Problem Set 7

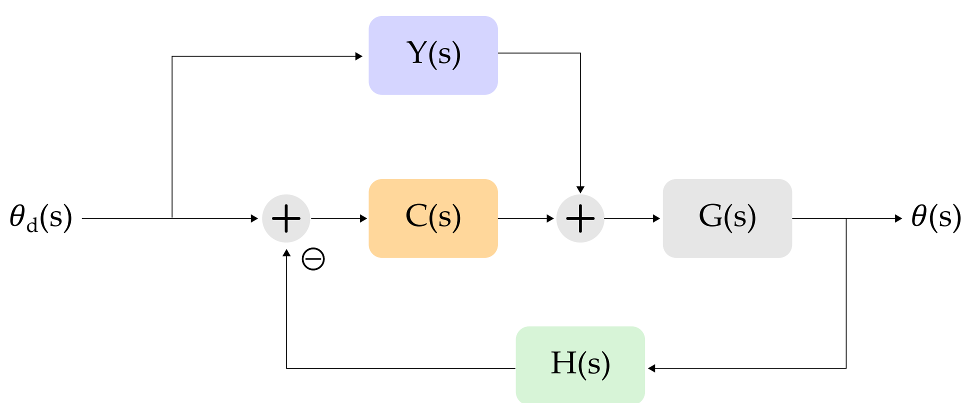
Robotics & Automation
Dylan Losey, Virginia Tech

Instructions. Please write legibly and do not attempt to fit your work into the smallest space possible. It is important to show all work, but basic arithmetic can be omitted. You are encouraged to use Matlab when possible to avoid hand calculations, but print and submit your commented code for non-trivial calculations. You can attach a pdf of your code to the homework, use live scripts or the publish feature in Matlab, or include a snapshot of your code. Do not submit .m files — we will not open or grade these files.

For this assignment we are asking you to also submit **videos** of your simulations. Follow the instructions to **label** these videos based on the problem number, and then submit them all within a **single zipped folder**.

1 Open-Loop and Closed-Loop Control

1.1 (10 points)



Find the closed-loop transfer function:

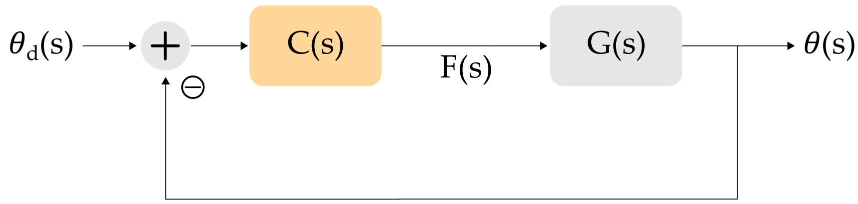
$$\frac{\theta(s)}{\theta_d(s)} \tag{1}$$

for the block diagram shown above. Here $Y(s)$ is a feed forward controller and $C(s)$ is a feedback controller.

2 Stability

Imagine that you purchased three separate 1-DoF robots. Each system comes with its own links, joint, motor, and motor controller, and some of the systems are acting in strange ways. Your job is to design controllers that will result in closed-loop stability.

Throughout this problem assume that you can directly measure θ. Use the block diagram above as a reference when determining closed-loop stability.



2.1 (5 points)

The first 1-DoF robot is a mass-damper:

$$f(t) = m\ddot{\theta} - b\dot{\theta} \tag{2}$$

Design a controller that results in closed-loop stability.

2.2 (10 points)

The second 1-DoF robot has plant dynamics:

$$G(s) = \frac{1}{s(s-3)(s-5)} \tag{3}$$

Design a controller that results in closed-loop stability.

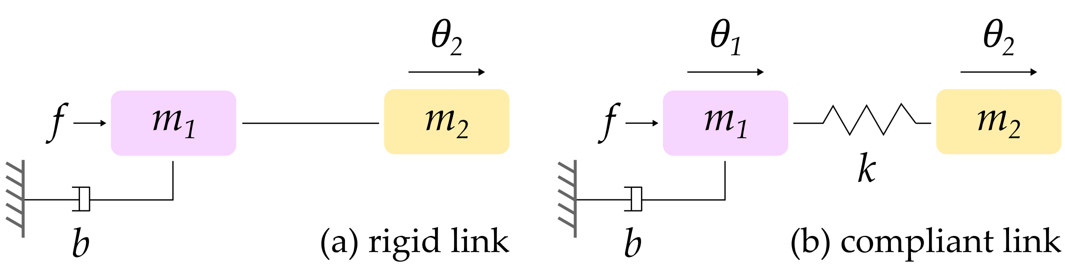
2.3 (5 points)

The third 1-DoF robot is a mass-spring-damper:

$$f(t) = 5\ddot{\theta} + 2\dot{\theta} - 15\theta \tag{4}$$

Design a controller that places both poles at $s = -5$. **Aside –** When both poles of a mass-spring-damper are equal negative real numbers, the system is *critically damped*.

3 Compliant Joints



We often introduce mechanical compliance to make the robot soft and safe during interaction. The drawing above compares a rigid link and (Left) and a compliant link (Right). Here m_1 is the motor mass and m_2 is the link mass. For rigid systems we have a rigid connection between motor output and the link, and the total mass is $m_1 + m_2$. For compliant systems we introduce a spring k between the motor output and the link. In both systems we control the actuator force f to regulate θ_2, the position of the link.

3.1 (5 points)

Find the plant dynamics $G_1(s)$ and $G_2(s)$. Here $G_1(s)$ is the plant for the rigid 1-DoF robot and $G_2(s)$ is the plant for the compliant 1-DoF robot. Both plants should be of the form:

$$G(s) = \frac{\theta_2(s)}{F(s)} \tag{5}$$

3.2 (10 points)

Assume we measure θ_2 in real-time and the closed-loop transfer function is:

$$\frac{\theta(s)}{\theta_d(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)} \tag{6}$$

We will use a proportional controller $C(s) = k_p$. Let $m_1 = 1$ kg, $m_2 = 2$ kg, $b = 0.1$ Ns/m, and $k = 100$ N/m.

