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 and learning the system response, automatic control, and the parameter identification of dynamic systems. The CIT app performs real-time experiments and permits tuning a traditional Proportional Integral Derivative controller (PID), whose performance under constant or noise disturbances, introduced by the app, can be analyzed. 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 an universal serial bus (USB), which is connected with Arduino Uno or Mega boards that carry out data acquisition. Experimental results obtained using first and second order low pass filters confirm the effectiveness of the proposed application. Moreover, undergraduate students evaluated the CIT app, and their responses are also presented.

# Keywords

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

# 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 are free and open software. In reference [1], the authors developed an open software package, named as Ch Control System Toolkit, with few lines of 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 [2] and [3] developed Virtual Control Laboratories (VCLs) for teaching automatic control courses. Ayas and Altas [2] designed the VCL in Embarcadero RAD Studio's C++, which simulates classical Proportional-Integral-Derivative (PID) controllers and fuzzy logic regulators. Demirtas et al. [3] create the VCL using LabVIEW in order to design sliding mode and PID controllers for a rotary inverted pendulum. In the literature, there also experimental platforms or control techniques that have been proposed for teaching graduate and undergraduate students. The authors Oliverira et al. [4] propose a methodology for teaching the system identification of open-loop systems using the step response and the particle Swarm optimization. Moreover, Galvao et al. [5] 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, Enikov and Campa [6] developed an aeropendulum experimental setup for demonstration of linear and nonlinear feedback control techniques using MATLAB/Simulink and a Freescale microcontroller. In addition, references [7]–[10], present laboratory setups for the identification and/or control of a DC motor using MATLAB/Simulink, where the experimental kits in [8], [9], and [10] 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, as described in [11], [12], and [13].

This article presents an open-source Android application, called the Control and Identification Toolbox (CIT), for developing experimental practices of system response, parameter identification, and automatic control of dynamic 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 using the universal serial bus (USB) port; these boards perform data acquisition and generates analog outputs. To our knowledge, the CIT is the first Android application developed for real-time hardware-in-the-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, and the Arduino boards are commonly use in engineering courses [14]. Additionally, the application CIT can be installed on laptops or personal computers through a virtual box with the program android-x86 [15], 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 steps 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, by means of 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 CIT application also permits the introduction of constant and noise disturbances to verify the performance of the PID controller. Additionally, the CIT application presents the discrete implementations of both the identification and control algorithms in order for the users to understand their programming.

A set of experiments realized with first and second order low pass filters are presented in this paper to show the results provided by the CIT application. These results have also been reproduced by 25 undergraduate students of the Faculty of Mechanical and Electrical Engineering (FIME) at the Universidad de Colima in Mexico during a set of six experimental practices carried out in a course-workshop. They answered a questionnaire survey, whose results are presented in this paper, which allows determining their satisfaction level on the CIT app.

The paper is organized as follows. Section 2 describes the CIT Android application and its graphical user interface. Section 3 presents the experimental responses of first and second order low pass filers under different inputs. The experimental results obtained by identifying and controlling these filters are presented in Sections 4 and 5, respectively. Section 6 shows a set of questions applied to undergraduate students to know their satisfaction level about the proposed app. Finally, concluding remarks are discussed in Section 7.

# CIT Android application

The Control and Identification Toolbox (CIT) application 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 [16] and the compiled Android PacKage (APK) is published at [17].

Figure 1 shows the home screen of the application that contains an action bar and a main view area. 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: system response, parameter identification, and PID controller, which are described in sections 3, 4, and 5, respectively. The home screen in Figure 1 also has a toolbar located at the right-hand side of the action bar, which contains five buttons that are: 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 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. On the other side, 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.

## 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. Figure 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 : It 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 tuning parameters of the identification or control algorithm are selected here.
4. Input signals (, and ): It permits selecting the reference input to the system, which is the sum of the input signals , and ). Each of them can be any of the following wave functions: step, sine, sawtooth, triangular, square, and rectangular. Reference [18] 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 desired 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 in the notes instead of estimating them from figures.

## 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 [19] and [20]. This communication is selected over the Bluetooth because the USB connection can also be employed for energizing the bridge device and the plant.

## Firmware

The firmware is taken from [21], which is downloaded into the Arduino board. The firmware algorithm 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 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 reply the voltage of the port B.

