MEAM 520
Lecture 3: Rotations

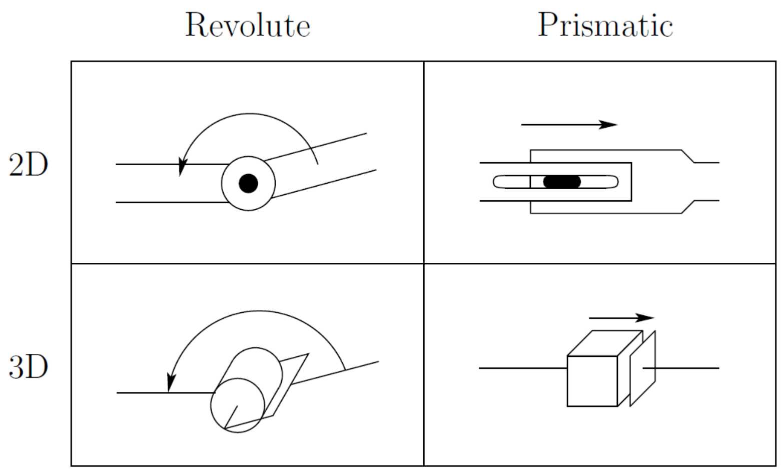
Cynthia Sung, Ph.D.

Mechanical Engineering & Applied Mechanics

University of Pennsylvania

Last Time

Manipulators are **links** connected by **joint**, which can be **R** or **P** type.

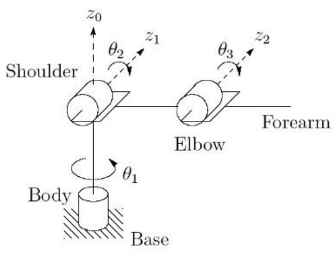
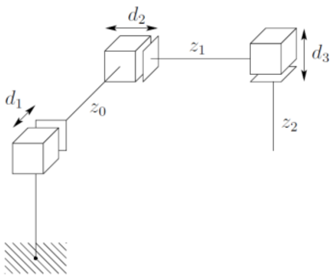
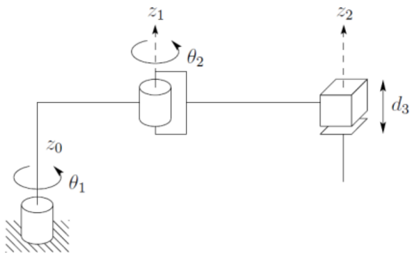
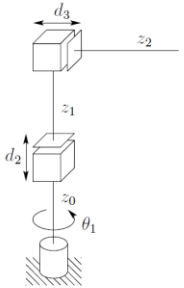
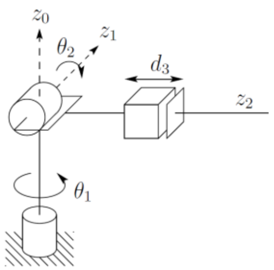
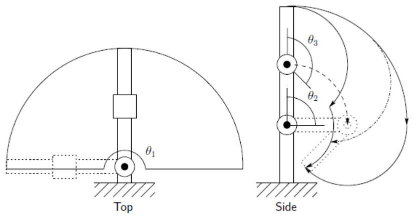
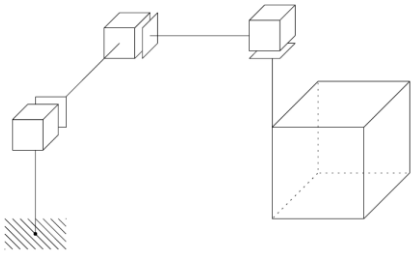
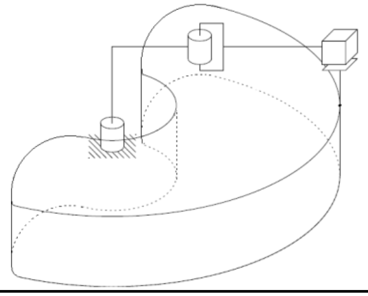
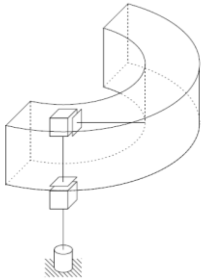
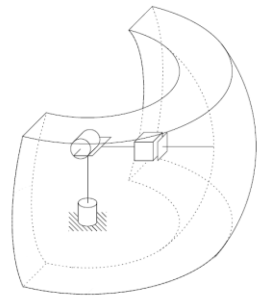


Joint variables θ and d

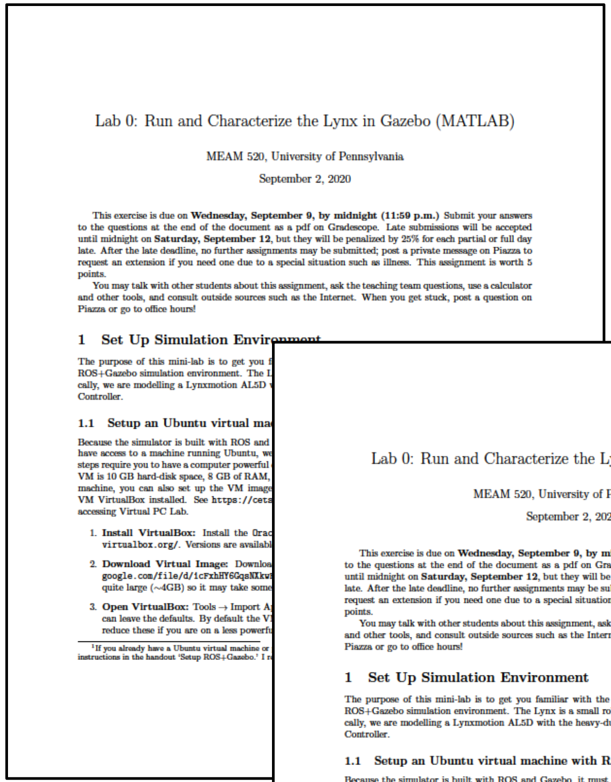
Zero configuration

Degrees of freedom (DOF)

Configuration Space vs Workspace vs Task Space vs State Space

<table>
<tr><th>Articulated (RRR)</th><th>Cartesian (PPP)</th><th>SCARA (RRP)</th><th>Cylindrical (RPP)</th><th>Spherical (RRP)</th></tr>
<tr><td>small workspaces</td><td>gantries</td><td>speed, planar tasks</td><td>material transfer</td><td>earliest designs</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>
</td><td>
</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td>
</td><td></td><td></td><td></td><td></td></tr>
</table>

Lab 0 is posted (due 9/9)



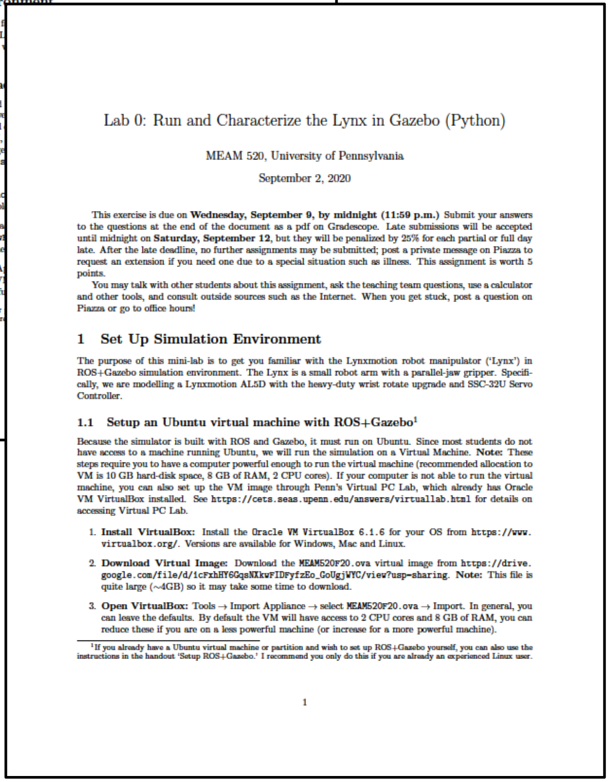
Lab 0: Run and Characterize the Lynx in Gazebo (MATLAB)

MEAM 520, University of Pennsylvania

September 2, 2020

This exercise is due on **Wednesday, September 9, by midnight (11:59 p.m.)** Submit your answers to the questions at the end of the document as a pdf on Gradescope. Late submissions will be accepted until midnight on **Saturday, September 12**, but they will be penalized by 25% for each partial or full day late. 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. This assignment is worth 5 points.

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. When you get stuck, post a question on Piazza or go to office hours!



Lab 0: Run and Characterize the Lynx in Gazebo (Python)

MEAM 520, University of Pennsylvania

September 2, 2020

This exercise is due on **Wednesday, September 9, by midnight (11:59 p.m.)** Submit your answers to the questions at the end of the document as a pdf on Gradescope. Late submissions will be accepted until midnight on **Saturday, September 12**, but they will be penalized by 25% for each partial or full day late. 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. This assignment is worth 5 points.

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. When you get stuck, post a question on Piazza or go to office hours!

1 Set Up Simulation Environment

The purpose of this mini-lab is to get you familiar with the Lynxmotion robot manipulator ('Lynx') in ROS+Gazebo simulation environment. The Lynx is a small robot arm with a parallel-jaw gripper. Specifically, we are modelling a Lynxmotion AL5D with the heavy-duty wrist rotate upgrade and SSC-32U Servo Controller.

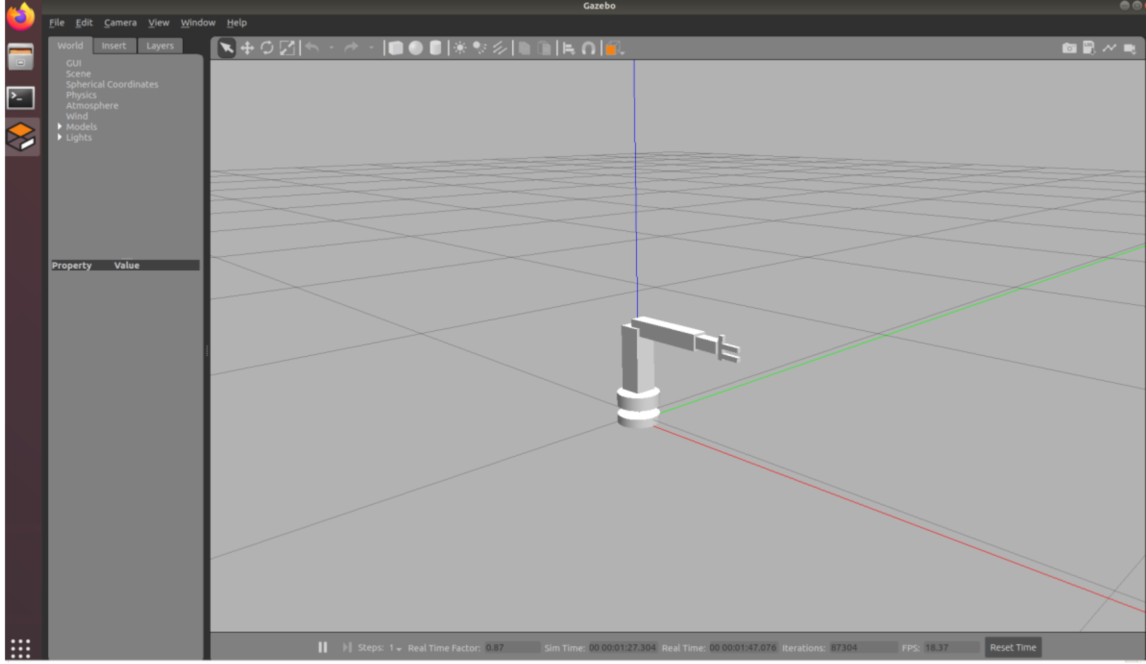
1.1 Setup an Ubuntu virtual machine with ROS+Gazebo[1]

Because the simulator is built with ROS and Gazebo, it must run on Ubuntu. Since most students do not have access to a machine running Ubuntu, we will run the simulation on a Virtual Machine. **Note:** These steps require you to have a computer powerful enough to run the virtual machine (recommended allocation to VM is 10 GB hard-disk space, 8 GB of RAM, 2 CPU cores). If your computer is not able to run the virtual machine, you can also set up the VM image through Penn's Virtual PC Lab, which already has Oracle VM VirtualBox installed. See `https://cets.seas.upenn.edu/answers/virtuallab.html` for details on accessing Virtual PC Lab.

1. **Install VirtualBox:** Install the `Oracle VM VirtualBox 6.1.6` for your OS from `https://www.virtualbox.org/`. Versions are available for Windows, Mac and Linux.
2. **Download Virtual Image:** Download the `MEAM520F20.ova` virtual image from `https://drive.google.com/file/d/1cFxhHY6GqsNXkwFIDFyfzEo_GoUgjWYC/view?usp=sharing`. **Note:** This file is quite large (~4GB) so it may take some time to download.
3. **Open VirtualBox:** Tools → Import Appliance → select `MEAM520F20.ova` → Import. In general, you can leave the defaults. By default the VM will have access to 2 CPU cores and 8 GB of RAM, you can reduce these if you are on a less powerful machine (or increase for a more powerful machine).

[1] If you already have a Ubuntu virtual machine or partition and wish to set up ROS+Gazebo yourself, you can also use the instructions in the handout 'Setup ROS+Gazebo.' I recommend you only do this if you are already an experienced Linux user.

