MAE263B: Dynamics of Robotic Systems Discussion Section – Week5

: Jacobian (SCARA)

Seungmin Jung

02.07.2020.

Contents

- Jacobian with SCARA example
  - Velocity propagation
  - Direct differentiation
- Frame of Representation

Kinematics Relations - Joint & Cartesian Spaces

- A robot is often used to manipulate object attached to its tip (end effector).
- The location of the robot tip may be specified using one of the following descriptions:
- ***Joint Space***

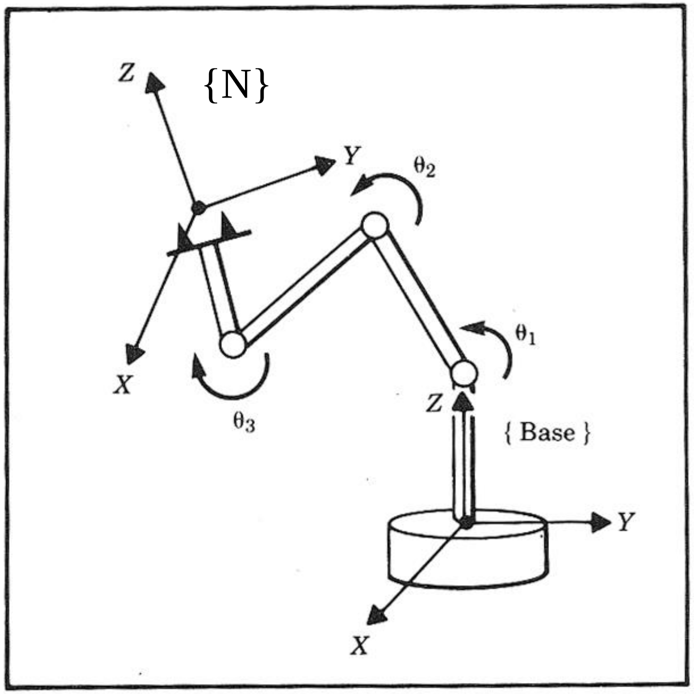
$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix}$$

- ***Cartesian Space / Operational Space***

$${}^0_N T = \begin{bmatrix} {}^0_N R & {}^0 P_N \\ 0 & 1 \end{bmatrix}$$

$$X = \begin{bmatrix} \left[{}^0 P_N \right] \\ \left[{}^0 r_N \right] \end{bmatrix}$$

Euler Angles



A minimal representation of orientation - Euler angles

- As an example, consider a transformation that uses ZYZ Euler angles $\underline{\phi}_e = [\varphi, \vartheta, \psi]^T$

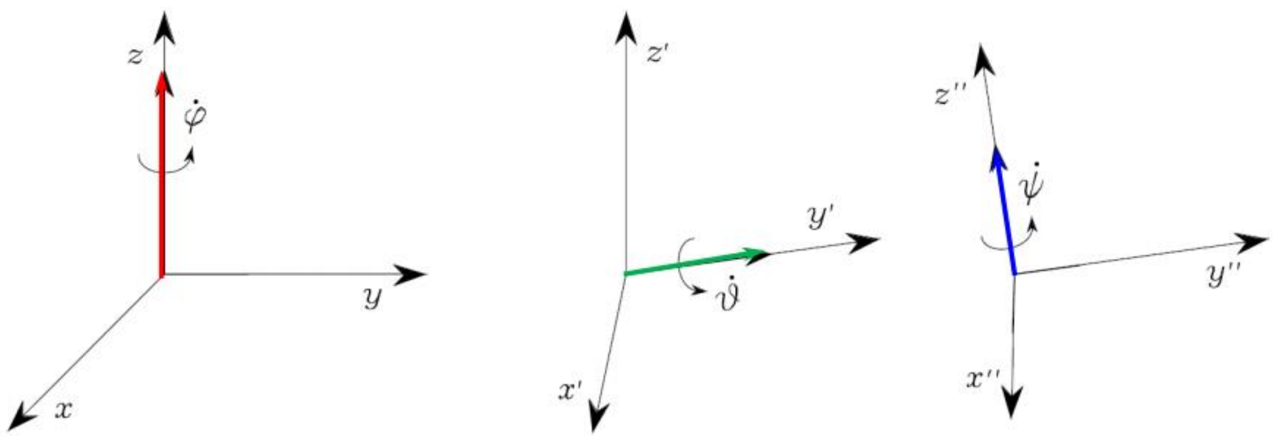


Fig. 3.8. Rotational velocities of Euler angles ZYZ in current frame

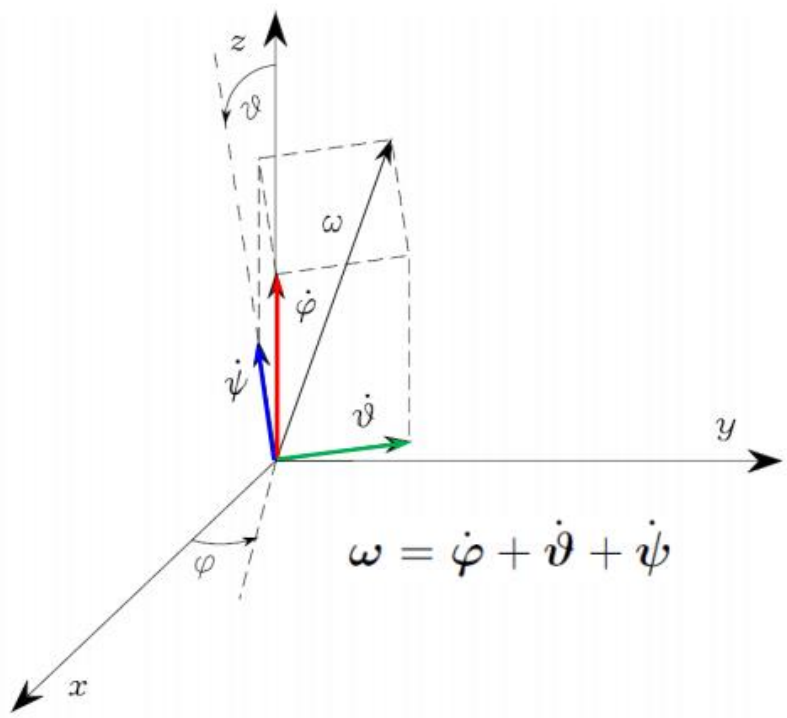
$$\phi = \begin{bmatrix} \varphi \\ \vartheta \\ \psi \end{bmatrix}$$

Siciliano, et al. Robotics: Modelling, Planning, and Control. **Sec. 3.6**. London: Springer-Verlag, 2009.

A minimal representation of orientation - Euler angles

Relationship between $\underline{\omega}_e$ and $\dot{\underline{\phi}}_e$

EXAMPLE



$$\boldsymbol{\omega} = \dot{\boldsymbol{\varphi}} + \dot{\boldsymbol{\vartheta}} + \dot{\boldsymbol{\psi}}$$

$$\dot{\underline{\phi}}_e = \begin{bmatrix} \dot{\varphi} \\ \dot{\vartheta} \\ \dot{\psi} \end{bmatrix}$$

$$\boldsymbol{\omega}_e = \boldsymbol{T}(\boldsymbol{\phi}_e)\dot{\boldsymbol{\phi}}_e$$

$$\boldsymbol{T} = \begin{bmatrix} 0 & -s_\varphi & c_\varphi s_\vartheta \\ 0 & c_\varphi & s_\varphi s_\vartheta \\ 1 & 0 & c_\vartheta \end{bmatrix}$$

Analytical Jacobian

- In order to design controllers in operational space, we must use the analytical Jacobian $J_A(\underline{q})$, a transformed version of the geometric Jacobian $J(\underline{q})$.
  - The analytical Jacobian is used in both Jacobian transpose control and Jacobian inverse control.

$$\boldsymbol{J}_A(\boldsymbol{q}) = \frac{\partial \boldsymbol{k}(\boldsymbol{q})}{\partial \boldsymbol{q}} \qquad \textit{or} \qquad \boldsymbol{J} = \boldsymbol{T}_A(\phi)\boldsymbol{J}_A$$

$$\text{where} \quad \boldsymbol{T}_A(\phi_e) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{T}(\phi_e) \end{bmatrix}$$

Siciliano, et al. Robotics: Modelling, Planning, and Control. Ch. 8. London: Springer-Verlag, 2009.

Comments on $\underline{\omega}_e$ and $\dot{\underline{\phi}}_e$

- The meaning of $\underline{\omega}_e$ is more intuitive than that of $\dot{\underline{\phi}}_e$.
  - $\underline{\omega}_e$ represents components of angular velocity with respect to a fixed base frame.
  - $\dot{\underline{\phi}}_e$ represents non-orthogonal components of angular velocity with respect to a frame that varies as the end-effector orientation varies.
- The integral of $\dot{\underline{\phi}}_e$ over time yields $\underline{\phi}_e$, but the integral of $\underline{\omega}_e$ does NOT have a clear physical interpretation.
- In general, $\underline{\omega}_e \neq \dot{\underline{\phi}}_e$ (and $J(\underline{q}) \neq J_A(\underline{q})$) unless you are considering a special case in which all DOFs cause rotations about the <u>same</u> fixed axis in space (e.g. z-axis for planar arm)

Siciliano, et al. <u>Robotics: Modelling, Planning, and Control</u>. **Sec. 3.6**. London: Springer-Verlag, 2009.

Kinematics Relations - Forward & Inverse

- The robot kinematic equations relate the two description of the robot tip location

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix}$$

$$X = FK(\theta)$$

$$\theta = IK(X)$$

$$X = \begin{bmatrix} \begin{bmatrix} {}^0P_N \end{bmatrix} \\ \begin{bmatrix} {}^0r_N \end{bmatrix} \end{bmatrix}$$

Tip Location in Joint Space

Tip Location in Cartesian Space

UCLA

Velocity relationship
: Jacobian Matrix – Joint velocity / End-effector velocity

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

$$\dot{\theta} = \frac{d}{dt}[\theta] = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_N \end{bmatrix}$$

$$\dot{X} = \frac{d}{dt}[X] = \begin{bmatrix} [\, v_N \,] \\ [\, \omega_N \,] \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Tip Velocity in Joint Space

$$\dot{\theta} = J^{-1}\dot{X}$$

Tip velocity in Cartesian Space

Jacobian Matrix - Introduction

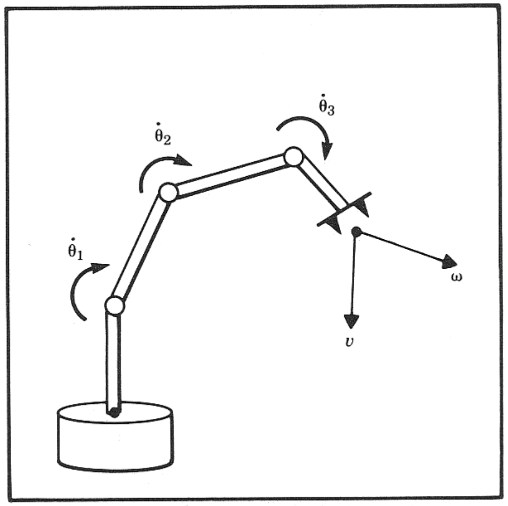
- The velocity relationship

  : The relationship between

  the joint angle rates ($\underline{\dot{\theta}}_N$)

  and the translation and rotation velocities of the end effector ($\underline{\dot{x}}$).

$$\underline{\dot{x}} = J(\underline{\theta})\underline{\dot{\theta}}$$



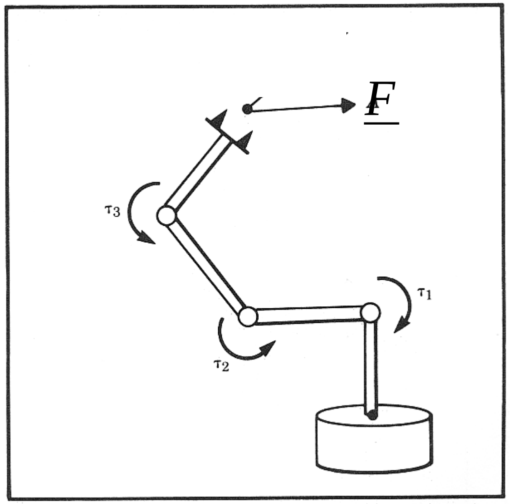
- The relationship between

  the robot joint torques ($\underline{\tau}$)

