

## **AC 2007-2991: DEVELOPING A MATLAB/SIMULINK RTWT BASED HYDRAULIC SERVO CONTROL DESIGN EXPERIMENT**

### **Charles Birdsong, California Polytechnic State University**

Charles Birdsong has expertise in vibrations, controls, signal processing, instrumentation, real-time control, active noise control, and dynamic system modeling. He received his BSME at Cal Poly San Luis Obispo, MS and Ph.D. at Michigan State University where he worked on active noise control applications for the automotive industry. He has worked in the vibration test and measurement industry helping to drive new technologies to market and working with industry to meet their emerging needs. He is currently an Assistant Professor at California Polytechnic State University at San Luis Obispo in the Department of Mechanical Engineering teaching dynamics, vibrations and controls and is involved in several undergraduate and master's level multidisciplinary projects.

# Developing a MATLAB/Simulink RTWT Based Hydraulic Servo Control Design Experiment

## Abstract

While one of the stated goals of the Mechanical Engineering Controls course is to develop the tools to design a controller, previous lab experiences did not include an experimental exercise in controller design. This was primarily due to the difficulty in implementing a controller that is robust and flexible enough to accommodate different student designs within the short time constraints of a three hour lab period. This paper describes an effort to use the Mathworks Real Time Windows Target to implement student controller designs on real hardware in a lab setting. Students use experimental data and a physically based model to design the controller for a hydraulic servo control system. A system transfer function is estimated from a frequency response experiment. Then a controller design is conducted using classical PID controller design techniques and a Simulink model. Finally when students have completed a controller design, they are allowed to implement it on the experimental apparatus and measure the system performance. They are then given a second chance to fine tune the model. The control is implemented with the Real Time Windows Target using a data acquisition card on a PC. A competition for the best performance also creates an exciting and competitive learning atmosphere. Assessments are presented that indicate the improvement in learning outcomes from the change in the lab exercise.

## Introduction

The Engineering College at California Polytechnic State University in San Luis Obispo has a strong tradition of hands-on, “learn by doing” education. A key component is the lab intensive courses where for example in the Mechanical Engineering Department a minimum of nine lab intensive courses are required in the core ME curriculum in addition to labs from support courses such as chemistry, physics, etc. Our philosophy is that students learn best through a combination of lecture and lab experiences and industry feedback indicates that our undergraduates are unparalleled at hitting the ground running and working with real world problems.

Mechanical Controls is a four unit, required senior level course that consists of three-one hour lectures and one-three hour lab per week for the ten week quarter. The course covers single input single output linear system modeling, time domain analysis, transfer functions, root locus, frequency response methods, PID and lead lag controllers. The lab is taken concurrently with the lecture and is designed to support the topics covered in lecture while also illustrating the realities of real world systems, modeling and controller implementation. There are currently four experiments: a two week analog DC servo position control experiment, a two week LabVIEW based two tank water level regulator experiment, a two week LabVIEW based hydraulic servo control experiment and a one week frequency response experiment using the DC servo apparatus.

An important goal is to periodically update the experiments to present hardware and software that is state of the art so that students get exposure to tools that will benefit them in their careers as engineers. In the last decade hardware has been updated from analog compensators to PLC controllers to LabVIEW based controllers and most recently to real-time rapid prototype software using MATLAB/Simulink and the Real Time Windows Target (RTWT). New lab development is a time consuming effort that must be done carefully so that it contributes to students learning and supports the theory learned in the classroom.

