MEAM 520

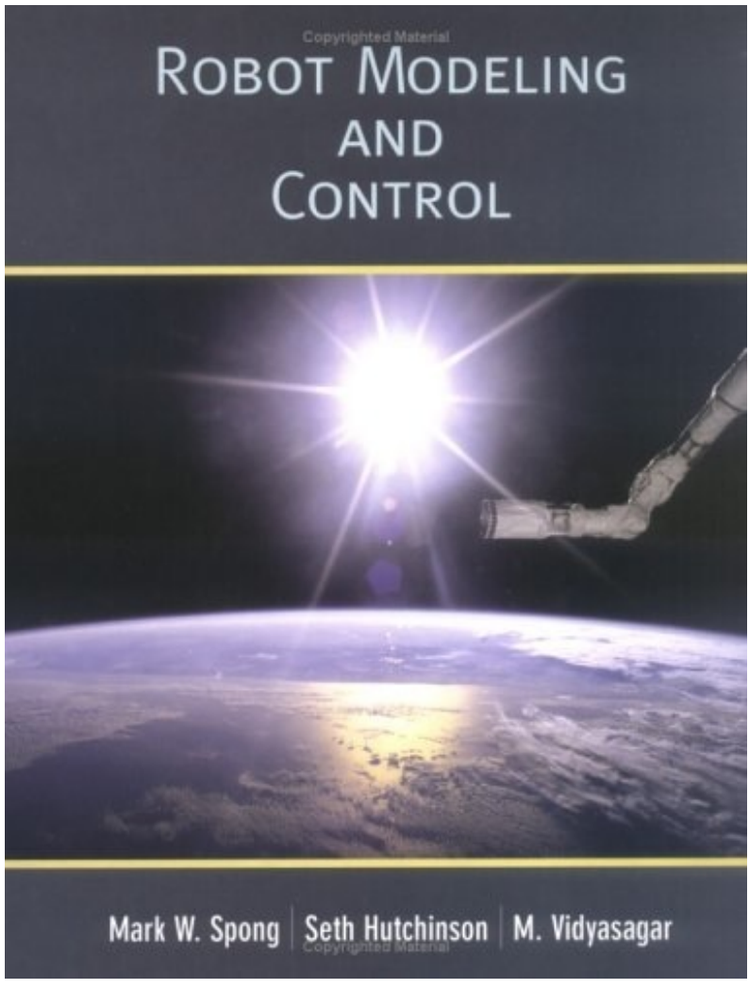
Homogeneous Transformations

Katherine J. Kuchenbecker, Ph.D.
Mechanical Engineering and Applied Mechanics Depa
SEAS, University of Pennsylvania

GRASP LABORATORY

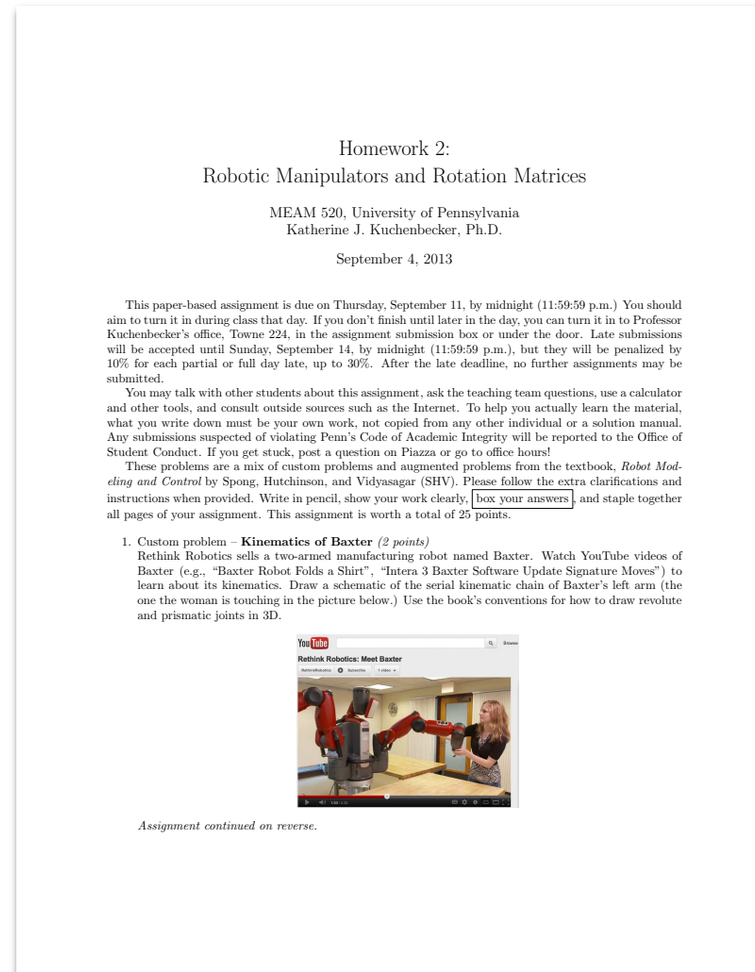
Lecture 5: September 11, 2014

Reading



For today,
read Sections 2.6,
2.7, and 2.8.

Homework



Homework 2:
Robotic Manipulators and Rotation Matrices

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

September 4, 2013

This paper-based assignment is due on Thursday, September 11, 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 assignment submission box or under the door. Late submissions will be accepted until Sunday, September 14, by midnight (11:59:59 p.m.), but they will be penalized by 10% for each partial or full day late, up to 30%. 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 must 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 a mix of custom problems and augmented problems 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. This assignment is worth a total of 25 points.

1. Custom problem – **Kinematics of Baxter** *(2 points)*
   Rethink Robotics sells a two-armed manufacturing robot named Baxter. Watch YouTube videos of Baxter (e.g., "Baxter Robot Folds a Shirt", "Intera 3 Baxter Software Update Signature Moves") to learn about its kinematics. Draw a schematic of the serial kinematic chain of Baxter's left arm (the one the woman is touching in the picture below.) Use the book's conventions for how to draw revolute and prismatic joints in 3D.

   Assignment continued on reverse.

Homework 2 is due by
11:59 p.m. today.

Parameterizations of Rotation Matrices

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

I figured out my confusion about gimbal lock!

Euler Angles

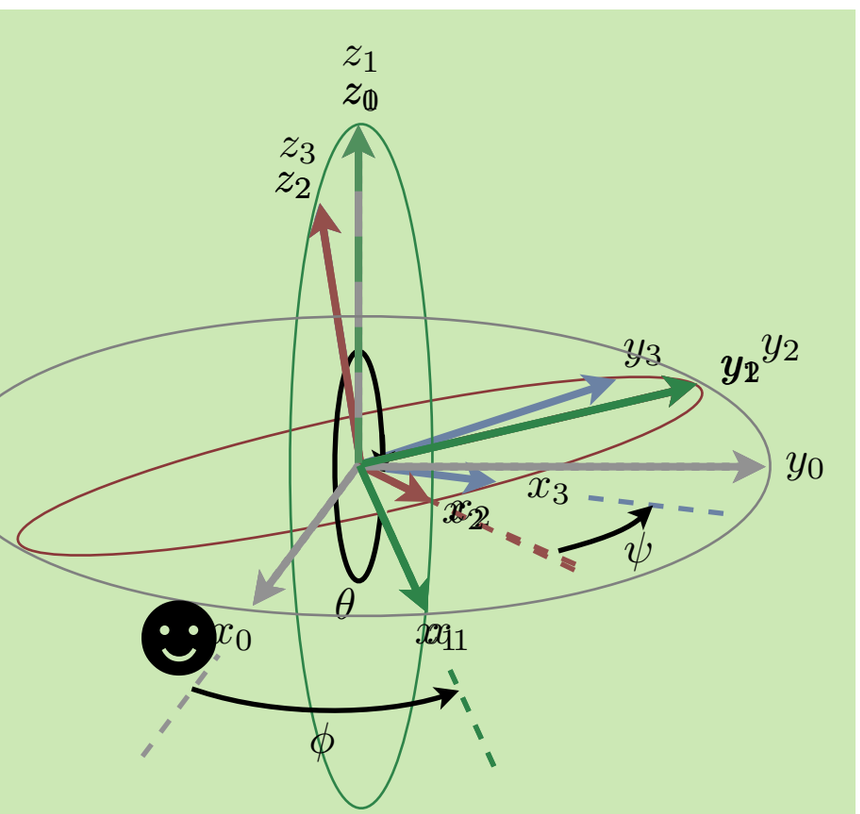
three rotations about intermediate frames

Roll, Pitch, Yaw Angles

three rotations about the fixed frame

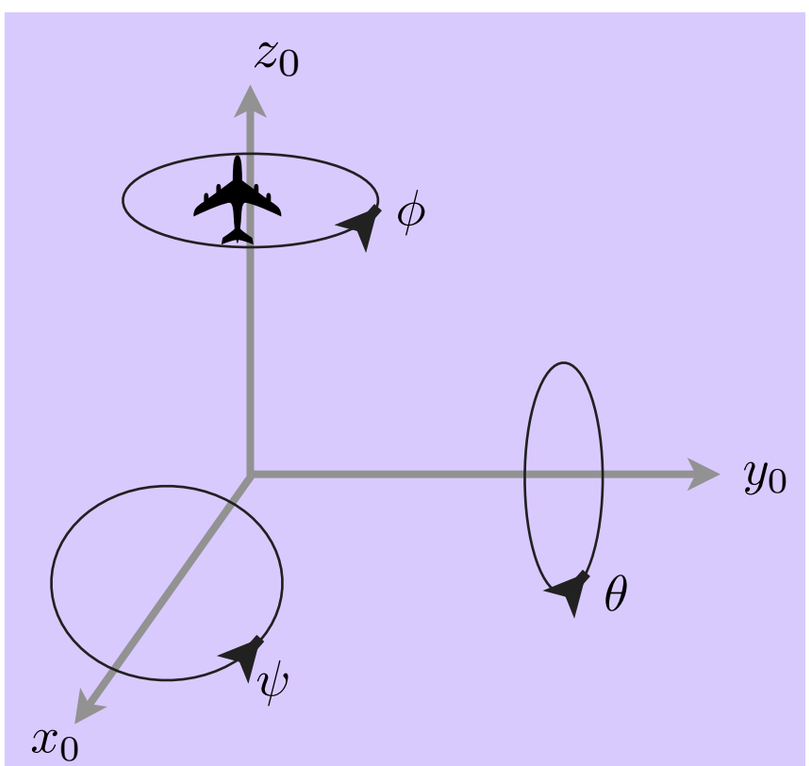
Axis/Angle

a unit vector (axis) and one angle



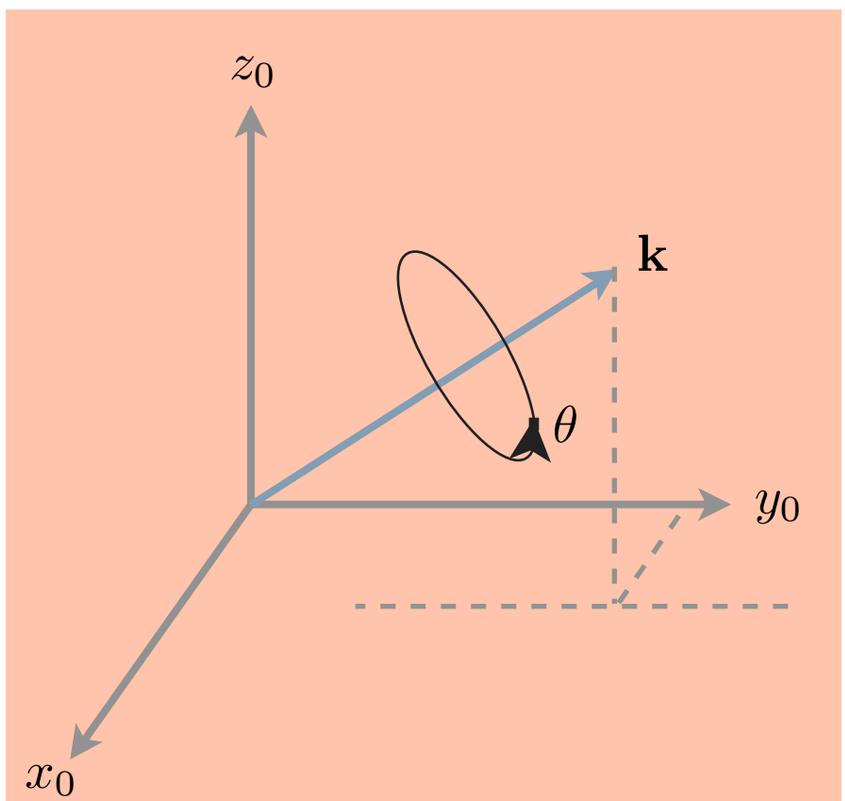
Our book uses ZYZ

$$\begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}$$



Our book uses XYZ

$$\begin{bmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}$$



$$\left[\begin{array}{c|c|c} k_x^2 v_\theta + c_\theta & k_x k_y v_\theta - k_z s_\theta & k_x k_z v_\theta + k_y s_\theta \\ k_x k_y v_\theta + k_z s_\theta & k_y^2 v_\theta + c_\theta & k_y k_z v_\theta - k_x s_\theta \\ k_x k_z v_\theta - k_y s_\theta & k_y k_z v_\theta + k_x s_\theta & k_z^2 v_\theta + c_\theta \end{array}\right]$$

Many of today's slides are adapted from ones created by Jonathan Fiene for MEAM 520 in Spring 2012.

Rigid Motions & Homogeneous Transformations

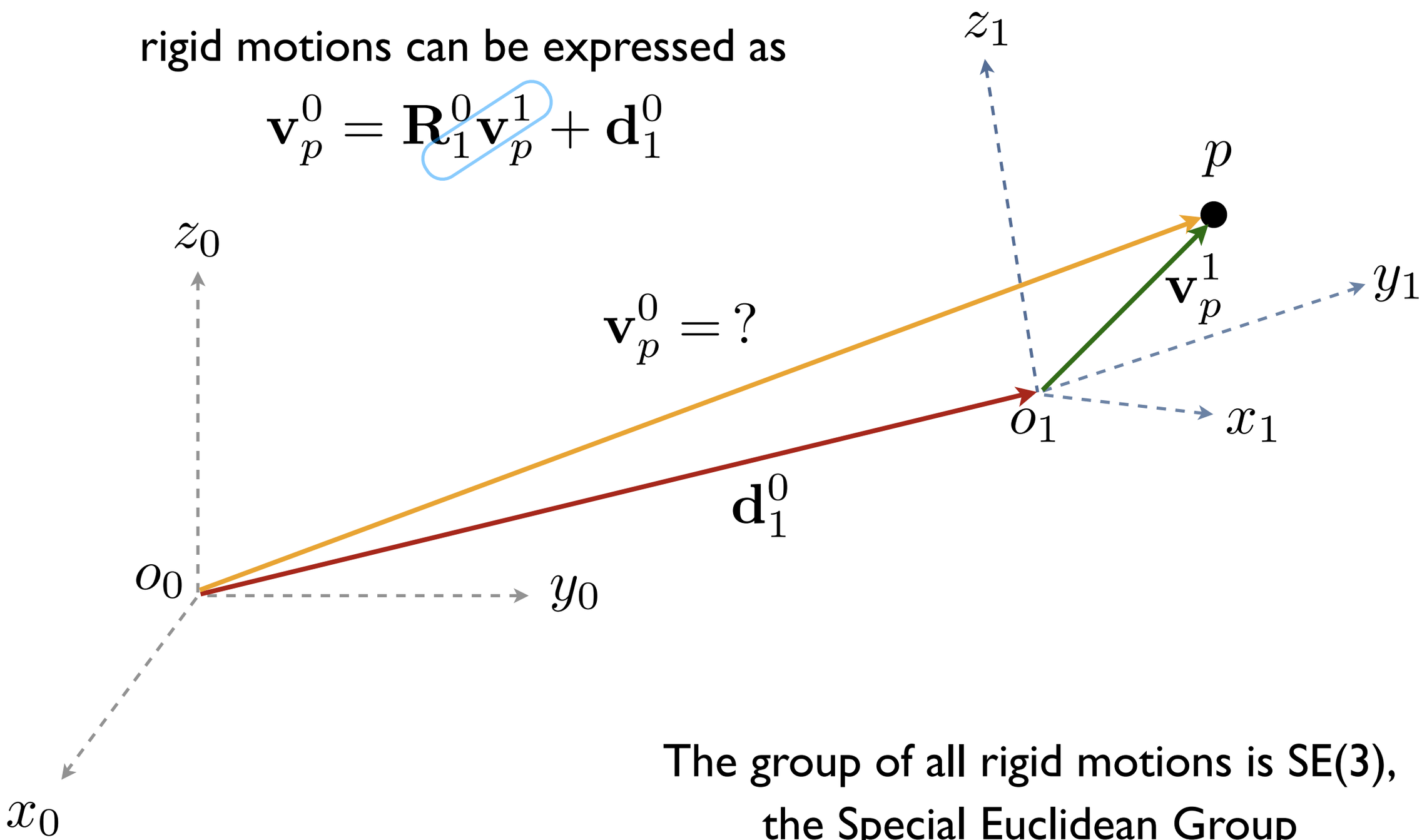
SHV 2.6 and 2.7

Rigid Motion

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

rigid motions can be expressed as

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



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

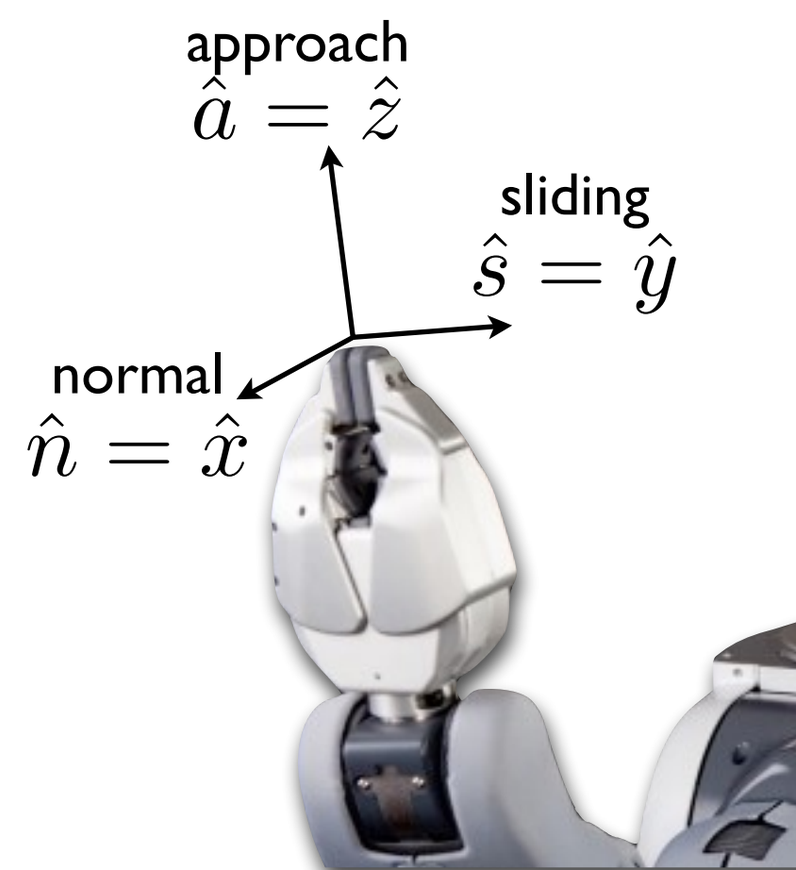
Homogeneous Transformations

a **homogeneous transformation** is a 4x4 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

