MEAM 520

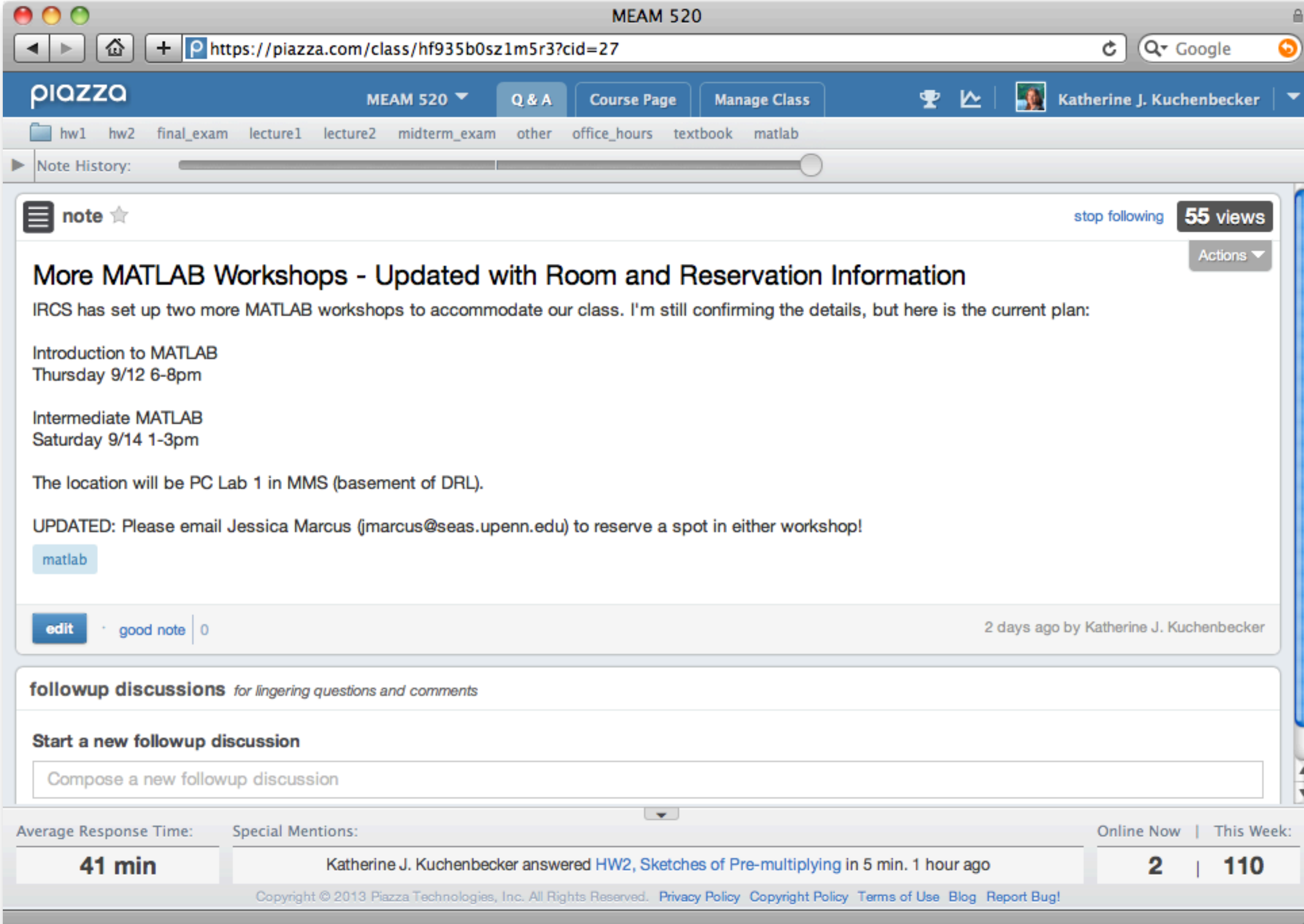
Homogeneous Transformations

Katherine J. Kuchenbecker, Ph.D.
General Robotics, Automation, Sensing, and Perception Lab (GRASP)
MEAM Department, SEAS, University of Pennsylvania

LEGES SINE MORIBUS VANAE

GRASP
LABORATORY

Lecture 5: September 12, 2013


MEAM 520
https://piazza.com/class/hf935b0sz1m5r3?cid=27
Google
piazza
MEAM 520
Q & A
Course Page
Manage Class
Katherine J. Kuchenbecker
hw1 hw2 final_exam lecture1 lecture2 midterm_exam other office_hours textbook matlab
Note History:
note
stop following
55 views
Actions
More MATLAB Workshops - Updated with Room and Reservation Information
IRCS has set up two more MATLAB workshops to accommodate our class. I'm still confirming the details, but here is the current plan:
Introduction to MATLAB
Thursday 9/12 6-8pm
Intermediate MATLAB
Saturday 9/14 1-3pm
The location will be PC Lab 1 in MMS (basement of DRL).
UPDATED: Please email Jessica Marcus (jmarcus@seas.upenn.edu) to reserve a spot in either workshop!
matlab
edit
good note | 0
2 days ago by Katherine J. Kuchenbecker
followup discussions for lingering questions and comments
Start a new followup discussion
Compose a new followup discussion
Average Response Time:
41 min
Special Mentions:
Katherine J. Kuchenbecker answered HW2, Sketches of Pre-multiplying in 5 min. 1 hour ago
Online Now | This Week:
2 | 110
Copyright © 2013 Piazza Technologies, Inc. All Rights Reserved. Privacy Policy Copyright Policy Terms of Use Blog Report Bug!

Homework 1:
MATLAB Programming and Reachable Workspace

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

September 4, 2013

This assignment is due on Tuesday, September 10, 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 Thursday, September 12, by noon (11:59:59 a.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

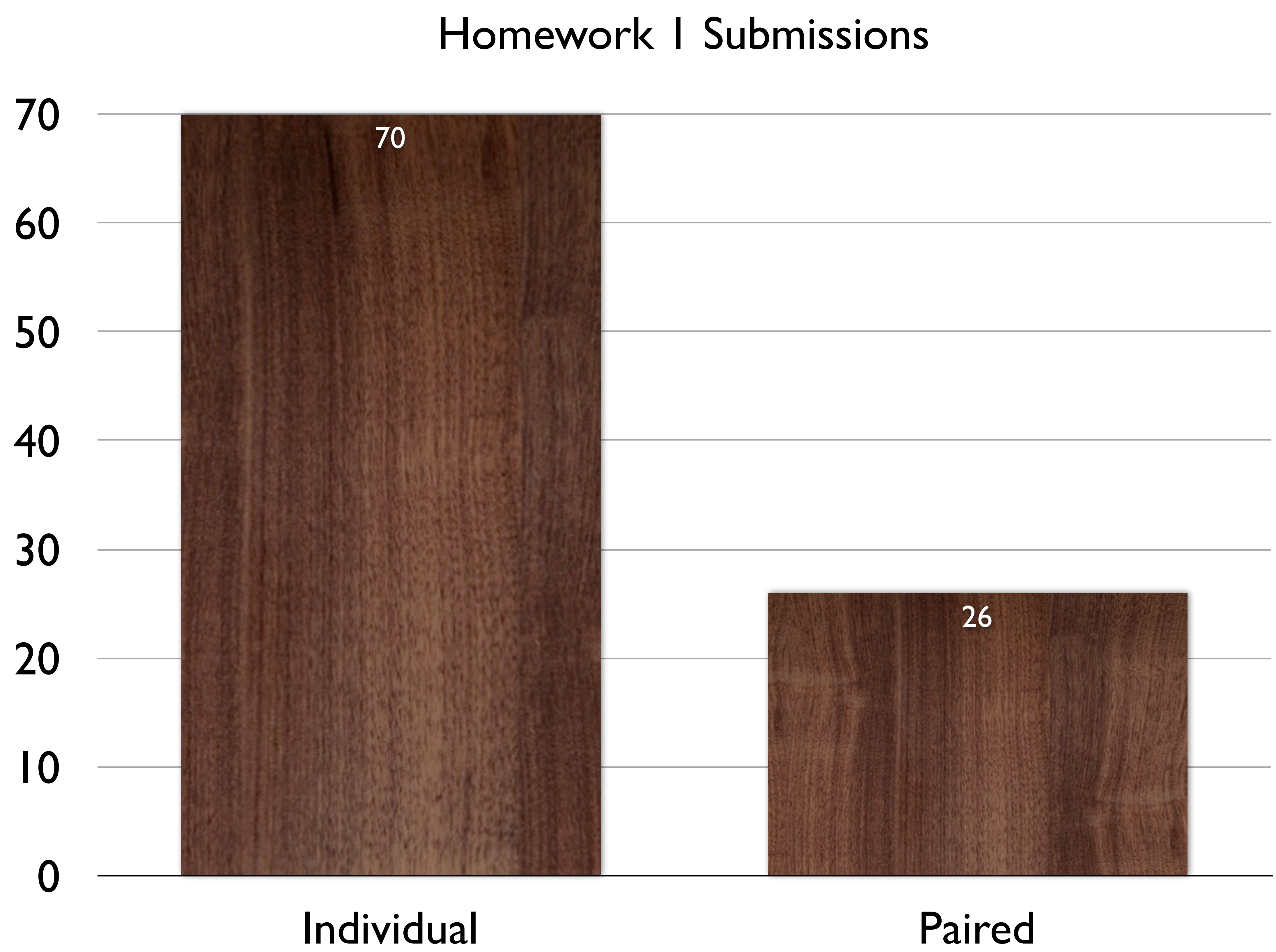
This class will use the programming language MATLAB to analyze and simulate robotic systems and also to control real robots. Some students in the class have never used MATLAB before, and others are quite familiar with it. The goal of this assignment is to get everyone starting to use MATLAB to improve their understanding of robotic systems.

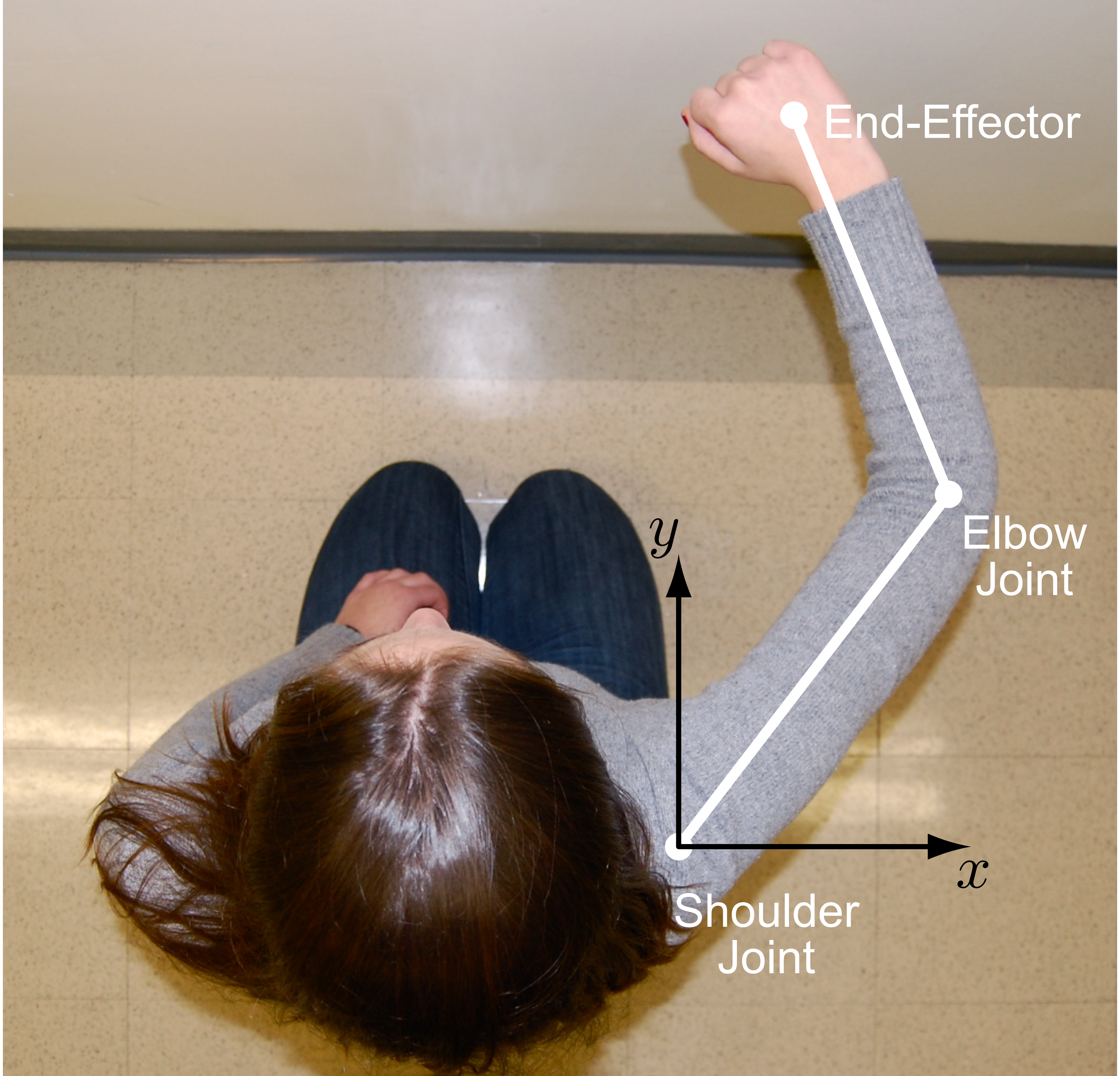
If you have not used MATLAB much before, you should do this assignment with another student in our class. If you are already pretty comfortable with MATLAB, you should do this assignment alone. Read the assignment to decide which option is right for you.

