# Analog Chaos

Prof. Dr. Bernd Ulmann, anabrid GmbH

10. Oktober 2026



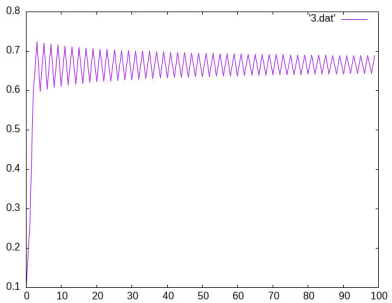


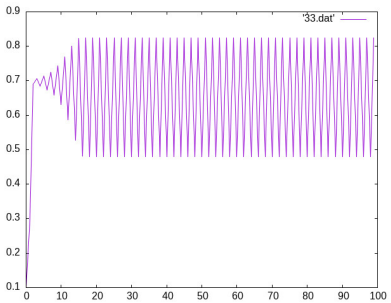

# Example

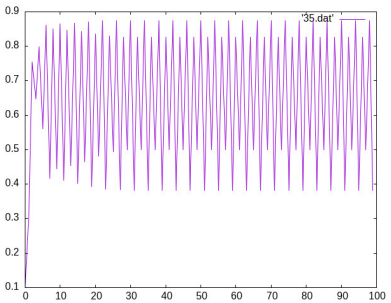
Imagine a simple iterative process:

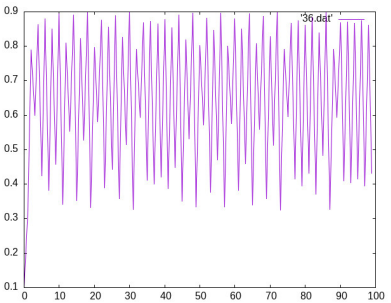
$$x_{n+1} = r x_n (1 - x_n)$$

with some initial condition like $x_0 = 0.1$ and a single control parameter $r$. What happens when $r$ changes?










These are results for $r = 3$ (convergent), $r = 3.3$, $r = 3.5$ (period doubling), and $r = 3.6$. What started rather orderly ended up in "chaos".[1]

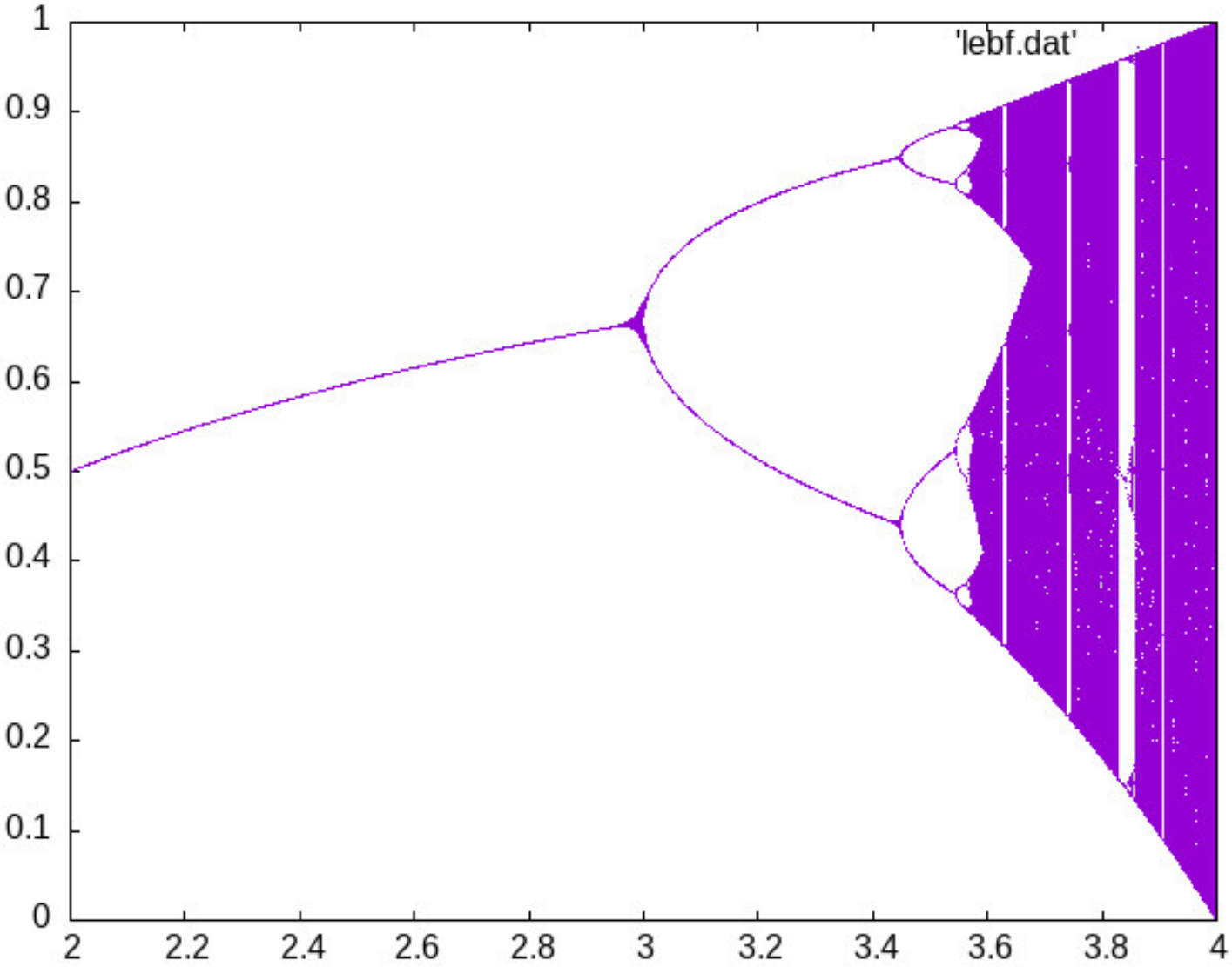
[1] This is an example of a *discrete* chaotic system – there are also *continuous* chaotic systems which will be in the focus in the following.

This simple system shows such a rich behavor that [May 1976, p. 467] writes:

> *"I would therefore urge that people be introduced to, say, [this] equation early in their mathematical education. [. . .] Such study would greatly enrich the student's intuition about nonlinear systems. Not only in research, but also in the everyday world of politics and economics, we would all be better off if more people realised that simple nonlinear systems do not necessarily possess simple dynamical properties."*

## The bifurcation diagram

A powerful tool to visualize the behavior of these (and other) systems is the *bifurcation diagram*:



## What is a chaotic system?

A chaotic system is

- a dynamical system, i. e., its state changes with time.
- It is deterministic, i. e., there is nothing "unknown" about the system itself. . .
- Yet, long term predictions are impossible (weather prediction is a good example – pretty good short term but basically useless long term).
- It is highly sensitive to initial conditions and parameters – even tiniest changes can and will alter the system's behavior substantially, i. e., nearby trajectories in phase space separate exponentially fast.
- It is aperiodic, i. e., there are no regular cycles, no periodicity.
- It is topologically mixing: Every region of its phase space will eventually overlap any other such region (think of mixing different colors in a bucket by stirring it).

## A very short history[3]

In 1814 Pierre-Simon de Laplace introduced what is now known as Laplace*'s demon* to illustrate the idea of causal determinism.

The basic idea is that an all-knowing entity (the demon) knowing position and momentum of every particle in the universe could compute all past as well as future values based on classical mechanics.

In 1880 Henri Poincaré, studying orbital mechanics, realized that orbits can be nonperiodic without either reaching a fixed point or growing infinitely – a rather unexpected result.[2] Which helped to kill the idea of Laplace's demon.

At the heart of such problems lie *differential equations*:

[2] See [Diacu et al. 1996].

[3] See [Oestreicher 2007].

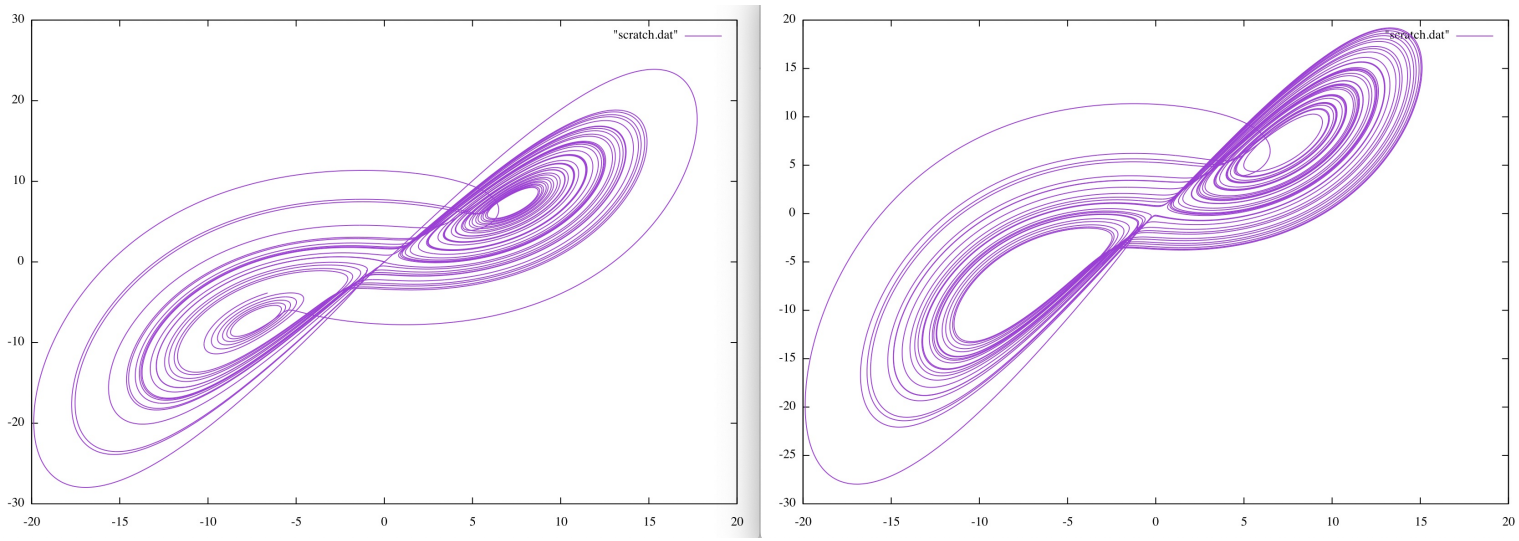
## Differential equations

What are *differential equations* (*DEQ*s)?

- They relate an unknown function to its derivative(s).
- Example: Which function $f(t)$ equals its first derivative with respect to $t$? Obviously something like $f(t) = ce^t$.
- The above question would be normally stated as either $\frac{\mathrm{d}}{\mathrm{d}t} f(t) = f(t)$ or just $\dot{f} = f$ when $t$ is the free variable (which is typically not even written as the argument).
- In contrast to simple algebra, the solution to a DEQ is not a single value but a function or typically a family of functions.
- DEQs are the "language of nature" – they are incredibly fascinating and notoriously difficult to solve. (Most do not even have analytical solutions.)
- Like in the case of coupled linear equations, there are, of course, also systems of coupled DEQs (linear and nonlinear), there are even *partial differential equations* (*PDE*s) but these are out of scope here.

## A very short history[6]

In the early 1960s, Edward Norton Lorenz developed a system of three coupled differential equations as a simplified model of atmospheric convection.[4]
Using a Royal McBee LGP-30 digital computer[5] he realized that even tiniest changes in the initial conditions of the system resulted in a vastly different trajectory of the system in its phase space (here with IC $(1, 1, 1$ and $(1, 1, 1.001))$:




[4] See [Lorenz, 1963].
[5] Programming was done by Ellen Fetter and Margret Hamilton.
[6] See [Oestreicher 2007].

How quickly do these differences grow?

A central characteristic of a dynamical system is its *spectrum of* LYAPUNOV *exponents* (this contains as many LYAPUNOV exponents as the system has dimensions in the phase space).

