An android application for system identification and automatic control

A. Concha [*] **C. M. Luna-Ochoa** [**] **Set Vejar** [**] **S. K. Gadi** [**]

[*] *Facultad de Ingeniería Mecánica y Eléctrica, Universidad de Colima, Coquimatlán Colima 28400, (e-mail: aconcha@ucol.mx).*
[**] *Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Coahuila, Torreón, Coahuila 27276, (e-mail: Cesarluna8a@hotmail.com setvejar@yahoo.com, research@skgadi.com).*

Abstract: This paper proposes a novel Android application, named Control and Identification Toolbox (CIT), for performing real-time experiments of system response, automatic control, and parameter identification of dynamic systems. It permits tuning a traditional Proportional Integral Derivative controller (PID), whose performance under constant or noise disturbances, introduced by the app, can be analyzed during the experiments. Moreover, this app allows estimating the parameters of first- and second-order linear systems by means of the Recursive Least Squares Method. The parameter estimates, as well as control system signals produced under common test input signals are displayed by the app. It 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. Experimental results obtained using a physical circuit, composed by a first-order low-pass filter, confirm the effectiveness of the proposed application.

Keywords: Android application, real-time experiments, system response, parameter identification, automatic control.

1. INTRODUCTION

The parameter identification and automatic control of systems are topics widely studied in undergraduate engineering programs related to robotics, mechanics, electrical and electronics engineering, chemistry, and biology just to mention some. 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 and processes. Moreover, the parametric estimation of systems is important for designing high-performance controllers, and state observers that estimate variables not available from direct measurements.

In order for the students to corroborate the control and identification theory, it is necessary to perform numerical simulations and experiments of dynamic systems. To this purpose, there are several commercial programs such as MATLAB/Simulink and LabVIEW; however, the infrastructure of hardware and software to teach aspects of parameter identification and automatic control is not always available in the educational institutions. An alternative to these programs is free and open software. In reference Chen et al. (2013), the authors developed an open software package, named as Ch Control System Toolkit, based on C/C++ code, where students can resolve problems like root locus, control design, and time and frequency response of time invariant linear systems. Moreover, references Ayas and Altas (2016) and Demirtas et al. (2013) developed Virtual Control Laboratories (VCLs) for teaching automatic control courses. Ayas and Altas (2016) designed the VCL in Embarcadero RAD Studio's C++, which simulates classical Proportional-Integral-Derivative (PID) controllers and fuzzy logic regulators. Demirtas et al. (2013) create the VCL using LabVIEW in order to design sliding mode and PID controllers for a rotary inverted pendulum. Remote laboratories have also been constructed for distance learning of system identification and automatic control, as described in Granado et al. (2007), Leva and Donida (2008), and Santana et al. (2013).

This article presents a proposed Android application, named Control and Identification Toolbox (CIT), for developing experiments of system response, parameter identification, and automatic control of dynamic systems. Experiments in the CIT application are carried out in real-time using any device based on the Android operating system with a version greater than or equal to 4.0.3. The processor of the Android device executes the identification and control algorithms, and it is communicated with Arduino Uno or Mega boards using the universal serial bus (USB) port; these boards perform data acquisition and generate analog outputs. To our knowledge, the CIT is the first Android application developed for real-time hardware-in-loop 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,

[⋆] Send all correspondance to Research@SKGadi.com

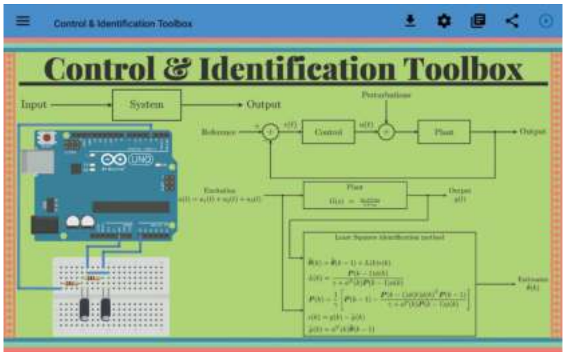

Control & Identification Toolbox
Control & Identification Toolbox
Input
System
Output
Perturbations
Reference
Control
Plant
Output
Excitation
Plant
Output
Least Squares identification method
Estimates

