Control and Identification Toolbox (CIT): An Android application for teaching automatic control and system identification

Abstract

This article presents a free Android application, named as Control and Identification Toolbox (CIT), for teaching the automatic control and parameter identification of first and second order linear systems. The proposed application runs in any android device supporting a universal serial bus (USB), which is connected with Arduino Uno or Mega boards that carry out data acquisition. The CIT app allows performing real-time experiments with these systems, and visualizing their open loop or closed-loop responses under common test input signals. The system parameter identification is performed by means of the Recursive Least Squares Method, whereas the automatic control is carried out with the traditional Proportional Integral Derivative controller, whose performance under constant or noise disturbances introduced by the CIT app can be analyzed. Experimental results obtained using first and second order low pass filters confirm the effectiveness of the proposed application.

Keywords

Android applications, real-time experiments, parameter identification, automatic control.

1 Introduction

The parameter identification and automatic control of systems are topics widely studied in careers related to robotics, mechanics and Electrical engineering, mechatronics, chemistry, and biology, just to mention a few. The reason is that automatic control is applied in robot manipulators, machine tools, process control valves, and servomechanisms, which are employed in a great variety of industrial applications. Moreover, the parametric estimation of systems is important for designing high-performance controllers, and state observers that estimate variables that are not available.

Courses of automatic control and parameter identification requires both theory and practice. To this end, there are several commercial programs such as MATLAB/Simulink and LABVIEW, where students can perform simulations and experiments; however, these programs are expensive, and as consequence it is not possible to use them in all courses of automatic control and parameter identification. An alternative to them are free programs as the ones presented in [1], [2] for automatic control simulations. In [1] Chen et al. developed a free program in C/C++, called as Ch Control System Toolkit, for teaching the root locus, control design, and time and frequency response of time invariant linear control systems. Moreover, in [2] Ayas and Altas crated the Virtual Control Laboratory (VCL), designed in Embarcadero RAD Studio's C++, for simulating classical proportional-integral-derivative controllers and fuzzy logic regulators.

In the literature, there are also simple techniques of control and/or identification that have been proposed for teaching graduate and undergraduate students. Oliverira et al. [3] propose a methodology for teaching the system identification of open-lop systems using the step response and the Particle

Swarm Optimization. Moreover, Galvao et al. [4] use the off-line Least Squares algorithm in order to estimate the parameters of a magnetic levitation system, which are subsequently employed for designing a digital lead compensator. On the other hand, references [5]–[8] present laboratory setups for the identification and/or control of a DC motor using MATLAB/Simulink, where the experimental kits in [6], [7], and [8] contain a dSPACE prototype, a FPGA, and an Arduino board, respectively. Remote laboratories have also been constructed for distance learning of system identification and automatic control [9], [10]. The most important characteristics of eight automatic control remote laboratories around the world are described in [11].

This article presents a free Android application, called the Control and Identification Toolbox (CIT), for developing experimental practices of system response, parameter identification, and automatic control of first and second order linear systems. Experiments in the CIT application are carried out in real-time using any Android device, whose processor executes the identification and control algorithms. The android device is communicated with Arduino Uno or Mega boards, which performs data acquisition, using a universal serial bus (USB). To our best knowledge, the CIT is the first app developed for real-time experiments of system response, parameter identification, and automatic control. The developed application takes advantage of the fact that currently most graduate and undergraduate students have an Android cellphone or tablet, and that the cost of Arduino boards is low. Moreover, this application can be installed on laptops or personal computers through a virtual box with the program android-x86 [12], thus allowing its use by students that do not have cell phones with the Android operative system. Through the proposed application, students can experimentally determine and visualize the responses of first and second order systems under common test input signals as steps and the following waves: sine, sawtooth, triangular, square, and rectangular. The system response produced by the CIT, in turn, can be compared with that obtained analytically during class. Students can also identify the parameters of these systems by means of the Recursive Least Squares method, and they can design and tune a conventional PID controller. The CIT app also permits introducing constant and noise disturbances in order to verify the performance of the PID controller under these disturbances. Moreover, the CIT app presents the discrete implementations of both the identification and control algorithms, which are also reproduced in this paper, in order for the students to understand their programming. A set of experiments, developed with first and second order low pass filters, are presented in this paper to show the results provided by the app. These results have also been reproduced by students of the Faculties of Mechanical and Electrical Engineering at the Universidad Autónoma de Coahuila, and the Universidad de Colima, who have been very enthusiastic since the control and identification theory seen in class has been experimentally verified through the application.