Of these exponents the maximal one (*MLE*) is of interest:

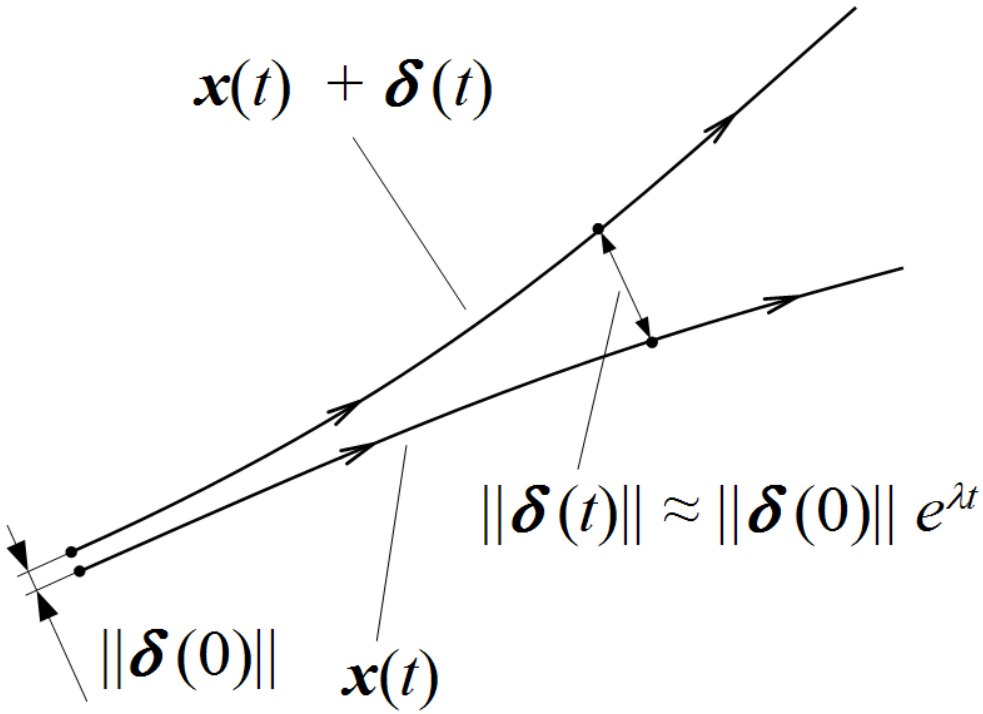
$$\lambda = \lim_{t\to\infty} \lim_{|\delta_0|\to 0} \frac{1}{t} \ln \frac{|\delta(t)|}{\delta_0}$$

$\delta_0$ is the *initial separation vector*. Two trajectories of the same system started with initial conditions $|\delta_0|$ apart from each other will diverge at a rate of

$$|\delta(t)| \approx |\delta_0| e^{\lambda t}.$$

If $\lambda > 0$, the system is chaotic. A "normal" chaotic system has one positive MLE.

# Lyapunov exponents[7]




Source: By Yapparina - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=42399467.

[7] If someone is looking for a thesis topic: Determining the MLE with a hybrid computer setup would be highly interesting.

## Poincaré-Bendixson

In the following continuous dynamical systems are considered. These are naturally described by systems of coupled differential equations (*DEQ*s).

How many equations are necessary to achieve chaotic behavior?

One is clearly not sufficient. Would two DEQs suffice?

No! The Poincaré-Bendixson theorem states that a two-dimensional continuous dynamical system cannot exhibit a chaotic behavior. At least three coupled DEQs are needed for that.

Note: That does not hold true for discrete dynamical systems. Here, even a single dimension (logistic equation) may suffice for chaotic behavior.

## Why should we care?

Why should we care?

- Chaotic systems are just highly interesting in themselves.
- Their phase space plots are often incredibly beautiful (you will see quite some in the following).
- They teach us about the complexity of real-world systems (and the impossibility to predict many things far into the future).
- They occur basically everywhere: Weather, climate, financial markets, biological systems, control systems, etc.
- They can be used for cryptographic purposes, to generate random numbers...
- They can be used for reservoir computing.

## Analog?

In the following chaotic systems will be implemented and treated using analog computers...

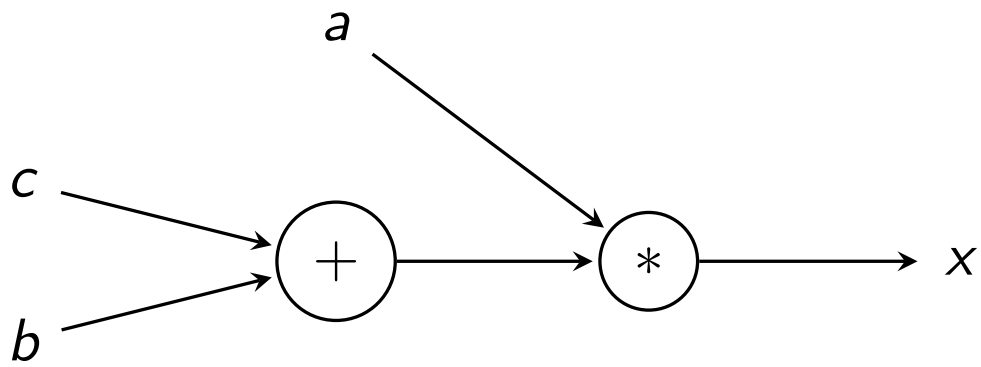
???...What is an analog computer and why should one do this?

- An analog computer is a *non-algorithmic* machine.
- It consists of a (large) number of computing elements such as integrators, summers, multipliers, etc.
- A *program* for an analog computer is a directed graph specifying how these computing elements are connected.
- Values are typically represented as voltages or currents.
- Analog computers are
  - things of beauty and joy forever![8]
  - Extremely powerful.
  - Extremely energy efficient (factors of $10^3$ over digital computers are not uncommon).
- There are signs that chaotic systems can not be fully simulated by digital computers (see [Coveney 2024])!

[8] See [McLeod et al. 1958, p. 488].

Example: Solve $a(b+c)$ on a digital and an analog computer:

```
LOAD  A, R0
LOAD  B, R1
LOAD  C, R2
ADD   R1, R2, R1
MULT  R0, R1, R0
STORE R0, ...
```



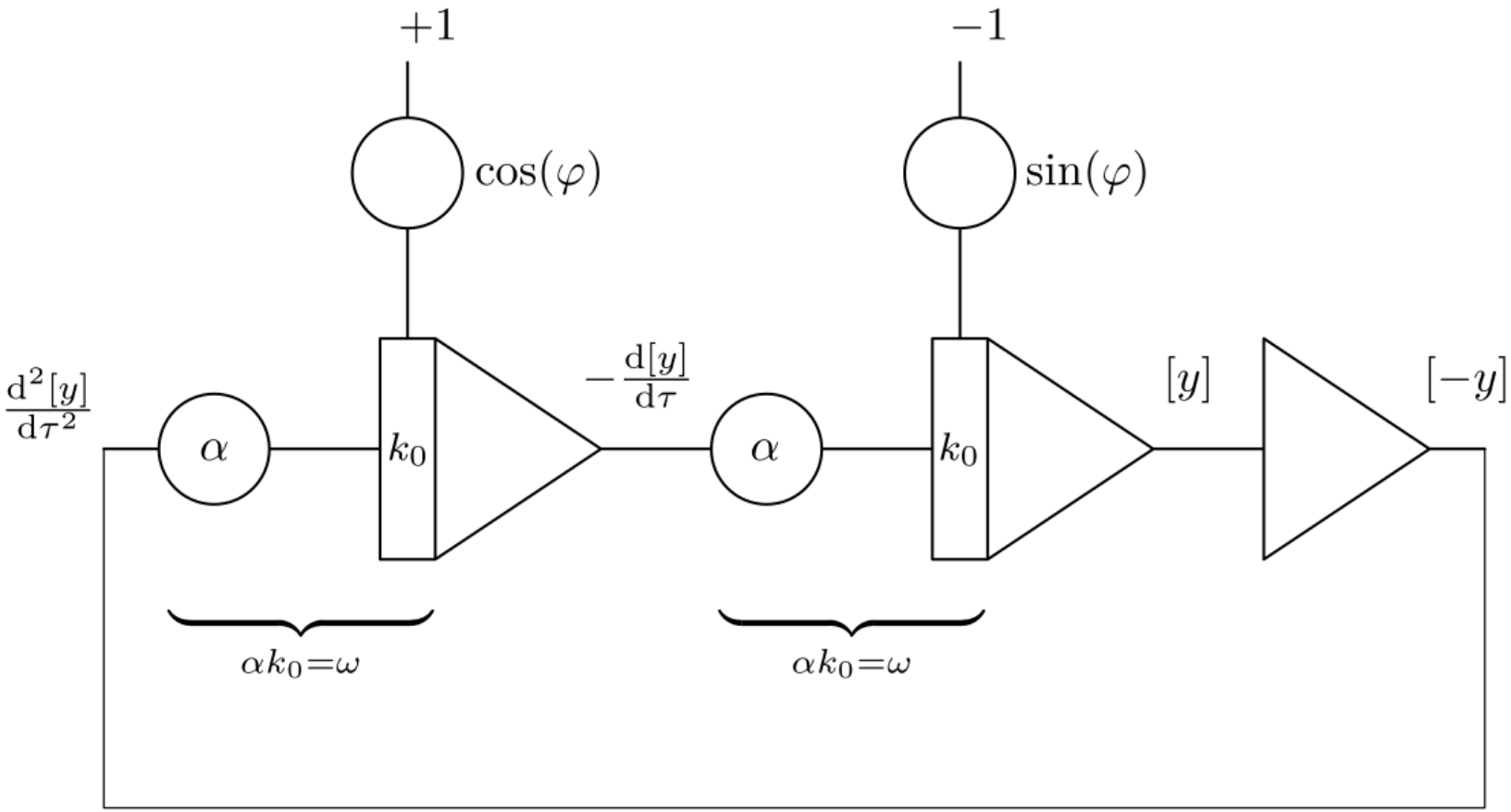
## Programming

Programming an analog computer is really straightforward:

- Write down the problem equations.
- Scale the equations since the voltage/current range for variables in an analog computer program is limited (typically to $[-1, 1]$).
- Rearrange the equations so that the highest derivative is on the left side.
- Create all lower derivatives and terms on the right side of the equation using computing elements such as integrators, summers, multipliers, coefficients, etc.
- Connect this result to the input of the very first integrator we started with (*feedback loop*).
- Define initial conditions for the integrators.
- Let the magic happen...

## Example

Let us solve a simple DEQ on an analog computer: What is the solution of $\ddot{y} = -\omega^2 y$?



## Advantages

Why should one use an analog computer at all?

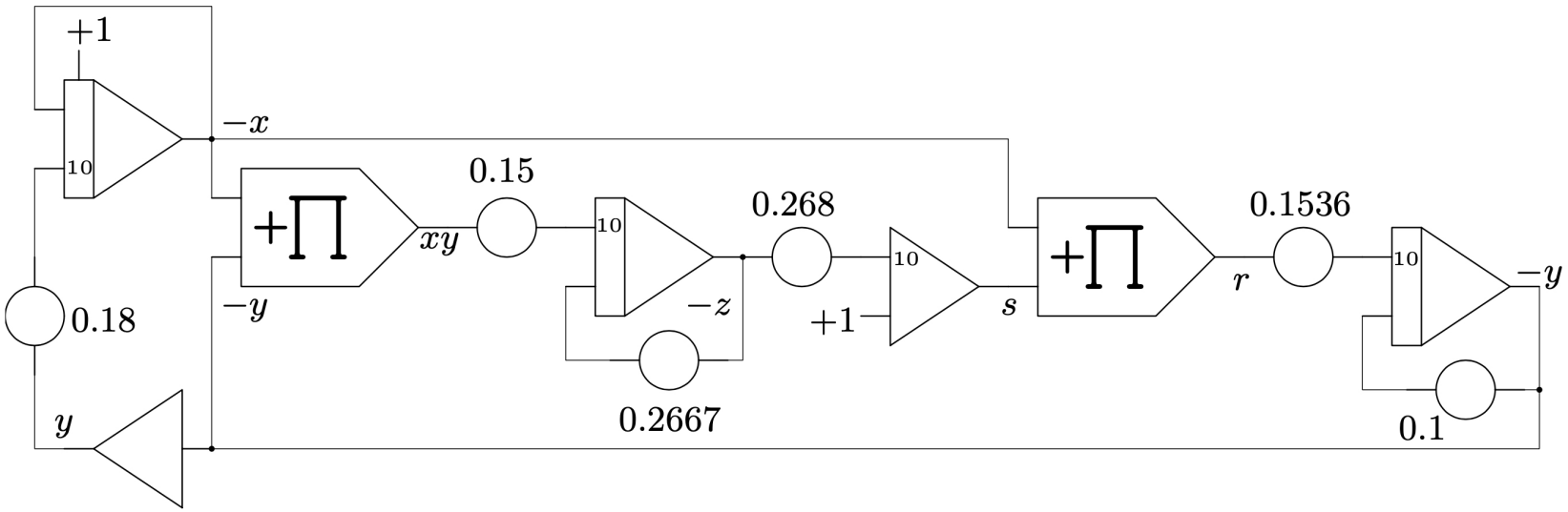
- They are ideally suited to treat problems described by systems of DEQs and to a certain extent even PDEs.
- All computing elements work 100% parallel.
- Solution times are typically much faster than on digital computers (depending on the analog computer factors of $\approx 10^3$ can be achieved).
- They are highly energy efficient.
- Modern analog computers are programmed and controlled from an attached digital computer so that one does not need to care about the fact that one is using an analog computer at all.

