Implementing and Validating Analog and Digital Controllers

Horacio Vasquez[*], Javier Kypuros[a], Roy Villanueva
[*]Corresponding Author. Assistant Professor
Phone: 956-292-7419, Fax: 956-381-3527; e-mail: vasqu002@panam.edu
[a]Assistant Professor
Department of Mechanical Engineering, The University of Texas-Pan American
1201 West University Drive, Edinburg, TX 78539

Abstract

Teaching control systems concepts to mechanical engineering students can be greatly improved by integrating theory with hands-on activities, implementation, and validation of controllers. In mechanical engineering, it is a common practice to teach modeling and analysis of dynamic system and control systems in a theoretical manner, usually complemented with attractive examples and simulations using sophisticated software packages. In most occasions, the control systems perfectly accomplish the task they are designed for and they are ready in just a few iterations, or perhaps, after rigorous theoretical modeling and controller design. However, there seems to be a need for hands-on activities for students to go through the implementation and validation of the designed controllers. The reason why there is a disconnection between theory and experimentation might occur because experimental work with control systems requires additional knowledge about low- and high-power electronics, microcontrollers, interfaces, and digital control systems, which are usually elective or graduate courses; another reason might be due to not having prototypes of real engineering system and the computer equipment required to implement the controllers. It is the purpose of this paper to present the result of several experiments and a practical procedure to design, implement, and validate a relatively simple controller in analog and digital form to regulate the position of prototype rotational arm driven by a geared brush-type DC motor. It is also the goal of this study to show undergraduate mechanical engineering students some practical options available to implement basic control systems laws using analog devices, data acquisition systems, and digital signal processors (DSPs).

Keywords: PI controller, Control Systems, DSPs

Introduction and Background

This work presents experimental results of several viable options for undergraduate mechanical engineering students to implement and validate basic control system compensators using components readily available in electronics, measurement and instrumentation, mechatronics, and/or control systems labs. A proportional and integral (PI) controller was used to control the position of a rotational arm connected to a geared brush-type DC motor. The approach taken in this study was on purpose backwards of what is commonly done to design a controller. Having constructed the motor-arm setup, the authors proceeded to determine the proportional and integral gains that provided satisfactory two-revolution step responses of the arm, by changing resistors and capacitors in the implementation of the analog controller using operational amplifiers. It was determined that a proportional gain of 15 and an integral gain of 1 was a good choice. After that, the same PI controller was implemented and tested in both LabVIEW using a data acquisition system board and MATLAB/SIMULINK using a DSP evaluation module. After that, the controller was simulated using MATLAB/SIMULINK and the experimental step responses were compared to the ones obtained by simulation.

In summary, the PI controller was implemented in 3 different ways: with operational amplifiers; with a data acquisition system and LabVIEW; and with an ezDSP TMS320F2812 evaluation module using MATLAB/SIMULINK. The first two implementations were achieved using power amplifier to drive the DC motor, because the electronics used for the controllers are not able to handle the amount of current required by the motor. Two types of operational amplifiers were used for comparison purposes: a Logosol and an OPA547. The third implementation of the controller was achieved using pulse width modulated (PWM) and rotation direction signals, both of them sent out by the controlling algorithm on the DSP towards an H-bridge power converter to drive the motor.

