An Interactive Tool for Servo Systems Learning

Nourdine Aliane, Rafael Pastor Vargas and Javier Fernández Andrés

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/46459

1. Introduction

Servo systems play an important role in industry. They are found in most automated manufacturing systems, machine tools, and robotic systems to cite but a few examples. Servo systems fundamentals have become an integral part of industrial electronics and other related fields in engineering, such as electrical engineering or computer disciplines. Servo systems topics should be extended to all engineering curricula [1]. It is therefore a major challenge to ensure that future engineers should be familiar with servo systems, and be able to analyze and control them.

A direct current controlled motor (DC motor) is considered to be the simplest form of servo system, and is used as a starting point for understanding all other electric machines. Control of DC motors is widely taught in control engineering and robotics courses, and is commonly used in laboratory experiments as providing an excellent case study. The importance of the material is well-evidenced in many textbooks [2]-[5]. However, in most of these texts, servo system problems are highly simplified for pedagogical purposes, and the given examples focus on the linear parts, and do not take into account practical issues such as compliance coupling, trajectory generation, the wind-up effect, feed-forward compensation, or torque limitation.

This paper presents an interactive learning module focused exclusively on servo systems. It is aimed at bridging the gap between theoretical background and experimentation, providing insight into fundamental concepts. The tool is based on exploiting interactivity as a pedagogical basis in teaching and learning activities. Although many interactive tools have been developed for general topics in control education [6]-[9], interactive tools focusing on servo systems are practically nonexistent.

There are several works in the literature which deal with the use of Matlab-Simulink for servo systems simulations for educational purposes [10]-[13], but the tools presented in these works lack interactivity. The Matlab Central File Exchange [14] is also used as a source to search for servo systems educational tools. Several excellent demonstrations can be found, but none of them meet the criteria for simulating practical aspects of servo systems. It is also worth mentioning the use of remote laboratories, where DC motors are successfully integrated in remote experimentation platforms [15]-[16]. However, remote experimentation is currently limited to performing simple experiments, and topics such as the effects of disturbances, anti-windup, or feed-forward, are not considered.

The development of the module presented in this paper is based on Simulink models used in combination with the Matlab graphical user interface (GUI). The choice of the Matlab platform is due to the fact that it is by far more productive than many other high level programming languages. Simulink, on the other hand, is used to easily model and simulate a variety of systems using intuitive block diagrams. Wrapping Simulink models within Matlab-GUI is advantageous and improves interactivity. This feature allows users to change parameters and view simulations without having to deal directly with Simulink blocks. Indeed, Matlab includes a built-in tool called GUIDE, which permits developers to design GUIs for Matlab applications.

The remainder of the paper is organized as follows: Section 2 presents the objectives and scope of the tool. Section 3 gives some theoretical background on servo systems. Section 4 describes the interactive module and shows how Simulink models can be integrated within an interactive Matalb-GUI. Section 5 describes usage experiences and gives some classroom examples. Section 6 describes the methodology used in evaluation and discusses the results obtained. Finally, Section 7 concludes the paper.

2. Objectives and scope

In the author’s institution, servo systems topics are an integral part of undergraduate control engineering and robotics courses. This part of the material examines servo systems principles, providing students with modeling techniques for DC motors, the formulation of the control problem, and an introduction to the basic treatment of feedback design. The theoretical background is supplemented with hands-on laboratory work. The laboratory activities are aimed at showing the students some qualitative aspects of servo systems control and giving them a broad picture of axis control. However, there is neither enough time nor sufficient equipment to experiment with all the practical aspects of servo systems.

To remedy this situation, the methodology was recently modified to place greater emphasis on fundamental concepts, because the tuning of servo systems can be confusing due rather to unfamiliar principles than to their complexity. Thus, to help students develop their knowledge of, and intuition for, servo systems and their control, an interactive module, dedicated exclusively to this topic, was developed. The tool is aimed at speeding up learning by reducing the time needed to design simulations. It is also intended to be interactive in the sense that it allows students to see immediately the effects of changing parameters on the behavior of systems. Finally, the tool is intended to complement rather than replace laboratory work.

The developed tool simulates the behavior of a separately-excited DC motor, and covers several practical issues of servo systems. Students can, among other capabilities, configure the model, perform simulations of position and speed control, interactively tune the controller, specify trajectories, and compare different control algorithms. Lastly, the user can perform comparisons of simulations by evaluating some performance measures.

3. Servo system background

3.1. Servo system modelling

Servo systems topics, as treated in many textbooks, consider simple models and assume that the motor shaft and its load are rigidly coupled. In practice, however, most servomechanisms contain flexible modes in the actuator or the structure being controlled. This is particularly true in systems such as disk drives, antennae and radar pointing or robotics axes, where coupling might have significant compliance. For a model to be useful, it must be realistic and yet simple enough to understand and manipulate. In general, flexible coupling is modeled as a rotational spring, which is sufficient to yield models of high order.

The interactive tool presented in this paper simulates a DC motor coupled to its load through a shaft with variable compliance (see Figure. 1). The user can set the model to what s/he considers relevant (a compliantly or rigidly coupled model).

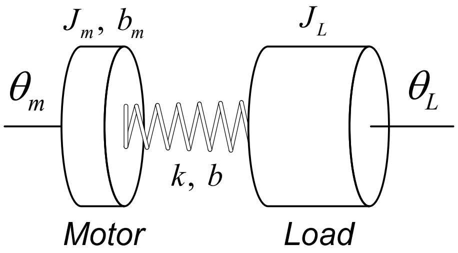


Figure 1. Model of motor and load compliantly coupled.

In the case of an armature-controlled DC motor, the control input is the armature voltage *U*. The torque *T* provided by the motor is proportional to the armature current *i*, and the back emf *e* is proportional to the motor's velocity *ω*, by the following equations

$$\begin{aligned} T &= k_m i \\ e &= k_m \omega \end{aligned} \tag{1}$$

The electrical equation is

$$U - e = L\frac{di}{dt} + Ri \tag{2}$$

where *R* is the armature resistance and *L* is the armature winding inductance.

