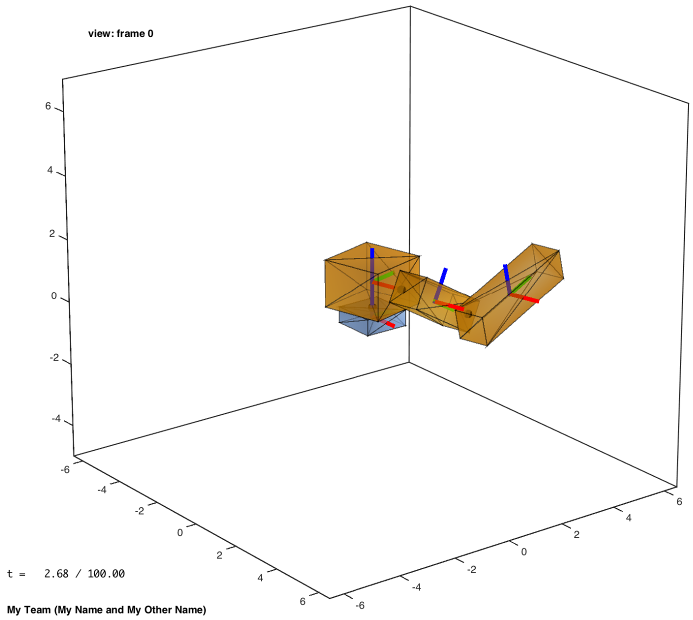
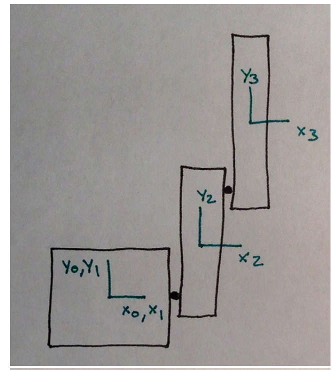
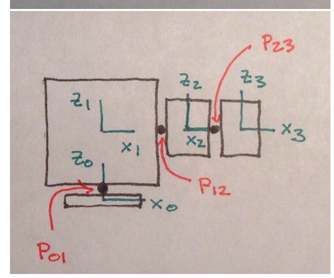
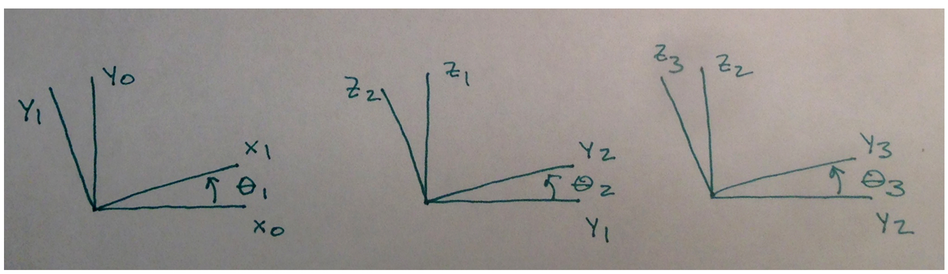
AE352 Homework #6:
More Systems of Rigid Bodies: Robot Arm

(due at the beginning of class on Wednesday, October 21)









The goal this week is to simulate the motion of the three-link robot arm shown above left (orange), which is mounted to a fixed base (blue). To do so, you will be adding code to the MATLAB script `hw6.m`, available on the course website. Groups of lines are labeled "must change" (for things like implementing coordinate transformations and finding rates of change), "can change" (for things like specifying initial conditions or making movies), and "can't change" (for things that happen behind the scenes). Much of the code will be familiar to you from HW1-HW5.

The base of the robot arm is shaped like a box, and so is each link. Frame 0 is fixed to the base. Frames 1, 2, and 3 are fixed to links 1, 2, and 3, respectively, with each origin at the center of mass. Each link rotates about a single axis that is fixed in the previous link (or in the base). In particular, link 1 rotates about z_0, link 2 rotates about x_1, and link 3 rotates about x_2. Assuming all frames are initially aligned, we call θ_1 the angle frame 1 has rotated about z_0, θ_2 the angle frame 2 has rotated about x_1, and θ_3 the angle frame 3 has rotated about x_2. The point at which link 0 is attached to link 1 is p_{01}. The point at which link 1 is attached to link 2 is p_{12}. The point at which link 2 is attached to link 3 is p_{23}. The pictures should make these relationships clear.

