2006-391: DSP-BASED REAL-TIME CONTROL SYSTEMS DESIGN, ANALYSIS, AND IMPLEMENTATION FOR REINFORCEMENT OF CONTROLS EDUCATION

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 the same year, he joined Howard University, Washington, D.C., as a faculty member, where he is presently a Professor of Electrical Engineering. His research interests include high performance motor drives, research and development of intelligent applications for manufacturing systems, engineering hardware testing in laboratory, and computer-aided design for undergraduate engineering education. Dr. Rubaai is a Senior Member of IEEE, and past Chair of the Industrial Automation and Control Committee of the Industry Applications Society (IAS). Currently, Dr. Rubaai is the Chair of the IEEE/IAS Manufacturing Systems Development and Applications Department, and a member of the IAS Executive Board. Dr. Rubaai can be contacted at arubaai@howard.edu.

James Johnson, Howard University

James H. Johnson, Jr. received his B.S. from Howard University, M.S. from the University of Illinois and Ph.D. from the University of Delaware. Currently, he is a professor of civil engineering and Dean of the College of Engineering, Architecture and Computer Sciences at Howard University. Prior to this appointment, he was the chair of the Department of Civil Engineering and interim associate vice president for Research for Howard University. Dr. Johnson's research interests include the treatment and disposal of hazardous substances, the evaluation of environmental policy issues in relation to minorities, the development of environmental curricula and strategies to increase the pool of underrepresented groups in the science, technology, engineering and math disciplines.

DSP-Based Real-Time Control Systems Design, Analysis, and Implementation for Reinforcement of Controls Education

Abstract

This article describes the development, implementation, and demonstration of a real-time laboratory platform for use in an undergraduate laboratory control course. The design, analysis, and implementation stages are carried out entirely using a dSPACE DS1104 digital signal processor (DSP)-based real-time data acquisition control (DAC) system and MATLAB/Simulink software tools. The control law is designed in Simulink and executed in real-time using the dSPACE DS 1104 DSP-board. Once the controller has been built in Simulink, the MATLAB Real-Time Workshop (RTW) routine is used to automatically convert the Simulink block-set to a machine code that runs on the DS1104 DSP processor. While an experiment is running, the dSPACE DS1104 provides a mechanism that allows the student to change controller parameters online. Thus, it is possible for students to view the real process while the experiment is in progress. Students are encouraged to compare their designs with those of other students. Students are enthusiastic about the lab; and, their comments are positive.

Introduction

Recent observations confirm the escalating concern of the general engineering educational community about enhancing the teaching of traditional control theory courses. Simulation tools are frequently used as an educational aid in automatic control courses. While simulations are an important component for teaching students about general system behavior, they cannot always account for all the details that must be considered in designing and analyzing a physical system in an interdisciplinary, team-oriented environment. The need to control real hardware, and not just simulations, is known to all who design and build real control systems. Experimentation is widely accepted as an important part of control-system education[1]. Arzen, et al.[2] describe the development of a control laboratory using real, rather than simulated processes. Their article presents the approach that the Department of Automatic Control at Lund Institute of Technology uses to maintain a high level of practical laboratory experiments. They claim that real-time systems help the students understand the theoretical material and they also help to increase the student's motivation to learn. A new senior-level laboratory for networked digital control systems is proposed by Varsakelis and Levine[3]. This laboratory attempts to introduce students to the complexities of modern control systems, which are often ignored during early control education. A low-cost laboratory module for control systems design using either MATLAB's intuitive programming language or in Simulink's block diagrams is reported in[4]. The laboratory module is based on a simple second-order dc servo motor that allows student to perform a wide range of laboratory experiments. A real time controlled laboratory plant for control education is proposed by Saco, et al.[5]. The proposed approach supports automatic code generation using Simulink, MATLAB RTW and dSPACE. However, the problem of testing before final implementation is not discussed in any detail.

In this article, a laboratory platform for performing control systems design, analysis, and implementation is described. This real-time environment is implemented and demonstrated by controlling a nonlinear time-varying permanent magnet synchronous servomotor drive system. The student can design his/her controller simply by drawing its block diagram using the graphical interface of Simulink. After the block diagram has been completed, the student can validate his/her design simply by the click of a mouse button. The dSPACE DS 1104 DSP-board is interfaced with the MATLAB Real-Time Workshop (RTW) to automatically generate optimized C code for real-time applications. After the board has been started, Simulink can be linked to the dSPACE DS 1104 using the external mode feature of Simulink. In this way, controller parameters can be tuned from Simulink, and control variables can be traced using Simulink scope blocks. Real time experimental results can be observed through online plots. Students are motivated to see that the real-life performance of the apparatus can be controlled and understood.