$$\mathbf{H} = \begin{bmatrix} \overbrace{\begin{matrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{matrix}}^{\mathbf{R}_1^0} & \overbrace{\begin{matrix} d_x \\ d_y \\ d_z \end{matrix}}^{\mathbf{d}_1^0} \\ \underbrace{\begin{matrix} 0 & 0 & 0 \end{matrix}}_{\text{three zeros}} & \underbrace{1}_{\text{one one}} \end{bmatrix}$$



Homogeneous Transformations

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

$$\mathbf{P} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}$$

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

$$\mathbf{P} = \begin{bmatrix} \overset{\mathbf{v}_p^1}{\begin{matrix} p_x \\ p_y \\ p_z \end{matrix}} \\ 1 \end{bmatrix}$$

Homogeneous Transformations

rigid body transformations are accomplished by pre-multiplying by the homogenous transformation into the homogeneous representation of the point.

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

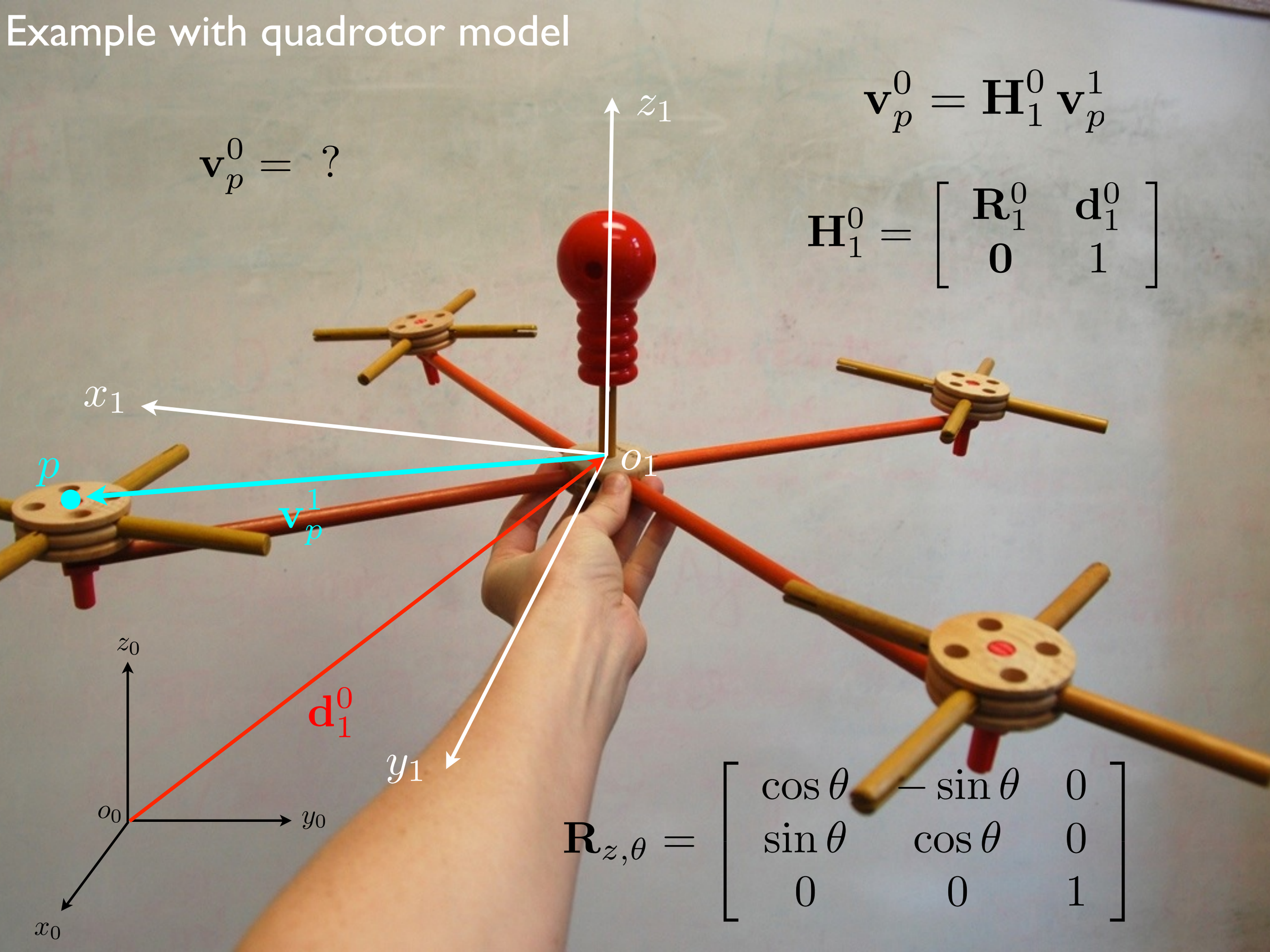
$$\mathbf{P}^0 = \begin{bmatrix} \overbrace{\begin{matrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{matrix}}^{\mathbf{R}_1^0} & \overbrace{\begin{matrix} d_x \\ d_y \\ d_z \end{matrix}}^{\mathbf{d}_1^0} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix} \begin{bmatrix} \overbrace{\begin{matrix} p_x \\ p_y \\ p_z \end{matrix}}^{\mathbf{v}_p^1} \\ 1 \end{bmatrix}$$

Example with quadrotor model

$$\mathbf{v}_p^0 = \ ?$$

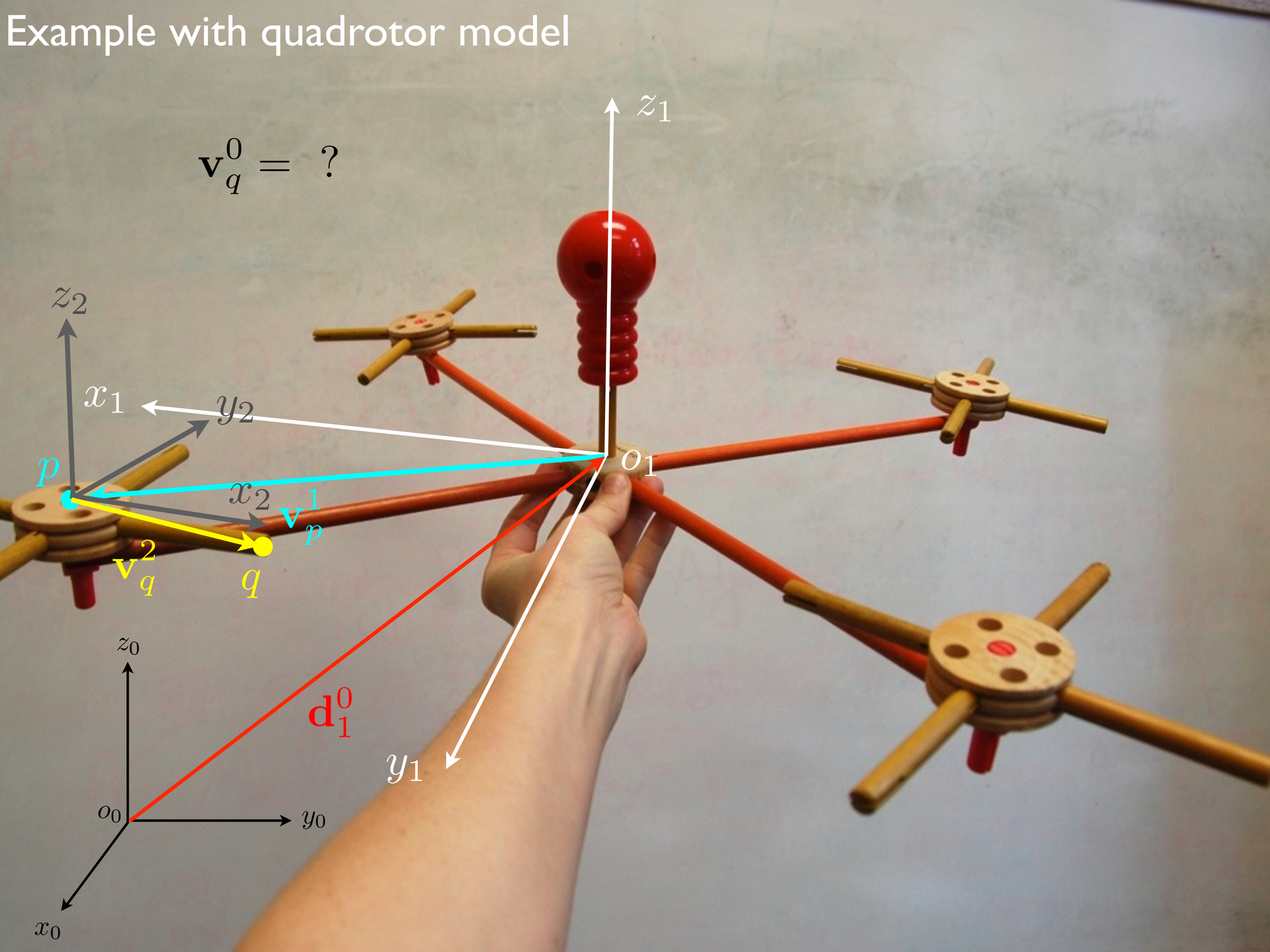
$$\mathbf{v}_p^0 = \mathbf{H}_1^0 \, \mathbf{v}_p^1$$

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



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

Example with quadrotor model



Homogeneous Transformations

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 via an 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

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

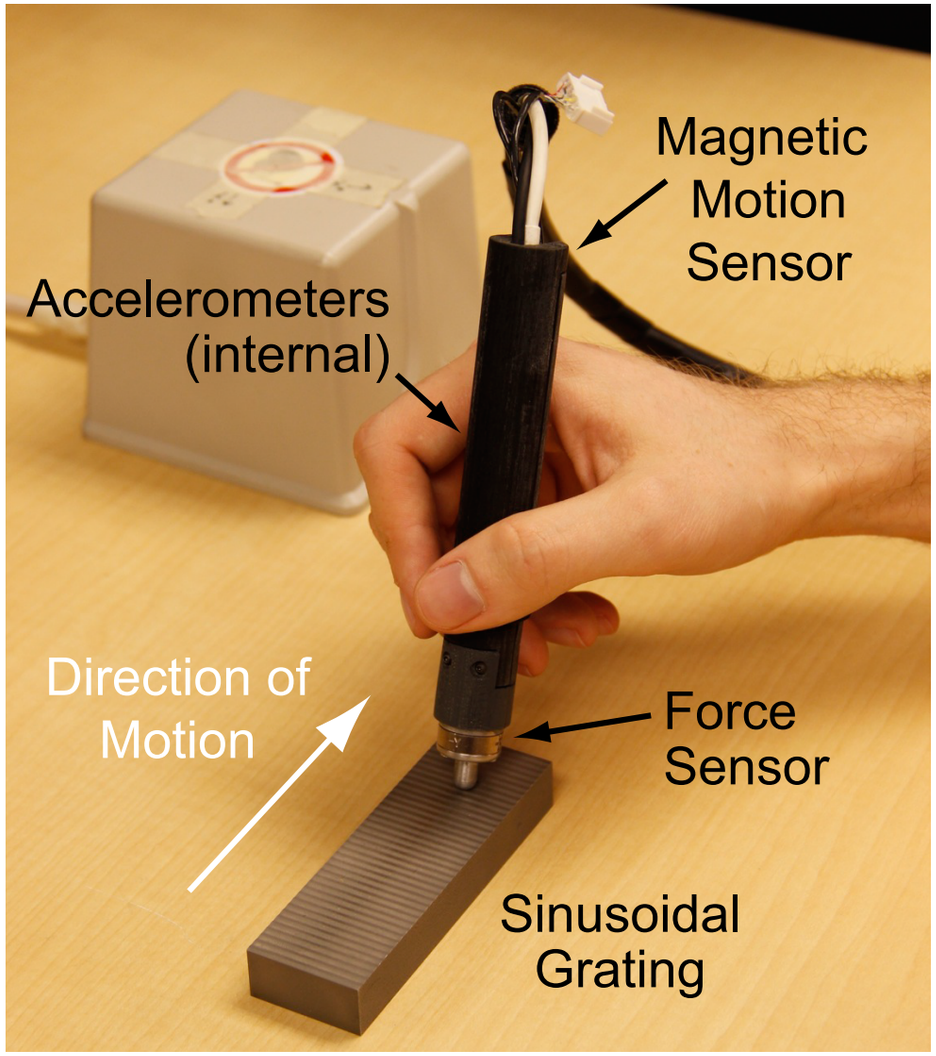
Basic Homogeneous Transformations

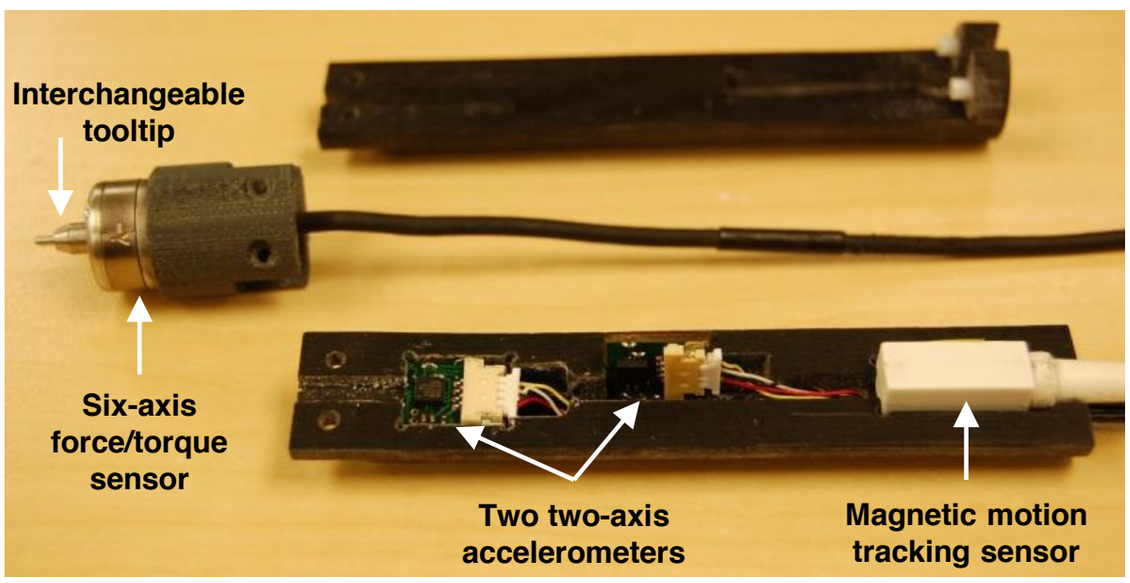
$$\mathrm{Trans}_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

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

Application of Rigid Motions and Homogeneous Transformations





I want to calculate the position of the tip of the handheld tool in the fixed frame based on magnetic motion tracking data.

