AC 2009-278: LABORATORY EXPERIMENTATION AND REAL-TIME COMPUTING: AN INTEGRATED ENVIRONMENT

Ahmed Rubaai, Howard University

AHMED RUBAAI received the M.S.E.E degree from Case Western Reserve University, Cleveland, Ohio, in 1983, and the Dr. Eng. degree from Cleveland State University, Cleveland, Ohio, in 1988. In 1988, he joined Howard University, Washington, D.C., as a faculty member, where he is presently a Professor of Electrical Engineering. He is the Founder and Lead Developer of Howard University Multipurpose and Multidisciplinary Automation and Control Laboratory. His research interests include built-in intelligent controller for high performance industrial drives, hardware testing in laboratory, and computer-aided design for undergraduate engineering education.

Ramesh Chawla, Howard University

Dr. RAMESH C. CHAWLA is Chair and professor of chemical engineering at Howard University. He has over thirty years of experience in teaching, research and industrial consulting in the fields of chemical and environmental engineering. His research and teaching interests include reaction engineering, separation processes, environmental engineering, and physical, chemical and biological treatment of hazardous wastes.

LABORATORY EXPERIMENTATION AND REAL-TIME COMPUTING: AN INTEGRATED ENVIRONMENT

ABSTRACT

This paper presents an integrated environment for rapid control prototyping that allows rapid realization of novel designs, from the initial design phase until the final steps of code generation. It uses a collection of tools that include both software (MATLAB/Simulink) and an off-the-shelf hardware (dSPACE DSP DS1104). The integrated environment presented in this paper has many educational advantages as compared to multi-environment settings. The main features of this environment are: 1) controller code can be generated automatically for hardware implementation; 2) different languages can be used to describe different parts of the system. In particular, Simulink block diagrams can be used to define the control structure, tune the controller parameters and reference signals online, while the experiments are in progress without having to rebuild and download a new Simulink model to the DS1104 board; and 3) ease of operation especially by means of a simple graphical user interface. The laboratory environment was used in teaching an introductory laboratory control course. The objective is to promote control-systems education with laboratory experimentation. Course assessment showed a high level of students' satisfaction with the course content and its structure. The students stated that the process helped them to apply modern design tools to a real time system.

INTRODUCTION

The study of control systems has been cited as a subject that is heavily based on abstract mathematical concepts[1]. This theoretical base has been considered a major problem with students unable to apply the coursework that is completed in the classroom to real-life systems. This problem has not gone unnoticed in the field of education today, and there have been great leaps in the creation of more “hands-on” teaching methods that lend themselves to industrial applications[2]. Throughout schools and universities within the United States and internationally, there has been growing interest in the use of practical control concepts in and beyond the classroom. This has been accomplished to a large extent through the use of laboratory courses, with incorporation of technology tools that enable students to work on different real-world control configurations. This adjustment to incorporate the more practical format into the classroom has taken different forms throughout the academic world. In the Technische Universiteit Eindhoven, The Netherlands, the modeling of control systems is an important part of their Bachelor’s in mechanical engineering degree curriculum[3]. There is a gradual introduction to real world systems that begins with a lower level course where the students are introduced to mathematical concepts and A/D conversion and ends with a final year project that incorporates the manipulation of various feedback controllers to accomplish a specific task. In this way the students are transported from the theoretical understanding to actual applications by the end of the degree program. At the Department of Automatic Control at the Lund Institute of Technology in Sweden[4], all disciplines in their four and a half year Master of Science degree, excluding chemical and biotechnical engineering, must complete a basic control course. The second half of this course involves the assignment of control projects in conjunction with the lectures, which is another clear indication that there is great importance placed on the practical applications of control theory. All control courses have three mandatory four-hour labs that make use of mobile desktop processes and standard computing equipment. The Institute is also credited with having “pioneered the teaching of real-time programming and real-time systems,” [4]. At the University of Maryland, College Park [5], their main focus with regard to the practical application of control systems is a multidisciplinary senior-level course (in the Bachelor’s degree program of computer and electrical, mechanical and aerospace engineering) that combines digital control and networks with information technology. One of the major advantages seen at Maryland is in the use of an all-digital controls lab, which allows controller-implementation using relatively cheap computers. Another article[6] promotes the control-systems laboratory at the University of Illinois at Urbana-Champaign. An appealing quality of this facility is that it is shared among several departments. At Howard University, the study of control has been accelerated by the integration of motion controls laboratory, which affords the student an opportunity to interact and utilize an “embeddable dSPACE digital signal processor (DSP)-based data acquisition and control system [7]. This is seen by Howard University as a solution to the need for a cost effective, “hands on instructional laboratory” which would “adequately provide hands on experience necessary for effective learning.” Another key aspect of this laboratory is the close integration of the conventional simulation tools MATLAB and Simulink™.

These are just some examples of the manner in which the institution of education has modified itself to incorporate the need for practical applications of control concepts. With regard to the software tools that have become popular for the creation and modeling of control systems in the lab, it has been found that many commercial entities offer several products that can be used in the laboratory environment to illustrate control systems. In each lab, there exists some consistency in the tools of choice. The MATLAB software package is undoubtedly the most common and most powerful tool for creating an environment for control systems design and simulation [2-4, 7]. There are several applications under MATLAB that have been used in this design and simulation process. These applications include QadScope and Wintarget [3]. QadScope is a scope-like application for measuring purposes. It supports a wide range of inputs and outputs with built-in frequency-domain analysis, while WinTarget is “a real-time target running under Simulink/Real-Time Workshop”. The two tools work together to create a real-time application that facilitates a simple method for the construction of Simulink™ models [3, 7]. Other software that is used in the experimental process is Linux 2.1.18 (with specific program extensions), and Java applications [4]. The use of MATLAB/Simulink™ overshadows all other mechanisms for control system modeling, as it is seen to generate the code independently, removing the need for Real-Time Workshop and other such software tools that were needed to facilitate coding. Another point in favor of using MATLAB/Simulink™ is in the creation of an environment similar to an ideal real-time control platform. Linux and Java are cited as incapable of producing the best real-time platform because of “the non-determinism caused by the automatic memory management in Java [3]”. While the speeds of most modern computers minimize this drawback, the Simulink™ model still offers the best real-time applications. A few other software tools that are utilized in laboratories today include RTLinux (Real-Time Linux) [2] and Simulinux-RT[5].

