Week 5 – Jacobians: Velocities and Static Forces

Advanced Robotic Systems – MANU2453

Dr Ehsan Asadi, School of Engineering
RMIT University, Victoria, Australia
Email: ehsan.asadi@rmit.edu.au

RMIT UNIVERSITY

Lectures

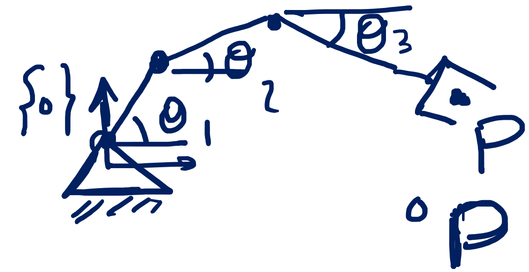
Wk	Date	Lecture (NOTE: video recording)	Maths Difficulty	Hands-on Activity	Related Assessment
1	24/7	• Introduction to the Course • Spatial Descriptions & Transformations			
2	31/7	• Spatial Descriptions & Transformations • Robot Cell Design			Robot Cell Design Assignment
3	7/8	• Forward Kinematics • Inverse Kinematics			
4	14/8	• ABB Robot Programming via Teaching Pendant • ABB RobotStudio Offline Programming		ABB RobotStudio Offline Programming	Offline Programming Assignment
5	21/8	• Jacobians: Velocities and Static Forces			
6	28/8	• Manipulator Dynamics			
7	11/9	• Manipulator Dynamics		MATLAB Simulink Simulation	
8	18/9	• Robotic Vision		MATLAB Simulation	Robotic Vision Assignment
9	25/9	• Robotic Vision		MATLAB Simulation	
10	2/10	• Trajectory Generation			
11	9/10	• Linear & Nonlinear Control		MATLAB Simulink Simulation	
12	16/10	• Introduction to I4.0 • Revision			Final Exam

Content

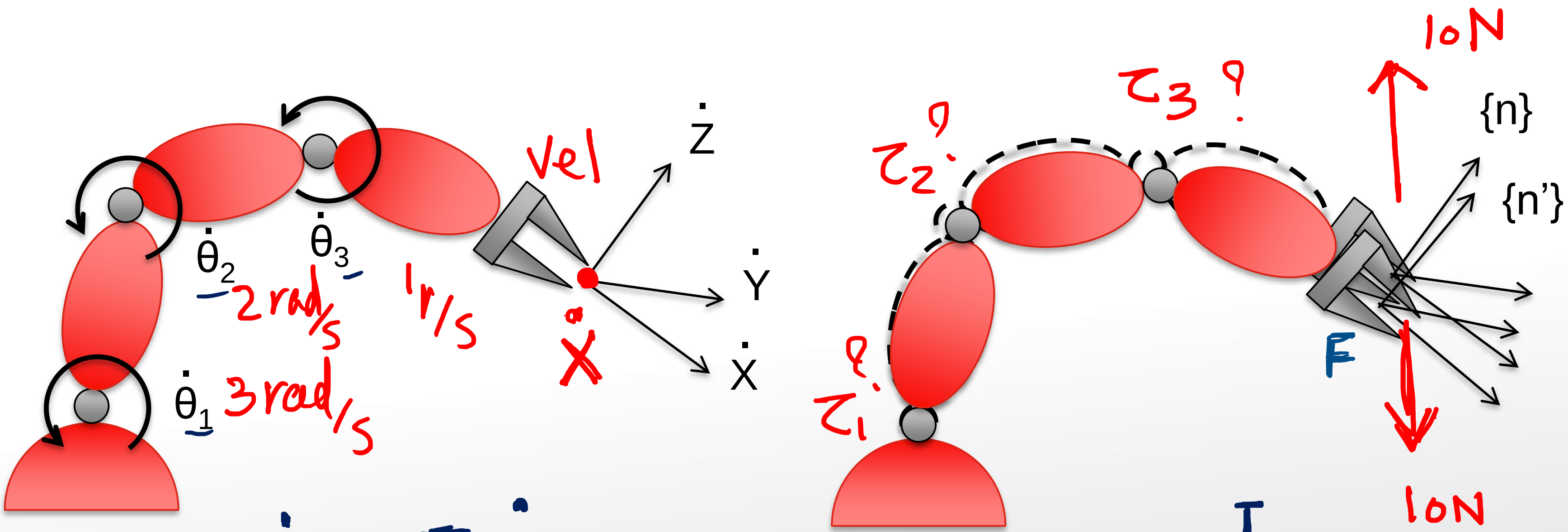
- Introduction - Jacobian
- Method 1 - Direct differentiation (for Linear Jacobian)
- Method 2 - Velocity Propagation from Link to Link
- Method 3 - Explicit Form (for your study, not included in exam)
- Static Forces in Manipulators
- Singularities

Introduction

Fwd-Kin



- In this lecture, we will learn:
  - Relationship between joint velocities and end-effector velocities
  - Relationship between the task-space force and the joint-space torques



$$\dot X = J \dot\theta$$

$$\tau = J^T F$$

Jacobian

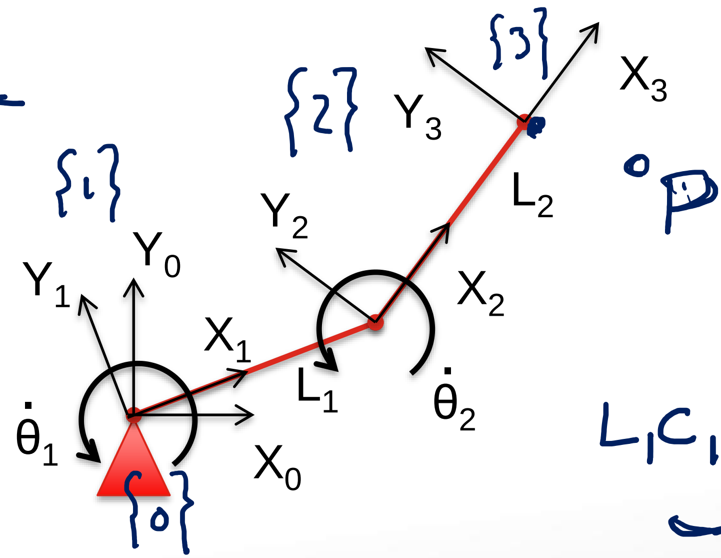
Content

- Introduction - Jacobian
- Method 1 - Direct differentiation (for Linear Jacobian)
- Method 2 - Velocity Propagation from Link to Link
- Method 3 - Explicit Form (for your study, not included in exam)
- Static Forces in Manipulators
- Singularities

Differentiation to Calculate Velocity

- In our previous lectures, we calculated the tip position for a two-link robot:
- Recall that the tip position is:

$$ {}^{0}\vec{P} = \begin{bmatrix} L_1c_1 + L_2c_{12} \\ L_1s_1 + L_2s_{12} \\ 0 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} $$



Differentiate the position vector wrt time

The velocity is $\dot{x} = \frac{dx}{dt}$

$L_1c_1 + L_2c_{12}$

$\theta_1 \,\&\, \theta_2$

t

$\frac{d\theta_1}{dt}, \frac{d\theta_2}{dt}$

$$\dot{x} = \frac{dx}{d\theta_1}\frac{d\theta_1}{dt} + \frac{dx}{d\theta_2}\frac{d\theta_2}{dt}$$

$$\dot{y} = \frac{dy}{d\theta_1}\frac{d\theta_1}{dt} + \frac{dy}{d\theta_2}\frac{d\theta_2}{dt}$$

$$\dot{z} = \text{----}$$

Chain Rule

Differentiation to Calculate Velocity

Differentiate the position vector wrt time, using chain rule:

$$\dot{x} = \frac{dx}{d\theta_1}\underbrace{\frac{d\theta_1}{dt}}_{\dot{\theta}_1} + \frac{dx}{d\theta_2}\underbrace{\frac{d\theta_2}{dt}}_{\dot{\theta}_2}$$

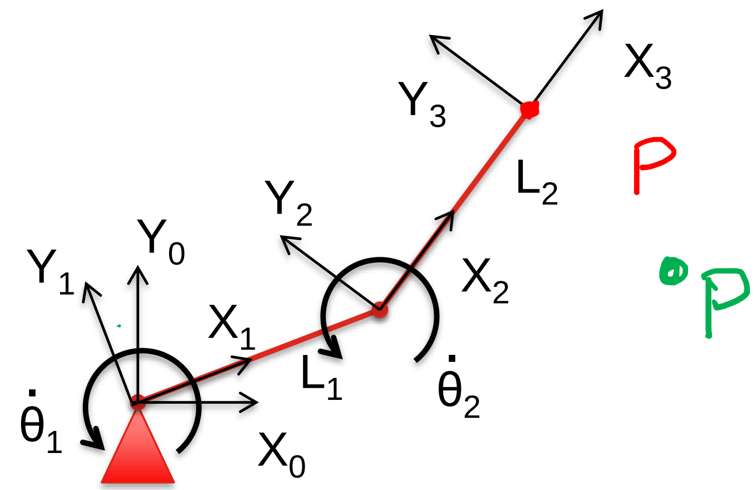
$$\dot{y} = \frac{dy}{d\theta_1}\underbrace{\frac{d\theta_1}{dt}}_{\dot{\theta}_1} + \frac{dy}{d\theta_2}\underbrace{\frac{d\theta_2}{dt}}_{\dot{\theta}_2}$$

$$\dot{z} = \cdots$$

Compare with $\dot{X} = J\dot{\Theta}$ → Jacobian

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix}_{3\times1} = \begin{bmatrix} dx/d\theta_1 & dx/d\theta_2 \\ dy/d\theta_1 & dy/d\theta_2 \\ dz/d\theta_1 & dz/d\theta_2 \end{bmatrix}_{3\times2} \begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}_{3\times1}$$

$\{0\}$ ${}^{0}J$ $\underline{v}$ → linear



Differentiation to Calculate Velocity

- Looking back at our two-link robot example:

$$\dot{x} = J\dot{\theta}$$

$$x \rightarrow \quad {}^0P = \begin{bmatrix} L_1c_1 + L_2c_{12} \\ L_1s_1 + L_2s_{12} \\ 0 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} dx/d\theta_1 & dx/d\theta_2 \\ dy/d\theta_1 & dy/d\theta_2 \\ dz/d\theta_1 & dz/d\theta_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$u = \theta_1 + \theta_2 \qquad x = L_1C_1 + L_2C_u$$

- Using the Jacobian method:

$$\frac{dx}{d\theta_1} = \frac{d(L_1C_1)}{d\theta_1} + \frac{d(L_2C_u)}{d\theta_1} \rightarrow \text{Chain Rule} \rightarrow = -L_1S_1 + \frac{d(L_2C_u)}{du}\frac{du}{d\theta_1}$$

$$= -L_1S_1 - L_2S_u = -L_1S_1 - L_2S_{12}$$

$$\frac{dx}{d\theta_2} = 0 + \frac{d(L_2C_u)}{du}\frac{du}{d\theta_2}$$

$$= 0 - L_2S_{12}$$

Another Way to Calculate Velocity

- Differentiate the position vector wrt time, using chain rule:

$$\frac{d}{dt}(P) = \begin{bmatrix} \frac{d}{dt}(L_1c_1 + L_2c_{12}) \\ \frac{d}{dt}(L_1s_1 + L_2s_{12}) \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{d}{d\theta_1}(L_1c_1 + L_2c_{12}) & \frac{d}{d\theta_2}(L_1c_1 + L_2c_{12}) \\ \frac{d}{d\theta_1}(L_1s_1 + L_2s_{12}) & \frac{d}{d\theta_2}(L_1s_1 + L_2s_{12}) \\ & 0 \end{bmatrix} \begin{bmatrix} \frac{d\theta_1}{dt} \\ \frac{d\theta_2}{dt} \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} (-L_1s_1 - L_2s_{12}) & -L_2s_{12} \\ (L_1c_1 + L_2c_{12}) & L_2c_{12} \\ 0 & 0 \end{bmatrix}}_{J} \times \underbrace{\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}}_{\dot{\theta}}$$

.

Velocity

$$\underline{{}^0\upsilon_3} = \begin{bmatrix} -L_1s_1\dot{\theta}_1 - L_2s_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ L_1c_1\dot{\theta}_1 + L_2c_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix}$$

Jacobian (notations)

- This result is not surprising, as velocity is the derivatives of position.
- We can perform this derivation in a systematic way.
- First we introduce a generalized joint coordinate.
- We knew:

$$q_i = \begin{cases} \theta_i & \text{revolute} \\ d_i & \text{prismatic} \end{cases}$$

R
P

- Combine both:

$$q_i = \bar{\varepsilon}_i \theta_i + \varepsilon_i d_i$$

  - where

$$\varepsilon_i = \begin{cases} 0 & \text{revolute} \\ 1 & \text{prismatic} \end{cases}$$

R
P

$$\bar{\varepsilon}_i = 1 - \varepsilon_i$$

$$\bar{\varepsilon}_i = \begin{cases} 1 & \text{revolute} \\ 0 & \text{prismatic} \end{cases}$$

q

Jacobian – Direct Differentiation

- Using this notation, the joint coordinate vector is: $q = [q_1, q_2, \dots q_n]^T$
- The Cartesian position of the tip of the arm is therefore:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f_x(q) \\ f_y(q) \\ f_z(q) \end{bmatrix} \quad \text{or} \quad X = f(q)$$

- Differentiation gives the Cartesian velocity of the tip of the arm:

$$\underbrace{\begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix}}_{m\times 1} = \underbrace{\begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \cdots & \frac{\partial f_x}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_z}{\partial q_1} & \cdots & \frac{\partial f_z}{\partial q_n} \end{bmatrix}}_{m\times n} \underbrace{\begin{bmatrix} \delta q_1 \\ \vdots \\ \delta q_n \end{bmatrix}}_{n\times 1}$$

or

$$\dot{x} = J(q)\dot{q}$$

Jacobian

- Summary: Given $\dot{q}$, we can calculate $\dot{x}$.

Jacobian – Direct Differentiation

- Using this notation, the joint coordinate vector is: $q = [q_1, q_2, \dots, q_n]^T$
- The Cartesian position of the tip of the arm is therefore:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f_x(q) \\ f_y(q) \\ f_z(q) \end{bmatrix} \quad \text{or} \quad x = f(q)$$

- Differentiation gives the Cartesian velocity of the tip of the arm:

$$J \rightarrow \underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}}_{m\times 1} = \underbrace{\begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \cdots & \frac{\partial f_x}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_z}{\partial q_1} & \cdots & \frac{\partial f_z}{\partial q_n} \end{bmatrix}}_{m\times n} \underbrace{\begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}}_{n\times 1} \quad \text{or} \quad \dot{x} = J(q)\dot{q}$$

Jacobian

- Summary: Given $\dot{q}$, we can calculate $\dot{x}$.

Jacobian – Notes:

- To indicate the frame in which the Cartesian velocity is expressed, we use leading superscripts:

$$ {}^{0}v = {}^{0}J(q)\dot{q} $$

^{0}v

  - This superscript may be ignored if the frame is obvious.

- Also, remember that the Jacobian is linear but time varying.
  - This means, the joint rates are related to the velocity of the tip in a linear manner, but this relationship is only instantaneous.
  - At the next instant, we have a new Jacobian and new relationship. $J(q)$

- Direct Differentiation is suitable for linear velocity, but not for rotational velocity.