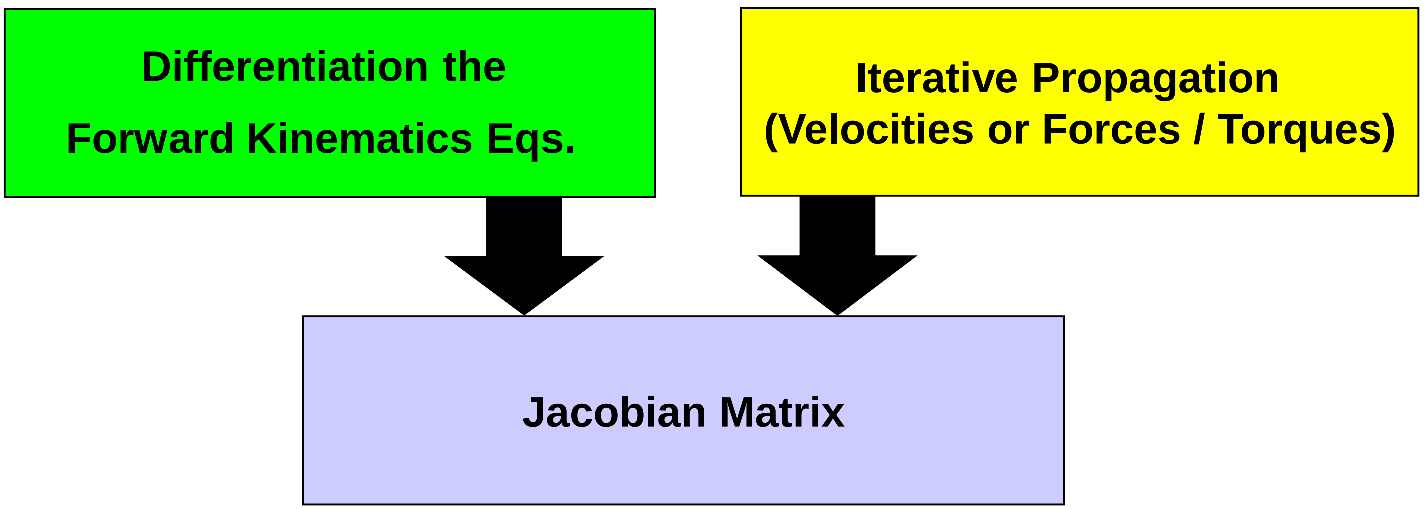
  and the forces and moments ($\underline{F}$)

  at the robot end effector (**Static Conditions**).

$$\underline{\tau} = J(\underline{\theta})^T \underline{F}$$



Jacobian Matrix - Calculation Methods



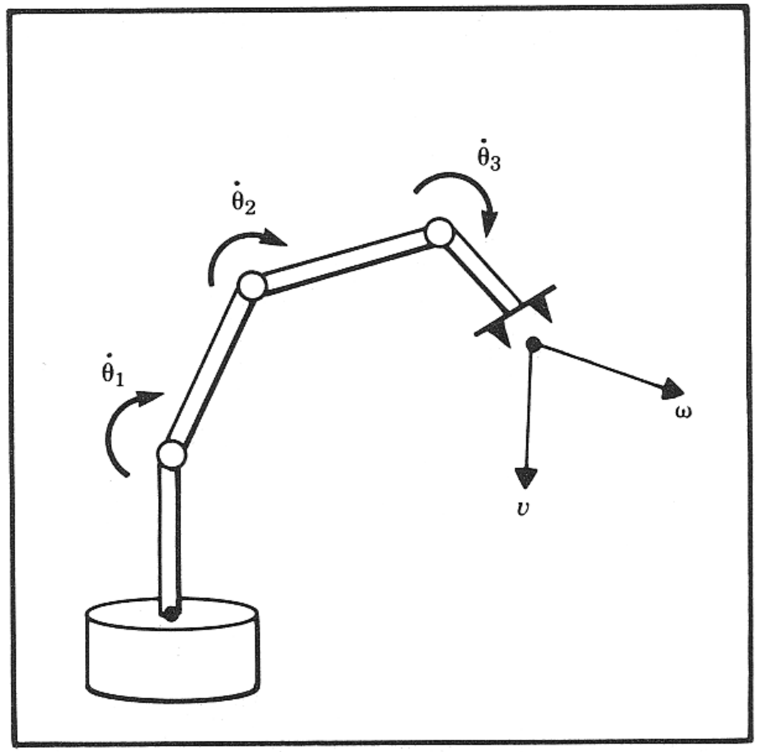
Jacobian Matrix - Introduction

- In the field of robotics the Jacobian matrix describe the relationship between the joint angle rates ($\dot{\underline{\theta}}_N$) and the translation and rotation velocities of the end effector ($\dot{\underline{x}}$). This relationship is given by:

$$\begin{Bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{Bmatrix} = \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{Bmatrix} \qquad \begin{Bmatrix} v \\ w \end{Bmatrix}$$

$$\dot{\underline{x}} = J(\underline{\theta})\dot{\underline{\theta}} \qquad \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \end{Bmatrix}$$

$$\dot{\underline{\theta}} = J(\underline{\theta})^{-1}\dot{\underline{x}}$$



UCLA

Jacobian Matrix - Introduction

- This expression can be expanded to:

- *Cartesian Space / Operational Space*

$$ {}^{0}_{N}T = \begin{bmatrix} {}^{0}_{N}R & {}^{0}P_N \\ 0 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} \begin{bmatrix} {}^{0}P_N \end{bmatrix} \\ \begin{bmatrix} {}^{0}r_N \end{bmatrix} \end{bmatrix} $$

Euler Angles

$$ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \overbrace{\qquad J(\underline{\theta}) \qquad}^{N} \end{bmatrix} \left. \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \cdots \\ \dot{\theta}_N \end{bmatrix} \right\} N $$

6x1 **6xN** **Nx1**

$$ \dot{p}_e = J_P(q)\dot{q} $$

$$ \omega_e = J_O(q)\dot{q}. $$

- Where:
  - $\underline{\dot{x}}$ is a 6x1 vector of the end effector linear and angular velocities
  - $J(\underline{\theta})$ is a 6xN Jacobian matrix
  - $\underline{\dot{\theta}}_N$ is a Nx1 vector of the manipulator joint velocities
  - N is the number of joints

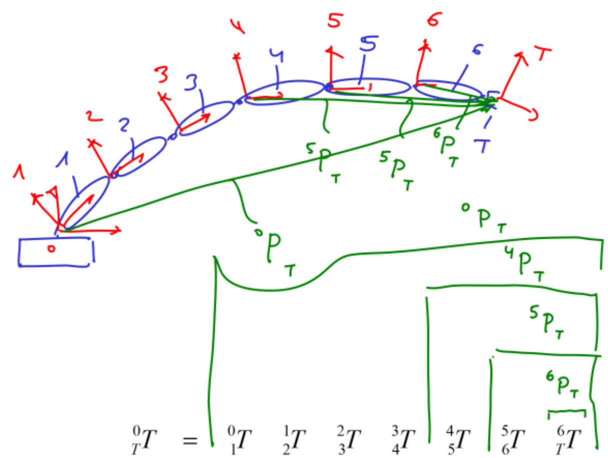
Position Propagation

- The homogeneous transform matrix provides a complete description of the linear and angular position relationship between adjacent links.

$$ {}^{i}_{i-1}T = \begin{bmatrix} {}^{i}_{i-1}R & {}^{i}P_{i-1} \\ 0 & 1 \end{bmatrix} $$

- These descriptions may be combined together to describe the position of a link relative to the robot base frame {0}.

$$ {}^{o}_{i}T = {}^{o}_{1}T\,{}^{1}_{2}T \cdots {}^{i-1}_{i}T $$



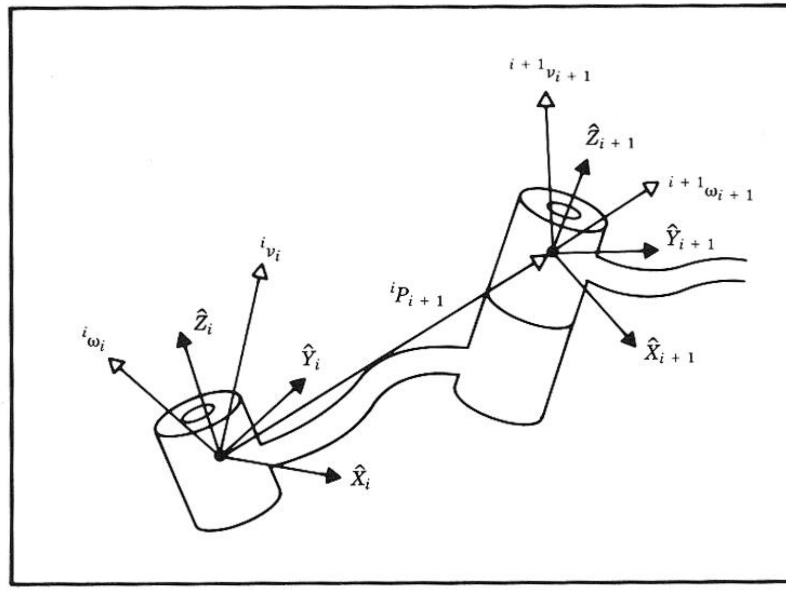
UCLA

Velocity Propagation

- ***Given:*** A manipulator - A chain of rigid bodies each one capable of moving relative to its neighbor

- ***Problem:*** Calculate the linear and angular velocities of the link of a robot

- ***Solution (Concept):*** Due to the robot structure (serial mechanism) we can **compute the velocities** of each link in order **starting from the base**.



The velocity of link *i*+1
= The velocity of link *i*
+ whatever new velocity components were added by joint *i*+1

Velocity Propagation – Intuitive Explanation

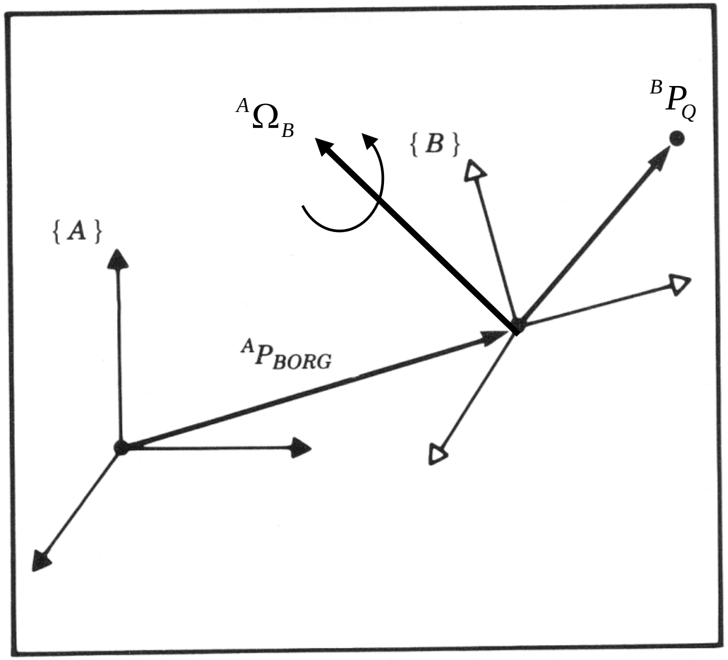
- Three Actions
  - The origin of frame B moves as a function of time with respect to the origin of frame A
  - Point Q moves with respect to frame B
  - Frame B rotates with respect to frame A along an axis defined by ${}^{A}\Omega_{B}$

- Linear and Rotational Velocity
  - Vector Form

$$\boxed{{}^{A}V_{Q} = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_{Q} + {}^{A}\Omega_{B} \times {}^{A}_{B}R\,{}^{B}P_{Q}}$$