With regard to the types of controllers that have been utilized in the educational arena, there are a number of practical approaches being used for the illustration of the control systems concepts [8-12]. Regardless of the particular software being used or the specific type of controller being built, it is obvious that educational bodies worldwide have adjusted their structure to facilitate a greater exposure to the application of the abstract theory behind control systems to real-world, real-time processes. With the technology available to various laboratories and schools continuously evolving, the students will soon be able to have all the required exposure and ability required to enter the work field with more than just a mere exposure to real-world applications of control theory. They will actually enter with a clear practical understanding.

Underlying Educational Objectives

Laboratory experiments using real-time systems are necessary in control education. Experiments help the students understand the theoretical concepts and provide important motivation. It is therefore essential for the students to have a thorough understanding of hands-on experimentation and real-time systems. Three fundamental educational objectives are:

1. To apply state-of-the-art knowledge to help students understand what they have learned.
2. To train a new cadre of graduates who value experimentation as an essential and natural part of solving engineering problems.
3. To develop good experimental skills.

Hence, the controls engineering education becomes more attractive and meaningful to the students. To achieve these objectives and make it possible for the students to perform experiments, the lead author has developed six novel laboratory workstations using state-of-the-art control systems technology.

Student Learning Outcomes

This paper describes a stimulating educational environment that emphasizes the role of hands-on experiments. The fundamental student learning outcomes of the control laboratory course are to demonstrate the following:

1) An ability to design, build, or assemble a part or product that configures control systems especially adapted to automation applications.
2) An ability to conduct experiments for measurements and analysis of feedback controls, and to write effective laboratory reports.
3) An ability to use MATLAB/ Simulink GUI to build a real-time model.
4) An ability to use dSPACE DSP ControlDesk GUI for real-time control.
5) An ability to achieve adequate learning skills in testing and debugging a prototype using appropriate engineering tools and learn how to be an experimenter.

Hardware Selection

Primarily, making a decision on a set of hardware to interface between the host computer and the process (system to be controlled) was a challenge. Finding a compatible set of hardware that would work with MATLAB/SIMULINK/Real-Time Windows Target by MathWorks[13] was also a challenge. The intention is to use a general-purpose DSP controller board that controls the operator interface, performs data acquisition, and executes the control algorithm. Our search led us to dSPACE DSP DS 1104 controller board[14] which could meet our specifications from a hardware point of view. The DS1104 controller board is equipped with a real-time interface (RTI) to MATLAB/Simulink by which the controller board converts the Simulink block-set to machine code. In this regard, the student writes no code at all. The dSPACE DSP has also its own ControlDesk graphical-user-interface (GUI), which helps the students to control experiments. One of the salient features of the dSPACE DSP DS 1104 is the ease of building real-time applications.

Rapid Control Prototyping Environment

The fusion of dSPACE DSP/MATLAB Simulink and MATLAB Real-Time Workshop (RTW) fruitfully produced a rapid control prototyping tool that provided a means for the rapid development and testing of control algorithms by real-time control of an actual target system through a flexible, extensible multiprocessor environment. Under this unique environment undergraduate students may well perform computer simulation, evaluate the simulated response of a system, develop, and verify the performance of traditional and advanced control laws in a simulated mode. The students can then easily install the developed controllers to hardware all within the same routine interface. The student has the ability to simulate a control system against a model of the actual target system, test and refine the controller, and then deploy the controller to the dSPACE DSP without directly coding any software. The dSPACE DSP executes, gathers inputs, computes the control algorithms, and commands the output to the connected actuators. For the drive system under study the students actually did control the actual target system since they could set it up in the lab. In the absence of the hardware equipment of the actual system, the student could use a model since he/she does not have an actual system to control.

To design a controller, the students download a Simulink template file and connect the signals describing the reference, output, and command signals with suitable Simulink blocks. Simulink provides a graphical user interface (GUI) for building system models as block diagrams using click-and-drag mouse operation. Once the controller is built in Simulink's block diagram, the student can utilize the MATLAB RTW routine that can automatically generate C code from the Simulink block diagram. The C code generated by RTW is used with the commercial hardware dSPACE DS 1104 DSP-board for real-time control. Then, the interface between Simulink and the dSPACE DS 1104 DSP allows the control algorithms to be run on the hardware of the DS1104. Once the control algorithm sets in motion, the student can validate the design using the laboratory platform at the click of a mouse button. A feature of this environment is a simple user interface, which requires a basic knowledge of MATLAB/Simulink for designing the controller.