The paper is organized as follows. Section 2 describes the CIT android application and its graphical user interface. The mathematical models of first and second order low pass filers, employed for experiments, are shown in Section 3. This section also shows the responses of these systems to different inputs that are plotted by the CIT application. Sections 4 and 5 present, respectively, the CIT app results produced by identifying and controlling these systems. Finally, concluding remarks of this paper are described in Section 6.

2 CIT Android application

The Control and Identification Toolbox (CIT) application (app) was developed using the Java language in the Android Studio environment. Table 1 presents the list of classes used in this app. The opensource code is available at [13] and the compiled Android PacKage (APK) is published at [14].

Figure 1 shows the home screen of the application that contain an action bar and a main view area. When a model is selected from the drawer, the main view area displays the view associated to the selected model. The action bar has a hamburger-Icon that opens the drawer shown in Figure 2 and that allows selecting between the following three main categories presented in Figure 2: system response, parameter identification, and PID controller, which are described in sections 3, 4, and 5, respectively. At the right hand side of the action bar, a toolbar is located, which contains the buttons download, settings, software and hardware, share, and hardware-in-the-loop. The download button captures the screen content and stores it as a jpg image file. Through the settings button, students define the desired settings during the real-time experiments, which are grouped in the following three categories: Simulation, Graphs, and Bridge device, which are described in Tables 2, 3 and 4, respectively. The software and hardware button displays another view that contains the link to download the firmware of the Arduino Uno or Mega boards, which performs data acquisition. This option also displays the connection diagram of the Arduino board to a first or a second order linear filter, which can be used for testing the CIT application or for developing simple practices of system response, automatic control, and parameter identification. These circuits are shown and described in Section 3. The share button allows sharing the screenshot produced with the CIT app using the compatible image sharing apps such as Gmail, WhatsApp, Photos, and so on. Finally, the hardware-in-the-loop button runs and stops the execution of real-time experiments. When they are in progress, the app does not permit the user to leave the app area by disabling the access to the drawer and buttons.

2.1 Experimental results screen

The experimental results screen (ERS) is displayed when any of the three main sections, that are open-loop system response, parameter identification, and PID controller, is selected. Figure 3 shows the ERS corresponding to the first-order parameter identification section. The ERS contains the following categories:

1. Diagram: Displays the diagram block corresponding to the identification or control algorithm.

2. Sampling time: Lets the user to change the sampling time directly from this screen without using the desired settings button. In this way, the student can change the sampling time in real-time while an algorithm is running.

3. Parameters: The parameters of the identification or control algorithm are selected here.

4. Input signals ($u_1(t)$, $u_2(t)$ and $u_3(t)$): Permits selecting the reference input to the system, which is the sum of these three input signals, that can be any of the following wave functions: step, sine, sawtooth, triangular, square, and rectangular. Reference [15] presents the definition of these functions.

5. Figures: Show the signals or parameter estimates in real-time. More than one figure can be displayed on the screen. The height of the figures can be adjusted from the desired settings button.

6. Instantaneous values: Displays the instantaneous values of the selected signals or the parameter estimates, which will help the student to write them down in the notes instead of estimating them from figures.

2.2 Data acquisition

The CIT application uses an Arduino board as a bridge device, commonly called as data acquisition system, between the Android system and the plant to be controlled or identified. The android device executes all the computations related to the system response and the identification or control algorithms. On the other side, the Arduino board is required to translate the output generated by these algorithms into real world voltage levels, and to capture the output of the plant, which is sent to the android device. The bridge and the android device use a USB communication, that is configured with the help of [16], [17]. This communication is selected over the Bluetooth because the USB connection can also be used for energizing the bridge device and the plant.

2.3 Firmware

The firmware is taken from [18], which is downloaded into the Arduino board. The firmware algorithm is shown in Figure 4, which converts the Arduino board into a data acquisition card. With this algorithm the Arduino board acts as a USB CDC class, which waits to receive a message from the host that sends the messages “analog write X in port A” or “analog read in port B”. In the case of the message “analog write X in port A”, the Arduino board set a PWM value of X into the port A. On the other hand, the message “analog read in port B”, means that the Arduino board need to reply the voltage into the port B.

2.4 Real-time execution

The Model, Figure, and Parameter classes described in Table 1 provide the necessary information to implement system response, identification or control algorithm in real-time, and to visualize their results. The real-time operations like receiving and sending messages to the bridge circuit, and executing the identification or control algorithms are carried out by the method doInBackground(Params...) of the Android **AsyncTask** class [19]. This software technique allows the real-time experiments to be performed using hardware-in-loop, where a separated thread runs the identification or control algorithm so that the GUI remains responsive during the experiments.