If you do this homework with a partner, you may work with anyone you choose; the only stipulation is that they also have only a little 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.


Homework 1 Submissions
70
60
50
40
30
20
10
0
70
26
Individual
Paired


End-Effector
Elbow Joint
y
x
Shoulder Joint

Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/01 workspace/matlab/arm_...

EDITOR PUBLISH VIEW

arm_workspace_kuchenbe.m

```
%% MEAM 520 Homework 1
%
% Starter code written by Professor Katherine J. Kuchenbecker
% University of Pennsylvania
%
% The goal of this assignment is to plot the reachable workspace of the
% human arm in the horizontal plane, treating it like an RR manipulator.

% Clear all of the existing variables.
clear


%% Define Variables

% Define minimum and maximum joint angles in degrees.

% First joint is the shoulder.
shouldermin_deg = -15;
shouldermax_deg = 90 + 45;

% Second joint is the elbow.
elbowmin_deg = -5;
elbowmax_deg = 90 + 70;

% Set angle increment.
angleinc_deg = 4;

% Set link lengths in centimeters.  Measurements were done in inches, so
% converting using 2.54 cm per inch.
upperarm_cm = 11.5 * 2.54;
forearm_cm = 12.5 * 2.54;

% Set the dimensions of the chest for collision checking.  I'm modeling the
% chest as a rectangular region that extends infinitely in the negative y
% direction.  Again, measurements were done in inches, so converting.
chest_x_min_cm = -17.5 * 2.54;
chest_x_max_cm = 1.5 * 2.54;
chest_y_min_cm = -inf;
chest_y_max_cm = 3 * 2.54;


%% Set Up Plot
```

script | Ln 29 Col 12

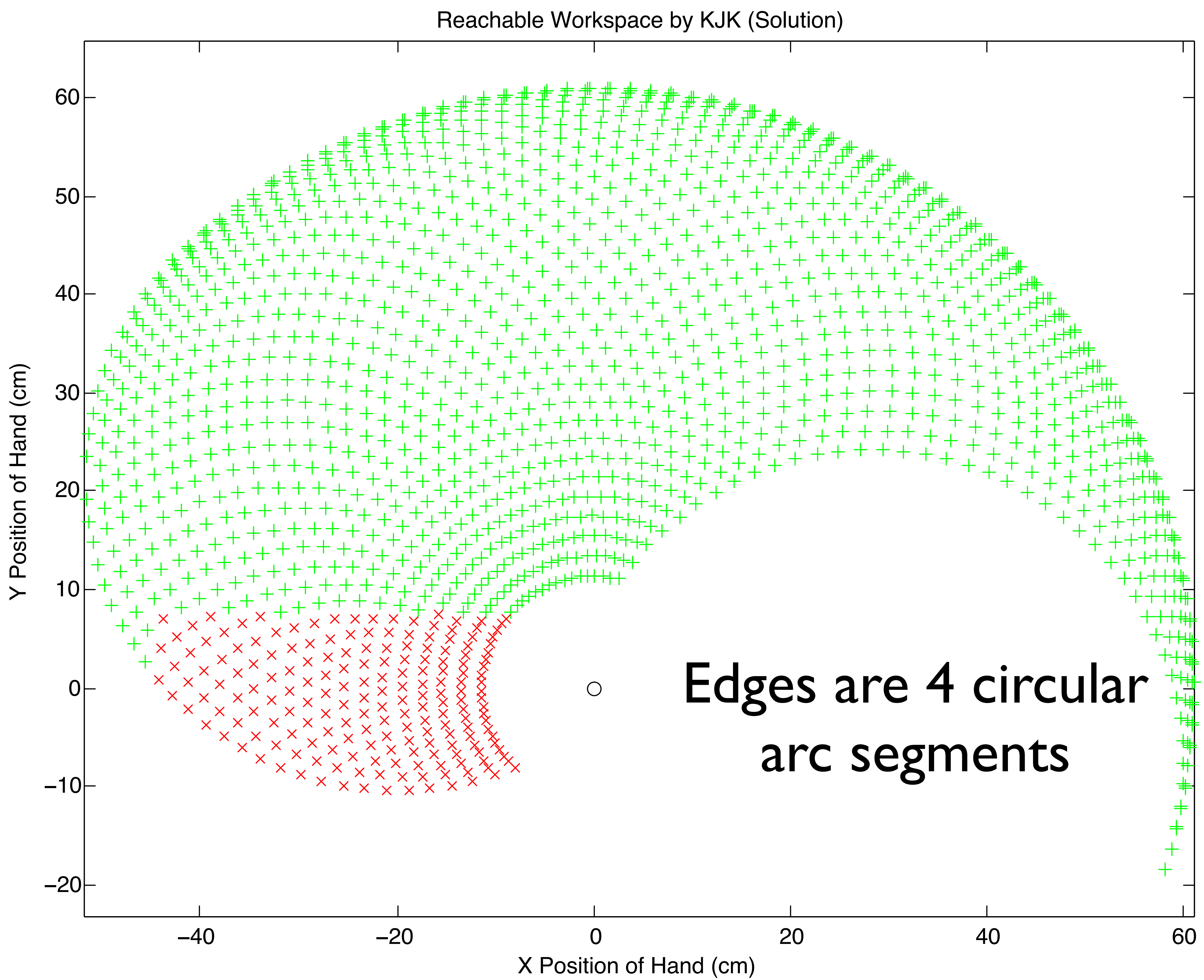
Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/01 workspace/matlab/arm_...

EDITOR PUBLISH VIEW

arm_workspace_kuchenbe.m

```
figure(2)
clf

% Plot the shoulder as a black circle at the origin.
plot(0,0,'ko')

% Turn hold on so that we can plot additional things on this figure.
hold on

% Force the axes to be displayed as the same scale.
axis equal

% Label the axes, with units.
xlabel('X Position of Hand (cm)')
ylabel('Y Position of Hand (cm)')

% Set the title of the plot.
title('Reachable Workspace by KJK (Solution)')


%% Plot the Reachable Workspace
% Update this code.

% Loop through all combinations of shoulder angle and elbow angle.
for theta1 = shouldermin_deg:angleinc_deg:shouldermax_deg
    for theta2 = elbowmin_deg:angleinc_deg:elbowmax_deg

        % Calculate x and y locations.  These equations are from the book.
        x = upperarm_cm*cosd(theta1) + forearm_cm*cosd(theta1+theta2);
        y = upperarm_cm*sind(theta1) + forearm_cm*sind(theta1+theta2);

        % Plot this position on the graph, setting the color based on the locati
        if (((x > chest_x_min_cm) && (x < chest_x_max_cm)) && ...
                ((y > chest_y_min_cm) && (y < chest_y_max_cm)))
            % Inside the body - make it red.
            plot(x,y,'rx')
        else
            % Not inside the body - make it green.
            plot(x,y,'g+')
        end
    end
end
```

script Ln 64 Col 1



Reachable Workspace by KJK (Solution)
Y Position of Hand (cm)
X Position of Hand (cm)
60
50
40
30
20
10
0
−10
−20
−40
−20
0
20
40
60
Edges are 4 circular arc segments

MEAM 520

https://piazza.com/class/hf935b0sz1m5r3?cid=26

piazza   MEAM 520   Q & A   Course Page   Manage Class   Katherine J. Kuchenbecker

hw1   hw2   final_exam   lecture1   lecture2   midterm_exam   other   office_hours   textbook   matlab

note   stop following   65 views

A question to ponder about the human arm...

After you complete the first homework, I encourage you to think about the following question:

Why do your joint limits permit your hand to move to places that are not reachable, such as inside your chest?

hw1

edit · good note | 0   3 days ago by Katherine J. Kuchenbecker

followup discussions *for lingering questions and comments*

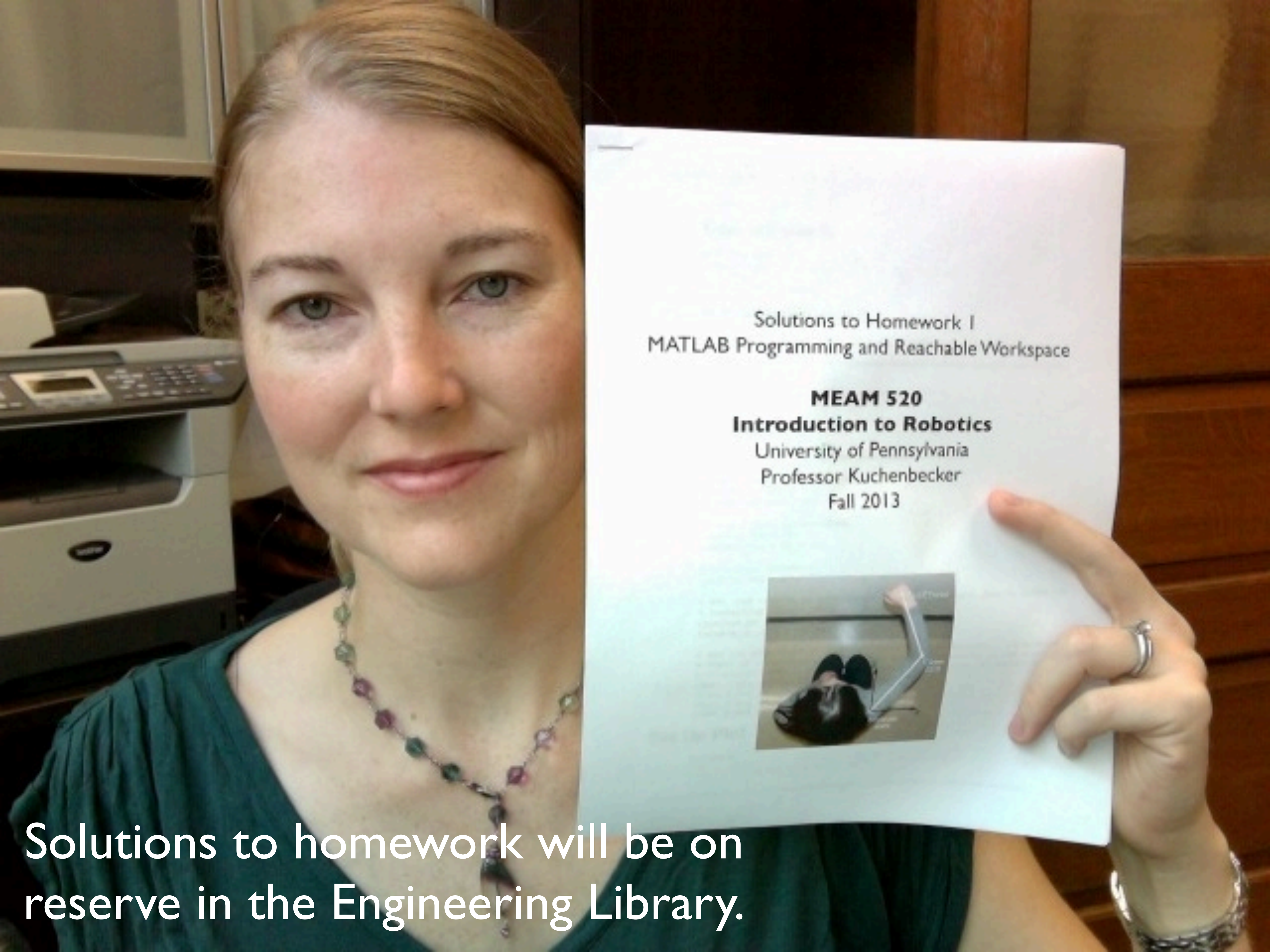
Start a new followup discussion

Compose a new followup discussion

Average Response Time: 41 min

Special Mentions: Katherine J. Kuchenbecker answered HW2, Sketches of Pre-multiplying in 5 min. 1 hour ago

Online Now | This Week: 2 | 110



Solutions to homework will be on reserve in the Engineering Library.

Grades will be posted on Canvas.

Total of 10 points possible.

Any deductions will be explained in the comment attached to the grade.

Some extra credit is always available for exceptional work.

Don’t forget to turn in your assignment on time! It is fine to submit multiple times.

Post a private message on Piazza if you want clarification on your grade or think we made a mistake.

Questions ❓

Homework 2:
Rotation Matrices and Homogeneous Transformations

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

September 10, 2013

This paper-based assignment is due on Tuesday, September 17, by midnight (11:59:59 p.m.) You should aim to turn it 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, in the bin or under the door. Late submissions will be accepted until Thursday, September 19, by noon (11:59:59 a.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 a solution manual. 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!

These problems are 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. This assignment is worth a total of 20 points.