The torque provided by the motor is transmitted to the load through the compliant shaft, which is modeled as a rotary spring with constant k and damping constant b. The mechanical equations of the motion are:

$$\begin{aligned} J_m\ddot{\theta}_m + b_m\dot{\theta}_m + b\left(\dot{\theta}_m - \dot{\theta}_L\right) + k\left(\theta_m - \theta_L\right) = k_m i \\ J_L\ddot{\theta}_L - b\left(\dot{\theta}_m - \dot{\theta}_L\right) - k\left(\theta_m - \theta_L\right) = 0 \end{aligned} \tag{3}$$

where subscript m refers to the motor, subscript L refers to the load, J is an inertia, and b is a damping constant.

Defining the sate vector as $x = \left[i, \theta_m, \dot{\theta}_m, \theta_L, \dot{\theta}_L\right]^T$, the state space representation of the DC motor with compliant coupling

$$\dot{x} = \begin{bmatrix} -R/L & 0 & -k'_m/L & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ k_m/J_m & -k/J_m & -(b+b_m)/J_m & k_m/J_m & b/J_m \\ 0 & 0 & 0 & 0 & 1 \\ 0 & k/J_L & b/J_L & -k/J_L & -b/J_L \end{bmatrix} x + \begin{bmatrix} 1/L \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \tag{4}$$

If the coupling shaft is rigid, then $k \to \infty$, and $\dot{\theta}_m = \dot{\theta}_L$. Adding the two mechanical subsystem equations (3) yields

$$J_m\ddot{\theta}_m + J_L\ddot{\theta}_L + b_m\dot{\theta}_m = k_m i \tag{5}$$

Thus, the mechanical subsystem becomes

$$\left(J_m + J_L\right)\ddot{\theta}_m + b_m\dot{\theta}_m = k_m i \tag{6}$$

Defining the total moment of inertia as $J = J_m + J_L$, and the state as $x = \left[i, \theta_m, \dot{\theta}_m\right]^T$, the state space equation is now simplified as

$$\dot{x} = \begin{bmatrix} -R/L & 0 & -k'_m/L \\ 0 & 0 & 1 \\ k_m/J & 0 & -b_m/J \end{bmatrix} x + \begin{bmatrix} 1/L \\ 0 \\ 0 \end{bmatrix} u \tag{7}$$

3.2. Servo system control

Despite the development of more advanced control techniques, the PID controller is still the most common algorithm used in servo systems applications [17]. The attractiveness of PID resides in the fact that an accurate model is not required and its control capabilities have been proven to be adequate for controlling servo systems. Indeed, PID controllers are used in many dedicated motion-control, such as LM628/629 [18], Magellan [19], Galil boards [20], or the Quanser DC motor model for training and education [21].

The mathematical expression in the time domain is given by the equation (8).

$$u(t) = K_p\left(r(t) - y(t)\right) + K_i \int \left(r(t) - y(t)\right)dt + K_d \frac{d\left(r(t) - y(t)\right)}{dt} \qquad (8)$$

where u(t) is the controller output, r(t) is the reference, y(t) is the measured system output, and *Kp, Ki* and *Kp* are the PID adjustable parameters. In real servo systems, PID controllers can perform poorly when used alone. To enhance the system performance, a number of additional mechanisms such as handling the wind-up effect properly, combining the feedback with feed-forward control, or using a trajectory generator, are adopted as technical solutions.

3.2.1. Anti-windup correction

The elimination of the steady state error in servo systems has long been performed using the integrator action. However, this action has the disadvantage of causing the wind-up effect, which occurs when the calculated control signal exceeds its saturation limits and the controller is unable to respond immediately to changes in the error signal. To prevent the windup effect, the operating range of the control signal should be limited to the range of the voltage input of the servo [22]. This ad-hoc solution provides instant recovery when the error signal changes signs. There are many way to ovoid the integrator wind-up and one of these possibilities is illustrated in the following Simulink diagram.

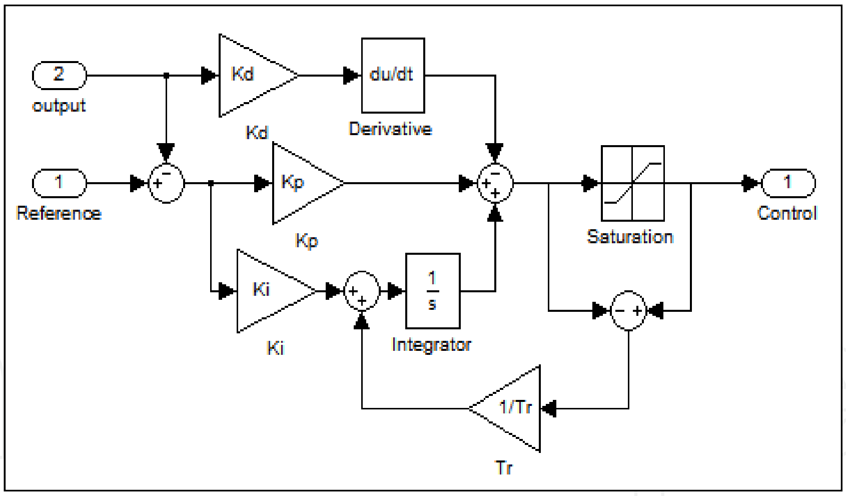


Figure 2. PID controller with Anti-Windup correction.

3.2.2. Feed-forward control

Feed-forward is another technique used to improve servo systems performance and is essentially used to reduce the tracking error in high performance motion control problems. Theoretical developments show that the feed-forward transfer function is the inverse dynamic of the servo. In general, feed-forward compensation is performed through the required acceleration and velocity.

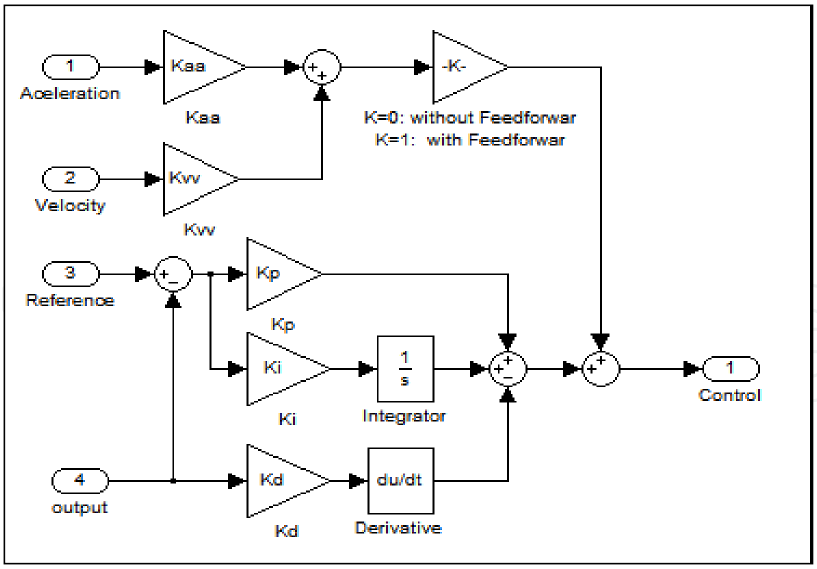


Figure 3. PID controller with feed-forward control.

