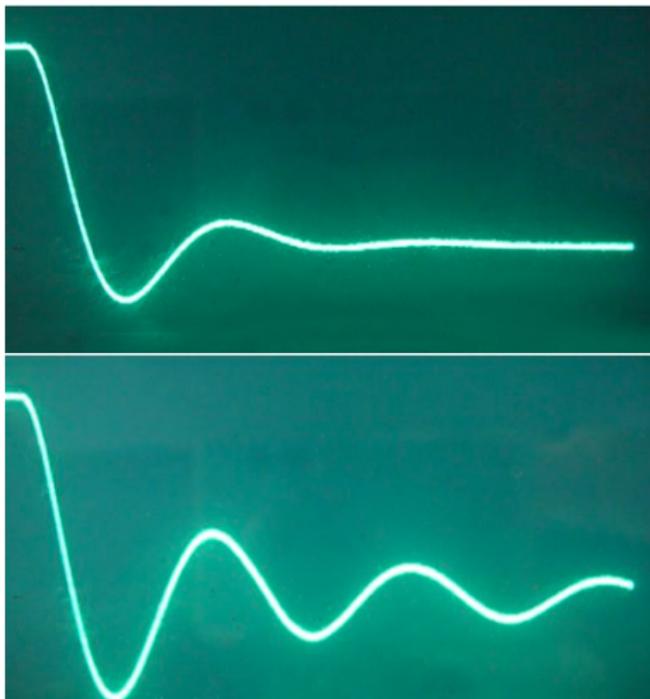
Bilden des Ausdruckes  $-(d\dot{y} + sy)$ :







Die beiden folgenden Bilder zeigen Simulationsergebnisse für unterschiedliche Dämpfungswerte:



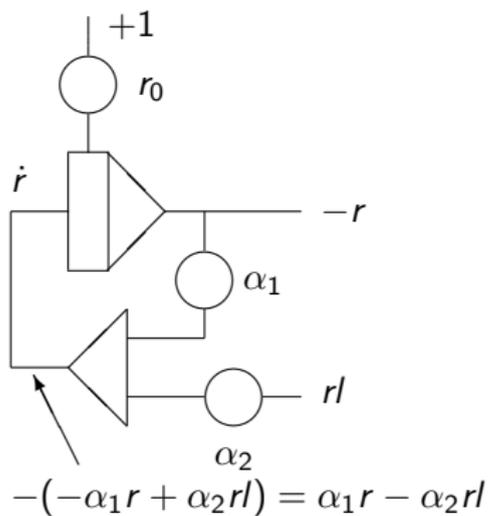
# Räuber-Beute-System

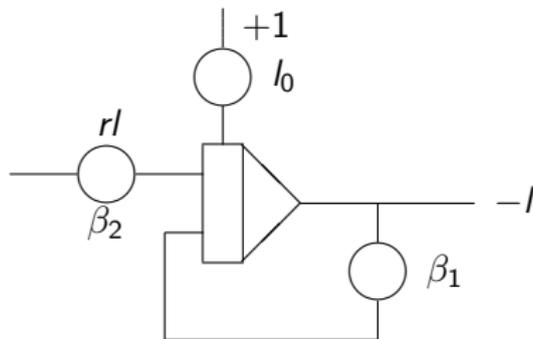
Das als zweites Beispiel dargestellte Räuber-Beute-System ist etwas komplexer als das Masse-Feder-Dämpfer-System, da hier erstmals zwei gekoppelte Differentialgleichungen auftreten.

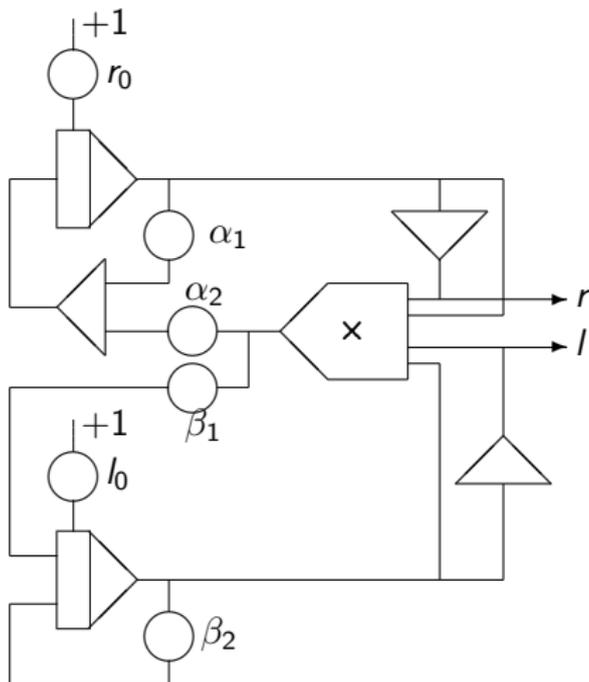
Betrachtet wird ein abgeschlossenes Ökosystem, bestehend aus einer Luchs- ( $l$ ) und einer Hasenpopulation ( $r$ ), für das