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

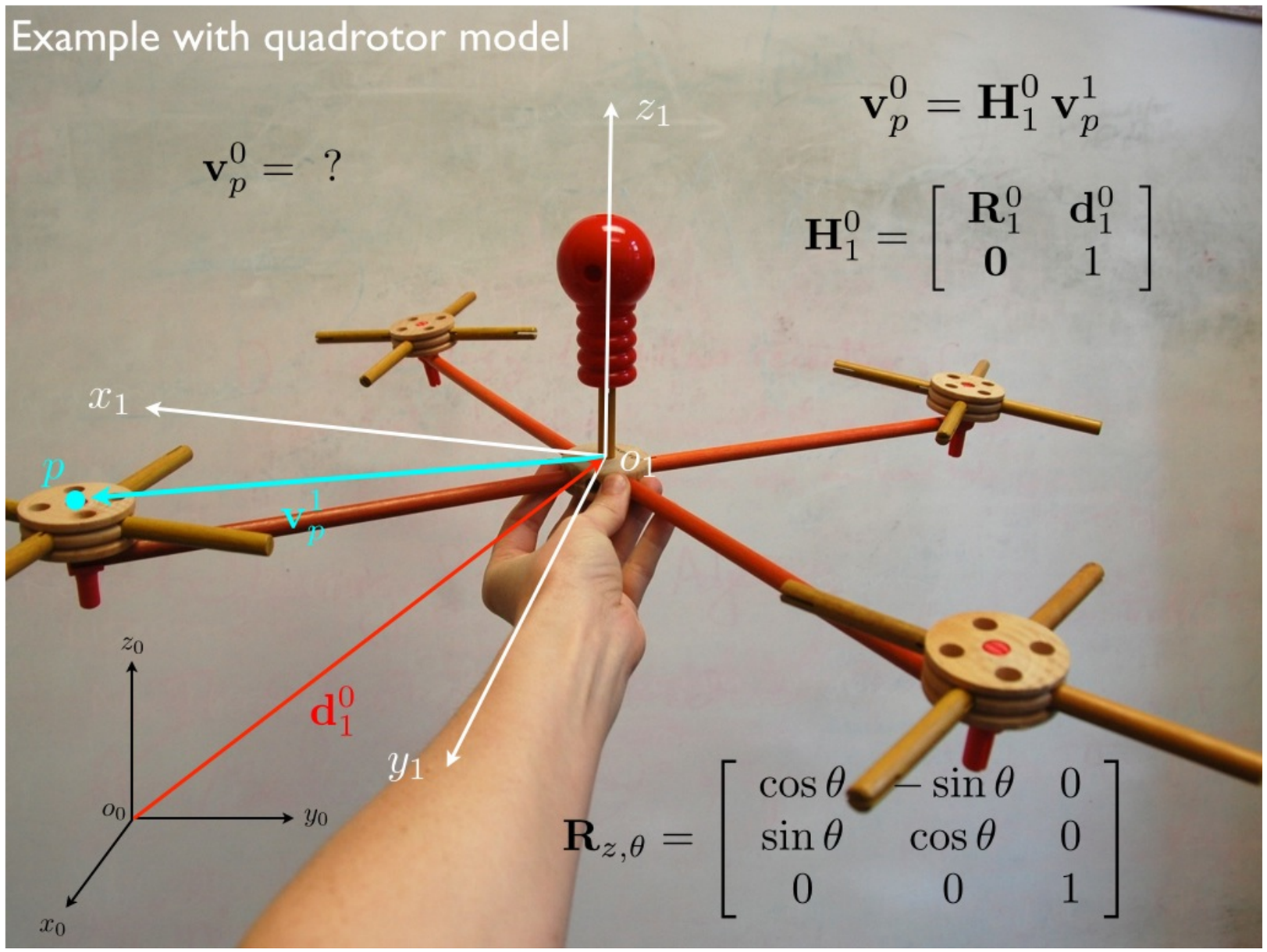
$$\mathbf{H} = \begin{bmatrix} \overbrace{\begin{matrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{matrix}}^{\mathbf{R}_1^0} & \overbrace{\begin{matrix} d_x \\ d_y \\ d_z \end{matrix}}^{\mathbf{d}_1^0} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix}$$

$$\mathbf{P} = \overbrace{\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}}^{\mathbf{v}_p^1}$$

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

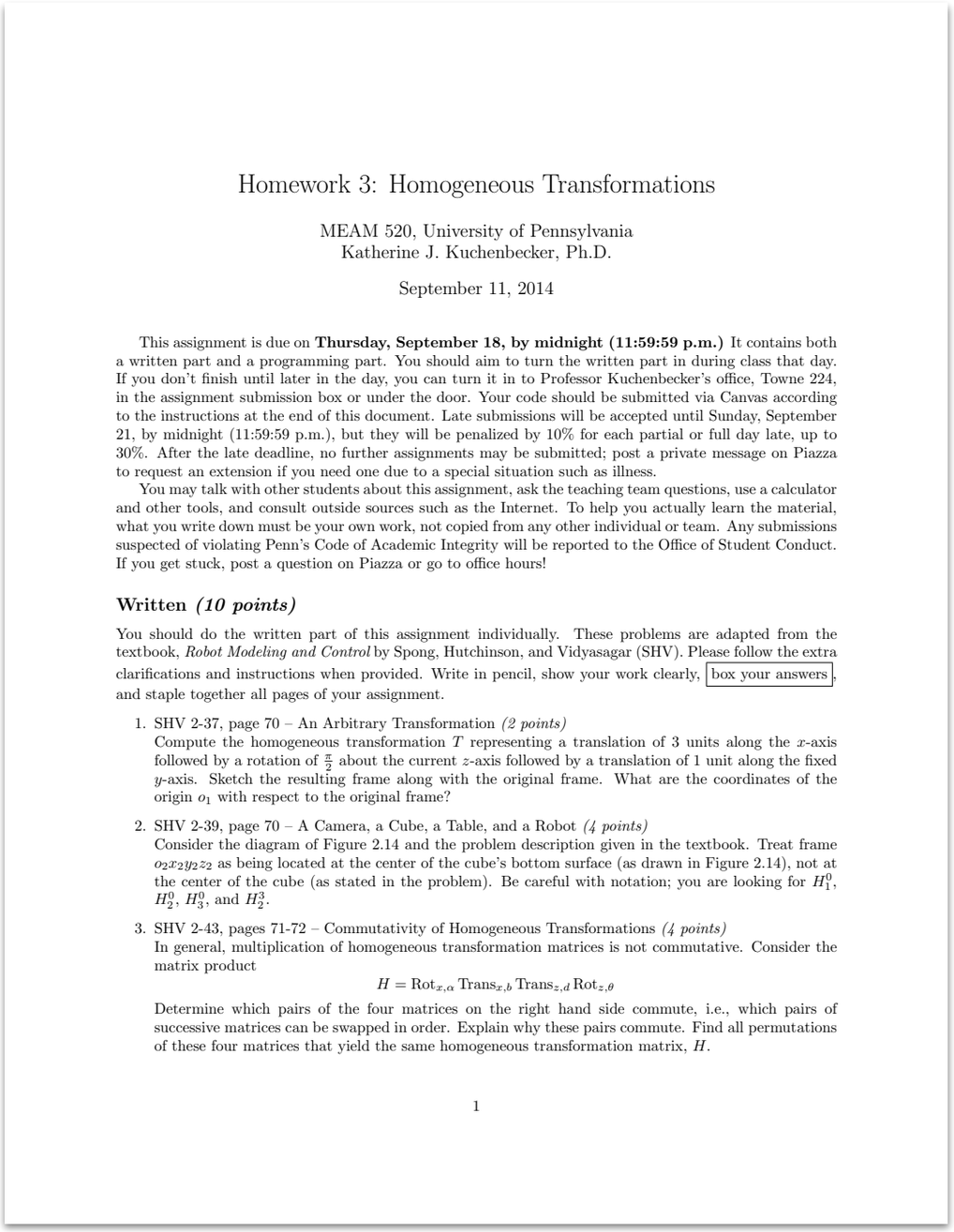
What questions do you have?

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



Homework 3

- Due by midnight (11:59 p.m.) on Thursday, September 18.
- Includes both written and programming portions.
- Written part includes three questions on homogeneous transformations. These should be done individually and are worth 10 points.
- Programming part can be done individually or in pairs, following the paradigm of pair programming. This part is worth 15 points.



Homework 3: Homogeneous Transformations

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

September 11, 2014

This assignment is due on **Thursday, September 18, by midnight (11:59:59 p.m.)** It contains both a written part and a programming part. You should aim to turn the written 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, in the assignment submission box or under the door. Your code should be submitted via Canvas according to the instructions at the end of this document. Late submissions will be accepted until Sunday, September 21, by midnight (11:59:59 p.m.), but they will be penalized by 10% for each partial or full day late, up to 30%. After the late deadline, no further assignments may be submitted; post a private message on Piazza to request an extension if you need one due to a special situation such as illness.

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 must 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!

Written ***(10 points)***

You should do the written part of this assignment individually. These problems are adapted 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 2-37, page 70 – An Arbitrary Transformation *(2 points)*
   Compute the homogeneous transformation T representing a translation of 3 units along the x-axis followed by a rotation of $\frac{\pi}{2}$ about the current z-axis followed by a translation of 1 unit along the fixed y-axis. Sketch the resulting frame along with the original frame. What are the coordinates of the origin o_1 with respect to the original frame?
2. SHV 2-39, page 70 – A Camera, a Cube, a Table, and a Robot *(4 points)*
   Consider the diagram of Figure 2.14 and the problem description given in the textbook. 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_2^3.
3. SHV 2-43, pages 71-72 – Commutativity of Homogeneous Transformations *(4 points)*
   In general, multiplication of homogeneous transformation matrices is not commutative. Consider the matrix product
   $$H = \mathrm{Rot}_{x,\alpha}\,\mathrm{Trans}_{x,b}\,\mathrm{Trans}_{z,d}\,\mathrm{Rot}_{z,\theta}$$
   Determine which pairs of the four matrices on the right hand side commute, i.e., which pairs of successive matrices can be swapped in order. Explain why these pairs commute. Find all permutations of these four matrices that yield the same homogeneous transformation matrix, H.

1

3D Guidance trakSTAR™

Class 1, Type B Applied Part

Desktop electronics unit tracks multiple sensors simultaneously

Track Objects with Magnetic DC Technology

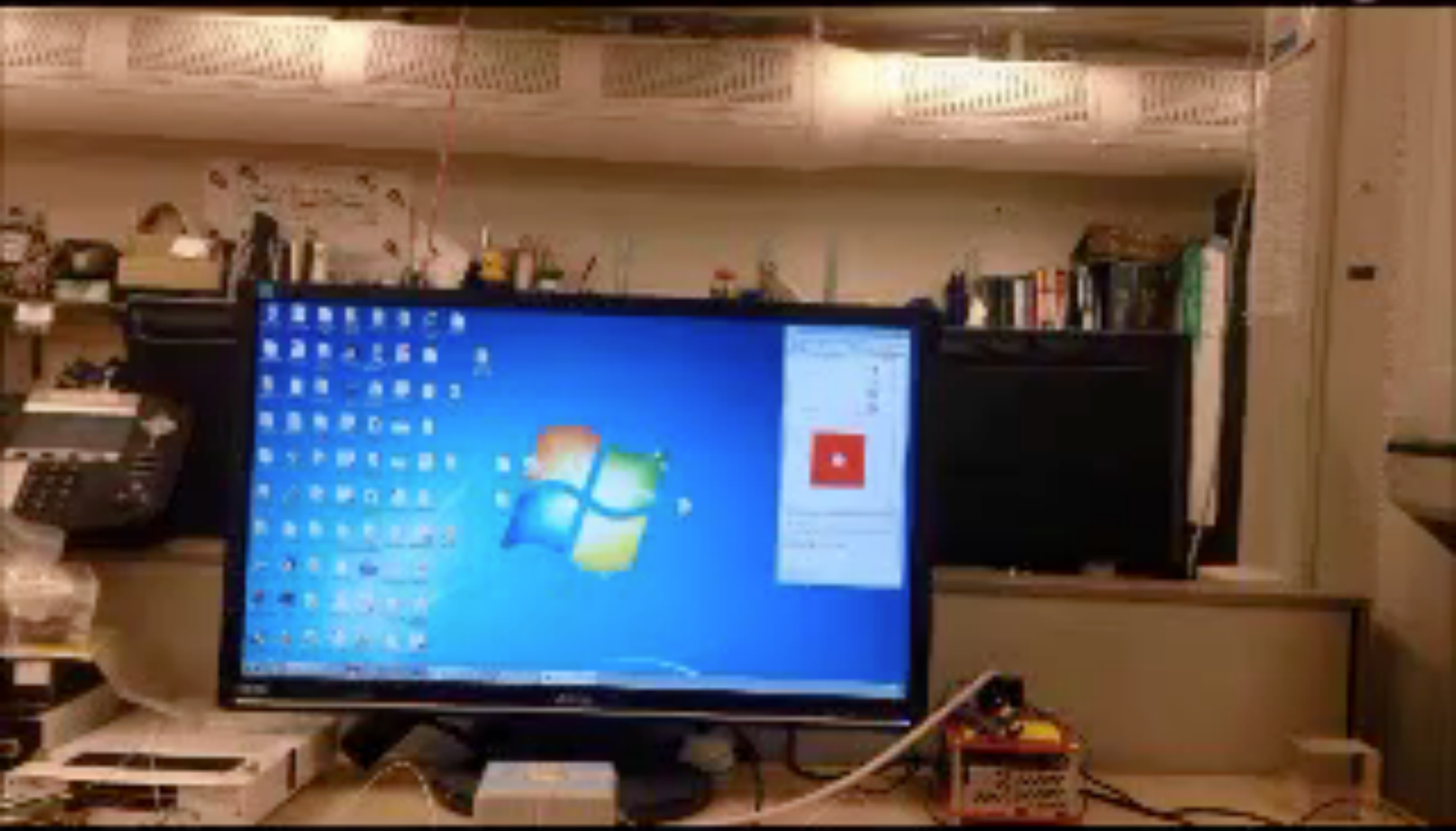
- **Fast, dynamic tracking** – 240 to 420 updates per second.
- **Miniaturized passive sensors** – outputs unaffected by "power-line" noise sources.
- **All attitude tracking** – no inertial drift or optical interference.
- **High metal immunity** – no distortion from non magnetic metals.

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



http://youtu.be/EEWjDd8g8p0

flying_box.txt

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

Sensor2:	0x0000	13.847	-9.224	-2.597	90.253	-5.885	96.257	0	1	1410210928.776
Sensor2:	0x0000	13.843	-9.229	-2.597	90.260	-5.878	96.228	0	1	1410210928.780
Sensor2:	0x0000	13.843	-9.229	-2.597	90.260	-5.864	96.207	0	1	1410210928.784
Sensor2:	0x0000	13.838	-9.233	-2.597	90.260	-5.850	96.193	0	1	1410210928.788
Sensor2:	0x0000	13.838	-9.237	-2.597	90.253	-5.836	96.179	0	1	1410210928.792
Sensor2:	0x0000	13.834	-9.242	-2.597	90.246	-5.815	96.164	0	1	1410210928.796
Sensor2:	0x0000	13.834	-9.242	-2.597	90.239	-5.808	96.150	0	1	1410210928.800
Sensor2:	0x0000	13.830	-9.246	-2.597	90.239	-5.794	96.136	0	1	1410210928.805
Sensor2:	0x0000	13.830	-9.246	-2.602	90.232	-5.787	96.128	0	1	1410210928.809
Sensor2:	0x0000	13.825	-9.250	-2.602	90.232	-5.780	96.128	0	1	1410210928.813
Sensor2:	0x0000	13.825	-9.250	-2.602	90.232	-5.773	96.128	0	1	1410210928.817
Sensor2:	0x0000	13.821	-9.255	-2.610	90.239	-5.766	96.128	0	1	1410210928.830
Sensor2:	0x0000	13.816	-9.255	-2.624	90.260	-5.766	96.128	0	1	1410210928.846
Sensor2:	0x0000	13.812	-9.255	-2.637	90.295	-5.752	96.128	0	1	1410210928.863
Sensor2:	0x0000	13.808	-9.255	-2.650	90.330	-5.745	96.114	0	1	1410210928.880
Sensor2:	0x0000	13.799	-9.255	-2.663	90.380	-5.724	96.086	0	1	1410210928.896
Sensor2:	0x0000	13.790	-9.255	-2.676	90.422	-5.724	96.064	0	1	1410210928.913
Sensor2:	0x0000	13.781	-9.255	-2.694	90.471	-5.717	96.029	0	1	1410210928.930
Sensor2:	0x0000	13.772	-9.259	-2.716	90.499	-5.710	95.993	0	1	1410210928.946
Sensor2:	0x0000	13.764	-9.264	-2.738	90.513	-5.675	95.951	0	1	1410210928.963
Sensor2:	0x0000	13.750	-9.268	-2.764	90.506	-5.639	95.929	0	1	1410210928.980
Sensor2:	0x0000	13.750	-9.272	-2.786	90.548	-5.576	95.879	0	1	1410210928.996
Sensor2:	0x0000	13.755	-9.277	-2.804	90.618	-5.527	95.836	0	1	1410210929.013
Sensor2:	0x0000	13.759	-9.272	-2.813	90.717	-5.450	95.757	0	1	1410210929.030
Sensor2:	0x0000	13.764	-9.272	-2.826	90.794	-5.372	95.686	0	1	1410210929.046
Sensor2:	0x0000	13.764	-9.272	-2.839	90.843	-5.274	95.601	0	1	1410210929.063
Sensor2:	0x0000	13.755	-9.281	-2.856	90.871	-5.190	95.529	0	1	1410210929.080
Sensor2:	0x0000	13.750	-9.281	-2.883	90.948	-5.127	95.451	0	1	1410210929.096
Sensor2:	0x0000	13.746	-9.281	-2.909	91.032	-5.077	95.380	0	1	1410210929.113
Sensor2:	0x0000	13.742	-9.272	-2.944	91.145	-5.056	95.316	0	1	1410210929.129
Sensor2:	0x0000	13.737	-9.268	-2.971	91.250	-5.028	95.217	0	1	1410210929.146
Sensor2:	0x0000	13.733	-9.268	-3.001	91.320	-5.007	95.147	0	1	1410210929.163
Sensor2:	0x0000	13.724	-9.268	-3.032	91.362	-4.979	95.048	0	1	1410210929.180
Sensor2:	0x0000	13.707	-9.277	-3.072	91.334	-4.958	95.020	0	1	1410210929.196
Sensor2:	0x0000	13.693	-9.281	-3.111	91.327	-4.951	94.977	0	1	1410210929.213
Sensor2:	0x0000	13.680	-9.277	-3.155	91.327	-4.965	94.949	0	1	1410210929.230
Sensor2:	0x0000	13.667	-9.268	-3.190	91.348	-5.000	94.872	0	1	1410210929.246
Sensor2:	0x0000	13.654	-9.259	-3.230	91.334	-5.063	94.760	0	1	1410210929.263
Sensor2:	0x0000	13.641	-9.246	-3.265	91.313	-5.169	94.613	0	1	1410210929.280
Sensor2:	0x0000	13.623	-9.237	-3.305	91.244	-5.309	94.480	0	1	1410210929.296
Sensor2:	0x0000	13.601	-9.224	-3.344	91.187	-5.436	94.353	0	1	1410210929.313
Sensor2:	0x0000	13.583	-9.211	-3.379	91.103	-5.520	94.234	0	1	1410210929.330
Sensor2:	0x0000	13.566	-9.207	-3.410	90.984	-5.569	94.087	0	1	1410210929.346
Sensor2:	0x0000	13.540	-9.207	-3.450	90.787	-5.597	93.946	0	1	1410210929.363
Sensor2:	0x0000	13.504	-9.211	-3.498	90.534	-5.675	93.785	0	1	1410210929.379
Sensor2:	0x0000	13.460	-9.220	-3.555	90.232	-5.766	93.623	0	1	1410210929.396

