Engaging Students in Hands-on Control System Design at the University of Arizona

By Eniko T. Enikov, University of Arizona

In their senior year, approximately 120 mechanical and aerospace engineering students at the University of Arizona attend their first and only required course in controls system design. Finding a way to engage the students in interesting, meaningful control design activities while keeping costs down has been a significant challenge for our department. As engineering labs have become more sophisticated, equipment costs have risen, making it difficult to provide each student with adequate experimentation time.

Products Used

- MATLAB®
- Simulink®
- Control System Toolbox™
- Real-Time Windows Target™
- Real-Time Workshop®

We recently introduced a low-cost, portable laboratory module based on MATLAB®, Simulink®, and an aeropendulum (Figure 1). Students take the device home to conduct experiments and complete assignments. This hands-on work complements the lecture portion of the course with active, sensory learning that gets students excited about control design.

Teaching Control Design Basics

The aeropendulum is introduced in the second half of the semester; in the first half, the students learn basic control theory and how to model simple mechanical systems, such as hydraulic pistons. In the lectures, I use MATLAB to illustrate new concepts. For example, when I cover the manual construction of root locus diagrams, I use MATLAB and Control System Toolbox™. I show them a step-by-step manual derivation using a few commands. I save the command history as a script and post it to the course Web site for students to use on assignments throughout the course. Later, I show them how they can save time by using the built-in root locus capabilities of Control System Toolbox.

When they begin *Control System Design* most students are familiar with MATLAB because they have used it in their numerical methods course. Few, however, have any experience with Simulink. To help the students learn Simulink, I incrementally build Simulink models in class. I save the complete model to the Web site for the students to download.

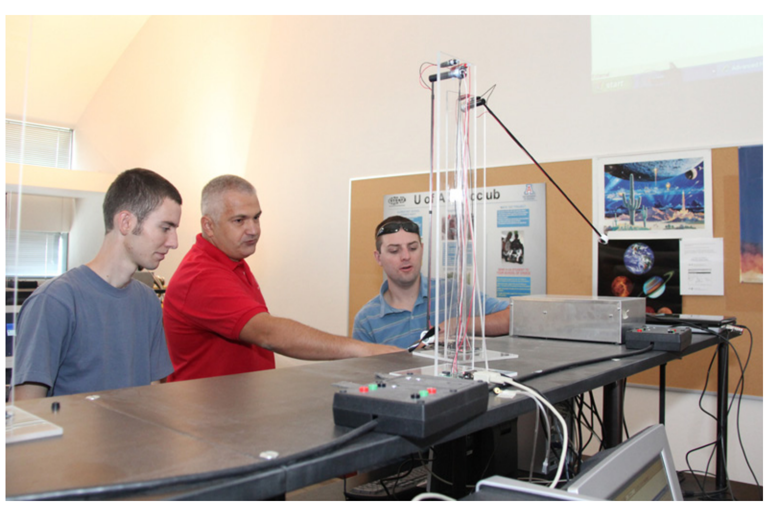

FIGURE 1. *Dr. Enikov and students experimenting with the aeropendulum in the lab. Image courtesy U.A. College of Engineering/Pete Brown.*

Experimenting with the Aeropendulum

In the third month of the semester, the students take an aeropendulum home and begin a series of three homework assignments. The ultimate goal is to develop a control system in MATLAB and Simulink that will keep the pendulum at a predefined angle by managing the voltage applied to the motor (and thus, the speed of the propeller). If an external disturbance is applied to the pendulum, the control system must return it to its correct position within two seconds.

In the first assignment, the students develop a nonlinear mathematical model of the aeropendulum and perform parameter identification by analyzing the system's step response in MATLAB. We keep this assignment simple by having the students consider the dynamics of the pendulum only, not those of the motor and other electronic components. The students then use feedback linearization to arrive at a linear system that has a straightforward transfer function with two real poles.

For the second assignment, students apply what they have learned about root locus design methods to build a closed-loop controller in Simulink. Most begin by plotting the root locus in MATLAB using steady state methods. They discover that the motor's internal friction keeps it from spinning for small input voltages (typically, less than 1 volt). Their designs must take this nonlinearity into account.

The Aeropendulum Setup

The aeropendulum includes a two-inch propeller driven by a small DC motor. The motor and propeller are attached to the free-swinging end of a carbon rod (Figure 2). The other end of the rod is fixed to a potentiometer that serves as a pivot point. A Microchip PIC16F690 microcontroller manages the voltage supply for the motor via pulse-width modulation (PWM). It also reads the voltage of the potentiometer, which is proportional to the angle of the rod. The controller communicates with a PC running MATLAB or Simulink via an RS-232 interface.

The complete device can be built for less than $100 and reused from semester to semester. Class fees pay for aeropendulum maintenance and repair.