3 System response

This section presents the experimental responses of two simple circuits, composed by a first and a second order low pass filters, which are used for testing the CIT app and for developing practices during the courses of automatic control and/or parameter identification. These circuits are shown in the CIT app when the software and hardware button of the home screen is pressed, and they are reproduced in the Figures 5 and 6 of this paper, where parameters R and C denote resistance and capacitance, respectively. Moreover, the connection between the Arduino board and the first order low pass filter is shown in Figure 7.

Note that the general mathematical model of a first and a second order linear dynamic systems is described through equations (1) and (2), respectively.

$$\dot{y}(t)+\alpha_3 y(t)=\alpha_1\dot{u}(t)+\alpha_2 u(t) \tag{1}$$

$$\ddot{y}(t)+\beta_4\dot{y}(t)+\beta_5 y(t)=\beta_1\ddot{u}(t)+\beta_2\dot{u}(t)+\beta_3 u(t) \tag{2}$$

where α_1, α_2, α_3, β_1, β_2, β_3, β_4 and β_5 are positive parameters. The transfer functions $H(s)$ of systems (1) and (2) are given by:

$$\frac{Y(s)}{U(s)}=H(s)=\frac{\alpha_1 s+\alpha_2}{s+\alpha_3} \tag{3}$$

$$\frac{Y(s)}{U(s)}=H(s)=\frac{\beta_1 s^2+\beta_2 s+\beta_3}{s^2+\beta_4 s+\beta_5} \tag{4}$$

where $Y(s)=\mathcal{L}[y(t)]$ and $U(s)=\mathcal{L}[u(t)]$, and $\mathcal{L}[\cdot]$ is the Laplace operator.

The structure of first and second order low pass linear filters, used to verify the performance of the CIT app, is a particular case of the structure of the systems (3) and (4), respectively. Note that the mathematical model of a first order low pass filter is given by:

$$\frac{Y(s)}{U(s)} = \frac{\frac{1}{\tau}}{s + \frac{1}{\tau}} \tag{5}$$

where $\tau = RC$ is called time constant, $U(s)$ and $Y(s)$ are the input and the output voltage of the filter, respectively; By comparing equations (3) and (5) yields $\alpha_2 = \alpha_3 = 1/\tau$, and $\alpha_1 = 0$.

On the other hand, a second order low pass filter is described through the following mathematical model:

$$\frac{Y(s)}{U(s)} = \frac{\frac{1}{\tau^2}}{s^2 + \frac{3}{\tau}s + \frac{1}{\tau^2}} \tag{6}$$

By observing equations (4) and (6), equalities $\beta_1 = \beta_2 = 0$, $\beta_3 = \beta_5 = 1/\tau^2$ and $\beta_4 = 3/\tau$ are deduced.

All the experiments shown in this paper use the parameters $R = 100$ kΩ and $C = 1$ μF, which produces $\tau = 0.1$ s, $\alpha_2 = \alpha_3 = 10$ s^{-1}, $\beta_3 = \beta_5 = 100$ s^{-2}, and $\beta_4 = 30$ s^{-1}. Moreover, the sampling time T_s defined for all the experiments is $T_s = 0.04$ s, and the simple moving average is selected as the sampling time type.

3.1 First order system

The responses plotted by the CIT app, when the first order low pass filter is excited by a step and a sine wave input, are shown below. Note that a step input allows determining how fast the system response can reach the amplitude of the input. Moreover, the sine wave input allows determining the frequency response of linear systems.

Applying the step function $u(t) = EH(t)$ in volts to the first order low pass filter model in (5) produces:

$$y(t) = E[1 - e^{-t/\tau}] \tag{7}$$

Figure 8 shows the response $y(t)$ plotted by the Android application using $E = 2.5$ V. Note that at $t = \tau = 0.1$s, the response $y(t)$ approximately reaches 63.2% of the amplitude E, which corroborates the expression (7).

On the other hand, let a sine wave input with offset $r(t) = F\sin(\omega t) + \kappa$, where F is the amplitude, ω the angular frequency, and κ the offset. Applying this input to (5) analytically produces the following steady-state response y_{ss}, that is obtained after a large period of time of $y(t)$.

$$y_{ss}(t) = \frac{F}{\sqrt{1 + \tau_c{}^2\omega^2}} \sin(\omega t - \tan^{-1}\tau_c\omega) + \kappa \tag{8}$$

The frequency $\omega_c = 1/\tau_c$ in rad/s is called cut-off frequency of the filter. Figure 9 presents the response $y_{ss}(t)$ provided by the CIT application using $r(t) = F\sin(\omega t) + \kappa$, where $F = 1$, $\kappa = 2.5$V, $\omega = \omega_c = 10$ rad/s. It is shown that the amplitude of $y_{ss}(t)$ is approximately $0.7F$, which coincides with the value provided by equation (8).