Changing Reference Frame

- Given a Jacobian written in frame {B}:

$$\begin{bmatrix} {}^{B}\upsilon \\ {}^{B}\omega \end{bmatrix} = {}^{B}v = {}^{B}J(q)\dot{q}$$

- We can transform it to frame {A}:

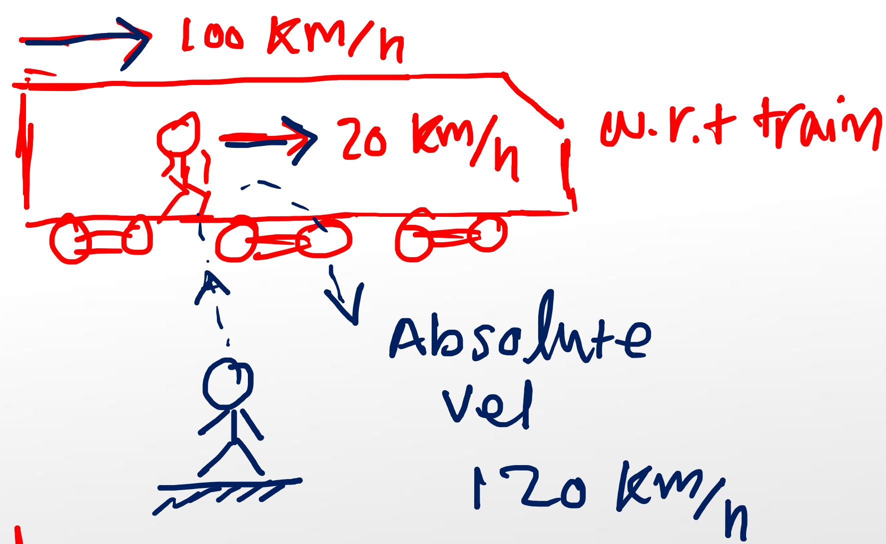
$$\begin{aligned} \begin{bmatrix} {}^{A}\upsilon \\ {}^{A}\omega \end{bmatrix} &= \begin{bmatrix} {}^{A}_{B}R & 0 \\ 0 & {}^{A}_{B}R \end{bmatrix} \begin{bmatrix} {}^{B}\upsilon \\ {}^{B}\omega \end{bmatrix} \\ &= \begin{bmatrix} {}^{A}_{B}R & 0 \\ 0 & {}^{A}_{B}R \end{bmatrix} \cdot {}^{B}J(q)\dot{q} \\ &= {}^{A}J(q)\dot{q} \end{aligned}$$

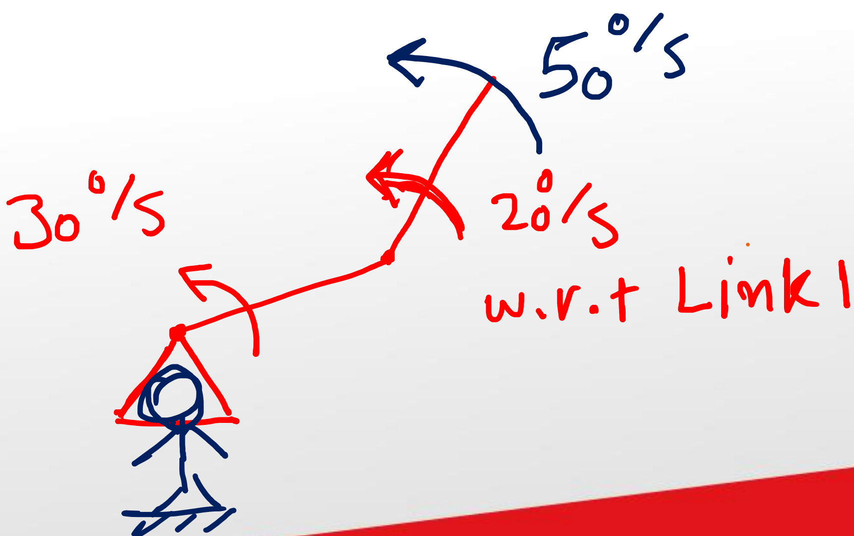
- Therefore:

$$\Rightarrow \quad {}^{A}J(q) = \begin{bmatrix} {}^{A}_{B}R & 0 \\ 0 & {}^{A}_{B}R \end{bmatrix} \cdot {}^{B}J(q)$$

Content

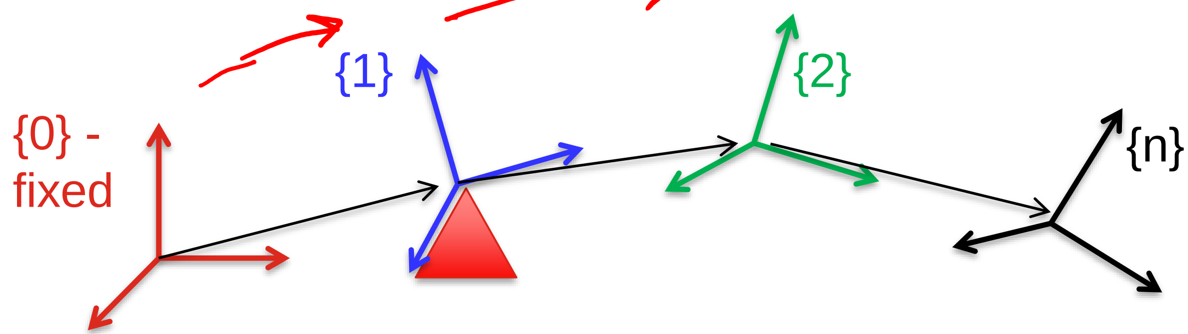
- Introduction - Jacobian
- Method 1 - Direct differentiation (for Linear Jacobian)
- Method 2 - Velocity Propagation from Link to Link → { linear vel, Rotational vel }
- Method 3 – Explicit Form (for Rotational Jacobian)
- Singularities
- Static Forces in Manipulators
- Resolved Motion Rate Control





Velocity Propagation

- For robots, we always use frame {0} as the reference frame.
- The robotic manipulator is a chain of bodies, each capable of moving relative to its neighbour.



- Therefore, the velocity of each link can be computed in order, starting from the base.

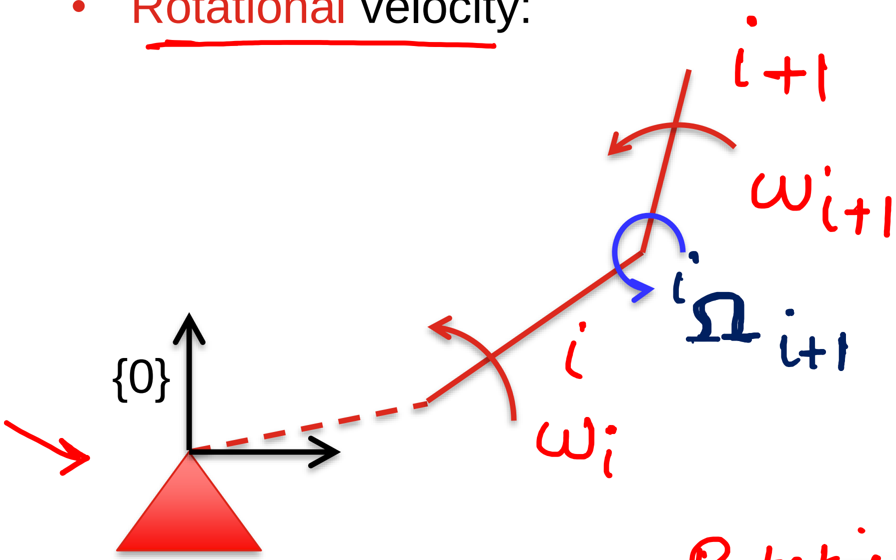
Velocity

- Velocity of link i+1 = Velocity of link i + new components due to joint i+1

- Note: “Velocity of a link”: v_i is the linear velocity of the origin of link frame and ω_i is the rotational velocity of the link.

Notations

- We need to differentiate between relative velocity and absolute velocity.
- Rotational velocity:



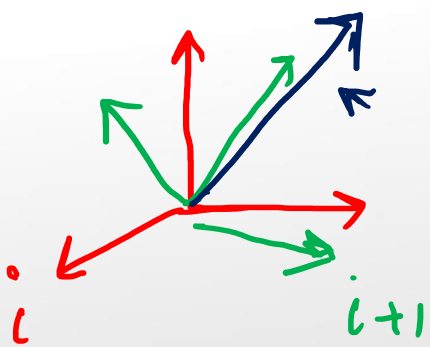
- ω_i, ω_{i+1}: "Absolute" velocity wrt. frame {0}.
- Ω_{i+1}: Angular speed of joint i+I wrt joint i, "relative".

How to get Ω & ω

Rotation from {i} to {i+1} = $^{i}_{i+1}R$

Angular vel from {i+1} to {i} = $^{i}\Omega_{i+1}$

3x1 vector



Magnitude = Speed

direction = instantaneous axis of Rot.

Notations

- How to get ω and Ω?
- We knew: Relative rotation from frame {i} to {i+1}:

$$ {}^{i}_{i+1}R $$

- Relative angular velocity of {i+1} wrt {i} is written as:

$$ {}^{i}\Omega_{i+1} $$

  - This is a 3x1 vector.
  - Direction = instantaneous axis of rotation
  - Magnitude = speed

Notations

$${}^{i}\Omega_{i+1} \xrightarrow{{}^{A}_{i}R} {}^{A}\Omega_{i+1} = {}^{A}_{i}R \; {}^{i}\Omega_{i+1}$$

- We can change the reference frame of expression using the rotation matrix.
- E.g. Relative angular velocity of {i+1} wrt {i}, expressed in frame {i+1} is:

$${}^{i+1}\Omega_{i+1} = {}^{i+1}_{i}R \cdot {}^{i}\Omega_{i+1}$$

- If we now change the reference frame of expression to frame {0}, we then get the absolute velocity. ω_{i+1} ?

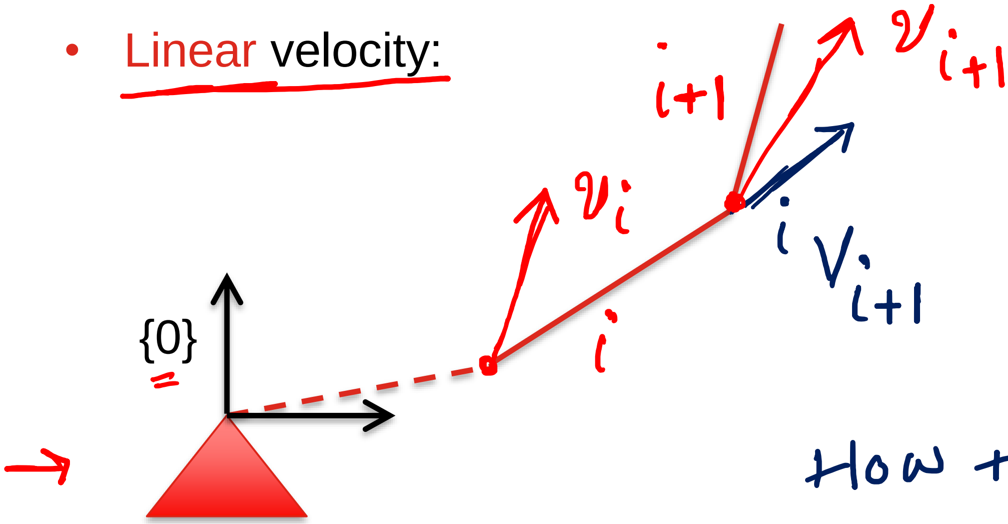
$${}^{0}\Omega_{i+1} \triangleq \omega_{i+1} = {}^{0}_{i}R \; {}^{i}\Omega_{i+1}$$

- Even the absolute velocity can have a change of reference frame, e.g.

$${}^{i+1}\omega_{i+1} = {}^{i+1}_{0}R \; \omega_{i+1} = {}^{i+1}_{0}R \; {}^{0}\Omega_{i+1}$$

Notations

- Linear velocity:

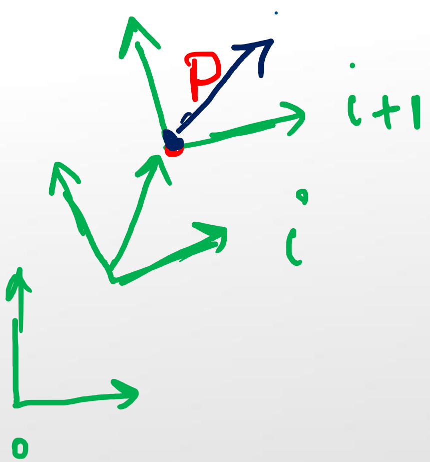


v

- v_i,v_{i+1}: “Absolute” velocity wrt. frame {0}.
- V_{i+1}: Linear speed of joint i+I wrt joint i, “relative”.

V

How to get v & V



Position of frame {i+1} w.r.t {i} $\quad {}^{i}P_{(i+1)ORG}$

vel of frame {i+1} w.r.t {i} $\quad {}^{i}V_{(i+1)ORG}$

Notations

$$ {}^{i}V_{i+1} \xrightarrow{{}^{A}_{i}R} {}^{A}V_{i+1} = {}^{A}_{i}R \cdot {}^{i}V_{i+1} $$

- We can change the reference frame of expression using the rotation matrix.
- E.g. Relative linear velocity of {i+1} wrt {i}, expressed in frame {i+1} is:

$$ {}^{i+1}V_{i+1} = {}^{i+1}_{i}R \cdot {}^{i}V_{i+1} $$

- If we now change the reference frame of expression to frame {0}, we then get the absolute velocity. v_{i+1}

$$ {}^{0}V_{i+1} = {}^{0}_{i}R \cdot {}^{i}V_{i+1} \triangleq v_{i+1} $$

- Even the absolute velocity can have a change of reference frame, e.g.

$$ {}^{i+1}v_{i+1} = {}^{i+1}_{0}R \, v_{i+1} = {}^{i+1}_{0}R \, {}^{0}V_{i+1} $$

1

Notations

- We can change the reference frame of expression using the rotation matrix.
- E.g. Relative linear velocity of {i+1} wrt {i}, expressed in frame {i+1} is:

$$ {}^{i+1}V_{i+1} = {}^{i+1}_{i}R \; {}^{i}V_{i+1} $$

- If we now change the reference frame of expression to frame {0}, we then get the absolute velocity.

$$ {}^{0}V_{i+1} = {}^{0}_{i}R \; {}^{i}V_{i+1} \triangleq v_{i+1} $$

- Even the absolute velocity can have a change of reference frame, e.g.

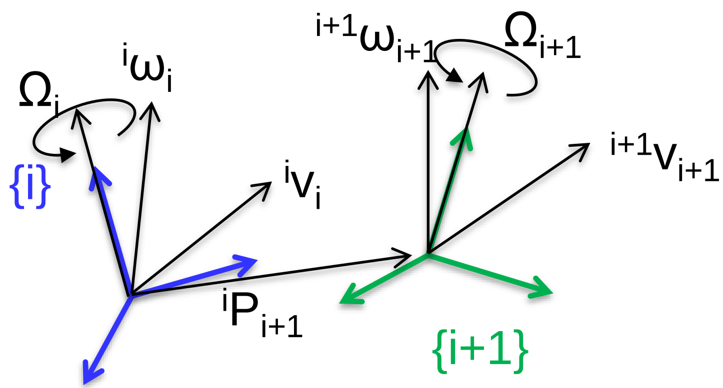
$$ {}^{i+1}v_{i+1} = {}^{i+1}\left({}^{0}V_{i+1} \right) = {}^{i+1}_{0}R \; {}^{0}V_{i+1} \triangleq {}^{i+1}_{0}R \cdot v_{i+1} $$

Sumarry

	Relative	Absolute
Rotational Velocity	${}^{i}\Omega_{i+1}$	${}^{0}\Omega_{i+1} \triangleq \omega_{i+1}$
Linear Velocity	${}^{i}V_{i+1} = {}^{i}V_{(i+1)ORG}$	${}^{0}V_{i+1} \triangleq v_{i+1}$

Velocity Propagation

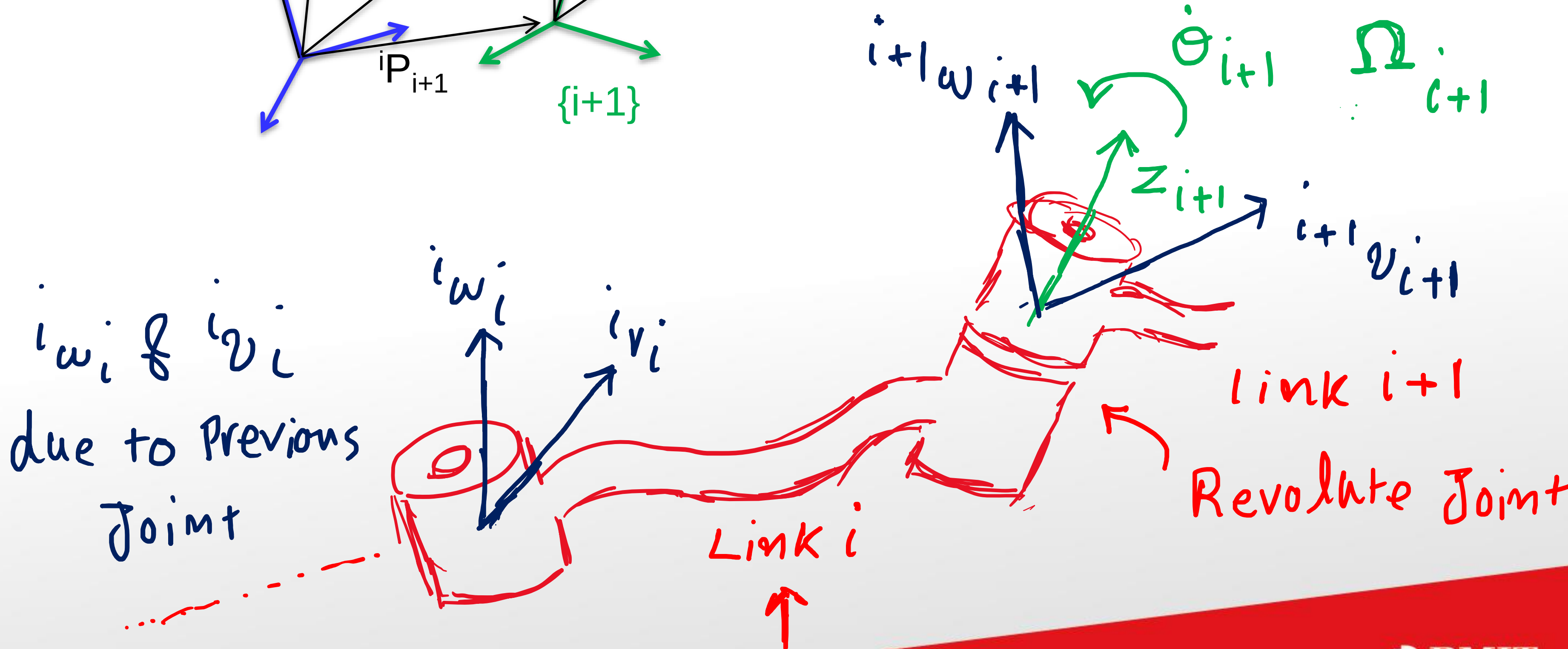
- Consider links i and i+1, together with their velocities.



