System modeling and identification techniques applied to an underdamped electronic system

1st Alan Sovano Gomes
Institute of Technology
Federal University of Pará
Belém, Brazil
alan.s.gomes@ieee.org

***Abstract*—This paper presents the modeling and identification of an underdamped electronic system, in order to discuss and review basic and fundamental methods in system modeling and identification theory. Three models were obtained: one using the system's step response, one through phenomenological modeling and a last one using the Recursive Least Squares (RLS) estimator. A theoretical background is provided to the reader for each method, followed by a comparison between the obtained model and experimental data. The model that could better describe the system behaviour was the one obtained using the RLS estimator, an fourth order ARX model. To define some parameters of the RLS estimator and obtain a well-suited data set, some information about the system obtained in the other two modeling methods was used. The ARX model presented an R^2 equal 0.9508, being considered adequate for many practical applications in system modeling and identification. To conclude, final comments are made regarding the discussed topics in this paper.**

***Index Terms*—Phenomenological modeling, recursive least squares estimator, step response, system modeling and identification**

I. Introduction

System modeling and identification is a broad and growing area, with a set of useful tools that can be used in control systems design, signal processing, applied econometrics, among other fields of application [1]. A model capable of describe a system accurately can be obtained in a lot of ways: through phenomenological modeling (according, for example, to the laws of physics that describe how the system works); through the system's response to a specific input signal (e.g., system modeling using its step response); and through a model estimator that is based on a optimization problem and a data set, such as Artificial Neural Networks or the Least Squares problem [1], [2].

To properly obtain a good mathematical model of the system, it is interesting that the model designer has knowledge about the system itself [1]. This kind of information can help in the design of a experimental setup to collect an appropriate data set for the identification process, and also guide the designer to the most adequate model in terms of complexity and final application.

Since system modeling and identification tools are widely used in several fields, it is important to have a solid knowledge regarding this topic. Hence, this paper aims to review some basic concepts and techniques regarding system modeling and identification, using as study object an underdamped electronic plant.

The layout of this paper is as follows. Section II describes the studied system and the modeling problem that needs to be solved. Section III discuss the modeling process using the step response of the system. Section IV presents a phenomenological model of the electronic plant. Section V discuss the identification of a model using the least squares method, through the recursive least squares algorithm. Finally, section VI presents some conclusions about the carried out study.

II. Study object description and problem definition

An underdamped electronic plant was designed in order to be used as a study object for this paper, with its schematics being shown in Fig. 1. The system was assembled using components that were new and that were retrieved from electronic waste. It is assumed that only the nominal values of the components and the model of the Operational Amplifiers (OpAmps) are known (the OpAmps reference is TL081). The designed system is basically a Sallen-Key low-pass filter in series with a first order RC low-pass filter, a negative diode clipping circuit and a resistor. To power the circuit, a $9\ V$ battery is used along with a virtual ground circuit, in order to generate a symmetric power supply that goes form $4.5\ V$ to $-4.5\ V$.

To apply the input signal and sample the output signal, an Arduino UNO was used. The clipping diode, D1, and the resistor R7 were used to give extra protection to the ADC converter of the ATMega328p (the microcontroller present in Arduino UNO). The system was assembled on a Printed Circuit Board (PCB). The designed PCB is shown in Fig. 2. The ATMega328p sampling period was configured to $250\ Hz$, and the signals were recorded using the software "SerialPlot". [3]. The system was designed to have its output always in the range of $0\ V$ to $4.5\ V$, since it iss within the reference voltage range used in the ADC converter ($0\ V$ to $5\ V$).

With the electronic system in a PCB, the modeling problem consists in finding an adequate model that describes the input-output relationship for this underdamped plant. Three classical approaches were chosen to achieve this goal:

- Modeling through step response;
- Phenomenological modeling;

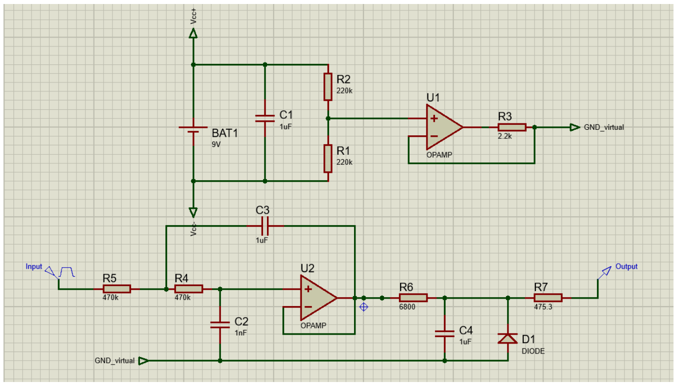

Fig. 1. Underdamped electronic system designed.