Laboratory Setup

The workstation described here offers many possibilities for experimentation on controls similar to those that students will encounter in the "the real world." Figure 1a shows a block diagram of the laboratory hardware apparatus. The key hardware element of such capability is an embeddable dSPACE digital signal processor (DSP) that can be connected to various sensors and actuators, depending upon the system objectives. The dSPACE DSP that was utilized in the laboratory is the DS1104 board. The DSP DS1104 equipped with its own graphical-user-interface (GUI), allows the drag-and-drop reconfiguration of the user interface and offers the required level of interaction with the process. The laboratory experiment runs completely on the real-time hardware dSPACE DSP DS1104 board. The workstation is equipped with state-of-the-art Dell OPTIPLEX 745 MINITOWER Personal Computer (PC) which is used as a host computer. Both MATLAB/SIMULINK^{TM} and the dSPACE ControlDesk[14] are installed on the PC. The DS1104 board transforms the host computer into powerful system for rapid control prototyping. The dSPACE ControlDesk made it possible to control the system in real-time and monitor the impact of the controller parameters. The real-time interface provides Simulink blocks for graphical configuration of A/D, D/A, digital I/O lines, incremental encoder interface and PWM. The DS1104 board is connected to the PC via a 5-V PCI slot. The workstation is also equipped with state-of-the-art 4-Channel Tektronix digital oscilloscope, 24V DC Programmable Power Supply (PSP-603), and custom made Switch-Mode Board with two independent 3-phase PWM inverters. The custom made switching board is used to vary the dynamic load of the PMDC generator and has complete digital/analog interface with the dSPACE controller board. The 4-Channel 200MHz oscilloscope is used for real-time measurements. The process to be controlled is 1-hp 3000 rev/min three-phase brushless DC servomotor, which was manufactured by Moog Aerospace[15]. It is equipped with a resolver and is coupled via a torque transducer. A Moog T200-410 adjustable speed drive[16] is employed to steer the motor. A variable auto-transformer is used to supply the driving circuit with AC voltage of 230V. A power supply is also used to supply the inverter component of the driving circuit with 24V DC. The laboratory workstation described here is used as a platform for teaching several fundamental concepts in controls and embedded computing. Figure 1b displays a snapshot of the laboratory setup.

