

Od: Lorenz-system-based voltages for the Music Thing Modular Workshop System Computer ¹

mcclintic.sphere.fx

January 21, 2025

¹ musicthing.co.uk/workshopsystem/

Od is a program for the Music Thing Modular Workshop System Computer module that simulates the Lorenz system in order to produce loop-able control voltage and pulse signals that display sensitivity to initial conditions.

The Lorenz System

THE LORENZ SYSTEM² IS A SET OF three coupled, first-order ordinary differential equations.

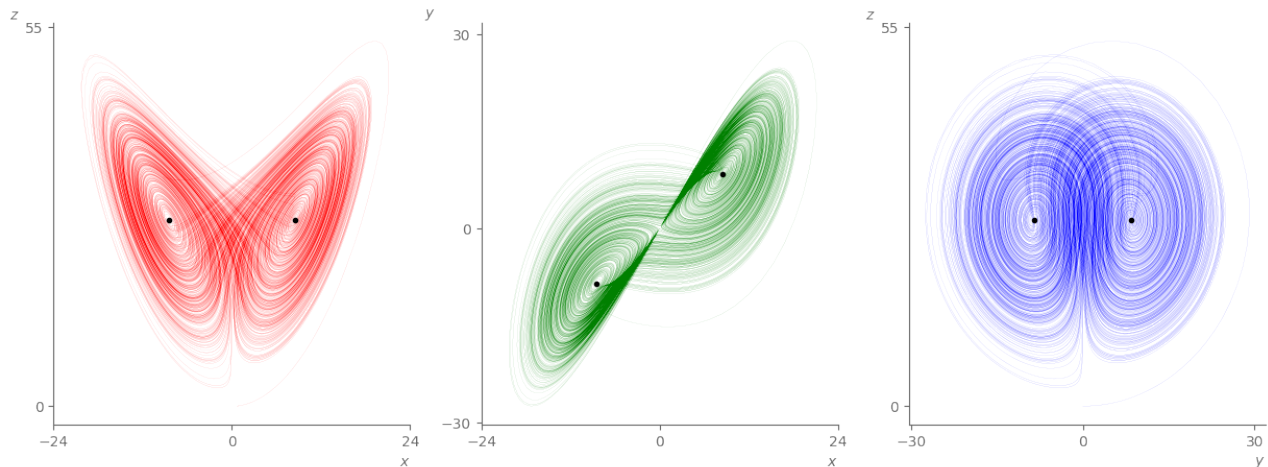
² Deterministic Nonperiodic Flow, Edward N. Lorenz

$$\frac{dx}{dt} = \sigma(y - x)$$

$$\frac{dy}{dt} = x(\rho - z) - y$$

$$\frac{dz}{dt} = xy - \beta z$$

Here, x , y and z are the "coordinates" of the system, their evolution through a three-dimensional space as a function of time are the solutions to the set of differential equations, and will be referred to as "trajectories" herein.



The constants ρ , β and σ determine the overall behavior of the system. For values of $\rho < 1$, all trajectories converge to the origin

Figure 1: Trajectories of the Lorenz attractor projected onto each Cartesian plane. The two fixed points are shown as gray dots.

$(x, y, z = 0)$. At $\rho = 1$, a bifurcation occurs such that for values of $\rho > 1$ there are a pair of fixed point attractors, and all trajectories will evolve towards one of the two.

For the values $\sigma = 10$, $\beta = 8/3$, $\rho = 28$ the system exhibits chaotic behaviour, and this corresponds to the "Lorenz Attractor". Trajectories of this system will propagate forever, never repeating or converging.

