Homework 2 (due 5pm Oct.14, in dropbox outside 3120A HFH)

2.1 Non-collocated Partial Feedback Linearization (PFL) control for the acrobot. In class, we looked at MATLAB simulation results for *collocated* PFL control of the acrobot (Figure 1), as outlined in Spong [2]. Download two m-files from the course homework website: `acrobot_collocated_linearization.m`, which solves the equations of motion (EOMs) for the system *with collocated PFL control*, and `acrobot_animate.m`, which will allow you to animate the motion. You can then run a simulation using the following MATLAB commands:

```
X0 = [-pi/2+.1;0;0;0]; % set an initial condition
[t,y] = ode45(@acrobot_collocated_linearization,[0 20],X0); % to simulate
figure(1)
acrobot_animate(t,y) % to animate
```

a) Modify `acrobot_collocated_linearization.m` to implement *non-collocated* PFL control, as outlined in [2]. (It is probably best to begin by copying this m-file to a *new* file called `acrobot_noncollocated_linearization.m`, so you have the old code to look at if you need to debug anything as you edit. In addition to [2], you may wish to reference the PFL class notes, handed out in class and also available for download from the class handouts website.)

(i) Include a print-out of your code in your homework.

(ii) Also include of a plot of the states over time, using X0 as given above.

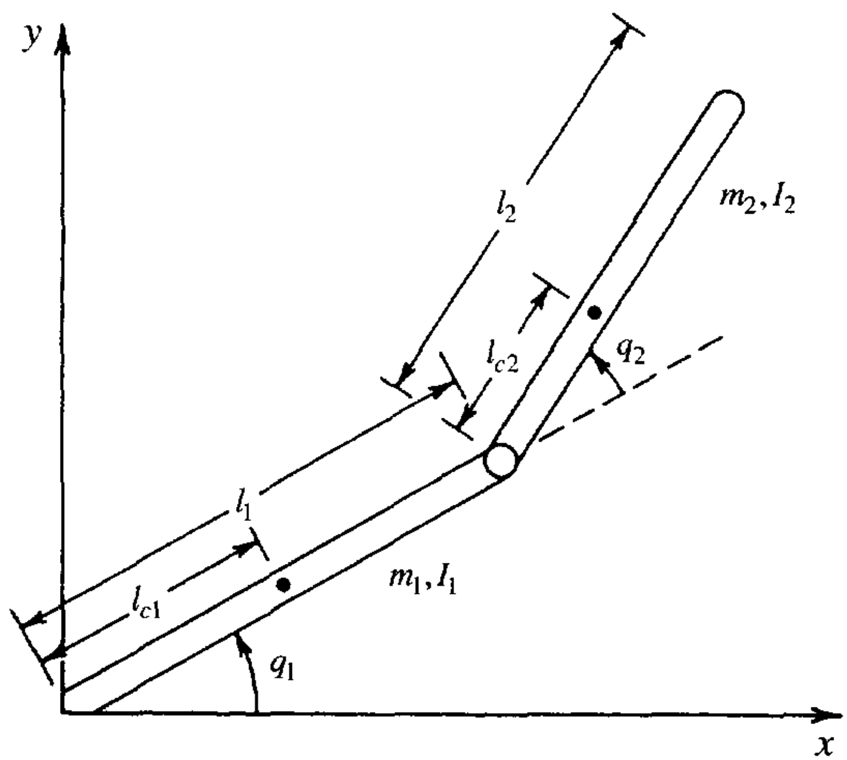

Figure 1: The acrobot (image taken from [2, 1]; torque input τ at elbow not shown).

