MeEn 537 Homework #6

1. Use the newly provided "puma560_torque_profile.mat" which is for the mdl_puma560 arm (instead of Baxter since the Baxter model had significant numerical issues) to calculate $\ddot{q}, \dot{q}, q$ for the entire torque profile. With regards to this, do the following:

   (a) Use the "p560_startup_script.m" and simulink model provided with this document to calculate the joint variables $\ddot{q}, \dot{q}, q$ (this should be straight forward, but I encourage you to look carefully at the model and code to make sure you understand it). To access the output of the simulink simulation, you'll need to do this "out.get('q_sim')" for q_sim, qd_sim, qdd_sim and t_sim. For this simulation, we are including viscous friction, but neglecting Coulomb friction. This is the purpose of line 8 in "p560_startup_script.m".

   (b) Use ode45 (or even first-order Euler integration) with the "accel" function that is a function of the robot model "p560" to get $\ddot{q}, \dot{q}, q$.

   (c) Check to see if both methods give the same values or not and report your findings. They should at least give very similar results. Turn in plots of results for both methods and code for part b).

2. For the 3 link RRR robot from HW 5, make the robot in the robotics toolbox (including adding the minimal necesary dynamic parameters - assume 1 kg masses for each link, rotational inertias I_{zz} of 0.01, and the COM at the geometric center of each link) and then do the following and turn in the associated code:

   (a) calculate the required torques for $q = [\frac{\pi}{4}, \frac{\pi}{4}, \frac{\pi}{4}]$, $\dot{q} = [\frac{\pi}{6}, -\frac{\pi}{4}, \frac{\pi}{3}]$,$\ddot{q} = [-\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{6}]$, using the recursive Newton Euler formulation we learned about in class

   (b) calcaulate the same torques using the toolbox function "rne" to see if it agrees with your calculations

   (c) now that you have code from part a), use it to calculate just the inertia matrix $D(q)$ for this same configuration and compare it to the "inertia" function from the toolbox.

3. After having loaded the MATLAB Robotics toolbox, run the following "help CodeGenerator" and carefully read the documentation. We will not use this in our class, but you should know it exists. Especially note in the "Notes" section where it says their methods may fail for robots with greater than 3 links. Sympybotics is another option for doing the same thing where we have used it for at least up to 7 degrees of freedom with no problems and very fast evaluation times (microseconds). By Friday November 18th, we'll look at some example sympybotics code and I'll post it for your future reference. Although we won't use it directly in this class, I encourage you to at least look at it.