Non-linear Circuits

Saumya Prakash Sharma-1911151

March 14, 2023

Abstract

Non-linear circuits are circuits that exhibit non-linear behavior due to the presence of non-linear elements such as diodes. These circuits can exhibit complex and unpredictable behavior, which makes their design and analysis challenging. In this lab report, we present the experimental observations of two non-linear circuits, Chhua's circuit, and Feigenbaum circuit. Chhua's circuit exhibits the double scroll attractor, while Feigenbaum's circuit exhibits bifurcations leading to chaos. We conducted the experiments using oscilloscopes, function generators, and resistors. The results obtained were in agreement with the theoretical predictions, and the observations and inferences from the experiments can help in the design and analysis of non-linear circuits.

1 Introduction and Theory

Non-linear circuits are circuits that exhibit non-linear behavior due to the presence of non-linear elements such as diodes and transistors. Unlike linear circuits, the behavior of non-linear circuits cannot be predicted by simple linear equations. Instead, non-linear circuits can exhibit complex and unpredictable behavior, including chaotic behavior. The analysis of non-linear circuits requires the use of mathematical models and simulations. Two examples of non-linear circuits are Chhua's circuit and Feigenbaum circuit. Chhua's circuit consists of three non-linear elements and can exhibit the double scroll attractor, while Feigenbaum's circuit consists of a series of bifurcations leading to chaos.

1.1 Chua's Circuit

Chhua's circuit is a non-linear electronic circuit that exhibits chaotic behavior. It was proposed by Chua et al. in 1993 and consists of three active components: two bipolar junction transistors (BJTs) and one operational amplifier (op-amp), as well as several passive components, including resistors and capacitors.

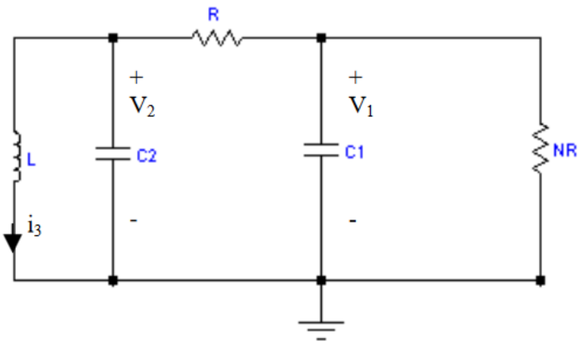

Figure 1: Chua Circuit

If we use Kirchhoff's voltage and current laws, we get the following differential equations for the Chua's circuit;

$$\begin{aligned} C_1\frac{dv_1}{dt} &= \frac{1}{R}\left(v_1 - v_2\right) - f\left(v_1\right) \\ C_2\frac{dv_2}{dt} &= \frac{1}{R}\left(v_1 - v_2\right) + i_3 \\ L\frac{di_3}{dt} &= -v_2 \end{aligned}$$

here, the nonlinear Chua's function, is described by

$$f\left(v_R\right) = m_o v_R + \frac{1}{2}\left(m_1 - m_o\right)\left\{\left|v_R + B_p\right| - \left|v_R - B_p\right|\right\}$$

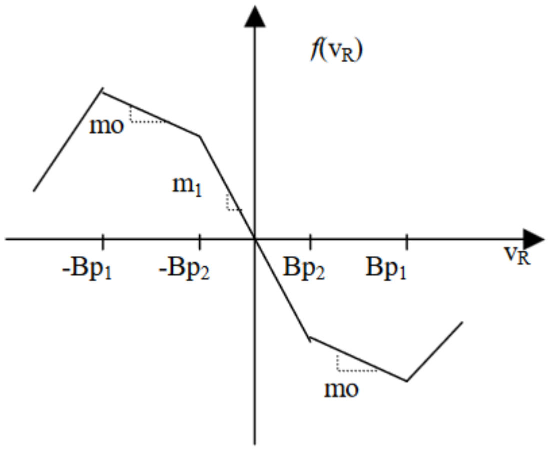

Figure 2: Non-linear function plot for Chua Cirucuit

To make this circuit, we first have to study the negative resistor, which is further used along with the OPAMPS.