2.2 Underpowered actuators. The torque required to achieve PFL control was not an issue addressed in [2] nor in class so far. Separate from the issue of being "underactuated" (where we cannot independently control every degree of freedom), the system dynamics may be noticeably "underpowered". Motors, for example, generally provide much lower torque and higher velocity than desired (which is why they are often geared down). However, gearing down a motor changes the actuator dynamics, making it challenging (if not impossible) to provide a pure "torque" output. So, we wish to use either a "direct drive" motor or very low transmission ratio (e.g., a belt drive increasing torque by a factor of 2-3 or so). All of which is to say: it is important to know whether our beautiful, theoretical control strategy can be implemented with real hardware!

a) Run an unaltered version of the m-file `acrobot_collocated_linearization.m` you downloaded in Problem 3.1 using the commands given below:

```
X0 = [-pi/2+.1;0;0;0]; % set an initial condition
[t,y] = ode45(@acrobot_collocated_linearization,[0 20],X0); % to simulate
tau = 0*t; % pre-allocate
for n=1:length(t)
   [dX,tau(n)] = acrobot_collocated_linearization(t(n),y(n,:));
end
figure(2); subplot(211); plot(t,tau); title('Collocated PFL');
xlabel('Time (s)'); ylabel('Torque (Nm)')
```

(i) What is the peak torque magnitude (positive or negative) required in a 10-second simulation? in a 20-second simulation?

(ii) Set a (generous) torque limit of 10 Nm by changing line 47 to the following:

```
torque_limit = 10; % [Nm] limit in torque magnitude
```

Animate the results. Does the first link of the acrobot still swing past vertical? How has the overall trajectory in state space changed (if at all) due to the torque limit? Explain (qualitatively) why or why not.

The goal of this entire problem is primarily to get you to think about the difference in the collocated and non-collocated control strategies.

b) Repeat this for your non-collocated m-file, plotting the results (with `torque_limit=1e12`, which is effectively no limit) on the lower half of the same figure:

```
X0 = [-pi/2+.1;0;0;0]; % set an initial condition
[t,y] = ode45(@acrobot_noncollocated_linearization,[0 20],X0); % to simulate
tau = 0*t; % pre-allocate
for n=1:length(t)
   [dX,tau(n)] = acrobot_noncollocated_linearization(t(n),y(n,:));
end
figure(2); subplot(212); plot(t,tau); title('Non-collocated PFL');
xlabel('Time (s)'); ylabel('Torque (Nm)')
```

(i) What is the peak torque in a 20-second simulation? (Use $K_p = 50$ and $K_d = 5$.)
(ii) Set a torque limit of 20 Nm in your non-collocated control code.
Animate the results. Does the first link of the acrobot still stabilize about a vertical position? Describe the difference between the animations with and without the torque limit.
(iii) Now, adjust the torque limit (up or down) until the system is "just barely" capable of stabilizing the first link in a near-vertical position. (The second link will continue to move in some way to provide this stabilization. These motions of the second link are, by the way, the "zero dynamics" of the system.) What is the smallest value for `torque_limit` that can still stabilize the first link near vertical? (*Please only calculate this value to within 5 Nm of accuracy!*)

c) Again working with your non-collocated control code, change the desired final position from $90°$ ($\pi/2$) to $80°$ ($80\pi/180$ radians).
(i) What is the magnitude of the peak torque for a 20-second simulation ($K_p = 50$, $K_d = 5$)?
(ii) How and why are the zero dynamics (second link motion) different than they were when the desired angle was $\pi/2$? (And why is the required torque so much higher now?)

2.3 Zero dynamics for collocated acrobot. As mentioned in class, the use of the arctan function in Equation (53) (as shown in Figure 4) in [2] is rather arbitrary. In fact, this particular choice results in zero dynamics result a third-order equation of motion for q_1, as shown in Equation (54). In Figure 2, below, is an alternative function. For $|\dot{q}_1| < 1$ (rad/s), $q_2^d = (\pi/3)\sin(\dot{q}_1\pi/2)$. For $|\dot{q}_1| \geq 1$ (rad/s), $q_2^d = (\pi/3)\,\text{sgn}\,(\dot{q}_1)$.