The material presented in this study is an effort to develop hands-on activities for students to implement control system compensators, and not only the basic required components but also several alternatives to accomplish such task are presented. There have been multiple efforts elsewhere to create prototype systems to implement control systems, in particular PID controllers. For example, Murphee et al.[1] controlled the position of a block attached to a screw nut that travels along a ball screw by implementing a PID controller on a DSP to create a demonstrative setup of control system concepts for undergraduate mechanical engineering students. A LabVIEW program was developed to serially communicate with a TMS320F243 DSP to allow students to vary the set point position of the block and to adjust the gains of the PID controller[1]. As another example, a PID controller was implemented to regulate the temperature in a kiln used by mechanical engineering students in a Strength of Materials course[2]. This controller allowed adjusting the duty cycle of a pulse width modulated (PWM) signal acting on a solid state relay that regulates the current going to the heating element; feedback was obtained with a K-type thermocouple[2]. In another project, a multidisciplinary approach that combines curricula in electrical, mechanical, and chemical engineering has been developed for instruction of control systems at Rowan University[3]. Experimental setups to control the velocity of a DC motor, the velocity of an internal combustion engine, and the fluid level of a process, were developed for instruction of PID and control systems concepts[3]. Basilio and Matos[3] presented a study on how to design PID controllers with transient performance specifications similarly to the Ziegler-Nichols method but with improved overshoot response. Kelly and Moreno[5] showed two PID structures used to regulate the shaft's angular position of a DC motor. As they described, a PID controller created with a position PI compensator and an internal velocity feedback loop allowed simple and convenient application of Routh-Hurwitz criteria and adequate selection of the compensator gains that guarantee stability of the closed-closed system[5]. Li et al.[8] and Ang et al.[9] presented advanced tuning techniques for PID controllers and mentioned that there are difficulties found in industry that are usually not found in academia when designing PID controllers. They also mentioned that using filtering techniques and switching between controller structures and operating modes is usually required to obtained satisfactory transient and steady-state performance of a system. The implementation and performance of PI controllers to regulate the position of a motor-arm setup using operational amplifiers, LabVIEW and a data acquisition system, and MATLAB/Simulink with a DSP evaluation module are presented next.

Description of the Problem

As presented in Figure 1, a DC motor with a gear reduction attached at its end was connected to an arm mounted on a shaft and supported by two bearings to create a simple setup for mechanical engineering students to implement and validate analog and digital control systems. A 10 kΩ, 10-turn, potentiometer was used at the end of the arm as sensor to measure angular position. A voltage of 5V was applied between the fixed ends of the potentiometer to obtain a linear relationship of 0.5V per revolution of the potentiometer wiper. This sensor could easily be replaced with a tachometer or an optical encoder to develop other experimental practices with different control strategies. Figure 1 shows a diagram of the entire motor-arm system.

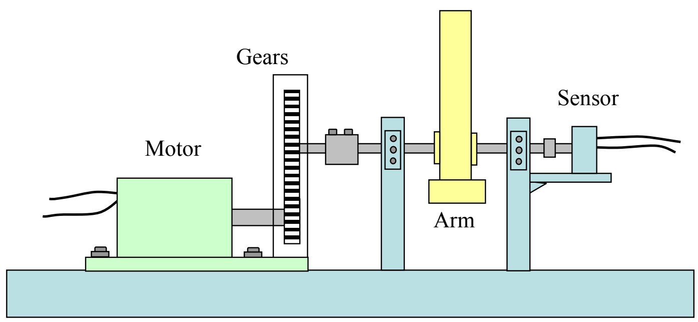

Figure 1: DC motor and arm setup

Figure 2 provides a more detailed schematic of the motor and load, and also provides a block diagram for the continuous control system. The parameters of the system are given in Table 1.

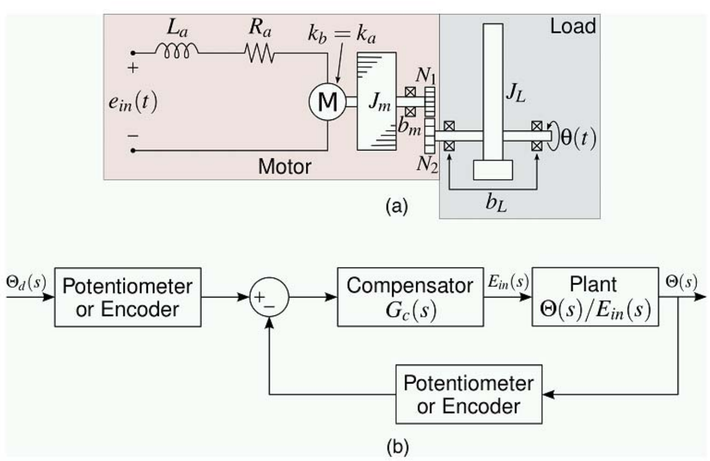

Figure 2: a) System schematic and b) Control system block diagram

The plant model was derived from the schematic in Figure 2, using the parameters in Table 1, which were obtained from motor manufacturer specifications, experiments, and physical measurements. The following transfer function model is used to design and simulate the PI controller,

$$\frac{\Theta(s)}{E_{in}(s)} = \frac{1}{s}\left(\frac{k_b N}{L_a(N^2 J_m + J_L)s^2 + [L_a(N^2 b_m + b_L) + R_a(N^2 J_m + J_L)]s + R_a(N^2 b_m + b_L) + k_b{}^2 N^2}\right) \quad (1)$$