3.2.3. *Motion control*

In real servo systems control, the position step references can cause controller saturation and lead to significant overshoot, and indeed, this kind of references are hardly used. The step response is actually used as a measure of system performance. To overcome these problems, references known as S-curves, such as parabolic or trapezoidal profiles, are used instead. These S-curves are provided by a trajectory generator, which is an algorithm at the top of control hierarchy.

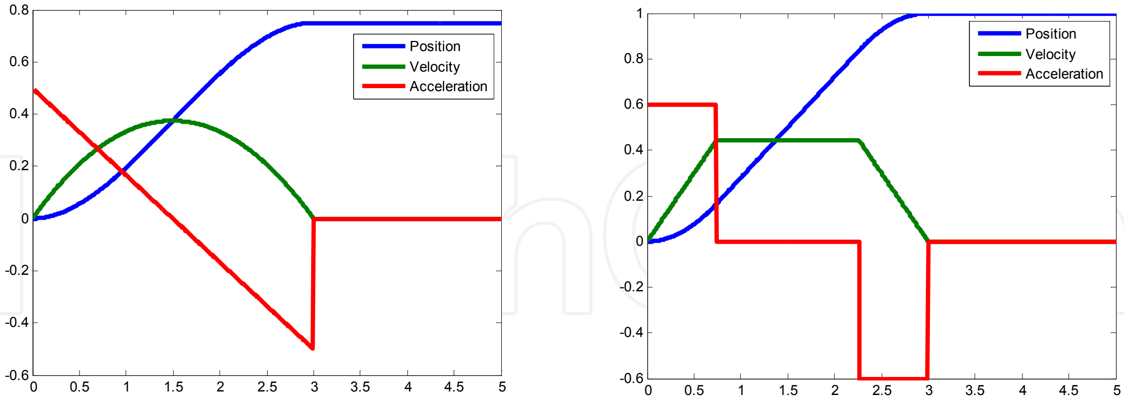
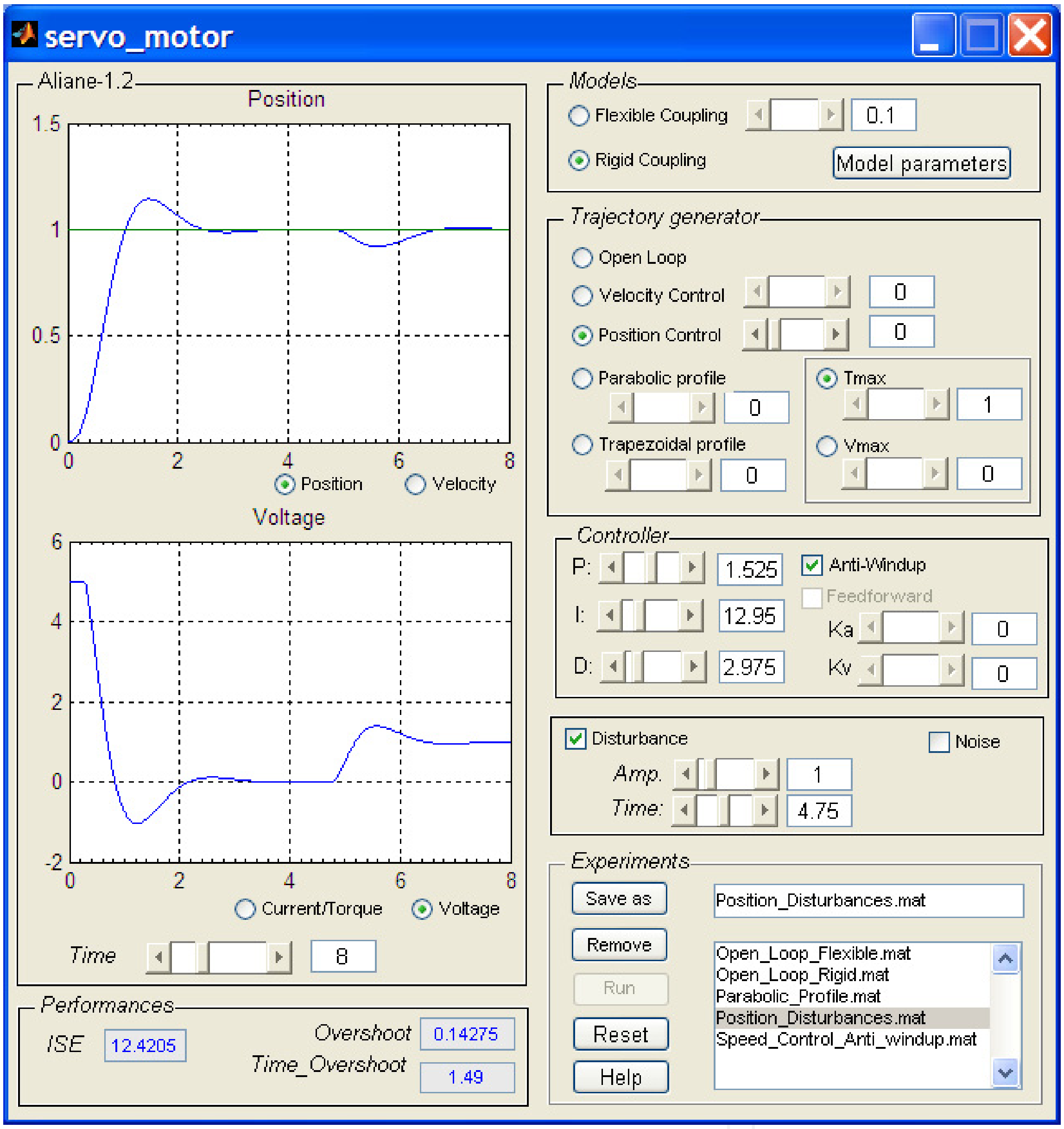


Figure 4. Parabolic and trapezoidal profiles for motion control.

