The Power of Analogs

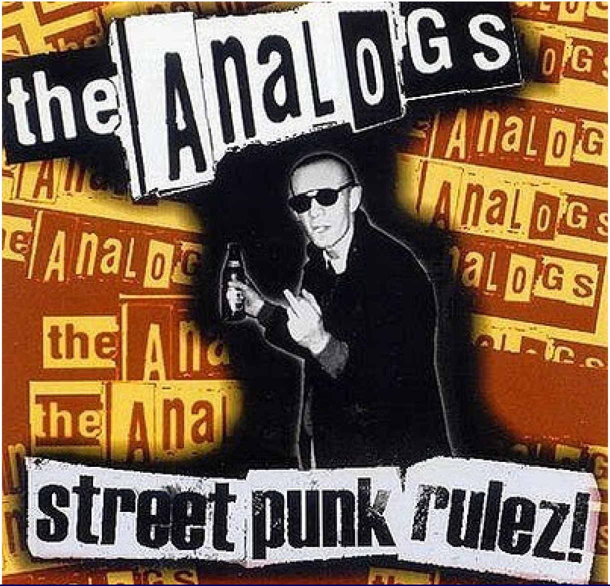
Bernd Ulmann

10-DEC-2010

What it is (not) about?

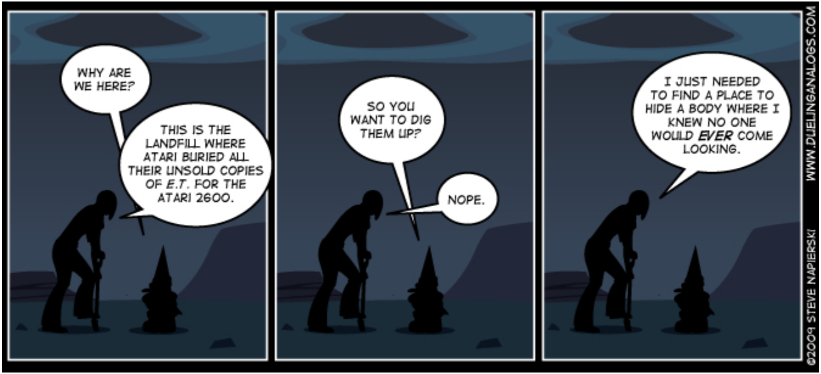
What it is not about?

. . . it's not about "The Analogs", a Polish street punk band. . .



What it is not about?

. . . it's also not about the "Dueling Analogs", a web comic by Steve Napierski[1]:



[1] I would like to thank Steve for his permission to use the above comic strip for this talk.

What it is about, then?

Since we will not talk about Polish street punk or Dueling Analogs (which, by the way, are both rather funny); what, then, is the theme of this talk?

It is about **models** – so called **analogs** – and their (future) use in simulation and computation.

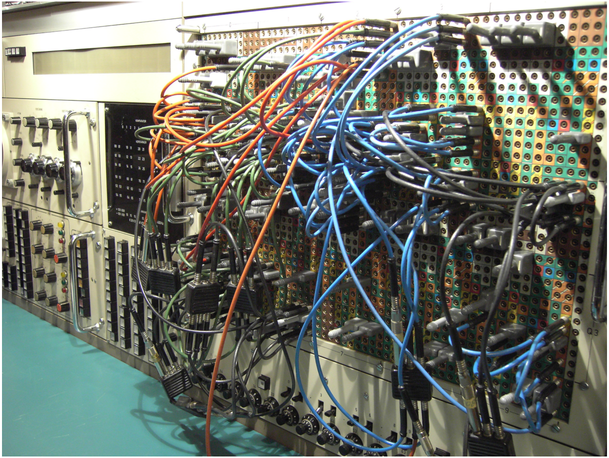
We will distinguish **direct** and **indirect** analogs (like soap bubbles for minimal surfaces and electrical circuits simulating a mechanical system).

The art of **analog computation/simulation** has been largely forgotten (like Atari's ET game mentioned in the Dueling Analogs comic) but this technology has enough potential to rise again to new heights in the future employing state of the art devices like FPGAs etc.

