Lab 8: Diophantine Control and Dynamic Prefilters

Overview

In this lab you will be controlling the one degree of freedom systems you previously modeled using Diophatine controllers with and without dynamic prefilters.

If you modeled two rectilinear 1 dof systems, start with the rectilinear systems. If you modeled two torsional 1 dof systems, start with the torsional systems.

You will need your Simulink model and the ***closedloop_driver.m*** *file from your homework for this lab.*

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

Torsional Systems (Model 205)

- Settling time less than 1.0 seconds.
- Steady state error less than 2 degrees for a 15 degree step, and less than 1 degree for a 10 degree step (*the input to the Model 205 must be in radians!*)
- Percent Overshoot less than 25%

Rectilineat Systems (Model 210)

- Settling time less than 0.6 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 three graphs for each of the 1 dof systems you used (type 0 system with constant prefilter, type1 system with a constant prefilter, type 1 system with a dynamic prefilte.r) Be sure to include the values of the closed loop poles and the dynamic prefilte in the captions for each figure. 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. How does the use of a type 1 system compare to that of a type 0 system? How does the dynamic prefilter change the response from that of a constant prefilter? Attach your Matlab driver file ***closedloop_driver.m***

For each of your three 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 it's model parameters.

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

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

Step 4: Diophantine Control (Type zero system, constant prefilter)

- Design a diophantine controller to meet the design specs (you may have already done this in the homework). Use a **constant prefilter**.
- Implement the correct gains into **closedloop_model.m**
- Simulate the system for 1.5 seconds. *Be sure to use radians for the Model 205 system!* 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)
- Compile the correct closed loop ECP Simulink driver, connect to the system, and run the simulation.
- Use the **compare1.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. The results for the torsional systems must be displayed in degrees.You need to include this graph in your memo. *(Note: the position error is likely to be off. There is nothing you can do about it.)*

Step 5: Diophantine Control (Type 1 system, constant prefilter)

- Design a diophantine controller to meet the design specs (you may have already done this in the homework). Use a **constant prefilter**.
- Implement the correct gains into **closedloop_model.m**
- Simulate the system for 1.5 seconds. *Be sure to use radians for the Model 205 system!* 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)
- Compile the correct closed loop ECP Simulink driver, connect to the system, and run the simulation.
- Use the **compare1.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. The results for the torsional systems must be displayed in degrees.You need to include this graph

in your memo. *Be sure to include the location of the closed loop poles in your memo.*

Step 6: Diophantine Control (Type 1 system, dynamic prefilter)

- Design a diophantine controller to meet the design specs (you may have already done this in the homework). Use a **dynamic prefilter** to cancel the zeros of the closed loop system. Be sure the zeros are in the left half plane!

- Implement the correct gains into **closedloop_model.m**

- Simulate the system for 1.5 seconds. *Be sure to use radians for the Model 205 system!* 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)

- Compile the correct closed loop ECP Simulink driver, connect to the system, and run the simulation.

- Use the **compare1.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. The results for the torsional systems must be displayed in degrees.You need to include this graph in your memo. *Be sure to include the location of the closed loop poles and the dynamic prefilter in your memo.*