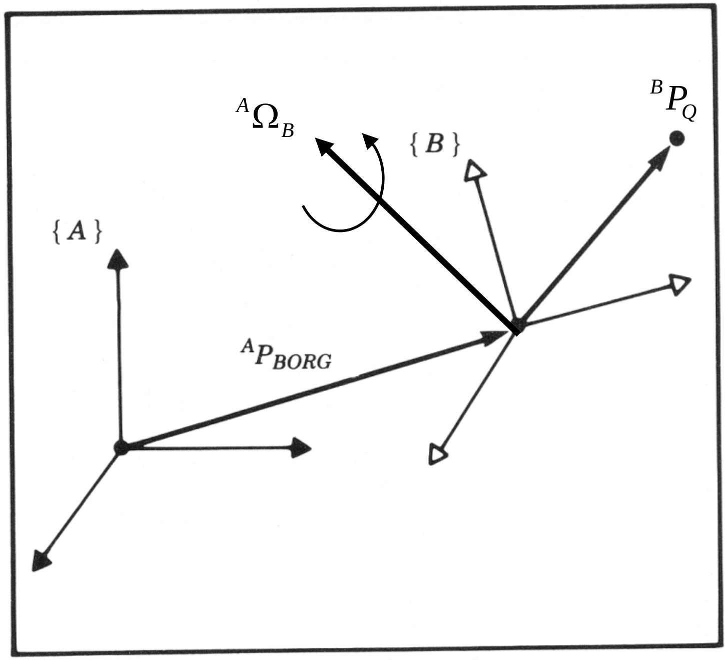
  - Matrix Form

$$\boxed{{}^{A}V_{Q} = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_{Q} + {}^{A}_{B}\dot{R}_{\Omega}\left({}^{A}_{B}R\,{}^{B}P_{Q}\right)}$$



Velocity Propagation – Intuitive Explanation

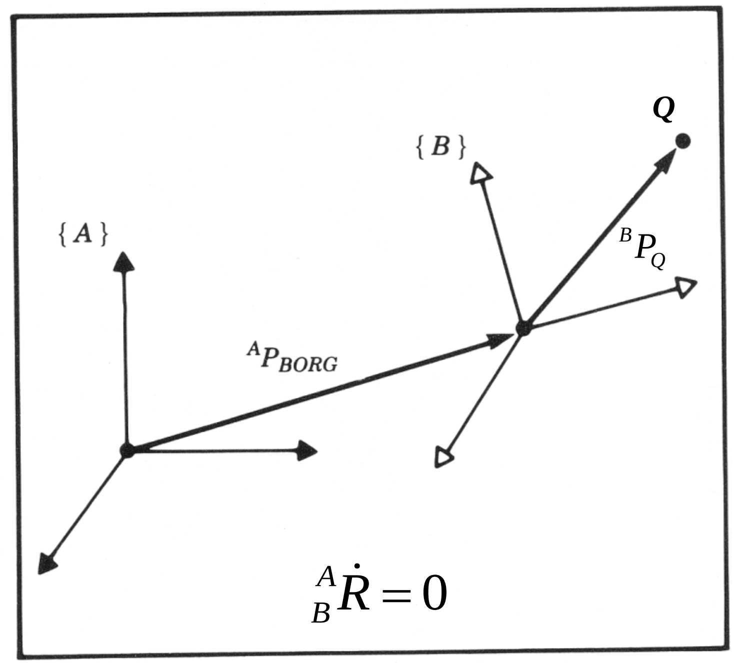
- Three Actions
  - The origin of frame B moves with respect to the origin of frame A
  - Point Q moves with respect to frame B
  - Frame B rotates with respect to frame A about an axis defined by ${}^{A}\Omega_{B}$



$$ {}^{A}V_{Q} = \boxed{{}^{A}V_{BORG}} + \boxed{{}^{A}_{B}R\,{}^{B}V_{Q}} + {}^{A}\Omega_{B} \times {}^{A}_{B}R\,{}^{B}P_{Q} \qquad {}^{A}V_{Q} = \boxed{{}^{A}V_{BORG}} + \boxed{{}^{A}_{B}R\,{}^{B}V_{Q}} + {}^{A}_{B}\dot{R}_{\Omega}\left({}^{A}_{B}R\,{}^{B}P_{Q}\right) $$

Linear Velocity - Rigid Body

- ***Given:*** Consider a frame {B} attached to a rigid body whereas frame {A} is fixed. The orientation of frame {A} with respect to frame {B} is not changing as a function of time $\,{}^{A}_{B}\dot{R} = 0$
- ***Problem:*** describe the motion of of the vector ${}^{B}P_{Q}$ relative to frame {A}
- ***Solution:*** Frame {B} is located relative to frame {A} by a position vector ${}^{A}P_{BORG}$ and the rotation matrix ${}^{A}_{B}R$ (assume that the orientation is not changing in time ${}^{A}_{B}\dot{R} = 0$) expressing both components of the velocity in terms of frame {A} gives



$$ \boxed{{}^{A}V_{Q} = {}^{A}V_{BORG} + {}^{A}({}^{B}V_{Q}) = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_{Q}} $$

Frame - Velocity

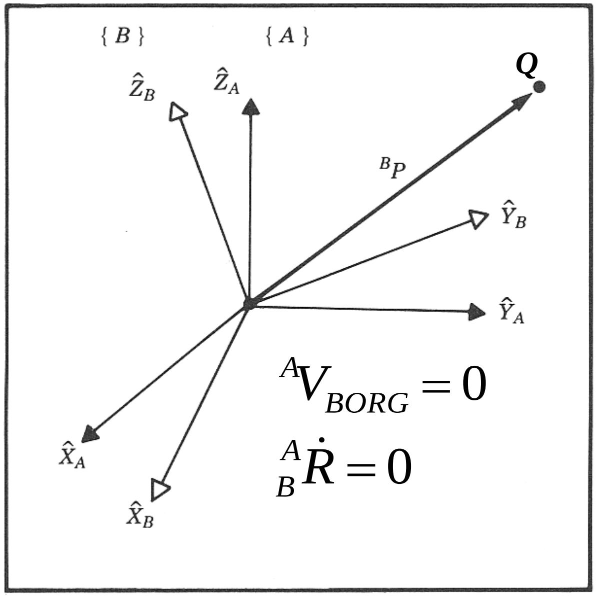
- As with any vector, a velocity vector may be described in terms of any frame, and this frame of reference is noted with a leading superscript.

- A velocity vector **computed** in frame {B} and **represented** in frame {A} would be written

Represented (Reference Frame)

$$ {}^{A}({}^{B}V_{Q}) = \frac{{}^{A}d}{dt}\,{}^{B}P_{Q} $$

Computed (Measured)



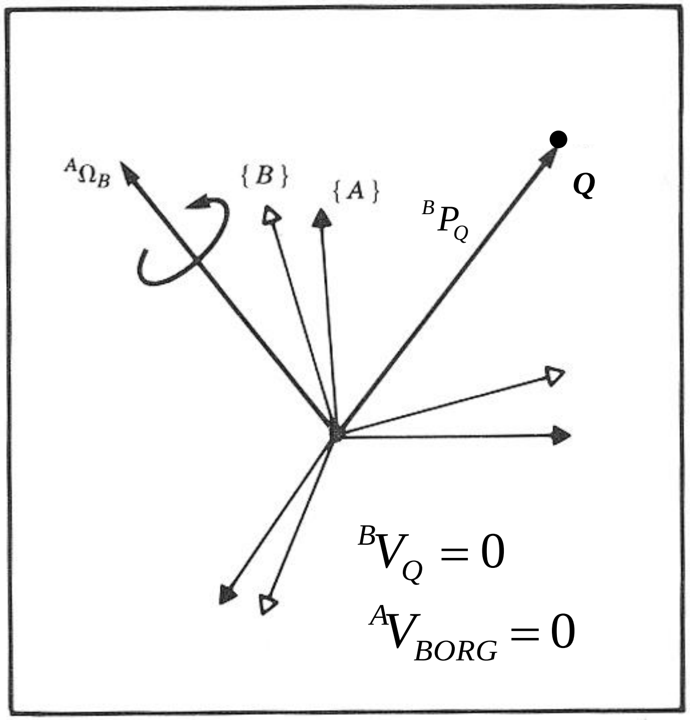
$$ {}^{A}V_{Q} = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_{Q} + \boxed{{}^{A}\Omega_{B}} \times {}^{A}_{B}R\,{}^{B}P_{Q} \qquad {}^{A}V_{Q} = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_{Q} + {}^{A}_{B}\dot{R}_{\Omega}\left({}^{A}_{B}R\,{}^{B}P_{Q}\right) $$

Angular Velocity - Rigid Body

- ***Given:*** Consider a frame {B} attached to a rigid body whereas frame {A} is fixed. The vector ${}^{B}P_{Q}$ is constant as view from frame {B} ${}^{B}V_{Q} = 0$

- ***Problem:*** describe the velocity of the vector ${}^{B}P_{Q}$ representing the the point $\mathbf{Q}$ relative to frame {A}

- ***Solution:*** Even though the vector ${}^{B}P_{Q}$ is constant as view from frame {B} it is clear that point $\mathbf{Q}$ will have a velocity as seen from frame {A} due to the rotational velocity ${}^{A}\Omega_{B}$



$${}^{A}V_{Q} = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_{Q} + \boxed{{}^{A}\Omega_{B} \times {}^{A}_{B}R\,{}^{B}P_{Q}} \qquad {}^{A}V_{Q} = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_{Q} + {}^{A}_{B}\dot{R}_{\Omega}\left({}^{A}_{B}R\,{}^{B}P_{Q}\right)$$

Angular Velocity - Rigid Body - Intuitive Approach

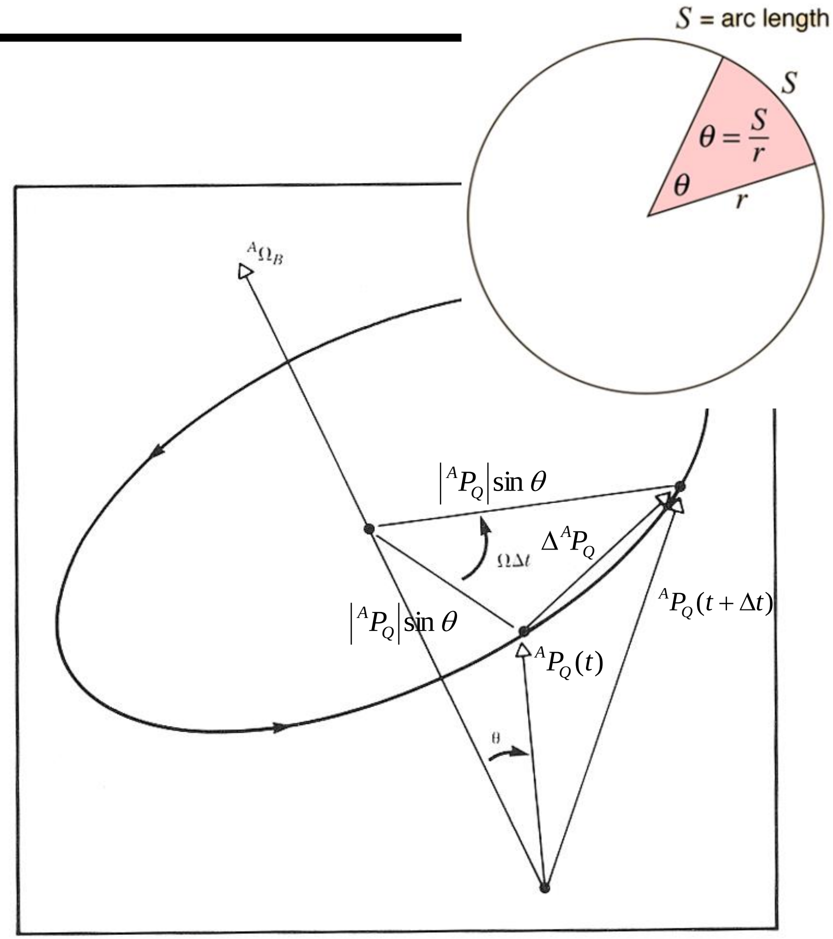
- The figure shows to instants of time as the vector ${}^{A}P_{Q}$ rotates around ${}^{A}\Omega_{B}$ This is what an observer in frame {A} would observe.

