ECE-320 Lab 8: Discrete-Time State Variable Feedback

Overview

In this lab you will be controlling both of the one degree of freedom systems you previously modeled using discrete-time state controllers. Both one degree of freedom systems must be controlled, and if there are two people in your lab group each lab partner should do a different system.

*You will need to download the files for **Lab8** from the class website.*

Design Specifications: *For each of your systems, you should try and adjust your parameters until you have achieved the following:*

Rectilinear Systems (Model 210)

- Settling time less than 1.5 seconds.
- Steady state error less than 0.1 cm for a 1 cm step, and less than 0.05 cm for a 0.5 cm step
- Percent Overshoot less than 25%

*Your memo should include **two** graphs **for each** of the 1 dof systems you used (two state variable feedback systems with different sample rates). Your memo should compare the difference between the predicted response (from the model) and the real response (from the real system) for each of the systems.*

For each of your two 1 dof systems, you will need to go through the following steps:

Step 1: Set up the 1 dof system exactly the way it was when you determined its model parameters.

Step 2: Modify **DT_sv1_driver.m** to read in the correct model file. You may have to copy this model file to the current folder.

Step 3: Modify **DT_sv1_driver.m** to use the correct *saturation_level* for the system you are using.

Step 4: Set the sampling interval to 0.05 seconds (the first time) and then something about half as large the second time. Note that you may have to play around with this and modify the pole locations as you go. As the sampling time get smaller it is more difficult to make the system respond quickly, so you will need to move the poles further from the origin.

Step 5: Design a discrete-time state variable controller using **DT_sv1_driver.m**

- Simulate the system for 1.5 seconds. If the design constrains are not met, or the control effort hits a limit, redesign your controller (you might also try a lower input signal). Try and stay away from the maximum allowed control values as much as possible, they are not as good a predictor with discrete-time systems as with continuous time systems.

- Although it should not matter, only use *positive* pole locations. Apparently the ECP systems are not particularly happy with negative pole locations.

- Start with poles at around 0.5 or 0.6, and then move them in closer once you see how the system is responding. We are not trying to make the systems go particularly fast here, but just see how discrete-time control systems work.

- The ECP systems really do not like poles at the origin, so don’t put any poles there (no deadbeat control.)

Step 6: Reset the system using **ECPDSPReset.mdl**

Step 7: Compile the correct closed loop ECP Simulink driver (**Model210_DT_sv1.mdl)**, connect to the system, and run the simulation.

Step 8: Use the **Compare_DT1.m** file (or a modification of it) to plot the results of both the simulation and the real system on one nice, neatly labeled graph. You need to include this graph in your memo.