Modeling, Simulation, and Control of a Real System

Robert D. Throne
Rose-Hulman Institute of Technology

Abstract

We have developed a preliminary laboratory sequence in our introductory linear control systems class that combines modeling a real system, developing a control system design based on this model, and then implementing the designed controller on the real system. After our first time through this new sequence we have found three educational benefits: the students realize that (1) their models are only approximate descriptions of the real systems, (2) even though the models are only approximate, feedback control can compensate for these modeling errors, and (3) real motors have limits on gains, which must be included in their design.

Background

Over the past few years our introductory controls class (ECE-320) has been taught primarily with laboratories that consisted of weekly Simulink and Matlab projects. These projects included transfer function and state space models of systems. Many nonlinear effects, such as limits on motor speed or torque, were easily incorporated into simulations with Simulink. However, the students often were left with the belief that the system models were exact, and did not appreciate the nonlinear effects we tried to add to the system models. In short, the students felt the Simulink models were not real.

 In an effort to address these issues eleven ECP-210a spring/mass/damper rectilinear systems were purchased through an NSF CCLI grant obtained by investigators from both the Electrical and Computer Engineering and Mechanical Engineering departments at Rose-Hulman. These systems allow for easy implementation of different standard controller types and are easily reconfigurable. Figure 1 shows one of the "carts" of the system, connected with two springs. The position encoder is shown toward the back of the system. The carts are moved via a motor with a rack and pinion mechanism, as shown in Figure 2.

We developed four groups of labs for the course, each group of labs covered one or two 3-hour laboratory sessions. The initial two labs focused on determining the system model in either the time domain or frequency domain. Since a system model is required for any controller design, and would need to be done first in all subsequent labs, we practiced the modeling process a great

deal so the students could model the systems quickly and accurately. In these preliminary labs we focused on modeling only second order systems. In the next group of two labs the students modeled different systems, and then designed closed loop controllers using their system model. The final group of two labs had the students model the systems, design a preliminary closed loop controller using their model, and then implement their controller on the real system. Since there was often a lot to do as the labs progressed, a number of Matlab graphical user interfaces (GUI's) were written so the students could concentrate on the higher level concepts, and not get caught up in Matlab. Each of these lab sequences will be described more fully in the following section.

Laboratory Descriptions

System Modeling in the Time Domain. In our first lab in this sequence, the students constructed six different second order systems by varying the masses, springs, and dampers on an ECP Model 210a system. This familiarized the students with configuring the ECP systems and using the ECP software. The systems were modeled using the log decrement method and then independently by matching the step response of the real system to a second order model. Two Matlab GUI's were written to allow students to quickly model the systems. One of the GUI's allowed rapid determination of a second order model based on the log decrement method, while the other determined the model using a fit to the step response.
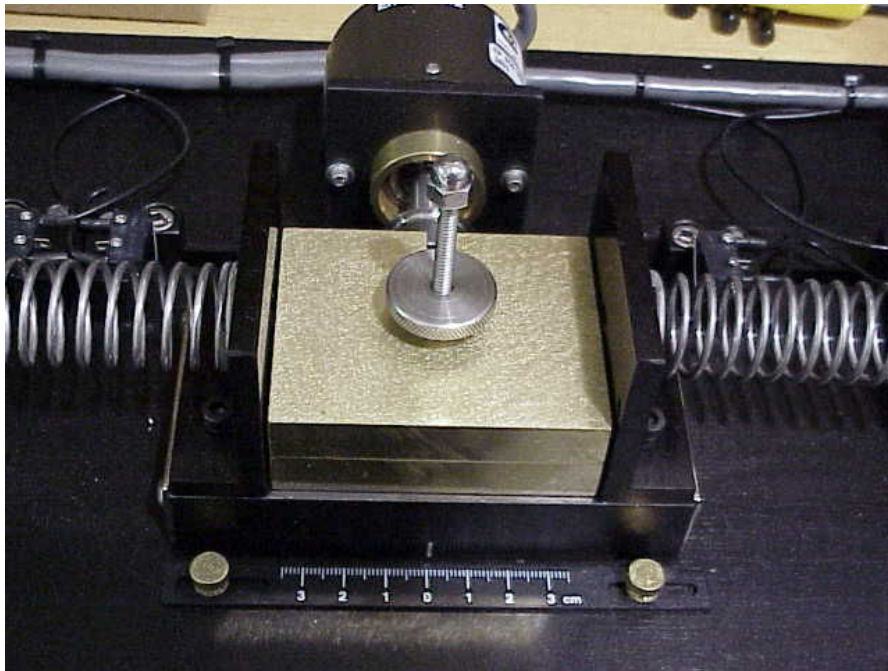

