Introduction to Robotics

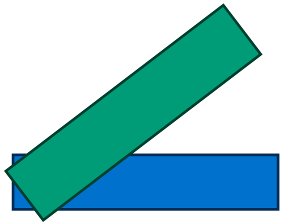
Kinematic Chains

Johns Hopkins
Whiting School of Engineering

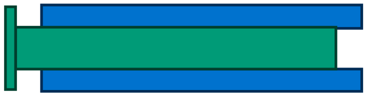
Forward Kinematics

Robot Manipulators

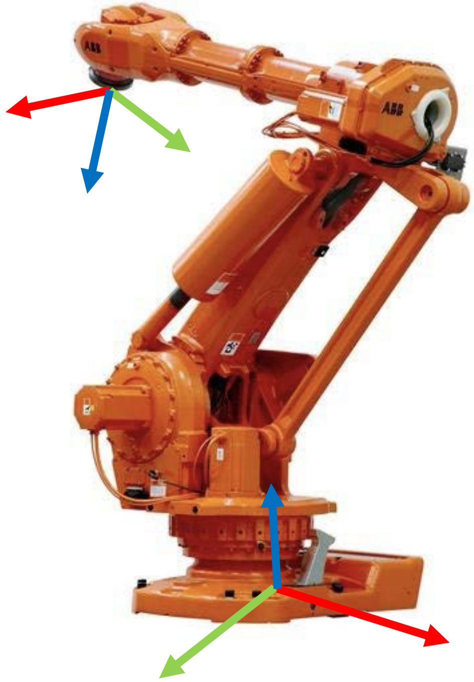
- A robot manipulator is typically moved through its joints
- Revolute: Rotate about an Axis
- Prismatic: Translate along an axis



Revolute



Prismatic



6 axes robot arm

Kinematics (1)

Linear Algebra

Rigid body motion
Transformation between coordinate frames

Forward Kinematics

Joint Space

Joint 1 = q1
Joint 2 = q2
…
Joint N = qn

$$f(q) = \left[R_t^B, t_T^B\right]$$

$$\left[q = f^{-1}\left(\left[R_t^B, t_T^B\right]\right)\right.$$

Cartesian Space
$[\,R_T^B\,, t_T^B\,]$
Tool Frame (T)
Base Frame (B)

R_t^B : Orientation of T wrt B
t_T^B : Orientation of T wrt B

…
Joint N = qn

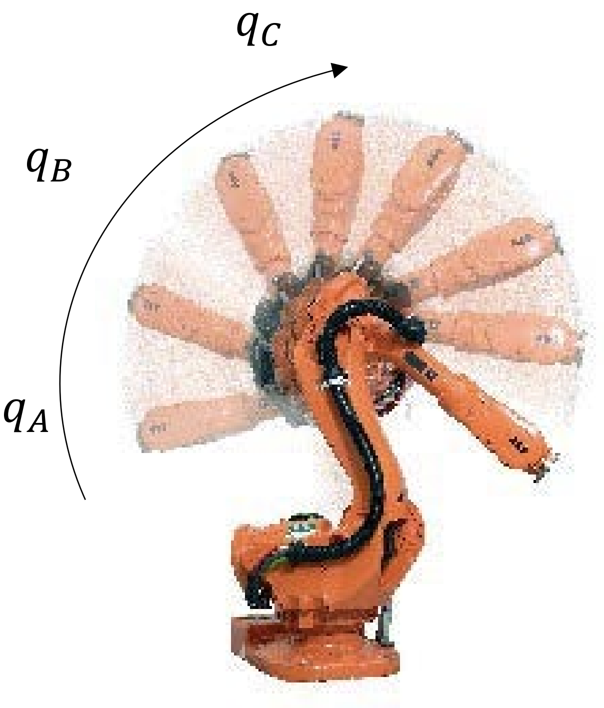
Inverse Kinematics

Joint Spaces

- Joint spaces are defined in $\mathbb{R}$
- Thus for a vector of joint values $q = \begin{bmatrix} q_1 \\ \cdots \\ q_N \end{bmatrix}$
- We can add/subtract joint values

$$q_c = q_A + q_B$$

- How many joints do you need? It depends on the task. But ISO 8373 requires all industrial robots to have at least three or more axes.

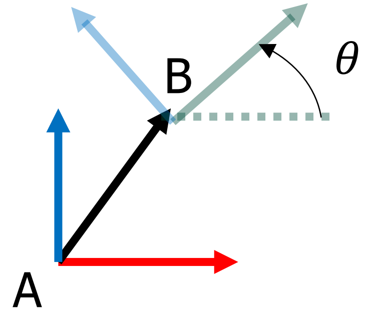


2D Rigid Motion

Special Orthogonal (SO)

- Combine position and orientation:
  - Special Euclidean Group: $SE(2)$

$$SE(2) = \left\{ (t, R) : t \in \mathbb{R}^2, R \in SO(2) \right\} = \mathbb{R}^2 \times SO(2)$$



$t_B^A \in \mathbb{R}^2$ is the translation between A and B

$R_B^A \in SO(2)$ is the rotation between A and B

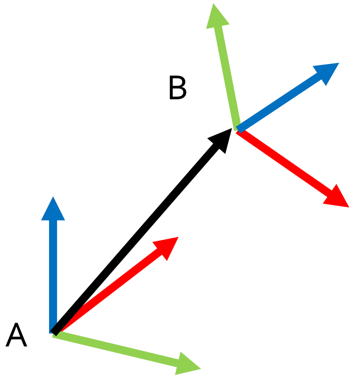
If $R \in SO(2)$ then $R \in \mathbb{R}^{2\times 2}, RR^T = I$ and $\det(R) = 1$

$$R_B^A = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

3D Rigid Motion

- Combine position and orientation:
  - Special Euclidean Group: $\mathrm{SE}(3)$

$$SE(3) = \left\{ (t, R) : t \in \mathbb{R}^3, R \in SO(3) \right\} = \mathbb{R}^3 \times SO(3)$$



$t_B^A \in \mathbb{R}^3$ is the translation between A and B

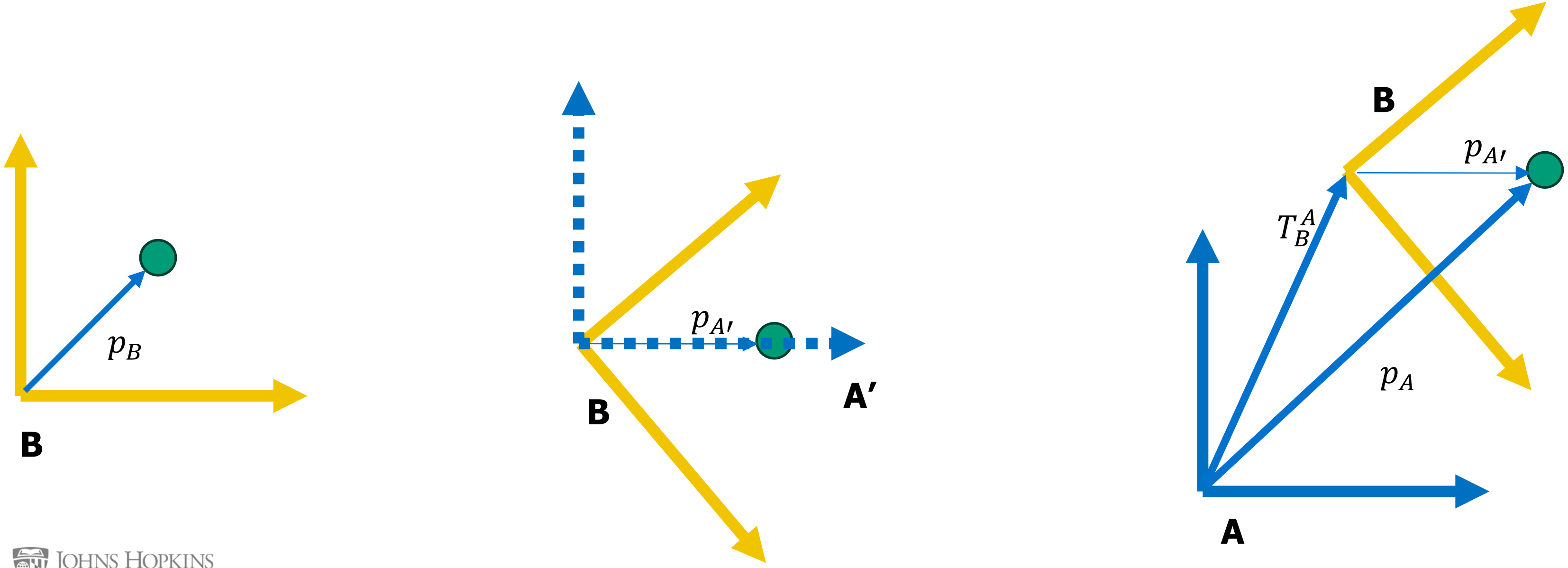
$R_B^A \in SO(3)$ is the rotation between A and B

If $R \in SO(3)$ then $R \in \mathbb{R}^{3\times 3}, RR^T = I$ and $\det(R) = 1$

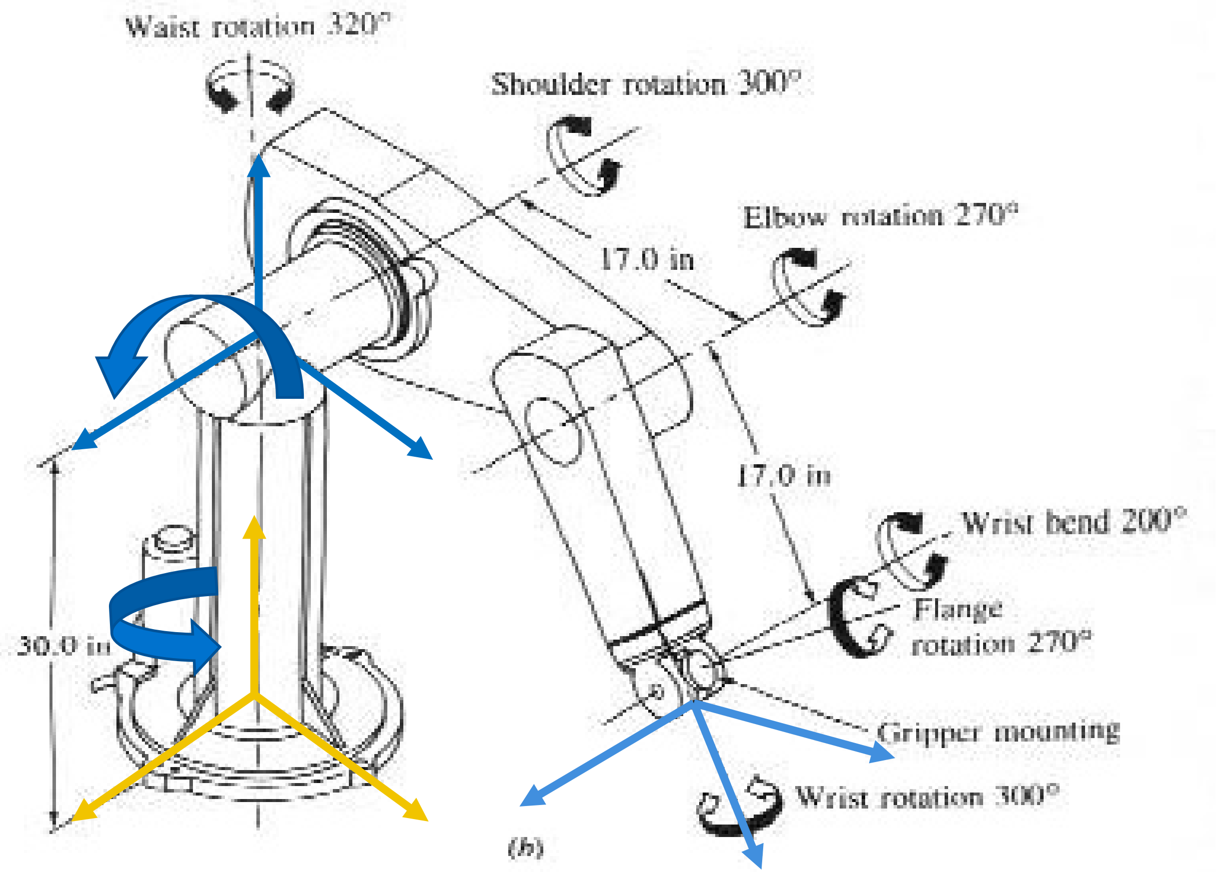
$$R_B^A = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Visual Representation

$$p_B \qquad p_{A'} = R_B^A p_B \qquad p_A = R_B^A p_B + T_B^A$$



Cartesian Transformation Kinematic Chain



Homogenous Representation

Rigid Body Motion in 2D

- We have the coordinates of a point in coordinate frame "C", P^C

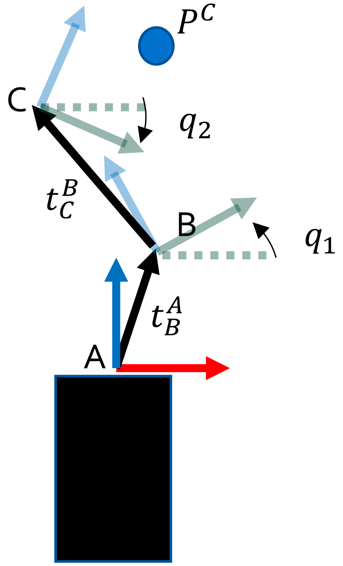
- Given the following robot what are the coordinates of P in the coordinate frame "A"? P^A

- Point Convention
  - Superscript – frame
  - Subscript – identifier

$$P_{id}^{frame}$$

- Transform conventions
  - Superscript - ending frame
  - Subscript - starting frame

$$t_{start}^{end}$$



Homogeneous Representation

- A 2D point is represented by appending a "1" to yield a vector in $\mathbb{R}^2, P = [x, y, 1]^T$
- A 3D point is represented by appending a "1" to yield a vector in $\mathbb{R}^3, P = [x, y, z, 1]^T$
- They are called homogenous coordinates
- The affine transformation of a point

$$P^A = R_B^A P^B + t_B^A$$

- Is represented by a linear transformation using a homogenous coordinates

$$\begin{bmatrix} P^A \\ 1 \end{bmatrix} = \begin{bmatrix} R_B^A & t_B^A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P^B \\ 1 \end{bmatrix}$$

Affine vs Linear Transforms

- Affine Transformations may seem more immediately intuitive but are more complicated to represent transformations between multiple coordinate systems.