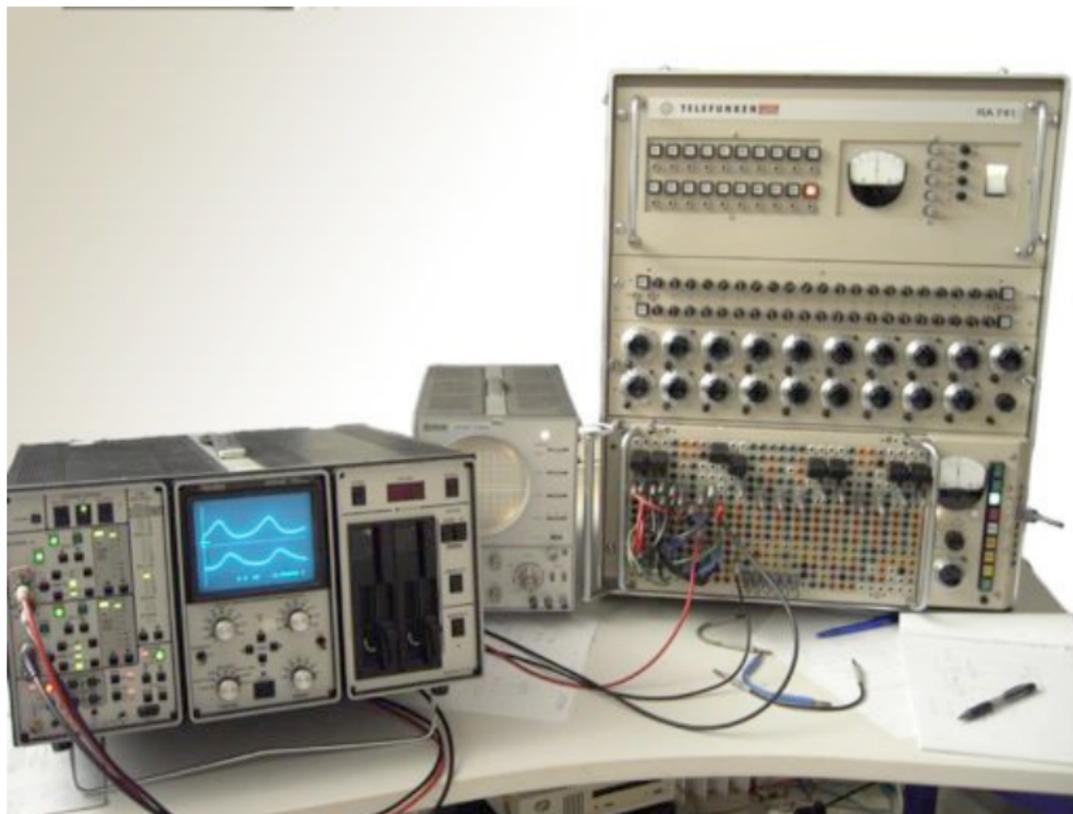
$$\begin{aligned}\dot{r} &= \alpha_1 r - \alpha_2 r l \\ \dot{l} &= -\beta_1 l + \beta_2 r l\end{aligned}$$

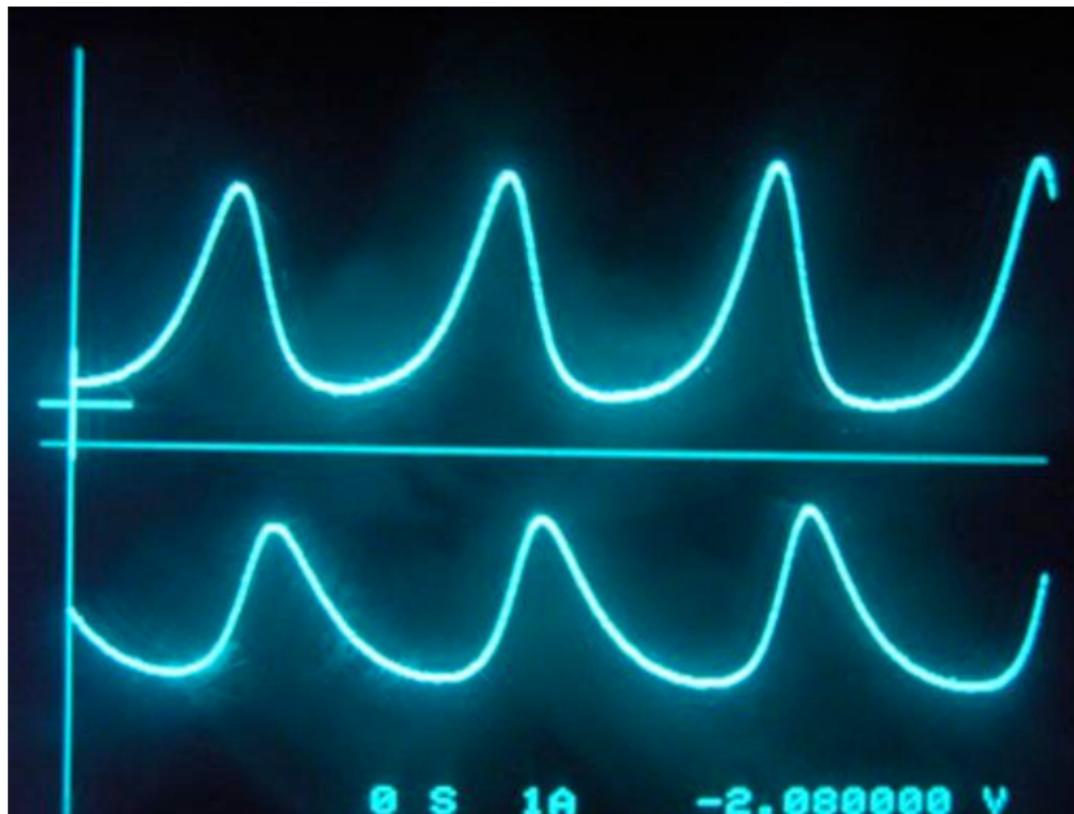
gilt. Hierbei ist  $\alpha_1$  die Geburtsrate der Hasen,  $\alpha_2$  die Rate der von Luchsen getöteten Hasen,  $\beta_1$  die Sterblichkeitsrate der Luchse und  $\beta_2$  der durch Futterzufuhr erzeugte Zuwachs der Luchspopulation.