The model accounts for armature inductance and resistance, gear ratio, the motor inertia and rotational damping. To more accurately model the dynamics of the motor, an alternative model that incorporate the stick-slip torque, T_f, and the motor saturation voltage, V_{sat}, was implemented in MATLAB/Simulink. Therefore, the transfer function was decomposed into two parts: the relationship between the input motor voltage and the torque at the output shaft,

$$\frac{T_2(s)}{E_{in}(s)} = \frac{N k_b (J_L s + b_L)}{L_a(N^2 J_m + J_L)s^2 + [L(N^2 b_m + b_L) + R_a(N^2 J_m + J_L)]s + R_a(N^2 b_m + b_L) + k_b{}^2 N^2} \quad (2)$$

and the relationship between the torque and the angular position at the output shaft is:

$$\frac{\Theta_2(s)}{T_2(s)} = \frac{\Theta(s)}{T_2(s)} = \frac{1}{s}\frac{1}{J_L s + b_L} \quad (3)$$

The effect of the stick-slip torque can be modeled by incorporating a dead-zone at the output of the $T_2(s)/E_{in}(s)$ transfer function. This will be illustrated in a later section.

Table 1: Parameter values for motor-arm system

Parameter	Value
Back-emf constant, k_b	0.0436 V-s/rad
Torque constant, k_a	0.0436 N-m/A
Moment of inertia of the motor, J_m	1.6×10^{-6} kg-m^2
Viscous damping of the motor, b_m	1.4×10^{-6} N-m-s/rad
Moment of inertia of the load, J_L	0.0027 kg-m^2
Reduction of gear box, $N = N_2/N_1$	95.9
Inductance of motor, L_a	9.35 mH
Resistance of motor, R_a	17 Ω
Stick-slip torque constant, T_f	2.5×10^{-3} N-m

Controller Design and Specifications

The focus of this study is on the experimental implementation of a PI controller first in an analog circuit using operational amplifiers and then in a digital form using a computer. After experimenting with the analog controller based on operational amplifiers, it was chosen to have only proportional and integral components, whose gains were determined by trial and error using electrical resistors and capacitors until a satisfactory response was accomplished. Such satisfactory response consisted of achieving approximately zero percent overshoot and 5-second settling time for a step response of 1V in the reference potentiometer, which is the same as a step command of 2 revolutions of the arm. Later on, simulations were performed in MATLAB/SIMULINK to compare theoretical and experimental results. A proportional gain of 15 and an integral gain of 1 for the PI controller generated a satisfactory step response of the arm. Some of the difficulties found in the real system are: first, the breakaway torque (dead zone), which requires about 100 mA of current applied to the motor to start moving; and second, the saturation of the amplifiers to provide only output voltage in the range of ±14V. For instance, the amplifiers were saturating at the beginning of the step command not accumulating all the position error that is actually present in the process. Therefore, when the error at the beginning of the step response is 2 revolutions, that is 1V, the proportional amplifier with a gain of 15 is trying to output 15 V, but it can only provide 14V due to its saturation. In order to compare the experimental results with the theoretical results, these non-linear effects were taken into account in the block diagram used for simulations.

Controller Implementation and Results

The implementation and performance of the PI controller for the position of the arm in Figure 1 is presented next using an analog method based on operational amplifiers, a digital controller using LabVIEW and a data acquisition system, and a digital controller using MATLAB/Simulink and a DSP evaluation module.