- For what range of k_p is the **rigid** robot stable?
- For what range of k_p is the **compliant** robot stable?

3.3 (10 points)

Simulate the rigid system and compliant system in Simulink (Matlab). For both systems set θ_d as a `Pulse Generator` with amplitude 1 m, period 5 s, and pulse width 50%. The model stop time should be 50 s.

Turn in **two separate plots**. One plot should show θ_d, θ_2 (for the rigid system) and θ_2 (for the compliant system) when $k_p = 10$. The other plot should show θ_d, θ_2 (for the rigid system) and θ_2 (for the compliant system) when $k_p = 110$. Include labels and captions.

3.4 (5 points)

Does introducing compliance (e.g., the spring between the motor and link) make it *easier* or *harder* to control the robot? Write a few sentences to explain your answer.

4 Lyapunov Stability Analysis

4.1 (20 points)

In this problem we will introduce *passivity-based motion control*. This approach is a basis for many robust and adaptive robot controllers. Consider a robot arm with dynamics:

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \tag{7}$$

In passivity-based control we select the controller:

$$\tau = M(\theta)y + C(\theta, \dot{\theta})x + g(\theta) - Kr \tag{8}$$

$$y = \ddot{\theta}_d - \Lambda(\dot{\theta} - \dot{\theta}_d), \quad x = \dot{\theta}_d - \Lambda(\theta - \theta_d), \quad r = (\dot{\theta} - \dot{\theta}_d) + \Lambda(\theta - \theta_d) \tag{9}$$

Here K and Λ are constant positive definite matrices. Apply Lyapunov's method to show that the closed-loop system is stable. The equilibrium should be $\dot{\theta} = \dot{\theta}_d$, $\theta = \theta_d$.

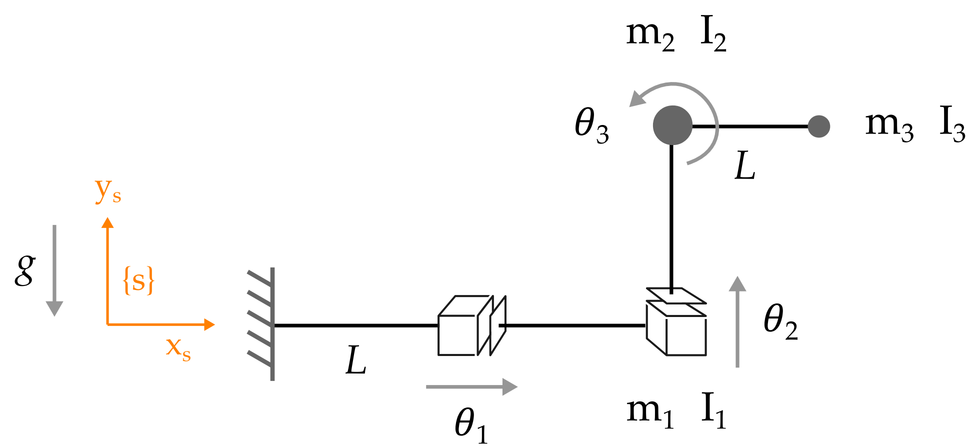
Hint 1. Let M be the mass matrix and let C be the Coriolis matrix. The matrix $\dot{M} - 2C$ is skew symmetric. For any vector a, this means $a^T(\dot{M} - 2C)a = 0$.

Hint 2. Try the following generalized energy function:

$$v = \frac{1}{2}r^T M(\theta)r + (\theta - \theta_d)^T \Lambda K(\theta - \theta_d) \tag{10}$$

5 Multivariable Control

In this problem you will simulate control the robot shown above. You have already obtained the dynamics of this robot in a previous assignment. The robot starts at joint position $\theta = 0$. Here $L = 1$, $m_1 = m_2 = m_3 = 1$, and $I_3 = 0.1$.



5.1 (20 points)

Download the Matlab file `make_controller.m` that was provided with this assignment. Use the given simulation parameters and frame rates; all videos should be 10 seconds in length. Modify the code as needed so that the robot reaches for θ_d using multivariable PD control with gravity compensation:

$$\tau = K_P(\theta_d - \theta) - K_D\dot{\theta} + g(\theta) \tag{11}$$

Turn in the following MP4 videos:

- Make a simulation where $K_P = I$ and $K_D = I$. Let the desired position be:

$$\theta_d = \begin{bmatrix} -2 \\ 2 \\ \pi/4 \end{bmatrix} \tag{12}$$

  Title this video **Problem2_1.mp4**

- Reach for the same θ_d as in the previous part, but now tune K_P and K_D to improve the robot's performance. Make a simulation with your best performing gains, and title this video **Problem2_2.mp4**

- Control the robot to move in a circle. Let t be the simulation time (variable `time` in the code), and define the desired trajectory as:

$$\theta_d = \begin{bmatrix} 2\cos(\frac{\pi t}{2}) \\ 2\sin(\frac{\pi t}{2}) \\ \pi/2 \end{bmatrix}, \qquad \dot{\theta}_d = \begin{bmatrix} -\pi\sin(\frac{\pi t}{2}) \\ \pi\cos(\frac{\pi t}{2}) \\ 0 \end{bmatrix} \tag{13}$$

  Update your PD controller to also include $\dot{\theta}_d$. *Write down* and submit the equation for your modified control law. Make a simulation with your best performing gains, and title this video **Problem2_3.mp4**