MATLAB R2013b

HOME | PLOTS | APPS | Search Documentation

/ ▸ Users ▸ kuchenbe ▸ Documents ▸ teaching ▸ meam 520 ▸ assignments ▸ 03 box ▸ hw03-code

Current Folder

Name
flying_box.mat
flying_box.wmv
flying_box_starter.m

Details

Command Window

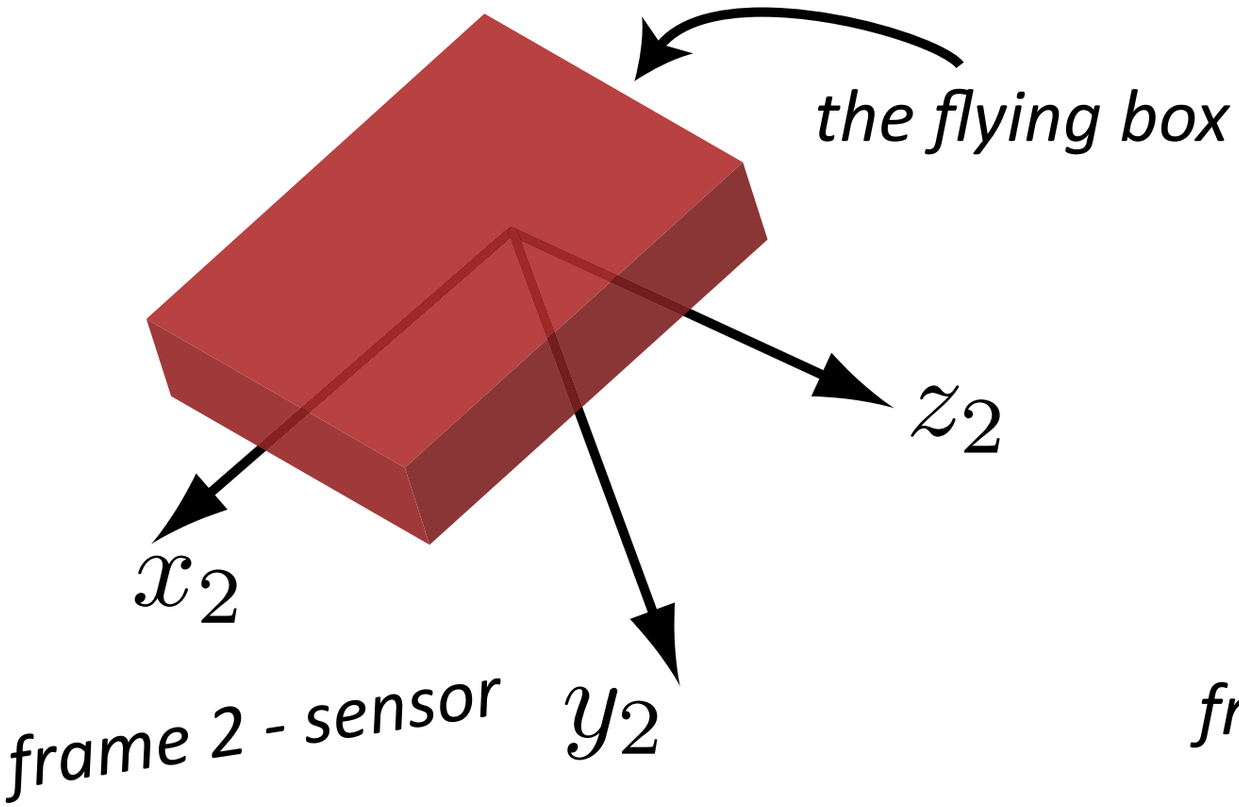
```
>> load flying_box
>> x_inches_history(1:10)

ans =

   13.8470
   13.8430
   13.8430
   13.8380
   13.8380
   13.8340
   13.8340
   13.8300
   13.8300
   13.8250

fx >>
```

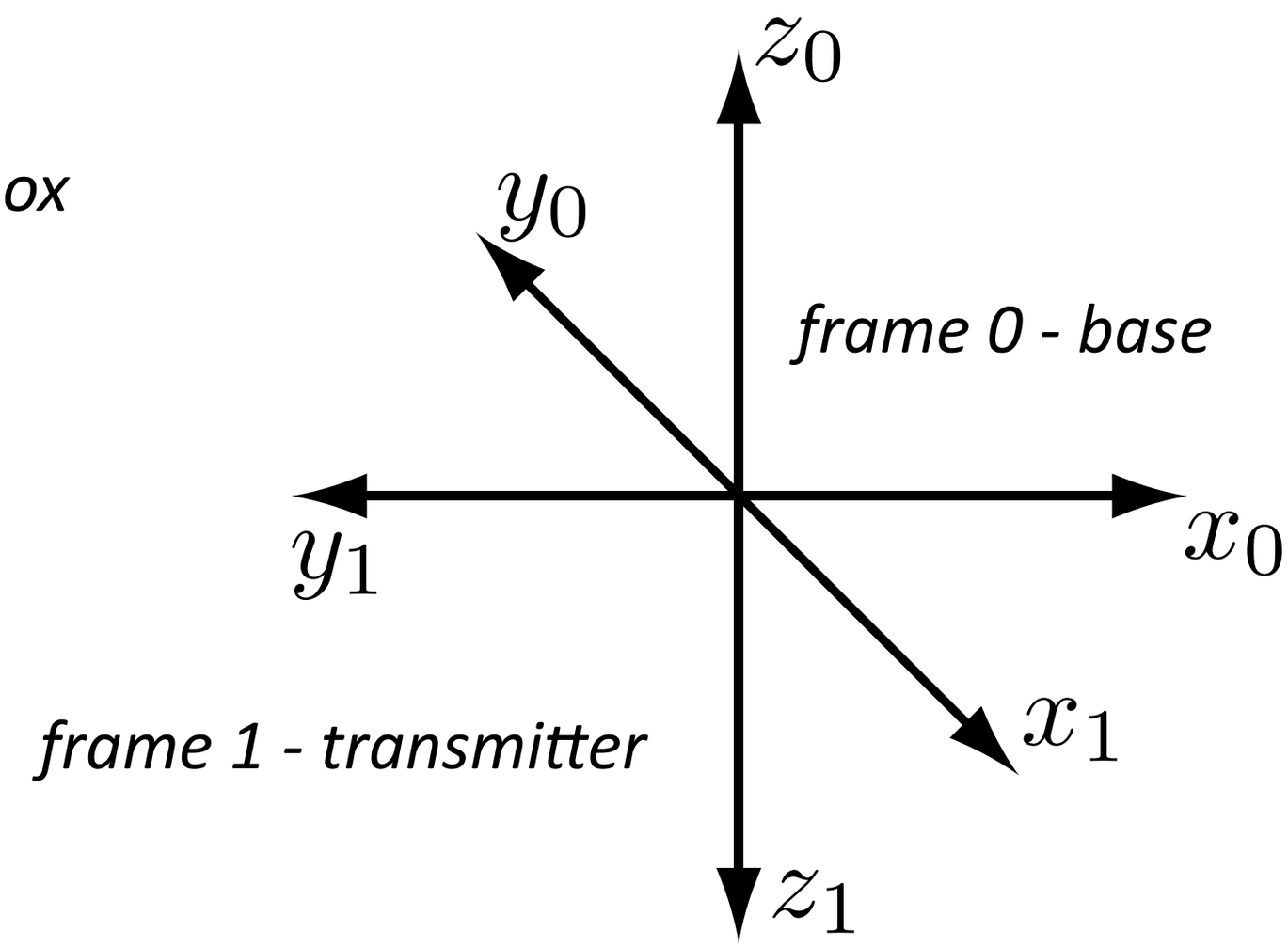
Workspace

Name	Value	Min	Max
alldata	1404x6 double	-17...	179.
ans	[13.8470;13.843...	13.8...	13.8
azimuth_degrees_history	1404x1 double	17.7...	142.
elevation_degrees_history	1404x1 double	-47....	50.4
roll_degrees_history	1404x1 double	-17...	179.
x_inches_history	1404x1 double	3.42...	14.7
y_inches_history	1404x1 double	-9.6...	4.79
z_inches_history	1404x1 double	-17....	-2.5

Command History

```
clear
cd ..
cd hw03-code/
ls
clear
clc
load flying_box
x_inches_history(1:10)
```


the flying box
z_2
x_2
y_2
frame 2 - sensor


z_0
y_0
frame 0 - base
y_1
x_0
x_1
frame 1 - transmitter
z_1

Editor - /Users/kuchenbe/Documents/teaching/meam 520/assignments/03 box/hw03-code/flying_box_starter.m

EDITOR PUBLISH VIEW

flying_box_starter.m

```
%% flying_box_starter.m
%
% This Matlab script provides the solution for the flying box problem on
% Homework 3 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 NAMES 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

% Animation Control
```

script Ln 4 Col 5

Editor - /Users/kuchenbe/Documents/teaching/meam 520/assignments/03 box/hw03-code/flying_box_starter.m

EDITOR PUBLISH VIEW

flying_box_starter.m

```
% Animation Control

% The number of data points to skip between plots.  Increase this to speed
% up playback.  The minimum value is 1.
skipfactor = 1;

% Delay time after showing each frame.  Increase this to slow down
% playback, and decrease it to speed up playback.
plot_delay = 0.001; % s

% Frame Definitions
%
% Frame 0 is the base frame, which aligns with the camera's view; x is
% positive to the right, y is positive straight back, and z is 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 through the front of the box, in the direction
% the hand is facing.  Its z-axis points out the box's left side, and its
% z-axis points out the bottom of the box.  See the assignment for a diagram.

% Box Definitions

% Define the size of the box
```

script Ln 45 Col 6

Editor - /Users/kuchenbe/Documents/teaching/meam 520/assignments/03 box/hw03-code/flying_box_starter.m

EDITOR PUBLISH VIEW

flying_box_starter.m

```
%% 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.
    % 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
```

script Ln 96 Col 7

Editor - /Users/kuchenbe/Documents/teaching/meam 520/assignments/03 box/hw03-code/flying_box_starter.m

EDITOR PUBLISH VIEW

flying_box_starter.m

```
    % ***********************************************************************

    % 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 base 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 + [i/100 0 0 0]'; % Make this corner grow over time so you can see t
    pb0 = pb2;
    pc0 = pc2;
    pd0 = pd2;
    pe0 = pe2;
    pf0 = pf2;
    pg0 = pg2;
    ph0 = ph2;

    % If your code runs but you can't see anything in your plot, modify the
    % axis([...]) command in the plot section below to show a larger volume.

    % 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, pd0, pe0, pf0, pg0, and ph0.
    % ***********************************************************************
```

script Ln 112 Col 34

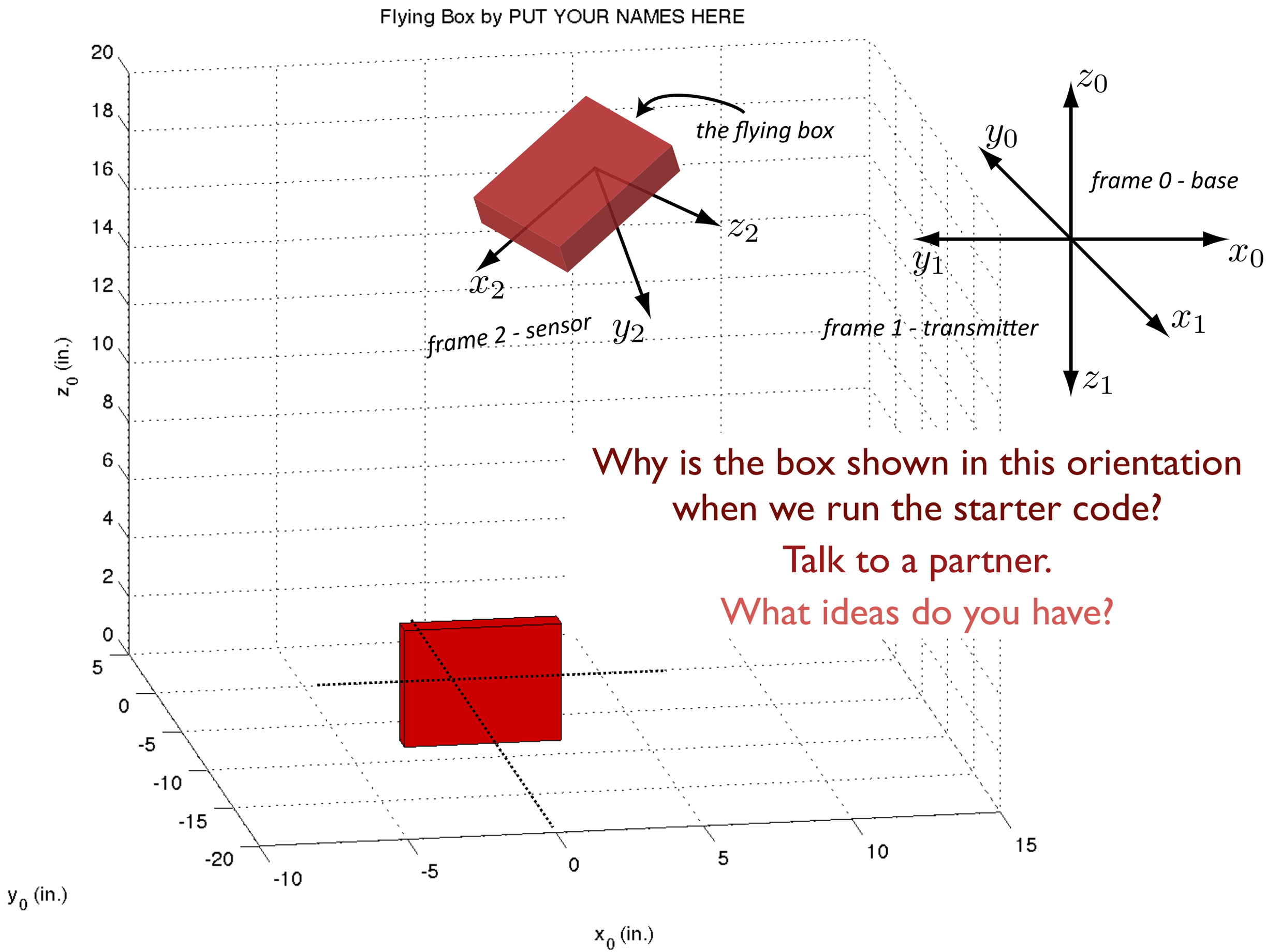
Editor - /Users/kuchenbe/Documents/teaching/meam 520/assignments/03 box/hw03-code/flying_box_starter.m

EDITOR PUBLISH VIEW

flying_box_starter.m

```
    % Define faces of the box using the corner points.
    top_face =    [pb0 ph0 pf0 pd0];
    front_face =  [pg0 ph0 pb0 pa0];
    left_face =   [pa0 pb0 pd0 pc0];
    back_face =   [pc0 pd0 pf0 pe0];
    right_face =  [pe0 pf0 ph0 pg0];
    bottom_face = [pa0 pc0 pe0 pg0];

    % Check for the first time through to format the plot.
    if (i == 1)

        % Plot the points as a red filled region.
        h = fill3([top_face(1,:)' front_face(1,:)' left_face(1,:)' back_face(1,:)'
                   [top_face(2,:)' front_face(2,:)' left_face(2,:)' back_face(2,:)'
                   [top_face(3,:)' front_face(3,:)' left_face(3,:)' back_face(3,:)'
                   [.8 0 0]);

        % Turn on hold so we can plot something else.
        hold on

        % Plot a fake shadow of the box to help the viewer see where it is.
        % We use the same x and y coordinates as the box itself but set the
        % z coordinates all to zero so that the shadow is flat.
        hshadow = fill3([top_face(1,:)' front_face(1,:)' left_face(1,:)' back_face
                         [top_face(2,:)' front_face(2,:)' left_face(2,:)' back_face
                       0*[top_face(3,:)' front_face(3,:)' left_face(3,:)' back_face
                         [.3 .3 .3],'edgealpha',0);

        % Enforce that one unit is displayed equivalently in x, y, and z
        % and that the graph doesn't grow or shrink when rotated
```

script Ln 152 Col 2



Why is the box shown in this orientation when we run the starter code?