1. SHV 2-6, page 65 – Verifying Three Properties of $R_{z,\theta}$ *(2 points)*
2. SHV 2-10, page 66 – Sequence of Rotations *(2 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.
3. SHV 2-14, page 67 – Rotating a Coordinate Frame *(4 points)*
   Sketch the initial, intermediate, and final frames by reading the text in the problem. Make your drawings big, and remember the right-hand rule. Then find R in two ways: by inspection of your sketch and by calculation. Check your solutions against one another.
4. SHV 2-23, page 68 – Axis/Angle Representation *(4 points)*
   Be careful with the sketch, and remember the right-hand rule.
5. SHV 2-39, page 70 – Homogeneous Transformations *(4 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). Be careful with notation; you are looking for H_1^0, H_2^0, H_3^0, and H_3^2.
6. SHV 2-43, page 71-72 – Commutativity of Homogeneous Transformations *(4 points)*

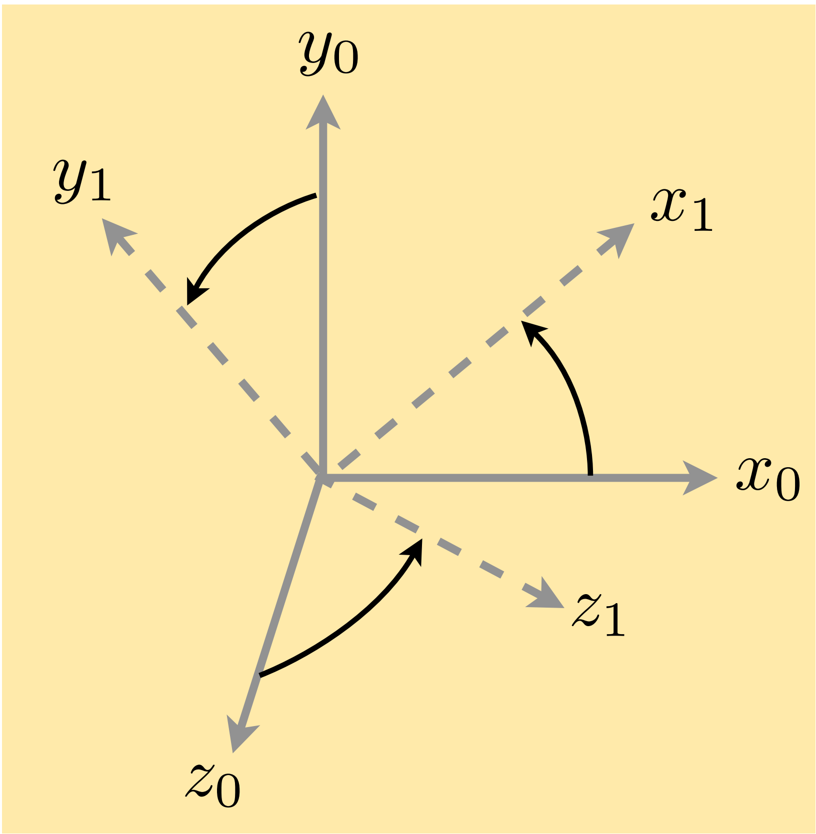
Extra paper copies of Homework 2

Due next Tuesday by midnight.

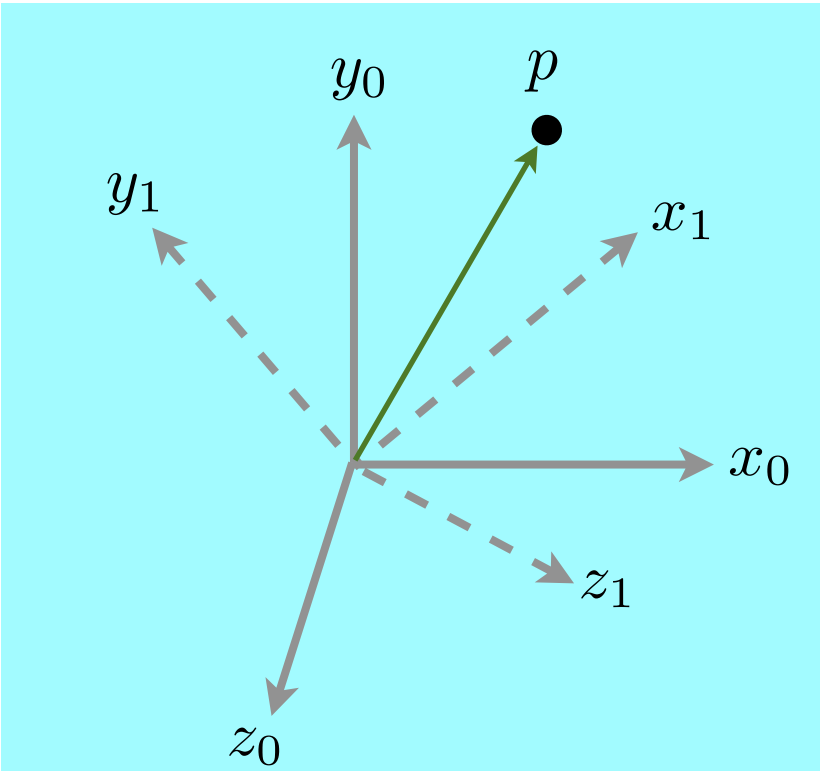
Late deadline is next Thursday at noon.

Interpretations of Rotation Matrices

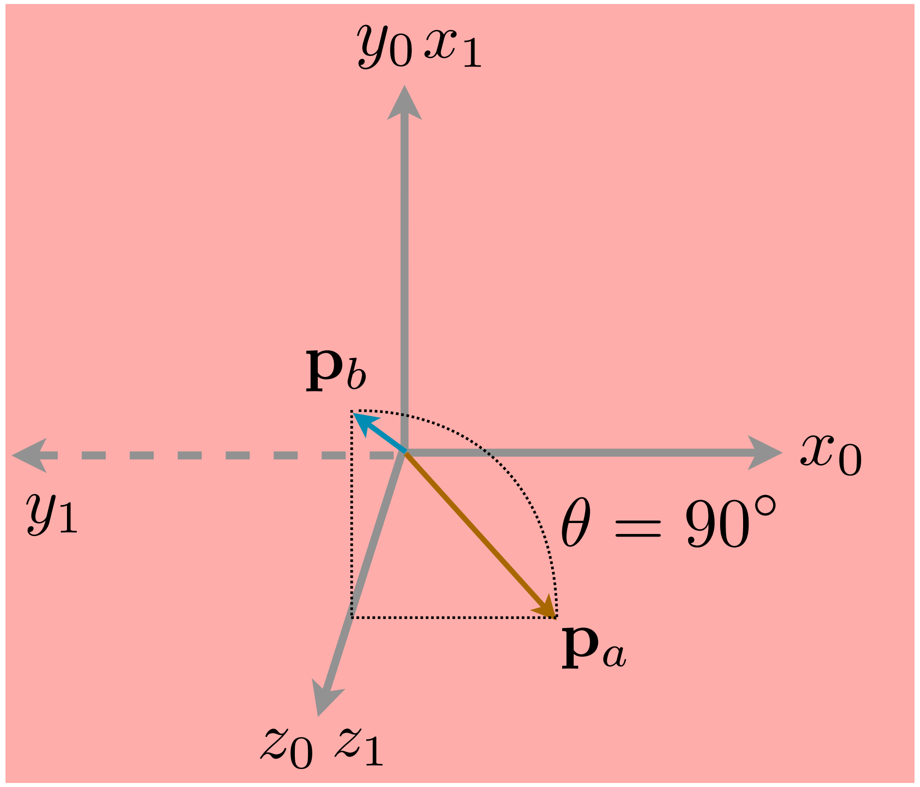
$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$



Orientation of one coordinate frame with respect to another frame



Coordinate transformation relating the coordinates of a point p in two different frames



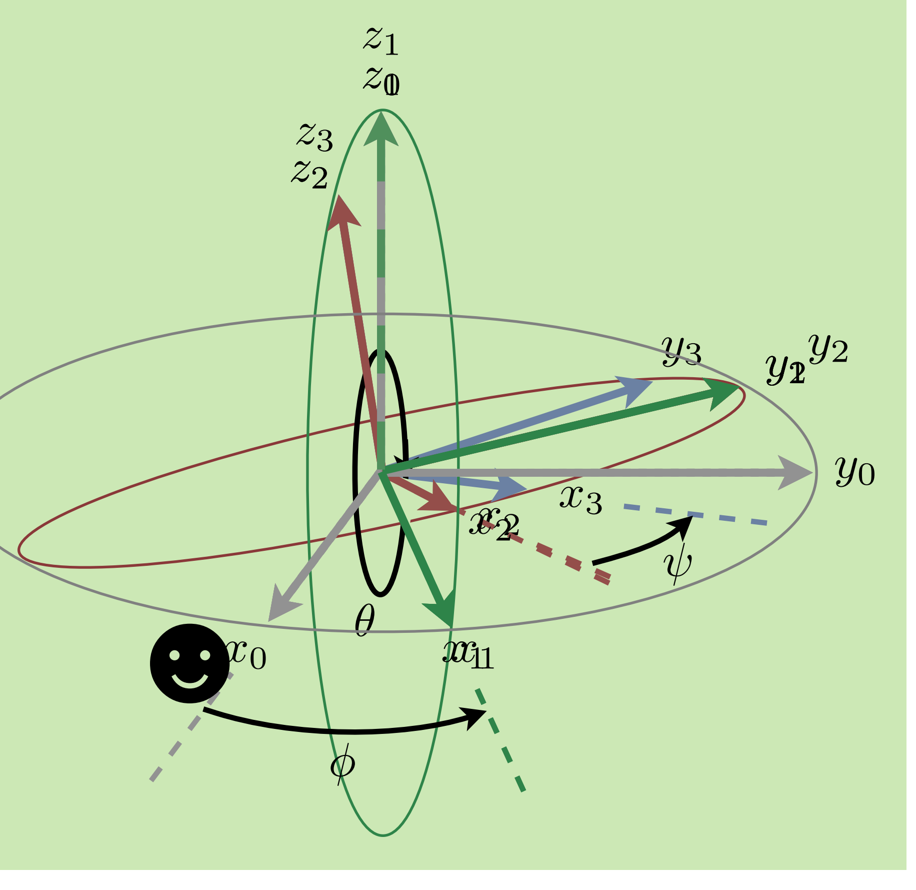
Operator taking a vector and rotating it to yield a new vector in the same coordinate frame

Parameterizations of Rotation Matrices

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Euler Angles

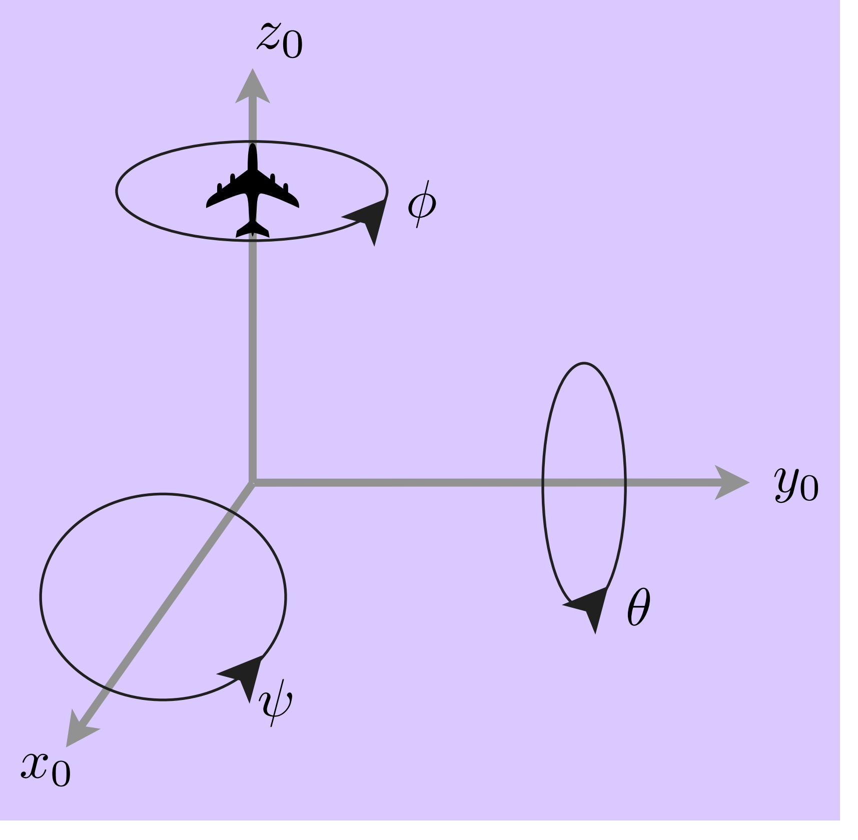
three rotations about intermediate frames