Fig. 1. Home-screen of the CIT android application.

and the Arduino boards are commonly used in engineering courses (Montironi et al., 2017). Additionally, the CIT app can be installed on laptops or personal computers through a virtual box with the program android-x86 (www.android x86.org, 2019), thus allowing its use in a powerful hardware. Through the CIT application, students can experimentally determine and visualize the responses of dynamic systems under common test input signals, such as step functions, sine, sawtooth, triangular, square, and rectangular waves. The system response produced by the CIT application, in turn, can be compared with that obtained analytically. Moreover, through this app, students can tune a conventional PID controller and can identify the parameters of first- and second-order linear systems by means of the Recursive Least Squares method.

The paper is organized as follows. Section 2 describes the CIT Android application and its graphical user interface. Section 3 presents a first-order low-pass filer, that is a physical plant employed to verify the performance of the CIT during real-time experiments. Section 4 shows the system response of this filter under different inputs. The experimental results obtained by identifying and controlling this filter are presented in Sections 5 and 6, respectively. Note that the control objective is that the capacitor's voltage follows a reference input voltage with the least possible error. Finally, concluding remarks are discussed in Section 7.

2. CIT

Fig. 1 shows the home screen of the application that contains an action bar and a main view area. The action bar, located at the top left corner, has a hamburger-icon that opens the menu shown in Fig. 2, and that allows selecting between the following three main categories: system response, parameter identification, and PID controller, which are described in sections 4, 5, and 6, respectively. The home screen in Fig. 1 also has a toolbar located at the right hand side of the action bar, which contains the buttons download, settings, hardware-in-the-loop, and software and hardware. The download button captures the screen content and stores it as a jpg image file. Through the settings button, students define the settings during the real-time experiments, which are grouped in the following three categories: Simulation, Graphs, and Arduino board. 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 menu and buttons. Finally, the software and hardware button displays another view that contains the link to download the firmware of the Arduino Uno or Mega boards, which perform data acquisition. This button 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. Additionally, the software and hardware button allows sharing the screenshot produced with the CIT app using the compatible image sharing apps such as Gmail, WhatsApp, Photos, and so on.

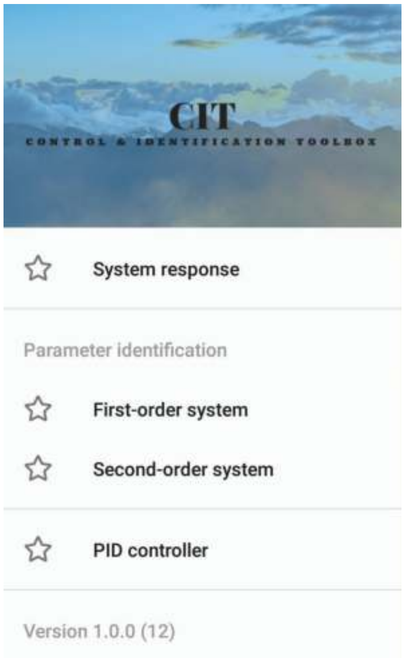


Fig. 2. Menu of the CIT android application.