## Real-time execution

The Model, Figure, and Parameter classes described in Table 1 provide the necessary information to implement the system response, identification or automatic control 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 [22]. Through this software technique, the real-time experiments are performed using hardware-in-loop, where a separated thread runs the identification or control algorithm so that the GUI remains responsive during the experiments.

# System response

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. These inputs are used to produced 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 application and for developing practices during the courses of automatic control and/or parameter identification. These circuits appear in the CIT application when the software and hardware button of the home screen is pressed.

The mathematical models of first and second order linear dynamic systems are described in equations and , respectively

where , , are input and output of the system, and , , , , , , and are constant parameters.

The transfer functions of systems and are respectively given as

and

On the other hand, the transfer functions of RC low pass filters of first and second order are presented in equations (6) and (7), respectively.

where is the time constant, and parameters and denote resistance and capacitance, respectively. Comparing equations and , yields and . Moreover, from equations and we have , and .

All the experiments in this paper use the parameters and , which produces , , , and . Moreover, the sampling time defined for all the experiments is , and the simple moving average is selected as the sampling time type.

## First order system

The responses plotted by the CIT application when the first order low pass filter is excited by a step and a sine wave input signals are shown in Figures 4 and 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 , where is the Heaviside function and is a constant parameter. Applying this input to the first order low pass filter model in produces

Figure 4 shows the response using . Note that at , the response approximately reaches of the amplitude , which corroborates the expression .

On the other hand, let a sine wave input with offset , where is the amplitude, the angular frequency, and the offset. Applying this input to analytically produces the following steady-state response , that is obtained after a large period of time of

where frequency in rad/s is called cut-off frequency of the filter.

Figure 5 presents the response corresponding to a sine wave input signal with parameters , , and  rad/s. It is shown that the amplitude of is approximately , which approximately coincides with the value provided by the steady-state response .

## Second order system

Figure 6 presents the response of the second order low-pass filter under a square wave input signal, which has a unitary amplitude, offset of 2.5 V, and frequency of 1 Hz. It is possible to see that the response of the second order low filter is smother than its input.

# Parameter identification

The CIT Android application allows to estimate the parameters of first and second order linear dynamic systems described by equations and , respectively. 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 [23]. Second, the parameters of the discrete time model are estimated by means of the Recursive Lest Squares Method (RLSM) [24]. Afterwards, this model is converted to its continuous-time counterpart.

## Parameter identification of a first-order system

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

,

where parameters , , and are expressed in terms of the unknown parameters , and as follows

,

where is the sampling period.

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

Expression is called linear parameterization, since it is simply a linear equation in terms of the unknown vector . The CIT Application uses the RLSM in order to estimate the vector parameter . This identification method is given by [24]:

where is an estimate of , the term is called forgetting factor and satisfies . Moreover, variable is called covariance matrix, the signal is the predicted output, and is the prediction error. The initial matrix is a diagonal matrix given by , where is a large value and is the identity matrix.

**Remark:** The estimate vector converges to if and only if the input has at least different frequencies, where is the number of parameters of [24].

According to the last remark, the signal must have at least two frequencies so that converges to .

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

,

where is a parameter projection of that takes only positive values.

An experiment is carried out with the CIT Application to identify the nominal parameters, and of the first-order linear filter. The experiment uses the RLSM parameters: and . Moreover, the filter is excited with the following input voltage

 V

The estimates , , and , , , that are obtained from the CIT application are shown in Figures 7 and 8, respectively. Additionally, Figures 7 and 8 present the prediction output 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 is very close to the system output , and that the estimates , , converge to their nominal values , , in approximately 2 s.

## Parameter identification of a second-order system

The continuous-time model in can be discretized as:

.

An estimate of the parameter vector in is determined through the RLSM given in . Since five parameters are estimated, convergence of to requires that  has at least three different frequencies.

For estimating the parameters , , , and of the continuous-time model , the following procedure is carried out.