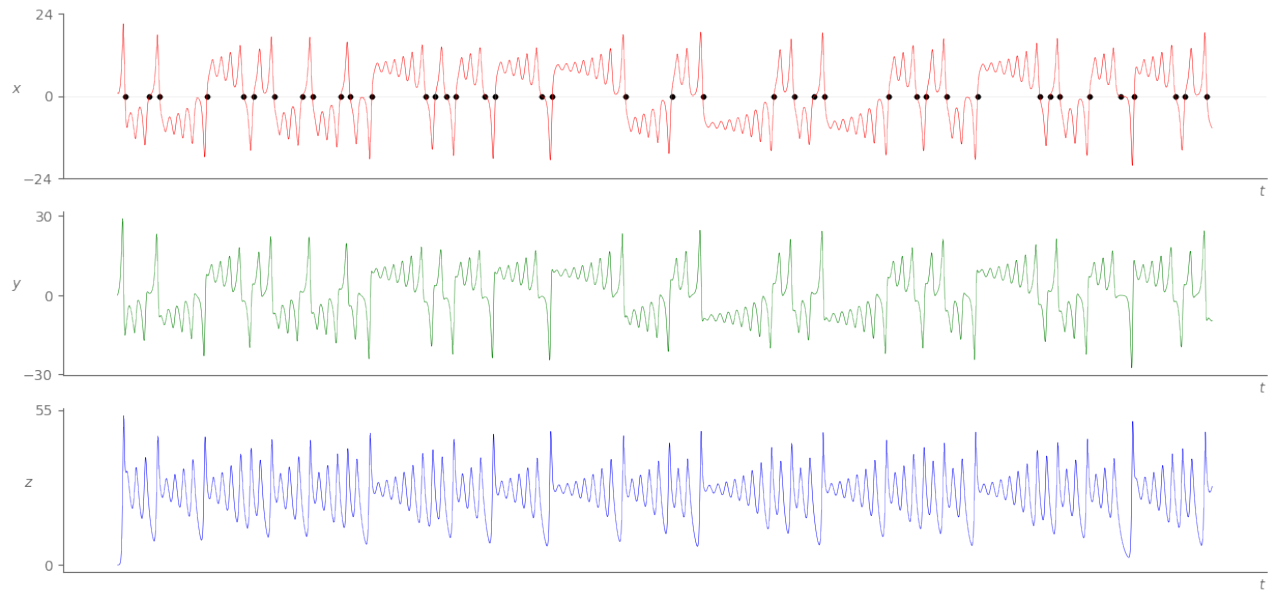


Figure 2: The first 10,000 steps of trajectories of the Lorenz attractor along each axis as a function of time. Note the similarity of the x and y traces. The x and z traces correspond to the CV outputs of the module. Times at which the x -coordinate crosses zero are marked with black dots.

Solving the System

The solution of the system of ODEs is an "initial value problem"³. That is, the coordinates of an initial point are given, and the goal is (in general) to find the values x , y , z at some discrete list of future time values. In this specific case, the goal is to be able to continually determine the next step in the trajectory so that the functions x , y and z can be exported as control voltage via the Computer's CV output sockets. The simplest approach to solution of an initial value problem is the "Euler method"⁴, which discretizes time into individual constant steps, Δt , allowing the next point on a trajectory to be obtained via a simple formula.

$$x_{n+1} = x_n + \Delta t f_x(t, x, y, z)$$

The error introduced by this approach is on the order of Δt^2 . For most scientific applications, this approach is woefully poor. However, when generating solutions for musical purposes, trajectories that

⁴ find a citation

have the appropriate character are sufficient, and the simplest (and cheapest) approach can be relied upon.

The Module

The module itself utilizes a pair of Lorenz systems (A and B) which run concurrently at different rates. Inspired by the øchd module⁵, the relative rate is fixed at a musically engaging ratio (in this case, $\Delta t_{fast} = \frac{11}{5} \Delta t_{slow}$). The overall rate is controlled by the main knob and the right-most CV/Audio input.

