# **Experimental Robotics (CS225A)** Homework #3

(Spring 2022/2023) Due: **Monday, May 8**

*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 trajectory tracking, joint limit avoidance, orientation control and velocity saturation for the Panda.

To download the assignment, you'll have to pull the latest updates from `cs225a.git`. If you want to keep your progress from previous homeworks, 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.

1. In the last homework, you implemented an operational space PD controller with null space posture control similar to the one below:

$$F = \Lambda(-k_p(x - x_d) - k_v\dot{x}),$$
$$\Gamma = J_v^T F + \mathcal{N}^T(-k_{pj}(q - q_d) - k_{vj}\dot{q}) + g.$$

where $k_p$ and $k_v$ are your operational space control gains, $\Lambda$ is your operational space mass matrix, $k_{pj}$ and $k_{vj}$ are your posture control gains, and $g$ is your joint space gravity vector. In this problem, use null space posture control to track a desired joint configuration of $q_d = [0, 0, 0, 0, 0, 0, 0]^T$. In the following scenarios, the desired end-effector trajectory is given by the following:

$$x_d = \begin{bmatrix} 0.3 \\ 0.1 \\ 0.5 \end{bmatrix} + 0.1 \begin{bmatrix} \sin(\pi t) \\ \cos(\pi t) \\ 0 \end{bmatrix}.$$

Use the following gains for all scenarios:

$$k_p = 100.0$$
$$k_v = 20.0$$
$$k_{pj} = 50.0$$
$$k_{vj} = 14.0$$

(a) In this part, only use the controller given at the beginning of this problem. Plot the actual vs. desired end-effector trajectories (not the joint trajectories). Does your actual trajectory perfectly match the desired trajectory? Why or why not?

(b) Now, we will add proper trajectory tracking to better track the desired end-effector trajectory. In order to do this, we need the appropriate values of $\dot{x}_d$ and $\ddot{x}_d$. Using the desired end-effector trajectory $x_d$, derive the expressions for the following:

i. $\dot{x}_d$
ii. $\ddot{x}_d$

(c) After adding proper trajectory tracking, our operational space controller is given by the following:

$$F = \Lambda(\ddot{x}_d - k_p(x - x_d) - k_v(\dot{x} - \dot{x}_d)),$$
$$\Gamma = J_v^T F + \mathcal{N}^T(-k_{pj}(q - q_d) - k_{vj}\dot{q}) + g.$$

Implement the new controller and track the desired trajectory. Plot the actual vs. desired end-effector trajectories (not the joint trajectories). Compare your results to 1 (a).

2. Now, we will utilize null space posture control to avoid joint limits. Let $\bar{q}_i$ and $\underline{q}_i$ be the upper and lower bounds on the $i^{th}$ joint position $q_i$. We construct the following potential function:

$$V_{mid}(q) = k_{mid} \sum_{i=1}^{n} \left( q_i - \frac{\bar{q}_i + \underline{q}_i}{2} \right)^2,$$

where $k_{mid}$ a constant gain and $n$ is the number of joints in the manipulator. The following gradient of this potential function provides the required attraction to the mid-range joint positions of the manipulator:

$$\Gamma_{mid} = -\nabla V_{mid}.$$

In order to avoid the interference of these additional torques with the end-effector dynamics, we project these additional torques into the dynamically consistent null space. The resulting torques are then given by $\mathcal{N}^T \Gamma_{mid}$. Next, we introduce a null space damping term $\mathcal{N}^T \Gamma_{damp}$, where

$$\Gamma_{damp} = -k_{damp} \dot{q}.$$

The operational space controller is given by the following:

$$\begin{aligned} F &= \Lambda(-k_p(x - x_d) - k_v \dot{x}), \\ \Gamma &= J_v^T F + \mathcal{N}^T \Gamma_{mid} + \mathcal{N}^T \Gamma_{damp} + g. \end{aligned}$$

For the intent of this problem, we will use 1-based indexing for the joints. This means that the first joint is joint 1 and the last joint is joint 7. Also, the lower and upper joint limits for the Panda are given by the following (these are not the real joint limits for the manipulator):

$$\begin{aligned} \underline{q} &= \begin{bmatrix} -165^\circ, & -100^\circ, & -165^\circ, & -170^\circ, & -165^\circ, & 0^\circ, & -165^\circ \end{bmatrix}^T, \\ \bar{q} &= \begin{bmatrix} 165^\circ, & 100^\circ, & 165^\circ, & -30^\circ, & 165^\circ, & 210^\circ, & 165^\circ \end{bmatrix}^T. \end{aligned}$$

(a) Write out the analytical expression for $\Gamma_{mid}$. Show your work.

(b) With your expression for $\Gamma_{mid}$, compare the command torques $\Gamma$ in this problem with the command torques from problem 1 ($\Gamma = J_v^T F + \mathcal{N}^T(-k_{pj}(q - q_d) - k_{vj}\dot{q}) + g$). Specifically, compare the terms ($\mathcal{N}^T \Gamma_{mid} + \mathcal{N}^T \Gamma_{damp}$) and $\mathcal{N}^T(-k_{pj}(q - q_d) - k_{vj}\dot{q})$.

(c) Based on your analysis in part (b), choose and report the values of $k_{mid}$ and $k_{damp}$ that achieve the same dynamic response as the one in problem 1.

