MEAM 520

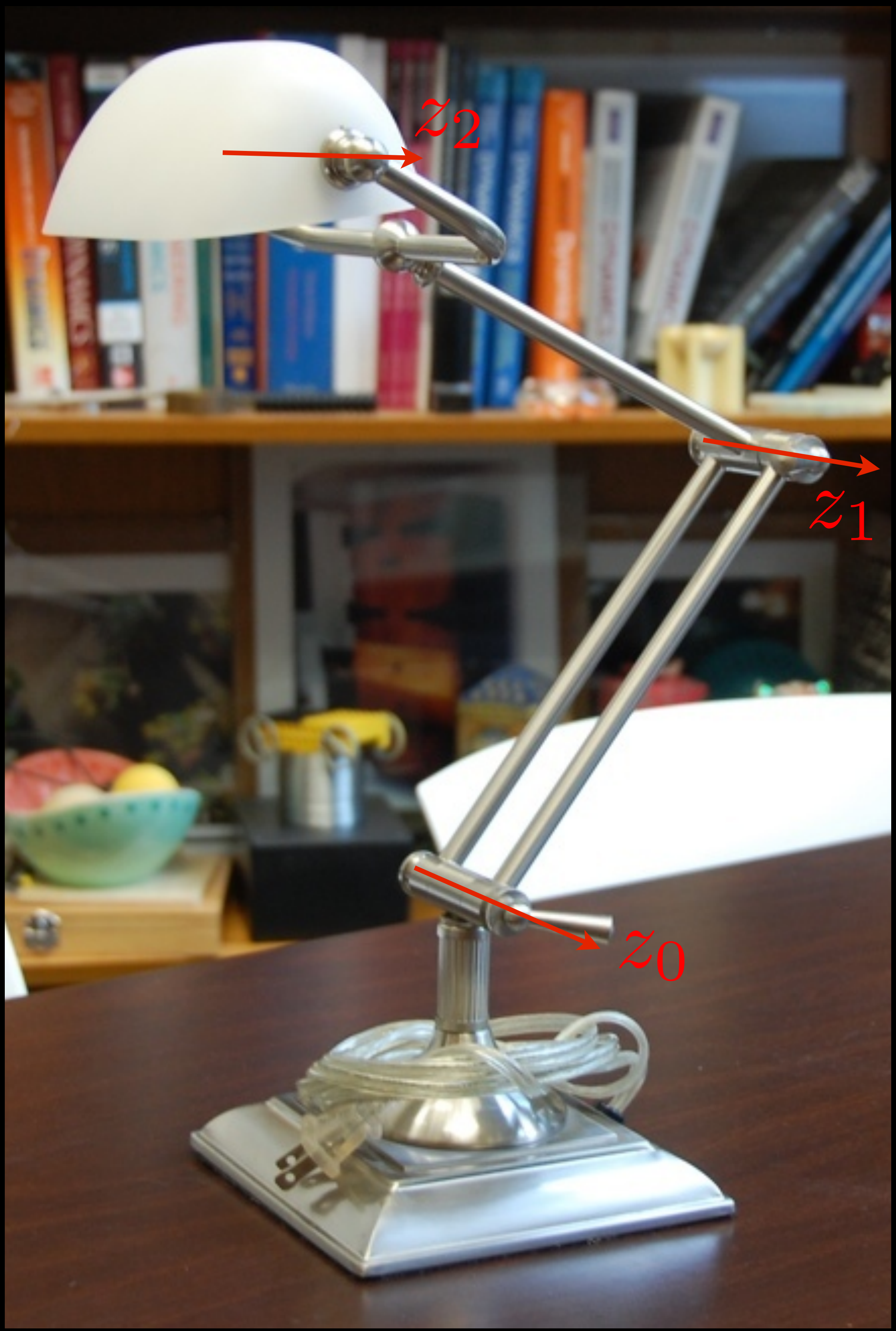
Rotation Matrices

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 3: September 5, 2013



Define a point on the shade along the final axis (z_2) as the end-effector location.

What is the **reachable workspace** of the lamp?

- Everywhere the point can go when joints are rotated.
- This is a two-dimensional shape.

What is the **dexterous workspace** of the lamp?

- If we consider motion only in 2D, dexterous is the same as reachable workspace.
- Considering 3D, there is none!

What if we let the **base move and rotate** on the table. What type of robot is the lamp now?

- **PPRRRR** manipulator
- Not all rotational axes are **parallel**
- **Spatial** mechanism

What is the **reachable workspace** of the lamp?

- Everywhere the shade can go when joints rotate, base slides.

What is the **dexterous workspace** of the lamp?

- There is none! The shade can't tilt left/right; z_5 always horizontal.

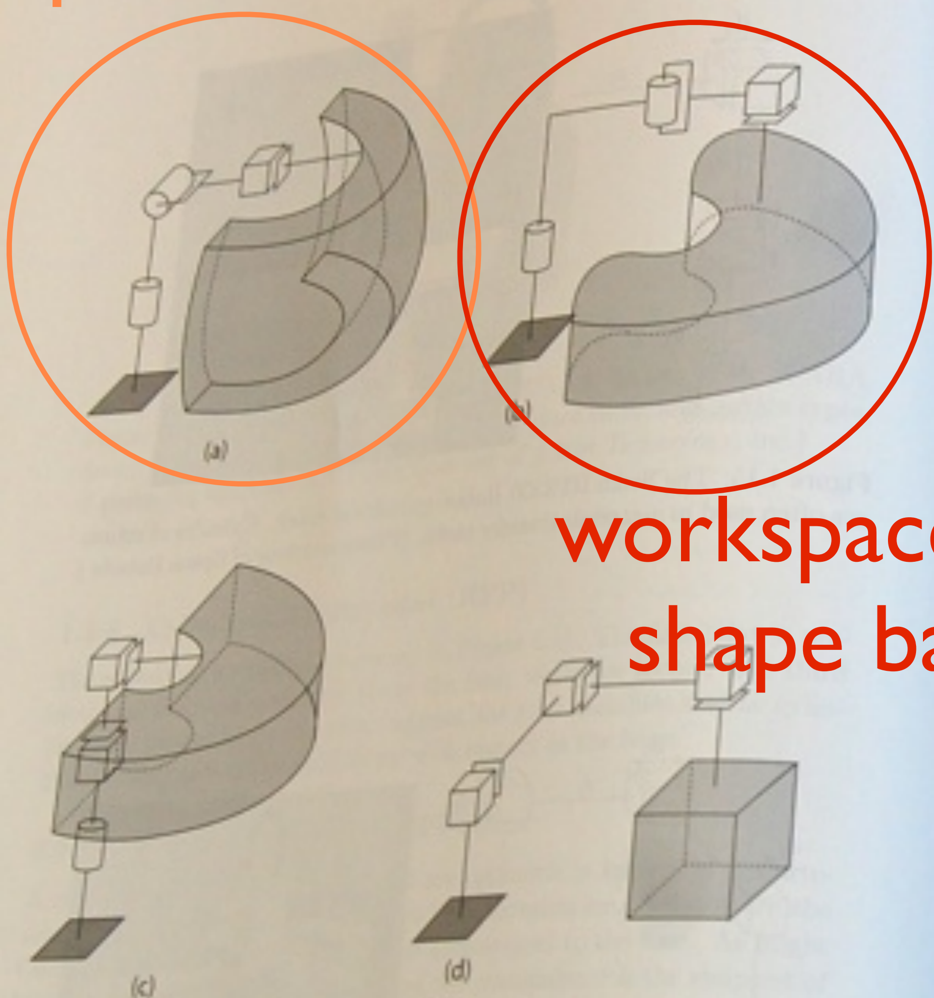


Figure 1.17: Comparison of the workspaces of the (a) spherical, (b) SCARA, (c) cylindrical, and (d) Cartesian robots. The nature of the workspace dictates the types of application for which each design can be used.

1.3.6 Parallel Manipulator

A **parallel manipulator** is one in which some subset of the links form a closed chain. More specifically, a parallel manipulator has two or more kinematic chains connecting the base to the end effector. Figure 1.18 shows the ABB IRB940 Tricept robot, which is a parallel manipulator. The closed-chain kinematics of parallel robots can result in greater structural rigidity, and hence greater accuracy, than open chain robots. The kinematic description of parallel robots is fundamentally different from that of serial link robots and therefore requires different methods of analysis.

Figure 1.18: The ABB IRB940 Tricept parallel robot. Parallel robots generally have much higher structural rigidity than serial link robots. (Photo courtesy of ABB.)

1.4 OUTLINE OF THE TEXT

A typical application involving an industrial manipulator is shown in Figure 1.19. The manipulator is shown with a grinding tool that it must use to remove a certain amount of metal from a surface. In the present text we are concerned with the following question: *What are the basic issues to be resolved and what must we learn in order to be able to program a robot to perform such tasks?* The ability to answer this question for a full six degree-of-freedom manipulator represents the goal of the present text. The answer is too complicated to be presented at this point. We can, however,

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

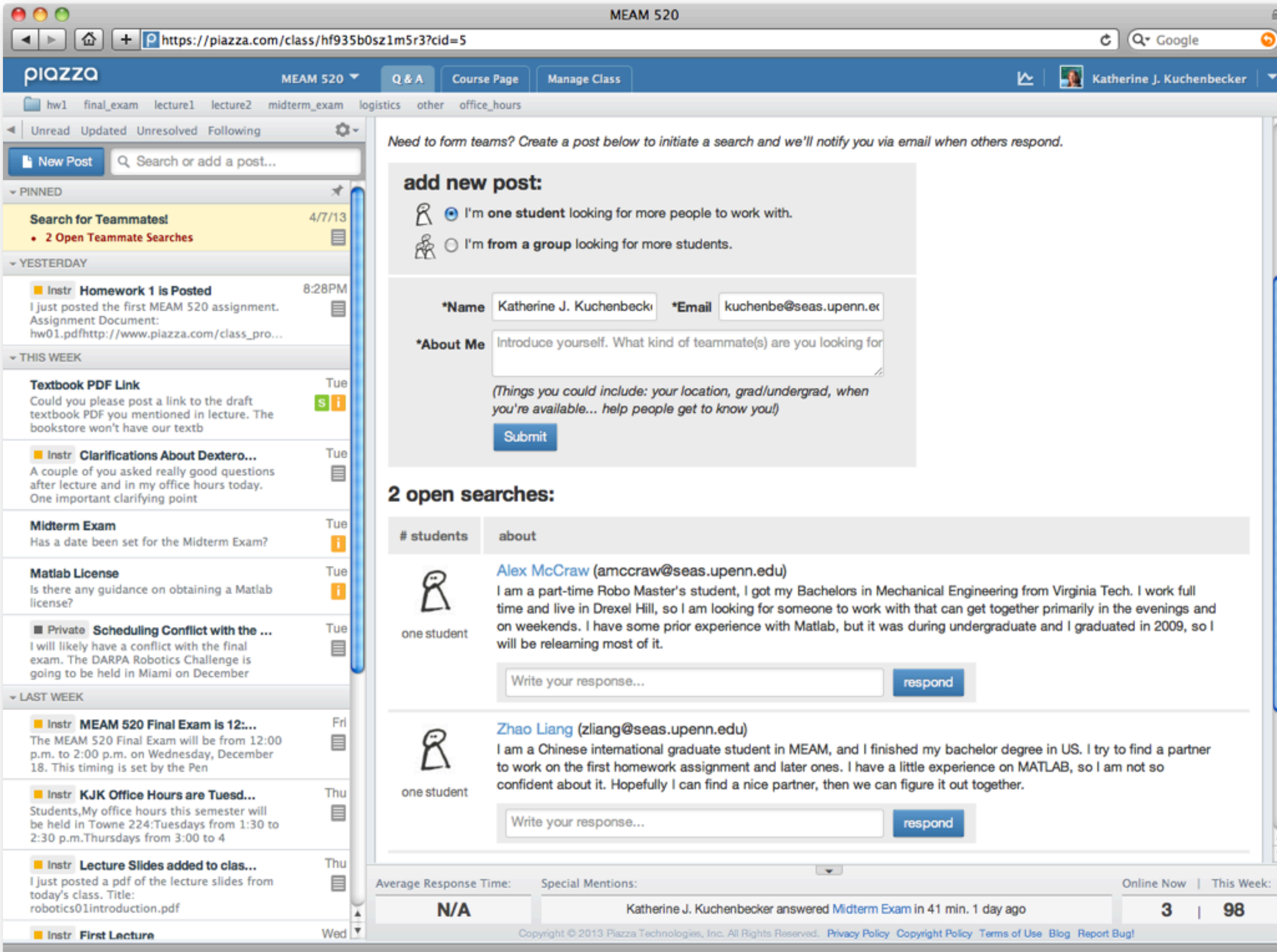
This class will use the programming language MATLAB to analyze and simulate robotic systems and also to control real robots. Some students in the class have never used MATLAB before, and others are quite familiar with it. The goal of this assignment is to get everyone starting to use MATLAB to improve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