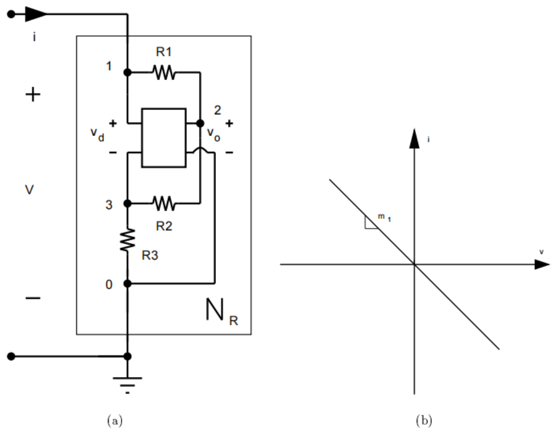

Figure 3: (A) A negative resistance converter, (B) i- v characteristic of the negative resistor

If the voltage controlled voltage source is linear with voltage transfer function $v_o = Av_d$. When A is sufficiently large, this negative resistance convertor has the following $v-i$ relation:

$$i = -\left[\frac{R_2}{R_1 R_3}\right] v$$

By choosing $R_2 = R_1$, this reduces to:

$$i = -\frac{1}{R_3} v$$

The above relation gives a negative resistance.

The slope of the $v-i$ characteristic is given by $m_1 = \left[\frac{(1-A)R_2+R_3}{R_1[R_2+(1+A)R_3]}\right]$; for sufficiently large $A, i \approx -\frac{R_2}{R_1 R_3} v$.
We have used two negative resistance converters in parallel,. N_{R_1} and N_{R_2}, and called it as N_R of the circuit.

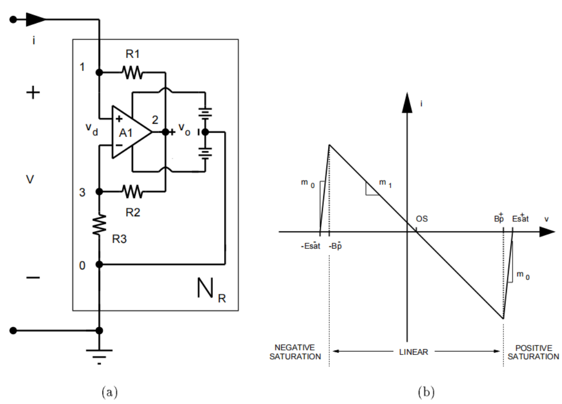

Figure 4: a) Negative resistor using OPAMP, (b) i-v characteristic of opamp negative resistance convertor

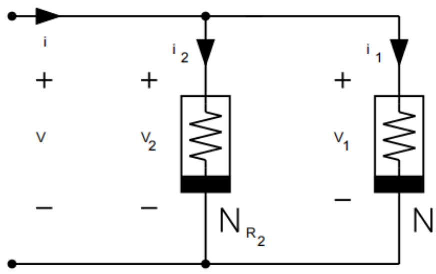

Figure 5: Two negative resistors in parallel

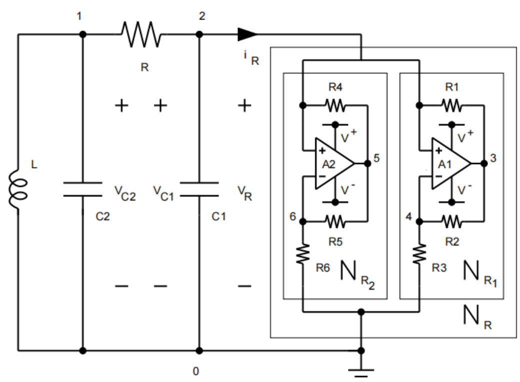

Figure 6: Chua circuit using Opamps and resistors

1.2 Feigenbaum's Bifurcation

Feigenbaum circuit is another non-linear electronic circuit that exhibits chaotic behavior, with bifurcations. It was proposed by Mitchell Feigenbaum in 1978 as a for studying the period-doubling mehtod to chaos in systems.