In the code, you'll also see the variables `robot.a1`, `robot.b1`, and so forth. These variables are used to describe the relative position of frames, as follows:

$$o_1^0 = a_1 + R_1^0 b_1$$
$$o_2^1 = a_2 + R_2^1 b_2$$
$$o_3^2 = a_3 + R_3^2 b_3$$

In particular, the parameters a_1, b_1, etc., are all constant. They do not vary with time (although the rotation matrices do, of course).

The robot arm is subject to three sources of force and torque:

- The force of gravity on link 1, written in the coordinates of frame 0, is

$$-m_1 g z_0^0$$

where m_1 is the mass of link 1 in kg, $g = 9.81$ m/s^2, and

$$z_0^0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

This force acts at the center of mass of link 1. Note that, as a consequence, this force results in no torque about the center of mass. It should be clear how to express the force of gravity on each of the other two links.

- The force applied to link 1 by link 0 (the base) through the revolute joint, written in the coordinates of frame 0, is something I recommend you call f_{01}^0. It acts at the point p_{01}. It is a constraint force, that needs to be solved for. As usual, there is an equal and opposite force applied to link 0 by link 1. It should be clear how to express the force on each of the other two links.

- The torque applied to link 1 by link 0 (the base) through the revolute joint, written in the coordinates of frame 0, is something I recommend you call τ_{01}^0. Like last week, the torque is partly due to the constraint and partly due to the motor. This week, we'll also model friction. In particular, I recommend you write

$$\tau_{01}^0 = t_{01}(u_1 - k_{\text{friction}}\dot{\theta}_1) + S_{01} r_{01}$$

where u_1 is the motor torque, k_{friction} is a constant (assumed the same for each joint), r_{01} is the part of the torque due to the constraint (an unknown, which needs to be solved for), and

$$t_{01} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad S_{01} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

As usual, there is an equal and opposite torque applied to link 0 by link 1. It should be clear how to express the torque on each of the other two links.

You can take exactly the same approach to modeling this system as you took last week for the spacecraft. The robot arm is, again, just a system of rigid bodies that are subject to constraints.

Please submit the following things. Like last week, you may—but are not required to—work in pairs. If you choose to work with a partner, please submit *one copy of your assignment* with both your and your partner's name on the front page.

1. (120 pts) You must derive expressions for the following things by hand, including any diagrams necessary to complete these derivations:

   - position and orientation of each link

   $$o_1^0, R_1^0, o_2^0, R_2^0 \text{ (also requires } R_2^1), o_3^0, R_3^0 \text{ (also requires } R_3^2)$$

   - linear and angular velocity of each link

   $$v_{0,1}^0, w_{0,1}^1, v_{0,2}^0, w_{0,2}^2 \text{ (also requires } w_{1,2}^2), v_{0,3}^0, w_{0,3}^3 \text{ (also requires } w_{2,3}^3)$$

   - linear and angular acceleration of each link (these will get lengthy—I strongly recommend that you group terms as much as possible, and that you immediately separate terms that involve the unknowns $\ddot{\theta}_1$, $\ddot{\theta}_2$, and $\ddot{\theta}_3$ from the other terms)

   $$\dot{v}_{0,1}^0, \dot{w}_{0,1}^1, \dot{v}_{0,1}^0, \dot{w}_{0,2}^2, \dot{v}_{0,3}^0, \dot{w}_{0,3}^3$$