Affine Transformations

$$P^A = R_B^A P^B + t_B^A$$

$$P^B = R_C^B P^C + t_C^B$$

$$P^A = R_B^A \left(R_C^B P^C + t_C^B \right) + t_B^A$$

Cumbersome representation

Linear Transformations

$$\begin{bmatrix} P^A \\ 1 \end{bmatrix} = \begin{bmatrix} R_B^A & t_B^A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P^B \\ 1 \end{bmatrix} = \mathrm{E}_{\mathrm{B}}^{\mathrm{A}} \begin{bmatrix} P^B \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} P^B \\ 1 \end{bmatrix} = \begin{bmatrix} R_C^B & t_C^B \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P^C \\ 1 \end{bmatrix} = \mathrm{E}_C^{\mathrm{B}} \begin{bmatrix} P^C \\ 1 \end{bmatrix}$$

$$P^A = E_B^A E_C^B \begin{bmatrix} P^C \\ 1 \end{bmatrix}$$

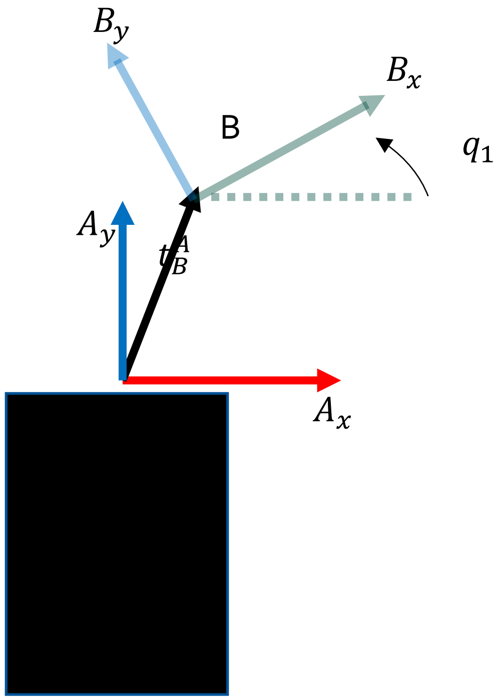
Compact representation

Homogenous Representation

- First what is the position and orientation of the coordinate frame "B" with respect to coordinate frame "A"
  - The position of B with respect to A is constant t_B^A
  - The orientation of B with respect to A depends on angle q_1

$$R_B^A = \begin{bmatrix} \cos(q_1) & -\sin(q_1) \\ \sin(q_1) & \cos(q_1) \end{bmatrix}$$

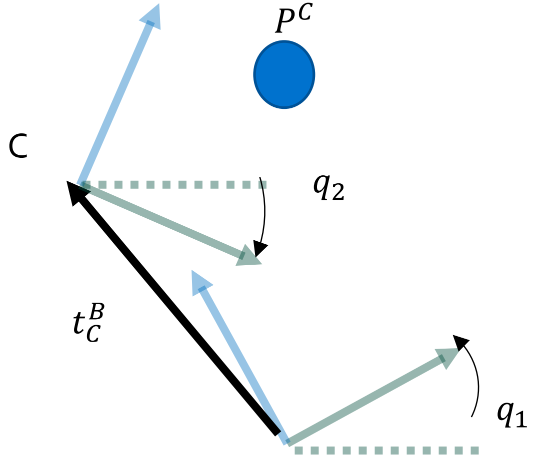
$$E_B^A = \begin{bmatrix} R_B^A & t_B^A \\ 0 & 1 \end{bmatrix}$$



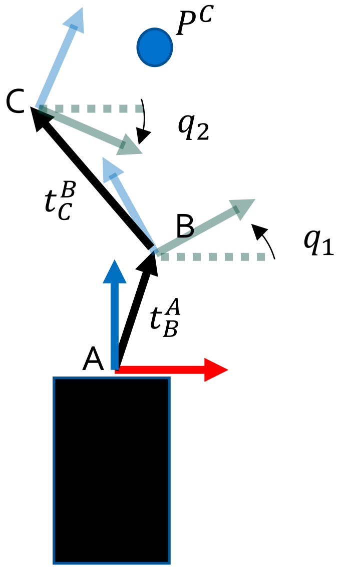
Homogenous Transforms

- Second what is the position and orientation of coordinate frame "C" with respect to coordinate frame "B"
- The position of C with respect to B is constant
- The orientation of C with respect to B depends on the angle q_2

$$R_C^B = \begin{bmatrix} \cos(q_2) & -\sin(q_2) \\ \sin(q_2) & \cos(q_2) \end{bmatrix}$$



Forward Kinematics in 2D

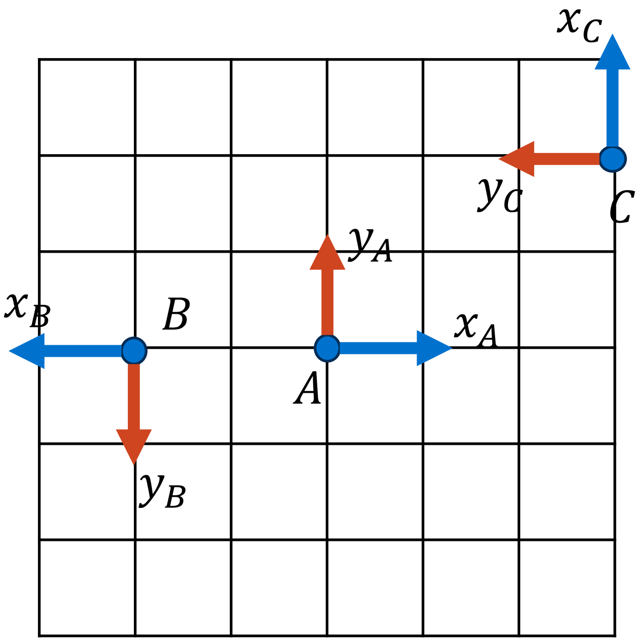


$$\begin{bmatrix} P^B \\ 1 \end{bmatrix} = \begin{bmatrix} R_C^B & t_C^B \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P^C \\ 1 \end{bmatrix} = \mathrm{E}_C^{\mathrm{B}} \begin{bmatrix} P^C \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} P^A \\ 1 \end{bmatrix} = \begin{bmatrix} R_B^A & t_B^A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P^B \\ 1 \end{bmatrix} = \mathrm{E}_{\mathrm{B}}^{\mathrm{A}} \begin{bmatrix} P^B \\ 1 \end{bmatrix}$$

$$P^A = E_B^A E_C^B \begin{bmatrix} P^C \\ 1 \end{bmatrix}$$

2D Transform Example



$$p_A^A = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$p_B^B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$p_C^C = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$q_{AB} = \pi$$
$$q_{BC} = -\frac{\pi}{2}$$

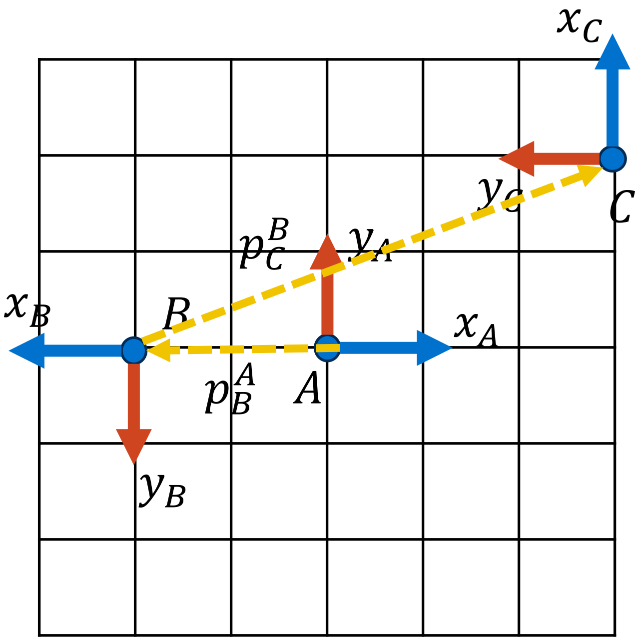
$$t_B^A = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

$$R_{q_{AB}} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$t_C^B = \begin{bmatrix} -5 \\ -2 \end{bmatrix}$$

$$R_{q_{BC}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

2D Transform Example



$$p_A^A = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$p_B^B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$p_C^C = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$p_B^A = R_{q_{AB}} p_B^B + t_B^A$$

$$p_B^A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

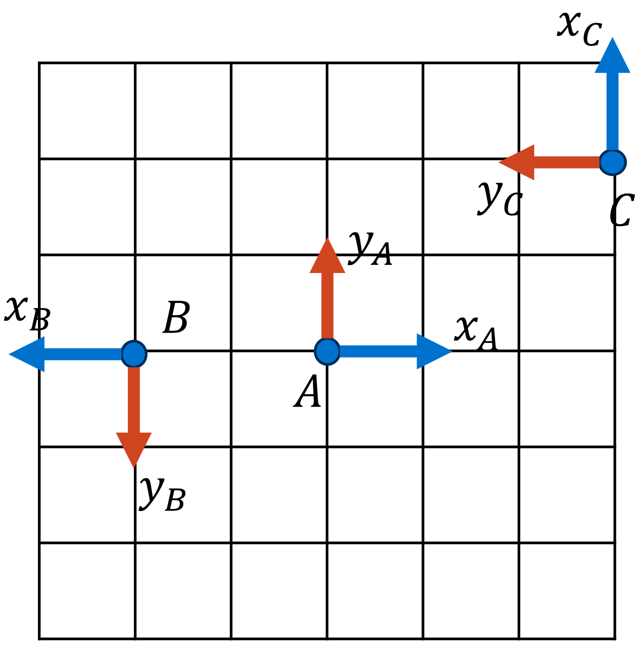
$$p_B^A = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

$$p_C^B = R_{q_{BC}} p_C^C + t_C^B$$

$$p_C^B = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -5 \\ -2 \end{bmatrix}$$

$$p_C^B = \begin{bmatrix} -5 \\ -2 \end{bmatrix}$$

2D Transform Example

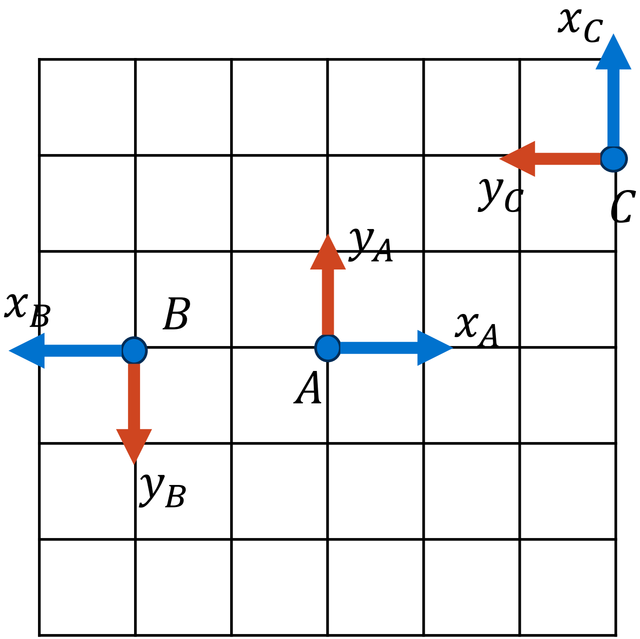


$$E_C^A = E_B^A E_C^B = \begin{bmatrix} -1 & 0 & -2 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & -5 \\ -1 & 0 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$E_A^B = \begin{bmatrix} R_B^A & t_B^A \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & -2 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$E_C^B = \begin{bmatrix} R_C^B & t_C^B \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -5 \\ -1 & 0 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

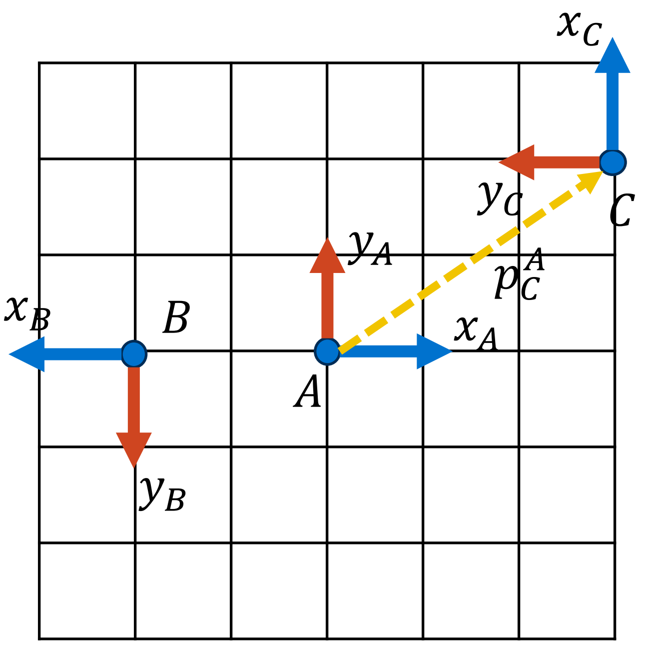
2D Transform Example



$$E_C^A = E_B^A E_C^B = \begin{bmatrix} -1 & 0 & -2 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & -5 \\ -1 & 0 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & -2 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & -5 \\ -1 & 0 & -2 \\ 0 & 0 & 1 \end{bmatrix} =$$