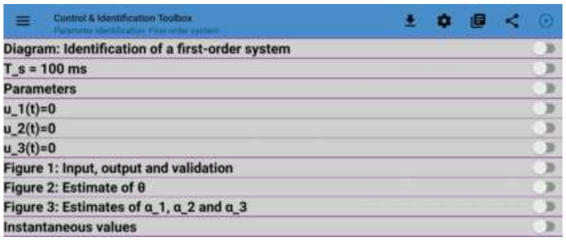

Control & Identification Toolbox
Parameter identification: First-order system

Diagram: Identification of a first-order system
T_s = 100 ms
Parameters
u_1(t)=0
u_2(t)=0
u_3(t)=0
Figure 1: Input, output and validation
Figure 2: Estimate of θ
Figure 3: Estimates of α_1, α_2 and α_3
Instantaneous values

Fig. 3. Experimental results screen (ERS) for first-order parameter identification.

2.1 *Experimental results screen*

The experimental results screen (ERS) is displayed by selecting any of the three main sections: system response, parameter identification, and PID controller. Fig. 3 shows the ERS corresponding to the first-order parameter identification section. The ERS contains the following parts:

(1) Diagram: It displays the diagram block corresponding to the identification or control algorithm.
(2) Sampling time T_s : It lets the user to change the sampling time directly from this screen without using the settings button. In this way, the student can change the sampling time in real-time while an algorithm is running.
(3) Parameters: The tuning of the parameters corresponding to the identification or control algorithms is carried out here.
(4) Input signals ($u_1(t)$, $u_2(t)$ and $u_3(t)$): It permits selecting the reference input to the system, which is the sum of the input signals $u_1(t)$, $u_2(t)$ and $u_3(t)$. Each of them can be any of the following wave functions: step, sine, sawtooth, triangular, square, and rectangular. Gadi (2019) presents the definition of these functions.
(5) Figures: More than one figure can be displayed on the screen. The height of the figures can be adjusted from the settings button.
(6) Instantaneous values: It displays the instantaneous values of the selected signals or the parameter estimates, which helps the student to write them down on the notes instead of estimating them from figures.

2.2 Data acquisition

The CIT application uses Arduino Uno or Mega boards as data acquisition systems, between the Android device 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. This board and the Android device use a USB communication. For most current Android devices, the minimum sampling time T_s achieved for real-time experiments with the app is 5 ms. Therefore, according to the Shannon theorem, the frequency band of the system signals should be less than 100 Hz in order to reconstruct them.

It is worth mentioning that during the experiments, the sampling time can vary due to the computational load and can be different to the sampling time T_s defined by the user. This variable sampling time is called actual. Through, the app the user specifies the maximum error that can be tolerated between the sampling time T_s and the actual sampling time. When this error is reached, the experiment is stopped.

2.3 Firmware

The firmware is taken from Giampiero Campa (2016), which is downloaded into the Arduino board. This firmware converts the Arduino board into a data acquisition card. With this algorithm the Arduino board acts as a USB communications device class, which waits to receive a message from the host that sends the messages "analog-write X in port A" or "analog-read the 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 the port B", means that the Arduino board need to provide the value of the voltage at the port B.

The Arduino board contains a 10-bit analog-to-digital converter, and as a consequence the minimum input voltage that is possible to measure is 5V/(1024)=4.9 mV. Furthermore, the resolution of the Arduino PWM channels is 8 bits, and they work at a frequency of 490 Hz. Then, the PWM values range from 0 to 255, where 0 is equivalent to 0% duty cycle, and 255 means 100% duty cycle.

3. EXPERIMENTAL SET-UP

The CIT application has been designed with the capacity of generating different analog wave forms, that are useful to analyze the behavior of dynamic systems under their effects. The generated analog outs are the next waveforms: step, sine, square, rectangular, triangular, and sawtooth.

A first-order linear dynamic system has the following mathematical model

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

where $u(t)$, $y(t)$, are input and output of the system, and α_1, α_2, α_3, are constant parameters.