Talk to a partner.

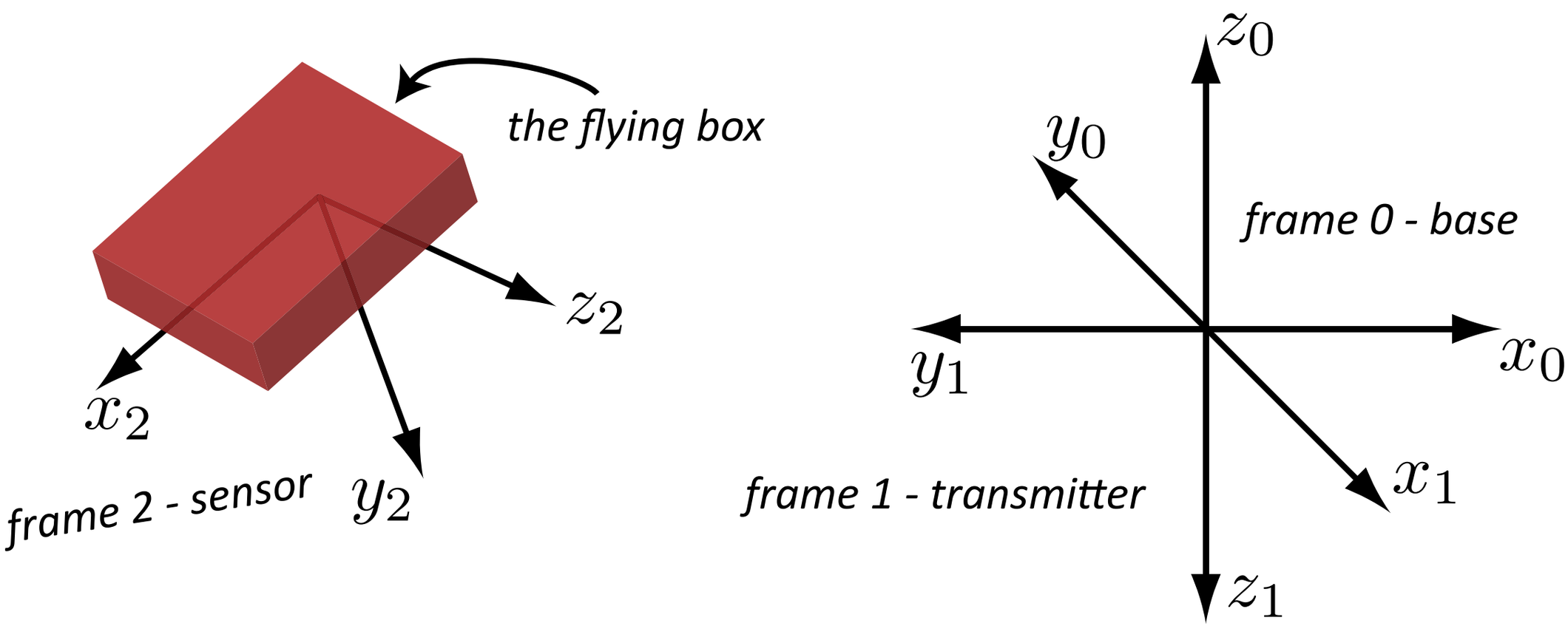
What ideas do you have?

Programming Homework Tips:

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

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

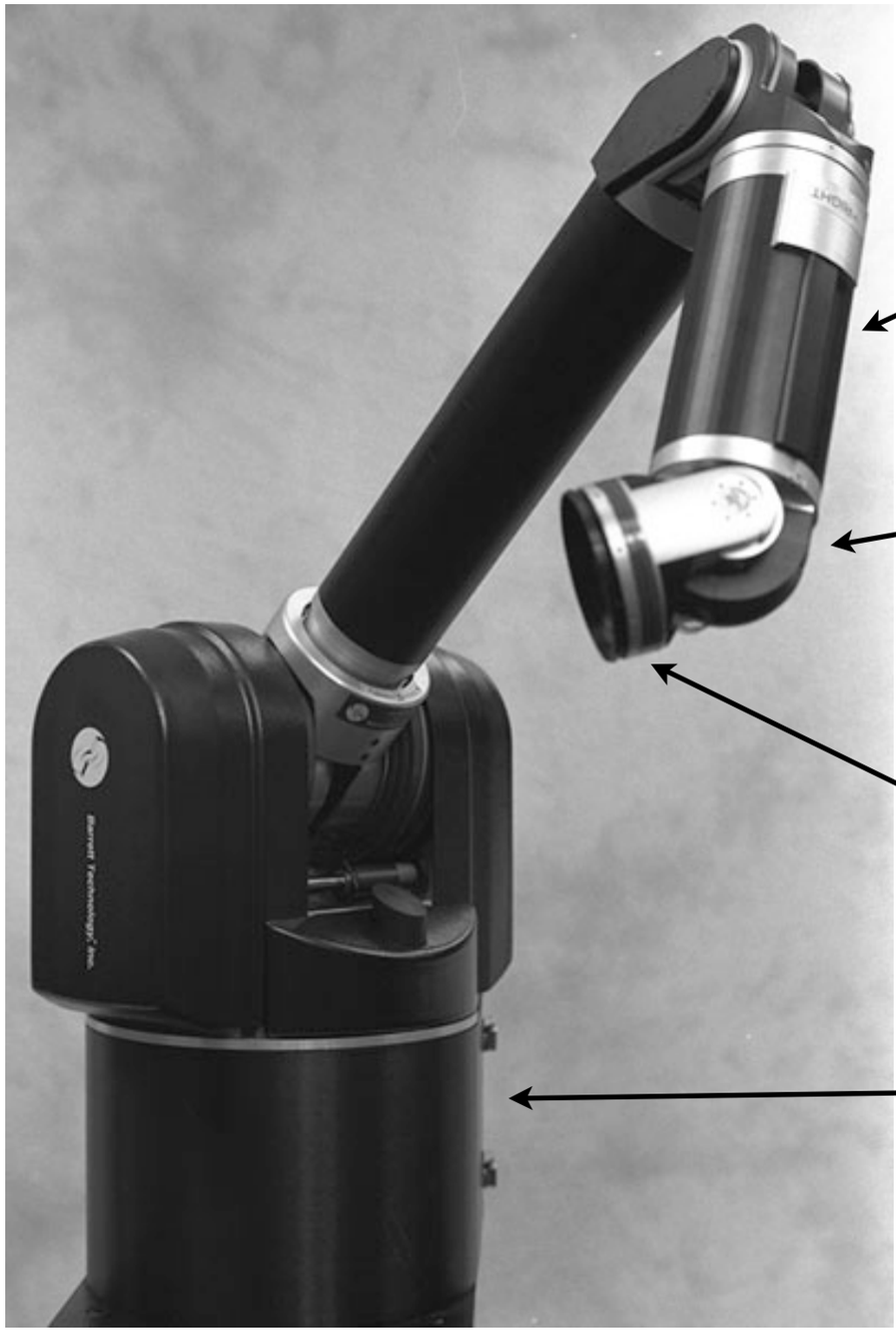
What questions do you have?



Manipulator Terminology Review

SHV 1.1-1.3, 3.1

General Terminology



Link : rigid body, 6 degrees of freedom

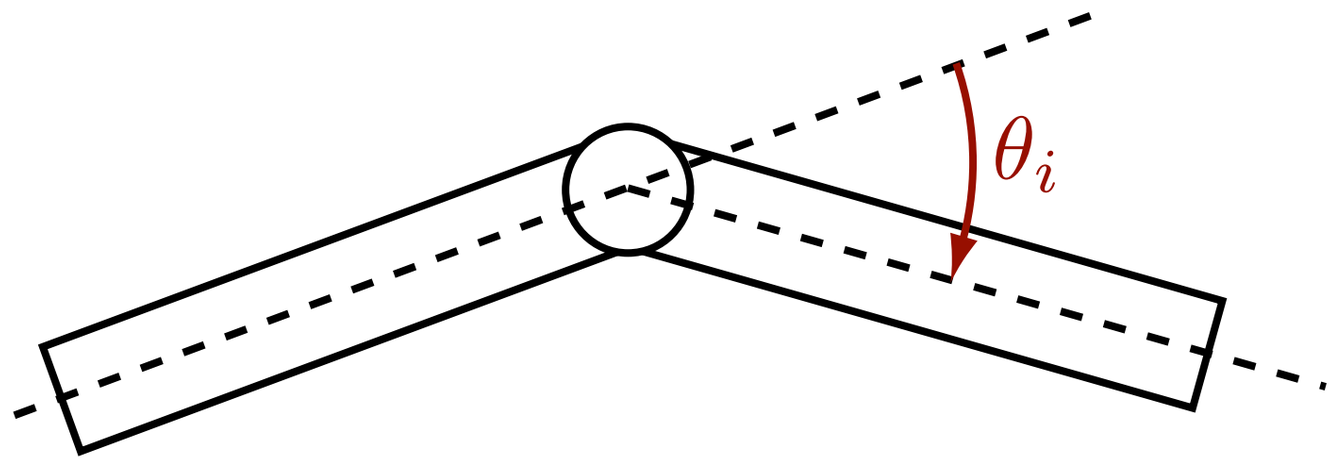
Joint : connection between two links

End-effector : interacts with the environment

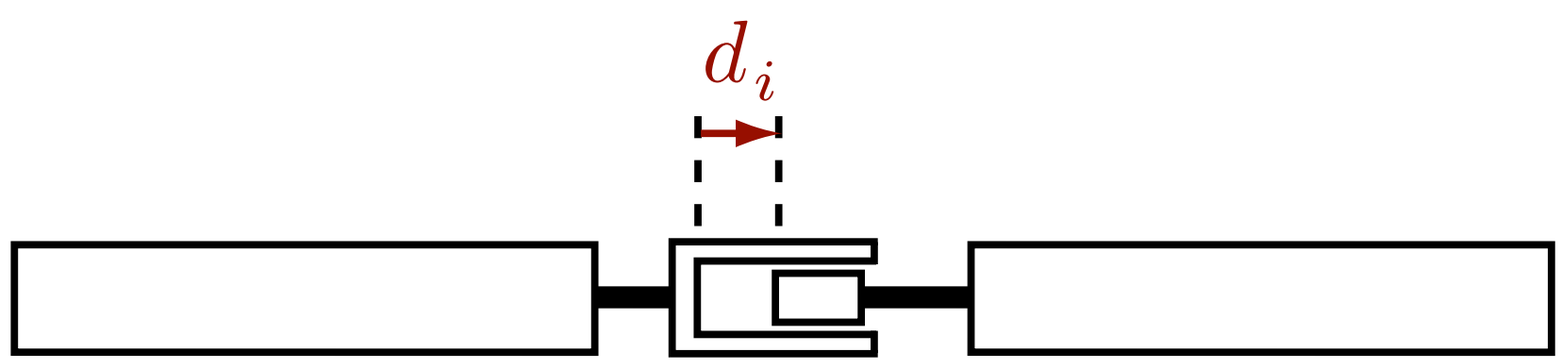
Base : connected to ground

Joint Descriptions

Revolute : angular displacement between adjacent links



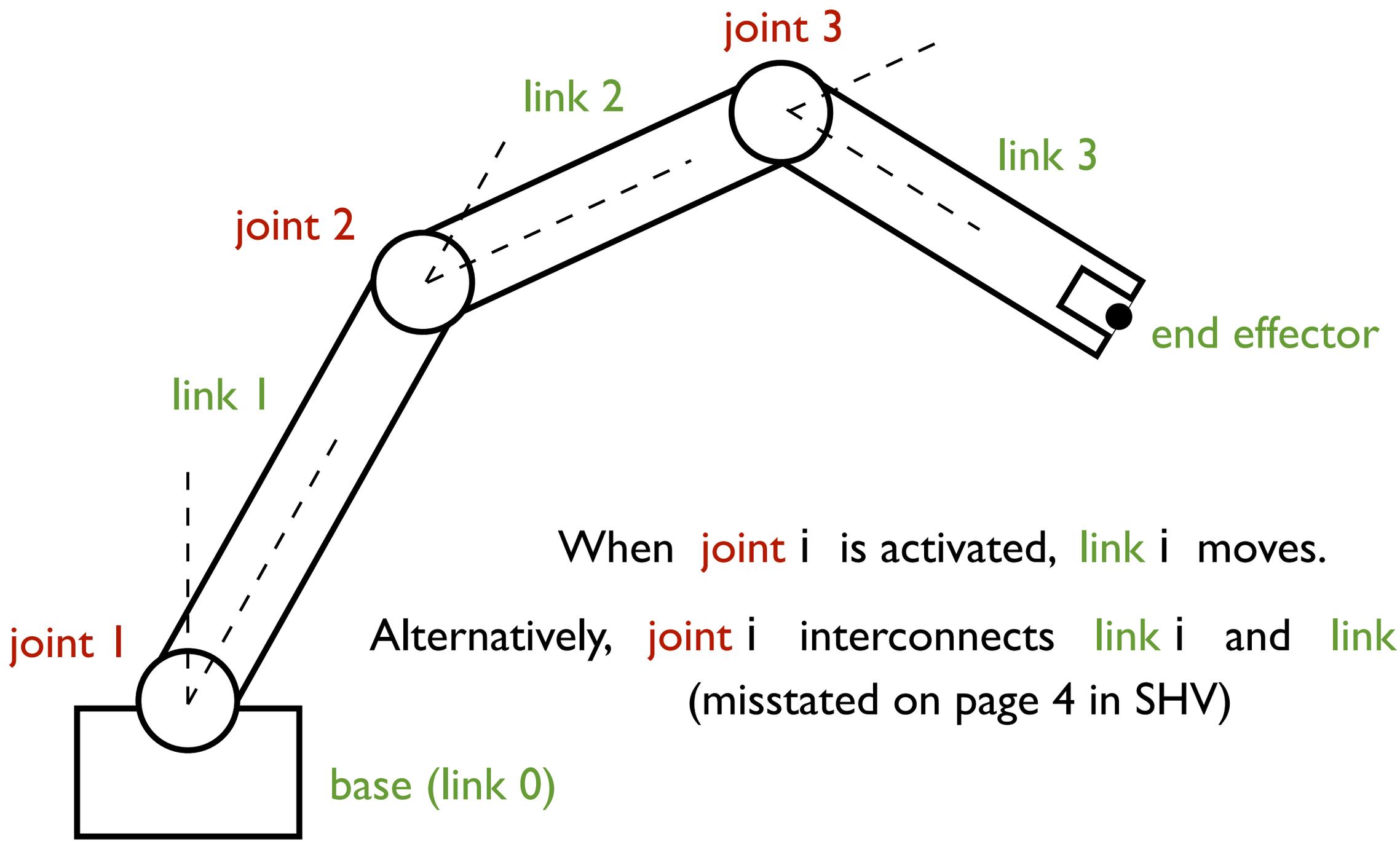
Prismatic : linear displacement between adjacent links



We use z-axes to denote joint axes.

Kinematic Chains

A **kinematic chain** is a system of rigid bodies connected by joints



When joint i is activated, link i moves.

Alternatively, joint i interconnects link i and link i−1

(misstated on page 4 in SHV)

In a **serial** kinematic chain, each intermediate link is connected to two others

Errata: Robot Modeling and Control

Mark W. Spong, Seth Hutchinson, and M. Vidyasagar

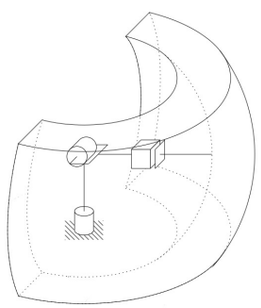
September 6, 2014

Compiled by Katherine J. Kuchenbecker and others affiliated with MEAM 520 at the University of Pennsylvania, expanding on the errata from Seth Hutchinson.

