MEAM 520

Rotation Matrices

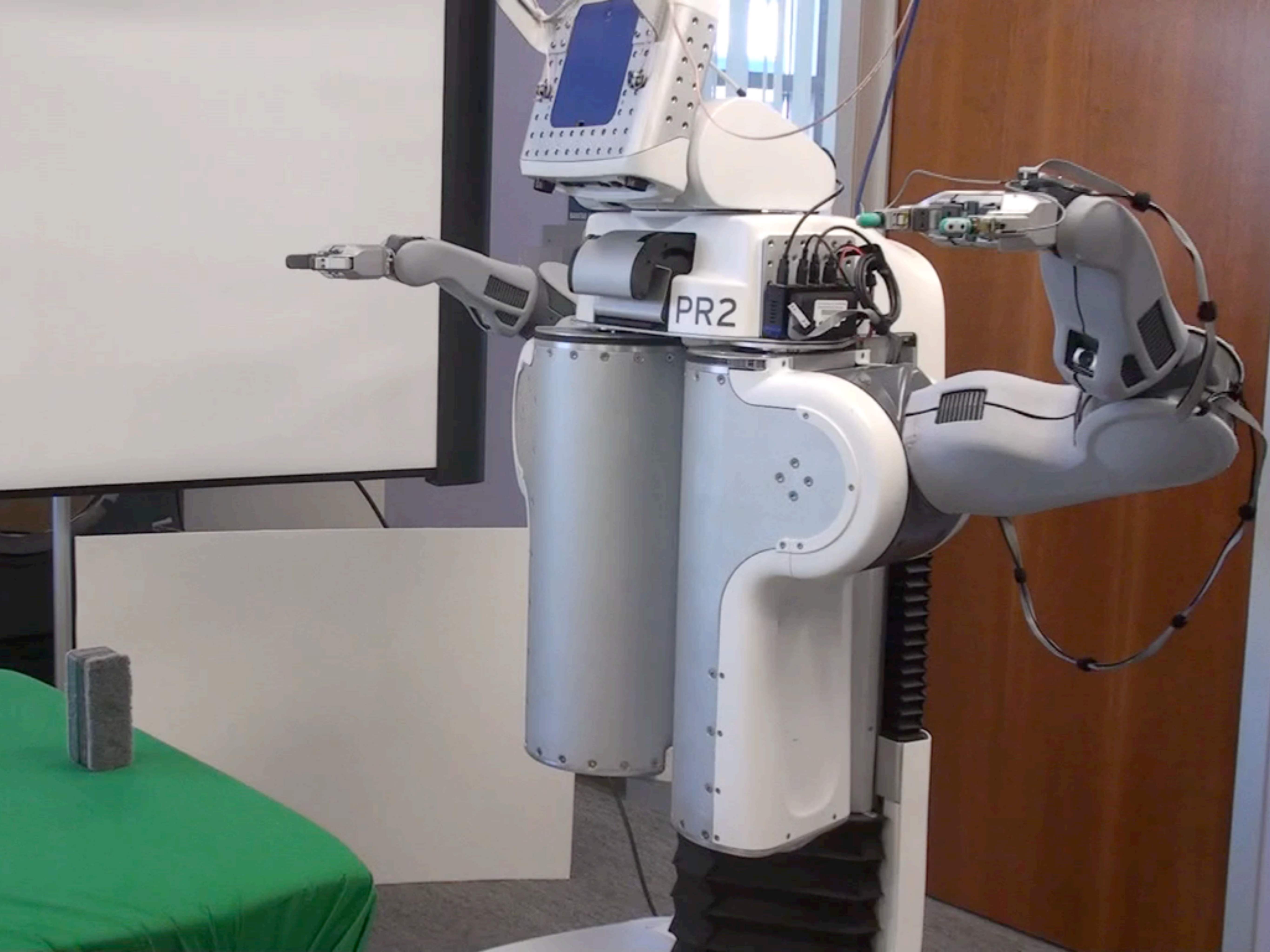
Katherine J. Kuchenbecker, Ph.D.