The transfer function $Y(s)/U(s)$ of system (1) is given by

$$\frac{Y(s)}{U(s)} = \frac{\alpha_1 s + \alpha_2}{s + \alpha_3} \tag{2}$$

On the other hand, the transfer function of the RC low-pass filter is presented in the following equation

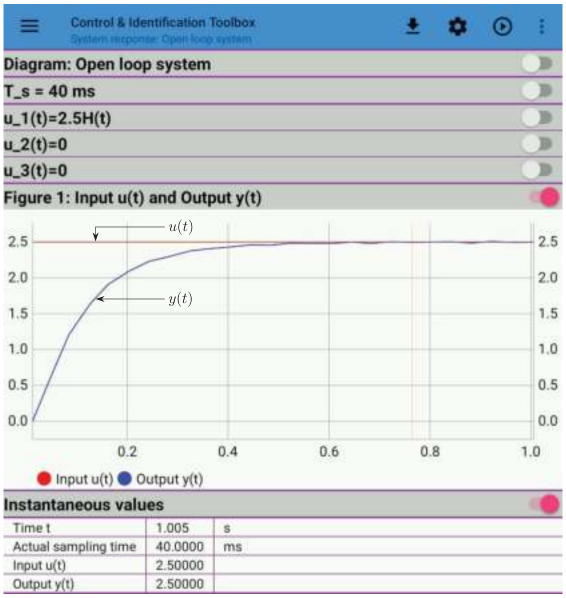
$$\frac{Y(s)}{U(s)} = \frac{\frac{1}{\tau}}{s + \frac{1}{\tau}} \tag{3}$$

where $\tau = RC$ is the time constant, and parameters R and C denote resistance and capacitance, respectively. Comparing equations (2) and (3), yields $\alpha_2 = \alpha_3 = 1/\tau$ and $\alpha_1 = 0$.

All the experiments in this paper are carried out through an Arduino Mega board, and a third generation Moto 3G cellphone with the Android 6.0 operating system. The graphs are refreshed after a time interval of 0.1 s in order to reduce the processing required by the app. The parameters of the low-pass filer are $R = 100$ kΩ and $C = 1$ μF, which produce $\tau = 0.1$ s and $\alpha_2 = \alpha_3 = 10$ s^{-1}. The sampling time T_s defined for all the experiments is $T_s = 0.02$ s. The maximum error that can be tolerated between T_s and the actual sampling time is 1 ms.

4. SYSTEM RESPONSE

The responses plotted by the CIT application when the first-order low-pass filter is excited by a step and a sine



Control & Identification Toolbox

System response: Open loop system

Diagram: Open loop system

T_s = 40 ms

u_1(t)=2.5H(t)

u_2(t)=0

u_3(t)=0

Figure 1: Input u(t) and Output y(t)

Instantaneous values

Time t	1.005	s
Actual sampling time	40.0000	ms
Input u(t)	2.50000	
Output y(t)	2.50000	

Fig. 4. Step response of the first-order low-pass filter.

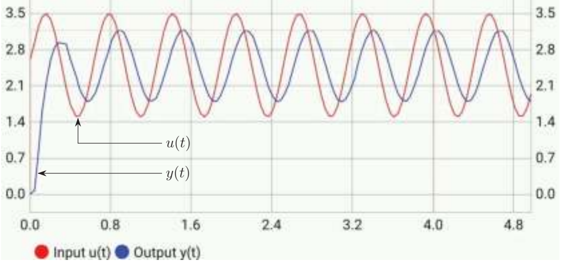


Fig. 5. Sinusoidal response of the fist-order low-pass filter.

wave input signals are shown in Fig. 4 and Fig. 5, respectively. Note that the step input allows determining how fast the system response can reach the amplitude of the input. On the other hand, the sine wave input allow obtaining the frequency response of linear systems.

Let the step input $u(t) = EH(t)$, where $H(t)$ is the Heaviside function and E is a constant parameter. Applying this input to the first-order low-pass filter model in (3) produces (Ogata, 2009)

$$y(t) = E\left[1 - e^{-t/\tau}\right] \tag{4}$$

Fig. 4 shows the response $y(t)$, plotted by the CIT app, 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 (4).