We are not interested in direct analogies although these are very powerful, too – NASA has a program called **Extreme Analogs**:

(Cf. `http://www.nasa.gov/exploration/analogs/`)

What we are interested in are **indirect analogies** like this one (but with more modern technology) showing an analog computer setup to compute airflow around a Joukowski air foil:



To create an indirect analog of a given problem the following steps will be executed:

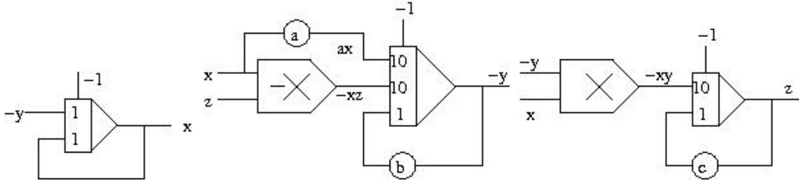
- Create a mathematical description of the problem – this will normally yield (coupled) differential equations.
- Transform these equations into a corresponding computer setup (a circuit consisting of basic elements like adders, integrators, multipliers etc.).
- Setup this circuit on an analog computer thus transforming it into a model of the initial problem.

The resulting model, the **analog** will now behave like the problem described initially. In contrast to a memory programmed digital computer the analog computer changes its structure according to the problem that is to be solved while the digital computer keeps its structure and only the algorithm stored in memory will change.

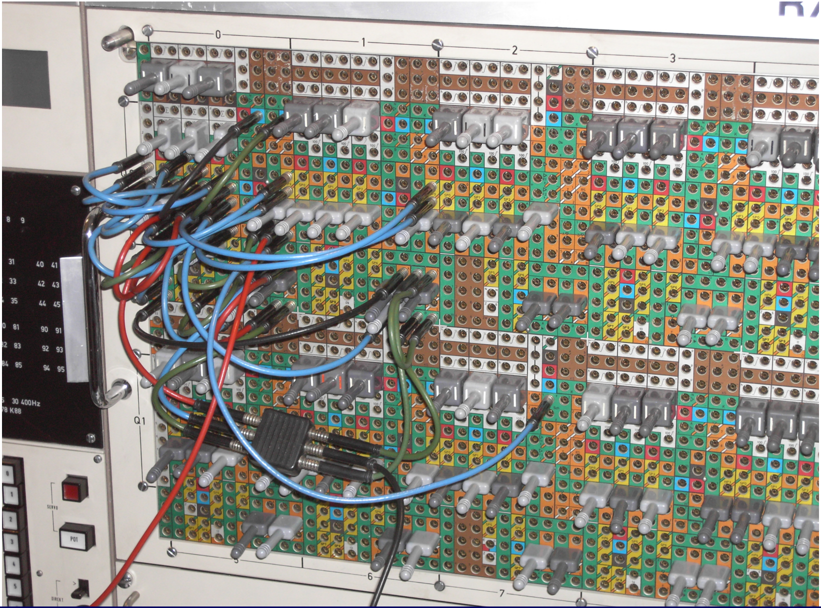
Let us have a look at an example – the Lorenz attractor that is described by the following three DEQs:

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y-x)\\ \frac{dy}{dt} &= x(\rho - z) - y\\ \frac{dz}{dt} &= xy - \beta z\end{aligned}$$

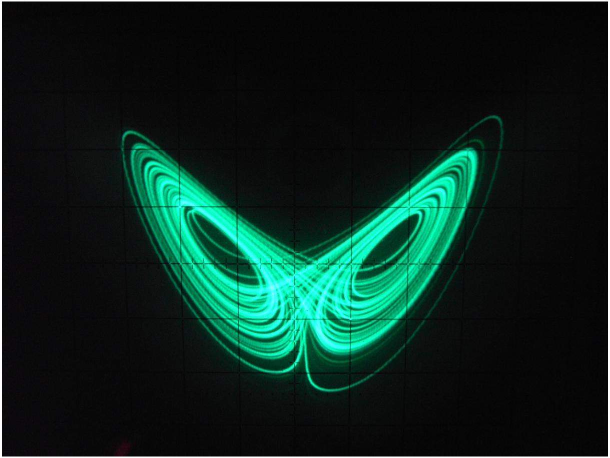
The resulting circuit looks like this:



This yields to the following computer setup:



The result of a computer run:



How did this technology evolve? Why was it forgotten and what might the future bring?