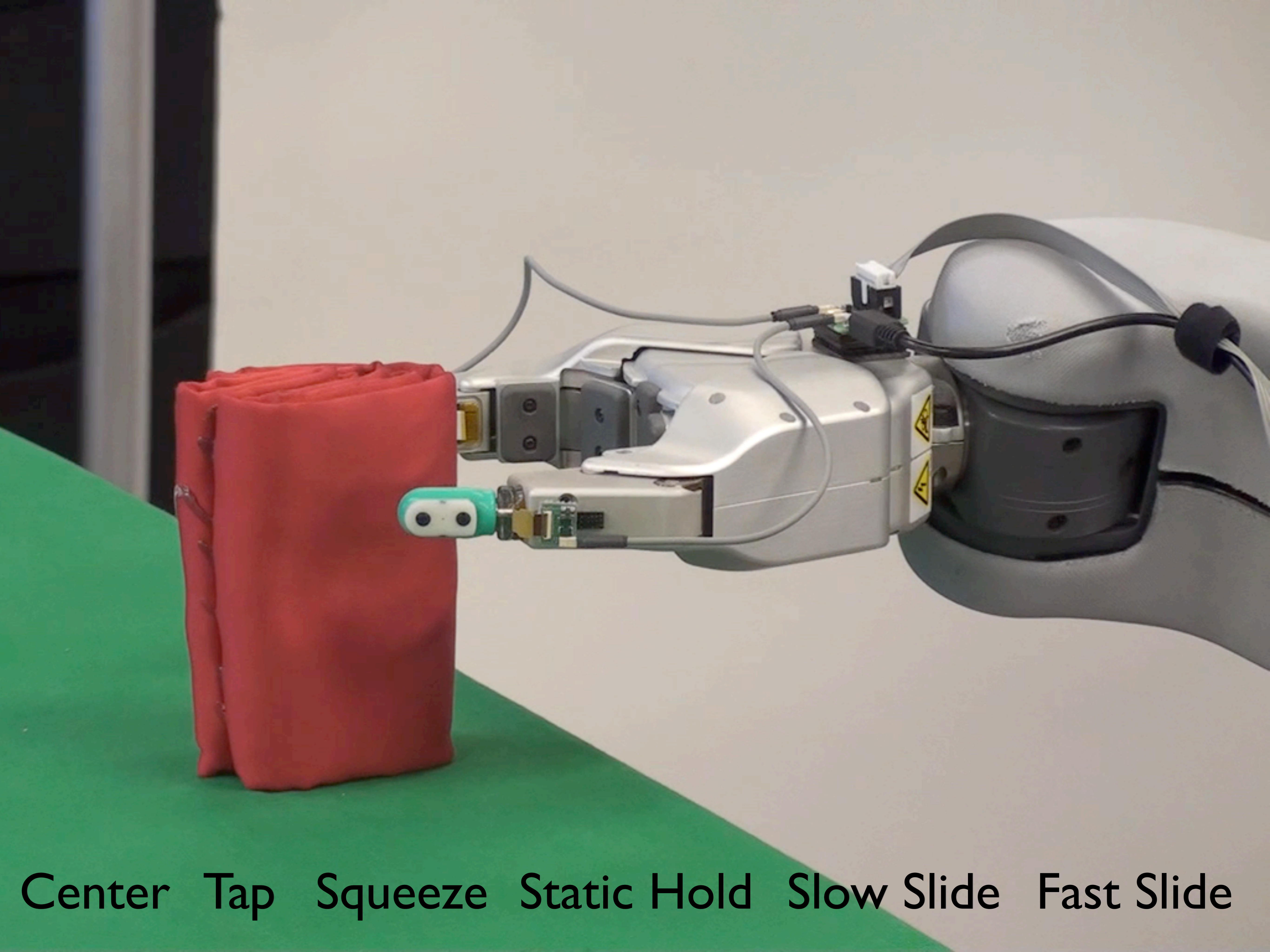
General Robotics, Automation, Sensing, and Perception Lab (GRASP)

MEAM Department, SEAS, University of Pennsylvania

Lecture 2: September 11, 2012

Last week I was at UC Berkeley in California
for a DARPA BOLT Activity E Evaluation on
“Perceptually Grounded Robotic Language Acquisition”


PR2


Center
Tap
Squeeze
Static Hold
Slow Slide
Fast Slide

Course Logistics

- Main points were explained during Lecture 1
- Those slides are posted on Lore
- Class website is hosted on Lore: join.lore.com/8GMYVB
- Barring technical glitches, lecture slides and audio will be recorded and posted to Blackboard.
- There was a glitch during Lecture 1. Hopefully today's attempt is more successful.
- You should have the book, and ideally you should have read pages1-19.
- What questions do you have?

How do you contact me?

Email: kuchenbe@seas.upenn.edu

Office: 224 Towne Building

Office Phone: (215) 573-2786

Office Hours: Tuesday 1:30 – 2:30 p.m.
Thursday 1:30 – 2:30 p.m.

Caveat

This is my first time teaching this class.

I sat through MEAM 520 in Spring 2012 and will largely be following the way Dr. Fiene taught the course.

I will do my best, but things may not be perfectly organized; we will all be learning and adapting together.

We’re going to need some math.

This week

Chapter 2 in SHV

Rigid Motions and Homogeneous Transformations

Appendix B in SHV

Linear Algebra

Key Linear Algebra Concepts

vector

transpose operator

scalar product (dot product) between two vectors

norm (length) of a vector

matrix

matrix multiplication

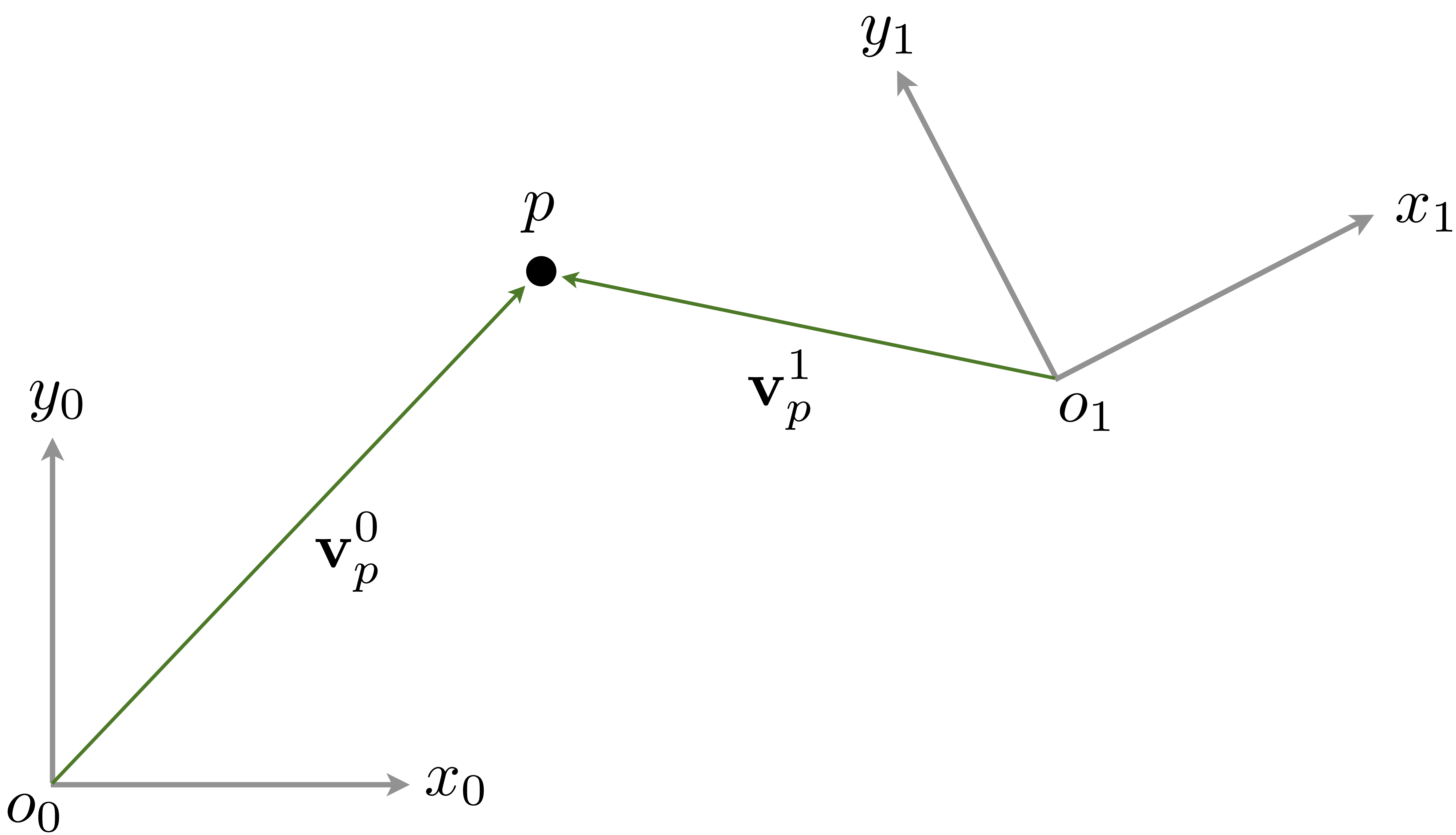
Questions ?