⁵ <https://www.instruomodular.com/product/ochd/>

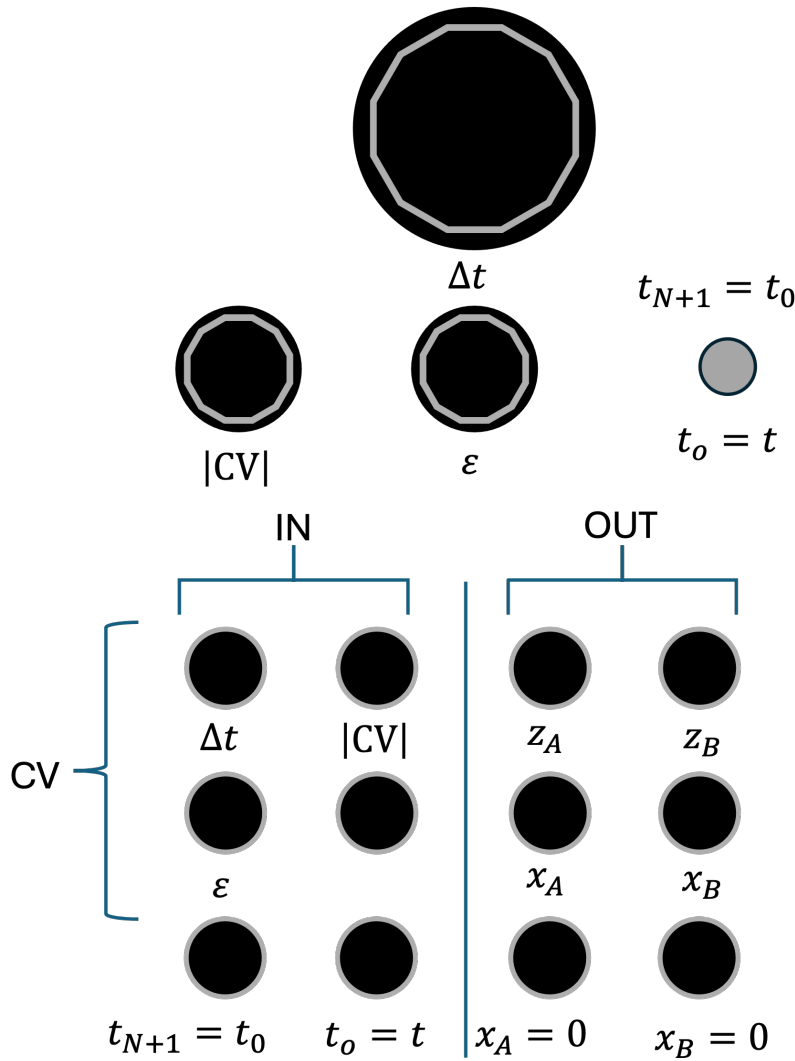


Figure 3: The labels of the Computer's faceplate. The CV inputs control properties shared by the two Lorenz systems, while the pulse inputs allow control of the looping behaviour. The CV outputs provide the coordinates of the trajectories of A and B , while the pulse outputs fire when the trajectories of A and B cross zero along the x -axis.

Control Voltage

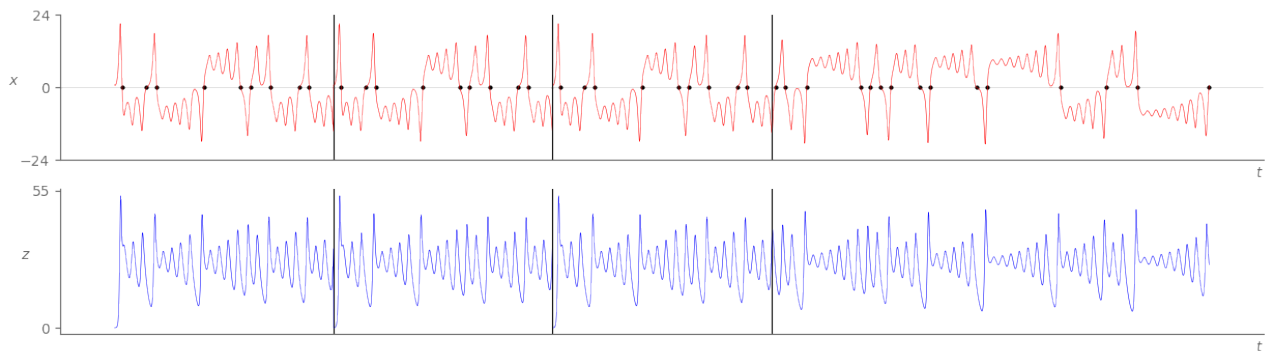
The trajectories of the system evolve according to the specified rate for A , and are made available as bipolar control voltage signals at the CV and CV/Audio output sockets of the Computer, using the full available Computer range of approximately -6 to 6 V. Two Lorenz systems produce three trajectories each for a total of six, however only four can be sent out of Computer. Given the strong similarity between the trajectories of x and y (see Fig. 2), variables x and z are chosen as the output for each system, being between approximately -6 and +6 volts. This allows plotting of the classic Lorenz butterfly in two dimensions, and the creation of sound pictures of the same. The X-knob acts as an overall attenuverter for the four CV outputs.