The Early Days

The mechanism of Antikythera is the first mechanical analog computer ever – it dates back to 150 BC:

(Cf. http://upload.wikimedia.org/wikipedia/commons/6/66/NAMA_Machine_d%27Anticythre_1.jpg.)

What could this lump of gears do[2]?

- It could calculate sidereal, synodic, draconic and anomalistic lunar cycles.
- It took into account the first lunar anomaly (the moon's velocity is higher in its perigee compared with its apogee).
- It predicted lunar and solar ecplises and it even predicted if an ecplise would be visible or not (day/night).
- . . . and more. . . (much?)

Isn't that impressive – accomplishing all of that with nothing more than cleverly arranged gears?

For those interested, there is a simulation of this mechanism described in [McCarthy 2009].

[2] Cf. [McCarthy 2009].

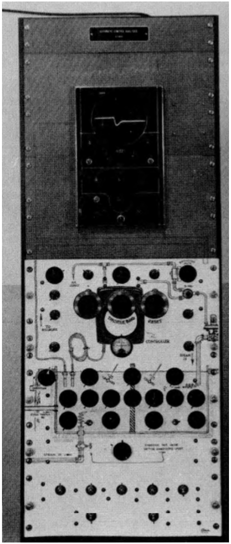
About 2000 years later, in 1876, Lord Kelvin developed the idea of using mechanical analogs to solve differential equations – his **feedback scheme** became the workhorse of the zenith of analog computation in the mid 20th century.

He discovered that a string of integrators in conjunction with other elements could be used to solve DEQs[3]:

> *[...] 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.*

Unfortunately, he did not build a real machine which would have become the world's first mechanical differential analyzer.

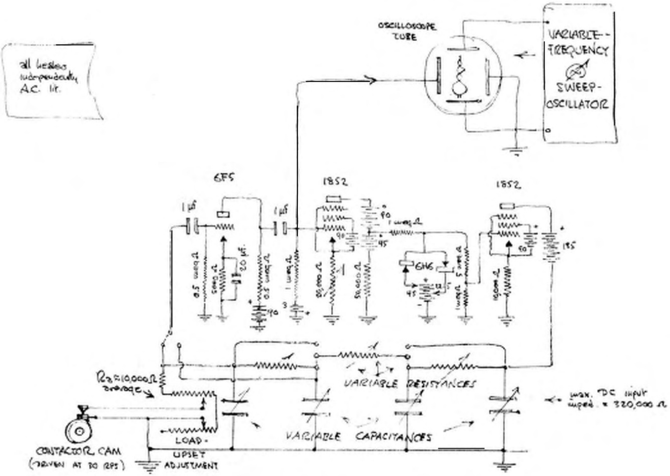
[3] Cf. [Thomson 1876].



Another 62 years later, George A. Philbrick developed **Polyphemus**, the first electronic training simulator.
The picture on the left shows a Polyphemus setup with an illustrated process-control model involving a two-stage liquid bath with steam and cold water inflows[a].

[a] Cf. [Holst 1982][p. 152].

This drawing of Philbrick's Electronic Control Analyzer gives an impression of Polyphemus' inner workings[4]:



[4] Cf. [Holst 1982][p. 151].

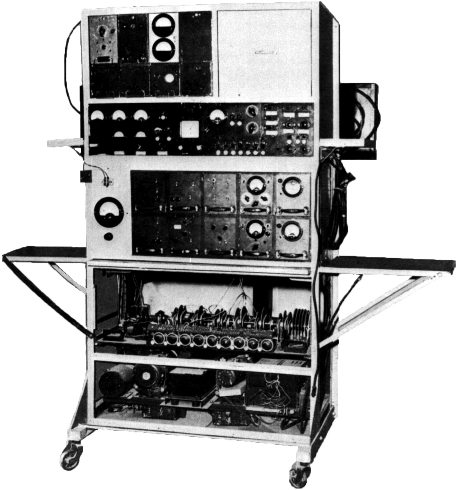
Parallel to Philbrick, Helmut Hoelzer developed another analog computer – in fact he developed two:

- The so called **Mischgeraet**, was the world's first on-board computer and was used in the guidance section of the V2 rocket.
- A truly general purpose electronic analog computer, used by NASA in rocket development at Redstone Arsenal, up through the late 1950s.