On the other hand, the amplitude of the responses $y_{ss}(t)$ obtained with the app under this input $r(t)$, where $\omega = 2\pi f$ varies from $f = 0.1$ Hz to $f = 10$ Hz, produce the magnitude of the frequency response of the system, which is shown in Figure 10.

3.2 Second order system

Figures 11 and 12 depict the responses of the second-order low pass filter, modelled by (6), when it is excited by a sawtooth and a square wave, respectively. The amplitude of these inputs is unitary, and they have an offset of 2.5 V, as well as a frequency of 1 Hz. Note that the responses of the filter are smother than its inputs.

4 Parameter identification

The CIT Android application permits estimating the parameters of the continuous-time models in equations (1) and (2), as well as the parameters of discrete-time models corresponding to (1) and (2). First, the parameters of the discrete time models are estimated using the Recursive Lest Squares Method (RLSM). Afterwards, these models are converted to their continuous-time counterpart.

4.1 First-order system

The zero-order-hold discretization method allows obtaining the following discrete time model corresponding to (1):

$$y(k) = \theta_1 u(k) + \theta_2 u(k-1) + \theta_3 y(k-1) \tag{9}$$

$$\theta_1 = \alpha_1, \qquad \theta_2 = \left[\frac{\alpha_2}{\alpha_3} - \theta_1\right]\left[1-\theta_3\right] - \theta_1\theta_3, \qquad \theta_3 = e^{-\alpha 3 T_s}$$

where parameters α_1, α_2 and α_3 are suppose to be unknown and will be estimated.

Equation (9) can be rewritten as:

$$y(k) = \phi^T(k)\boldsymbol{\theta} \tag{10}$$

$$\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^T, \qquad \phi(k) = [u(k), u(k-1), y(k-1)\ ]^T \tag{11}$$

Expression (10) is called linear parameterization, since it is simply a linear equation in terms of the unknown vector $\boldsymbol{\theta}$ [20].

The CIT Application uses the RLSM in order to estimate the vector parameter $\boldsymbol{\theta}$ in (10). The RLSM is given by [21]:

$$
\begin{aligned}
&\hat{\boldsymbol{\theta}}(k)=\hat{\boldsymbol{\theta}}(k-1)+\boldsymbol{L}(k)\epsilon(k)\\
&\boldsymbol{L}(k)=\frac{\mathbf{P}(k-1)\phi(k)}{\gamma+\phi^{T}(k)\mathbf{P}(k-1)\phi(k)}\\
&\mathbf{P}(k)=\frac{1}{\gamma}\left[\mathbf{P}(k-1)-\frac{\mathbf{P}(k-1)\phi(k)\phi(k)^{T}\mathbf{P}(k-1)}{\gamma+\phi^{T}(k)\mathbf{P}(k-1)\phi(k)}\right]\\
&\epsilon(k)=y(k)-\hat{y}(k)\\
&\hat{y}(k)=\phi^{T}(k)\hat{\boldsymbol{\theta}}(k-1)
\end{aligned}
\tag{12}
$$

where $\hat{\boldsymbol{\theta}}(k)=[\hat{\theta}_1(k),\hat{\theta}_2(k),\hat{\theta}_3(k)]^T$ is an estimate of $\boldsymbol{\theta}$, γ is called forgetting factor and satisfies $0<\gamma\leq 1$. Moreover, variable $\mathbf{P}(k)=\mathbf{P}^T(k)$ is called covariance matrix, $\hat{y}(k)$ is the predicted output, and $\epsilon(k)$ is the prediction error.

Remark: The estimate vector $\hat{\boldsymbol{\theta}}(k)$ converges to $\boldsymbol{\theta}$ if and only if the input $u(t)$ has at least $n/2$ different frequencies, where n is the number of parameters of vector $\hat{\boldsymbol{\theta}}$ [21].

According to the last remark, the signal $u(t)$ must have at least two frequencies so that $\hat{\boldsymbol{\theta}}$ converges to $\boldsymbol{\theta}$.

