Linear Analysis of Rössler System based on Circuits

Juan S. Cárdenas R.
Student
Universidad EAFIT
Medellín, Colombia
jscardenar@eafit.edu.co

David Plazas E.
Student
Universidad EAFIT
Medellín, Colombia
dplazas@eafit.edu.co

***Abstract*—In this work, the Rössler system is studied: a short introduction about the history of the system is given, as well as some state-of-the-art applications; a circuit implementation is presented based on the literature and translated to a state equation. The first part of this document is focused on give some basic theory, concepts and procedures to make a successful analysis of a linear system, which are applied and presented in the results section: a preliminary validation of the linear system obtained is performed making a comparison with the original Rössler system and an stability analysis is performed based on transfer functions and Bode diagrams. The simulations and the findings showed that the linear system in the selected operation point is stable, likewise intervals for one of the parameters that make the nonlinear and linear model stable as well; furthermore, the closed-loop stability was analyzed and properly validated. A linear approximation to Rössler equations was found with success (under some conditions).**

***Index Terms*—Rössler system, simulation, state equation, dynamic system, transfer function, Bode diagram, stability analysis, linearization, order reduction, discrete dynamics, non-minimum phase system.**

I. Introduction

The system in study was proposed by O.E. Rössler in 1976, as a simplified model with shape and behavior similar to spirals in Lorenz system, which was not fully understood at the time due to the techniques known to study oscillators were not applicable to Lorenz model [1]. The Rössler equations are:

$$\begin{aligned} \dot{x} &= -y - z \\ \dot{y} &= x + ay \\ \dot{z} &= b + z\,(x - c) \end{aligned} \tag{1}$$

Although Rössler affirmed that the system did not have immediate physical interpretation [1], nowadays some applications can be found using the model as a mechanism and not as an abstraction of a physical system. The model presented has been used as a tool for image cryptography as it was shown by Mandal *et al.* in [2]; in further work, Laiphrakpam and Khumanthem proposed improvements to Mandal's algorithm, as it is shown in [3]. On the other hand, coupled Rössler system with different inputs have been used to measure the correlation of time series, as Weule *et al.* showed in [4].

In order to bring the system to the real world, Rössler equations can be represented by a circuit, as Canals *et al.* show in [5]. The proposed circuit is shown in Fig. 1 and can be translated to

$$\begin{aligned} RC\dot{x} &= -y - z \\ RC\dot{y} &= x + ay \\ RC\dot{z} &= b + z\,(x - c) \end{aligned} \qquad \begin{aligned} a &= \frac{100k\Omega}{R_a} \\ b &= V_{cc}\frac{100k\Omega}{R_b} \\ c &= \frac{100k\Omega}{R_c} \end{aligned} \tag{2}$$

In [5] they use this circuit to generate true random numbers using the output of the voltage of the node z. The nodes x and y have a fixed frequency of oscillation if the other variable is set to 0, since their rate of change are linear. In contrast, z induces chaos to the circuit, due to its nonlinear behavior. In this manner, this variable was selected to be the output as its chaotic behavior is useful to generate random numbers [5].

In previous work [6], it was found that the Rössler system is stable for large input values. Based on this, it will be attempted to analyze this system as a linear model.

In this work, the question "which linear model can successfully represent the Rössler circuit for certain input and parameters?". It is difficult to give a hypothetical model without prior linear analysis, although it is believed that this model to be found will represent the system behavior, at least, in stationary state.

In order to achieve this, this work will begin linearizing the system in an equilibrium point, comparing the time response for both systems, close and far away from the operation point. Then, the respective continuous and discrete transfer function will be obtained and compared with the continuous linear model. It will be attempted to reduce the model's order using both analytic and software methods and compare this models with the linearized Rössler system; on the other hand, an stability analysis will be performed, using Ruth-Hurwitz criteria and root locus method. Finally a frequency response analysis will be performed using Bode diagrams of the original linearized model and the reduced ones; based on the Bode diagram for the linearized system, the closed-loop stability will be determined using phase and gain margins.

In section II, the dynamic system used and some theory required to make the linear analysis of the Rössler system can be found. In section III, the results of every procedure are presented and applied to the Rössler system. In section IV,

the analysis of the obtained results and their justification is made. Finally, the conclusions are presented in section V.

II. Methods

A. Dynamic System

In the circuit presented, according to Canals *et al.* [5], the variables x, y and z represent the voltages through the nodes shown in Fig. 1. The RC parameter defines the system's time (in seconds), the supplied voltages are $V_{cc} = 15V$ and $V_{ss} = -15V$; R_a, R_b y R_c are resistors in $k\Omega$ used to calculate the system's original parameters, as shown in (2).

The circuit can be represented through a dynamic equation, as follows:

$$\begin{cases} \dot{x_1} = \dfrac{1}{RC}\left(-x_2 - x_3\right) \\ \dot{x_2} = \dfrac{1}{RC}\left(x_1 + \dfrac{100k\Omega}{R_a}x_2\right) \\ \dot{x_3} = \dfrac{1}{RC}\left[\left(V_{cc0} + u\left(t\right)\right)\dfrac{100k\Omega}{R_b} + x_3\left(x_1 - \dfrac{100k\Omega}{R_c}\right)\right] \\ y = x_3 \end{cases} \tag{3}$$

where y is the output and u the input; note that the parameter V_{cc} was selected as input, based on equation (2), thus we select an initial value V_{cc0} and add the input $u\left(t\right)$ in Volts. For the rest of this document, the state variable x_3 can sometimes be referred as y, since it has been chosen as the output.

B. Simulation Diagram

Using equation system (3), a simulation diagram was constructed using *Simulink*, as Fig. 2 shows.

C. Linearity Curve

The linearity curve is a tool that helps determine the range where the linear system can provide a suitable approximation to the stable nonlinear system. The linearity curve is obtained by simulating an stable system and registering the final value (stationary state) for a constant input, then the input is increased in a constant factor and then simulated until a new final value is found and so forth.

For the system in study, the following simulation diagram (Fig. 3) was constructed, using *Simulink*, in order to obtain the desired curve. As this figure shows, both systems are simulated with the same input. This one begins with a clock that should represent a continuous "ramp" over time, which is first scaled by a factor of $2/10$ and then passes through a Zero-Order-Hold (ZOH). This ZOH ensures that the continuous "ramp" is transformed into a discrete one, i.e. a "staircase" input since the signal is retained with a sampling time of $T = 100s$. This means that both systems are simulated for $100s$ with a **constant** input. After this, the ZOH passes a new constant signal, where each iteration of the ZOH increases the input by $20V$ (this explains why it is scaled by $2/10$). Note that the initial input value $1000V$ has been included in the nonlinear simulation diagram, as Fig. 2 shows.

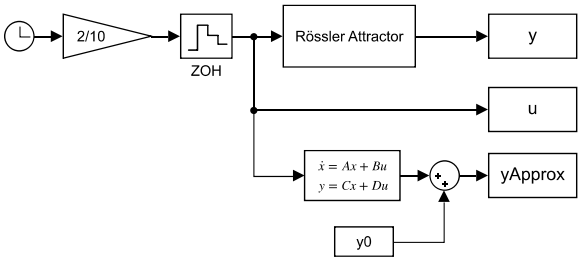


Figure 3: Simulation diagram for linearity curve.

D. Linearization

In order to linearize the system in study, the classical procedure will be performed. For more detailed notes on this method, see [7, pp. 21-27]. Given a nonlinear dynamic system

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= g(\mathbf{x}, \mathbf{u}) \end{aligned} \tag{4}$$

We can obtain a linearization of this system around an operation point $(\mathbf{x_0}, \mathbf{u_0})$, in the form

$$\begin{aligned} \Delta\dot{\mathbf{x}} &= \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u} \\ \Delta\mathbf{y} &= \mathbf{C}\Delta\mathbf{x} + \mathbf{D}\Delta\mathbf{u} \end{aligned} \tag{5}$$

Where $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x_0}$, $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u_0}$, $\Delta\mathbf{y} = \mathbf{y} - \mathbf{y_0}$; and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are Jacobians of the functions f and g as follows

$$\begin{aligned} \mathbf{A} &= \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x_0}, \mathbf{u_0}) \qquad \mathbf{B} = \frac{\partial f}{\partial \mathbf{u}}(\mathbf{x_0}, \mathbf{u_0}) \\ \mathbf{C} &= \frac{\partial g}{\partial \mathbf{x}}(\mathbf{y_0}, \mathbf{u_0}) \qquad \mathbf{D} = \frac{\partial g}{\partial \mathbf{u}}(\mathbf{y_0}, \mathbf{u_0}) \end{aligned} \tag{6}$$

This procedure can be also performed using *Matlab*, through the command `linmod` with the following syntax: `[A,B,C,D] = linmod(sys,x0,u0)`, where *sys* is a *Simulink* model, and $(\mathbf{x}_0, \mathbf{u}_0)$ is the operation point.

E. Linear and Nonlinear systems comparison

As it has been already depicted, the linearization of a nonlinear model is given in Δ-variables; in order to compare properly the output of a nonlinear system and its linearization, the scheme shown in Fig. 4 can be used. Note that, for the linear system, is required to subtract the initial input $\mathbf{u_0}$ to obtain $\Delta\mathbf{u}$, and the output is given as $\Delta\mathbf{y}$, then the initial value $\mathbf{y_0}$ is added to obtain the standard output $\mathbf{y}(t)$

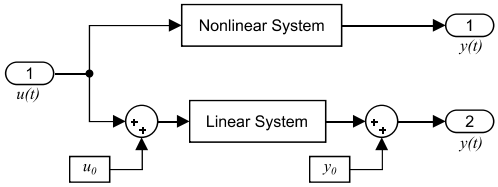


Figure 4: Diagram for comparing a nonlinear system with its linearization.