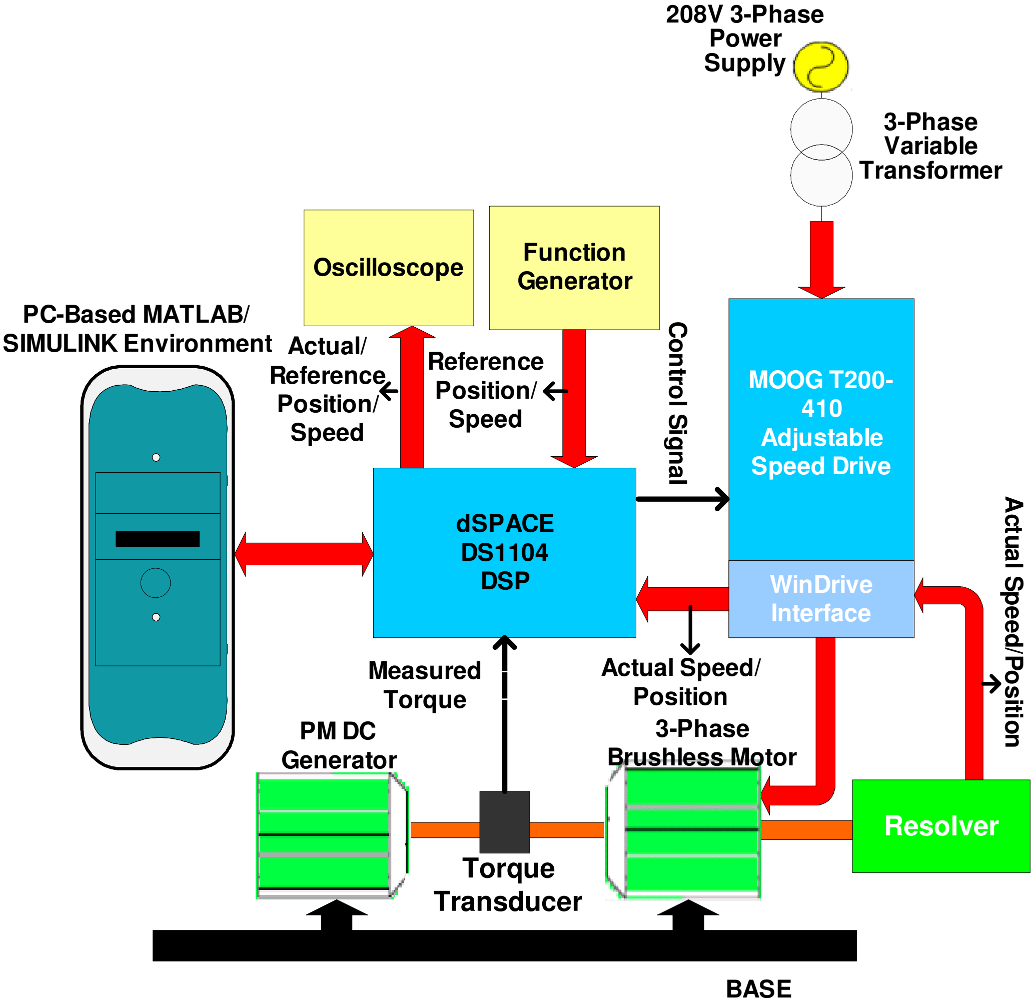


Fig. 1a Block diagram of the laboratory hardware apparatus

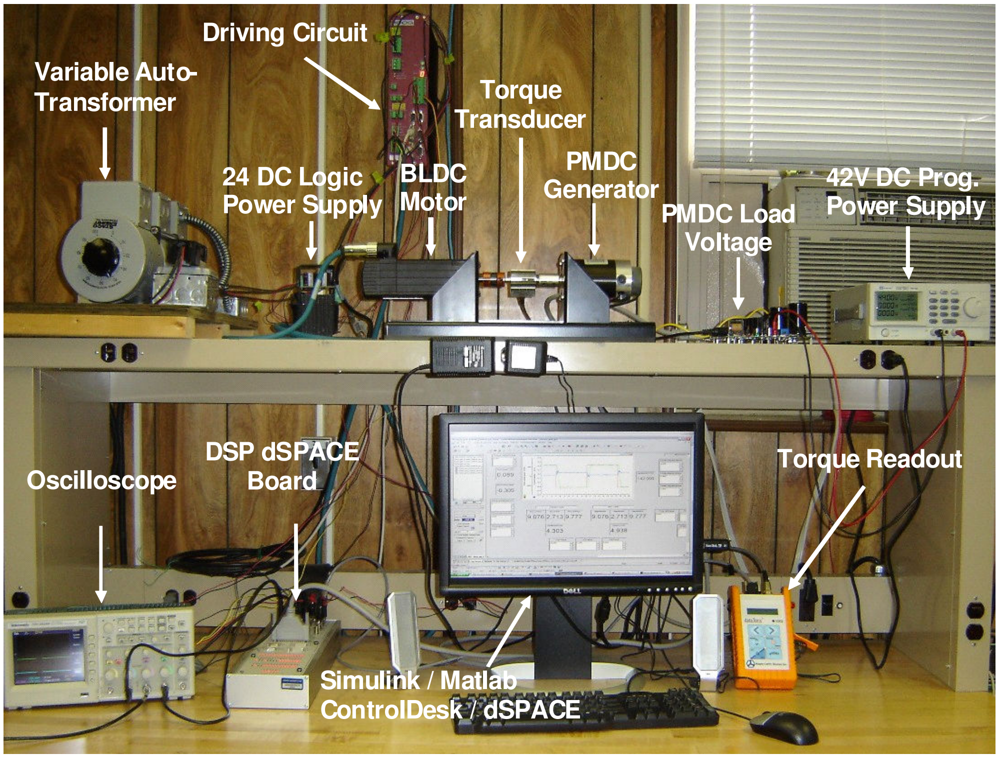


Fig. 1b Photo of the laboratory setup

Hands on Laboratory Experimentation

In controls engineering education there is a need to apply theoretical control algorithms to laboratory applications to foster better understanding and increased experience in the student. A challenge to having students implement control experiments is their unfamiliarity with the various hardware and software at the outset of education lab courses. In many engineering undergraduate control courses, however, only traditional control and state variable regulator methods are considered. Therefore, the students focused on these methods as a foundation for future development to expand into other design methods. As a part of the control laboratory course for the senior class of 2008, different groups of students were assigned to design and implement several controllers currently in use in practice:

1) Proportional-Plus-Integral (PI) controller that incorporates an anti-windup scheme,
2) Proportional-Plus-Derivative (PID) controller (also with an anti-windup scheme),
3) State-Feedback Pole Placement controller-based observer.

Through the modeling of different types of controllers, students are expected to gain a better understanding for the differences between various controllers as well as a greater appreciation of how they are applied to real life. The controllers have been designed to meet the following specifications:

- Minimize the overshoot that may occur during the transients;
- Achieve stability robustness and short transient;

- Regulate load disturbance rejection, including performance at steady state;
- Attenuate noise and robustness against environment uncertainty.

The students were asked to design a state feedback pole placement speed tracking controller-based observer to specifications and analyze its responses to square-wave inputs, and sinusoidal inputs. This controller was chosen since it is a product of modern control theory and it is desirable to expose the student to an algorithm based on this theory. The inclusion of state-feedback also provides the student the opportunity to contrast modern and classical approaches to a control problem. The controller is built in Simulink and implemented in the dSPACE DS1104 DSP.

Theoretical background

The experiment uses a three-phase brushless DC motor (BDCM) which is electronically controlled by the MOOG WinDrive which performs the electronic motor commutation required by a BDCM. The dSPACE DS1104 digital signal processor (DSP)-based real-time data acquisition control (DAC) board provides the interface between the WinDrive and the Simulink programs. The basic equivalent circuit for a three phase BDCM is shown in Figure 2.

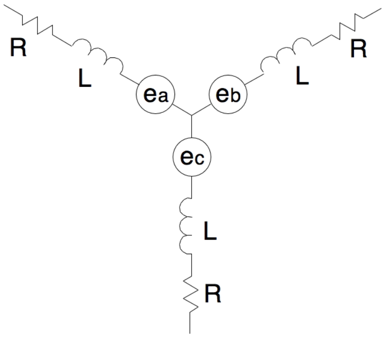
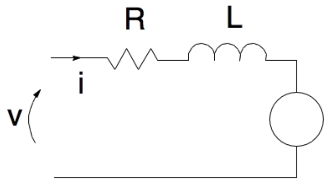


Fig. 2 BDCM equivalent circuit

The circuit equations for the motor are:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \tag{1}$$

On a per phase basis, the equivalent BDCM circuit can be shown as Figure 3.



$$e = K_b \omega$$
$$T = K_T i$$

Fig. 3 Dynamic per phase BDCM equivalent circuit

The per phase equivalent circuit is similar to that of the standard DC motor and by controlling the input voltage to the motor, the motor speed (ω) is controlled. The controller design uses the pole placement technique to obtain the desired speed control output. The general state equations for a system are:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{2}$$