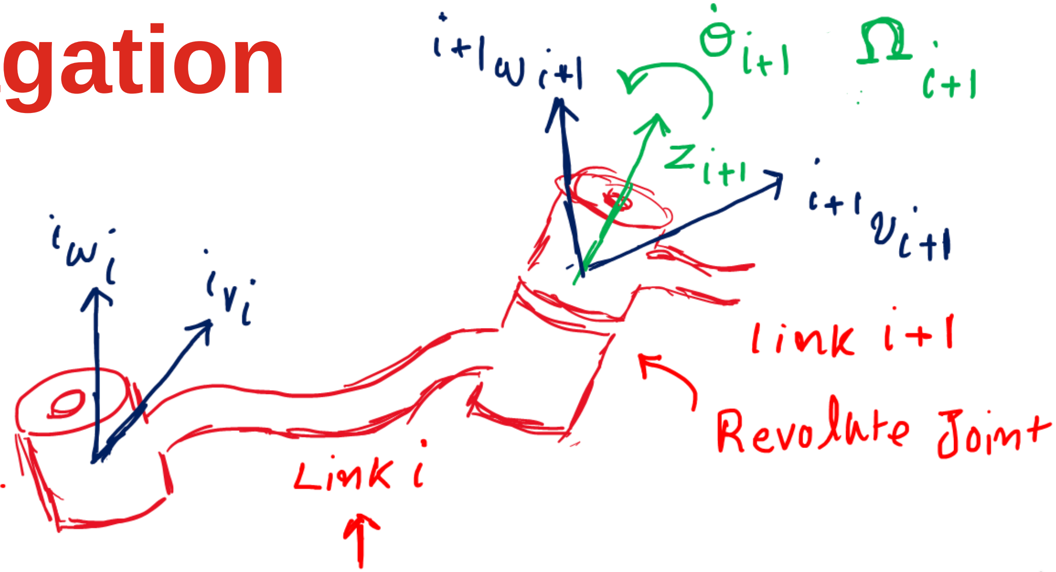
Velocity of link i+1 =

Velocity of link i +

new components due to joint i+1



Velocity Propagation



- Rotational velocities can be added, if both ω vectors are expressed in the same frame. Therefore:

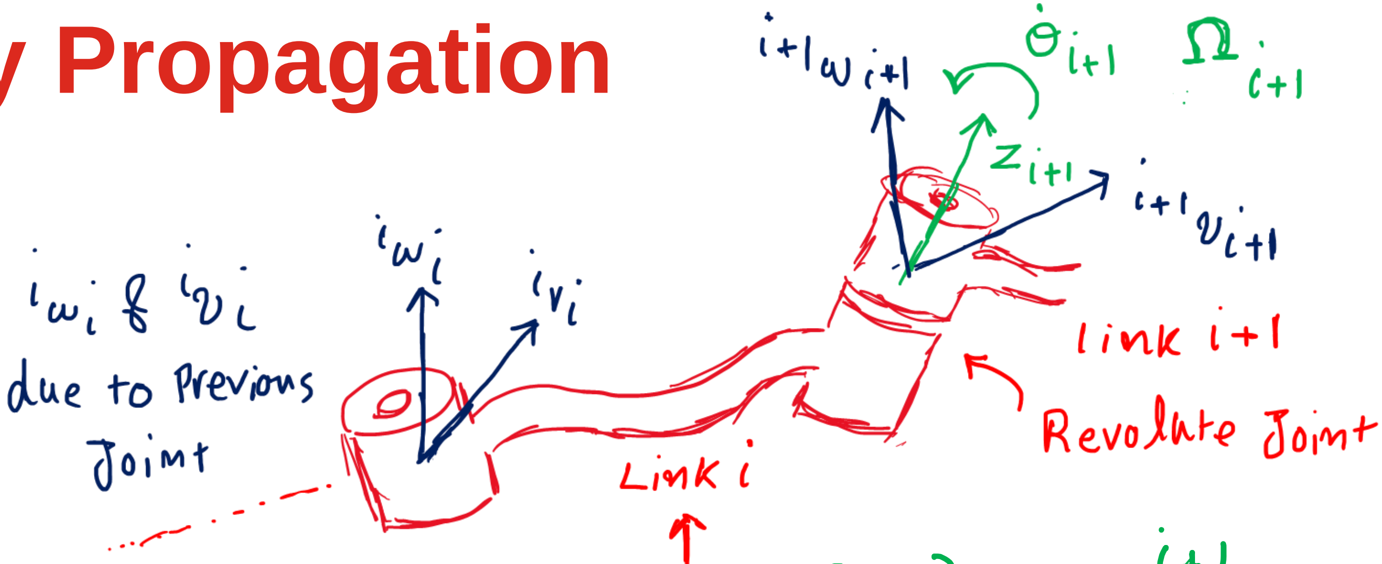
$$^{i}\omega_{i+1} = {}^{i}\omega_i + {}^{i}\Omega_{i+1}$$

However, writing in frame {i+1} is better

- Pre-multiply every term with ${}^{i+1}_{i}R$ gives:

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R \cdot {}^{i}\omega_i + {}^{i+1}\Omega_{i+1}$$

Velocity Propagation



$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{\ \ i}R \cdot {}^{i}\omega_i + {}^{i+1}\Omega_{i+1} $$

in {i+1}, the ${}^{i+1}\Omega_{i+1}$ is always in Z-axis

in other words

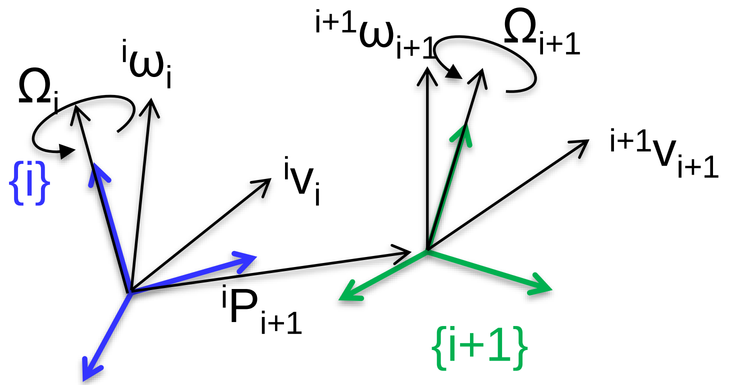
$$ {}^{i+1}\Omega_{i+1} = \dot\theta_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} $$

$$ = \dot\theta_{i+1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} $$

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{\ \ i}R \cdot {}^{i}\omega_i + \dot\theta_{i+1}\, {}^{i+1}\hat{Z}_{i+1} $$

Rot. vel (Revolute Joint)

Velocity Propagation (Rot. Vel)



- Rotational velocities can be added, if both ω vectors are expressed in the same frame. Therefore:

$$\begin{aligned} {}^{i}\omega_{i+1} &= {}^{i}\omega_i + {}^{i}_{i+1}R \cdot {}^{i+1}\Omega_{i+1} \\ &= {}^{i}\omega_i + {}^{i}_{i+1}R \cdot \dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} \end{aligned}$$

where

$$\dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1} = \begin{bmatrix} 0 \\ 0 \\ {}^{i+1}\dot{\theta}_{i+1} \end{bmatrix}$$

scalar value, thus no frame

vector

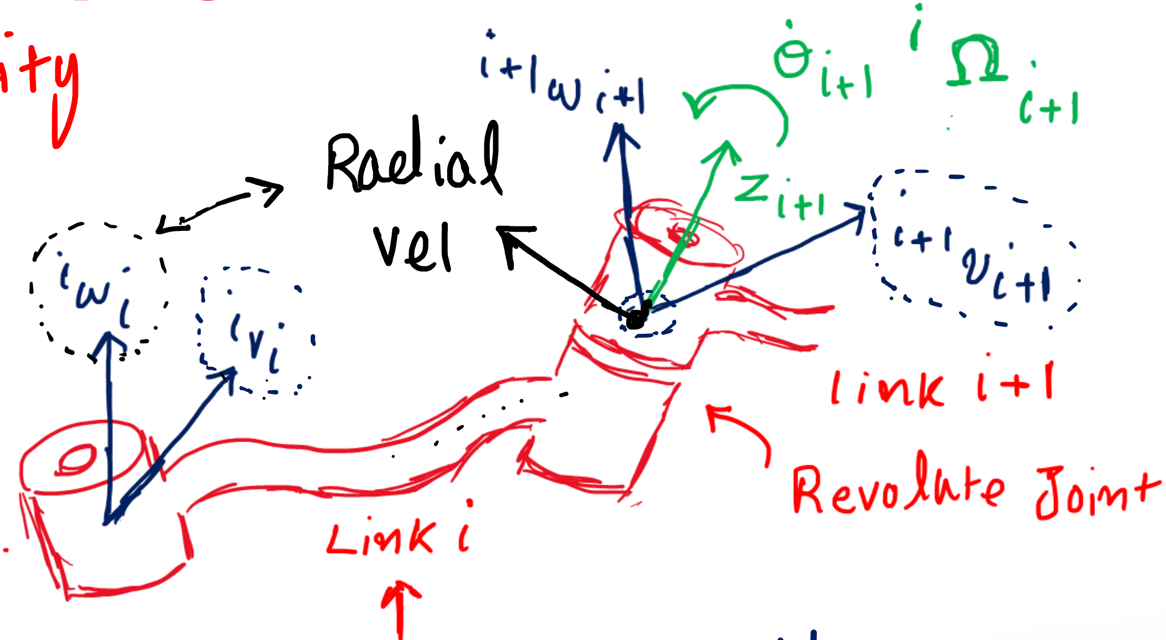
- Pre-multiply every term with $^{i+1}{}_{i}R$ gives:

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R \cdot {}^{i}\omega_i + \dot{\theta}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1}$$

Velocity Propagation

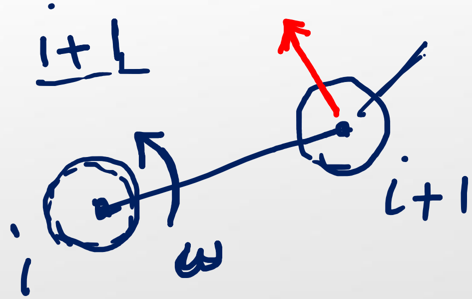
Linear Velocity

$^{i}\omega_i$ & $^{i}v_i$ due to Previous Joint



The rotation of joint i+1 wouldn't add any new linear velocity to the joint i+1

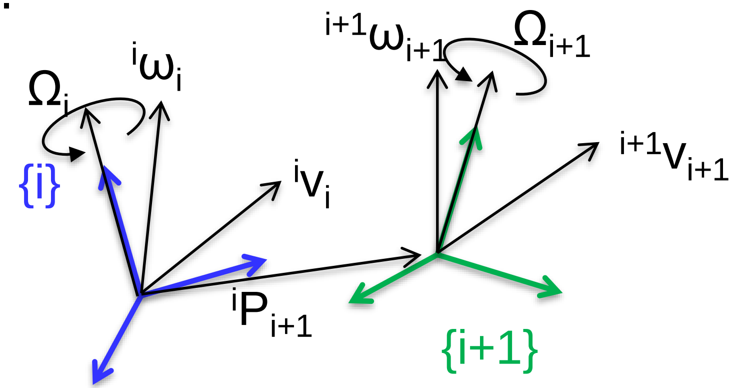
However, the rotation of joint i creates a "Radial" velocity for joint i+1



Radial vel.

Velocity Propagation (Linear Vel.)

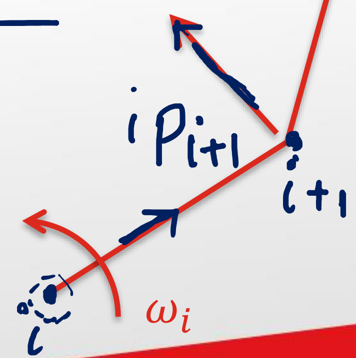
- As for linear velocities, if the joint is revolute, then there is no linear velocity between the two frames.



- The linear velocity of the origin of {i+1} = linear velocity of origin of {i} + new component caused by rotational velocity of link i.

$$^{i}v_{i+1} = {}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}$$

Radial linear vel caused by rotation

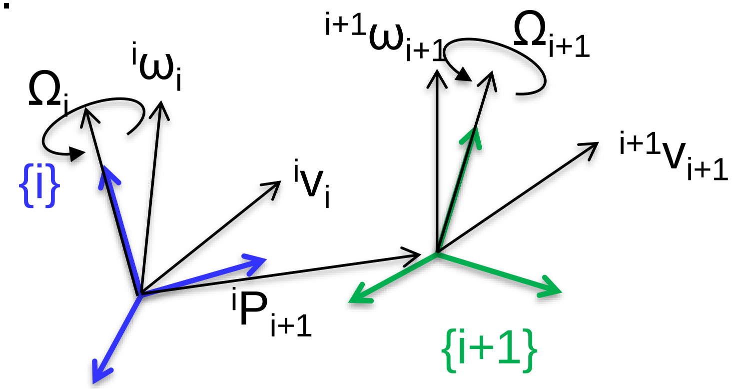


- Pre-multiply every term with ${}^{i+1}_{i}R$ gives:

$$^{i+1}v_{i+1} = {}^{i+1}_{i}R\,[{}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}]$$

Velocity Propagation (Linear Vel.)

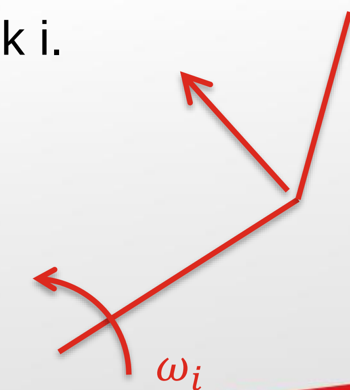
- As for linear velocities, if the joint is revolute, then there is no linear velocity between the two frames.



- The linear velocity of the origin of {i+1} = linear velocity of origin of {i}

  + new component caused by rotational velocity of link i.

$$ {}^{i}\upsilon_{i+1} = {}^{i}\upsilon_i + \underbrace{{}^{i}\omega_i \times {}^{i}P_{i+1}} $$



Radial linear vel caused by rotation

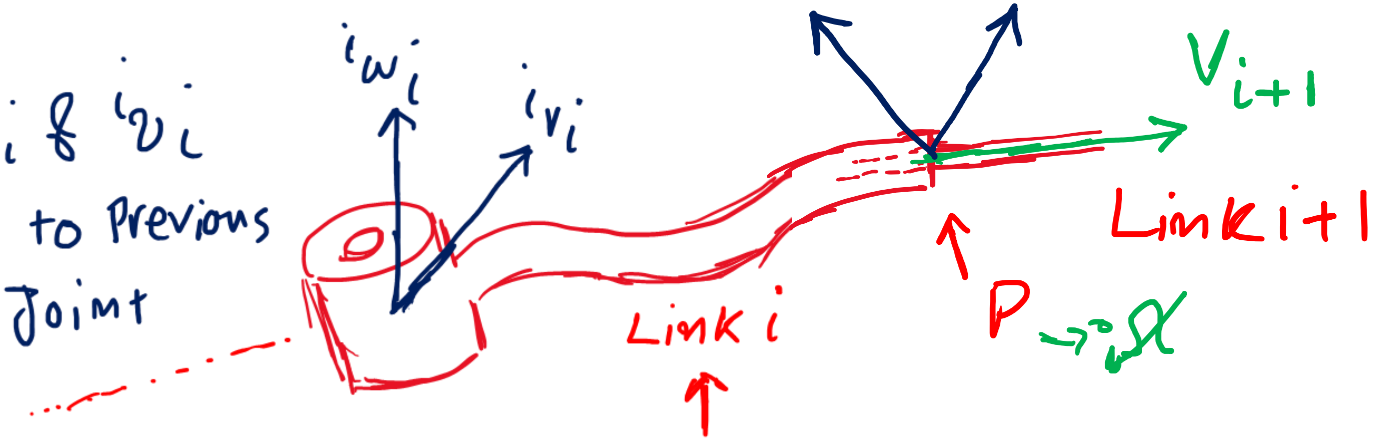
- Pre-multiply every term with ${}^{i+1}_{i}R$ gives:

$$ {}^{i+1}\upsilon_{i+1} = {}^{i+1}_{i}R \cdot \left({}^{i}\upsilon_i + {}^{i}\omega_i \times {}^{i}P_{i+1} \right) $$