On the other hand, let a sine wave input with offset given by $r(t) = F\sin(\omega t) + \kappa$, where F is the amplitude, ω the angular frequency, and κ the offset. Applying this input to (3) analytically produces the steady-state response $y_{ss}(t)$ in (5), that describes the behaviour of $y(t)$ as t approaches to infinity (Ogata, 2009).

$$y_{ss}(t) = \frac{F}{\sqrt{1+\tau_c^2\omega^2}}\sin\left(\omega t - \tan^{-1}\left(\tau_c\omega\right)\right) + \kappa \tag{5}$$

where frequency $\omega_c = 1/\tau_c$ in rad/s is called cut-off frequency of the filter.

Fig. 5 presents the low-pass filter response $y(t)$ corresponding to a sine wave input signal with parameters $F = 1$, $\kappa = 2.5$ V, and $\omega = \omega_c = 10$ rad/s. It is shown that the amplitude of $y(t)$ is very close to the value $0.707F$ provided by the steady-state response $y_{ss}(t)$ in (5).

5. PARAMETER IDENTIFICATION

The CIT Android application allows to estimate the parameters of first- and second-order linear dynamic systems. The process to realize parameter identification is the following. First, the continuous-time model is converted to a discrete time model by the zero-order-hold discretization method (Oppenheim et al., 1997). Second, the parameters of the discrete-time model are estimated by means of the Recursive Least Squares Method (RLSM) (Ioannou and Fidan, 2006). Afterwards, this model is converted to its continuous-time counterpart.

5.1 *Experiments of parameter identification*

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

$$y(k) = \theta_1 u(k) + \theta_2 u(k-1) + \theta_3 y(k-1) \tag{6}$$

where parameters θ_1, θ_2, and θ_3 are expressed in terms of the unknown parameters α_1, α_2 and α_3; they are defined as

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

where T_s is the sampling period.

The discrete time model (6) can be rewritten in the following vector form

$$y(k) = \boldsymbol{\phi}^T(k)\boldsymbol{\theta} \tag{7}$$

$$\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^T, \quad \boldsymbol{\phi}(k) = [u(k), u(k-1), y(k-1)]^T \tag{8}$$

Expression (7) is called linear parameterization, and it is a linear equation in terms of the unknown vector $\boldsymbol{\theta}$. The CIT Application uses the RLSM in order to estimate the vector parameter $\boldsymbol{\theta}$. This identification method is given by (Ioannou and Fidan, 2006):

$$\begin{aligned}
\hat{\boldsymbol{\theta}}(k) &= \hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{L}(k)\epsilon(k)\\
\boldsymbol{L}(k) &= \frac{\mathbf{P}(k-1)\boldsymbol{\phi}(k)}{\gamma + \boldsymbol{\phi}^T(k)\mathbf{P}(k-1)\boldsymbol{\phi}(k)}\\
\mathbf{P}(k) &= \frac{1}{\gamma}\left[\mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\boldsymbol{\phi}(k)\boldsymbol{\phi}(k)^T\mathbf{P}(k-1)}{\gamma + \boldsymbol{\phi}^T(k)\mathbf{P}(k-1)\boldsymbol{\phi}(k)}\right]\\
\epsilon(k) &= y(k) - \hat{y}(k)\\
\hat{y}(k) &= \boldsymbol{\phi}^T(k)\hat{\boldsymbol{\theta}}(k-1)
\end{aligned} \tag{9}$$

where $\hat{\boldsymbol{\theta}}(k) = \left[\hat{\theta}_1(k), \hat{\theta}_2(k), \hat{\theta}_3(k)\right]^T$ is an estimate of $\boldsymbol{\theta}$, the term γ is called forgetting factor and satisfies $0 < \gamma \leq 1$. Moreover, variable $\mathbf{P}(k) = \mathbf{P}^T(k)$ is called covariance matrix, the signal $\hat{y}(k)$ 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 $\boldsymbol{\theta}$ (Ioannou and Fidan, 2006).