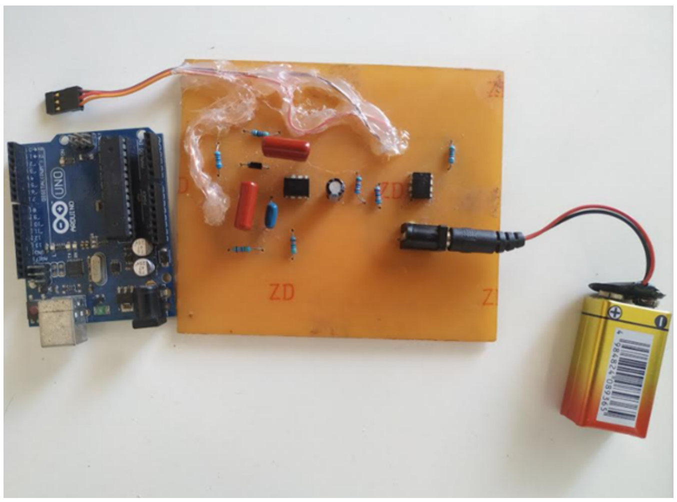

Fig. 2. Designed PCB alongside the Arduino UNO.

- Model identification using the Recursive Least Squares estimator.

The following sections better describe each one of these methods.

III. Modeling through step response

A. Theoretical background

One of the most elementary and consolidated ways of modeling a dynamic system, in special first and second-order ones, is using its step response [1], [2], [4]. An underdamped system can be often modeled by a second-order transfer function with the following structure [4]:

$$G(s) = \frac{K_{dc} \cdot {\omega_n}^2}{s^2 + 2 \cdot \zeta \cdot \omega_n \cdot s + {\omega_n}^2} \tag{1}$$

Where K_{dc} is the static gain of the transfer function $G(s)$, ζ is the damping factor and ω_n is the natural frequency of the system.

If one have access to the data set that describes the step response of the system in a adequate way, it is possible to find the second-order model parameters as follows [2]:

- Calculate K_{dc} using the input and output signals, $u(t)$ and $y(t)$, after the system is in steady-state, that is:

$$K_{dc} = \frac{y(\infty)}{u(\infty)} \tag{2}$$

- Find the maximum peak of the output signal, $y(t_p)$, and the time instant of the peak, t_p in the data set;
- Calculate the normalized peak of the output signal, $y_n(t_p)$:

$$y_n(t_p) = \frac{y(t_p)}{K_{dc}} \tag{3}$$

- Calculate ζ using the following equation:

$$\zeta = \sqrt{\frac{\{\ln[y_n(t_p) - 1]\}^2}{\{\ln[y_n(t_p) - 1]\}^2 + \pi^2}} \tag{4}$$

- Finally, calculate ω_n using the the equation that follows:

$$\omega_n = \frac{\pi}{t_p \cdot \sqrt{1 - \zeta^2}} \tag{5}$$

These steps can be easily translated into any programming language, in order to have an automatized script for modeling a second-order system from a step response. In the presented case, an algorithm that realize the above mentioned steps was written in MATLAB.

B. Comparison between experimental data and obtained model

In order to obtain a data set to the modeling process, a unit step was applied to the system's input and the output data was saved in a text file. The obtained experimental step response is shown in Fig. 3. From the figure, it is possible to conclude, in a graphical way, that the oscillation period is approximately 0.1 *seconds*.

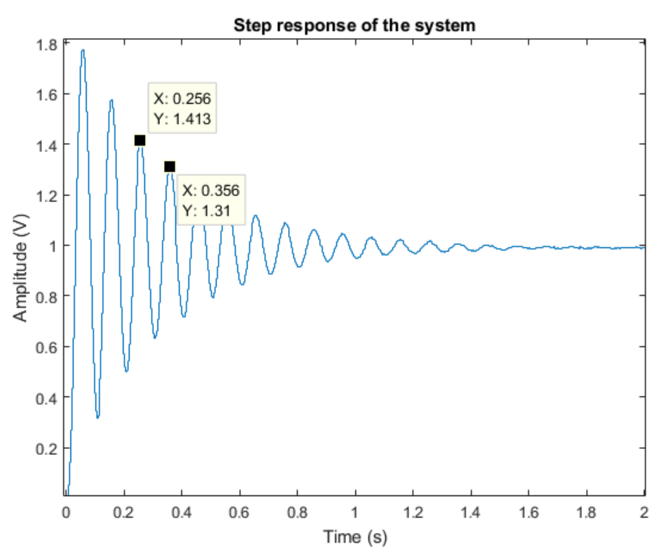

Fig. 3. Experimental unit step response of the system.