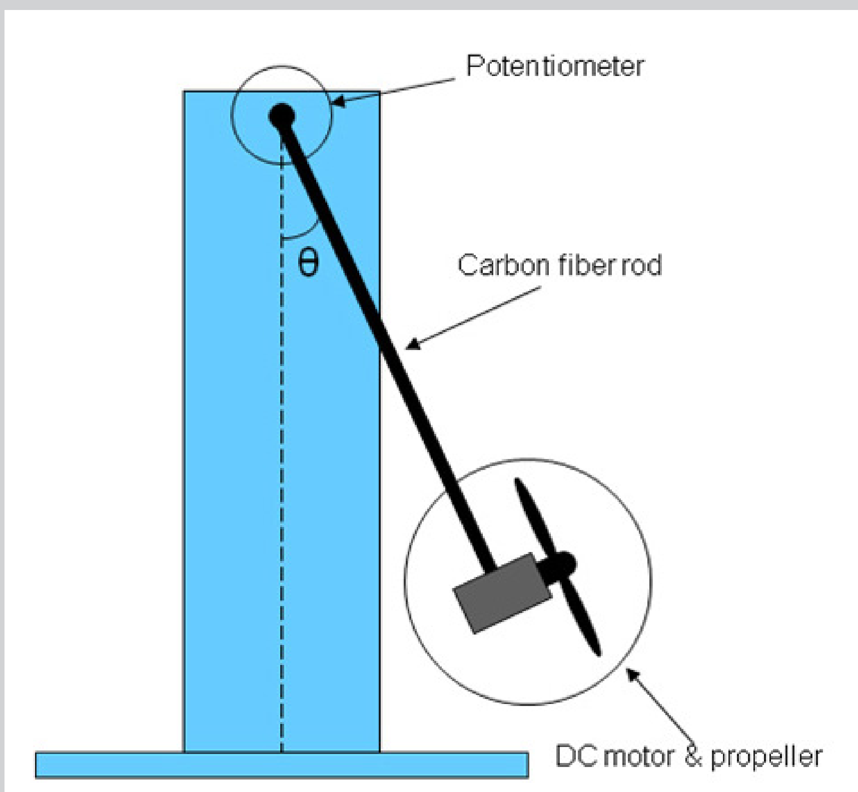

FIGURE 2. *Diagram of the aeropendulum showing the potentiometer, rod, motor, and propeller.*