Figure 1: One cart (with variable masses) connected to two (variable) springs of the ECP-210a system. The position encoder is shown in the background.
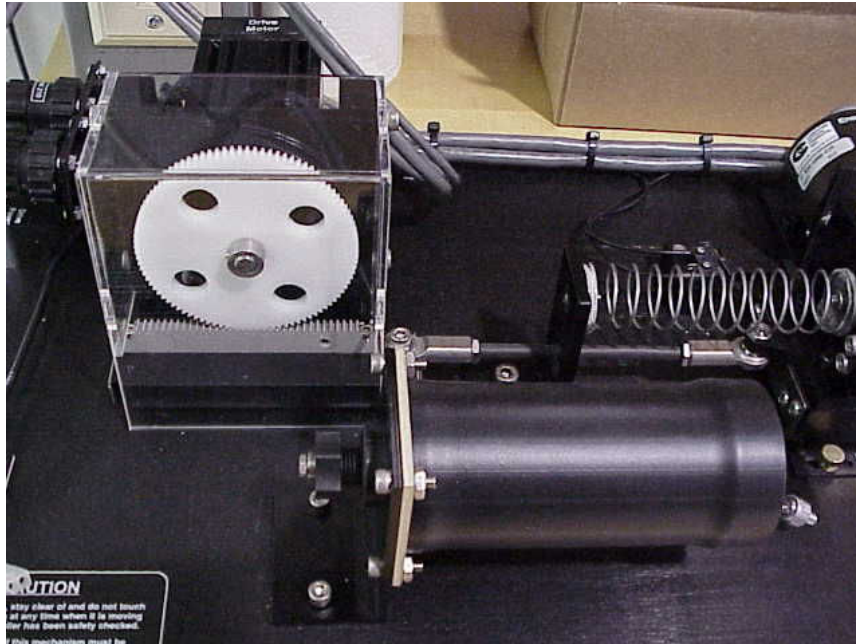
Figure2. The rack and pinion (at the end of the spring on the far left) is connecting the motor to the first cart.

System Modeling in the Frequency Domain. The next lab had the students model a second order system in the time domain. Next, sinusoids of different frequencies were used to excite the system and the ratios of the output amplitude to the input voltages were recorded at 10 to 12 frequencies for each system. Based on the magnitude portion of this Bode plot the students attempted to identify the second order system (primarily by looking at the resonant frequency and bandwidth). Matlab software was given to the students to compare the measured frequency response with the frequency response of their estimated transfer function. Once this initial transfer function was computed, additional frequency response data was obtained to better identify the resonant frequency, and the model of the system was updated. Finally, the students compared the frequency response of their time-domain estimated transfer function, the frequency response of their frequency domain estimated transfer function, and the measured frequency response of the system. The students followed this sequence for two or three different second order systems.

Controller Design Based on the System Model. Once the students became proficient at modeling second order systems, the next group of two labs had the students develop a model for a second order system and then explore the use of different types of controllers for obtaining the desired response. At this point, the system motor also must be modeled. We model this as a simple gain, which is determined by examining the steady state closed loop values for inputs with three or more amplitudes.