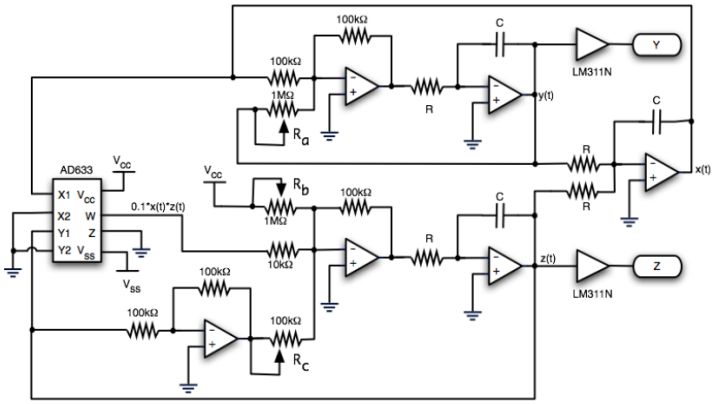


Figure 1: Rössler circuit representation [5].

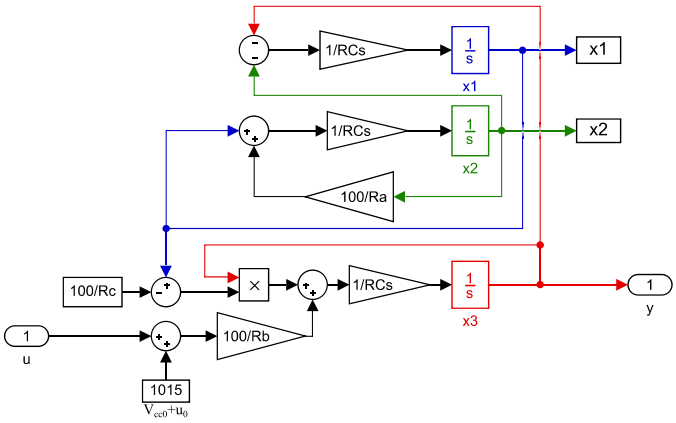


Figure 2: Simulation diagram for Rössler system.

F. Transfer Function

The transfer function is a tool to represent a linear system with null initial conditions. For a single-input single-output (SISO) system, it is defined as the relation between the Laplace transform ($\mathcal{L}$) of the output and the Laplace transform of the input, as follows:

$$G(s) = \frac{Y(s)}{U(s)} \tag{7}$$

Where $Y(s) = \mathcal{L}\{y(t)\}$, $U(s) = \mathcal{L}\{u(t)\}$. For a discrete system, it would be worked with the $\mathcal{Z}$ transform. If the system is in a state-space representation, as in equation (5), the transfer function can be also obtained with

$$G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \tag{8}$$

And, finally, this can be achieved using *Matlab*, using the command `tf(LinSys)`, where `LinSys` is a linear system object (can be created with the command `ss(A,B,C,D)`).

G. Sampling Time

The sampling time is a number that determines how often a continuous signal will be sampled in order to obtain a discrete signal.

In this work, the sample time was originally selected empirically, based on simulation: it was chosen such that the signal is sampled correctly (keeping oscillation period and most transitory behavior) and such that it can be easily evidenced the "retained" signal.

From a theoretical perspective, the sampling time can be obtained from the growth time (T_r) of the system in study. The growth time is defined at the time that the system takes to reach 90% of stationary state from the respective 10%, and then apply the following criterion:

$$\frac{T_r}{10} < T < \frac{T_r}{2} \tag{9}$$

It is important to mention that the sample time selection is a non-trivial procedure, since a bad selection of the sample time can lead to undesired phenomena like the aliasing; for more information regarding the sample theorem and aliasing, refer to [8, pp. 34-54]. In this work, both methods are performed.

H. Discretization of Transfer Functions

Given a continuous transfer function $G(s)$, the discrete transfer function is obtained through the following expression:

$$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\mathcal{L}^{-1}\left\{\frac{G(s)}{s}\right\}_{t\to kT}\right\} \tag{10}$$

Where T is a predefined sampling time for the discretization. This procedure can also be carried out using *Matlab* with the command `c2d(transfer,T)` and, in this case, `transfer` is a continuous transfer function.

I. Ponderation Sequence

The transfer function can be used to obtain a sequence of numbers that, given an input, the respective system output can be acquired. This numbers, in a continuous system, are infinite; thus, this sequence is often calculated for the discrete case.

Let $G(z)$ a discrete transfer function. The discrete ponderation sequence is found from the inverse $\mathcal{Z}$ transform of this transfer function:

$$g(k) = \mathcal{Z}^{-1}\{G(z)\} \tag{11}$$

And the output can be calculated with the following procedure:

$$\begin{aligned} G(z) &= \frac{Y(z)}{U(z)} \\ Y(z) &= G(z)U(z) \\ \mathcal{Z}^{-1}\{Y(z)\} &= \mathcal{Z}^{-1}\{G(z)U(z)\} \\ y(k) &= g(k) * u(k) \end{aligned} \tag{12}$$

Where $*$ represent the convolution product, in this case, discrete.

J. Order Reduction

In order to reduce the system order, two straight-forward methods are often considered; this methods do not require formal procedures.

The first method is to cancel out stable zeros and poles that are close enough to each other and add a gain to assure that the stationary state for both systems is equal. The second method is to cancel out stable poles that are far enough from the dominant pole (at least 10 times the dominant pole) and, as the previous method, add a gain to compensate for the reduction.

On the other hand, in order to reduce the system order with more formal methods, *Matlab* has the command `balred(Linear,order)`, where `Linear` is an state-space model **or** a transfer function, and `order` is the desired order after the reduction.

2nd order approximation:
This method is a slightly more empirical/experimental approach. This method is based on the general formula for 2nd order transfer functions:

$$G(s) = \frac{k\omega_0^2 e^{-s\tau}}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{13}$$

Where ω_0 is the natural frequency of the system, ζ is the damping, k is the gain and τ is the delay. An example is shown in Fig. 5.

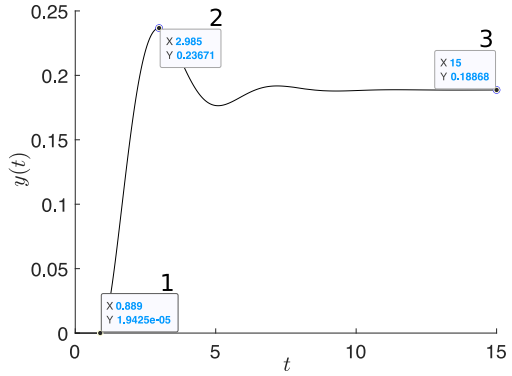


Figure 5: Second order system.

In order to calculate the parameters, points 1, 2 and 3 are required. The first point gives information about the delay τ, the second has information about the peak time t_p and magnitude y_p. Finally, the last one shows the final value y_{ss} (stationary state) and is needed to calculate the gain.

The following expression is used to calculate ζ:

$$\zeta = \frac{1}{\sqrt{1 + \left(\frac{\pi}{\ln Mp}\right)^2}} \tag{14}$$

Where

$$Mp = \frac{y_p - y_{ss}}{y_{ss}} \tag{15}$$

is the maximum overshoot; and through the following equation, ω_0 is obtained:

$$\omega_0 = \frac{\pi}{t_p\sqrt{1-\zeta^2}} \tag{16}$$

K. Stability

The following methods can only be applied to linear systems. For stability of nonlinear systems see reference [9, Ch. 13], for Lyapunov stability.

1) Poles Calculation: The simplest way of knowing whether a linear system is stable is calculating all the poles λ_i and checking if $\mathrm{Re}(\lambda_i) < 0$. In *Matlab*, the poles can be calculated directly from the state-space model or transfer function with the command `pole(Linear)`.

2) Bounded-Input Bounded-Output Criterion (BIBO): Another way to determine if a linear system is stable is through the BIBO criterion [7, Pag. 170]: a bounded input (e.g. bounded waves, step, impulse train, etc.) generates a bounded output. It is important to highlight that this criterion does not necessarily imply that for every bounded input, the output will be bounded, but at least, for sine inputs. Another note is that this method is specially important when poles are exclusively imaginary ($\mathrm{Re}(\lambda_i) = 0$).

3) Routh-Hurwitz Method: The Routh-Hurwitz method works with the characteristic polynomial of a linear system, that can be obtained from the denominator of the transfer function or with $P(s) = |s\mathbf{I} - \mathbf{A}| = 0$.

Consider the general case:

$$P(s) = \sum_{j=0}^{n} a_j s^{n-j} \quad a_0 \neq 0 \tag{17}$$

Necessary Conditions:

- $(\forall j)(a_j > 0)$

Sufficient Conditions:

- The first column of the Routh-Hurwitz array is non-negative.

For a more detailed explanation of the Routh-Hurwitz array, refer to [10, pp. 212-214]. In *Matlab*, an implementation has been developed and can be found in [11].

L. Root Locus

The root locus is a plot of how the roots of a polynomial change for a parameter $k > 0$. It uses a closed-loop transfer function with a gain k that is equivalent to this polynomial.

Given a polynomial with the parameter k in one or more coefficients of the polynomial, it can always be expressed as

$$1 + k\frac{N(s)}{D(s)} = 0 \tag{18}$$

Note that this can be interpreted as a denominator of a closed-loop transfer function. Thus, the associated plant is

$$G(s) = \frac{N(s)}{D(s)} \tag{19}$$

The root locus starts ($k = 0$) in the poles of $G(s)$ and finishes ($k \to \infty$) in the finite and infinite zeros of $G(s)$. The root locus is useful to determine ranges of k where the system associated with the characteristic polynomial $P(s)$ is stable. This procedure can also be used to determine the infinite zeros of a transfer function. The root locus can be obtained as well using *Matlab* using `rlocus(Linear)`, where `Linear` is a state-space model or a transfer function.

M. Bode Diagram

The Bode diagram is a tool to analyze the frequency response. The frequency response is useful to analyze the system output when the input is a sine wave, since the output in stationary state will be another sine wave with equal frequency and different amplitude and phase.

The bode diagram is composed of two plots in logarithmic scale, since it is often desired to analyzed the frequency response for a wide spectrum of frequencies: one for the relative amplitude and another one for the phase of the output (in stationary state). For the amplitude, it is given in decibels (dB) and for the phase, it is given in degrees.