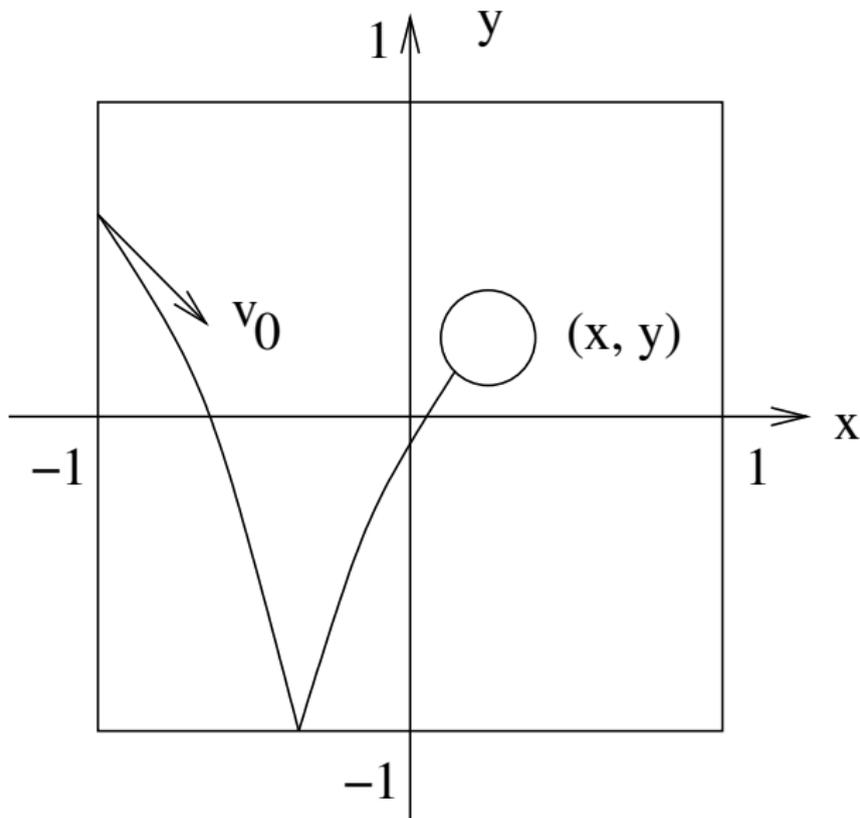
# Ball in Kiste

Ein in einer (gedachten) rechteckigen Kiste springender Ball soll mit Hilfe eines Analogrechners simuliert werden. Hierzu bietet sich folgendes Vorgehen an:<sup>8</sup>

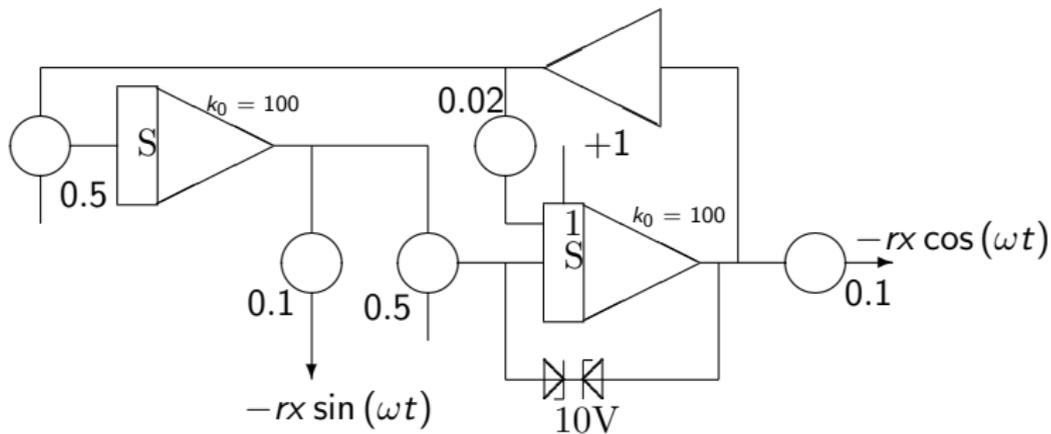
- Bewegung des Balles in  $x$ - und  $y$ -Richtung getrennt simulieren.
- Erzeugung eines  $\sin()/\cos()$ -Signalpaares für die Darstellung des Balles.
- Überlagerung der Ballfigur mit den generierten Koordinatenwerten.

---

<sup>8</sup>Idee: [Telefunken].



Dieser Teil ist am einfachsten – durch Lösen von  $\ddot{y} = -y$  entsteht ein Quadratursignalpaar, das einen Einheitskreis beschreibt:

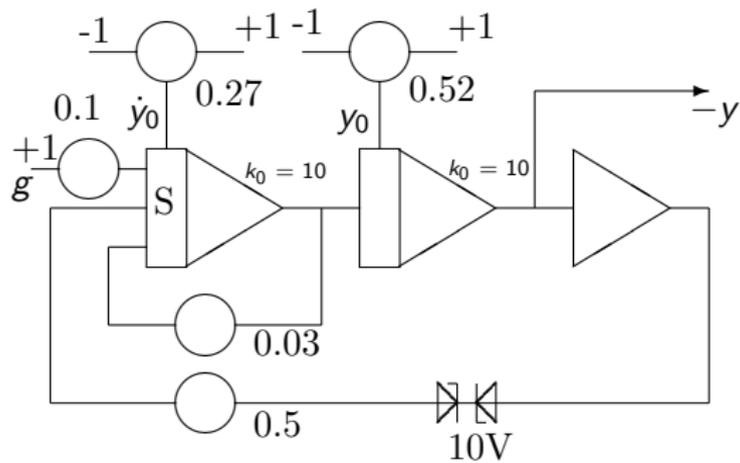


$\ddot{y}$  setzt sich aus drei Termen zusammen; der Erdbeschleunigung  $g$ , einer zu  $\dot{y}$  proportionalen Dämpfung durch Luftreibung sowie dem elastischen Stoß, wenn der Ball den Boden (oder die Decke) des Kastens trifft:

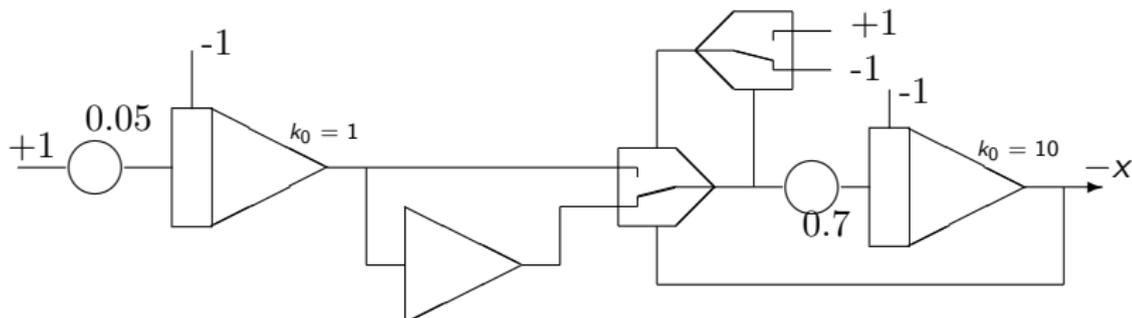
$$\ddot{y} = -g + d\dot{y} \begin{cases} +\frac{c}{m} (|y| + 1) & \text{if } y < -1 \\ -\frac{c}{m} (y - 1) & \text{if } y > 1 \end{cases}$$

