Interactive Educational Tool for Compensators Design in MATLAB® Using Frequency Response Analysis

JUAN MANUEL RAMIREZ-CORTES,[1] VICENTE ALARCON-AQUINO,[2] PILAR GOMEZ-GIL,[3]
ALEJANDRO DIAZ-MENDEZ,[1] MARIANA IBARRA-BONILLA,[1] IRMA GARCÍA-ENRIQUEZ[4]

[1]*Department of Electronics, National Institute of Astrophysics, Optics, and Electronics, Tonantzintla, Puebla, Mexico*

[2]*Department of Electronics, University of the Americas, Cholula, Puebla, Mexico*

[3]*Department of Computer Science, National Institute of Astrophysics, Optics, and Electronics, Tonantzintla, Puebla, Mexico*

[4]*Department of Mechatronics, Atlixco Technological Institute, Puebla, Mexico*

Received 20 October 2011; accepted 5 March 2012

ABSTRACT: This article presents an educational platform developed to support the teaching of compensators design in a basic control theory course. The application consists of a graphical user interface in MATLAB®, and further connection to the plant under study through the data acquisition toolbox, and a data acquisition card. The developed system allows the students to experiment with parameter changes in the controllers under study, such as gain, overshoot, settling time, and peak time, and visualize results obtained from simulated or real signals. The methodology is based on the frequency response analysis. Typical Bode, root locus, and unit step response plots are easily obtained for a system before and after compensation, in a dynamical way. A modular design allows the students to easily upgrade the application in order to include further methodologies. Results derived from its use in undergraduate and graduate courses are presented. MATLAB is a registered trademark of The MathWorks, Inc. © 2012 Wiley Periodicals, Inc. Comput Appl Eng Educ 22:699–707, 2014; View this article online at wileyonlinelibrary.com/journal/cae; DOI 10.1002/cae.21562

Keywords: control; compensators; interface; MATLAB; education

INTRODUCTION

The use of interactive software tools in instruction is particularly significant because it provides an immediate insight of the theoretical concepts involved in every field of engineering education [1,2]. There are several high-level software authoring packages that have been used to develop engineering education applications, such as MATLAB, LabVIEW, MathCAD, MATHEMATICA, Maple, and others [3–6], as well as JAVA or c++ for the developing of web-based educational resources [7–9]. Among these, MATLAB [10] has emerged as a standard language for science and technology due to its flexibility and ease of use, as well as the availability of a large number of toolboxes that have been developed for many different applications. Particularly, the data acquisition toolbox allows interfacing a physical system using the corresponding acquisition hardware for a number of applications. This article describes an application whose purpose is to provide a tool to be used as a teaching aid in basic control courses, specifically in the design of controllers and compensators. The purpose was to design an intuitive and flexible tool that the students could use to experiment freely with the control techniques, without getting overly involved in programming. The application has been developed in MATLAB and Simulink taking advantage of the toolboxes about data acquisition and control. MATLAB is now available in any university or industry, and it is used, among many other things, in the design of control systems. As users of MATLAB knows, Simulink is a graphical block diagramming tool that works in conjunction with MATLAB through an intuitive interface based

---

Correspondence to J. M. Ramirez-Cortes (jmram@inaoep.mx, jmram57@hotmail.com).

© 2012 Wiley Periodicals, Inc.

on manipulation of operational blocks, using drag-drop operations. The use of MATLAB for teaching and research is supported by an extensive literature that provides examples of applications in various engineering topics, such as chemical engineering, media signal processing, digital signal processing, electromagnetic theory, communications, engineering design, metallurgy, control systems, and many others [11–15]. Numerous toolboxes have been developed for MATLAB to facilitate a variety of engineering and educational tasks such as algorithm development, modeling, simulation, data analysis, visualization, engineering graphics, and application development in many fields; however, use of these toolboxes requires certain length of time to master them, thus discouraging novice users, in some cases, to use them.

The educational graphical user interface presented in this work aims to help in reducing the time spent in developing computational tasks to solve problems and reinforce the students understanding of theoretical principles through simulations and interactive practice. In the past years, several applications based on MATLAB with engineering education purposes have been reported: digital image processing, electromagnetic theory, biomedical engineering, neural networks, power systems, fuzzy control, and many others [16–20]. There are several works reported in the literature, which address from various perspectives the vast field of education on control theory. Differences with the work described in this communication reside in the used approach of the specific technique involved in the controller design [21–23], whether the web is involved or not for interacting with the tool [9,23,24], or whether the access to the system through dedicated hardware is considered or not [25–27]. The main advantage of the control design software package presented in this article is described as follows. It makes use of operations contained in the MATLAB control toolbox, accessed through a graphical user interface which allows, even to the novice user, to do experimentation with the design of controllers, and the effect of parameters changing, without any previous knowledge of MATLAB or the control toolbox. Furthermore, the interface allows the student to interact with the real world using the data acquisition capabilities. In that way, the student gets an instantaneous insight on the performance of the designed controller. Using the software application described in this article, the student is less prone to errors as compared to the case of having to program each line of code in any given language. Classical control design methods are normally based on several assumptions derived from the specifications, which allow the simplification of the design procedures. Usually, the controllers and compensators designed give students some insight into the control problem, providing an initial approach to the problem solution. However, further effects derived from nonidealities can produce some deviations from the desired specifications. For this reason, it is a standard procedure the use of an iterative procedure in which the students simulate the system, controllers, and compensators, analyzing in each step the effect in the intermediate results, until they get the fulfillment of the specifications. This iterative procedure can be very time consuming and confusing without computational tools that automates much of the iteration and gives good insights about the progress of the design. With the graphical user interface described in this article, the students can solve in a graphical environment most of the exercises presented on the topic of compensators in a basic control theory course. The rest of the article is organized as follows. The next section presents a review on the principles of compensators design, and the frequency response methodology used in this work. The following section describes the developed MATLAB-based graphical user interface for the design and implementation of compensators. Concluding remarks about the didactic relevance of the project are discussed in the final section.