The algorithm written in MATLAB for the modelling using the step response of the system returned the following parameters: $K_{dc} = 0.9922$, $\omega_n = 56.2607\ rad/s$ and $\zeta = 0.0756$. These parameters, thus, results in the following transfer function:

$$G(s) = \frac{3141}{s^2 + 8.502 \cdot s + 3165} \tag{6}$$

The step response of the model presented in (6) is compared in Fig. 4 with the experimental data previously obtained. It is possible to notice that the obtained model can not explain the system's behaviour adequately. In order to better understand the system's properties, the next section discuss the phenomenological modeling of the plant.

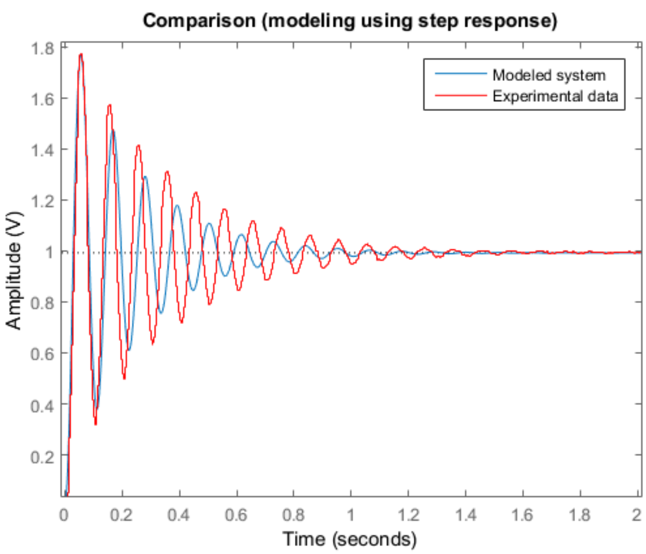

Fig. 4. Comparison between the experimental data and the model obtained from the step response.

IV. Phenomenological Modeling

A. Theoretical background and modeling conventions

To model the studied system, it is necessary to understand how it works. As discussed in section II, the electronic system consists in a virtual ground circuit, two filters in series and some protection components. In order to simplify the modeling procedure, the following conventions are assumed:

- The virtual ground circuit works in an ideal way (the reference does not oscillate);
- The OpAmps are ideal;
- The protection components does not have significant influence in the system dynamics;
- The component values are their nominal values.

The previous conditions, although not entirely true, simplifies the phenomenological modeling process, since the relevant dynamics can now be described only by the response of the two filters in cascade. Fig. 5 highlights each one of the filters in the system's schematic.

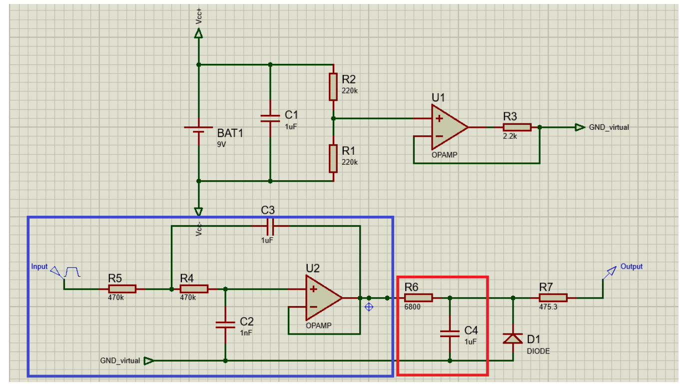

Fig. 5. Schematics of the studied electronic system, with the low-pass Sallen-Key filter (in blue) and the low-pass RC filter (in red) highlighted.