Zweifache Integration liefert dann die gesuchte  $y$ -Koordinate, wobei  $y_0$  der Anfangshöhe des Balles entspricht:

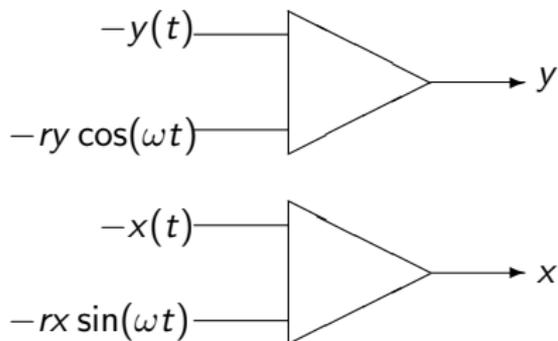
$$y = \iint_0^T \ddot{y} dt + y_0$$

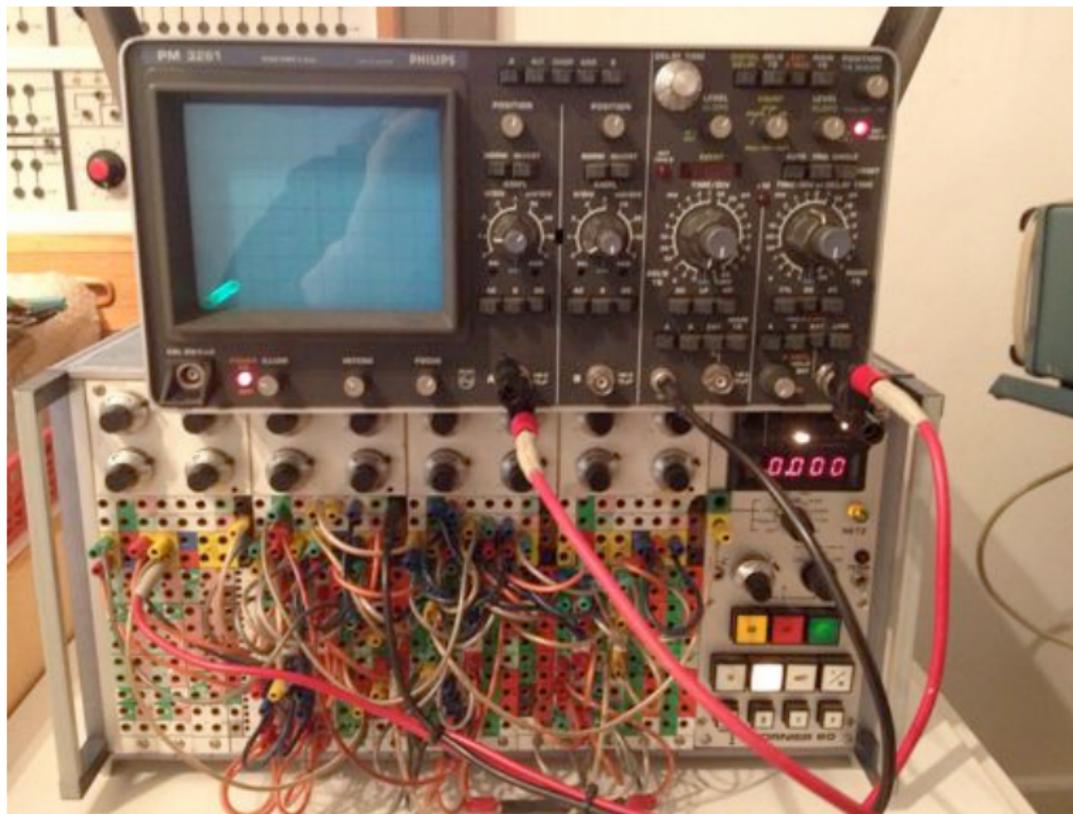


Bei der Erzeugung der  $x$ -Koordinate wird davon ausgegangen, dass die Geschwindigkeit linear mit der Zeit  $t$  abnimmt, bis sie 0 erreicht. Die Richtungsumkehr bei Erreichen einer Wand wird mit Hilfe zweier Komparatoren implementiert:



Die solchermaßen generierten  $x$ - und  $y$ -Koordinatenwerte müssen nun noch mit dem Quadratursignalpärchen überlagert werden, um das Abbild eines springenden Balles zu erzeugen:





# Strömungssimulation

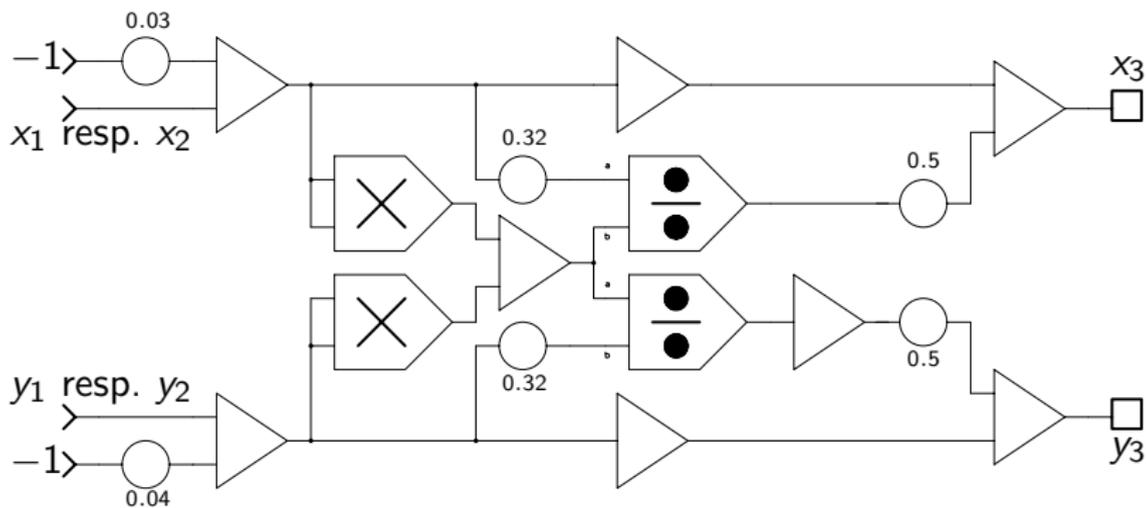
Mit Hilfe einer konformen Abbildung der Form

