CMPSCI 403: Introduction to Robotics: Perception, Mechanics, Dynamics, and Control

HW 07

Due Online, Friday, Nov. 12nd, 11:50 PM

---

1 Operational Space Control

In HW6, you made a dynamics simulation for the robot shown in Figure 1. Given a desired position of the foot $\mathbf{r}_C^d = [x_d,\, y_d]^\top$, your control law took the form:

$$\tau = J^T \left[K(\mathbf{r}_C^d - \mathbf{r}_C) + D(\dot{\mathbf{r}}_C^d - \dot{\mathbf{r}}_C) \right] . \tag{1}$$

As we watched in the lecture, this method effectively controlled the position of the foot using a virtual spring-damper in task-space. However, the Jacobian $J = \frac{\partial \mathbf{r}_C}{\partial q}$ was the only part of this control law that required information about the model! This homework will explore methods to improve the performance of the controller using more information about the model. Specifically, we will use information about its dynamics.

As we have seen in class, the dynamics of the leg are given by the **configuration-space** equations of motion:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau \tag{2}$$

After a bit of algebraic manipulation, these equations can be rearranged into the **operational-space** equations of motion to describe the dynamics of the end-effector:

$$\Lambda(q)\, \ddot{\mathbf{r}}_C + \mu(q, \dot{q}) + \rho(q) = F \tag{3}$$

where $\Lambda(q)$ is the effective mass felt at the foot, $\mu(q, \dot{q})$ gives the coriolis and centripetal forces on the foot, and $\rho(q)$ gives the gravity force felt on the foot. Formula for these quantities are given below:

$$\Lambda(q) = (JM^{-1}J^T)^{-1} \tag{4}$$

$$\mu(q, \dot{q}) = \Lambda J M^{-1}\, C - \Lambda\, \dot{J}\, \dot{q} \tag{5}$$

$$\rho(q) = \Lambda J M^{-1}\, G \tag{6}$$

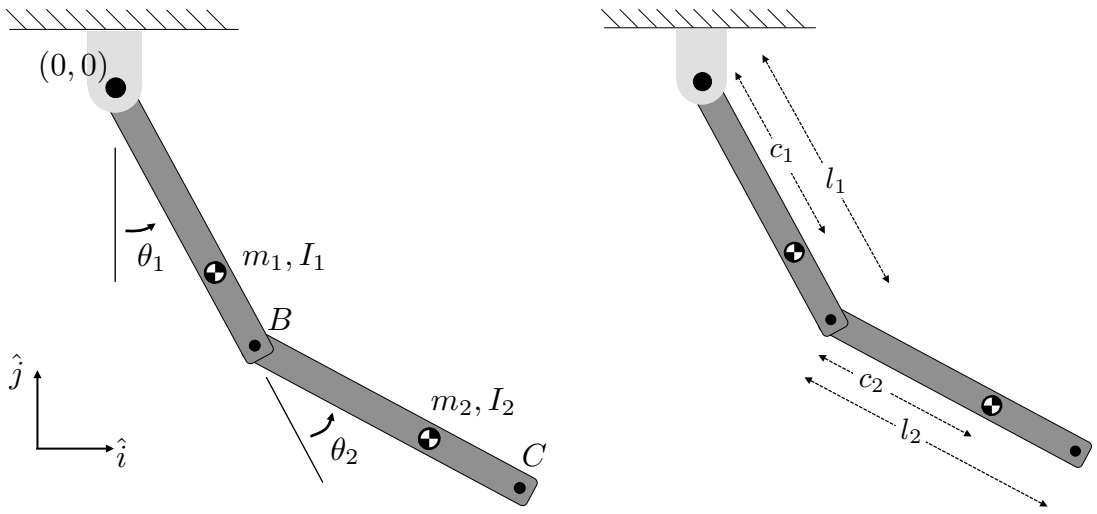

Figure 1: Double pendulum and parameter definitions.

In this assignment, you will explore the use of an extended version of Eq. 1 given by:

$$\tau = J^\top \left[\Lambda \left(\ddot{\mathbf{r}}_C^d + K(\mathbf{r}_C^d - \mathbf{r}_C) + D(\dot{\mathbf{r}}_C^d - \dot{\mathbf{r}}_C) \right) + \mu + \rho \right] . \tag{7}$$

Under mild assumptions, it can be show that the foot will converge to the desired trajectory with this control law.

1. Download the `derive_eqns.m` code. Using the `simulate_pend.m` code as a simulator, implement the control law in Eq. 7 in `controller.m`. The script `derive_eqns.m` creates a number of useful functions what will help in implementing the controller once you specify kinetic/potential energy and generalized forces.

   With your operational space controller, run the circular trajectory tracking simulation. You should use parameters:

   $$\mathbf{r}_C^d(t) = \begin{bmatrix} 0.5 + 0.25\cos(\omega t) \\ -1 + 0.25\sin(\omega t) \end{bmatrix} \tag{8}$$

   $$\omega = 2\pi \text{ rad/s} \tag{9}$$

   $$K = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix} \tag{10}$$

   $$D = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \tag{11}$$

   $$q|_{t=0} = \begin{bmatrix} \pi/6 \\ \pi/6 \end{bmatrix} \tag{12}$$

   $$\dot{q}|_{t=0} = [0, 0]^\top \tag{13}$$

   Turn in a plot of of x, y, x_d, and y_d versus time.

2. Repeat step 2. using modified control laws:

   - $\tau = J^T \left[\Lambda \left(\ddot{\mathbf{r}}_C^d + K(\mathbf{r}_C^d - \mathbf{r}_C) + D(\dot{\mathbf{r}}_C^d - \dot{\mathbf{r}}_C) \right) + \mu \right]$
   - $\tau = J^T \left[\Lambda \left(\ddot{\mathbf{r}}_C^d + K(\mathbf{r}_C^d - \mathbf{r}_C) + D(\dot{\mathbf{r}}_C^d - \dot{\mathbf{r}}_C) \right) + \rho \right]$
   - $\tau = J^T \left[\Lambda \left(K(\mathbf{r}_C^d - \mathbf{r}_C) + D(\dot{\mathbf{r}}_C^d - \dot{\mathbf{r}}_C) \right) + \mu + \rho \right]$

   Turn in a plot of of x, y, x_d, and y_d versus time for each case. Also, include a short description of what is neglected in each controller, and how that relates with the observed performance. For comparison, you probably need to try different trajectory speed, link masses, or feedback gains.

3. **Turn in** your `controller.m` and `derive_eqns.m` code.