Suppose the input is

$$u(t) = A\sin(\omega t) \tag{20}$$

And the output in stationary state

$$y_{ss}(t) = B\sin(\omega t + \phi) \tag{21}$$

It can be proved (see [10, pp. 399-400]) that

$$B = A|G(i\omega)|$$
$$\phi = \arctan\left(\frac{\mathrm{Im}[G(i\omega)]}{\mathrm{Re}[G(i\omega)]}\right)$$

And the amplitude in decibels is defined as

$$|G(i\omega)|\text{dB} = 20\log_{10}\left(\frac{B}{A}\right) \tag{22}$$

The Bode diagram is useful for several reasons. In first place, it can be used to determine the regions where the system intensifies or reduces the input amplitude. On the other hand, it can be used to determine the ranges of gain and delay where the closed-loop system is stable. Lastly, it can help determine whether the system in study is a minimum phase system.

The bode diagram can be obtained using *Matlab* through the command `bode(Linear)`, where `Linear` is a transfer function or a linear state-space model.

As it was previously mentioned, the Bode diagram can be used to determine closed-loop stability, using the phase and gain margins. For a more detailed explanation regarding the phase and gain margins see [10, pp. 464-468]. With *Matlab*, the margins can be calculated with the command `allmargin(Linear)`, where Linear is a state-space model or a transfer function.

III. Results

A. Equilibrium Points

The system in study has equilibrium points when

$$\begin{cases} -x_2 - x_3 = 0 \\ x_1 + \dfrac{100}{Ra}x_2 = 0 \\ \dfrac{100}{R_b}\left[V_{cc0} + u(t)\right] + x_3\left(x_1 - \dfrac{100}{R_c}\right) = 0 \end{cases} \tag{23}$$

After carrying out some algebraic procedures, it can be proved that the solution is given by

$$\begin{cases} x_1 = \dfrac{100x_3}{R_a} \\ x_2 = -x_3 \\ x_3 = \dfrac{R_a}{2R_c}\left(1 \pm \sqrt{1 - \dfrac{4R_c^2\left[V_{cc0} + u(t)\right]}{R_a R_b}}\right) \end{cases} \tag{24}$$

Note the double sign in x_3, therefore the Rössler system has two equilibrium points. These depend on both the parameters and the input $u(t)$.

In this work, the same parameters studied in [6] will be used. Thus, $R_a = 500k\Omega$, $R_b = 7500k\Omega$, $R_c = 17.5439$, $V_{cc0} = 15V$, $RC = 1$ and input

$$u(t) = (1000V)H(t) \tag{25}$$

where $H(t)$ is the Heaviside or step function. Therefore, the equilibrium points for the Rössler system with said parameters are

$$\begin{aligned} P_1(x_1, x_2, x_3) &= (0.5228, -2.6140, 2.6140) \\ P_2(x_1, x_2, x_3) &= (5.1772, -25.8859, 25.8859) \end{aligned} \tag{26}$$

In this paper, the first equilibrium point P_1 will be considered for the linearization. In Figure 6, the system output with initial conditions in P_1 is shown. Note that the system's state does not change over time, validating the equilibrium point found. This simulation was conducted using fourth-order Runge-Kutta method [12], with fixed step size of 0.001, $t_0 = 0s$ and $t_{\text{end}} = 100s$.

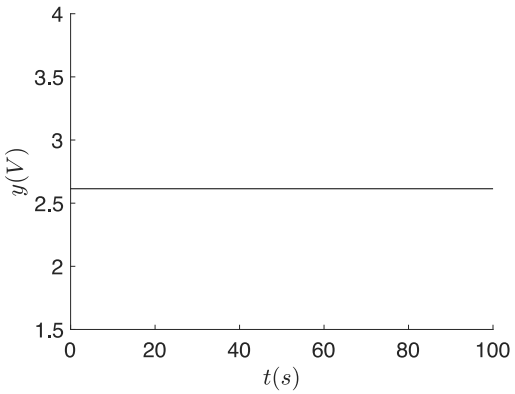


Figure 6: System output with initial conditions at equilibrium point.

B. Linear Model Approximation

For the linearization, equation (3) was used. Let $\mathbf{x} = [x_1(t)\ x_2(t)\ x_3(t)]^T$, $\mathbf{u} = u(t)$ and $\mathbf{y} = y(t)$, as the system is a SISO system. Now, the Jacobians for the linear system are

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{-1}{RC} & \frac{-1}{RC} \\ \frac{1}{RC} & \frac{100}{RCR_a} & 0 \\ \frac{x_3}{RC} & 0 & \frac{1}{RC}\left(x_1 - \frac{100}{RC}\right) \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \frac{100}{RCR_b} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 0 \end{bmatrix}$$

Evaluating at the operation point P_1 from expression (26), the linear system is given by the following state space representation:

$$\begin{aligned} \Delta\dot{\mathbf{x}} &= \begin{bmatrix} 0 & -1 & -1 \\ 1. & 0.2 & 0 \\ 2.614 & 0 & -5.1772 \end{bmatrix} \Delta\mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0.0133 \end{bmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{y} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \Delta\mathbf{x} \end{aligned} \tag{27}$$

The linearization was performed as well using the command `linmod` from Matlab through a simulation diagram, like 2, in Simulink and the same results were obtained.

C. Linearity Curve

In this section we present the linearity curve in order to compare the linear approximation with the nonlinear dynamic system in study. In Fig. 7 this curve can be observed.

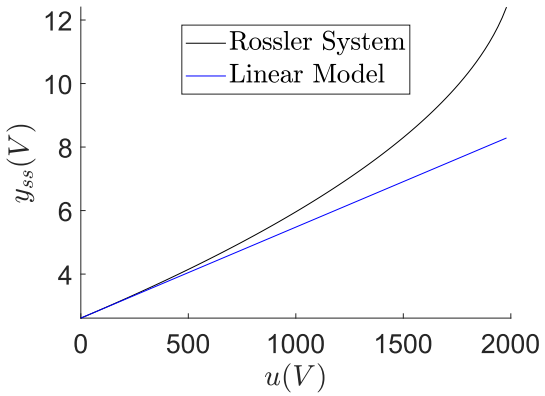


Figure 7: Linearity Curve

D. Comparison: Linear and Nonlinear System Response

In the following results, the same parameters described in section III-A will be used (for both the system and for the simulation).

1) Comparison regarding the Input: The previously selected input (equation 25) will be changed by a factor ε. For the nonlinear system, the input will be

$$u(t) = (1000V + \varepsilon)H(t)$$

whereas, for the linear system, the input will be

$$\Delta\mathbf{u} = \mathbf{u} - \mathbf{u_0} = (1000V + \varepsilon)H(t) - (1000V)H(t) = \varepsilon H(t)$$

The comparison will be performed, first, exactly at the operation point, then close to it and finally far from the initial input.

For a comparison exactly at the operation point, $\varepsilon = 0V$. The results are presented in Fig. 8.

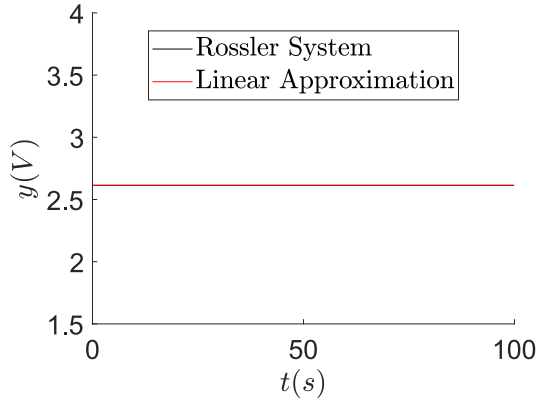


Figure 8: Output for $\varepsilon = 0V$.

For a comparison close to the operation point, $\varepsilon = 50V$. The system output is shown in Fig. 9. Note that the behaviour is almost identical.

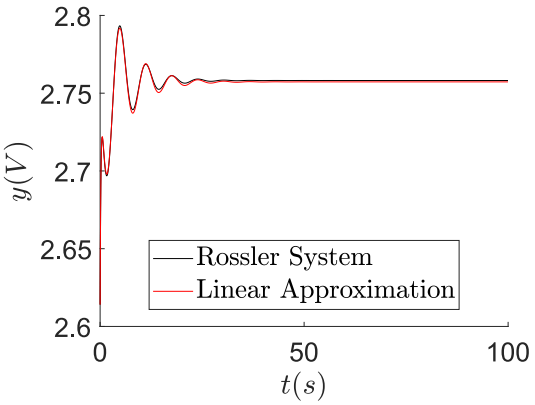


Figure 9: Output for $\varepsilon = 50V$.

For a comparison far from the operation point, $\varepsilon = 200V$. The system output is shown in Fig. 10. Note that the two systems start to differ.

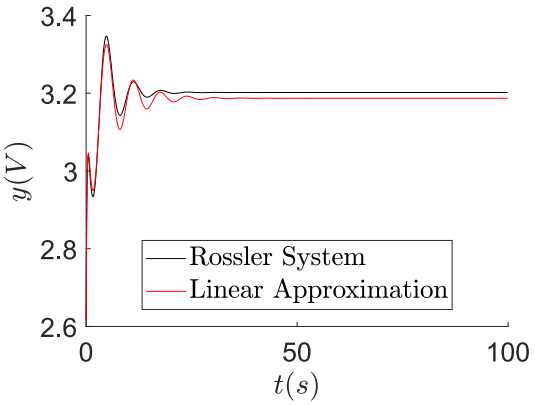


Figure 10: Output for $\varepsilon = 200V$.

In order to show how the systems differ over time with inputs significantly far from the operation point, one last simulation will be shown. For the following results (Fig. 11), $\varepsilon = 750V$ is used; note that the output is notably different: not stabilizing in the same value nor showing the same oscillation.

