ME EN 5230/6230, CS 6330, ECE 6651

Intro to Robot Control – Spring 2023

LAB ASSIGNMENT #3: ROBUST AND ADAPTIVE CONTROL

Introduction

The purpose of this assignment is to test the robustness of the controllers you designed in Lab #2, and to compare the robustness of a sliding mode controller, as you simulated in PS#6. We will use the Quanser 2-DOF serial robots, in the same configuration as in Lab #2. We have placed some small brass weights on each bench, which you can attach at various points along Link 2 to introduce some disturbances to your mass parameters.

You will not need a new Simulink template for this assignment. For the sliding mode controller, you can simply build onto your inverse dynamics controller from Lab #2. Please review the lab protocols from Lab 2 handout before beginning.

Lab Exercises

1. Implement a **sliding mode control** action on top of your inverse dynamics control from Lab #1 and experimentally tune your sliding mode parameters to optimize the tracking performance at low speed (f=0.2 circles/s) and high speed (f=1 circles/s). You should be able to use the same sliding mode structure and similar sliding mode parameter values as you used in PS#7. Compare the tracking performance of your sliding mode controller to the controllers you implemented in Lab #2.
2. Now **test the robustness** of your previously designed controllers by attaching one or more brass weights to Link 2 (but don't change any of the parameter values in your compensators). Compare the robustness of the following controllers at high speed (f=1 circles/s):

   2.1 PD Control with feedforward compensation (from Lab #2)
   2.2 PD Control with inverse dynamics control (from Lab #2)
   2.3 PD Control with inverse dynamics control and sliding mode control (from part 1)
3. **GRADS ONLY:** *(Required for 6000 level students, extra credit for 5000 level students)* Now assume that all the mass/inertia parameter values are unknown to your controller. Construct and test an **adaptive controller**. Compare the tracking performance with other controllers you have implemented (especially your computed torque feedforward controller from Lab #2). Compare the parameter convergence with your simulations in PS#6. Try it with and without the brass weights and see if the mass parameters converge to different values.

For part 1, run the robot at low speed (0.2 circles/s) and high speed (1 circles/s). For parts 2 and 3, only run the robot at high speed. If you wish you can use the *RobotApp* program that is posted on Canvas to operate your Simulink models in real time. For each new control scheme, provide an image of your Simulink model and printouts of any new Embedded MATLAB Functions. For each trial, send the data to the workspace and make a plot of the x-y trajectory (for reference, plot the actual trajectory overtop of the desired trajectory), and a plot of the joint angle errors. Be sure to properly title your plots and label your axes. When comparing the tracking performance of different controllers, you may want to use both peak joint errors and the RMS (Root-Mean-Square) of the joint errors as metrics. For a fair comparison, be sure you use the same simulation configuration (solver, step time, time duration).