Homework 3:
Flying Box

MEAM 520, University of Pennsylvania
Katherine J. Kuchenbecker, Ph.D.

September 12, 2013

This assignment is due on **Friday, September 20, by midnight (11:59:59 p.m.)** Your code should be submitted via email according to the instructions at the end of this document. Late submissions will be accepted until Sunday, September 22, by midnight (11:59:59 p.m.), but they will be penalized by 10% for each partial or full day late, up to 20%. After the late deadline, no further assignments may be submitted.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you write down should be your own work, not copied from any other individual or team. Any submissions suspected of violating Penn's Code of Academic Integrity will be reported to the Office of Student Conduct. If you get stuck, post a question on Piazza or go to office hours!

Individual vs. Pair Programming

You may do this assignment either individually or with a partner, according to your personal preference. Read the assignment to decide which option is right for you. If you do this homework with a partner, you may work with anyone you choose, even someone with substantial MATLAB experience. If you are looking for a partner, consider using the "Search for Teammates!" tool on Piazza.

If you are in a pair, you should work closely with your partner throughout this assignment, following the paradigm of pair programming. You will turn in one MATLAB script for which you are both jointly responsible, and you will both receive the same grade. Please follow these pair programming guidelines, which were adapted from "All I really need to know about pair programming I learned in kindergarten," by Williams and Kessler, *Communications of the ACM*, May 2000:

- Start with a good attitude, setting aside any skepticism and expecting to jell with your partner.
- Don't start writing code alone. Arrange a meeting with your partner as soon as you can.
- Use just one computer, and sit side by side; a desktop computer with a large monitor is better for this than a laptop. Make sure both partners can see the screen.
- At each instant, one partner should be driving (using the mouse and keyboard or recording design ideas) while the other is continuously reviewing the work (thinking and making suggestions).
- Change driving/reviewing roles at least every thirty minutes, *even if one partner is much more experienced than the other.* You may want to set a timer to help you remember to switch.
- If you notice a bug in the code your partner is typing, wait until they finish the line to correct them.
- Stay focused and on-task the whole time you are working together.
- Recognize that pair programming usually takes more effort than programming alone, but it produces better code, deeper learning, and a more positive experience for the participants.
- Take a break periodically to refresh your perspective.
- Share responsibility for your project; avoid blaming either partner for challenges you run into.

Animating the Motion of a Flying Box

Your task for this homework is to update a provided MATLAB script so that it animates the motion of a rectangular block that was moved in a specific way. The flying block was captured on video, and its positions and orientations were recorded over time using a Ascension TrakStar magnetic motion tracking system that includes a sensor located inside the block.

A. Watch the movie of the flying block: `http://www.youtube.com/watch?v=FTC83piKiX0`

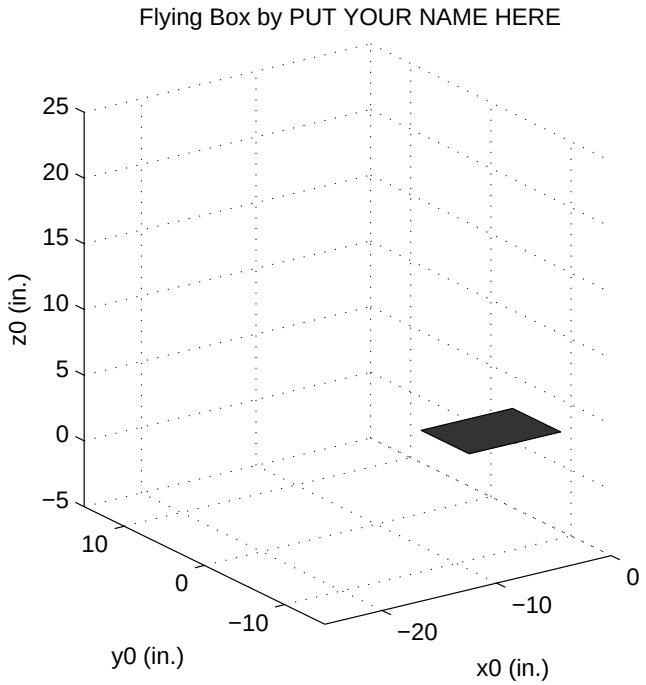
B. Download the following items from the Course Page / Resources / Homework section of our Piazza site.