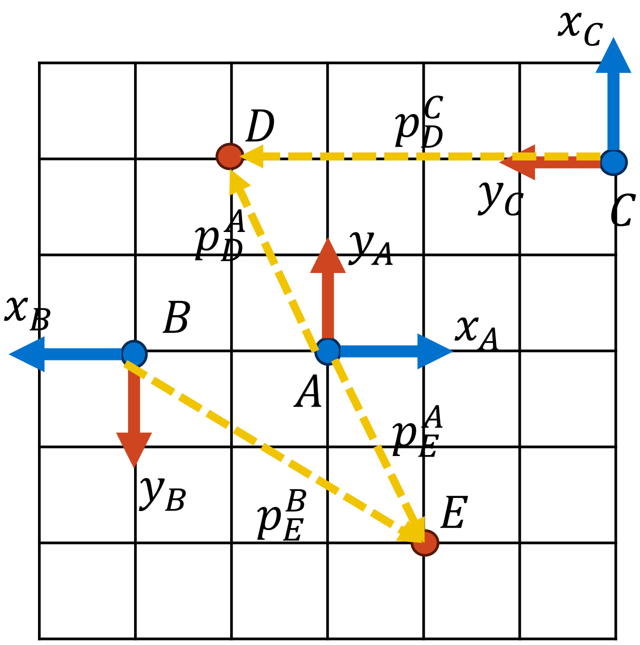
2D Transform Example



$$E_C^A = E_B^A E_C^B = \begin{bmatrix} 0 & -1 & 3 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$p_C^A = E_C^A p_C^C = \begin{bmatrix} 0 & -1 & 3 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

2D Transform Example



$$p_D^C = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

$$p_D^A = E_C^A p_D^C = \begin{bmatrix} 0 & -1 & 3 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$$

$$p_E^B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$p_D^A = E_B^A p_E^B = \begin{bmatrix} -1 & 0 & -2 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

Inverse Kinematics

Inverse Homogenous Representation

- What is the position of joint A in frame B?

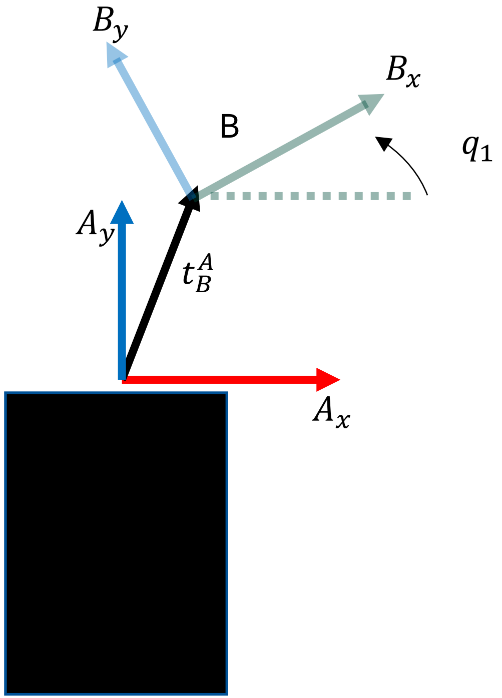
E_B^A lets us convert from B to A,

how to we inverse this to get E_A^B

$$p^A = R_B^A p^B + t_B^A$$

$$p^B = R_B^{A^T}\left(p^A - t_B^A\right)$$

$$E_A^B = \begin{bmatrix} R_B^{A^T} & -R_B^{A^T} t_B^A \\ 0 & 1 \end{bmatrix}$$



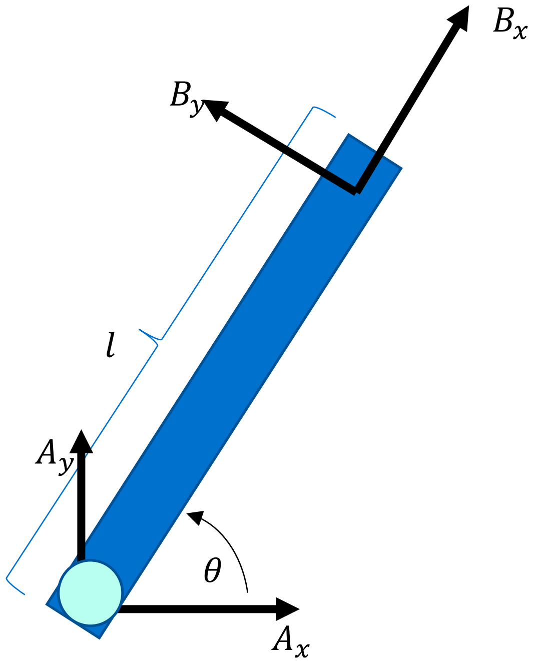
Selecting Correct Frames - Incorrect

$$R_B^A = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad t_B^A = \begin{bmatrix} l*\cos(\theta) \\ l*\sin(\theta) \\ 0 \end{bmatrix}$$

$$E_B^A = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & l*\cos(\theta) \\ \sin(\theta) & \cos(\theta) & 0 & l*\sin(\theta) \\ 0. & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad E_A^B = \begin{bmatrix} R_B^{A^T} & -R_B^{A^T} t_B^A \\ 0 & 1 \end{bmatrix}$$

$$R_A^B = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad t_A^B = \begin{bmatrix} -l \\ 0 \\ 0 \end{bmatrix} \qquad E_B^A = \begin{bmatrix} R_A^{B^T} & -R_A^{B^T} t_A^B \\ 0 & 1 \end{bmatrix}$$

$$E_A^B = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & -l \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



The pivot point for R_A^B is not co-located with the origin of **B**

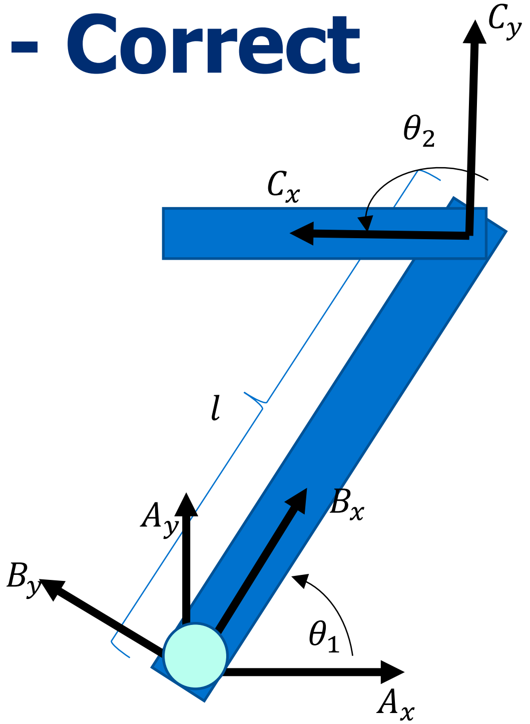
Selecting Correct Frames - Correct

$$R_B^A = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad t_B^A = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$E_B^A = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_C^B = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ -\sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad t_C^B = \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix}$$

$$E_C^B = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

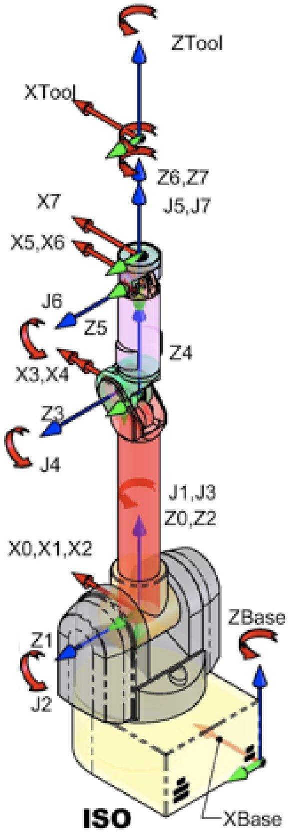


The pivot points for R_B^A and R_C^B are co-located with the origin of **B** and **C**

Forward Kinematics

- Guidelines for assigning frames to robot links:
- There are several conventions
  - Denavit Hartenberg (DH), modified DH, Hayati, etc.
  - They are ”conventions” not “laws”
  - Mainly used for legacy reasons (when using 4 numbers instead of 6 per link made a difference)

1. Choose your base and tool coordinate frame
2. Start from the base and move towards the tool
3. 3) Align each coordinate frame with a joint actuator
   1. Traditionally it’s the Z axis but this is not necessary and any axis can be used to represent the motion of a joint.

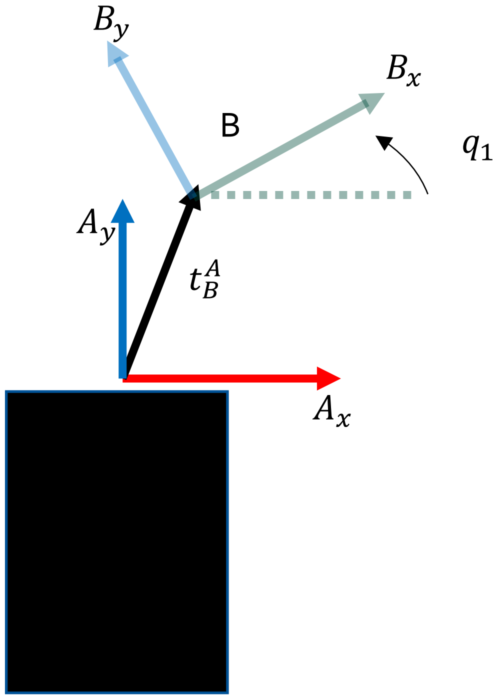


Barrett WAM

Inverse Kinematics 2D

$$E_B^A(q_1) = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & t_x \\ \sin(q_1) & \cos(q_1) & t_y \\ 0 & 0 & 1 \end{bmatrix}_B$$

- Given R_B^A and t_B^A find q
- q only appears in R_B^A
  so solving R for q is pretty easy.
- Adding more joints and things
  start to get messy.



Inverse Kinematics 3D

- Likewise in 3D we want to solve for the position and orientation of the last coordinate frame: Find q_1 and q_2 such that

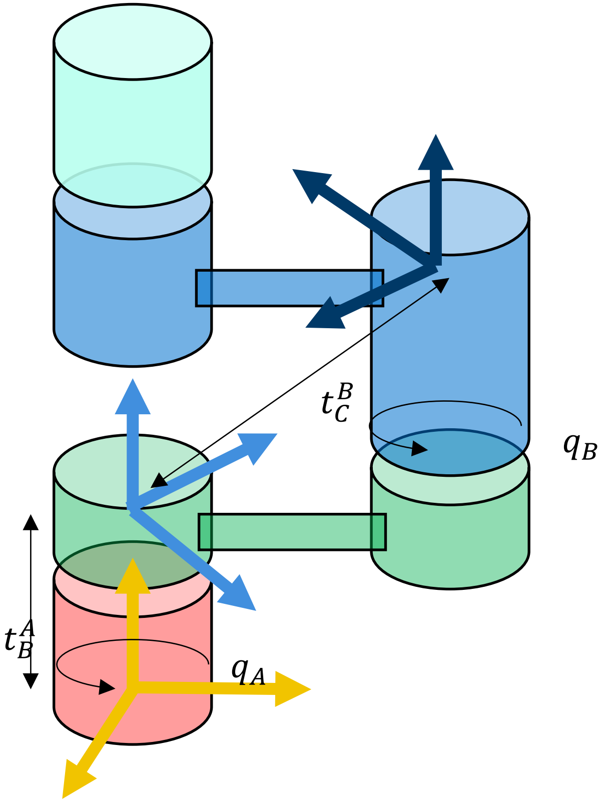
$$E_C^A = E_B^A E_C^B = \begin{bmatrix} R_Z(q_1)R_Z(q_2) & t_B^A + R_Z(q_1)t_B^C \\ 0 & 1 \end{bmatrix}$$

Solving the inverse kinematics gets messy fast

A. For a robot with several joints a symbolic solution can be difficult to get

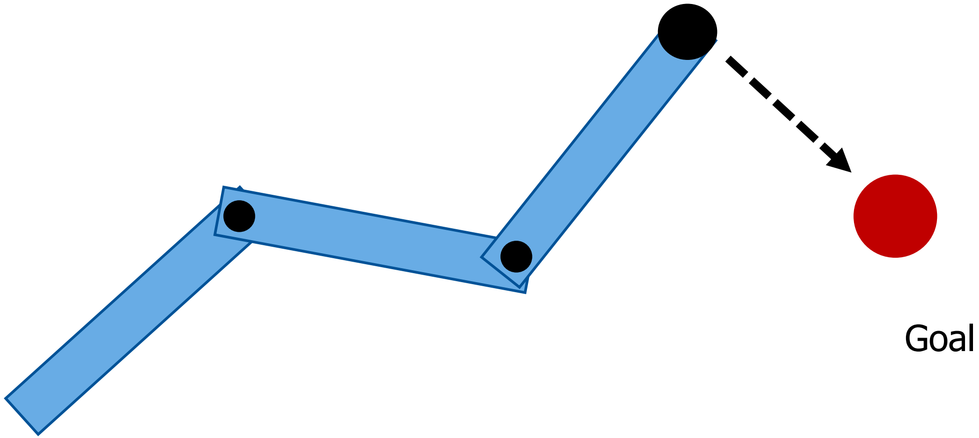
B. A numerical solution (Newton's method) is more generic

Note that the inverse kinematics is not the inverse of the forward kinematics $E_C^{A^{-1}}$



Possible Solution – Motion Rate Control

- Difficult: Directly solving for the joint positions
  - Multiple motion solving packages exist that rely on numerical optimization
  - Outside the scope of this course
- More tractable: Given a starting position can we drive the arm towards a goal?
  - We can provide a velocity in Cartersian space for the end-effector
  - Can we compute the required joint velocities?