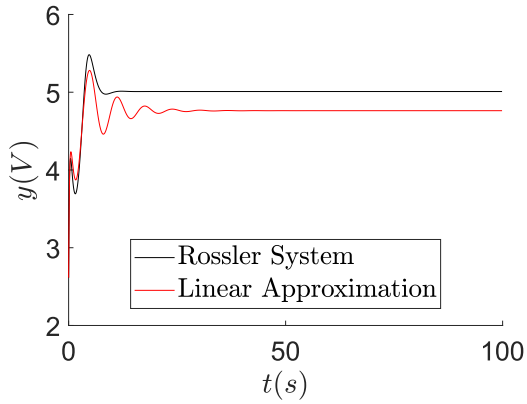


Figure 11: Output for $\varepsilon = 750V$.

One last comparison was performed for a sine input

$$u(t) = (1V)\sin(t) + 1000V \tag{28}$$

which yields the following input for the linear approximation:

$$\Delta\mathbf{u} = (1V)\sin(t) \tag{29}$$

In the following plot (Fig. 12), the systems responses are displayed. Note that both systems overlap; it was expected that the linear model will have the same behavior only in stationary state, but, surprisingly, also in transitory state the response is reproduced.

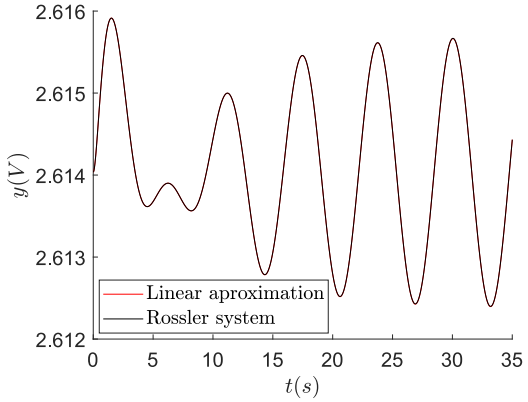


Figure 12: Comparison between the linear and nonlinear model for a sine input.

2) Comparison regarding the Initial Conditions: In this section, the initial conditions for x_3 will be changed using a small, medium and large factor ε, that is $x_3(0) = 2.6140V + \varepsilon$. All simulations were performed with all same parameters from previous section, except for $t_{\text{end}} = 35s$ since the system stabilizes quickly and for $t_{\text{end}} = 100s$ it would show a lot of not relevant behavior.

For initial conditions close (small change) to the operation point, $\varepsilon = 0.1V$. In Fig. 13 the comparison between the linear and the nonlinear is shown. Note that the change is so subtle that the output responses overlap.

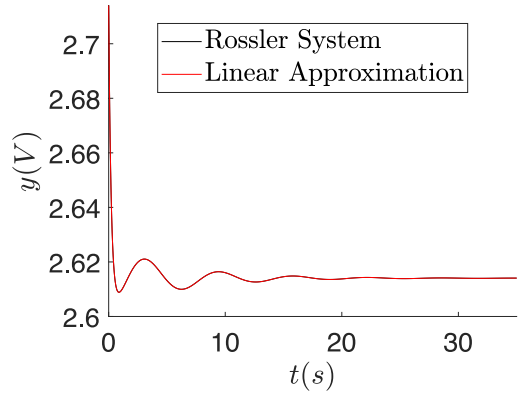


Figure 13: Output for $\varepsilon = 0.1V$ in initial conditions.

For a medium change, $\varepsilon = 50V$ is considered. As it can be observed in Fig. 14, both outputs are similar, only differing in the magnitude of each oscillation.

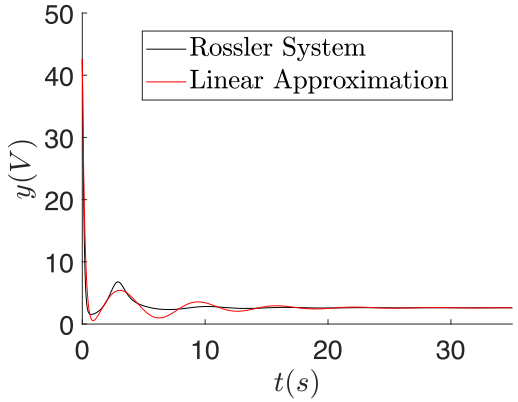


Figure 14: Output for $\varepsilon = 50V$ in initial conditions.

Finally, $\varepsilon = 400V$ was set for a large change in the initial conditions for x_3. The results are displayed in Fig. 15. This plot shows that, at this point, both systems diverge one from another over time.

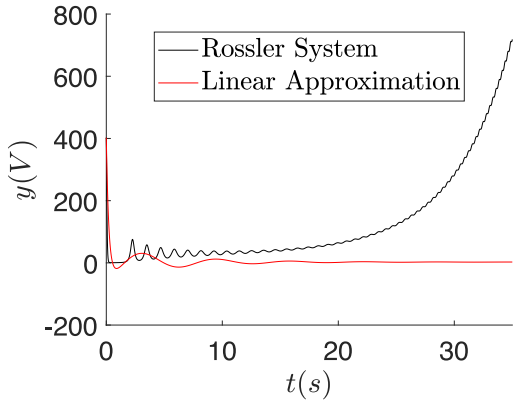


Figure 15: Output for $\varepsilon = 400V$ in initial conditions.

Two last simulations were performed changing all initial conditions and the input. For the first one, making a small change in all parameters: the input was increased by $3V$ and **each** of the initial conditions were changed by adding $0.1V$; the results for this simulation are presented in Fig. 16.

Finally, for a relatively large change in all parameters: the input was increased by $30V$ and $10V$ was added to all initial conditions. The outcome of this simulation can be seen in Fig. 17. Note that the nonlinear Rössler system starts to show a strange behavior, which cannot be reproduced by the linear system.

E. Continuous Transfer Function

Based on the method presented in section II-F, the transfer function was obtained from the linear state-space model from

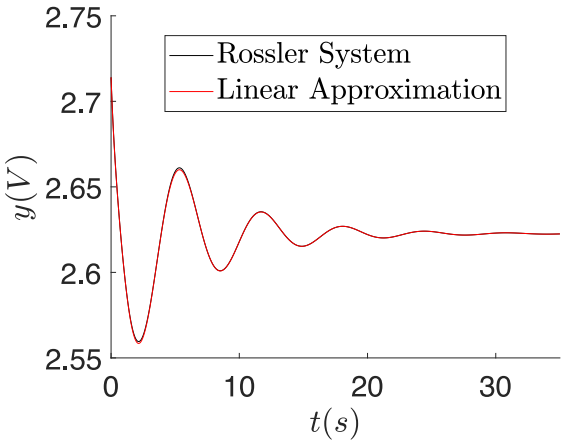


Figure 16: Output for small changes in all parameters.

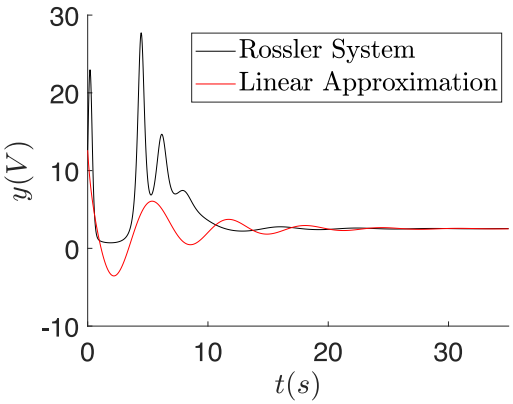


Figure 17: Output for relatively large changes in all parameters.

equation (27) and using equation (8), the transfer function for the linearized system is

$$G(s) = \frac{0.01333s^2 - 0.002667s + 0.01333}{s^3 + 4.977s^2 + 2.597s + 4.654} \tag{30}$$

which is the exact same equation that was calculated using *Matlab*.

F. Sampling Time Selection

The selection of sampling time for the discretization was done accordingly with was discussed in section II-G. The growth time was extracted from a simulation with the same parameters for the linealization (section III-A), except for the input of $u(t) = 1002V$. The obtained time response is shown in Fig. 18 as well as the two points needed to calculate the growth time.

From here, it was the growth time was gotten: $T_r = 3.073s$. Now applying the criterion mentioned in section II-G, the sampling time must satisfy

$$0.3073s < T < 1.5365s \tag{31}$$

Therefore, the selected sampling time for the discretization is $T = 1s$.

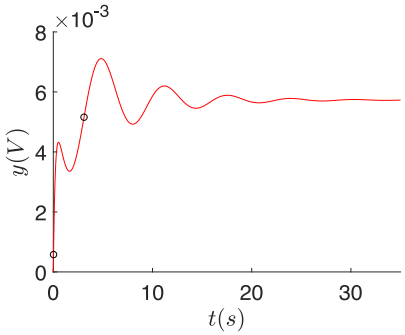


Figure 18: Time response and points for the growth time.

G. Discrete Transfer Function

The discretization was performed with the strategy presented in section II-H and the transfer function (equation (30)) obtained in section III-E. The discrete transfer function is

$$G(z) = \frac{0.001906z^2 - 0.001946z + 0.00226}{z^3 - 0.9395z^2 + 0.721z - 0.006893} \tag{32}$$

H. Comparison regarding the Transfer Functions and Linear Model

In this section, the time response of the system will be compared using three tools: the continuous (equation (30)) and discrete (equation (32)) transfer functions and the linearized model (equation (27)).

The simulation was performed with two inputs: a unitary step and a unitary impulse. The parameters are the same as in section III-A. In Figs. 19-20

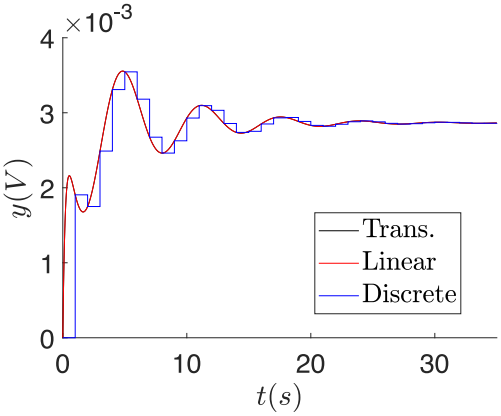


Figure 19: System's output to step input through three different methods.