The Mischgeraet[5]:

[5] Photo by Adri de Keijzer.

Hoelzer's analog computer as it appeared at the end of WWII[6]:



[6]Source: NASA, Marshall Space Flight Center.

The Zenith

Analog computing gained a lot of interest after WWII especially in areas like high performance air planes and space flight etc.

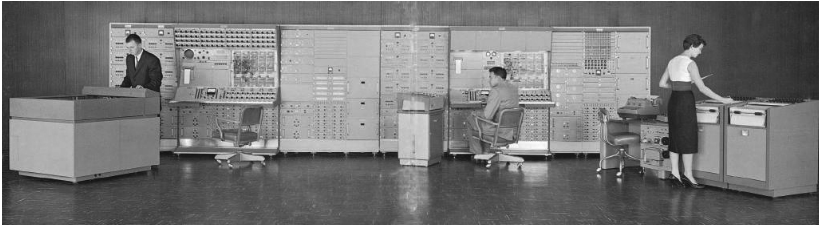
NASA had some of the largest analog computer centers from about 1950 and well into the 70s.

During this time analog computing transformed from an arcane technology into the backbone of our technological culture.

Especially in test pilot circles analog computing and simulation was not taken seriously. After all, what could these guys wearing white lab coats and fiddling with slide rules teach pilots about handling a real aircraft?

A lot...

Using analog computer systems like this[7]



high performance planes like the Bell X2 were successfully simulated. Initially this dismayed the pilots who regarded it is certainly **not** the "Right Stuff". Dick Day remembers[8]:

> *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.*

[7] Cf. [EAI].

[8] Cf. [Waltman 2000][p. 138 f.].

Mel Apt's X2 Flight

In a particular impressive example, it was shown by simulation that the X2 would become unstable at around Mach 3.0 under certain circumstances[9]:

> *We showed [Mel Apt (the test pilot)] if he increased AOA [(angle of attack)] 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[. . .]*

Maybe due to the distrust of pilots concerning analog simulation techniques Mel Apt nevertheless did the maneuver mentioned above when he ran out of fuel and tried to head back to the base. . . Apt did not survive (as predicted). . .

[9] Cf. [Waltman 2000][p. 138 f.]

Mal Apt in the cockpit of the X2:

(Cf. `http://upload.wikimedia.org/wikipedia/commons/2/2f/Captain_Mel_Apt_in_Bell_X-2_1956.jpg`.)

The end of this flight which boosted the trust in analog computing:

(Cf. [Merlin, Moore 2008][p. 20].)

The Zenith

In the following years electronic analog computers reached their zenith – without their help the technological achievements of the 1960s, 1970s and later would have not been possible. These machines were used most prominently in the following areas:

- Flight simulation and space flight
- Optimization of processes
- Process control and simulation
- Chemistry (reaction kinetics, ...)
- Biology and medicine (metabolism research, immunology, neuro science etc.)
- Power plant planning, scheduling, power grid simulations, ...
- Mechanical systems (rail road cars, street cars etc.)
- Military applications (embedded control, simulations, research, ...)

What was the advantage of analog computing that led to this wide spread use?

- Programming an analog computer is close to the problem to be solved and not abstract as traditional algorithmic approaches.
- Analog computers offer an uniquely high amount of interactivity.
- Analog computers are highly parallel machines thus surpassing even modern memory programmed digital computers in terms of parallelism and maybe even in terms of sheer computing power.
- Analog computers do not have a von Neumann-bottleneck since they change their structure according to a problem without needing memory.
- Analog computers can be easily expanded – if a problem grows larger the computer can grow with the problem.