Representing Positions

The following slides are adapted from those created by Jonathan Fiene for MEAM 520 in Spring 2012

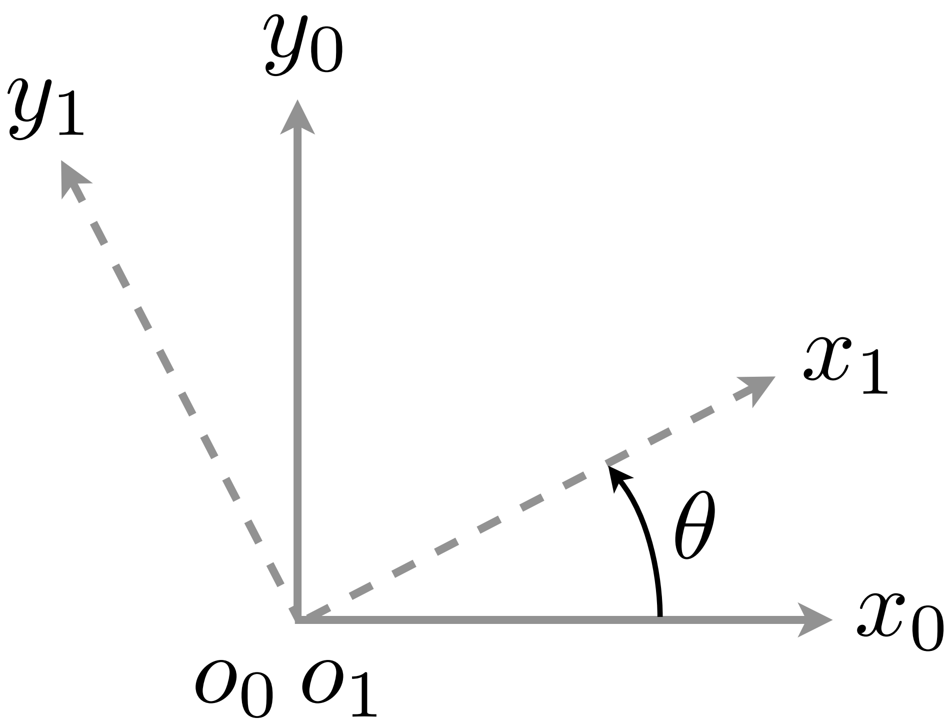
Rotation Matrices

The reference frame is designated using **superscript** notation



to perform algebraic manipulation,
vectors must be expressed in the **same** frame or in **parallel** frames

Planar Coordinate Rotations



project frame 1 into frame 0

$$\mathbf{x}_1^0 = \begin{bmatrix} x_1 \cdot x_0 \\ x_1 \cdot y_0 \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

$$\mathbf{y}_1^0 = \begin{bmatrix} y_1 \cdot x_0 \\ y_1 \cdot y_0 \end{bmatrix} = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