1



Symbolic representation for the Lynx

Get familiar with the simulation

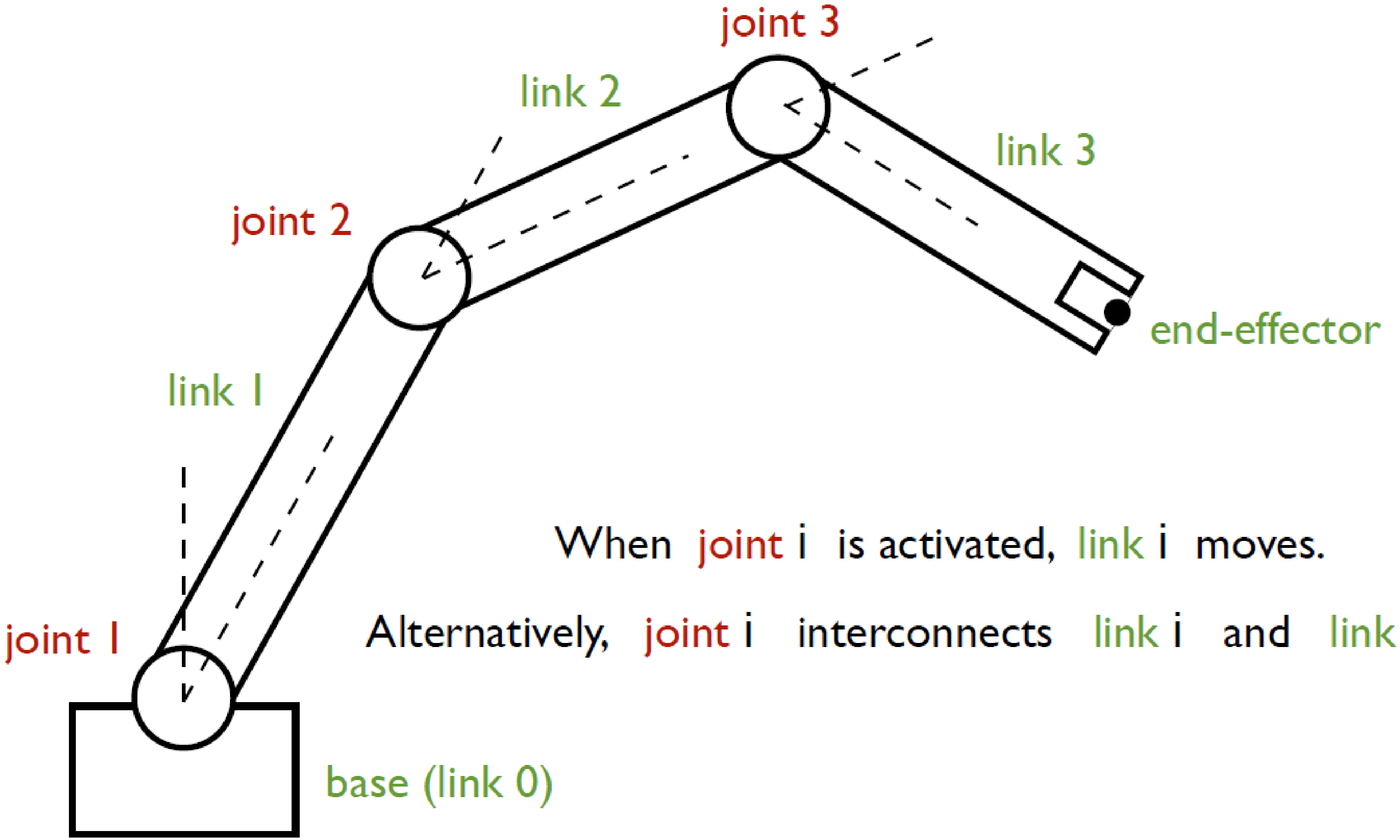
Sketch the workspace
(does not need to be to scale!)

Today: Kinematics

Kinematics is the study of motion without

reference to the causes of that motion.

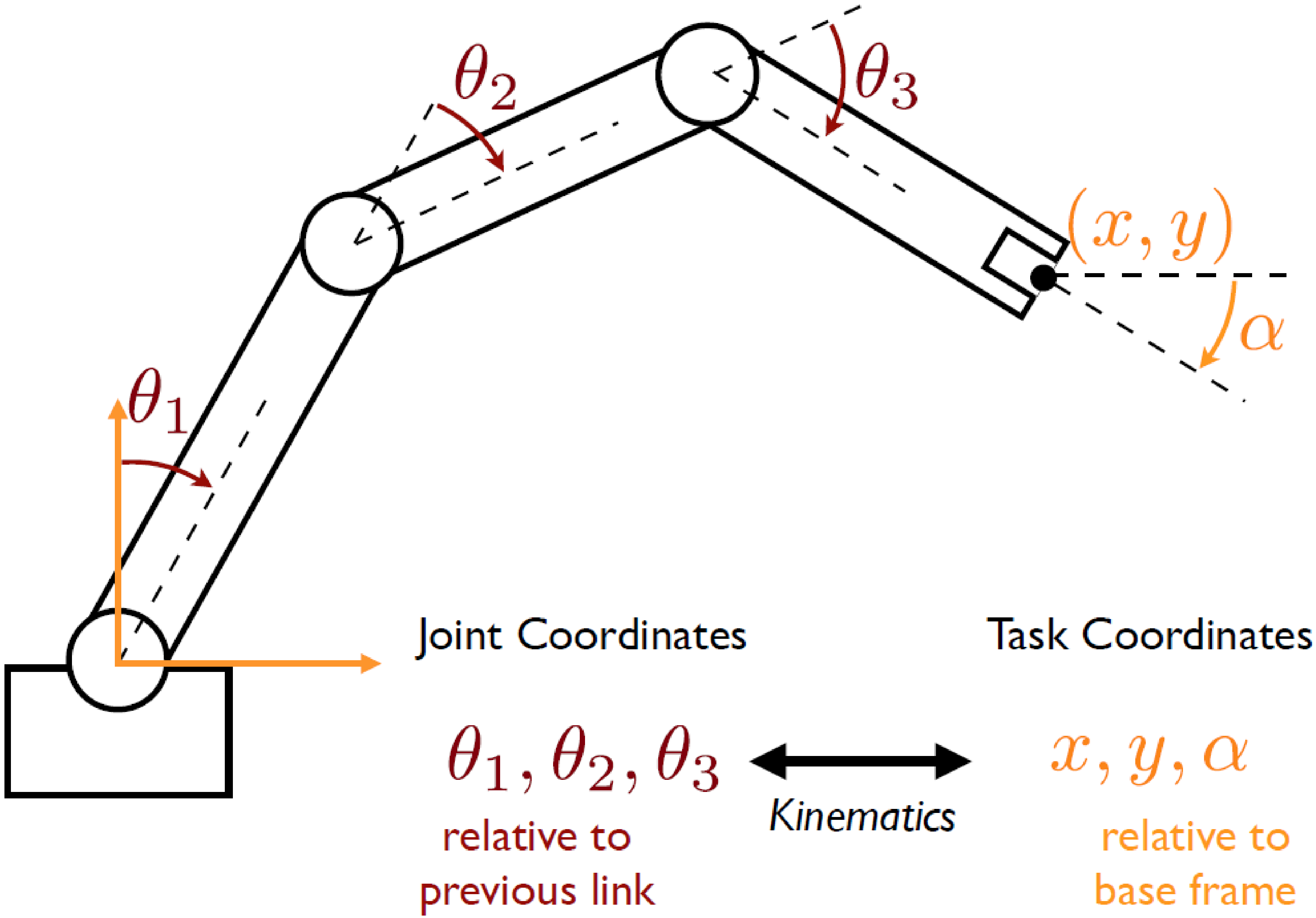
Kinematics



When joint i is activated, link i moves.

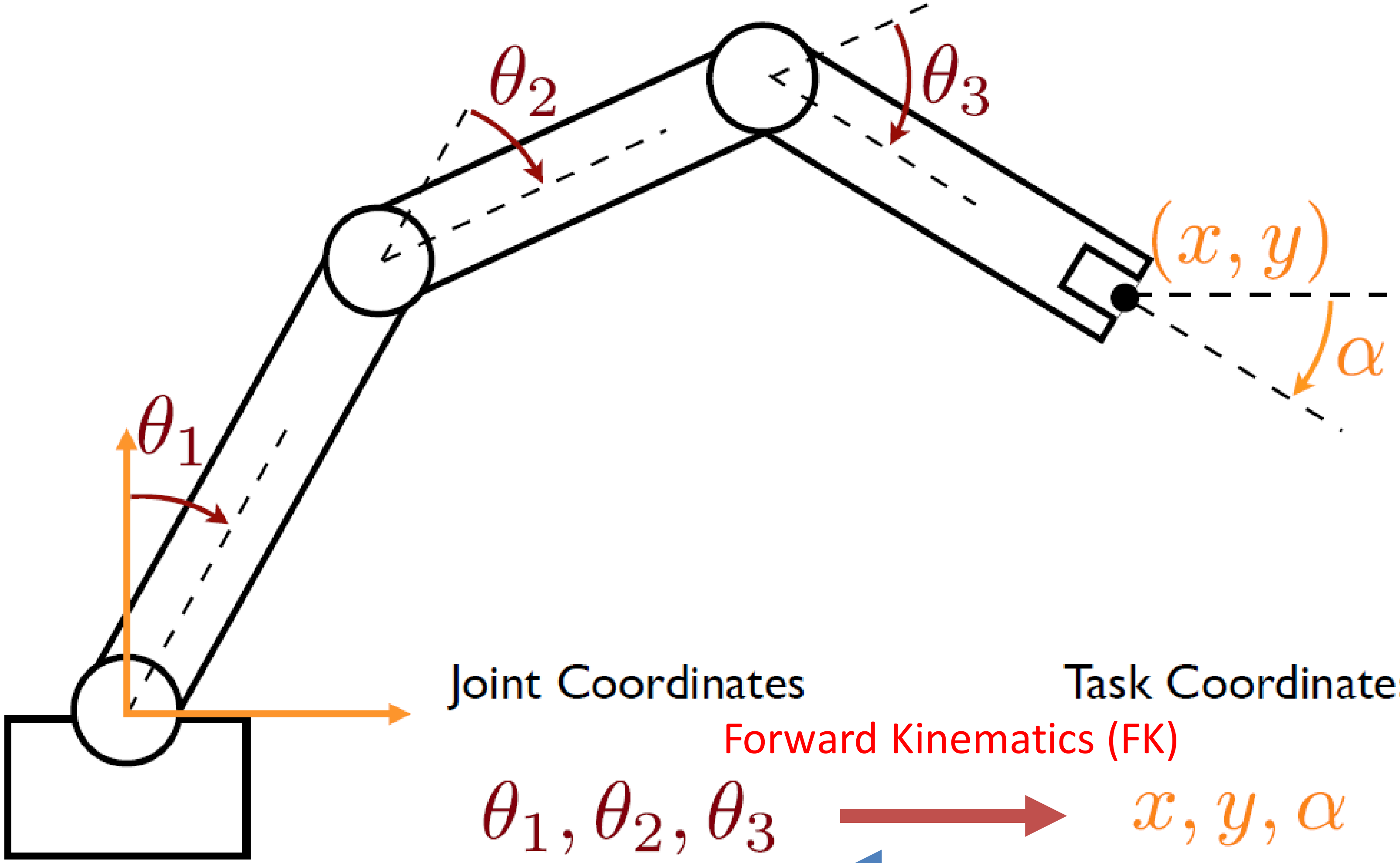
Alternatively, joint i interconnects link i and link i−1

Kinematics



θ_2
θ_3
(x, y)
α
θ_1
Joint Coordinates
Task Coordinates
$\theta_1, \theta_2, \theta_3$
Kinematics
x, y, α
relative to previous link
relative to base frame

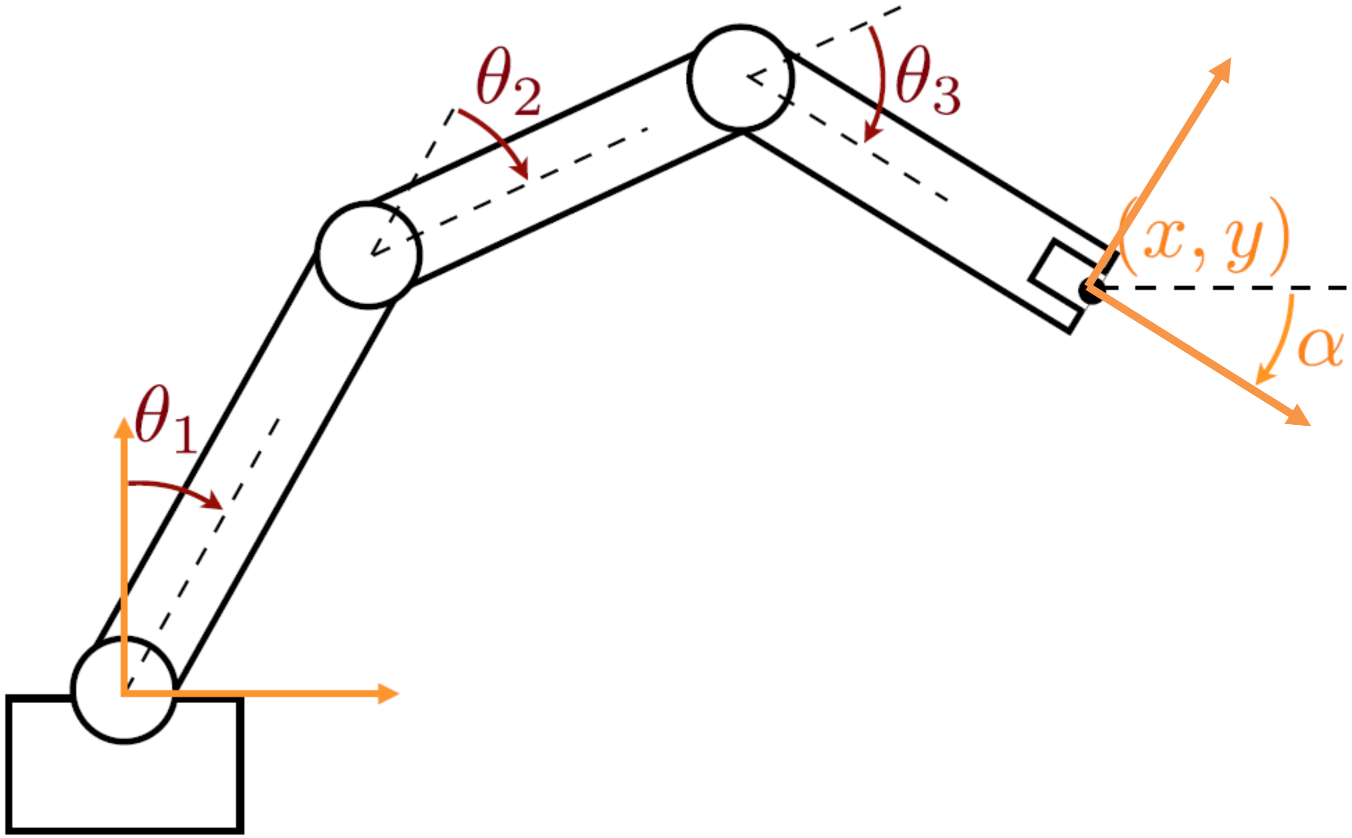
Kinematics



Joint Coordinates		Task Coordinates
	Forward Kinematics (FK) →	
$\theta_1, \theta_2, \theta_3$		x, y, α
relative to previous link	← Inverse Kinematics (IK)	relative to base frame

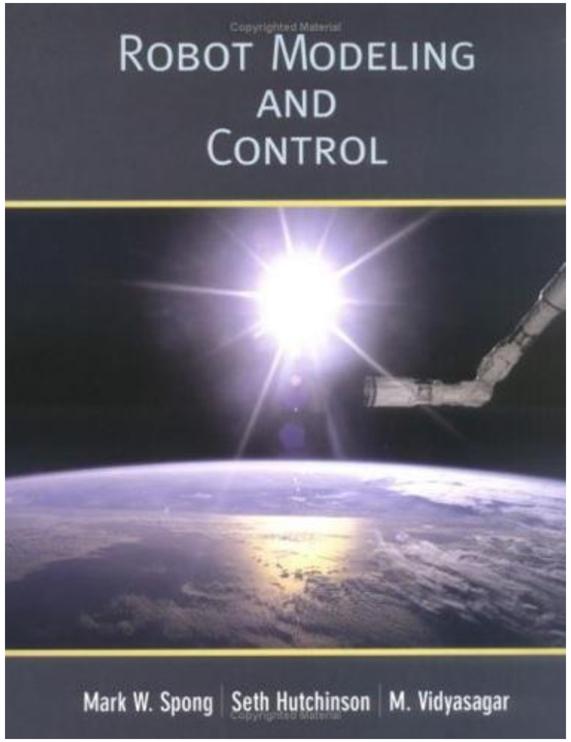
Forward Kinematics

Given the joint coordinates, what are the task coordinates?



