ME EN 5230/6230, CS 6330, ECE 6651

Intro to Robot Control – Spring 2023

LAB ASSIGNMENT #7: TELEOPERATION

Introduction

The purpose of this assignment is to implement **teleoperation** using the Quanser 2-DOF serial robots. For this exercise, we will use a real lab robot as the master, but we will use a virtual (simulated) slave robot in contact with a virtual (simulated) benchtop in order to avoid dealing with noisy force sensors. Simulated force feedback will be available from the virtual interaction between the slave and the benchtop.

Please use one of the robots without force sensors for the master robot. The master robot should be oriented on the desk in the same configuration as in prior labs, and the same lab protocols (including homing) will be used.

Lab Exercises

On Canvas, you are provided with a Simulink Template <Lab7_template2023.slx>. The template has subsystems in place for interfacing with the master lab robot and the simulated slave/environment. Gravity compensators have been provided for both robots, and the initial conditions on the slave robot have been adjusted to match the home position of the master robot. The behavior of the slave robot can be visualized using the *RobotApp*. If you run the template as provided, you should be able to easily manipulate the master robot with your hand (starting at the home position) and the gravity compensation should allow it to remain stationary at any configuration in the workspace.

Your task will be to design and simulate a series of controllers to implement master/slave teleoperation as we discussed in lecture.

1. **Bilateral Servo in Joint Space**. Penalize the difference in joint angles between the robots and feed it back to both master and slave. Recall from lecture that this method should ideally be stable for all gains. However in practice, we know that high gains will excite unmodeled dynamics and/or cause amplifier saturation. You should be able to use comparable PD gains to what you used in Lab 2. Be mindful of your sign conventions on your error and your penalties, such that you are providing negative feedback to both robots.
2. **Bilateral Servo in Operational Space**. Compute the operational space error between the two robots and feed it back to both master and slave. You should be able to use comparable PD gains to what you used in Lab 4. This controller should behave similarly to #1. However note that this time, if you move the master through its singularity, the master and slave may end up in opposite elbow up/down configurations.
3. **Slave Servo with Direct Force Feedback to Master**. Implement this in operational space. Penalize the trajectory error between the two robots, but this time feed the penalty back to the slave only. Use the simulated force feedback from the slave robot to generate torque on the master robot. Again be mindful of your sign conventions on the force feedback so that the force from the benchtop is repulsive rather than attractive. Recall from lecture that this method should be generally unstable, though you can try it.

4. **Bilateral Servo with Direct Force Feedback to Master**. Implement this in operational space. This time the torque to the master robot should be a combination of #2 and #3. This method should be generally stable, while giving you a more realistic perception of contact force. Try different gains on your force feedback.
5. **Velocity vs. Position Control**. Using your controller from part 4, show that if you use PI control on velocity error (instead of PD control on position error), the slave robot can start from a different initial position than the master (i.e. change the initial condition on the *v2p* integrator block inside the slave robot subsystem). In this case, the bilateral servo in operational space should just maintain the same relative distance between the master and slave end-effectors.

For each of the above control schemes, you can try with both a soft wall (k_w = 1 N/mm) and a hard wall (k_w = 10 N/mm). In each case, comment on what you qualitatively experience in terms of stability and your ability to perceive the contact between the slave and the benchtop.

Note: Make plots of both the position tracking error vs. time and the force feedback. Be sure to properly title your plots and label your axes. For each controller, provide an image of your Simulink model and printouts of any new Embedded MATLAB Functions. Be sure to properly title your plots and label your axes. You should definitely use the *RobotApp* to visualize the motion of the virtual slave. Feel free to alter the Refresh Rate of the App to get it to run smoothly in real-time.