Chapter 1

Page 4 Near the bottom of the page, change "if the joint is the interconnection of links i and $i+1$" to "if the joint is the interconnection of links i and $i-1$".

Page 18 In Figure 1.17(a) the workspace for the spherical robot is drawn too low. The spherical surface should be centered on the intersection between the two revolute joints, as shown in the image below, which was taken from an early draft of the book:

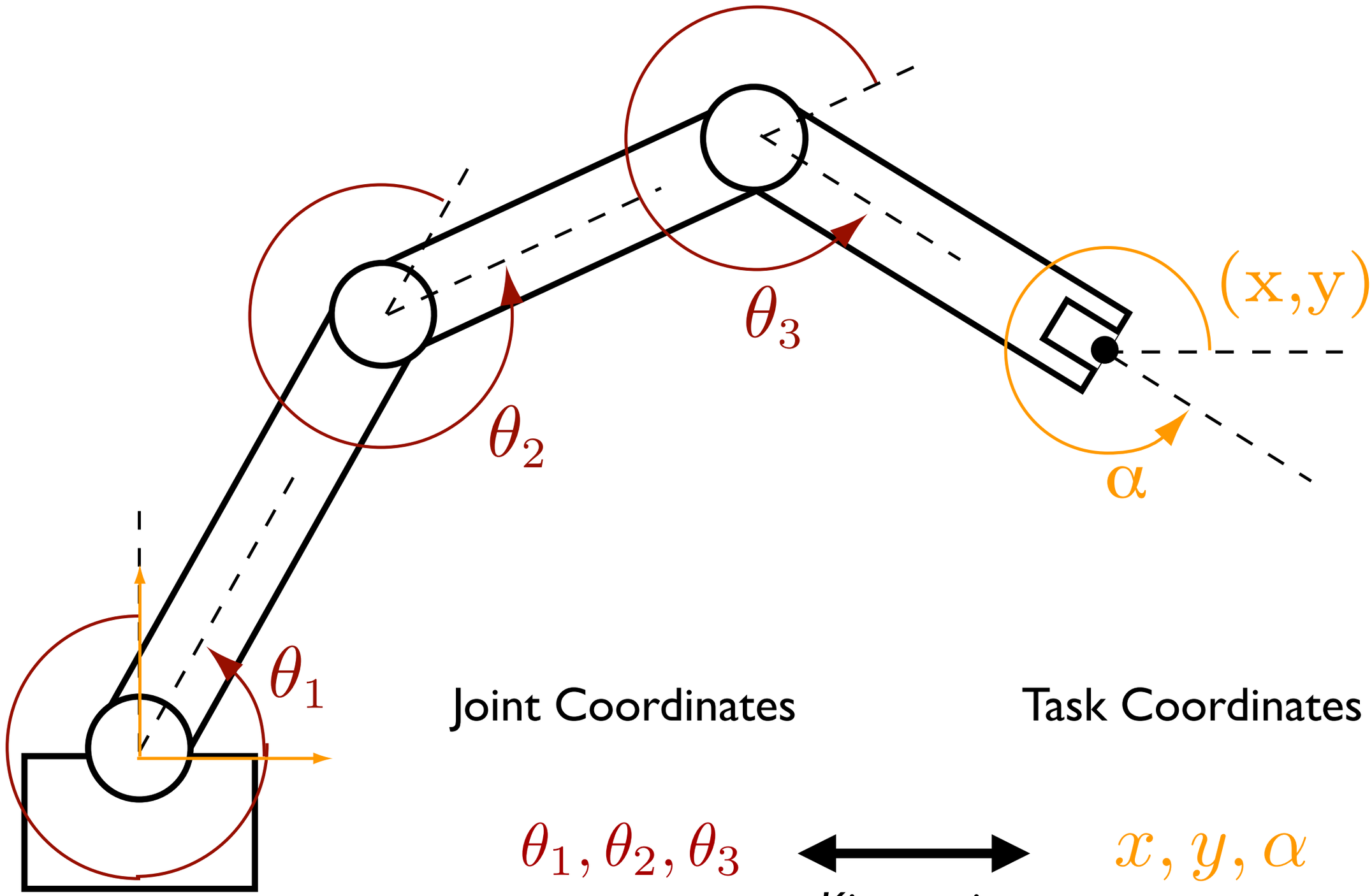


Who has looked at the errata?

Correct the known errors in your textbook.

Report any new errors that you find, and earn extra credit!

Joint and Task Coordinates

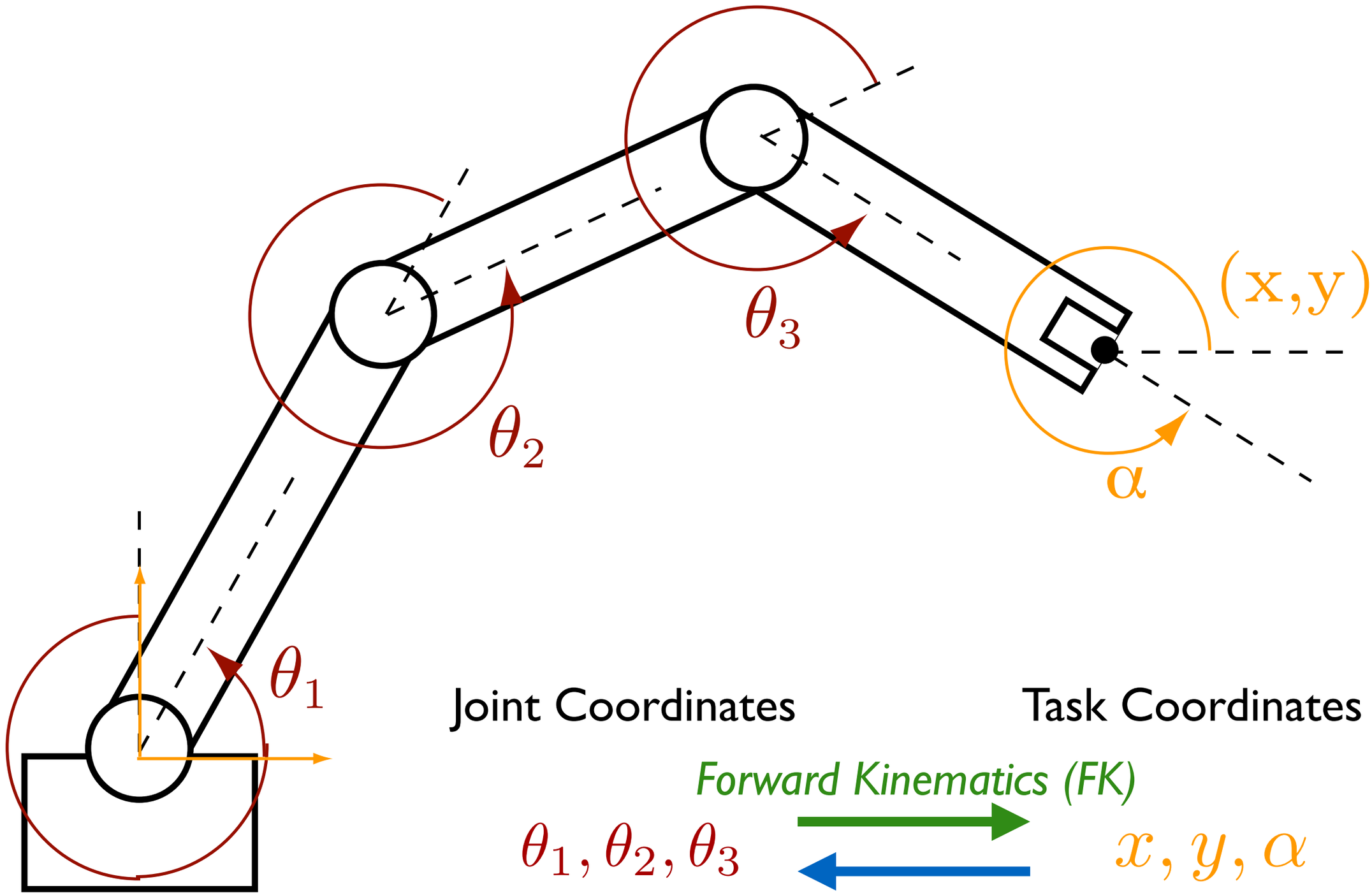


Joint Coordinates | Task Coordinates

$\theta_1, \theta_2, \theta_3 \longleftrightarrow x, y, \alpha$

Kinematics

Joint and Task Coordinates



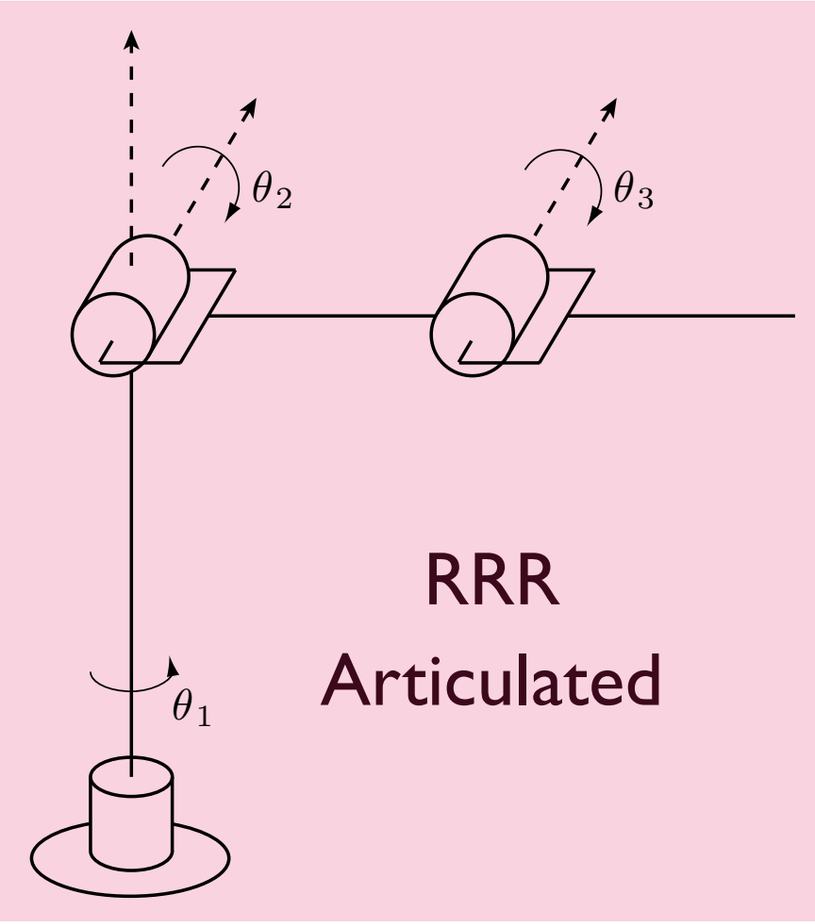
Joint Coordinates | Task Coordinates

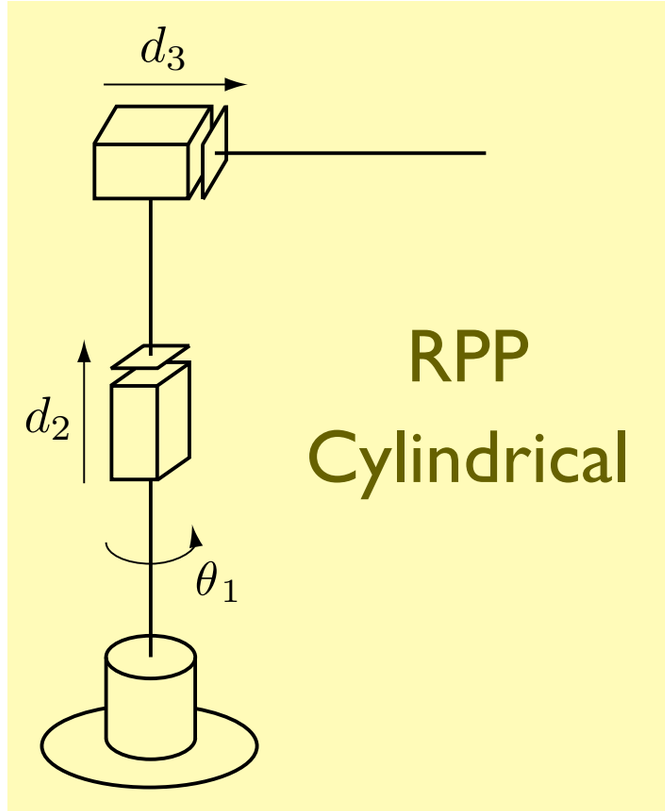
Forward Kinematics (FK)

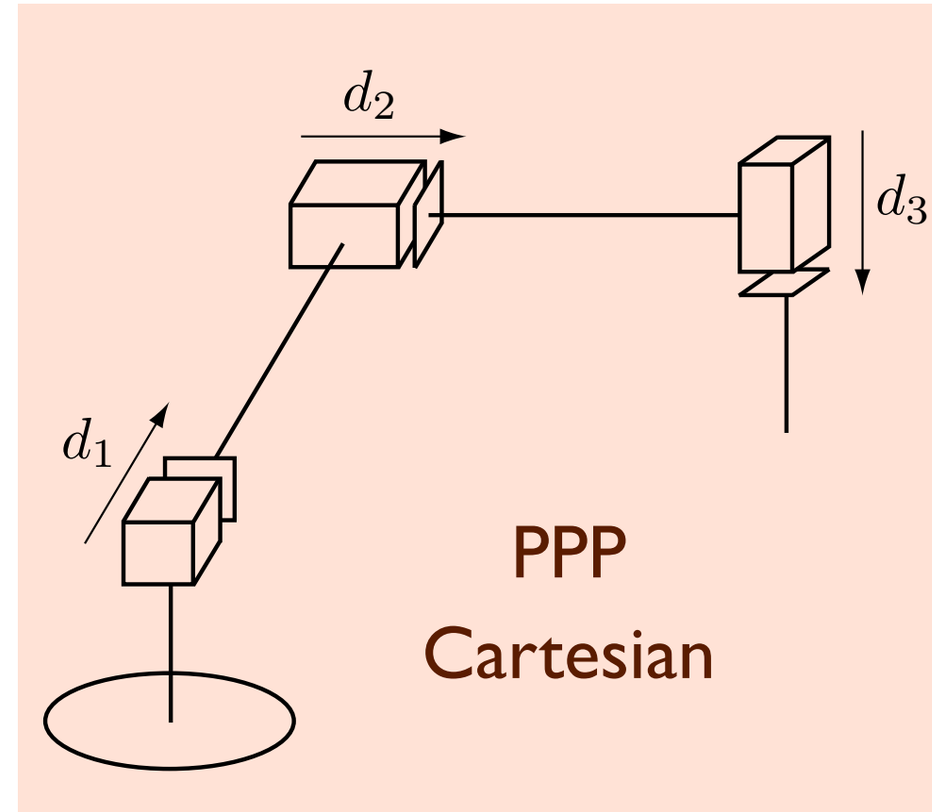
$\theta_1, \theta_2, \theta_3 \rightleftarrows x, y, \alpha$

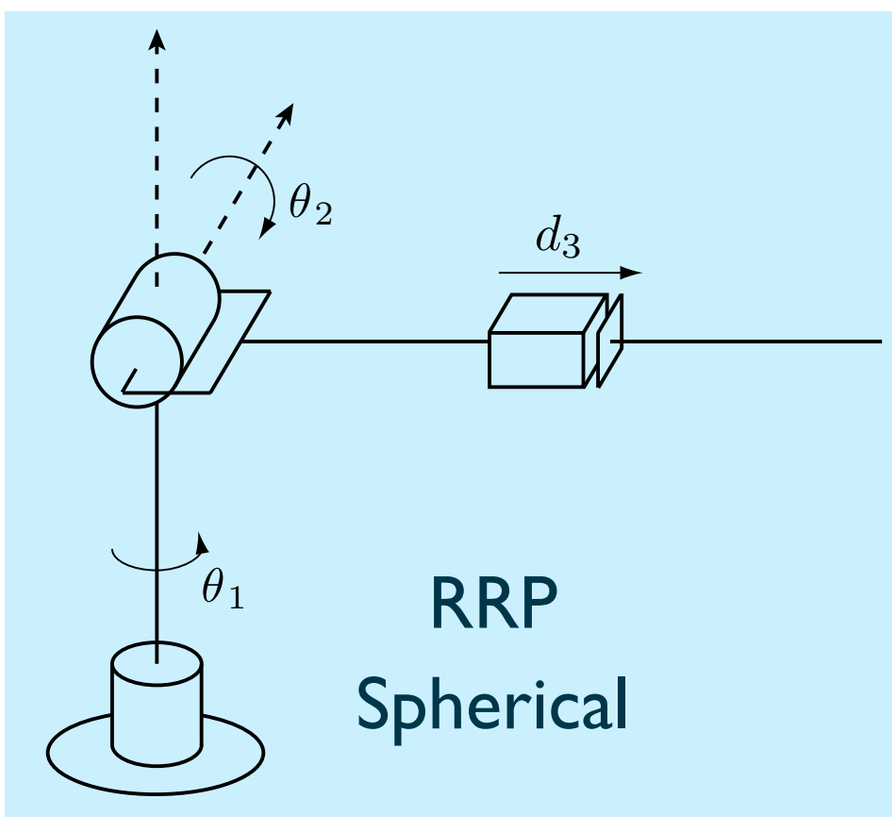
Inverse Kinematics (IK)