Analog PI Controller using Operation Amplifiers

Figure 3 presents the schematic representation of an analog proportional and integral (PI) compensator using 741 operational amplifiers. The voltage at the wiper of a 10-turn potentiometer attached to the arm shaft is subtracted from a reference voltage set by another potentiometer to generate the position error of the arm. This error voltage goes through an inverting amplifier with a gain of 15 and through an integrating amplifier with a gain of 1, and the outputs of these two amplifiers are added with a summer amplifier to complete the analog PI controller implementation. The output of the summer amplifier is the input of a power operational amplifier, such as the OPA547 from Texas Instruments, required to handle the current that goes to the DC motor. The power amplifier was adjusted with a gain of 2 in order to compare results with a Logosol LS-5Y power amplifier whose gain was set also to 2. All the output voltages from the operational amplifiers saturated at about ±14V when using a bipolar power supply of ±16V. In other experiments, instead of the OPA547, a Logosol LS-5Y power amplifier was used, its gain was set to 2 and it suffered saturation at ±19V.

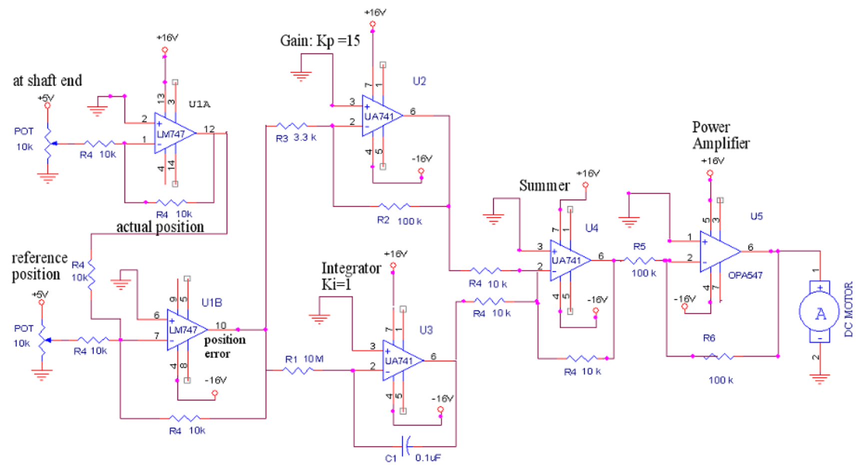

Figure 3: Implementation of an analog PI position controller using operational amplifiers

Figure 4 shows the experimental step responses of the motor-arm system using all three different PI controller implementations. At this moment, notice the two step responses using the analog implementation of the controller with operational amplifiers. The step input was 1V, which corresponds to 2 revolutions of rotation of the arm due to the 0 and 5V applied to the fixed terminals of the 10-turn potentiometers. As mentioned above, for the PI compensator, the proportional gain was set to 15 and the integral gain to 1.

Digital PI Controller using LabVIEW

With less difficulty in the construction of the circuitry but with the relative complexity of requiring programming and interfacing of the computer, data acquisition system, DC motor and potentiometer, the same PI controller was implemented in LabVIEW. For the output from the data acquisition system to drive the DC motor the same power amplifiers used with the analog controller were used. The LabVIEW controller was implemented with a sampling rate of 0.01 seconds set by an external interrupt on the rising edge of a TTL signal from a function generator. It can be observed from Figure 4 that the step responses are very similar in both cases and the steady state error converges to zero. However, it can also be observed that the PI controller implemented with LabVIEW is slightly slower than the same controller implemented with operational amplifiers. This is due to the fact that the analog output (AO) of the data acquisition system is limited to ±10 Vdc, which with the gain of 2 of the power amplifiers might be doubled, but still saturates at the indicated values.

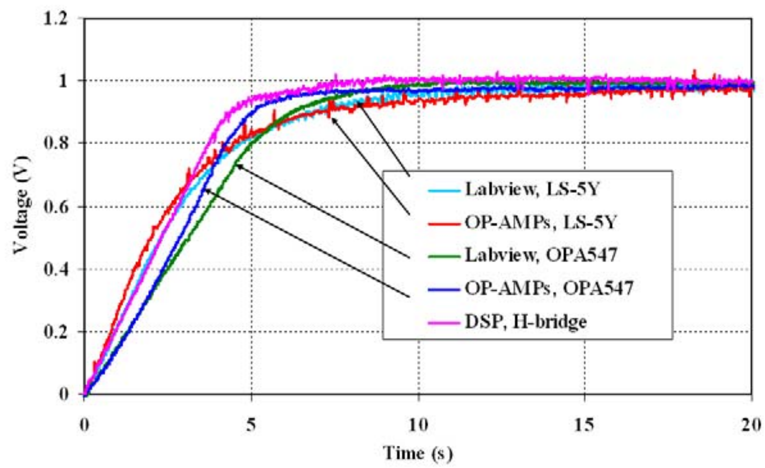

Figure 4: Experimental step responses of various PI position controllers

Digital PI Controller using MATLAB/Simulink and a Digital Signal Processor (DSP)