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

The following expressions allow calculating the continuous-time parameters from the discrete-time estimates.

$$\begin{aligned}
\hat{\alpha}_1(k) &= \hat{\theta}_1(k) \\
\hat{\alpha}_2(k) &= \hat{\alpha}_3(k)\left[\frac{\hat{\theta}_2(k) + \hat{\theta}_1(k)\hat{\theta}_3(k)}{1 - \hat{\theta}_3(k)} + \hat{\theta}_1(k)\right] \qquad (10) \\
\hat{\alpha}_3(k) &= -\frac{\ln\left(\hat{\theta}_{3*}(k)\right)}{T_s}
\end{aligned}$$

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

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 $\alpha_1 = 0$, and $\alpha_2 = \alpha_3 = 10\ s^{-1}$ of the first-order linear filter, which is excited with the following input voltage

$$\begin{aligned} u(t) = &0.8\sin(0.4\pi t) + 0.5\sin(1.6\pi t) \\ &+ 0.5\sin(2.4\pi t) + 2.5 \end{aligned} \qquad (11)$$

The estimates $\hat{\theta}_1(t)$, $\hat{\theta}_2(t)$, $\hat{\theta}_3(t)$ and $\hat{\alpha}_1(t)$, $\hat{\alpha}_2(t)$, $\hat{\alpha}_3(t)$ that are obtained from the CIT application are shown in Figures 6 and 7, respectively. Moreover, Figures 6 and 7 present the predicted 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 $y(t)$ and that the estimates $\hat{\alpha}_1(t), \hat{\alpha}_2(t)$ and $\hat{\alpha}_3(t)$ converge to their nominal values α_1, α_2 and α_3 in approximately 2s.

6. PID CONTROLLER

A closed-loop system perturbed by a disturbance $d(t)$ and controlled through a PID regulator is shown in Fig. 8. This controller is defined as (Astrom and Hagglund, 1995)

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \dot{e}(t) \qquad (12)$$

where K_P, K_I, and K_D are constant gains that characterize the contribution of the proportional, integral, and derivative terms of $u(t)$, respectively.

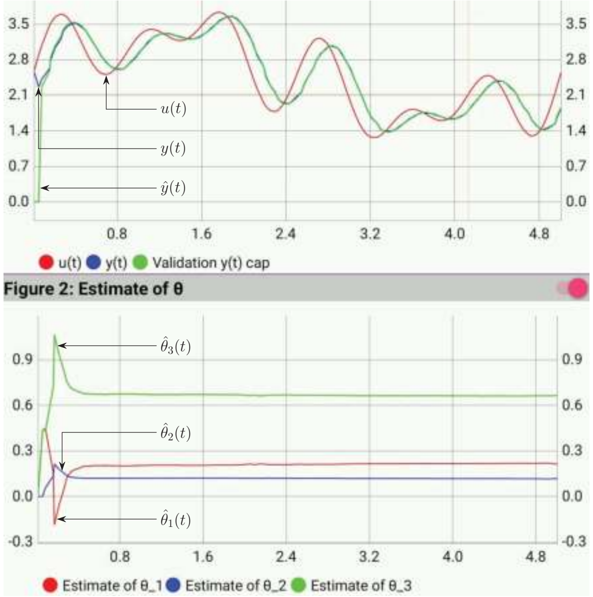
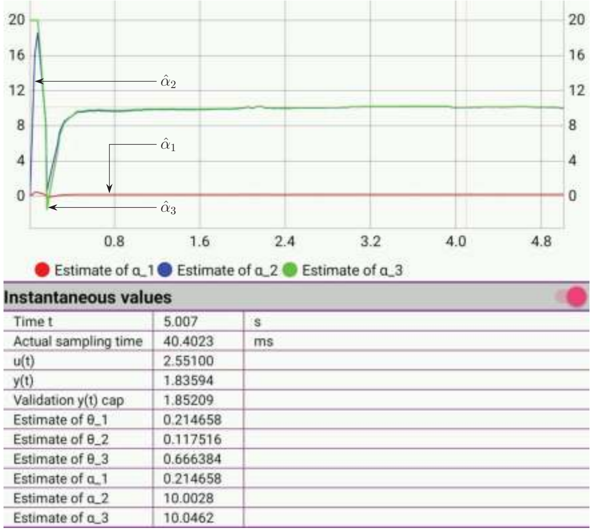