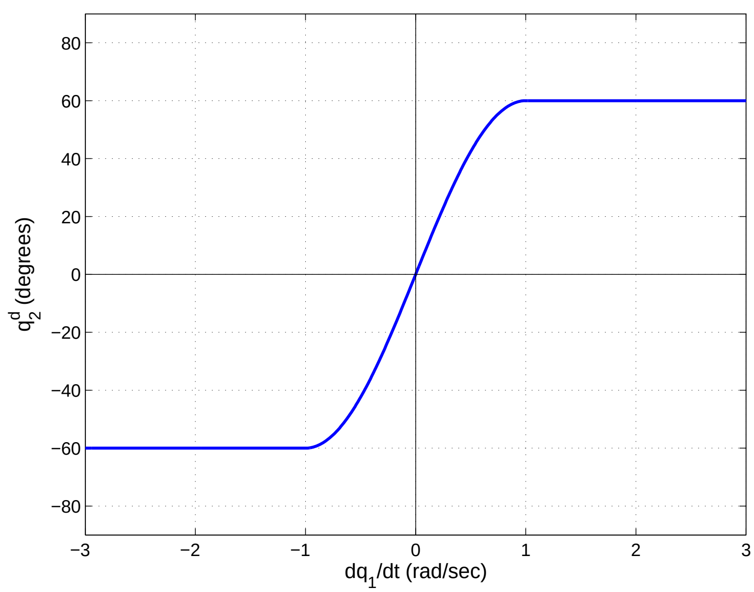

Figure 2: An alternative smooth function for $q_2^d(\dot{q}_1)$.

a) Modify `acrobot_collocated_linearization.m` to use the function shown in Figure 2, instead of the arctan function from [2]. Using the same initial condition, X0, from Problem 2.1, simulate 30 seconds of the dynamics from the original collocated controller (downloaded from the web) and from your newly modified code. Produce a phase plot of $\dot{q}_1$ vs q_1 for each set of data. (These plots should look similar to Figure 5 in [2].)

b) Assume that at steady state, $q_2 = q_2^d = \pi/3$, and $\dot{q}_2 = \ddot{q}_2 = 0$. Write the equation of motion for the resulting zero dynamics. (i.e., find the new EOM for q_1, which now replaces Eq. (54) from [2].)

2.4 Zero dynamics of a linear, 4th-order system. In this problem, you are to implement the feedback strategies outlined for PFL on a system that is already linear, shown below.

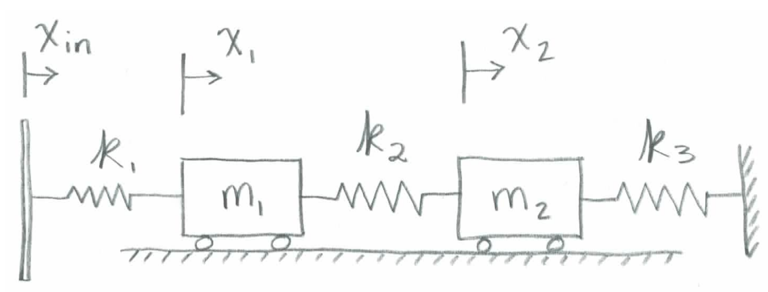

Figure 3: Fourth-order spring-mass system, for Problem 2.4.

a) What are the equations of motion for the system shown in Figure 3? (Assume that when $x_{in} = x_1 = x_2 = 0$, all of the springs are at their neutral, undeflected length. Also assume that x_{in} is a control input that we can set directly over time.)

b) Implement control to drive x_1 to zero, such that the dynamics of x_1 over time obey the following equation:

$$\ddot{x}_1 + 2\zeta\omega_n\dot{x}_1 + \omega_n^2 = 0$$

where $\omega = 2\pi$ and $\zeta = 0.7$. Turn in MATLAB code, along with plots of $x_1(t)$ and $x_2(t)$ given initial conditions $x_1(0) = 0.5$ (m), $x_2(0) = 0$, $\dot{x}_1(0) = 0$, $\dot{x}_2(0) = 0$. Use the following values for parameters: $m_1 = m_2 = 1$ (kg), $k_1 = k_2 = k_3 = 100$ (N/m).

c) What are the zero dynamics for your controller above? (Hint, the dynamics of x_1 evolve such that a steady state of $x_1 = \dot{x}_1 = \ddot{x}_1 = 0$ is approached over time...)

References

[1] Mark Spong. The swingup control problem for the acrobot. *IEEE Control Systems Magazine*, 15(1):49–55, February 1995.

[2] Mark W. Spong. Partial feedback linearization of underactuated mechanical systems. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 1, pages 314–321, September 1994.