## The Lorenz system

The following example is much more difficult: The classic LORENZ system is described by the *scaled* equations

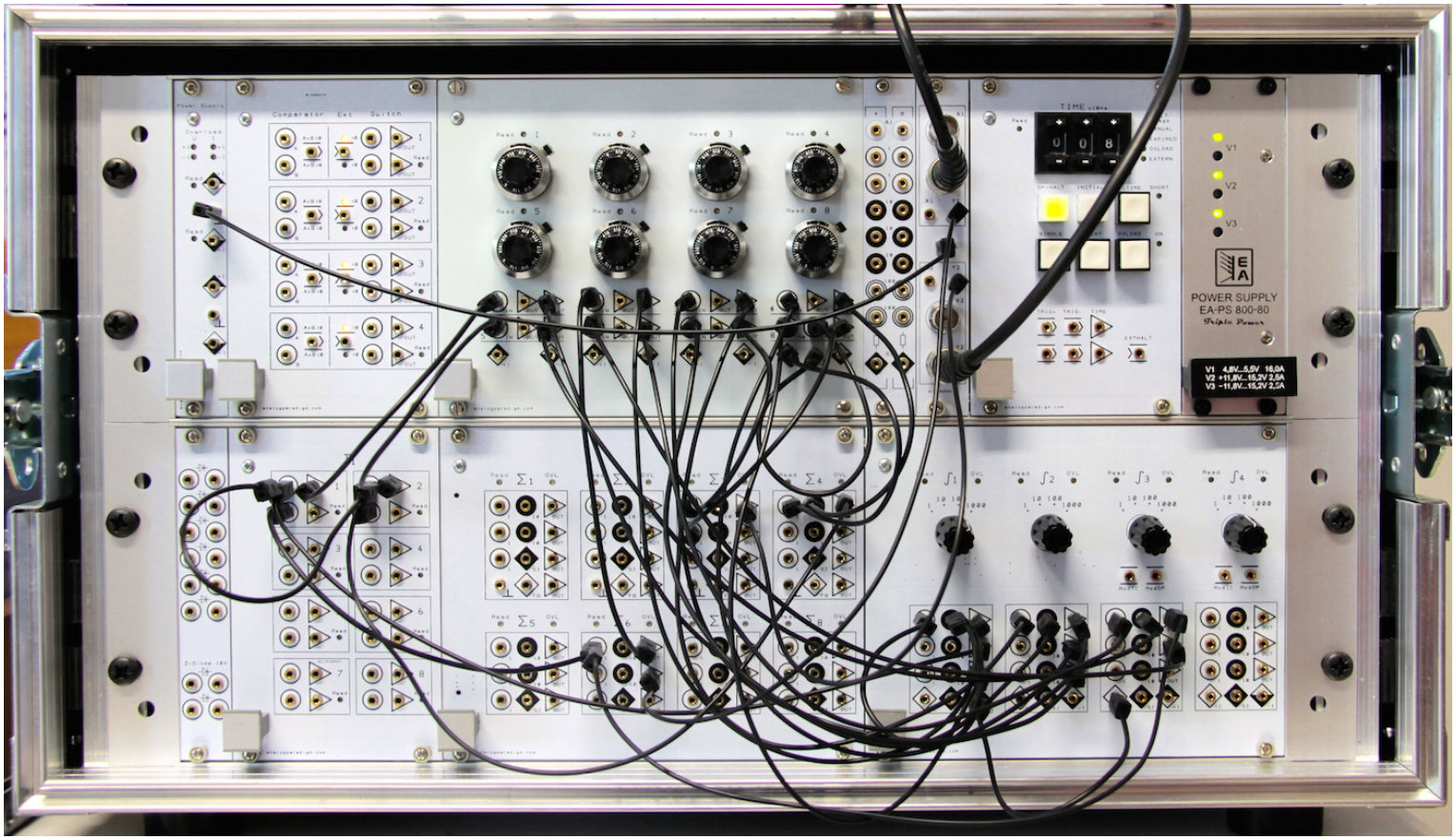
$$\dot{x} = 1.8y - x,$$
$$\dot{y} = 1.56x(1 - 2.678z) - 0.1y, \text{ and}$$
$$\dot{z} = 1.5xy - 0.2667z$$

The corresponding program looks like this:



## Implementation

On a classic/traditional analog computer it looks like this:




It should be noted that recent systems such as anabrid GmbH's *LUCIDAC* and REDAC systems no longer need manual patching. Programming is done using a Python library and a compiler stack.

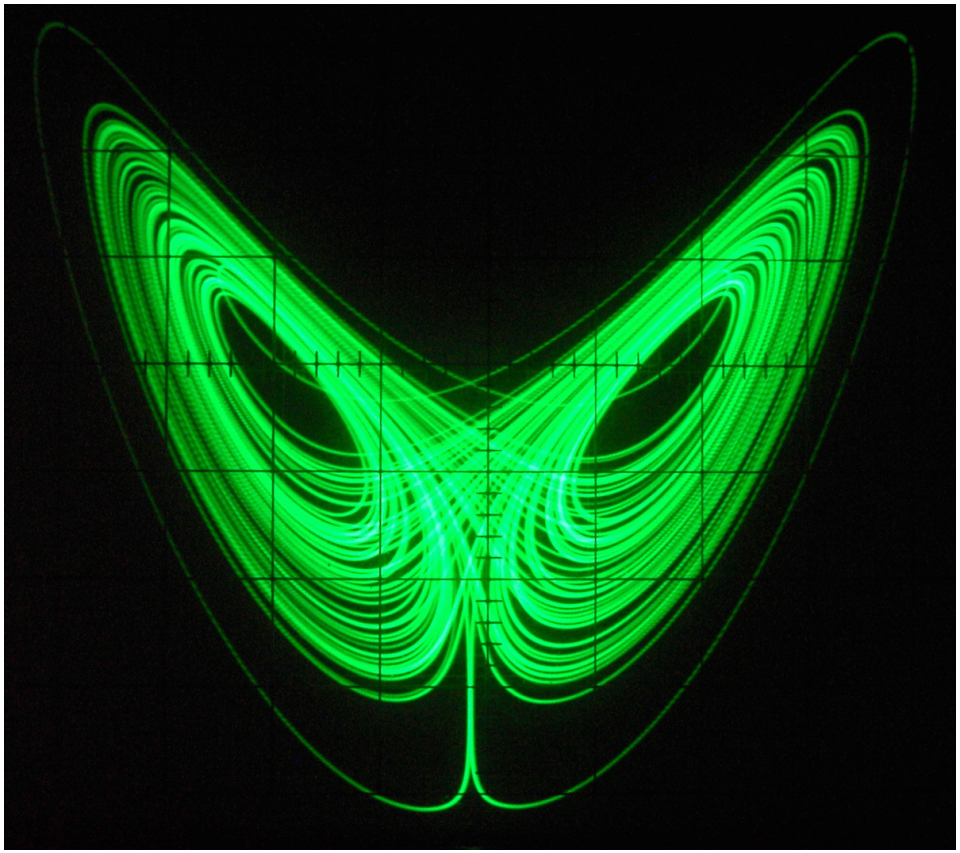
# DSL

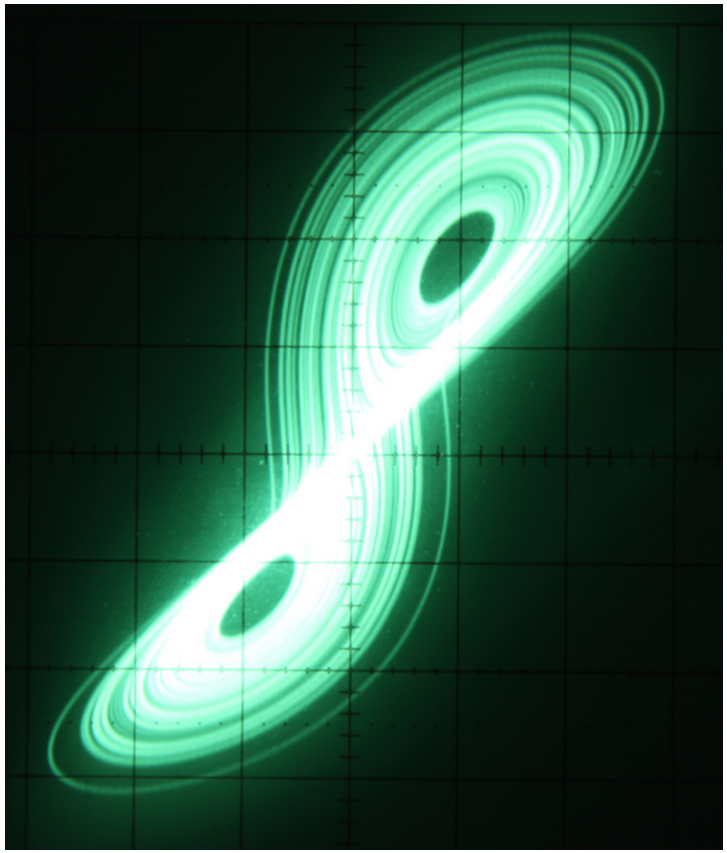
Nowadays the corresponding analog computer program just looks like this:

```
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);
```

## Phase space plots

Here are two analog phase space plots of this system:





Isn't that beautiful?

## So, what?

Now, why should we bother implementing chaotic systems using analog computers (or, more generally, analog electronics)?

- Using the analog computer as a co-processor for a digital computer, it makes it possible to explore the behavior of dynamical systems highly efficiently. The digital computer controls parameters and initial conditions and reads results from the analog computer.
- Even with a stand alone analog computer one can explore the behavior of highly complex systems as the following example shows.

Let us have a look at another fascinating system:

1996, Lorenz published another chaotic system which consists of $N \in \mathbb{N}, N \geq 4$ coupled differential equations and only a single control parameter $F$:[9]

$$\dot{x}_i = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F$$

with

$$x_0 = x_N,$$
$$x_{-1} = x_{N-1}, \text{ and}$$
$$x_{N+1} = x_1.$$

In the following we use $N = 4$:

$$\dot{x_0} = 20(x_1 - x_2)x_3 - x_0 + F^*, \tag{1}$$
$$\dot{x_1} = 20(x_2 - x_3)x_0 - x_1 + F^*, \tag{2}$$
$$\dot{x_2} = 20(x_3 - x_0)x_1 - x_2 + F^*, \text{ and} \tag{3}$$
$$\dot{x_3} = 20(x_0 - x_1)x_2 - x_3 + F^* \tag{4}$$

[9] See [Lorenz 1996, pp. 4 f.] and [Sterk et al. 2017].

## Attractor types

Basically, a dynamical system may exhibit one of four attractor types:

Point attractor: This is the most basic attractor type – a point to which neighboring states in phase space are attracted. There is no "escape" once the point attractor has been reached.

