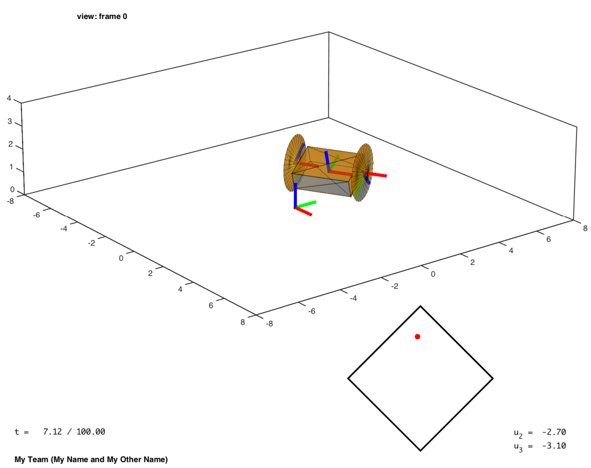
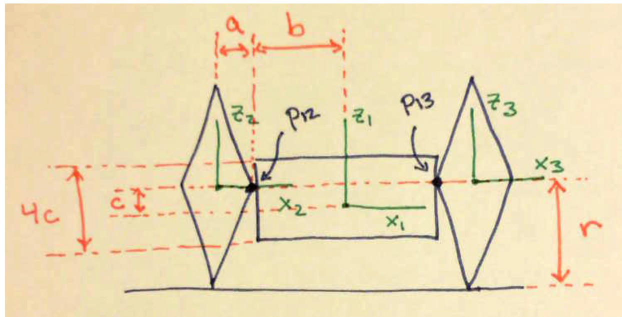
AE352 Homework #7: Rolling Contact Between Rigid Bodies: Segway

(due at the beginning of class on Friday, November 6)





The goal this week is to simulate the motion of the two-wheeled robot shown above left, which resembles a "segway." To do so, you will be adding code to the MATLAB script `hw7.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-HW6. One difference this week is that every function ends with an "`end`" statement. If you create one or more functions yourself, you'll need to end them with an "`end`" statement as well.

The chassis of the robot is shaped like a box. Each wheel is shaped like two cones glued together. Frame 0 is fixed to the ground. Frame 1 is fixed to the chassis, with its origin at the center of mass. Frames 2 and 3 are fixed to wheels 2 (left) and 3 (right), respectively, with each origin at the center of mass. Each wheel rotates about an axis parallel to x_1. The points at which the chassis is attached to wheel 2 and wheel 3 are p_{12} and p_{13}, respectively. It should be clear from the picture shown above right that:

$$p_{12}^1 = \begin{bmatrix} -b \\ 0 \\ c \end{bmatrix} \qquad p_{13}^1 = \begin{bmatrix} b \\ 0 \\ c \end{bmatrix} \qquad o_2^1 = \begin{bmatrix} -a-b \\ 0 \\ c \end{bmatrix} \qquad o_3^1 = \begin{bmatrix} a+b \\ 0 \\ c \end{bmatrix}$$

These parameters are all constant, and are all given to you in the code. Other useful parameters are also given, for example the mass and moment of inertia of everything (m_1, J_1^1, etc.) and the gravity force vector written in the coordinates of frame 0:

$$g^0 = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix}.$$

As usual, see `GetGeometryOfRobot` for a list of these parameters.

The robot is subject to four sources of force and torque:

- The force of gravity. This force acts on the chassis and on each wheel. It should be clear how to express this force in each case, in terms of g^0.
- The forces applied by the chassis to each wheel through the revolute joints, denoted by f_{12}^1 and f_{13}^1, acting at p_{12} and p_{13} respectively. Equal and opposite forces act on the chassis.
- The torques applied by the chassis to each wheel through the revolute joints, denoted by τ_{12}^1 and τ_{13}^1. Like last week, the torque is due to the constraint, the motor, and viscous friction. For the left wheel, I recommend you write

$$\tau_{12}^1 = t_{12}\left(u_2 - k\left(\text{“blah”}\right)\right) + S_{12}r_{12}$$

  for appropriate constants t_{12} and S_{12}, where "blah" is an expression for the angular velocity of the wheel relative to the chassis about the joint axis (i.e., it is the projection of $w_{1,2}^2$ onto the x_1 axis). It should be clear how to express the torque on the other wheel. Equal and opposite torques act on the chassis.
- The forces applied by the ground to each wheel, assuming that friction is sufficiently high that both wheels roll without slipping all the time. Denote these forces, written in the coordinates of frame 0, by f_{02}^0 and f_{03}^0. The z_0 component of each force is the "normal force" that keeps each wheel from falling through the ground. The x_0 and y_0 components of each force make up the "tangential force" or "friction force" that keeps each wheel from slipping.

You can take almost exactly the same approach to modeling this system as you took in the last two weeks. The robot is, again, just a collection of three rigid bodies that are subject to constraints. The only change is that we're going to use MATLAB to eliminate not only constraint forces and torques, but also redundant configuration variables (as we discussed in class on Wednesday, October 28). In particular, we're going to describe the position and orientation of the chassis and of each wheel separately—as (o_1^0, R_1^0), (o_2^0, R_2^0), and (o_3^0, R_3^0), where each rotation matrix is parameterized by a separate ZYX Euler Angle sequence[1]. There will be a total of 34 unknowns:

- the linear and angular acceleration of each frame (18 unknowns)

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

- the constraint forces and torques associated with the revolute joints (10 unknowns)

$$f_{12}^1, \tau_{12}^1, f_{13}^1, \tau_{13}^1$$

- the constraint forces associated with the rolling contact (6 unknowns)

$$f_{02}^0, f_{03}^0$$

You will derive a total of 34 equations:

- Newton's Equation and Euler's Equation for each body (18 equations)

[1] Yes, I said "ZYX" and not "XYZ." You will have to re-derive an expression for the rotation matrix produced by a ZYX sequence, and for the relationship between the angular velocity and angular rates.

- the constraint on the relative position of each wheel with respect to the chassis due to the revolute joint—to derive this constraint, write an expression for o_2^1 and o_3^1, and take the time derivative of each expression twice (6 equations)
- the constraint on the relative orientation of each wheel with respect to the chassis due to the revolute joint—to derive this constraint, write an expression for $S_{12}^T w_{1,2}^2$ and $S_{13}^T w_{1,3}^3$, and take the time derivative of each expression (4 equations)
- the constraint that each wheel rolls without slipping—we will discuss how to derive this constraint in class (5 equations… it may surprise you that there are only 5, not 6)
- the constraint that "friction forces don't squeeze"—we will discuss what this constraint means, and how to derive it, in class (1 equation… it may surprise you that we need this "extra" equation—think about the fact that there are two points of contact between the robot and the ground, and what this means about indeterminate forces)

This may sound like a lot of equations, but it's not really, and you'll see that each equation is very simple. You should find the paperwork easier this week than in either of the last two weeks.

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 the equations of motion and write them in matrix form as $F\gamma = h$, where γ is a column matrix of unknowns, so you can solve easily in MATLAB as $\gamma = F^{-1}h$.
2. (60 pts) You must implement everything marked "must change" in `hw7.m`. (You may, of course, also play around with anything marked "can change.") Submit your entire code online—details of the submission process will be forthcoming.
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-HW6.

   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 and methodically through each part of the derivation and implementation.

Remember how important it is to be clear and organized in your approach to the derivation, so that you can compare your work to others. If you are working with a partner, both of you should do the derivation, and check to see if you agree. If you are working alone, it will still benefit you to check your work with others.