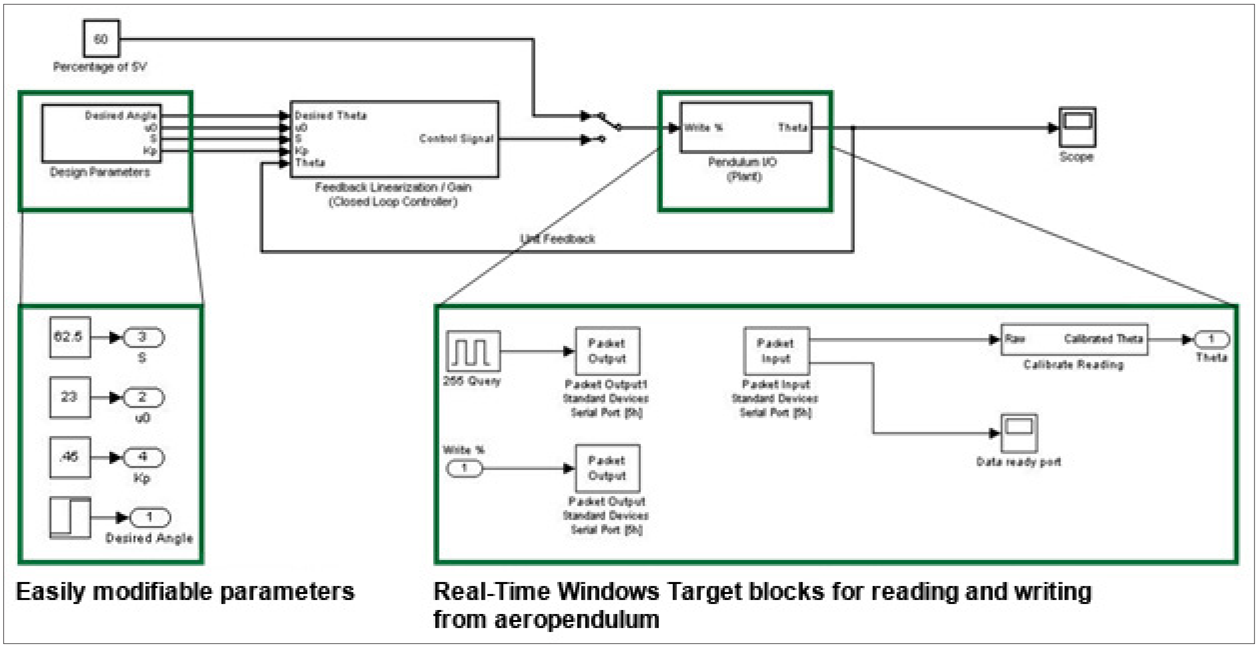

FIGURE 3. *Simulink model of the aeropendulum controller.*

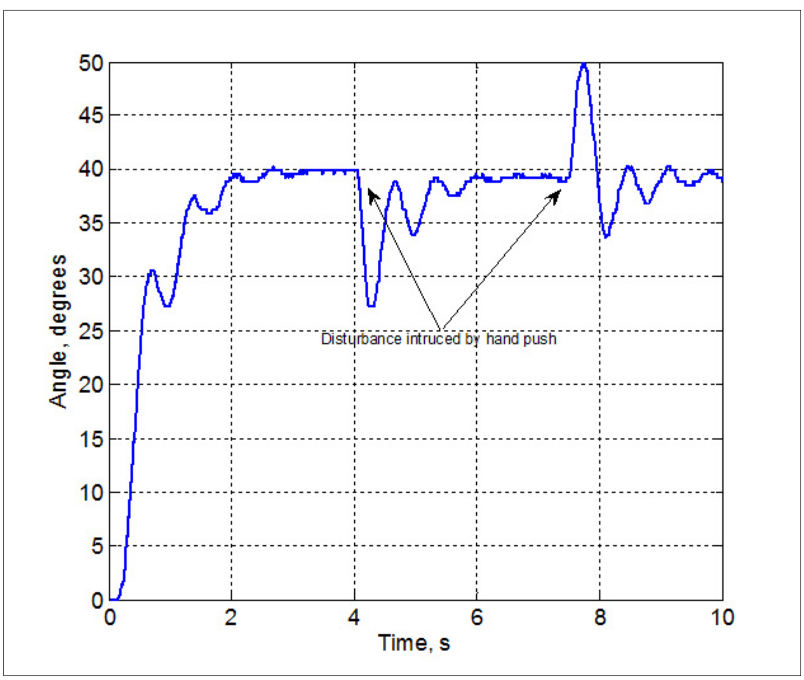

FIGURE 4. *Simulink scope output showing the angle of the aeropendulum (theta) as a function of time.*

The students then build a Simulink model of the controller (Figure 3), which they run on their PCs. The model receives the potentiometer voltage via an RS-232 interface and computes the angle of the aeropendulum. Based on the current angle and the target angle, the model generates a control signal for the motor and sends it to the embedded processor, where it is converted to a PWM signal. The students use Simulink to visualize the angle of the aeropendulum in real time as they run their experiments (Figure 4).

Based on what they have learned about root locus methods and their second-order Simulink models by this point in the course, the students believe that the system should be stable for any gain value. In the third assignment, they learn that this is not the case. We have them incrementally increase the gain until they discover that above a certain threshold the system becomes unstable and the aeropendulum oscillates wildly. Students are tasked with refining their model to reflect this observed behavior.

Measuring the Frequency Response with Real-Time Windows Target

We recently began using Real-Time Windows Target™ to enable the students to run their Simulink controller models in real time on their laptops. The students generate C code from the model using Real-Time Workshop® and then compile the code for execution on a Windows PC using Real-Time Windows Target. This approach reduces sampling time from 50 milliseconds to 5 milliseconds.

My graduate student assistant benefited greatly from this reduced sampling time because it enabled him to measure the frequency response of the aeropendulum and modulate it at higher frequencies. He found that the instability of the system at higher gains was related to the position of the propeller as it spun.

By providing an order of magnitude improvement in sampling rate, Real-Time Windows Target enabled more advanced analysis and made the basic analysis and control more consistent and repeatable. We plan to use Real-Time Windows Target more frequently in future course assignments.

Student Feedback

Student response to the course and to the hands-on assignments has been very positive. In a course evaluation, students reported that the aeropendulum exercises exceeded their expectations in illustrating a number of technical concepts, including second-order system response, the relationship between stability and gain, the use of root locus, and system recovery after a disturbance.

For many, the biggest benefit of the course was the ability to apply their newly acquired control design skills to their senior-year capstone design projects. More than half cited this design experience on their resumes. In fact, one of the largest engineering companies in the Tucson area already employs several former students who took the course. When the company recruits on campus, they often discuss the aeropendulum project with my current students to gauge their understanding of control design. Professors at other universities have expressed interest in using the aeropendulum, and we recently sent 16 of the devices to California State University, Sacramento, for use in control systems courses.

When I ask the students to imagine the controls needed to turn a rocket, it can be difficult for them to fully appreciate the dynamics at work. That is never the case with the aeropendulum. They can actually touch the aeropendulum to produce a disturbance and then observe its response. That gets them thinking about how to improve the control system. When they see the system respond in a way that the model didn't predict (for example, when it oscillated at large gains), they immediately understand the limits of their model and begin thinking of ways to improve that as well.

These experiences do not require expensive lab equipment. In fact, we have found that MATLAB, Simulink and a simple low-cost device are all that is needed. ■

About the Author

Dr. Eniko T. Enikov is an associate professor in Aerospace and Mechanical Engineering at the University of Arizona. He is the director of the Advanced Microsystems Laboratory, which specializes in the development of microelectromechanical systems (MEMS) and in the emerging field of nanotechnology.

Learn More Online

- Embedded Control Systems Education at the University of Michigan
  mathworks.com/embedded-control-systems
- Undergraduate Engineers Develop Hydraulic Servo Control Systems Using Model-Based Design with Simulink
  mathworks.com/hydraulic-servo

91891v00 09/10