THAT on steroids

After showing how simple it is to implement something as complex as the AIZAWA system on the REDAC with anabrid's compiler technique,[1] I felt the urge to spend my Sunday afternoon doing the same on a THAT just for fun.

The governing equations are

$$\dot{x} = (z - \beta)x - \omega y \tag{1}$$

$$\dot{y} = \omega x + (z - \beta)y \tag{2}$$

$$\dot{z} = \lambda + \alpha z - \frac{z^3}{3} - \left(x^2 + y^2\right)(1 + \rho z) + \varepsilon z x^3 \tag{3}$$

with $\alpha = 1$, $\beta = 0.7$, $\lambda = 0.6$, $\omega = 3.5$, $\rho = 0.25$ and ε "small", i. e., $0 \leq \varepsilon \leq 0.1$.[2] That is obviously way too complex to be implemented on a single THAT. Just counting how many products are involved shows that this problem requires nine multipliers – far beyond the two multipliers of a THAT.

However, being an analog computer, is it quite easy to add external computing elements to a THAT. In this case a spare Model-1 MLT8 module containing eight multipliers was added. Using a spare 10-inch chassis, a simple linear power supply yielding ± 15 V and $+5$ V was built. The chassis holds an Arduino based hybrid controller,[3] a small adapter board with four BNC connectors, a homebrew sample&hold circuit,[4] and the aforementioned MLT8 module.[5] The THAT is connected to the expansion chassis by means of its hybrid port from which it is also powered by the external $+5$ V source. The overall system is shown in figure 1. The THAT fits nicely on top of the expansion chassis.

[1] See https://analogparadigm.com/downloads/alpaca_64.pdf, retrieved 15.02.2026.

[2] See [LANGFORD 1984] for details. To be honest, it is not clear at the moment why this particular system is called AIZAWA system (see https://www.algosome.com/articles/aizawa-attractor-chaos.html or https://sequelaencollection.home.blog/3d-chaotic-attractors/, both retrieved 15.02.2026). Maybe LANGFORD system might be more appropriate?

[3] See https://github.com/anabrid/THAThc (retrieved 15.02.2026) for details.

[4] This was used in application note #56, see https://analogparadigm.com/downloads/alpaca_56.pdf (retrieved 15.02.2026).

[5] It is be quite simple to build something similar from scratch using eight integrated multiplier circuits such as the AD633 (see https://www.analog.com/media/en/technical-documentation/data-sheets/ad633.pdf, retrieved 15.02.2026) on a prototyping perf-board.

