Homework 3 (due 5pm 2/5, in HFH 5115)

3.1 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 [3]. 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 [3]. (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 [3], 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) Include a print-out of a plot of the states over time, e.g., from: `plot(t,y)`

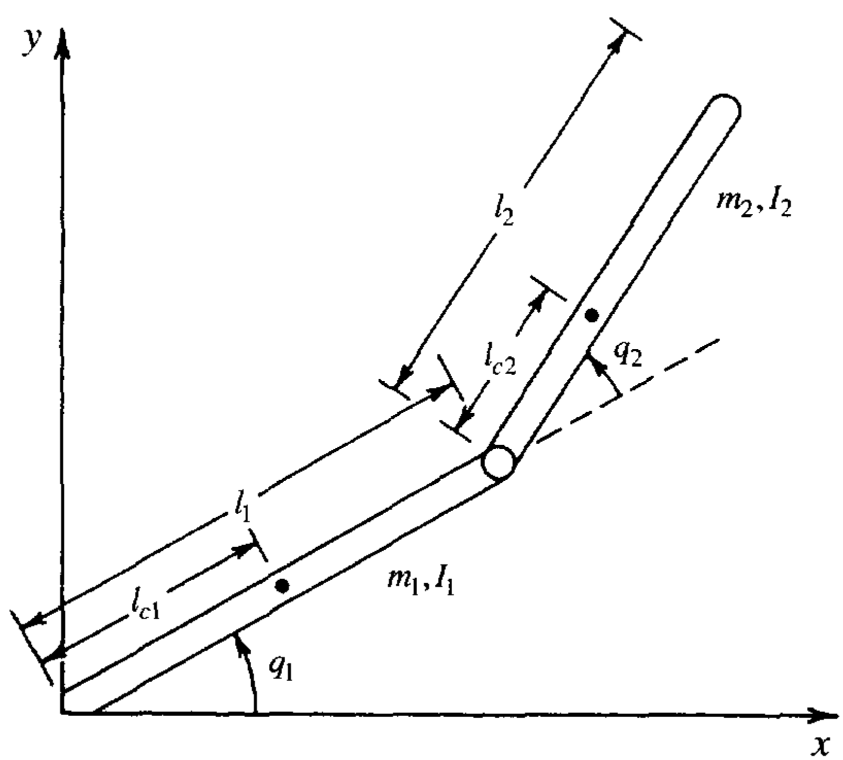

Figure 1: The acrobot (image taken from [2]; torque input τ at elbow not shown).

3.2 Underpowered actuators. The torque required to achieve PFL control was not an issue addressed in [3] 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?)

3.3 Zero-Moment Point (ZMP) for table with a moving mass. In this problem, we will calculate the ZMP for a table with a printer that has a moving mass (the printer head), using the "cart-table" model presented in [1]. (See also class notes on ZMP, available on the "notes" website for the course.)

a) We wish to modify the cart-table model to include the non-moving (static) mass of the table, M, whose center of mass remains at $x_T = 0$ for all time. Assume the moving mass is m and travels at a constant height z_c and is at some location $x(t)$. (For now, assume the support polygon of the table is "infinitely wide", so the ZMP could theoretically exist at any point on the ground.)

(i) Write an equation for the ZMP location, p_x, of the entire cart-tablel system, as a function of: M, m, x, $\ddot{x}$, z_c, and g.

(ii) Show that when $M = 0$, your equation reduces to equation 9 in [1], shown below:

$$p_x = x - \frac{z_c}{g}\ddot{x}$$

b) Assume the mass travels with the following trajectory over time:

$$x(t) = A\sin(\omega t)$$

where $A = 0.5$ meters, $\omega = 2\pi$ (i.e., 1 Hertz). Also, use the following constants: $M = 4$ kg, $m = 1$ kg, $z_c = 0.98$ meters, $g = 9.8$ m/s^2.

(i) How wide must the table base be in x to ensure the table does not begin to tip or rock? (That is, what is $(p_x)_{max} - (p_x)_{min}$?)

(ii) Now, assume that $M = 0$. How wide must the table be?

(iii) Let $M = 4$ kg again, but speed up the motion so $\omega = 8\pi$ (4 Hertz). How wide must the table be?

c) (i) Is/are there any non-trivial motions over time, $x(t)$, for which the steady state solution for the ZMP location *does not move*? ($x(t) = 0$ is the trivial solution and does not count here.) If so, provide an analytical solution (in terms of the variables). If not, why not? (Such a motion, $x(t)$, is then the "zero dynamics" for maintaining a desired, stationary Zero-Moment Point location.)

3.4 Zero-Moment Point (ZMP) trajectories over time. Assume the center of mass of a cart-table "robot" remains at a constant height of $z_c = 0.98$ meters and that it moves from $x = 0$ to $x = 1$ smoothly as shown in Figure 2, so that:

$$x = \begin{cases} 0 & \text{if } t < 0, \\ \frac{1}{2}\left(1 - \cos(\pi t)\right) & \text{if } 0 \leq t \leq 1, \\ 1 & \text{if } t > 1. \end{cases}$$

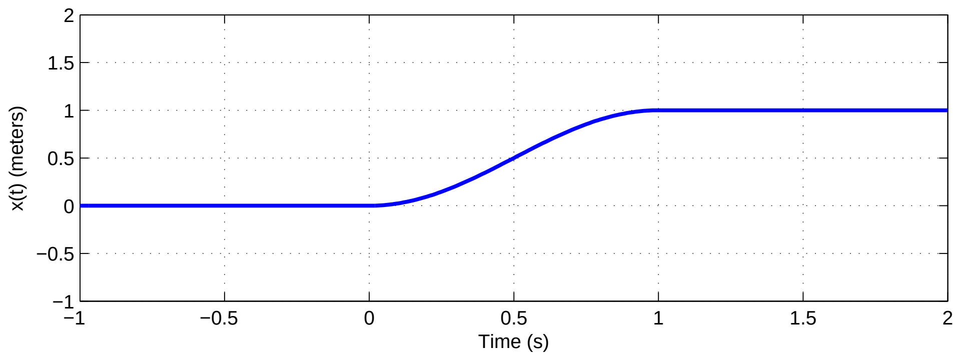

Figure 2: x(t) for Problem 4.

a) Plot (by hand) and label $\dot{x}(t)$ and $\ddot{x}(t)$.

b) Assume $x(t)$ is the motion of the cart in the cart-table model of [1], where the table has zero mass, and use the following constants: $z_c = 0.98$ [m], $g = 9.8$ [m/s^2]
Use equation 9 of [1]: $p_x = x - \frac{z_c}{g}\ddot{x}$

(i) What is the x location of the ZMP over time, $p_x(t)$?

(ii) Plot $p_x(t)$ (by hand).
Note that although both $x(t)$ and $\dot{x}(t)$ are continuous over time, p_x is not.

c) Now, assume an initial condition $x_o = 0$, $\dot{x}_o = 0$. For $t \geq 0$, assume the desired location of the ZMP is a constant location of $p_x = -0.1$ meters.

(i) Use MATLAB to solve for the approximate trajectory $x(t)$. Approximately what will $x(t = 1)$ be? What about $x(t = 2)$? And $x(t = 4)$?

References

[1] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiware, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *ICRA IEEE International Conference on Robotics and Automation*, pages 1620–1626. IEEE, Sep 2003.

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

[3] 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.