Note that in both simulations, the output through the linearized system and its respective continuous transfer function is the same, since both graphs overlap, validating the function obtained. The discrete system gives an acceptable approximation, missing out the some of the first peaks of the system.

I. Ponderation Sequence

Following the ideas of section II-I, the ponderation sequence was obtained by taking the inverse $\mathcal{Z}$-transform of the discrete

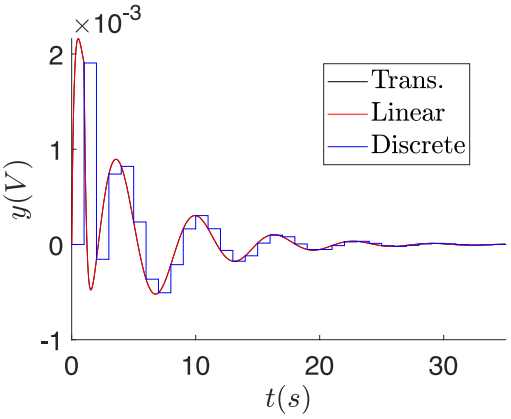


Figure 20: System's output to impulse input through three different methods.

transfer function (32), which is equivalent to calculate the system response to an unitary impulse; table I shows the first 10 terms of this sequence.

k	$g(k)$
0	0
1	0.0019
2	-0.0002
3	0.0007
4	0.0008
5	0.0002
6	-0.0004
7	-0.0005
8	-0.0002
9	0.0002

Table I: Ponderation Sequence

From this values, given an input, the respective output can be calculated (as described in section II-I) using discrete convolution. The following results show the response of the discrete system through this sequence for two inputs:

$$u_1(k) = H(k) \tag{33}$$

$$u_2(k) = \delta(k) \tag{34}$$

where H is the Heaviside or step function and δ is a unitary impulse or Kronecker delta. The response to the first input is displayed in Fig. 21, and for the second one in Fig. 22. Notice that in both cases, for both inputs, the outputs overlap with the simulation of the original linear state-space model; note as well that the discrete output is only calculated for the 10 values of the ponderation sequence. In order to obtain more points on the curve, more points for the ponderation sequence must be calculated.

J. Order Reduction

This section presents the results obtained using the three methods discussed in section II-J. The poles and zeros of the transfer function (30) will be calculated first.

The zeros of the continuous transfer function are given by the solutions to

$$0.01333s^2 - 0.002667s + 0.01333 = 0 \tag{35}$$

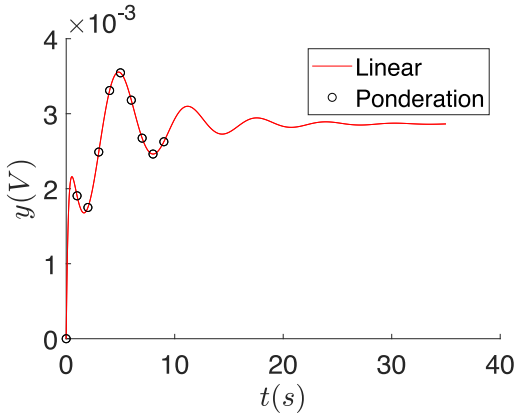


Figure 21: System output for u_1 using the ponderation values.

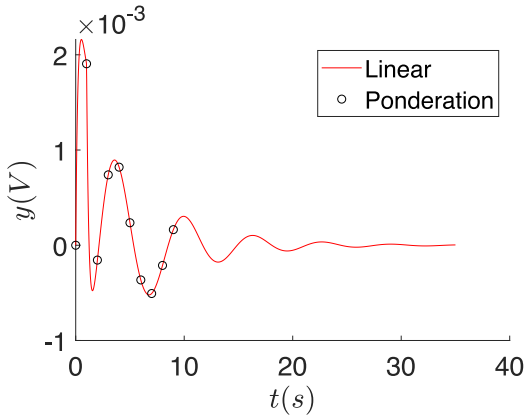


Figure 22: System output for u_2 using the ponderation sequence.

Therefore, $s_{z_1,z_2} = 0.100 \pm 0.995i$ are the two **finite** zeros of the transfer function; and for the poles, the following equation must be solved:

$$s^3 + 4.977s^2 + 2.579s + 4.654 = 0 \tag{36}$$

which solutions are $s_{p_1,p_2} = -0.1698 \pm 0.9873i$ and $s_{p_3} = -4.6376$. Clearly, the first method cannot be applied; note that the two finite zeros are complex and cannot be compared directly with the poles. On the other hand, for the poles, for the same reason as the zeros, they cannot be canceled out directly, since they provide essential properties to the system such as the oscillation. Finally the pole at $s_{p_3} = -4.6376$ does not have a close zero to be canceled out with.

For the second method, the complex poles and zeros cannot be canceled out; but for the pole s_{p_3}, the method can be applied: the dominant pole can be considered as $\mathrm{Re}(-0.1698 \pm 0.9873i) = -0.1698$ and the pole $s_{p_3} = -4.6376$ is more than 10 times away from the dominant pole. Therefore, the reduced transfer function using the second method is

$$\tilde{G}(s) = \frac{0.002875s^2 - 0.000575s + 0.002875}{s^2 + 0.3396s + 1.004} \tag{37}$$

The following plot (Fig. 23) shows the linear system's response to an input

$$u(t) = (2V)H(t) \tag{38}$$

compared with the response to the linear non-reduced system (equation (30)). In the figure it can be observed that both systems almost overlap starting from $t = 2s$.

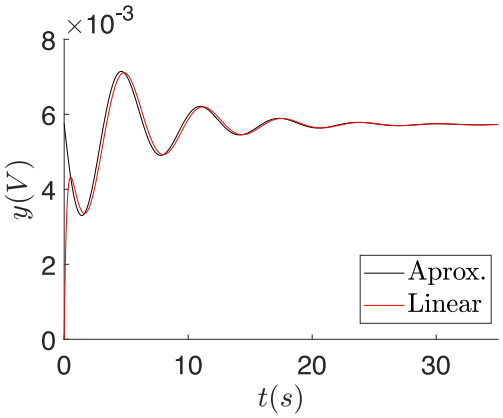


Figure 23: Results obtained with the reduced model (37).

Next, the 2nd order approximation, presented in section II-J, was performed based on the response to the original linearization of the Rössler system, based on a response to a step input. The parameters for the transfer function were calculated: $\zeta = 0.4125$, $\omega_0 = 0.7144$ and $k = 0.0029$. With this results, the following 2nd order transfer function was acquired:

$$\tilde{G}(s) = \frac{0.001461}{s^2 + 0.5887s + 0.5102} \tag{39}$$

The response of the system for an input as (38) is presented in Fig. 24. As expected, since it is an experimental approach, the results are not really good; the model successfully approximates the growth time, the maximum overshoot and the stationary state, but the oscillation frequency of the approximated system is slower than the actual model.

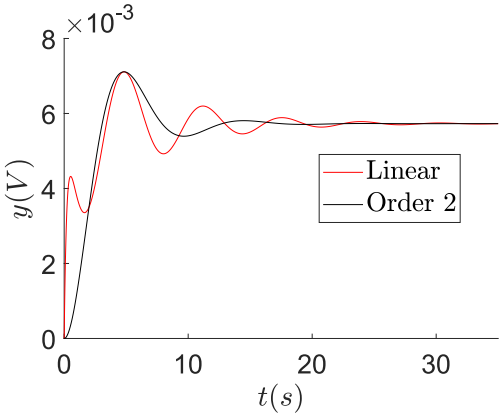


Figure 24: Results obtained with the reduced model (39).

Finally, the order reduction was performed using *Matlab*, with the command mentioned in section II-J. The resulting transfer function is

$$\tilde{G}(s) = \frac{0.0027s^2 - 0.0005665s + 0.002697}{s^2 + 0.3575s + 0.9416} \tag{40}$$

The same simulation was conducted for this new transfer function, as the previous reductions, and the result can be observed in Fig. 25.

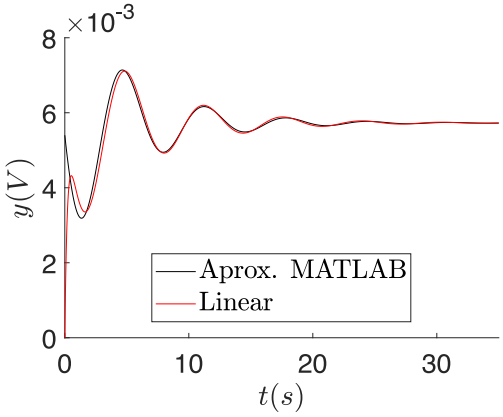


Figure 25: Results obtained with the reduced model (40).

This plot shows that the approximated model is just as good as the one obtained through the insignificant pole cancellation, since both graphs overlap from around $t = 4s$.

K. Routh-Hurwitz

For the following stability analysis tools, the parameter R_a was chosen to be analyzed. It is desired to find the region where the linear system is stable. In order to achieve this goal, the characteristic polynomial needs to be obtained in terms of the parameter R_a. Applying the formula $P(s) = |s\mathbf{I} - \mathbf{A}| = 0$, the resulting polynomial is

$$R_a s^3 + (5.1772R_a - 100.0)s^2 + (3.614R_a - 517.72)s + 5.1772R_a - 261.4 = 0 \tag{41}$$

And the respective Routh-Hurwitz array for this polynomial is

s^3	R_a	$3.614R_a - 517.72$
s^2	$5.1772R_a - 100.0$	$5.1772R_a - 261.4$
s^1	$\frac{2.614R_a^2 - 537.03R_a + 10000}{R_a - 19.316}$	
s^0	$5.1772R_a - 261.4$	

Applying the necessary and sufficient conditions, the stability region is $R_a > 184.7341k\Omega$; this means that the linearized Rössler system is stable for all values greater than $184.7341k\Omega$.