$$w(z) = (z - z_0) + \frac{\lambda^2}{z - z_0}$$

kann ein Einheitskreis in ein sogenanntes JOUKOWSKI-Profil verwandelt werden. Nach Aufspaltung dieser Gleichung in einen Real- und einen Imaginärteil ergibt sich Folgendes:

$$u(x(t), y(t)) = (x(t) - x_0(t)) + \frac{\lambda^2(x(t) - x_0(t))}{(x(t) - x_0(t))^2 + (y(t) - y_0(t))^2}$$
$$v(x(t), y(t)) = (y(t) - y_0(t)) - \frac{\lambda^2(y(t) - y_0(t))}{(x(t) - x_0(t))^2 + (y(t) - y_0(t))^2}$$

Die Rechenschaltung für die konforme Abbildung hat folgende Gestalt:



Der Weg eines Stromfadens um einen rotierenden Zylinder lässt sich durch ein komplexes Geschwindigkeitspotenzial der folgenden Form beschreiben:

$$f(z) = v_0 \left( e^{i\delta} z + \frac{e^{-i\delta} r^2}{z} \right) - i \frac{\Gamma}{2\pi} \ln(z),$$

Die Umsetzung dieser Funktion in eine Rechenschaltung zeigt folgendes Bild:

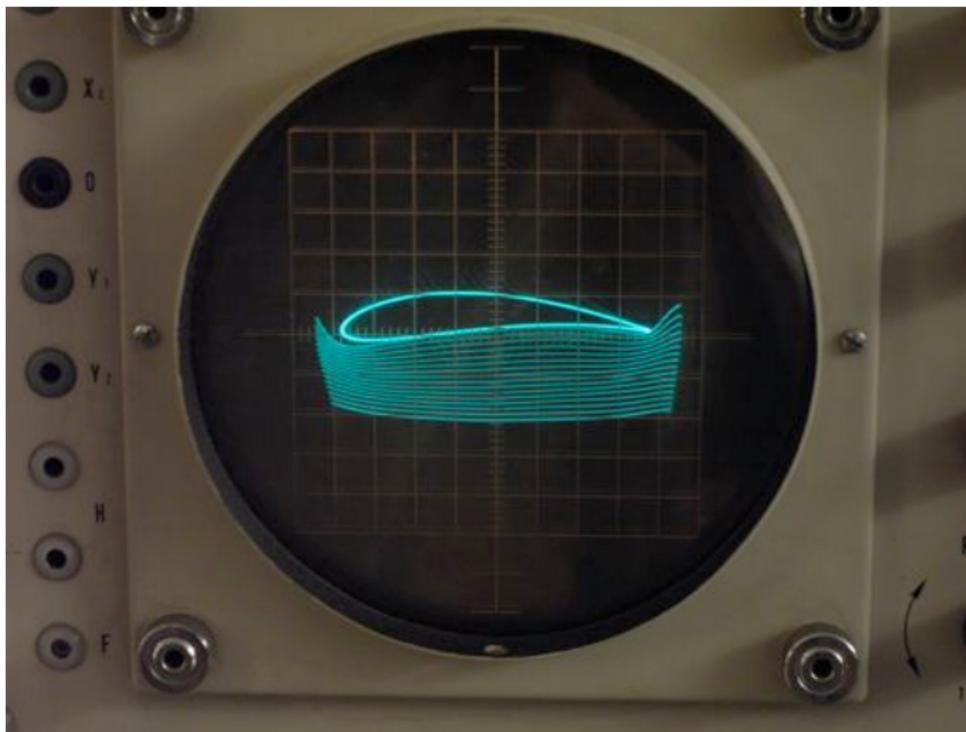


Generiert man nun durch Lösen von  $\ddot{y} = -y$  ein Sinus-/Cosinus-Signalpaar, mit dessen Hilfe ein Einheitskreis dargestellt werden kann, so dient dieses einerseits als Eingang für die JOUKOWSKI-Transformation zur Darstellung der Tragfläche.

Andererseits dient dieses Signalpaar als Eingang für die Berechnung des Weges eines Stromteilchens um einen rotierenden Zylinder. Das Ergebnis dieser Berechnung wird in der Folge ebenfalls einer JOUKOWSKI-Transformation unterworfen.

Mit Hilfe einiger elektronischer Schalter kann der Schaltungsteil, mit dem die konforme Abbildung implementiert ist, schnell zwischen beiden Aufgaben hin und her geschaltet werden, so dass nur **eine** Implementation notwendig ist.

Bei hinreichend kleinen Zeitkonstanten der Integrierer und der eben erwähnten Umschaltung sowie einer automatischen Erzeugung von Startkoordinaten für einen Stromfaden ergibt sich folgende Ansicht:



# DDAs

All das (und noch mehr, weil nicht mehr nur die Zeit als freie Variable der Integration zur Verfügung steht) geht nicht nur mit analogelektronischen Analogrechnern, sondern auch mit ihren digitalen Verwandten, den sogenannten *DDAs*, kurz für *Digital Differential Analyzers*.

Die Grundidee ist einfach und naheliegend: Alle Rechenelemente werden mit Hilfe digitaler Schaltungen implementiert. Historisch gesehen gibt es zwei Varianten hiervon:

**Serielle DDAs** : Diese Form eines DDAs verfügt nur über eine einzige *ALU* (kurz für *Arithmetic Logic Unit*), die quasi im Timesharing nacheinander alle Rechenschritte der Rechenelemente einer Rechnung durchführt.