4. Interactive tool description

As mentioned above, the interactive tool is composed of two parts: a Matlab-GUI application and a Simulink model. The Simulink model is completely transparent for end users, and is automatically opened and runs in the background. The tool is freely available on the World Wide Web [23]. The layout of the GUI of the interactive module is shown in Figure 5.



servo_motor

Aliane-1.2

○ Position ○ Velocity

○ Current/Torque ● Voltage

Time 8

Performances

ISE 12.4205

Overshoot 0.14275

Time_Overshoot 1.49

Models

○ Flexible Coupling 0.1

● Rigid Coupling

Trajectory generator

○ Open Loop

○ Velocity Control 0

● Position Control 0

○ Parabolic profile 0

○ Trapezoidal profile 0

● Tmax 1

○ Vmax 0

Controller

P: 1.525

I: 12.95

D: 2.975

☑ Anti-Windup

□ Feedforward

Ka 0

Kv 0

☑ Disturbance

□ Noise

Amp. 1

Time: 4.75

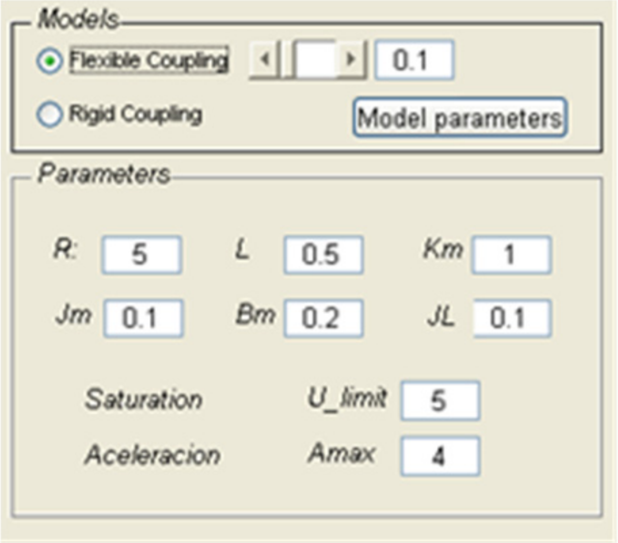
Experiments

Position_Disturbances.mat

Open_Loop_Flexible.mat
Open_Loop_Rigid.mat
Parabolic_Profile.mat
Position_Disturbances.mat
Speed_Control_Anti_windup.mat

Figure 5. The servo systems learning module main user interface.

Its interface visually displays all the actions that users may execute. In the first group of controls, users can select between a rigid or flexible coupling model. The difference between these models can be visualized by their open-loop step response. Furthermore, the compliance of the flexible coupling can be changed interactively which allows students visualize the significance of coupling stiffness better. Users can also customize the model by setting the model parameters in the "model parameters" button, which displays a window for capturing model parameters.



Models

● Flexible Coupling 0.1

○ Rigid Coupling

Parameters

R: 5 L 0.5 Km 1

Jm 0.1 Bm 0.2 JL 0.1

Saturation U_limit 5

Aceleracion Amax 4

R:	Electric resistance
L:	Electric inductance
Km:	Electromotive force constant.
Jm:	Moment of inertia of the rotor
Bm:	Damping ratio factor
JL:	Moment of inertia of the load
U_limit:	Maximal input source voltage
Amax:	Maximal motor acceleration

Figure 6. DC-Motor model parameters.

This is suitable for motor modeling, and it is particularly useful for investigating the effects of some DC-motor's parameters on the speed and how the induced current is modified.

In the second group, users can select between simulating a velocity or position control. In the position besides the step input reference, users can choose parabolic or trapezoidal profiles as a trajectory reference. The third group deals with control, where users can interactively modify the PID tuning parameters and immediately see their effect on the system behavior. In addition, the feedback control can be augmented with a feed-forward and an anti-windup correction. The fourth group allows the simulation of the effect of exogenous factors such as external disturbances and noise measurements. The fifth group lets users save and retrieve simulations, which is particularly useful for drawing straightforward comparisons between different scenarios. Finally, as well as visualizing velocity, position, induced current, and voltage input, some performance measures, such as the integral of squared error (ISE) and the overshoot, are also displayed.

The design of the Matlab-GUI is easy using the Matlab GUIDE. This toolkit is opened by entering the command ">>guide" in the command line. Then, images of all the elements (sliders, axes plotting, etc) are dragged out, and then the user writes their respective callback functions. The reader can go further with the help of the Matlab GUI manuals.

The Simulink block diagram implementing the whole servo systems simulation is shown in Figure 7. Interfacing Simulink models with Matlab GUI applications is easy, but not trivial operation. The control of Simulink models can be performed through suitable Matlab commands. Simulink blocks can be accessed and their parameters changed by using the "set_param()" function. Alternatively, the communication between Simulink models and Matlab-GUI can be performed by using variables defined in the main workspace. These two mechanisms are also valid for specifying other Simulink parameters such as sample time or time simulation through the GUI. Lastly, a Simulink simulation can be run by using the "sim()" function.

As far as the trajectory generator is concerned, this is implemented as an "embedded Matlab function block," which is a Simulink block that contains a compiled Matlab code. Unlike "Matlab function blocks," which accept multiple inputs and support only a single scalar output, the functions in "embedded Matlab function blocks" allow multiple inputs and return multiple outputs.