Students used the speed (ω) and current (i) as the state variables x_1 and x_2; and voltage (v) as the input u which gives the following state equations:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -b/J & K_T/J \\ -K_b/L & -R/L \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/L \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3}$$

where J= 0.01 N.m.sec^2/rad; L= 0.05 H; R= 2.2 ohms; K_b= 0.25v sec/ rad; b= 0.03 N-m-sec/rad. Note that for this state model, the output y equals speed ω. From equation (3), the matrices A, B, and C are known. MATLAB can be used to find the control matrix K. Select poles (p1, p2) anywhere on the left hand plane (complex conjugates are typical) and use the "place" command [`K=place (A,B,[p1,p2]);`]. The observer equation is given by

$$\dot{\tilde{x}} = A\tilde{x} + Bu + L(y - \tilde{y}) \tag{4}$$

In this system, a replica of the original plant is modified with the observer matrix L. A set of predicted states, $\tilde{x}$, is calculated and used with the control matrix K to create the error signal e as

$$e = K\tilde{x} \tag{5}$$

Since we require the dynamics of the observer to be much faster than the original system, we place the observer poles (op1, op2) at least five times farther to the left than the original poles. MATLAB can be used here again to calculate L

`L=place (A',C',[op1,op2]');` (6)

Also note that there will also be a saturation block added to the controller to prevent the motor from being damaged by excessive input voltages. The input voltage equations become

$$\begin{aligned} &u = r - K\tilde{x} \text{ where} \\ &u = u_c \text{ when } |u_c| \le |u_{SAT}| \\ &u = u_{SAT} \text{ when } |u_c| \ge |u_{SAT}| \end{aligned} \tag{7}$$

where u is the actual input voltage to the motor, u_{SAT} is the max/min value allowed for the input voltage, and u_c is the input voltage prior to the saturation block. All information is now available to set up the system in Simulink. The controller is designed as a state feedback pole-placement controller with observer. The Simulink model implementing the state feedback controller-based observer is shown in Figure 4.

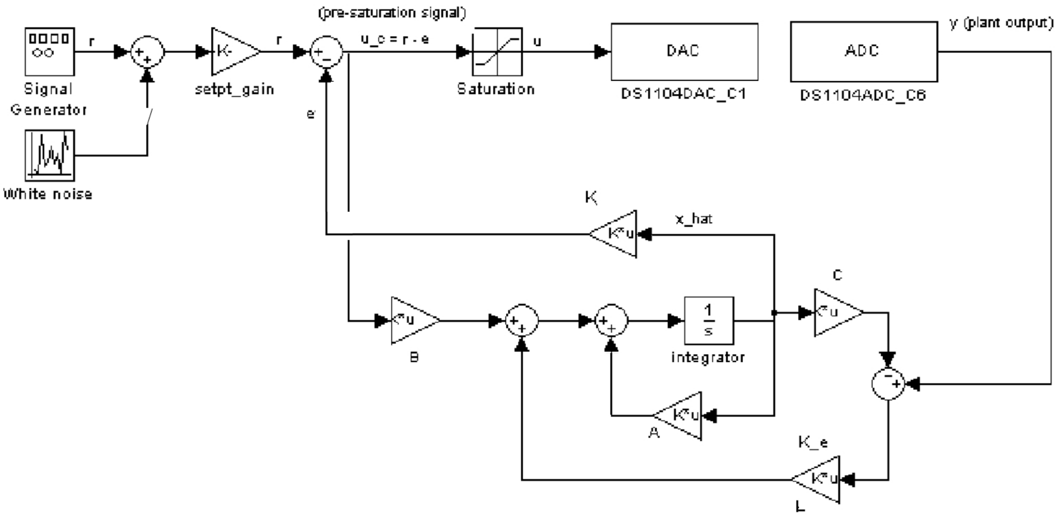


Fig. 4 Simulink block diagram of control system

Input to the controller is the reference signal provided by the signal Generator. The DAC (Digital to Analog Conversion) and the ADC (Analog to Digital Conversion) blocks represent the interface between dSPACE DSP and the motor drive system (plant). The only known state is velocity, which is the output of the ADC block and is represented by y (refer to (2)). Figure 5 shows the block diagram of the observer described by (4).

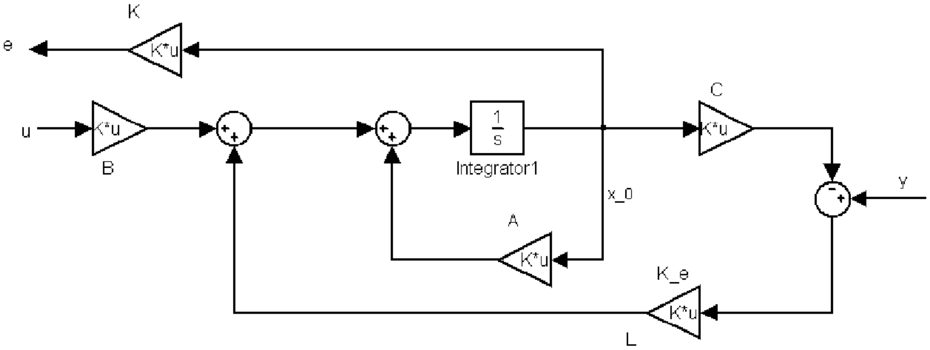


Fig. 5 Block diagram of observer

Experimental Implementation

After the students observed the controller baseline operation, the performance of the proposed controller was evaluated by completing several test cases in the laboratory under different operating conditions. Two reference signals were used in the experiment, a square wave and a sinusoidal wave. Figures 6a and 6b illustrate the performance of the controller under loading conditions for square-wave inputs and sinusoidal-wave inputs, respectively. The students observed that, in every case, the controller brings the measured rotor speed to the desired value smoothly and with a slight overshoot. Figures 6a and 6b indicate that the controller tracked these reference signals very well. The slight overshoot was almost diminished with the sinusoidal reference since the controller was not required to react to the abrupt change of direction. The discussion by the students can address these issues with reference to the performance specifications. For the baseline case, the rotor speed was set to 750 rpm and this was also the limit set within the saturation block.