Need to find both orientation and position

Today: Rotations in 2D and 3D

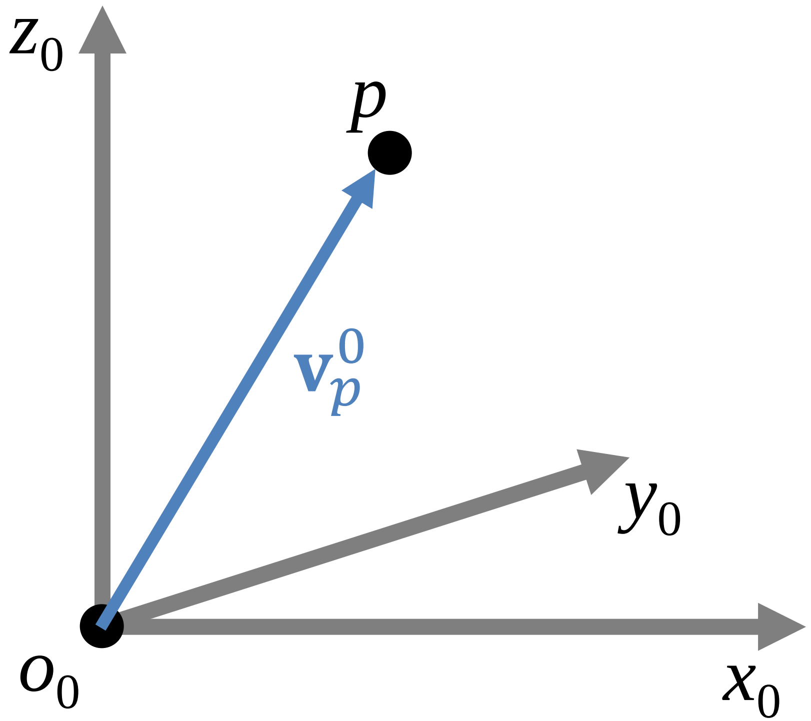


Chapter 2: Rigid Motions

- Sec. 2.intro-2.5 and B.1-B.4

Representing positions

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



Coordinate frame

- an origin (point in space)
- 2 or 3 orthogonal coordinate axes

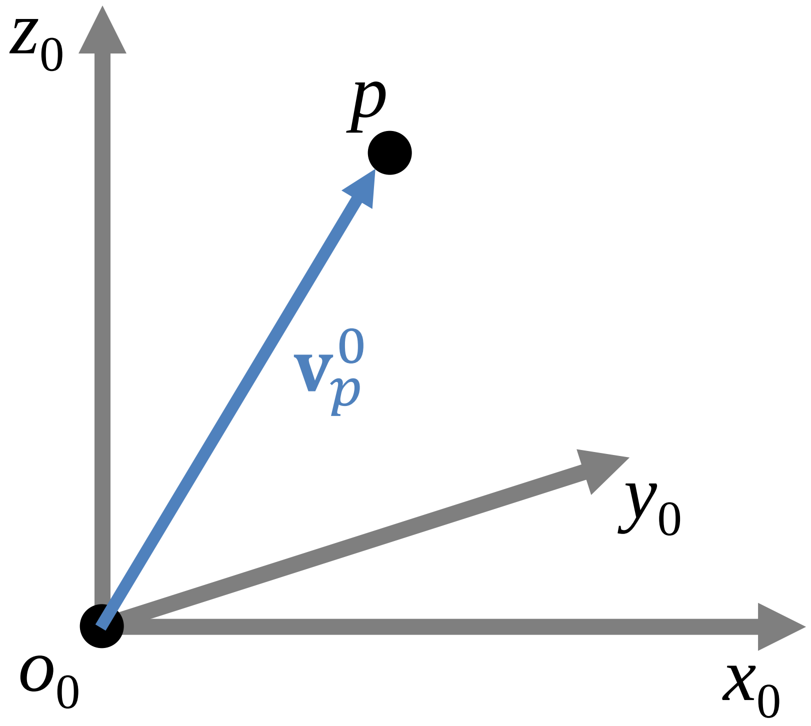
Call this frame $\boldsymbol{o_0 x_0 y_0 z_0}$ or **frame 0**

Point p is written as a vector

superscript = frame

$$\mathbf{v}_p^0 = p^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = [x \quad y \quad z]^\top$$

Vectors



A vector has a **magnitude/length**

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

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

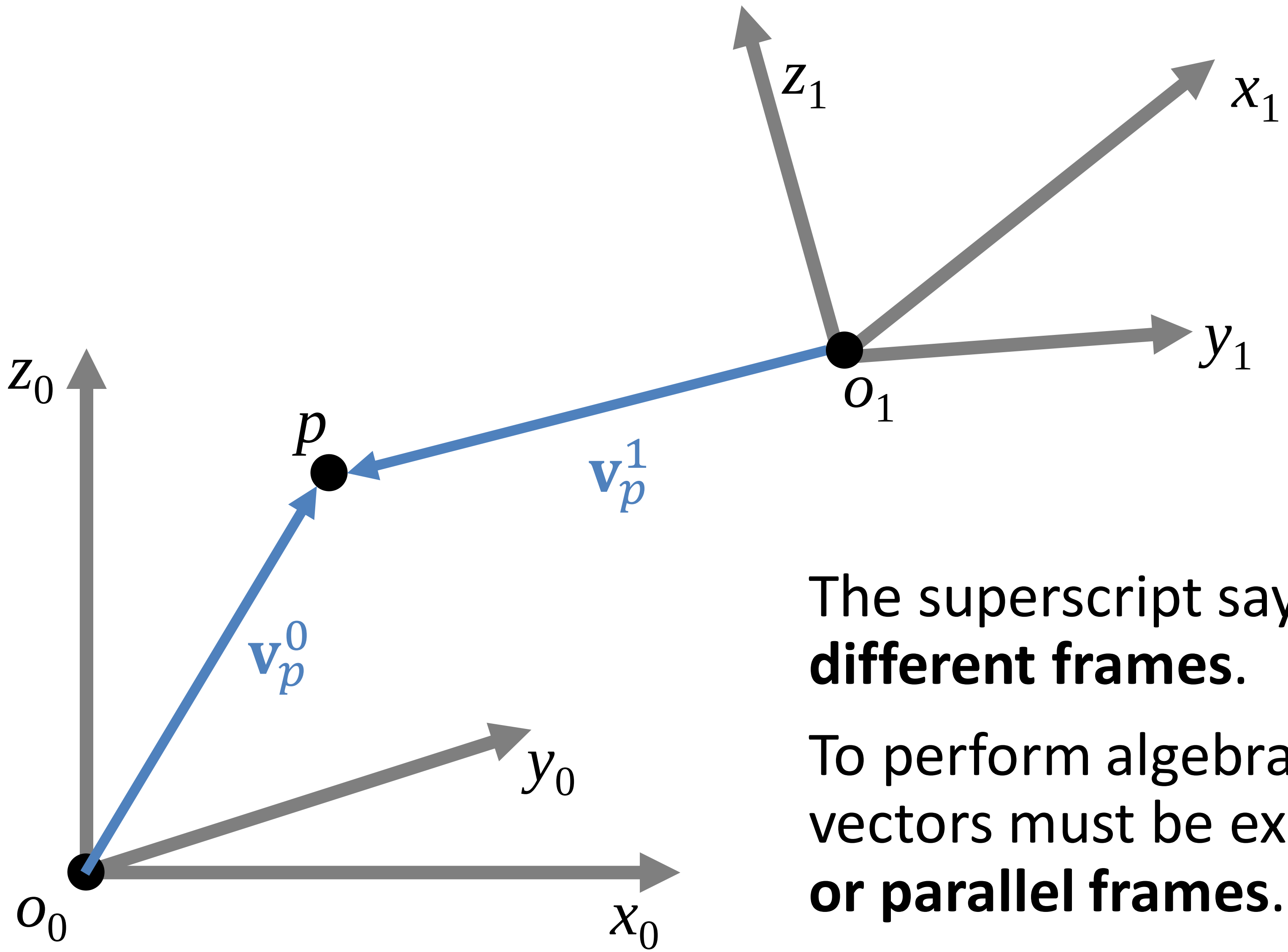
This is called the **ℓ^2 norm**.

A vector has a **direction**

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

This **unit vector** has length 1

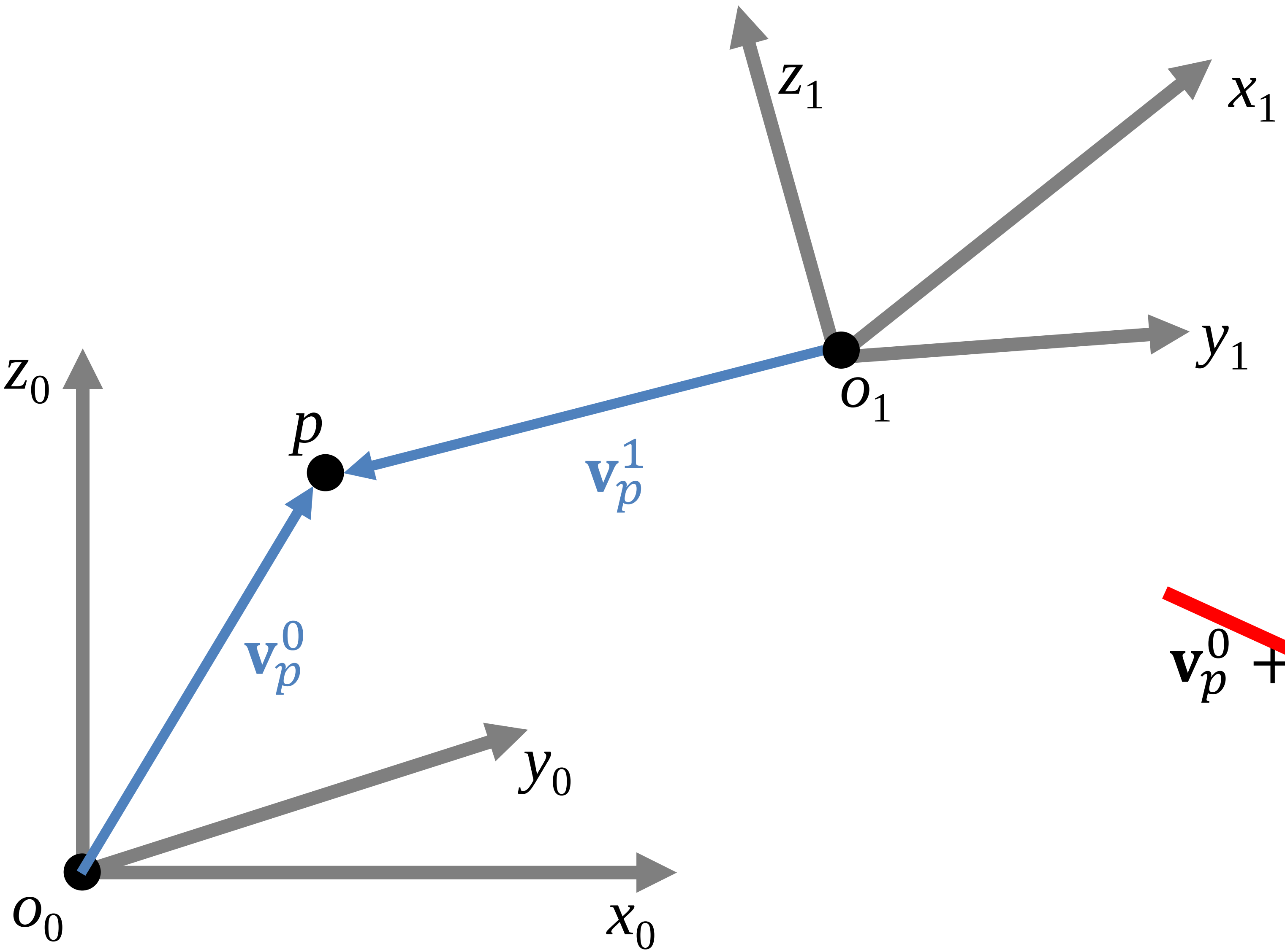
Multiple coordinate frames



The superscript says they are in **different frames**.

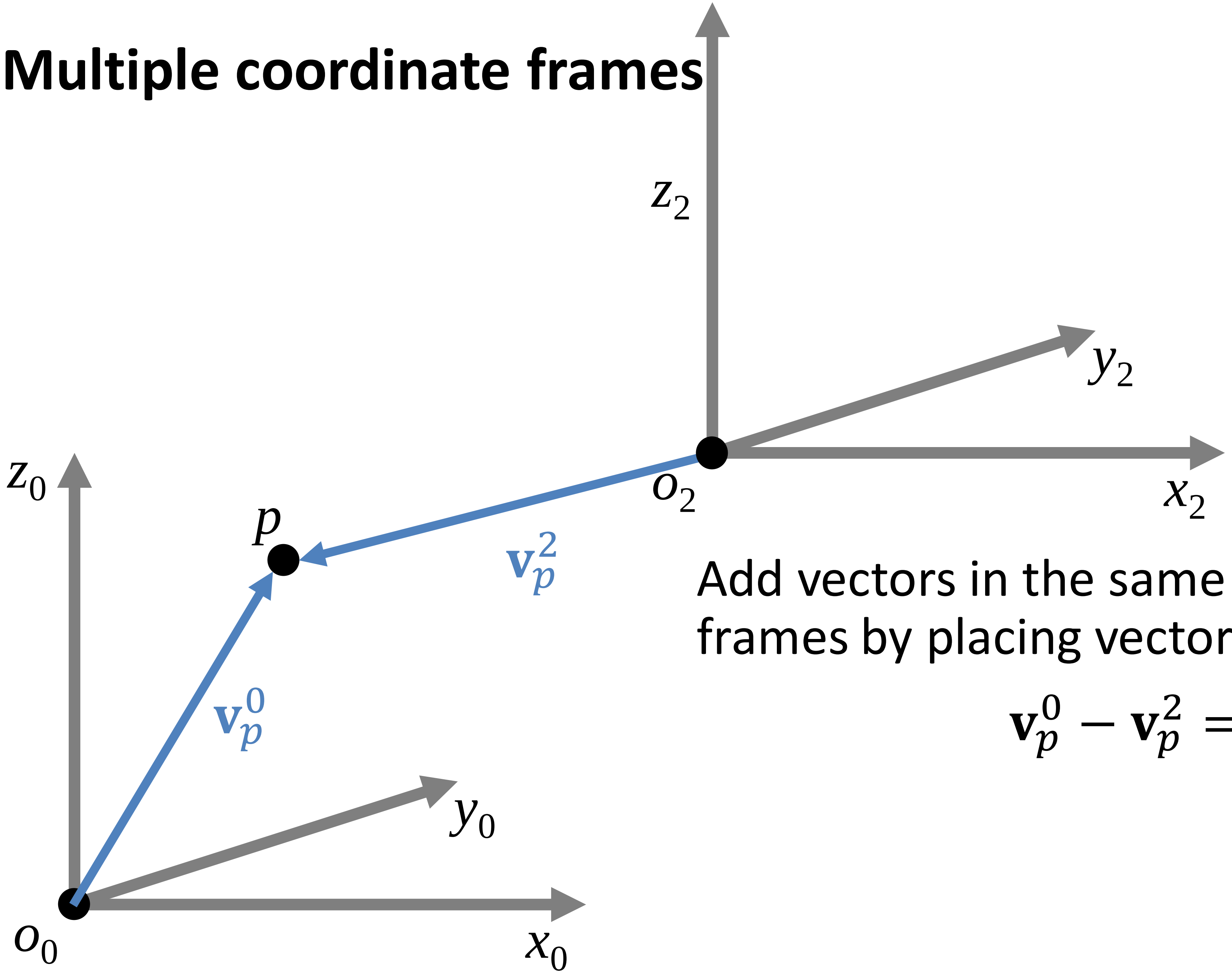
To perform algebraic manipulation, the vectors must be expressed in the **same or parallel frames**.