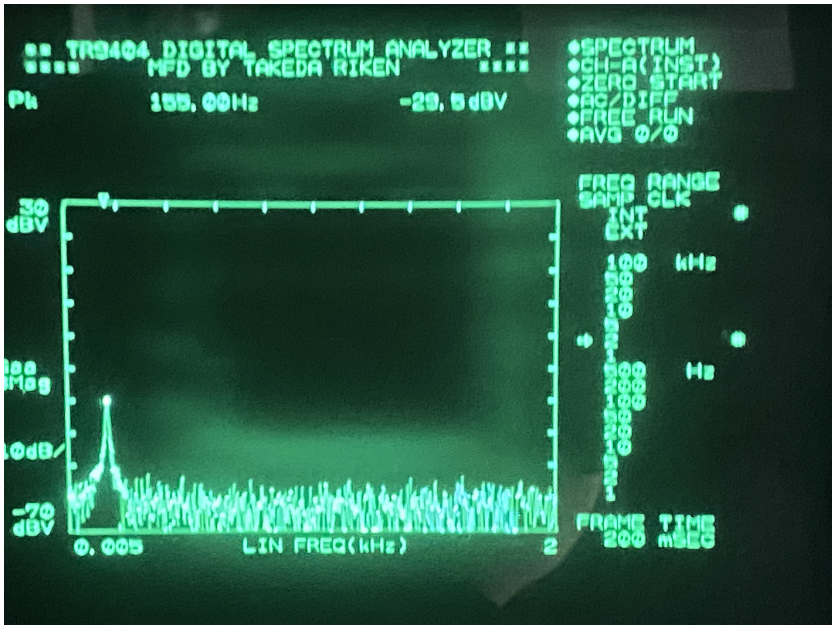
Periodic attractor: This is a set of states of a dynamical system towards which neighboring states converge in a cyclic pattern (limit cycles).

Torus attractor: The system states lie on the surface of an $n$-dimensional torus (example: a double pendulum).

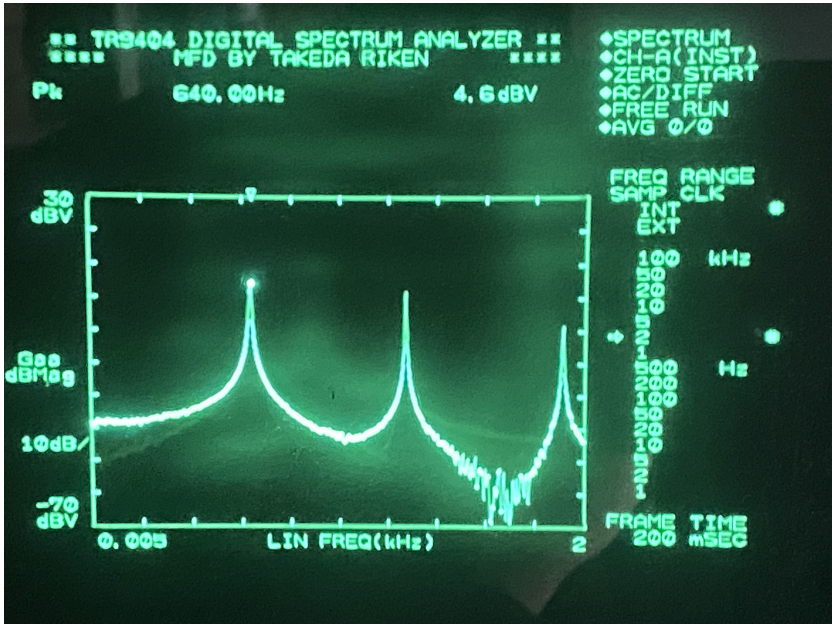
Strange attractor: The system is aperiodic, i. e., it never repeats but stays within a bounded region. Tiny differences in initial conditions grow exponentially over time.

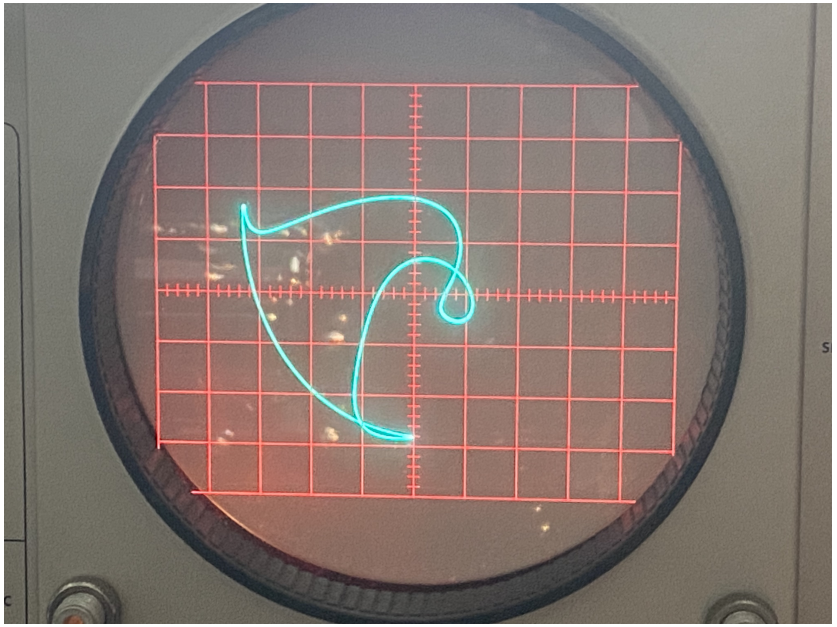
These can now be directly observed (FFT spectrum on the left, phase space on the right (the point attractor is boring and omitted) by varying $F$:

# Periodic attractor (start – end)

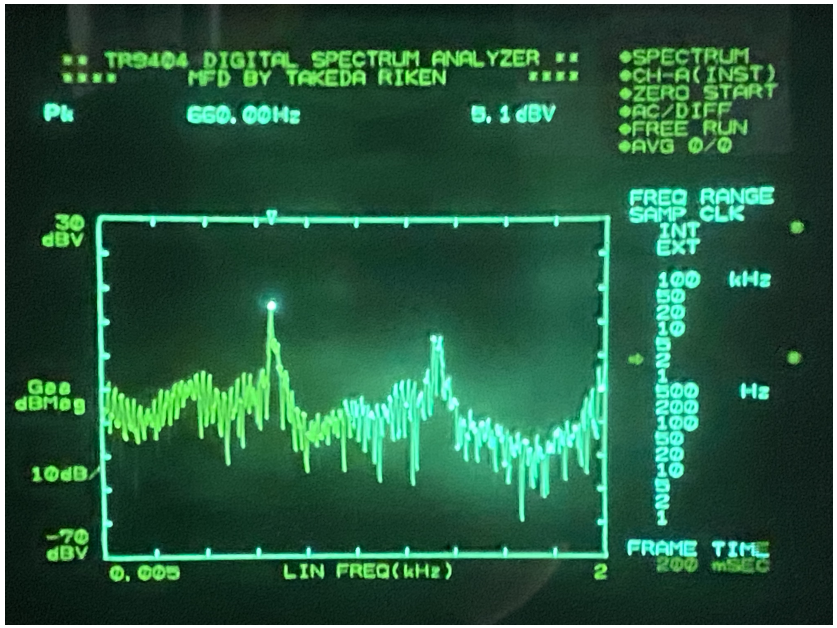


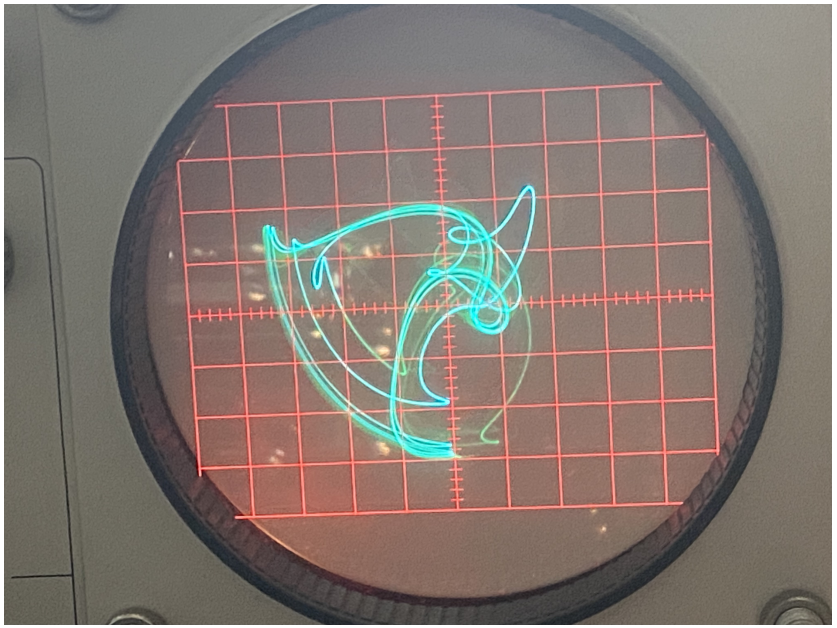


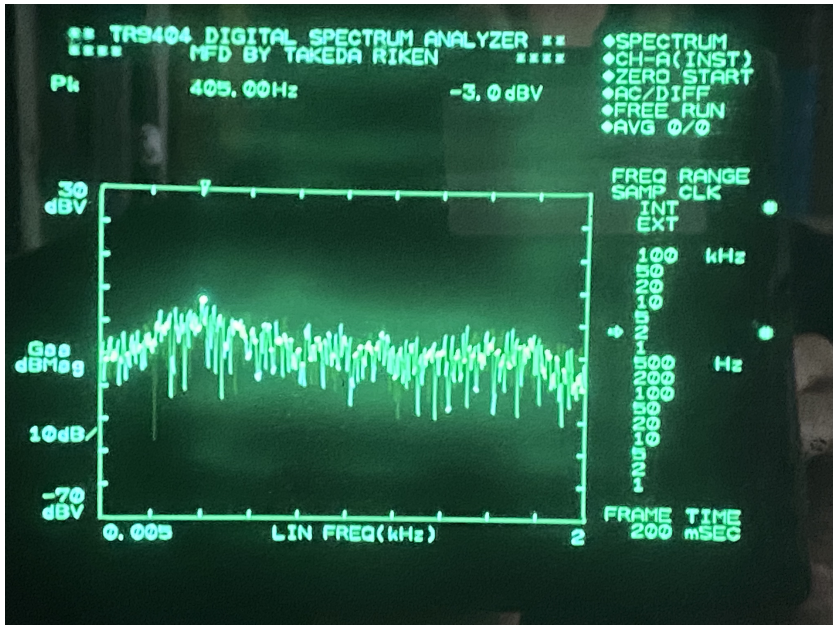



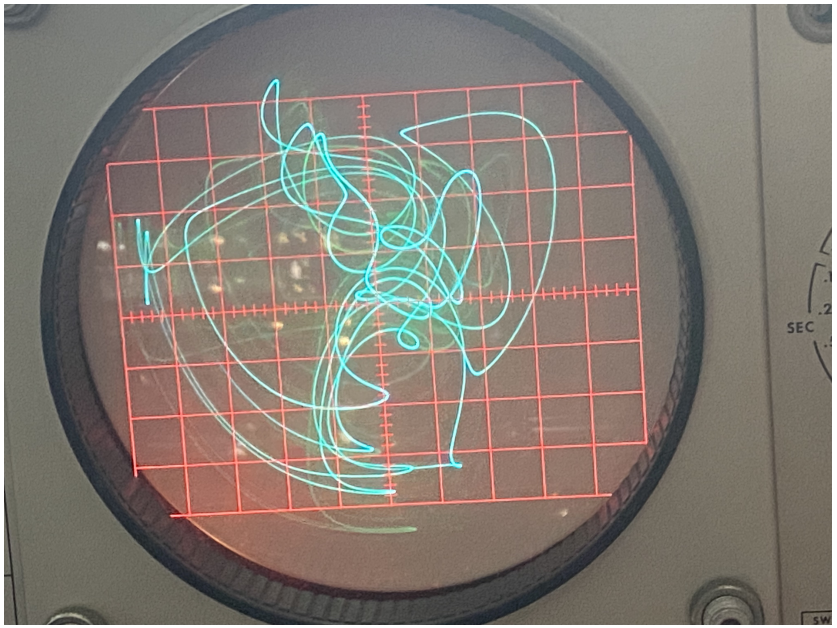



# Strange attractor (start – end)



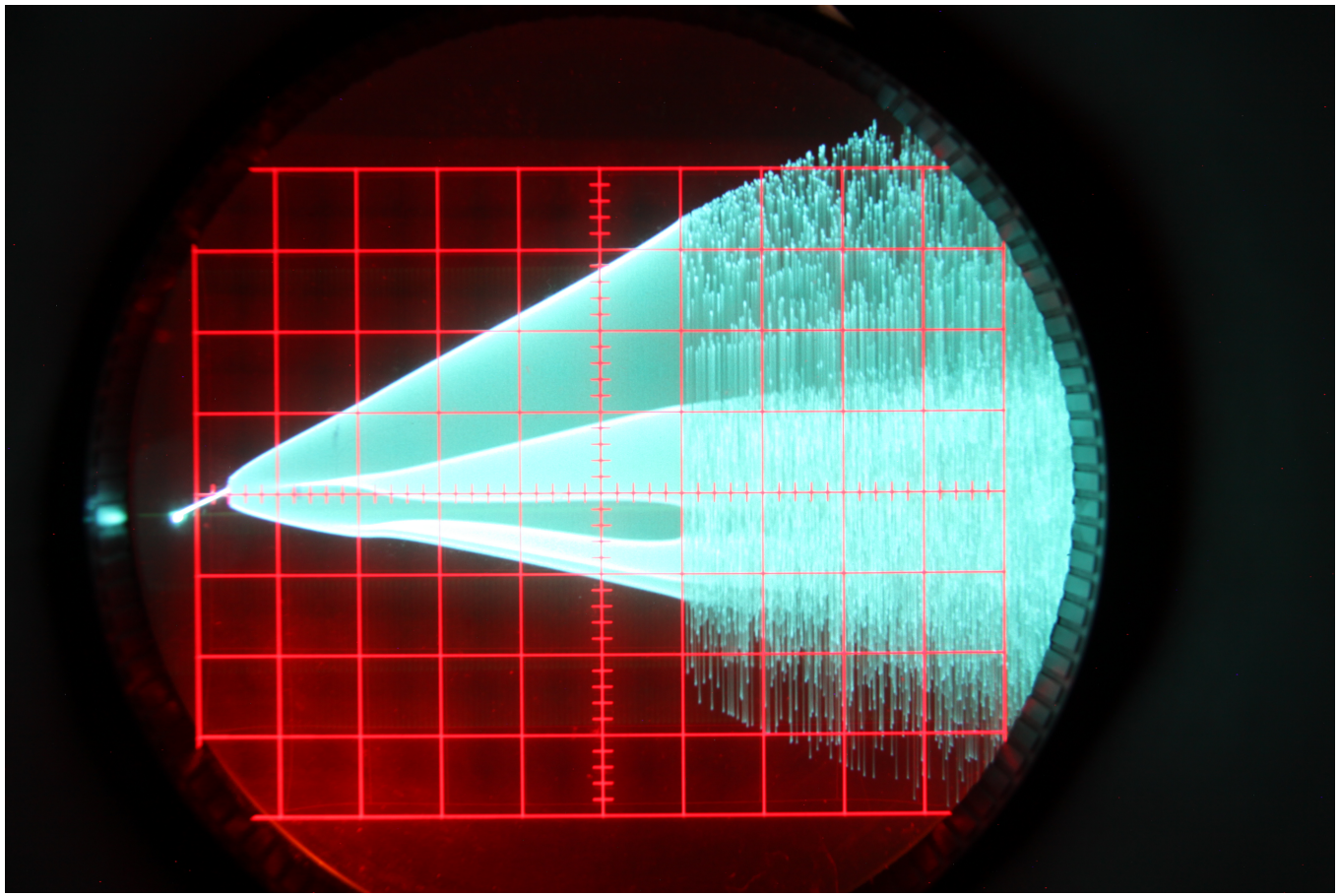





## Bifurcation diagram

A *bifurcation diagram* shows values reached (possibly asymptotically) by a dynamical system, typically depending on a control variable. This is the resulting display from varying $F$ linearly from 0 to a certain maximum value for the LORENZ '96 system:



Is there more than "just chaos"?

While a normal chaotic system has one positive LYAPUNOV exponent, *hyperchaotic systems* have at least two positive LYAPUNOV exponents. This, in turn, means that at least four coupled DEQs are necessary for hyperchaos to emerge.

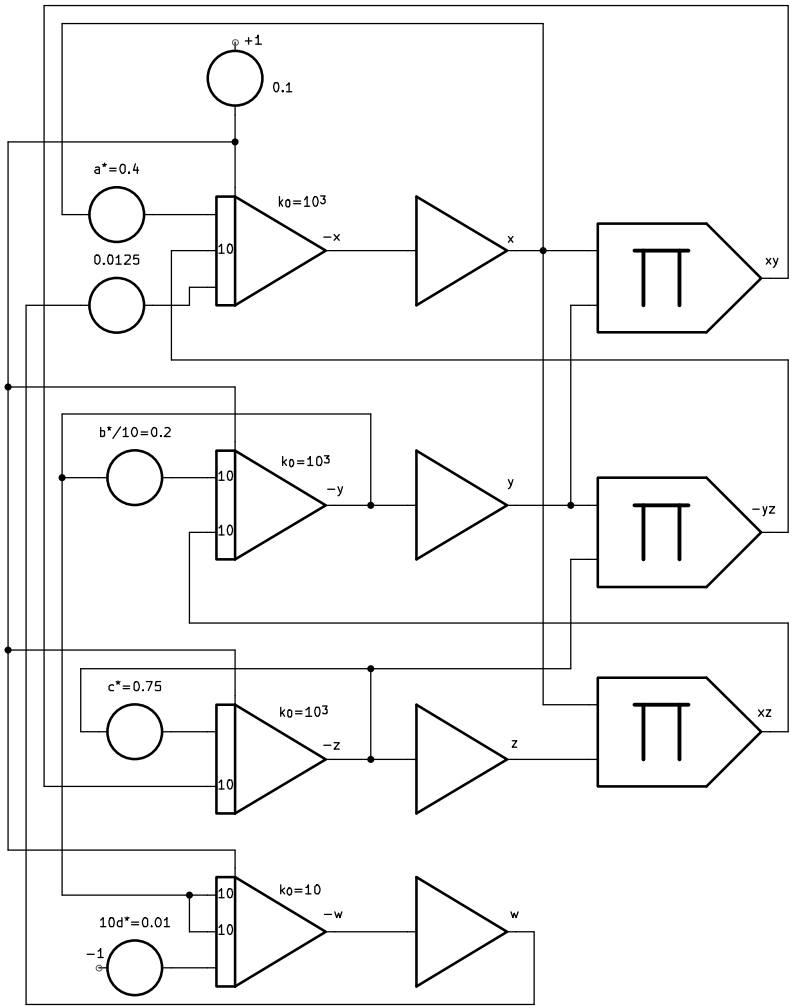
A nice example is the system

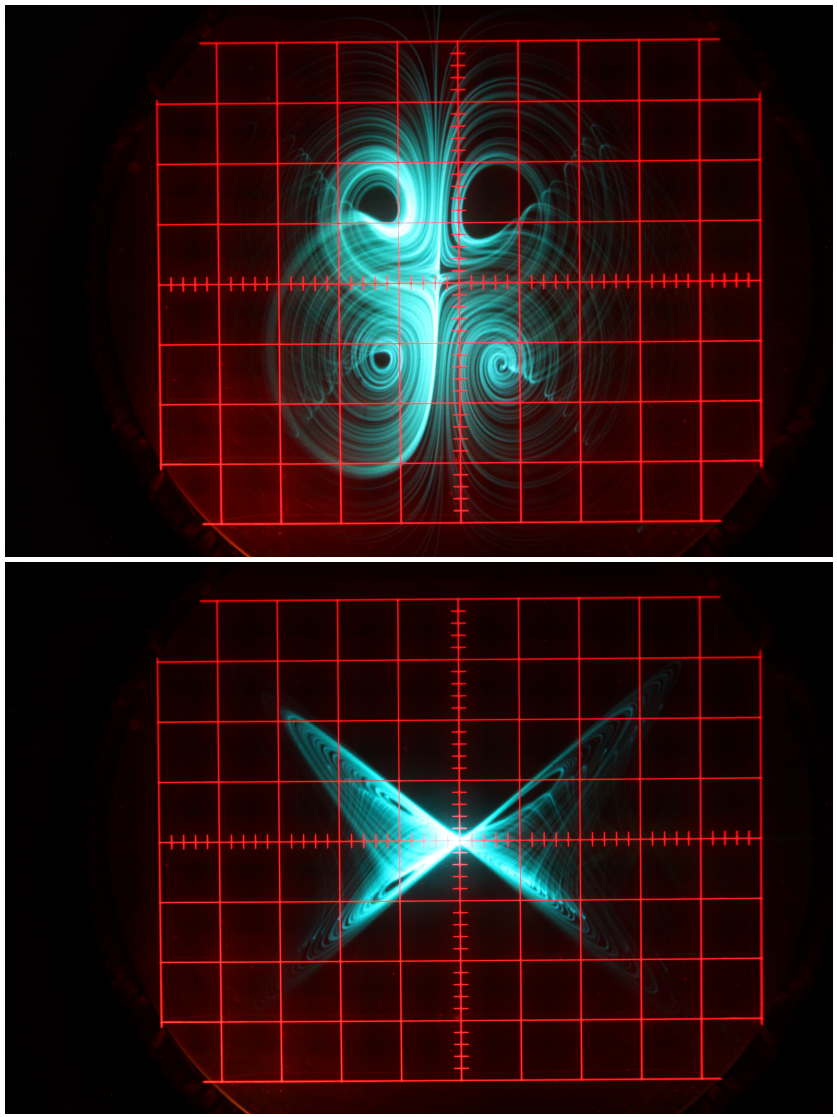
$$\begin{aligned}\dot{x} &= ax - yz + w,\\ \dot{y} &= xz - by,\\ \dot{z} &= xy - cy, \text{ and}\\ \dot{w} &= -y - d\end{aligned}$$

with $a = 8$, $b = 40$, $c = 15$, and $d = 0.1$.[10] This has two positive LYAPUNOV exponents: 3.036 and 0.01.

[10] See [BENKOUIDER et al. 2024].

# Hyperchaos


+1
0.1
a*=0.4
k0=10^3
10
−x
x
0.0125
xy
b*/10=0.2
k0=10^3
10
10
−y
y
−yz
c*=0.75
k0=10^3
10
−z
z
xz
k0=10
10
10
−w
w
10d*=0.01
−1



## Randomness

For sufficiently large values of $F$ the frequency spectrum of the Lorenz '96 system looks like that of a noise source.

Can a chaotic system be used as a source of randomness? Yes![11]

This can be an alternative to more direct means of noise/random generation such as Johnson–Nyquist noise voltage over a resistor, shot noise in a Zener diode, or radioactive decay processes.