Our book uses ZYZ

Roll, Pitch, Yaw Angles

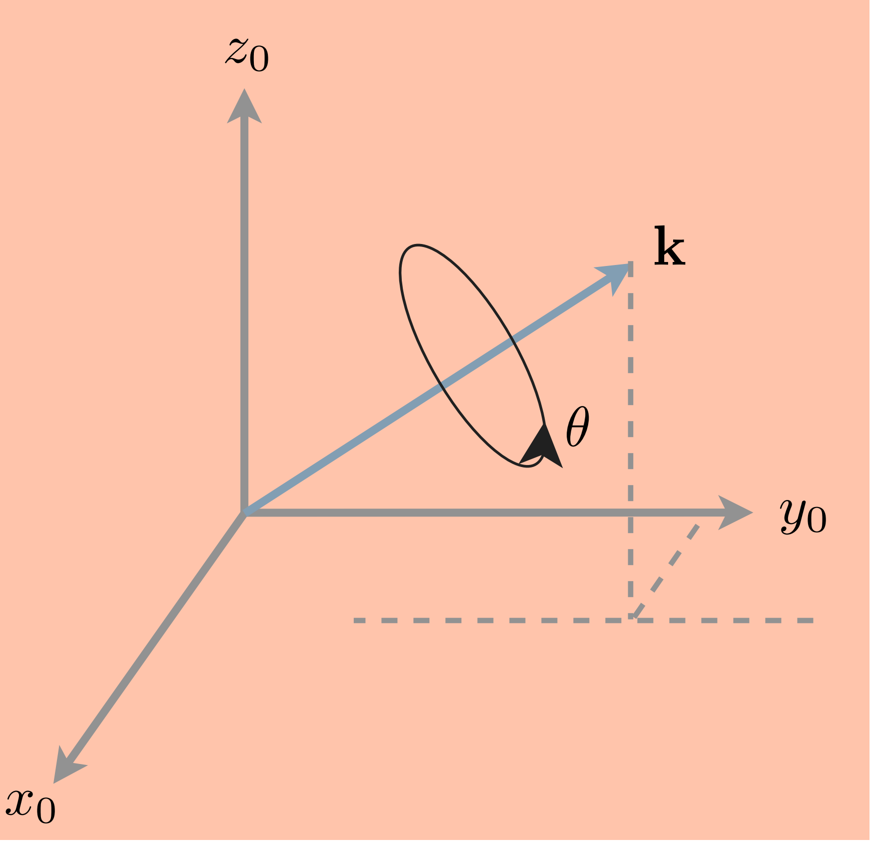
three rotations about the fixed frame

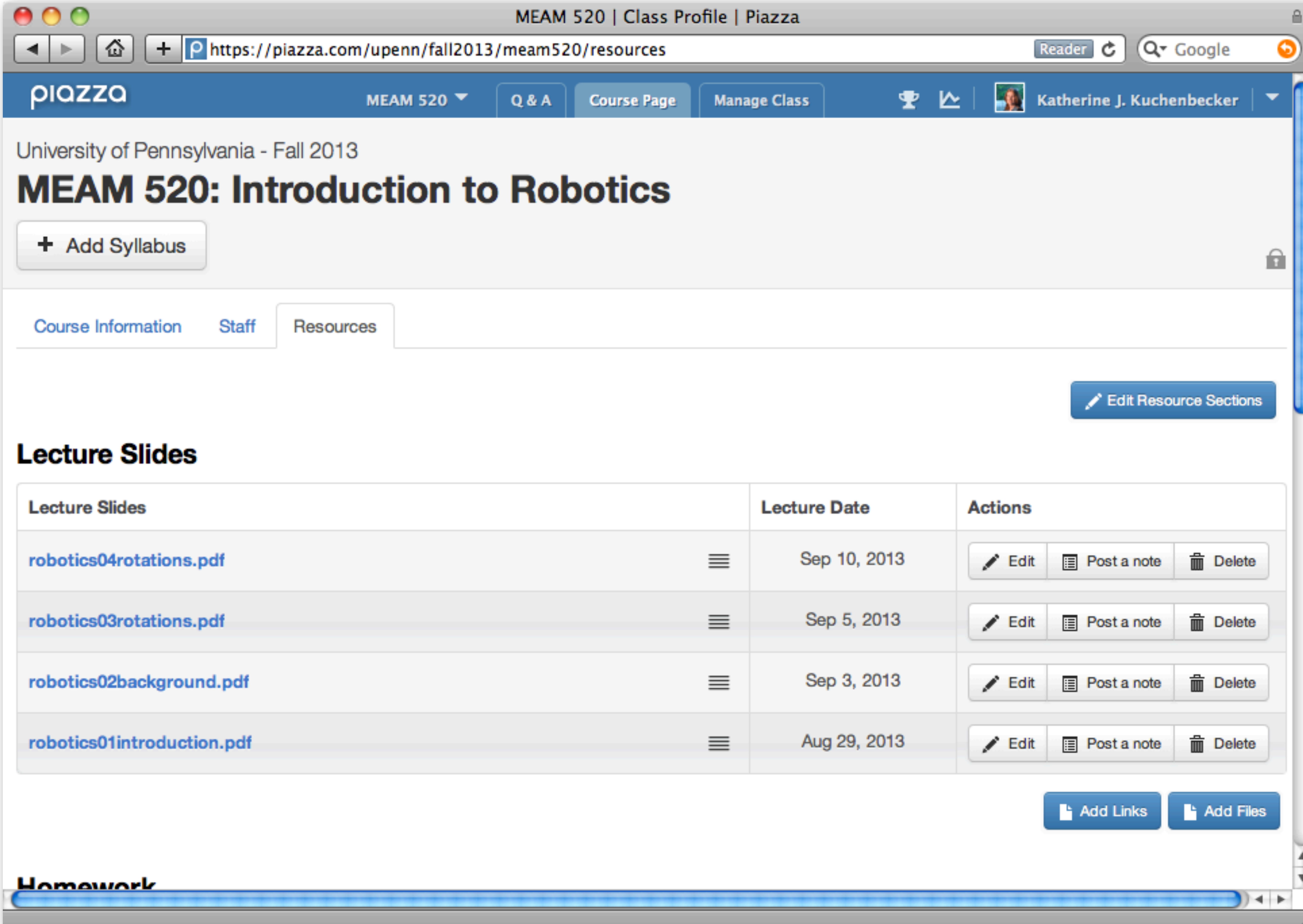


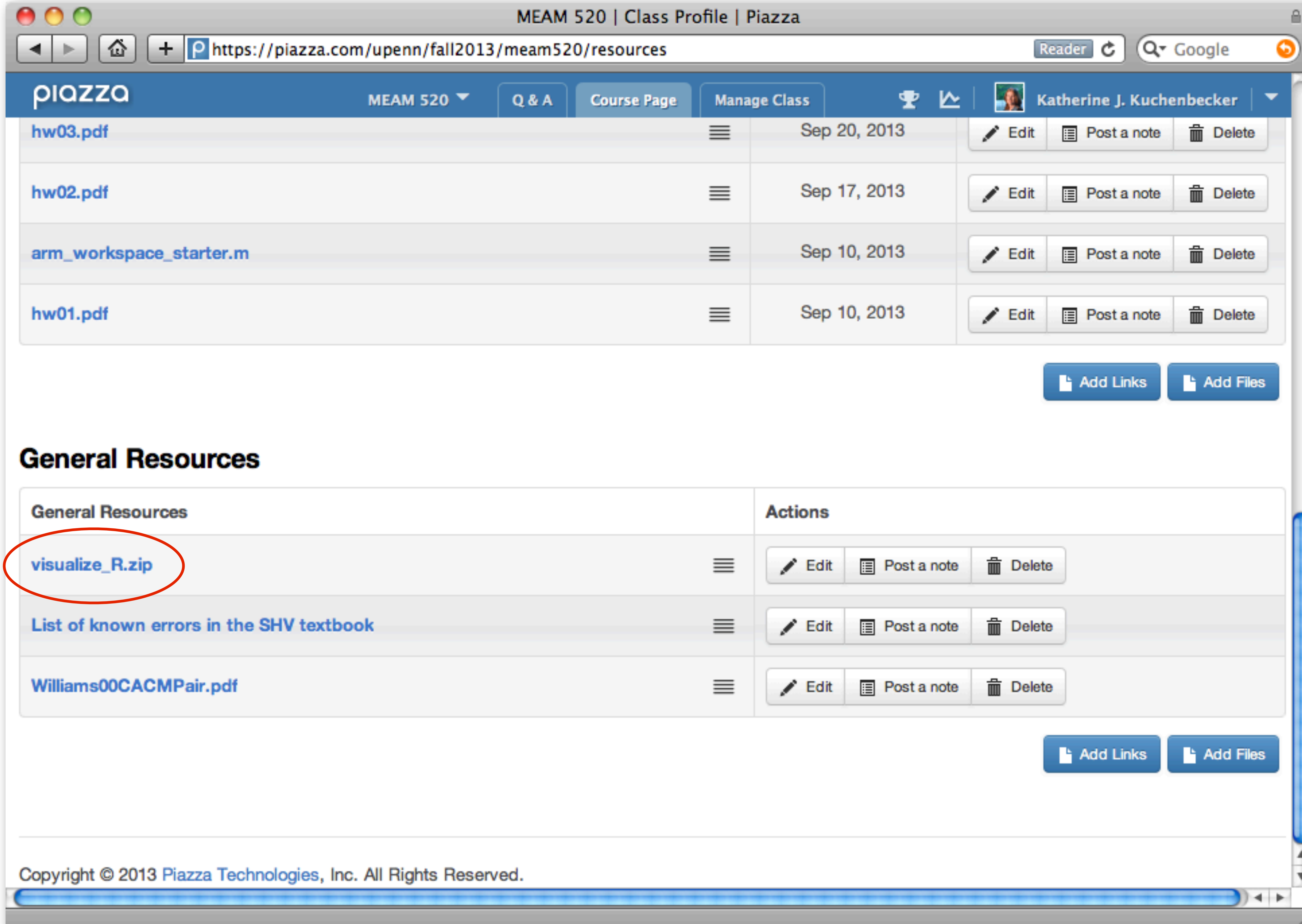
Our book uses XYZ

Axis/Angle

a unit vector (axis) and one angle




MEAM 520 | Class Profile | Piazza
https://piazza.com/upenn/fall2013/meam520/resources
Reader
Google
piazza
MEAM 520
Q & A
Course Page
Manage Class
Katherine J. Kuchenbecker
University of Pennsylvania - Fall 2013
MEAM 520: Introduction to Robotics
+ Add Syllabus
Course Information
Staff
Resources
Edit Resource Sections
Lecture Slides
Lecture Slides
Lecture Date
Actions
robotics04rotations.pdf
Sep 10, 2013
Edit
Post a note
Delete
robotics03rotations.pdf
Sep 5, 2013
Edit
Post a note
Delete
robotics02background.pdf
Sep 3, 2013
Edit
Post a note
Delete
robotics01introduction.pdf
Aug 29, 2013
Edit
Post a note
Delete
Add Links
Add Files


MEAM 520 | Class Profile | Piazza
https://piazza.com/upenn/fall2013/meam520/resources
Reader
Google
piazza
MEAM 520
Q & A
Course Page
Manage Class
Katherine J. Kuchenbecker
hw03.pdf
Sep 20, 2013
Edit
Post a note
Delete
hw02.pdf
Sep 17, 2013
arm_workspace_starter.m
Sep 10, 2013
hw01.pdf
Sep 10, 2013
Add Links
Add Files
General Resources
General Resources
Actions
visualize_R.zip
List of known errors in the SHV textbook
Williams00CACMPair.pdf
Edit
Post a note
Delete
Add Links
Add Files
Copyright © 2013 Piazza Technologies, Inc. All Rights Reserved.