- The Magnitude of the differential change is

$$\left|\Delta\,{}^{A}P_{Q}\right| = \left(\left|{}^{A}\Omega_{B}\right|\Delta t\right)\left(\left|{}^{A}P_{Q}\right|\sin\theta\right)$$

- Using a vector cross product we get

$$\frac{\Delta\,{}^{A}P_{Q}}{\Delta t} = {}^{A}V_{Q} = {}^{A}\Omega_{B} \times {}^{A}P_{Q}$$



Simultaneous Linear and Rotational Velocity

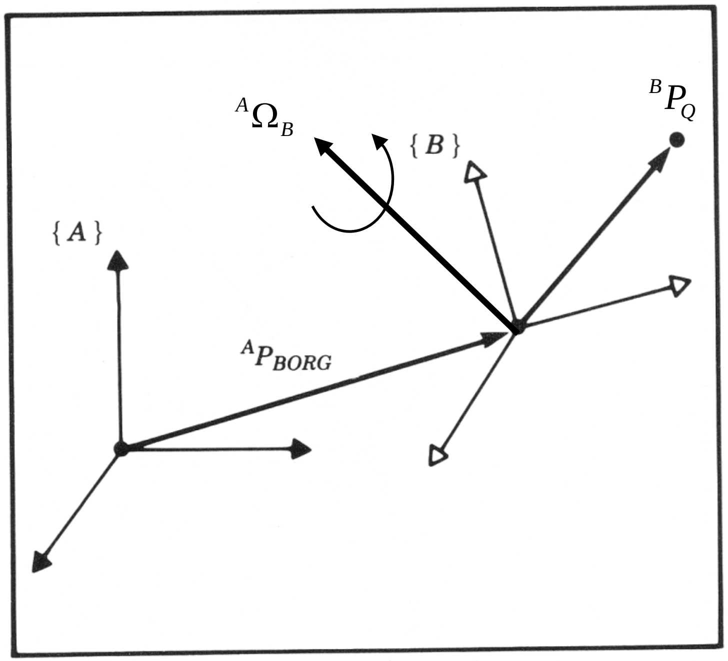
- The final results for the derivative of a vector in a moving frame (linear and rotation velocities) as seen from a stationary frame

- Vector Form

$$\boxed{{}^{A}V_{Q} = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_{Q} + {}^{A}\Omega_{B} \times {}^{A}_{B}R\,{}^{B}P_{Q}}$$

- Matrix Form

$$\boxed{{}^{A}V_{Q} = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_{Q} + {}^{A}_{B}\dot{R}_{\Omega}\left({}^{A}_{B}R\,{}^{B}P_{Q}\right)}$$



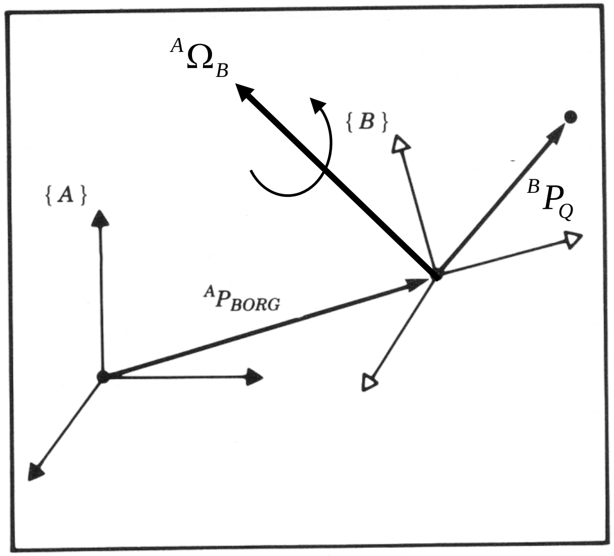
Simultaneous Linear and Rotational Velocity

- Linear and Rotational Velocity
  - Vector Form

$$ {}^{A}V_Q = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_Q + {}^{A}\Omega_B \times {}^{A}_{B}R\,{}^{B}P_Q $$

  - Matrix Form

$$ {}^{A}V_Q = {}^{A}V_{BORG} + {}^{A}_{B}R\,{}^{B}V_Q + {}^{A}_{B}\dot{R}_\Omega \left({}^{A}_{B}R\,{}^{B}P_Q\right) $$



- Angular Velocity
  - Vector Form

$$ {}^{A}\Omega_C = {}^{A}\Omega_B + {}^{A}_{B}R\,{}^{B}\Omega_C $$

  - Matrix Form

$$ {}^{A}_{C}\dot{R}_\Omega = {}^{A}_{B}\dot{R}_\Omega + {}^{A}_{B}R\,{}^{B}_{C}\dot{R}_\Omega\,{}^{A}_{B}R^T $$

Velocity of Adjacent Links - Summary

- Angular Velocity

0 - Prismatic Joint

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{\ i}R\,{}^{i}\omega_i + f \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix} $$

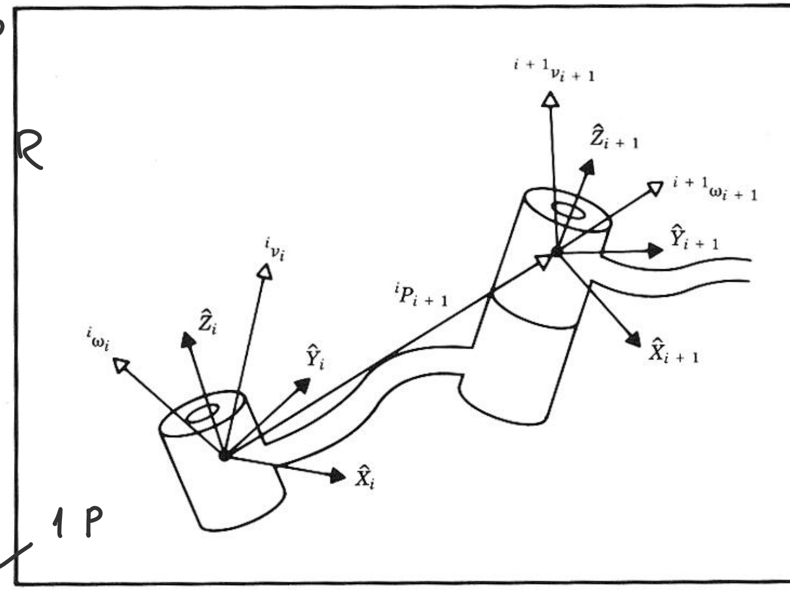
f 0-P
1-R

- Linear Velocity

0 - Revolute Joint

$$ {}^{i+1}v_{i+1} = {}^{i+1}_{\ i}R\left({}^{i}\omega \times {}^{i}P_{i+1} + {}^{i}v_i\right) + f \begin{bmatrix} 0 \\ 0 \\ \dot{d}_{i+1} \end{bmatrix} $$

f 1 P
0 R



The velocity of link i+1
= The velocity of link i
+ whatever new velocity components were added by joint i+1

Jacobian: Velocity propagation

- Therefore the recursive expressions for the adjacent joint linear and angular velocities can be used to determine the Jacobian in the end effector frame

$$ {}^{N}\dot{X} = {}^{N}J(\theta)\dot{\theta} $$

- This equation can be expanded to:

$$ {}^{N}\dot{X} = {}^{N}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \Omega x \\ \Omega y \\ \Omega z \end{bmatrix} = \begin{bmatrix} {}^{N}v_{N} \\ {}^{N}\omega_{N} \end{bmatrix} = {}^{N}\begin{bmatrix} & J(\theta) & \end{bmatrix} \begin{bmatrix} \dot{\theta}1 \\ \dot{\theta}2 \\ \cdots \\ \dot{\theta}_{N} \end{bmatrix} $$

UCLA

Velocity of Adjacent Links - Angular Velocity 5/5

- The result is a ***recursive equation*** that shows the angular velocity of one link in terms of the angular velocity of the previous link plus the relative motion of the two links.

$$\boxed{{}^{i+1}\omega_{i+1} = {}^{i+1}_{\;\;i}R\,{}^{i}\omega_i + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix}}$$

- Since the term ${}^{i+1}\omega_{i+1}$ depends on all previous links through this recursion, the angular velocity is said to propagate from the base to subsequent links.

Velocity of Adjacent Links - Angular Velocity 1/5

- From the relationship developed previously

$$ {}^{A}\Omega_{C} = {}^{A}\Omega_{B} + {}^{A}_{B}R\,{}^{B}\Omega_{C} $$

- we can re-assign link names to calculate the velocity of any link i relative to the base frame {0}

$$ \begin{cases} A \rightarrow 0 \\ B \rightarrow i \\ C \rightarrow i+1 \end{cases} $$

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

- By pre-multiplying both sides of the equation by ${}^{i+1}_{0}R$,we can convert the frame of reference for the base {0} to frame {i+1}

Velocity of Adjacent Links - Angular Velocity 2/5

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

- Using the recently defined notation, we have

$$ \boxed{{}^{i+1}\omega_{i+1} = {}^{i+1}\omega_{i} + {}^{i+1}_{i}R\,{}^{i}\Omega_{i+1}} $$

${}^{i+1}\omega_{i+1}$ - Angular velocity of frame {i+1} measured relative to the robot base, and expressed in frame {i+1} - **Recall the car example** ${}^{c}\left[{}^{w}V_{c}\right] = {}^{c}v_{c}$

${}^{i+1}\omega_{i}$ - Angular velocity of frame {i} measured relative to the robot base, and expressed in frame {i+1}

${}^{i+1}_{i}R\,{}^{i}\Omega_{i+1}$ - Angular velocity of frame {$i+1$} measured relative to frame {i} and expressed in frame {i+1}

Velocity of Adjacent Links - Angular Velocity 3/5

$$ {}^{i+1}\omega_{i+1} = \boxed{{}^{i+1}\omega_i} + {}^{i+1}_{\;\;i}R\,{}^{i}\Omega_{i+1} $$

- Angular velocity of frame {i} measured relative to the robot base, expressed in frame {i+1}

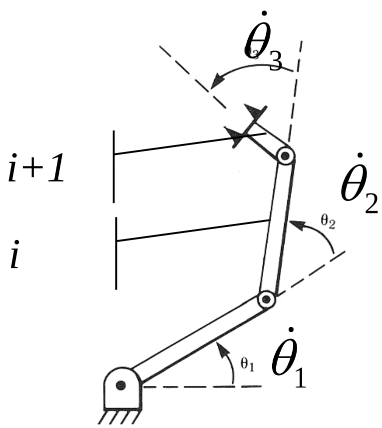
$$ {}^{i+1}\omega_i = {}^{i+1}_{\;\;i}R\,\underbrace{{}^{i}\omega_i} $$

$$ {}^{i+1}_{\;\;i}R\left[{}^{i}_{0}R\,{}^{0}\Omega_i\right] $$

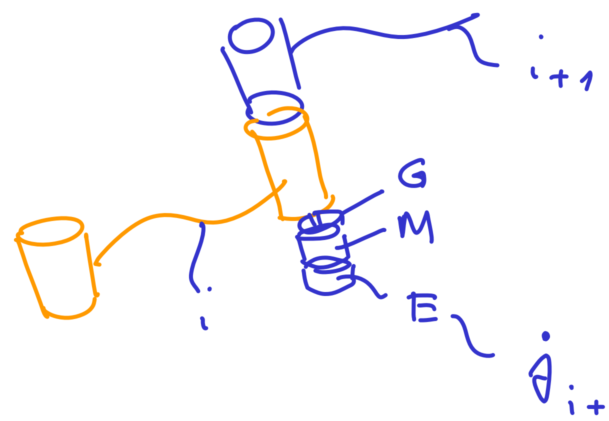
Velocity of Adjacent Links - Angular Velocity 4/5