Multiple coordinate frames



$$\cancel{\mathbf{v}_p^0 + \mathbf{v}_p^1 = ?}$$

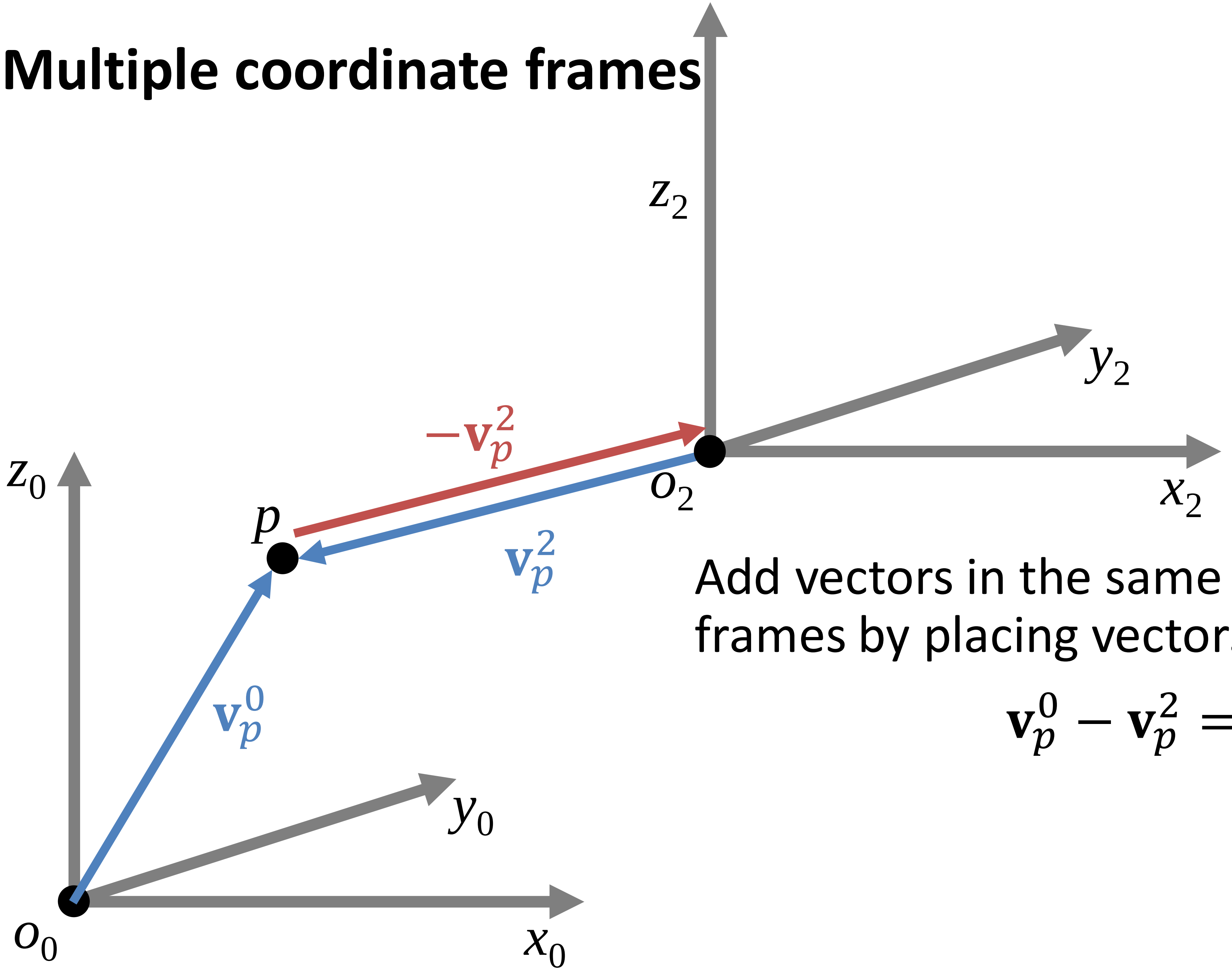
Multiple coordinate frames



Add vectors in the same or parallel frames by placing vectors tip to tail

$$\mathbf{v}_p^0 - \mathbf{v}_p^2 = ?$$

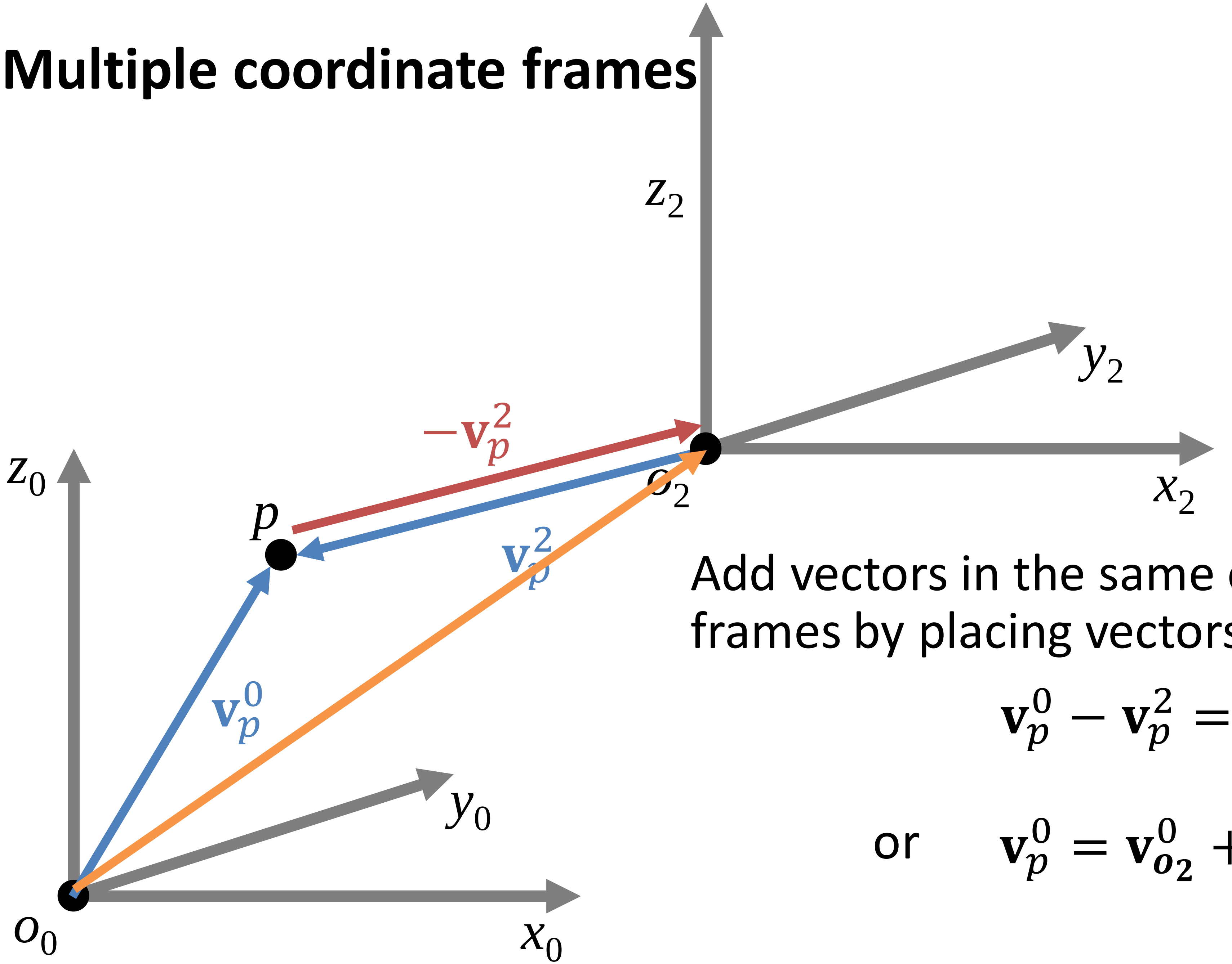
Multiple coordinate frames



Add vectors in the same or parallel frames by placing vectors tip to tail

$$\mathbf{v}_p^0 - \mathbf{v}_p^2 = ?$$

Multiple coordinate frames

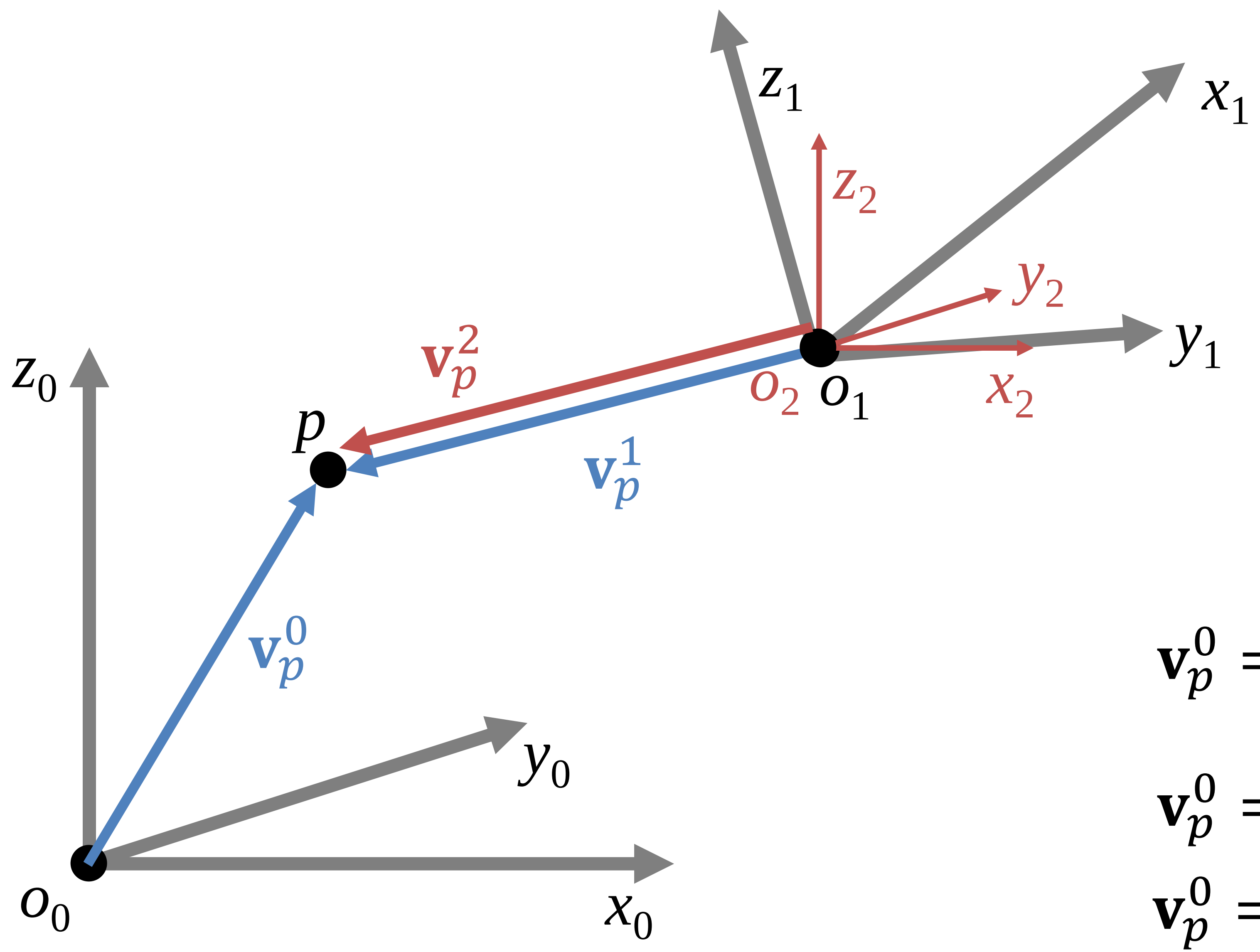


Add vectors in the same or parallel frames by placing vectors tip to tail

$$\mathbf{v}_p^0 - \mathbf{v}_p^2 = \mathbf{v}_{\boldsymbol{o_2}}^0$$

or $$\mathbf{v}_p^0 = \mathbf{v}_{\boldsymbol{o_2}}^0 + \mathbf{v}_p^2$$

How do we deal with rotated frames?