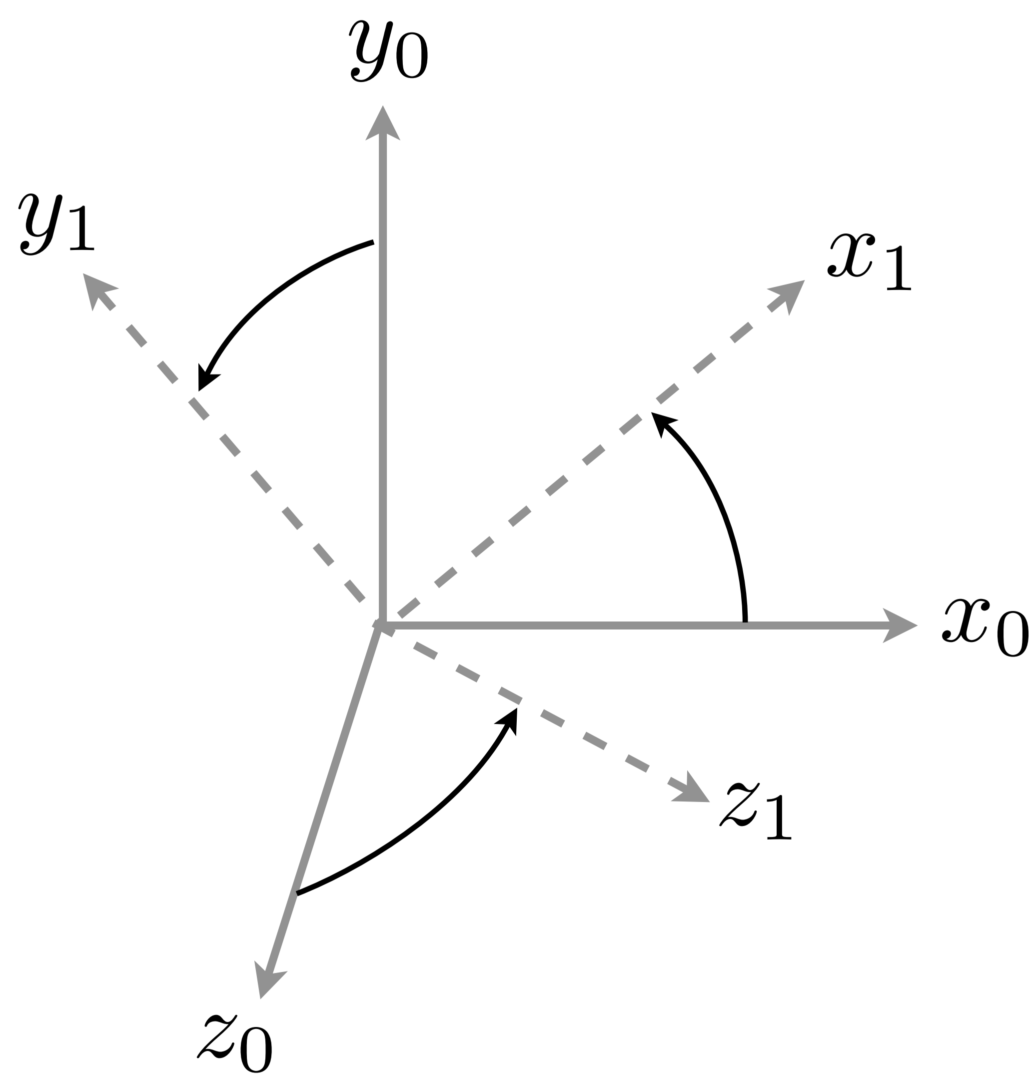
which can be expressed as a **rotation matrix**

$$\mathbf{R}_1^0 = \begin{bmatrix} \mathbf{x}_1^0 & \mathbf{y}_1^0 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

the inverse of which is the matrix transpose

$$\mathbf{R}_0^1 = (\mathbf{R}_1^0)^\top = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Three-Dimensional Coordinate Rotations



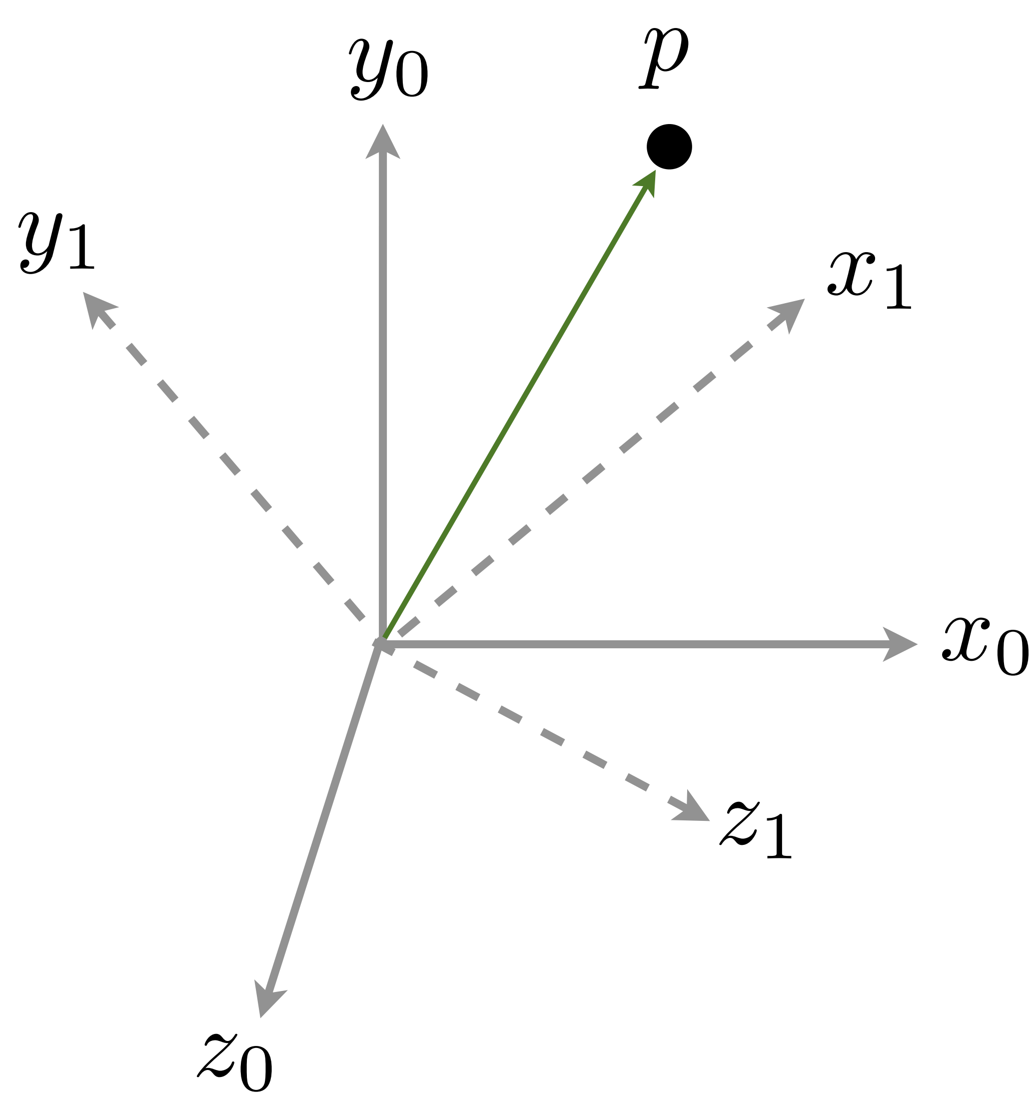
$$\mathbf{R}_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}$$

$$\mathbf{R}_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$\mathbf{R}_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