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

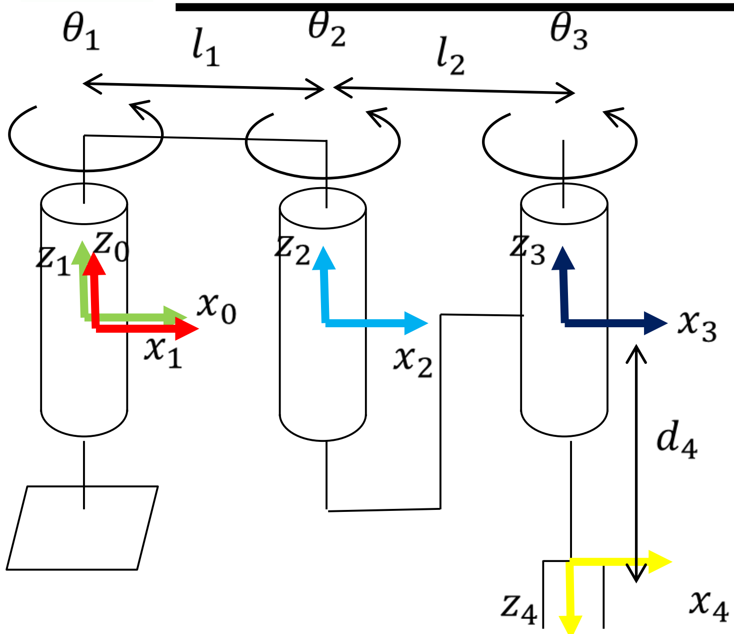
- Angular velocity of frame $\{i+1\}$ measured (differentiate) in frame $\{i\}$ and represented (expressed) in frame $\{i+1\}$
- Assuming that a joint has only 1 DOF. The joint configuration can be either revolute joint (***angular velocity***) or prismatic joint (Linear velocity).
- Based on the frame attachment convention in which we assign the Z axis pointing along the $i+1$ joint axis such that the two are coincide (rotations of a link is preformed only along its Z- axis) we can rewrite this term as follows:



$$ {}^{i+1}_{\ \ i}R\,{}^{i}\Omega_{i+1} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix} $$



SCARA – RRRP – DH Parameter (Modified form)



Modified form:

a_{i-1} - The distance from $\hat{Z}_{i-1}$ to $\hat{Z}_i$ measured along $\hat{X}_{i-1}$

α_{i-1} - The angle between $\hat{Z}_{i-1}$ and $\hat{Z}_i$ measured about $\hat{X}_{i-1}$

d_i - The distance from $\hat{X}_{i-1}$ to $\hat{X}_i$ measured along $\hat{Z}_i$

θ_i - The angle between $\hat{X}_{i-1}$ and $\hat{X}_i$ measured about $\hat{Z}_i$

i-1	*i*	α_{i-1}	a_{i-1}	d_i	θ_i
0	1	0	0	0	θ_1
1	2	0	l_1	0	θ_2
2	3	0	l_2	0	θ_3
3	4	180°	0	d_4	0

SCARA – RRRP

Modified DH parameter

```
syms pi

L(1) = Link('revolute','d',0,'a',0,'alpha',0,'modified');
L(2) = Link('revolute','d',0,'a',l1,'alpha',0,'modified');
L(3) = Link('revolute','d',0,'a',l2,'alpha',0,'modified');
L(4) = Link('prismatic','alpha',pi,'theta', 0 ,'a',0,'modified')

SCARA = SerialLink(L,'name','SCARA')
```

```
SCARA:: 4 axis, RRRP, modDH, slowRNE
+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|          0|          0|          0|          0|
|  2|         q2|          0|         l1|          0|          0|
|  3|         q3|          0|         l2|          0|          0|
|  4|          0|         q4|          0|         pi|          0|
+---+-----------+-----------+-----------+-----------+-----------+
```

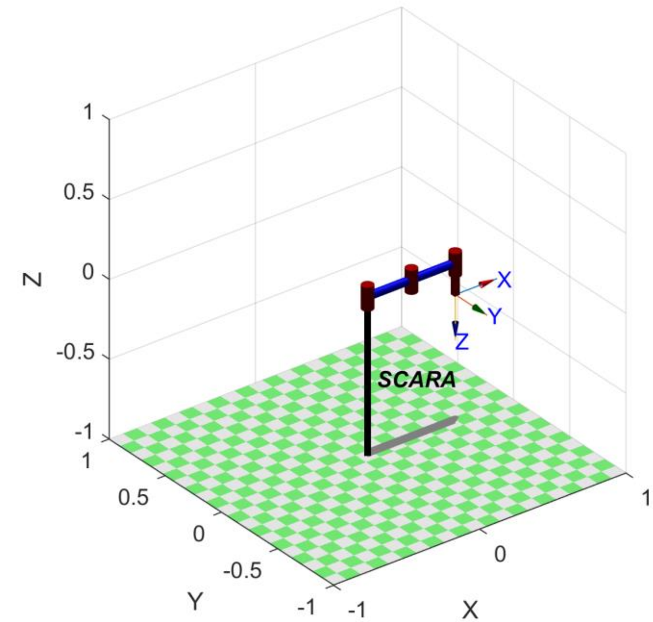
For plot [mm]

```
l1 = 0.3; l2=0.3; pi= 3.14;
t1 =0; t2=0; t3=0; d4=0.2;
th= [t1 t2 t3 d4];

L_P(1) = Link('revolute','d',0,'a',0,'alpha',0,'modified');
L_P(2) = Link('revolute','d',0,'a',l1,'alpha',0,'modified');
L_P(3) = Link('revolute','d',0,'a',l2,'alpha',0,'modified');
L_P(4) = Link('prismatic','alpha',pi,'theta', 0 ,'a',0,'modified')

SCARA_P = SerialLink(L_P,'name','SCARA')

figure()
SCARA_P.plot(th,'workspace',[-1 1 -1 1 -1 1])
```



SCARA – RRRP – Forward Kinematics

```
q = [t1 t2 t3 d4]
```

```
T0_1 = SCARA.A([1],q)
T1_2 = SCARA.A([2],q)
T2_3 = SCARA.A([3],q)
T3_4 = SCARA.A([4],q)
T0_4 = simplify(SCARA.A([1 2 3 4],q))
```

```
T0_2 = simplify(SCARA.A([1,2],q))
T0_3 = simplify(SCARA.A([1,2,3],q))
```

T0_2 =

$$\begin{pmatrix} \cos(t_1+t_2) & -\sin(t_1+t_2) & 0 & l_1\cos(t_1) \\ \sin(t_1+t_2) & \cos(t_1+t_2) & 0 & l_1\sin(t_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T0_3 =

$$\begin{pmatrix} \cos(t_1+t_2+t_3) & -\sin(t_1+t_2+t_3) & 0 & l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ \sin(t_1+t_2+t_3) & \cos(t_1+t_2+t_3) & 0 & l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T0_1 =

$$\begin{pmatrix} \cos(t_1) & -\sin(t_1) & 0 & 0 \\ \sin(t_1) & \cos(t_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T1_2 =

$$\begin{pmatrix} \cos(t_2) & -\sin(t_2) & 0 & l_1 \\ \sin(t_2) & \cos(t_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T2_3 =

$$\begin{pmatrix} \cos(t_3) & -\sin(t_3) & 0 & l_2 \\ \sin(t_3) & \cos(t_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T3_4 =

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T0_4 =

$$\begin{pmatrix} \cos(t_1+t_2+t_3) & \sin(t_1+t_2+t_3) & 0 & l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ \sin(t_1+t_2+t_3) & -\cos(t_1+t_2+t_3) & 0 & l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ 0 & 0 & -1 & -d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Simplify Function

```
T0_4 = SCARA.A([1 2 3 4],q)
```

T0_4 =

$$\begin{pmatrix} \cos(t_3)\,\sigma_2 - \sin(t_3)\,\sigma_3 & \sigma_1 & 0 & l_2\,\sigma_2 + l_1\cos(t_1) \\ \sigma_1 & \sin(t_3)\,\sigma_3 - \cos(t_3)\,\sigma_2 & 0 & l_2\,\sigma_3 + l_1\sin(t_1) \\ 0 & 0 & -1 & -d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = \cos(t_3)\,\sigma_3 + \sin(t_3)\,\sigma_2$$

$$\sigma_2 = \cos(t_1)\cos(t_2) - \sin(t_1)\sin(t_2)$$

$$\sigma_3 = \cos(t_1)\sin(t_2) + \cos(t_2)\sin(t_1)$$

```
simplified_T0_4 = simplify(SCARA.A([1 2 3 4],q))
```

simplified_T0_4 =

$$\begin{pmatrix} \cos(t_1+t_2+t_3) & \sin(t_1+t_2+t_3) & 0 & l_2\cos(t_1+t_2) + l_1\cos(t_1) \\ \sin(t_1+t_2+t_3) & -\cos(t_1+t_2+t_3) & 0 & l_2\sin(t_1+t_2) + l_1\sin(t_1) \\ 0 & 0 & -1 & -d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

SCARA example _ Jacobian: Velocity propagation

- The recursive equation for the Angular Velocity is

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{\ i}R\ {}^{i}\omega_i + \rho \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix}, \begin{matrix} \rho = 0\ in\ the\ \ prismatic\ joint \\ \boldsymbol{\rho = 1\ in\ the\ \ revolute\ joint} \end{matrix} $$

${}^{i+1}_{\ i}R$ is the transpose of ${}_{i+1}^{\ \ i}R$ (${}^{i+1}_{\ i}R = \left({}_{i+1}^{\ \ i}R\right)^T$) and ${}_{i+1}^{\ \ i}R$ = ${}_{i+1}^{\ \ i}T(1:3,1:3)$

${}_{i+1}^{\ \ i}R$ can be obtained from the transformation matrix in the forward kinematics.

SCARA example _ Jacobian: Velocity propagation

```
R0_1 = t2r(T0_1)
R1_2 = t2r(T1_2)
R2_3 = t2r(T2_3)
R3_4 = t2r(T3_4)
```

R0_1 =

$$\begin{pmatrix} \cos(t_1) & -\sin(t_1) & 0 \\ \sin(t_1) & \cos(t_1) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

R1_2 =

$$\begin{pmatrix} \cos(t_2) & -\sin(t_2) & 0 \\ \sin(t_2) & \cos(t_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

R2_3 =

$$\begin{pmatrix} \cos(t_3) & -\sin(t_3) & 0 \\ \sin(t_3) & \cos(t_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

R3_4 =

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

```
R1_0 = transpose(R0_1)
R2_1 = transpose(R1_2)
R3_2 = transpose(R2_3)
R4_3 = transpose(R3_4)
```

R1_0 =

$$\begin{pmatrix} \cos(t_1) & \sin(t_1) & 0 \\ -\sin(t_1) & \cos(t_1) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

R2_1 =

$$\begin{pmatrix} \cos(t_2) & \sin(t_2) & 0 \\ -\sin(t_2) & \cos(t_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

R3_2 =

$$\begin{pmatrix} \cos(t_3) & \sin(t_3) & 0 \\ -\sin(t_3) & \cos(t_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

R4_3 =

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

SCARA example _ Jacobian: Velocity propagation

- The recursive equation for the Angular Velocity is

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\ {}^{i}\omega_i + \rho \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix}, \begin{matrix} \rho = 0\ \mathit{in\ the\ \ prismatic\ joint} \\ \boldsymbol{\rho = 1\ in\ the\ \ revolute\ joint} \end{matrix}$$

- **The base frame does not move**

$$^{0}\omega_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{(The base frame does not move)}$$

- **Three revolute joints**

$$^{1}\omega_1 = {}^{1}_{0}R\ {}^{0}\omega_0 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \qquad ^{2}\omega_2 = {}^{2}_{1}R\ {}^{1}\omega_1 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} \qquad ^{3}\omega_3 = {}^{3}_{2}R\ {}^{2}\omega_2 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix}$$

- **One prismatic joint**

$$^{4}\omega_4 = {}^{4}_{3}R\ {}^{3}\omega_3 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_4 = 0 \end{bmatrix}$$

SCARA example _ Jacobian: Velocity propagation

- The recursive equation for the Angular Velocity is

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\ {}^{i}\omega_i + \rho \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix}, \quad \begin{array}{l} \rho = 0 \textit{ in the prismatic joint} \\ \boldsymbol{\rho = 1 \textit{ in the revolute joint}} \end{array}$$

- **The base frame does not move**

$$^{0}\omega_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{(The base frame does not move)}$$

- **Three revolute joints**

$$^{1}\omega_1 = {}^{1}_{0}R\ {}^{0}\omega_0 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \qquad ^{2}\omega_2 = {}^{2}_{1}R\ {}^{1}\omega_1 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} \qquad ^{3}\omega_3 = {}^{3}_{2}R\ {}^{2}\omega_2 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix}$$