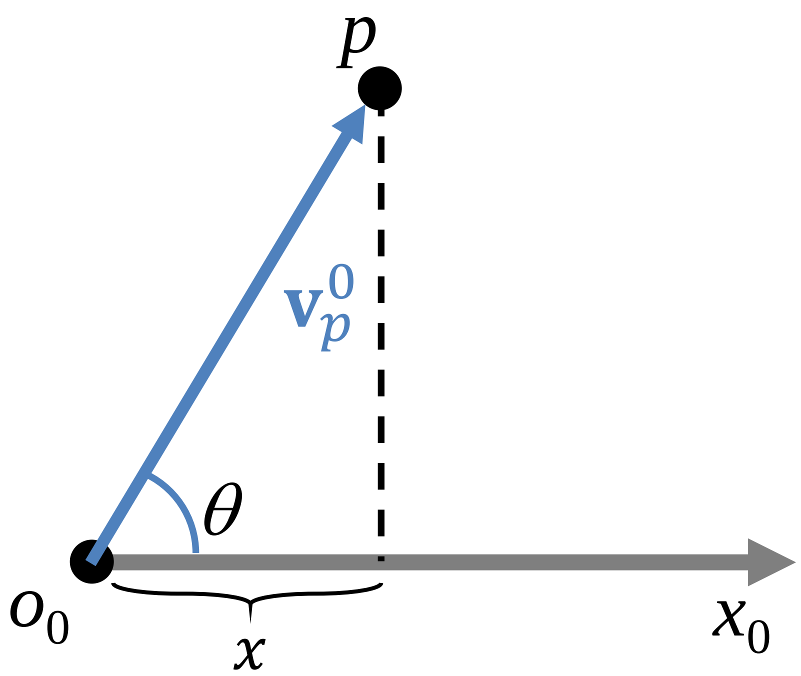


$$\mathbf{v}_p^0 = f\left(\mathbf{v}_p^1\right)$$

$$\mathbf{v}_p^0 = \mathbf{v}_{\boldsymbol{o_2}}^0 + \mathbf{v}_p^2$$

$$\mathbf{v}_p^0 = \mathbf{v}_{\boldsymbol{o_2}}^0 + \mathrm{Rotate}\left(\mathbf{v}_p^1\right)$$

Dot Products



$$
\begin{aligned}
\mathbf{v}_p^0 \cdot \hat{x}_0 &= \left\|\mathbf{v}_p^0\right\| \|\hat{x}_0\| \cos\theta \\
&= \left\|\mathbf{v}_p^0\right\| (1) \cos\theta \\
&= x
\end{aligned}
$$

$$\mathbf{v}_p^0 \cdot \hat{y}_0 = y$$

$$\mathbf{v}_p^0 \cdot \hat{z}_0 = z$$

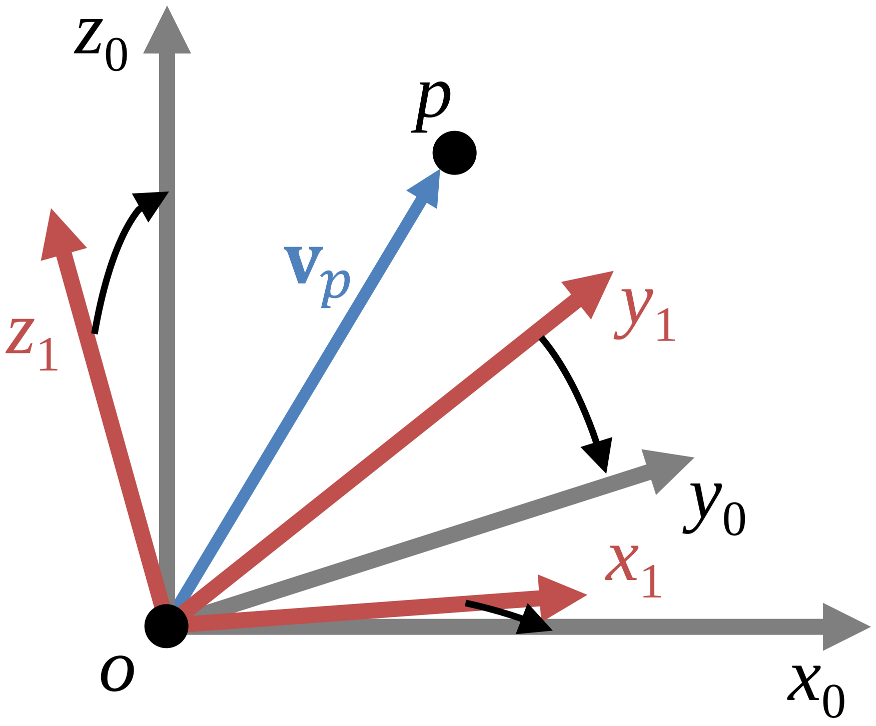
Rotation Matrices

$$\mathbf{v}_p^0 \cdot \hat{x}_0 = x \qquad \mathbf{v}_p^0 \cdot \hat{y}_0 = y \qquad \mathbf{v}_p^0 \cdot \hat{z}_0 = z$$

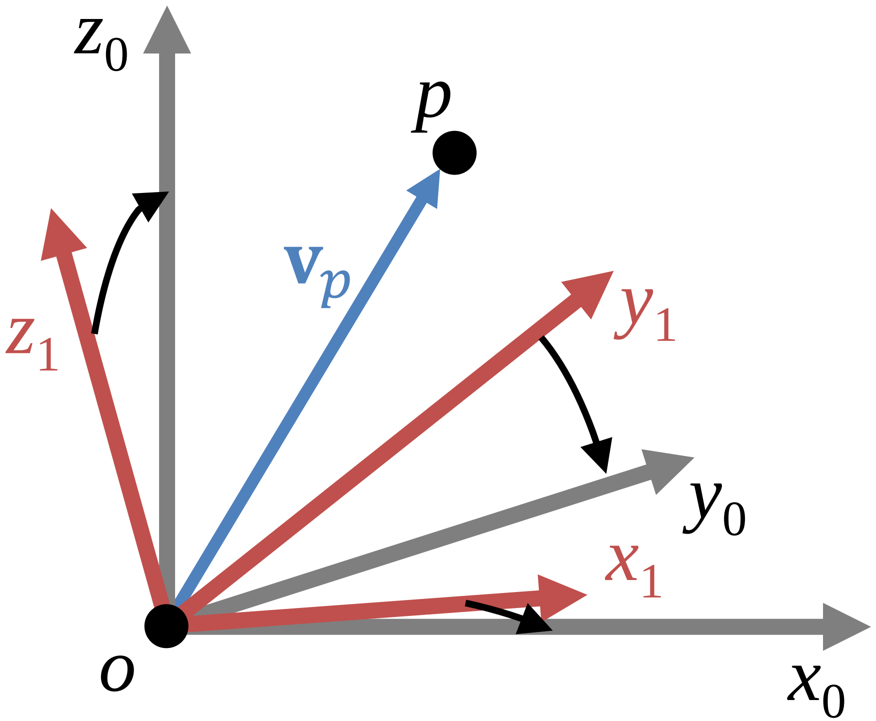
$$\mathbf{v}_p^1 = x_1\hat{x}_1 + y_1\hat{y}_1 + z_1\hat{z}_1$$

$$\mathbf{v}_p^0 = x_1\hat{x}_1^0 + y_1\hat{y}_1^0 + z_1\hat{z}_1^0$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (x_1\hat{x}_1^0 + y_1\hat{y}_1^0 + z_1\hat{z}_1^0) \cdot \hat{x}_0 \\ (x_1\hat{x}_1^0 + y_1\hat{y}_1^0 + z_1\hat{z}_1^0) \cdot \hat{y}_0 \\ (x_1\hat{x}_1^0 + y_1\hat{y}_1^0 + z_1\hat{z}_1^0) \cdot \hat{z}_0 \end{bmatrix}$$



Rotation Matrices



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (x_1\hat{x}_1^0 + y_1\hat{y}_1^0 + z_1\hat{z}_1^0) \cdot \hat{x}_0 \\ (x_1\hat{x}_1^0 + y_1\hat{y}_1^0 + z_1\hat{z}_1^0) \cdot \hat{y}_0 \\ (x_1\hat{x}_1^0 + y_1\hat{y}_1^0 + z_1\hat{z}_1^0) \cdot \hat{z}_0 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\mathbf{v}_p^0} = \underbrace{\begin{bmatrix} \hat{x}_1^0 \cdot \hat{x}_0 & \hat{y}_1^0 \cdot \hat{x}_0 & \hat{z}_1^0 \cdot \hat{x}_0 \\ \hat{x}_1^0 \cdot \hat{y}_0 & \hat{y}_1^0 \cdot \hat{y}_0 & \hat{z}_1^0 \cdot \hat{y}_0 \\ \hat{x}_1^0 \cdot z_0 & \hat{y}_1^0 \cdot \hat{z}_0 & \hat{z}_1^0 \cdot \hat{z}_0 \end{bmatrix}}_{\mathbf{R}_1^0} \underbrace{\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}}_{\mathbf{v}_p^1}$$

rotation from frame 1 to frame 0

subscript and superscript “cancel”

Properties of Rotation Matrices

$$\mathbf{R}_1^0 = \begin{bmatrix} \hat{x}_1^0 \cdot \hat{x}_0 & \hat{y}_1^0 \cdot \hat{x}_0 & \hat{z}_1^0 \cdot \hat{x}_0 \\ \hat{x}_1^0 \cdot \hat{y}_0 & \hat{y}_1^0 \cdot \hat{y}_0 & \hat{z}_1^0 \cdot \hat{y}_0 \\ \hat{x}_1^0 \cdot z_0 & \hat{y}_1^0 \cdot \hat{z}_0 & \hat{z}_1^0 \cdot \hat{z}_0 \end{bmatrix}$$

The rows show you the three unit vectors of the base frame expressed in the rotated frame.

$$\mathbf{R}_1^0 = \begin{bmatrix} \hat{x}_0^1 \\ \hat{y}_0^1 \\ \hat{z}_0^1 \end{bmatrix}$$

The columns show you the three unit vectors of the rotated frame expressed in the base frame.

$$\mathbf{R}_1^0 = \begin{bmatrix} \hat{x}_1^0 & \hat{y}_1^0 & \hat{z}_1^0 \end{bmatrix}$$

Properties of Rotation Matrices

$$\mathbf{R}_1^0 = \begin{bmatrix} \hat{x}_1^0 \cdot \hat{x}_0 & \hat{y}_1^0 \cdot \hat{x}_0 & \hat{z}_1^0 \cdot \hat{x}_0 \\ \hat{x}_1^0 \cdot \hat{y}_0 & \hat{y}_1^0 \cdot \hat{y}_0 & \hat{z}_1^0 \cdot \hat{y}_0 \\ \hat{x}_1^0 \cdot z_0 & \hat{y}_1^0 \cdot \hat{z}_0 & \hat{z}_1^0 \cdot \hat{z}_0 \end{bmatrix}$$

- Every row and column is a unit vector.
- The columns (and rows) are orthogonal.
- $(\mathbf{R}_1^0)^\top \mathbf{R}_1^0 = \mathbf{I} \implies \mathbf{R}_0^1 = (\mathbf{R}_1^0)^{-1} = (\mathbf{R}_1^0)^\top$
- If you are transforming between 2 right-handed coordinate frames, then $\det \mathbf{R}_1^0 = +1$

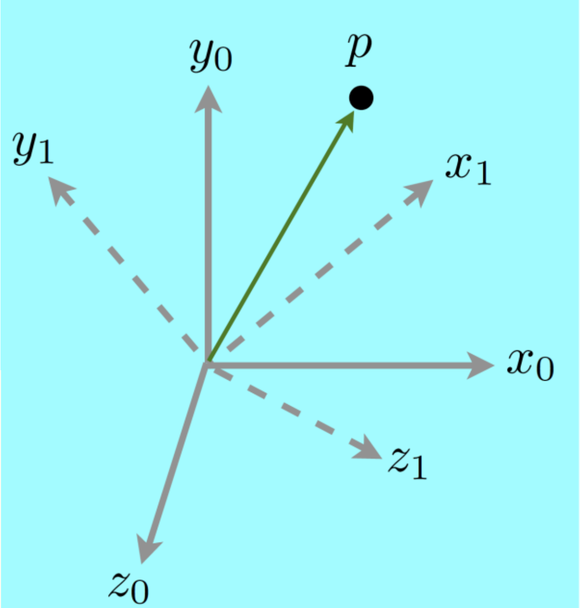
  We call these matrices SO(3) (“special orthogonal group of order 3”)

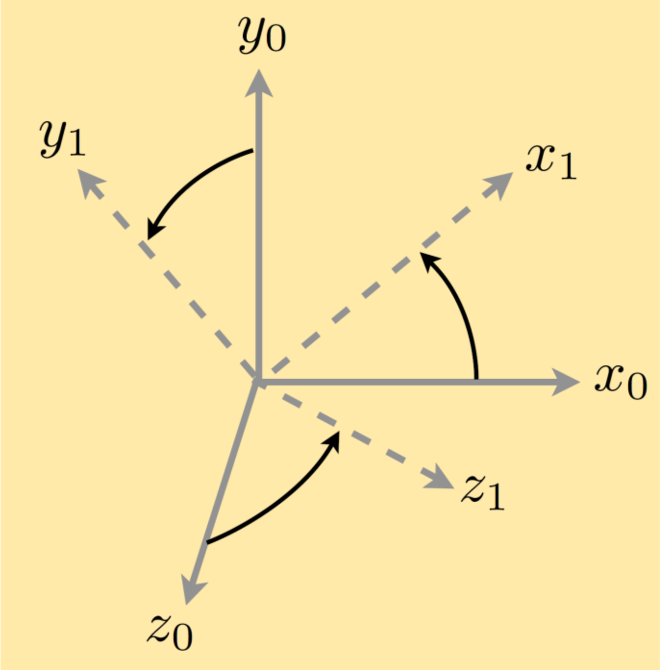
Rotation Matrices

Serve 3 purposes (p. 47 of SHV):

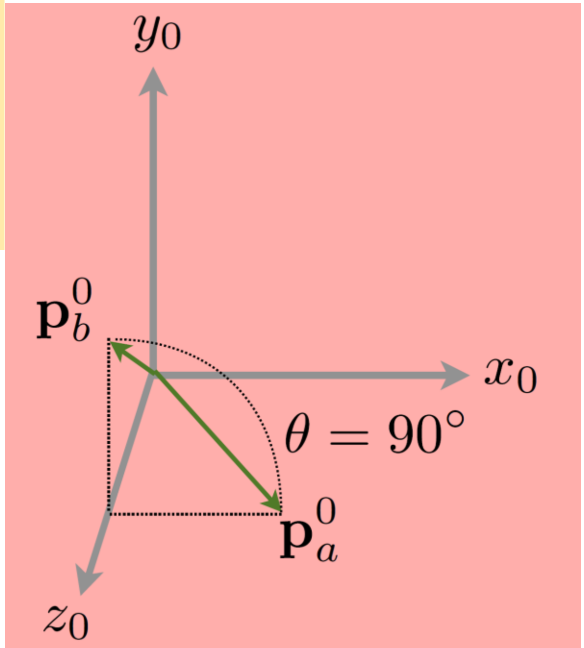
1. Coordinate transformations relating coordinates of a point p in two different frames
2. Orientation of a transformed coordinate frame with respect to a fixed frame
3. Operator taking a vector and rotating it to yield a new vector in the same coordinate frame

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





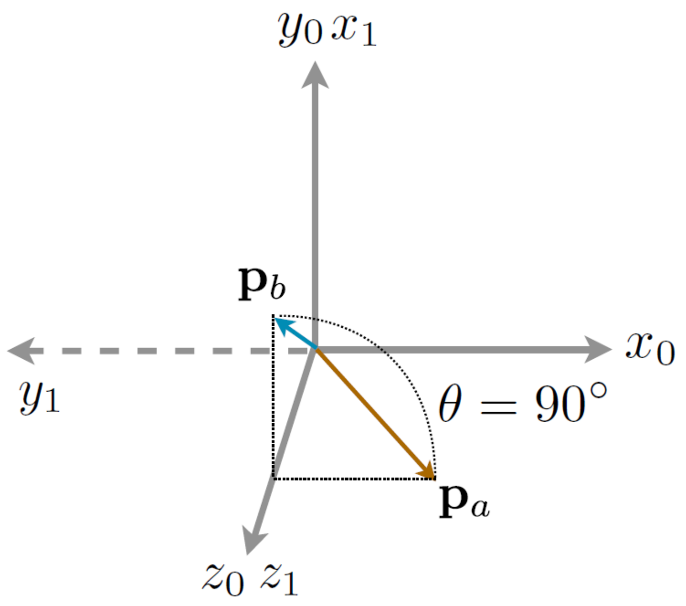
$$\mathbf{R}_1^0 = [\hat{x}_1^0 \quad \hat{y}_1^0 \quad \hat{z}_1^0]$$



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

Composite Rotations

What if I want to apply multiple rotations to a vector?