COMPENSATORS DESIGN

Compensators theory in control systems is a topic extensively presented in detail in many textbooks, following several approaches. Roughly, compensators aim to improve the transient response of a system, as well as the steady-state error characteristics, by augmenting or compensating the system with additional poles and zeros. This is achieved by placing the compensator either in cascade with the plant, or in the feedback path. In every case the open-loop poles and zeros change, thereby creating a new root locus that goes through the desired closed-loop pole location [28]. As part of the process in many design techniques, it is usual to observe a tradeoff between system-gain, steady-state error, percent overshoot, and so on. In general, root locus techniques require repeated trials to find the desired design point. In that sense, frequency response techniques can take advantage of the availability of computer programs, such as MATLAB, in the design of compensators. A very interesting discussion on the value of interactivity associated with design of control systems can be found in Ref. [2]. When designing a cascade compensation to improve the transient response, the open-loop frequency response is reshaped to meet both the phase margin requirement (percent overshoot), and the bandwidth requirement (settling or peak time). Thus, this reshaping can lead to several trials until all transient response requirements are met. The approach used for the application presented in this article is based on the methodology described in detail in Ref. [28] for cascade compensators design. A short summary of the design procedure corresponding just to the *LEAD* compensation case, for illustrative purposes, is described as follows.

LEAD COMPENSATOR; DESIGN PROCEDURE

Consider a unity negative feedback system with an open-loop transfer function represented by $G(s)$, and a cascade compensator transfer function represented by $G_c(s)$. The closed-loop transfer function without any compensation is given by Equation (1) as follows:

$$T(s) = \frac{G(s)}{1 + G(s)}, \tag{1}$$

Restricting the analysis to a second order unity feedback system, with ω_n and ζ representing the natural frequency and the damping factor, respectively, the open-loop function is given by Equation (2).

$$G(s) = \frac{\omega_n^2}{s(s + 2\zeta\omega_n)}, \tag{2}$$

The closed-loop transfer function is then given by Equation (3).

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \tag{3}$$

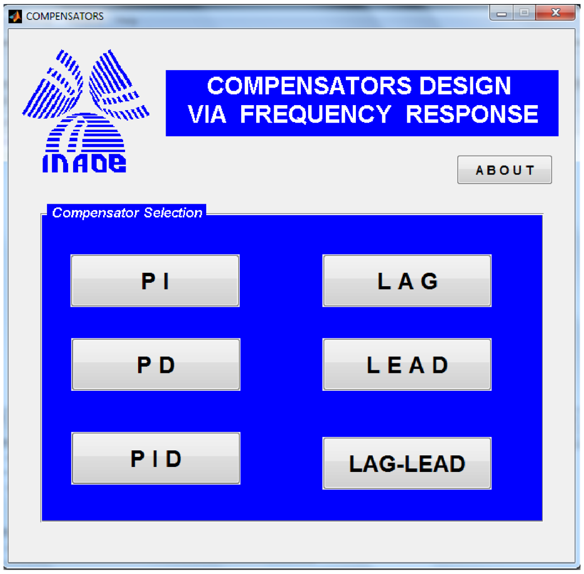


Figure 1 Graphical user interface main screen. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

When designing lead compensators via Bode plots, the objective is usually to change the phase diagram, increasing the phase margin to reduce the percent overshoot, and increasing the gain crossover to obtain a faster transient response. By increasing the gain crossover frequency the lead compensator increases the bandwidth, and simultaneously, the phase diagram is raised at higher frequencies. The consequence is a larger phase margin and a higher phase-margin frequency. The effect in the time domain is the presence of lower percent overshoots with smaller peak times.

The transfer function of the lead network is given by Equation (4), for $\beta < 1$.

$$G_c(s) = \frac{1}{\beta}\left(\frac{s+(1/T)}{s+(1/\beta T)}\right) \tag{4}$$

The dc gain of the compensator is set to unity in order not to change the dc gain designed for the static error constant when the compensator is inserted into the system.

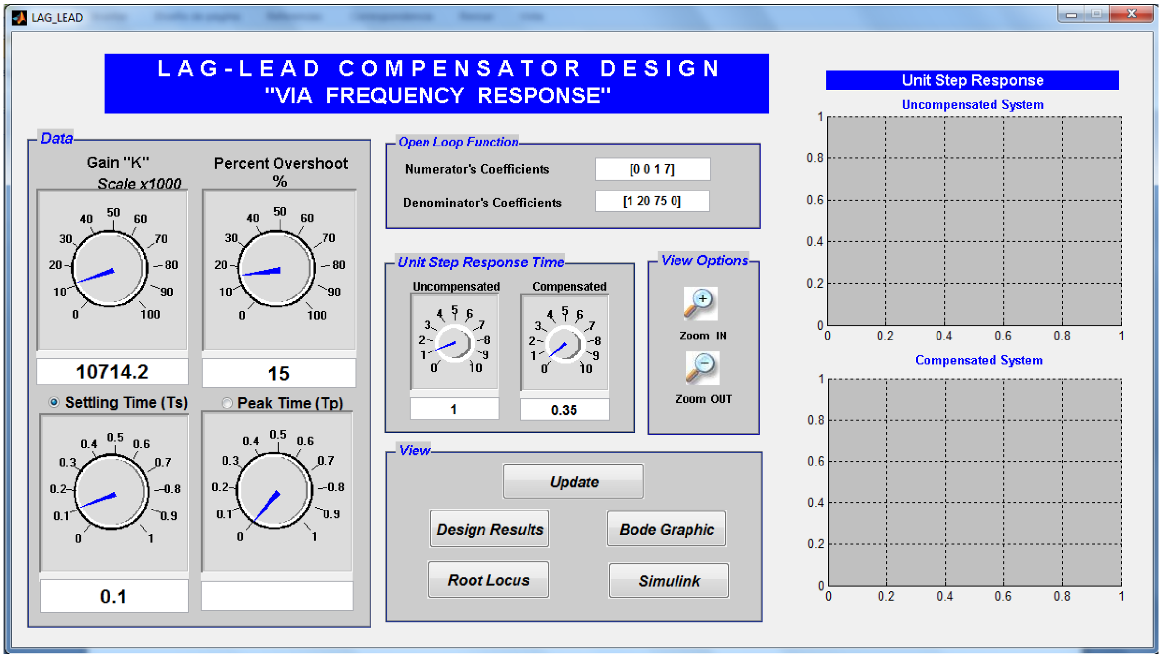
As initial procedure, the program asks the user to input the desired values by defining: numerator and denominator coefficients corresponding to the transfer function of the plant, overshoot (OS), system gain (K), and either the peak time (T_p) or the settling time (T_s). The transfer function is then obtained from the numerator and denominator coefficients and the system gain K. The damping factor ζ is obtained according to Equation (5).