Pulses

The two pulse outputs of the computer fire whenever the x -coordinate of the associated Lorenz system crosses zero. Examining the plot of the trajectories shows that this will produce a rhythm that feels random, although it is completely deterministic. The combination of zero-crossing triggers from the two Lorenz systems will provide contrasting fast and slow rhythms, similar in the sense that they re deterministically to the output of Rollz⁶ circuits. The top row of LEDs is used to show when the pulses are fired, left is the slower system and right the faster system.

⁶ add reference

Looping



It seems ignorant to implement a module using the Lorenz system without making use of the fact that these dynamical systems are highly sensitive to initial conditions. This implies that two trajectories whose starting positions differ by a small amount will initially behave similarly, but over time will drift apart. This suggests the

Figure 4: Looping the two Lorenz systems (without sensitivity to initial conditions). The simulation starts at an arbitrary t_0 and begins looping at t_n , marked with a black line. After the second repeat, looping ends and the trajectory evolves freely.

idea of returning to the start of a trajectory, but perturbing the initial position before computing the trajectory. The magnitude of this perturbation will determine for how long the first and second solutions of the system will remain similar before diverging.

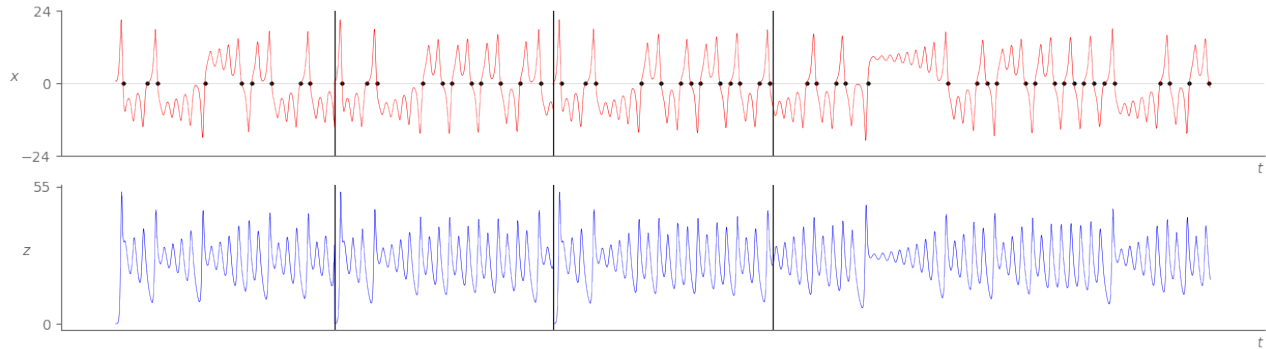


Figure 5: Looping the two Lorenz systems with sensitivity to initial conditions.