Using the *Target for TI C2000 Toolbox* in MATLAB/SIMULINK, the design of a digital controller can be linked to *Code Composer Studio* to implement real-time control using a digital signal processor (DSP). A DSP can be programmed directly from MATLAB/SIMULINK and continuous communication of the user with the DSP, and vice versa, can be established using Real Time Data eXchange (RTDX) methods. The reference position of the rotational arm is set through a slider on the computer screen using the RTDX capability in Simulink. To setup RTDX communication, several steps must be taken: a) The RTDX block must be incorporated within the control algorithm, b) RTDX must be enabled, c) RTDX diagnostics must be run, and 4) a MATLAB script free of errors must be created to achieve communication from the host computer to the target DSP, or vice versa. The same proportional-integral controller implemented using LabVIEW and operational amplifiers, was implemented using a PID block in the C28x *Digital Motor Control Library* (DMC) in MATLAB/SIMULINK. Therefore, the proportional, integral, and derivative gains were set to 15, 1, and 0, respectively, and the sampling rate was 0.01 seconds.

An interfacing circuit between the DSP and the DC motor was also required, as shown in Figure 5. This circuit consisted of a buffer, an inverting logic gate, and an H-bridge rated up to 1A of current. The buffer was used to convert the 3.3V signals from the DSP into 5V signals to drive the logic gate and the H-bridge. The inverting logic gate was used to create two opposite logic signals to set the direction of rotation of the motor. The H-bridge was used not only to change the direction of rotation of the motor but also to change its velocity using a pulse width modulated (PWM) signal. The H-bridge was powered using a bipolar ±16V power supply and the maximum voltage provided to the motor was ±14.5V.

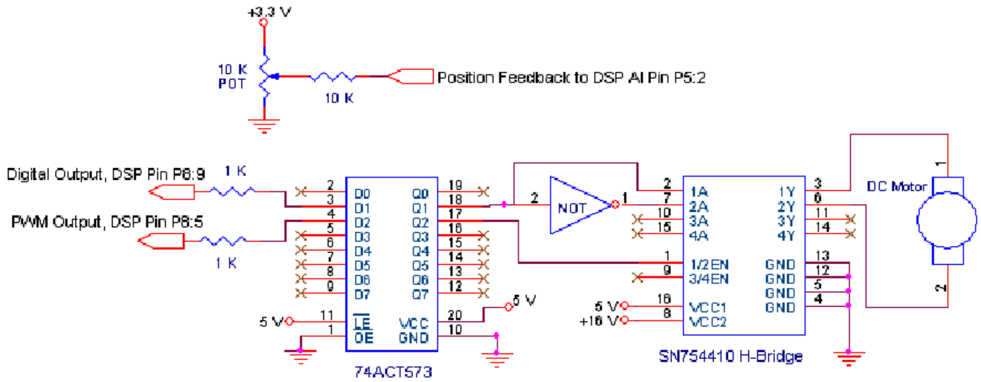

Figure 5: Interface connections between DSP and DC motor.

The feedback voltage used in the algorithm was obtained from a 10KΩ potentiometer attached to the rotating shaft of the arm. To control the direction of rotation of the motor, the sign of the position error was determined in order to set the logic of a digital output (DO) pin on the DSP. Both, the digital output (DO) and its inverted value were used to set the motor's rotation direction. The SN754410 integrated circuit is a dual H-bridge device with enable (EN) pins to which a PWM signal from the DSP was connected. The duty cycle of the PWM signal was regulated by the PID controller.