A bifurcation is a critical point where the qualitative behavior of the system changes as a parameter is varied.
The Feigenbaum bifurcation is a type of period-doubling bifurcation that occurs in non-linear dynamical systems. It occurs when a system undergoes successive bifurcations that lead to a period-doubling cascade, in which the period of the system doubles at each step.

One can detect chaos by plotting a quadratic function dependent on a parameter,say, r,

$$x_{n+1} = rx_n\left(1 - x_n\right)$$

The above function is a logistic map and can be modelled to depict dynamical systems, where, ' r ' is a chaos parameter. One example where this model is valid is the population increment. Here, r determines how population in the future is going to be, stabilize or undergo chaos.

While doing the experiment, the value of r is constantly changed to observe the bifurcation.

2 Observations

2.1 Chua Circuit

Non-linear resistor:

Non-linear resistors, N_{R1} and N_{R2} are made using IC TL082. We have made two separate -segment non-resistor parts to generate a 5 segment non-linearity. The 3 segment non-linear curves obtained are as follows:

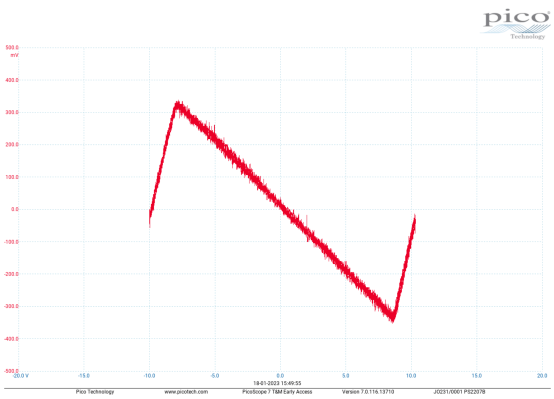

Figure 8: Nonlinear curve 1, from N_{R1}

After a certain value of r the population curve bifurcates and then it goes on and on forming what is called as the feigenbaum tree.

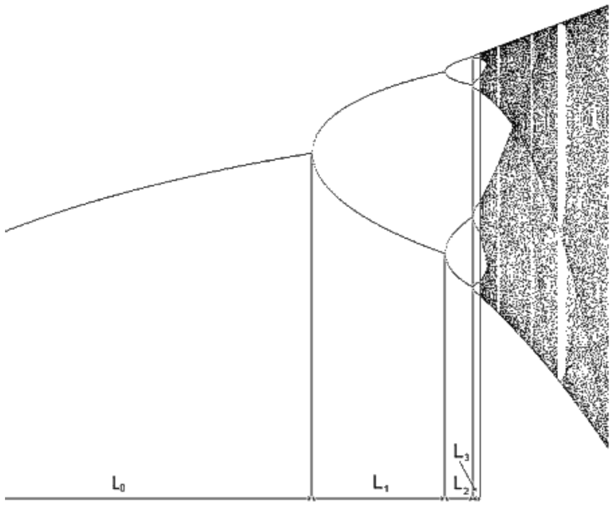

Figure 7: Feigenbaum Bifurcation

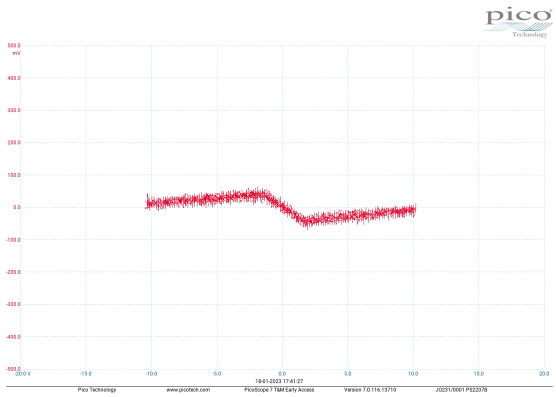

Figure 9: Non-linear resistor curve 2, from N_{R2}

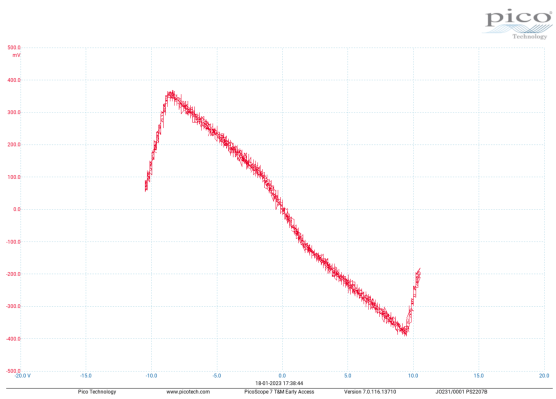

Figure 10: 5 segment non-linear curve

Chua circuit plots: We got the double scroll attractor as shown in the figure.