Editor – /Users/kuchenbe/Documents/teaching/meam 520/lectures/05 transformations/visualize_R/visualize_R.m

EDITOR | PUBLISH | VIEW

visualize_R.m | plot_coordinate_frame.m | plot_vector.m

```matlab
%% Visualize a rotation matrix and its parameterizations.
%
% Written by Katherine J. Kuchenbecker, University of Pennsylvania
% For MEAM 520: Introduction to Robotics
% September 2013
% Version 1


%% Find out which matrix to use

% If the matrix already exists, ask if we should use it.
if exist('R01','var')
    button = questdlg('Which rotation matrix do you want to use?','','Existing R01', 'Random', 'R
else
    button = 'Random';
end


%% Create a random three-by-three rotation matrix if needed

if strcmp(button, 'Random')
    % This method was suggested on this MATLAB Central thread:
    % http://www.mathworks.com/matlabcentral/newsreader/view_thread/298500
    % The source credited there is "How to generate random matrices from the
    % classical compact groups", Francesco Mezzadri, Notices of ACM, Volume 54,
    % Number 5, 2007.

    % Create a 3 x 3 matrix of normally distributed pseudorandom numbers.
    random_matrix = randn(3);

    % Decompose the random matrix using QR decomposition.
    [Q, R] = qr(random_matrix);

    % Create what will be our rotation matrix by manipulation of Q and R.
    % We call this R01 because it shows the orientation of frame 1 in frame 0.
    R01 = Q*diag(sign(diag(R)));

    % Delete Q and R matrices because we don't need them any more.
    clear Q R

```

script | Ln 1 | Col 1

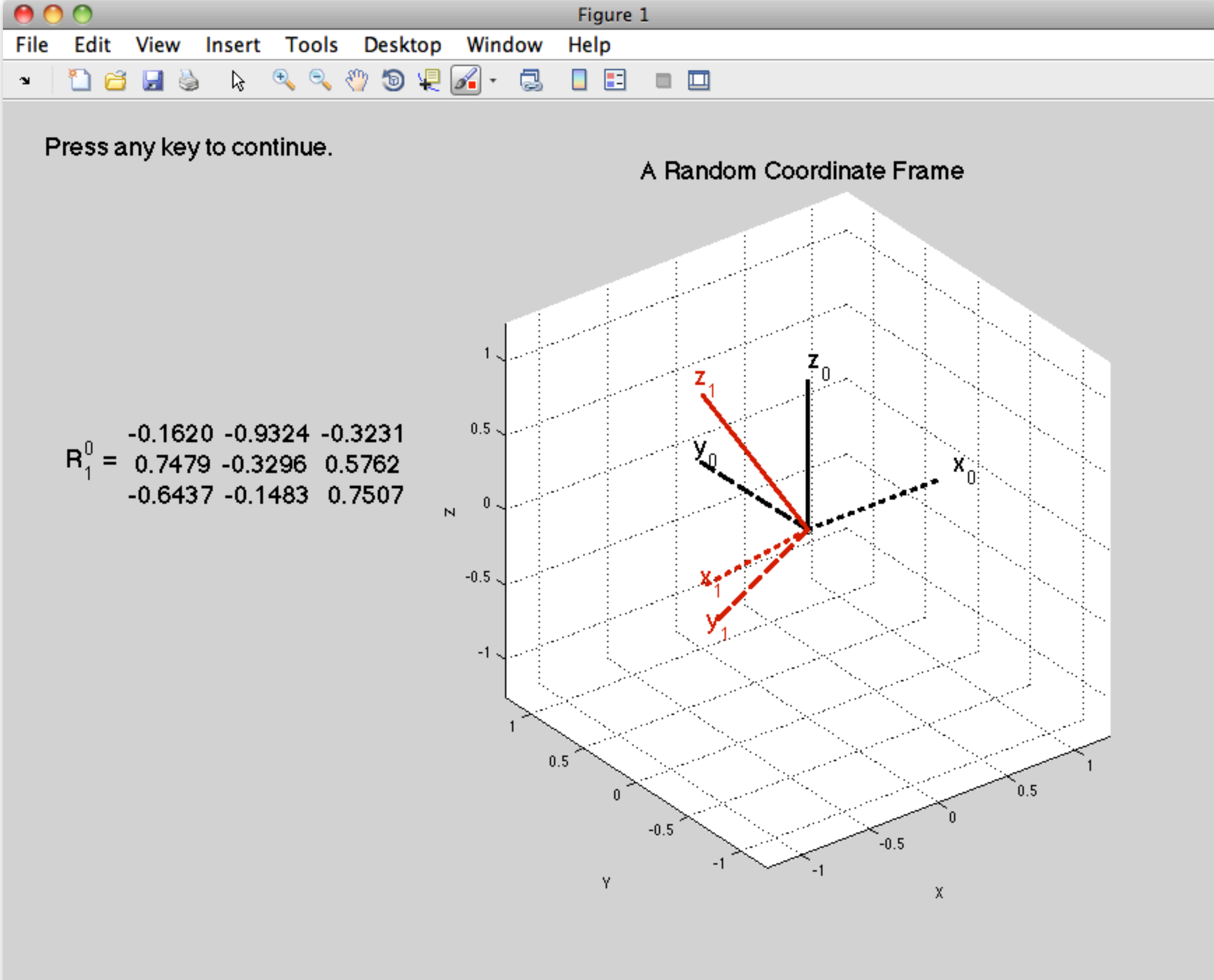
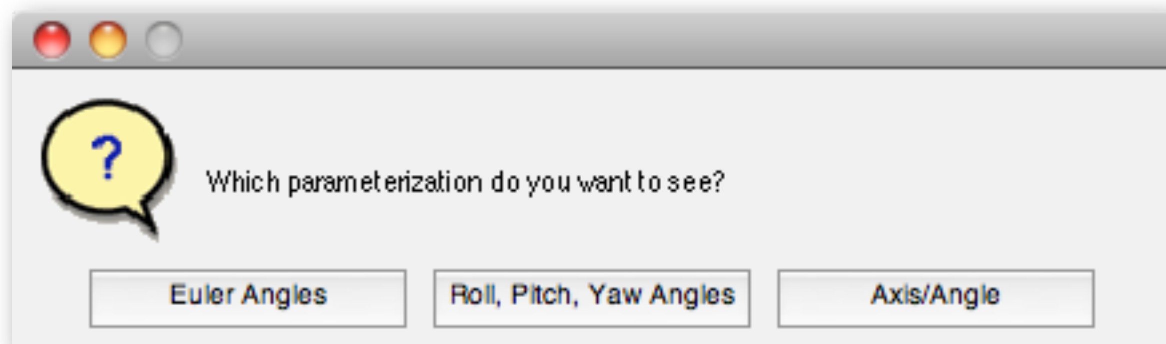


Figure 1
File Edit View Insert Tools Desktop Window Help
Press any key to continue.
A Random Coordinate Frame
R
0
1
=
-0.1620 -0.9324 -0.3231
0.7479 -0.3296 0.5762
-0.6437 -0.1483 0.7507
z
0
y
0
x
0
z
1
x
1
y
1
Z
Y
X
1
0.5
0
-0.5
-1


Which parameterization do you want to see?
Euler Angles
Roll, Pitch, Yaw Angles
Axis/Angle

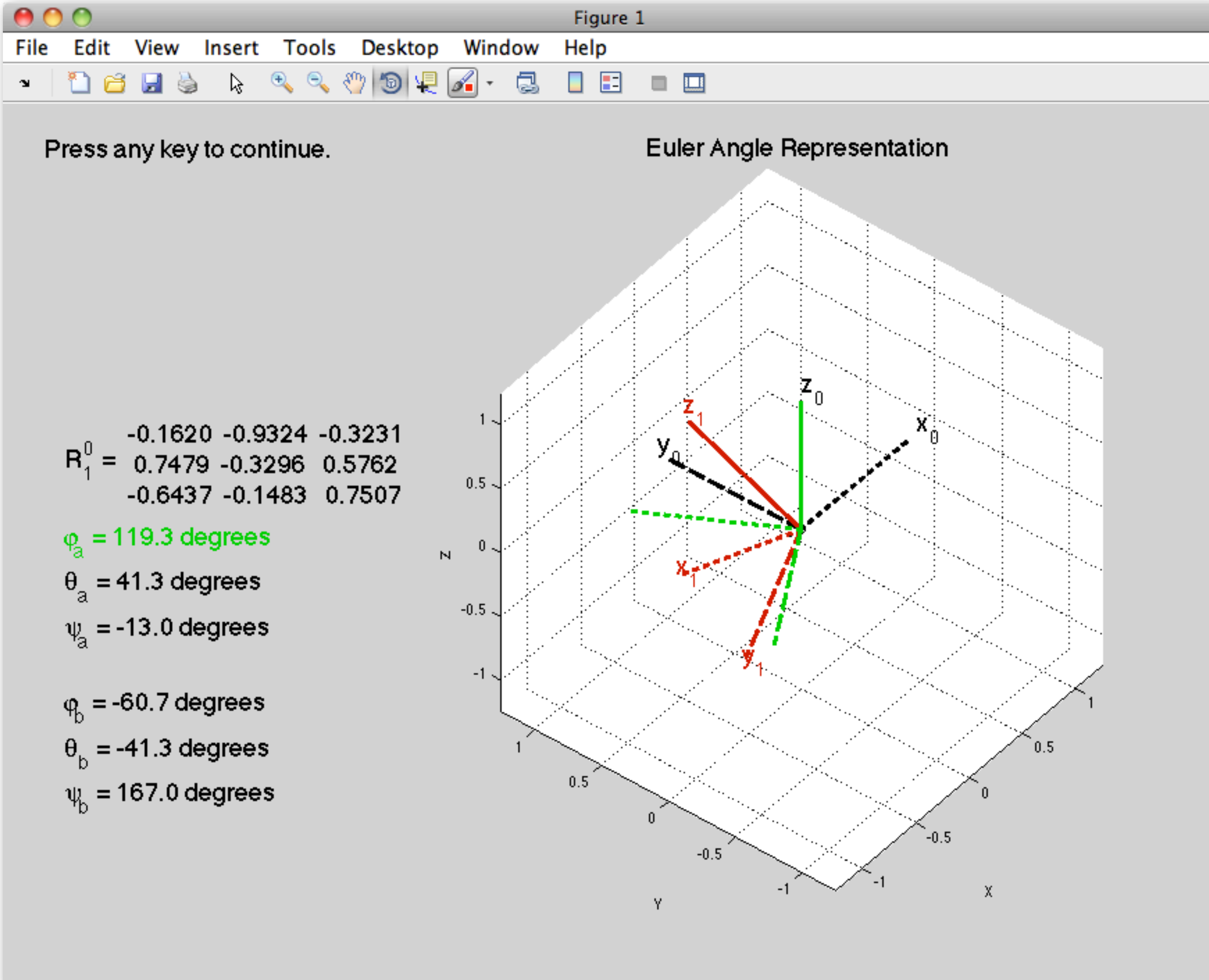


Figure 1
File Edit View Insert Tools Desktop Window Help
Press any key to continue.
Euler Angle Representation
R_1^0 = -0.1620 -0.9324 -0.3231
0.7479 -0.3296 0.5762
-0.6437 -0.1483 0.7507
φ_a = 119.3 degrees
θ_a = 41.3 degrees
ψ_a = -13.0 degrees
φ_b = -60.7 degrees
θ_b = -41.3 degrees
ψ_b = 167.0 degrees
z_0
x_0
y_0
z_1
x_1
y_1
z
Y
X

I strongly recommend that you play with this tool to hone your skills at visualizing rotation matrices and their parameterizations.

Caveat: I wrote this last night. Some parts are not carefully commented and not super efficient. There may be bugs or errors. If you find one, report it on Piazza with the specific steps needed to produce the error.

If you're interested, look at the code to see how I implemented the calculations and created the animations.

If you make an improvement, send it to me!

Questions ?

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.

1

Homework 3

Due next Friday 9/20 by midnight.

Late deadline is next Sunday 9/22 at midnight.

Done individually or in pairs – your choice.

3D Guidance trakSTAR™

Class 1, Type B Applied Part

Desktop electronics unit tracks multiple sensors simultaneously

Track Objects with Magnetic DC Technology

- **Fast, dynamic tracking** – 240 to 420 updates per second.
- **Miniaturized passive sensors** – outputs unaffected by "power-line" noise sources.
- **All attitude tracking** – no inertial drift or optical interference.
- **High metal immunity** – no distortion from non magnetic metals.

Interchangeable sensor sizes for full six degrees–of–freedom tracking

Magnetic field transmitter options for mid and short-range tracking

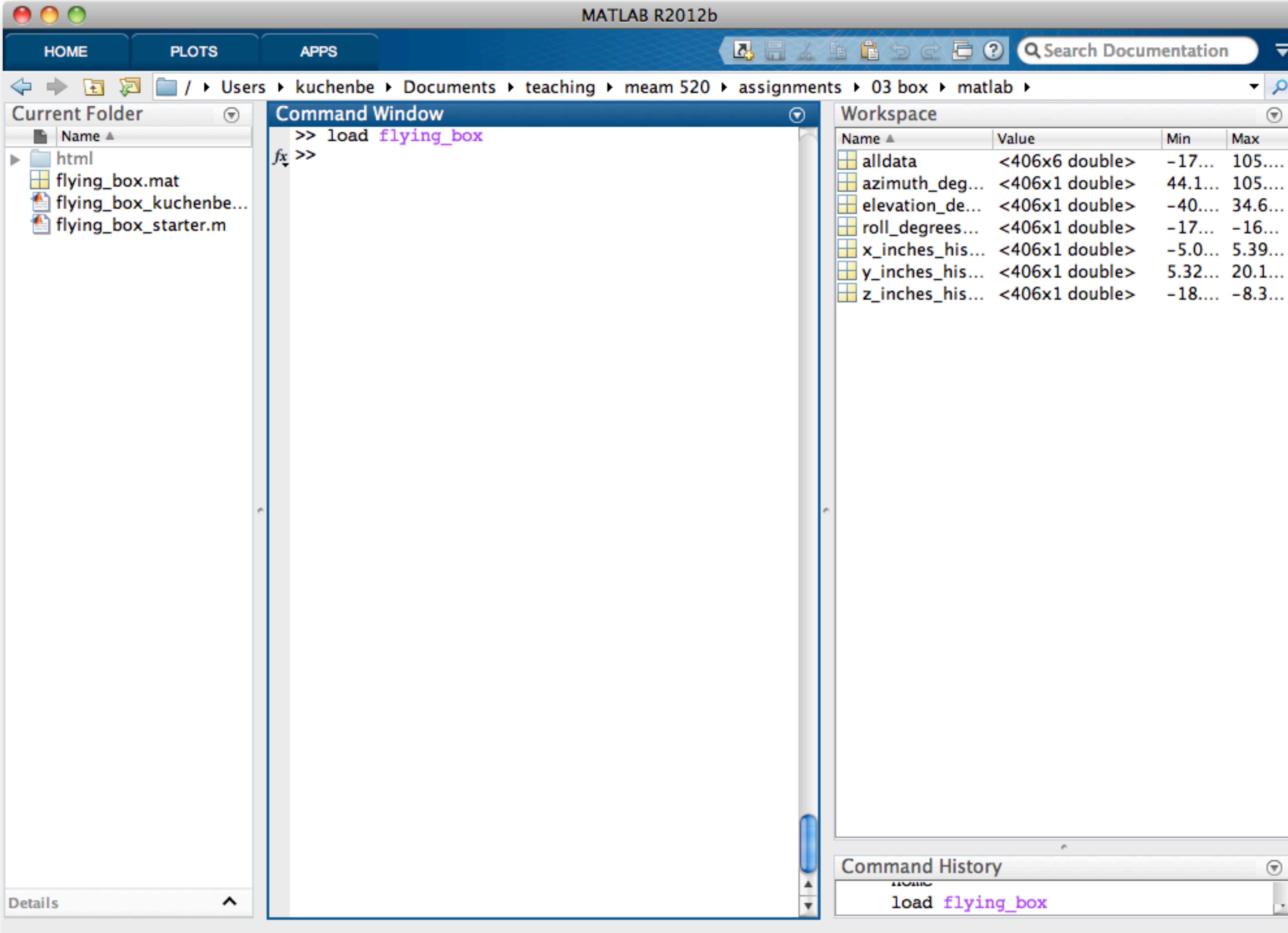
Ascension
Technology Corporation
an **NDI** company