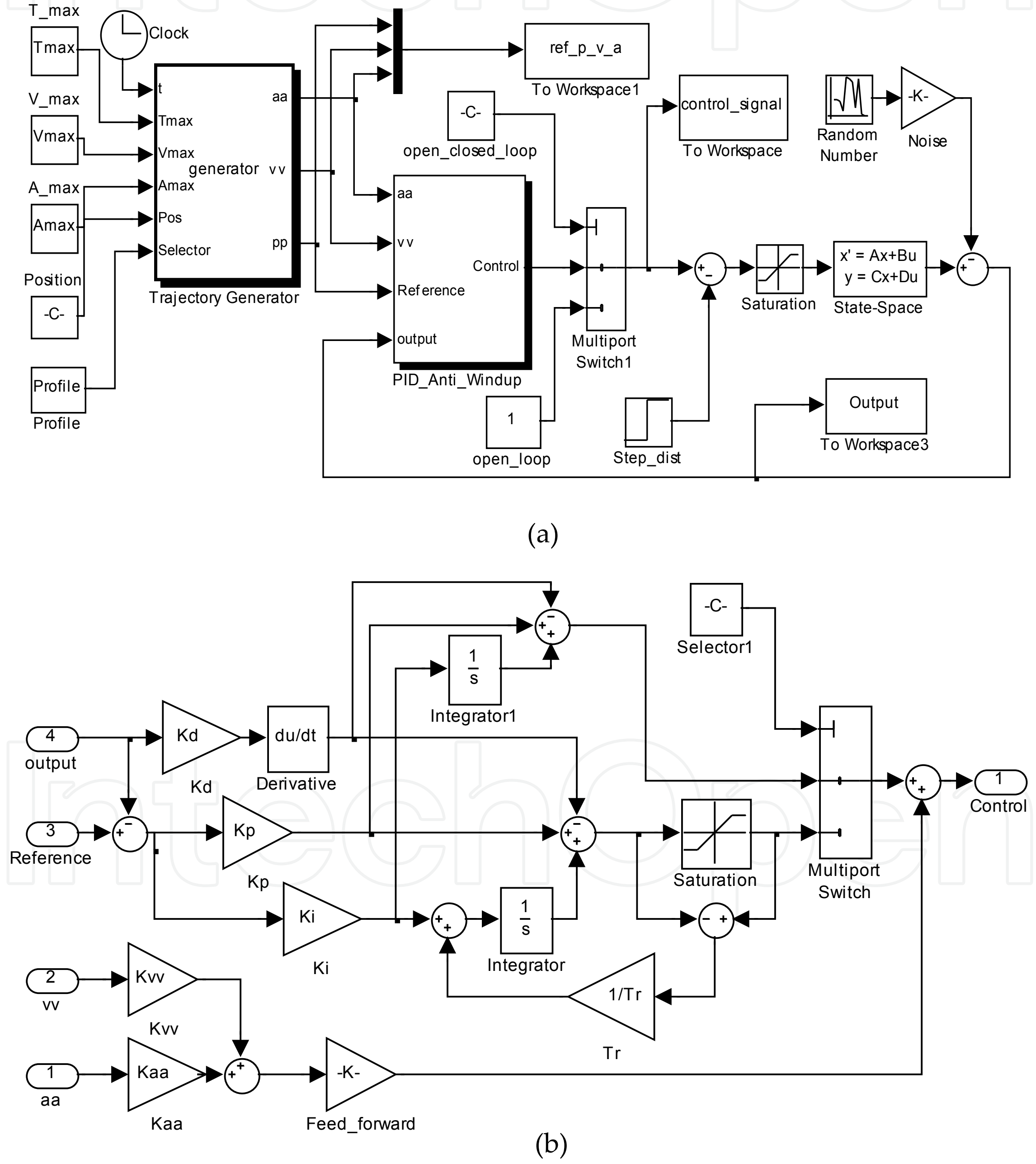


Figure 7. a) Simulink block diagrams: General structure for servo systems simulation; b) The controller subsystem: the PID, anti-windup, and feed-forward control.

5. Classroom experience

The tool was used during the 2006-07 and 2007-08 academic years at Universidad Europea de Madrid (Spain), in control engineering and robotics subjects, providing the instructor with a valuable supplement to lectures by projecting the module screenshots. The first advantage of the tool presented here is that it allows the instructor to set up simulations with minimum effort, which obviously helps him to maintain the flow of the lecture, and allows him to pose different cases quickly in response to the students' questions. The module proved to be an effective instrument to facilitate teaching, since the interactive nature of the module helps to draw comparisons between simulations, explore the effect of varying parameters, and give explanations through "what-if" scenarios.

The learning module was also used by approximately 35 students in the classroom, following guided exercises. These exercises were focused on students getting a feel for the qualitative aspects of servo systems, having them explore what would happen if a given parameter were increased or decreased. The quantitative aspects were not emphasized as much. For example, students are asked to undertake the problem of position control and tracking trajectories. After a review of the PID controller, students are asked to tune the controller through a trial and error process, taking advantage of the interactivity of the tool. As a first step, students use proportional action (P) and examine steady-state error. Next, they introduce the integral action (I) to see how the steady-state error is eliminated. Finally, the derivative action (D) is activated to increase the dumping of the system. Robustness is also an important aspect in servo systems. In another exercise, simulations are performed to assess control strategy with respect to external disturbances, considering cases where a motor is holding a final position and a disturbance is applied to its shaft. Thanks to the module's interactivity, many other concepts such as flexible coupling, anti-windup and feed-forward corrections, or the use of S-curve references are presented with animated graphs examining the effects of some parameters. These exercises are solved in group, and the use of the interactive tool provides an excellent context for provoking discussions and reflections.

To foster interest in the use of the tool, a comprehensive tutorial, providing a number of handouts on classroom activities, was also developed [23]. This tutorial starts with an introductory level on modeling techniques, and goes into more detail on using the PID controller, anti-windup, feed-forward and trajectory generation. It provides a total of ten guided examples; some are projected as demonstrations during lectures while others are developed as exercises and as a basis for group discussion on more in-depth theoretical topics.

5.1. Classroom examples

Many are the classroom activities that can be performed by the tool presented. This subsection shows three classroom examples that illustrate the module's capabilities. Although these examples do not fully exploit all of the tool's features, they illustrate some important aspects of servo systems.

Example-1: Modeling and open-loop simulation

Models are an essential element of servo system analysis and design, and serve to understand how systems behave with and without control. This aspect is covered through demonstrations shown during a lecture, carrying out comparisons of a motor's open-loop responses with rigid and flexible coupling, as well as observing dynamically how changes in the compliance parameter affects the shapes of the responses. Furthermore, different parameters of the model are taken into account and are changed dynamically in order to assess their effect in light of the modeling concept. Step responses of a DC motor with rigid and flexible coupling are shown in Figure. 8.