**Parallele DDAs** : Solche DDAs sind, wie analogelektronische Analogrechner, hochgradig parallel, was jedoch einen immensen Hardwareaufwand nach sich zieht.

Bereits 1949 begann die Entwicklung von *MADDIDA*, dem *Magnetic Drum Differential Analyzer* bei Northrop (serieller DDA):<sup>9</sup>



<sup>9</sup> Picture: Navy Electronics Laboratory, File Number E1278.

Ein weiterer serieller DDA, die *D-12*, wurde in den frühen 1950er Jahren von Bendix entwickelt:<sup>10</sup>



<sup>10</sup> Siehe [Bendix][p. 2].

Ein paralleler DDA war *TRICE*, in den 1960ern von Packard Bell entwickelt:<sup>11</sup>

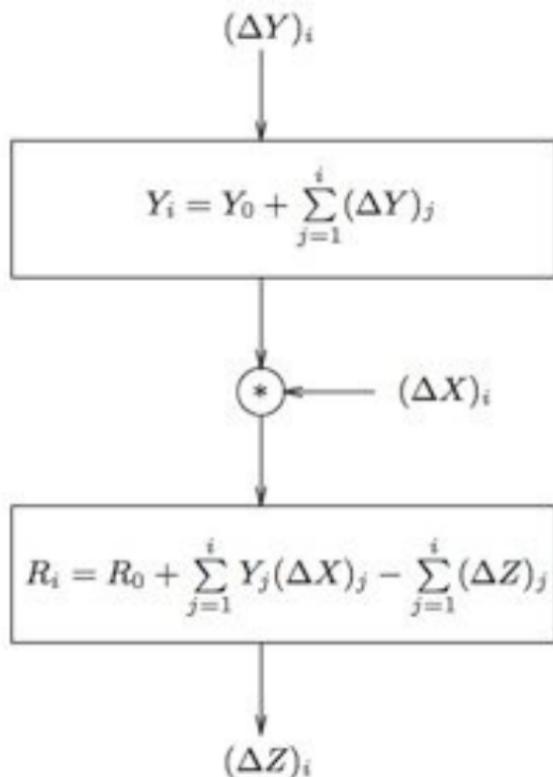


<sup>11</sup> Siehe [AMELING 1963][p. 30].

Solche DDAs arbeiteten bitseriell (heutige Implementationen auf Basis von FPGAs wären von solchen, der Hardwarekomplexität geschuldeten Einschränkungen frei). Zentrales Element ist der Integrierer, auf den fast alle anderen Rechenelemente zurückgeführt werden.

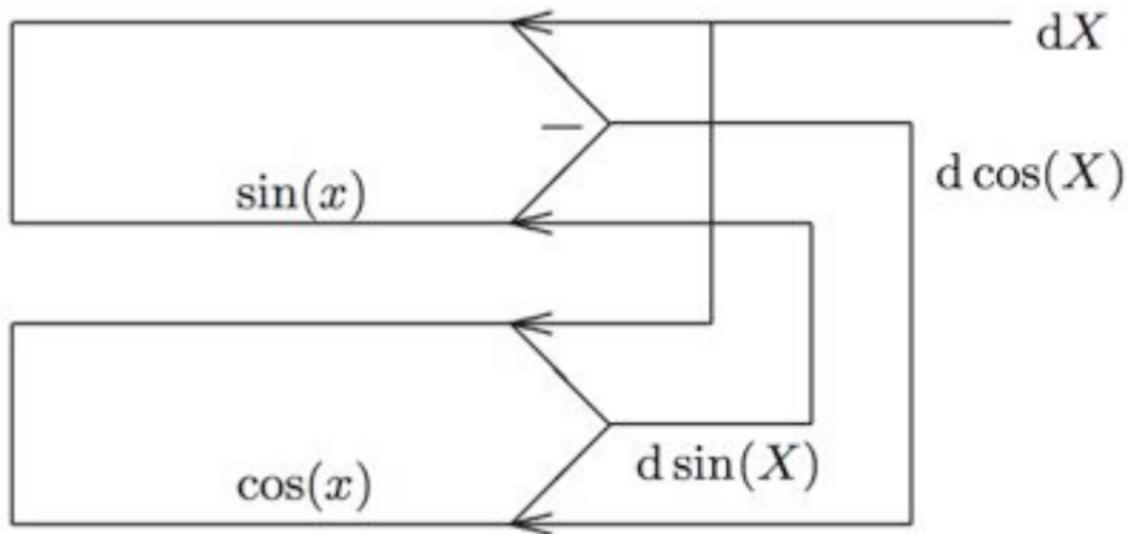
Bei bitserieller Arbeitsweise werden sogenannte *Inkments* wie  $\Delta X$  zwischen den Rechenelementen übertragen, welche die Werte  $-1$ ,  $0$  und  $+1$  annehmen können.

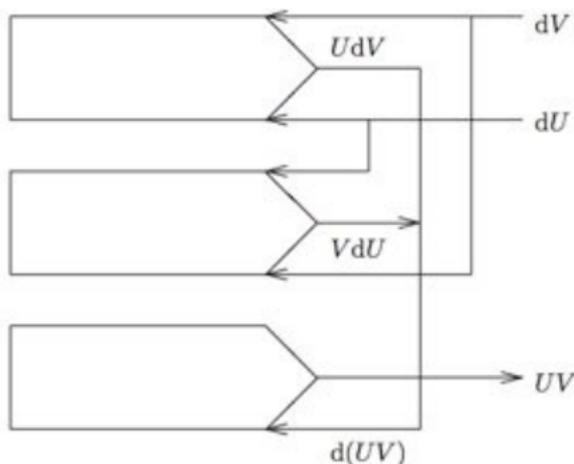
Integrierer, Summierer etc. bestehen im Wesentlichen aus Zählern:



Ein typischer Integrierer besteht aus zwei Zählern, wie links dargestellt. Der obere summiert über  $\Delta Y$ , während der untere, gesteuert durch  $\Delta X$ , welches das  $dx$  der Integration repräsentiert, über die Werte  $Y$  integriert. Ausgangswert ist wieder ein Inkrement, das durch Über-/Unterlauf des unteren Zählers generiert wird.