Kinematics (2)

Linear Algebra

Rigid body motion
Transformation between coordinate frames

Forward Kinematics

Joint Space

Joint 1 = q1
Joint 2 = q2
…
Joint N = qn

$$f(q) = \left[R_t^B, t_T^B \right]$$

$$\left[q = f^{-1}\left(\left[R_t^B, t_T^B \right] \right) \right.$$

Cartesian Space Velocity

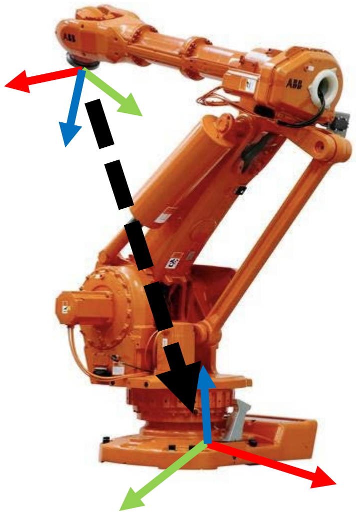
ω : *angular velocityv*
: *linear velocity*$[v, \omega]$

Inverse Kinematics

Velocity Transforms

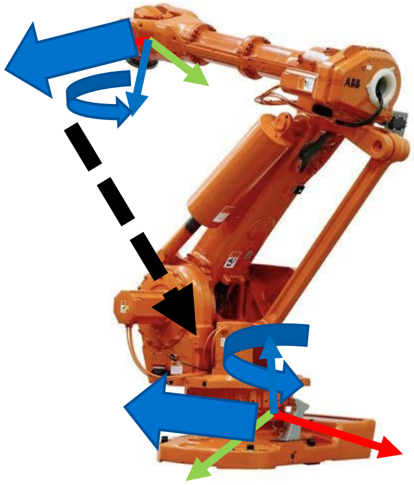
Rigid Body Transformation

Relates two coordinate frames



Rigid Body Velocity

Relate a 3D velocity in one coordinate frame to an equivalent velocity in another coordinate frame



Rotational Velocity (1)

- We note that a rotation relates the coordinates of 3D points with

$$p^A(t) = R_B^A(t)\, p^B$$

- Deriving on both sides with respect to time we get

$$v_{p^A}(t) = \frac{dp^A(t)}{dt} = \dot{R}_B^A(t)\, p^B$$

$$v_{p^A}(t) = \dot{R}_B^A \left[R_B^{A-1} R_B^A \right] p^B$$

$$v_{p^A}(t) = \left(\dot{R}_B^A\, R_B^{A^{-1}} \right) p^A$$

Rotational Velocity (2)

$$\dot{R}_B^A R_B^{A-1} \text{ is skew symmetric } S = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_X & 0 \end{bmatrix} \text{ (aka } S = -S^T \text{)}$$

- The instantaneous spatial angular velocity is defined by

$$\widehat{\omega_B^A} = \hat{k}_B^A = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix} = \dot{R}_B^A R_B^{A^{-1}}$$

where k_B^A is the axis of rotation $\begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$

- There is another way of writing this: A cross product!

$$S(x)p = x \times p$$

Rigid Body Velocity (1)

- We note that a rotation relates to the coordinates of 3D points with

$$\begin{bmatrix} P^A \\ 1 \end{bmatrix} = \begin{bmatrix} R_B^A & t_B^A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P^B \\ 1 \end{bmatrix} = \mathrm{E}_\mathrm{B}^\mathrm{A} \begin{bmatrix} P^B \\ 1 \end{bmatrix}$$

- Just like we did for rotations, deriving on both sides with respect to time we get

$$v_{p^A}(t) = \mathrm{E}_\mathrm{B}^\mathrm{A} \mathrm{E}_\mathrm{B}^{\mathrm{A}^{-1}} p^A$$

- And we expand the matrices to be

$$\dot{\mathrm{E}}_\mathrm{B}^{\dot{A}^A \mathrm{E}_\mathrm{B}^\mathrm{A} -1} = \begin{bmatrix} \dot{R}_B^A & \dot{t}_B^A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_B^{A^T} & -R_B^{A^T} t_B^A \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \dot{R}_B^A R_B^{AT} & -\dot{R}_B^A R_B^{A^T} t_B^A + \dot{t}_B^A \\ 0 & 1 \end{bmatrix}$$

Rigid Body Velocity (2)

- The spatial velocity is defined by $$\hat{V}_B^A = \dot{E}_B^A E_B^{A^{-1}}$$
- Where the linear velocity is defined by $$v_B^A = -\dot{R}_B^A R_B^{A^T} t_B^A + \dot{t}_B^A$$
- And the angular velocity is defined as before by $$\hat{\omega}_B^A = \dot{R}_B^A R_B^{A^T}$$
- Combine these two we obtain the 6D vector $$V_B^A = \begin{bmatrix} v_B^A \\ \omega_B^A \end{bmatrix}$$

The Jacobian

- A matrix of all the first-order partial derivatives.

$$J = \frac{d\mathbf{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial}{\partial x_1}\mathbf{f} & \cdots & \frac{\partial}{\partial x_1}\mathbf{f} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1 & \frac{\partial}{\partial x_2} f_1 & & \frac{\partial}{\partial x_N} f_1 \\ \frac{\partial}{\partial x_1} f_2 & \frac{\partial}{\partial x_2} f_2 & \cdots & \frac{\partial}{\partial x_N} f_2 \\ \frac{\partial}{\partial x_1} f_N & \frac{\partial}{\partial x_2} f_N & \cdots & \frac{\partial}{x_N} f_N \end{bmatrix}$$

- In our context it lets us relate input (joint) velocity to output (cartesian) velocity

$$\dot{Q} = J\dot{V}$$

The Jacobian (cont.)

- Instead of representing the rotation as a derivative of the homogeneous equations it is easier to think of it in terms of the Jacobians.

$$v = J_v \dot{q}$$

$$\omega = J_\omega \dot{q}$$

$$V = \begin{bmatrix} v_B^A \\ \omega_B^A \end{bmatrix} \dot{q} = \begin{bmatrix} J_v \\ J_w \end{bmatrix} \dot{q}$$

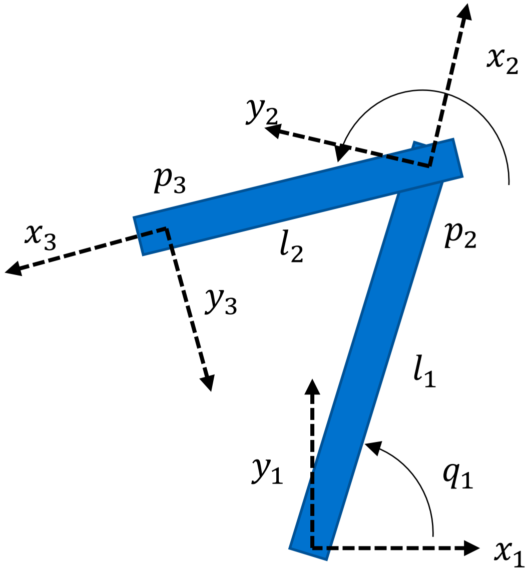
- Relation to the derivative $J_v = \dfrac{\partial x}{\partial q}$ **But** $\boxed{J_\omega \neq \dfrac{\partial r_{\psi\theta\phi}}{\partial q}}$

That's not an angular velocity

Calculating the Jacobian

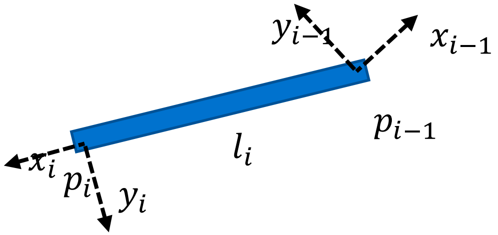
Approach:

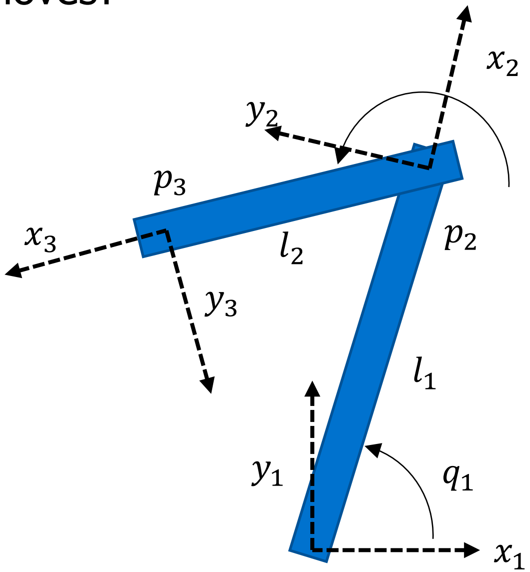
- Calculate the Jacobian one column at a time
- Each column describes the motion at the end effector due to the motion of **that joint only**
- For each joint *I*, pretend all the other joints are frozen and calculate the motion of the end effector caused by *i*



Calculating the Jacobian (cont.)

How does the end effector translate as the ith link moves?





$$p^B_{\text{eff}} = R^B_{i-1} p_{i-1,eff}$$

Orientation of the i-1 link

Vector from the reference frame i-1

To the end effector

Calculating the Jacobian: Velocity

$$p_{\text{eff}}^{B} = R_{i-1}^{B} p_{i-1,\ \text{eff}}^{B}$$

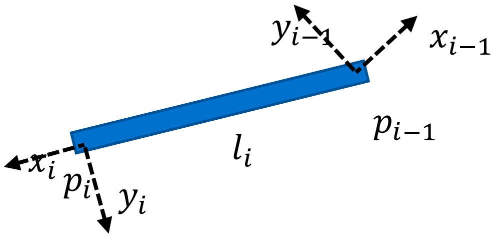
$$\dot{p}_{eff}^{B} = \dot{R}_{i-1}^{B} p_{i-1,eff}^{B} + R_{i-1}^{B} \dot{p}_{i-1,eff}^{B}$$

$$\dot{p}_{\text{eff}}^{B} = \dot{R}_{i-1}^{B} R_{i-1}^{B\ T} R_{i-1}^{B} p_{i-1,eff}^{B} + \dot{p}_{i-1,eff}^{B}$$

$$S\left(k_{i-1}\right) = \dot{R}_{i-1}^{B} R_{i-1}^{B^T}$$

$$\dot{p}_{eff}^{B} = S\left(k_{i-1}\right) p_{i-1}^{B} + \dot{p}_{i-1,eff}^{B}$$

$$\dot{p}_{eff}^{B} = k_i \times p_{i-q,eff}^{B} + \dot{p}_{i-1,eff}^{B}$$



Velocity at end effector due to change in length of i-1

Velocity at end effector due to rotation at joint i-1

Calculating the Jacobian: Velocity (cont.)

Rotational DOF

Rotates about z^{i-1}

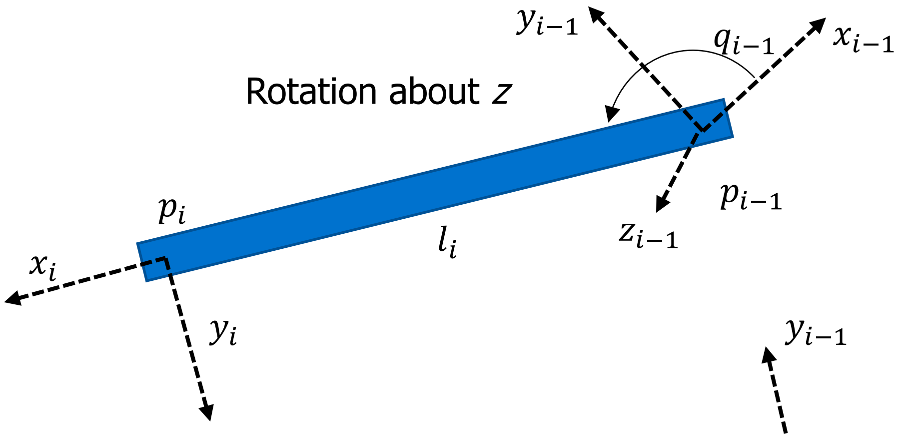
$$J_{v_i} = z_{i-1}^{b} \times p_{i-1,eff}^{B}$$

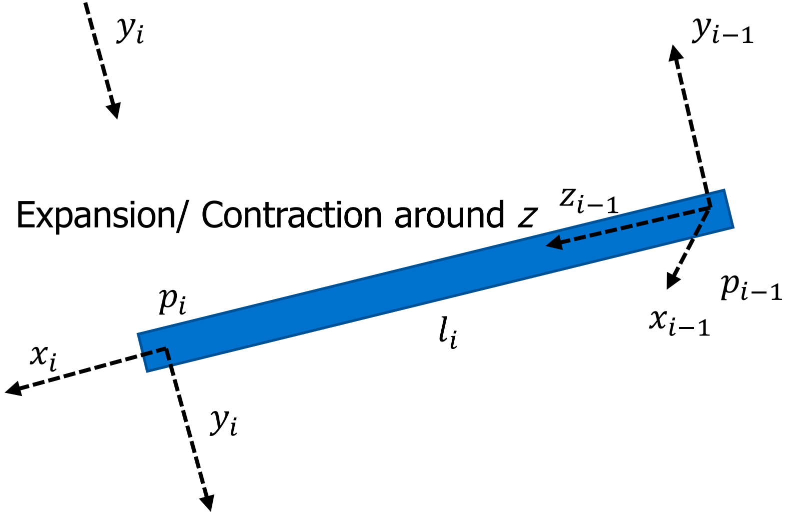
$$J_{v_i} = z_{i-1}^{b} \times \left(p_{eff}^{B} - p_{i-1}^{B} \right)$$