Velocity Propagation

→ Prismatic Joint

${}^{i}\dot{\omega}_i$ & ${}^{i}\dot{v}_i$
due to Previous
Joint

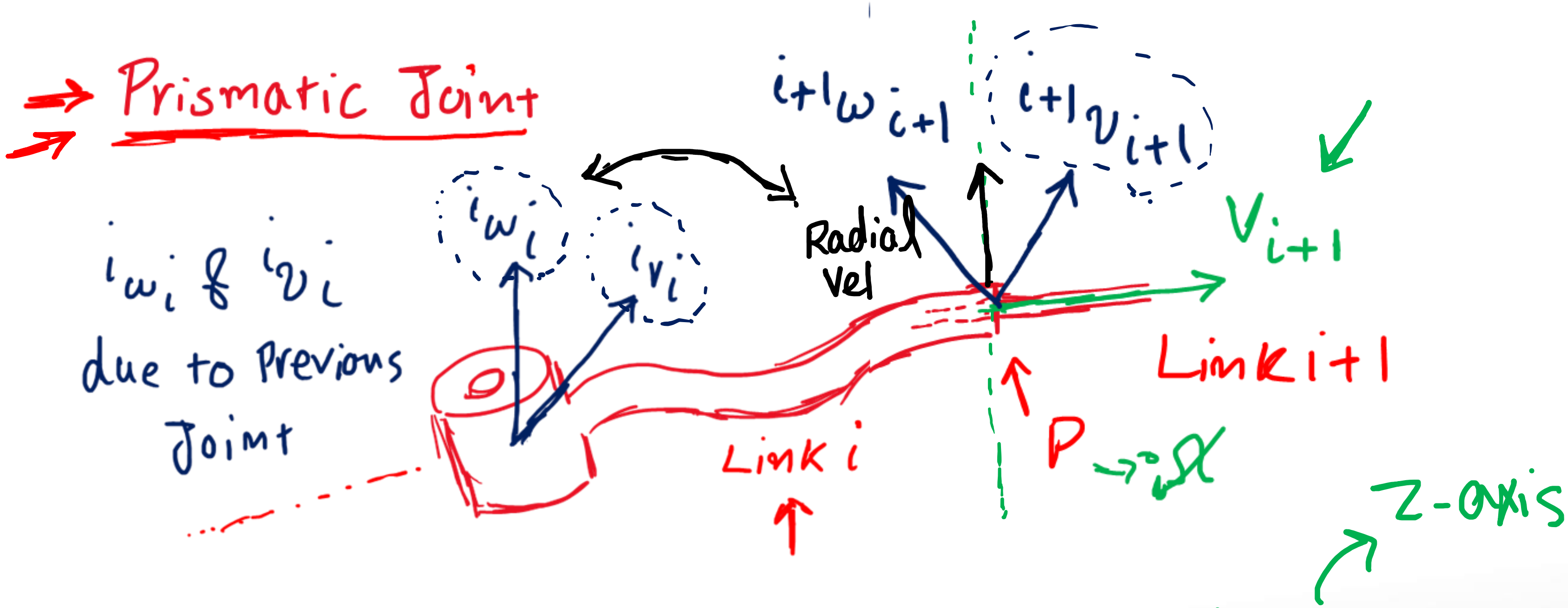


Rot. velocity : Prismatic Joint doesn't Rotate

thus $\quad {}^{i}\dot{\omega}_{i+1} = {}^{i}\dot{\omega}_i$ $\quad {}^{i+1}_{i}R$

Express in {i+1} $\quad {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R \cdot {}^{i}\dot{\omega}_i$

Velocity Propagation



Linear Velocity:

$$ {}^{i+1}\dot{v}_{i+1} = \underbrace{{}^{i+1}_{i}R\left({}^{i}\dot{v}_i + {}^{i}\dot{\omega}_i \times {}^{i}P_{i+1}\right)}_{\text{linear Velocities caused before the P. Joint}} + \overbrace{\ddot{d}_{i+1} \cdot {}^{i+1}\hat{Z}_{i+1}}^{{}^{i+1}\dot{V}_{i+1}} $$

Velocity Propagation - Algorithm

1 • Start with $^{0}\omega_0, ^{0}\upsilon_0$ $^{0}\omega_0 , ^{0}v_0$

2 • Calculate recursively from link to link:

- For revolute joints: $^{i+1}\omega_{i+1} = ^{i+1}_{i}R \cdot ^{i}\omega_i + \dot{\theta}_{i+1} \cdot ^{i+1}\hat{Z}_{i+1}$ Rot.

R

$$^{i+1}\upsilon_{i+1} = ^{i+1}_{i}R \cdot \left(^{i}\upsilon_i + ^{i}\omega_i \times ^{i}P_{i+1}\right)$$ Linear

- For prismatic joints: $^{i+1}\omega_{i+1} = ^{i+1}_{i}R \cdot ^{i}\omega_i$ ← Rot.

P

$$^{i+1}\upsilon_{i+1} = ^{i+1}_{i}R \cdot \left(^{i}\upsilon_i + ^{i}\omega_i \times ^{i}P_{i+1}\right) + \dot{d}_{i+1} \cdot ^{i+1}\hat{Z}_{i+1}$$ Linear

untill Join n

3 • Finally, we will arrive at $^{n}\omega_n, ^{n}\upsilon_n$. $^{n}\omega_n , ^{n}v_n$

4 • Transform these back to base frame by:

$$\begin{bmatrix} ^{0}\upsilon_n \\ ^{0}\omega_n \end{bmatrix} = \begin{bmatrix} ^{0}_{n}R & 0 \\ 0 & ^{0}_{n}R \end{bmatrix} \begin{bmatrix} ^{n}\upsilon_n \\ ^{n}\omega_n \end{bmatrix}$$

{0}

⇒ Velocity Propagation - Algorithm

1 • Start with $^{0}\omega_0, ^{0}\upsilon_0$ $^{0}\omega_0, ^{0}\upsilon_0$

2 • Calculate recursively from link to link:

- For revolute joints: $^{i+1}\omega_{i+1} = ^{i+1}_{i}R \cdot ^{i}\omega_i + \dot{\theta}_{i+1} \cdot ^{i+1}\hat{Z}_{i+1}$ Eq. 1

R

$$^{i+1}\upsilon_{i+1} = ^{i+1}_{i}R \cdot \left(^{i}\upsilon_i + ^{i}\omega_i \times ^{i}P_{i+1}\right)$$ Eq. 2

- For prismatic joints: $^{i+1}\omega_{i+1} = ^{i+1}_{i}R \cdot ^{i}\omega_i$ Eq. 3

P

$$^{i+1}\upsilon_{i+1} = ^{i+1}_{i}R \cdot \left(^{i}\upsilon_i + ^{i}\omega_i \times ^{i}P_{i+1}\right) + \dot{d}_{i+1} \cdot ^{i+1}\hat{Z}_{i+1}$$ Eq. 4

Untill joint n

3 • Finally, we will arrive at $^{n}\omega_n, ^{n}\upsilon_n$. $^{n}\omega_n, ^{n}\upsilon_n$

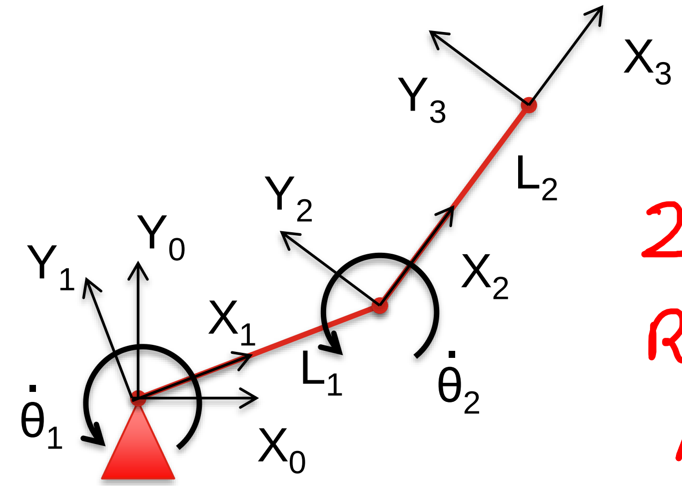
4 • Transform these back to base frame by: {0}

$$\begin{bmatrix} ^{0}\upsilon_n \\ ^{0}\omega_n \end{bmatrix} = \begin{bmatrix} ^{0}_{n}R & 0 \\ 0 & ^{0}_{n}R \end{bmatrix} \begin{bmatrix} ^{n}\upsilon_n \\ ^{n}\omega_n \end{bmatrix}$$

3

Example

- Given a two link robot as shown.
- What is the velocity at its tip?



2-Link Robotic Arm

- Because the formulae needs the transformation matrices, they are computed first.

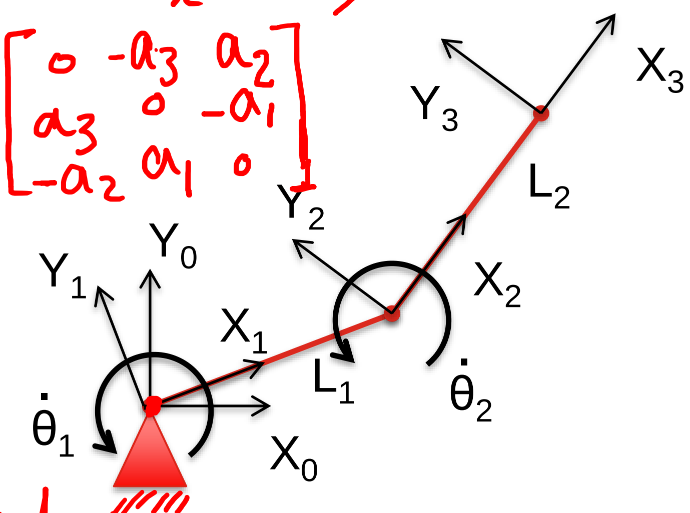
${}^0_1R$ ${}^1_2R$ ${}^2_3R$

$$ {}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1_2T = \begin{bmatrix} c_2 & -s_2 & 0 & L_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^2_3T = \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Example

$a \times b = [a]_x b$

$[a]_x = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$ → skew-symmetric



- Next, we apply the iterative formulae:

1 • Start with: $\{0\}$

$$ {}^0\omega_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad {}^0\upsilon_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} $$

Fixed

2 • Then: Joint1, R, Eq.1

$^1\upsilon_1$: Joint1, R, Eq.2

$$ {}^1\omega_1 = {}^1_0R \cdot {}^0\omega_0 + \dot\theta_1 \cdot {}^1\hat Z_1 = {}^0_1R^T \cdot {}^0\omega_0 + \dot\theta_1 \cdot {}^1\hat Z_1 = \begin{bmatrix} & & \\ & \cdot & \\ & & \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 \end{bmatrix} $$

$$ {}^1\upsilon_1 = {}^1_0R \cdot \left({}^0\upsilon_0 + {}^0\omega_0 \times {}^0P_1\right) = {}^0_1R^T \cdot \left({}^0\upsilon_0 + {}^0\omega_0 \times {}^0P_1\right) $$

$$ = \begin{bmatrix} & & \\ & {}^1_0R & \\ & & \end{bmatrix} \cdot \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & -{}^0\omega_{0z} & {}^0\omega_{0y} \\ {}^0\omega_{0z} & 0 & -{}^0\omega_{0x} \\ -{}^0\omega_{0y} & {}^0\omega_{0x} & 0 \end{bmatrix}}_{[{}^0\omega_0]_x} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) $$

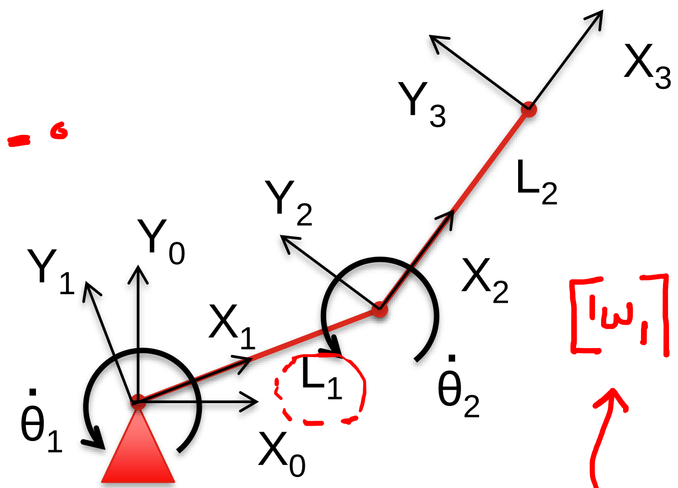
$$ = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = 0 $$

Example

- Next we have:

$${}^1\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 \end{bmatrix} \qquad {}^1v_1 = 0$$

${}^1\omega_{1z}$



$[{}^1\omega_1]_\times \, {}^1P_2$

Step 2 : Link 2

set i=1

${}^2\omega_2$: Joint 2, R, Eq.1

$$
\begin{aligned}
{}^2\omega_2 &= {}^2_1R \cdot {}^1\omega_1 + \dot\theta_2 \cdot {}^2\hat{Z}_2 \\
&= {}^1_2R^T \cdot {}^1\omega_1 + \dot\theta_2 \cdot {}^2\hat{Z}_2 \\
&= \begin{bmatrix} C_2 & S_2 & 0 \\ -S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot\theta_2 \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 + \dot\theta_2 \end{bmatrix}
\end{aligned}
$$

Joint 2, R, Eq.2

$$
\begin{aligned}
{}^2\upsilon_2 &= {}^2_1R \cdot \left({}^1\upsilon_1 + {}^1\omega_1 \times {}^1P_2\right) = {}^1_2R^T \cdot \left(\cancel{{}^1\upsilon_1}^{\,0} + {}^1\omega_1 \times {}^1P_2\right) \\
&= \begin{bmatrix} C_2 & S_2 & 0 \\ -S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -{}^1\omega_{1z} & {}^1\omega_{1y} \\ {}^1\omega_{1z} & 0 & -{}^1\omega_{1x} \\ -{}^1\omega_{1y} & {}^1\omega_{1x} & 0 \end{bmatrix} \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix} \right) \\
&= \begin{bmatrix} & \downarrow & \\ & // & \\ & & \end{bmatrix} \begin{bmatrix} 0 \\ L_1\,{}^1\omega_{1z} \\ -L_1\,\cancel{{}^1\omega_{1y}}^{\,0} \end{bmatrix} = \begin{bmatrix} L_1\,{}^1\omega_{1z} S_2 \\ L_1\,{}^1\omega_{1z} C_2 \\ 0 \end{bmatrix} = \begin{bmatrix} L_1 S_2 \dot\theta_1 \\ L_1 C_2 \dot\theta_1 \\ 0 \end{bmatrix}
\end{aligned}
$$

Example

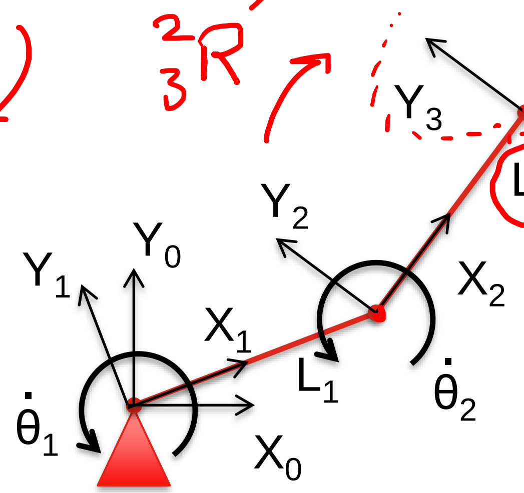
2-Link Robot

$^2\omega_2$ & $^2v_2$ Previous step

Step 3

- If the end-effector is of interest, we can propagate the velocity from the last link to the end-effector as well:

$^2_3R \to I$



{3}

No R. or P. Joint

$[{}^2\omega_2]_\times \, {}^2P_3$

Eq.1

$${}^3\omega_3 = {}^3_2R \cdot {}^2\omega_2 + \dot{\theta}_3 \cdot {}^3\hat{Z}_3$$

$$= {}^2_3R^T \, {}^2\omega_2 + \dot{\theta}_3 \cdot {}^3\hat{Z}_3$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}$$

$$= {}^2\omega_2$$

Eq.2

$${}^3\upsilon_3 = {}^3_2R \cdot \left({}^2\upsilon_2 + {}^2\omega_2 \times {}^2P_3 \right) = {}^2_3R^T \cdot \left({}^2\upsilon_2 + {}^2\omega_2 \times {}^2P_3 \right)$$

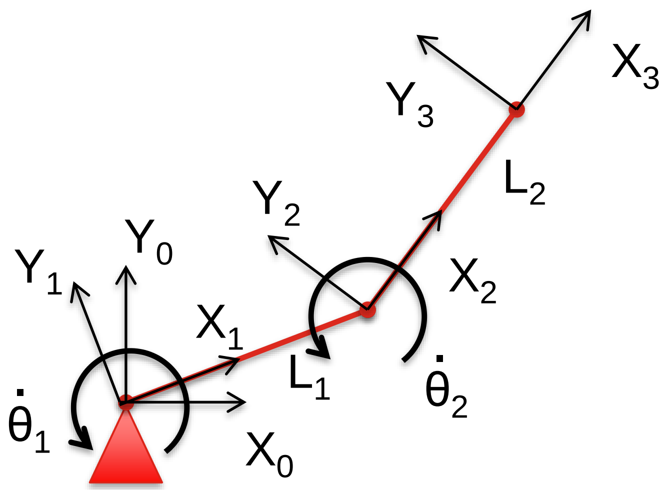
$$= \begin{bmatrix} & & \\ & I & \\ & & \end{bmatrix} \cdot \left(\begin{bmatrix} L_1 S_2 \dot{\theta}_1 \\ L_1 C_2 \dot{\theta}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -{}^2\omega_{2z} & {}^2\omega_{2y} \\ {}^2\omega_{2z} & 0 & -{}^2\omega_{2x} \\ -{}^2\omega_{2y} & {}^2\omega_{2x} & 0 \end{bmatrix} \begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix} \right)$$

$$= \begin{bmatrix} L_1 S_2 \dot{\theta}_1 \\ L_1 C_2 \dot{\theta}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ L_2 \, {}^2\omega_{2z} \\ -L_2 \, {}^2\omega_{2y} \end{bmatrix} = \begin{bmatrix} L_1 S_2 \dot{\theta}_1 \\ L_1 C_2 \dot{\theta}_1 + L_2 (\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix}$$