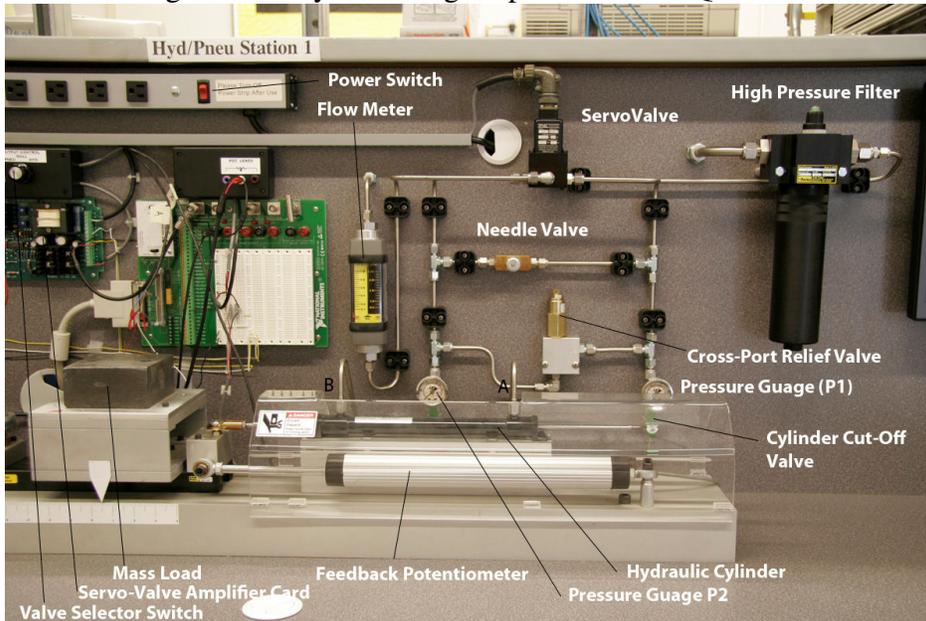
The latest development, described in this paper is the change of the hydraulic servo control system from a LabVIEW based controller to a RTWT based control system and the addition of a new controller design exercise. The previous LabVIEW system was adequate for a simple proportional controller but was unable to provide accurate integral and derivative control due to the high sampling rate required and the latencies in the Windows based PC and LabVIEW software. Furthermore modifying the software required a high level understanding of the LabVIEW programming language which is beyond the abilities of most undergraduate students in the limited time constraints of the lab session. Finally the newer versions of LabVIEW are not compatible with programs developed in earlier versions. These reasons motivated a search for a better solution.

Several solutions were considered including dedicated servo position controllers, LabVIEW or DSpace real time hardware and the MATLAB/Simulink RTWT software. The RTWT was selected because it allows the students to implement the controller using Simulink which is already an integral part of the course for system modeling and also because the incremental cost was very low compared to the other solutions<sup>1</sup>. The system requires a data acquisition card (DAQ) which was already installed for the existing LabVIEW system and additional MATLAB/Simulink software which was available at no additional cost in the current university MATLAB license. The LabVIEW or DSpace real time solutions have been used in similar lab projects<sup>2</sup> and would provide higher performance in the form of a faster sampling rate, however it was found that the RTWT system provided more than adequate performance for this application. These systems also have a relatively high cost. Dedicated industrial servo position controllers provide very good performance but do not provide enough flexibility to meet the educational needs of the lab. If new hardware and software had to be purchased to implement the RTWT system then the cost comparison might change. However as other have proposed Simulink and the RTWT system seems to meet the pedagogical needs of a controls lab more than the other solutions<sup>3</sup>.

The following sections describe the hardware and software used in this experiment, the new controller design project developed for the new system and an assessment of the new design project and the impact on learning in the lab and the lecture.

## Hardware and Software

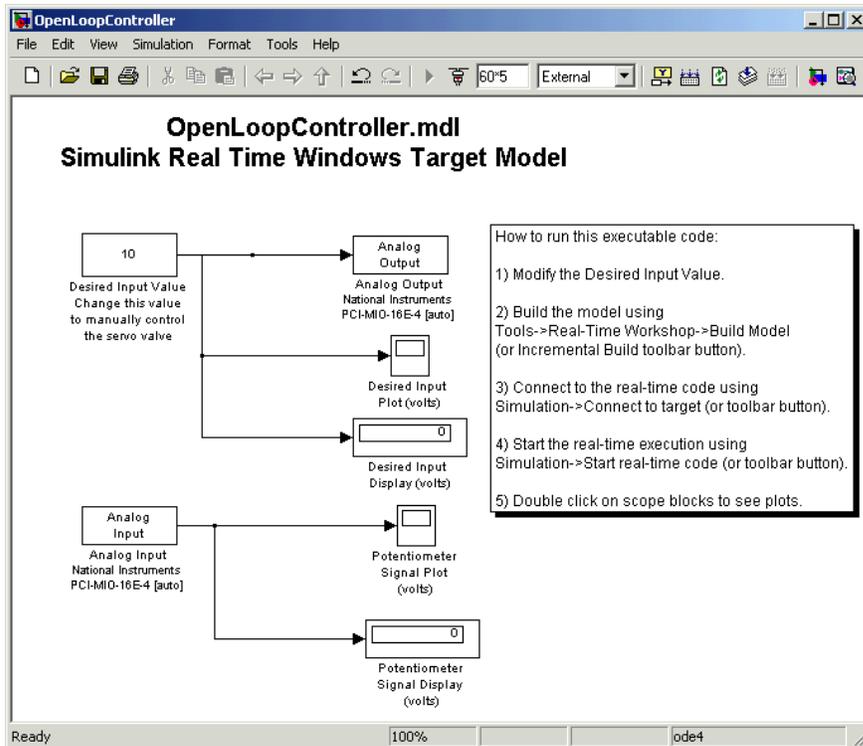
The experimental apparatus shown in Figure 1 is a hydraulic servo control system that was developed at Cal Poly through financial support from Parker Hannifin Corporation who is a sponsor of the Parker Hannifin Controls Lab. It consists of a lead mass mounted on a linear bearing and driven by a hydraulic double ended cylinder. The cylinder can be driven in both directions by a hydraulic pump and is controlled with a servo amplifier and valve. The mass position is measured with a linear feedback potentiometer. The potentiometer voltage is measured by an analog input on a National Instruments PCI MIO 16E-4 data acquisition board (DAQ). The command voltage to the servo amplifier and valve is generated by an analog output on the DAQ.



