Experimental Robotics (CS225A) — Homework #1

(Spring 2020/2021) — Due: **Monday, April 19**

Make sure to provide justification for your answers. This includes labeling all of your plots (title, axes, legend, etc.) and explaining what is shown in the plots. Otherwise, you will lose points.

In this homework assignment, you will implement joint space control for the Franka Panda and experiment with control gains in simulation.

To download the assignment, you'll have to pull the latest updates from `cs225a.git`. If you want to keep your progress from Homework 0, first call `git status` to see what files you've modified, and then call `git add <filename>` for all the files you want to save. Next call `git commit -m "Your commit message here"` to commit the changes to these files. For the rest of the files you don't care about, call `git stash` to revert them back to the original version (if you ever decide you want to bring back the modified files, you can call `git stash pop`). At this point, `git status` should show no modified files (untracked files are fine).

Now, you are ready to download the assignment. Call `git pull`. This will likely ask you to save a commit message for merging `cs225a.git` with your local repository - you can simply save and exit. If there were merging issues, you'll have to go into the problem files, manually fix the merging, and then commit those changes again. Now you're ready to start Homework 1!

Note: You may use `simviz_hw1_slow.cpp` instead of `simviz_hw1.cpp` if your computer has trouble running the simulation smoothly. The normal simviz runs the physics simulation loop at 1,000 Hz, whereas the slow version runs the loop at 200 Hz.

Note: For all problems, the initial robot configuration is $q_0 = (-80°, -45°, 0°, -125°, 0°, 80°, 0°)$. The desired robot configuration is $q_d = (90°, -45°, 0°, -125°, 0°, 80°, 0°)$.

1. Let the joint positions and velocities of the robot be given by q and $\dot{q}$, respectively. Let q_d be the desired joint positions of the robot. Implement the joint space control law:

$$\Gamma = -k_p(q - q_d) - k_v\dot{q}$$

   where k_p and k_v are your control gains.

   (a) Tune your gains to achieve critical damping on joint 1 with $k_p = 400$ and report your chosen k_v.

   (b) Move the robot from its initial configuration to its desired configuration. Plot the joint trajectory for joints 1, 3, and 4. Why do some joints converge closer to the desired position than others?

2. Now, implement the joint space control law:

$$\Gamma = -k_p(q - q_d) - k_v\dot{q} + g(q)$$

   where $g(q)$ is the joint space gravity compensation vector.

   (a) Tune your gains to achieve critical damping on joint 1 with $k_p = 400$ and report your chosen k_v.

   (b) Again, move the robot from its initial configuration to its desired configuration. Plot the joint trajectory for joints 1, 3, and 4. Compare these plots to the ones in Problem 1 and explain what you see.

3. Now, implement the joint space control law that takes into account the dynamics of the robot:

$$\Gamma = A(q)(-k_p(q - q_d) - k_v\dot{q}) + g(q)$$

where $A(q)$ is the joint space mass matrix.

   (a) Tune your gains to achieve critical damping on joint 1 with $k_p = 400$ and report your chosen k_v.

   (b) Again, move the robot from its initial configuration to its desired configuration. Plot the joint trajectory for joints 1, 3, and 4. Compare these plots to the ones in Problem 2 and explain what you see.

4. Now, implement the joint space control law:

$$\Gamma = A(q)(-k_p(q - q_d) - k_v\dot{q}) + b(q, \dot{q}) + g(q)$$

where $b(q, \dot{q})$ is the joint space coriolis and centrifugal force compensation vector.

   (a) Tune your gains to achieve critical damping on joint 1 with $k_p = 400$ and report your chosen k_v.

   (b) Again, move the robot from its initial configuration to its desired configuration. Plot the joint trajectory for joints 1, 3, and 4. Compare these plots to the ones in Problem 3 and explain what you see.

5. Now, let's assume the robot is carying a payload of 2.5 kg at the end-effector. In order to simulate this, change the mass of "link 7" in `panda_arm_simulation.urdf` from 0.5 kg to 3.0 kg. Using the same controller from Problem 4, move the robot from its initial configuration to its desired configuration. Plot the joint trajectory for joints 1, 3, and 4. Compare these plots to the ones in Problem 4 and explain what you see. *Hint:* You will need to call the cmake command in order to copy the modifications to your urdf file to the runtime directory.

6. **Extra credit**: What do you need to do in the controller in order to take the payload into account? (You can't modify the urdf file for the controller.) Implement this change and then move the robot from its initial configuration to its desired configuration. Plot the joint trajectory for joints 1, 3, and 4. Compare these plots to the ones in Problem 5 and explain what you see.

7. Submit your SAI code (`hw1.cpp`).