L. Root Locus

Following the procedure described in section II-L and using the characteristic polynomial (41), the expression to calculate the root locus is

$$1 + R_a\left(\frac{-0.01s^3 - 0.05177s^2 - 0.03614s - 0.05177}{s^2 + 5.177s + 2.614}\right) = 0 \tag{42}$$

From where the transfer function for the root locus can be extracted. Hence,

$$G(s) = \frac{-0.01s^3 - 0.05177s^2 - 0.03614s - 0.05177}{s^2 + 5.177s + 2.614} \tag{43}$$

The poles of this transfer function are $s_{p_1} = -4.6102$ and $s_{p_2} = -0.5670$, and the zeros are $s_{z_1} = -4.6387$ and $s_{z_{2,3}} = -0.2693 \pm 1.0216i$. This indicates that the root locus will commence in the two finite poles and an infinite pole (in $+\infty$); on the other hand, the root locus will end in the three finite zeros of the transfer function. The root locus was obtained using *Matlab* and is shown in Fig. 26.

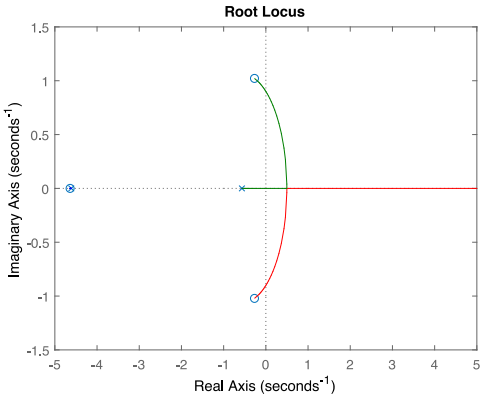
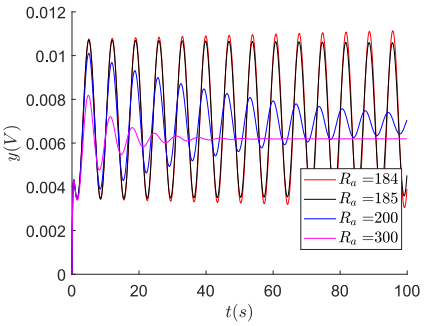


Figure 26: Root locus for the linearized Rössler system.

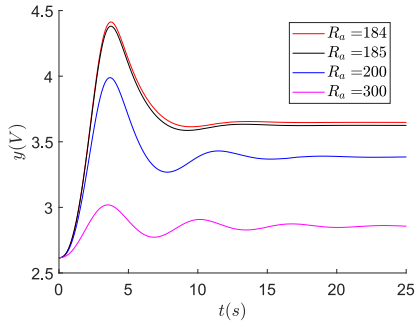
M. Stability Analysis

The following simulations were performed using the results from the Ruth-Hurwitz stability criterion. The first simulation, presented in Fig. 27, show the comparison between the original Rössler system and the linearization when changing the parameter R_a inside the stability region, for three different values. Note that both systems show stable behavior but they do not present similar responses.

The second simulation was performed using three values for R_a outside the stability region. The time responses for both the linear and nonlinear are shown in Fig. 28.

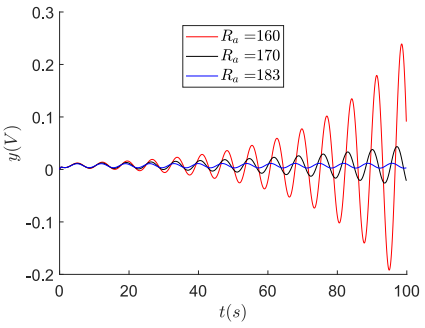


(a) Linear.

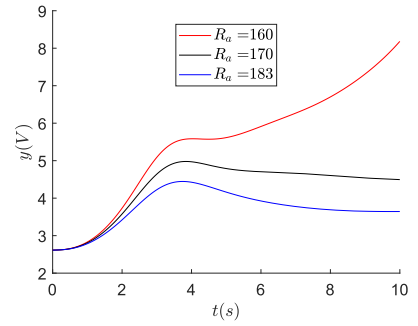


(b) Nonlinear.

Figure 27: Comparison for the linear and nonlinear systems for R_a is the stability region.



(a) Linear.



(b) Nonlinear.

Figure 28: Comparison for the linear and nonlinear systems for R_a is the stability region.

As it can be appreciated, the linear system shows a unstable behavior for all values, even though, for $R_a = 183k\Omega$, it looks like it is critically stable. On the other hand, for the nonlinear system, the response is stable for values where the linear is unstable, and is unstable for $R_a = 160k\Omega$.

N. Bode Diagram

The Bode diagram for the continuous system was obtained using *Matlab* through the continuous transfer function (30); in Fig. 29.

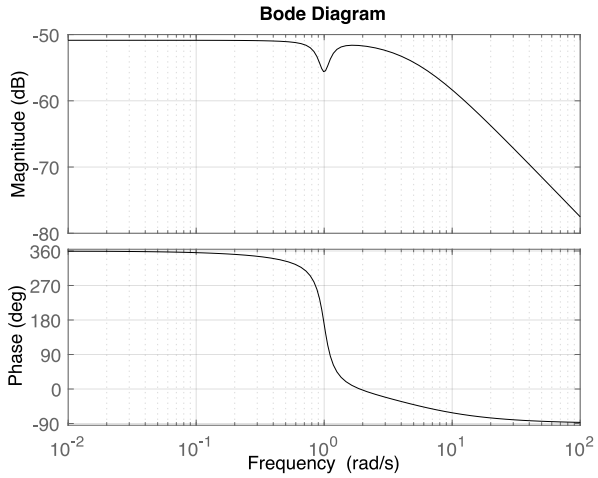


Figure 29: Bode diagram for the continuous linearized Rössler system.

The Bode diagram for the discrete system was calculated with *Matlab* as well, using the discrete transfer function (32) and the obtained Bode diagram is shown in Fig. 30.

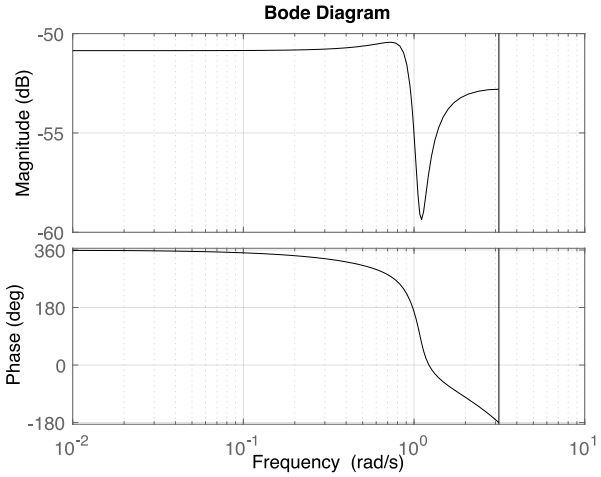


Figure 30: Bode diagram for the discrete system.

As it was discussed in section II-M, the Bode diagram can be used to determine the amplitude and phase for the output in stationary state, when the input is a sine wave. A simulation was conducted using a sine input with $A = 1V$, $\omega = 1rad/s$, hence

$$u(t) = (1V)\sin(t) \tag{44}$$

From the Bode diagram, the respective amplitude relation and phase of the output in stationary state can be obtained. In Fig. 31 the respective phase and relative amplitude are shown.

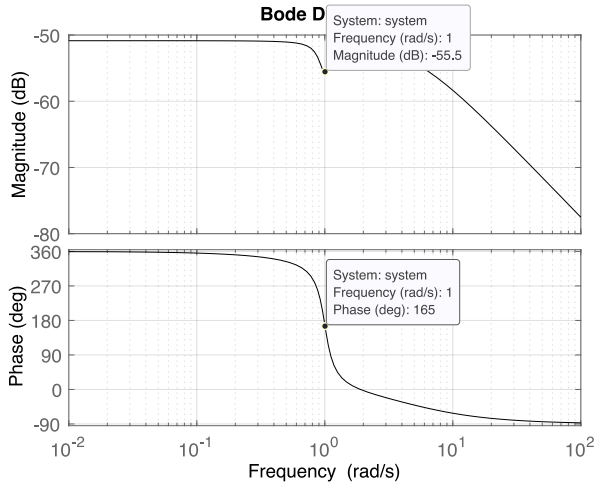


Figure 31: Phase and relative amplitude for $\omega = 1rad/s$.

Thus, the stationary state is given by

$$y_{ss}(t) = 0.0017\sin(t + 2.8798) \tag{45}$$

In the following graph (Fig. 32), the system output to this input is shown, as well as the sine wave obtained for the stationary state. Note that the system, over time, adjusts to the expected sine wave.

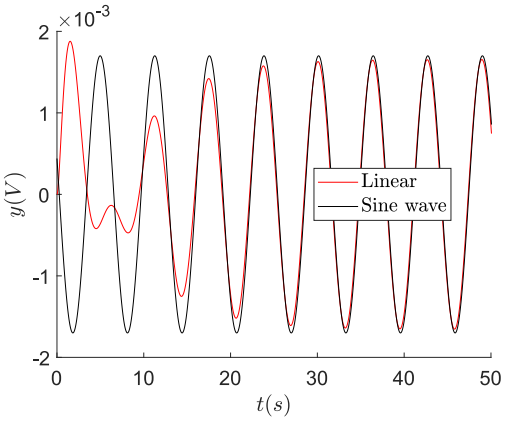


Figure 32: System output for sine input.

O. Frequency Response Comparison with the Reduced Model

The reduced model (37) gave a good approximation to the original linear system, as it was proved through Fig. 23. In this section, this model is used to compare a frequency response in both systems; in order to achieve this, the Bode diagram for the reduced model was obtained, as shown in Fig. 33.