The laboratory platform is implemented for a senior-level undergraduate control laboratory course that emphasizes hardware/software design for embedded controlled systems. The laboratory course brings together undergraduates from two engineering departments (electrical and mechanical) with plans to include systems and computer science students in the near future.

Laboratory Hardware/Software Interface

To concentrate fully on the actual control design task and allow students to gain experience with industrial control development tools, a dSPACE DS1104 DSP board[6] is chosen as the main interface between the controlled system (process) and the host computer. One of the salient features of the dSPACE DS 1104 DSP-board is the ease of building real-time applications. In order for students to access the I/O dSPACE DS 1104 DSP-board, a software interface to the board is required. Here, the student has two options: 1) the student can choose to write C-language code and interrupt-service routines, or 2) the student can chose to use a MATLAB/Simulink interface[7]. Simulink is a block-diagram graphical-user-interface based simulation package that works within the MATLAB environment and allows linear and nonlinear, continuous-time and discrete-time simulation. In addition, students can choose from a broad range of control design toolboxes. The Control-Systems Toolbox, in particular, greatly aids control-system analysis and design.

The dSPACE DS1104 has software tools which help the student manage the laboratory experiments. For example, the student can display or store variables and change parameters with ControlDesk panel. The software associated with the dSPACE DS 1104 DSP-board consists of the following: 1) MATLAB/Simulink, a Window based analysis environment from the MathWorks, and 2) a Real Time Workshop (RTW) Window interface between the MATLAB/Simulink environment and the dSPACE DS1104 DSP-board. This seamless interface makes it possible to download and generate C-code algorithms on the dSPACE DS1104 from the Simulink blocks within the MATLAB environment. Control simulation and implementation are performed using the same generated C-code blocks. This interface provides a two-way pipe between the MATLAB environment and the dSPACE DS1104, allowing the student to manipulate the real-time DAC system using MATLAB's full analytical capabilities.

Laboratory Platform for Experimental Teaching

The undergraduate control systems laboratory in the Department of Electrical and Computer Engineering at Howard University has currently four workstations. Each laboratory station has a host computer with resident dSPACE hardware and software. The experimental assembly described here provides a platform for teaching several fundamental concepts in controls and embedded computing. Fig. 1 illustrates a block diagram of the hardware configuration. It consists of a three-phase motor, a driving circuit, a PM DC generator as a dynamic load, a torque transducer, a variable transformer, a power supply, a variable resistive load, a dSPACE DS1104 DSP-board, an oscilloscope, a function generator, and a personal computer (PC) with Windows NT 4.0. The motor is 1-hp 3000 rev/min three-phase permanent magnet synchronous servomotor, which was manufactured by Moog Aerospace[8]. It is equipped with a resolver and is coupled via a torque transducer. The driving circuit is also a Moog T200-410 designed for high performance motor drives[9]. 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 dSPACE DS-1104 DSP board is installed in a Pentium III 500-MHz PC.