The controller design is mostly done through the use of a Matlab GUI, shown in Figure 3. (We plan to migrate to Matlab's sisotool in the spring.) The students enter the system model they want to try and control (from their workspace). They then select the type of controller they want to use. They can choose from proportional (P), integral (I), proportional+integral (PI), proportional+derivative (PD), lead, and proportional+integral+derivative (PID) controllers. For each of these controllers they enter the desired poles and/or zeros of the controller. They can then make a root locus plot of the system with the chosen controller and choose a gain. Finally, they can plot the step response of their controller and the system model to see if it meets the time-domain specifications. Allowing the students to use this GUI interface allows them to examine different controller schemes very quickly. The students are told to try and meet the design specs with each type of controller, and if they feel a particular type of controller will not work they are to provide some justification. Part of the design specifications at this point are limits on the controller gains, which the students often feel are arbitrary.
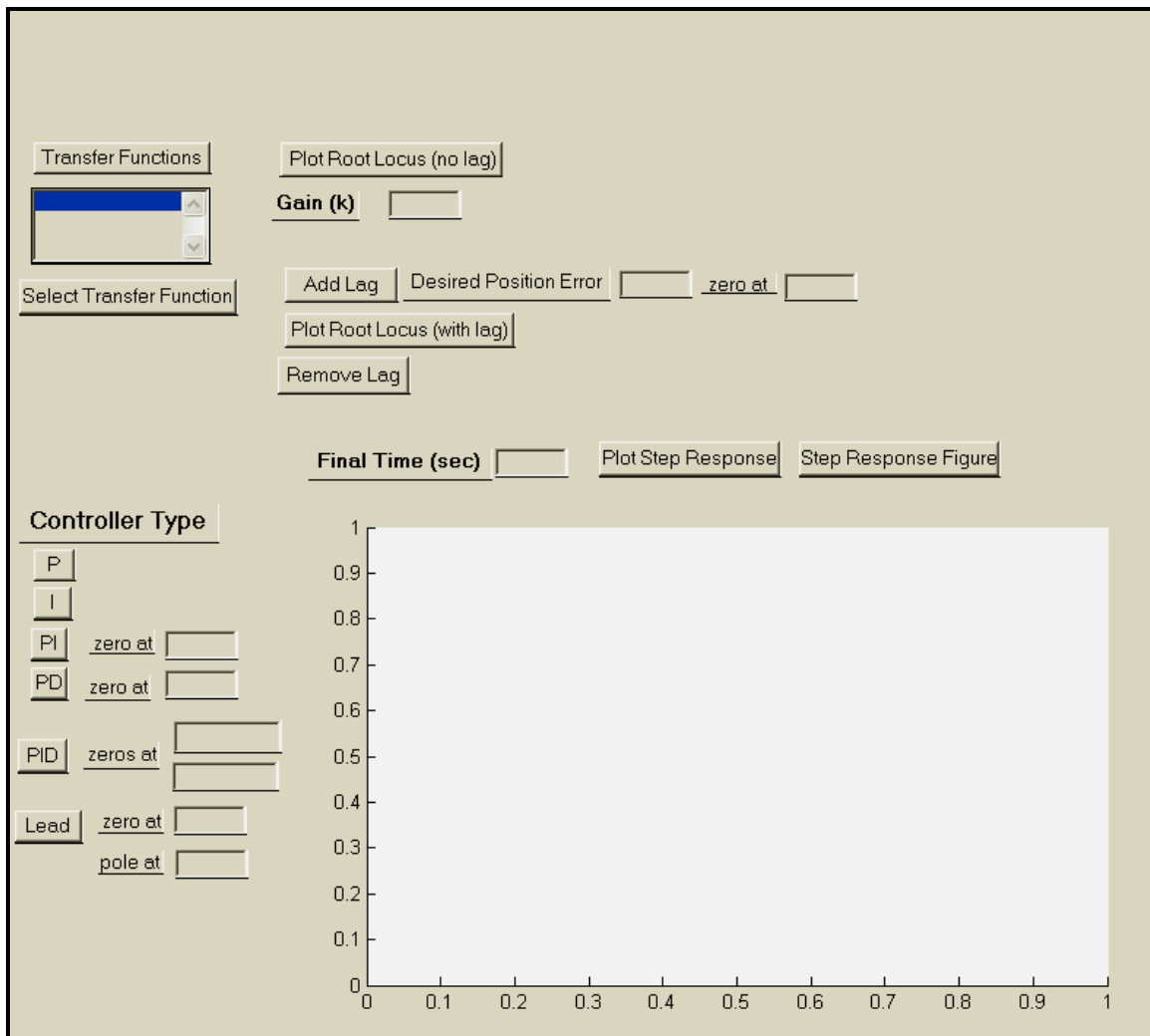

Figure 3: Matlab GUI for designing the controller for the system model.

Closing the Loop: Implementing Controllers on the Real Systems. The last set of two labs has the students go through the entire process, culminating with implementing their controllers on the real system. Specifically, the students develop a transfer function model of a system, develop a controller they think will meet the design specifications and simulate the system with this controller, and then implement the controller on the real system and see if the real system behaves as the model of the system predicts it will. In most cases the model of the system is reasonable, but often not too close to the real system. Often slight tweaking of the model parameters improves the response of the real system. It is at this point that the students also realize that there are real limits on the motor, and the design specs limiting the gains of the controller are real. Many of the students feel a real sense of accomplishment at the conclusion of these labs.

Concluding Remarks

At this point, we have not yet examined modeling and controlling higher order systems, or modeling using state space techniques, though those topics will be included in future labs. The use of the equipment make the subject of controls seem much more real to the students, and developing control algorithms and implementing them on real systems give the students a real appreciation for the role of modeling systems and real world limits on controller design.

ROBERT D. THRONE is currently an Associate Professor in the ECE department at Rose-Hulman Institute of Technology. He was in the EE department at the University of Nebraska-Lincoln for ten years prior to moving to Rose-Hulman. His interests are in control systems and the solution of inverse problems.