- **One prismatic joint**

$$^{4}\omega_4 = {}^{4}_{3}R\ {}^{3}\omega_3 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_4 = 0 \end{bmatrix}$$

```
% Angular Velocity Propagation
w0 = [0; 0; 0;]
w1 = [0; 0; dt1] % for a revolute joint
w2 = R2_1*w1 + [0; 0; dt2;] % for a revolute joint
w3 = simplify(R3_2*w2 + [0; 0; dt3;]) % for a revolute joint
w4 = simplify(R4_3*w3)% for a prismatic joint
```

w1 =

$$\begin{pmatrix} 0 \\ 0 \\ \mathrm{dt}_1 \end{pmatrix}$$

w2 =

$$\begin{pmatrix} 0 \\ 0 \\ \mathrm{dt}_1 + \mathrm{dt}_2 \end{pmatrix}$$

w3 =

$$\begin{pmatrix} 0 \\ 0 \\ \mathrm{dt}_1 + \mathrm{dt}_2 + \mathrm{dt}_3 \end{pmatrix}$$

w4 =

$$\begin{pmatrix} 0 \\ 0 \\ -\mathrm{dt}_1 - \mathrm{dt}_2 - \mathrm{dt}_3 \end{pmatrix}$$

SCARA example _ Jacobian: Velocity propagation

- The recursive equation for Linear Velocity is

$$ {}^{i+1}v_{i+1} = {}^{i+1}_{i}R\left({}^{i}\omega_i \times {}^{i}P_{i+1} + {}^{i}v_i \right) + \rho \begin{bmatrix} 0 \\ 0 \\ \dot{d} \end{bmatrix}, \quad \begin{matrix} \rho = 1 \; in \; the \; prismatic \; joint \\ \boldsymbol{\rho = 0 \; in \; the \; revolute \; joint} \end{matrix} $$

- **The base frame does not move**

$$ {}^{0}v_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} $$

- **Three revolute joints**

$$ \begin{aligned} {}^{1}v_1 &= {}^{1}_{0}R\left({}^{0}\omega_0 \times {}^{0}P_1 + {}^{0}v_0 \right) \\ {}^{2}v_2 &= {}^{2}_{1}R\left({}^{1}\omega_1 \times {}^{1}P_2 + {}^{1}v_1 \right) \\ {}^{3}v_3 &= {}^{3}_{2}R\left({}^{2}\omega_2 \times {}^{2}P_3 + {}^{2}v_2 \right) \end{aligned} $$

- **One prismatic joint**

$$ {}^{4}v_4 = {}^{4}_{3}R\left({}^{3}\omega_3 \times {}^{3}P_4 + {}^{3}v_3 \right) + \begin{bmatrix} 0 \\ 0 \\ \dot{d}_4 \end{bmatrix} $$

SCARA example
Jacobian: Velocity propagation

- The recursive equation for Linear Velocity is

$$ {}^{i+1}v_{i+1} = {}^{i+1}_{i}R\left({}^{i}\omega_i \times {}^{i}P_{i+1} + {}^{i}v_i \right) + \rho \begin{bmatrix} 0 \\ 0 \\ \dot{d} \end{bmatrix}, \quad \begin{array}{l} \rho = 1 \ \textit{in the prismatic joint} \\ \boldsymbol{\rho = 0 \ in \ the \ revolute \ joint} \end{array} $$

- **The base frame does not move**

$$ {}^{0}v_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} $$

- **Three revolute joints**

$$ \begin{aligned} {}^{1}v_1 &= {}^{1}_{0}R\left({}^{0}\omega_0 \times {}^{0}P_1 + {}^{0}v_0 \right) \\ {}^{2}v_2 &= {}^{2}_{1}R\left({}^{1}\omega_1 \times {}^{1}P_2 + {}^{1}v_1 \right) \\ {}^{3}v_3 &= {}^{3}_{2}R\left({}^{2}\omega_2 \times {}^{2}P_3 + {}^{2}v_2 \right) \end{aligned} $$

- **One prismatic joint**

$$ {}^{4}v_4 = {}^{4}_{3}R\left({}^{3}\omega_3 \times {}^{3}P_4 + {}^{3}v_3 \right) + \begin{bmatrix} 0 \\ 0 \\ \dot{d}_4 \end{bmatrix} $$

```
% Linear Velocity Propagation
v0 = [0; 0; 0;]
v1 = collect(R1_0*(cross(w0,P0_1)+v0),[dt1 dt2 dt3 dd4])
v2 = collect(R2_1*(cross(w1,P1_2)+v1),[dt1 dt2 dt3 dd4])
v3_c = collect(R3_2*(cross(w2,P2_3)+v2),[dt1 dt2 dt3 dd4])
v3_s = simplify(v3_c)
v3 = collect(v3_s,[dt1 dt2 dt3 dd4])
v4 = collect(R4_3*(cross(w3,P3_4)+v3)+[0;0;dd4],[dt1 dt2 dt3 dd4])
```

```
v0 = 3×1
     0
     0
     0
```

v1 =

$$ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} $$

v2 =

$$ \begin{pmatrix} (l_1 \sin(t_2))\, \mathrm{dt}_1 \\ (l_1 \cos(t_2))\, \mathrm{dt}_1 \\ 0 \end{pmatrix} $$

v3_c =

$$ \begin{pmatrix} (\sin(t_3)\ (l_2 + l_1 \cos(t_2)) + l_1 \cos(t_3) \sin(t_2))\, \mathrm{dt}_1 + (l_2 \sin(t_3))\, \mathrm{dt}_2 \\ (\cos(t_3)\ (l_2 + l_1 \cos(t_2)) - l_1 \sin(t_2) \sin(t_3))\, \mathrm{dt}_1 + (l_2 \cos(t_3))\, \mathrm{dt}_2 \\ 0 \end{pmatrix} $$

v3_s =

$$ \begin{pmatrix} \mathrm{dt}_1\, l_2 \sin(t_3) + \mathrm{dt}_2\, l_2 \sin(t_3) + \mathrm{dt}_1\, l_1 \sin(t_2 + t_3) \\ \mathrm{dt}_1\, l_2 \cos(t_3) + \mathrm{dt}_2\, l_2 \cos(t_3) + \mathrm{dt}_1\, l_1 \cos(t_2 + t_3) \\ 0 \end{pmatrix} $$

v3 =

$$ \begin{pmatrix} (l_1 \sin(t_2 + t_3) + l_2 \sin(t_3))\, \mathrm{dt}_1 + (l_2 \sin(t_3))\, \mathrm{dt}_2 \\ (l_1 \cos(t_2 + t_3) + l_2 \cos(t_3))\, \mathrm{dt}_1 + (l_2 \cos(t_3))\, \mathrm{dt}_2 \\ 0 \end{pmatrix} $$

v4 =

$$ \begin{pmatrix} (l_1 \sin(t_2 + t_3) + l_2 \sin(t_3))\, \mathrm{dt}_1 + (l_2 \sin(t_3))\, \mathrm{dt}_2 \\ (-l_1 \cos(t_2 + t_3) - l_2 \cos(t_3))\, \mathrm{dt}_1 + (-l_2 \cos(t_3))\, \mathrm{dt}_2 \\ \mathrm{dd}_4 \end{pmatrix} $$

SCARA example _ Jacobian: Velocity propagation

- The Jacobian ${}^{4}J_4$ is defined as $\begin{bmatrix} {}^{4}v_4 \\ {}^{4}\omega_4 \end{bmatrix} = \begin{bmatrix} {}^{4}J_4 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{d}_4 \end{bmatrix}$

- According to the definition $\begin{bmatrix} {}^{4}v_4 \\ {}^{4}\omega_4 \end{bmatrix} = \begin{bmatrix} {}^{4}v_{4,1} \\ {}^{4}v_{4,2} \\ {}^{4}v_{4,3} \\ {}^{4}w_{4,1} \\ {}^{4}w_{4,2} \\ {}^{4}w_{4,3} \end{bmatrix}, \quad \begin{bmatrix} {}^{4}v_{4,1} \\ {}^{4}v_{4,2} \\ {}^{4}v_{4,3} \\ {}^{4}w_{4,1} \\ {}^{4}w_{4,2} \\ {}^{4}w_{4,3} \end{bmatrix} = \begin{bmatrix} {}^{4}J_4 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{d}_4 \end{bmatrix}$

SCARA example _ Jacobian: Velocity propagation

```
[JV4]=equationsToMatrix([v4],[dt1 dt2 dt3 dd4])
[JW4]=equationsToMatrix([w4],[dt1 dt2 dt3 dd4])
J4_VP=simplify([JV4;JW4])
```

v4 =

$$\begin{pmatrix} (l_1 \sin(t_2+t_3) + l_2 \sin(t_3))\,\mathrm{dt}_1 + (l_2 \sin(t_3))\,\mathrm{dt}_2 \\ (-l_1 \cos(t_2+t_3) - l_2 \cos(t_3))\,\mathrm{dt}_1 + (-l_2 \cos(t_3))\,\mathrm{dt}_2 \\ \mathrm{dd}_4 \end{pmatrix}$$

w4 =

$$\begin{pmatrix} 0 \\ 0 \\ -\mathrm{dt}_1 - \mathrm{dt}_2 - \mathrm{dt}_3 \end{pmatrix}$$

JV4 =

$$\begin{pmatrix} l_1 \sin(t_2+t_3) + l_2 \sin(t_3) & l_2 \sin(t_3) & 0 & 0 \\ -l_1 \cos(t_2+t_3) - l_2 \cos(t_3) & -l_2 \cos(t_3) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

JW4 =

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 \end{pmatrix}$$

J4_VP =

$$\begin{pmatrix} l_1 \sin(t_2+t_3) + l_2 \sin(t_3) & l_2 \sin(t_3) & 0 & 0 \\ -l_1 \cos(t_2+t_3) - l_2 \cos(t_3) & -l_2 \cos(t_3) & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 \end{pmatrix}$$

Jacobian: Frame of Representation

- The Jacobian provides the relationship between the end effector’s Cartesian velocity measured relative to the robot base frame {0}

- For velocity expressed in frame {N}

$$ {}^{N}\dot{X} = {}^{N}J(\theta)\dot{\theta} $$

- For velocity expressed in frame {0}

$$ {}^{0}\dot{X} = {}^{0}J(\theta)\dot{\theta} $$

Jacobian: Frame of Representation

- Consider the velocities in a different frame {B}

$$ {}^{B}\dot{X} = \begin{bmatrix} {}^{B}v_N \\ {}^{B}\omega_N \end{bmatrix} = {}^{B}J(\theta)\dot{\theta} $$

- We may use the rotation matrix to find the velocities in frame {A}:

$$ {}^{A}\dot{X} = \begin{bmatrix} {}^{A}v_N \\ {}^{A}\omega_N \end{bmatrix} = \begin{bmatrix} {}^{A}_{B}R\,{}^{B}v_N \\ {}^{A}_{B}R\,{}^{B}\omega_N \end{bmatrix} $$

FOR 6 DOF

$$ {}^{0}_{6}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T $$

$$ \left[\begin{array}{ccc|c} & \boxed{{}^{0}_{6}R} & & {}^{0}P_6 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] $$

- The Jacobian transformation is given by a rotation matrix

$$ {}^{A}\dot{X} = {}^{A}J(\theta)\dot{\theta} = {}^{A}_{B}R_J\,{}^{B}J(\theta)\dot{\theta} $$

Jacobian: Frame of Representation

- where ${}^{A}_{B}R_{J}$ is given by

$$
{}^{A}_{B}R_{J} = \begin{bmatrix} \begin{bmatrix} {}^{A}_{B}R \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} {}^{A}_{B}R \end{bmatrix} \end{bmatrix}
$$

or equivalently,

$$
{}^{A}J(\theta) = \begin{bmatrix} \begin{bmatrix} {}^{A}_{B}R \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} {}^{A}_{B}R \end{bmatrix} \end{bmatrix} {}^{B}J(\theta)
$$