Fig. 6. Estimates $\hat{\theta}_1(t)$, $\hat{\theta}_2(t)$, $\hat{\theta}_3(t)$ and output $\hat{y}(k)$ of the estimated model.



Instantaneous values

Time t	5.007	s
Actual sampling time	40.4023	ms
u(t)	2.55100	
y(t)	1.83594	
Validation y(t) cap	1.85209	
Estimate of θ_1	0.214658	
Estimate of θ_2	0.117516	
Estimate of θ_3	0.666384	
Estimate of α_1	0.214658	
Estimate of α_2	10.0028	
Estimate of α_3	10.0462	

Fig. 7. Estimates $\hat{\alpha}_1(t)$, $\hat{\alpha}_2(t)$, $\hat{\alpha}_3(t)$ and the instantaneous values of signals and parameters.

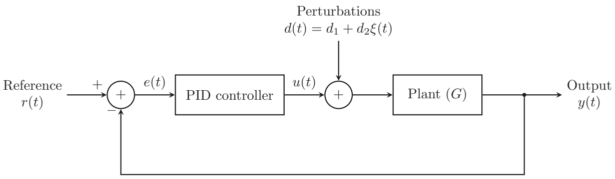


Fig. 8. Closed-loop system with a PID Controller.

The disturbance $d(t)$ is modelled as:

$$d(t) = d_1 + d_2\xi(t) \qquad (13)$$

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

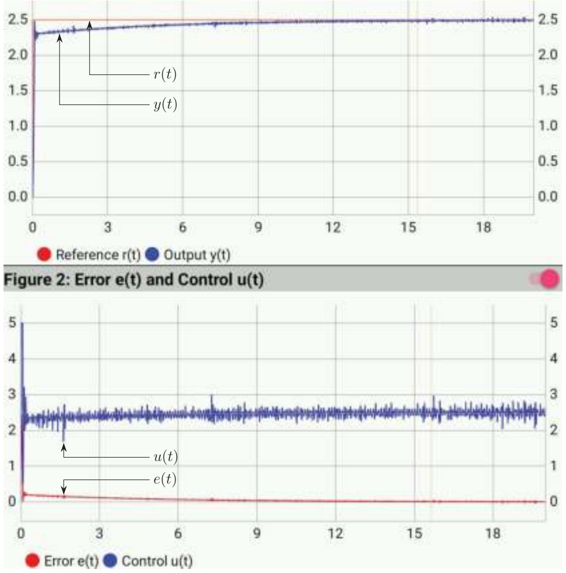

2.5
2.0
1.5
1.0
0.5
0.0
$r(t)$
$y(t)$
0
3
6
9
12
15
18
● Reference r(t) ● Output y(t)
Figure 2: Error e(t) and Control u(t)
5
4
3
2
1
0
$u(t)$
$e(t)$
● Error e(t) ● Control u(t)

Fig. 9. Closed-loop system signals.

6.1 *Experiments using the PID controller*

This section presents the results of an experiment produced with the first-order low-pass filter operating in closed-loop under the PID controller (12), whose gains are $K_P = 10, K_I = 2$, and $K_D = 0$. The reference input $r(t)$ is $r(t) = 2.5H(t)$ V, and the filter is perturbed by means of a constant disturbance $d_1 = 0.05$ V that is generated with the CIT application. It is well known that the effect of the constant disturbance d_1 is eliminated by means of the integral action of the PID controller. Fig. 9 depicts the output $y(t)$ of the experiment, and shows that the $y(t)$ converges to the reference $r(t)$, thus corroborating the aforementioned fact.