**Figure 1. Photograph of hydraulic servo control apparatus**

The DAQ communicates with the MATLAB/Simulink RTWT software<sup>4,5</sup>. The software is a special blockset within the Simulink software that allows DAQ inputs and outputs to be used as sources and sinks in a Simulink model. Before the model can be run in external real-time mode it must be compiled into C code. This process is initiated by clicking on a button and is then handled automatically by MATLAB. The student does not need to have any special programming skills beyond Simulink programming.

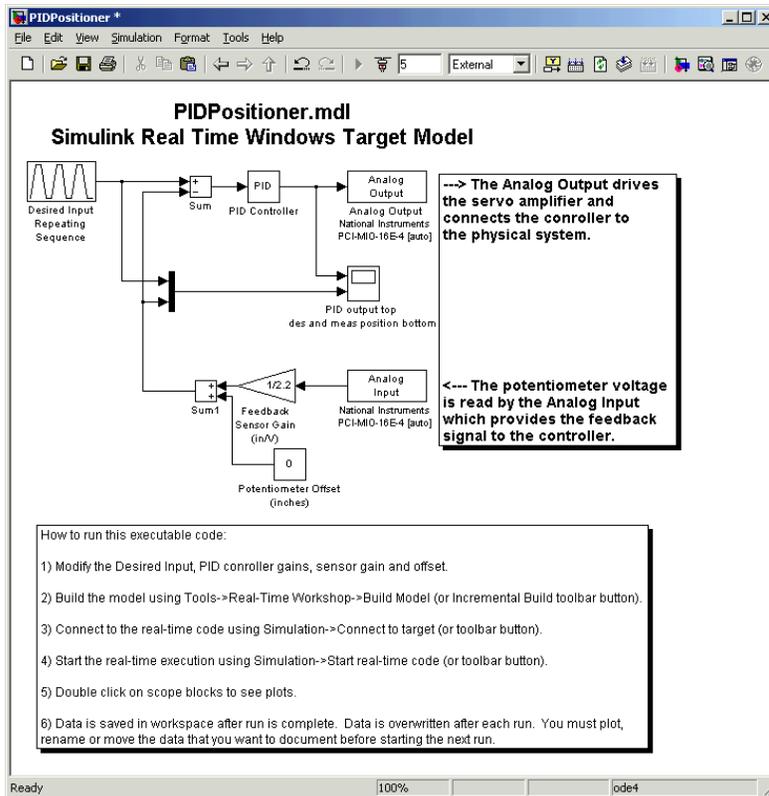
Figure 2 shows the interface for a simple open-loop controller model that measures the position sensor voltage and sets the servo amplifier/valve voltage. This interface can be used to measure the system parameters such as the servo amplifier, servo valve and potentiometer gains. When the program is run, the desired input value (servo amplifier input voltage) is written to the DAQ output and displayed on the interface and the DAQ input. The user can change the desired input on the interface. The DAQ input (position sensor voltage) is also displayed. This program operates with a loop time of 1000 cycles per second.



**Figure 2. Open-Loop Controller RTWT Interface**

Figure 3 shows the closed-loop PID controller model that is used to implement closed loop position control of the hydraulic system. This interface has the advantage that it looks like the closed-loop feedback block diagram that the students study in the lecture which is not the case for LabVIEW programs or other real time code. This helps the students understand and visualize the program very quickly.