$$\zeta = \frac{-\log(\mathrm{OS}/100)}{\sqrt{\pi^2 + \log(\mathrm{OS}/100)^2}}. \tag{5}$$

Using the obtained values, the closed-loop bandwidth required to meet the specifications of settling time, peak time, or rise time is obtained as:

$$\omega_{\mathrm{BW}} = \omega_n\sqrt{1 - 2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}, \tag{6}$$

where ω_n is given by either Equation (7) or (8), depending on the parameter that was initially specified; settling time T_s or the



Gain "K" Scale x1000: 10714.2

Percent Overshoot %: 15

Settling Time (Ts): 0.1

Numerator's Coefficients: [0 0 1 7]

Denominator's Coefficients: [1 20 75 0]

Unit Step Response Time — Uncompensated: 1; Compensated: 0.35

Figure 2 Window accessed through the "LAG-LEAD" option. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

peak time T_p, respectively.

$$\omega_n = \frac{4}{\zeta T_\mathrm{s}} \tag{7}$$

$$\omega_n = \frac{\pi}{T_\mathrm{p}\sqrt{1-\zeta^2}}. \tag{8}$$

The next step consists of finding the phase margin to meet the damping ratio or percent overshoot requirement. The required compensator phase angle φ_c is obtained through an iterative process, and then is used in calculating the beta value according to Equation (9).

$$\beta = \frac{1-\sin\varphi_\mathrm{c}}{1+\sin\varphi_\mathrm{c}} \tag{9}$$

At this point, it is required to determine the compensator's magnitude at the peak of the phase curve, and the new phase-margin frequency. Zero (z_L), pole (p_L), and gain (K), of the lead compensator transfer function, are then determined according to Equations (10)–(12).

$$z_\mathrm{L} = \omega_\mathrm{max}\sqrt{\beta} \tag{10}$$

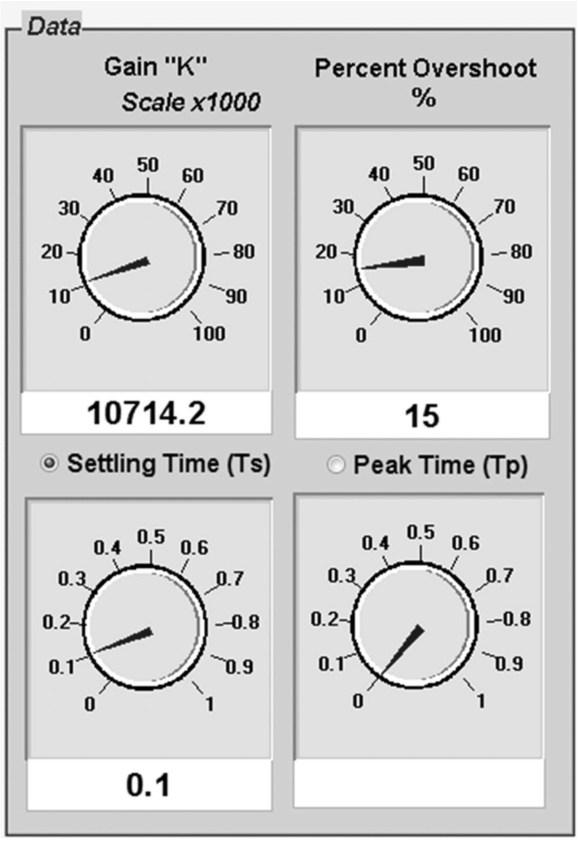
$$p_\mathrm{L} = \frac{z_\mathrm{L}}{\beta} \tag{11}$$

$$K = \frac{1}{\beta} \tag{12}$$

Using these values, the compensator transfer function is finally obtained. The compensator is added in cascade structure to the plant. At this point, simulation is required to be sure that all requirements are met, with the possibility of redesign if necessary. The system performance can be easily evaluated through typical Bode, root locus, and unit step response plots, before and after compensation, in a dynamical way. The design procedures used for the other compensators included in the application follow a similar methodology. Procedure details can be consulted in Ref. 28.

THE COMPENSATORS DESIGN GRAPHICAL USER INTERFACE

Figure 1 shows the main screen of the interface. From this window, the user is able to select the compensator type to be used in the simulations. In the described application, the following choices are available: proportional-integral (PI), proportional-derivative (PD), proportional-integral-derivative (PID), lag, lead, and lag-lead compensators. By clicking the desired option the user accesses a new screen with several input blocks. Figure 2 shows, by instance, the screen corresponding to lead-lag compensator design. Every option displays some default values, which can be easily changed at the designer convenience. Figure 3 shows the window used to enter the numerator and denominator coefficients defining the transfer function of the plant in open loop.



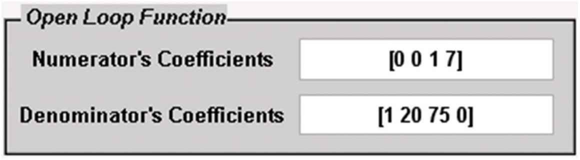
Gain "K" Scale x1000: 10714.2

Percent Overshoot %: 15

● Settling Time (Ts): 0.1

○ Peak Time (Tp):

Figure 4 Parameter definition window for the LAG-LEAD compensator option.