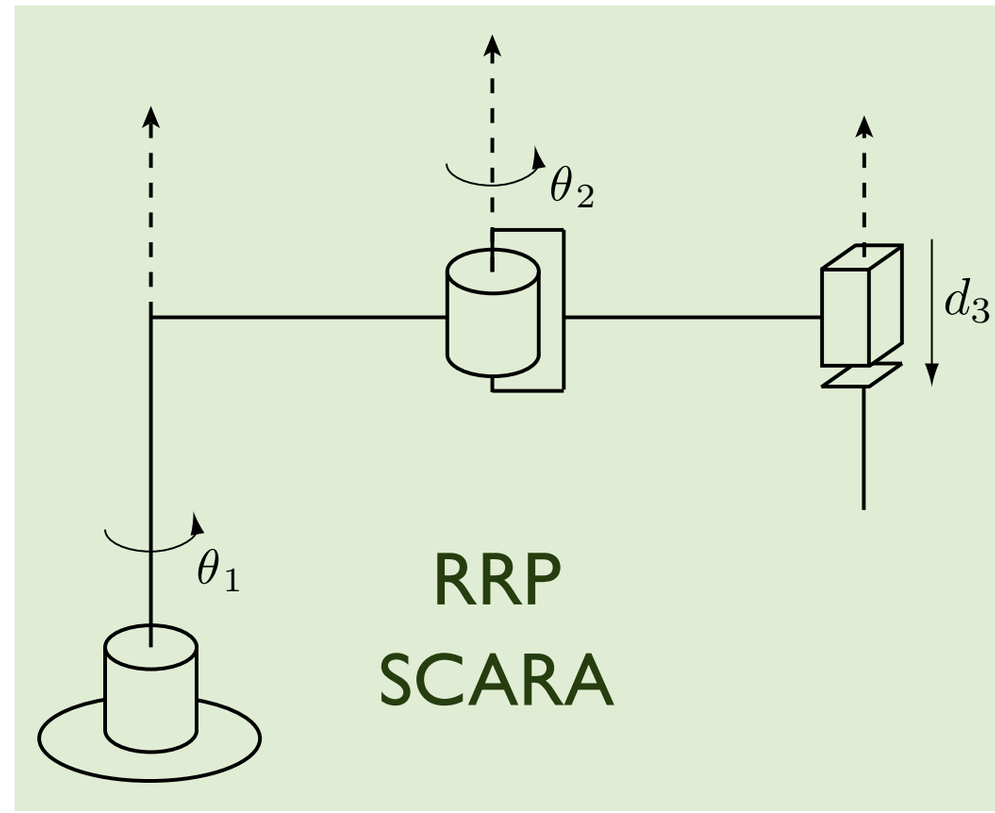
Common Manipulator Designs



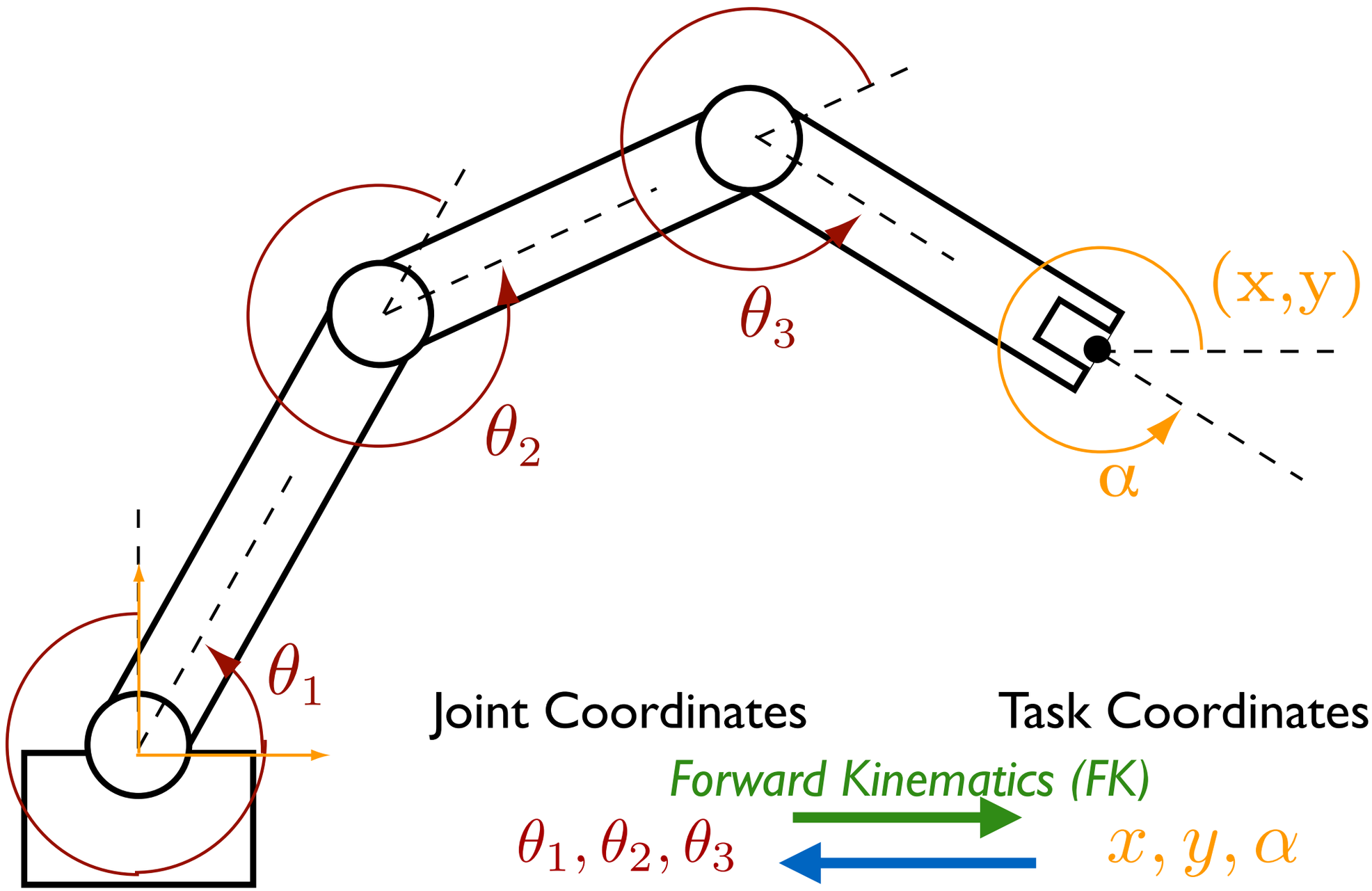








What questions do you have?

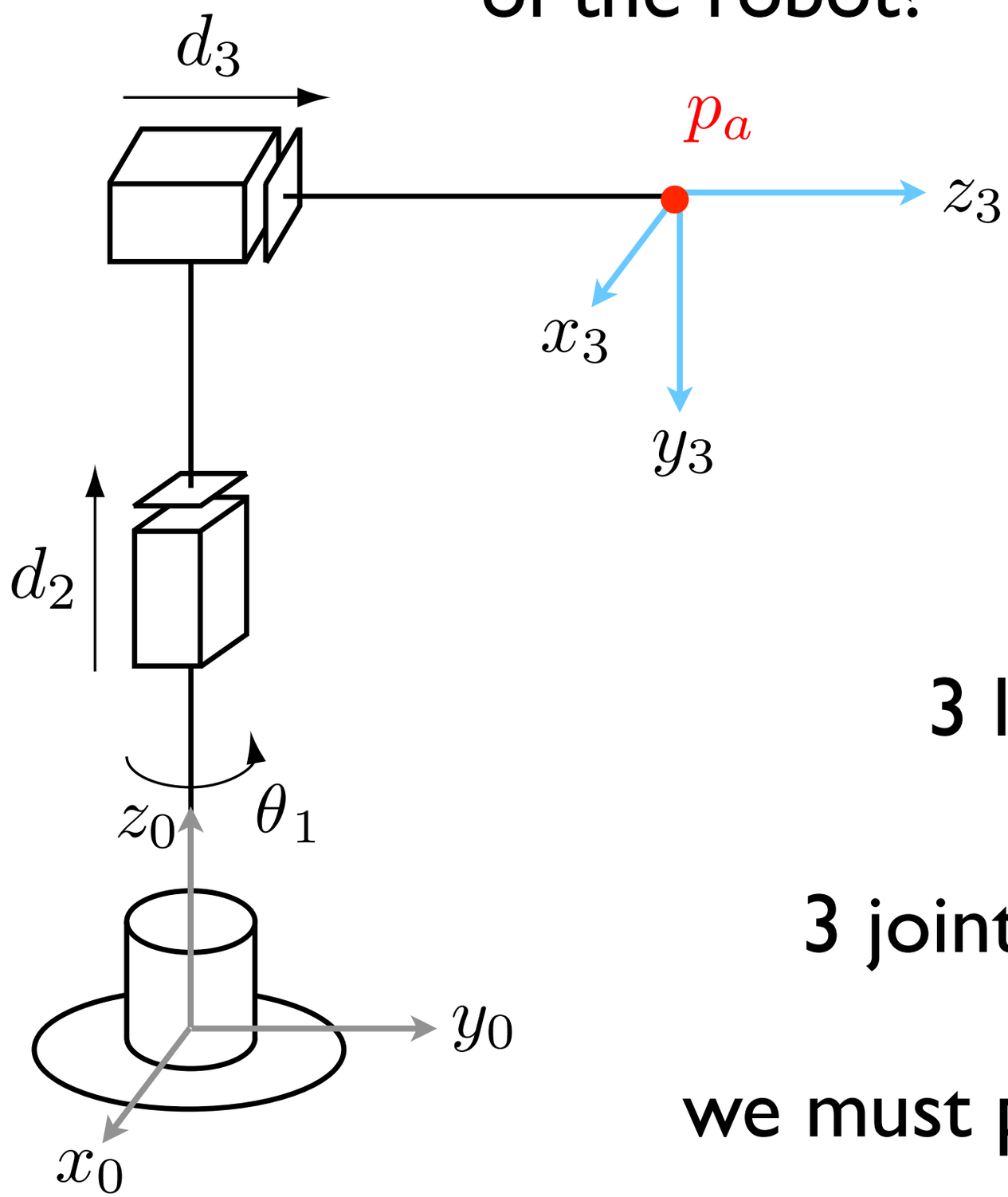


Where is the tip of the robot?

Forward Kinematics

Ignore final (vertical) joint.

Given (q_1, q_2, q_3), where is the tip of the robot?



3 links plus ground

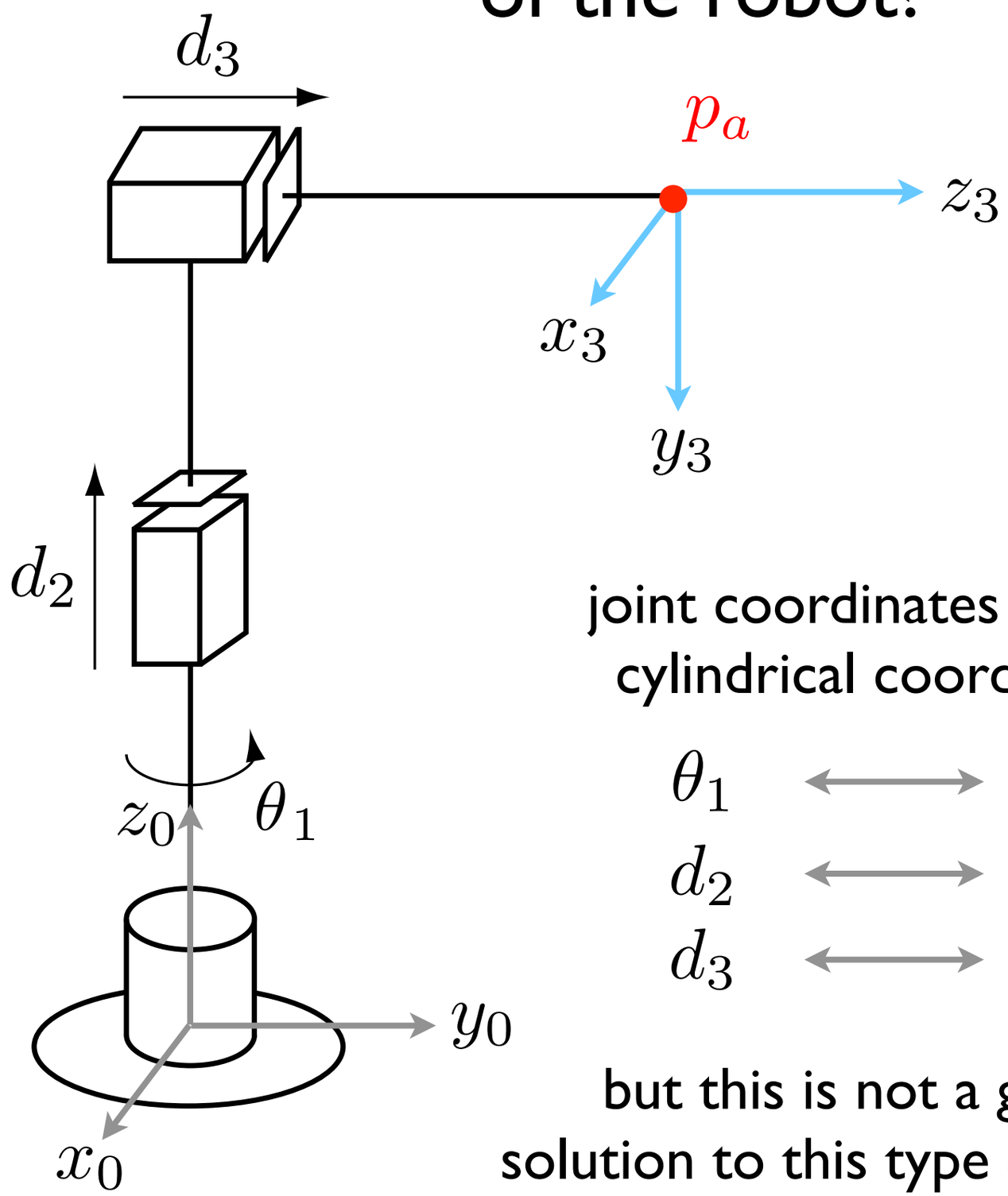
3 joints

3 joint variables (q_1, q_2, q_3)

we must place coordinate frames

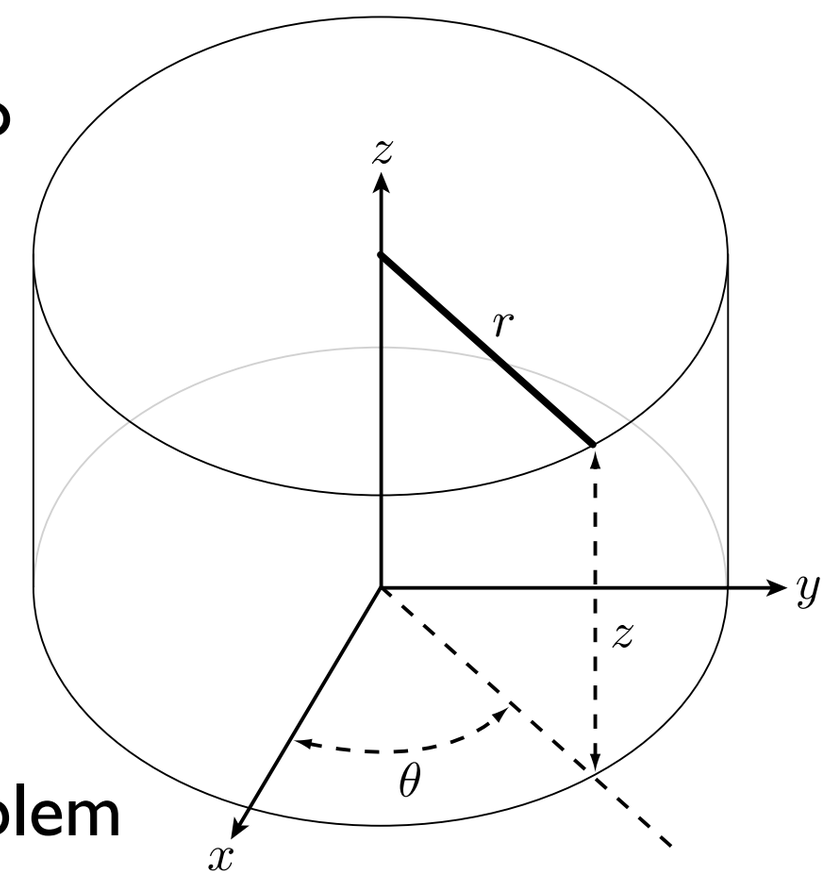
Ignore final (vertical) joint.