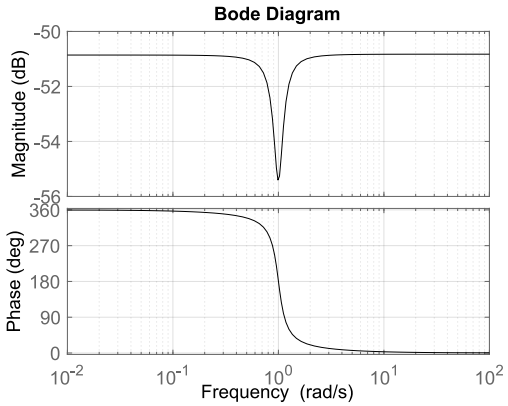


Figure 33: Bode diagram for reduced order approximation.

The selected input for the comparison was chosen as the same in equation (44). Hence, a simulation was conducted an the results for the systems' responses are presented in Fig. 34.

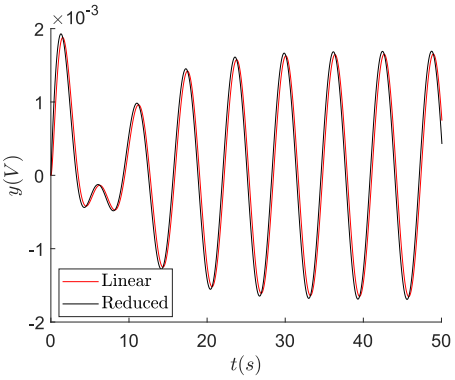


Figure 34: Comparison between outputs of the original model and the reduced for a sine wave input.

P. Closed-Loop Stability

The closed-loop stability analysis is based on the Bode diagram for the linearization of the Rössler system. The closed-loop diagram for this analysis is presented in Fig. 35.

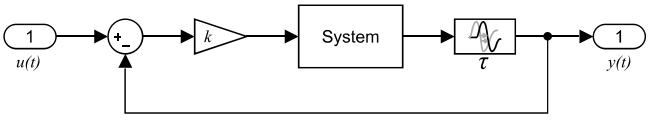


Figure 35: Closed-loop scheme.

Using *MATLAB*, as explained in II-M, the gain margin and the phase margin were calculated obtaining a maximum value for the gain of the closed-loop system of $k_{max} = 601.2835$ to preserve a stable system. On the other hand, it was found that a delay can not unstabilize the linear Rössler system. This first simulation, in Fig. 36, the k was selected for three different values inside the stability region.

The following simulation, in Fig. 37, it was selected three different values for k outside the stability region. It is important to notice that for $k = 620$ it does not seem as it had any

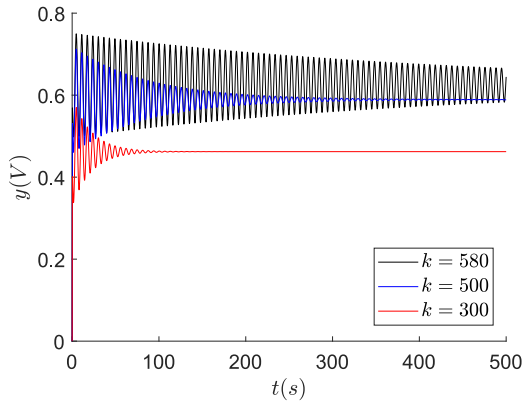


Figure 36: Stable system's response to some gains.

unstable behaviour as for the other values chosen; this will be explained in IV.

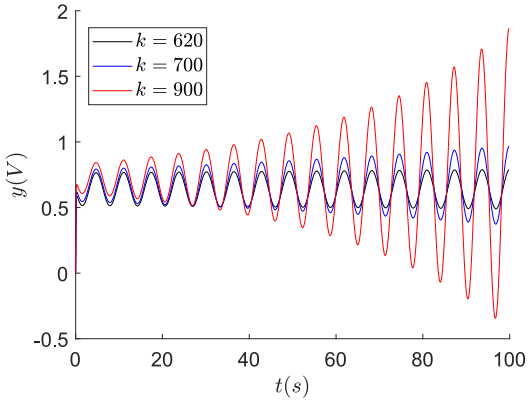


Figure 37: Unstable system's response to some gains.

Lastly, in Fig. 38, the closed-loop system was simulated with three different delays. As predicted by the margins calculated, the delay does not unstabilize the system.

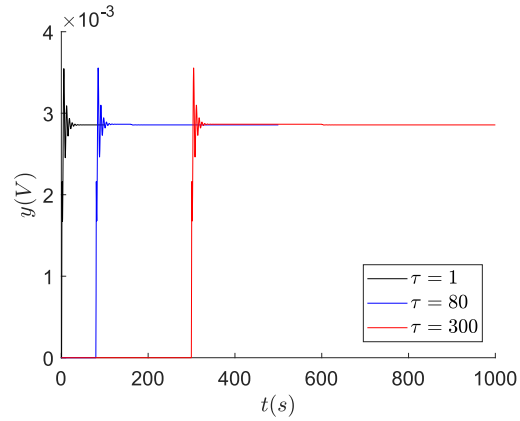


Figure 38: System's response to some delays.

IV. Results analysis

A. Comparison between the Linear and Nonlinear Systems

As it has been mentioned before, the linearization of the Rössler system (3) was performed based on previous results in [6]; there, it was found that the Rössler system is completely stable for large inputs, around $1000V$. Therefore, the equilibrium points were calculated based on this input and the linear model was obtained.

The linearity curve, discussed in section III-C and presented in Fig. 7, was developed in order to see where the linear system can successfully reach the same stationary state as the nonlinear. From the linearity curve, the linearity interval can also be extracted: in the curve, it can be observed that both systems (in stationary state) are really close from one another for u between $0V$ to $250V$. This means that the systems have the almost the same stationary state for inputs $u \in [0V, 250V]$, and this can also be verified in the simulations performed.

In section III-D, the inputs were chosen according to the linearity curve. Recall Fig. 8, note that both systems do not present changes in the output and they completely overlap; this is caused by the input selected: $1000V$. Since both systems are exactly at the operation point, which was chosen as an equilibrium point, their states do not change over time. In the next simulation, $1050V$ was selected as input; Fig. 9 shows the responses to this input, note that they present almost the exact same behavior, only differing in the peaks and valleys of the oscillation only by a really small factor. This can be partially explained with the linearity curve: both systems are equal in stationary state. The linearity curve does not provide information about the transitory state but as the input selected was chosen near to the operation point, both systems present equal behavior in transitory state as well.

In Fig. 10, the responses are shown for an input of $1200V$. Note that they start to differ a little bit, but it is still a value inside the range of linearity, that is why the stationary state is close. It is important to highlight that the transitory state of the linear approximation is not too different from the nonlinear one, they present similar behavior and same oscillation frequency; it can be observed that the Rössler system present larger damping, as it stabilizes faster than the linear approximation. Finally, in Fig. 11, the responses of the systems are displayed, but for an input outside of the linearity range ($1750V$). As expected, the systems stabilize in different values, but the linear system does a good approximation at the start of the simulation, but couldn't reproduce the strange and quick stabilization of the nonlinear system. The last graph (Fig. 12) shows the output for both systems to a sine input; since the linear system can successfully represent the nonlinear Rössler, for inputs close to the operation point, it was expected that the output for the nonlinear system would be a sine wave as well and, as this figure shows, the output was exactly the same. This is due to the sine wave chosen: as the amplitude is so small compared with the linearity range, all the values that the sine input take are contained in the linearity interval.

For a comparison in regarding the initial conditions, section III-D2, the initial conditions for the output state x_3 were changed. For a really small change $\varepsilon = 0.1V$ and a medium change $\varepsilon = 50V$, the system is stable and its state returns to the equilibrium point; this yields that the operation point chosen is an stable equilibrium point (at least in x_3) since the nonlinear system, for small changes in the initial conditions, it returns over time to the same equilibrium point, as it is shown in Figs. 13 and 14, even though their transitory state differ. The next simulation was performed with $\varepsilon = 400V$, which is significantly far from the operation point; in Fig. 15, the responses were presented; notice that the nonlinear system starts to diverge from the operation point; the reason for this is explained by the nonlinear dynamics associated with the Rössler system: it is likely that the system is going to another equilibrium point or just growing unboundedly, or it will start to show strange attractor properties and chaos.

One last comparison was developed with small and medium changes in the initial conditions and the input. From Fig. 16, the response of both systems in shown; as the input selected $(1003V)$ is so close to the operation point (and in the linearity range) and as the changes in the initial conditions are so subtle (stable operation point), both systems will show exactly the same behavior. On the other hand, as it can be observed in Fig. 17, the nonlinear system shows a strange transitory state that clearly cannot be obtained with the linear approximation; although, both systems stabilize in the same equilibrium point.

B. Continuous and Discrete Transfer function

In section II-F, the continuous transfer function was found. This function has some important properties to notice; in first place, it represents a non-minimum phase system as the zeros, found in III-J, have a positive real part. This is really noticeable in Fig. 18 where when the system is reaching its maximum value, instead of growing constantly to the peak it decreases for a moment and then proceeds to keep growing. On the other hand, it is confirmed that the system is stable as all the poles have negative real part.

For the sample time selection, it was specially important that the discretization does not lose main properties of the system; specially, the non-minimum phase property discussed before. As seen in Fig. 19, from the discrete system the original signal could be reconstructed without losing too much information about it. Note that even if the sample time selection was selected using a different simulation, for Fig. 20, the sample time is also a sufficient value to be able to reconstruct the signal without losing too much information.

It is important to analyze why is there an existence of a lower bound for the sample time, if a big selection could generate problems, such as aliasing. This is explained as, if it was chosen to be indefinitely small, it would be impossible to keep sampling the output in an experiment, as a really low sampling time cannot be replicated easily; also, the discrete signal would almost be continuous therefore, losing its purpose of making the signal discrete.

Additionally, about the response found in Fig. 19 and Fig. 20, it is important to notice how, in the first one, it stabilized in other place different from the initial conditions as the input was a constant step that made the system increase it's energy. On the other hand, when the impulse was given to the system in the second figure, it gain a high momentum for a bit and then comes back to the initial point, as the input goes to zero as soon as it transfers that momentum to the system.

Lastly, the discrete ponderation sequence is verified as every point calculated using it overlaps the continuous graph. It was noticed that this weighting sequence is really useful, as the convolution product is an easy operation to calculate and inverse zeta transform would be only used one time instead of using it every time it is needed to simulate with a different input.