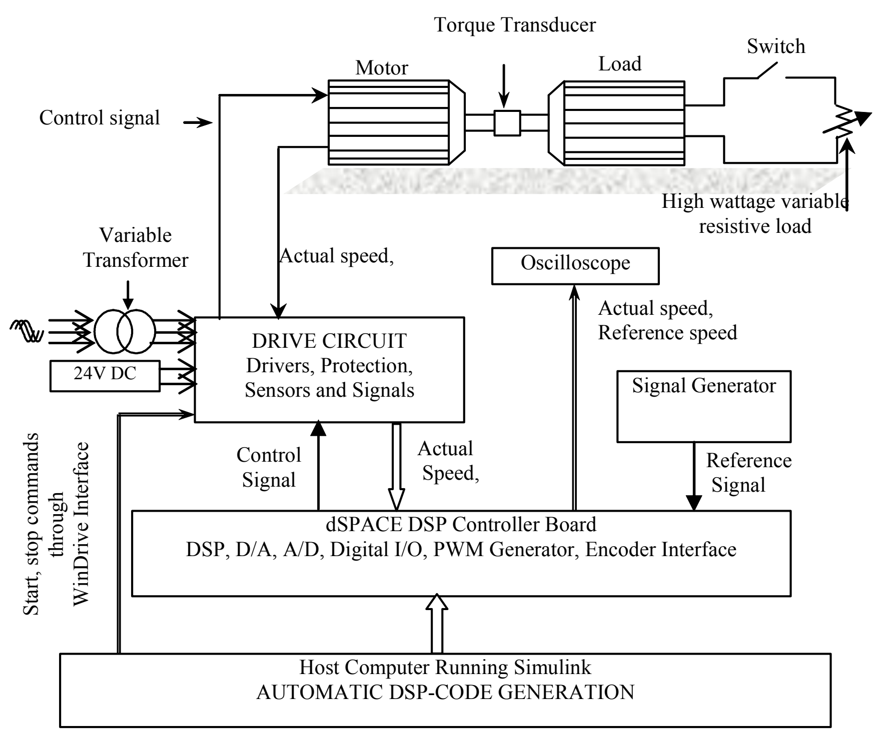

Fig. 1 Block diagram of the hardware configuration

The control algorithm is built within Simulink environment combined with the Real-Time Interface (RTI) provided by dSPACE and is implemented by the main processor of the DS-1104 board in real-time. Safety sockets have been used so that students can connect to these wires with modern "banana" plugs to measure speed, position, and current without the risk of contacting exposed conductors. It is interesting to note that in the experimental setup, the motor is coupled with a PM DC Generator. However, the test setup does not have a proper dynamometer facility to load the DC generator with some well defined load. Therefore, to test the load characteristics of the controller, a variable resistive load is connected to the terminals of the DC generator. To achieve sudden disturbances, the switches of the resistive load are turned on and off. Much of the wiring has been linked so that students can see how the components are interconnected. A typical lab workbench is shown in Fig. 2. As a result of an equipment gift, the developed workstation utilizes Moog and dSPACE components almost exclusively. If the workstations were to be duplicated, equipment from a variety of vendors could be used with similar functionality.

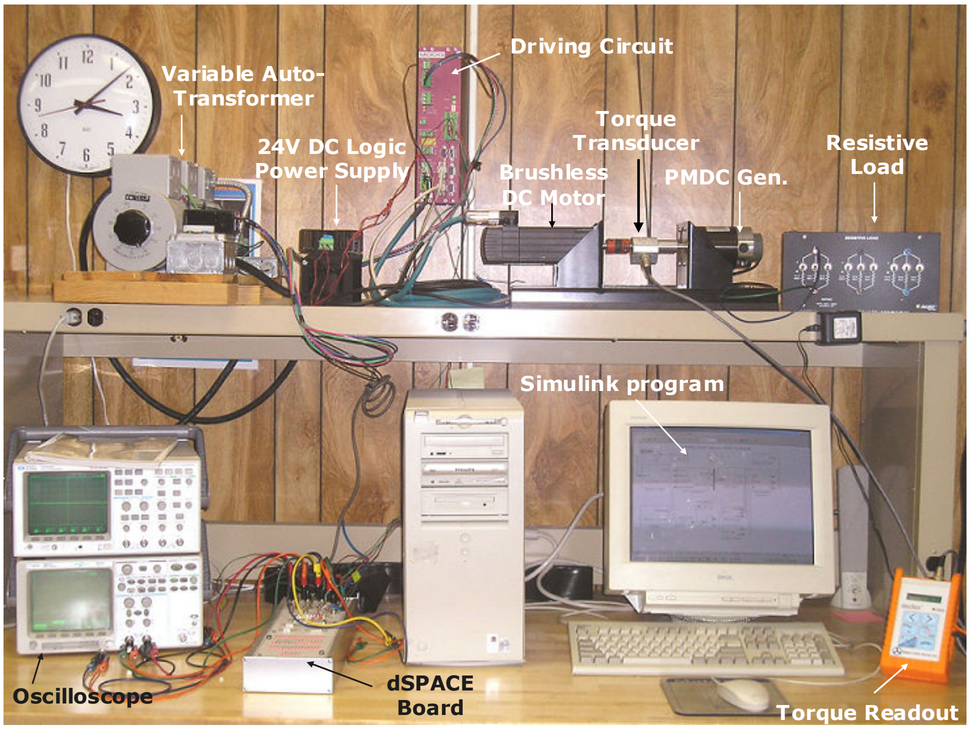

Fig. 2 Typical lab bench in the control systems laboratory at Howard University

Control Laboratory Course

The new undergraduate control systems laboratory at Howard University originated in summer 2004 with funds from Moog Aerospace. The primary goal is to provide a platform for hands-on demonstrations and projects in control courses that require lab facilities. Laboratory experiments are integrated with previously taken lectures in control courses. The laboratory course consists of lecture material and hands-on laboratory experiments. The laboratory course has six mandatory four-hour laboratory exercises offered every spring semester. The laboratory is in use 12 hours per week, with three four-hour laboratory sessions per week. Students normally work in teams of two per workstation