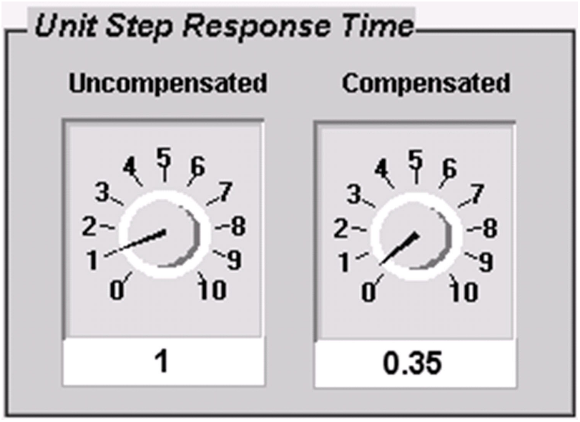


Numerator's Coefficients [0 0 1 7]

Denominator's Coefficients [1 20 75 0]

Figure 3 Window "Open Loop Function."

In the case of choosing the compensators PI and LAG, the user has the possibility to specify the values of the gain (K) of the system and the overshoot percentage, either using the buttons or introducing directly the desired value in the space



Uncompensated: 1

Compensated: 0.35

Figure 5 Window "Unit Step Response."

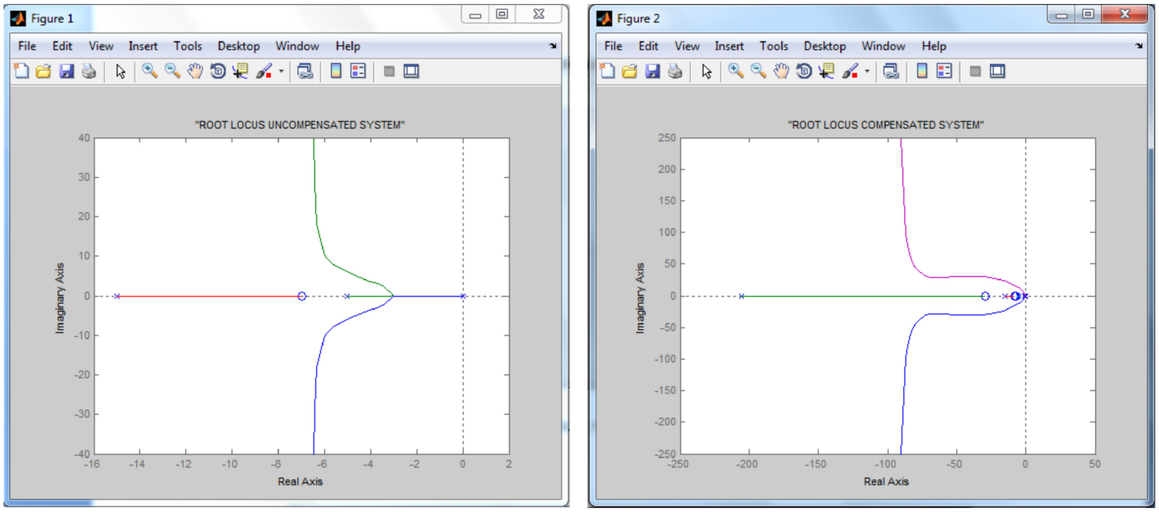
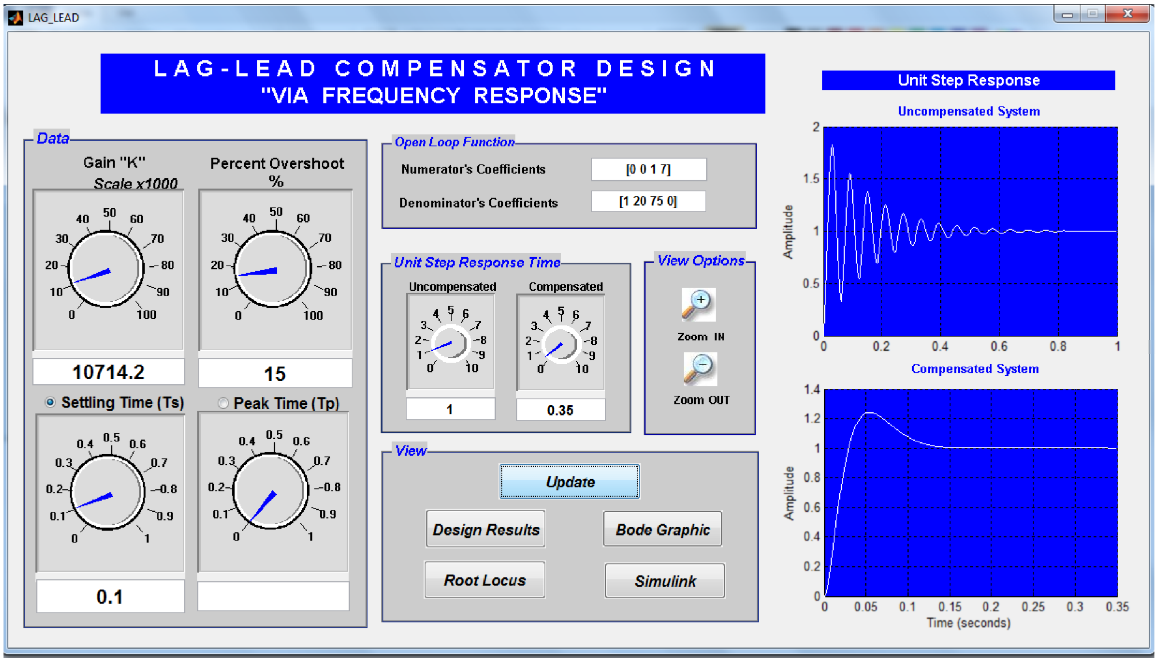


Figure 6 System root locus with and without compensation. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

located under the corresponding bar. For the rest of the compensators (PD, PID, LEAD, and LAG-LEAD), besides the two previous parameters (gain and overshoot), the user can specify the values of settling time (Ts) and peak time (Tp), as it is shown in Figure 4. In a similar way to the previous options, the user can select the desired value by means of the corresponding button or introducing the values in the space located specifically for that purpose. Figure 5 shows the window named "*Unit step response.*" This option allows the user to define the appropriate scale of time in order to analyze the unit step response in both cases: compensated and noncompensated system.