$\dot{\theta}_1 + \dot{\theta}_2 \to {}^2\omega_{2z}$

Example

Step 4



- The last step is to transform these back to the base frame.
- We need: ${}^3v_3, {}^3\omega_3$

$$\begin{bmatrix} {}^0\upsilon_n \\ {}^0\omega_n \end{bmatrix} = \begin{bmatrix} {}^0_nR & 0 \\ 0 & {}^0_nR \end{bmatrix} \begin{bmatrix} {}^n\upsilon_n \\ {}^n\omega_n \end{bmatrix}$$

- where: ${}^0_3R = {}^0_1R \cdot {}^1_2R \cdot {}^2_3R$

$$= \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$ {}^0\omega_3 = \begin{bmatrix} C_{12} & -S_{12} & 0 \\ S_{12} & C_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1+\dot{\theta}_2 \end{bmatrix}}_{{}^3\omega_3} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1+\dot{\theta}_2 \end{bmatrix}$$

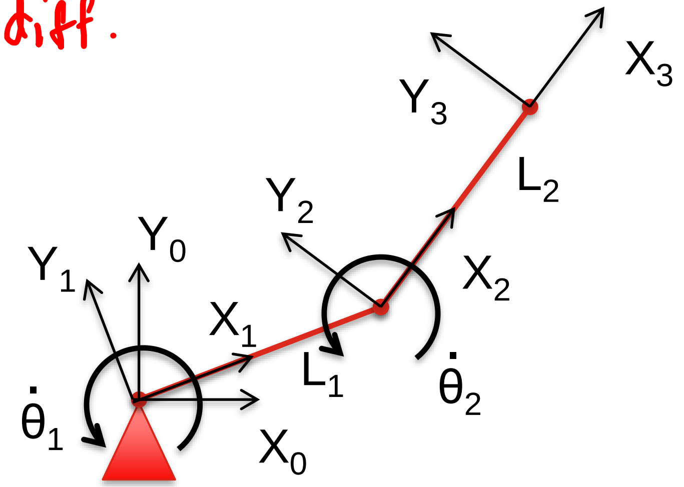
$$ {}^0v_3 = \begin{bmatrix} C_{12} & -S_{12} & 0 \\ S_{12} & C_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} L_1S_2\dot{\theta}_1 \\ L_1C_2\dot{\theta}_1 + L_2(\dot{\theta}_1+\dot{\theta}_2) \\ 0 \end{bmatrix}}_{{}^3v_3} = \begin{bmatrix} -L_1S_1\dot{\theta}_1 - L_2S_{12}\dot{\theta}_1 - L_2S_{12}\dot{\theta}_2 \\ L_1C_1\dot{\theta}_1 + L_2C_{12}\dot{\theta}_1 + L_2C_{12}\dot{\theta}_2 \\ 0 \end{bmatrix}$$

Example - Jacobian

diff.

- What about Jacobian Matrices:

$\left\{ \begin{array}{l} {}^0J_v \\ {}^0J_\omega \end{array} \right.$



$$0w_3 = \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 + \dot\theta_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot\theta_1 + \dot\theta_2 \end{bmatrix}$$

${}^0\omega_z$

$$0v_3 = \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} L_1 s_2 \dot\theta_1 \\ L_1 c_2 \dot\theta_1 + L_2(\dot\theta_1 + \dot\theta_2) \\ 0 \end{bmatrix} = \begin{bmatrix} -L_1 s_1 \dot\theta_1 - L_2 s_{12}(\dot\theta_1 + \dot\theta_2) \\ L_1 c_1 \dot\theta_1 + L_2 c_{12}(\dot\theta_1 + \dot\theta_2) \\ 0 \end{bmatrix}$$

${}^0J_v$: the same as direct diff. $\Rightarrow$

$\dot X = J \dot\theta$ ${}^0J_\omega$

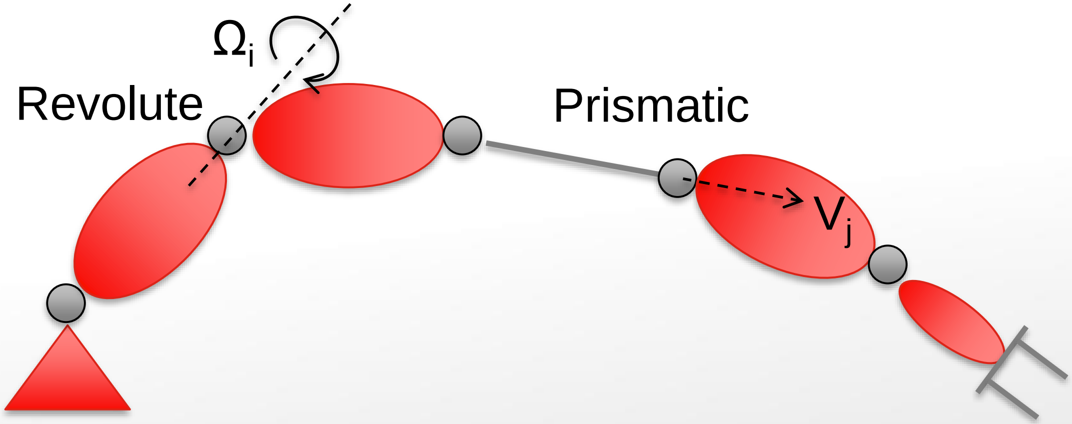
$${}^0J_\omega : \begin{bmatrix} {}^0\omega_x \\ {}^0\omega_y \\ {}^0\omega_z \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot\theta_1 \\ \dot\theta_2 \end{bmatrix}$$

Content

- Introduction - Jacobian
- Method 1 - Direct differentiation (for Linear Jacobian)
- Method 2 - Velocity Propagation from Link to Link
- Method 3 - Explicit Form (for your study, not included in exam)
- Static Forces in Manipulators
- Singularities

Explicit Form for Jacobian

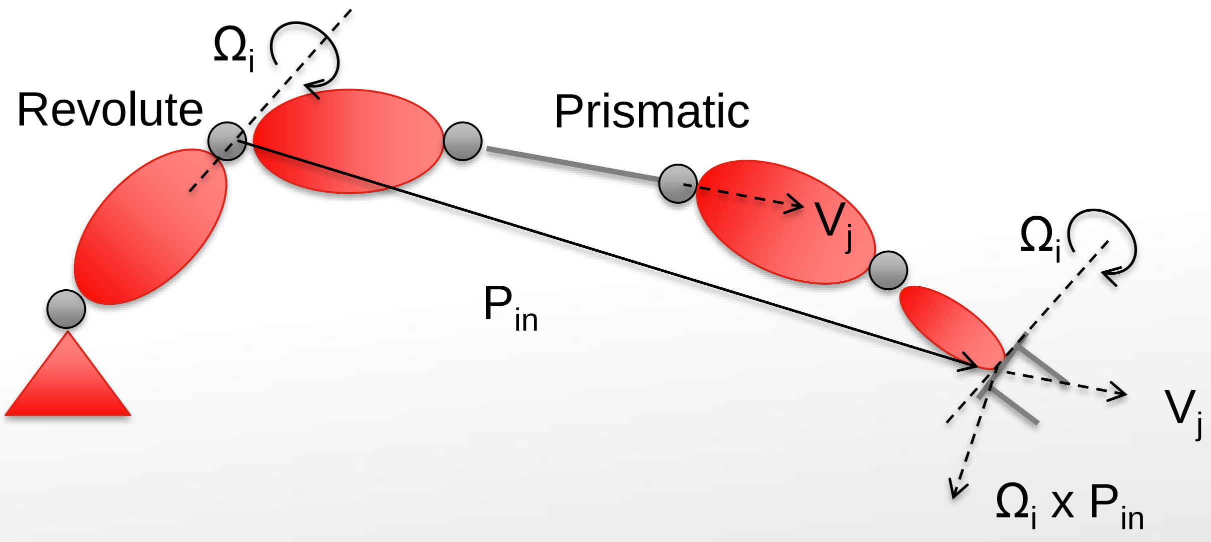
- We have learnt how to obtain the Jacobian matrix through:
  - Velocity propagation (linear and rotational).
  - Direct differentiation of position vector (linear).
- Is there another way to get the rotational Jacobian?
- Yes! We can "look at the mechanism" and directly see the rotational Jacobian!



  - We look at each joint axis, seeing whether it is revolute or prismatic, and analyze its impact on ω.
  - Finally, we map these into the Jacobian.

Explicit Form for Jacobian

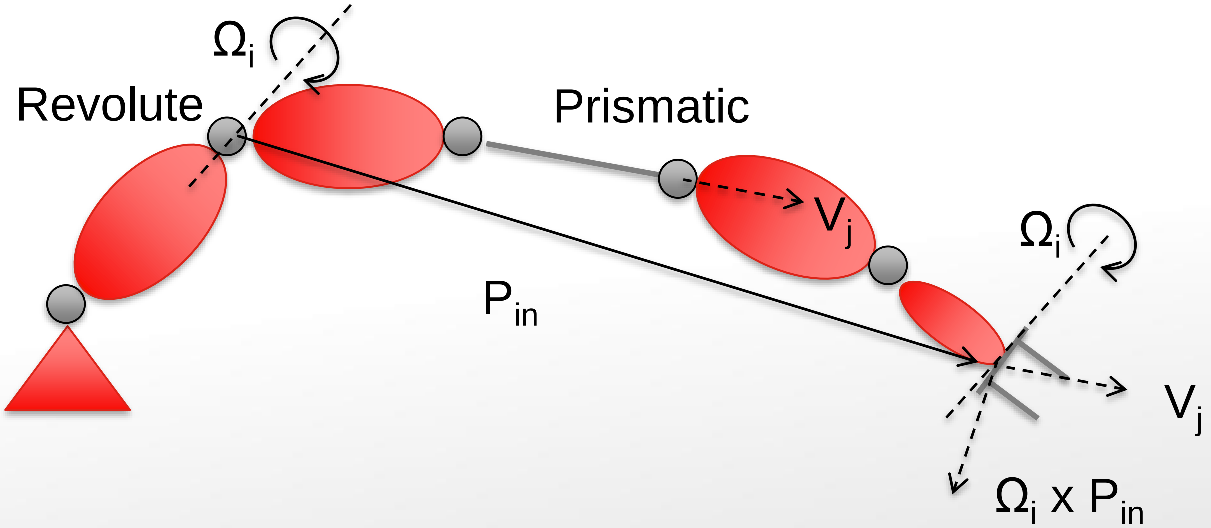
- We knew that:
  - Ω_i is generated by $\dot{\theta}_i$ along Z_i for revolute joint: $\Omega_i = Z_i \dot{q}_i$
  - V_j is generated by $\dot{d}_i$ along Z_i for prismatic joint: $V_i = Z_i \dot{q}_i$



Explicit Form for Jacobian

- Each (one) of these joints create the following motion to the end-effector:

End-Effector	Prismatic Joint	Revolute Joint
Linear Velocity	V_j	$\Omega_i \times P_{in}$
Angular Velocity	0	Ω_i



- And here, we only focus on the angular velocity.

Explicit Form for Jacobian

- Combining the effect of all joints, we have:
  - Angular velocity of end-effector:
    - If all joints are prismatic: $\omega = 0$
    - If all joints are revolute: $$\omega = \sum_{i=1}^{n} \bar{\varepsilon}_i \Omega_i$$
    - If mixed: $$\omega = \sum_{i=1}^{n} \bar{\varepsilon}_i \Omega_i$$

Explicit Form for Jacobian

- In summary:

$$\omega = \sum_{i=1}^{n} \bar{\varepsilon}_i \Omega_i$$

- We want to write these in terms of $\dot{q}$.
- Recall that: $\Omega_i = Z_i \dot{q}_i$
- Therefore:

$$\Rightarrow \quad \omega = \sum_{i=1}^{n} \bar{\varepsilon}_i Z_i \dot{q}_i = \sum_{i=1}^{n} \left(\bar{\varepsilon}_i Z_i\right) \dot{q}_i$$

Explicit Form for Jacobian

- Repeated here: $$\omega = \sum_{i=1}^{n} (\bar{\varepsilon}_i Z_i)\dot{q}_i$$

- Let's expand the expression:

$$\begin{aligned}
\omega &= \sum_{i=1}^{n} (\bar{\varepsilon}_i Z_i)\dot{q}_i \\
&= (\bar{\varepsilon}_1 Z_1)\dot{q}_1 + (\bar{\varepsilon}_2 Z_2)\dot{q}_2 + \cdots + (\bar{\varepsilon}_n Z_n)\dot{q}_n \\
&= \underbrace{\begin{bmatrix} \bar{\varepsilon}_1 Z_1 & \bar{\varepsilon}_2 Z_2 & \cdots & \bar{\varepsilon}_n Z_n \end{bmatrix}}_{J_\omega} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}
\end{aligned}$$

- Thus we have obtained the Jacobian for angular velocity!

Explicit Form for Jacobian

- IMPORTANT: All terms in the above expression must be in the same frame.
- E.g. in frame {0}: $$^0J_\omega = \left[\bar{\varepsilon}_1\,{}^0Z_1 \quad \bar{\varepsilon}_2\,{}^0Z_2 \quad \cdots \quad \bar{\varepsilon}_n\,{}^0Z_n\right]$$
- But $$^iZ_i = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$$
- Therefore: $$^0Z_i = {}^0_iR \cdot {}^iZ_i$$
- And we have: $$^0J_\omega = \left[\bar{\varepsilon}_1\left({}^0_1R\cdot{}^1Z_1\right) \quad \bar{\varepsilon}_2{}^0\left({}^0_2R\cdot{}^2Z_2\right) \quad \cdots \quad \bar{\varepsilon}_n{}^0\left({}^0_nR\cdot{}^nZ_n\right)\right]$$
- and: $$^0\omega = \left[\bar{\varepsilon}_1\left({}^0_1R\cdot{}^1Z_1\right) \quad \bar{\varepsilon}_2{}^0\left({}^0_2R\cdot{}^2Z_2\right) \quad \cdots \quad \bar{\varepsilon}_n{}^0\left({}^0_nR\cdot{}^nZ_n\right)\right]\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \vdots\\ \dot{q}_n\end{bmatrix}$$
- Note that $^0_iR^iZ_i$ means the 3rd column of ^0_iR matrix because $^iZ_i = [0,0,1]^T$.

Therefore: ith column of the $^0J_\omega$ matrix = 3rd column of ^0_iR matrix!!