MEAM 520
https://piazza.com/class/hf935b0sz1m5r3?cid=5
Google
piazza
MEAM 520
Q & A
Course Page
Manage Class
Katherine J. Kuchenbecker
hw1 final_exam lecture1 lecture2 midterm_exam logistics other office_hours
Unread Updated Unresolved Following
New Post
Search or add a post...
PINNED
Search for Teammates! 4/7/13
2 Open Teammate Searches
YESTERDAY
Instr Homework 1 is Posted 8:28PM
I just posted the first MEAM 520 assignment. Assignment Document: hw01.pdfhttp://www.piazza.com/class_pro...
THIS WEEK
Textbook PDF Link Tue
Could you please post a link to the draft textbook PDF you mentioned in lecture. The bookstore won't have our textb
Instr Clarifications About Dextero... Tue
A couple of you asked really good questions after lecture and in my office hours today. One important clarifying point
Midterm Exam Tue
Has a date been set for the Midterm Exam?
Matlab License Tue
Is there any guidance on obtaining a Matlab license?
Private Scheduling Conflict with the ... Tue
I will likely have a conflict with the final exam. The DARPA Robotics Challenge is going to be held in Miami on December
LAST WEEK
Instr MEAM 520 Final Exam is 12:... Fri
The MEAM 520 Final Exam will be from 12:00 p.m. to 2:00 p.m. on Wednesday, December 18. This timing is set by the Pen
Instr KJK Office Hours are Tuesd... Thu
Students,My office hours this semester will be held in Towne 224:Tuesdays from 1:30 to 2:30 p.m.Thursdays from 3:00 to 4
Instr Lecture Slides added to clas... Thu
I just posted a pdf of the lecture slides from today's class. Title: robotics01introduction.pdf
Instr First Lecture Wed
Need to form teams? Create a post below to initiate a search and we'll notify you via email when others respond.
add new post:
I'm one student looking for more people to work with.
I'm from a group looking for more students.
*Name Katherine J. Kuchenbeck
*Email kuchenbe@seas.upenn.ed
*About Me Introduce yourself. What kind of teammate(s) are you looking for
(Things you could include: your location, grad/undergrad, when you're available... help people get to know you!)
Submit
2 open searches:
students
about
Alex McCraw (amccraw@seas.upenn.edu)
I am a part-time Robo Master's student, I got my Bachelors in Mechanical Engineering from Virginia Tech. I work full time and live in Drexel Hill, so I am looking for someone to work with that can get together primarily in the evenings and on weekends. I have some prior experience with Matlab, but it was during undergraduate and I graduated in 2009, so I will be relearning most of it.
one student
Write your response...
respond
Zhao Liang (zliang@seas.upenn.edu)
I am a Chinese international graduate student in MEAM, and I finished my bachelor degree in US. I try to find a partner to work on the first homework assignment and later ones. I have a little experience on MATLAB, so I am not so confident about it. Hopefully I can find a nice partner, then we can figure it out together.
one student
Write your response...
respond
Average Response Time:
N/A
Special Mentions:
Katherine J. Kuchenbecker answered Midterm Exam in 41 min. 1 day ago
Online Now | This Week:
3 | 98
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.

- 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.

MATLAB

Completing this assignment requires 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`

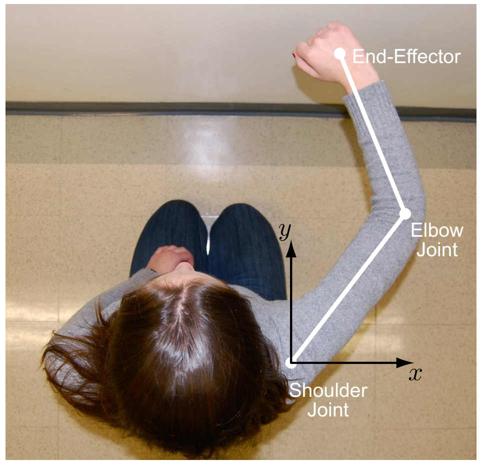
SEAS is still planning to pay for MATLAB licenses for all students in this class, but the software distribution system has not yet been set up. It will be announced on Piazza as soon as it's available.

If you haven't used MATLAB much before, you should consider attending one of the MATLAB workshops that were announced in lecture. We'll announce any additional ones once they are organized.

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

Plotting the Horizontal Human Arm's Reachable Workspace

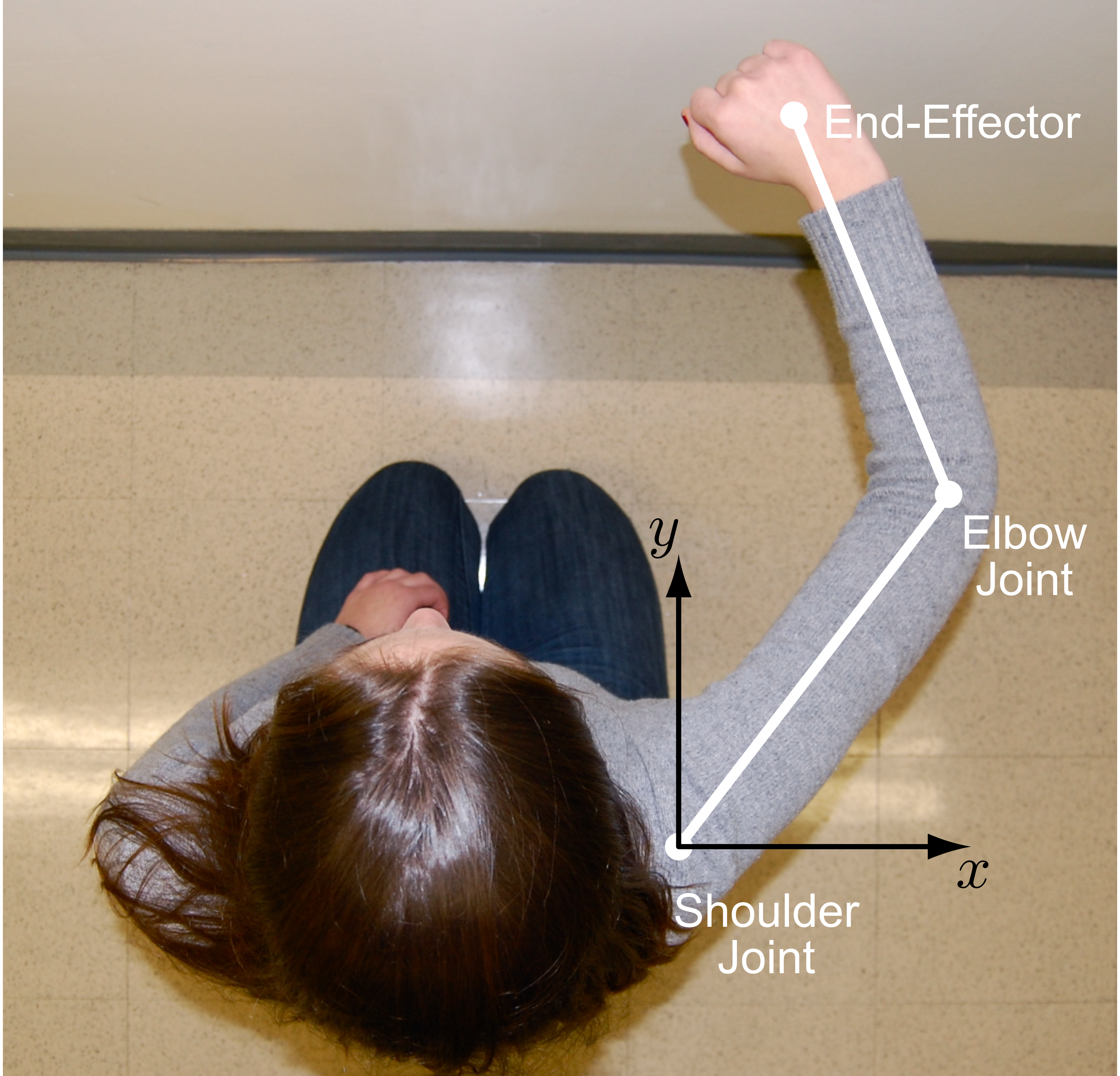
Your task for this homework is to write a MATLAB script that plots the **reachable workspace of your right arm**, as seen from overhead. Recall that the reachable workspace is the entire set of points that a manipulator's end-effector can reach.



As shown in the illustration above, you should treat the human arm as a planar RR manipulator. This decision models the shoulder and the elbow both as revolute joints whose rotational axes are vertical. Treat the center of the hand as the end-effector location.

Put the origin of the coordinate frame at the shoulder joint. The x axis points out to your right, and the y axis points straight forward.

How far can your elbow flex and extend? What about your shoulder's range of motion in the horizontal plane? Examine your own arm to estimate the mechanical limits of each joint's movement. Also measure your arm to obtain the appropriate link lengths in centimeters. If you're in a team, you may use either partner's arm for this step. Write down the values you find; you'll need them in your code.


End-Effector
Elbow
Joint
y
x
Shoulder
Joint

Look on pages 20 to 21 of the textbook to find equations for the x and y coordinates of the end-effector of a two-link planar RR robot as functions of the joint angles θ_1 and θ_2. Specifically, you should be looking at equations (1.1) and (1.2) and the associated diagram in Figure 1.20.