By selecting an option in the block "View", the user can access some choice, depending on the desired output plot: the root locus, system unit step response, magnitude Bode plot, or phase Bode plot. In every case the two plots corresponding to the compensated and uncompensated cases are displayed. The root locus of the system in both cases, compensated and noncompensated, can be easily obtained by selecting the option named "Root Locus", which is located in the screen shown in Figure 2, and typical result plots are shown in Figure 6. Some examples of the displayed graphics corresponding to the magnitude and phase Bode plots are shown in Figures 7 and 8. These


LAG-LEAD COMPENSATOR DESIGN
"VIA FREQUENCY RESPONSE"

Data

Gain "K" *Scale x1000*: 10714.2

Percent Overshoot %: 15

● Settling Time (Ts): 0.1

○ Peak Time (Tp):

Open Loop Function

Numerator's Coefficients: [0 0 1 7]

Denominator's Coefficients: [1 20 75 0]

Unit Step Response Time

Uncompensated: 1

Compensated: 0.35

Figure 7 Unit step response with and without compensation. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

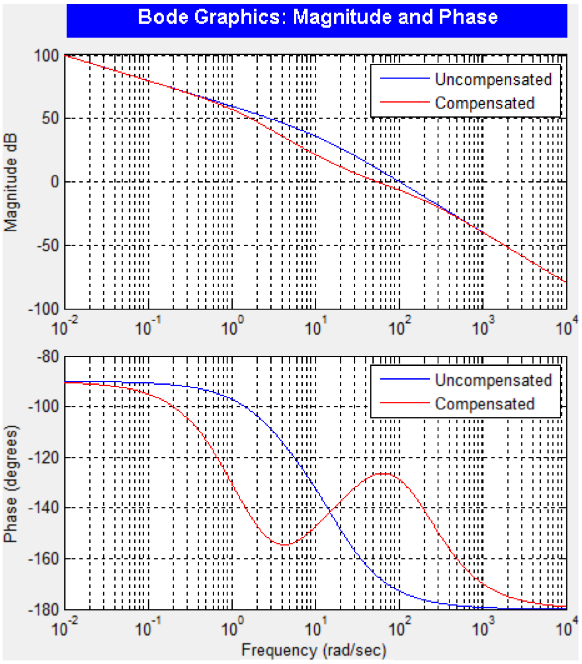


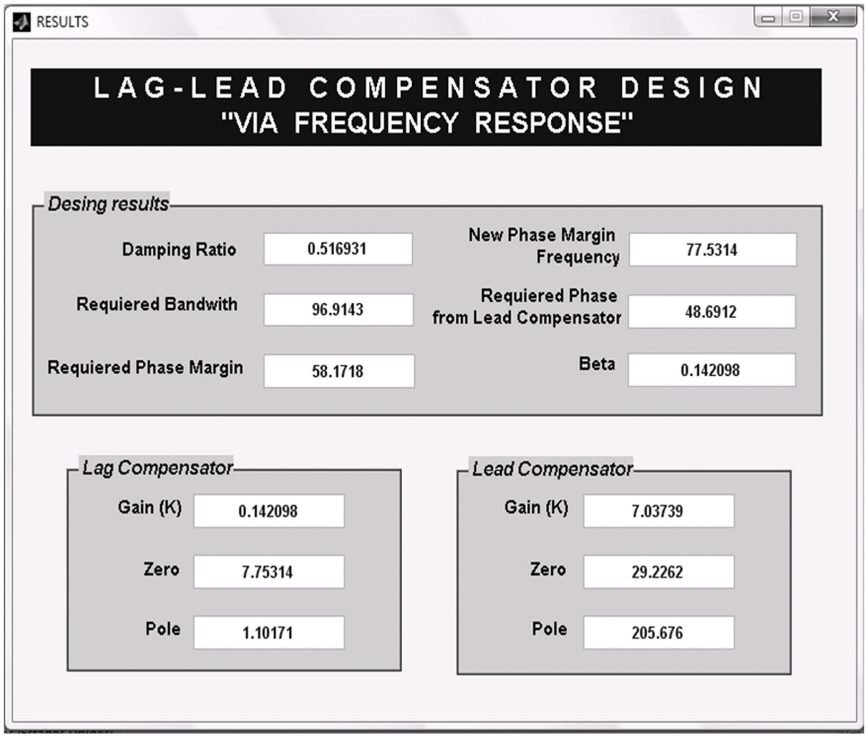
Figure 8 Magnitude and phase Bode plots for the system before and after compensation. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

plots are updated every time a parameter change occurs. For a numerical evaluation of the system, the system provides an option named "Design Results", which displays the screen shown in Figure 9, with the parameters obtained in each case.

The application allows the user to analyze the block diagram of the system under study by accessing the Simulink option, which is located in the inferior part of the main screen shown in Figure 2. A typical Simulink block diagram is shown in Figure 10.

