ECE-520 Lab 5

State Variable Feedback Control For One and Two Degree of Freedom Systems

Overview

In this lab you will be utilizing state variable feedback control to place the poles in a closed loop system to improve the performance of your open loop one and two degree of freedom systems. In addition, you will be comparing the models you created for your systems either by sampling the continuous time model or by using Matlab's system identification toolbox.

For each of the systems you use in this lab (and for the remainder of the labs in this course) you will go through the following basic procedure:

1) *Modify the Simulink driver you are using to load the mathematical model file (.mat file) that corresponds to the way you have the system configured.*
2) *Simulate the system to determine if your model meets the desired specifications. If it does not, modify the pole locations until it does meet the specifications. In addition, you need to be sure the control effort does not reach the saturation level. The simulated control effort for the discrete-time system does not model the real control effort as well as it does for the continuous time system. It would work better if we were to sample at a higher rate. Hence, if your control effort is near the saturation level it is not likely to work well.*
3) *Once you have simulated the system all of the variables you used are in Matlab's workspace. Now compile the correct ECP driver file that replaces the model of the ECP system with the ECP system driver (and the real ECP system). Reset the ECP system, and run the system.*
4) *Finally, compare the predicted response (from your mathematical model) with the real response (from the ECP system). A graph showing the predicted and real response is to be included in your memo (as an attachment) for each system you simulate.*

Notes and Guidlines:

- Although it should not matter, only use *positive* pole locations. Apparently the ECP system 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.)

- Run the systems for at least one second, but don't run the system so long most of your graph shows the system at steady state.

- Reset the system each time before you run it.

- As soon as you start you controller (click on play) be prepared to stop the system. In particular, listen for vibrations that are growing louder and stop the system as soon as possible after this.

Design Specifications

For each of your systems try and have the simulated systems meet the following design specifications

a) Settling time less than or equal to one second
b) Steady state error is zero for a 1 cm step input (or a 15 degree step input)
c) Percent overshoot less than 20%

You should try for the 1cm (or 15 degree) inputs, but if your system is unwilling to cooperate try a 0.5 cm input (or a 10 degree input). This is particularly true when trying to control the position of the second cart/disk.

At this point don't worry if your real system is does not meet the steady state error requirement, though it should meet the other requirements. In next weeks lab we'll see how to produce a zero steady state error (by inserting an integrator and making the system a type 1 system).

*Note: For each system you should have four models generated using the system identification toolbox. These were the files that the program **compare_sys_id.m** wrote out with a prefix of **sys_id_...** You should have one model generated by sampling the continuous time model. The files the program **compare_sys_id.m** wrote out with the prefix **c2d_...** are all the same. There was only one model we made by sampling the continuous time model*

Part A: One Degree of Freedom Rectiliner Systems

a) Load the files from the **basic files** folder into a folder for **Lab 5**.
b) Load all of the mathematical model files from **Lab 3** into the **Lab 5** folder.
c) Configure your one degree of freedom system the way you did in **Lab 1**.
d) Modify the Matlab driver file **DT_sv1_driver.m** to read in one of the mathematical model files
e) Use state variable feedback to place the poles in such a way that you think you will meet the system requirements given above. Remember there is usually a tradeoff between speed of response and the required control effort, which is limited by our motors.
f) Run the simulation for at least one second (*Tf should be at least one second*). This is because the ECP systems tend to hang up if they run for less than a second.
g) Simulate the systems, check to see that they meet the design requirements and the control effort does not reach the saturatation level. If there is a problem, go to step (e) and try a different set of pole locations.
h) The Simulink file **Model210_DT_sv1.mdl** replaces your mathematical model of the system (used in **DT_sv1.mdl**) with the ECP drivers (and hence the real system). The states are named differently in this file than they are in **DT_sv1.mdl** so we can compare the model and the real system later. Compile **Model210_DT_sv1.mdl.** This Simulink file will read in the value of K (the state feedback gains), the value of Gpre (the prefilter gain), and the value of C (the output matrix) from the Matlab workspace.
i) Reset the ECP system.
j) Connect **Model210_DT_sv1.mdl** to the ECP system and run it.
k) Use the program **Compare_DT1.m** to produce a plot comparing the predicted response using the mathematical model of the system with the real response of the system. Note that the third state is not plotted since it is not really all that important.
l) Copy and paste this graph into your **Word** document that will eventually become your memo for this lab. It is a real good idea to write a short caption at this time so you don't forget what you just did. In particular, you will want to remember which mathematical model was use to generate the predicted response.
m) Repeat steps d-l for each of the different models you generated for the system. You should have a total of 5 models to compare.
n) Of the five mathematical models you tried, pick the one you think is the best predictor of the behavior of the real system.