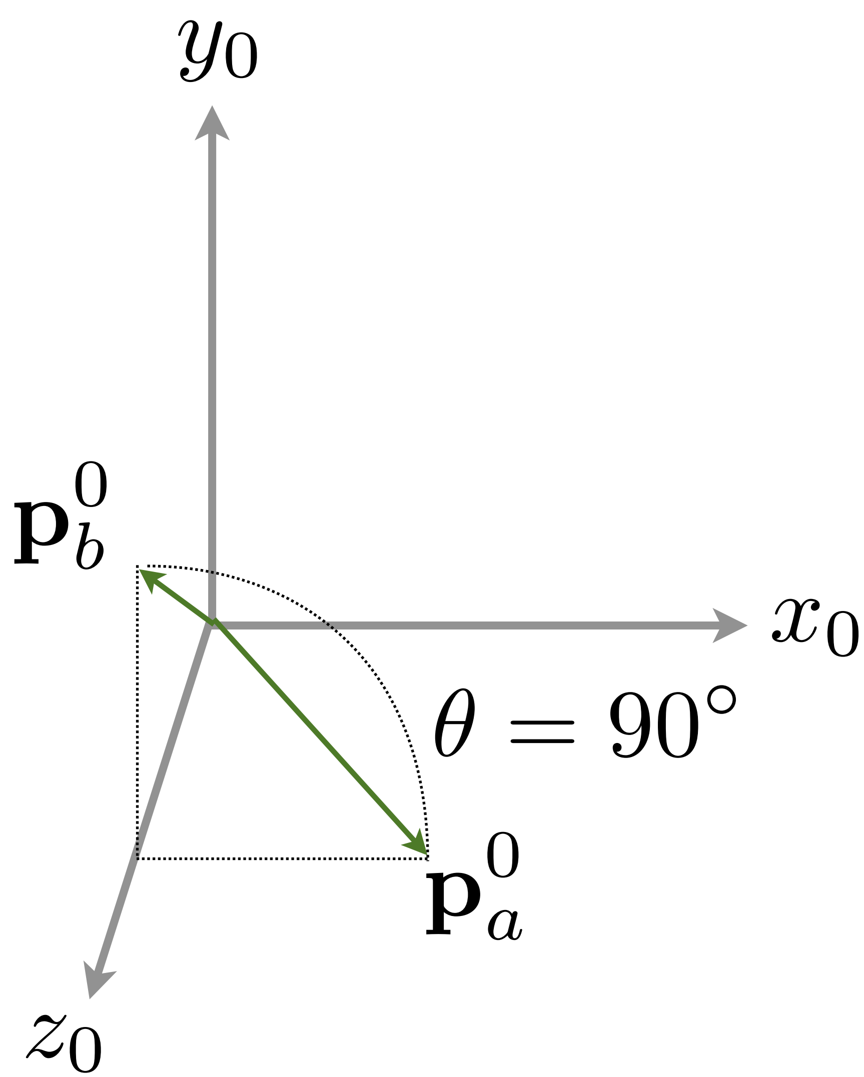
$$\mathbf{R}_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The **basic rotation matrices** define rotations about the three coordinate axes



For pure coordinate rotation, a point in frame 1 can be expressed in frame 0 using the rotation matrix

$$\mathbf{v}_p^0 = \mathbf{R}_1^0\ \mathbf{v}_p^1$$



The rotation matrix can also be used to perform rotations on vectors

$$\mathbf{p}_a^0 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{p}_b^0 = \mathbf{R}_{z,\theta}\ \mathbf{p}_a^0 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Rotation matrices serve three purposes (p. 47 in SHV):

1. Coordinate transformation relating the coordinates of a point p in two different frames
2. Orientation of a transformed coordinate frame with respect to a fixed frame
3. Operator taking a vector and rotating it to yield a new vector in the same coordinate frame.

Homework 1

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, 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)*