Data can be logged and exported to the MATLAB workspace for controller evaluation. The sampling rate of the system depends on the complexity of the model and the speed of the PC processor. The PC processor is used to perform the real time calculations while the RTWT software manages the Windows operating system and guarantees the control process receives the maximum processor resources. Using a Pentium 1.9 GHz PC the PID Positioner model can be run as fast as 10 kHz however the system was limited to 1 kHz which is adequate for the servo control experiment. Faster sampling rates could probably be attained for other applications with faster PC processors.



**Figure 3. Closed-Loop PID Controller Interface**

The students use the apparatus for a two week experiment that consists of modeling the system using Simulink, characterizing the system components and then measuring the system response for various PID control designs including proportional only, proportional plus integral and full PID control. The experimental results are compared with the model results and students are asked to include nonlinear effects such as static friction to improve the model results. The students change the reference function from a step to a ramp to observe how integral control improves the steady state error.

The system model<sup>6</sup> is used and produces very good agreement with experimental results. Figure 4 shows the Simulink model and Figure 5 shows a typical comparison of model and experimental results.

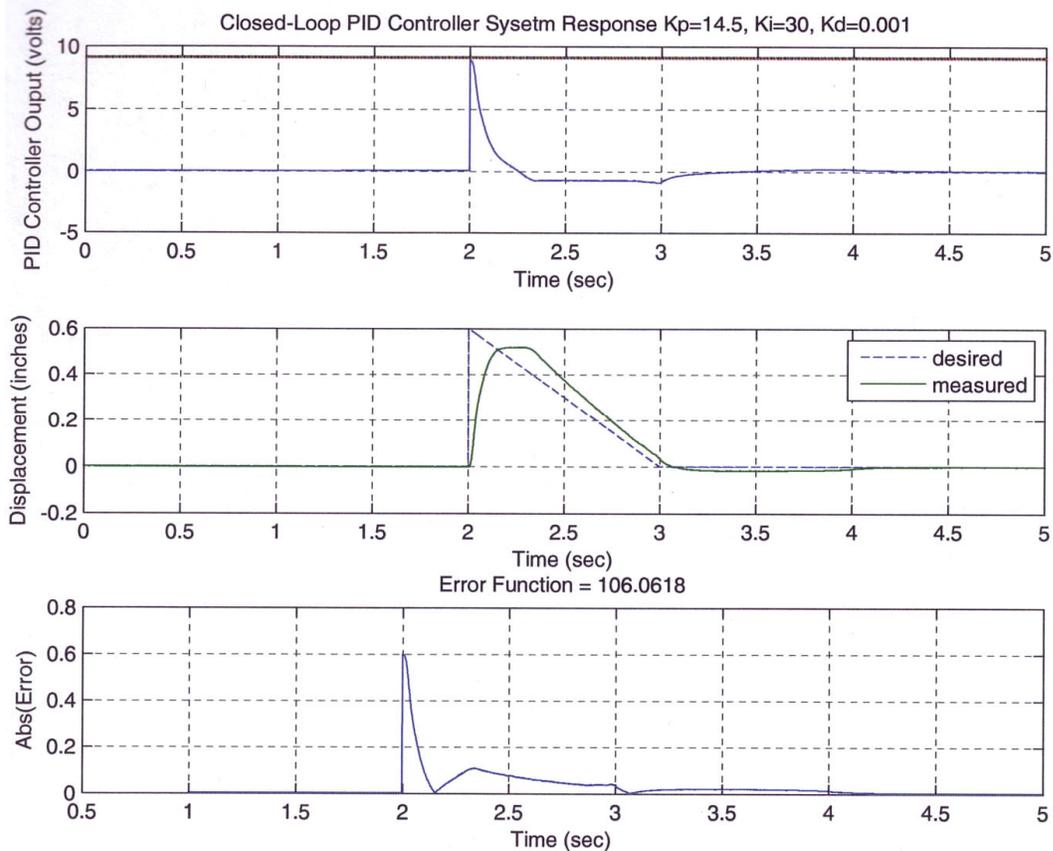


students an opportunity to get a hands-on understanding of the effects of a PID controller. Because the lab is conducted before the theory for controller design is complete, it helps motivate the concept of controller design as an improvement over the ad hoc method of tuning the controller used in the lab.