HARDWARE SETUP FOR REAL TIME EXPERIMENTATION

The effect of the designed compensators can be immediately visualized in real-time experiments by taking advantage of the predefined acquisition functions in the MATLAB *Data Acquisition Toolbox.* Figure 11 shows a setup used to do some experiments on AC motor-speed control, which basically consists of the acquisition module DT-9812 from *Data Translation*, connected to the USB port of the laptop computer, a variable-speed drive, an AC 1/3 hp motor, and a speed sensor. System identification is performed experimentally using a first order model, with a motor transfer function given by Equation (12), where K_m and τ_m represent the gain constant and time constant, respectively. These parameters depend on physical characteristics such as motor torque, moment of inertia, frictional coefficient, and so on [29]. The values obtained from an experimental procedure are $K_m = 3$ and $\tau_m = 0.089$ s. Figure 12 shows the screen of the graphical user interface with



LAG-LEAD COMPENSATOR DESIGN
"VIA FREQUENCY RESPONSE"

Desing results

Damping Ratio: 0.516931
New Phase Margin Frequency: 77.5314
Requiered Bandwith: 96.9143
Requiered Phase from Lead Compensator: 48.6912
Requiered Phase Margin: 58.1718
Beta: 0.142098

Lag Compensator

Gain (K): 0.142098
Zero: 7.75314
Pole: 1.10171

Lead Compensator

Gain (K): 7.03739
Zero: 29.2262
Pole: 205.676

Figure 9 System parameters obtained for the LAG-LEAD compensator.

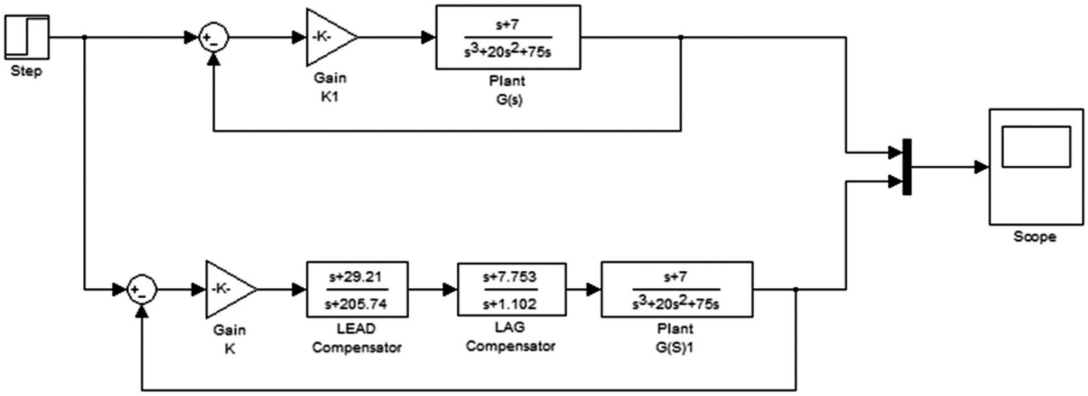


Figure 10 Simulink block diagram of the compensated and uncompensated control system.

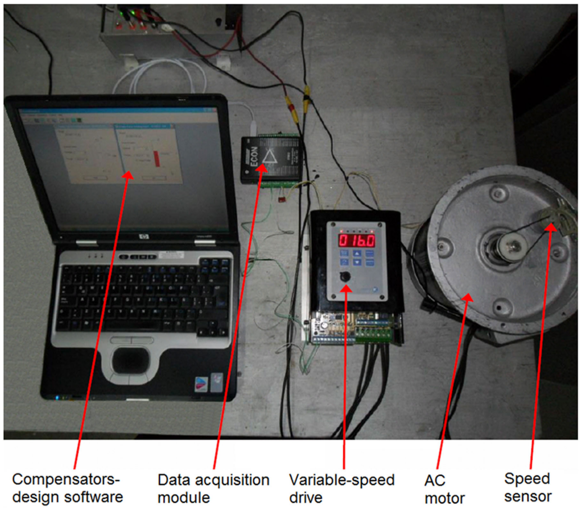


Figure 11 AC Motor-speed control; experimental setup. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

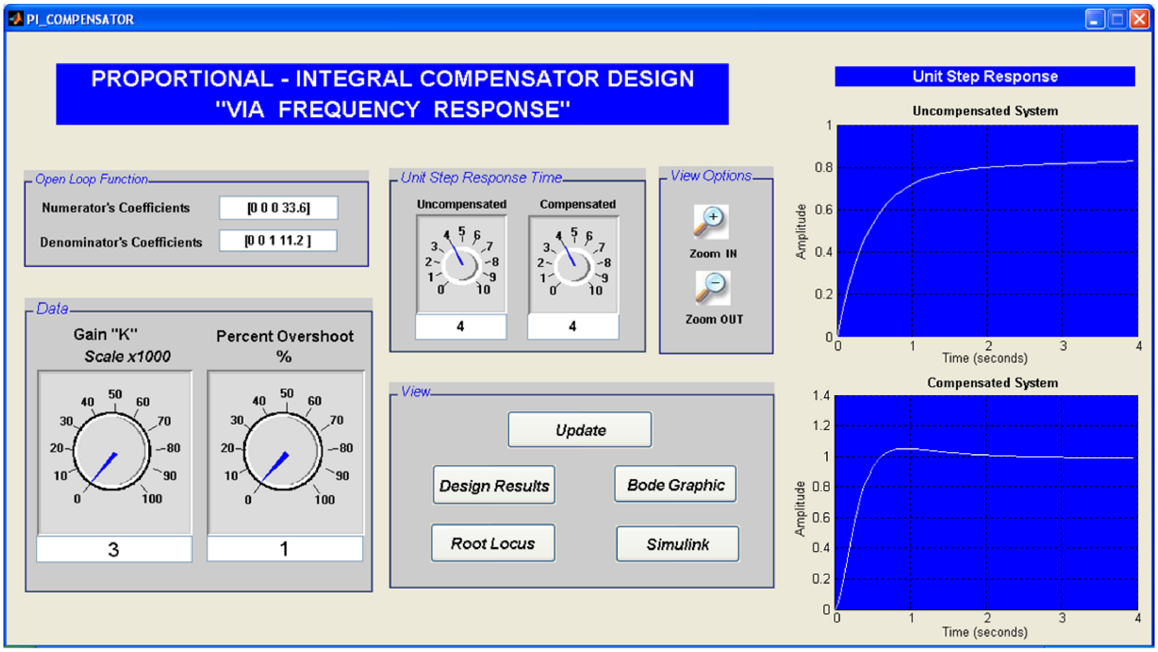
the unit step response of the system before and after compensation

$$H(s) = \frac{K_m}{1 + s\tau_m} \tag{12}$$

Figure 13 shows a typical obtained window with the transient and steady state response corresponding to the AC motor-speed control experiment.