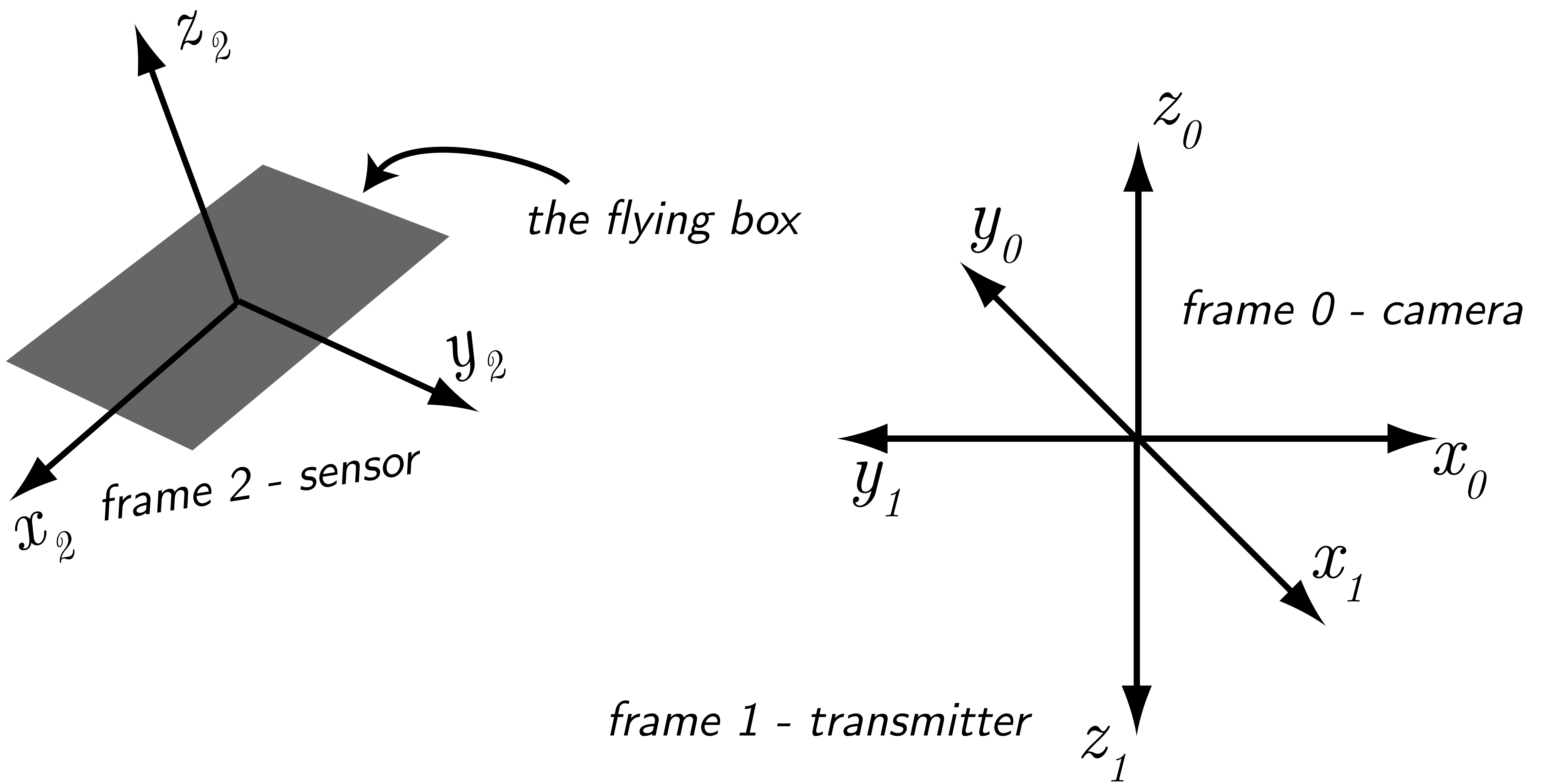
This class will use MATLAB to analyze and simulate robotic systems and also to control real robots. While Professor Kuchenbecker loves MATLAB, she recognizes that it can be difficult to use at the start. Even if you don't like MATLAB now, please give it a chance, and come to office hours or contact the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

flying_box.txt

Format for each sensor is: status x, y, z, azimuth, elevation, roll, button, quality, time

Sensor1:	0x0000	0.312	8.886	-15.579	85.703	2.694	-173.770	0	4	1347328434.211
Sensor1:	0x0000	0.312	8.881	-15.579	85.703	2.687	-173.784	0	4	1347328434.228
Sensor1:	0x0000	0.312	8.873	-15.579	85.696	2.673	-173.798	0	4	1347328434.244
Sensor1:	0x0000	0.312	8.868	-15.579	85.703	2.645	-173.820	0	4	1347328434.261
Sensor1:	0x0000	0.312	8.864	-15.583	85.718	2.624	-173.841	0	4	1347328434.277
Sensor1:	0x0000	0.312	8.859	-15.583	85.746	2.603	-173.862	0	4	1347328434.294
Sensor1:	0x0000	0.312	8.855	-15.587	85.788	2.589	-173.897	0	4	1347328434.315
Sensor1:	0x0000	-0.237	7.163	-13.179	85.729	0.776	-174.592	0	3	1347328468.754
Sensor1:	0x0000	-0.237	7.163	-13.175	85.729	0.776	-174.592	0	3	1347328468.763
Sensor1:	0x0000	-0.233	7.163	-13.175	85.729	0.783	-174.592	0	3	1347328468.775
Sensor1:	0x0000	-0.233	7.163	-13.175	85.729	0.783	-174.592	0	3	1347328468.783
Sensor1:	0x0000	-0.233	7.163	-13.175	85.729	0.790	-174.599	0	3	1347328468.800
Sensor1:	0x0000	-0.229	7.163	-13.175	85.729	0.797	-174.606	0	3	1347328468.817
Sensor1:	0x0000	-0.229	7.163	-13.175	85.729	0.804	-174.606	0	3	1347328468.833
Sensor1:	0x0000	-0.224	7.159	-13.175	85.722	0.811	-174.613	0	3	1347328468.850
Sensor1:	0x0000	-0.220	7.154	-13.179	85.721	0.825	-174.613	0	3	1347328468.867
Sensor1:	0x0000	-0.215	7.150	-13.179	85.728	0.846	-174.627	0	3	1347328468.883
Sensor1:	0x0000	-0.211	7.146	-13.184	85.736	0.860	-174.641	0	3	1347328468.900
Sensor1:	0x0000	-0.207	7.137	-13.188	85.764	0.867	-174.662	0	3	1347328468.917
Sensor1:	0x0000	-0.202	7.132	-13.188	85.778	0.874	-174.676	0	3	1347328468.933
Sensor1:	0x0000	-0.202	7.128	-13.192	85.792	0.874	-174.683	0	3	1347328468.950
Sensor1:	0x0000	-0.198	7.124	-13.192	85.799	0.874	-174.690	0	3	1347328468.967
Sensor1:	0x0000	-0.198	7.119	-13.192	85.813	0.874	-174.697	0	3	1347328468.983
Sensor1:	0x0000	-0.198	7.115	-13.197	85.820	0.867	-174.704	0	3	1347328469.000
Sensor1:	0x0000	-0.193	7.110	-13.197	85.827	0.860	-174.711	0	3	1347328469.017
Sensor1:	0x0000	-0.189	7.106	-13.201	85.834	0.846	-174.711	0	3	1347328469.033
Sensor1:	0x0000	-0.189	7.102	-13.201	85.841	0.825	-174.711	0	3	1347328469.050
Sensor1:	0x0000	-0.185	7.097	-13.201	85.855	0.804	-174.711	0	3	1347328469.067
Sensor1:	0x0000	-0.185	7.093	-13.201	85.869	0.783	-174.697	0	3	1347328469.083
Sensor1:	0x0000	-0.185	7.088	-13.201	85.876	0.762	-174.690	0	3	1347328469.100
Sensor1:	0x0000	-0.185	7.088	-13.201	85.883	0.748	-174.683	0	3	1347328469.117
Sensor1:	0x0000	-0.185	7.088	-13.201	85.883	0.741	-174.676	0	3	1347328469.133
Sensor1:	0x0000	-0.189	7.088	-13.201	85.876	0.734	-174.669	0	3	1347328469.150
Sensor1:	0x0000	-0.189	7.088	-13.201	85.869	0.727	-174.662	0	3	1347328469.167
Sensor1:	0x0000	-0.189	7.088	-13.201	85.862	0.720	-174.655	0	3	1347328469.183
Sensor1:	0x0000	-0.189	7.088	-13.201	85.848	0.706	-174.655	0	3	1347328469.200
Sensor1:	0x0000	-0.193	7.088	-13.201	85.813	0.692	-174.655	0	3	1347328469.217
Sensor1:	0x0000	-0.193	7.093	-13.206	85.715	0.650	-174.655	0	3	1347328469.233
Sensor1:	0x0000	-0.198	7.097	-13.210	85.497	0.581	-174.635	0	3	1347328469.250
Sensor1:	0x0000	-0.193	7.106	-13.219	85.062	0.490	-174.592	0	3	1347328469.267
Sensor1:	0x0000	-0.180	7.110	-13.228	84.437	0.427	-174.536	0	3	1347328469.283
Sensor1:	0x0000	-0.154	7.110	-13.236	83.650	0.399	-174.536	0	3	1347328469.300
Sensor1:	0x0000	-0.114	7.106	-13.254	82.409	0.399	-174.614	0	4	1347328469.321
Sensor1:	0x0000	-0.083	7.102	-13.267	81.540	0.420	-174.677	0	4	1347328469.333
Sensor1:	0x0000	-0.022	7.088	-13.285	79.896	0.476	-174.761	0	4	1347328469.354