Part B: Two Degree of Freedom Rectiliner Systems

For this system you need to basically go through the same steps you used in **PART A**, except you will use the programs **DT_sv2.mdl** and **DT_sv2_driver.m** you wrote for your homework. You will need to copy the program **Model210_DT_sv1.mdl** to **Model210_DT_sv2.mdl** and modify the program **Model210_DT_sv2.mdl** to work with the two degree of freedom system**.** You will also have to copy **Compare_DT1.m** to **Compare_DT2.m** and then modify **Compare_DT2.m** to plot the first four predicted states compared to the first four real states (we don't care about the delayed input state). You should use subplot and put all four graphs on one (neatly labeled) plot.

For each of your 5 mathematical models assume we are trying to control the position of the first cart, and then we are trying to control the position of the second cart. Hence you will have a total of ten plots as you analyze your system.

After you have all of the simulations, determine which mathematical model seems to best predict the behavior of the real system.

Part C: One Degree of Freedom Torsional Systems

For this system you need to basically go through the same steps you used in **PART A**, except you will need to modify the programs **DT_sv1.mdl** and **DT_sv1_driver.m** to work with a rotational system. Even though I indicated the desired input in degrees, your mathematical model works in radians, so you will need to convert from radians to degrees. You should, however, plot your outputs in degrees or degrees/second

You will need to use the program **Model205_DT_sv1.mdl** to interface with the torsional system**.** You will also have to modify **Compare_DT1.m** to plot the predicted states compared to the real states in degrees or degrees/second (we don't care about the delayed input state).

For each of your 5 mathematical models you should compare the predicted response using your mathematical models with the response of the real system. Hence you will have a total of five plots as you analyze your system.

After you have all of the simulations, determine which mathematical model seems to best predict the behavior of the real system.

Part D: Two Degree of Freedom Torsional Systems

For this system you need to basically go through the same steps you used in **PART B**, except you will need to modify the programs **DT_sv2.mdl** and **DT_sv2_driver.m** to work with the torsional systems. You will need to copy the program **Model205_DT_sv1.mdl** to **Model205_DT_sv2.mdl** and then modify it to work with the two degree of freedom system**.** You will also have to modify **Compare_DT2.m** to plot the first four predicted states compared to the first four real states (we don't care about the delayed input state) in degrees or degrees/second. You should use subplot and put all four graphs on one (neatly labeled) plot.

For each of your 5 mathematical models assume we are trying to control the position of the first disk, and then we are trying to control the position of the second disk. Hence you will have a total of ten plots as you analyze your system.

After you have all of the simulations, determine which mathematical model seems to best predict the behavior of the real system.

Your memo should summarize which mathematical model seemed to work best at predicting the response of the real system, for each of the four systems you modeled. In particular, how well did the discretized continuous time model work compared to the other models?Was there any pattern in which models seemed to predict the system response better than other models? Also, if the mathematical models did not seem to work very well, was there any pattern to the failures? For example, was the final value the same for the mathematical model and the real system? Did the mathematical models work well predicting the positions but not the velocities? If you see any patterns of discrepancies between the predicted response and the response of the real system let me know in your memo. If you saw no patterns in the difference between the predicted and real response also indicate this in your memo. Your memo should also include all of the plots required above as attachments. Each plot should have a figure number and a caption. Since there are so many plots you should try and put two to four plots on each page.