Simulations and Theoretical Results

A block diagram was created in MATLAB/Simulink, as shown in Figure 6, to estimate the step response of the motor-arm system. Note that a dead zone was included to take into account the breakaway or stick-slip torque required to start moving the arm. The motor did not move until a voltage greater than 1.8V was applied to it. The manufacturer provides a friction torque rating for the motor. That was used to set the limits on the dead-zone in the Simulink block diagram. Also a saturation block was added to account for the fact that the operational as well as the power amplifiers have a voltage limit due to the power supply.

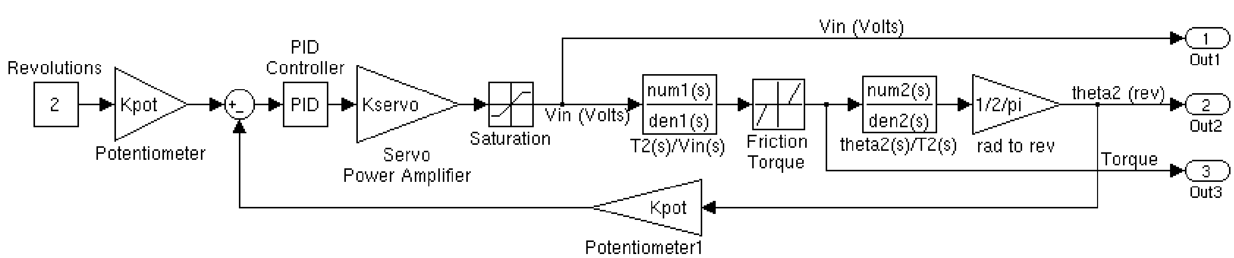

Figure 3: Simulink block diagram

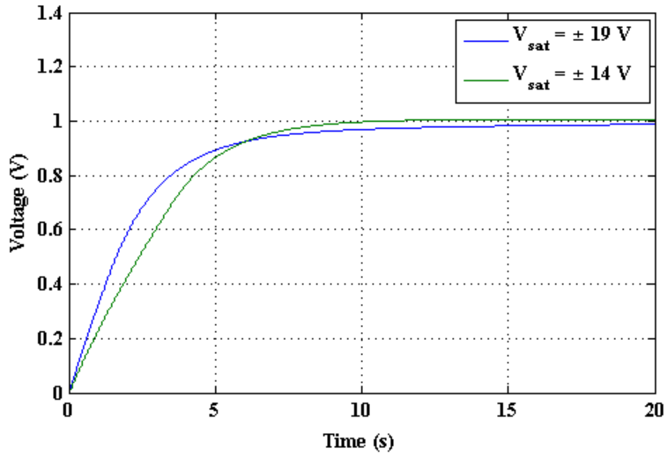

Figure 4: Simulated step responses

The results of the simulation in Figure 7, show two step responses equivalent to motion of two revolutions of the arm. One of the step responses is when the saturation voltage is ±19V and the other when it is ±14V; which occur when using a Logosol power amplifier with a 24Vdc power supply, and when using an OPA547 power amplifier with a bipolar power supply of ±16V, respectively.

Discussion of Results

There seems to be a disconnection between theoretical control systems courses and the actual practice and implementation of control systems in real engineering applications, in particular for mechanical engineering students at UTPA and in other institutions[1,10]. In theoretical control systems courses, students learn to design many types of controllers using different manual and sophisticated computer techniques that allow obtaining results through simulations. However, not very often students get to implement their controllers, not even the simple ones, in actual systems. There is a need for prototype systems to improve student learning of control systems concepts that might be otherwise difficult to understand[1]. Though it is more challenging to implement such controllers in real engineering applications, it is also more interesting and fun.

By developing hand-on activities and didactic setups, like the one in Figure 1, for Control Systems and Mechatronics instruction, this study will contribute to strengthen the weak connection that exits between theory and experimental control systems instruction at UTPA. Sophisticated computer software packages has been developed to avoid having to build and implement every control system that is designed; however, the experimental work in control systems is a truly enriching experience for any learner involved in this area. Throughout this study, it was determined that many conditions and experiences occurring in real life while implementing control systems, even though it was a relatively simple PI controller, are not apparent when designing the control system in theory. For instance, voltage and current limits as well as saturation effects are common in control system experimentation. Besides that, for the motor-arm to start moving, it requires a voltage of at least ±1.8V, which indicates a dead zone due mainly to static friction and inertia in the motor, gears, and bearings. This type of conditions can be taken into account in the simulations but usually they are avoided for simplicity.