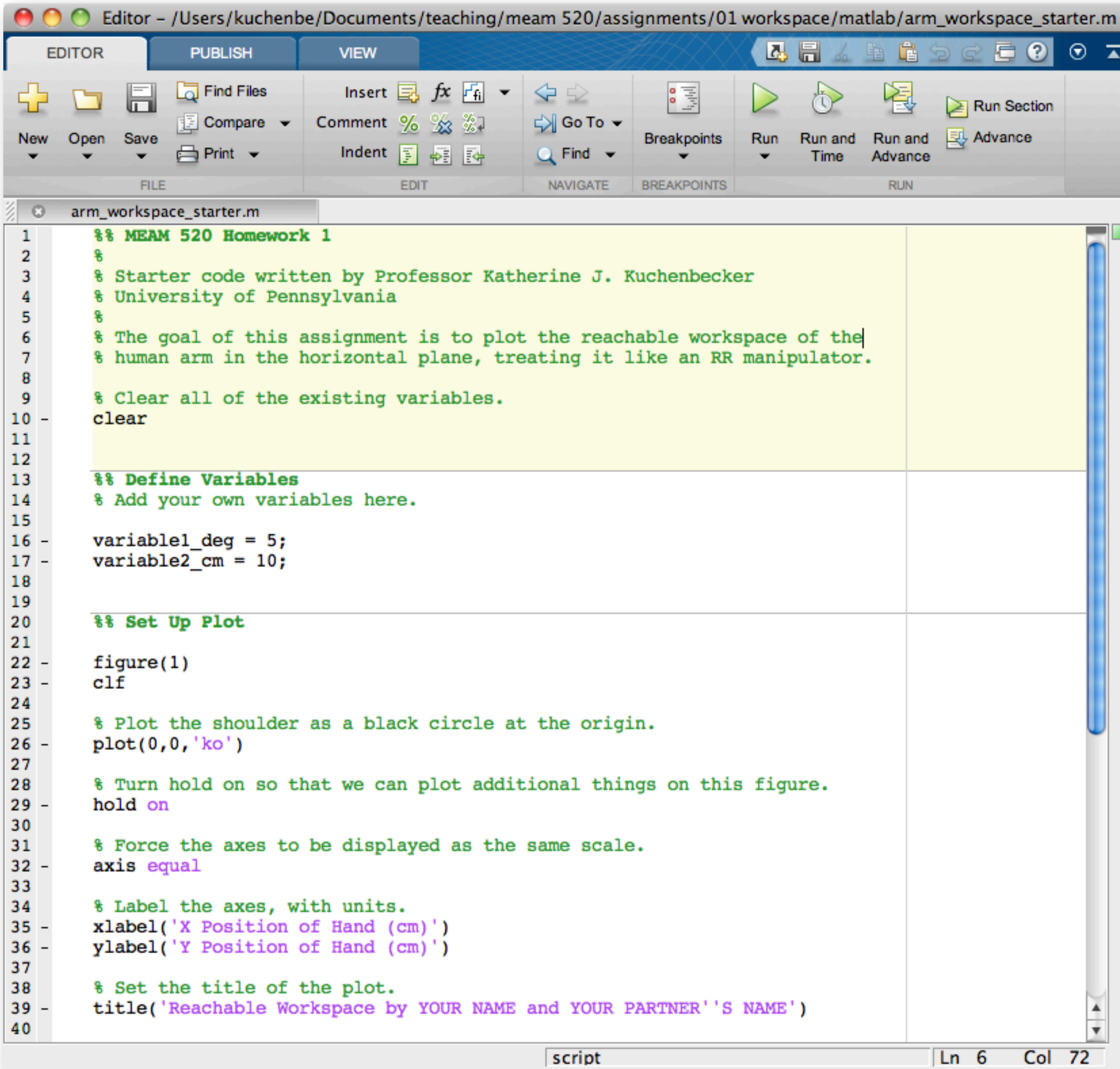
Combine all of these pieces to write a MATLAB script to plot the reachable workspace of your right arm. Here are a few additional tips and requirements:

- Start by downloading the **starter code** from Piazza. The filename is `arm_workspace_starter.m`
- Rename your file `arm_workspace_yourpennkey.m` or `arm_workspace_pennkey1_pennkey2.m` (if you're working with a partner). Your PennKey is the first part of your SEAS email address. Naming your file in this way ensures that the submissions all have unique, identifiable file names.
- Inside your script, create **sensibly named variables** to hold the values you measured above, instead of typing the numbers directly into equations. For example, the variable `elbowmin_deg` could hold the minimum angle your elbow can reach in degrees, and the variable `forearmlength_cm` could hold the length of your forearm in centimeters. Then use the name of the variable every time you need that value. This practice makes your code easier to read, debug, and update later.
- Plot a **black circle** at the location of the shoulder joint. This is already done in the starter code.
- Instead of plotting continuous regions, we recommend that you just plot a **finite set of points** to show the reachable workspace of the human arm. The starter code plots a simple set of points to show you how to do this.
- Make the plot marker **green** at locations where the end-effector can physically go and **red** at locations where the end-effector encounters a collision with your body. Use reasonable measurements for the location of your body in the workspace.
- Use units of **centimeters** in both directions.
- Note that the MATLAB functions `sin` and `cos` expect arguments in **radians**. If you want to use **degrees**, try the functions `sind` and `cosd`.
- Ensure that the view of the workspace is not distorted by calling the MATLAB function `axis equal` after you've plotted at least one thing in the figure window; this step is already done in the starter code. This function makes one unit in the x-direction be displayed at the same visual scale as one unit in the y-direction.
- For the **title** of your plot, use "Reachable Workspace by Your Name" or "Reachable Workspace by Teammate 1 and Teammate 2", inserting your full given name(s).

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* or *Homework 1: 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.


Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/01 workspace/matlab/arm_workspace_starter.m
EDITOR
PUBLISH
VIEW
New
Open
Save
Find Files
Compare
Print
FILE
Insert
Comment
Indent
EDIT
Go To
Find
NAVIGATE
Breakpoints
BREAKPOINTS
Run
Run and Time
Run and Advance
Run Section
Advance
RUN
arm_workspace_starter.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
% Add your own variables here.

variable1_deg = 5;
variable2_cm = 10;

%% Set Up Plot

figure(1)
clf

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

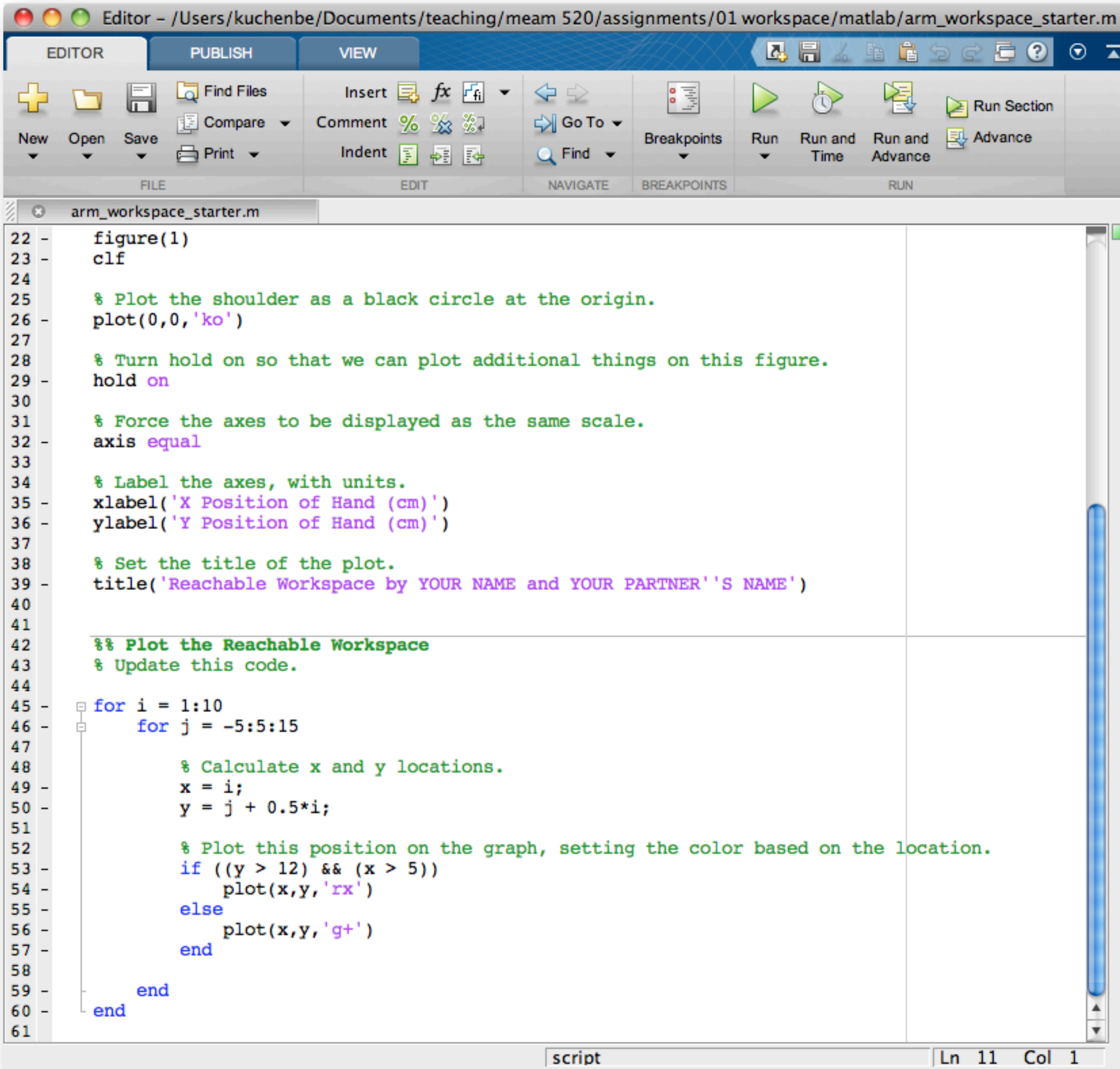
% Turn hold on so that we can plot additional things on this figure.
hold on

% 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 YOUR NAME and YOUR PARTNER''S NAME')

script
Ln 6
Col 72


Editor – /Users/kuchenbe/Documents/teaching/meam 520/assignments/01 workspace/matlab/arm_workspace_starter.m
EDITOR
PUBLISH
VIEW
New
Open
Save
Find Files
Compare
Print
FILE
Insert
Comment
Indent
EDIT
Go To
Find
NAVIGATE
Breakpoints
BREAKPOINTS
Run
Run and Time
Run and Advance
Run Section
Advance
RUN
arm_workspace_starter.m
figure(1)
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 YOUR NAME and YOUR PARTNER''S NAME')
%% Plot the Reachable Workspace
% Update this code.
for i = 1:10
for j = -5:5:15
% Calculate x and y locations.
x = i;
y = j + 0.5*i;
% Plot this position on the graph, setting the color based on the location.
if ((y > 12) && (x > 5))
plot(x,y,'rx')
else
plot(x,y,'g+')
end
end
end
script
Ln 11 Col 1