Each lab exercise has a prelab preparation segment in which the students build their controllers in Simulink, and prepare real-time models that are subsequently used during the experiment. This method ensures that students are ready to carry out the experiment. The prelab activity is followed by an in-lab activity where students perform the experiments in the lab. In the first two laboratory experiments students are taught to become familiar with MATLAB/Simulink by writing their own script files and developing their own block diagrams. We believe that eminent software environments, such as MATLAB/Simulink, are advantageous since students do not desire to learn control languages that are tailored for a particular laboratory. One teaching assistant, normally a Ph.D. student, supervises each laboratory session. During this period the teaching assistant and instructor walk through the room answering questions, addressing problems, and supervising students performing dSPACE-based MATLAB/Simulink real-time experiments.

Running the Experiments

Students can demonstrate many control concepts using the laboratory platform. After choosing which experiment to run, students can design their controllers in Simulink blocks. Once the controllers have been built, the MATLAB Real-Time Workshop (RTW) routine is used to automatically generate optimized C code for real-time application. When using RTW with the dSPACE DS 1104 DSP-board, target specific software is developed using dSPACE DS1104 Real-Time Interface (RTI) to Simulink. MATLAB RTW and dSPACE RTI provide an integral environment that automates the task for building an executable file that is downloaded onto the dSPACE processor. Subsequently, the interface between Simulink and the dSPACE DS 1104 allows the control algorithm to be run on the hardware of the dSPACE DS1104 which is an MPC8240 processor. Using these interfacing routines, students can implement and test their design simply by the click of a mouse button. While the experiments are in progress, the student can change controller parameters, reference signals, and observe results without having to rebuild and download a new Simulink model to the dSPACE DS1104. When the student stops the experiment, it is possible to download a file in a MATLAB format where the experimental results have been stored. This file can be used to perform offline analysis, such as evaluation of the maximum overshoot, steady state error, and settling time.

The following experiments were successfully performed using the laboratory platform under MATLAB/Simulink and dSPACE DS1104 environment:

- Learning How to use MATLAB/Simulink to Build a Model.
- Building Real-Time Models in Simulink. Learning about dSPACE DS1104 DSP. Familiarizing students with DS1104 programming tool, ControlDesk, and how to build layouts; and, compare simulation and real-time results.
- Studying the proportional-integral (PI) control algorithm and the effects of non-ideal motor characteristics on the closed loop performance of the controller.
- Studying the dual loop (outer proportional position loop with inner proportional-integral speed loop) control algorithm and tuning.

These exercises provide insight into the hardware aspects of the controller.

Implementation Results

The proportional-integral (PI) controller is perhaps the most commonly used controller in the industry, and students should be familiar with its practicality. With its two-term functionality covering treatment to both transient and steady state responses, PI controller offers the simplest and yet most effective solution to many real-world control problem. Students are asked to look into the qualitative properties of proportional and integral action to develop an intuitive feel for the PI controller. Controller parameters are tuned such that the closed-loop control system will be stable and will meet given specifications associated with the following:

- Stability robustness.
- Set-point following and tracking performance at transient, including rise-time, overshoot; and, settling time.
- Regulation performance at steady-state, including load disturbance rejection.
- Robustness against environmental uncertainty.

The student is asked to design a proportional-integral controller to specifications and analyze its responses to step inputs, square-wave inputs, triangular inputs, and sinusoidal inputs. The controller is built in Simulink and implemented in the dSPACE DS1104 DSP. The Simulink model implementing the PI controller is shown in Fig. 3. The coefficients of the proportional and integral actions can be tuned while the experiment is running. The response plots and the tracking performance of the controller under dynamic load using these reference inputs are displayed in Figs. 4 through 7. The student assesses the effects of the two gains on system performance.