Analog Computer Applications

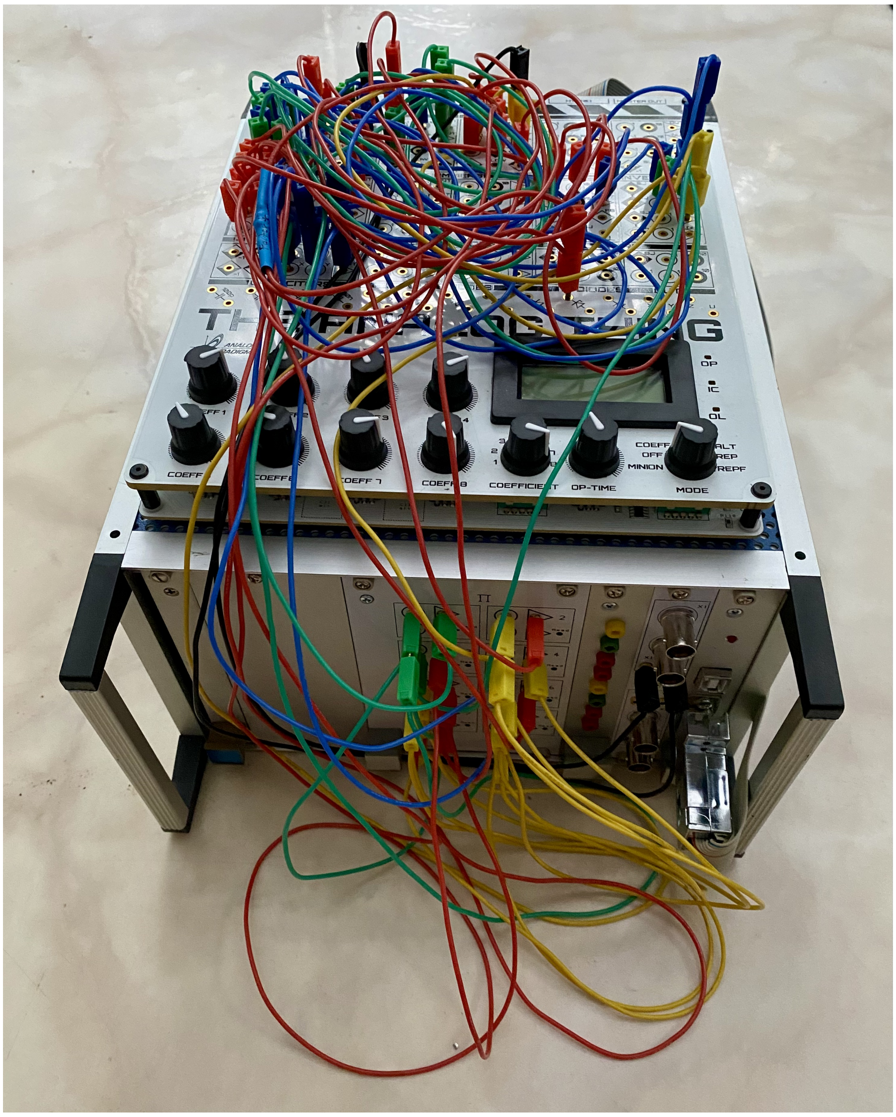

Figure 1: THAT with attached expansion module

Dr. Bernd Ulmann, Issue #65, 15.02.2026

This setup now features ten multipliers, enough to tackle equations (1), (2), and (3). As a quick numerical experiment shows, x, y, and z are all well within the interval $[-2, 2]$, so scaling is simple with $\lambda_x = \lambda_y = \lambda_z = \frac{1}{2}$. The scaling process, which is straightforward, yields

$$\begin{aligned}
\dot{x} &= 2zx - \beta x - \omega y \\
\dot{y} &= \omega x + 2zy - \beta y \\
\dot{z} &= \lambda + z - 1.333z^3 - 8\left(x^2 + y^2\right) - \rho z\left(x^2 + y^2\right) + \varepsilon z x^3
\end{aligned}$$

with the scaled parameters $\beta = 0.7$, $\lambda = 0.3$, $\omega = 3.5$, $\rho = 4$, $0 \le \varepsilon \le 1$, and $x(0) \approx 0.1$.

Unfortunately, this requires ten coefficients, while the THAT only features eight. Thus, ρ has been implemented as an external resistor of 250k connected to the z-integrator, while $x(0) \approx 0.1$ is generated with a summer with a negative feedback of 10 (connect an input with weight 10 to the output of the summer) and an input value of $+1$. This leaves eight coefficients which perfectly fits the THAT. The factors of 2 are implemented by using two inputs with weight 1 each on the x- and y-integrators instead of two dedicated coefficient potentiometers.

The products required are x^2, x^3, y^2, zx, zy, z^2, $-z^3$, zx^3, and $z\left(-\left(x^2 + y^2\right)\right)$, requiring both multipliers of the THAT as well as seven multipliers in the expansion chassis. The analog computer circuit can be directly patched from the scaled equations above and these product terms. The program works very well and yields beautiful results as shown in the x/z phase space plot in figure 2.

References

[LANGFORD 1984] W. F. LANGFORD, "Numerical Studies of Torus Bifurcations", in *International Series on Numerical Mathematics*, Vol. 70, 1984, pp. 285–295

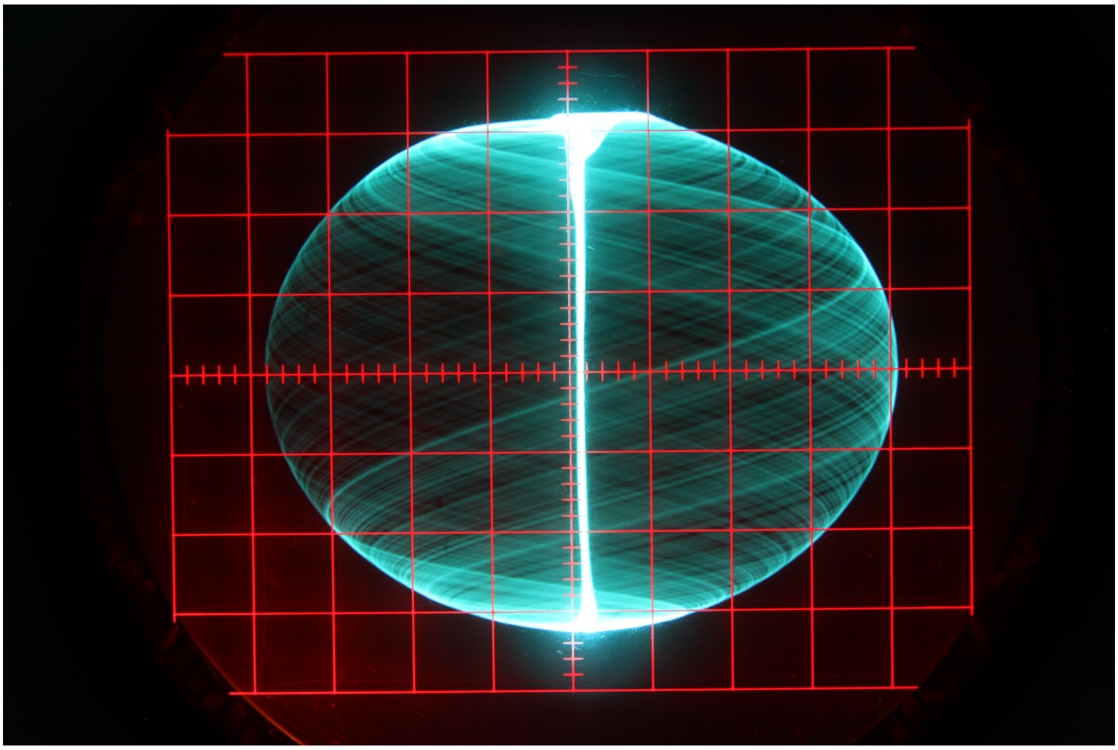

Figure 2: Typical x/z phase space plot of the Aizawa system