## Controller Design Project

After the control software was changed from LabVIEW to Simulink RTWT, the improved and more accurate PID controller implementation provided an opportunity to allow the students to conduct a controller design experiment. With the simple user interface the students could implement any realizable controller including noise filtering, etc. This new development was conducted and implemented in the fall of 2006. The system was modified from the conditions in the previous experiment so that same system model used in the earlier project could not be used: a spring was installed between the mass and the base and the mass was increased. Rather than perform system identification, students were shown a demonstration of an open-loop swept sine frequency response test using an LDS-Dactron Focus real-time signal analyzer and given the experimentally measured frequency response (bode) data. They were asked to use the bode data to estimate the system transfer function, and then use the transfer function to perform analytical PID controller design. To emphasize the importance of modeling and design versus trial and error controller tuning, they were then given only two chances to test their design with the real system. Students filled out a form with the controller transfer function and PID gains and submitted it to the instructor who entered the gains on the test system and ran a test. An additional design constraint was required that the command signal may not saturate the servo amplifier which occurred at greater than 10 volts. This was done to force the students to work within the linear range of the system so that the classical analytical PID control design procedures would provide good results.

A special triangle pulse reference function was used to force the students to consider several different effects. Figure 6 shows the results of a typical PID controller performance evaluation. The top graph shows the controller output in volts with a line indicating the maximum value to verify that the servo amplifier was not saturated. The middle plot shows the desired reference signal and the measured position under the closed-loop PID control. The sluggish response after 2 seconds is due to the 10 volt limit in the controller output. This constraint essentially limited the proportional gain to low values resulting in poor transient performance. This constraint also eliminated the possibility of using derivative control since any significant derivative control would saturate the amplifier due to the infinite slope at 2 seconds. The ramp from 2 to 3 seconds was included to force the students to consider adding integral control to reduce the steady state error in this region. The bottom plot shows the error function which was computed from the absolute value of the difference between the reference and measured signals and summed for each data point. The sum of the error values was indicated as the Error Function and used to measure the performance of the PID controller design. If the controller output saturated the amplifier then no error function was given and the design was deemed unacceptable. The evaluation and the figure was automatically generated after the test by implementing an m-file in the Simulink callback function.



**Figure 6. Reference function for PID controller design evaluation**

Students were asked to document their design methods and turn in a report showing the design steps and any supporting calculations used in the process. Students were encouraged to model the closed-loop response with their PID control design before trying it on the real system. This step helped most students identify that the amplifier saturation was a key limiting factor. Students were told that the report grade would be based on documentation of the design procedure and the results and that the top three designs would receive extra credit. This seemed to help motivate the students and make the project more fun.

The design project will be used in the future and has the benefit that the system can be modified to change the dynamics by modifying the mass and spring stiffness. Also different reference functions can be used to change the optimal controller design results. Hopefully this will eliminate the possibility that students will get results from previous years and get a short cut to a good design. The challenge is in formulating the design constraints in a way that allows classical linear control theory and PID controller design methods to result in good solutions while at the same time illustrating the real world effects such as saturation and other nonlinear phenomena.

## Assessment

The new PID Controller Design Project development was implemented in fall 2006 however the lab experiments for the course already comprise a full schedule. Therefore, an effort was made to assess the learning outcomes of this new development to determine if it was an improvement over the DC Servo Frequency Response Exercise that it would replace. It was felt that the DC Servo Frequency Response Exercise which consisted of a manual sine sweep frequency response measurement of the DC Servo apparatus was a relatively simple procedure that had been covered in the prerequisite vibrations course and replacing this experiment with a design project was justified. Two of the four lab sections were assigned the new PID Controller Design project (PID Design group) and two lab sections were assigned the old DC Servo Frequency Response Exercise (DC Servo group). The justification to replace the old lab with the new lab could be evaluated by comparing an assessment of these two groups. The assessment consisted of a Self Confidence Survey.

A Self Confidence Survey<sup>8</sup> was given to both groups before the final exercise. The survey asked the students to evaluate their confidence in several areas. For each question the student could answer: strongly agree, agree, somewhat agree; somewhat disagree; strongly disagree and also enter comments for each question. The theory for all areas had been covered in lecture before the survey was given. The survey questions were as follows:

1. I understand the meaning and significance of the concept of root locus and its use as a design tool.
2. I understand the concept and design methods for a PID controller.
3. I am confident that I understand how to perform a frequency response measurement on a control system and obtain an experimental Bode diagram.
4. I understand the meaning and significance of the concept of phase and gain margin.
5. The Mechanical Controls Lab helps to understand and practice the material covered in the lecture.

The same survey was also given to both groups of students a week after the final exercise to measure the change in confidence levels as a result of the lab experience. The new lab development was then assessed by comparing the change in confidence levels on the DC Servo group with the PID Design group. The results are summarized in Tables 1 and 2. Table 1 shows the average and standard deviation of all students in the PID Design Group before and after the PID Design Exercise. Responses were graded strongly agree = 5, agree = 4, somewhat agree = 3, somewhat disagree = 2 and strongly disagree = 1. Table 2 shows the same results for the DC Servo Group.