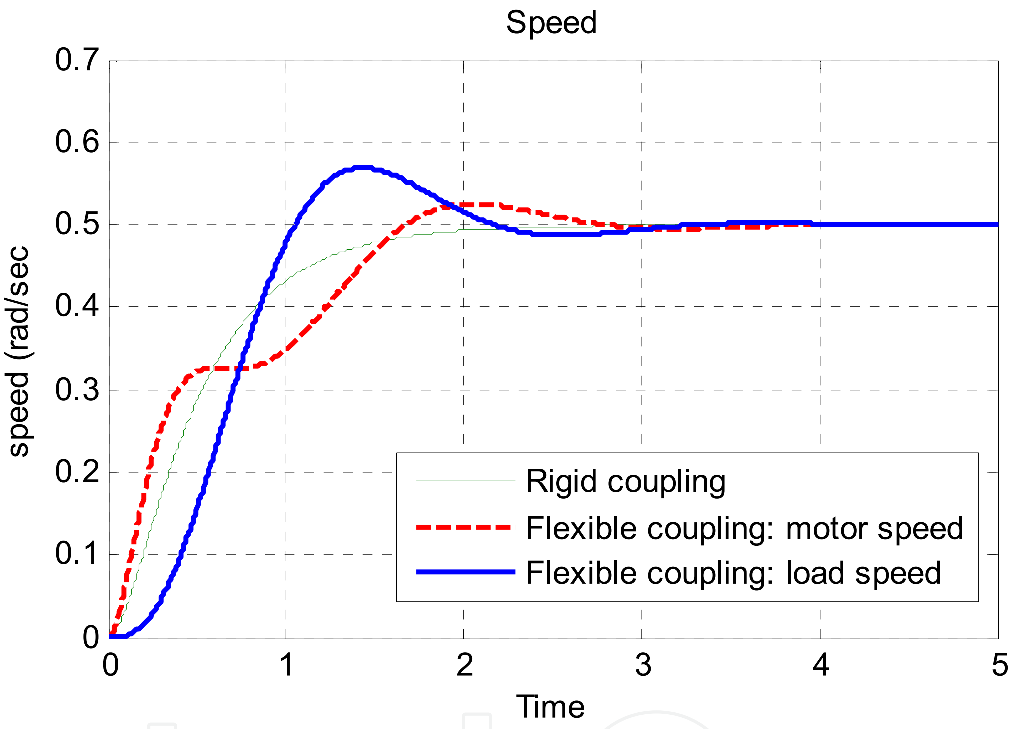


Figure 8. Step responses of a DC motor with rigid and flexible coupling (k = 0.7).

Example-2: Speed control and anti-windup correction

The tool is used to study velocity control and its characteristics. For example, one exercise is devoted to simulating PI control, applying poles placement strategy. The tuning parameters are adjusted analytically assuming a simplified model. Afterwards, students are required to compare simulation outputs applying the same controller tuning to the model with flexible coupling. In the context of speed control, another exercise is dedicated to demonstrating the benefits of anti-windup correction. In this section, students are required to compare the system's output under different conditions: namely, a simulation with no saturation limits, a simulation with saturation limits, and a simulation with anti-windup compensation. This comparison is illustrated in Figure. 9.

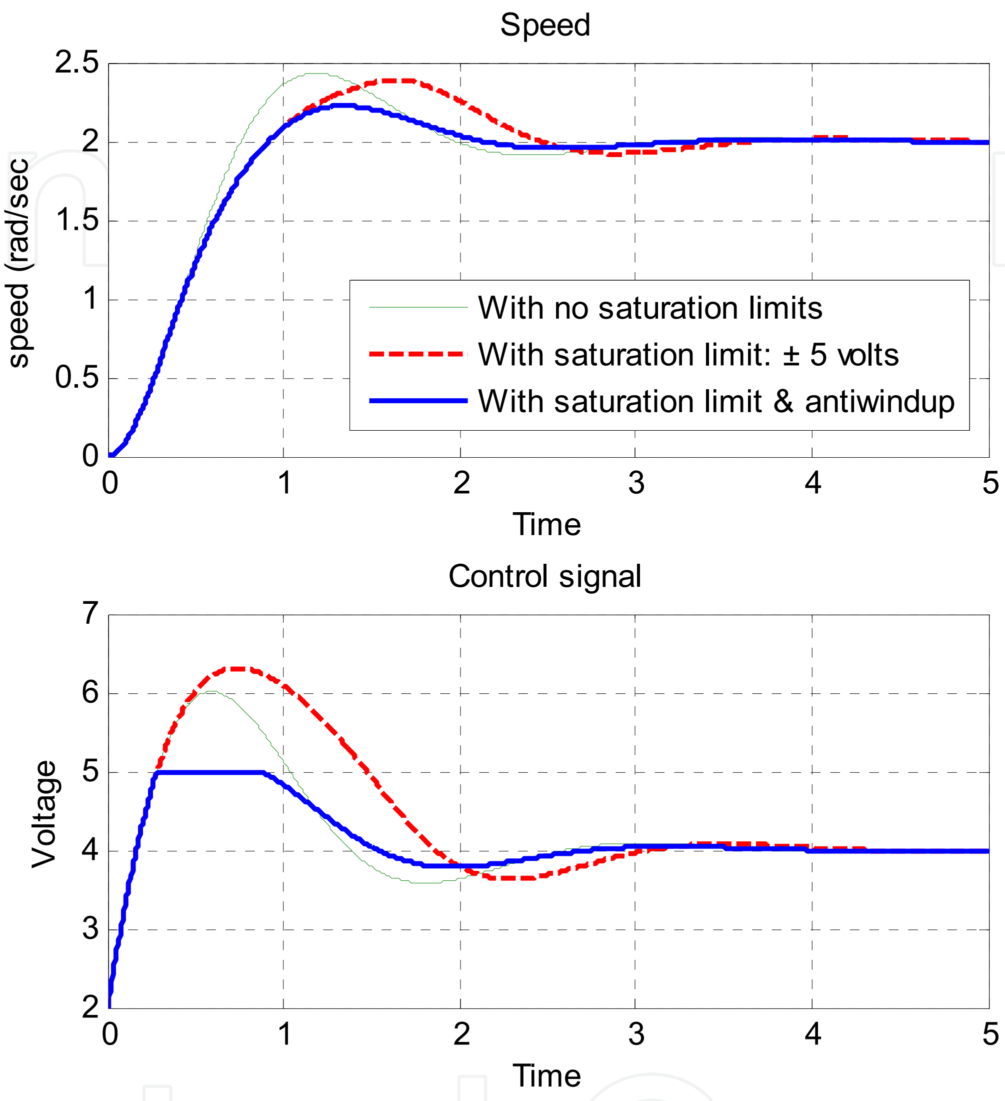


Figure 9. Speed control using a PI controller with and without anti-windup effect (P=1 I=7).