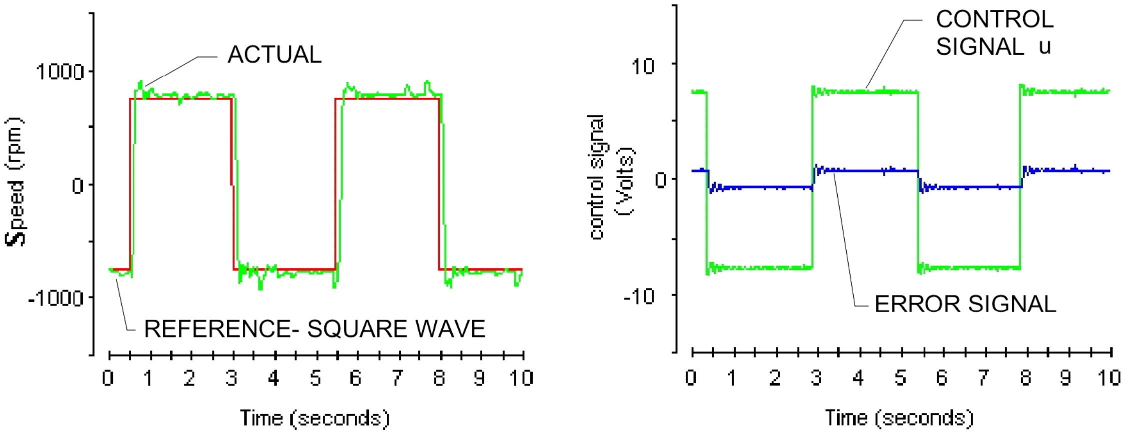


Figure 6a: Initial setup- Baseline results- Square Wave

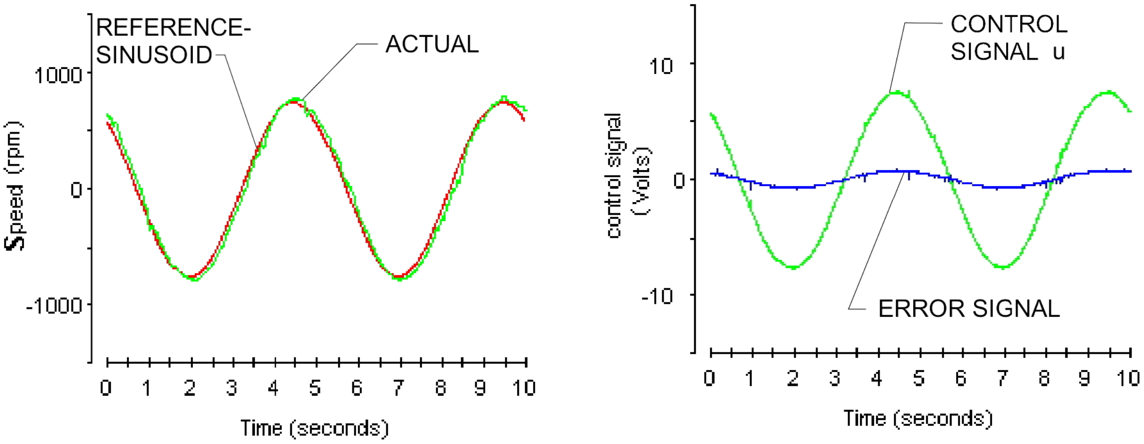


Figure 6b: Initial setup- Baseline results- Sinusoidal Wave

The saturation limits was tested by setting the reference signal to a 900 rpm (9 volt) set-point which is beyond the 750 rpm saturation limit. The baseline test was run again and the effect of the saturation on the system was observed. Figures 7a and 7b show that the pre-saturation control signal was limited by the saturation limits and that the motor was controlled to the 750 rpm saturation block setting.