This idea is also based on the concept that repeating randomness becomes not random at all. While the trajectories are not "random" in any sense - they are in fact completely deterministic - they are not repetitive in an obvious way and give a sense of back-and-forth evolution. It is not predictable to a listener when or how quickly the system will move from one attractor to another, or how long an orbit on an attractor will last. However, if a chunk of the trajectory is repeated over a short enough time range, the repetition will become obvious to the listener. Thus, we can take an unpredictable system and make it predictable by looping a section of a trajectory between an initial position x_0, y_0, z_0 and a final position x_N, y_N, z_N where N is the number of steps taken between the initial and final positions. Given this loop, the sensitivity to initial conditions can be made use of. By allowing the musician to control the magnitude of random noise added to the initial position, the loop can be tuned to be either the same every time ($\zeta = 0$), or to diverge on each repeat to a desired degree. This opens the door to a fuzzier definition of repetition where the loop is "similar enough" each time for the listener to detect repetition, but different enough to remain somewhat unpredictable, at the direction of the musician. This effect will also be felt on the pulse outputs. The Y-knob is used to set the magnitude of randomness.

The start point is set by sending a trigger into the leftmost CV input or clicking the Z-switch down. This will store the position at which the trigger (or switch click) was detected as r_0 , and begin counting steps on the trajectory. Looping can be controlled either by a trigger at the rightmost pulse input or by setting the Z-switch to the up position (middle corresponds to turning looping off). The goal of including the switch is playability of the loop, and the trigger en-

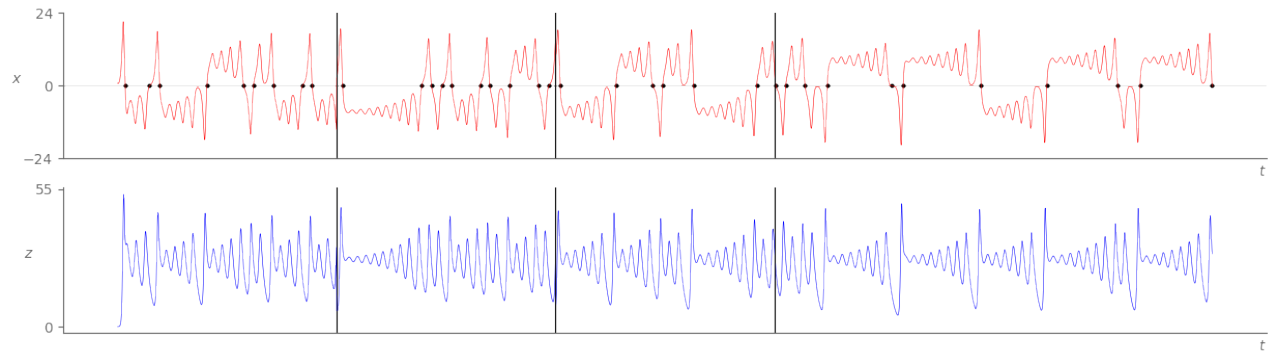


Figure 6: Looping the two Lorenz systems with a high degree of sensitivity to initial conditions.

ables some amount of repeatability. When looping is turned on, the position at the current time is used as r_N and the system will immediately return to r_0 , evolving until the number of steps N has elapsed then returning to the position at t_0 . Turning looping off un-loops the time circle back to a flat line and the system will continue to freely evolve without affecting t_0 ; t_N is forgotten at the point looping is turned off.

The center row of LEDs shows the loop status. The leftmost LED is lit when looping is enabled (NB this is redundant with looking at the switch), and the rightmost flashes when the loop boundary is crossed (i.e. at the simultaneous start/end of the loop).

NB: maybe it would be nice if you could always return to your exact same loop after meandering, i.e after the first time you start a loop t_N would be retained unless explicitly altered, but would need to decide how (switch click?)

Future Work

The implementation of Od takes care to avoid stable solutions of the Lorenz system (i.e. those with $\rho < 1$, or $\rho > 1$ where the stability test is able to confirm stability of the system. This is done to ensure the outputs of the module are chaotic. However, stable solutions could provide interesting, loop-able decaying oscillatory envelopes. This could be incorporated in Od or act as the basis for a separate module or half-program of Computer.

The assumption that the Euler method is sufficient for solving the Lorenz system warrants further investigation, and the implementation of a method with smaller error could be attempted.