(d) Here, we will move the Panda to the desired position $x_d = \begin{bmatrix} -0.1, & 0.15, & 0.2 \end{bmatrix}^T$. Use the following operational space controller (note that we do not use the potential field here):

$$\begin{aligned} F &= \Lambda(-k_p(x - x_d) - k_v \dot{x}), \\ \Gamma &= J_v^T F + \mathcal{N}^T \Gamma_{damp} + g. \end{aligned}$$

Plot your actual vs. desired end-effector trajectories. Are you able to track the desired end-effector position? Also plot the joint trajectories and joint limits for joints 4 and 6. Do these joints approach their joint limits?

(e) Now, we will still move the Panda to the desired position $x_d = \begin{bmatrix} -0.1, & 0.15, & 0.2 \end{bmatrix}^T$. However, we will add the potential field to avoid joint limits in the null space of the task:

$$F = \Lambda(-k_p(x - x_d) - k_v\dot{x}),$$
$$\Gamma = J_v^T F + \mathcal{N}^T \Gamma_{mid} + \mathcal{N}^T \Gamma_{damp} + g.$$

Plot your actual vs. desired end-effector trajectories. Are you able to track the desired end-effector position? Also plot the joint trajectories and joint limits for joints 4 and 6. Do these joints approach their joint limits? Compare your results to 2 (d) and explain any differences that you see.

(f) Now, we will change the Panda's desired position to $x_d = \begin{bmatrix} -0.65, & -0.45, & 0.7 \end{bmatrix}^T$. Use the same operational space controller as the previous part:

$$F = \Lambda(-k_p(x - x_d) - k_v\dot{x}),$$
$$\Gamma = J_v^T F + \mathcal{N}^T \Gamma_{mid} + \mathcal{N}^T \Gamma_{damp} + g.$$

Plot your actual vs. desired end-effector trajectories. Are you able to track the desired end-effector position? Also plot the joint trajectories and joint limits for joints 4 and 6. Do these joints approach their joint limits? Compare your results to 2 (e) and explain any differences that you see.

(g) Again, we will move the Panda to the desired position $x_d = \begin{bmatrix} -0.65, & -0.45, & 0.7 \end{bmatrix}^T$. However, we will not project the torques designed to avoid joint limits onto the null space. The new operational space controller is given by the following:

$$F = \Lambda(-k_p(x - x_d) - k_v\dot{x}),$$
$$\Gamma = J_v^T F + \Gamma_{mid} + \mathcal{N}^T \Gamma_{damp} + g.$$

Plot your actual vs. desired end-effector trajectories. Are you able to track the desired end-effector position? Also plot the joint trajectories and joint limits for joints 4 and 6. Do these joints approach their joint limits? Compare your results to 2 (f) and explain any differences that you see. In your opinion, which controller is preferable between the one from part (f) and part (g)? Explain why.

(h) **Extra credit**: What would you do to implement a controller that guarantees joint limit avoidance, but does not affect the task when all the joints are far from their limits?

3. We will implement orientation control with the full jacobian $\begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$.

   Recall orientation error and full control is:

$$\delta\phi = -\frac{1}{2}\sum_{n=1}^{3} R_i \times (R_d)_i$$
$$F = \Lambda_0 \begin{bmatrix} k_p(x_d - x) - k_v\dot{x} \\ k_p(-\delta\phi) - k_v\omega \end{bmatrix}$$
$$\Gamma = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}^T F - \mathcal{N}^T k_{vj}\dot{q} + g$$

   where $R_i$ and $(R_d)_i$ are the i-th columns of $R$ and $R_d$, respectively, and $\omega$ is the angular velocity. Note that you need to use the function `robot->J_0` in order to get the Jacobian in the form $\begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$.

   (a) Control your robot to orientation:

$$R_d = \begin{bmatrix} \cos\frac{\pi}{3} & 0 & \sin\frac{\pi}{3} \\ 0 & 1 & 0 \\ -\sin\frac{\pi}{3} & 0 & \cos\frac{\pi}{3} \end{bmatrix}$$

   and position:

$$x_d = \begin{bmatrix} 0.6 \\ 0.3 \\ 0.5 \end{bmatrix}$$

   Plot the actual vs. desired end-effector trajectories. Also plot $\delta\phi$ over time. Explain what you see in your plots. Choose gains appropriately to achieve good tracking but critical damping. If your plot is not precise enough because the robot is moving too fast, reduce your gains.

4. We will implement velocity saturation in the following operational space PD controller with null space posture control from the last homework:

$$F = \Lambda(k_p(x_d - x) - k_v\dot{x})$$
$$\Gamma = J_v^T F + \mathcal{N}^T M(-k_{pj}(q - q_d) - k_{vj}\dot{q}) + g$$

Reach a desired position of $x_d = \begin{bmatrix}0.6, & 0.3, & 0.4\end{bmatrix}^T$.

(a) With operational space $k_p = 200.0$ and a critically damped $k_v$, plot $x$, $x_d$ (on the same plot) and $\dot{x}$. Explain what you see.

(b) Now, implement velocity saturation with:

$$\dot{x}_d = \frac{k_p}{k_v}(x_d - x)$$
$$F = \Lambda(-k_v(\dot{x} - \nu\dot{x}_d))$$
$$\nu = sat\left(\frac{V_{max}}{|\dot{x}_d|}\right)$$

where $sat()$ is the saturation function:

$$sat(x) = \begin{cases} x & \text{if } |x| \leq 1.0 \\ sgn(x) & \text{if } |x| > 1.0 \end{cases}$$

and $sgn(x)$ is the sign function. In this problem, use $V_{max} = 0.1$m/s. Plot $x$, $x_d$ (on the same plot) and $\dot{x}$, $V_{max}$ (on the same plot) and compare your results to 4 (a).

5. Submit your SAI code (`hw3.cpp`).