Since the filters are in series, the transfer function that describes the phenomenological model, $H(s)$, can be written as:

$$H(s) = H_1(s) \cdot H_2(s) \tag{7}$$

Where $H_1(s)$ is the transfer function of the Sallen-Key Filter and $H_2(s)$ is the transfer function of the RC filter. The mathematical model of these filter can be easily found in classical electronic textbooks, such as [5], which shows that the low-pass Sallen-Key transfer function is:

$$H_1(s) = \frac{\dfrac{1}{R_{in} \cdot R_{V_+} \cdot C_f \cdot C_{V_+}}}{s^2 + \dfrac{R_{in} + R_{V_+}}{R_{in} \cdot R_{V_+} \cdot C_f} s + \dfrac{1}{R_{in} \cdot R_{+} \cdot C_f \cdot C_{V_+}}} \tag{8}$$

Where R_{in} is the resistance where the input signal is connected, R_{V_+} is the resistor connected in the non-inverting input of the OpAmp, C_f is the feedback capacitor and C_{V_+} is the capacitance connected in the non-inverting output. For the low-pass RC filter, $H_2(s)$ is:

$$H_2(s) = \frac{\dfrac{1}{R \cdot C}}{s + \dfrac{1}{R \cdot C}} \tag{9}$$

Where R is the resistance and C is the capacitance of the RC filter.

B. Comparison between experimental data and obtained model

Substituting the nominal values shown in Fig. 1 and Fig. 5 into the transfer functions $H_1(s)$ and $H_2(s)$, and then multiplying them to obtain $H(s)$, results in:

$$H(s) = \frac{6.657 \cdot 10^5}{s^3 + 151.4 \cdot s^2 + 5153 \cdot s + 6.657 \cdot 10^5} \tag{10}$$

The step response of the phenomenological model and the experimental data are compared in Fig. 6. The new model, although presented a response different from $G(s)$, still does not describe the electronic system in an adequate way. Possible causes for this are:

- The consideration that the OpAmps are ideal (their non-ideal characteristics could have a non-neglectable influence in the system's output);
- The virtual ground and the protection circuit, somehow, are influencing the system's behaviour;
- The real values of all or some of the components significantly deviates from the nominal values;
- The amateur project of the PCB, which was handmade, could have some kind of non-identified problem, resulting in unmodeled dynamics.

Since the possible unmodeled dynamics are too complicated to be measured and evaluated, the next best alternative is to use the information obtained from this and the previous procedure to estimate a new model using the least squares method.

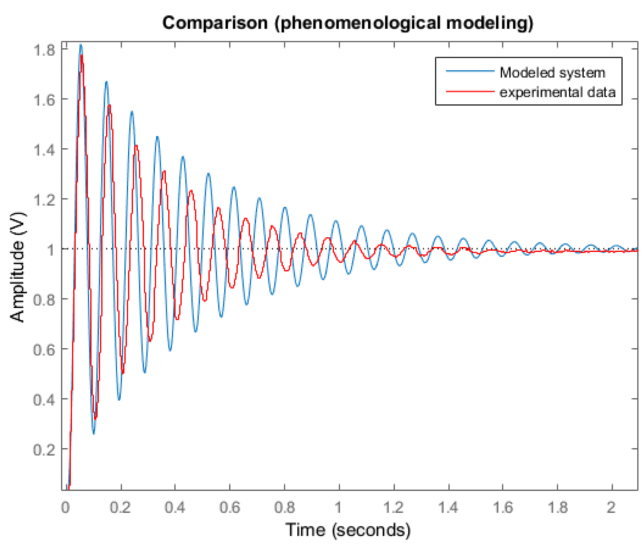

Fig. 6. Comparison between the experimental data and the phenomenological model step response.

V. Least squares method for system identification

The least squares method is a mathematical formulation based on an optimization problem that is widely used in system identification [6]. In this paper, the least squares algorithm used is the Recursive Least Squares (RLS) estimator, which will be further discussed in the following subsections.

To identify a model using the RLS estimator, one must follow the following steps:

- Define the the dynamics of interest that the model must describe;
- Design a input signal of the type Pseudo-Random Binary Sequence (PRBS), capable of exciting all the dynamics of interest;
- Apply the PRBS signal to the system and save the input-output data;
- Define the structure of the model to be identified;
- Define the initial conditions to the RLS estimator algorithm;
- Use the RLS estimator to the input-output data;
- Validate the model, comparing its output with the real system's output;
- Analyze other pertinent factors (e.g., the identification error and the evolution of the model parameters along the iterations).

Thus, the identification process will start with the design of a PRBS signal.