From equation (9), it is possible to compute the parameters α_1, α_2 and α_3 of the continuous time model (1) as follows:

$$
\hat{\alpha}_1(k)=\hat{\theta}_1(k),\qquad \hat{\alpha}_2(k)=\hat{\alpha}_3(k)\left[\frac{\hat{\theta}_2(k)+\hat{\theta}_1(k)\hat{\theta}_3(k)}{1-\hat{\theta}_3(k)}+\hat{\theta}_1(k)\right],\qquad \hat{\alpha}_3(k)=-\frac{\ln(\hat{\theta}_{3*}(k))}{T_s}
\tag{13}
$$

$$
\hat{\theta}_{3*}(k)=\begin{cases}\hat{\theta}_3(k) & \text{if} \quad \hat{\theta}_3(k)>0.01\\ 0.01 & \text{if} \quad \hat{\theta}_{3*}(k)\leq 0.01\end{cases}
$$

where $\hat{\theta}_{3*}(k)$ is a parameter projection of $\hat{\theta}_3(k)$, that takes only positive values.

An experiment is carried out with the CIT Application to identify the nominal parameters $\alpha_1 = 0$, $\alpha_2 = 10$ and $\alpha_3 = 10$ of the first-order linear filter, which is excited with the following input:

$$u(t) = 0.8\sin(0.4\pi t) + 0.5\sin(1.6\pi t) + 0.5\sin(2.4\pi t) + 2.5 \quad \text{V} \tag{14}$$

Figures 13 and 14 show the estimates $\hat{\theta}_1(t)$, $\hat{\theta}_2(t)$, $\hat{\theta}_3(t)$ and $\hat{\alpha}_1(t)$, $\hat{\alpha}_2(t)$, $\hat{\alpha}_3(t)$, respectively. Moreover, Figures 13 and 14 present the prediction output $\hat{y}(k)$ of the estimated model, and the instantaneous values of parameters and signals, respectively. From these Figures, It is possible to conclude that the predicted output $\hat{y}(k)$ is very close to the system output $y(k)$, and that the estimates $\hat{\alpha}_1(t)$, $\hat{\alpha}_2(t)$, $\hat{\alpha}_3(t)$ converge to their nominal values $\alpha_1(t)$, $\alpha_2(t)$, $\alpha_3(t)$ in approximately 2 s, .

4.2 Second-order system

The continuous-time model in (2) can be discretized as:

$$\begin{aligned} y(k) &= \theta_1 u(k) + \theta_2 u(k-1) + \theta_3 u(k-2) - \theta_4 y(k-1) - \theta_5 y(k-2) = \phi^T(k)\boldsymbol{\theta} \\ \boldsymbol{\theta} &= [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]^T, \qquad \phi(k) = [u(k), u(k-1), u(k-2), -y(k-1), -y(k-2)]^T \end{aligned} \tag{15}$$

An on-line estimate $\hat{\boldsymbol{\theta}}(k) = [\hat{\theta}_1(k), \hat{\theta}_2(k), \hat{\theta}_3(k), \hat{\theta}_4(k), \hat{\theta}_5(k)]^T$ of the parameter vector $\boldsymbol{\theta}$ in (15) is determined through the RLSM given in (12). Since five parameters are estimated, convergence of $\hat{\boldsymbol{\theta}}$ to $\boldsymbol{\theta}$ requires that $u(t)$ has at least three different frequencies.

For estimating the parameters β_1, β_2, β_3, β_4 and β_5 of the continuous-time model (2), the following procedure is carried out:

- In each sample time the following matrices are computed:

$$\widehat{\mathbf{A}}_d = \begin{bmatrix} 0 & 1 \\ -\hat{\theta}_5 & -\hat{\theta}_4 \end{bmatrix}, \qquad \widehat{\mathbf{B}}_d = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad \widehat{\mathbf{C}}_d = [\hat{\theta}_3 - \hat{\theta}_5\hat{\theta}_1, \hat{\theta}_2 - \hat{\theta}_4\hat{\theta}_1], \qquad \widehat{\mathrm{D}}_d = \hat{\theta}_1 \tag{16}$$

which correspond to the next state space representation of the estimated model in (15):

$$\begin{aligned} \mathbf{x}(k+1) &= \widehat{\mathbf{A}}_d \mathbf{x}(k) + \widehat{\mathbf{B}}_d u(k) \\ y(k+1) &= \widehat{\mathbf{C}}_d \mathbf{x}(k) + \widehat{\mathrm{D}}_d u(k) \end{aligned} \tag{17}$$

