ME EN 5230/6230, CS 6330, ECE 6651

Intro to Robot Control – Spring 2023

# LAB ASSIGNMENT #4: OPERATIONAL SPACE CONTROL

## Introduction

The purpose of this assignment is to implement coordinated motion control in **operational space** using the Quanser 2-DOF serial robots. The robot should be oriented on the desk in the same configuration as in Lab 2&3, and the same lab protocols (including homing) will be used. The desired trajectory is the same circle as in previous assignments, but take note that since we are doing operational space control this time, the robot will have no preference between elbow-up and elbow-down configurations. As long as you properly home the robot in the elbow up configuration, and stay away from toggle points (singularities), the robot should stay in the elbow-up configuration.

## Lab Exercises

Begin with the Simulink template from Lab 2, which has the necessary motor outputs and sensor inputs built into the *Robot subsystem*. You can then modify this template to implement the same control schemes in real-time that you simulated in PS#7.

1. First implement **Jacobian Inverse Control**, in which case you can use the same the same PD gains that you used when doing decentralized control in joint space (Lab 2, problem 1). Simulate at low speed (f=0.2 circles/s) and high speed (f=1 circles/s), and compare the **trajectory** and **joint errors** from with Problem 1.1 of Lab 2.
2. Next, you will implement a series of operational space controllers based on the **Jacobian Transpose**. For each of the following control schemes, simulate at low speed (f=0.2 circles/s) and high speed (f=1 circles/s) and compare the tracking performance with the other control schemes. For these controllers, you will not be computing the joint errors at all, so just compare the errors in operational space. Since the PD gains for these controllers will now be penalizing errors in operational space, you will have to find a different set of PD gains than you used in problem 1. For part 2.1, find a set of PD gains that results in comparable RMS errors to problem 1. Then use the same PD gains in parts 2.1-2.3 for a fair comparison.
   - **2.1** Jacobian Transpose Control
   - **2.2** Inverse Dynamics Control in Operational Space
   - **2.3** Robust Control in Operational Space

**Note:** For problem 1, make a plot of the x-y trajectory (for reference, plot the actual trajectory overtop of the desired trajectory) and plots for both joint errors vs. time and operational space errors vs. time. For problem 2, make plots of the x-y trajectory and only the operational space errors vs. time. For each control scheme, provide an image of your Simulink model and printouts of your Embedded MATLAB Functions. Be sure to properly title your plots and label your axes. If you wish you can use the *RobotApp* that is posted on Canvas to control the experiments.