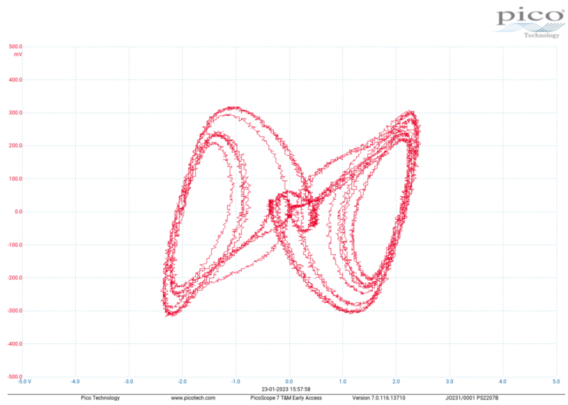

Figure 11: Double scroll attractor with acquisition time 5ns

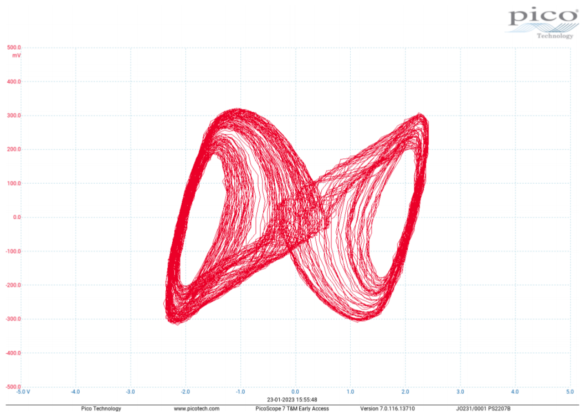

Figure 12: Double scroll attractor with acquisition time 10 ns

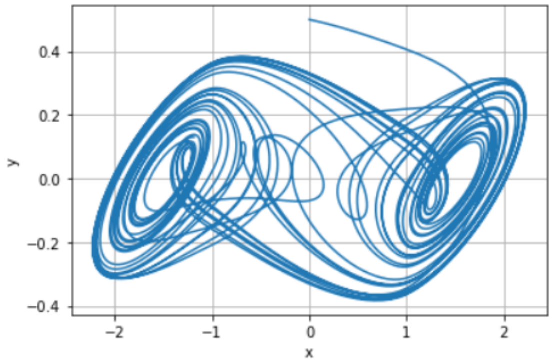

Figure 13: Chua circuit simulated using python

3 Feigenbaum Bifurcation circuit

we have used AD633 ICs as multipliers and LF398 Ics make up the sample and hold part so as to hold the circuit between each iteration of the function. We used ExpEyes-17 to generate a square wave instead of using traditional IC 555 timer.

