Homework 1:
Rigid Motions and Homogeneous Transformations

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

September 11, 2012

This assignment is due on Tuesday, September 18, by 5:00 p.m. sharp. You should aim to turn the paper part in during class that day. If you don't finish until later in the day, you can turn it in to Professor Kuchenbecker's office, Towne 224. The code must be emailed according to the instructions at the end of this document. Late submissions of either or both parts will be accepted until 5:00 p.m. on Wednesday, but they will be penalized by 25%. After that 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 a peer or a solution manual.

Book Problems *(30 points)*

The first set of problems is from the textbook, *Robot Modeling and Control* by Spong, Hutchinson, and Vidyasagar (SHV). Please follow the extra clarifications and instructions when provided. Write in pencil, show your work clearly, $\boxed{\text{box your answers}}$, and staple together all pages of your assignment.

1. SHV 2-10, page 66 – Sequence of Rotations *(5 points)*
   Please specify each element of each matrix in symbolic form and show the order in which the matrices should be multiplied; as stated in the problem, you do not need to perform the matrix multiplication.

2. SHV 2-14, page 67 – Rotating a Coordinate Frame *(5 points)*
   Sketch the initial, intermediate, and final frames by reading the text in the problem. Then find R in two ways: by inspection of your sketch and by calculation. Check your solutions against one another.

3. SHV 2-23, page 68 – Axis/Angle Representation *(10 points)*

4. SHV 2-39, page 70 – Homogeneous Transformations *(10 points)*
   Treat frame $o_2x_2y_2z_2$ as being located at the center of the cube's bottom surface (as drawn in Figure 2.14), not at the center of the cube (as stated in the problem).

MATLAB Programming *(30 points)*

This class will use MATLAB to analyze and simulate robotic systems and also to control real robots. While Professor Kuchenbecker loves MATLAB, she recognizes that it can be difficult to use at the start. Even if you don't like MATLAB now, please give it a chance, and come to office hours or contact the teaching team if you feel lost or frustrated.

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

A. If you're not yet confident in your MATLAB skills, you might want to complete the *Interactive MATLAB Tutorial* provided online by The MathWorks:
`http://www.mathworks.com/academia/student_center/tutorials/mltutorial_launchpad.html`

B. Get access to a computer that has MATLAB installed. All of the computers in the SEAS computer labs will work, or you can remotely connect to a SEAS computer lab PC by following these instructions: `http://www.seas.upenn.edu/cets/answers/virtualLab.html` You can also purchase your own copy of MATLAB at the Penn Computer Connection for about $100.

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

D. Download the following items from Homework 1 on the Timeline section on the MEAM 520 Lore website.

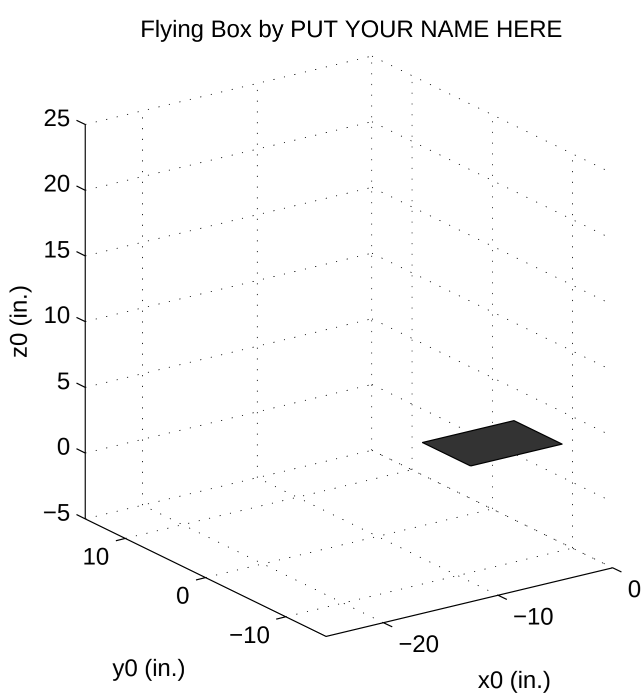
- A script called `flying_box_starter.m` to get you started on this assignment. It loads the data into the workspace, defines the animation parameters, sets up a `for` loop to process the data, and graphs a flat gray rectangle that represents the box.
- A MATLAB data file called `flying_box.mat`. This 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.

E. Rename the starter file to `flying_box_yourpennkey.m` so that everyone will have a unique file name. Your PennKey is the first part of you Penn email address. Note that this is not the same as the number on your PennCard.

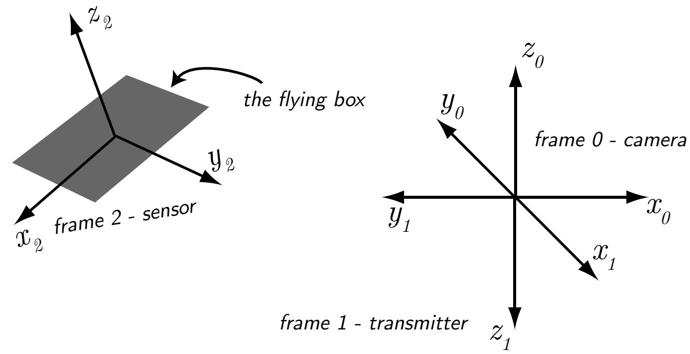
F. Put your name at the top of the file where indicated in the comments.

G. Put your name in the title of the created graph by modifying the command at the bottom.

H. 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.



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



J. 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. Your job is to figure out which it is, and in what order the rotations should be applied. 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 1: Your Name*, replacing *Your Name* with your name.
3. Attach your correctly named MATLAB script to the email.
4. Optionally include any comments you have about this assignment.
5. Send the email.