The **order** in which a sequence of rotations is performed is crucial.

Thus, the **order** in which rotation matrices are multiplied together is crucial

For example:

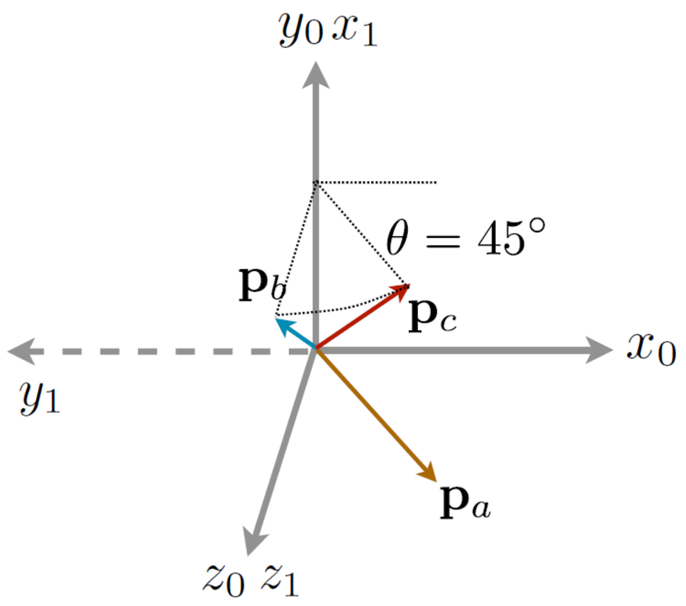
Rotate 45° around y_0

vs

Rotate 45° around y_1

Composite Rotations

What if I want to apply multiple rotations to a vector?



$$\mathbf{p}_b = \mathbf{R}\mathbf{p}_a \qquad \mathbf{R}' = \mathbf{R}_{y,45°}$$

$$\mathbf{p}_c = ?$$

$$\mathbf{p}_c = \mathbf{R}'\mathbf{p}_b$$

$$\mathbf{p}_c = \mathbf{R}'\mathbf{R}\mathbf{p}_a$$

For example:

Rotate 45° around y_0

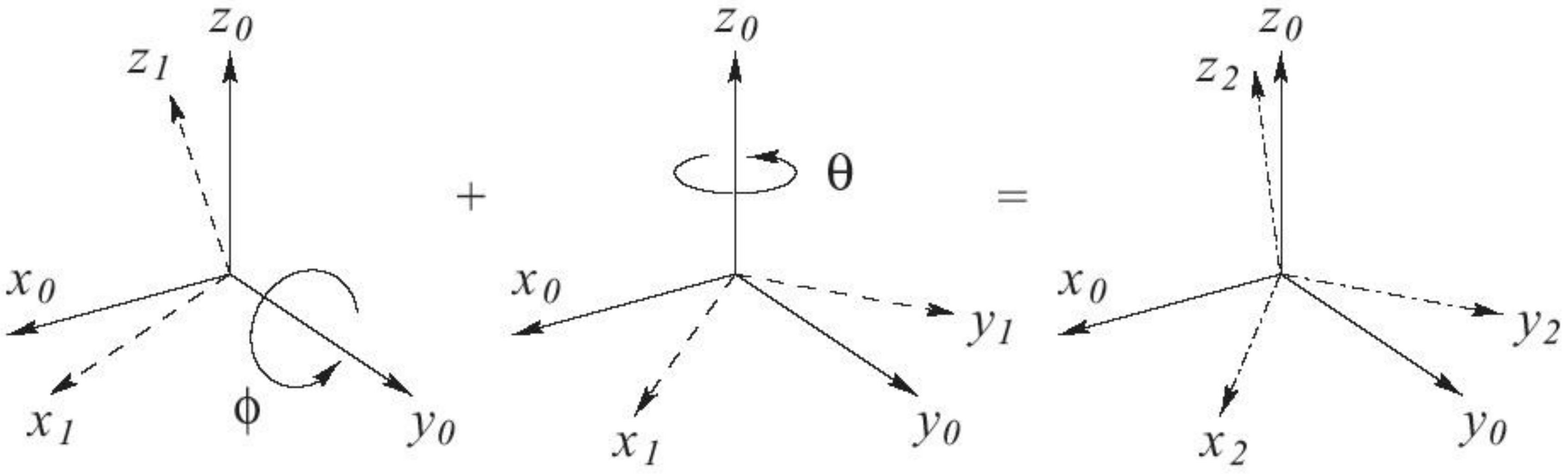
vs

Rotate 45° around y_1

Compositions of Rotations with Respect to a Fixed Frame

the result of a successive rotation about a fixed frame can be found by **pre-multiplying** by the corresponding rotation matrix

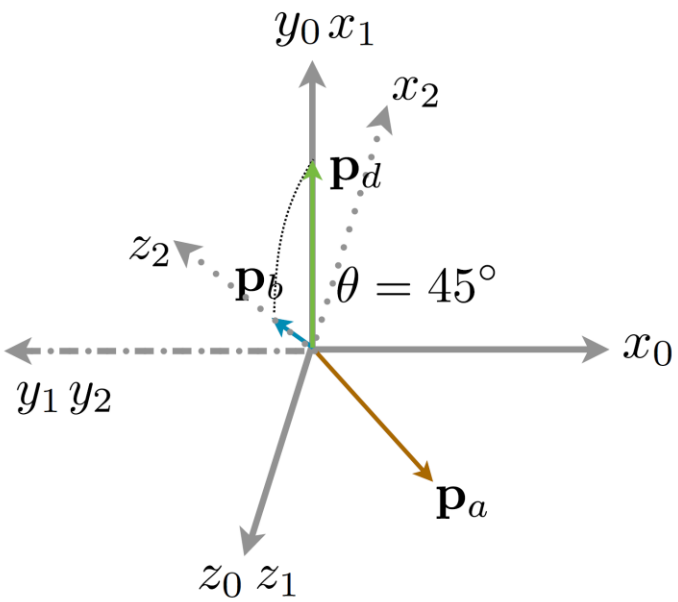
$$\mathbf{R}_2^0 = \mathbf{R}\mathbf{R}_1^0$$



Note that $\mathbf{R}$ is a rotation about the original frame

Composite Rotations

What if I want to apply multiple rotations to a vector?



For example:
Rotate 45° around y_0
vs
Rotate 45° around y_1

$$\mathbf{p}_b = \mathbf{R}\mathbf{p}_a \qquad \mathbf{R}' = \mathbf{R}_{y,45°}$$

$$\mathbf{p}_d = ?$$

$$\mathbf{p}_d^2 = \mathbf{p}_a^0$$

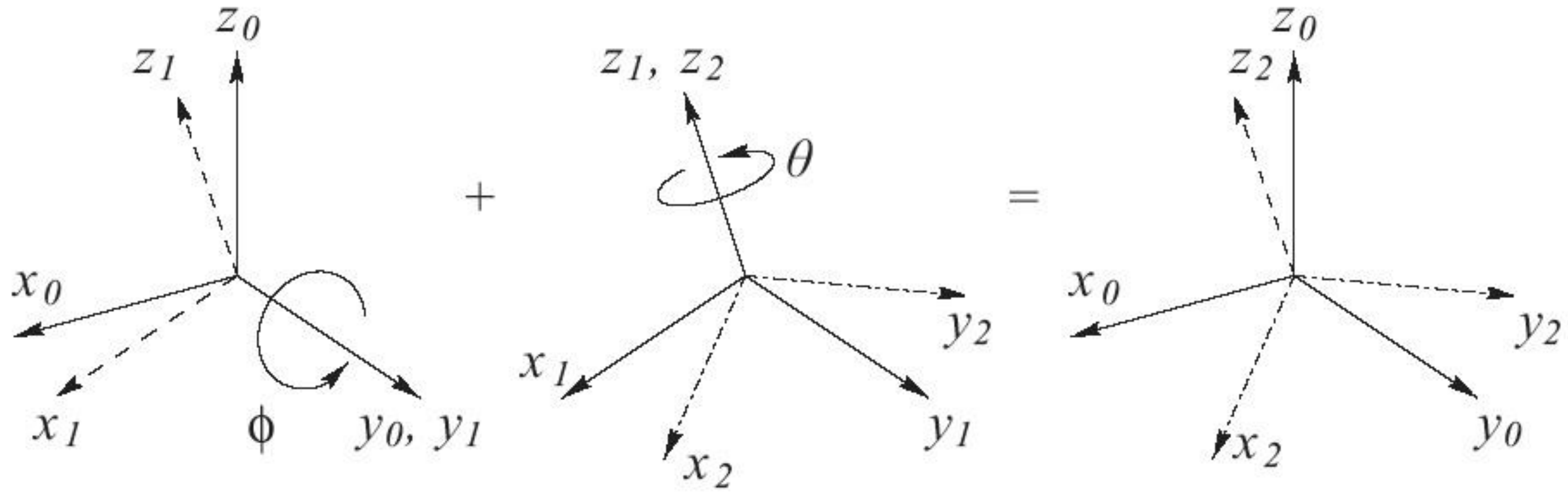
$$\mathbf{p}_d^0 = \mathbf{R}_1^0\mathbf{R}_2^1\mathbf{p}_d^2$$

$$\mathbf{p}_d = \mathbf{R}\mathbf{R}'\mathbf{p}_a$$

Compositions of Rotations with Respect to an Intermediate Frame

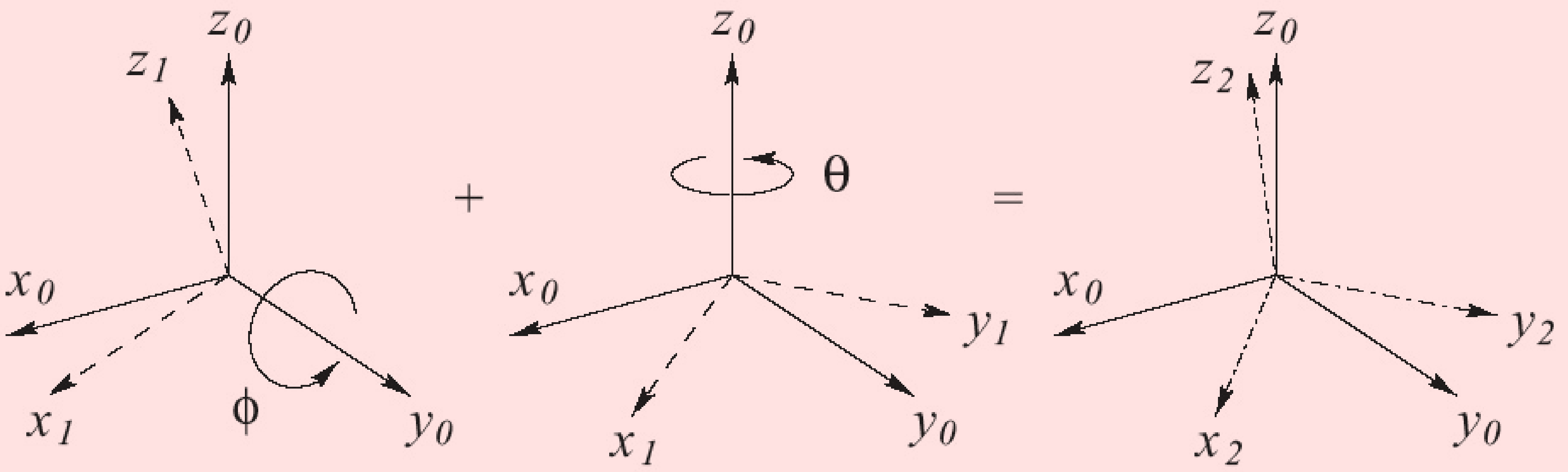
the result of a successive rotation about the current (intermediate) frame can be found by **post-multiplying** by the corresponding rotation matrix

$$\mathbf{R}_2^0 = \mathbf{R}_1^0 \mathbf{R}_2^1$$

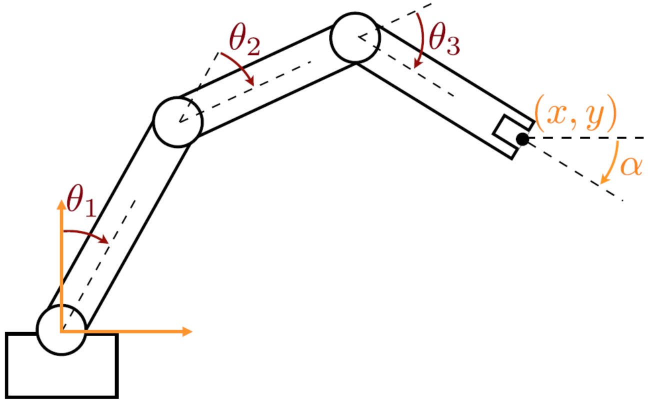


Rotation about a fixed frame? **pre-multiply**

$$\mathbf{R}_2^0 = \mathbf{R}\mathbf{R}_1^0$$

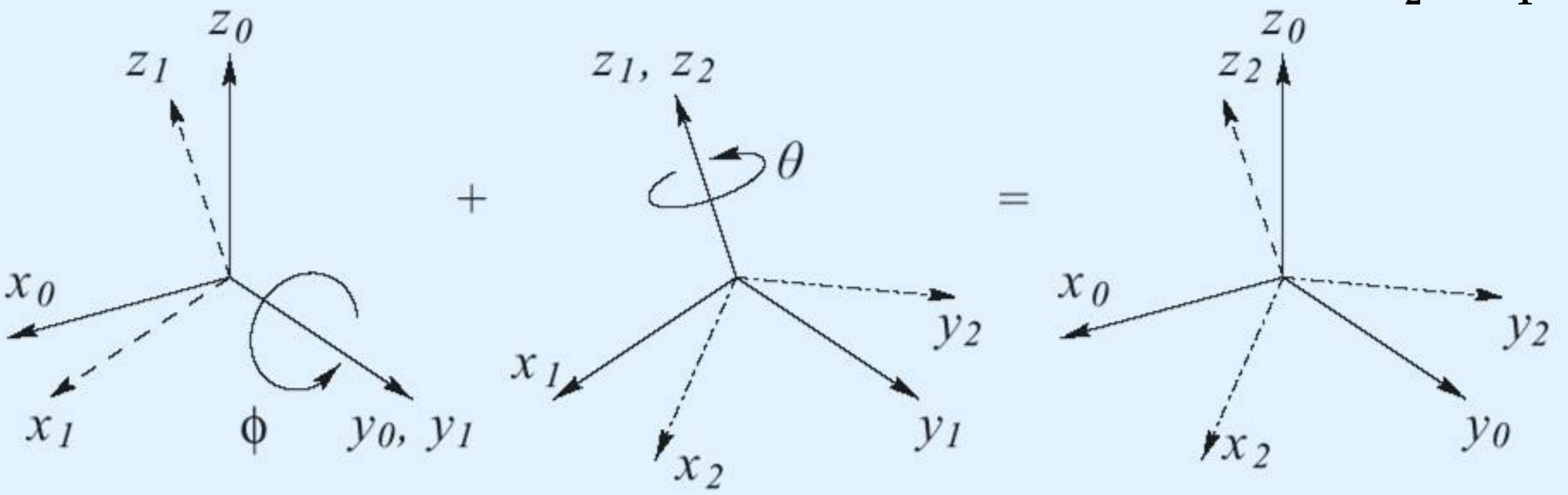


Which of these is more commonly used in robotics?