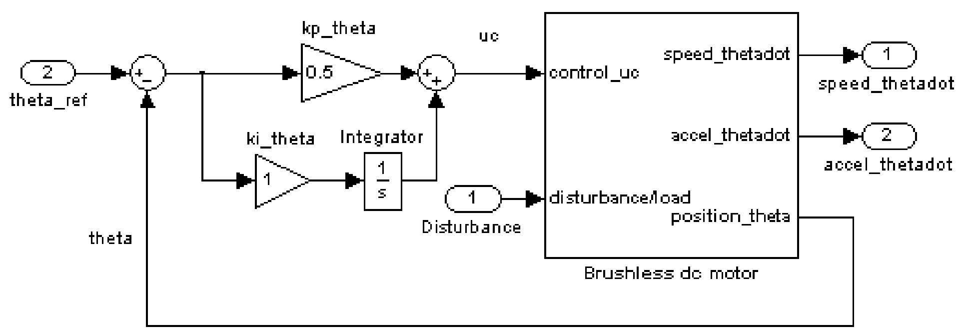

Fig. 3 Simulink model implementing the PI controller where all coefficients are tunable online

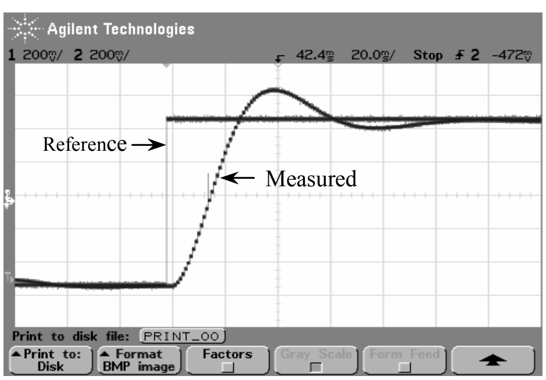

Fig. 4 Step input position tracking, X=20 ms/div, Y=2 rev/div

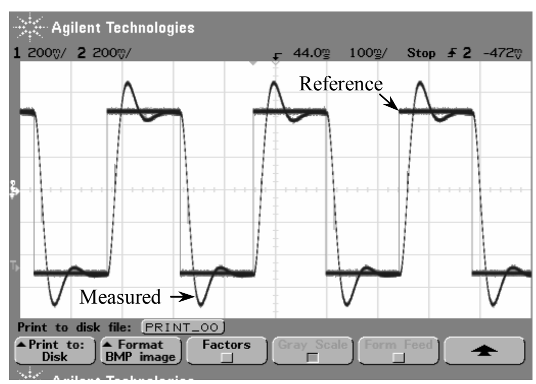

Fig. 5 Square-wave position tracking, X=100 ms/div, Y=2 rev/div

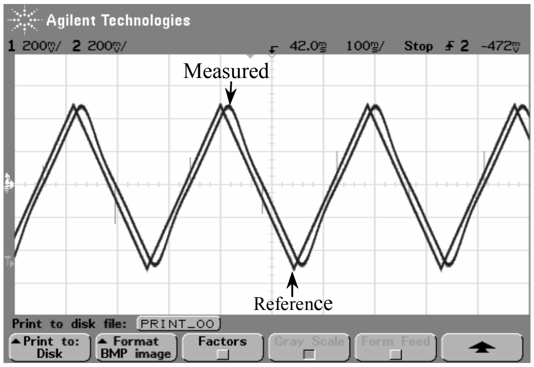

Fig. 6 Triangular-wave position tracking, X=100 ms/div, Y=2 rev/div

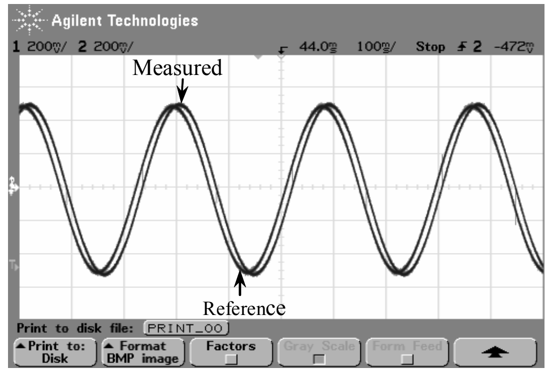

Fig. 7 Sine-wave position tracking, X=100 ms/div, Y=2 rev/div

Figures 4 through 7 illustrate comparisons between the actual measurements and the corresponding reference inputs for several angular positions. Clearly, the PI controller exhibits some overshoot as the measured angular position approaches the desired reference input. The steady-sate error indicates a need for increased integral gain in the control loop. Students also observed that with a large enough value of the integral gain the closed loop response may become unstable (Figs. 8 and 9).