- Matrices in (16) are used to obtain

$$\begin{aligned} &\widehat{\mathbf{A}}_c = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \frac{2}{T_s}\mathbf{R}\left[\mathbf{I}_{2x2} - \frac{8}{21}\mathbf{R}^2 - \frac{4}{105}\mathbf{R}^4\right]\left[\mathbf{I}_{2x2} - \frac{5}{7}\mathbf{R}^2\right]^{-1} \\ &\widehat{\mathbf{B}}_c = \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} = \widehat{\mathbf{A}}_c\left[\widehat{\mathbf{A}}_d - \mathbf{I}_{2x2}\right]^{-1}\widehat{\mathbf{B}}_d, \qquad \widehat{\mathbf{C}}_c = [c_{11}, c_{12}] = \widehat{\mathbf{C}}_d, \qquad \widehat{\mathrm{D}}_c = \widehat{\mathrm{D}}_d \\ &\mathbf{R} = \left[\widehat{\mathbf{A}}_d - \mathbf{I}_{2x2}\right]\left[\widehat{\mathbf{A}}_d + \mathbf{I}_{2x2}\right]^{-1} \end{aligned} \tag{18}$$

where $\widehat{\mathbf{A}}_c$, $\widehat{\mathbf{B}}_c$, $\widehat{\mathbf{C}}_c$ and $\widehat{\mathrm{D}}_c$ correspond to the following continuous-time counterpart of (17), which are computed using the Geometric series method [22];

$$\dot{\mathbf{x}}(t) = \widehat{\mathbf{A}}_c x(t) + \widehat{\mathbf{B}}_c u(t) \tag{19}$$

$$y(t) = \widehat{\mathbf{C}}_c \mathbf{x}(t) + \widehat{\mathrm{D}}_c \mathbf{x}(t) \tag{20}$$

Finally, in each sample time the estimates $\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{\beta}_3$, $\hat{\beta}_4$ and $\hat{\beta}_5$ are computed as follows

$$\begin{aligned} &\hat{\beta}_1 = \widehat{\mathrm{D}}_c, \quad \hat{\beta}_2 = c_{11}b_{11} + c_{12}b_{21} - \widehat{\mathrm{D}}_c\left[a_{11} + a_{22}\right], \\ &\hat{\beta}_3 = c_{11}(b_{21}a_{12} - b_{11}a_{22}) + c_{12}(b_{11}a_{21} - b_{21}a_{11}) + \hat{\beta}_5\widehat{\mathrm{D}}_c, \quad \hat{\beta}_4 = -(a_{11} + a_{22}), \quad \hat{\beta}_5 = a_{11}a_{22} - a_{12}a_{21} \end{aligned} \tag{21}$$

where equations in (21) result by equalling the transfer function $H(s)$ in (4) with the transfer function

$$H(s) = \widehat{\mathbf{C}}_c(s\mathbf{I} - \widehat{\mathbf{A}}_c)^{-1}\widehat{\mathbf{B}}_c + \widehat{\mathrm{D}}_c \tag{22}$$

The CIT app is used to estimate the parameters θ_1, θ_2, θ_3, θ_4, θ_5, β_1, β_2, β_3, β_4, and β_5 corresponding to the second-order filter model in (6), whose input is given by the expression in (14). Figure 15 shows the parameter estimates $\hat{\theta}_1$, $\hat{\theta}_2$, $\hat{\theta}_3$, $\hat{\theta}_4$ and $\hat{\theta}_5$ of the discrete-time model corresponding to the filter, whereas Figure 16 depicts the parameters $\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{\beta}_3$, $\hat{\beta}_4$ and $\hat{\beta}_5$ of the continuous-time model. Note that the parameter estimates $\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{\beta}_3$, $\hat{\beta}_4$ and $\hat{\beta}_5$ converge close to their nominal values, which are $\beta_1 = \beta_2 = 0$, $\beta_3 = \beta_5 = 100$ and $\beta_4 = 30$. Note that Figure 15 also presents the predicted output $\hat{y}(k)$ that converges to $y(k)$.

5 Proportional Integral Derivative (PID) controller

Figure 17 shows a closed-loop system, where $r(t)$ and $y(t)$ are the reference input and the output of the system, respectively. Signal $e(t)$ is the difference between $r(t)$ and $y(t)$; furthermore, the system is corrupted by a disturbance $d(t)$ and is controlled through a PID controller, which produces a control signal $u(t)$ that reduces $e(t)$ to zero or to a small value. The PID controller is defined as [18]:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \dot{e}(t), \qquad e(t) = r(t) - y(t) \tag{23}$$

The disturbance $d(t)$ is given by:

$$d(t) = d_1 + d_2 \xi(t) \tag{24}$$

where $d_1 \in R$ is a constant disturbance, $\xi(t)$ is a zero mean white noise, and d_2 is its power.

The PID controller is implemented in the CIT app in the following discrete-time version [23]

$$u(k) = u(k-1) + \lambda_1 e(k) + \lambda_2 e(k-1) + \lambda_3 e(k-2)$$
$$\alpha_1 = K_P + \frac{K_I T_s}{2} + \frac{K_D}{T_s}, \qquad \alpha_2 = -K_P + \frac{K_I T_s}{2} - \frac{2K_D}{T_s}, \qquad \alpha_3 = \frac{K_D}{T_s} \tag{25}$$

where the integral and derivative terms are approximated by means of the Trapezoidal and the Backward Euler methods [24], respectively.