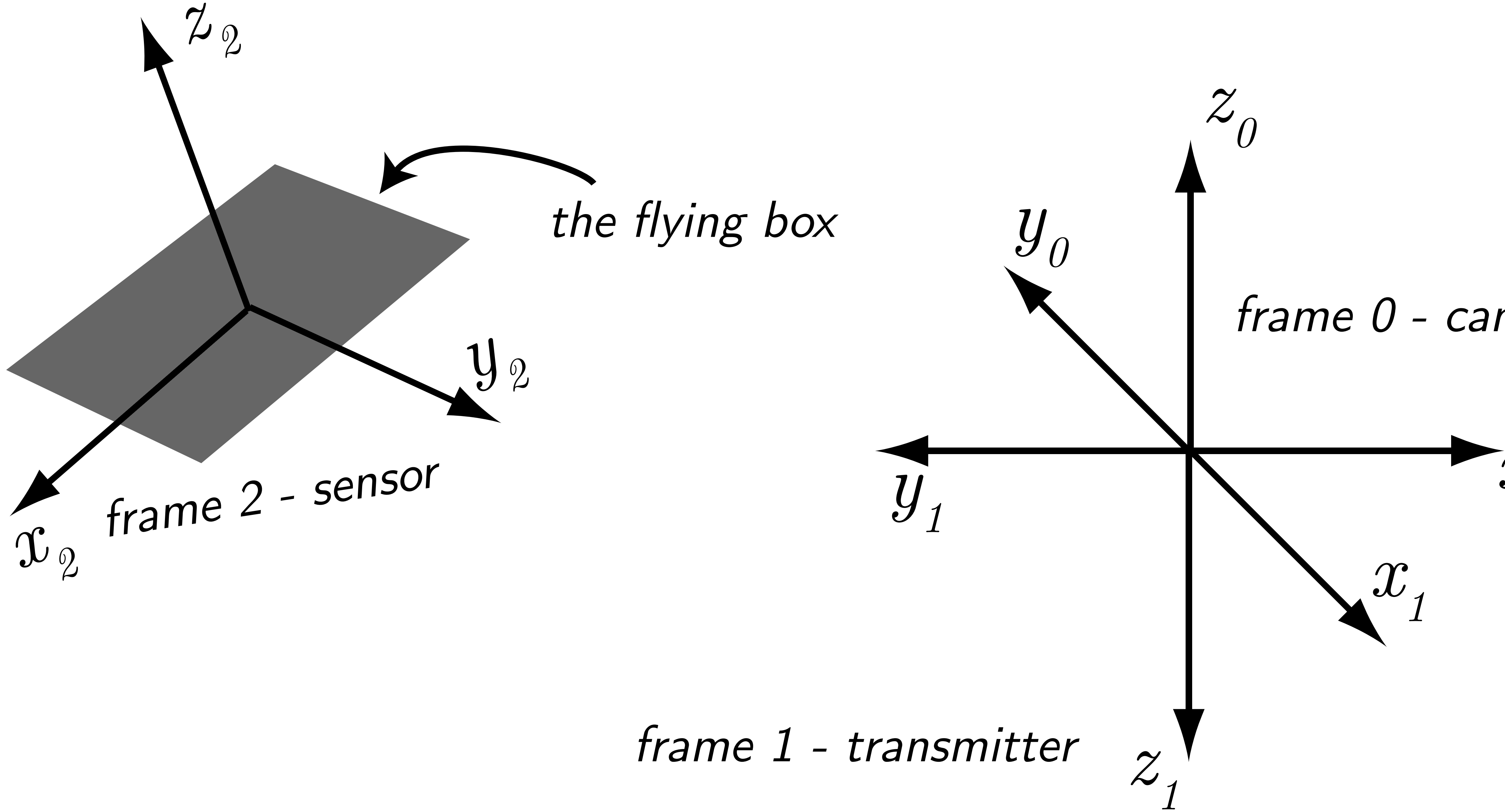
FAST PORTABLE AFFORDABLE

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


MATLAB R2012b
HOME
PLOTS
APPS
Search Documentation
/ ▸ Users ▸ kuchenbe ▸ Documents ▸ teaching ▸ meam 520 ▸ assignments ▸ 03 box ▸ matlab ▸
Current Folder
Name
html
flying_box.mat
flying_box_kuchenbe...
flying_box_starter.m
Details
Command Window
>> load flying_box
>>
Workspace
Name
Value
Min
Max
alldata <406x6 double> -17... 105....
azimuth_deg... <406x1 double> 44.1... 105....
elevation_de... <406x1 double> -40.... 34.6...
roll_degrees... <406x1 double> -17... -16...
x_inches_his... <406x1 double> -5.0... 5.39...
y_inches_his... <406x1 double> 5.32... 20.1...
z_inches_his... <406x1 double> -18.... -8.3...
Command History
load flying_box



z2
the flying box
y2
x2
frame 2 - sensor
z0
y0
frame 0 - camera
y1
x0
x1
frame 1 - transmitter
z1

Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/03 box/matlab/flying_box_starter.m

EDITOR PUBLISH VIEW

flying_box_starter.m

```
%% 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
% (kuchenbe@seas.upenn.edu). Students will modify this code to create
% their own script.
%
% Change the name of this file to replace "starter" with your PennKey.  For
% example, Professor Kuchenbecker's script would be flying_box_kuchenbe.m

%% SETUP

% Delete all variables from our workspace.
clear

% Set student names.
studentNames = 'PUT YOUR NAME HERE';

% 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;


%% 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.
```

script | Ln 17 Col 21

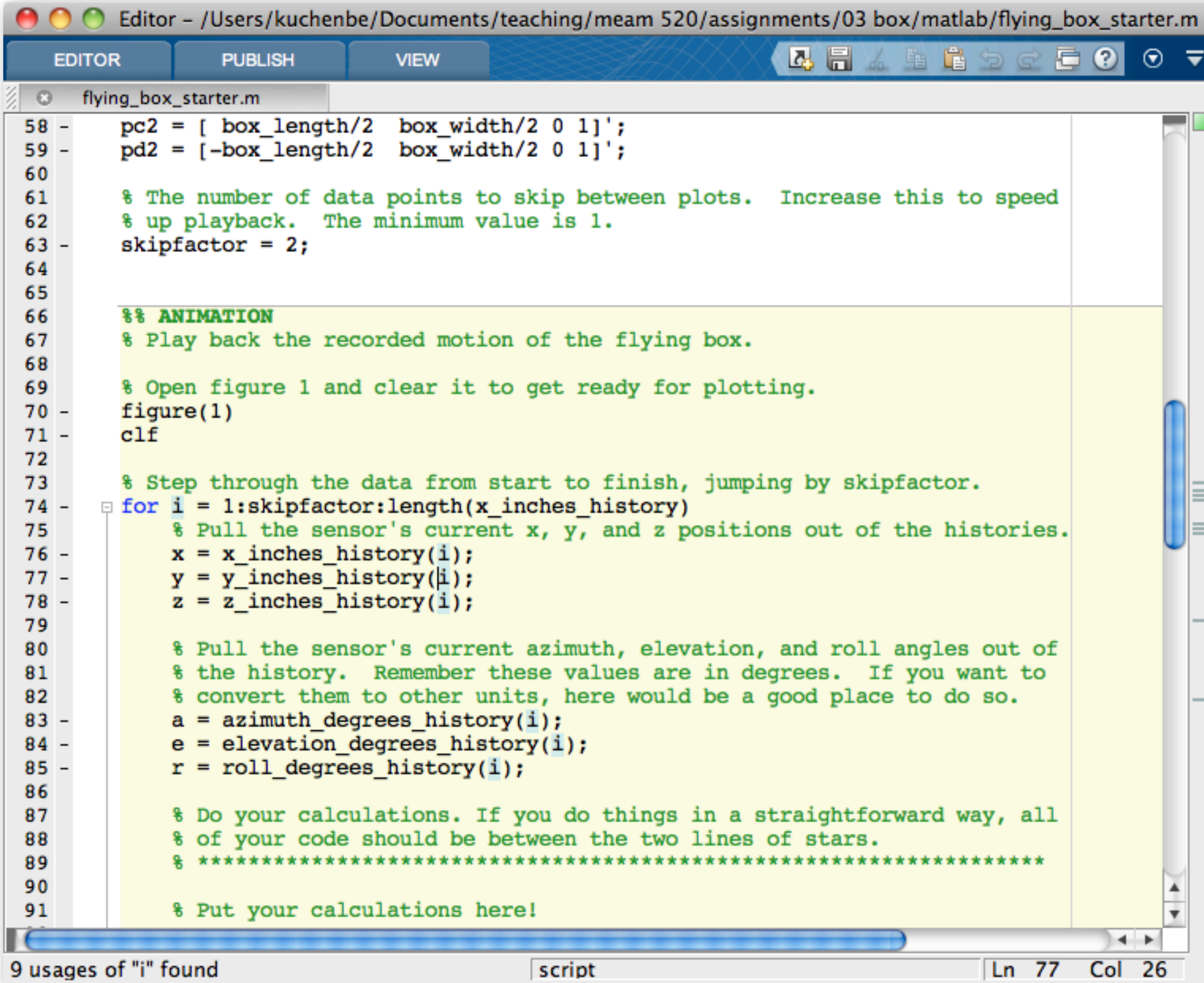
Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/03 box/matlab/flying_box_starter.m

EDITOR | PUBLISH | VIEW

flying_box_starter.m

```matlab

%% 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
% down.  This is the frame in which the sensor positions and orientations
% are expressed.  Its origin is near the center of the transmitter's beige
% cube, which can be seen in the video.
%
% Frame 2 is the frame of the TrakStar sensor, which is being moved around.
% Its x-axis is straight out horizontal through the front of the box, in
% the direction the hand is facing.  Its y-axis is mostly horizontal during
% the video, and its z-axis is mostly vertical.

% Define the locations of the box's four corners (ignoring thickness) in
% the sensor's frame (frame 2).  The length is in the direction of the
% sensor's x-axis, and the width is in the direction of the sensor's
% y-axis.  We call the corners points a, b, c, and d (pa, pb, pc, and pd),
% and we assume the sensor is at the center of the box.
% We include a 1 at the end of each of these column vectors for use with
% homogenous transformations. You may remove the 1's if you find you do not
% need them but please update this comment accordingly.
box_length = 8;  % inches
box_width = 6;   % inches
pa2 = [-box_length/2 -box_width/2 0 1]';
pb2 = [ box_length/2 -box_width/2 0 1]';
pc2 = [ box_length/2  box_width/2 0 1]';
pd2 = [-box_length/2  box_width/2 0 1]';

```

script | Ln 26 Col 1


Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/03 box/matlab/flying_box_starter.m
EDITOR
PUBLISH
VIEW
flying_box_starter.m
pc2 = [box_length/2 box_width/2 0 1]';
pd2 = [-box_length/2 box_width/2 0 1]';
% The number of data points to skip between plots. Increase this to speed
% up playback. The minimum value is 1.
skipfactor = 2;
%% ANIMATION
% Play back the recorded motion of the flying box.
% Open figure 1 and clear it to get ready for plotting.
figure(1)
clf
% Step through the data from start to finish, jumping by skipfactor.
for i = 1:skipfactor:length(x_inches_history)
% Pull the sensor's current x, y, and z positions out of the histories.
x = x_inches_history(i);
y = y_inches_history(i);
z = z_inches_history(i);
% Pull the sensor's current azimuth, elevation, and roll angles out of
% the history. Remember these values are in degrees. If you want to
% convert them to other units, here would be a good place to do so.
a = azimuth_degrees_history(i);
e = elevation_degrees_history(i);
r = roll_degrees_history(i);
% Do your calculations. If you do things in a straightforward way, all
% of your code should be between the two lines of stars.
% **
% Put your calculations here!
9 usages of "i" found
script
Ln 77
Col 26

Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/03 box/matlab/flying_box_starter.m