* In each sample time the CIT app computes the matrices

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

* Matrices in are used to obtain

where , , and correspond to the following continuous-time counterpart of , which are computed using the Geometric series method [25];

Finally, in each sample time the estimates , , , and are computed as follows

where equations in equation result by equaling the transfer function in with the following transfer function corresponding of the state space system in and

The CIT application is used to estimate the parameters , , , , , , , , , and corresponding to the second-order linear model in , whose input is given by the expression in . The RLSM parameters are and . Figure 9 depicts the parameters , , , and of the continuous-time model. Note that these estimates converge close to their nominal values, which are , and .

# Proportional Integral Derivative (PID) controller

The CIT application implements a PID algorithm, which is the more widely used controller in the industry and real applications. The closed-loop system for a PID controller is shown in Figure 10, where and are the reference input and the output of the system, respectively. The error signal is the difference between and , i.e., . The system is perturbed by a disturbance , and is controlled through the PID controller, which produces a control signal that reduces to zero or to a small value. The PID controller is defined by [26]

where , , and are constant gains that characterize the contribution of the proportional, integral, and derivative terms of , respectively.

The disturbance is modelled as

where is a constant disturbance, is the zero mean white noise, and is its power.

The PID controller is implemented in the CIT application by the following discrete-time version [26]

where the constants , and are defined by

It is worth mentioning that the integral and the derivative terms of the PID controller are approximated by means of the Trapezoidal and the Backward Euler methods [27], respectively.

## PID controller for a first-order system

This section presents the results of an experiment produced with the first order low pass filter operating in closed-loop under the PID controller , whose gains are , , and . The reference input is set to V, and the filter is perturbed by means of a constant disturbance V that is generated with the CIT application. It is well known that the effect of the constant disturbance is eliminated by means of the integral action of the PID controller. Figure 11 depicts the output of the experiment and shows that the output converges to the input , thus corroborating the aforementioned fact.

## PID controller for a second-order system

An experiment is realized using the CIT application by connecting the second order filter to the PID controller and configuring the input as a triangular wave with unitary amplitude, offset of 2.5 V, and frequency of 0.25 Hz. In this experiment, the system is perturbed by a constant disturbance V and white noise with a power of . Furthermore, the gains of the PID controller are fixed to , , and . Figure 12 shows the obtained results, where the output signal is very close to the reference input . Note that this Figure shows high frequency components in the control signal , which are associated to the white noise perturbation .

# Student evaluation

In order to analyze the satisfaction of using the CIT app for learning system identification and automatic control, a set of 25 undergraduate students of the FIME at the Universidad de Colima were invited to respond thirteen statements S1, S2, …, S13, which are shown in Table 5. These students were enrolled in a course-workshop entitled “Using the CIT Android application for experiments of automatic control and system identification”. The course-workshop is a teaching-learning methodology characterized by the interrelation between theory and practice. Here the instructor exposes the theoretical and procedural foundations, which serve as the basis for students to perform a set of activities or practices previously designed in order to link conceptualization and implementation [28].

The aforementioned course-workshop was composed by a number of 8, 2, 5, 9, and 1 students of the 2nd, 4th, 6th, 8th, and 10th semesters of the Mechatronic engineering program respectively. All the students of the 2nd and 4th semesters had not been enrolled in courses of automatic control and system identification before they took the course-workshop. It had a duration of 15 hours divided in 3 sessions of 5 hours per day, during the period of June 3-5, 2019. In each day, the students carried out two experimental practices using the CIT app, as well as the first and second order filters described in Section 3. All the experimental practices are found at [29] and are the following. P1:First ordersystem response; P2: Second order system response; P3*:* Estimation of the parameters of a first order system; P4: Estimation of the parameters of a second order system; P5: PID controller applied to a first order system; and P6: PID controller applied to a second order system.

A scale from 0 to 10 was used to evaluate the statements S1, S2, …, S13. A score between the interval 0-2 means completely disagree (CD), 3-4 corresponds to disagree (D), 5-6 means indifferent (I), 7-8 corresponds to agree (A), and 9-10 indicates completely agree (CA). Table 5 shows the scores for these statements, as well as their corresponding average denoted as AVG. Note that the students’ evaluations were quite positive, since all of the statements S1, S2, …, S13 have an average score ranked as agree or completely agree, and they corroborate that most of the students were engaged and motivated with the classroom/laboratory activity.