7. CONCLUSIONS

This paper proposed the CIT Android application, which was designed 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 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 results validated the identification and control theory behind these systems, where such results were visualized with graphs and instantaneous values.

REFERENCES

Astrom, K.J. and Hagglund, T. (1995). *PID controllers*. International Society for Measurement and Control, Research Triangle Park, NC. URL `https://ww2.isa.org/store/products/product-detail/?productId=116103`.

Ayas, M.S. and Altas, I.H. (2016). A virtual laboratory for system simulation and control with undergraduate curriculum. *Computer Applications in Engineering Education*, 24(1), 122–130. doi:10.1002/cae.21678. URL `http://doi.wiley.com/10.1002/cae.21678`.

Chen, B., Chou, Y.C., and Cheng, H.H. (2013). Open source Ch Control System Toolkit and web-based control system design for teaching automatic control of linear time-invariant systems. *Computer Applications in Engineering Education*, 21(1), 95–112. doi:10.1002/cae.20454. URL `http://doi.wiley.com/10.1002/cae.20454`.

Demirtas, M., Altun, Y., and Istanbullu, A. (2013). Virtual laboratory for sliding mode and PID control of rotary inverted pendulum. *Computer Applications in Engineering Education*, 21(3), 400–409. doi:10.1002/cae.20484. URL `http://doi.wiley.com/10.1002/cae.20484`.

Gadi, S.K. (2019). Implementation of basic waveforms used in control systems - Suresh Kumar Gadi. URL `https://skgadi.com/posts/basic-waveforms/`.

Giampiero Campa (2016). Legacy MATLAB and Simulink Support for Arduino - File Exchange - MATLAB Central. URL `https://la.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink`.

Granado, E., Colmenares, W., Strefezza, M., and Alonso, A. (2007). A web-based virtual laboratory for teaching automatic control. *Computer Applications in Engineering Education*, 15(2), 192–197. doi:10.1002/cae.20111. URL `http://doi.wiley.com/10.1002/cae.20111`.

Ioannou, P. and Fidan, B. (2006). *Adaptive control tutorial*. Siam, Philadelphia.

Leva, A. and Donida, F. (2008). Multifunctional Remote Laboratory for Education in Automatic Control: The CrAutoLab Experience. *IEEE Transactions on Industrial Electronics*, 55(6), 2376–2385. doi:10.1109/TIE.2008.922590. URL `http://ieeexplore.ieee.org/document/4531704/`.

Montironi, M.A., Qian, B., and Cheng, H.H. (2017). Development and application of the ChArduino toolkit for teaching how to program Arduino boards through the C/C++ interpreter Ch. *Computer Applications in Engineering Education*, 25(6), 1053–1065. doi:10.1002/cae.21854. URL `http://doi.wiley.com/10.1002/cae.21854`.

Ogata, K. (2009). *Modern control engineering*. Prentice Hall Upper Saddle River, NJ.

Oppenheim, A.V., Willsky, A.S., and Nawab, S.H. (1997). *Signals & systems*. Prentice Hall. URL `https://books.google.com.mx/books?id=O9ZHSAAACAAJ`.

Santana, I., Ferre, M., Izaguirre, E., Aracil, R., and Hernandez, L. (2013). Remote Laboratories for Education and Research Purposes in Automatic Control Systems. *IEEE Transactions on Industrial Informatics*, 9(1), 547–556. doi:10.1109/TII.2011.2182518. URL `http://ieeexplore.ieee.org/document/6140966/`.

www.android x86.org (2019). Android-x86 - Porting Android to x86. URL `http://www.android-x86.org/`.