

Chaotic Circuit Dynamics

Matthew Beckler

Lab Partners:
Vanessa Cheesbrough, David Toyli

April 14, 2005

1 Introduction

When researching a project for this lab, my lab group and I decided that we wanted to do something out of the ordinary. The project that we developed, which is the subject of this report, certainly was unique. We decided to investigate a class of circuits which are described in a paper titled “Simple chaotic systems and circuits” by J. C. Sprott. In this paper, Sprott describes the history of the search for chaotic behavior in simple differential equations. One of the equations described can be constructed using relatively few simple electronic components, including resistors, capacitors, diodes, and operational amplifiers.

2 Predictions

Based on information and results in Sprott’s paper, we thought that it would be a simple matter of properly constructing the circuit, and observing the results on the oscilloscope. Due to the simplicity of the circuit, and the simple parts used, the circuit was expected to function immediately. We also wanted to compare the results from the circuit with results from a numerical analysis, using a fourth-order Runge-Kutta algorithm. As demonstrated in Sprott’s paper, we expected our oscilloscope traces to look almost exactly the same as the output from the numeric simulation.

3 Description of Experiment

In our experiment, we constructed our circuit to match the schematic given in Sprott’s paper. Initially, we ordered incorrect parts, receiving tiny surface-mount components instead of breadboard sized components. We eventually acquired all the proper parts, and commenced construction. The resistors in the circuit are $1\text{k}\Omega$, the capacitors are $0.1\mu\text{F}$, and the variable resistor varied between $1\text{k}\Omega$ and $2\text{k}\Omega$. As the circuit is fairly simple, we were able to transfer the circuit onto our breadboard relatively quickly. Unfortunately, the circuit did not work, nor did it produce any meaningful output to either oscilloscope or multimeter. We decided to test one of the simple mathematic operator circuits also included in Sprott’s paper, to determine if we were using the operational amplifier chips properly. We decided to construct the basic integrator, of which there are three such integrators in our full circuit. We were unable to get even this simple circuit sub-element working. We tried many combinations of resistors and capacitors, of varying sizes and orientations, but nothing seemed to

work. We tried changing the voltage on our power supply, but that did nothing. We eventually abandoned the actual circuit, and focused on understanding the mathematics behind the circuit and equation, and also the mathematical model we were simulating.

4 Results & Analysis

The equation we decided to model is:

$$\ddot{x} = -A\dot{x} - \dot{x} \pm (|x| - 1) \quad (1)$$

It is called a "Jerk Function" because it is an ordinary differential equation of order three that can be written in the form:

$$\ddot{x} = J(\ddot{x}, \dot{x}, x) \quad (2)$$

Sprott selected this equation by doing a computer search for possible equations, based on this general form:

$$\ddot{x} = a_1\ddot{x} + a_2\phi(\ddot{x}) + a_3\dot{x} + a_4\phi(\dot{x}) + a_5x + a_6\phi(x) + a_7 \quad (3)$$

where $\phi(x)$ is a simple nonlinear function that can be implemented using simple electronic components. Some examples of these functions include $abs(x)$, $max(x, 0)$, $min(x, 0)$, and $-sgn(x)$. Sprott was able to significantly narrow his search parameters by restricting any solution that had $|\ddot{x}|$, $|\dot{x}|$, or $|x|$ exceed 10, a voltage that will usually saturate an operational amplifier. This saturation is a property of the transistors that are used to construct the op amp, but is simply a point where the transistors can supply no more output voltage, resulting in a leveling-off of a graph of output voltage.

During the search, Sprott also calculated what is known as the *largest Lyapunov exponent*. The largest Lyapunov exponent is a fairly standard method to check for chaos. The general idea of calculating the Lyapunov exponent is to follow two close orbits, and calculate their average logarithmic rate of separation. The procedure is fairly simple, but will be omitted. If the reader wishes to read more about the Lyapunov exponent, Sprott has a procedure available electronically at <http://sprott.physics.wisc.edu/chaos/lyapexp.htm>. A positive value for the largest Lyapunov exponent is generally indicative of chaotic behavior. Sprott includes a table of various equations he tried, and their associated largest Lyapunov exponent.