Lösen der DGL  $\ddot{y} = -y$  auf einem DDA:





Da dank  $\Delta X$  beliebige Integrationsvariablen, und nicht nur  $t$ , genutzt werden können, lässt sich auch die Multiplikation auf zwei Integrationen zurück führen:

Die Grundidee hierbei ist Folgende:

$$UV = \int_{U_0}^{U_1} V dU + \int_{V_0}^{V_1} U dV + U_0 V_0$$

Entsprechend können auch partielle DGLs mit einem DDA elegant gelöst werden.

# Zukunft

Die Idee des Analogrechnens – sowohl des analogelektronischen als auch des digitalen – besitzt eine Reihe von Eigenschaften, die sie auch und gerade für zukünftige Entwicklungen hochinteressant werden lassen:

- Extrem hoher Grad an Parallelität innerhalb einer Rechnung möglich – denkbar sind hier DDA-Coprozessoren auf Basis von FPGAs im Zusammenspiel mit herkömmlichen VON NEUMANN-Architekturen.
- Analogelektronische Implementationen sind hinsichtlich ihrer Rechengenauigkeit zwar recht eingeschränkt, bieten aber (extrem) hohe Rechenleistung bei sehr geringer Leistungsaufnahme. Auch hiermit sind Coprozessoren denkbar (und wurden bereits umgesetzt), mit deren Hilfe beispielsweise Startwerte für nachgelagerte numerische Verfahren generiert werden können.

Welche konkreten Vorteile bietet die Implementation eines DDA auf Basis von FPGAs (zum Teil verglichen mit analogelektronischen Analogrechnern):

- Extreme Parallelität, keine explizite Synchronisation (Semaphore o.ä.) notwendig
- Beliebige Rechengenauigkeit möglich
- Keine Drift- und Alterungseffekte
- Keine Signalwandlung zwischen Hostrechner und DDA notwendig
- Passt gut in das gegenwärtig dominierende Paradigma des speicherprogrammierten Rechnens

Einige praktische Beispiele, z.B. ein Masse-Feder-Dämpfer-System, finden sich unter

<https://courses.cit.cornell.edu/ece576/DDA/>.

```

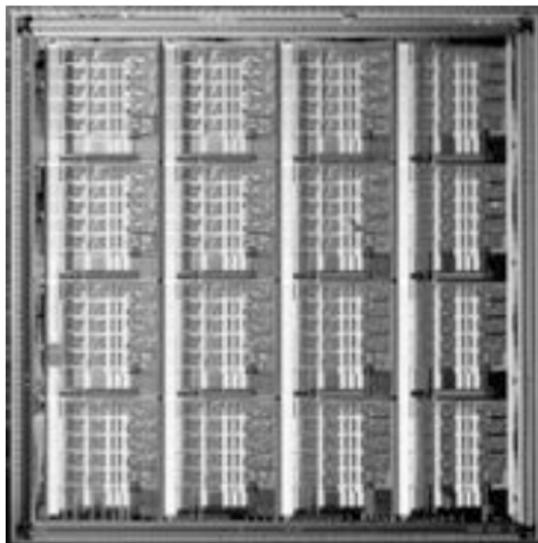
reg signed [17:0] v1, v2 ;
wire signed [17:0] v1new, v2new ;
wire signed [17:0] v1xK_M, v2xD_M ; //signed mult output
reg [4:0] count; // the clock divider
    always @ (posedge CLOCK_50) //Update state variables of simulation of spring- mass
    begin
        count <= count + 1;
        if (KEY[3]==0) //reset
            begin
                v1 <= 32'h10000 ; //
                v2 <= 32'h00000 ;

            end
        else if (count==0)
            begin
                v1 <= v1new ;
                v2 <= v2new ;

            end
        end
    // Compute new F(t,v) with dt = 2>>9, v1(n+1) = v1(n) + dt*v2(n)
    assign v1new = v1 + (v2>>>9);
    // v2(n+1) = v2(n) + dt*(-k/m*v1(n) - d/m*v2(n))
    signed_mult K_M(v1xK_M, v1, 18'h10000);
    signed_mult D_M(v2xD_M, v2, 18'h00800);
    assign v2new = v2 - ((v1xK_M + v2xD_M)>>>9);
module signed_mult (out, a, b);
    output [17:0] out;
    input signed [17:0] a;
    input signed [17:0] b;
    wire signed [17:0] out;
    wire signed [35:0] mult_out;
    assign mult_out = a * b;
    assign out = {mult_out[35], mult_out[32:16]};
endmodule

```

2005 entwickelte GLENN EDWARD RUSSEL COWAN einen analogelektronischen *VLSI* Analogrechner (kurz für *Very Large Scale Integration*) für den Einsatz als Co-Prozessor für herkömmliche speicherprogrammierte Rechner:<sup>12</sup>



<sup>12</sup>See [COWAN 2005].

Der Chip weist eine Fläche von  $100 \text{ mm}^2$  auf, enthält 416 Rechenelemente mit der zugehörigen Verbindungselektronik und wurde zur Lösung gewöhnlicher, partieller und stochastischer Differentialgleichungen eingesetzt.<sup>13</sup>

In [COWAN et al. 2009][pp. 13 f.] werden einige Ansätze zur Lösung solcher stochastischer Differentialgleichungen hinsichtlich der benötigten Energie betrachtet.

Verglichen wurden eine traditionelle Implementation auf einem typischen Mikroprozessor, eine Lösung auf einem *DSP* (kurz für *Digital Signal Processor*) sowie eine Lösung mit Hilfe des VLSI Analogrechners von COWAN:

---

<sup>13</sup>Siehe [COWAN 2005][abstract].