Prismatic DOF

Translates along z^{i-1}

$$J_{v_i} = z_{i-1}^{B}$$





Calculating the Jacobian: Rotational

Rotational DOF

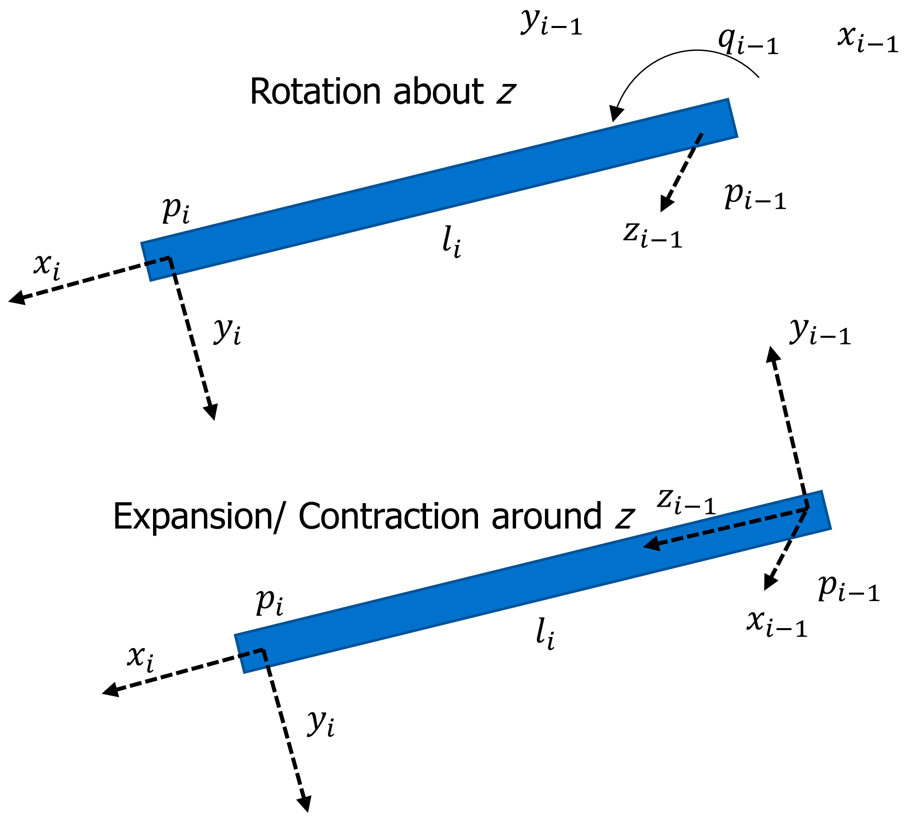
Rotates about z^{i-1}

$$J_{v_i} = z_{i-1}^{b}$$

Prismatic DOF

Translates along z^{i-1}

$$J_{\omega_i} = 0$$



JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Putting It Together

Rotational DOF

Rotates about axis k^{i-1}

$$\mathrm{J}_{\mathrm{v_i}} = \mathrm{k}_{\mathrm{i-1}} \times p^{B}_{i-1,eff}$$

$$\mathrm{J}_{\mathrm{v_i}} = \mathrm{k}_{\mathrm{i-1}} \times \left(p^{B}_{eff} - p^{B}_{i-1}\right)$$

Prismatic DOF

Translates along k^{i-1}

$$J_{v_i} = k^{i-1}$$

Rotational DOF

Rotates about axis k^{i-1}

$$\mathrm{J}_{\omega_{\mathrm{i}}} = k^{i-1}$$

Prismatic DOF

Translates along k^{i-1}

$$J_{\omega_i} = 0$$

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} = \begin{bmatrix} J_{v_1} & \cdots & J_{v_n} \\ J_{\omega_1} & \cdots & J_{\omega_n} \end{bmatrix}$$

Manipulator Jacobian

- We just derived that given a vector of joint velocities the Velocity of the tool as seen in the base of the robot is given by

$$\begin{bmatrix} v_T^{SB} \\ \omega_T^{sB} \end{bmatrix} = J(q)\dot{q}$$

- If instead we want the tool to move with a velocity expressed in the base frame the corresponding joint veolocities can be computed by

$$J^{-1}(q)\begin{bmatrix} v_T^{sB} \\ \omega_T^{sB} \end{bmatrix} = \dot{q}$$

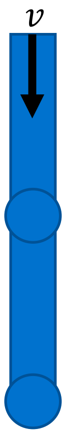
- Inverting a matrix is much easier than computing the inverse kinematics!

Manipulator Jacobian (cont.)

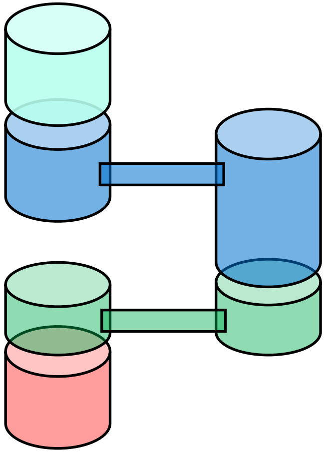
- What if the Jacobian has no inverse?

A. No solution: The velocity is impossible

B. Infinity of solutions: Some joints can be moved without affecting the velocity (i.e. when two axes are colinear)



A)



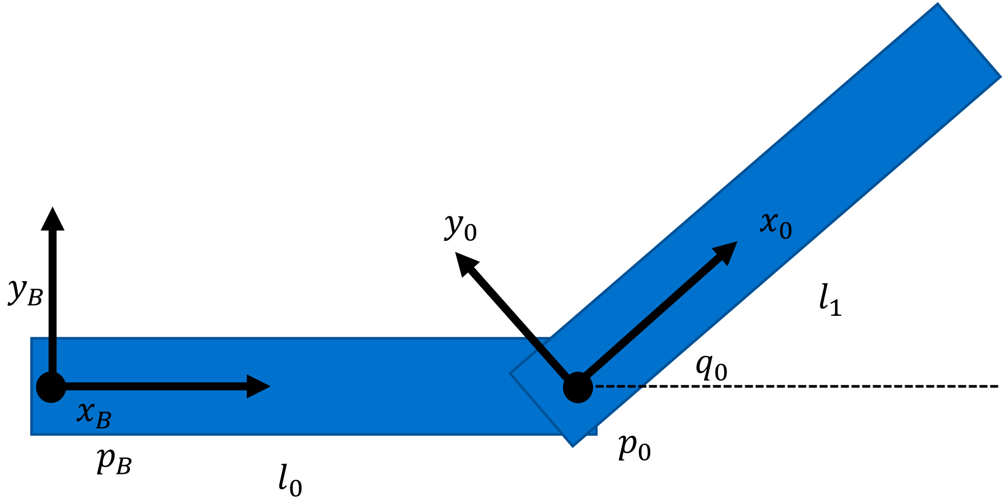
B)

Example for a 3DOF Arm in 2D

Joint Convention for Kinematic Chains

Joint Conventions

- In this class we place the origin for each coordinate system at the location of the joint with the x-axis aligned with the link. Pointing towards the next joint.
- This is similar to the Denavit–Hartenberg Convention and makes the transformations easier.



q_i	The angle of joint *i*
t_i^{i-1}	The translation from the previous joint to the next joint
l_0	The length of the previous link

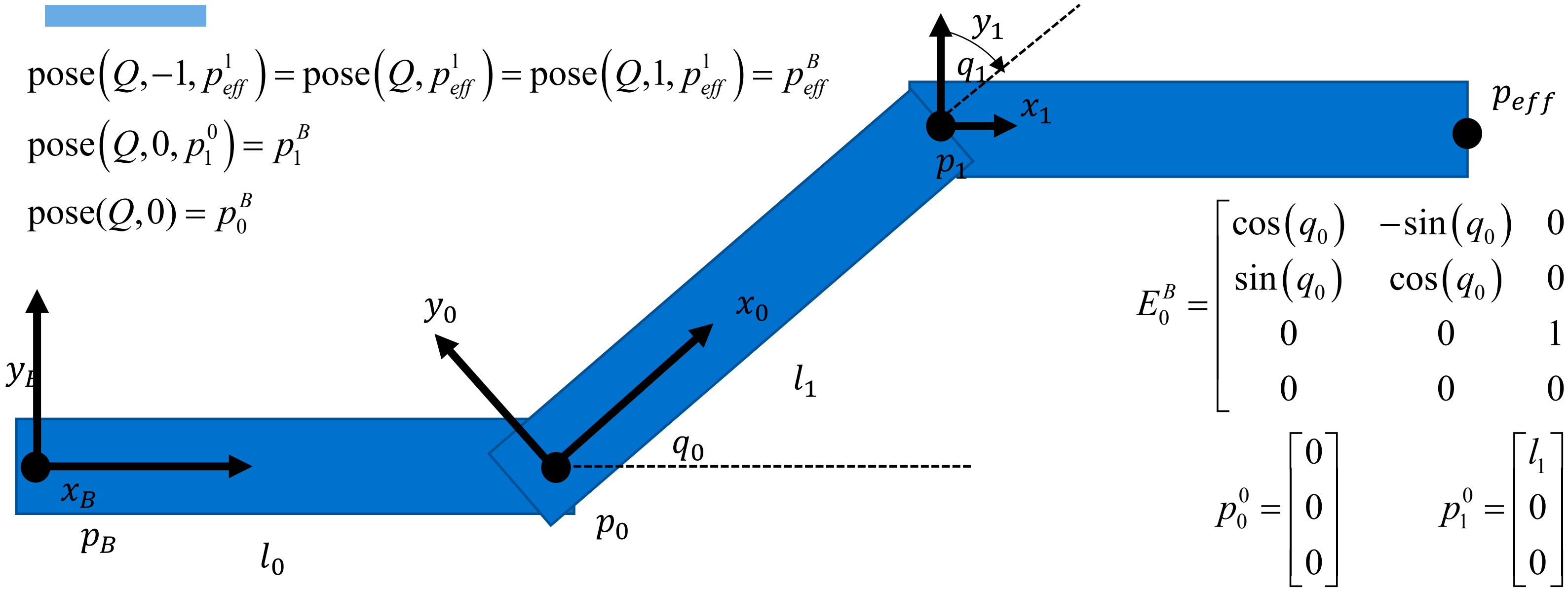
$$R_0^B = \begin{bmatrix} \cos q_0 & -\sin q_0 \\ \sin q_0 & \cos q_0 \end{bmatrix} \qquad t_0^B = \begin{bmatrix} l_0 \\ 0 \\ 0 \end{bmatrix} \qquad E_0^B = \begin{bmatrix} \cos(q_0) & -\sin(q_0) & 0 & l_0 \\ \sin(q_0) & \cos(q_0) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joint Convention for Assignment 4

$$\text{pose}\left(Q,-1,p_{eff}^{1}\right)=\text{pose}\left(Q,p_{eff}^{1}\right)=\text{pose}\left(Q,1,p_{eff}^{1}\right)=p_{eff}^{B}$$

$$\text{pose}\left(Q,0,p_{1}^{0}\right)=p_{1}^{B}$$

$$\text{pose}(Q,0)=p_{0}^{B}$$



$$E_0^B=\begin{bmatrix}\cos(q_0) & -\sin(q_0) & 0 & l_0\\ \sin(q_0) & \cos(q_0) & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$p_0^0=\begin{bmatrix}0\\0\\0\end{bmatrix}\qquad p_1^0=\begin{bmatrix}l_1\\0\\0\end{bmatrix}$$

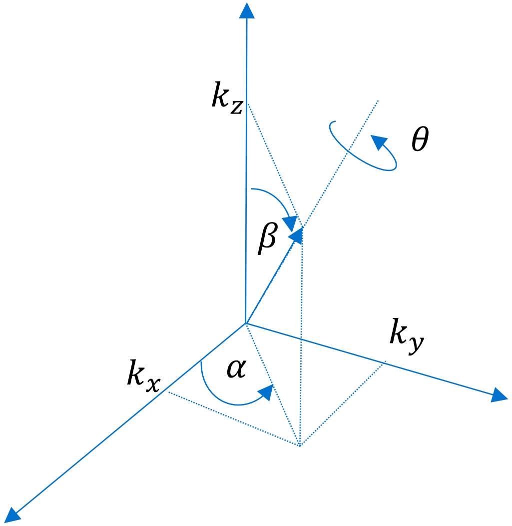
$$p_0^B=\begin{bmatrix}l_0\\0\\0\end{bmatrix}\qquad p_1^B=E_0^Bp_1^0=\begin{bmatrix}l_1\left(\cos q_0-\sin q_0\right)+l_0\\ l_1\left(\sin q_0+\cos q_0\right)\\ 0\end{bmatrix}$$

Remember in python lists are indexed
by zero so the first joint is 0

Axis-Angle Representation

$$R_{k,\theta} = R_{z,\alpha} R_{y,\beta} R_{z,\theta} R_{y,-\beta} R_{z,\alpha} \qquad v_\theta = 1 - c_\theta$$