5.1 First-order system

This section presents two experimental results produced with the first order low pass filter operating in closed-loop under the PID controller (23). In these experiments, reference input $r(t)$ is given by $r(t) = 2.5H(t)$ V, and the filter is corrupted by means of a constant disturbance $d_1 = 0.05$ V, which is generated with the CIT app. It is well known that the effect of the constant disturbance d_1 is eliminated by means of the integral action of the controller. In order to verify this fact, the first experiment uses the gains $K_P = 10$, $K_I = 0$, and $K_D = 0$, whereas the second experiment employs $K_P = 10$, $K_I = 2$, and $K_D = 0$. Figures 18 and 19 show the output $y(t)$ of the fist and the second experiment, respectively. From these figures, it is evident that only the output $y(t)$ of the first experiment is corrupted by the disturbance d_1, thus corroborating the aforementioned fact.

5.2 Second-order system

Figure 20 shows an experiment, obtained with the second order filer controlled by the PID controller, where the reference input $r(t)$ is a triangular wave with unitary amplitude, offset of 2.5 V, and frequency of 0.25 Hz. In this experiment, the system is corrupted by a constant disturbance $d_1 = 0.05$V and a white noise $\xi(t)$ with a power of $d_2 = 0.01$. Moreover, the gains of the PID controller are $K_P = 20$, $K_I = 5$, and $K_D = 0.1$. From this figure, it is possible to see that signal $y(t)$ is very close to $r(t)$; moreover, the control signal $u(t)$ has high frequency components due to the measurement noise $\xi(t)$.

6 Conclusions

This paper proposed the CIT android application, which has been used for undergraduate or graduate students, in order to carry out real-time experiments related to the system response, parameter identification, and automatic control. The CIT permits estimating the parameters of first and second order linear systems using the Least Squares method, and allows controlling single input-single output linear systems of any order using a PID regulator. The processor of the Android device executes all the control and the identification algorithms, whereas data acquisition is carried out with Arduino Uno or Mega boards, which are communicated with the android device through a USB connection. Experiments, developed with first and second order low pass filters, show that their results validate the identification and control theory behind these systems, where these results are visualized with graphs and instantaneous values. It is worth mentioning that the CIT application has provoked that students are very enthusiastic and engaged with the classroom activity, and that classes are very interactive.

7 References

[1] B. Chen, Y.-C. Chou, and H. H. Cheng, "Open source Ch Control System Toolkit and web-based control system design for teaching automatic control of linear time-invariant systems," *Comput. Appl. Eng. Educ.*, vol. 21, no. 1, pp. 95–112, Mar. 2013 [Online]. Available: http://doi.wiley.com/10.1002/cae.20454. [Accessed: 02-Jul-2018]

[2] M. S. Ayas and I. H. Altas, "A virtual laboratory for system simulation and control with undergraduate curriculum," *Comput. Appl. Eng. Educ.*, vol. 24, no. 1, pp. 122–130, Jan. 2016 [Online]. Available: http://doi.wiley.com/10.1002/cae.21678. [Accessed: 02-Jul-2018]

[3] P. M. Oliveira, D. Vrančić, J. B. Cunha, and E. J. S. Pires, “Teaching particle swarm optimization through an open-loop system identification project,” *Comput. Appl. Eng. Educ.*, vol. 22, no. 2, pp. 227–237, Jun. 2014 [Online]. Available: http://doi.wiley.com/10.1002/cae.20549. [Accessed: 02-Jul-2018]

[4] R. K. H. Galvao, T. Yoneyama, F. M. U. de Araujo, and R. G. Machado, “A simple technique for identifying a linearized model for a didactic magnetic levitation system,” *IEEE Trans. Educ.*, vol. 46, no. 1, pp. 22–25, Feb. 2003 [Online]. Available: http://ieeexplore.ieee.org/document/1183663/. [Accessed: 02-Jul-2018]

[5] J. C. Basilio and M. V. Moreira, “State–Space Parameter Identification in a Second Control Laboratory,” *IEEE Trans. Educ.*, vol. 47, no. 2, pp. 204–210, May 2004 [Online]. Available: http://ieeexplore.ieee.org/document/1296780/. [Accessed: 02-Jul-2018]