EDITOR PUBLISH VIEW

flying_box_starter.m

```matlab

    % Do your calculations. If you do things in a straightforward way, all
    % of your code should be between the two lines of stars.
    % *******************************************************************

    % Put your calculations here!

    % You 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, usin only low-level functions such as
    % sind, cosd, and vector/matrix math.

    % Calculate the position of each corner in the camera's frame. For now,
    % we just set the locations to be the positions in frame 2, with one
    % augmentation.  You will need to change this.
    pa0 = pa2;
    pb0 = pb2;
    pc0 = pc2 + [0 0 i/30 0]'; % Make this corner grow over time so you can see th
    pd0 = pd2;

    % If your code runs but you can't see anything in your plot, comment
    % out or modify the axis([...]) command in the plot section below.

    % All your calculations should be done by here.  Your answers for the
    % coordinates of the box's corners in frame 0 should be in the
    % variables pa0, pb0, pc0, and pd0.
    % *******************************************************************

    % Put all four corner points together to make it easier to get the x,
    % y, and z coordinates for plotting.  These are all in the camera's
    % frame (frame 0).
    points0 = [pa0 pb0 pc0 pd0];

```

script | Ln 103 Col 15

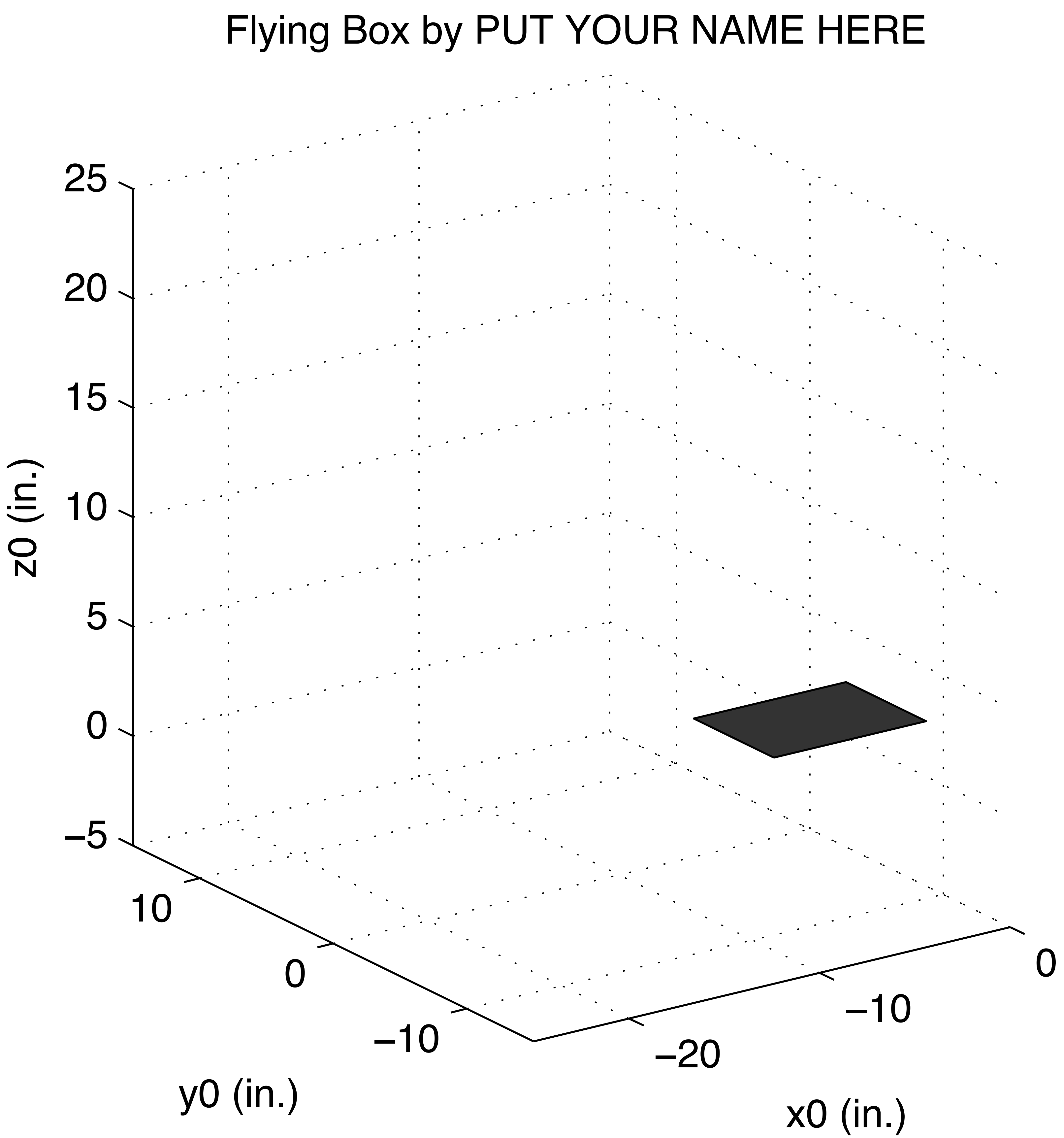
Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/03 box/matlab/flying_box_starter.m

EDITOR PUBLISH VIEW

flying_box_starter.m

```matlab

        % Set the viewing volume to the values known to be correct.  If
        % you cannot see anything in your plot, you should increase the
        % ranges here or comment this out.  The order is xmin xmax ymin
        % ymax zmin zmax.
        axis([-25 0 -15 15 -5 25])

        % Set the viewing angle to be similar to the camera view.
        view(-35,20)

        % Label the axes, including abbreviated units of measure in parentheses.
        xlabel('x0 (in.)')
        ylabel('y0 (in.)')
        zlabel('z0 (in.)')

        % Turn on the grid to make it easy to see the walls.
        grid on

        title(['Flying Box by ' studentNames])

    else

        % Set the locations of the corners to the current points.  Using
        % set and the plot handle in this way is faster than replotting
        % everything.
        set(h,'xdata',points0(1,:),'ydata',points0(2,:),'zdata',points0(3,:))

    end

    % Pause for a short time to allow the viewer to see the animation play.
    pause(0.001)

end

```

script | Ln 149 Col 9


Flying Box by PUT YOUR NAME HERE
z0 (in.)
25
20
15
10
5
0
−5
y0 (in.)
10
0
−10
x0 (in.)
−20
−10
0

Programming Homework Tips:

1. Write out your approach before sitting down to program.

2. Start early to give yourself time to figure things out.

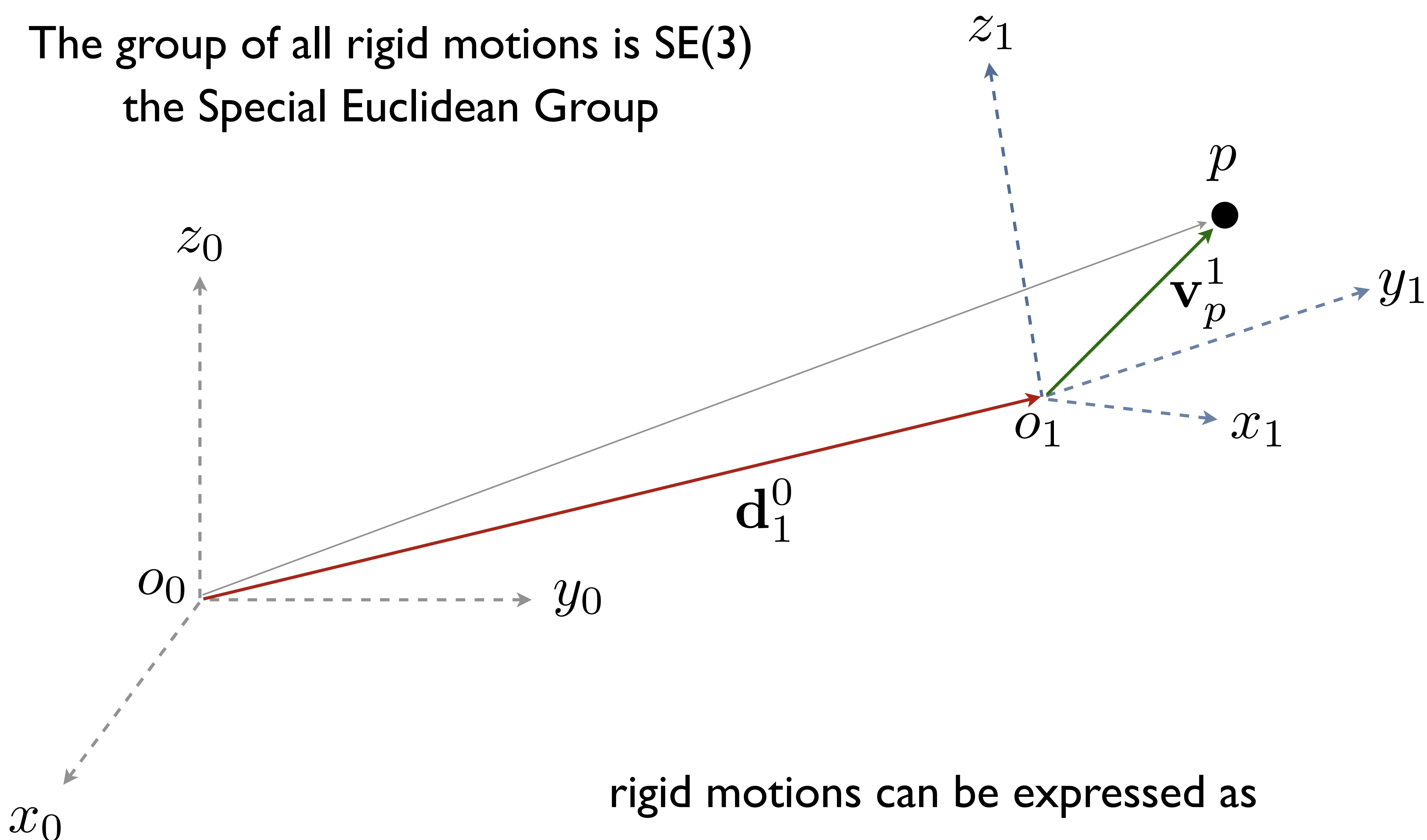
Questions ?

One more big thing we need to learn for HW2 and HW3!

Rigid Motions & Homogeneous Transformations

a **rigid motion** couples pure translation with pure rotation

The group of all rigid motions is SE(3)
the Special Euclidean Group



rigid motions can be expressed as

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

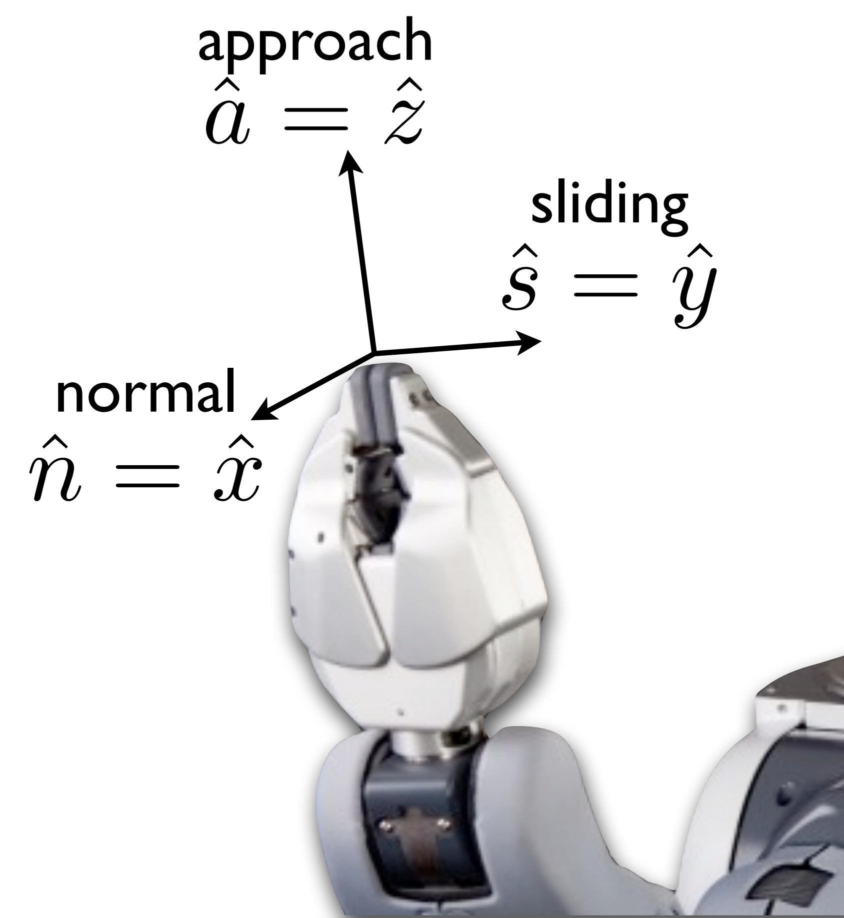
Homogeneous Transformations

a **homogeneous transformation** is a matrix representation of rigid motion, defined as