Example-3: Parabolic velocity profiling and feed-forward

Servo systems behavior is often characterized by their step responses. However, it is also useful to see how servo systems behave under S-curves, such as parabolic or trapezoidal profiles. Students are asked to investigate the effect of these trajectories and how input voltage and current are influenced. Lastly, activities concerning feed-forward compensation are also possible, demonstrating how near-zero tracking error can be achieved. Figure. 10 shows a comparison of position control through a parabolic profile, and illustrates the effect of feed-forward.

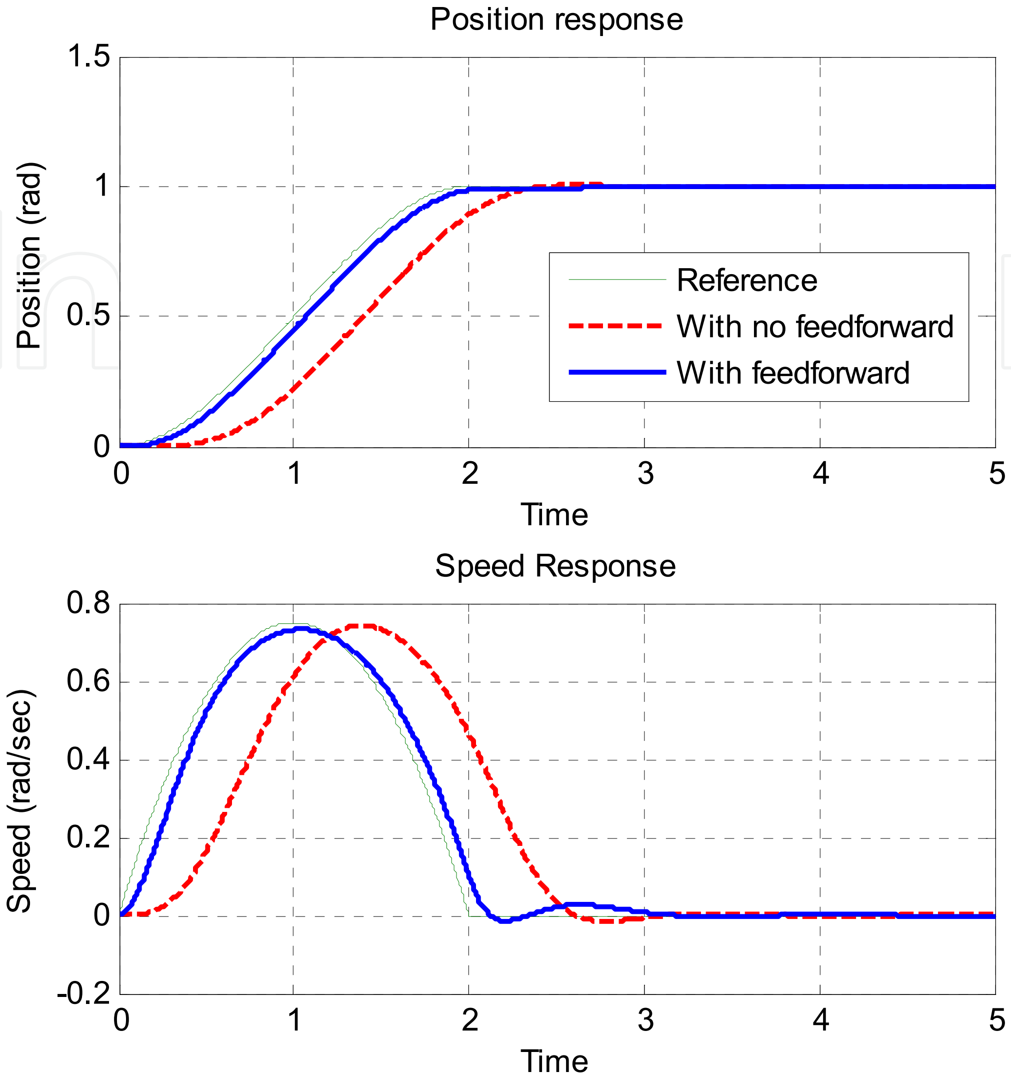


Figure 10. Illustration of a parabolic velocity move profile with and without feed-forward. (P = 12.6, I = 0.5, D = 4.25, Kv = 4.95, Ka = 0.85, T = 2).

6. Assessment and evaluation

The use of the interactive module, from the students' point of view, and how this contributes effectively to enhancing the learning of servo systems are important issues for the instructor. To this end, a qualitative assessment based on a focus group interview [24] and a traditional course survey were conducted by the instructor. Interviewing students constitutes an excellent way of assessing the use of the new learning module as well as learning outcomes. The interview consisted of discussing a set of open-ended questions with the students, raising two primary issues: the usability of the module and its effectiveness in learning. After the discussion, students were asked to give an evaluation (from 1 to 5) for each of the discussed questions. The assessment was conducted with two groups (a control engineering and robotics courses) during the second 2007 semester. In all, 34 students took part in the interview. The students' questionnaire and corresponding evaluations are summarized in Table 1.

Nº.	Questions	score
1	The module is easy to use.	3.91 (0.79)
2	The module is productive and allows setting up simulations quickly.	3.56 (0.93)
3	The module is useful for complementing lectures.	3.88 (1.01)
4	The module gave me more motivation to learn.	3.79 (0.77)
5	Overall, the module helped me to learn servo systems.	3.88 (0.88)

Table 1. Student questionnaire. The table gives the mean and the standard deviation of students' responses. The grading scale is from 1 (low) to 5 (high).

Concerning the usability of the module, students were receptive to using the new tool. The majority agreed that it is easy to use and does not add cognitive load. They also agreed that the use of a simulation module with a GUI is much better than directly using Simulink blocks, since it helps them to proceed quickly. Students agreed that its most valuable feature is the sliders provided as a means to explore the effects of varying parameters on system behavior. The companion tutorial, with its guided exercises, is also considered important for the correct use of the module. In this aspect, students presumably limit their instructional material to the tutorials designed by the teacher.

The results from the question on whether the module is useful for complementing lectures showed that students consider that interactive tools represent a convenient way of enhancing in-class sessions and bringing new dimensions to traditional teaching methods. The majority considered that the tool presented constitutes a good complement for lectures, and many of them advocated extending the use of such modules to more topics and subjects.