The largest Lyapunov exponent for Eq. (1) was found to be 0.036, which indicates that this equation can exhibit chaotic behavior, provided proper initial values of $(0, 0, 0)$ for (\ddot{x}, \dot{x}, x) respectively, and a proper value for the other parameters, of which there is only A . In Sprott's trials, he used a value of 0.6. This value is adjusted by adjusting the variable resistor in the circuit, using $R = \left(\frac{1}{A}\right) \approx 1.67 \Omega$

As was noted above, the circuit did not work, so all efforts were shifted towards the numeric simulation. In this simulation, we used a fourth-order Runge-Kutta algorithm, to calculate each step of the function. To calculate the y-value of the next iteration, using a step size of h , you first must calculate the

values of k_1 through k_4 , and then combine them with y_n to produce y_{n+1} :

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\ k_4 &= f(t_n + h, y_n + h k_3) \\ y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

We ran our simulation for 100,000 iterations, with $h = 0.001$. The simulation made a plot of \dot{x} vs. x for various values of the variable resistor R. In the images below, the axes are labeled \dot{V} and V for voltage.

In his paper, Sprott indicated that this circuit/equation would only exhibit chaotic behavior if the value of A was ‘valid’. Since the value for A is determined by the variable resistor R, we wanted to see what values of R would produce chaotic behavior, and which values of R would produce either divergent or convergent behavior. The values of R that we wanted to see were values of R for which the values of \dot{x} and x would form a closed path. Here are a few samples of the output from the numeric simulation:

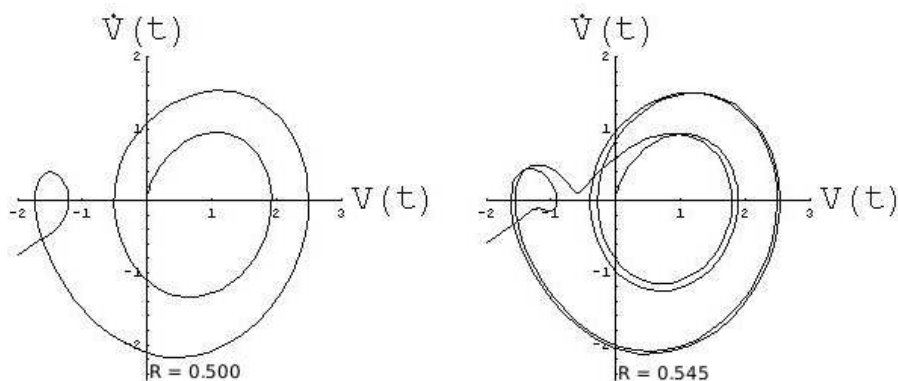


Figure 1: Two examples of divergent behavior

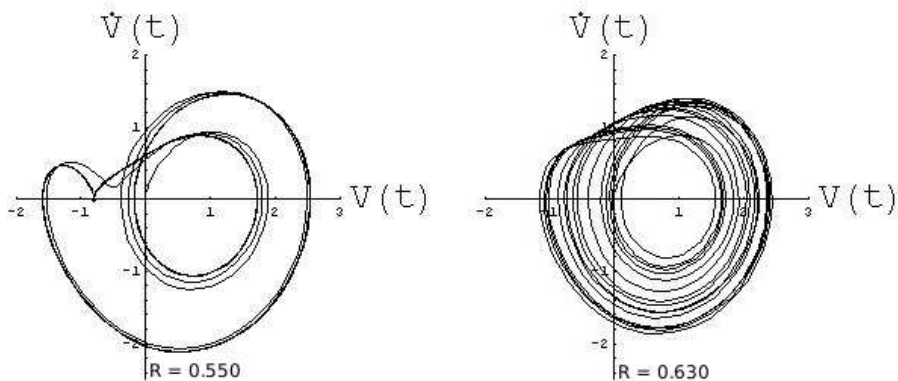


Figure 2: Two examples of closed paths

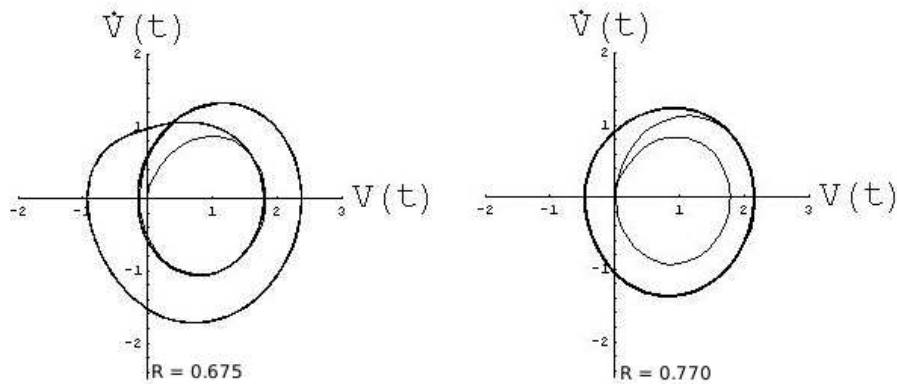


Figure 3: Two sparse closed loops

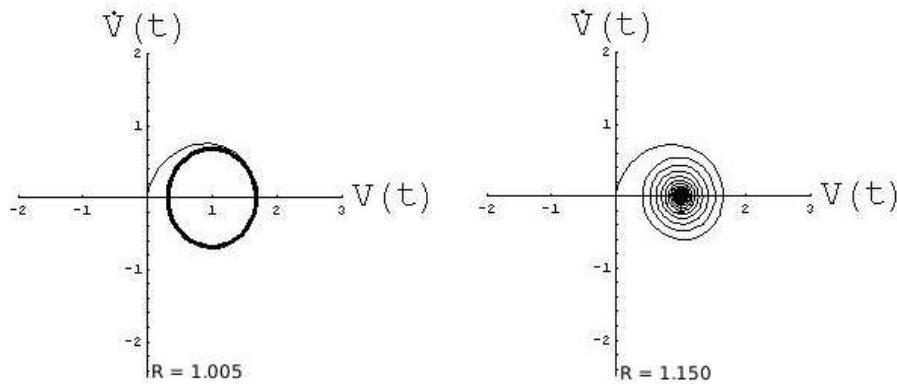


Figure 4: One loop and a converging solution

A computer animation of the evolution of the solution, as R is increased from 0.50 to 1.15 in increments of 0.01, is available electronically at: http://www.tc.umn.edu/~beck0778/videos/chaotic_circuit_video.avi.

5 Interpretation of Graphs

In Figure 1, the graph of $R = 0.500$ is the initial graph of the system. You can see that the solution leaves the localized area, and diverges to infinity. The same is true for all R until 0.550, in Figure 2. Here, this is the first time that the solution to the equation has not diverged outwards, but instead has been trapped in a loop. When we let this simulation run for an extended period of time, the curve did not move, indicating that these loops are the actual solution, and not just a byproduct of numeric approximation. Throughout the range of R -values for which the solution is bounded in a closed loop, there is a great variety of interesting shapes. This shows how a small change in an initial condition, here R , can have a large effect on the end result of the system. This is yet another indication that this system exhibits chaotic behavior. In the rightmost graph of Figure 2, you can see how dense the loops are, especially when compared to the relatively sparse loops in Figure 3. Between $R = 0.770$ and $R = 1.005$, the radius of the main loop decreases until it arrives at the size when $R = 1.005$. It appears that $R = 1.005$ is the other critical value, because for any $R > 1.005$,

the solution converges to a point on the V-axis, as can be seen by the rightmost graph of Figure 4.

6 Error Attribution

There are two possible sources for errors in our numeric simulation. The first type of error is called *Roundoff error*, and is the error associated with the computer's limitation on storage of floating-point numbers. Since computers have a limited amount of memory, some truncation must occur. On a typical 32 bit computer, using a mathematical oriented program, such as *Mathematica*, *Maple*, or *MATLAB*, floating point numbers are usually stored in an 8-byte double-precision floating point variable. This means that the variable will have a 52-bit mantissa, which translates into a precision of $2^{-52} = 2.22 \cdot 10^{-16}$. This is very small compared to the actual values we are working with. However, there is always the possibility that these very small errors can accumulate over many millions of iterations.

The other general type of error when using computers to do numeric simulations is called *Truncation error*, and is caused by using a finite number of steps in a calculation that requires an infinite number of steps to do exactly. In our simulation, the truncation error is introduced because we cannot use an infinitely small step size, and are therefore making a finite approximation to the solution. Since we are still using a relatively small step size (0.001), this error is quite small.

Another factor to consider is the choice of the numeric integration algorithm. As stated before, we used a fourth-order Runge-Kutta algorithm, which is much more accurate than Euler's Method. When comparing the error with various numeric methods, it is important to realize that any truncation error made, no matter how small, can be exponentially magnified in certain cases, leading to inaccurate results. For Euler's method, the Error is approximately proportional to the first power of the step size ($E_h \approx \lambda h^1$). The second-order Runge-Kutta method has $E_h \approx \lambda h^2$. The presence of the power of two on the step size is how this method got the name 'second order'. The method that we used in our simulation, the fourth-order Runge-Kutta method, has $E_h \approx \lambda h^4$, hence the name, 'fourth order'. Due to the differing error estimates, using the fourth-order Runge-Kutta method will have about $\frac{1}{1000}$ the error of Euler's Method.