The Aizawa system on REDAC

This application note is the first to demonstrate an actual program for the REDAC, anabrid's large scale programmable analog computer.[1] The system chosen for this demonstration is the AIZAWA system which was already covered in application note #17[2] and is described by the following system of three coupled differential equations:[3]

$$\dot{x} = x(z-\beta) - \delta y,$$
$$\dot{y} = \delta x + y(z-\beta) \text{ and}$$
$$\dot{z} = \gamma + \alpha z - \frac{z^3}{3} + \varepsilon z x^3$$

Scaling these equations to ensure that no variable exceeds the machine units of ± 1 is pretty straight-forward but an error prone process when done manually. The resulting computer setup is quite convoluted and its implementation on a classic analog computer is a time consuming and error prone process. All in all it requires three integrators, four inverters, seven multipliers, and eleven coefficients resulting in a rather convoluted patch panel.

Fortunately, with modern fully reconfigurable analog computers such as the REDAC and anabrid's compiler implementing this particular problem can now be done in a matter of minutes with no manual patching or scaling involved at all. The source code describing this system of coupled differential equations is shown in figure 1.

The compiler parses this input, performs the necessary scaling, and generates the corresponding bitstreams for the configuration of the analog computer, i. e., a representation of the connection graph, all required coefficients as well as the necessary initial conditions for the integrators.

[1]The REDAC system can be expanded to up about 1000 integrators, 500 multipliers, and 8000 coefficients.
[2]See `https://analogparadigm.com/downloads/alpaca_17.pdf`, retrieved 06.02.2026.
[3]A thorough numerical study of the behavior of this particular system can be found in [LANGFORD 1984].

```
let diff[x, t](t) = ( x(t) * ( 2.0 * z(t) - 0.7 ) - 3.5 * y(t) );
let diff[y, t](t) = ( 3.5 * x(t) + y(t) * ( 2.0 * z(t) - 0.7 ) );
let diff[z, t](t) = ( 0.3 + z(t) - 1.3333 * (z(t)*z(t)*z(t)) -
                      2.0 * (x(t) * x(t) + y(t) * y(t)) -
                      z(t) * (x(t) * x(t) + y(t) * y(t)) +
                      0.2 * z(t) * x(t) * x(t) * x(t) );
let x(t: 0) = 0.5;
let y(t: 0) = 0.0;
let z(t: 0) = 0.0;
probe x(t);
probe z(t);
```

Figure 1: Input to the compiler defining the AIZAWA system

This program can now be compiled and run on a system like the REDAC with the following commands and yield an output like the one shown in figure 2:[4]

```
redacc --backend NETWORK --output aizawa.apb aizawa.ana
pybrid redac -h <device_address>  run -c aizawa.apb
```

The world of analog computing has changed a lot in the last couple of years. The classic patch panel has been finally relegated to a museum. Programming these systems no longer differs substantially from programming classic digital computers with seamless integration in a Python environment.

References

[LANGFORD 1984] WILLIAM FINLAY LANGFORD, "Numerical Studies of Torus Bifurcations", in INTERNATIONAL SERIES OF NUMERICAL MATHEMATICS, Vol. 70, 1984 Birkhäuser Verlag Basel, pp. 285–295

[4] redacc is the compiler while pybrid is the Python based interface between the digital and the analog computer.

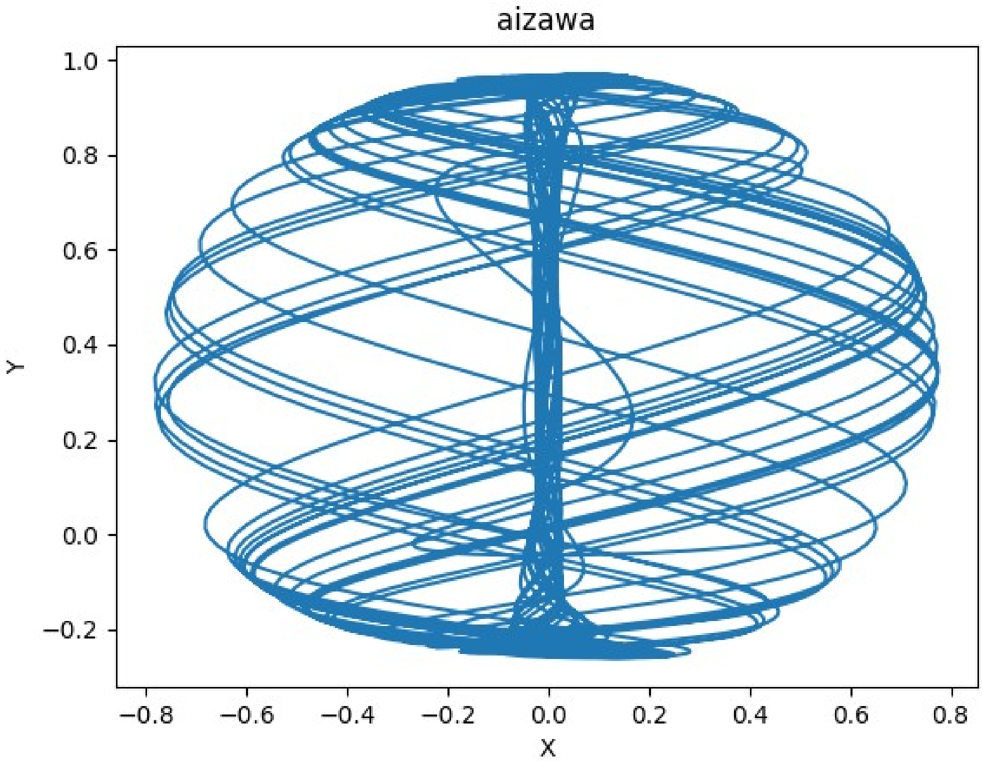

Figure 2: Typical output from the AIZAWA program