   - mass and moment of inertia of each link

   $$m_1, J_1^1, m_2, J_2^2, m_3, J_3^3,$$

   - constraint forces and torques

   $$f_{01}^0, \tau_{01}^0, f_{12}^1, \tau_{12}^1, f_{23}^2, \tau_{23}^2$$

   - equations of motion
     (a) write both Newton's Equation and Euler's Equation for each link separately
     (b) plug in for the linear and angular accelerations ($\dot{v}_{0,1}^0, \dot{w}_{0,1}^1, \dot{v}_{0,1}^0, \dot{w}_{0,2}^2, \dot{v}_{0,3}^0, \dot{w}_{0,3}^3$) and for the torques ($\tau_{01}^0, \tau_{12}^1, \tau_{23}^2$)
     (c) put unknowns on left-hand-side and knowns on right-hand-side
     (d) write in matrix form as $F\gamma = h$ where γ is a column matrix of unknowns, so you can solve easily in MATLAB as $\gamma = F^{-1}h$ (note that I switched from "g" to "γ" this week because "g" is now being used to denote the acceleration of gravity)

2. (60 pts) You must implement everything marked "must change" in `hw6.m`. (You may, of course, also play around with anything marked "can change.") Submit a print-out *only* of the lines of code that are changed, in the order in which these lines appear in `hw6.m`.

3. (60 pts) You must choose a task that is of interest to you and complete it. Examples of "a task" will be forthcoming. You must submit the following things:

   - A description of what you wanted to accomplish and why.
   - A movie showing the results you were able to achieve, submitted as in HW4-HW5.

   If you believe that you were not successful in doing what you wanted (e.g., if you encountered a coding error that you could not resolve), then please attempt to describe what went wrong in your video. If you do this then you can still receive full credit for this part of the assignment.

Like last week, it is important that you start this assignment right away. You should proceed carefully, methodically, and peacefully through each part of the derivation and implementation.

Examples of "a task"

Here are five examples of "a task" that you might choose to accomplish.

- *Gravity compensation.* Suppose all joint velocities are initially zero. Derive an expression for the joint torques u_1, u_2, and u_3 that would need to be applied in order for all joint velocities to stay zero (i.e., for the robot to remain motionless), as a function of the initial joint angles. You might consider if this is always possible, for any set of joint angles. Create a composite video that shows results for several different choices of joint angles. You might consider—and describe in your video—the way in which "gravity compensation" is useful for a real robot.

- *Dynamic equilibrium.* Suppose you choose $u_1 \neq 0$ and $u_2 = u_3 = 0$. You will likely observe that $\dot{\theta}_1$ converges to a constant value over time, and that θ_2 and θ_3 also converge to constant values. You might explore—either through analysis, experiment, or both—how these constant values depend on the choice of u_1. You might consider whether or not constant values will always be reached, for any choice of u_1. You might consider what happens if the coefficient of friction (k_{friction}) is increased or decreased, or even set to zero. You might consider—and describe in your video—what these results mean.

- *Swing-up.* Suppose the arm begins hanging motionless, pointing straight down. Suppose you want the arm to end motionless, pointing straight up. See if you can do it while keeping each motor torque between -50 and 50. You will likely want to change the increment of torque that is caused by each keypress—to do so, search for "`du=1;`" and replace it with "`du=25;`" or whatever you want (the effect should be obvious). For a real challenge, see if you can do this task using only one of the three joint torques. You might consider—and describe in your video—why such a maneuver might be useful in practice.

- *Balance.* Suppose the arm begins motionless, pointing almost straight up, but not quite. Without the application of any torque, you'll see that the arm will fall. Apply torque to balance it, so that it does not fall, for as long as you can. You may wish to change the increment of torque that is caused by each keypress—to do so, search for "`du=1;`" and replace it with "`du=5;`" or whatever you want (the effect should be obvious). For a real challenge, see if you can do this task using only one of the three joint torques. You might consider—and describe in your video—why such a maneuver might be useful in practice.

- *Chaos.* Choose some initial set of joint angles. Simulate the system for some fixed length of time (e.g., 10 seconds). Choose another initial set of joint angles that is very close to, but not exactly the same as, the original set of joint angles. Simulate the system again, for the same fixed length of time. What happens? Something very similar, or something very different? Does your answer depend on your choice of the original set of joint angles? You might consider—and describe in your video—what these results mean.

You may, of course, choose any other task of similar scope that is of more interest to you instead.