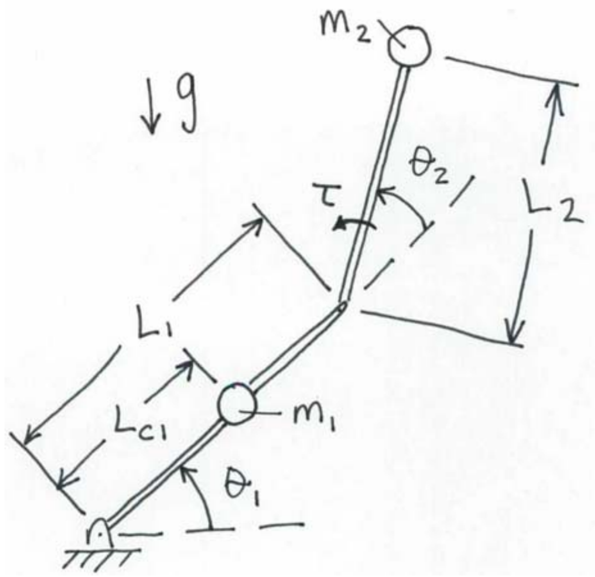
Homework 5

Problem 1 – Acrobot dynamics and swing-up control.
This problem involves the classic "acrobot" two-link system, shown below. **You must turn in your working m-files electronically** (email to katiebyl@gmail.com) for credit.



Acrobot:

In the figure at left, there are only two point masses (m1 and m2). The rest of each arm is assumed to be massless (zero inertia). There is a single torque, applied between the two links (at the "elbow"). Theta 2 is relative to theta 1.

a) Write a MATLAB m-file (script or function) to derive the equations of motion for the system above. (See Lecture 17 notes for general approach.) *Note that you may refer to the attached paper by Mark Spong to check your equations of motion. Your equations will not include the inertia terms that are included in Spong, however*.)

b) For the set of parameters listed below, solve for the state space matrices **A** and **B** for small deviations about the vertical equilibrium point where $\theta_1 = \frac{\pi}{2}$, and $\theta_2 = 0$.

```
L1 = 1;     L2 = 1;
Lc1 = .6;   Lc2 = L2;
m1 = 1;     m2 = 2;
g = 9.81;
```

Note that Spong uses a different set of parameters. However, you can plug in the Spong's parameters to check against A and B given in the paper. Also note that although the equilibrium point for the first angle is pi/2 (not zero), $\cos\left(\frac{\pi}{2} + \Delta\right) = -\sin(\Delta)$, for small delta, and A and B are calculated for deviations in states about equilibrium…

c) Assume the current state is: $X = [\theta_1 \quad \theta_2 \quad \dot{\theta}_1 \quad \dot{\theta}_2]^T = [1 \quad 1 \quad 1 \quad 1]^T$

1) Calculate the accelerations, $\ddot{\theta}_1$ and $\ddot{\theta}_2$, for an input torque $\tau = 0$.
2) Calculate the accelerations, $\ddot{\theta}_1$ and $\ddot{\theta}_2$, for $\tau = 1\ (Nm)$.
3) Calculate the required torque, τ, that will result in $\ddot{\theta}_1 = 0$.
4) Calculate the required torque, τ, that will result in $\ddot{\theta}_2 = 0$.

d) Write MATLAB code to simulate 20 seconds of the "collocated" stand-up controller from Spong, starting with the initial condition $X = [\theta_1 \quad \theta_2 \quad \dot{\theta}_1 \quad \dot{\theta}_2]^T = [-1 \quad 0 \quad 0 \quad 0]^T$. Suggested steps are as follows:

1) Using the approach/strategy in part c, write a MATLAB function that takes the current (4x1) state vector (`X`) and the desired $\ddot{\theta}_2$ (`d2th2_des`) as INPUTS, calculates the required torque to achieve this (through "partial feedback linearization") and then OUTPUTS the derivative of the state vector: `dX` $= \dot{X} = [\dot{\theta}_1 \quad \dot{\theta}_2 \quad \ddot{\theta}_1 \quad \ddot{\theta}_2]^T$. (Only $\ddot{\theta}_1$ must be actually be calculated to create this output vector, since $\dot{\theta}_1$ and $\dot{\theta}_2$ are states, and $\ddot{\theta}_2$ must match the desired input…)

2) Write a MATLAB function that takes time, `t`, and the current state vector, `X`, as INPUTS and gives the derivative of the states, `dX`, as its OUTPUT, as described below. In fact, you will ignore the time input, but we will still include it so that the function syntax matches what is required for you to use the function "ode45" for simulation. This m-file should be named "`twolink_swingup.m`" and should have the following text on its first line:

```
function dX = twolink_swingup(t,X)
```

This function will calculate a desired angle for theta 2 that is between -30 and 30 degrees, depending on the velocity of the first link, as shown below:

$$\theta_2^{des} = \left(\frac{30\pi}{180}\right) \text{atan}\,(2\dot{\theta}_1)$$

Then, it will calculate the desired acceleration of the second link, to drive it toward this desired angle, using the following PD control law:

$$\ddot{\theta}_2^{des} = K_p\left(\theta_2^{des} - \theta_2\right) + K_d(0 - \dot{\theta}_2)$$

Here, use `Kp=20`, `Kd=5`. Finally, you will call the function you wrote in part d-1 to calculate dX, given the current state, X, and a desired acceleration for the second link, $\ddot{\theta}_2^{des}$, as inputs.

3) Simulate the dynamics for 20 seconds, using ode45, using the code below:

```
tspan = [0 20];
Xinit = [-1 0 0 0]';
[tout,xout] = ode45(@twolink_swingup, tspan, Xinit);
twolink_ani(tout,xout)
```

(The function `twolink_ani.m` is provided on the homework website and will animate the acrobot motions over time, give inputs `tout` and `xout` from running MATLAB's `ode45`. The mat-file `twolink_data.mat` includes example `tout` and `xout` data that you can use as inputs to see how the command `twolink_ani(tout,xout)` is supposed to work.)