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



Rotational Velocity (3)

- We note that a rotation relates the coordinates of 3D points with

$$p^A(t) = R_B^A(t)\, p^B$$

- Deriving on both sides with respect to time we get

$$v_{p^A}(t) = \frac{dp^A(t)}{dt} = \dot{R}_B^A(t)\, p^B$$

$$v_{p^A}(t) = \dot{R}_B^A \left[R_B^{A^{-1}} R_B^A \right] p^B$$

$$v_{p^A}(t) = \left(\dot{R}_B^A\, R_B^{A^{-1}} \right) p^A$$

Rotational Velocity (4)

$\dot{R}_B^A R_B^{A^{-1}}$ is equivalent to the following function

$$S\left(\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}\right) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

S is skew symmetric $S = -S^T$

The instantaneous spatial angular velocity is defined by

$$S(k) = \hat{k}_B^A = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}$$

where k_B^A is the axis of rotation $\begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$

There is another way of writing this: A cross product!

$$\boxed{v_{p^A}(t) = \left(\dot{R}_B^A R_B^{A^{-1}}\right) p^A = S(k)\, p^A = \hat{k} p^A = k \times p^A}$$

Rigid Body Velocity (3)

- We note that a rotation relates to the coordinates of 3D points with

$$\begin{bmatrix} P^A \\ 1 \end{bmatrix} = \begin{bmatrix} R_B^A & t_B^A \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} P^B \\ 1 \end{bmatrix} = \mathrm{E}_\mathrm{B}^\mathrm{A} \begin{bmatrix} P^B \\ 1 \end{bmatrix}$$

- Just like we did for rotations, deriving on both sides with respect to time we get

$$v_{p^A}(t) = \dot{\mathrm{E}}_\mathrm{B}^\mathrm{A}\, \mathrm{E}_\mathrm{B}^{\mathrm{A}^{-1}} p^A$$

- And we expand the matrices to be

$$\dot{\mathrm{E}}_\mathrm{B}^A\, \mathrm{E}_\mathrm{B}^{\mathrm{A}-1} = \begin{bmatrix} \dot{R}_B^A & \dot{t}_B^A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_B^{A^T} & -R_B^{A^T} t_B^A \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \dot{R}_B^A R_B^{A^T} & -\dot{R}_B^A R_B^{A^T} t_B^A + \dot{t}_B^A \\ 0 & 1 \end{bmatrix}$$

Rigid Body Velocity (4)

- The **spatial** velocity is defined by

$$\hat{V}_B^{As} = \dot{E}_B^A E_B^{A^{-1}}$$

- Where the linear velocity is defined by

$$v_B^{AS} = -\dot{R}_B^A R_B^{A^T} t_B^A + \dot{t}_B^A = k \times t_B^A + \dot{t}_B^A$$

- And the angular velocity is defined as before by

$$\hat{\omega}_B^{AS} = \dot{R}_B^A R_B^{A^T} = \hat{k}$$

- Combine these two we obtain the 6D vector

$$V_B^{As} = \begin{bmatrix} v_B^{As} \\ \omega_B^{As} \end{bmatrix}$$

Also known as a Twist vector

Forward and Inverse Kinematics

Forward Kinematics

$$\dot{X} = F(X, u)$$

Inverse Kinematics

$$u = F^{-1}(X, \dot{X})$$

Forward Kinematics

$$V = J * \dot{Q}$$

Inverse Kinematics

$$\dot{Q} = J^{-1} V$$

F = Forward kinematic function

F^{-1} = Inverse kinematic function

V = Velocity of a point on the robot

J = Jacobian

$\dot{Q}$ = The rate of change in robot configuration (i.e., joints)

The Jacobian may not be directly invertible so we use a pseudo inverse

$$J^{+} = \left(J^{T} J\right)^{-1} J^{T}$$

Jacobian Calculation Example (1)

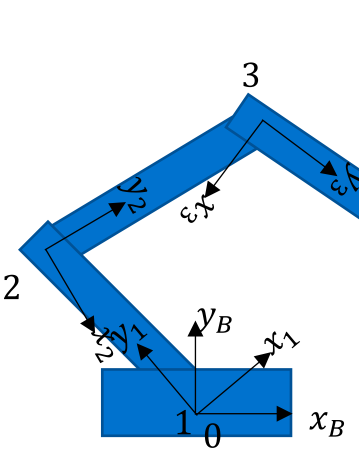
Linear velocity

Jacobian

Joint velocities

$$\dot{V} = J * \dot{Q}$$

$$\dot{Q} = J^{-1}\dot{V}$$

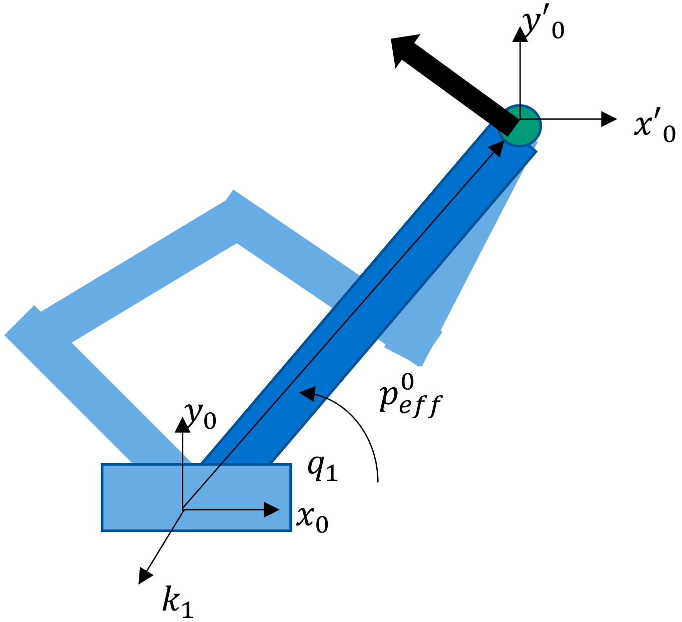


$$p_{\text{eff}} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = J_v \dot{q} = \begin{bmatrix} \frac{dv_x}{dq_1} & \frac{dv_x}{dq_2} & \frac{dv_x}{dq_3} & \frac{dv_x}{dq_4} \\ \frac{dv_y}{dq_1} & \frac{dv_y}{dq_2} & \frac{dv_y}{dq_3} & \frac{dv_y}{dq_4} \\ \frac{dv_z}{dq_1} & \frac{dv_z}{dq_2} & \frac{dv_z}{dq_3} & \frac{dv_z}{dq_4} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix}$$

$$J_{vi}^{eff,0} = \begin{bmatrix} \frac{dv_x}{dq_i} \\ \frac{dv_y}{dq_i} \\ \frac{dv_z}{dq_i} \end{bmatrix} = k \times \left(p_{eff}^0 - p_i^0 \right)$$

$$J_v = \left[J_{v1}, J_{v2}, J_{v3}, J_{v4} \right]$$

Jacobian Calculation Example (2)



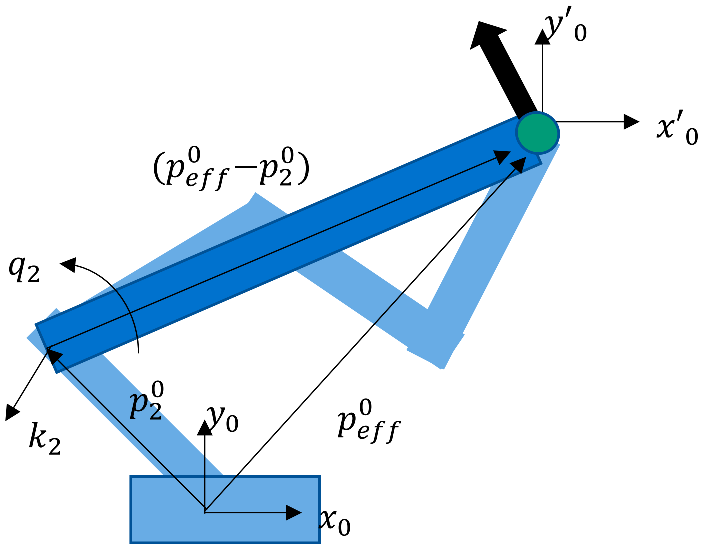
$$J_v = \begin{bmatrix} \boxed{\begin{matrix} \frac{dv_x}{dq_1} \\ \frac{dv_y}{dq_1} \\ \frac{dv_z}{dq_1} \end{matrix}} & \begin{matrix} \frac{dv_x}{dq_2} \\ \frac{dv_y}{dq_2} \\ \frac{dv_z}{dq_2} \end{matrix} & \begin{matrix} \frac{dv_x}{dq_3} \\ \frac{dv_y}{dq_3} \\ \frac{dv_z}{dq_3} \end{matrix} & \begin{matrix} \frac{dv_x}{dq_4} \\ \frac{dv_y}{dq_4} \\ \frac{dv_z}{dq_4} \end{matrix} \end{bmatrix}$$

$$J_{v1}^{eff,0} = \begin{bmatrix} \frac{dv_x}{dq_1} \\ \frac{dv_y}{dq_1} \\ \frac{dv_z}{dq_1} \end{bmatrix} = k \times \left(p^0_{eff} - p^0_1 \right)$$

$$J_v = \left[\boxed{J_{v1}}, J_{v2}, J_{v3}, J_{v4} \right]$$

Note: the base and first joint are collocated so $p_1^B = 0$

Jacobian Calculation Example (3)

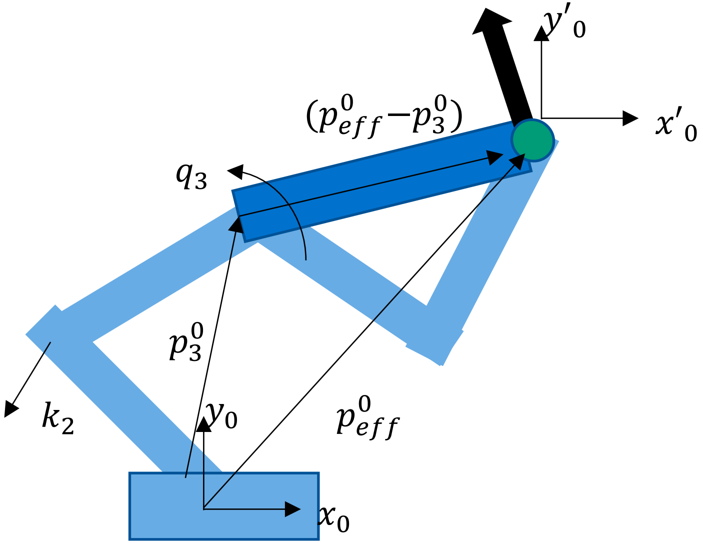


$$J_v = \begin{bmatrix} \frac{dv_x}{dq_1} & \boxed{\frac{dv_x}{dq_2}} & \frac{dv_x}{dq_3} & \frac{dv_x}{dq_4} \\ \frac{dv_y}{dq_1} & \boxed{\frac{dv_y}{dq_2}} & \frac{dv_y}{dq_3} & \frac{dv_y}{dq_4} \\ \frac{dv_z}{dq_1} & \boxed{\frac{dv_z}{dq_2}} & \frac{dv_z}{dq_3} & \frac{dv_z}{dq_4} \end{bmatrix}$$

$$J_{v1}^{eff,0} = \begin{bmatrix} \frac{dv_x}{dq_1} \\ \frac{dv_y}{dq_1} \\ \frac{dv_z}{dq_1} \end{bmatrix} = k \times \left(p_{eff}^0 - p_1^0\right)$$

$$J_v = \left[J_{v1}, \boxed{J_{v2}}, J_{v3}, J_{v4}\right]$$

Jacobian Calculation Example (4)

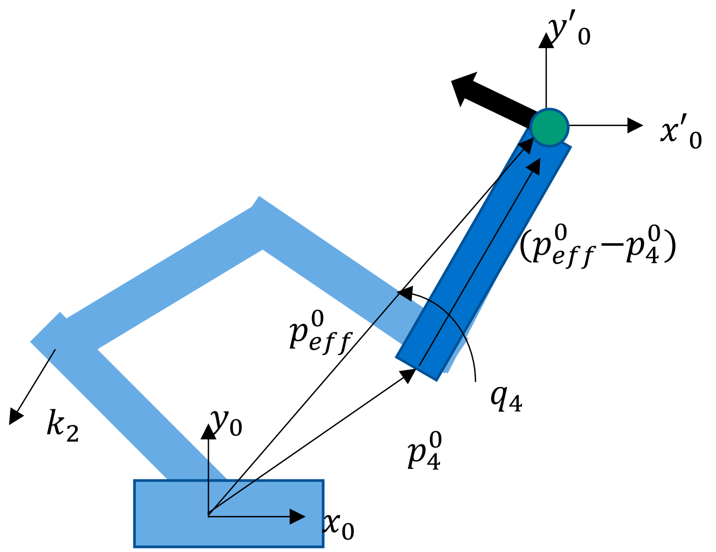


$$J_v = \begin{bmatrix} \frac{dv_x}{dq_1} & \frac{dv_x}{dq_2} & \boxed{\frac{dv_x}{dq_3}} & \frac{dv_x}{dq_4} \\ \frac{dv_y}{dq_1} & \frac{dv_y}{dq_2} & \boxed{\frac{dv_y}{dq_3}} & \frac{dv_y}{dq_4} \\ \frac{dv_z}{dq_1} & \frac{dv_z}{dq_2} & \boxed{\frac{dv_z}{dq_3}} & \frac{dv_z}{dq_4} \end{bmatrix}$$

$$J_{v1}^{eff,0} = \begin{bmatrix} \frac{dv_x}{dq_1} \\ \frac{dv_y}{dq_1} \\ \frac{dv_z}{dq_1} \end{bmatrix} = k \times \left(p_{eff}^0 - p_1^0 \right)$$

$$J_v = \left[J_{v1}, J_{v2}, \boxed{J_{v3}}, J_{v4} \right]$$

Jacobian Calculation Example (5)