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}$$

where $\mathbf{R}$ is the 3x3 rotation matrix, and $\mathbf{d}$ is the 3x1 translation vector

$$\begin{array}{c} \qquad\qquad \mathbf{R}_1^0 \qquad\qquad\qquad \mathbf{d}_1^0 \\ \mathbf{H} = \begin{bmatrix} n_x & s_x & a_x & d_x \\ n_y & s_y & a_y & d_y \\ n_z & s_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$



the **homogeneous representation** of a vector is formed by concatenating the original vector with a unit scalar

$$\mathbf{P} = \left[\begin{array}{c} \mathbf{p} \\ 1 \end{array} \right]$$

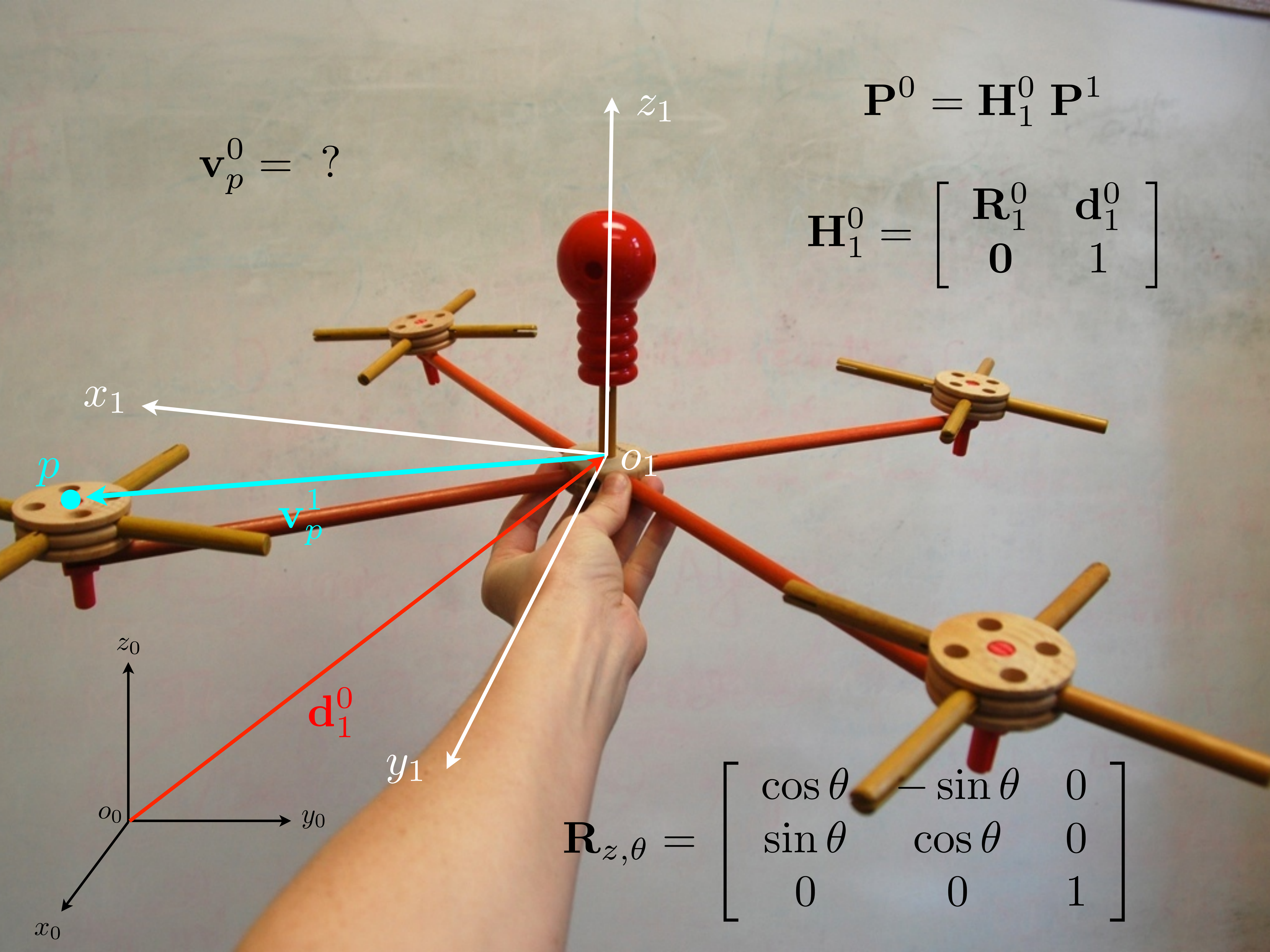
where $\mathbf{p}$ is the 3x1 vector

$$\mathbf{P} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

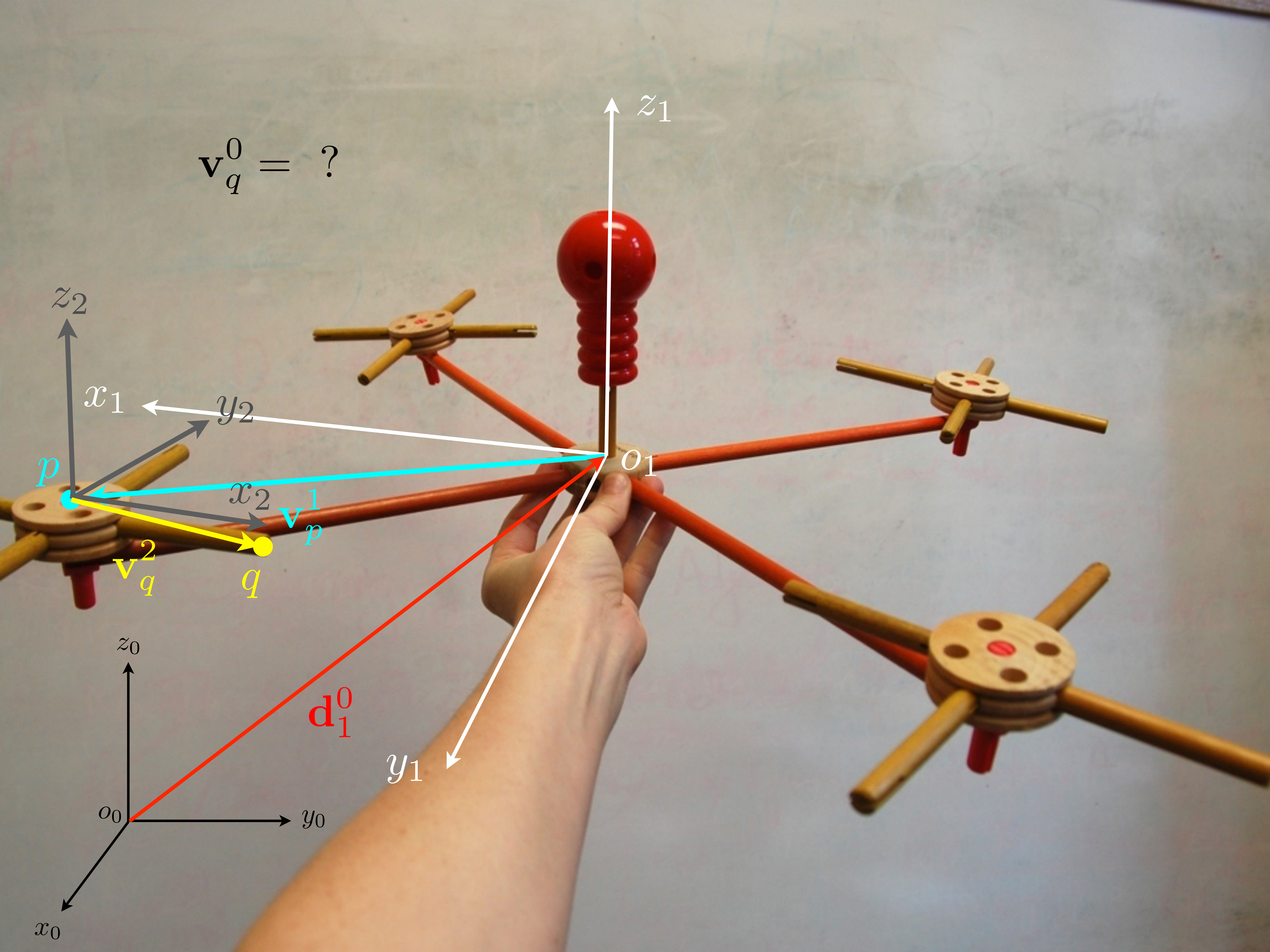
rigid body transformations are accomplished by pre-multiplying by the homogenous transform

$$\mathbf{P}^0 = \mathbf{H}_1^0\, \mathbf{P}^1$$

{example with quadrotor model}



$\mathbf{P}^0 = \mathbf{H}_1^0 \, \mathbf{P}^1$
$\mathbf{v}_p^0 = \; ?$
$\mathbf{H}_1^0 = \begin{bmatrix} \mathbf{R}_1^0 & \mathbf{d}_1^0 \\ \mathbf{0} & 1 \end{bmatrix}$
z_1
x_1
o_1
p
$\mathbf{v}_p^1$
y_1
$\mathbf{d}_1^0$
z_0
o_0
y_0
x_0
$\mathbf{R}_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$



$\mathbf{v}_q^0 = \ ?$
z_1
z_2
x_1
y_2
p
o_1
x_2
$\mathbf{v}_p^1$
$\mathbf{v}_q^2$
q
z_0
$\mathbf{d}_1^0$
y_1
o_0
y_0
x_0

composition of multiple transforms is the same as for rotation matrices:

post-multiply when successive rotations are relative to intermediate frames

$$\mathbf{H}_2^0 = \mathbf{H}_1^0\,\mathbf{H}_2^1$$

pre-multiply when successive rotations are relative to the first fixed frame

$$\mathbf{H}_2^0 = \mathbf{H}\,\mathbf{H}_1^0$$

Composition (intermediate frame)

$$\mathbf{H}_2^0 = \mathbf{H}_1^0\,\mathbf{H}_2^1 = \begin{bmatrix} \mathbf{R}_1^0 & \mathbf{d}_1^0 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_2^1 & \mathbf{d}_2^1 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_2^0 & \mathbf{R}_1^0\mathbf{d}_2^1 + \mathbf{d}_1^0 \\ \mathbf{0} & 1 \end{bmatrix}$$

Inverse Transform

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

Basic Homogeneous Transformations

$$\text{Trans}_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Trans}_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{y,\beta} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Trans}_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{z,\gamma} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Questions ?

GRASP Seminar this Friday, September 13th – Manuela Veloso — grasp

Delete Junk Reply Reply All Forward Print To Do

From: Charity Payne
Subject: **GRASP Seminar this Friday, September 13th - Manuela Veloso**
Date: September 9, 2013 2:28:55 PM EDT
To: grasp-seminar@lists.seas.upenn.edu , cis-faculty@lists.seas.upenn.edu , cis-postdocs@lists.seas.upenn.edu , cis-phd-owner@lists.seas.upenn.edu , cis-mcit@lists.seas.upenn.edu , cis-ms@lists.seas.upenn.edu , and 1 more...

Free pizza afterward on Levine 4th floor

Fall 2013 GRASP Seminar

Friday, September 13th at 11am
Wu & Chen Auditorium

Manuela Veloso
Carnegie Mellon University

"Symbiotic Autonomy: Robots, Humans, and the Web"

Abstract: We envision ubiquitous autonomous mobile robots that coexist and interact with humans while performing tasks. Such robots are still far from common, as our environments offer great challenges to robust autonomous robot perception, cognition, and action. In this talk, I present symbiotic robot autonomy in which robots are robustly autonomous in their localization and navigation, as well as handle they limitations by proactively asking for help from humans, accessing the web for missing knowledge, and coordinating with other robots. Such symbiotic autonomy has enabled our CoBot robots to move in our multi-floor buildings performing a variety of service tasks, including escorting visitors, and transporting packages between locations. I will describe CoBot's fully autonomous effective mobile robot indoor localization and navigation algorithms, its human-centered task planning, and its symbiotic interaction with the humans, the web, and other robots, namely other CoBots and Baxter. I will further present our ongoing research on knowledge learning from our speech-based robot interaction with humans. The talk will be illustrated with results and examples from many hours-long runs of the robots in our buildings. The work is joint with Joydeep Biswas, Brian Coltin, Stephanie Rosenthal, Mehdi Samadi, Tom Kollar, Vittorio Perera, Robin Soetens, and Yichao Sun. Special thanks to Cetin Mericli and Daniele Nardi.

Biography: Manuela M. Veloso is Herbert A. Simon Professor in the Computer Science Department at Carnegie Mellon University. She researches in Artificial Intelligence and Robotics. She founded and directs the CORAL research laboratory, for the study of multiagent systems where agents Collaborate, Observe, Reason, Act, and Learn, www.cs.cmu.edu/~coral. Professor Veloso is IEEE Fellow, AAAS Fellow, and AAAI Fellow. She is the current President of AAAI, and the past President of RoboCup. She received the 2009 ACM/SIGART Autonomous Agents Research Award for her contributions to agents in uncertain and dynamic environments, including distributed robot localization and world modeling, strategy selection in multiagent systems in the presence of adversaries, and robot learning from demonstration. Professor Veloso and her students have worked with a variety of autonomous robots, for robot soccer, education, and service robots. See www.cs.cmu.edu/~mmv for further information, including publications.

Want to get on the GRASP email list?

Ask Charity Payne
charity@cis.upenn.edu