Explicit Form for Jacobian

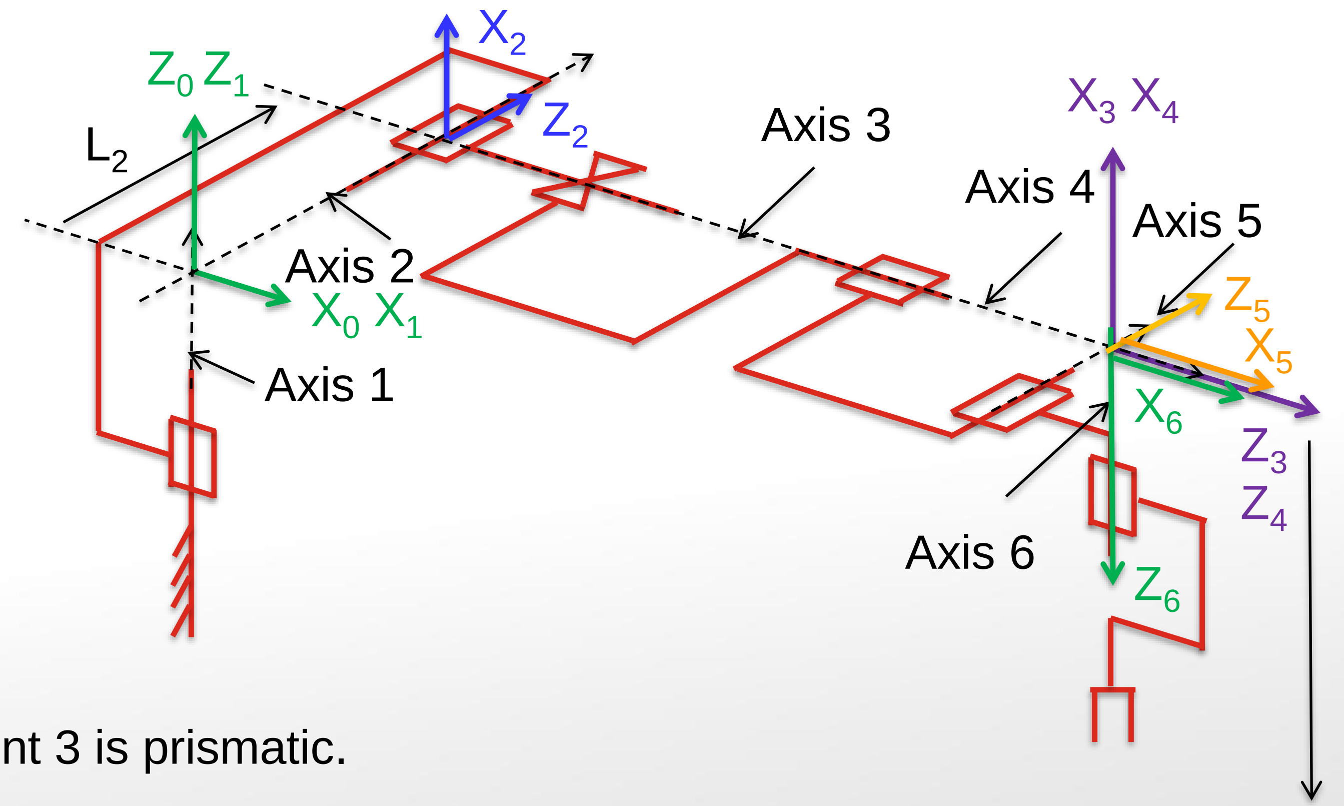
- Summary:

$$ {}^{0}J_{\omega} = \begin{bmatrix} \bar{\varepsilon}_1 \left({}^{0}_{1}R \cdot {}^{1}Z_1 \right) & \bar{\varepsilon}_2 {}^{0}\left({}^{0}_{2}R \cdot {}^{2}Z_2 \right) & \cdots & \bar{\varepsilon}_n {}^{0}\left({}^{0}_{n}R \cdot {}^{n}Z_n \right) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} $$

3rd column of ${}^{0}_{i}R$ matrix
or ${}^{0}_{i}T$ matrix

Example

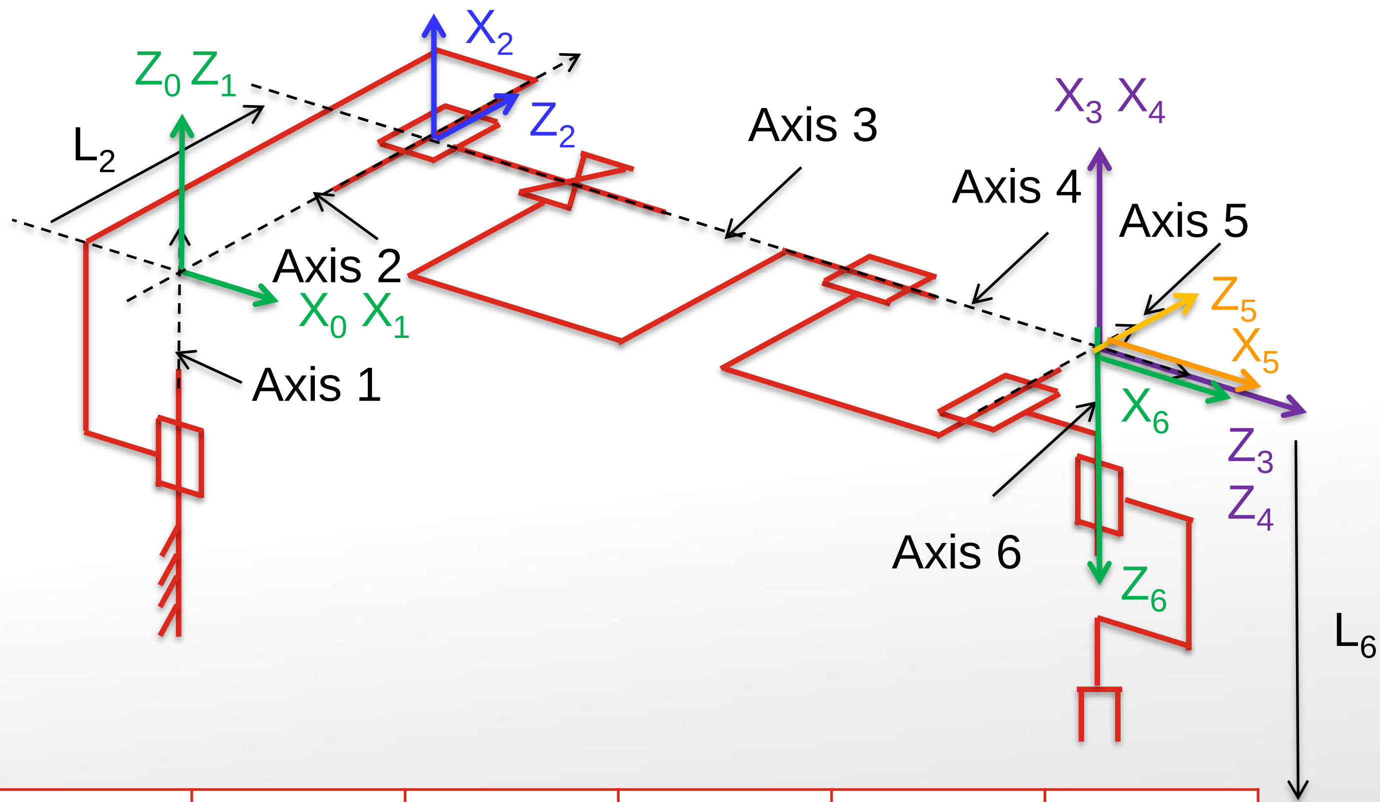
- Stanford Scheinman Arm:



- Joint 3 is prismatic.
- All others are revolute.

Example

- Stanford Scheinman Arm:



- J_w =

Z_1	Z_2	0	Z_4	Z_5	Z_6

Example

- In lecture 3, we had derived:

$$ {}^0_1T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1_2T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^2_3T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^3_4T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^4_5T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^5_6T = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Example

- We learnt that ith column of the $^0J_\omega$ matrix = 3rd column of ^0_iT matrix.
- 1st column of $^0J_\omega$ = 3rd column of ^0_1T →

$$^0_1T = \begin{bmatrix} c\theta_1 & -s\theta_1 & \boxed{0} & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- 2nd column of $^0J_\omega$ = 3rd column of ^0_2T →

$$^0_2T = {}^0_1T \cdot {}^1_2T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_1c_2 & -c_1s_2 & -s_1 & -L_2s_1 \\ s_1c_2 & -s_1s_2 & c_1 & L_2c_1 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

- 3rd column of ${}^0J_\omega$ = 3rd column of ${}^0_3T$ →

$$
{}^0_3T = {}^0_2T \cdot {}^2_3T = \begin{bmatrix} c_1c_2 & -c_1s_2 & -s_1 & -L_2s_1 \\ s_1c_2 & -s_1s_2 & c_1 & L_2c_1 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
= \begin{bmatrix} c_1c_2 & s_1 & \boxed{-c_1s_2} & -c_1d_3s_2 - L_2s_1 \\ s_1c_2 & -c_1 & \boxed{-s_1s_2} & -s_1d_3s_2 + L_2c_1 \\ -s_2 & 0 & \boxed{-c_2} & -d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

- However, this is $[0\ 0\ 0]^T$ because of prismatic joint.
- Continue the same method up to 6th column, and we would have fully specified

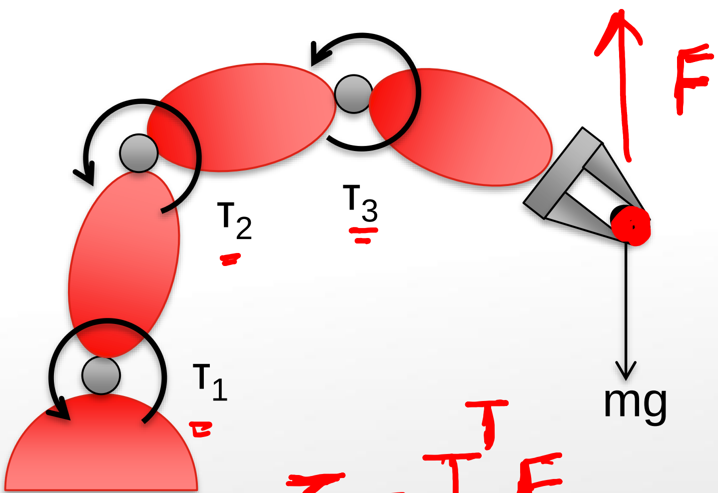
$$
{}^0J_\omega = \begin{bmatrix} 0 & -s_1 & 0 & \text{3rd} & \text{3rd} & \text{3rd} \\ 0 & c_1 & 0 & \text{column} & \text{column} & \text{column} \\ 1 & 0 & 0 & \text{of } {}^0_4T & \text{of } {}^0_5T & \text{of } {}^0_6T \end{bmatrix}
$$

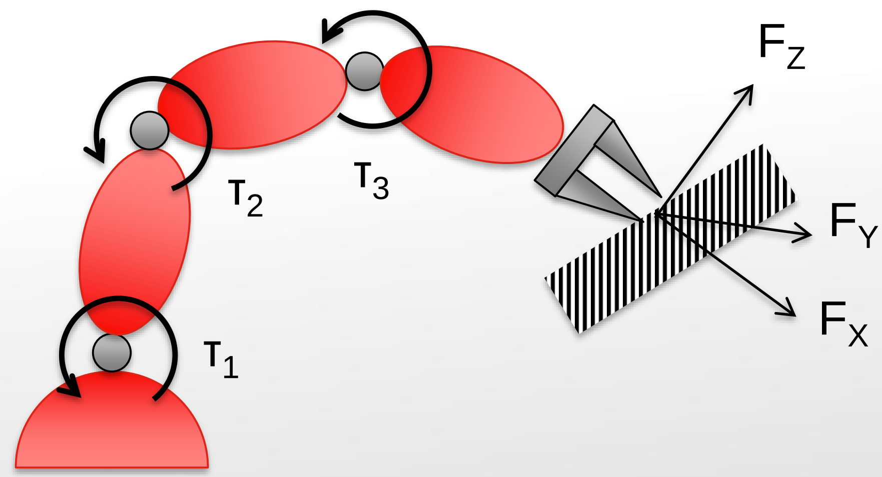
Content

- Introduction - Jacobian
- Method 1 - Direct differentiation (for Linear Jacobian)
- Method 2 - Velocity Propagation from Link to Link
- Method 3 - Explicit Form (for your study, not included in exam)
- Static Forces in Manipulators
- Singularities

Static Forces in Manipulator

- The question we would like to answer in this section is as follows:
  - The robot is holding an object with mass m (left), or
  - The robot is pushing the environment with force F (right).
  - What would be the joint torques needed to keep the system in static https://rmit.instructure.com/courses/51269/external_tools/23547?



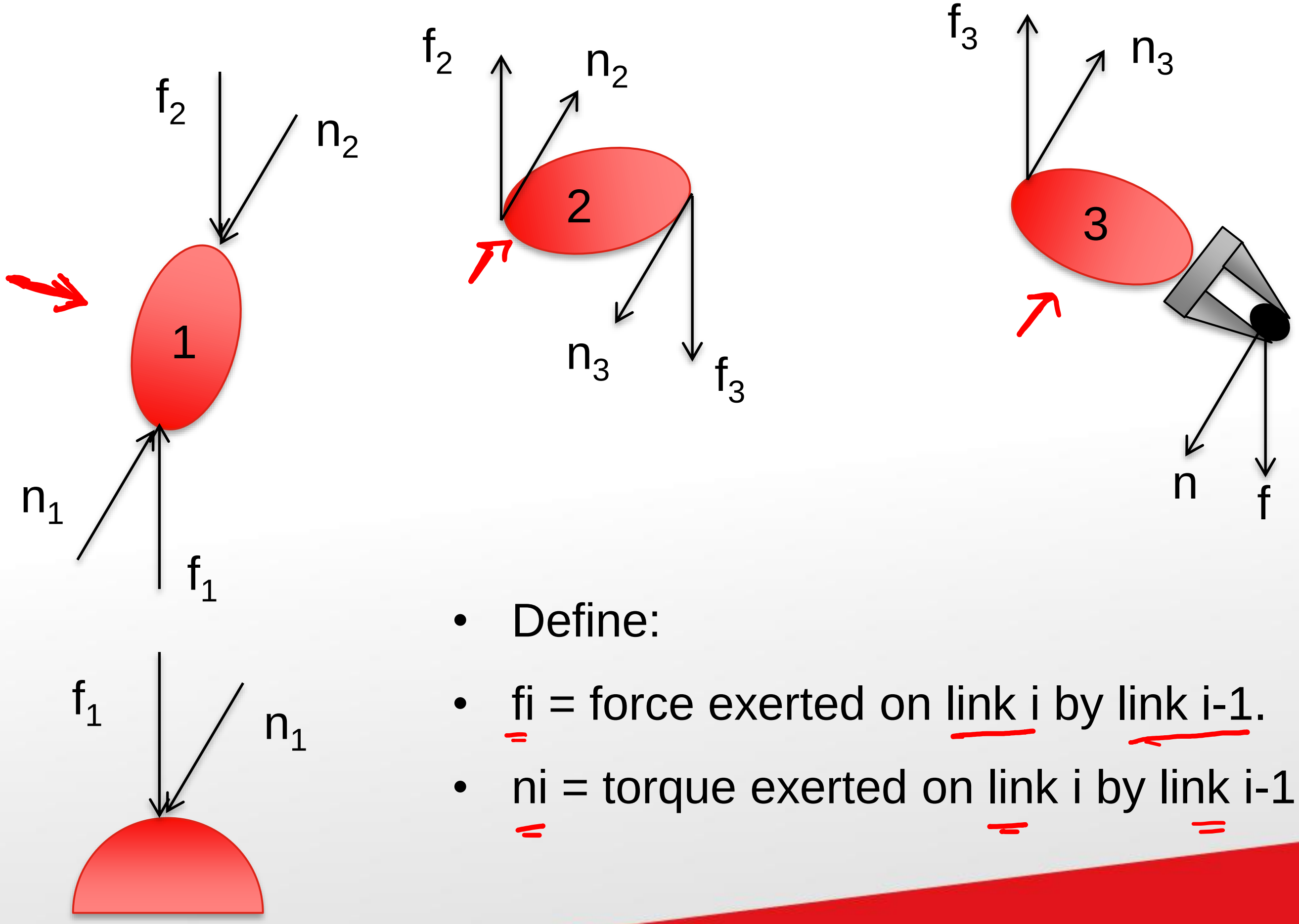


$$\tau = J^T F$$

$$\dot{x} = J\dot{\theta}$$

Static Forces in Manipulator

- This can be solved by separating each link, and find a force-moment balance relationship in terms of the link frames.

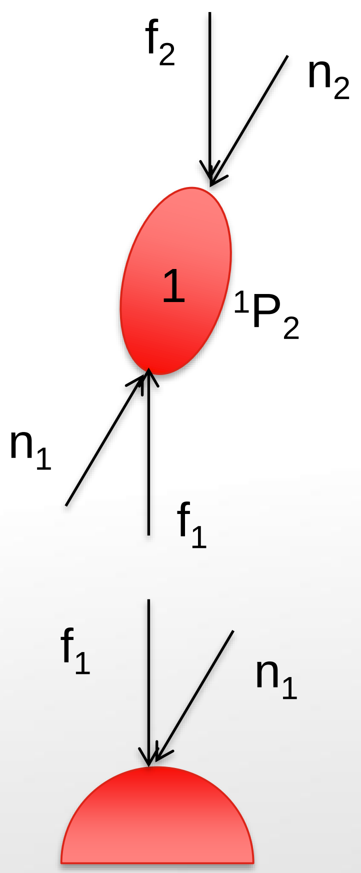


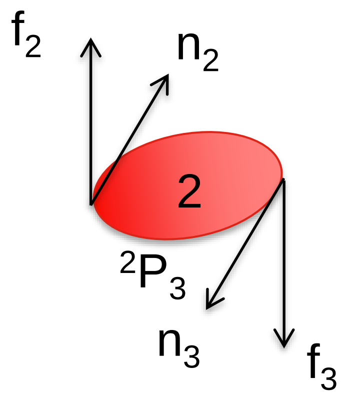
- Define:
- fi = force exerted on link i by link i-1.
- ni = torque exerted on link i by link i-1.