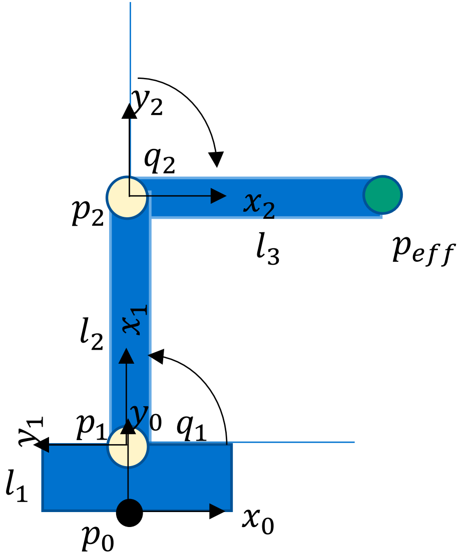
$$J_v = \begin{bmatrix} \frac{dv_x}{dq_1} & \frac{dv_x}{dq_2} & \frac{dv_x}{dq_3} & \frac{dv_x}{dq_4} \\ \frac{dv_y}{dq_1} & \frac{dv_y}{dq_2} & \frac{dv_y}{dq_3} & \frac{dv_y}{dq_4} \\ \frac{dv_z}{dq_1} & \frac{dv_z}{dq_2} & \frac{dv_z}{dq_3} & \frac{dv_z}{dq_4} \end{bmatrix}$$

$$J_{v1}^{eff,0} = \begin{bmatrix} \frac{dv_x}{dq_1} \\ \frac{dv_y}{dq_1} \\ \frac{dv_z}{dq_1} \end{bmatrix} = k \times \left(p_{eff}^0 - p_1^0\right)$$

$$J_v = [J_{v1}, J_{v2}, J_{v3}, \boxed{J_{v4}}]$$

Example (1)

$$J_v^{eff,0} = \begin{bmatrix} J_{v_1} & J_{v2} \end{bmatrix} = \begin{bmatrix} \frac{dv_x}{dq_1} & \frac{dv_x}{dq_2} \\ \frac{dv_y}{dq_1} & \frac{dv_y}{dq_2} \\ \frac{dv_z}{dq_1} & \frac{dv_z}{dq_2} \end{bmatrix}$$



$$l_1 = 0.5 \qquad l_2 = l_3 = 1$$

$$q_1 = \frac{\pi}{2}, q_2 = -\frac{\pi}{2}$$

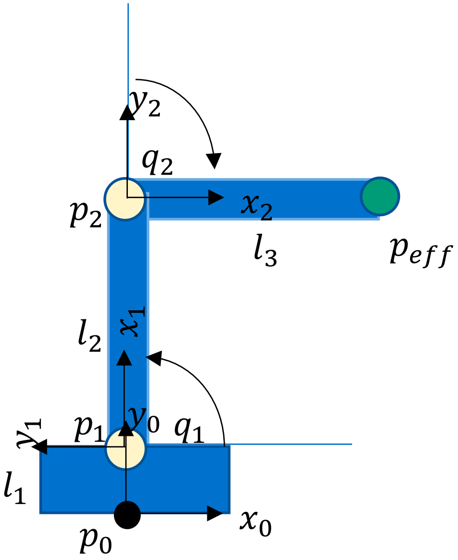
$$R_1^0 = \begin{bmatrix} \cos q_1 & -\sin q_1 \\ \sin q_1 & \cos q_1 \end{bmatrix} \qquad t_1^0 = \begin{bmatrix} 0 \\ l_0 \end{bmatrix}$$

$$E_1^0 = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ \sin q_1 & \cos q_1 & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix} \qquad p_1^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$E_1^0 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix} \qquad p_1^0 = \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix}$$

Example (2)

$$J_v^{eff,0} = \begin{bmatrix} J_{v_1} & J_{v2} \end{bmatrix} = \begin{bmatrix} \frac{dv_x}{dq_1} & \frac{dv_x}{dq_2} \\ \frac{dv_y}{dq_1} & \frac{dv_y}{dq_2} \\ \frac{dv_z}{dq_1} & \frac{dv_z}{dq_2} \end{bmatrix}$$



$$l_0 = 0.5 \qquad l_1 = l_2 = 1$$

$$q_1 = \frac{\pi}{2}, q_2 = -\frac{\pi}{2}$$

$$E_1^0 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix} \qquad p_1^0 = \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix}$$

$$E_2^1 = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & l_2 \\ \sin q_2 & \cos q_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix} \qquad p_2^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

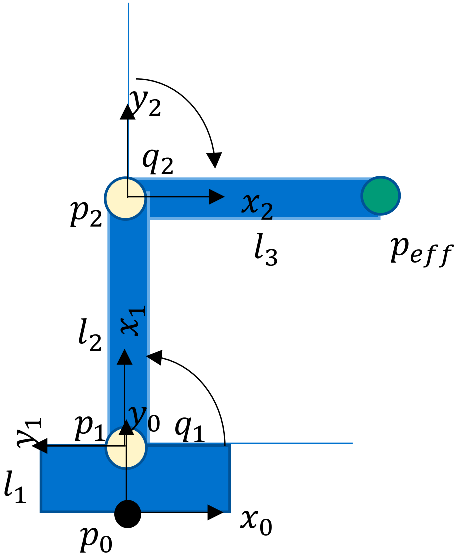
$$E_2^1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix} \qquad p_2^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$E_2^0 = E_1^0 E_2^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_2 + l_1 \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix} \qquad p_2^0 = \begin{bmatrix} 0 \\ 1.5 \\ 0 \end{bmatrix}$$

Example (3)

$$E_2^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_2 + l_1 \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix} \qquad p_{eff}^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$J_v^{eff,0} = \begin{bmatrix} J_{v_1} & J_{v2} \end{bmatrix} = \begin{bmatrix} \frac{dv_x}{dq_1} & \frac{dv_x}{dq_2} \\ \frac{dv_y}{dq_1} & \frac{dv_y}{dq_2} \\ \frac{dv_z}{dq_1} & \frac{dv_z}{dq_2} \end{bmatrix}$$



$$p_{\text{eff}}^0 = \begin{bmatrix} 1 \\ 1.5 \\ 0 \end{bmatrix} \qquad p_2^0 = \begin{bmatrix} 0 \\ 1.5 \\ 0 \end{bmatrix} \qquad p_1^0 = \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix}$$

$$k_1 = [0,0,1]^T \qquad k_2 = [0,0,1]^T$$

$$J_{v_1} = \mathrm{k}_1 \times \left(p_{eff}^0 - p_1^0 \right)$$

$$J_{v_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} 1 \\ 1.5 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix} \right)$$

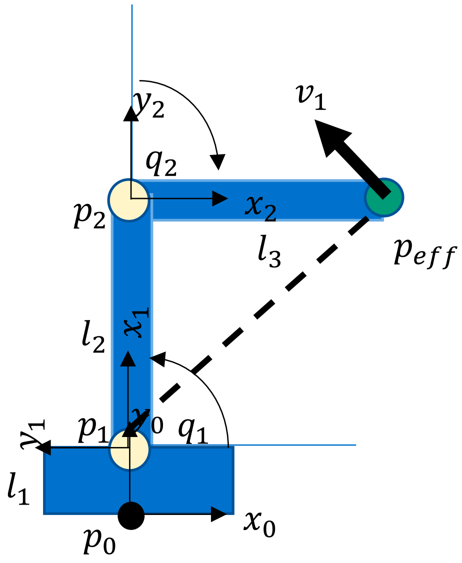
$$l_0 = 0.5 \qquad l_1 = l_2 = 1$$

$$q_1 = \frac{\pi}{2}, q_2 = -\frac{\pi}{2}$$

Example (4)

$$J_{v_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} 1 \\ 1.5 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix} \right)$$

$$J_v^{eff,0} = \begin{bmatrix} J_{v_1} & J_{v2} \end{bmatrix} = \begin{bmatrix} -1 & \frac{dv_x}{dq_2} \\ 1 & \frac{dv_y}{dq_2} \\ 0 & \frac{dv_z}{dq_2} \end{bmatrix}$$



$$J_{v_1} = \begin{bmatrix} -1*1+0*0 \\ 1*1-0*0 \\ -0*1+0*1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

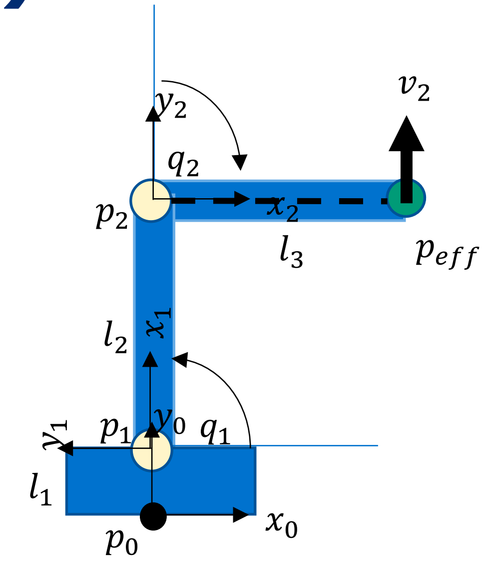
$$J_v = \begin{bmatrix} -1 & - \\ 1 & - \\ 0 & - \end{bmatrix}$$

$$l_0 = 0.5 \qquad l_1 = l_2 = 1$$

$$q_1 = \frac{\pi}{2}, q_2 = -\frac{\pi}{2}$$

Example (5)

$$J_v^{eff,0} = \begin{bmatrix} J_{v_1} & J_{v2} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}$$



$$l_0 = 0.5 \qquad l_1 = l_2 = 1$$

$$q_1 = \frac{\pi}{2}, q_2 = -\frac{\pi}{2}$$

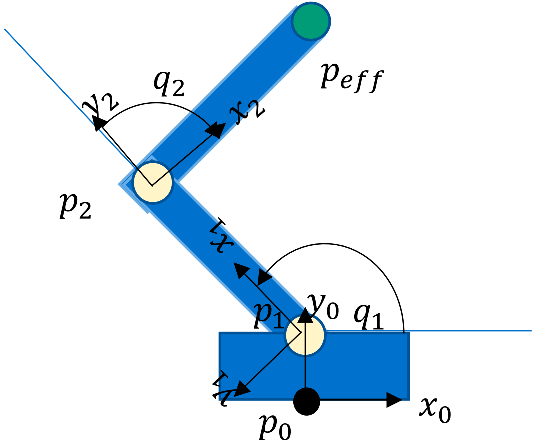
$$J_{v_2} = \mathrm{k}_1 \times \left(p_{eff}^0 - p_2^0\right)$$

$$J_{v_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} 1 \\ 1.5 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1.5 \\ 0 \end{bmatrix} \right)$$

$$J_{v_2} = \begin{bmatrix} -1*0+0*0 \\ 1*1-0*0 \\ -0*1+0*1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$J_v = \begin{bmatrix} -1 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}$$

Example (6)



$$E_1^0 = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0.5 \\ 0 & 0 & 1 & \\ & 0 & & 1 \end{bmatrix} \qquad p_1^0 = \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix}$$

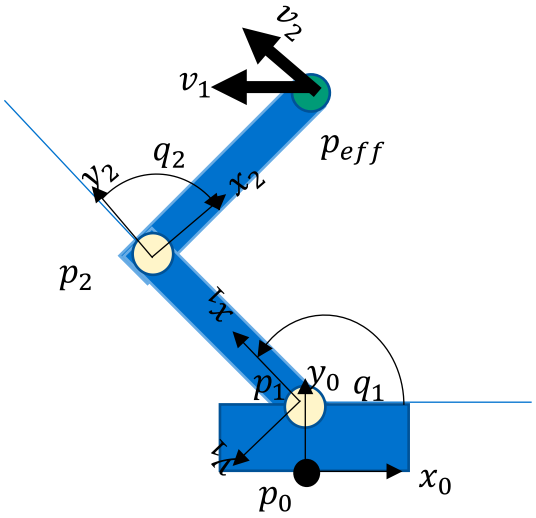
$$E_2^1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ & 0 & & 1 \end{bmatrix} \qquad p_2^0 = \begin{bmatrix} -0.707 \\ 1.207 \\ 0 \end{bmatrix}$$

$$l_0 = 0.5 \qquad l_1 = l_2 = 1$$

$$q_1 = \boxed{\frac{3\pi}{4}}, q_2 = -\frac{\pi}{2}$$

$$E_2^0 = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} + \frac{1}{2} \\ 0 & 0 & 1 & 0 \\ & & & \end{bmatrix} \qquad p_{\mathrm{eff}}^0 = \begin{bmatrix} 0 \\ 1.9142 \\ 0 \end{bmatrix}$$