Die Resultate sind beeindruckend:

- Der traditionelle Lösungsansatz auf einem Mikroprozessor erforderte etwa 400 MFLOPS, was einer Leistungsaufnahme von 1.2 bis 40 J entspricht.
- Der DSP benötigt hierfür zwischen 0.04 und 0.4 J.
- Der VLSI Analogrechner benötigt für die gleiche Aufgabe lediglich 0.008 J!
- Die VLSI-Implementation erzielt etwa 6.4 GFLOPS (bei zugegebenermaßen eingeschränkter Genauigkeit) bei einer Leistungsaufnahme von lediglich 300 mW!

Bezüglich *Performance per Watt* findet sich folgende Aussage:<sup>14</sup>

*„As of June 2012, the Green500 list rates BlueGene/Q, Power BQC 16C as the most efficient supercomputer on the TOP500 in terms of FLOPS per watt, running at 2,100.88 MFLOPS/watt.“*

COWANS VLSI Analogrechner liegt mit

21 GFLOPS/Watt

um einen Faktor 10 vorne, verglichen mit einem Rechner (BlueGene/Q), der sieben Jahre später gebaut wurde!

---

<sup>14</sup>See [http://en.wikipedia.org/wiki/Performance\\_per\\_watt](http://en.wikipedia.org/wiki/Performance_per_watt), retrieved 11.06.2013.

In der Zukunft sollten/könnten die folgenden Punkte angegangen werden:

- Implementation eines sequentiell arbeitenden DDAs auf einem FPGA als einfacher und preiswerter Coprozessor. Zu untersuchen ist, ob vielleicht auf Gleitkommazahlen verzichtet und stattdessen mit (extrem) langen Integers vorteilhaft gerechnet werden kann (siehe die Arbeiten KULISCHS). Die Programmierung würde, wie bei MADDIDA, über Konfigurationsdaten in einem RAM-Bereich des FPGA abgelegt werden, so dass im Betrieb keine Rekonfiguration vorzunehmen ist.
- Implementation eines Frameworks, um parallele DDAs mit Hilfe von FPGAs zu implementieren. Eine der Hauptschwierigkeiten hierbei sind die proprietären Entwicklungsumgebungen der Hersteller, die einer automatisierten Codegenerierung entgegen stehen.
- Entwicklung eines hochintegrierten analogelektronischen Analogrechners auf der Basis der Ergebnisse COWANS.

Wo lassen sich weitere Informationen finden?

- <http://www.analogmuseum.org>
- Einer der besten historischen Texte über das Analogrechnen ist [GILOI et al. 1963].
- Zwei moderne Texte hierzu sind [ULMANN 2013] (Englisch) bzw. [ULMANN 2010] (Deutsch).
- DDAs werden detailliert in [SHILEIKO 1964] behandelt.

## Vielen Dank für Ihre Aufmerksamkeit

Der Autor kann unter

[ulmann@analogmuseum.org](mailto:ulmann@analogmuseum.org)

erreicht werden.

In eigener Sache: Das Analogrechnermuseum sucht stets Analogrechner, Artefakte, Dokumentationen etc. für die Sammlung. Falls Sie von Rechnern und anderen Dingen wissen, die ein gutes neues Zuhause suchen, lassen Sie es bitte den Autor wissen.

# Bibliographie



[AMELING 1963] Walter Ameling, „Aufbau und Arbeitsweise des Hybrid-Rechners TRICE“, in *Elektronische Rechenanlagen*, 5 (1963), Heft 1



[Bendix] Bendix Computer, „Digital Differential Analyzer D-12“



[BORD44] Bureau of Ordnance Publication (Hrsg.), *Torpedo Data Computer, Mark 3, Mods. 5 to 12 inclusive*, June, 1944



[BORD46] Bureau of Ordnance (Hrsg.), *Basic Fire Control Mechanisms: Maintenance*, OP 1140 A, 1946



[COWAN 2005] GLENN EDWARD RUSSELL COWAN, *A VLSI Analog Computer / Math Co-processor for a Digital Computer*, Columbia University, 2005



[COWAN et al. 2009] GLENN EDWARD RUSSELL COWAN, Y. TSIVIDIS, *Analog and Digital Continuous-Time Computation and Signal Processing*, CMOSSET 2009

-  [DRESSLER 1972] Fritz Dressler, „Das Dach“ in *hobby – Das Magazin der Technik*, Nr. 8/72, pp. 50
-  [GERWIN 1964] Robert Gerwin, „Intelligente Automaten“, Chr. Belser Verlag, 1964
-  [GILOI et al. 1963] WOLFGANG GILOI, RUDOLF LAUBER, *Analogrechnen*, Springer-Verlag, 1963
-  [HOELZER 1946] HELMUT HOELZER, *Anwendung elektrischer Netzwerke zur Lösung von Differentialgleichungen*, Dissertation TH Darmstadt, 1946
-  [KARPLUS et al. 1958] Walter J. Karplus, Walter W. Soroka, *Analog Methods – Computation and Simulation*, McGraw-Hill Book Company, Inc., 1958



[McCARTHY 2009] Jerry McCarthy, „Der Mechanismus von Antikythera“, in Tagungsband, 15. Internationales Treffen der Rechenschiebersammler und 4. Symposium zur Entwicklung der Rechentechnik, Universitaet Greifswald, 2009, pp. 55 – 64



[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



[PHILBRICK 1948] GEORGE A. PHILBRICK, “Designing Industrial Controllers by Analog”, in *Electronics*, June, 1948, pp. 108–111



[SHILEIKO 1964] A. V. Shileiko, *Digital Differential Analysers*, The Macmillan Company, New York, 1964



[Telefunken] AEG Telefunken, „Demonstrationsbeispiel Nr.5, Ball im Kasten“



[THOMSON 1876] William Thomson, „Mechanical integration of linear differential equations of the second order with variable



[TRUITT et al. 1960] Thos. D. Truitt, A. E. Rogers, *Basics of Analog Computers*, John F. Rider Publisher, Inc., New York, December 1960



[ULMANN 2010] Bernd Ulmann, *Analogrechner, Wunderwerke der Technik – Grundlagen, Geschichte und Anwendung*, Oldenbourg Verlag, 2010



[ULMANN 2013] Bernd Ulmann, *Analog Computing*, Oldenbourg Verlag, 2013