Applications for this are abundant:

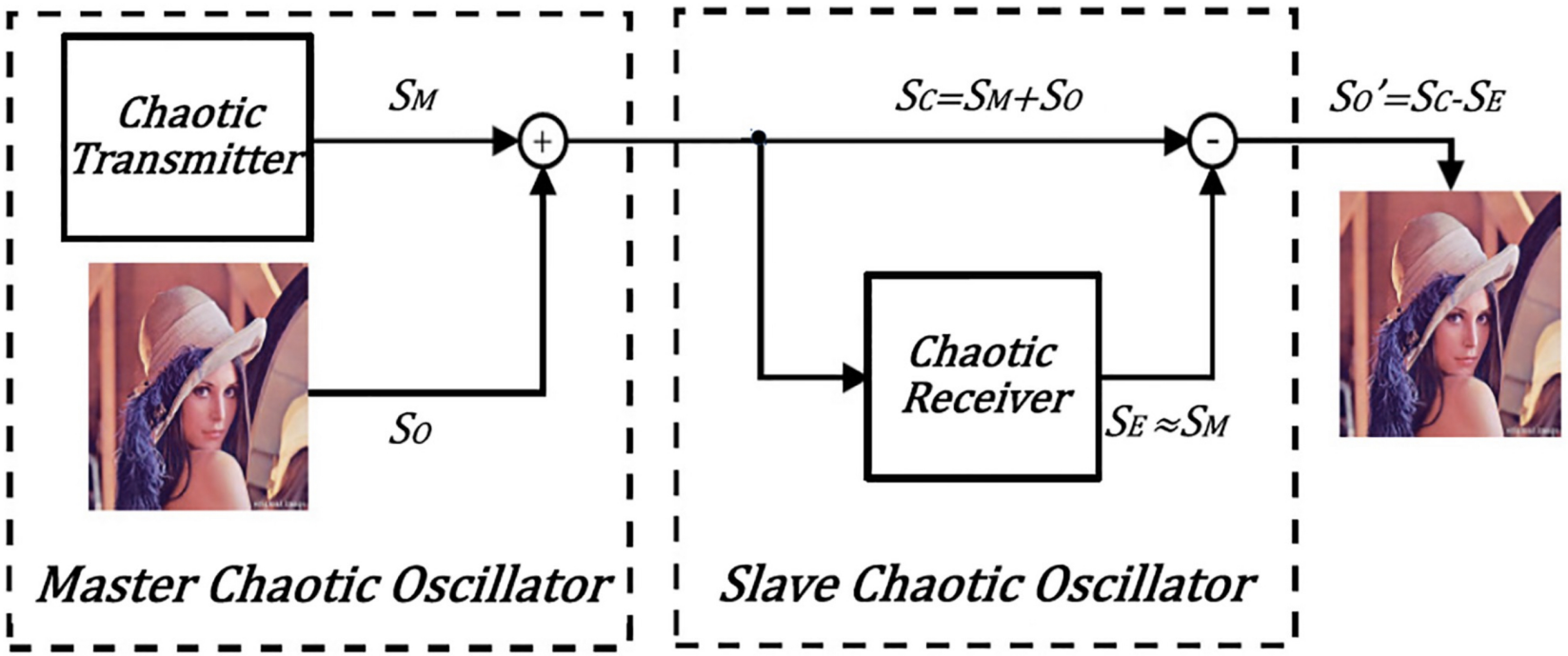
- Key generation for cryptographic and security applications.
- Random number generation for simulations, etc.
- Random selection.

[11] See [Yu et al. 2019]. [Nguyen et al. 2020] uses a hyperchaotic system as random source.

- Using chaotic systems for secure key generation.
- *BARN* (*bury among random numbers*): Generate a true random bitstream and interject (or XOR) data bits at positions known only to transmitter and receiver[12] (Effectively, this is a steganographic stream cipher.)
- Another application of chaotic systems to cryptography is based on the observation that chaotic systems can be synchronized.[13]
- Using this approach, speech can be masked[14] or images can be transmitted.[15]

[12] See [Tatarkiewicz et al. 2024].
[13] See [Pecora et al. 2015] and [Padmanaban et al. 2011].
[14] See [Cuomo et al. 1993].
[15] See [Guillén-Fernández et al. 2019].

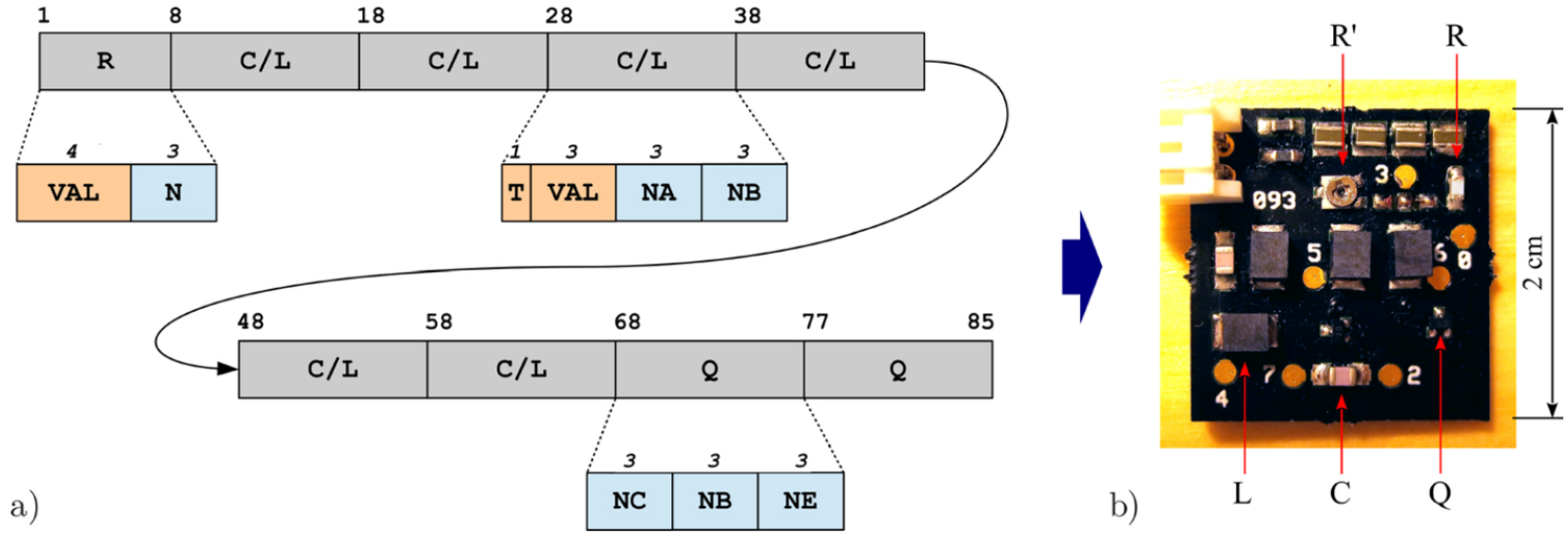
Chaotic systems can be used like this, e. g.:[16]




[16] See [Guillén-Fernández et al. 2019]. Here, digital implementations (FPGA based) of such systems are used but it would be worthwhile to dive deeper into analog realizations of such systems.

## Simple transistor circuits

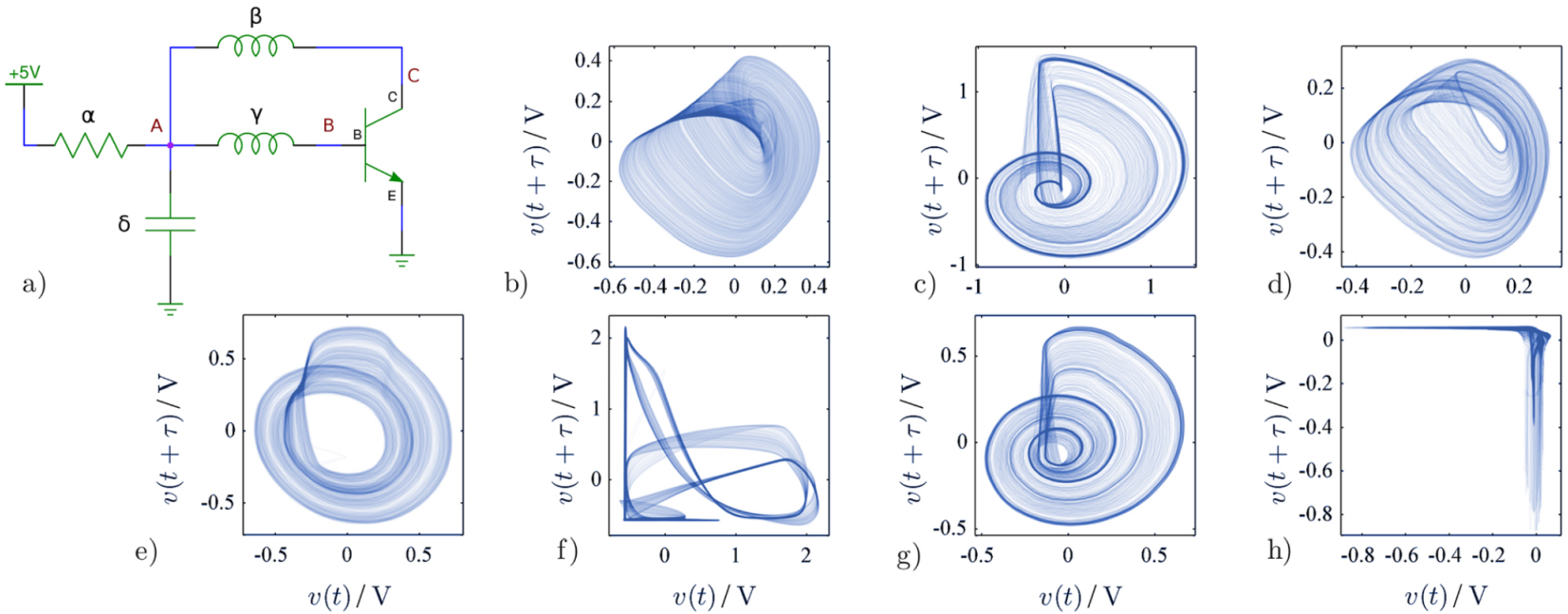
A fascinating approach was taken by MINATI and colleagues[17]: Electronic circuits of up to eight nodes, two of which were BJTs, were encoded by an 85 bit random sequence. The circuit was then pruned to remove short circuited components, etc. Circuits containing at least one inductor, one capacitor, and one BJT were considered valid and fed into a SPICE simulation.




[17] See [MINATI et al. 2017], etc.

# Simple transistor circuits

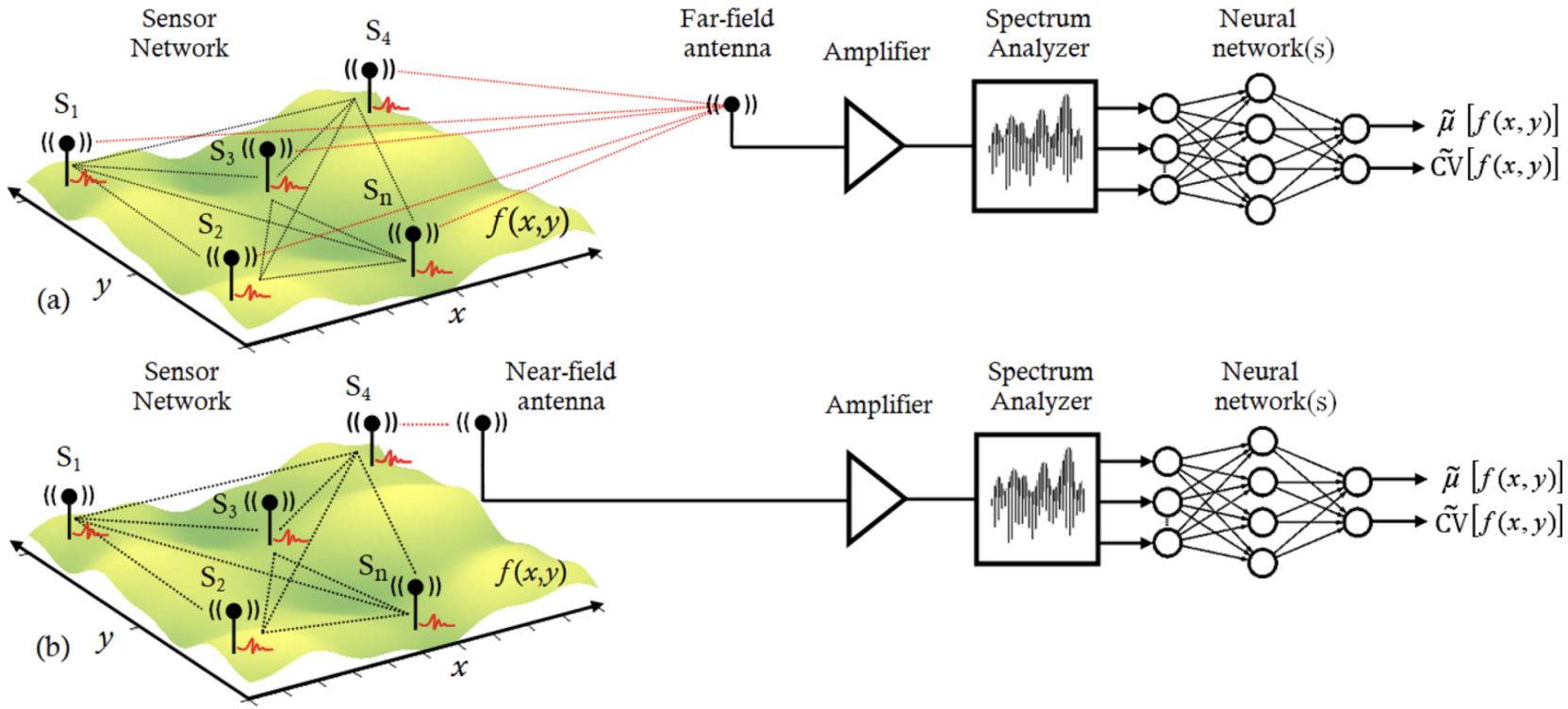
This resulted in 49 novel basic circuits exhibiting chaotic behavior with this being the smallest one:[18]




[18] The attractors were obtained with different component values.