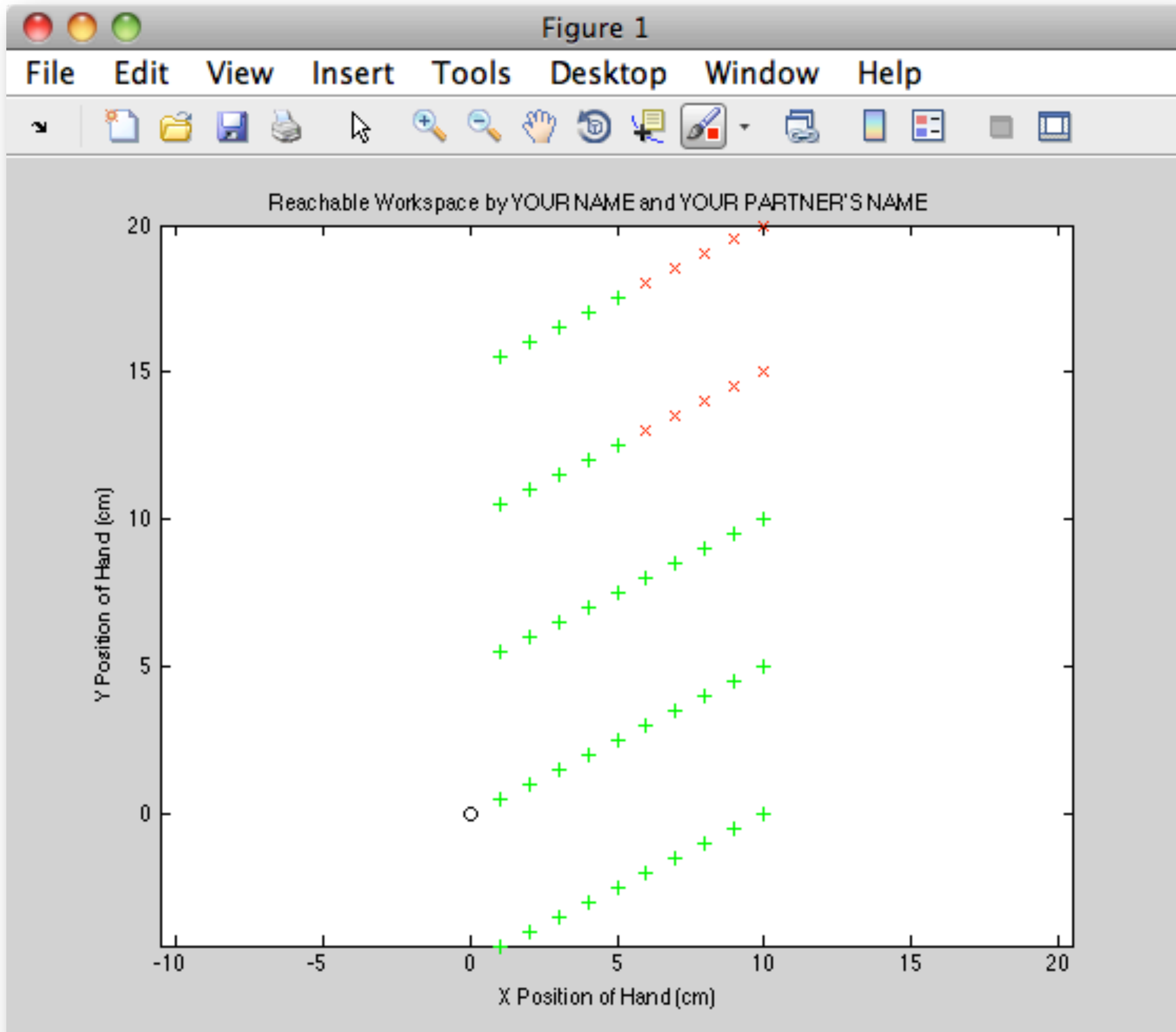
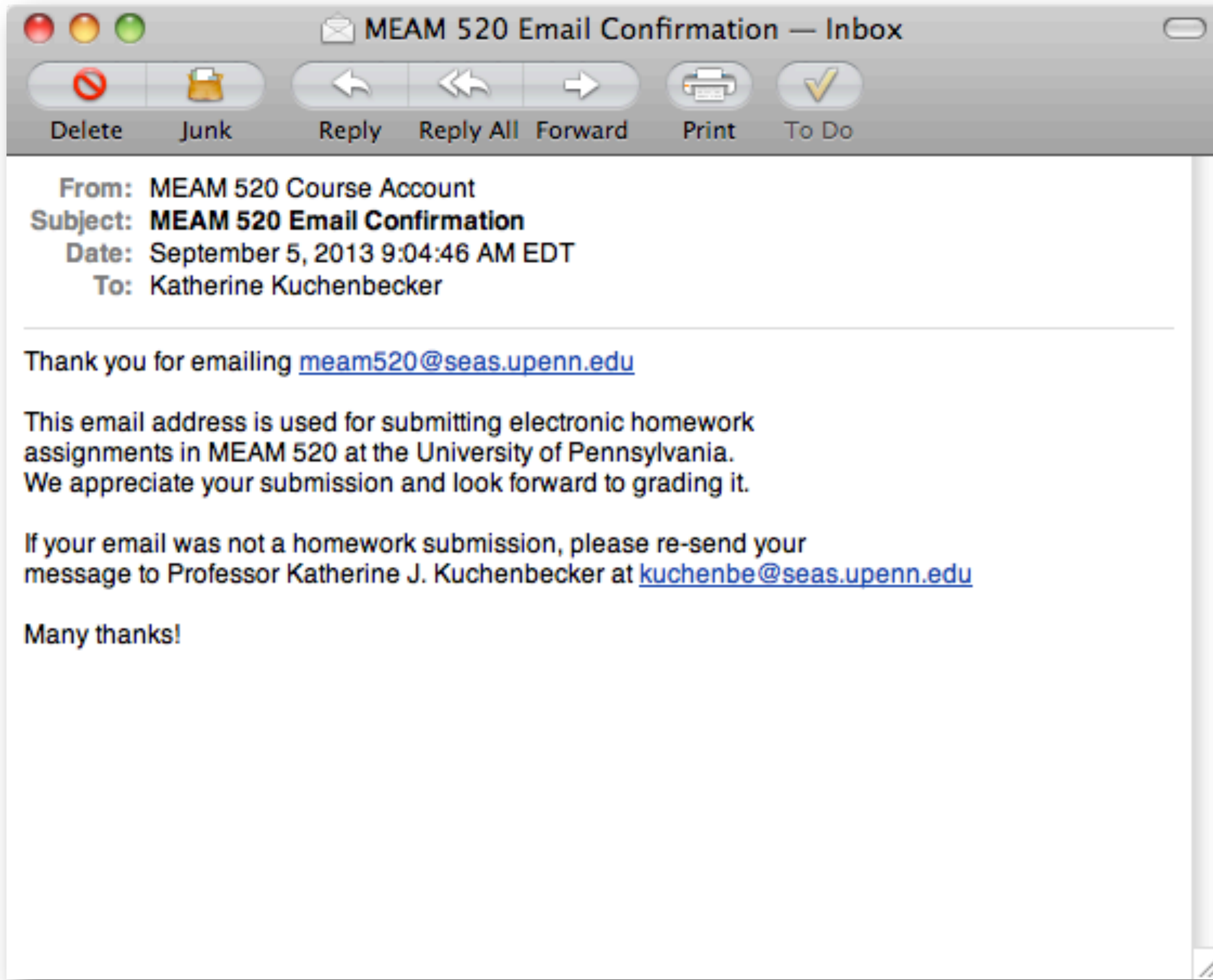


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Reachable Workspace by YOUR NAME and YOUR PARTNER'S NAME
Y Position of Hand (cm)
X Position of Hand (cm)
20
15
10
5
0
-10
-5
0
5
10
15
20



MEAM 520 Email Confirmation — Inbox

Delete Junk Reply Reply All Forward Print To Do

From: MEAM 520 Course Account
Subject: **MEAM 520 Email Confirmation**
Date: September 5, 2013 9:04:46 AM EDT
To: Katherine Kuchenbecker

Thank you for emailing meam520@seas.upenn.edu

This email address is used for submitting electronic homework assignments in MEAM 520 at the University of Pennsylvania. We appreciate your submission and look forward to grading it.

If your email was not a homework submission, please re-send your message to Professor Katherine J. Kuchenbecker at kuchenbe@seas.upenn.edu

Many thanks!

- Naomi will run my office hours today, still from 3:00 to 4:00 p.m. in Towne 224
- Our teaching team will hold other office hours this weekend to help you with Homework 1.
- OH times and locations will be announced on Piazza.