ACADEMIC REMARKS

This section describes comments obtained from course evaluations, as well as written observations included in the lab reports. The tool described in this work has been used in graduate and undergraduate courses on control theory at University of the Americas and the National Institute of Astrophysics Optics and Electronics, Puebla, Mexico. According to the obtained evaluations, the students found the described system a very good tool, which help them to explore the effects of compensators in a


PI_COMPENSATOR

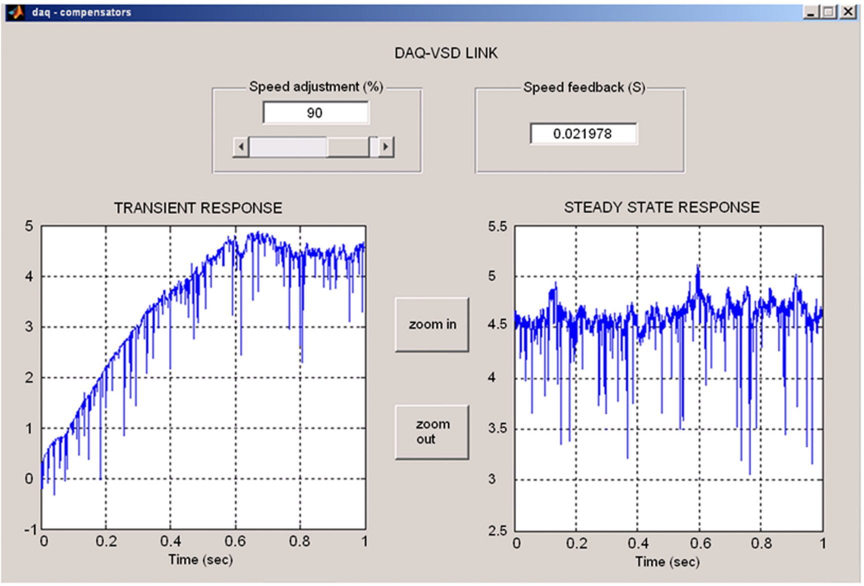
PROPORTIONAL - INTEGRAL COMPENSATOR DESIGN
"VIA FREQUENCY RESPONSE"

Open Loop Function
Numerator's Coefficients: [0 0 0 33.6]
Denominator's Coefficients: [0 0 1 11.2]

Unit Step Response Time
Uncompensated: 4
Compensated: 4

Data
Gain "K" Scale x1000: 3
Percent Overshoot %: 1

Figure 12 Unit step response before and after compensation. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]



Speed adjustment (%): 90

Speed feedback (S): 0.021978

Figure 13 Transient and steady-state response of the AC motor-speed control. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

very interactive way, while they were able to check the results in real time experiments. The graphical user interface was found to be a friendly way to explore the theoretical and experimental effect on the system response when some parameter changes. The general consensus was that they enjoyed the experiments which reinforced the theoretical concepts, with an improvement in their attention during the lectures.

CONCLUSIONS

We have presented a MATLAB-based graphical user interface for compensators design as an educational aid for basic control theory courses, which takes advantages of the available toolboxes on control and data acquisition. The educational graphical user interface helped our students to concentrate on the theoretical concepts, reducing the time spent in developing computational tasks, and reinforcing the students understanding of theoretical principles through simulations and interactive practice. An additional interesting result was the observation of a decrease in the number of errors which usually arise by having to program a great deal of code, avoiding also the need for debugging long programs. In short, it was found that the use of the software tool helped the students to reinforce some theoretical concepts, while improving their motivation to explore interactively the theoretical and practical effect of compensators in a classical control theory course. Additional experiments around the presented topic are being continuously developed by the students and the instructor.

REFERENCES

[1] B. Balamuralithara and P. C. Woods, Virtual laboratories in engineering education: The simulation lab and remote lab, Comput Appl Eng Educ 17 (2009), 108–118.

[2] J. L. Guzman, S. Dormido, and M. Berenguel, Interactivity in education: An experience in the automatic control field, Comput Appl Eng Educ, Published online in Wiley Online Library, DOI: 10.1002/cae.20480

[3] C. Domnisoru, Using MATHCAD in teaching power engineering, IEEE Trans Educ 48 (2005), 157–161.

[4] A. M. Dabrowski, S. A. Mitkowski, and A. Porebska, The use of mathematical programs and numerical methods in teaching selected topics in circuit theory based on Maple and MATLAB, Global J Eng Educ 13 (2011), 132–139.

[5] M. Stefanovic, V. Cvijetkovic, M. Matijevic, and V. Simic, A LabVIEW-based remote laboratory experiments for control engineering education, Comput Appl Eng Educ 19 (2011), 538–549.

[6] P. Kujan, M. Hromcik, and M. Sebek, Web-based MATHEMATICA platform for systems and controls education, IEEE International Symposium on Mediterranean Conference on Control and Automation, Limassol, Cyprus, June 2005, pp 376–381.

[7] J. Sanchez, S. Dormido, R. Pastor, and F. Morilla, A Java/Matlab-based environment for remote control system laboratories: illustrated with an inverted pendulum, IEEE Trans Educ 47 (2004), 321–329.

[8] B. Chen, Y. Cheng-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, Published online in Wiley Online Library, DOI: 10.1002/cae.20454

[9] J. A. Mendez, C. Lorenzo, L. Acosta, S. Torres, and E. Gonzalez, A web-based tool for control engineering teaching, Comput Appl Eng Educ 14 (2006), 178–187.

[10] A. Gilat, MATLAB: An introduction with applications, 3rd ed., John Wiley & Sons, Inc, Hoboken, NJ, 2008.