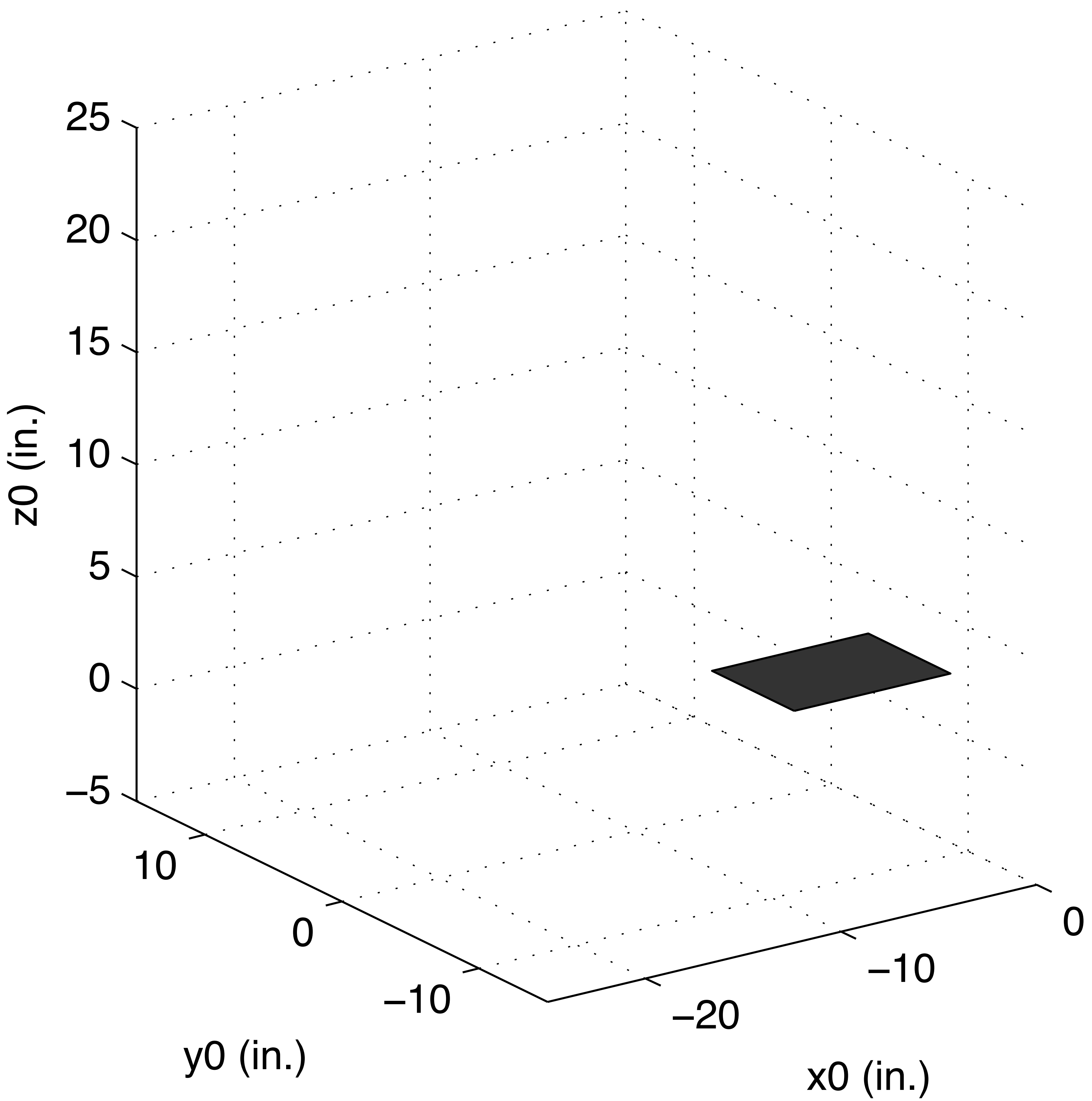
z_2
the flying box
y_2
x_2
frame 2 - sensor
z_0
y_0
frame 0 - camera
y_1
x_0
x_1
frame 1 - transmitter
z_1