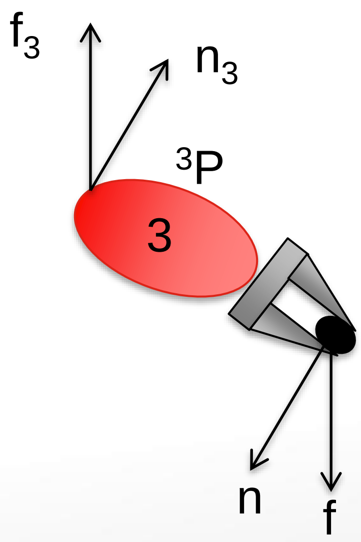
Static Forces in Manipulator

- For static equilibrium: $\sum f = 0 \ \& \ \sum n = 0$

Here: about frame origin







- Therefore:

$$^1f_1 - {}^1f_2 = 0; \quad {}^2f_2 - {}^2f_3 = 0; \quad {}^3f_3 - {}^3f = 0$$

$$^1n_1 - {}^1n_2 - {}^1P_2 \times {}^1f_2 = 0 \qquad {}^2n_2 - {}^2n_3 - {}^2P_3 \times {}^2f_3 = 0$$

$$^3n_3 - {}^3n - {}^3P \times {}^3f = 0$$

Static Forces in Manipulator

- We see that: $\begin{cases} {}^{i}f_i = {}^{i}f_{i+1} \\ {}^{i}n_i = {}^{i}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_{i+1} \end{cases}$ length or $\begin{matrix} {}^{i}f_i = {}^{i}_{i+1}R \cdot {}^{i+1}f_{i+1} \\ {}^{i}n_i = {}^{i}_{i+1}R \cdot {}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i \end{matrix}$ same
- The equations on the right use only forces and moments described within their own link frames.
- Hence we have the algorithm:

1 • Start from last link. n

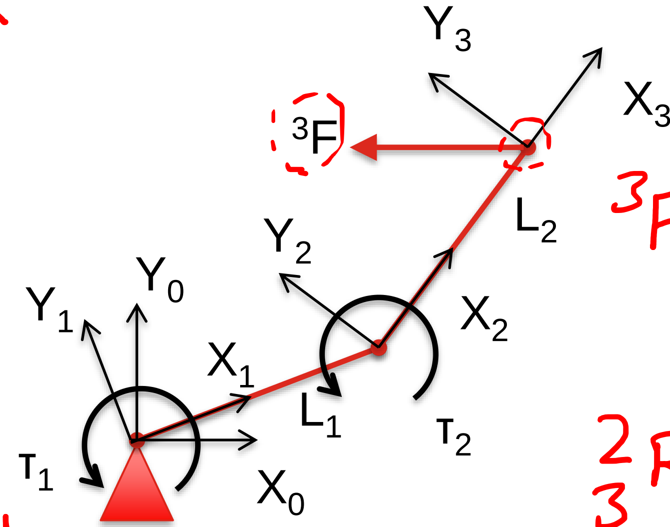
2 • Calculate iteratively, down to the base: $\begin{cases} {}^{i}f_i = {}^{i}_{i+1}R \cdot {}^{i+1}f_{i+1} \\ {}^{i}n_i = {}^{i}_{i+1}R \cdot {}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i \end{cases}$

3 • The joint torques required to maintain the static equilibrium are then calculated as dot-product of joint-axis vector and the moment vector acting on the link:
  - R Revolute: $\tau_i = {}^{i}n_i^T \cdot {}^{i}\hat{Z}_i$ P Prismatic: $\tau_i = {}^{i}f_i^T \cdot {}^{i}\hat{Z}_i$

Example

2-Link

- A force F acts horizontally (wrt {0}) on the origin of {3}, of the 2-link robot.
- What are the joint torques needed to hold the robot in equilibrium?
- We apply the recursive algorithm:
  - Start from: $n=2$



$$^3F = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix}$$

$$^2_3R = I$$

$$^2f_2 = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} \qquad {}^2n_2 = \underbrace{L_2\hat{X}_2 \times \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix}}_{} = \begin{bmatrix} 0 \\ 0 \\ L_2 f_y \end{bmatrix}$$

$$\begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix} \times$$

- where f_x and f_y are the x and y components of ^3F in {3}, and they are the same for {2}.

$$\begin{vmatrix} i & j & k \\ L_2 & 0 & 0 \\ f_x & f_y & 0 \end{vmatrix}$$

0 0

Example

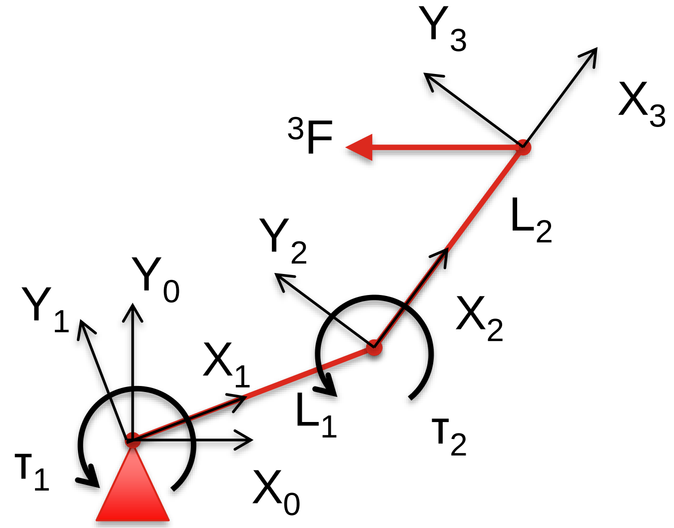
Step 2

- Next:

$^1f_1$

$$^1f_1 = {}^1_2R \cdot {}^2f_2$$

$$= \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} c_2 f_x - s_2 f_y \\ s_2 f_x + c_2 f_y \\ 0 \end{bmatrix}$$



Last link

$$^1n_1 = {}^1_2R \cdot {}^2n_2 + {}^1P_2 \times {}^1f_1$$

$\begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}$ a b

$$= \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ L_2 f_y \end{bmatrix} + L_1 \hat{X}_1 \times \begin{bmatrix} c_2 f_x - s_2 f_y \\ s_2 f_x - c_2 f_y \\ 0 \end{bmatrix}$$

$$\begin{vmatrix} i & j & k \\ L_1 & 0 & 0 \\ a & b & 0 \end{vmatrix}$$

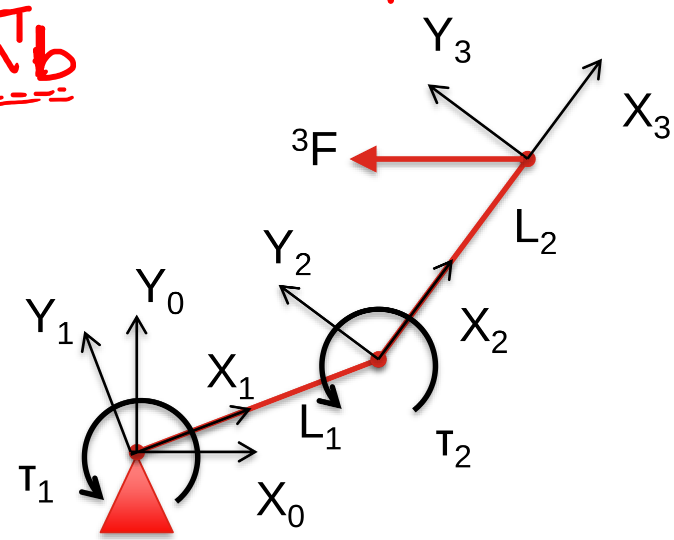
$$= \begin{bmatrix} 0 \\ 0 \\ L_2 f_y \end{bmatrix} + \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} c_2 f_x - s_2 f_y \\ s_2 f_x + c_2 f_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ L_2 f_y + L_1 s_2 f_x + L_1 c_2 f_y \end{bmatrix}$$

Previous Steps → $^1n_1$ & $^2n_2$ -

Example

$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \end{bmatrix}$ $a \cdot b = a^T b$

Step 3



- Finally, we calculate the torques:

J1

$$\tau_1 = {}^1n_1^T \cdot {}^1\hat{Z}_1$$

$^1n_1^T$

$$= \begin{bmatrix} 0 & 0 & L_2 f_y + L_1 s_2 f_x + L_1 c_2 f_y \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

1x3 3x1

$$\tau_1 = L_1 s_2 f_x + \left(L_2 + L_1 c_2\right) f_y$$

J2

$$\tau_2 = {}^2n_2^T \cdot {}^2\hat{Z}_2$$

$$= \begin{bmatrix} 0 & 0 & L_2 f_y \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\tau_2 = L_2 f_y$$

$$^3\tau = {}^3J^T \cdot {}^3F$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} L_1 s_2 & L_2 + L_1 c_2 \\ 0 & L_2 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Remember that this is in frame {3}

This is exactly $^3J^T$ → we had this expression before:

^3J

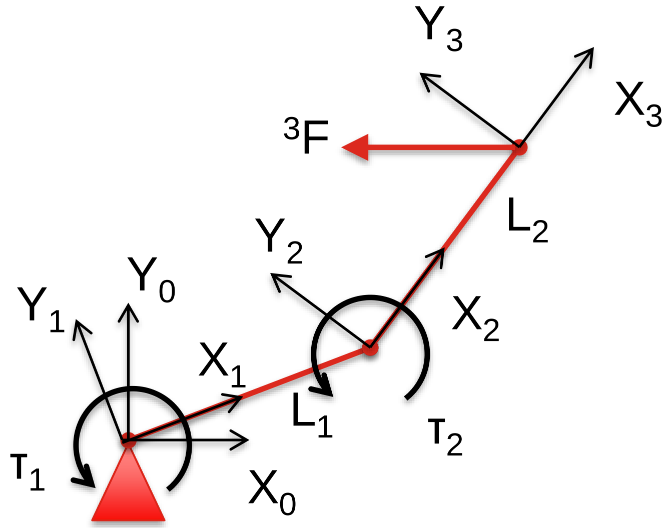
$$^3\upsilon_3 = \begin{bmatrix} L_1 s_2 \dot{\theta}_1 \\ L_1 c_2 \dot{\theta}_1 + L_2\left(\dot{\theta}_1 + \dot{\theta}_2\right) \end{bmatrix} = \begin{bmatrix} L_1 s_2 & 0 \\ L_2 + L_1 c_2 & L_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$^0J \to ?$

Example

^0_3R

- Therefore: $\tau = {}^3J^T \cdot {}^3F$
- We can also express everything in frame {0}:



$$\tau = {}^0J^T \cdot {}^0F$$

$$= \left({}^0_3R \cdot {}^3J\right)^T \cdot {}^0F$$

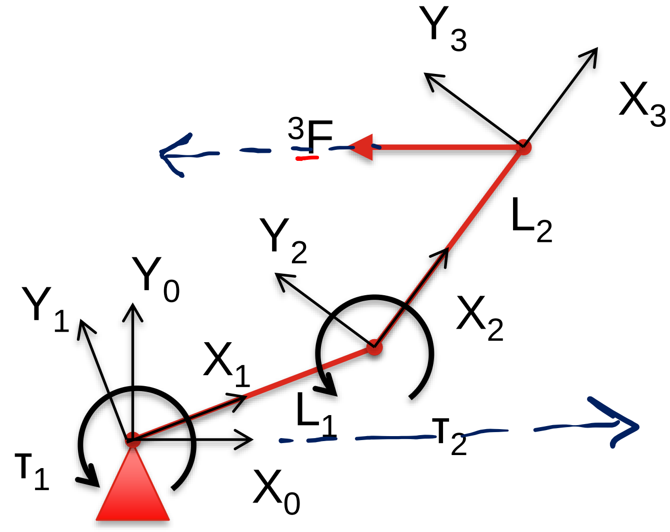
$${}^0\tau = \left(\begin{bmatrix} c_{12} & -s_{12} \\ s_{12} & c_{12} \end{bmatrix}\begin{bmatrix} L_1s_2 & 0 \\ L_2 + L_1c_2 & L_2 \end{bmatrix}\right)^T \cdot \begin{bmatrix} {}^0f_x \\ {}^0f_y \end{bmatrix}$$

$$= \begin{bmatrix} -L_1s_1 - L_2s_{12} & -L_2s_{12} \\ L_1c_1 + L_2c_{12} & L_2c_{12} \end{bmatrix}^T \cdot \begin{bmatrix} {}^0f_x \\ {}^0f_y \end{bmatrix}$$

$$= \begin{bmatrix} -L_1s_1 - L_2s_{12} & L_1c_1 + L_2c_{12} \\ -L_2s_{12} & L_2c_{12} \end{bmatrix} \cdot \begin{bmatrix} {}^0f_x \\ {}^0f_y \end{bmatrix}$$

${}^0J^T$ ${}^0J^T$ ${}^0F$

Example



- With numerical values:
  - $L_1 = 1$, $L_2 = 1$
  - $\theta_1 = 30\text{deg}$, $\theta_2 = 30\text{deg}$
  - $|^3F| = 10\text{N}$
- With these, we have:

$$f_x = {}^3f_x = -5 \qquad {}^0f_x = \boxed{-10}$$
$$f_y = {}^3f_y = 8.66 \qquad {}^0f_y = \boxed{0}$$

- Substituting the values into either

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix} L_1 s_2 & L_2 + L_1 c_2 \\ 0 & L_2 \end{bmatrix}\begin{bmatrix} f_x \\ f_y\end{bmatrix}$$

or

$$\tau = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & L_1 c_1 + L_2 c_{12} \\ -L_2 s_{12} & L_2 c_{12}\end{bmatrix} \cdot \begin{bmatrix} {}^0f_x \\ {}^0f_y \end{bmatrix}$$

gives

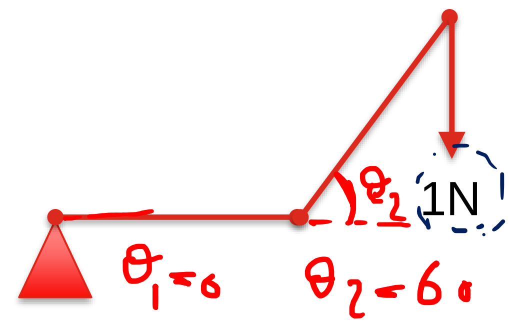
$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix} 13.66 \\ 8.66 \end{bmatrix}$$

Another Example

- Still the same two link robot, but at different configuration and force.
- Case 1: $\theta_1 = 0$, $\theta_2 = 60$ deg, F = 1N.

$$\tau = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & L_1 c_1 + L_2 c_{12} \\ -L_2 s_{12} & L_2 c_{12} \end{bmatrix} \cdot \begin{bmatrix} {}^0 f_x \\ {}^0 f_y \end{bmatrix}$$

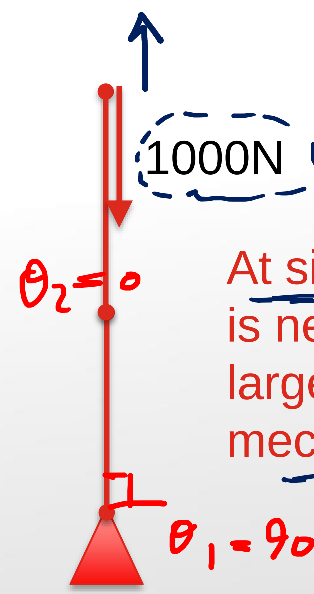
$$= \begin{bmatrix} -0.866 & 1.5 \\ -0.866 & 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix} \begin{matrix} \tau_1 \\ \tau_2 \end{matrix}$$



- Case 2: $\theta_1 = 90$ deg, $\theta_2 = 0$, F = 1000N.

$$\tau = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & L_1 c_1 + L_2 c_{12} \\ -L_2 s_{12} & L_2 c_{12} \end{bmatrix} \cdot \begin{bmatrix} {}^0 f_x \\ {}^0 f_y \end{bmatrix}$$

$$= \begin{bmatrix} -2 & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -1000 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{matrix} \tau_1 \\ \tau_2 \end{matrix}$$



At singularity, no torque is needed to withstand large forces → mechanical advantage!