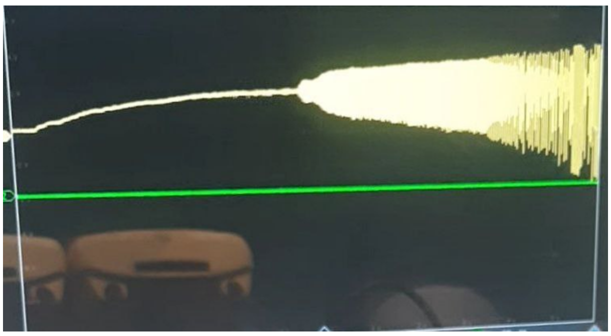

Figure 14: Feigenbaum Bifurcation curve 1

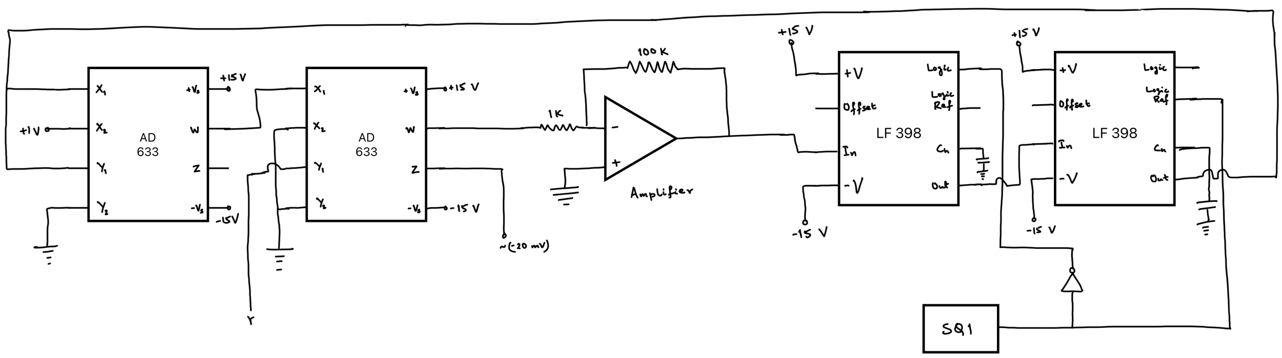

Figure 15: Feigenbaum Circuit

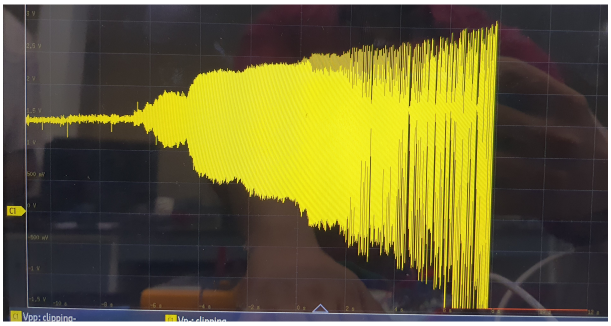

Figure 16: Bifurcation curve 2

The bifurcations for the circuit are observed at 3 , 3.41, 3.5.
So, the Feigenbaum constant for the above circuit is found out to be,
$= \frac{3.5-3.41}{3.41-13.1} = 4.56$

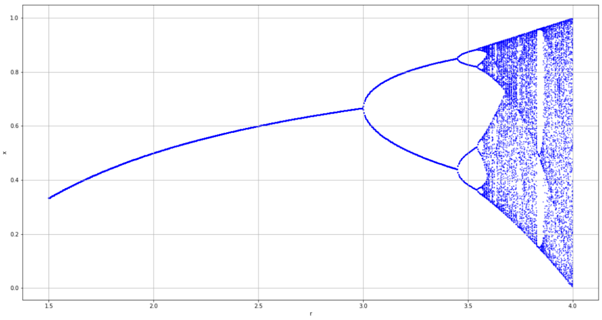

Figure 17: Bifurcation curve simulated using Python

The bifurcations for the circuit obtained from the python simulation are at 3, 3.45, 3.55. So, the Feigenbaum constant for the simulation is found out to be, $= \frac{3.45-3}{3.55-3.45} = 4.5$

We have simulated another finite-difference equation and obtained the bifurcation plot.

$$x_{i+1} = f\left(x_i\right) = 1 - ax_i^2.$$

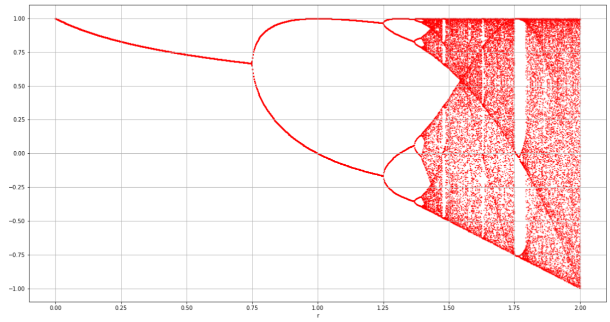

Figure 18: Bifurcation diagram for the above equation simulated using python

The first bifurcation for this curve occurs at $a = 0.75$. The Feigenbaum constant $= \frac{1.25-0.75}{1.37-1.25}$ $= 4.167$.