A. Design of the PRBS Signal

The PRBS signal is widely used in system identification. Its autocorrelation function approximates the autocorrelation function of a random signal, although the signal is generated in a deterministic way [1]. Using this kind of signal in an identification process that uses the least squares method provides additional noise immunity, resulting in a more robust model, and it is possible to easily design a PRBS signal capable of exciting the system in a well-defined frequency bandwidth [1], [2].

A good "rule of thumb" to define the bandwidth of interest to be excited is:

$$f_{\min} = \frac{1}{20 \cdot T_{oscillation}} \leq f \leq \frac{5}{T_{oscillation}} = f_{\max} \tag{11}$$

Where f_{min} and f_{max} are the inferior and superior limits of the bandwidth of interest, respectively, and $T_{oscillation}$ is the oscillation period of the system. From the step response modeling procedure, it was found that the oscillation procedure was approximately $T_oscillation = 0.1seconds$, which results in the following bandwidth:

$$0.5\ Hz \leq f \leq 50\ Hz \tag{12}$$

The values of f_{min} and f_{max} are, then, used to calculate the time between bits, T_b, and the number of registers in the signal generator, n. Basically, the PRBS signal can be generated using a shift register with n bits and the XOR operator between some of these bits (the bits that go through the operator depend on the size of the shift register) [1]. The parameter T_b indicates for how long the last bit sent to the system's input must be maintained, in order to properly excite the system. A 9 bit PRBS generator is shown in Fig. 7 for illustration purposes.

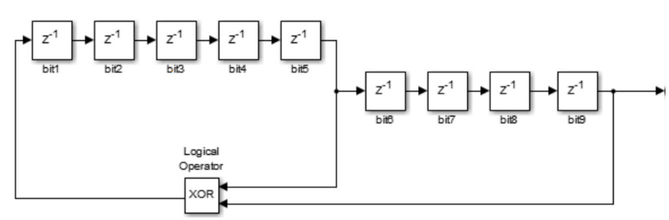

Fig. 7. A 9 bit PRBS generator.

A practical rule to obtain an appropriate interval for T_b is [1]:

$$\frac{1}{10 \cdot f_{\max}} \leq T_b \leq \frac{1}{3 \cdot f_{\max}} \tag{13}$$

Which results in the following interval:

$$0.002\ seconds \leq T_b \leq 0.0067\ seconds \tag{14}$$

Since the sampling period (the inverse of the sampling frequency) is $0.004\ seconds$ and is within the interval described in (14), it was defined that $T_b = 0.004\ seconds$, so the generation of the PRBS and the sampling period of the data would be all the same. To calculate the size of the shift register, n, the following equation can be used [2]:

$$f_{\min} = \frac{1}{(2^n - 1) \cdot T_b} \tag{15}$$

Which results in $n = 9$ bits, if the value of n is rounded to the nearest higher integer. Thus, the PRBS generator will have its block diagram equal to Fig. 7, with the XOR operation occurring between the fifth and ninth bits. The PRBS signal will repeat its sequence after the 512 samples for a 9 bit shift register, which can be seen in its autocorrelation function [1]. Fig. 8 shows the autocorrelation function of the designed PRBS for a signal of duration between 11 and 12 seconds.

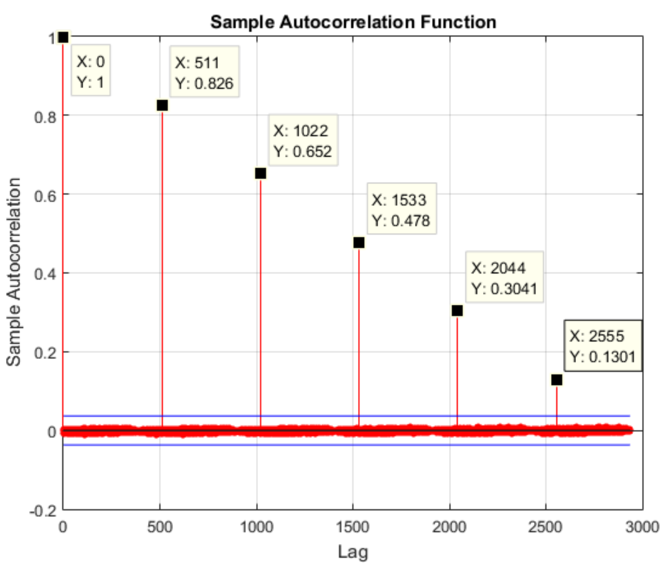

Fig. 8. Autocorrelation function of a PRBS signal generated with a 9 bit shift-register.