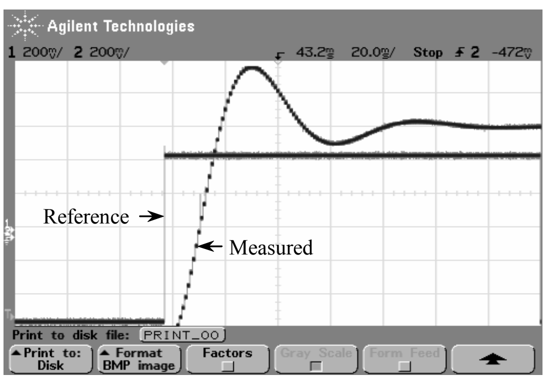

Fig. 8 Step-input position tracking, X=20 ms/div, Y=2 rev/div

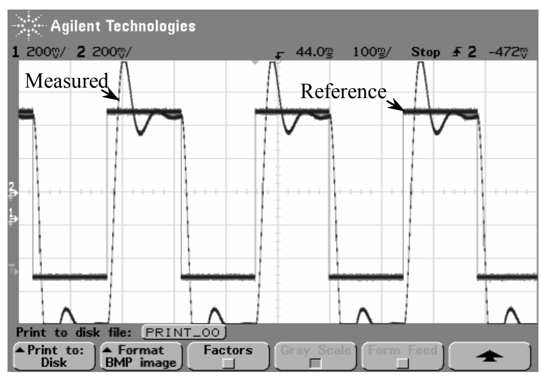

Fig. 9 Square-wave position tracking, X=100 ms/div, Y=2 rev/div

Following the design and implementation of the PI controller, the student is introduced to the dual loop controller (proportional position with an inner proportional-integral speed loop controller).This controller represents a modification of the PI controller. The dual loop controller is employed frequently in servo position loops found on numerically controlled machines and has been chosen for this reason. The Simulink model implementing the dual loop controller is shown in Fig. 10. The students, with no difficulty, modified the values of the controller's three parameters-the proportional gain in the outer loop, and the proportional and integral gains for the inner speed loop in the Simulink diagram. These are modified without recompiling the control code and their effects are observed in real-time on the scope. Figures 11 through 14 illustrate the performance of the dual loop controller under loading conditions for step inputs, square-wave inputs, triangular inputs, and sinusoidal inputs, respectively. The students observed that, in every case, the dual loop controller brings the measured angular position to the desired value smoothly and with a slight overshoot as apposed to Figs. 4 and 5 of the PI controller. It is also observed that the response under dual loop controller is somewhat slower than that the response under PI controller. The discussion by the students can address these issues with reference to the performance specifications.

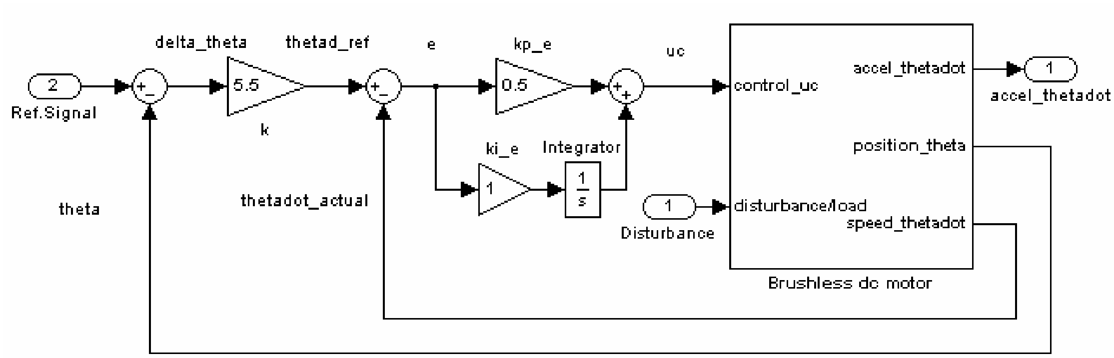

Fig. 10 Simulink model implementing the dual loop controller where all gains are tunable online