Now

The Decline

Despite these advantages the decline of analog computers started in the 1970s for a number of reasons:

- Memory programmed digital computers became fast enough to compete at least with small and medium analog computers in terms of computing power.
- Memory programmed digital machines became simpler and thus cheaper each month while analog computers continued being highly complex machines – expensive and hard to maintain and run.
- Digital computers offered more flexibility due to time sharing etc. which allowed a better utilization of the machines.
- Over time, the expertise in analog computing was lost – abstract algorithmic approaches which are adapted to the machines won over analog approaches which were adapted to the problems to be solved.

Now that analog computing is largely a forgotten art, its particular advantages are still too precious to relinquish – imagine a machine consisting of

- an off-the-shelf x86-workstation and
- additional extension cards containing digital implementations of typical analog computer elements.

Such a machine would easily outperform even most of todays traditional parallel computers due to the inherent fine grain parallelism in analog problem solving.

Even without specialized hardware, analog computing would be a great basis to generate code for highly parallel architectures like graphic cards.

Such a digital implementation of analog computer parts is nothing new – first attempts were done in the form of **DDA**s[10] like **MADDIDA**, built by Northtrop at about 1950 (cf. [Reed 2006]) or TRICE (cf. [Ameling 1963]).

A modern direct implementation could use FPGAs for example while another approach could use highly parallel computers like the **Green Arrays GA144** chip[11] which contains 144 independent F18A processors – each of which might implement a group of traditional analog computing elements thus forming high level building blocks[12].

[10] Digital Differential Analyzer

[11] Cf. [Green Array 2010].

[12] This approach would renew interest in the **Hannauer matrix** – cf. [Hannauer 1968].

Why would one want to build such a machine?

- It would be ideally suited to tackle dynamic systems of all kinds (mechanics, electronics, biology, nuclear physics, ...) – just as a traditional electronic analog computer but without its drawbacks.
- It could possibly deliver more computing power for the buck in these areas than traditional parallel digital computers.
- Programming this system would be based mainly on the DEQs describing the problem to be solved, simplifying programming substantially.
- It could be a great tool in education due to its high performance and possible high interactivity.

Thank you for your interest!

The author can be reached at

ulmann@analogmuseum.org

The author would like to thank Scott Kelbell, Dr. Reinhard Steffens and Arno Jacobs for valuable suggestions and corrections to this talk.

Personal note: If you have any old analog computing equipment, documentation etc. looking for a good new home, please contact the author.

Bibliography

- [Ameling 1963] W. Ameling, "Aufbau und Arbeitsweise des Hybrid-Rechners TRICE", in *Elektronische Rechenanlagen*, 5 (1963), Heft 1., pp. 28 – 41
- [EAI] N. N., *PACE 231R analog computer*, Electronic Associates, Inc., Long Branch, New Jersey, Bulletin No. AC 6007
- [Green Array 2010] N. N., "Green Array GA144 – 144-Computer Chip", `http://www.greenarraychips.com/home/documents/greg/PB001-100503-GA144-1-10.pdf`
- [Hannauer 1968] George Hannauer, *Stored Program Concept for Analog Computers*, EAI Project 320009, NASA Order NAS8-21228, 1968
- [Holst 1982] Per A. Holst, "George A. Philbick and Polyphemus – The First Electronic Training Simulator", in *Annals of the History of Computing*, Vol. 4, Number 2, April 1982, pp. 143 – 156

Bibliography

- [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
- [Merlin, Moore 2008] Peter W. Merlin, Tony Moore, *X-Plane Crashes*, Specialty Press, 2008
- [Reed 2006] Irving S. Reed, "The Dawn of the Computer Age", in *Engineering & Science*, No. 1, 2006, pp. 7 – 12
- [Thomson 1876] William Thomson, "Mechanical integration of linear differential equations of the second order with variable coefficients", Proceedings of the Royal Society, Vol. 24, No. 167, 1876, pp. 269 – 270
- [Waltman 2000] Gene L. Waltman, *Black Magic and Gremlins: Analog Flight Simulations at NASA's Flight Research Center*, NASA SP-2000-4520, 2000