Jacobian: Frame of Representation - 3R Example

$$
{}^{0}J_4(\theta) = \begin{bmatrix} \begin{bmatrix} {}^{0}_{4}R \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} {}^{0}_{4}R \end{bmatrix} \end{bmatrix} {}^{4}J_4(\theta)
$$

```
% convert 4J4_VP to 0J4_VP
J0_VP=[R0_4 zeros(3,3); zeros(3,3) R0_4]*J4_VP;
J0_VP=simplify(J0_VP,50)
```

J4_VP =

$$
\begin{pmatrix} l_1\sin(t_2+t_3)+l_2\sin(t_3) & l_2\sin(t_3) & 0 & 0 \\ -l_1\cos(t_2+t_3)-l_2\cos(t_3) & -l_2\cos(t_3) & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 \end{pmatrix}
$$

J0_VP =

$$
\begin{pmatrix} -l_2\sin(t_1+t_2)-l_1\sin(t_1) & -l_2\sin(t_1+t_2) & 0 & 0 \\ l_2\cos(t_1+t_2)+l_1\cos(t_1) & l_2\cos(t_1+t_2) & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}
$$

Jacobian Methods of Derivation & the Corresponding Reference Frame – Summary

Method	Jacobian Matrix Reference Frame	Transformation to Base Frame (Frame 0)	
Explicit (Diff. the Forward Kinematic Eq.)	${}^{0}J_{N}$	None	
Iterative Velocity Eq.	${}^{N}J_{N}$	Transform Method 1: Transform Method 2:	${}^{0}v_{N} = {}^{0}_{N}R\,{}^{N}v_{N}$ ${}^{0}\omega_{N} = {}^{0}_{N}R\,{}^{N}\omega_{N}$ ${}^{0}J_{N}(\theta) = \begin{bmatrix} {}^{0}_{N}R & 0 \\ 0 & {}^{0}_{N}R \end{bmatrix} {}^{N}J_{N}(\theta)$
Iterative Force Eq.	${}^{N}J_{N}^{T}$	Transpose Transform	${}^{N}J_{N} = [{}^{N}J_{N}^{T}]^{T}$ ${}^{0}J_{N}(\theta) = \begin{bmatrix} {}^{0}_{N}R & 0 \\ 0 & {}^{0}_{N}R \end{bmatrix} {}^{N}J_{N}(\theta)$

Jacobian: Direct Differentiation

- This expression can be expanded to:

- *Cartesian Space / Operational Space*

$$ {}^{0}_{N}T = \begin{bmatrix} {}^{0}_{N}R & {}^{0}P_N \\ 0 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} \begin{bmatrix} {}^{0}P_N \end{bmatrix} \\ \begin{bmatrix} {}^{0}r_N \end{bmatrix} \end{bmatrix} $$

Euler Angles

$$ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \overbrace{\begin{bmatrix} \\ \\ & J(\underline{\theta}) & \\ \\ \\ \end{bmatrix}}^{N} \left.\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \\ \ldots \\ \\ \dot{\theta}_N \end{bmatrix}\right\} N $$

6x1 **6xN** **Nx1**

$$ \boxed{\begin{aligned} \dot{p}_e &= J_P(q)\dot{q} \\ \omega_e &= J_O(q)\dot{q}. \end{aligned}} $$

Jacobian: Direct Differentiation

$$\dot{\boldsymbol{p}}_e = \boldsymbol{J}_P(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.2}$$

$$\boldsymbol{\omega}_e = \boldsymbol{J}_O(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3.3}$$

In (3.2) $\boldsymbol{J}_P$ is the $(3 \times n)$ matrix relating the contribution of the joint velocities $\dot{\boldsymbol{q}}$ to the end-effector *linear* velocity $\dot{\boldsymbol{p}}_e$, while in (3.3) $\boldsymbol{J}_O$ is the $(3 \times n)$ matrix relating the contribution of the joint velocities $\dot{\boldsymbol{q}}$ to the end-effector *angular* velocity $\boldsymbol{\omega}_e$. In compact form, (3.2), (3.3) can be written as

$$\boldsymbol{v}_e = \begin{bmatrix} \dot{\boldsymbol{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.4}$$

which represents the manipulator *differential kinematics equation*. The $(6 \times n)$ matrix $\boldsymbol{J}$ is the manipulator *geometric Jacobian*

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_P \\ \boldsymbol{J}_O \end{bmatrix}, \tag{3.5}$$

which in general is a function of the joint variables.

Jacobian: Direct Differentiation

```
T0_4 =
```

$$
\begin{pmatrix}
\cos(t_1+t_2+t_3) & \sin(t_1+t_2+t_3) & 0 & \boxed{l_2\cos(t_1+t_2)+l_1\cos(t_1)} \\
\sin(t_1+t_2+t_3) & -\cos(t_1+t_2+t_3) & 0 & l_2\sin(t_1+t_2)+l_1\sin(t_1) \\
0 & 0 & -1 & -d_4 \\
0 & 0 & 0 & 1
\end{pmatrix}
$$

$$
{}^0_N T = \begin{bmatrix} {}^0_N R & {}^0 P_N \\ 0 & 1 \end{bmatrix}
$$

$$
\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ -d_4 \end{bmatrix}
$$

$$
\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix} = \begin{bmatrix} [-l_2\sin(t_1+t_2)-l_1\sin(t_1)]\dot{t}_1+[-l_2\sin(t_1+t_2)]\dot{t}_2 \\ [l_2\cos(t_1+t_2)-l_1\cos(t_1)]\dot{t}_1+[l_2\cos(t_1+t_2)]\dot{t}_2 \\ -\dot{d}_4 \end{bmatrix}
$$

Jacobian: Direct Differentiation (Jp)

$$\dot{p}_e = J_P(q)\dot{q}$$

$$\omega_e = J_O(q)\dot{q}.$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = J_p\left[\dot{\theta}\right]$$

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix} = \boxed{J_p} \begin{bmatrix} \dot{t}_1 \\ \dot{t}_2 \\ \dot{t}_3 \\ \dot{d}_4 \end{bmatrix}$$

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ -d_4 \end{bmatrix}$$

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix} = \begin{bmatrix} [-l_2\sin(t_1+t_2)-l_1\sin(t_1)]\dot{t}_1+[-l_2\sin(t_1+t_2)]\dot{t}_2 \\ [l_2\cos(t_1+t_2)-l_1\cos(t_1)]\dot{t}_1+[l_2\cos(t_1+t_2)]\dot{t}_2 \\ -\dot{d}_4 \end{bmatrix}$$

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix} = \boxed{\begin{bmatrix} (-l_2\sin(t_1+t_2)-l_1\sin(t_1)) & (-l_2\sin(t_1+t_2)) & 0 & 0 \\ (l_2\cos(t_1+t_2)-l_1\cos(t_1)) & (l_2\cos(t_1+t_2)) & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}} \begin{bmatrix} \dot{t}_1 \\ \dot{t}_2 \\ \dot{t}_3 \\ \dot{d}_4 \end{bmatrix}$$

J0_VP =

$$\begin{pmatrix} -l_2\sin(t_1+t_2)-l_1\sin(t_1) & -l_2\sin(t_1+t_2) & 0 & 0 \\ l_2\cos(t_1+t_2)+l_1\cos(t_1) & l_2\cos(t_1+t_2) & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\dot{p}_e = J_P(q)\dot{q}$$

Jacobian: Direct Differentiation (Jo)

$$\omega_e = J_O(q)\dot{q}.$$

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = Jo\left[\dot{\theta}\right]$$

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \boxed{Jo}\begin{bmatrix} \dot{t}_1 \\ \dot{t}_2 \\ \dot{t}_3 \\ \dot{d}_4 \end{bmatrix}$$

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + 0 \end{bmatrix}$$

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \boxed{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}}\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{d}_4 \end{bmatrix}$$

```
J0_VP =
```

$$\begin{pmatrix} -l_2\sin(t_1+t_2) - l_1\sin(t_1) & -l_2\sin(t_1+t_2) & 0 & 0 \\ l_2\cos(t_1+t_2) + l_1\cos(t_1) & l_2\cos(t_1+t_2) & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Jacobian: Direct Differentiation

$$v_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = J(q)\dot{q}$$

$$\boxed{\begin{aligned} \dot{p}_e &= J_P(q)\dot{q} \\ \omega_e &= J_O(q)\dot{q}. \end{aligned}}$$

$$J = \begin{bmatrix} J_P \\ J_O \end{bmatrix}$$

J0_VP =

$$\begin{pmatrix} -l_2\sin(t_1+t_2) - l_1\sin(t_1) & -l_2\sin(t_1+t_2) & 0 & 0 \\ l_2\cos(t_1+t_2) + l_1\cos(t_1) & l_2\cos(t_1+t_2) & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Jacobian: Direct Differentiation

In summary, the Jacobian in (3.5) can be partitioned into the (3×1) column vectors $\jmath_{Pi}$ and $\jmath_{Oi}$ as

$$\boldsymbol{J} = \begin{bmatrix} \jmath_{P1} & & \jmath_{Pn} \\ & \dots & \\ \jmath_{O1} & & \jmath_{On} \end{bmatrix}, \tag{3.29}$$

where

$$\begin{bmatrix} \jmath_{Pi} \\ \jmath_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} \boldsymbol{z}_{i-1} \\ \boldsymbol{0} \end{bmatrix} & \text{for a } \textit{prismatic} \text{ joint} \\ \begin{bmatrix} \boldsymbol{z}_{i-1} \times (\boldsymbol{p}_e - \boldsymbol{p}_{i-1}) \\ \boldsymbol{z}_{i-1} \end{bmatrix} & \text{for a } \textit{revolute} \text{ joint.} \end{cases} \tag{3.30}$$

The expressions in (3.30) allow Jacobian computation in a simple, systematic way on the basis of direct kinematics relations. In fact, the vectors $\boldsymbol{z}_{i-1}$, $\boldsymbol{p}_e$ and $\boldsymbol{p}_{i-1}$ are all functions of the joint variables. In particular:

Jacobian: Direct Differentiation

- $\boldsymbol{z}_{i-1}$ is given by the third column of the rotation matrix $\boldsymbol{R}_{i-1}^0$, i.e.,

$$\boldsymbol{z}_{i-1} = \boldsymbol{R}_1^0(q_1) \dots \boldsymbol{R}_{i-1}^{i-2}(q_{i-1})\boldsymbol{z}_0 \tag{3.31}$$

  where $\boldsymbol{z}_0 = [\,0 \quad 0 \quad 1\,]^T$ allows the selection of the third column.

- $\boldsymbol{p}_e$ is given by the first three elements of the fourth column of the transformation matrix $\boldsymbol{T}_e^0$, i.e., by expressing $\widetilde{\boldsymbol{p}}_e$ in the (4×1) homogeneous form

$$\widetilde{\boldsymbol{p}}_e = \boldsymbol{A}_1^0(q_1) \dots \boldsymbol{A}_n^{n-1}(q_n)\widetilde{\boldsymbol{p}}_0 \tag{3.32}$$

  where $\widetilde{\boldsymbol{p}}_0 = [\,0 \quad 0 \quad 0 \quad 1\,]^T$ allows the selection of the fourth column.