The most important technical results of this work were presented in Figure 5, which shows the motor-arm experimental step responses obtained with a proportional and integral, PI, controller with gains of 15 and 1, respectively. The implementation of the controller was performed using operational amplifiers and LabVIEW, both using two different types of power amplifiers; and, in addition, the same controller was implemented using a MATLAB/SIMULINK and a DSP, but in this case an H-bridge was used as the motor's driver. Comparing the experimental step responses with the simulations in Figure 7, it is concluded that the controller was validated in practice and that the mathematical model of the system is closely represents the actual system.

Conclusions

Students could achieve better understanding of concepts by having hands-on activities implementing basic control system compensators in engineering classic control, mechatronic, and similar courses. Even though, there are multiple ways to implement simple control compensators, many mechanical engineering curricula expose students only to theory and simulation practices, leaving out a complexity of factors and experiences that students would otherwise confront when performing real implementation of control systems.

Definitely, digital control systems are much more flexible than analog controllers, allowing modifications, tuning, and any changes in the control system to be much easier to perform in digital than in analog controller implementation.

In this study a PI controller was designed and implemented in three different ways and it was also studied by simulation using MATLAB/Simulink. The step responses of the motor-arm system were similar in the experiments and in the simulations, which validated the mathematical model of the system.

References

[1] Murphree, J., Brezinski, B., Parker, J. 2002, Using a fixed-point digital signal processor as a PID controller, ASEE Annual Conference Proceedings Vive L'ingenieur, 9419-9426

[2] Oelschlegel, M., Bachnak, R., 2003, A project in PID temperature control and loop tuning, ASEE Annual Conference Proceedings Staying in Tune with Engineering Education, 13863-13873.

[3] Ramachandran, R., Ordonez, R., Farrell, S., Gephardt, Z., Zhang, H., 2001, Multidiscplinary control experiments based on the proportional-integral-derivative (PID) concept, ASEE Annual Conference Proceedings Peppers, Papers, Pueblos and Professors, 8759-8775

[4] Basilio,J., Matos, S., 2002, Design of PI and PID controllers with transient performance specification, IEEE Transactions on Education, 45 (4) November 364-370

[5] Kelly, R., Moreno, J., 2001, Learning PID structures in an introductory course of automatic control, IEEE Transactions on Education, 44 (4) November 373-376

[6] Katsuhiko, O., 1995, Discrete-Time Control System, Prentice Hall.

[7] URL: http://www.yildiz.edu.tr/~adelibas/pdf/AD_101_mechrob04.pdf, Real-Time DC Motor Position Control by Fuzzy Logic and PID Controllers Using LabVIEW.

[8] Li, Y., Ang, K., Chong, G., 2006, PID control system analysis and design, IEEE Control Systems Magazine, 26 (1) February 32-41

[9] Ang, K., Chong, G., Li, Y., 2005, PID control system analysis, design, and technology, IEEE Transactions on Control Systems Technology 13 (4) July 559 – 576

[10] Newcomer, J., 1998, Design project based approach to teaching automatic control theory to mechanical engineers, ASEE Annual Conference and Exposition 7775-7789.

HORACIO VASQUEZ
Dr. Vasquez is an Assistant Professor of Mechanical Engineering at the University of Texas Pan-American. He teaches courses in Mechatronics and Digital Control Systems. His research interests include Mechatronics, Instrumentation, and Automatic Control Systems.

JAVIER KYPUROS
Dr. Kypuros is an Assistant Professor of Mechanical Engineering at UTPA where he teaches courses that include System Dynamics and Automatic Control Systems. Currently, Dr. Kypuros researches methods to model, sense, and control smart systems and structures.

ROY VILLANUEVA
Roy is a graduate student pursuing a Master's degree in Mechanical Engineering at the University of Texas Pan-American. He graduated from UTPA in 2007 with a B.Sc.degree in Mechanical Engineering.