[6] Z. Kamis, E. E. Topcu, and I. Yuksel, “Computer-aided automatic control education with a real-time development system,” *Comput. Appl. Eng. Educ.*, vol. 13, no. 3, pp. 181–191, 2005 [Online]. Available: http://doi.wiley.com/10.1002/cae.20045. [Accessed: 02-Jul-2018]

[7] J. de Jesus Rangel-Magdaleno, J. R. Rivera-Guillen, R. de Jesus Romero-Troncoso, and R. A. Osornio-Rios, “FPGA-Matlab-based open core for three-time controllers in automatic control applications,” *Comput. Appl. Eng. Educ.*, vol. 21, no. S1, pp. E132–E140, Aug. 2013 [Online]. Available: http://doi.wiley.com/10.1002/cae.20526. [Accessed: 02-Jul-2018]

[8] I. Uyanik and B. Catalbas, “A low-cost feedback control systems laboratory setup via Arduino-Simulink interface,” *Comput. Appl. Eng. Educ.*, vol. 26, no. 3, pp. 718–726, May 2018 [Online]. Available: http://doi.wiley.com/10.1002/cae.21917. [Accessed: 02-Jul-2018]

[9] E. Granado, W. Colmenares, M. Strefezza, and A. Alonso, “A web-based virtual laboratory for

teaching automatic control," *Comput. Appl. Eng. Educ.*, vol. 15, no. 2, pp. 192–197, 2007 [Online]. Available: http://doi.wiley.com/10.1002/cae.20111. [Accessed: 02-Jul-2018]

[10] A. Leva and F. Donida, "Multifunctional Remote Laboratory for Education in Automatic Control: The CrAutoLab Experience," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2376–2385, Jun. 2008 [Online]. Available: http://ieeexplore.ieee.org/document/4531704/. [Accessed: 02-Jul-2018]

[11] I. Santana, M. Ferre, E. Izaguirre, R. Aracil, and L. Hernandez, "Remote Laboratories for Education and Research Purposes in Automatic Control Systems," *IEEE Trans. Ind. Informatics*, vol. 9, no. 1, pp. 547–556, Feb. 2013 [Online]. Available: http://ieeexplore.ieee.org/document/6140966/. [Accessed: 02-Jul-2018]

[12] "Android-x86 - Porting Android to x86." 2018 [Online]. Available: http://www.android-x86.org/. [Accessed: 17-Sep-2018]

[13] S. K. Gadi, "skgadi/CIT." GitHub repository, 2018 [Online]. Available: https://github.com/skgadi/CIT

[14] S. K. Gadi, "Control & Identification Toolbox (CIT) – Apps on Google Play," *Google Play*. [Online]. Available: https://play.google.com/store/apps/details?id=com.skgadi.cit. [Accessed: 24-Sep-2018]

[15] S. K. Gadi, "Implementation of basic waveforms used in control systems - Suresh Kumar Gadi." [Online]. Available: https://skgadi.com/posts/basic-waveforms/. [Accessed: 12-Sep-2018]

[16] O. Aflak, "Arduino Library." GitHub repository [Online]. Available: https://github.com/OmarAflak/Arduino-Library

[17] F. Herranz, "UsbSerial." GitHub repository, 2018 [Online]. Available: https://github.com/felHR85/UsbSerial

[18] Giampiero Campa, “Legacy MATLAB and Simulink Support for Arduino - File Exchange - MATLAB Central.” 2016 [Online]. Available: https://la.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink-support-for-arduino. [Accessed: 12-Sep-2018]

[19] “AsyncTask | Android Developers.” [Online]. Available: https://developer.android.com/reference/android/os/AsyncTask. [Accessed: 12-Jun-2018]

[20] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. New Jersey: Prentice Hall, 1991.

[21] P. Ioannou and B. Fidan, *Adaptive control tutorial*. Philadelphia: Siam, 2006.

[22] L. S. Shieh, H. Wang, and R. E. Yates, “Discrete-continuous model conversion,” *Appl. Math. Model.*, vol. 4, no. 6, pp. 449–455, Dec. 1980 [Online]. Available: https://www.sciencedirect.com/science/article/pii/0307904X80901778. [Accessed: 12-Jun-2018]

[23] K. J. Åström and T. Hägglund, *PID controllers*. Research Triangle Park, NC: International Society for Measurement and Control, 1995 [Online]. Available: https://ww2.isa.org/store/products/product-detail/?productId=116103

[24] S. Buso and P. Mattavelli, *Digital control in power electronics*. Morgan & Claypool Publishers, 2006 [Online]. Available: https://books.google.com.mx/books?id=yTc-yzAaRCQC&dq=Digital+Control+in+Power+Electronics&source=gbs_navlinks_s. [Accessed: 12-Jun-2018]