[11] P. R. A. Pereira, Implementation of IDP technique by the use of MATLAB as a tool for the chemical engineering education, International Conference on Engineering Education, Coimbra, Portugal, September 2007.

[12] M. Boulmalf, Y. Semmar, A. Lakas, and K. Shuaib, Teaching digital and analog modulation to undergraduate information technology students using Matlab and Simulink, IEEE EDUCON Education Engineering, Madrid, Spain, 2010, pp 685–691.

[13] I. Ngamroo, B. Somritvanitcha, and K. Hongesombut, Incorporating ObjectStab library and fuzzy logic toolbox for design of power system damping controller, Comput Appl Eng Educ 16 (2008), 243–255.

[14] M. Magistris, A MATLAB-based virtual laboratory for teaching introductory quasi-stationary electromagnetics, IEEE Trans Educ 48 (2005), 81–88.

[15] R. M. Del Toro-Matamoros, R. E. Haber, and A. Bustillo, Modelling and simulation of the high-speed milling process based on MATLAB-SIMULINK, Rev Metal 44 (2008), 176–188.

[16] U. Rajashekar, G. C. Panayi, F. P. Baumgartner, and A. C. Bovik, The SIVA demonstration gallery for signal, image, and video processing education, IEEE Trans Educ 45 (2002), 1–14.

[17] E. D. Ubeyli and I. Guler, MATLAB toolboxes: Teaching feature extraction from time-varying biomedical signals, Comput Appl Eng Educ 14 (2006), 321–332.

[18] W. S. Gan and S. M. Kuo, Transition from Simulink to MATLAB in Real-Time Digital Signal Processing Education, Int J Eng Educ 21 (2005), 587–595.

[19] A. Ugur and A. C. Kinaci, A web-based tool for teaching neural network concepts, Comput Appl Eng Educ 18 (2010), 449–457.

[20] E. D. Ubeyli, Teaching application of MATLAB fuzzy logic toolbox to modeling coplanar waveguides, Comput Appl Eng Educ 16 (2008), 223–232.

[21] W. C. Messner, M. D. Bedillion, L. Xia, and D. C. Karns, Lead and lag compensators with complex poles and zeros, IEEE Control Syst Mag 27 (2007), 44–54.

[22] D. A. Marquez and O. Cardenas, Implementation of a virtual laboratory for teaching PID controller, Inf Tecnol 19 (2008), 75–78.

[23] R. Bucher and S. Balemi, Rapid controller prototyping with Matlab/Simulink and Linux, Control Eng Pract 14 (2006), 185–192.

[24] W. E. Dixon, D. M. Dawson, B. T. Costic, and M. S. de Queiroz, A MATLAB-based control systems laboratory experience for undergraduate students: Toward standardization and shared resources, IEEE Trans Educ 45 (2002), 218–226.

[25] R. D. Keyser and C. Ionescu, Matlab-modelling, programming and simulations. In: A Matlab® interactive tool for computer aided control systems design in frequency domain: FRTool, Chap 5, E. P. Leite, (Ed.), Sciyo, Rijeka, Croatia, 2010, pp 87–98.

[26] Z. Kamis, E. E. Topcu, and I. Yuksel, Computer-aided automatic control education with a real-time development system, Comput Appl Eng Educ 13 (2005), 181–191.

[27] I. H. Altas and H. Aydar, A real-time computer-controlled simulator: For control systems, Comput Appl Eng Educ 16 (2008), 115–126.

[28] S. Norman, Nise, control systems engineering, 6th ed., John Wiley & Sons, Hoboken, NJ, 2011.

[29] A. Hughes, Electric motors and drives; fundamentals, types and applications, 3rd ed., Elsevier, Burlington, MA, 2006.

BIOGRAPHIES

Juan Manuel Ramirez-Cortes was born in Puebla, Mexico. He received the BSc degree from the National Polytechnic Institute, Mexico, the MSc degree from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Mexico, and the PhD degree from Texas Tech University, all in electrical engineering. He is currently a Titular Researcher at the Electronics Department, INAOE, in Mexico. His research interests include signal and image processing, biometry, neural networks, fuzzy logic control, and digital systems. He is a senior member of IEEE.

Vicente Alarcon-Aquino received the BSc degree from the *Instituto Tecnologico de Veracruz*, the MSc degree from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Mexico, and the PhD and DIC degrees from Imperial College London, University of London, London, UK, all in electrical engineering. He is a Titular Professor in the Department of Electrical and Electronic Engineering at the *Universidad de las Americas,* Puebla, Mexico. His research interests include network protocols, time-series prediction, wavelet-based digital signal processing, hardware description languages, and multiresolution neural networks.

Pilar Gomez-Gil was born in Puebla, Mexico. She received the BSc degree from the *Universidad de las Americas A.C*, Mexico, the MSc and PhD degrees from Texas Tech University, USA, all in computer science. She is currently an Associate Researcher in computer science at INAOE, Mexico. Her research interests include neural networks, image processing, pattern recognition, and software engineering. She is a senior member of IEEE, and a member of ACM.

Alejandro Diaz-Mendez received the BSc degree from Universidad Veracruzana, the MSc and PhD degrees from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Mexico, all in electrical engineering. He is currently a Titular Researcher at the Electronics Department, INAOE, in Mexico. His research interests include circuits and systems design, knowledge-based intelligent systems, and integrated circuits design. He is a senior member of IEEE.

Mariana Ibarra-Bonilla received the BSc degree from *Instituto Tecnologico de Veracruz*, and the MSc from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Mexico, both in electrical engineering. She is currently pursuing a doctoral degree in electronics, at INAOE, Mexico. Her research areas are control systems, sensors, and simultaneous localization and mapping systems.

Irma García-Enriquez received the BSc degree from *Instituto Tecnologico de Orizaba*, and the MSc from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Mexico, both in electrical engineering. She currently holds a faculty position at the *Instituto Tecnologico Superior de Atlixco*, Puebla, Mexico. Her teaching and research areas are control theory, mechatronics, and systems engineering.