In conclusion, the transfer function is a useful tool for simulating and saving the information about a linear system with null initial conditions specially for a SISO system.

C. Order reduction

In first place, it is seen that the *Matlab* reduction and the elimination of an insignificant pole are barely different as seen in equation 40 and 37. This happens because the software used utilizes numerical algorithms to find a fitting reduction for the system; instead, the other one was using analytical methods which would have more precision. In this manner, it is assured that both models are the same. Furthermore, if Fig. 25 and Fig. 23 are compared this thesis is confirmed even more, as the minor differences between the two plots can be explained by the same numerical errors.

In second place, there is an apparent contradiction that occurs in Fig. 23. It occurs, apparently, that the initial conditions of the system simulated are not 0, as it is seen in the comparison between the reduced system and the original system. This happens due to the obtained transfer function is strictly proper therefore, by the initial value theorem, the output would have a not null initial condition. Although this is a good reason why this occurs it is still an incomplete view of the problem.

In third place, the approximated approach did exactly what the system was purposed of as both systems have the same growth time, stationary state and maximum peak. On the other hand, the approximated approach does not take into account the non-minimum property as that system has not finite zeros; therefore, in the graph it grows continuously instead of taking a slight step backwards as in the original system. This behaviour, although not very representative of the original system, it is desirable as it would represent a causal system; in this manner, this reduced system is preferable than the others if the circuit is going to be recreated experimentally.

In conclusion, there was found two different models (the *Matlab* reduction and analytic one are considered the same as explained before) that represent the system to a certain degree of lower order than the original linear Rössler system.. The analytic system, ensures to represent the system in it's transitory state and final state except at the start in which it

changes the initial conditions; the approximated approach was close to the system in it's final state and peak but not during it's transitory state but, it is more practical to use in real life as this reduced system is minimum phase.

D. Stability Analysis

The parameter R_a, chosen to make the stability analysis has no special properties; the same procedure could have been made for parameter R_c or RC. As presented in section III-J, all the poles of the linearized Rössler system satisfy $\mathrm{Re}(\lambda_i) < 0$, therefore the system is stable; from the simulations performed in III-D, it can also be proved, graphically, that the system is stable.

It was desired to analyze for which values of R_a, the linear system is stable; in order to achieve this, it was required to obtain the characteristic polynomial in terms of R_a, since the roots of this polynomial are the poles of the linear system. This polynomial, presented in equation (41), will have all their roots in the left half-plane when the Routh-Hurwitz necessary and sufficient conditions are satisfied. This process lead to the condition $R_a > 184.7341k\Omega$; this implies that the resistor R_a can take values greater than this condition and the linear system will be stable. This is presented as well with the simulations performed, showed in Fig. 27a); as expected, the linear system is stable. On the other hand, for the stability of the nonlinear system, it would be necessary to make a deeper analysis, with Lyapunov theory for example; although, the simulations performed and showed in 27b) proves that the nonlinear system is stable for $184k\Omega \leq R_a \leq 300k\Omega$. It could be also proposed that the Rössler system

In Fig. 28a), the results of the linear system for values of R_a outside the stability region found are presented, this is another validation of the region found. Another important annotation is that the nonlinear system still has stable values outside the region of stability for the linear model, again, since the nonlinear system requires much further study; notice in Fig. 28b) that the nonlinear system is still stable for $R_a \geq 170k\Omega$. Hence, it can be asserted that the Rössler system in study is stable for $170k\Omega \leq R_a \leq 300k\Omega$, which is an important property that had to be mentioned.

All the previous analysis could have been performed with the root locus method, recall Fig. 26, remember that the root locus starts in the poles and ends in the zeros of the associated transfer function for the parameter in study, as depicted in II-L. The poles are marked with an $\times$ and the zeros with a $\circ$. Note that one root make almost no movement for $k \geq 0$ but is in the left half-plane; one of the other two roots starts in the left half-plane as well, whereas the other one start in the right half-plane in $+\infty$, but this plot also shows that there exists some values of R_a that stabilize the system; note that, after the roots reach the break point, both of them start moving towards the left half-plane and actually crossing into it towards the "end" of their trajectories, making all of the characteristic polynomial roots satisfy $\mathrm{Re}(\lambda_i) < 0$.

E. Bode Diagram

Firstly, yet again it is confirmed that the linear system is non-minimum phase. This happens because, if it would, then the change of phase of the bode diagram would be -180° instead of -360° as seen in Fig. 29. It is commonly said that for non-minimum phase systems the bode diagram is not a good tool for representing the original system as it gives wrong information; in this manner, it is said in the literature, that the Nyquist diagram is more useful. On the other hand, in our testing the bode diagram gave correct information about the original system.

Furthermore, in Fig. 31, it can be seen that for an sine wave input with a frequency of $\omega = 1rad/s$ the system would have decibels magnitude of -55.5. Using the formulas explained in methods, the stationary state would be given by the wave in equation 45. So, the system was simulated with this conditions and as seen in Fig. 32, the stationary state matches almost perfectly with the sine wave predicted by the bode diagram. In this manner, the small discrepancies between both simulations can be due to small numerical errors made by the software.

In second place, the discrete bode diagram does have the expected behaviour as it is very similar to the continuous one until it starts approach the Nyquist frequency which, start distorting the original signal so it starts not giving reliable information. Although there was not any simulation performed using this information, it is argued that because the similar behaviour of the discrete bode diagram to the continuous one it is confirmed that it does represent the original system at least for frequencies much smaller than the Nyquist one.

In third place, for the reduced model it is important to notice that the valley encountered at $\omega = 1rad/s$ has the same magnitude as the one encountered in the original system. Therefore, for frequencies in that valley and before the reduced system is a very good approximation of the original system. On the other hand, for frequencies beyond this valley it would seem as the magnitude becomes constant again not representing at all the behaviour of the not reduced model. This happens because, when the model was reduced the transfer function became not strictly proper therefore, making it have a rate of change of 0; at the same time, this warns that if the reduced model is going to be used instead of the original system, for frequencies larger than $\omega = 2rad/s$ the reduced model is useless to examine the original system.

In third place, for the closed-loop stability in the bode diagram it is seen that the phase plot does not cross the -180° value therefore it would seem as it did not had any crossover phase frequency. But, *Matlab* presented a value for this property because, there are several distinct definitions to calculate the different margins of the system and instead of seeing when the phase plot crosses the value selected it could be analyzed, instead, for 180°. In this manner, this value for the k_max was verified ins Fig. 36 and Fig. 37 in which inside and outside the stability region it behaves as it is predicted.

On the other hand, for the unstable plot it would seem the first two values are critically stable instead of unstable;

this happens because the growth of the value of the system is changing slightly, therefore it would be needed a lot more time to show this and it would be difficult to present the results in an organized way. Lastly, in Fig. 38, it was shown that for any delay chosen the system can not be unstabilized by this value.

In conclusion, the bode diagram was a useful tool which presented good information of the system when the input is a sine wave. This information allowed to calculate an interval for the stability of the closed loop system which was confirmed for the simulations presented.

V. Conclusions

In this work, a linear approximation to the Rössler circuit proposed by [5] was successfully proposed for an interval of the input of $u \in [1000V, 1200V]$ in a selected operation point with initial conditions $(0.5228, -2.6140, 2.6140)$ and initial input of $u_0 = 1000V$; this model was constructed and analyzed based on the theory and procedures presented in section II, all the results obtained were shown and properly explained and justified. Furthermore, a valid discrete system was found and verified with the same procedures. In this results, the stability region for parameter R_a was $[184.7341k\Omega, \infty)$, which makes sense, since higher values for this resistor would make small average voltage in the circuit and clearly the linear system cannot have unstable behavior. On the other hand, even though the order reduction was not completely successful, a second order linear system is easier to implement in a real life circuit; it was also found that the order reduction works fairly well for certain inputs. Finally, it was verified that the Bode diagram is a powerful tool to analyze the output for sine wave inputs and a concise way of presenting the information about a system; lastly, it is useful to determine the closed-loop stability in of the system for certain gains and delays, information that is necessary for further work in control design and beyond.

References

[1] O. E. Rössler, "An equation for continuous chaos," *Physics Letters A*, vol. 57, no. 5, pp. 397–398, 1976.

[2] M. K. Mandal, M. Kar, S. K. Singh, and V. K. Barnwal, "Symmetric key image encryption using chaotic rossler system," *Security and Communication Networks*, vol. 7, no. 11, pp. 2145–2152, 2014.

[3] D. S. Laiphrakpam and M. S. Khumanthem, "Cryptanalysis of symmetric key image encryption using chaotic rossler system," *Optik-International Journal for Light and Electron Optics*, vol. 135, pp. 200–209, 2017.

[4] J. Weule *et al.*, "Detection of n: m phase locking from noisy data: application to magnetoencephalography," *Phys. Rev. Lett*, vol. 81, no. 15, pp. 3291–3294, 1998.

[5] V. Canals, A. Morro, and J. L. Rosselló, "Random number generation based on the rossler attractor," *IEICE Proceeding Series*, vol. 1, pp. 272–275, 2014.

[6] J. S. Cárdenas and D. Plazas, "Simulation and analysis of chaotic rössler system based on circuits." 2019.

[7] P. J. Antsaklis and A. N. Michel, *A linear systems primer*. Springer Science & Business Media, 2007.

[8] P. S. Diniz, E. A. Da Silva, and S. L. Netto, *Digital signal processing: system analysis and design*. Cambridge University Press, 2010.

[9] M. Dahleh, M. A. Dahleh, and G. Verghese, "Lectures on dynamic systems and control," *MIT OpenCourseWare*, 2004.

[10] K. Ogata, *Modern Control Engineering*. Pearson, 2010.

[11] C. M. Vélez. Routh's array in symbolic way. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/33926-routh-s-array-in-symbolic-way

[12] J. Kiusalaas, *Numerical methods in engineering with python 3*. Cambridge University Press, 2013.