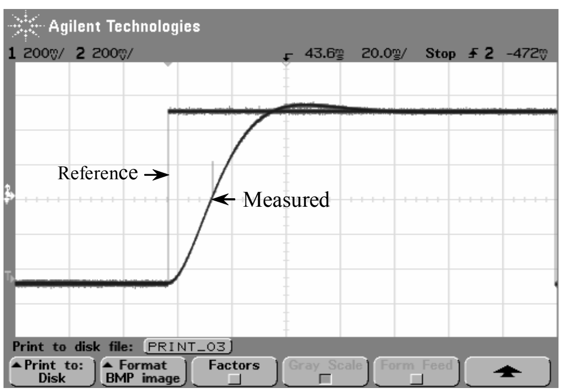

Fig. 11 Step-input position tracking,
X=20 ms/div, Y=2 rev/div

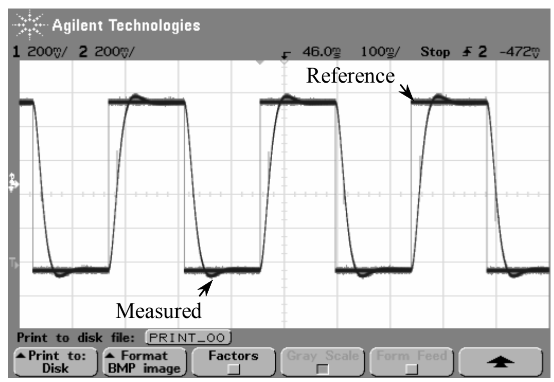

Fig. 12 Square-wave position tracking,
X=100 ms/div, Y=2 rev/div

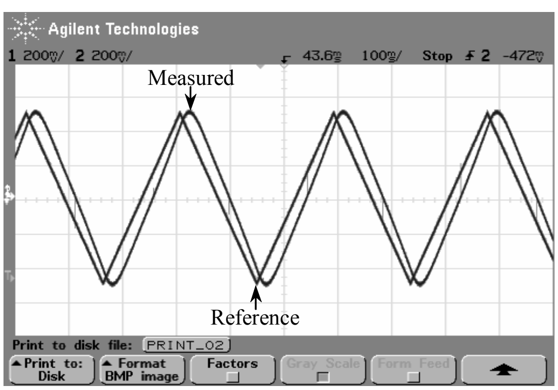

Fig. 13 Triangular-wave position tracking,
X=100 ms/div, Y=2 rev/div

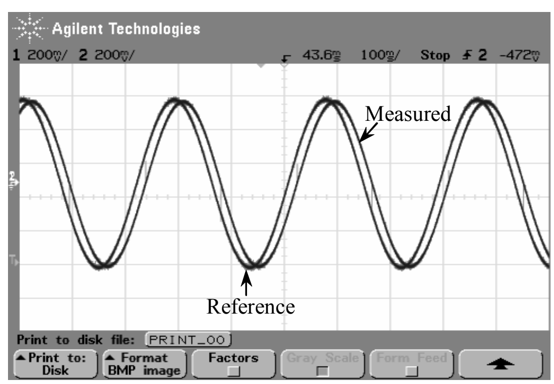

Fig. 14 Sine-wave position tracking,
X=100 ms/div, Y=2 rev/div

The disturbance rejection aspect of closed-loop control is also introduced to students. The students are instructed to turn alternately to the on and off positions of the switches of the resistive load. Figures 15 and 16 illustrate the effects of the external disturbance on the performance of the controller. 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.