Naomi Fitter

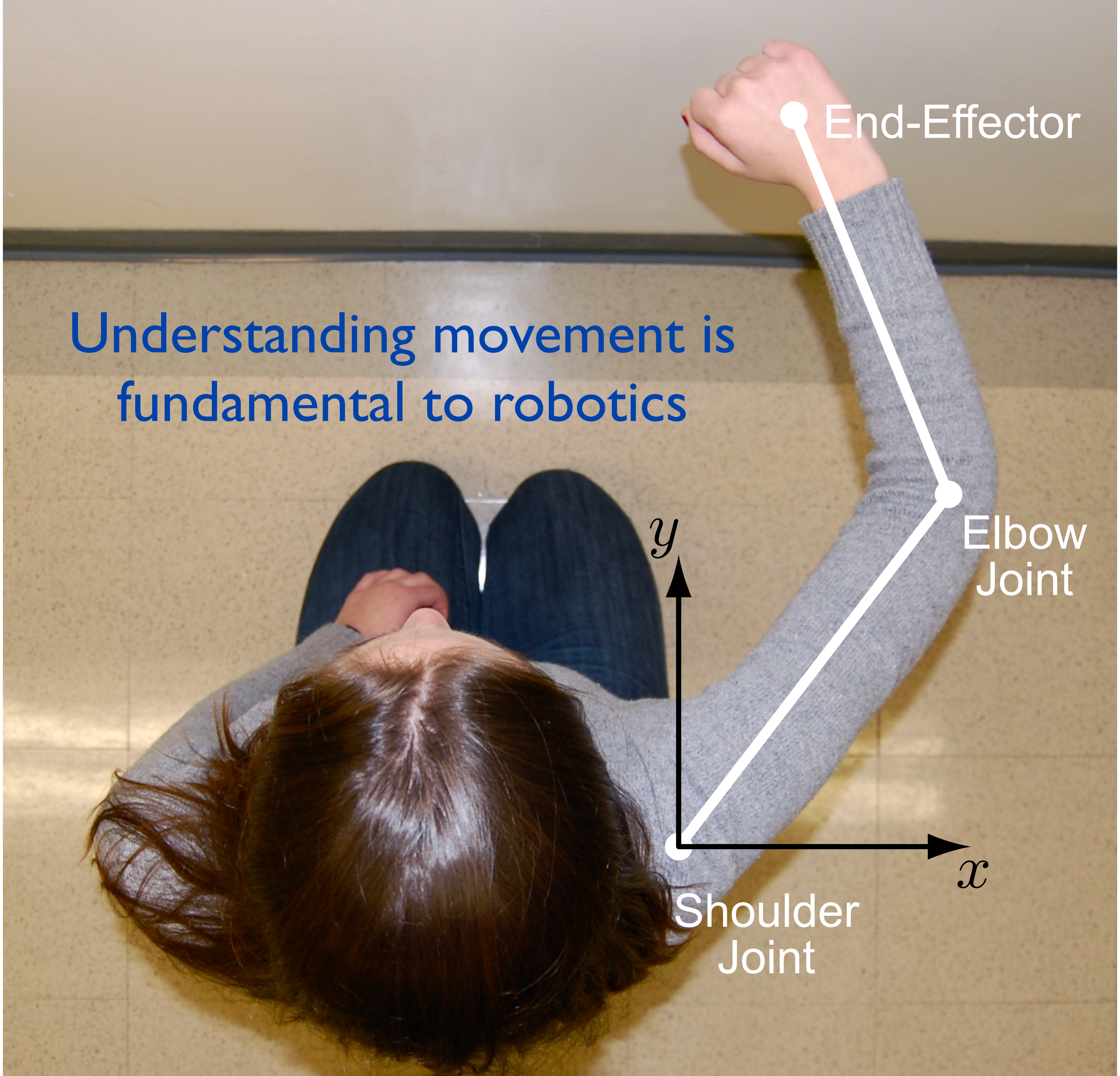
Tyler Barkin

Samarth Brahmbhatt

Chaitanya Bhargava

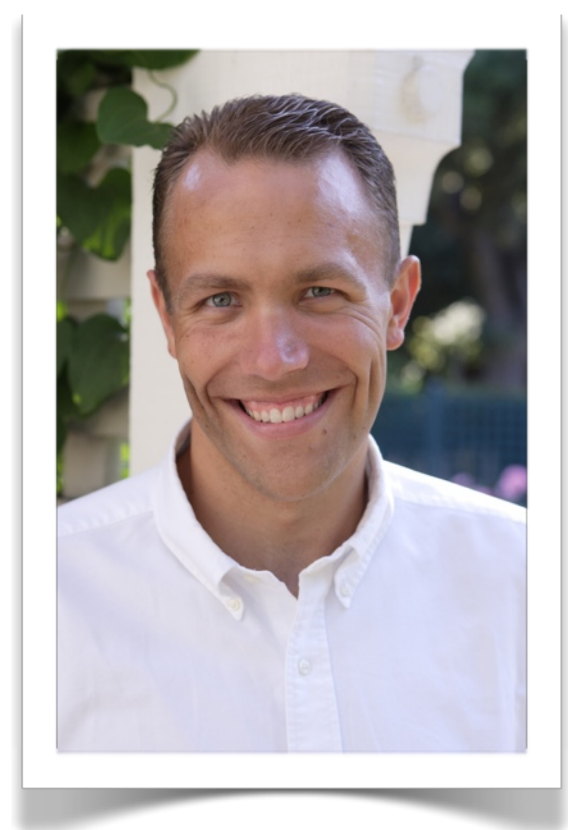
Edward (Yunkai) Cui

Annie Mroz


Understanding movement is fundamental to robotics
End-Effector
Elbow Joint
y
x
Shoulder Joint

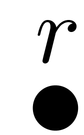
Points & Frames

These slides are adapted from ones created by Jonathan Fiene for MEAM 520 in Spring 2012.

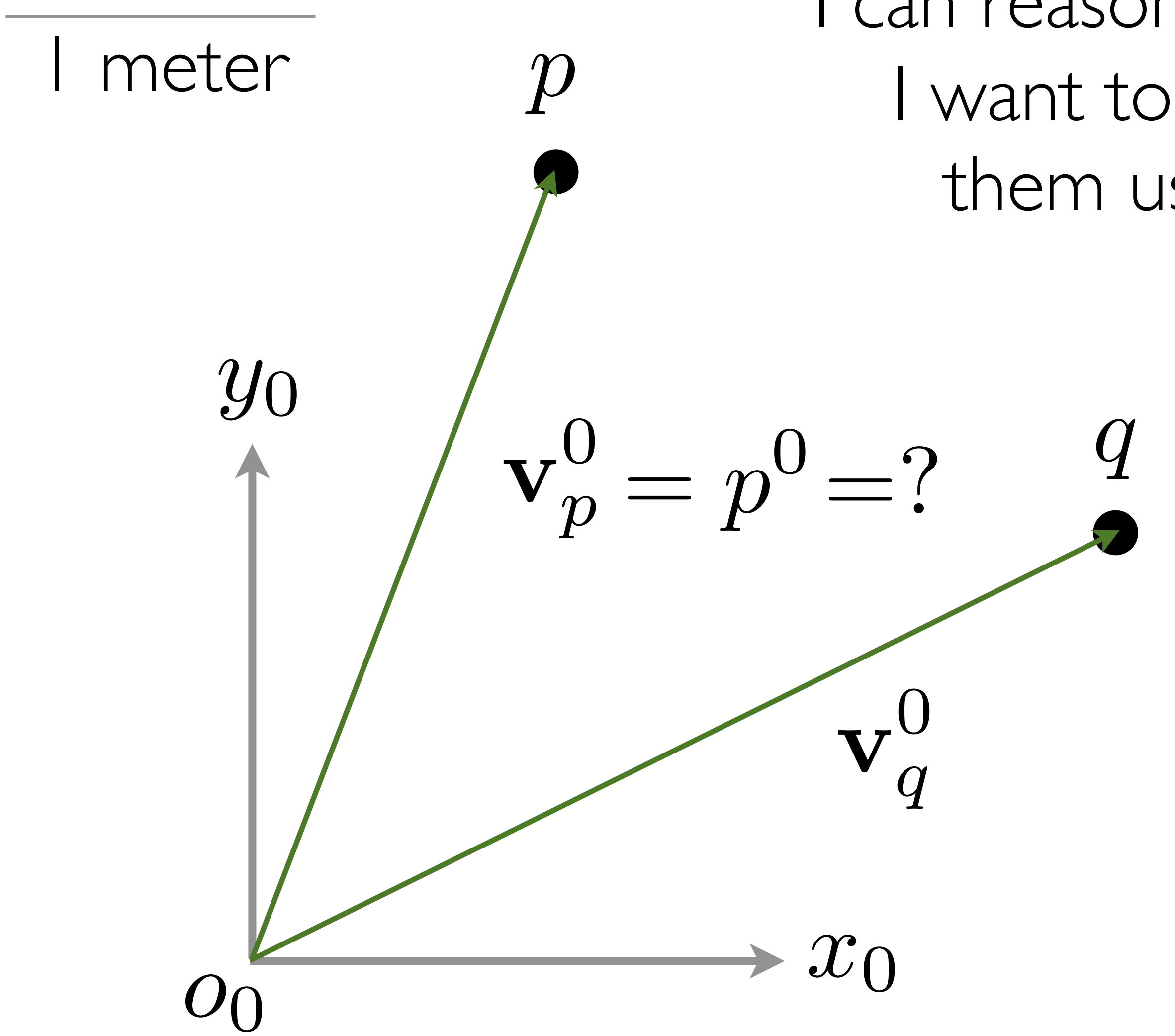


Representing positions

A **point** exists in space as a geometric entity



I can reason directly about these points, but if I want to **analyze** them, I must represent them using coordinates or equations.



The coordinate frame being used is designated using **superscript** notation

"It's *super* to specify the frame!"

What if we were doing this in **3D**.

How would the coordinate frame change?

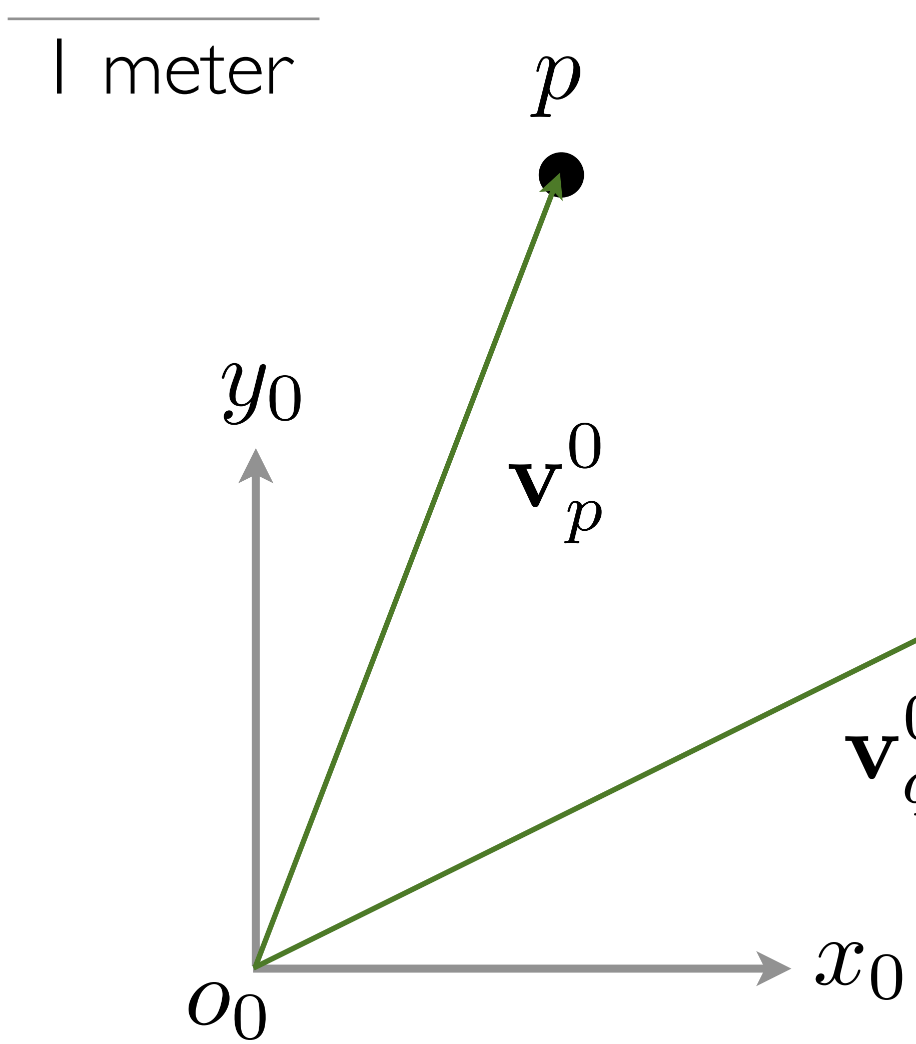
How would the coordinates change?