Content

- Introduction - Jacobian
- Method 1 - Direct differentiation (for Linear Jacobian)
- Method 2 - Velocity Propagation from Link to Link
- Method 3 - Explicit Form (for your study, not included in exam)
- Static Forces in Manipulators
- Singularities

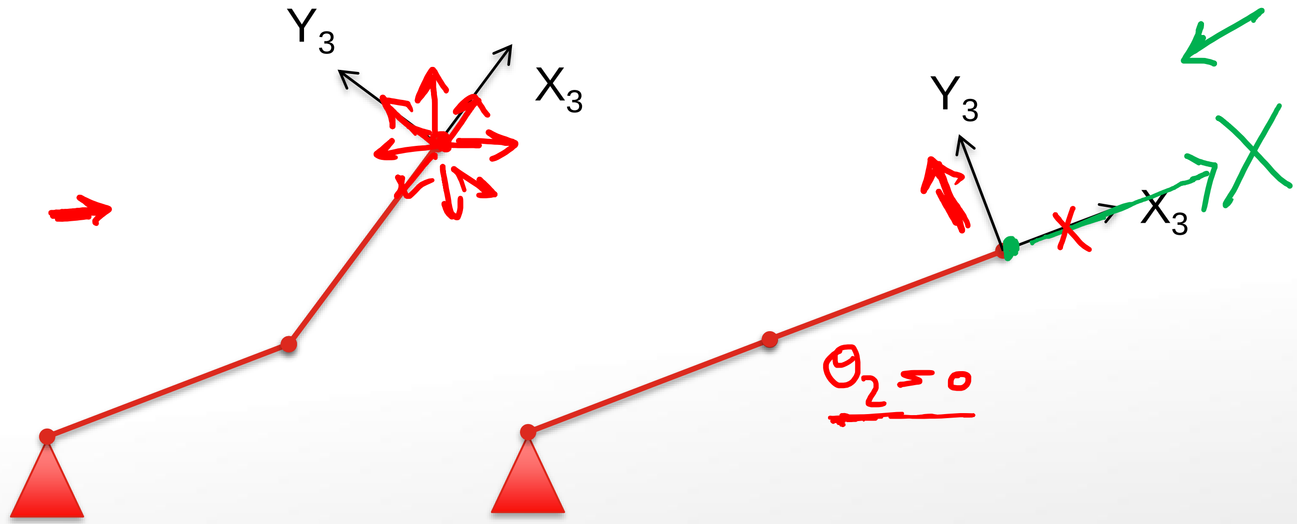
Kinematic Singularities

The robot loses some DoF

Rotations R
& Express the Rot $\begin{cases}\alpha\\ \beta\\ \gamma\end{cases}$

$\beta = 0, -180$
α, γ

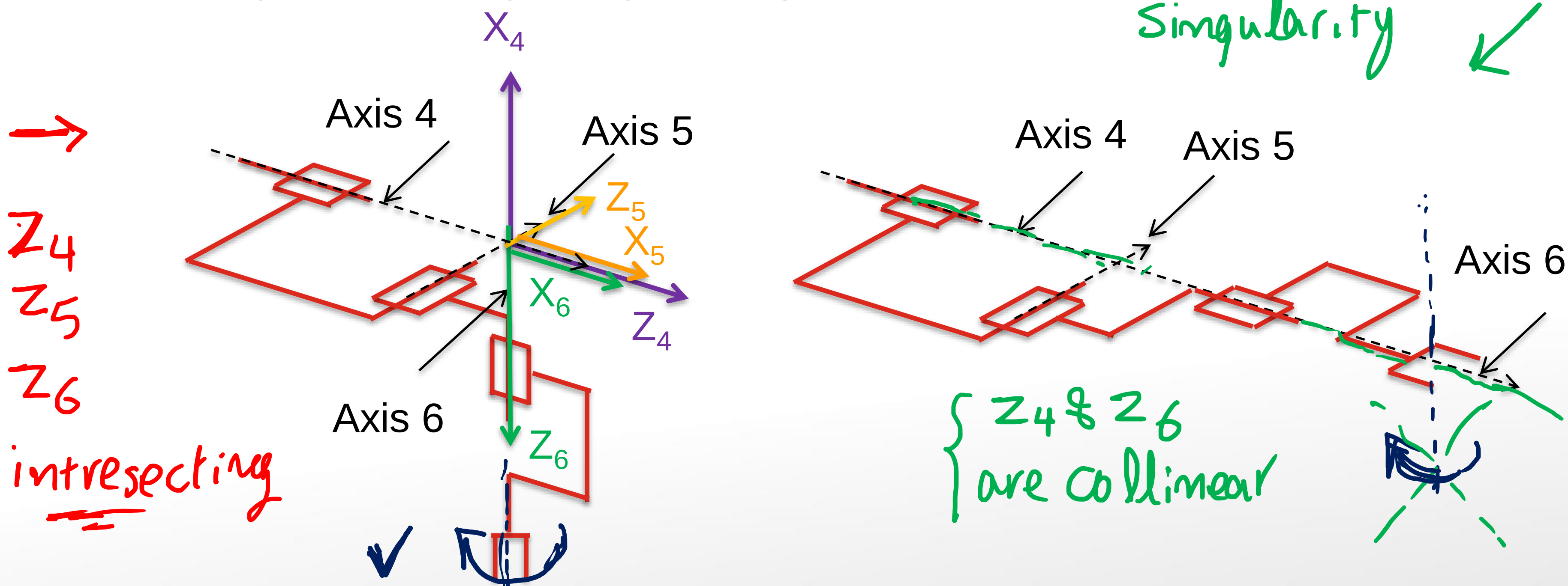
- Kinematic singularity happens when the end-effector loses the ability to move in a direction, or to rotate about a direction.
  - THE direction is called the "Singular Direction".
- E.g. Two-link robot:



- In the left figure, the end-effector can move in any direction instantly.
- In the right figure, the end-effector loses the ability to move in the x-direction.

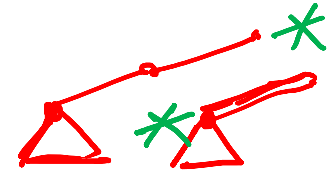
Kinematic Singularities

- Another example: 6-link robot, with the last 3 axes intersecting.
- Looking at the last 3 joints (the wrist):



- In the left figure, the end-effector can rotate about a "vertical axis".
- In the right figure, when axes 4 and 6 are collinear, the end-effector loses ability to rotate about the axis.

Kinematic Singularities



- The first example is a case of workspace-boundary singularities.
  - Manipulator is fully stretched or folded back on itself.
- The second example is a case of workspace-interior singularities.
  - Generally caused by lining up two or more axes.

Collinear

6 Link Z_4 & Z_6

- Mathematically, these singularities happen when the Jacobian matrix becomes non-invertible / singular.
  - We knew: $v = J(q)\dot{q}$

$v = J\dot{\theta} \rightarrow \dot{\theta} = J^{-1} v$ (≈ 0, ?, ∞)

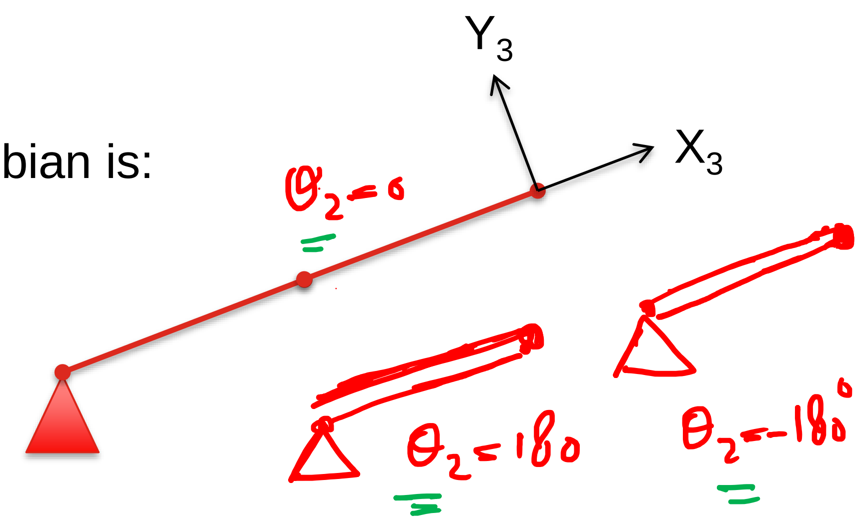
  - The inverse question is: If we "want" a certain v, what should the joint rate be?
  - This can be obtained via: $\dot{q} = J^{-1}(q)v$

$|J| \rightarrow 0 \rightarrow$ Singularity

  - If the Jabobian matrix is not invertible or ill-conditioned (determinant close to zero), then we need an infinitely big joint rate to obtain the v. (just imagine q_dot = 1/J * v with J = 0 or J ≈ 0, for scalar cases)
    - Not possible or practical.

Example

2-Link



- For the two-link robot example, the Jacobian is:

$v = J \dot{\theta}$

$$\begin{bmatrix} {}^0v_{3x} \\ {}^0v_{3y} \end{bmatrix} = \underbrace{\begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix}}_{{}^0J_v} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

- The determinant of the Jacobian is: $|J| = 0$

$$\det \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix} = -L_1 L_2 s_1 c_{12} - L_2^2 s_{12} c_{12} + L_1 L_2 c_1 s_{12} + L_2^2 s_{12} c_{12}$$

$$= \boxed{L_1 L_2 s_2}$$

L_1 & L_2

- The determinant is zero, i.e. the Jacobian becomes singular, when $\boxed{s_2 = 0}$ i.e.

$\sin \theta_2$

$$\theta_2 = k\pi$$

$\begin{cases} \theta_2 = 0 \\ \theta_2 = \pm 180 \end{cases}$

Resolved Motion Rate Control

- Here, we would like to show another useful application of the Jacobians, apart from calculating velocity and force.
- The Jacobian is widely used to control the robots:
  - At the current moment, the joint angles are q=(q1,…,qn), and we know the tip's position (from forward kinematics) at x.
  - Next, we want the tip's position to move somewhere else. How should the joint angles change?

    A • $\dot{\theta}$? • B

    - Last week, we saw how difficult it is to solve this inverse kinematics question, because it is a nonlinear problem.

      $\dot{x} = J\dot{q}$

    - However, we now know that for small changes: $\delta x = J(q)\delta q$

      $\delta x = J\,\delta q$

    - Outside of singularity, we have: $\delta q = J^{-1}(q)\delta x$

      $\delta q = J^{-1}\delta x$

    - Hence: Split the path into many small paths δx

      δx

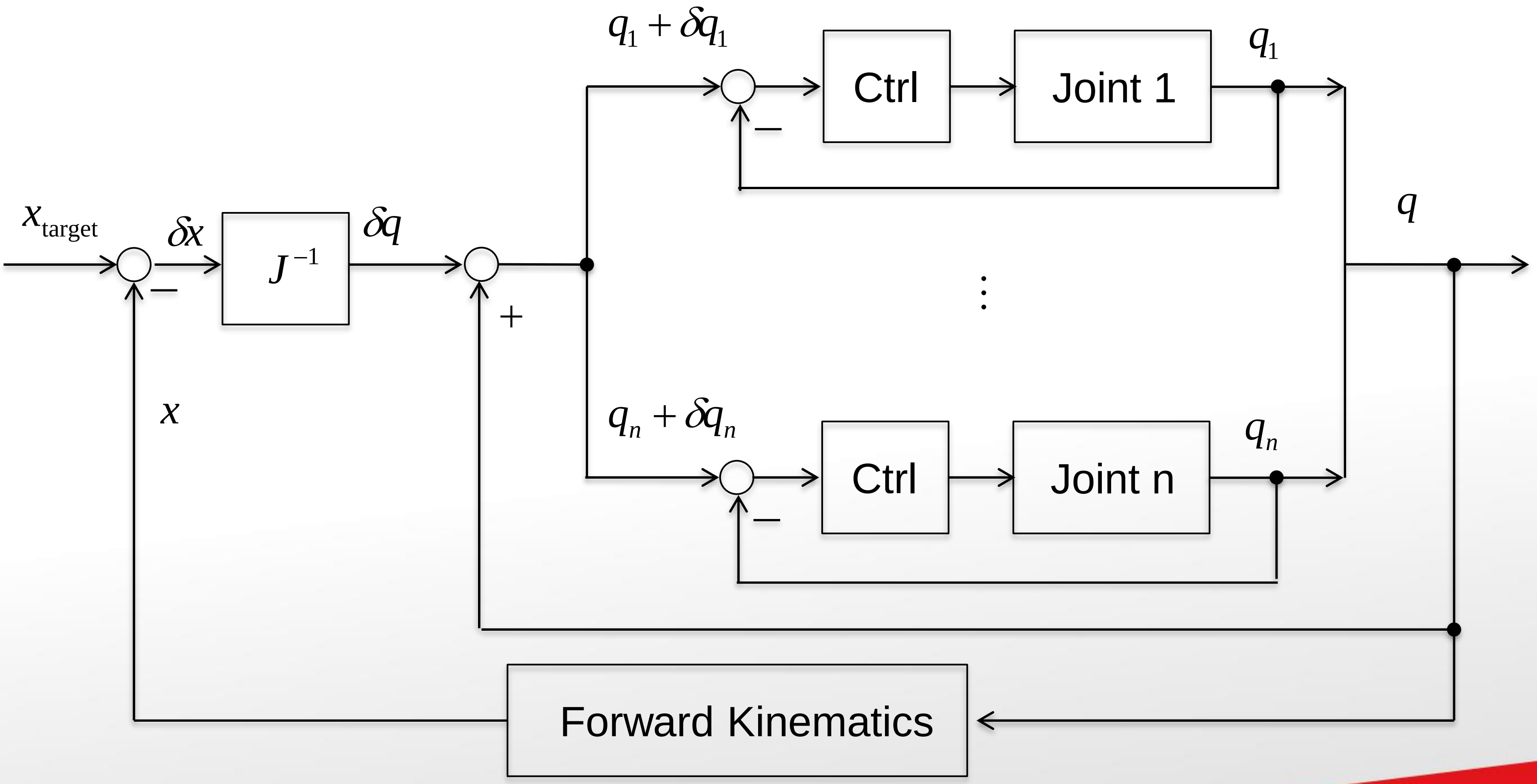
      A B

    - And at each step, calculate $\delta q = J^{-1}(q)\delta x$ and set $q_{\text{new}} = q + \delta q$

      δq

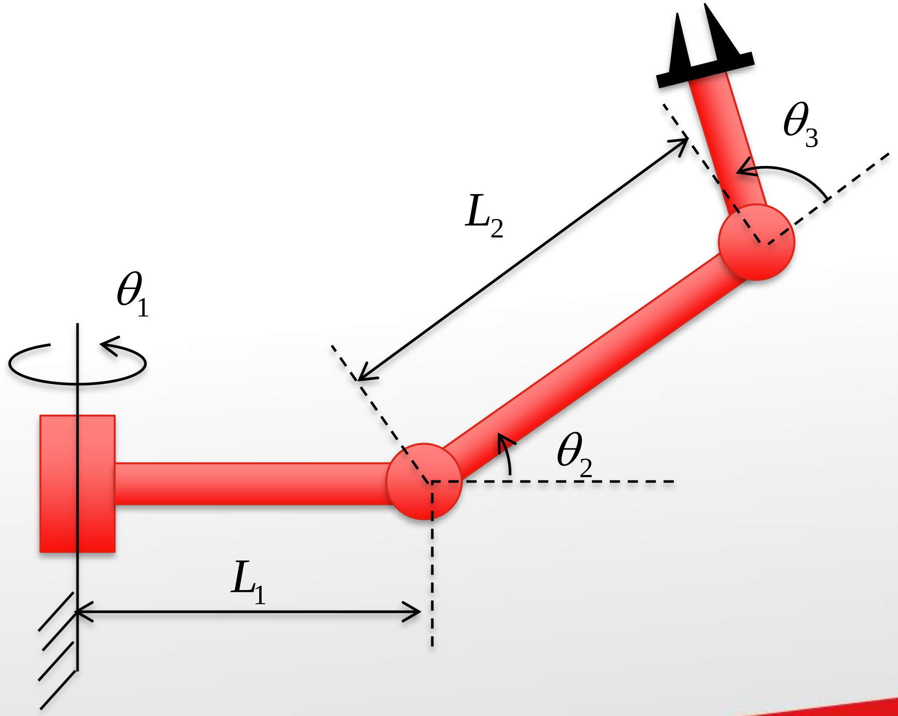
Resolved Motion Rate Control

- The block diagram is as follows:



Tutorial Assignments

- **Question 1:**
  - Find the Jacobian of the manipulator shown on the right. (You should already have some information about this robot in earlier tutorials).
  - Write it in terms of frame {3} at the wrist of robot.
    - Velocity propagation method.
    - Differentiation of kinematic equations.
  - Write also in terms of frame {4} at the tip of hand, having same orientation as {3}.



Tutorial Assignments

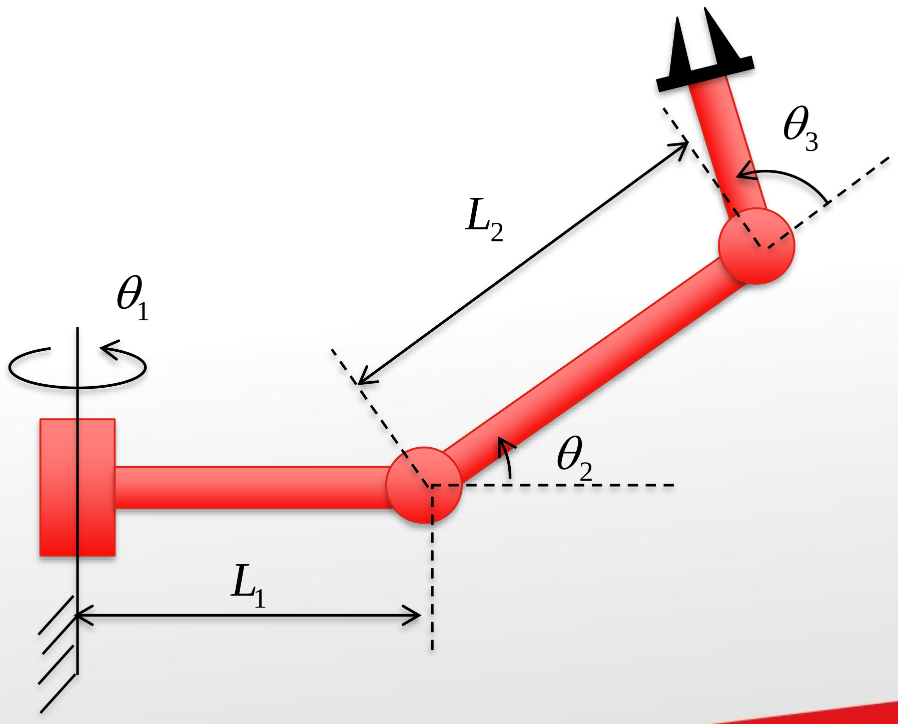
- **Question 2:**
  - A 2-ink manipulator has the following Jacobian:

$$^{0}J(\theta)=\begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}$$

  - If we ignore gravity, what are the joint torques required so that the manipulator can apply a static force of $^{0}F = 10\hat{X}_0$?

Tutorial Assignments

- **Question 3:**
  - Find the Jacobian of the manipulator shown on the right. (You should already have some information about this robot in earlier tutorials).
  - Write it in terms of frame {3} at the wrist of robot.
    - Explicit form for Jacobian.
  - Write also in terms of frame {4} at the tip of hand, having same orientation as {3}.



Thank you!

Have a good evening.