The designed PRBS signal was applied in the input of the system, using a PWM output of the Arduino UNO. It was defined that the PRBS signal would vary from $1\ V$ to $2\ V$, in order to the output stay in the range from $0\ V$ to $4.5\ V$. The new input-output data set experimentally obtained is graphically shown in Fig. 9. With this new data set, the RLS estimator can be properly applied.

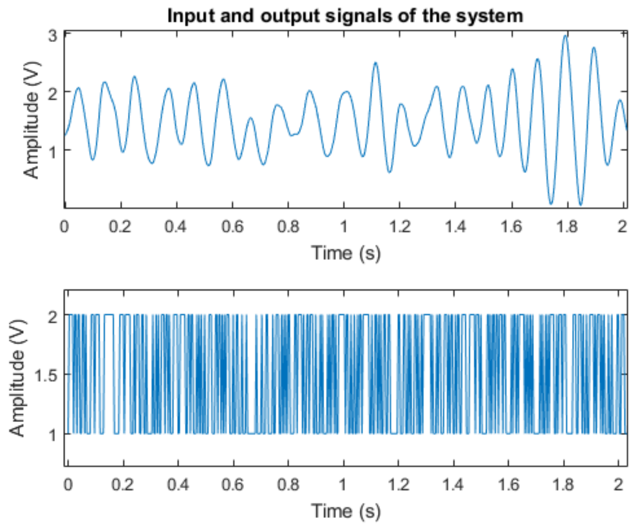

Fig. 9. System's response to the designed PRBS signal.

B. RLS estimator

The RLS estimator algorithm can be summarized in the following steps [1]:

- Define the initial conditions for the covariance matrix, P, the estimation error, e, the estimated parameter vector, $\widehat{\boldsymbol{\theta}}$, and the estimation gain vector, $\mathbf{K}$;
- Define and start the regressor vector, $\boldsymbol{\phi}\,(\boldsymbol{k})$;
- Calculate the estimated output of the model, $\hat{y}$, defined as:

$$\widehat{y}(k) = \boldsymbol{\phi}\,(\boldsymbol{k})\,\widehat{\boldsymbol{\theta}}\,(\boldsymbol{k}-\mathbf{1}) \tag{16}$$

- Update the estimation error:

$$e\,(k) = y\,(k) - \widehat{y}\,(k) \tag{17}$$

- update the gain vector:

$$\boldsymbol{K}\,(\boldsymbol{k}) = \frac{P\,(k-1)\,\boldsymbol{\phi}\,(\boldsymbol{k})}{1 + \boldsymbol{\phi^T}\,(\boldsymbol{k}) P\,(k-1)\,\boldsymbol{\phi}\,(\boldsymbol{k})} \tag{18}$$

- Update the estimated parameter vector:

$$\widehat{\boldsymbol{\theta}}\,(\boldsymbol{k}) = \widehat{\boldsymbol{\theta}}\,(\boldsymbol{k}-\mathbf{1}) + \boldsymbol{K}\,(\boldsymbol{k}) \cdot e\,(k) \tag{19}$$

- Update the covariance matrix:

$$P\,(k) = P\,(k-1) - \boldsymbol{K}\,(\boldsymbol{k})[P\,(k-1)\,\boldsymbol{\phi}\,(\boldsymbol{k})]^T \tag{20}$$

- Repeat all steps (until it is possible or desired).

The chosen structure of the model to be identified was the following fourth order ARX model:

$$T\,(z) = \frac{b_0 \cdot z^{-2} + b_1 \cdot z^{-3}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + a_3 \cdot z^{-3} + a_4 \cdot z^{-4}} \tag{21}$$

A fourth order ARX model was chosen because, in the phenomenological modeling, it was possible to conclude that the system had, at least, three poles, thus, the natural next step would be to include an additional pole to the model structure. Also, zeroes were also included in the model, since previous models did not had this characteristic, and this dynamic characteristic may help to explain the data. To identify the ARX model chosen, the following regressor vector was used:

$$\phi(k) = \begin{bmatrix} -y(k-1) \\ -y(k-2) \\ -y(k-3) \\ -y(k-4) \\ u(k-2) \\ u(k-3) \end{bmatrix} \tag{22}$$