- A script called `flying_box_starter.m` to get you started on this assignment. It loads the data into the workspace, defines the animation parameters, sets up a `for` loop to process the data, and graphs a flat gray rectangle that represents the box.
- A MATLAB data file called `flying_box.mat`. This file contains the time history of the position and orientation of the flying box recorded during the movie.
- The movie `flying_box.mov`, in case you need to download it for offline viewing.

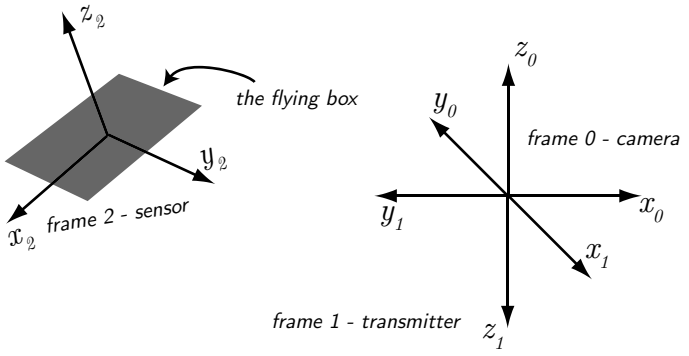
C. Rename the starter file to `flying_box_yourpennkey.m` or `flying_box_pennkey1_pennkey2.m` so that everyone will have a unique file name. Your PennKey is the first part of your Penn email address.

D. Set the value of the `studentNames` variable at the top of the file to include your full name or the full names of both teammates.

E. Run the code and watch what happens. As provided, the first step of the animation should look like the plot below, and then the back corner of the box should move straight up so you can see movement.



F. Read about the frames defined in the code comments and look at the diagram below.



G. Update the code between the two lines of stars to calculate the coordinates of the four corners of the box in the frame of the camera (frame 0). You should start with the provided coordinates of these corners in the sensor's frame (`pa2`, `pb2`, `pc2`, and `pd2`), and you should store your four final answers as `pa0`, `pb0`, `pc0`, and `pd0`.

Important: Your calculations may not use any built-in or downloaded functions dealing with rotation matrices, homogeneous transformations, Euler angles, roll/pitch/yaw angles, or related topics. Instead, you must type all your calculations yourself, using only low-level functions such as `sind`, `cosd`, and vector/matrix math.

The variables you have to work with inside the loop are `x`, `y`, `z`, `a`, `e`, and `r`, which were recorded by the magnetic tracking system. These variables give you the position and orientation of the sensor (frame 2) in the frame of the transmitter (frame 1). Positions are measured in inches. The manufacturer's documentation states that `a` is the azimuth angle in degrees, denoting a rotation about the z axis. Similarly, `e` is the elevation angle in degrees, a rotation around the y axis, and `r` is the roll angle in degrees, a rotation around the x axis.

Unfortunately, the TrakStar documentation is ambiguous about whether these three rotations should be made about the sensor's axes or the transmitter's axes, and it isn't clear about the order in which they should be performed. Your job is to figure out what order the rotations should be applied. **You should not modify the angles themselves (by adding constants or changing the sign). Instead, you should change the order in which you apply the transformations.**

Use the motion you can see in the video to check the output of your code. When you get the transformation correct, the animated gray box will move just like the box in the video. If you can't get it perfect, do the best you can, and submit what you have for partial credit.

Submitting Your Code

Follow these instructions to submit your code:

1. Start an email to `meam520@seas.upenn.edu`
2. Make the subject *Homework 3: Your Name* or *Homework 3: Your Name and Your Teammate's Name*, replacing *Your Name* and *Your Teammate's Name* with the appropriate full names.
3. Attach your correctly named MATLAB script to the email. Please do not put it in a zip file or include any other attachments.
4. Optionally include any comments you have about this assignment and the experience of pair programming if you worked with a teammate.
5. Send the email.