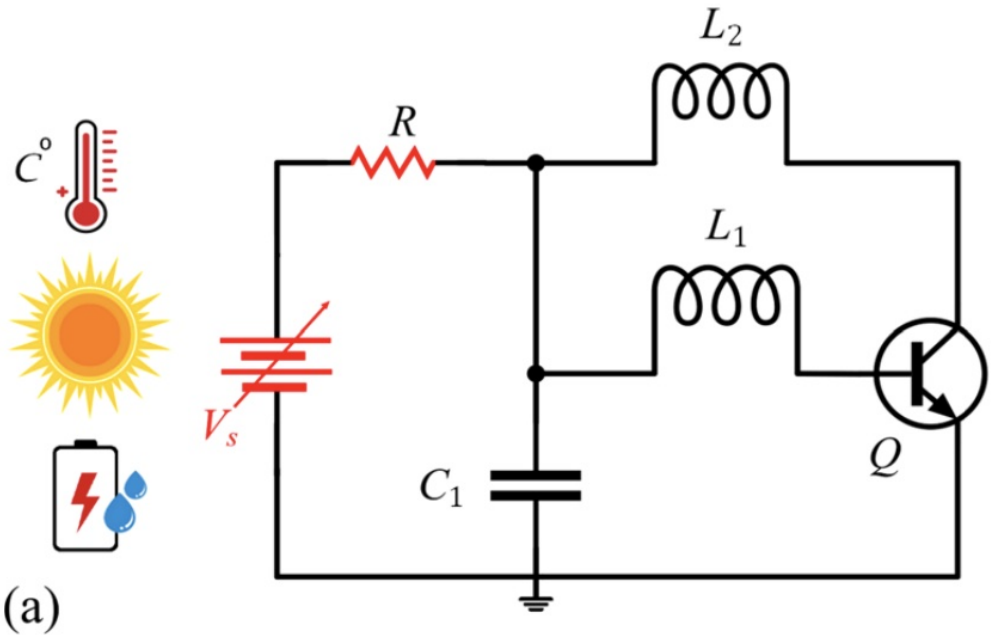
# Application to sensors

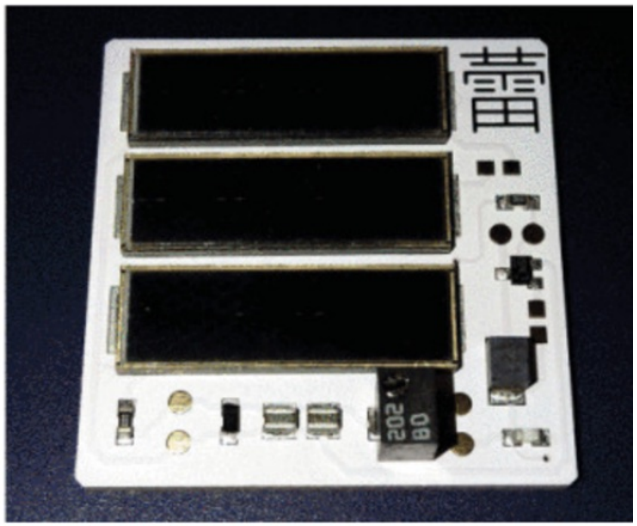
Such simple chaotic circuits can be used as a basis for distributed networks of sensors in a large area.[19]




[19] See [Fossi et al. 2025].

A typical circuit can be as simple as this one:[20]




[20] See [Fossi et al. 2025].

So much about continuous chaos. What about discrete systems? In contrast to the continuous systems shown before they evolve in discrete time steps and thus require some form of *clock* and some storage elements (*sample&hold*, *SAH*).

The archetypal discrete system exhibiting chaotic behavior is the *logistic equation* of the general form

$$x_{n+1} = r x_n (1 - x_n)$$

with a single parameter $r$ which has already be shown.[21]

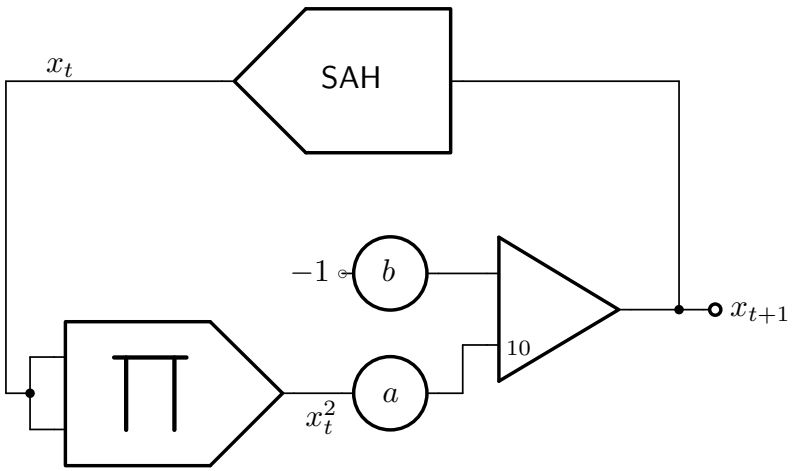
[21] Details can be found in [May 1976], [Strogatz 2015, pp. 360 ff.], and [Steeb et al. 1989, pp. 14 ff.].

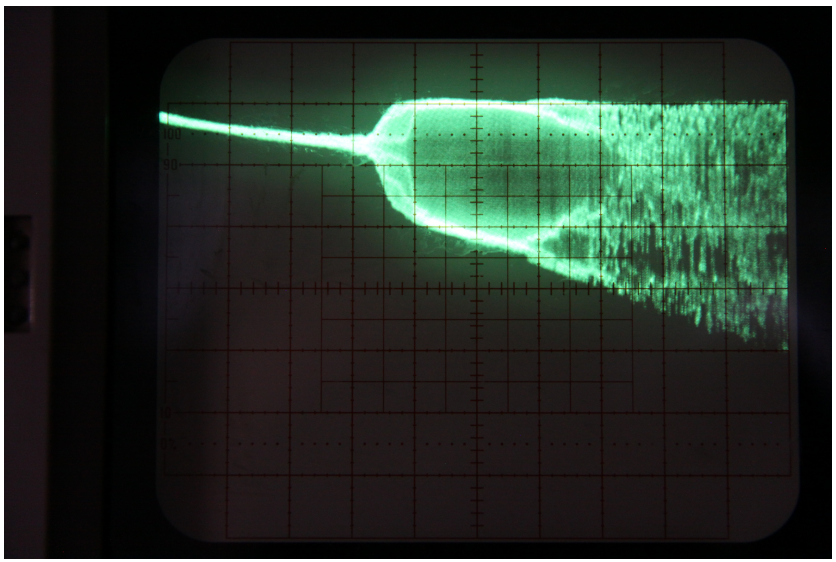
## The logistic equation

[Wu et al. 2022] propose a slightly different model exhibiting similar dynamics while avoiding this initial value problem:

$$x_{t+1} = b - ax_t^2 \tag{5}$$

The resulting program is straightforward and varying the parameters automatically, a bifurcation diagram can be created:



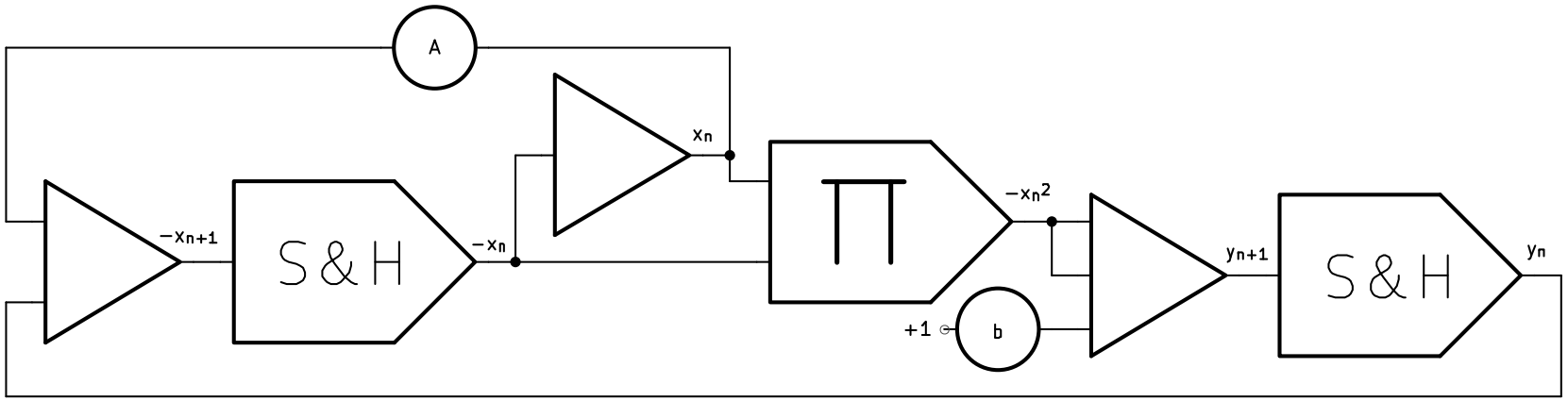



## The Cathala attractor

A more complex example is the CATHALA attractor[22]:

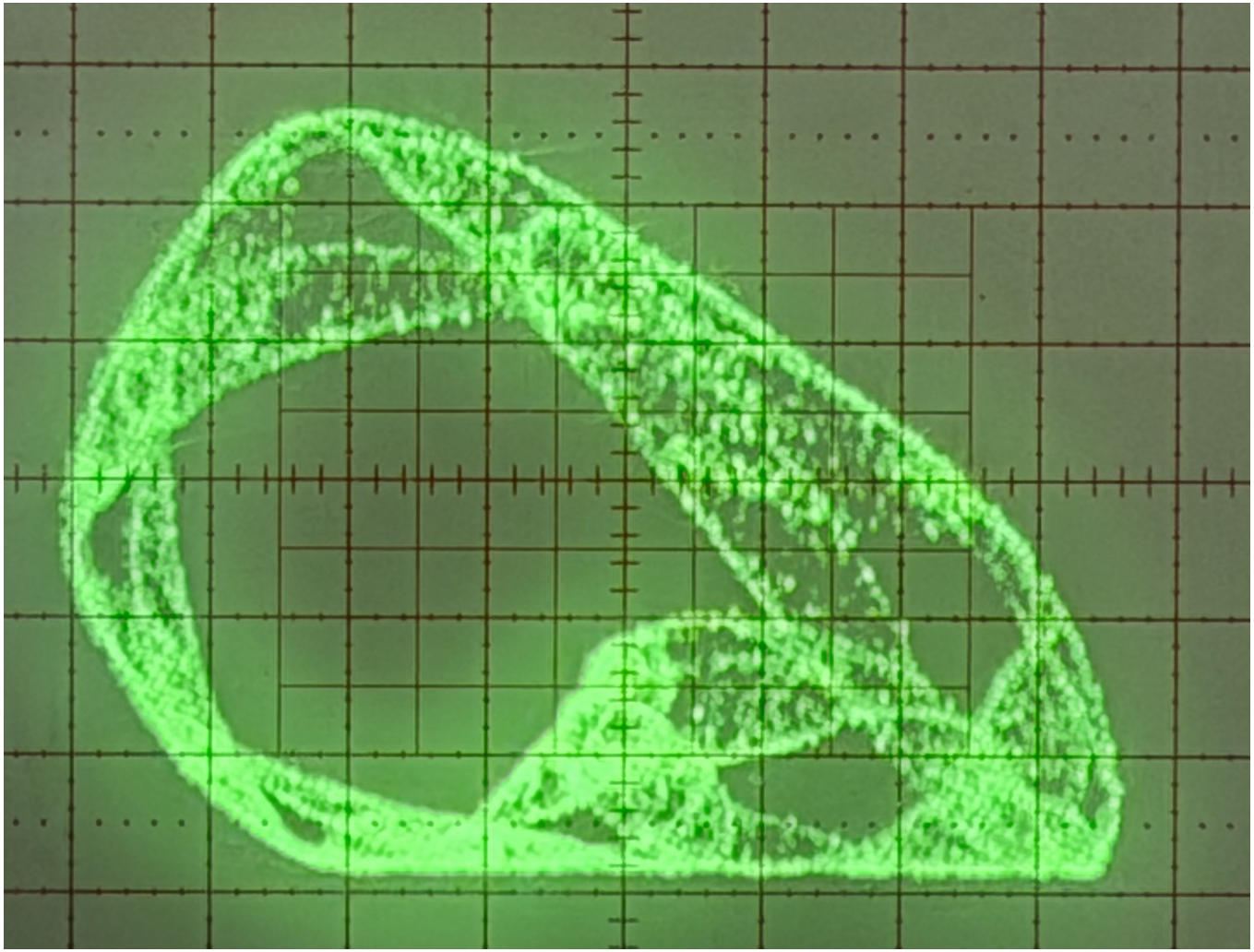
$$x_{n+1} = ax_n + y_n \text{ and}$$
$$y_{n+1} = -b + x_n^2$$

with $a = 0.7$ and $b = 0.82$. This requires two sample&hold elements:

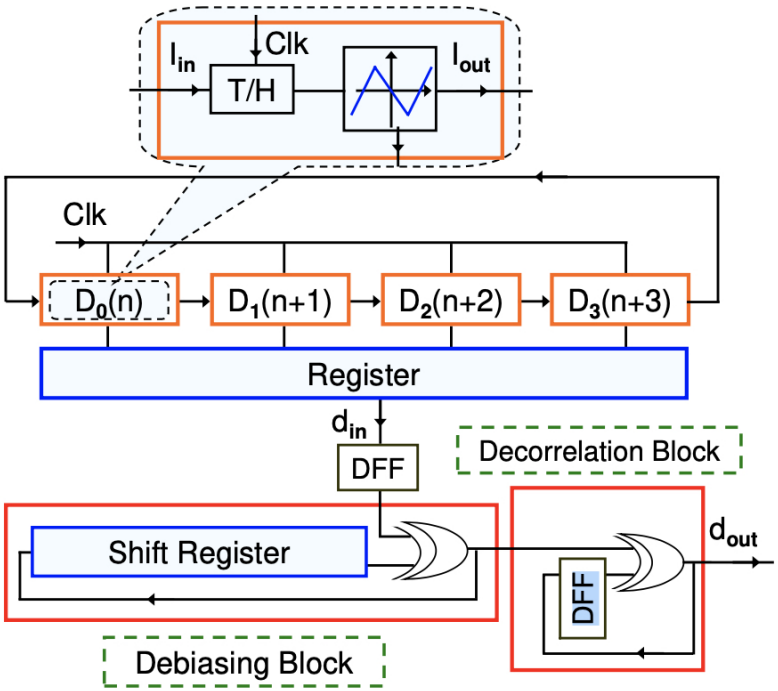



[22] See `http://www.3d-meier.de/tut19/Seite53.html`, retrieved 30.09.2026.

# The Cathala attractor



Such time discrete chaotic systems are ideally suited as the basis for random generators.
The underlying clock signal and the SAH elements are ideal for analog-digital-conversion (*ADC*). The ADC may be as simple as a one-bit ADC feeding some digital post-processing logic, such as a *debiasing block* (basically a feedback shift-register) followed by a simple *decorrelation block*. A nice example can be found in [Nejati et al. 2012].



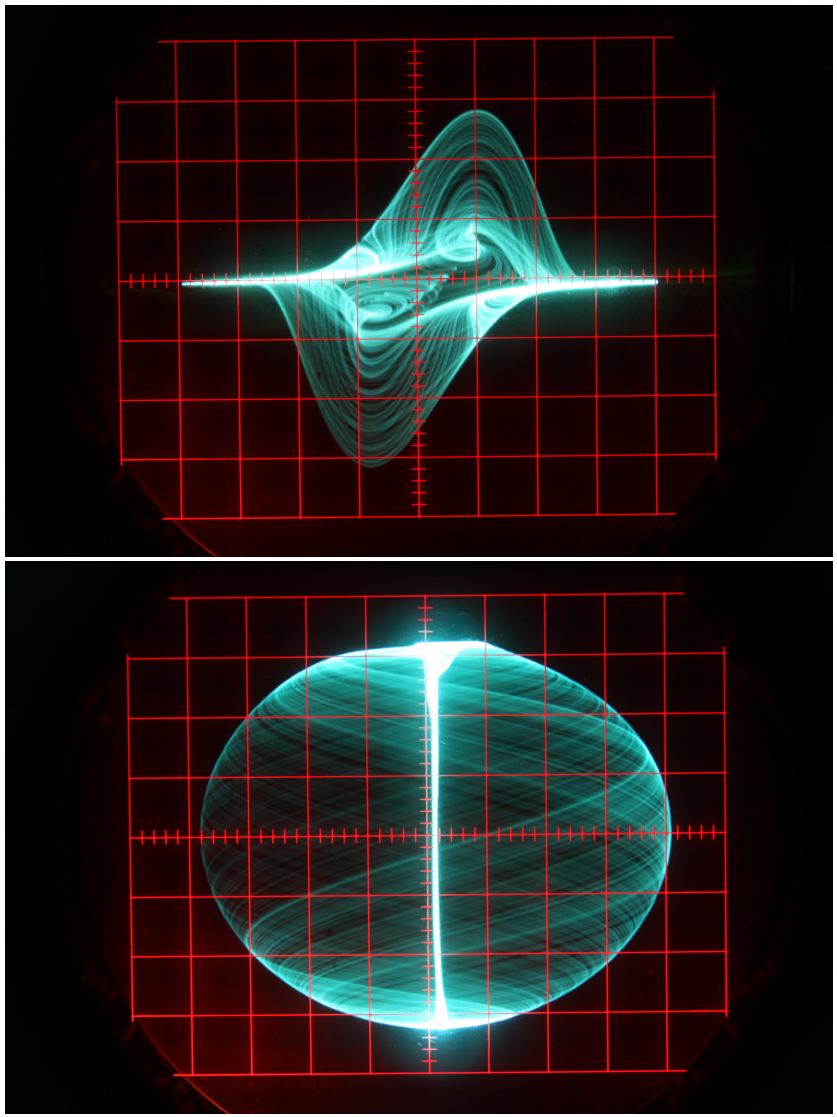
## Thank you!

**Thank you for your interest and happy analog computing!**

If you are (now :-) ) also fascinated by chaotic systems and analog computers, please contact the author at

ulmann@anabrid.com.

Save the date: Analog Computing Days '27 in Frankfurt/Main, Germany, 9./10.08.2027, https://analog-computing-days.org.



## Bibliography




Khaled Benkouider, Aceng Sambas, Talal Bonny, Wafaa Al Nassan, Issam A. R. Moghrabi, Ibrahim Mohammed Sulaiman, Basim A. Hassan, Mustafa Mamat, "A comprehensive study of the novel 4D hyperchaotic system with self-excited multistability and application in the voice encryption", in Nature, scientific reports, https://doi.org/10.1038/s41598-024-63779-1



Peter V. Coveney, "Sharkovskii's theorem and the limits of digital computers for the simulation of chaotic dynamical systems", in *Journal of Computational Science*, 83, 2024, 102449



Kevin M. Cuomo, Alan V. Oppenheim, "Circuit Implementation of Synchronized Chaos with Applications to Communications", in *Physical Review Letters*, Volume 71, Number 1, pp. 65–68

## Bibliography




Florin Diacu, Philip Holmes, *Celestial Encounters – The Origins of Chaos and Stabiliy*, Princeton University Press, 1996



Jules Tagne Fossi, Paul Didier Kamdem Kuate, Vitrice Ruben Folifack Signing, Zeric Tabekoueng Njitacke, Manyu Zhao, Theophile Fonzin Fozin, Pedro A. Valdes-Sosa, Andrea Gaiardo, Hiroyuki Ito, Arturo Buscarino, Ludovico Minati, "Distributed Estimation of Sensor Statistics Using Wireless Networks of Single-Transistor Chaotic Oscillators", in *Nonlinear Dynamics*, 2025



Omar Guillén-Fernández, Ashley Meléndez-Cano, Esteban Tlelo-Cuautle, Jose Cruz Núñez-Pérez, Jose de Jesus Rangel-Magdaleno, "On the synchronization techniques of chaotic oscillators and their FPGA-based implementation for secure image transmission", in *PLoS ONE*, 14(2): e0209618

## Bibliography



Edward Norton Lorenz, "Deterministic Nonperiodic Flow", in *Journal of the Atmospheric Sciences*, Volume 20, pp. 130–141, March 1963


Edward Norton Lorenz, "Predictability – a Problem Partly Solved", in *Proceedings on Seminar on Predictability*, 1, pp. 1–18


Robert M. May, "Simple mathematical models with very complicated dynamics", in *Nature*, Vol. 261, June 10, 1976, pp. 459–467


John H. McLeod, Suzette McLeod, "The Simulation Council Newsletter", in *Instruments and Automation*, Vol. 31, March 1958, p. 488


Ludovico Minati, Mattia Frasca, PawełOświęcimka, Luca Faes, Stanisław Drożdż, "Atypical transistor-based chaotic oscillators: Design, realization, and diversity", in *Chaos*, 073113, 2017

## Bibliography



Hamid Nejati, Ahmad Beirami, Warsame H. Ali, "Discrete-time Chaotic-Map Truly Random Number Generators: Design, Implementation, and Variability – Analysis of the Zigzag Map", `https://arxiv.org/pdf/1206.1039v1`, retrieved 03.10.2026


Ngoc Nguyen, Georges Kaddoum, Fabio Pareschi, Riccardo Rovatti, Gianluca Setti, "A fully CMOS true random number generator based on hidden attractor hyperchaotic system", in *Nonlinear Dynamics*, 2020, 102, pp. 2887–2904


Christian Oestreicher, "A history of chaos theory", in *Dialogues in Clinical Neuroscience*, 2007, September, 9(3), pp. 279–289


E. Padmanaban, Chittaranjan Hens, Syamal K. Dana, "Engineering synchronization of chaotic oscillators using controller based coupling design", in *Chaos*, 21, 01311, 2011

## Bibliography



Louis M. Pecora, Thomas L. Carroll, "Synchronization of chaotic systems", in *Chaos*, 25, 097611, 2015


W.-H. Steeb, A. Kunick, *Chaos in dynamischen Systemen*, BI Wissenschaftsverlag, 1989


A. E. Sterk, D. L. van Kekem, "Predictability of Extreme Waves in the Lorenz-96 Model Near Intermittency and Quasi-Periodicity", in *Complexity*, 2017, Article ID 9419024, `https://doi.org/10.1155/2017/9419024`


Steven H. Strogatz, *Nonlnead Dynamics and Chaos with Applications to Physics, Biology, Chemistry, and Engineering*, 2nd edition, Westview Press, 2015


Jan J. Tatarkiewicz, Wiesław Kuźmicz, "New Class of Ciphers Using Hardware Entropy Source", `https://arxiv.org/pdf/2404.09288v1`, retrieved 03.10.2026

# Bibliography




Xianming Wu, Longxiang Fu, Shaobo He, Huihai Wang, "Analogue circuit implementation of a new logistic-like map", in *Electronics Letters*, July 2022, Vol. 58, No. 14, pp. 533–535



Fei Yu, Lixiang Li, Qiang Tang, Shuo Cai, Yun Song, Quan Xu, "A Survey on True Random Number Generators Based on Chaos", in *Discrete Dynamics in Nature and Society*, 2019, Article ID 2545123