Concerning learning outcomes, students uniformly agreed that the module allowed them to pay more attention to simulation results, rather than to designing Simulink diagrams. Some students declared that the interactive feature of the module activated their curiosity to discover its contents. As the interaction with the module is driven mainly by sliders, students can explore the effects of a full range of parameters by adjusting the values from low to high, or vice versa. In this aspect, the majority believed that this routine enables them to perceive clearly the qualitative aspects of servo systems. Finally, many students agreed that the tool helped them to understand servo systems problems, and many of them felt that the interactive module was an effective learning medium.

7. Conclusion

In this paper, the development of a Matlab/Simulink-GUI application for servo systems learning is described, and its use in the classroom is also addressed. The tool exposes students to servo systems concepts and allows them to experience a variety of practical scenarios. Its principal feature is that it dramatically reduces the effort needed to specify simulations. It also allows students to explore the effects of varying parameters and instantly observe their influence on the system's behavior. Interaction with the module is straightforward and is done essentially using sliders. This sort of interaction proved to be helpful since it stimulates the students, and it is found that the interactive tool provides a stronger motivation to learn than does the traditional use of Simulink blocks. Finally, extensive use of the module can give students an enhanced intuition and help to them to gain better understanding of the qualitative aspects of servo systems.

Author details

Nourdine Aliane[*] and Javier Fernández Andrés
Dpto: Sistemas Informáticos Automática y Comunicaciones, Universidad Europea de Madrid (UEM), Villaviciosa de Odón, Madrid, Spain

Rafael Pastor Vargas
Dpto. Sistemas de Comunicación y Control, Escuela Técnica Superior Ingeniería Informática, UNED, Spain

Acknowledgement

The authors wish to thank the Spanish Ministry of Science and Innovation and the National Plan R&D TIN2008-06083-C03/TSI "s-Labs: Integration of open services for remote and distributed virtual laboratories, reusable and safe".

8. References

[1] D. S. Bernstein, The Quanser DC motor Control Trainer (2005), *IEEE Cont. Syst. Mag.*, vol. 25, no. 3, pp. 90-94.

[2] G. F Franklin, J. D. Powell, and A. Emami-Naeini, (2006), *A. Feedback Control of Dynamic* Systems. 5th ed, New Jersey: Prentice-Hall.

[3] F. L. Lewis, (1992), *Applied Optimal Control & Estimation: Digital Design & Implementation.* 1st ed, New Jersey: Prentice Hall.

[4] G. F. Franklin, J. D. Powell and M. L. Workman, (1997) *Digital Control of Dynamic Systems.* 3rd ed, Boston: Addison Wesley.

[5] R. N. Clark, (1997) *Control System Dynamics.* Cambridge: Cambridge University.

[6] B. Wittenmark, H. Hägglund, and M. Johansson, (1998), Dynamic pictures and interactive learning, *IEEE Cont. Syst. Mag.*, vol. 18, no. 3, pp. 26-32.

[7] M. Johansson, M. Gäfvert, and K. J. Åström, (1998), Interactive tools for education in automatic control, *IEEE Cont. Syst. Mag.*, vol. 18, no. 3, pp. 33-40.

[8] S. Dormido, S. Dormido-Canto, R. Dormido-Canto, J. Sanchez, and N. Duro, (2005) The role of interactivity in control learning," *Int. J. Eng. Educ.*, vol. 21, no. 6, pp. 1122-1133.

[*] Corresponding Author

[9] J. L. Guzmán, K. J. Åström, S. Dormido, T. Hägglund, and Y. Piguet, (2008), Interactive learning modules for PID control, *IEEE Cont. Syst. Mag.*, vol. 28, no. 5, pp. 118-134.

[10] T. Kikuchi, T. Kenjo and S. Fukuda, (2002), "Developing an educational simulation program for the PM stepping motor," *IEEE Trans. Educ.*, vol 45, no 1, pp-70-78.

[11] S. Ayasun and C. O. Nwankpa, (2005), Induction motor tests using Matlab/Simulink and their integration into undergraduate electric machinery courses, *IEEE Trans. Educ.*, vol 48, no 1, pp. 37–46.

[12] N. Patrascoiu, (2005), Modeling and Simulation of the DC Motor Using Matlab and LabVIEW," *Int. J. Eng. Educ.*, vol 21, no 1, pp. 49-54.

[13] D. Hercog and K. Jezernik, (2005), Rapid Control Prototyping using Matlab/Simulink and DSP-based Motor Control," *Int. J. Eng. Educ.*, vol. 21 no. 4, pp.596-605.

[14] MATLAB-Central-File-Exchange [on-line], Retrieved March 2012 from:
http://www.mathworks.com/matlabcentral/fileexchange

[15] Casini, M. y D. Prattichizzo, (2003), The automatic control Telelab: A user-friendly interface for distance learning," *IEEE Trans. Educ.*, vol.46, no. 2, pp. 252-257.

[16] R. Pastor, C. Martín, J. Sánchez, and S. Dormido, (2005), Development of a XML-based lab for remote control experiments on a servo motor, *Int. J. Elect. Eng. Educ.*, vol. 42, nº 2, pp. 173-184.

[17] R. Kelly and J. Moreno, (2001), Learning PID structures in an introductory course of automatic control," *IEEE Trans. Educ.*, vol. 44, no. 4, pp. 373-376.

[18] LM628/LM629 Precision Motion Controller, [on-line] Retrieved March 2012 from:
http://cache.national.com/ds/LM/LM628.pdf

[19] MAGELLAN Motion Control ICs, [on-line] Retrieved March 2012 from:
http://www.pmdcorp.com/downloads/Magellan_PUG_v23_Feb_2008.pdf

[20] GALIL Controllers, [on-line] Retrieved March 2012 from:
http://www.galilmc.com/about/index.html

[21] J. Apkarian and K. J. Astrom, (2004), A laptop servo for control education, *IEEE Cont. Syst. Mag.*, vol. 24, no. 5, pp-70-73.

[22] K. J. Åström and T. Hägglund, (2005), Advanced PID Control, ISA-Society.

[23] The Matlab interactive module for servo systems learning: [on-line]:
http://www.esp.uem.es/aliane/servosystems/ss.zip

[24] B. M. Olds, B. M. Moskal and R. L. Miller, (2005), Assessment in Engineering Education: Evolution, Approaches, and Future Collaborations," *J. Eng. Educ.*, vol. 94, no. 1, pp. 13-25.