C. Evaluation of the identified model

The RLS estimator resulted approximately in the following model in the z domain, which was obtained using a MATLAB script with the RLS estimator algorithm:

$$T(z) = \frac{0.01921z^{-2} + 0.0182z^{-3}}{1 - 1.90z^{-1} + 0.55z^{-2} + 0.79z^{-3} - 0.40z^{-4}} \tag{23}$$

Fig. 10 shows a comparison between the experimental data and the output of the identified model for an applied PRBS. It is possible to see that the system has a response compatible with the experimental data. In the beginning of the graphics, the model appears to be not well adjusted to the data set because the model is considering that the initial conditions are zero, while this is not true for the experimental data. After the effect of these initial conditions are ceased, the both the model is capable of reproducing the output of the real electronic system.

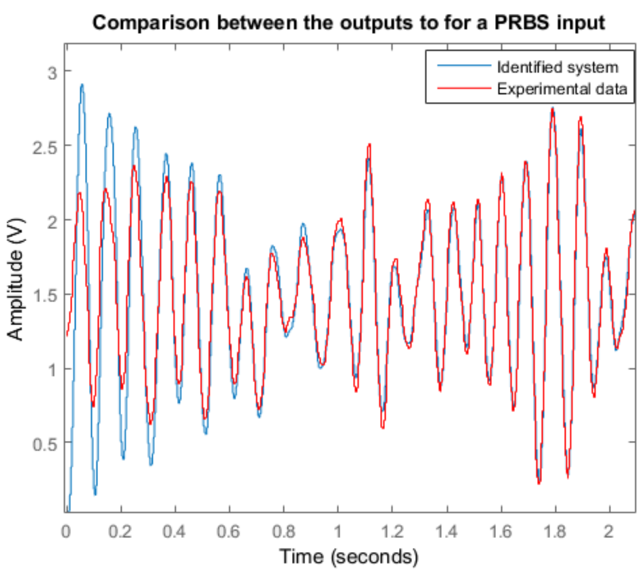

Fig. 10. Comparison between the identified model and the experimental data for a PRBS input.

In order to evaluate the convergence of the algorithm, the absolute error and the evolution of the parameters along the iteration were also analyzed. Fig. 11 shows the absolute identification error, and Fig. 12 shows the parameters along the iterations. It is possible to conclude that the absolute error reached low values in the last iterations, and the parameters presented some stability in the last iterations of the estimation process.

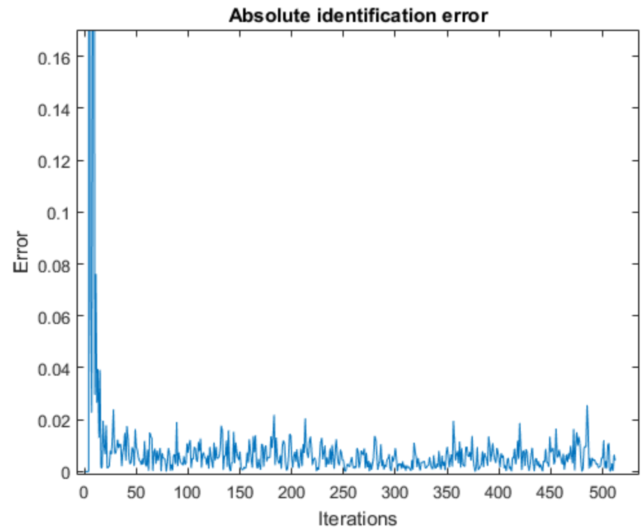

Fig. 11. Absolute identification error of the RLS estimator.

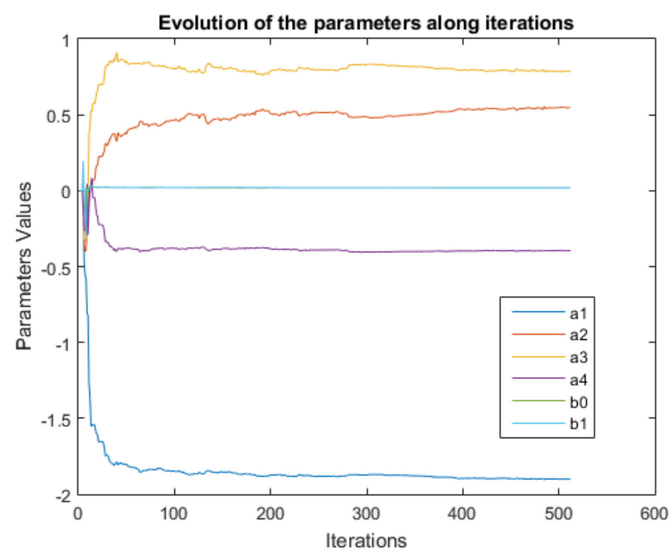

Fig. 12. Evolution of the parameters of the estimated model.