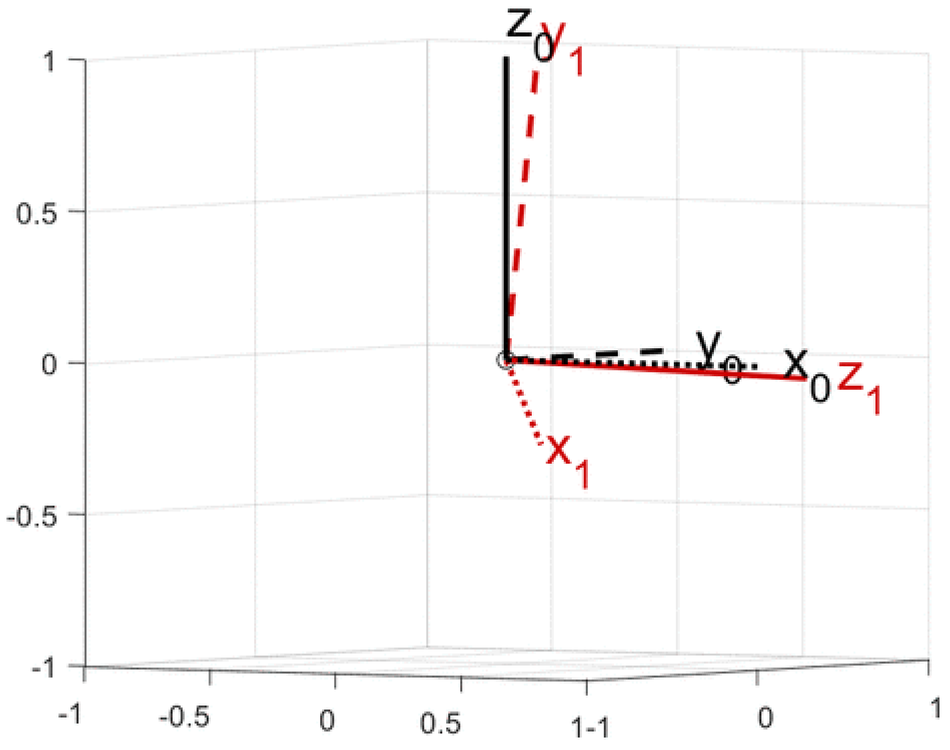


Rotation about intermediate frame? **post-multiply**

$$\mathbf{R}_2^0 = \mathbf{R}_1^0\mathbf{R}_2^1$$



Practice: Choose the $\mathbf{R}_1^0$ matrix corresponding to the following visual representation.



$$\mathbf{R}_1^0 = \begin{bmatrix} 0.028 & 0.538 & -0.843 \\ 0.899 & 0.355 & 0.256 \\ 0.437 & -0.765 & -0.474 \end{bmatrix}$$

$$\mathbf{R}_1^0 = \begin{bmatrix} -0.434 & -0.092 & 0.895 \\ 0.842 & 0.310 & 0.443 \\ -0.318 & 0.946 & -0.057 \end{bmatrix}$$

$$\mathbf{R}_1^0 = \begin{bmatrix} -0.453 & -0.591 & -0.668 \\ 0.298 & -0.806 & 0.511 \\ -0.840 & 0.033 & 0.541 \end{bmatrix}$$

Further practice: MATLAB scripts visualizeR.m and testR.m posted on Canvas under Files > Resources

Parameterizing Rotations

Can also figure out 4th value using the remaining 3

$$\mathbf{R}_1^0 = \begin{bmatrix} \hat{x}_1 \cdot \hat{x}_0 & \boxed{\hat{y}_1 \cdot \hat{x}_0} & \hat{z}_1 \cdot \hat{x}_0 \\ \hat{x}_1 \cdot \hat{y}_0 & \hat{y}_1 \cdot \hat{y}_0 & \hat{z}_1 \cdot \hat{y}_0 \\ \hat{x}_1 \cdot z_0 & \hat{y}_1 \cdot \hat{z}_0 & \hat{z}_1 \cdot \hat{z}_0 \end{bmatrix}$$

Figure out 1 column from the others using RHR

Figure out 1 row using normality

In 3 dimensions, no more than **3 independent values** are needed to specify an arbitrary rotation.

The 9-element rotation matrix has **6 redundancies** (3 DOF)

Numerous methods have been developed to represent rotation/orientation more compactly

Euler Angles   Roll/Pitch/Yaw   Axis/Angle

Conventions vary, so always check definitions!

Three Special Rotation Matrices

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

The 0s and 1s are always in the row and column of the rotation axis

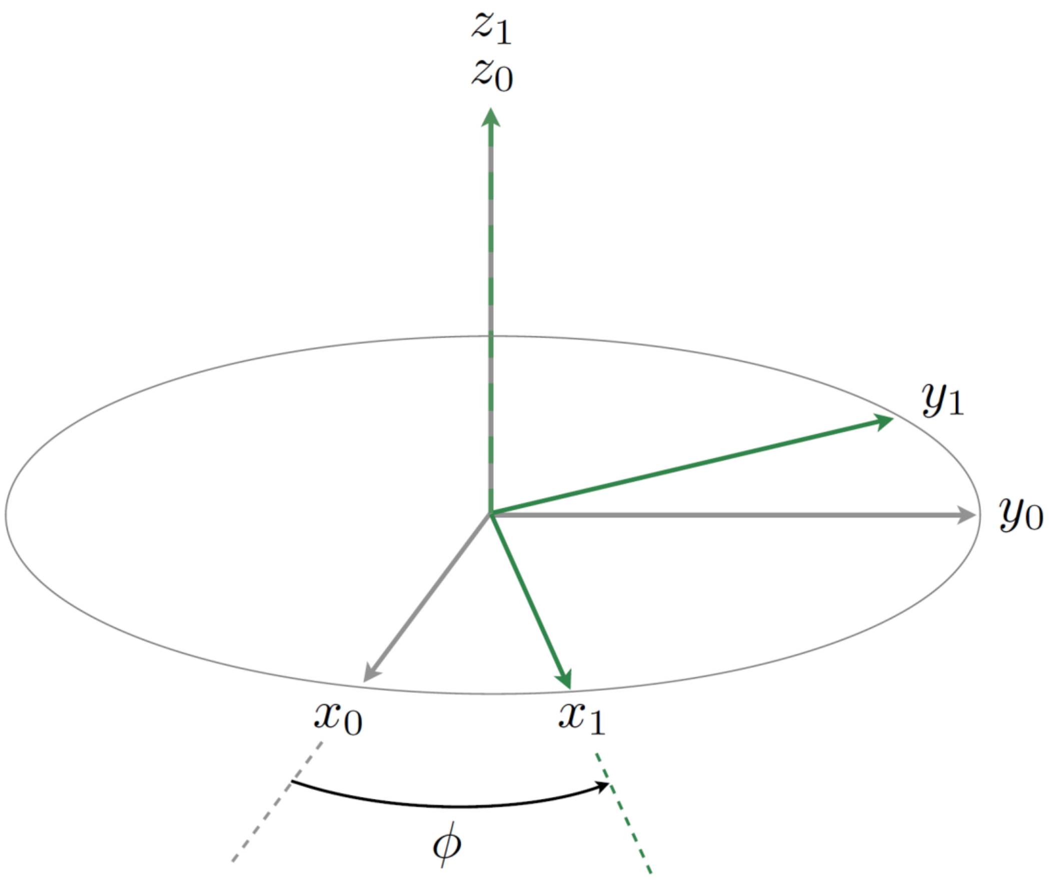
$$\mathbf{R}_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

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

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

Euler Angles

Define a set of 3 angles ϕ, θ, ψ to go from 0 → 3 by rotating around the axes of the **current frame**.

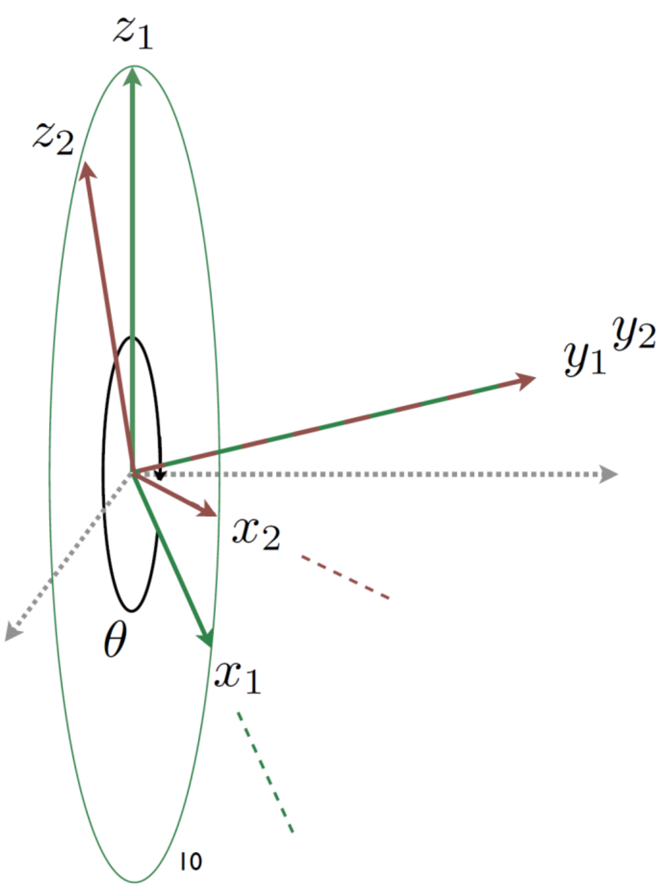


Using Z-Y-Z convention:

1. Rotate by ϕ about z_0

Euler Angles

Define a set of 3 angles ϕ, θ, ψ to go from 0 → 3
by rotating around the axes of the **current frame**.

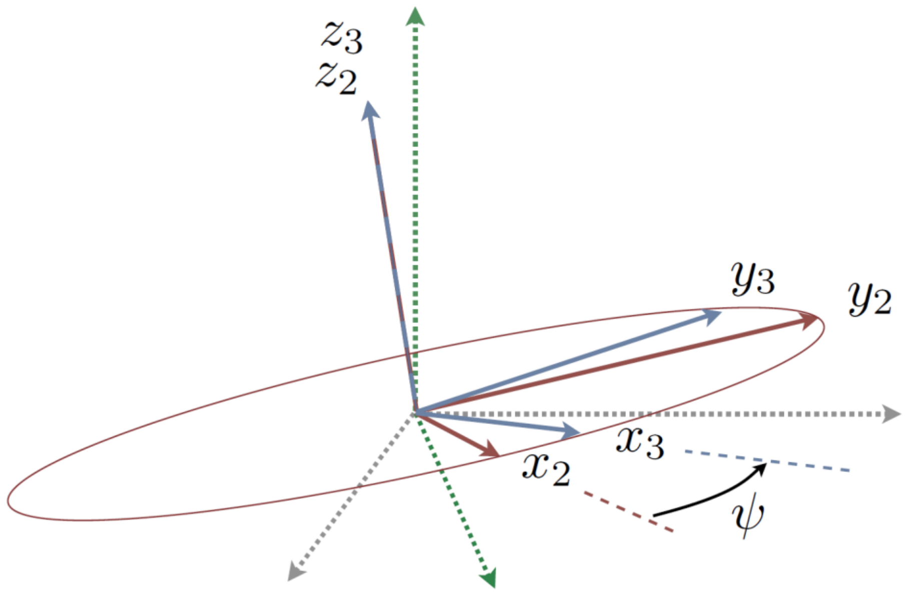


Using Z-Y-Z convention:

1. Rotate by ϕ about z_0
2. Rotate by θ about y_1

Euler Angles

Define a set of 3 angles ϕ, θ, ψ to go from 0 → 3 by rotating around the axes of the **current frame**.



Using Z-Y-Z convention:

1. Rotate by ϕ about z_0
2. Rotate by θ about y_1
3. Rotate by ψ about z_2

Q: Should we **pre-** or **post-**multiply?

Euler Angles to Rotation Matrices

Post-multiply using the **basic rotation matrices**

$$\mathbf{R} = \mathbf{R}_{z,\phi}\mathbf{R}_{y,\theta}\,\mathbf{R}_{z,\psi}$$

$$s_\theta = sin\,\theta\,, c_\theta = \cos\theta$$

$$\mathbf{R} = \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}\begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Rotation Matrices to Euler Angles

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

Plug in to solve for ϕ

Plug in to solve for ψ

Solve for θ

NOTE: Two solutions for θ because sign of s_θ is not known.

In general, you will end up with two sets of valid ϕ, θ, ψ values.