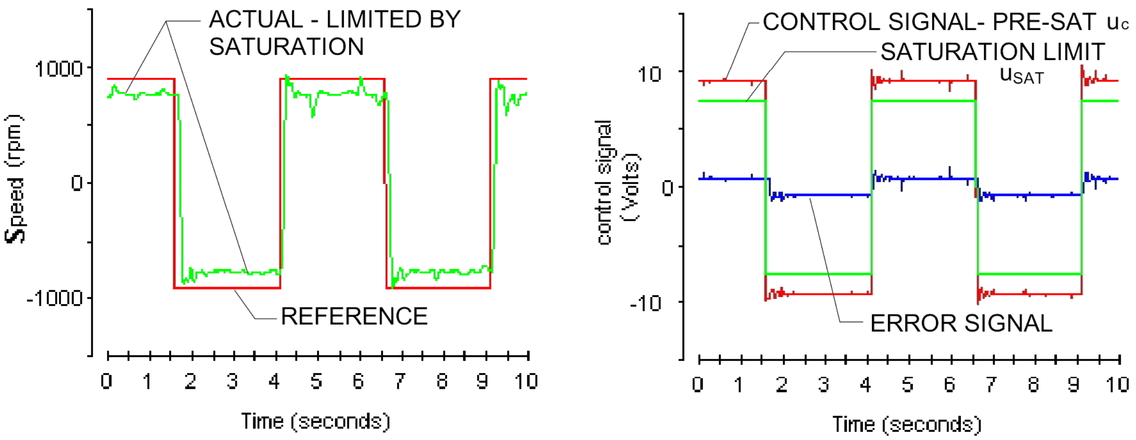


Figure 7a – Effects of saturation block- Square Wave

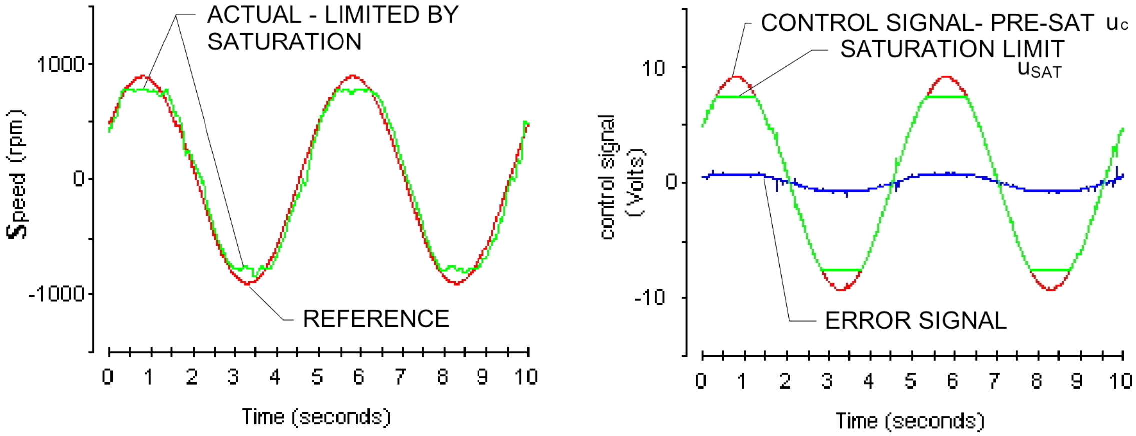


Figure 7b – Effects of saturation block- Sinusoidal Wave

The disturbance rejection aspect of closed-loop control is also introduced to students. Figure 8 illustrates the effects of the external disturbance on the performance of the controller. The effect of external disturbances applied momentarily at 1.5 seconds, 5 seconds and 7.5 seconds. At each point where the disturbance was introduced, note that the motor slows but then the controller acts to return the motor rotor speed to the desired reference setting. The typical motor overshoot is also observed as the motor reaches the reference setting and then settles back. Disturbance response is also assessed with respect to the robustness concepts. The discussion by the students can address the effect of the external disturbance on the performance of the system. At this stage, the design appears to be satisfactory even under sudden external disturbance.

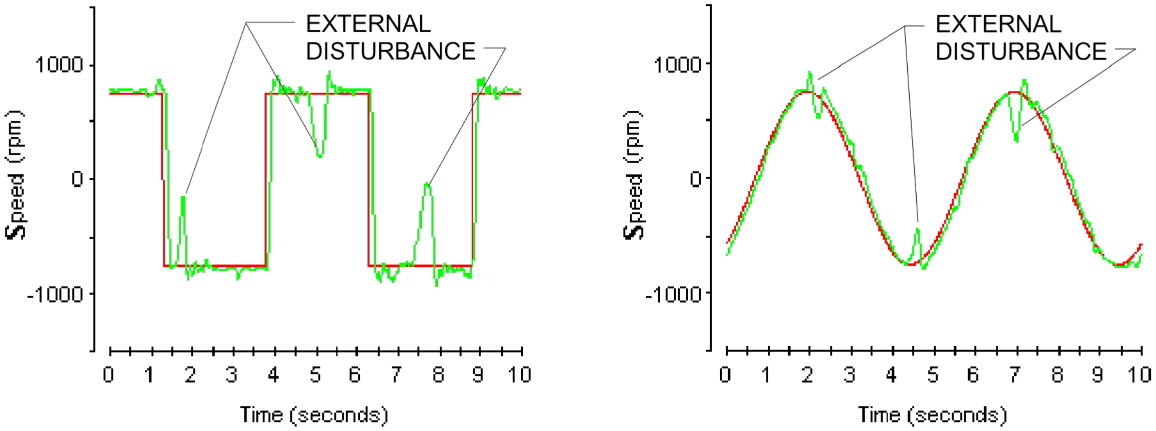


Figure 8 Effects of external disturbances

Clearly, the students manage to reduce both the overshoot and the extent of oscillations conditions by manipulating the control gains.

Assessment/Evaluation

At the end of the 2008 spring semester student are asked to rate the value of the component of the laboratory exercises. The students respond with a rating of 1 (Strongly Disagree) to 4 (Strongly Agree). Table I displays the student self survey. Over 30 undergraduates participated in the survey and provided anonymous comments about the laboratory exercises.

Review of these comments indicated that the students liked using the framework of MATLAB/Simulink/dSPACE DSP and believed that the process helped them to apply modern design tools to a real time system. Additionally, the students commented that more formal instruction of MATLAB/Simulink in courses prior to the laboratory control course would be helpful. The lab course was a success because of the value it added to strengthen their understanding of learned principles, applying their engineering knowledge to understand new systems, and the opportunity to practice their engineering skills. Implementation of the laboratory exercises gave the students a sense of accomplishment. Much enjoyment was realized in the implementation of the dSPACE DSP system and Simulink intuitive model-based programming. Students also improved their written and oral communication skills as well as experimental proficiency from the hands-on approach. In addition, working in a team of diverse students encouraged constant discourse and exchange of ideas through the course of the design of the controllers. All students have reacted positively to the experiment and overall interest has definitely been increased. The authors found the laboratory experiment to help reach the goal of lab-based teaching. The experiment definitely required critical thinking by the students. Some comments from student evaluations concerning the laboratory control course are as follows:

- “The laboratory course was the most challenging and the most rewarding course that I have taken thus far.”
- “I found the control laboratory course to be a first-rate lab which prepared me for the industry---not only by the laboratory exercises it offered but most importantly, by teaching me how to use deductive logic in solving problems.”