Finally, to validate the model, the step response data was used, since this data set did not participate in the identification of the model using the RLS estimator. A graphical comparison between the experimental data and the model output for a unit step is shown in Fig. 13. To numerically verify if the model is adequate, the sum of the square errors, SSE, and the coefficient of determination, R^2, were calculated, resulting in:

$$SSE = \sum_{k=1}^{N} \left[y(k) - \widehat{y}(k) \right]^2 = 0.9478 \tag{24}$$

$$R^2 = 1 - \frac{SSE}{\sum_{k=1}^{N} \left[y(k) - \bar{y}(k) \right]^2} = 0.9508 \tag{25}$$

According to [2], an R^2 value between 0.9 and 1 is considerate adequate for many practical applications in system identification. Thus, the ARX model obtained can be used to describe and simulate the studied electronic system behaviour.

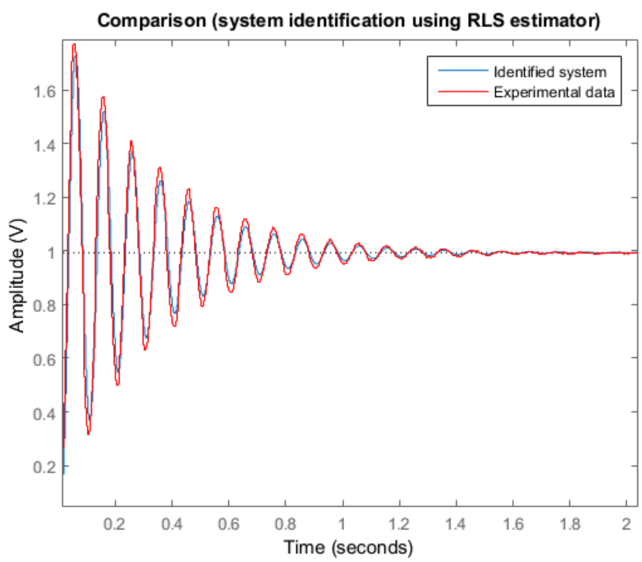

Fig. 13. Comparison between experimental data and the step response of the identified model.

VI. Conclusions

This paper discussed three different methods for obtaining a mathematical model of a underdamped electronic system. Although the models obtained through the step response and the phenomenological modeling could not describe the system in an adequate way, useful information was obtained from both procedures, since it was possible to estimate the oscillation period of the real system and have an idea of its minimal order. This information guided some decisions in the estimation process using the RLS estimator, which resulted in adequate model for describing the dynamical characteristics of the studied plant.

Although the RLS estimator generated the best model, sometimes it is important to insist in a better phenomenological modeling. In some cases, the researcher seeks to understand the physical, chemical and/or biological relationships between certain measurable quantities, and a identified model will not be as useful. An interesting discussion regarding this dilemma can be read in [7].

To conclude, it is important to emphasize that system modeling and identification theory gives a set of tools that can be used to solve a large set of problems. One must know, although, that to obtain an adequate model, some experience and information-based decisions are necessary, and knowledge about the problem will always be decisive between obtaining an useful or an useless model.

References

[1] L. A. Aguirre, *Introdução à identificação de sistemas*. Editora UFMG, 2007.

[2] A. A. R. Coelho and L. dos Santos Coelho, *Identificação de sistemas dinâmicos lineares*. Editora da UFSC, 2004.

[3] H. Y. Özderya. (2015) Serialplot software. [Online]. Available: https://hackaday.io/project/5334-serialplot-realtime-plotting-software

[4] K. Ogata, *Modern control engineering*. New Jersey, USA: Prentice hall, 2010.

[5] A. P. Malvino and D. J. Bates, *Eletrônica*. AMGH Editora Ltda., 2011.

[6] R. F. Stengel, *Optimal control and estimation*. Courier Corporation, 1994.

[7] C. Rudin and J. Radin, "Why are we using black box models in ai when we don't need to? a lesson from an explainable ai competition," *Harvard Data Science Review*, vol. 1, no. 2, 2019.