Example: Rotation Matrices to Euler Angles

$$\mathbf{R} = \begin{bmatrix} 0.090 & -0.785 & 0.612 \\ 0.574 & 0.544 & 0.612 \\ -0.814 & 0.296 & 0.5 \end{bmatrix}$$

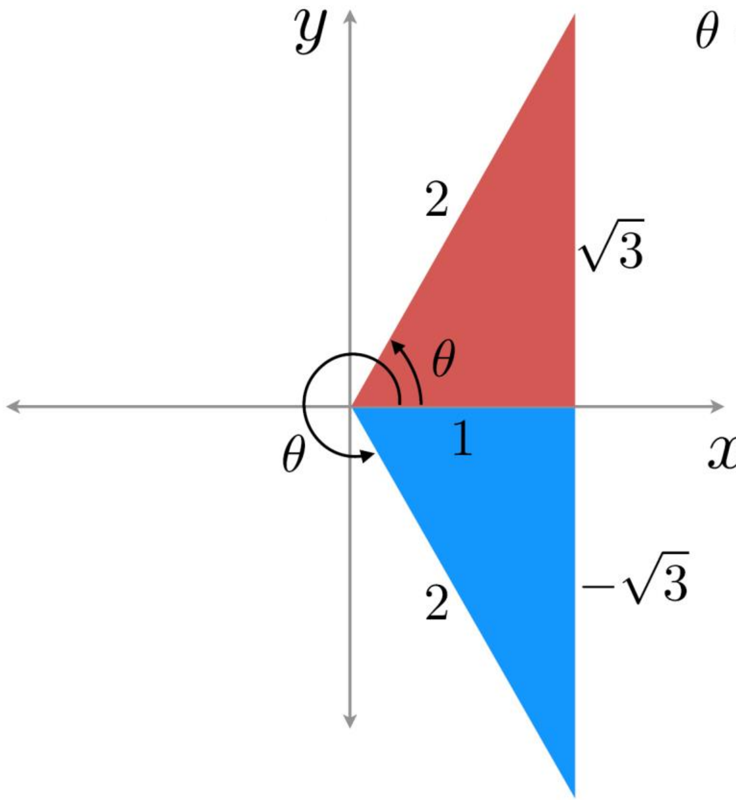
Check: Is the matrix orthonormal? ✓

Is the determinant +1? ✓

Example: Rotation Matrices to Euler Angles

$$\mathbf{R} = \begin{bmatrix} 0.090 & -0.785 & 0.612 \\ 0.574 & 0.544 & 0.612 \\ -0.814 & 0.296 & 0.5 \end{bmatrix}$$

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



$\theta = \cos^{-1}(0.500)$

Example: Rotation Matrices to Euler Angles

Side note:
Book uses atan2(x, y)
MATLAB uses atan2(y, x)
We'll use atan2(y/x)

$$\mathbf{R} = \begin{bmatrix} 0.090 & -0.785 & 0.612 \\ 0.574 & 0.544 & 0.612 \\ -0.814 & 0.296 & 0.5 \end{bmatrix}$$

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

$$\theta = \frac{\pi}{3}$$

$$\sin\theta = 0.866 \Longrightarrow \begin{cases} \cos\psi = 0.940 \\ \sin\psi = 0.342 \end{cases}$$

$$\boldsymbol{\psi} = \mathrm{atan2}\left(\frac{\sin\psi}{\cos\psi}\right)$$

$$\psi = \mathrm{atan2}\left(\frac{0.342}{0.940}\right) = \frac{\pi}{9}$$

$$\theta = \frac{5\pi}{3}$$

$$\sin\theta = -0.866 \Longrightarrow \begin{cases} \cos\psi = -0.940 \\ \sin\psi = -0.342 \end{cases}$$

$$\boldsymbol{\psi} = \mathrm{atan2}\left(\frac{\sin\psi}{\cos\psi}\right)$$

$$\psi = \mathrm{atan2}\left(\frac{-0.342}{-0.940}\right) = -\frac{8\pi}{9}$$

Example: Rotation Matrices to Euler Angles

$$\mathbf{R} = \begin{bmatrix} 0.090 & -0.785 & 0.612 \\ 0.574 & 0.544 & 0.612 \\ -0.814 & 0.296 & 0.5 \end{bmatrix}$$

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

$$\theta = \frac{\pi}{3}$$

$$\sin\theta = 0.866 \Longrightarrow \begin{cases} \cos\phi = 0.707 \\ \sin\phi = 0.707 \end{cases}$$

$$\phi = \operatorname{atan2}\left(\frac{\sin\phi}{\cos\phi}\right)$$

$$\phi = \operatorname{atan2}\left(\frac{0.707}{0.707}\right) = \frac{\pi}{4}$$

$$\theta = \frac{5\pi}{3}$$

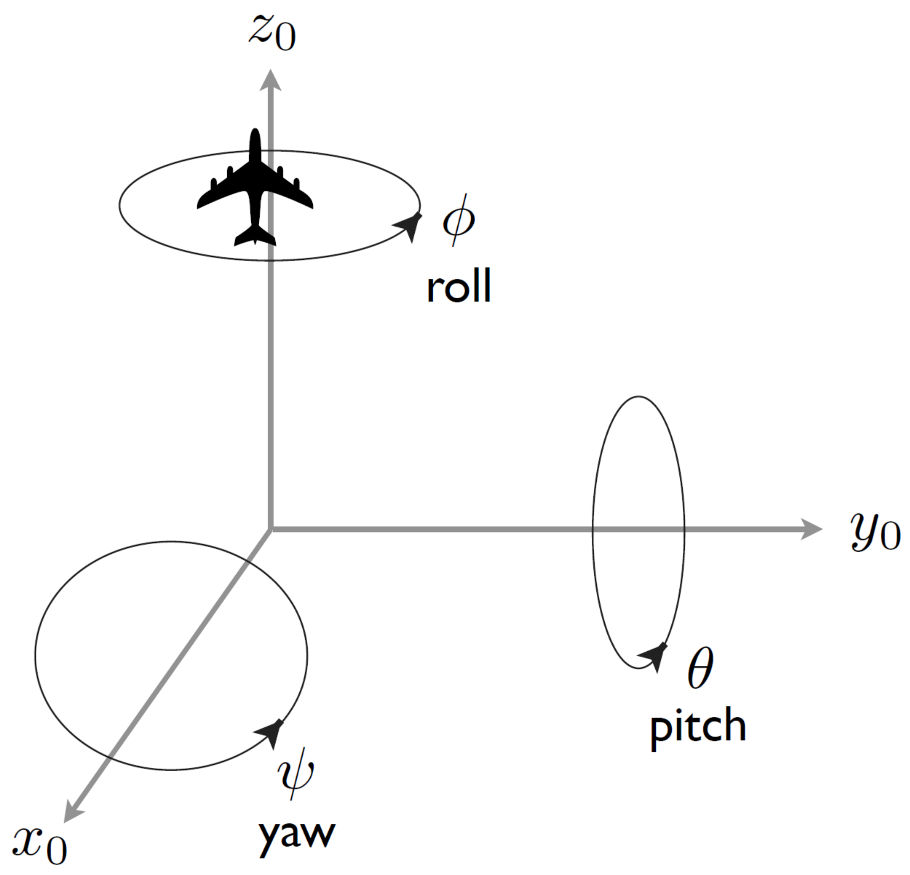
$$\sin\theta = -0.866 \Longrightarrow \begin{cases} \cos\phi = -0.707 \\ \sin\phi = -0.707 \end{cases}$$

$$\phi = \operatorname{atan2}\left(\frac{\sin\phi}{\cos\phi}\right)$$

$$\phi = \operatorname{atan2}\left(\frac{-0.707}{-0.707}\right) = -\frac{3\pi}{4}$$

Yaw, Pitch, Roll Angles

Define a set of 3 angles ϕ, θ, ψ to go from 0 → 3 by rotating around **fixed axes**.



Our book uses X-Y-Z convention.

Think about a plane flying in the z direction. Yaw is left/right, Pitch is up/down, and roll is rotating about z.

Q: Should we **pre-** or **post-**multiply?

Yaw, Pitch, Roll Angles to Rotation Matrices

Pre-multiply using the **basic rotation matrices**

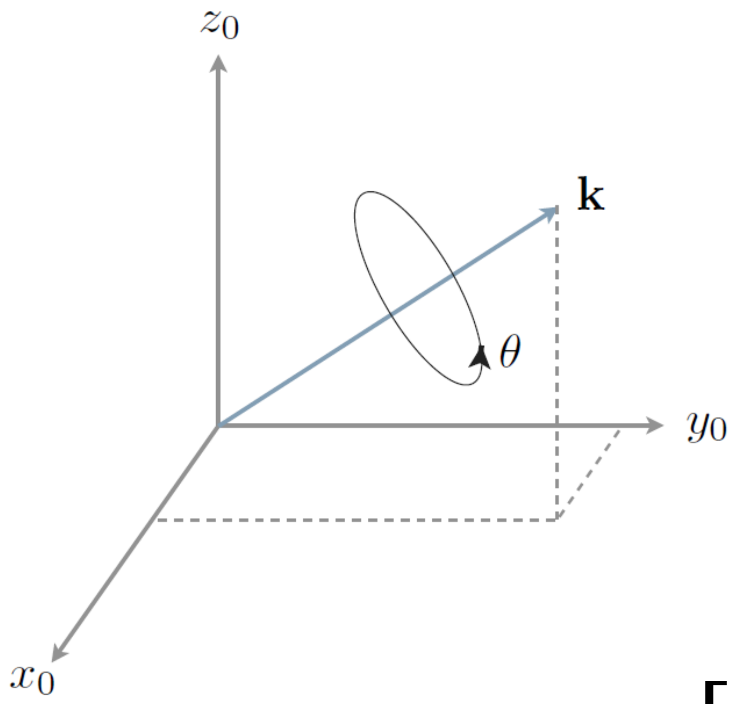
$$\mathbf{R} = \mathbf{R}_{z,\phi}\mathbf{R}_{y,\theta}\,\mathbf{R}_{x,\psi}$$

$$\mathbf{R} = \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix}$$

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

Angle/Axis Representation

Rotation by an angle about an axis in space



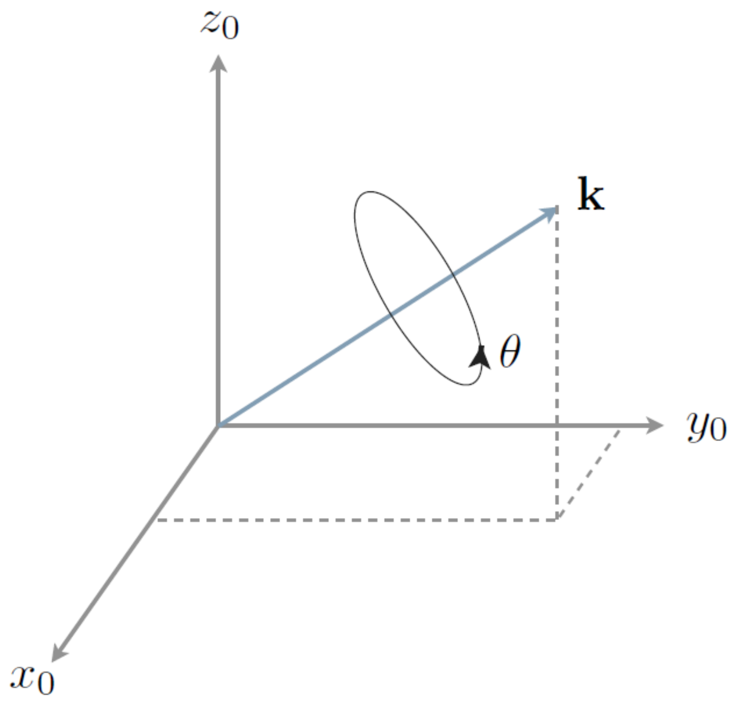
$$\mathbf{k} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} \text{ with } \|\mathbf{k}\| = 1$$

Let $v_\theta = \operatorname{vers}\theta = 1 - c_\theta$

$$\mathbf{R}_{k,\theta} = \begin{bmatrix} k_x^2 v_\theta + c_\theta & k_x k_y v_\theta - k_z s_\theta & k_x k_z v_\theta + k_y s_\theta \\ k_x k_y v_\theta + k_z s_\theta & k_y^2 v_\theta + c_\theta & k_y k_z v_\theta - k_x s_\theta \\ k_x k_z v_\theta - k_y s_\theta & k_y k_z v_\theta + k_x s_\theta & k_z^2 v_\theta + c_\theta \end{bmatrix}$$

Angle/Axis Representation

Any rotation matrix can be represented this way!



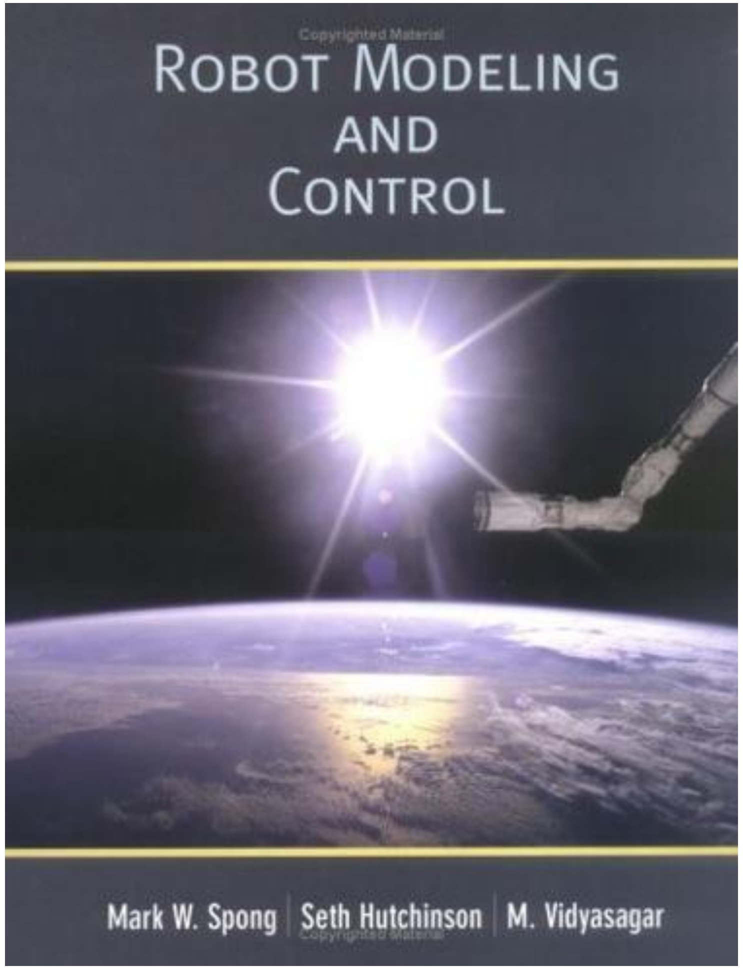
$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\mathbf{k} = \frac{1}{2s_\theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

$$\theta = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right)$$

Why?

Next time: Homogeneous Transformations



Chapter 2: Rigid Motions

- Read Sec. 2.6 – 2.8