		Table 1. PID Group Comparison				
		Q1	Q2	Q3	Q4	Q5
Before	Average	4.074	3.741	2.963	3.259	3.704
	Std Dev	0.917	0.944	0.980	0.903	1.103
After	Average	4.067	3.700	3.233	3.500	3.767
	Std Dev	0.740	0.877	0.935	0.900	0.774
Change		-0.01	-0.04	0.27	0.24	0.06

		Table 2. DC Servo Group Comparison				
		Q1	Q2	Q3	Q4	Q5
Before	Average	3.957	3.739	3.136	3.348	4.174
	Std Dev	0.825	0.619	0.990	0.935	0.778
After	Average	3.542	3.333	3.625	3.333	3.708
	Std Dev	1.062	0.917	1.377	0.917	1.122
Change		-0.42	-0.41	0.49	-0.01	-0.47

The results are somewhat surprising. The confidence of the PID Design Group seemed marginally unchanged in Q1, Q2 and Q5 and somewhat improved in Q3 and Q4. This was disappointing as it was expected that the confidence level would be improved in the questions relating to PID design (Q1 and Q2). However, when these results are compared to the DC Servo Group there is some clarity. The DC Servo Group change is significantly negative in Q1, Q2 and Q5, significantly improved in Q3 and negligible change in Q4. The improvement in Q3 was expected because the DC Servo Group performed a manual sine sweep frequency response measurement while the PID Design Group only observed a demonstration.

These results do show that the new lab development was an improvement; however the results seem to be biased. A possible explanation for the bias is that the second survey was administered a week after the lab experience and after the students received their graded midterms which tested them for PID design theory. It seems that before they received the midterms they overestimated their confidence in their skills and afterwards downgraded their confidence evaluation. The data shows that the PID Design Group had a higher confidence in PID design concepts compared with the other group. A possible remedy to this bias is to give the survey immediately before and after the lab experience to eliminate biasing of other factors.

Student comments regarding the PID design project related to the fact that the design was more constrained by the actuation limit constraint than by the position response characteristics. Some students mentioned that there seemed to be small changes in the results with large changes in the controller design. These comments point out the difficulty in balancing the real world aspects of controller design with the objective of reinforcing the linear control theory.

## Conclusions

The MATLAB/Simulink RTWT software was introduced into the Mechanical Engineering Controls lab at Cal Poly and applied in the hydraulic servo control experiment replacing the previous LabVIEW control software. The software is easier to program and easier for the students to understand in the context of linear control theory. This lab development resulted in more accurate control and the ability to demonstrate the effects of PID control. In addition the ability to modify the controller with very little

effort enabled a PID design project to be added to the curriculum. Assessment shows that the new project improved students self confidence in the concepts of controller design.

It should be noted that LabVIEW continues to be used successfully in other experiments that require lower sampling rates (below 100 Hz) and with applications that do not need to be modified significantly by the students. It is to the student's benefit to see more than one data measuring and control software to expand their breadth of experiences so that they can find the best solution for specific industrial applications in their future careers.

## References

---

- <sup>1</sup> R. Throne, "*Developing Models and an Understanding of Their Limitations*", ASEE 2006 Conference
- <sup>2</sup> A. Rubaai and J. Johnson, "*DSP-Based Real-Time Control Systems Design, Analysis, and Implementation for Reinforcement of Controls Education*," ASEE Conference 2006.
- <sup>3</sup> A. Mohammadzadeh, S. Haidar, "*Analysis and Design of Vehicle Suspension System Using MATLAB and Simulink*", ASEE Conference 2006.
- <sup>4</sup> MathWorks Inc., *MATLAB, version 7*, Natick, MA: Math Works, Inc., 2005.
- <sup>5</sup> Math Works Inc., "*Real-time Windows Target User's Guide Version 2*," Natick, MA: MathWorks, Inc., 2004.
- <sup>6</sup> H. Merrit, "*Hydraulic Control Systems*", John Wiley and Sons, 1969,
- <sup>7</sup> N. Nise, "*Control Systems Engineering*," Wiley & Sons 2004
- <sup>8</sup> T. Angelo and K. Cross, "*Classroom Assessment Techniques, 2<sup>nd</sup> ed.*," Jossey-Bass, San Francisco 1993