Coordinate Frame

Needs an origin (a single point in space)

and two or three orthogonal coordinate axes

How do I find the **magnitude** or **length** or **norm** of a vector?



$$\mathbf{v}_p^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \mathbf{v}_p^0 = [x, y, z]^T$$

$$||\mathbf{v}_p^0|| = \sqrt{x^2 + y^2 + z^2}$$

$$||\mathbf{v}_p^0|| = (x^2 + y^2 + z^2)^{\frac{1}{2}}$$

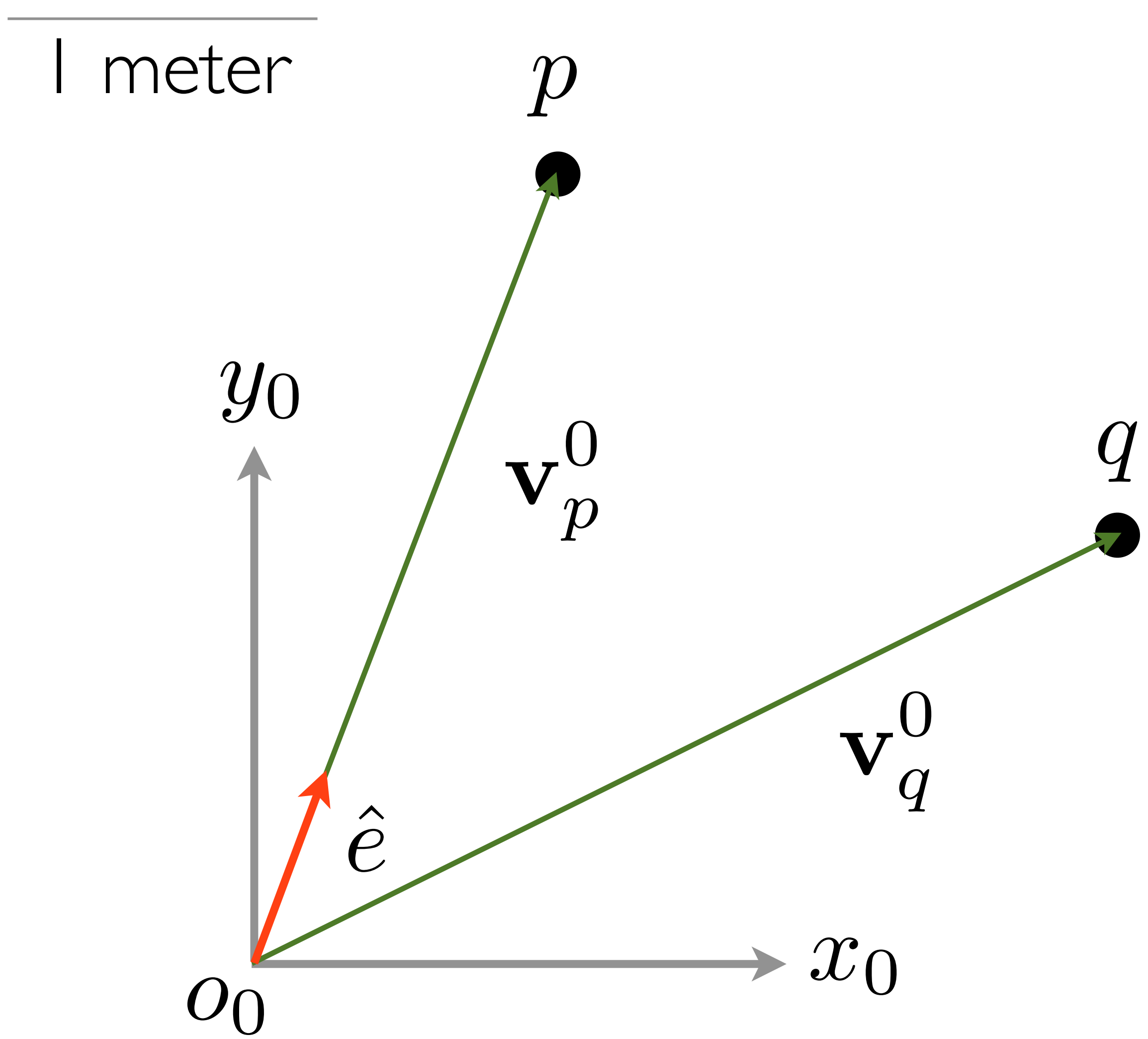
$$||\mathbf{v}_p^0|| = \left((\mathbf{v}_p^0)^T \mathbf{v}_p^0\right)^{\frac{1}{2}}$$

$$||\mathbf{v}_p^0|| = \langle \mathbf{v}_p^0, \mathbf{v}_p^0 \rangle^{\frac{1}{2}}$$

In MATLAB?

Vectors

How do I represent the **direction** of a vector?



$$\mathbf{v}_p^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \mathbf{v}_p^0 = [x, y, z]^T$$

Create a **unit vector**.

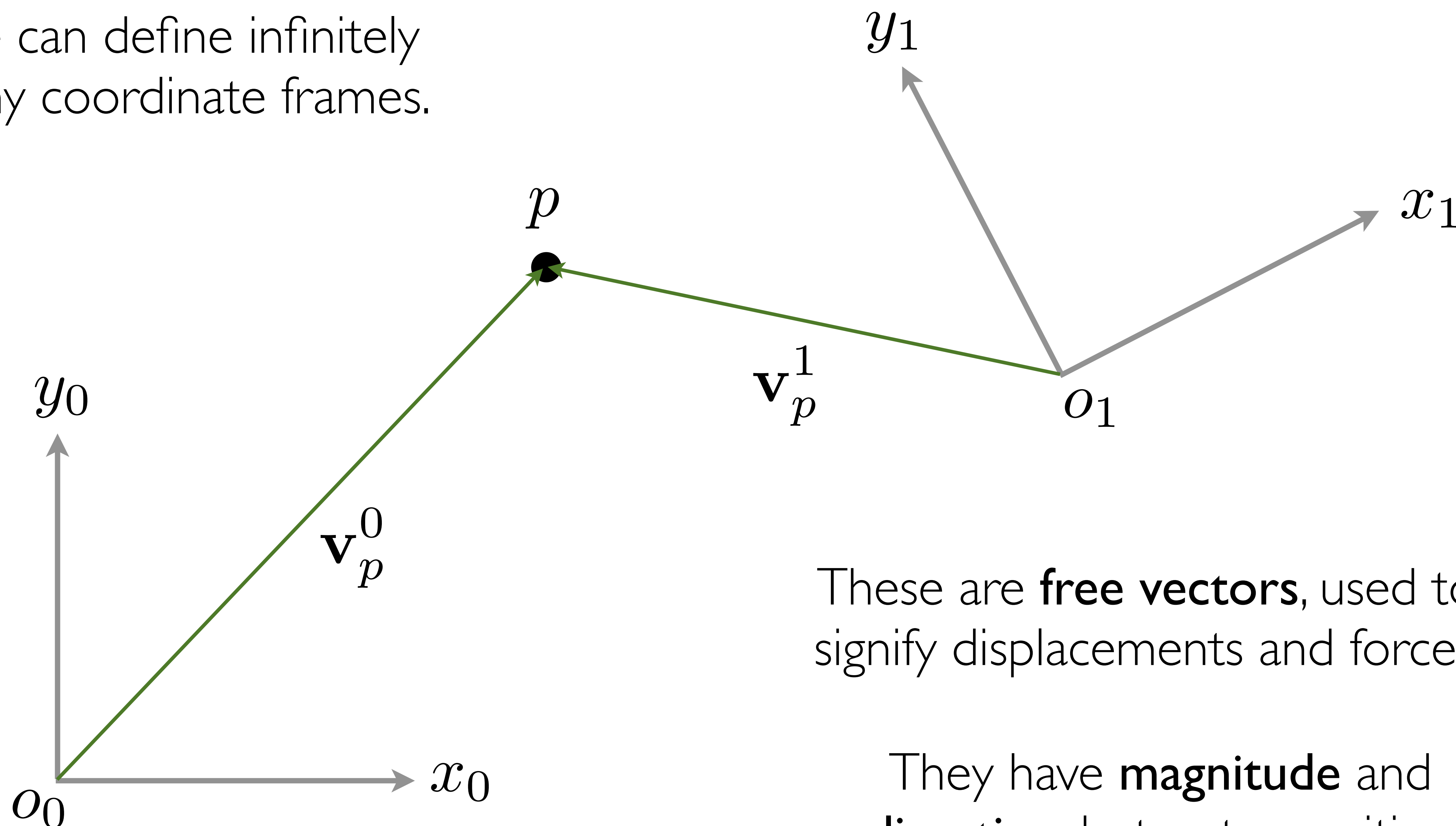
$$\hat{e} = \frac{\mathbf{v}_p^0}{||\mathbf{v}_p^0||}$$

In MATLAB?

Multiple coordinate frames

The **superscript** should make you turn your head to consider world from the orientation of the coordinate frame.

We can define infinitely many coordinate frames.

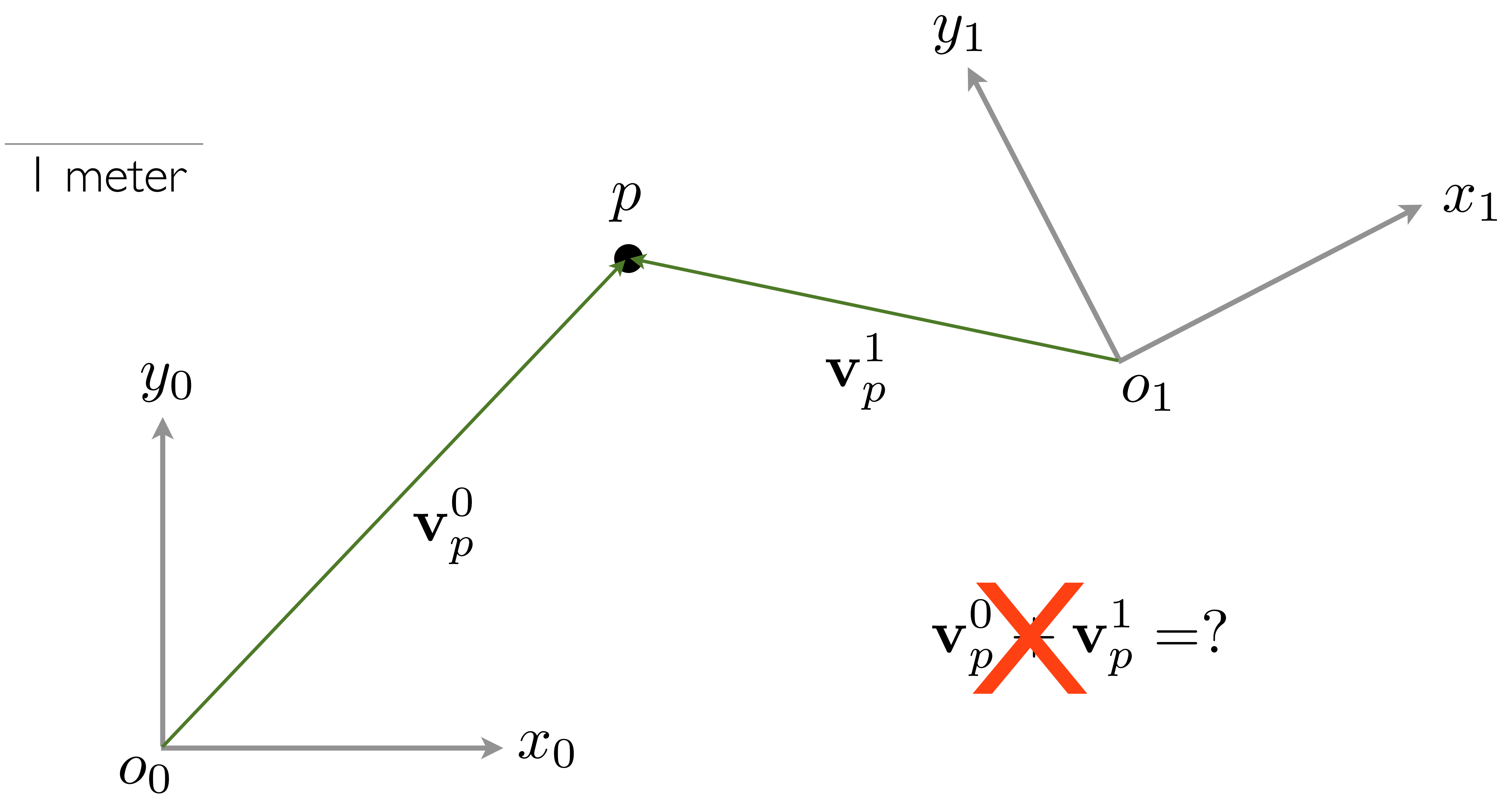


These are **free vectors**, used to signify displacements and forces.

They have **magnitude** and **direction**, but not a position from which they start.

We draw them at different locations for convenience.