The 25 students also answered the questions Q1, Q2, and Q3from Table 6 twice. Through this activity, we could know their responses with the CIT tool compared without it. In other words, they answered these questions at the beginning of the course-workshop before using the CIT app, and they answered the same questions at the end of the course-workshop after using it. Questions Q1 and Q2 were evaluated through a scale from 0 to 10, where 0 means unenthusiastic and 10 means enthusiastic. Moreover, question Q3was answered as yes or not. It is possible to observe in Table 6 that the average score of questions Q1 and Q2is very high and similar before and after the students used the CIT tool, which implies that the course-workshop with this tool met their expectations. Moreover, Table 6 shows that at the end of the course-workshop, the number of students with practical skills in system identification and automatic control increased 16%, thus confirming the utility of this app for teaching or learning system identification and automatic control.

In addition, the students were asked to provide feedback on the CIT app through the questions F1, F2 and F3 in Table 7, whose responses were used to know the positive and negative experiences of the students by using the CIT tool. Moreover, through this survey they gave us some recommendations to increase the use of the CIT tool in courses. All the answers for these questions are also summarized in Table 7. As positive experiences, students highlight the importance of knowing the CIT tool that will be a useful for their career. Moreover, the students suggested to improve the interface of the app, and to produce versions running in more operating systems. Finally, they suggested promoting the app in order to increase its utility in control and identification courses.

1. **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 methods 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, 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 was verified through a questionnaire survey applied to undergraduate students, in a course-workshop related to automatic control and system identification, that the CIT application provoked that most of them were enthusiastic and engaged with the classroom activity and the course-workshop was interactive. As a future work, we propose to integrate the CIT app as an experimental tool for all the control and identification courses of our faculties.

# Acknowledgements

The authors thank the anonymous reviewers for their valuable comments, as well as the students enrolled in the course-workshop.

# 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] M. Demirtas, Y. Altun, and A. Istanbullu, “Virtual laboratory for sliding mode and PID control of rotary inverted pendulum,” *Comput. Appl. Eng. Educ.*, vol. 21, no. 3, pp. 400–409, Sep. 2013 [Online]. Available: http://doi.wiley.com/10.1002/cae.20484. [Accessed: 04-Feb-2019]

[4] 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]

[5] 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]

[6] E. T. Enikov and G. Campa, “Mechatronic Aeropendulum: Demonstration of Linear and Nonlinear Feedback Control Principles With MATLAB/Simulink Real-Time Windows Target,” *IEEE Trans. Educ.*, vol. 55, no. 4, pp. 538–545, Nov. 2012 [Online]. Available: http://ieeexplore.ieee.org/document/6193186/. [Accessed: 04-Feb-2019]

[7] 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]

[8] 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]

[9] 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]

[10] 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]

[11] 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]

[12] 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]

[13] 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]

[14] M. A. Montironi, B. Qian, and H. H. Cheng, “Development and application of the ChArduino toolkit for teaching how to program Arduino boards through the C/C++ interpreter Ch,” *Comput. Appl. Eng. Educ.*, vol. 25, no. 6, pp. 1053–1065, Nov. 2017 [Online]. Available: http://doi.wiley.com/10.1002/cae.21854. [Accessed: 04-Feb-2019]

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

[16] S. K. Gadi, “skgadi/CIT.” GitHub repository, 2018 [Online]. Available: https://github.com/skgadi/CIT

[17] 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]

[18] 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]

[19] O. Aflak, “Arduino Library.” GitHub repository [Online]. Available: https://github.com/OmarAflak/Arduino-Library

[20] F. Herranz, “UsbSerial.” GitHub repository, 2018 [Online]. Available: https://github.com/felHR85/UsbSerial

[21] 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]

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

[23] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals &amp; systems*. Prentice Hall, 1997 [Online]. Available: https://books.google.com.mx/books?id=O9ZHSAAACAAJ. [Accessed: 27-Nov-2018]

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

[25] 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]

[26] 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

[27] 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]

[28] A. Saroyan and C. Amundsen, *Rethinking teaching in higher education : from a course design workshop to a faculty development framework*. Stylus Pub, 2004.

[29] A. Concha and S. K. Gadi, “CIT manuals - Suresh Kumar Gadi,” 2019. [Online]. Available: https://skgadi.com/posts/cit-manuals/. [Accessed: 27-Jun-2019]