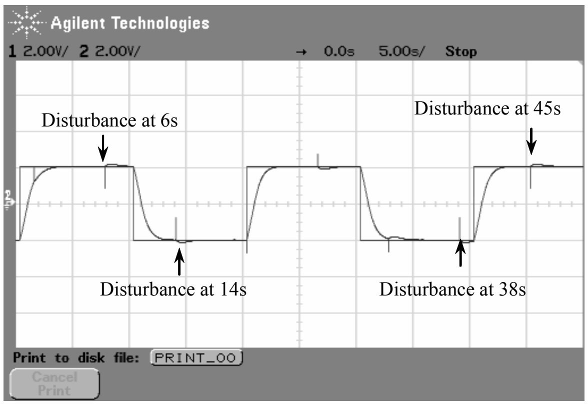

Fig. 15 Square-wave position tracking, X=5 s/div, Y=10 rev/div

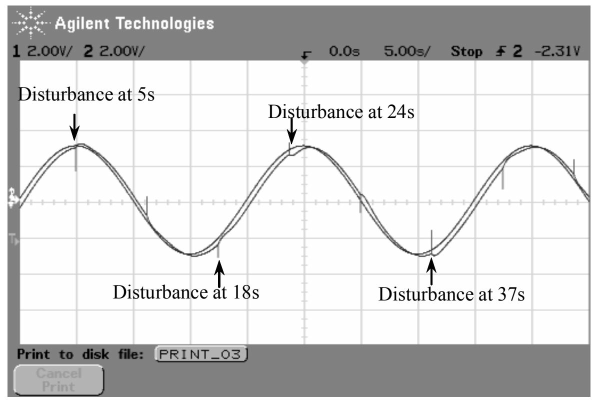

Fig. 16 Sine-wave position tracking, X=5 s/div, Y=10 rev/div

Clearly, the students manage to reduce both the overshoot and the extent of oscillations conditions by manipulating the control gains. The students also concluded that the dual loop controller offers substantial performance improvement over the PI controller.

Laboratory Features

- *Simulink-Based Interface*

The Simulink-based interface is used to design controllers that drive the real process. Only a basic knowledge of the MATLAB/Simulink environment is required. Students design, analyze, and verify control strategies without software installation, system configuration or programming. Students can focus on the controller design and obtain the results interactively without need for any special programming or debugging.

- *Easy-to-use Interface*

Simplicity is vital for realizing an interface that is easy to use. One of the features of the dSPACE DS1104 DSP board is the ease of building real-time applications. The DS1104 DSP equipped with its own graphical-user-interface (ControlDesk) allows the drag-and-drop reconfiguration of the user interface and offers the required level of interaction with the process. Compiling and downloading the object code into the DS1104 DSP board is automated and totally transparent to the students.

- *Reference Signals*

Students can choose reference signals from a given list or create new reference signals by building a Simulink block. It is possible to change the reference signal while an experiment is running. Thus, the student does not have to start a new experiment to verify the response of the system to different input signals.

- *Controller Types*

Students choose a predefined controller or synthesize a new controller through the MATLAB/Simulink environment. In the first case, a student can select a proportional and integral (PI) controller to run the experiment and choose the values of its coefficients (gains). On the other hand, students can synthesize a new controller using Simulink graphical interface.

Assessment/Evaluation

The laboratory control course provides a unique opportunity to achieve department-wide integration of practical training in the curriculum. In particular, our educational control laboratory covers key points of the ABET requirements pertaining to curriculum content, adequate laboratory experience, engineering practice, and written and oral communications. The laboratory sessions are adaptable and enable a large group of undergraduate students to carry out variety of laboratory experiments. Using detailed student questionnaires, the control laboratory course has been evaluated over the past year. The students are enthusiastic about the laboratory sessions with practical experiments. They believe that control laboratory experiments help them to learn the material from lectures, which satisfies the first part of our educational goal. Not only have students developed better experimental skills, they also foster an understanding about the design, implementation, and testing of different control algorithms. The use of the laboratory experiments has generated many positive results. The student reaction to the experiments has been very good and interest has been increased. The students seem to appreciate the "feel" that they gain from the laboratory course. In order to realize the educational objectives of the laboratory exercises, students are encouraged to compare the design of their controllers with those of other students. The students are enthusiastic about the lab, and their comments are positive.

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. 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. The combination of dSPACE DS1104 DSP and MATLAB/Simulink/RTW effectively created a rapid control prototype environment in which students focused on control design rather than programming details or debugging control languages. In this way, the lab offers an unparalleled experience and is a great source of attracting students and exciting their interest.

References

1. R. Molengraft, M. Steinbuch, and B. Karker, "Integrating Experimentation into Control Courses," IEEE Control Syst. Mag., vol. 25, no. 1, pp. 40-44, Feb. 2005.
2. 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.
3. 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.
4. 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.
5. R. Saco, E. Pires, and C. Godfrid, "Real Time Controlled Laboratory Plant for Control Education," 32nd ASEE/IEEE Frontiers in Education Conference, pp. 12-16, Nov. 6-9, 2002, Boston, Ma
6. dSPACE User's Guide, Digital Signal Processing and Control Engineering, dSPACE, Paderborn, Germany, 2003.
7. MATLAB/Simulink User's Guide, The MathWorks Inc., Natick, MA, 1998
8. G413-817 Technical Data Manual, Moog Aerospace, East Aurora, New York, 2000
9. T200-410 Technical Data Manual, Moog Aerospace, East Aurora, New York, 2000