MeEn 537 Homework #5

1. Work the following problems (from Spong, Robot Modeling and Control):

   (a) Find the moments of inertia and cross products of inertia of a uniform rectangular solid with sides a, b, and c with respect to a coordinate frame with the origin at one corner and the axes along the edges of the solid.

   (b) Show that the inertia matrix M(q) for an n-link robot is always positive definite. (Look at the definition of a positive definite matrix on wikipedia if needed. The form of Kinetic energy for a n-link robot may help you here.)

   (c) Consider a 3-link Cartesian manipulator and do the following:

      i. Compute the inertia tensor I_i for each link $i = 1, 2, 3$ assuming that the links are uniform rectangular solids of length 1, width 1/4, height 1/4, and mass of 1.

      ii. Compute the 3x3 inertia matrix M(q) for this manipulator

      iii. Show that the Christoffel symbols c_{ijk} are all zero for this robot. Interpret the meaning of this for the dynamic equations of motion.

      iv. Derive the equations of motion in matrix form:

      $$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = (\tau \ or \ f)$$

   (d) Use symbolic software (like the "syms" command in MATLAB) to derive the Euler-Lagrange equations of motion for a planar three-link RRR with each link length being 0.4m. It is recommended that you use the robotics toolbox, but not the dynamic functions in the toolbox.

2. Use the dynamic model of Baxter in the toolbox to do the following:

   (a) Given the file "desired_accel.mat" calculate the joint torques τ necessary to cause this acceleration. Turn in plots of the joint torques for each joint.

3. Use the provided "puma560_torque_profile.mat" which is for the mdl_puma560 arm 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 simuilink 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 each of the following variable names - 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).