Editor - /Users/kuchenbe/Documents/teaching/meam 520/assignments/01 transformations/matlab/flying_box_starter.m

File Edit Text Go Cell Tools Debug Desktop Window Help

```matlab
%% flying_box_starter.m
%
% This Matlab script provides the starter code for the flying box
% problem on Homework 1 in MEAM 520 at the University of Pennsylvania.
% The original was written by Professor Katherine J. Kuchenbecker in
% September of 2012. Students will modify this code to create their own
% script. Email kuchenbe@seas.upenn.edu with questions.
%
% Student Name:
%
% Change the name of this file to replace "starter" with your PennKey.  For
% example, Professor Kuchenbecker's script would be visualize_kuchenbe.m

%% SETUP

% Delete all variables from our workspace.
clear

% Load the TrakStar data recorded during the movie.
% This MATLAB data file includes time histories of the x, y, and z
% coordinates in inches, as well as time histories of the azimuth,
% elevation, and roll angles in degrees.
load flying_box;

% Open figure 1 and clear it to get ready for plotting.
figure(1)
clf

%% DEFINITIONS

% We need to keep track of three frames in this code.
%
% Frame 0 is the frame of the camera's view, with x positive to the right,
% y positive straight back, and z positive up.  The is the base frame, and
% it's what we plot in.  Its origin coincides with the origin of frame 1.
%
% Frame 1 is the frame of the TrakStar transmitter, which sits on the desk.
% It has x positive straight out, y positive to the left, and z positive
```

Flying Box by PUT YOUR NAME HERE



MATLAB

Session 1: Thursday, 9/13: 6-8pm

Session 1: Saturday, 9/15: 3-5pm

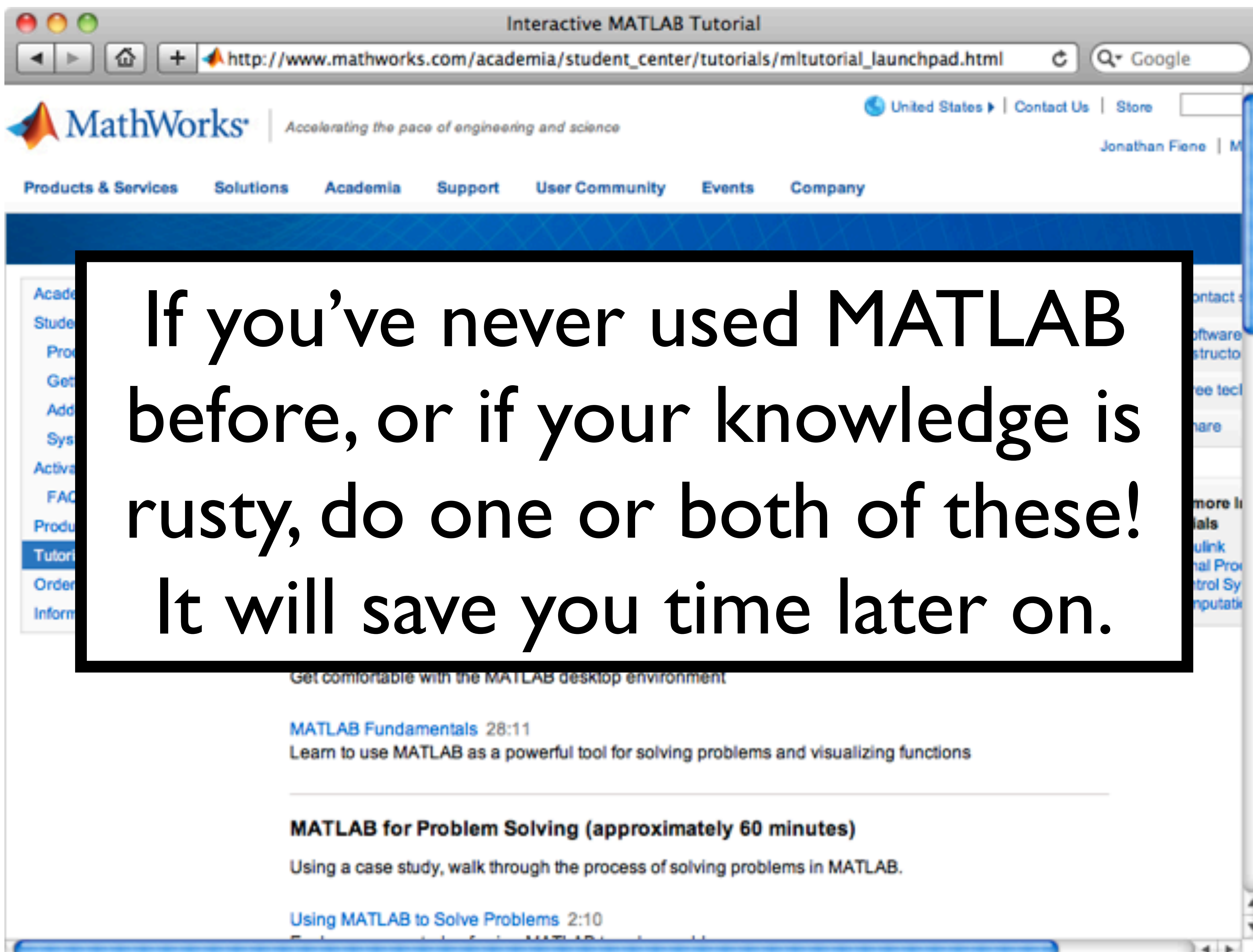
Session 2: Thursday, 9/20: 6-8pm

Session 2: Saturday, 9/22: 3-5pm

email jmarcus@seas.upenn.edu to sign up

MATLAB

Free online tutorial: http://www.mathworks.com/academia/student_center/tutorials/mltutorial_launchpad.html



If you've never used MATLAB before, or if your knowledge is rusty, do one or both of these! It will save you time later on.

Questions ❓