4 Results and Discussion

- For the Chua circuit, we have made 2 different non-linear resistors that resulted in respective 3 segment non-linear curves. They are represented as N_{R1}, and N_{R2}. The value of resistors used are $R_1 = R_2 = 220\Omega$, $R_3 = 2.2$ kΩ

- For N_{R2}, $R_1 = R_2 = 22$ kΩ, and $R_3 = 3.3$ kΩ. The value of capacitors are 10 and 100 nano Farads, and the value of the inductor is 18mH.

- N_{R1} and N_{R2} are connected in parallel to get the 5-segment non-linear characteristic curve.

- We have used a potentiometer for the resistance. On adjusting the potentiometer, we get a single then, a double scroll attractor. The outputs are observed in an oscilloscope and a pico-scope.

- For the Feigenbaum circuit, we have used Expeyes to generate a square wave instead of using a IC 555 timer. By using expeyes, we could also supply small signal to the circuit input whenever required without disturbing other sources.

- We used the function generator to provide the *ramp* signal.

- We have re-built the circuit and eliminated multiple components. For example, instead of using two different amplifiers of amplification 10, we have used a single amplifier of amplification value 100.

- The Feigenbaum constant for the circuit is found to be 4.56 from the data obtained from the oscilloscope and 4.5 from the simulated curve.

- Bifurcation simulation of another equation results in a constant of 4.167.

- This shows that Feigenbaum constant remains same for different logistic maps.

5 Conclusion

- The Chua circuit and Feigenbaum bifurcation circuit has been studied properly. The double scroll attractor and the bifurcation curves haven been obtained.
- The output signal was chaotic with a double scroll attractor, for Chua circuit.
- The output signal exhibited bifurcation that led to chaos in Feigenbaum's circuit.

6 Appendix

```
import numpy as np
import matplotlib.pyplot as plt

r_a = np.linspace(1.5, 5.0, 1000) # r parameter
x0 = 0.2 #initial value

x_points = []
r_points = []


def f(r):
    p = [x0]
    i = 0
    while i < 1000:
        p.append(r*p[i]*(1-p[i])) # the logistic equation
        i += 1
    return p[901:]

for r in r_a:            # collecting all data points to plot
    x_points.append(f(r))
    r_points.append([r] * 100)



plt.figure(figsize=(19, 10))
plt.grid(True)
plt.xlabel('r')
plt.ylabel('x')
plt.scatter(r_points, x_points, color='blue', s=0.5)
plt.show()
```

Figure 19: Python code for solving the differential equations for Chua circuit

```
: import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import ode


#Scroll attractor parameters. These are ratio of conductance and capacitance v
slp1= - 1.14
slp2 = -0.71
alpha = 15.3
# alpha = 15.1
beta = 27

func = lambda x: slp2 * x + ( (slp1 - slp2) * (abs(x + 1) - abs(x - 1)) / 2)

#Chua differential equations
def chua(t, d):
    dc1_dot = alpha * (v[1] - v[0] - func(v[0]))
    dc2_dot = v[0] - v[1] + v[2]
    dc3_dot = - beta * v[1]
    return [dC1_dot, dc2_dot, dc3_dot]


#intialization parameters
s0 = [0.0, 0.5, 0.0]
t0 = 0
val = ode(derivative).set_integrator('dop853')
val.set_initial_value(s0, t0).set_f_params(1)

t_max = 80
dt = 0.02


#Number of steps
revs = int(t_max / dt) + 2
x = np.zeros(revs - 1)
y = np.zeros(revs - 1)
z = np.zeros(revs - 1)

x[0], y[0], z[0] = s0

nx = 1
while val.successful() and val.t < t_max:
    val.integrate(val.t+dt)
    x[nx], y[nx], z[nx] = val.y[0:]
    nx = nx + 1


plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.grid()
```

Figure 20: Bifurcation curve simulated using Python