Example (7)



$$l_0 = 0.5 \qquad l_1 = l_2 = 1$$

$$q_1 = \boxed{\frac{3\pi}{4}}, q_2 = -\frac{\pi}{2}$$

$$p_1^0 = \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix} \qquad p_2^0 = \begin{bmatrix} -0.707 \\ 1.207 \\ 0 \end{bmatrix} \qquad p_{\text{eff}}^0 = \begin{bmatrix} 0 \\ 1.9142 \\ 0 \end{bmatrix}$$

$$J_{v_1} = \mathrm{k}_1 \times \left(p_{eff}^0 - p_1^0 \right)$$

$$J_{v_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} 0 \\ 1.9142 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} -1.4142 \\ 0 \\ 0 \end{bmatrix}$$

$$J_{v_2} = \mathrm{k}_1 \times \left(p_{eff}^0 - p_1^0 \right)$$

$$J_{v_2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} 0 \\ 1.9142 \\ 0 \end{bmatrix} - \begin{bmatrix} -0.707 \\ 1.207 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} -0.707 \\ 0.707 \\ 0 \end{bmatrix}$$

$$J_v = \begin{bmatrix} 1.414 & -0.707 \\ 0 & 0.707 \\ 0 & 0 \end{bmatrix}$$

Assignment Recap

Assignment 2 Visualization, Assignment 3 Demo

Quaternion Matrix Discussion

- Frequently asked question. Why do the slides not match the equation on Wikipedia?

$$R(Q)=\begin{bmatrix} 1-2s\left(q_3^2+q_2^2\right) & 2s\left(q_1q_2-q_0q_3\right) & 2s\left(q_1q_3+q_0q_2\right) \\ 2s\left(q_1q_2+q_0q_3\right) & 1-2s\left(q_1^2+q_3^2\right) & 2s\left(q_2q_3-q_0q_1\right) \\ 2s\left(q_1q_3-q_0q_2\right) & 2s\left(q_2q_3+q_0q_1\right) & 1-2s\left(q_1^2+q_2^2\right) \end{bmatrix}$$

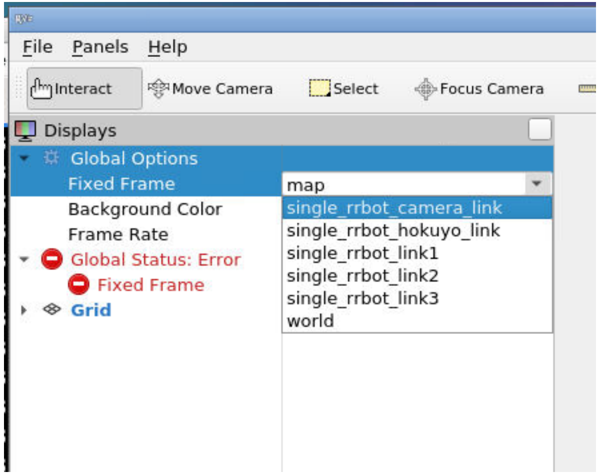
- Answer:

  $1-2s\left(q_3^2+q_2^2\right)$ and $2s\left(q_0^2+q_1^2\right)-1$

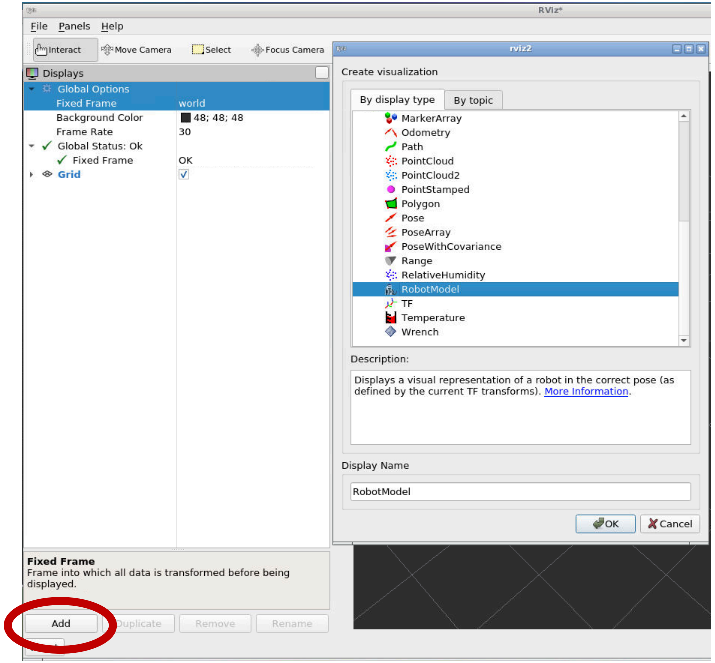
  - The S factor is only required if the quaternion is not a unit

Viewing Robots in RVIZ

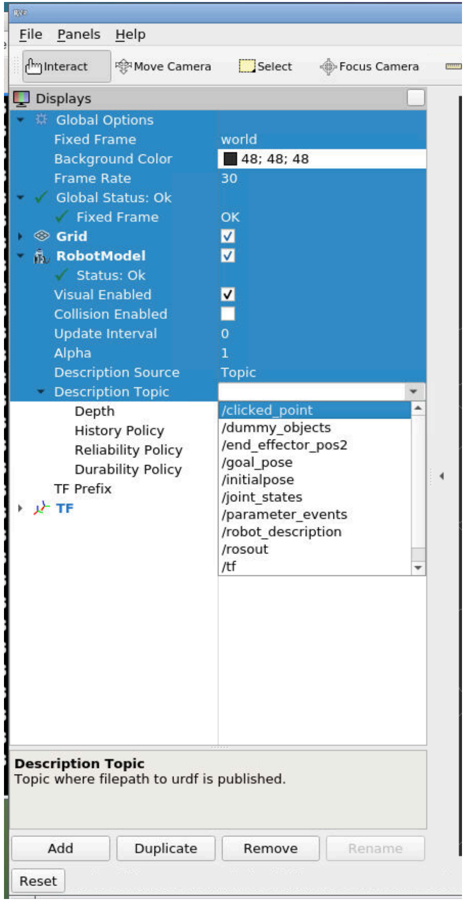
Step 1.
Select the correct frame



Step 2.
Add RobotModel and TF elements to the display



Step 3.
Select the topics to display (e.g. /robot_description)

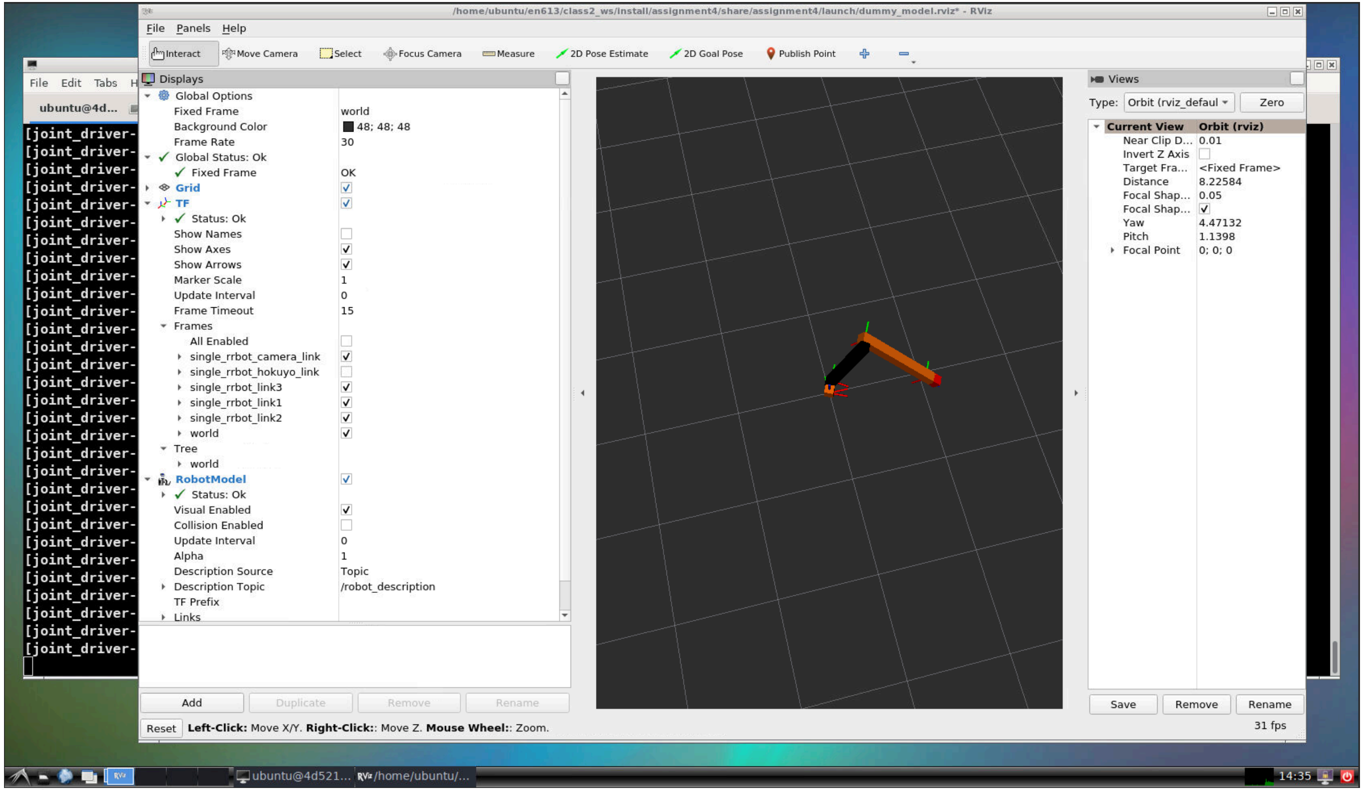


Assignment 3

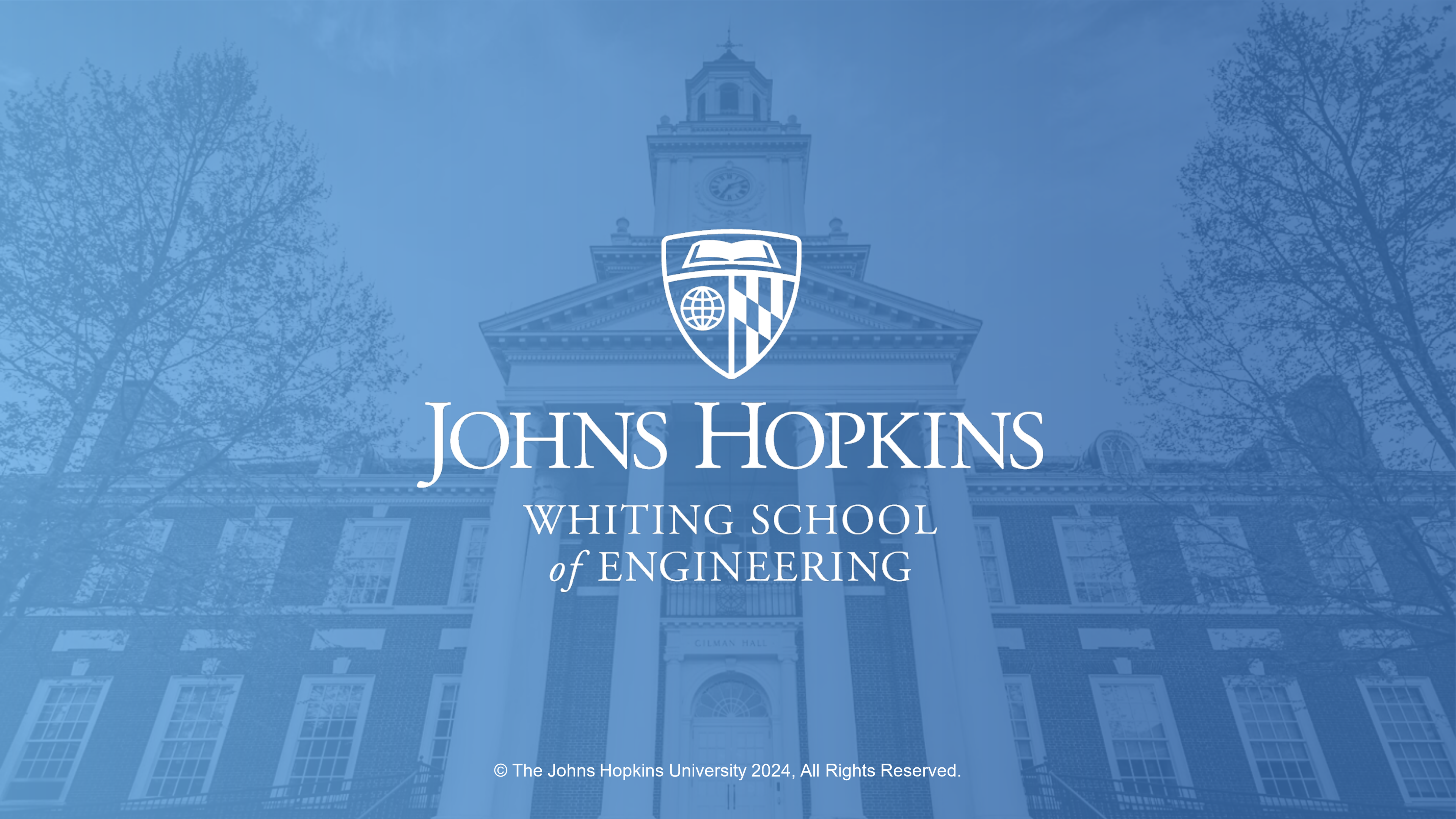
- Goal: Students will compute the forward kinematics for a Kinematic Chain.
- The inverse kinematic function and visualization code will be provided but rely on correct computation of the forward kinematics and jacobian to execute

1. Setup class4_ws and download assignment4.zip
2. Implement the following functions in assignment4.py
   a) axis_angle_rot_matrix 15 pts
   b) hr_matrix 15 pts
   c) inverse_hr_matrix 10 pts
   d) KinematicChain.pose 30 pts
   e) KinematicChain.jacobian 30 pts

Assignment 3 Output



JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING


JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING
GILMAN HALL