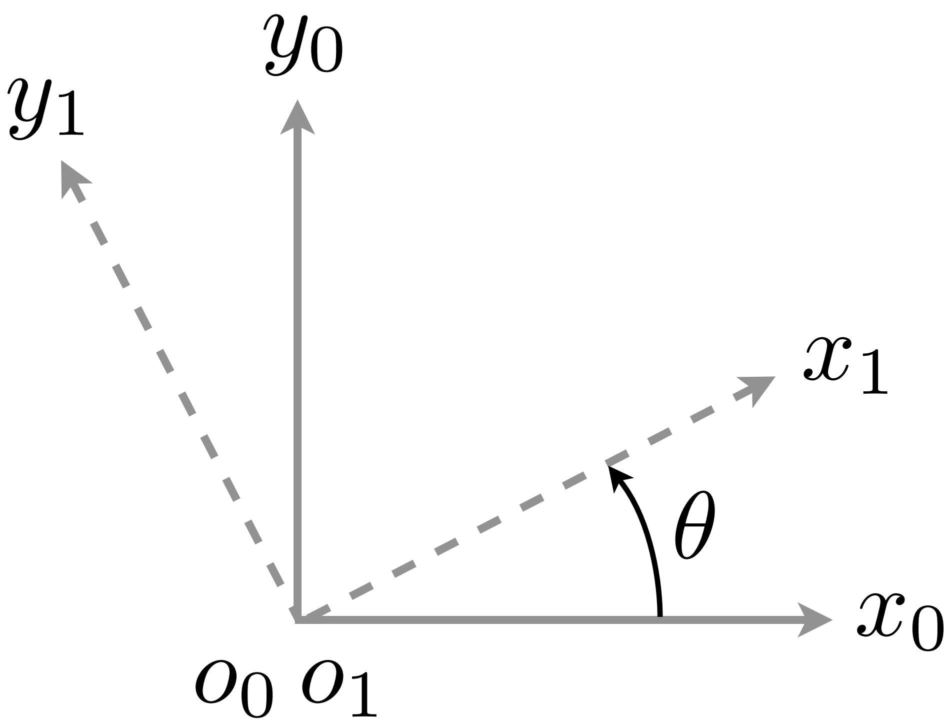
Multiple coordinate frames



To perform algebraic manipulation, you must express vectors in the same frame or in parallel frames

Rotation Matrices

Planar Coordinate Rotations



project frame 1 into frame 0

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

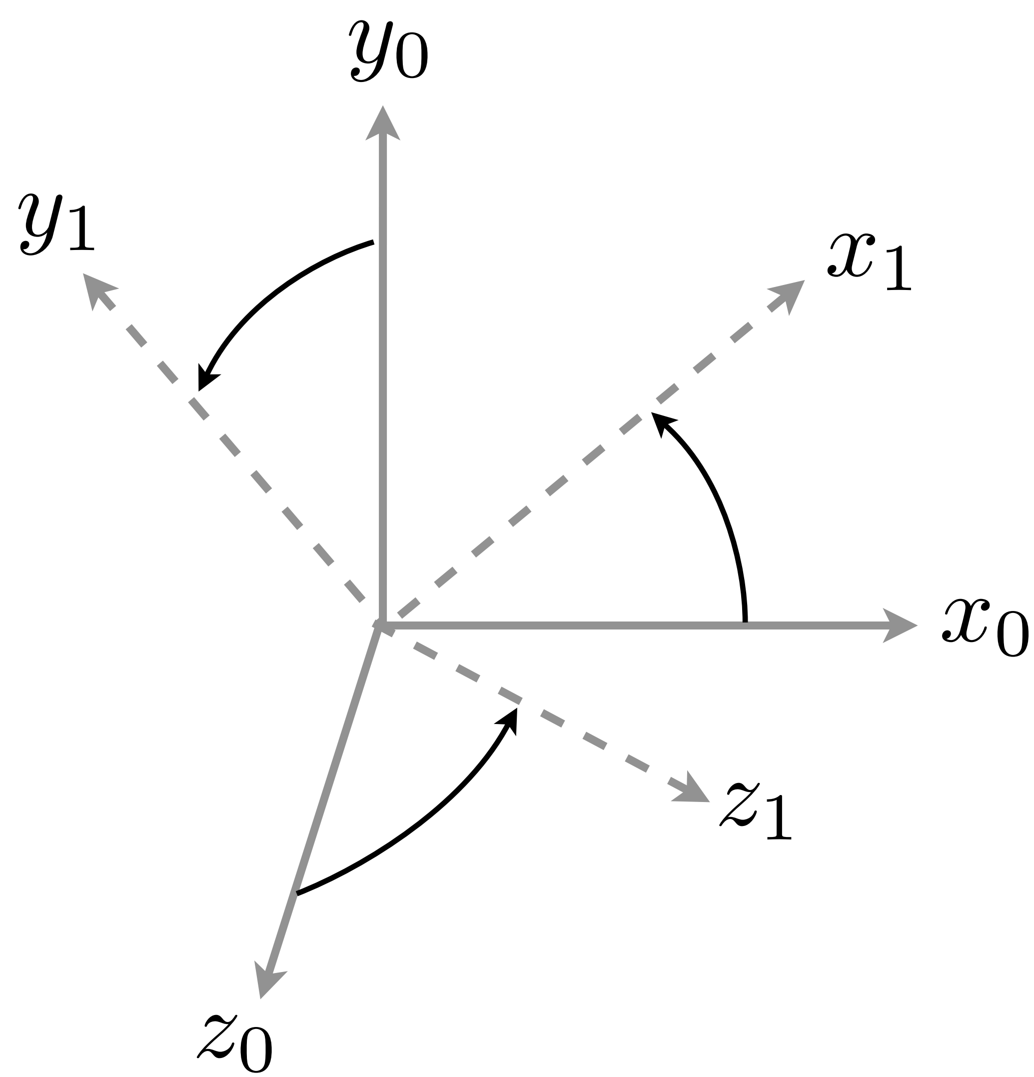
$$\mathbf{y}_1^0 = \begin{bmatrix} y_1 \cdot x_0 \\ y_1 \cdot y_0 \end{bmatrix} = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

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

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



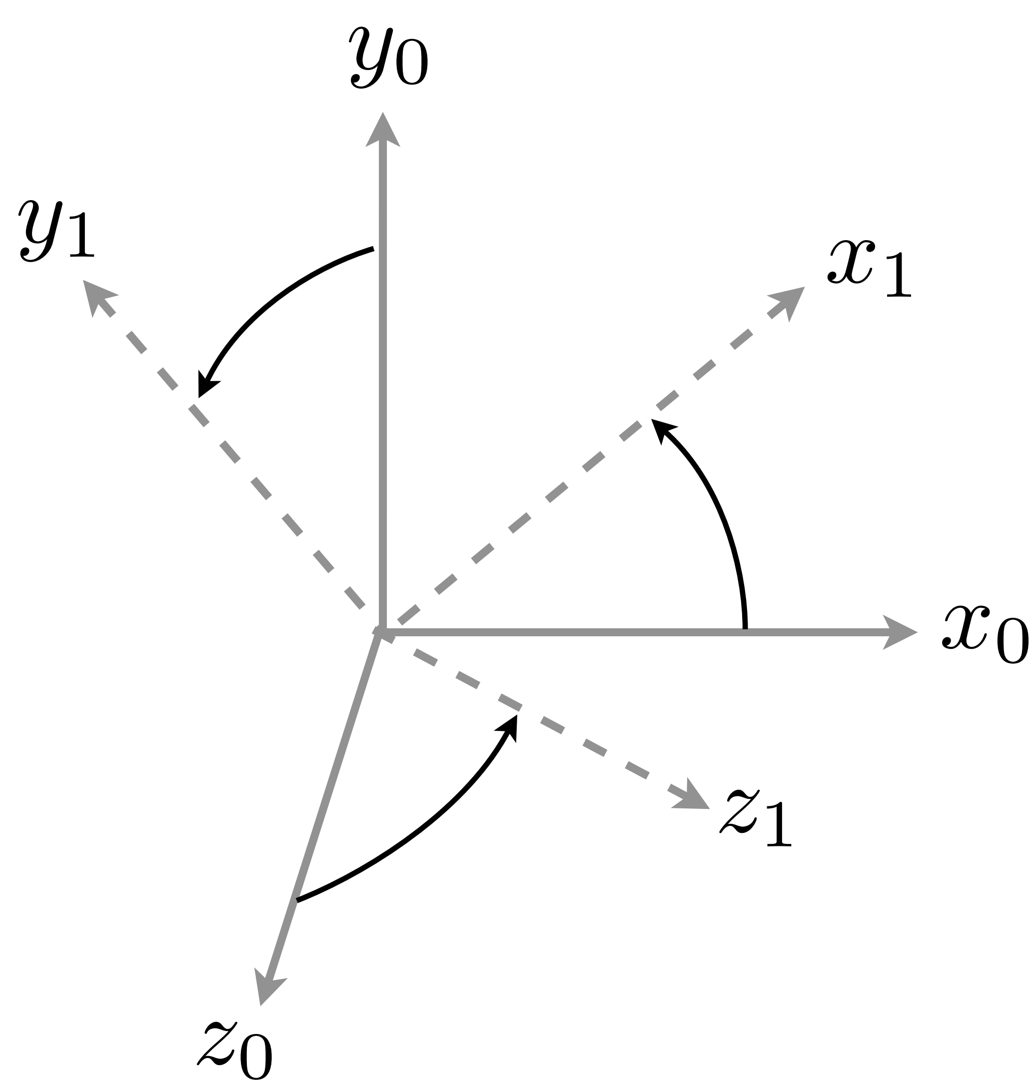
Represent orientation of one frame with respect to another frame

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

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

SO(3)