- “I find it easier to understand material presented via laboratory experiment than solely via a textbook.”
- “A very good practical experience. Finally, hands-on labs.”

Table I Control Laboratory Assessment

Use the following scale: 4 – Strongly Agree 3 – Agree 2 – Disagree 1–Strongly Disagree

Average	
3.5	1) Using WinDrive Graphical Interface helped my understanding of second order system.
3.5	2) Using WinDrive helped my understanding of proportional and integral gains.
4.0	3) Using dSPACE DSP helped my understanding of building real-time applications.
4.0	4) I am comfortable using MATLAB/Simulink to build a model
4.0	5) I am comfortable using MATLAB/Simulink to design a controller
4.0	6) I am comfortable using MATLAB Real-Time Workshop and dSPACE DSP to build a real-time model in Simulink
4.0	7) I am comfortable using ControlDesk of dSPACE DSP to implement a real-time controller by varying gains and observing the response
3.5	8) Implementing a PI controller in dSPACE DSP helped my understanding of industrial controls
3.5	9) Implementing a dual-loop controller in dSPACE DSP helped my understanding of industrial controls
3.5	10) Implementing a PID controller in dSPACE DSP helped my understanding of industrial controls
3.5	11) Implementing a state-variable controller in dSPACE DSP helped my understanding of industrial controls

I learned the most from the control lab that: ____

I learned the least from the control lab that: ____

Conclusions

This article has described the development, implementation, and demonstration of real time laboratory platform for use in undergraduate laboratory control courses. The platform has been implemented using off-the-shelf components, such as dSPACE DSP boards and standard personal computers. Controllers were implemented using MATLAB/Simulink environment. The values of the parameters of the controller and reference input signals were modified dynamically and their effects were observed on the scope without having to download the new Simulink model. The laboratory sessions are adaptable and enable a large group of undergraduate students to carry out variety of laboratory experiments. Students have gained extensive experience with the implementation of industrial controllers. They were excited to have access to real hardware to synthesize their controllers through MATLAB/Simulink programs, validate the controller on a Simulink model, run the dSPACE DS 1104 DSP-board experiment, download data, and analyze the control system performance offline without being distracted by software implementation issues. This environment allowed for extensive experimentation, performance comparison, and development of several practical control algorithms. It is expected that the techniques employed in the controller designed for the laboratory experiment will likely be used by the students in their subsequent employment after completion of their college education.

References

1. H. Ashrafiuon and D. S. Bernstein, “Innovations in undergraduate education: Part II,” "IEEE Control Syst. Mag., vol. 25, no. 1, pp. 21-22, Feb. 2005.
2. L. D. Feisel and A. J. Rosa, “The role of the laboratory in undergraduate engineering education,” Journal of Engineering Education, vol. 94, no. 1, pp. 121-130, January 2005.
3. R. Molengraft, ML Steinbuch, and B. Karker, "Integrating Experimentation into Control Courses, "IEEE Control Syst. Mag., vol. 25, no. 1, pp. 40-44, Feb. 2005.
4. K. Arzen, A. Blomdell, and B. Wittenmark, "Laboratories and Real-Time Computing," IEEE Control Syst. Mag., vol. 25, no. 1, pp. 30-34, Feb. 2005.
5. D. Varsakelis and W. Levine," An Undergraduate Laboratory for Networked Digital Control Systems," IEEE Control Syst. Mag., vol. 25, no. 1, pp. 60-62, Feb. 2005.
6. A. G. Alleyne, D. J. Block, S. P. Meyn, W. R. Perkins, and M. W. Spong, “An interdisciplinary, interdepartmental control systems laboratory, IEEE Control Syst. Mag., vol. 25, no. 1, pp. 50-55, Feb. 2005.
7. A. Rubaai, J. H. Johnson, and D. Cobbinah," The New Motors and Controls Laboratory at Howard University," Proceedings of the 2005 ASEE for Engineering Education Annual Conference and Exposition, June 12-15, 2005.
8. Y. Chen and J. Naughton, "An Undergraduate Laboratory Platform for Control System Design, Simulation, and Implementation," IEEE Control Syst. Mag., vol. 20, no. 3, pp. 12-20, June. 2000.
9. P. S. Shiakolas and D. Piyabongkarn, “Development of a real-time digital control system with a hardware-in-the-loop magnetic levitation device for reinforcement of controls education,” IEEE Trans. Education, vol. 46, no. 1, pp. 79-87, Feb. 2003
10. L. Güvenç and B. A. Güvenç, “Design projects on automotive controls,” IEEE Control Syst. Mag., vol. 24, no. 5, pp. 92-94, Oct. 2004.
11. J. Apkarian and K. J. Astrom, “A laptop servo for control education,” IEEE Control Syst. Mag., vol. 24, no. 5, pp. 70-73, Oct. 2004
12. S. Bex, S. Doclo, G. Ysebaert, G. Gielen, W. Dehaene, H. De Man, and B. De Moor, “The peopleMover educational project ,” IEEE Control Syst. Mag., vol. 24, no. 5, pp. 70-73, Oct. 2004
13. MATLAB/Simulink User’s Guide, The MathWorks Inc., Natick, MA, 1998
14. SPACE User’s Guide, Digital Signal Processing and Control Engineering, dSPACE, Paderborn, Germany, 2003.
15. G413-817 Technical Data Manual, Moog Aerospace, East Aurora, New York, 200
16. T200-410 Technical Data Manual, Moog Aerospace, East Aurora, New York, 2000