Given (q_1, q_2, q_3), where is the tip of the robot?



joint coordinates map to cylindrical coordinates

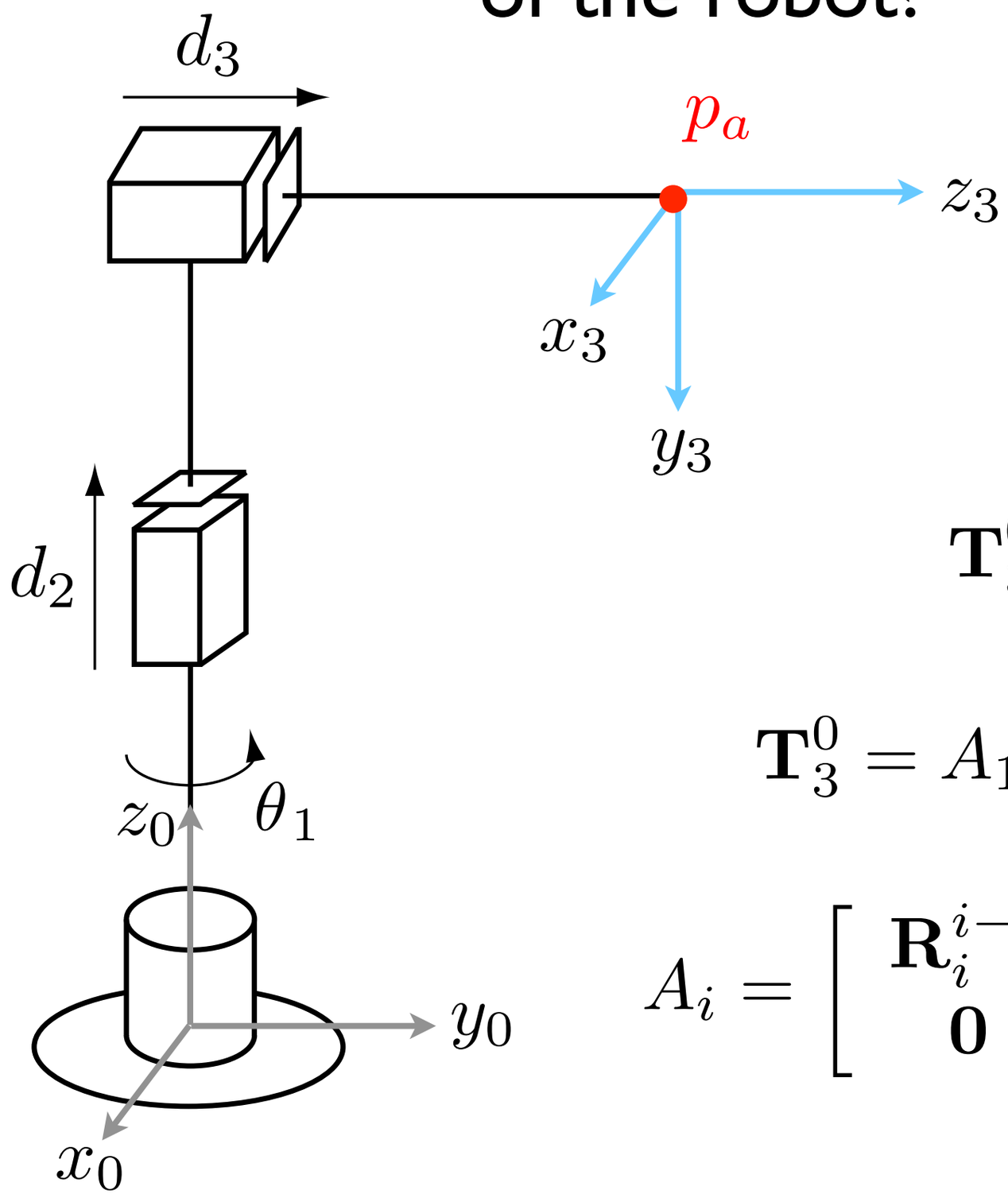
$$\theta_1 \longleftrightarrow \theta$$
$$d_2 \longleftrightarrow z$$
$$d_3 \longleftrightarrow r$$



but this is not a general solution to this type of problem

Ignore final (vertical) joint.

Given (q_1, q_2, q_3), where is the tip of the robot?



$$\mathbf{T}_3^0 = \ ?$$

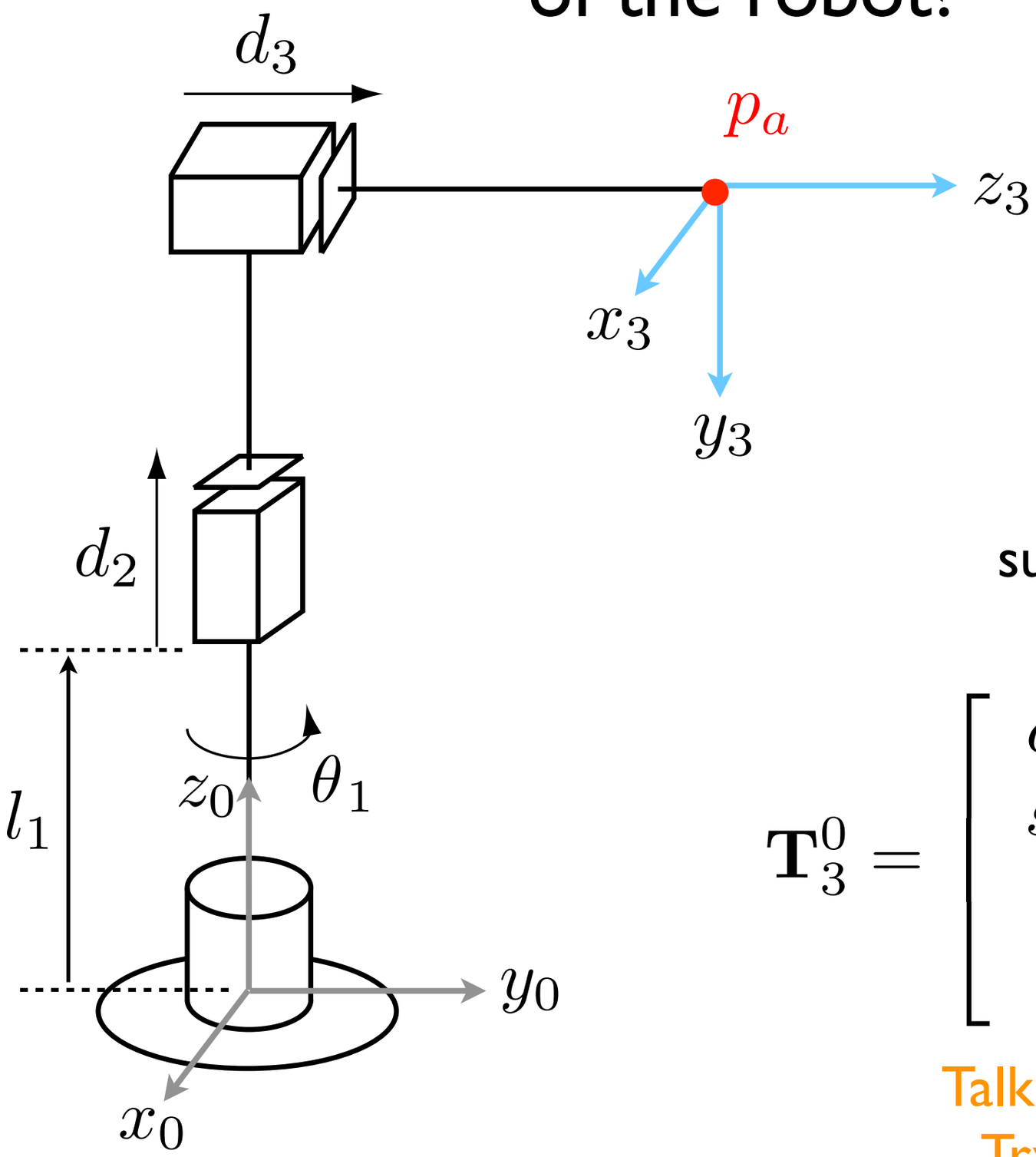
$$\mathbf{T}_3^0 = A_1(q_1)A_2(q_2)A_3(q_3)$$

$$A_i = \begin{bmatrix} \mathbf{R}_i^{i-1} & \mathbf{d}_i^{i-1} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$\mathbf{P}_a^0 = \mathbf{T}_3^0 \mathbf{P}_a^3$$

Ignore final (vertical) joint.

Given (q_1, q_2, q_3), where is the tip of the robot?



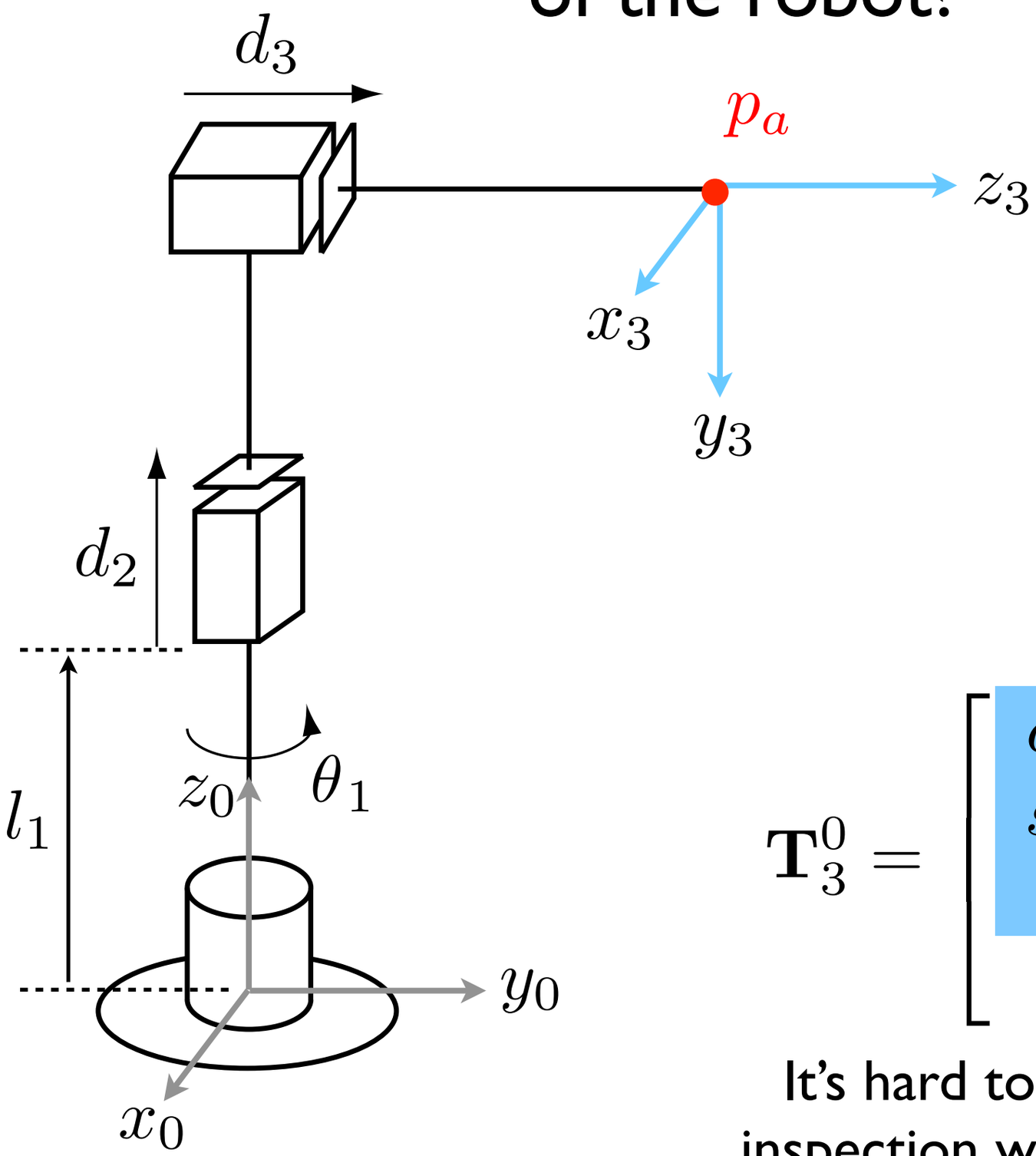
superscript * marks joint angles, which vary over time

$$\mathbf{T}_3^0 = \begin{bmatrix} c_1^* & 0 & -s_1^* & -d_3^* s_1^* \\ s_1^* & 0 & c_1^* & d_3^* c_1^* \\ 0 & -1 & 0 & d_2^* + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Talk about this matrix with a partner.
Try to understand every element.

Ignore final (vertical) joint.

Given (q_1, q_2, q_3), where is the tip of the robot?



Explain this matrix to me.

$$\mathbf{T}_3^0 = \begin{bmatrix} c_1^* & 0 & -s_1^* & -d_3^* s_1^* \\ s_1^* & 0 & c_1^* & d_3^* c_1^* \\ 0 & -1 & 0 & d_2^* + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It's hard to write transformation matrices by inspection when the robot is more complicated.

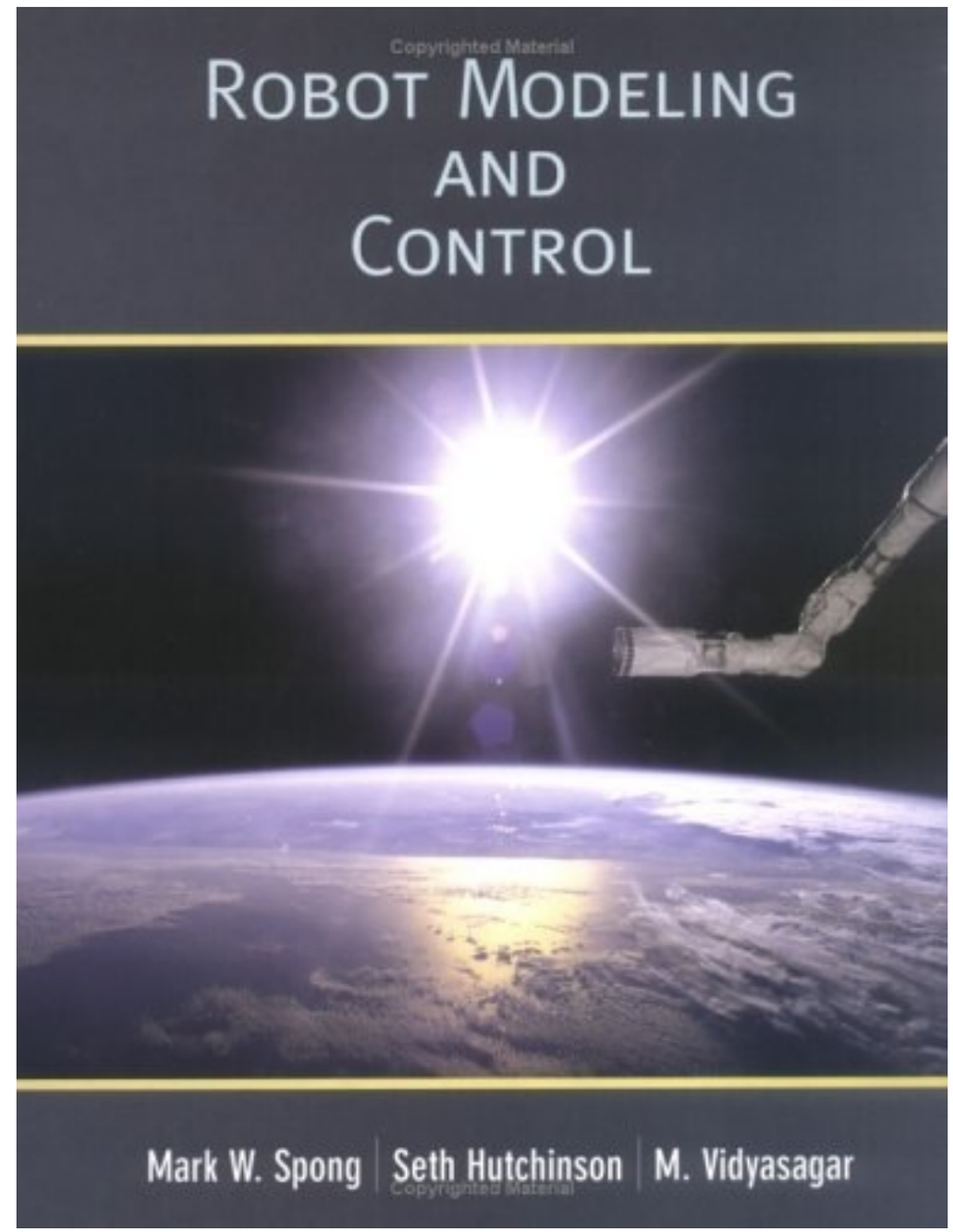
This is the general idea of forward kinematics for manipulators.

There are many choices one must make regarding placement of intermediate frames and definition of joint variables, which means there are many equally good ways to reach the same final solution.

The robotics community has agreed on a set of conventions to ensure uniformity:
The Denavit-Hartenberg (DH) Convention.

Reading Assignment

Within Chapter 3: Forward and Inverse Kinematics

- Read Sec. 3.1 and the first half of Sec. 3.2, up through Example 3.2 (pages 73–86)

Deadline: Tuesday lecture