Special Orthogonal group of order 3



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

Represent orientation of one frame with respect to another frame

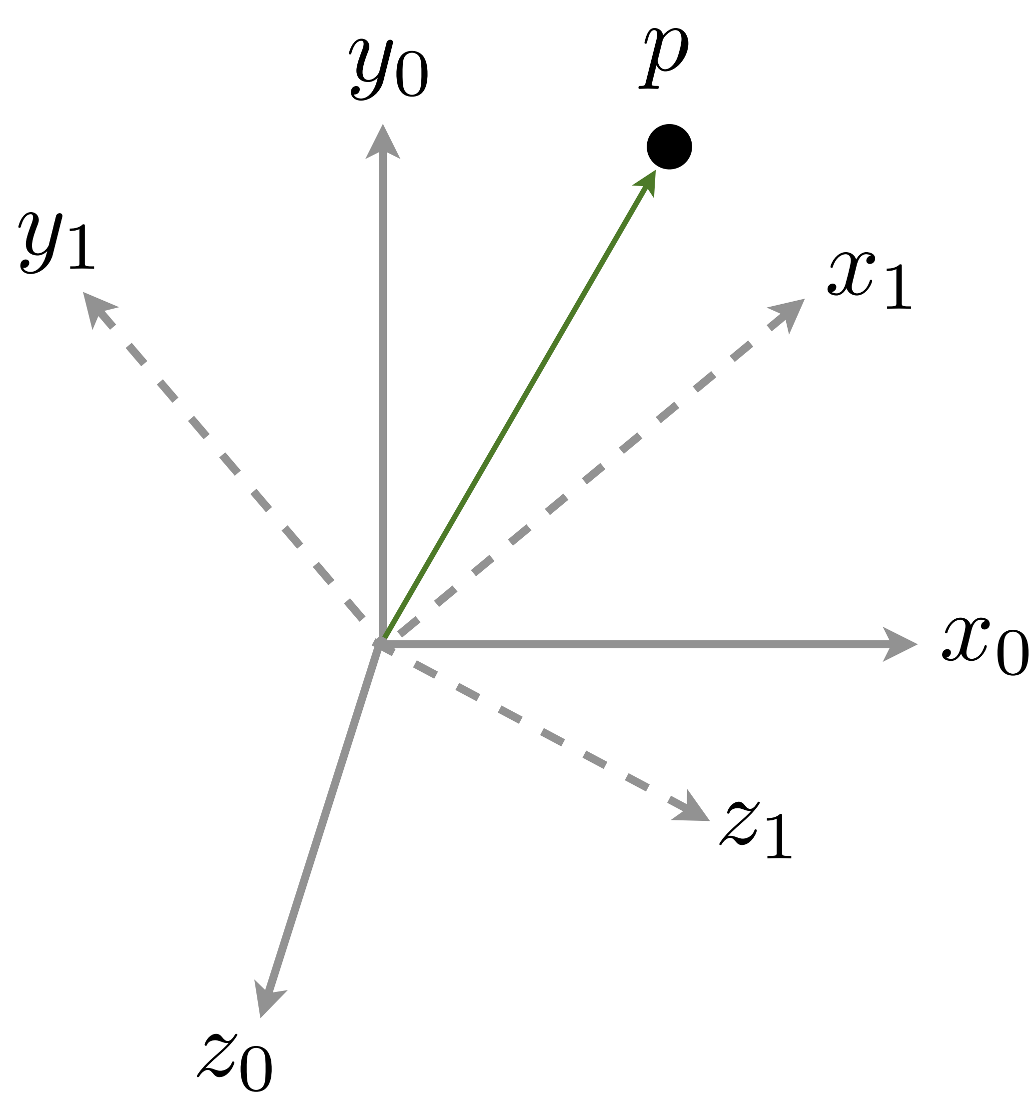
$$\mathbf{R}_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}$$

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

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

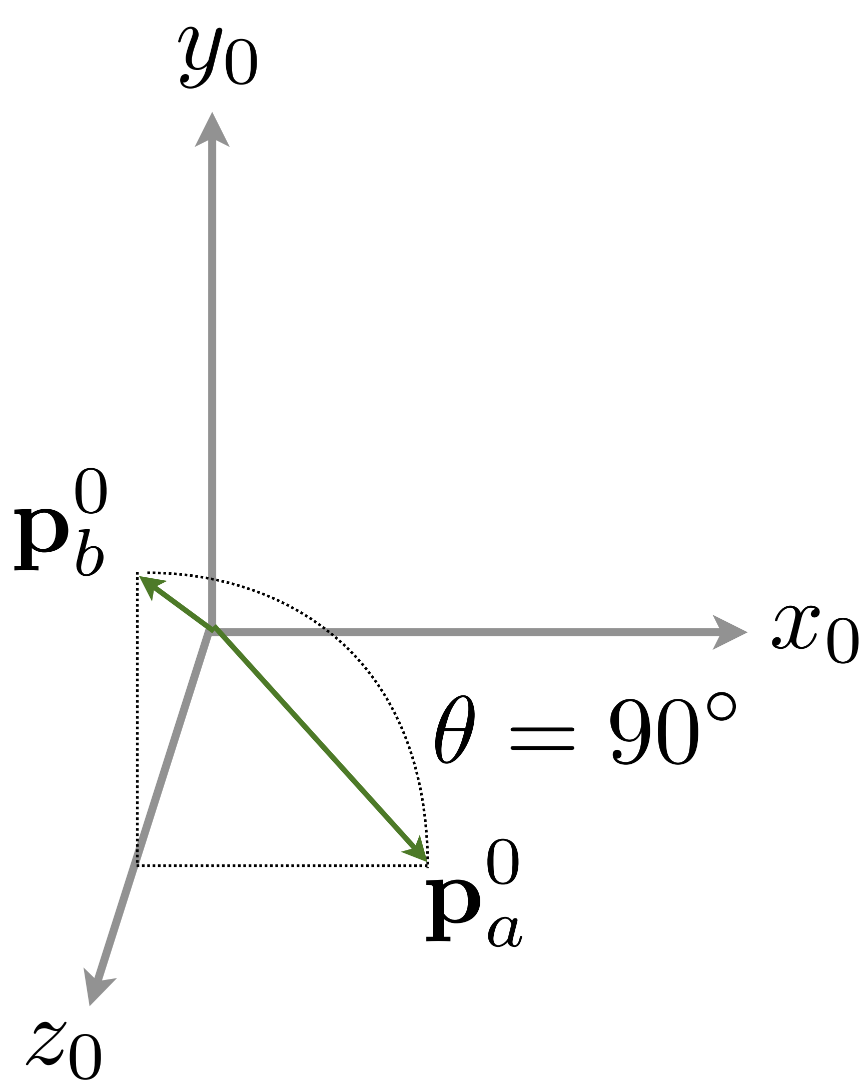
$$\mathbf{R}_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotational Transformations



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

$$\mathbf{p}_b^0 = \mathbf{R}\,\mathbf{p}_a^0$$

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

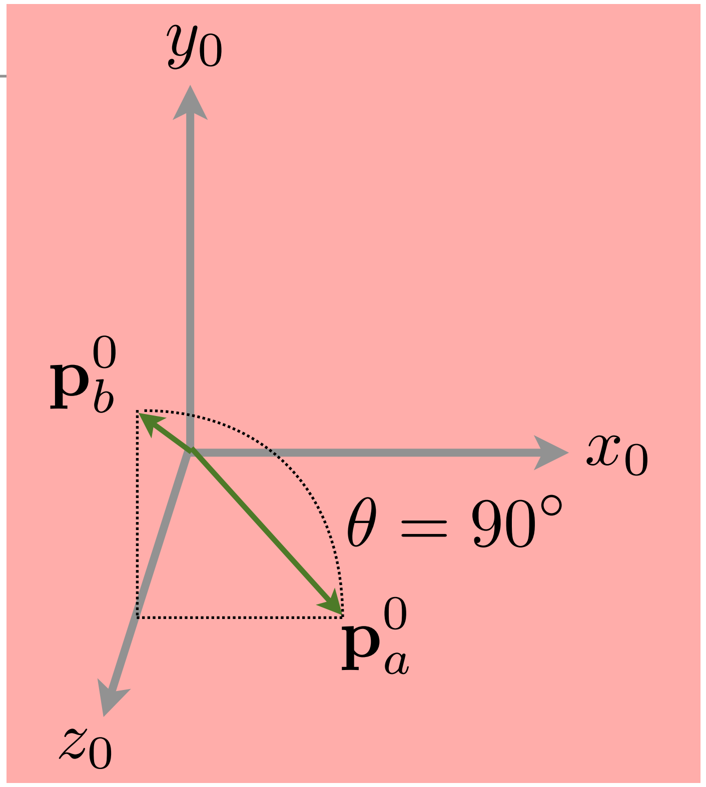
$$\mathbf{R}_{z,90^\circ} = ?$$

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

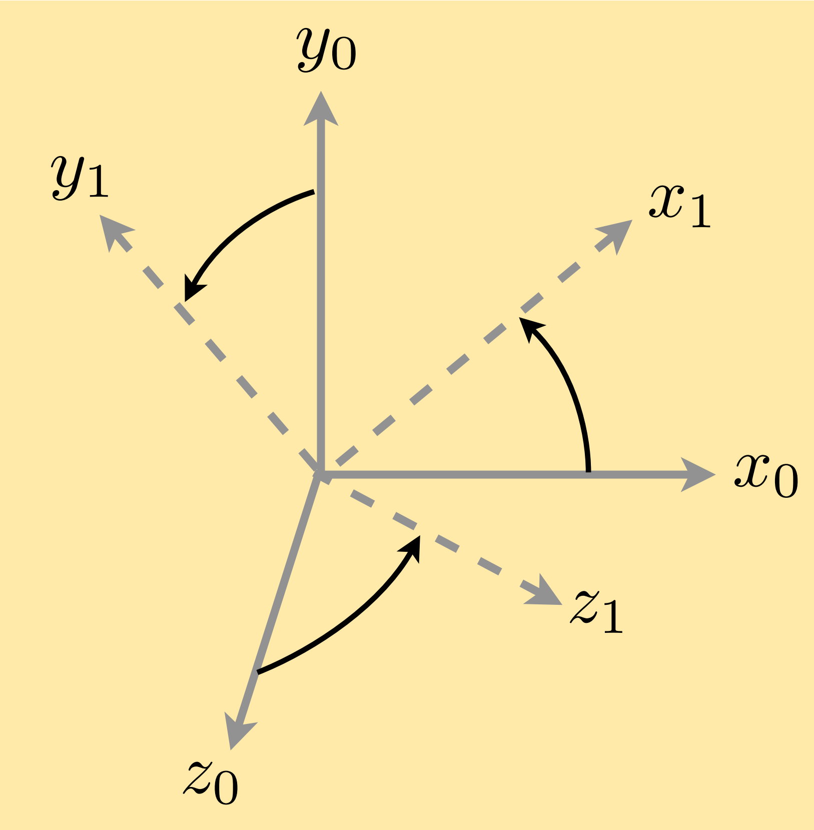
In MATLAB?

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

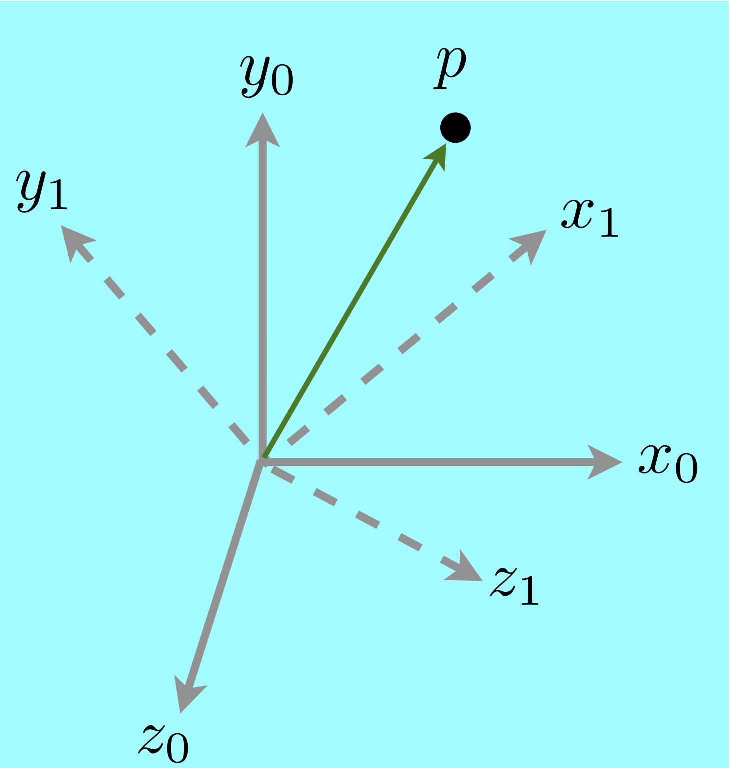
1. Coordinate transformation relating the coordinates of a point p in two different frames



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



Questions ❓