- $\boldsymbol{p}_{i-1}$ is given by the first three elements of the fourth column of the transformation matrix $\boldsymbol{T}_{i-1}^0$, i.e., it can be extracted from

$$\widetilde{\boldsymbol{p}}_{i-1} = \boldsymbol{A}_1^0(q_1) \dots \boldsymbol{A}_{i-1}^{i-2}(q_{i-1})\widetilde{\boldsymbol{p}}_0. \tag{3.33}$$

Jacobian: Direct Differentiation

$$\begin{bmatrix} J_{Pi} \\ J_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \text{for a } \textit{prismatic} \text{ joint} \\[2ex] \begin{bmatrix} \boxed{z_{i-1}} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{for a } \textit{revolute} \text{ joint.} \end{cases}$$

T0_1 =

$$\begin{pmatrix} \cos(t_1) & -\sin(t_1) & 0 & 0 \\ \sin(t_1) & \cos(t_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T0_2 =

$$\begin{pmatrix} \cos(t_1+t_2) & -\sin(t_1+t_2) & 0 & l_1\cos(t_1) \\ \sin(t_1+t_2) & \cos(t_1+t_2) & 0 & l_1\sin(t_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T0_3 =

$$\begin{pmatrix} \cos(t_1+t_2+t_3) & -\sin(t_1+t_2+t_3) & 0 & l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ \sin(t_1+t_2+t_3) & \cos(t_1+t_2+t_3) & 0 & l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T0_4 =

$$\begin{pmatrix} \cos(t_1+t_2+t_3) & \sin(t_1+t_2+t_3) & 0 & l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ \sin(t_1+t_2+t_3) & -\cos(t_1+t_2+t_3) & 0 & l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ 0 & 0 & -1 & -d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

R0_1 =

$$\begin{pmatrix} \cos(t_1) & -\sin(t_1) & 0 \\ \sin(t_1) & \cos(t_1) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

R0_2 =

$$\begin{pmatrix} \cos(t_1+t_2) & -\sin(t_1+t_2) & 0 \\ \sin(t_1+t_2) & \cos(t_1+t_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

R0_3 =

$$\begin{pmatrix} \cos(t_1+t_2+t_3) & -\sin(t_1+t_2+t_3) & 0 \\ \sin(t_1+t_2+t_3) & \cos(t_1+t_2+t_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

R0_4 =

$$\begin{pmatrix} \cos(t_1+t_2+t_3) & \sin(t_1+t_2+t_3) & 0 \\ \sin(t_1+t_2+t_3) & -\cos(t_1+t_2+t_3) & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

Z0_1 =

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Z0_2 =

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Z0_3 =

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Z0_4 =

$$\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

```
%t2r : rotation matrix from the homogeneous matrix
R0_1 = t2r(T0_1)
R0_2 = t2r(T0_2)
R0_3 = t2r(T0_3)
R0_4 = t2r(T0_4)
```

```
Z0_1 = R0_1(1:3,3)
Z0_2 = R0_2(1:3,3)
Z0_3 = R0_3(1:3,3)
Z0_4 = R0_4(1:3,3)
```

Jacobian: Direct Differ

$$\begin{bmatrix} J_{Pi} \\ J_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ \mathbf{0} \end{bmatrix} & \text{for a } \textit{prismatic} \text{ joint} \\ \begin{bmatrix} z_{i-1} \times (\boxed{p_e} - \boxed{p_{i-1}}) \\ z_{i-1} \end{bmatrix} & \text{for a } \textit{revolute} \text{ joint.} \end{cases}$$

T0_1 =

$$\begin{pmatrix} \cos(t_1) & -\sin(t_1) & 0 & 0 \\ \sin(t_1) & \cos(t_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T0_2 =

$$\begin{pmatrix} \cos(t_1+t_2) & -\sin(t_1+t_2) & 0 & l_1\cos(t_1) \\ \sin(t_1+t_2) & \cos(t_1+t_2) & 0 & l_1\sin(t_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T0_3 =

$$\begin{pmatrix} \cos(t_1+t_2+t_3) & -\sin(t_1+t_2+t_3) & 0 & l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ \sin(t_1+t_2+t_3) & \cos(t_1+t_2+t_3) & 0 & l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T0_4 =

$$\begin{pmatrix} \cos(t_1+t_2+t_3) & \sin(t_1+t_2+t_3) & 0 & l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ \sin(t_1+t_2+t_3) & -\cos(t_1+t_2+t_3) & 0 & l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ 0 & 0 & -1 & -d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

P0_1 =

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

P0_2 =

$$\begin{pmatrix} l_1\cos(t_1) \\ l_1\sin(t_1) \\ 0 \end{pmatrix}$$

P0_3 =

$$\begin{pmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ 0 \end{pmatrix}$$

P0_4 =

$$\begin{pmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ -d_4 \end{pmatrix}$$

```
%.t : translation matrix from the homogeneous matrix
P0_1 = T0_1.t
P0_2 = T0_2.t
P0_3 = T0_3.t
P0_4 = T0_4.t
```

Jacobian: Direct Differe

$$\begin{bmatrix} J_{Pi} \\ J_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \text{for a } \textit{prismatic} \text{ joint} \\ \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{for a } \textit{revolute} \text{ joint.} \end{cases}$$

$$J_{P1} = z_{0\_1} \times (p_e - p_{0\_1})$$

$$= \underset{\texttt{z0\_1 =}}{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}} \times \left(\underset{\texttt{P0\_4 =}}{\begin{pmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ -d_4 \end{pmatrix}} - \underset{\texttt{P0\_1 =}}{\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}} \right) = \underset{\texttt{J1 =}}{\begin{pmatrix} -l_2\sin(t_1+t_2)-l_1\sin(t_1) \\ l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ 0 \end{pmatrix}}$$

$$J_{P2} = z_{0\_2} \times (p_e - p_{0\_2})$$

$$= \underset{\texttt{z0\_2 =}}{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}} \times \left(\underset{\texttt{P0\_4 =}}{\begin{pmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ -d_4 \end{pmatrix}} - \underset{\texttt{P0\_2 =}}{\begin{pmatrix} l_1\cos(t_1) \\ l_1\sin(t_1) \\ 0 \end{pmatrix}} \right) = \underset{\texttt{J2 =}}{\begin{pmatrix} -l_2\sin(t_1+t_2) \\ l_2\cos(t_1+t_2) \\ 0 \end{pmatrix}}$$

$$J_{P3} = z_{0\_3} \times (p_e - p_{0\_3})$$

$$= \underset{\texttt{z0\_3 =}}{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}} \times \left(\underset{\texttt{P0\_4 =}}{\begin{pmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ -d_4 \end{pmatrix}} - \underset{\texttt{P0\_3 =}}{\begin{pmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ 0 \end{pmatrix}} \right) = \underset{\texttt{J3 =}}{\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}}$$

$$J_{P4} = z_{0\_4}$$

$$= \underset{\texttt{z0\_4 =}}{\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}} = \underset{\texttt{J4 =}}{\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}}$$

Jacobian: Direct Differe

$$\begin{bmatrix} \boxed{J_{Pi}} \\ J_{Oi} \end{bmatrix} = \begin{cases} \boxed{\begin{bmatrix} \boxed{z_{i-1}} \\ 0 \end{bmatrix} \quad \text{for a } \textit{prismatic} \text{ joint}} \\ \begin{bmatrix} \boxed{z_{i-1}} \times (p_e - \boxed{p_{i-1}}) \\ z_{i-1} \end{bmatrix} \quad \text{for a } \textit{revolute} \text{ joint.} \end{cases}$$

$J_{P1} = z_{0\_1} \times (p_e - p_{0\_1})$

$$= \text{Z0\_1} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \left(\boxed{\text{P0\_4} = \begin{pmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ -d_4 \end{pmatrix}} - \text{P0\_1} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right) = \text{J1} = \begin{pmatrix} -l_2\sin(t_1+t_2)-l_1\sin(t_1) \\ l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ 0 \end{pmatrix}$$

$J_{P2} = z_{0\_2} \times (p_e - p_{0\_2})$

$$= \text{Z0\_2} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \left(\boxed{\text{P0\_4} = \begin{pmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ -d_4 \end{pmatrix}} - \text{P0\_2} = \begin{pmatrix} l_1\cos(t_1) \\ l_1\sin(t_1) \\ 0 \end{pmatrix} \right) = \text{J2} = \begin{pmatrix} -l_2\sin(t_1+t_2) \\ l_2\cos(t_1+t_2) \\ 0 \end{pmatrix}$$

$J_{P3} = z_{0\_3} \times (p_e - p_{0\_3})$

$$= \text{Z0\_3} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \left(\boxed{\text{P0\_4} = \begin{pmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ -d_4 \end{pmatrix}} - \text{P0\_3} = \begin{pmatrix} l_2\cos(t_1+t_2)+l_1\cos(t_1) \\ l_2\sin(t_1+t_2)+l_1\sin(t_1) \\ 0 \end{pmatrix} \right) = \text{J3} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\boxed{\begin{aligned} &J_{P4} = z_{0\_4} \\ &= \text{Z0\_4} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} = \text{J4} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \end{aligned}}$$

Jacobian: Direct Differentiation

```
%.t : translation matrix from the homogeneous matrix
P0_1 = T0_1.t
P0_2 = T0_2.t
P0_3 = T0_3.t
P0_4 = T0_4.t
```

```
%t2r : rotation matrix from the homogeneous matrix
R0_1 = t2r(T0_1)
R0_2 = t2r(T0_2)
R0_3 = t2r(T0_3)
R0_4 = t2r(T0_4)
```

```
Z0_1 = R0_1(1:3,3)
Z0_2 = R0_2(1:3,3)
Z0_3 = R0_3(1:3,3)
Z0_4 = R0_4(1:3,3)
```

```
J1 = cross(Z0_1,(P0_4-P0_1)) % Column:1, Row:1-3 % for a revolute joint
J2 = cross(Z0_2,(P0_4-P0_2)) % Column:2, Row:1-3 % for a revolute joint
J3 = cross(Z0_3,(P0_4-P0_3)) % Column:3, Row:1-3 % for a revolute joint
J4 = Z0_4 % Column:4, Row:1-3 %for a prismatic joint
```

```
J0_DD = [J1 J2 J3 J4; Z0_1 Z0_2 Z0_3 [0; 0; 0]];
J0_DD = simplify(J0_DD)
```

```
fprintf('T0_2 = ')
T0_2 = simplify(SCARA.A([1,2],q))
fprintf('T0_3 = ')
T0_3 = simplify(SCARA.A([1,2,3],q))
fprintf('T0_4 = ')
T0_4 = simplify(SCARA.A([1,2,3,4],q))
```

```
R1_0 = transpose(R0_1);
R2_1 = transpose(R1_2);
R3_2 = transpose(R2_3);
R4_3 = transpose(R3_4);
```

UCLA

Jacobian: Direct Differentiation

$$J = \begin{bmatrix} J_{P_1} & J_{P_2} & J_{P_3} & J_{P_4} \\ J_{O_1} & J_{O_2} & J_{O_3} & J_{O_4} \end{bmatrix}$$

J1 =

$$\begin{pmatrix} -l_2 \sin(t_1+t_2) - l_1 \sin(t_1) \\ l_2 \cos(t_1+t_2) + l_1 \cos(t_1) \\ 0 \end{pmatrix}$$

J2 =

$$\begin{pmatrix} -l_2 \sin(t_1+t_2) \\ l_2 \cos(t_1+t_2) \\ 0 \end{pmatrix}$$

J3 =

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

J4 =

$$\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

J0_DD =

$$\begin{pmatrix} -l_2 \sin(t_1+t_2) - l_1 \sin(t_1) & -l_2 \sin(t_1+t_2) & 0 & 0 \\ l_2 \cos(t_1+t_2) + l_1 \cos(t_1) & l_2 \cos(t_1+t_2) & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Z0_1 =

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Z0_2 =

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Z0_3 =

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Z0_4 =

$$\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

$$\begin{bmatrix} J_{Pi} \\ J_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ \mathbf{0} \end{bmatrix} & \text{for a } \textit{prismatic} \text{ joint} \\ \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{for a } \textit{revolute} \text{ joint.} \end{cases}$$

Jacobian - Comparison

J4_VP =

$$\begin{pmatrix} l_1 \sin(t_2+t_3) + l_2 \sin(t_3) & l_2 \sin(t_3) & 0 & 0 \\ -l_1 \cos(t_2+t_3) - l_2 \cos(t_3) & -l_2 \cos(t_3) & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 \end{pmatrix}$$

J0_VP =

$$\begin{pmatrix} -l_2 \sin(t_1+t_2) - l_1 \sin(t_1) & -l_2 \sin(t_1+t_2) & 0 & 0 \\ l_2 \cos(t_1+t_2) + l_1 \cos(t_1) & l_2 \cos(t_1+t_2) & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

J0_DD =

$$\begin{pmatrix} -l_2 \sin(t_1+t_2) - l_1 \sin(t_1) & -l_2 \sin(t_1+t_2) & 0 & 0 \\ l_2 \cos(t_1+t_2) + l_1 \cos(t_1) & l_2 \cos(t_1+t_2) & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Jacobian Methods of Derivation & the Corresponding Reference Frame – Summary

Method	Jacobian Matrix Reference Frame	Transformation to Base Frame (Frame 0)
Explicit (Diff. the Forward Kinematic Eq.)	${}^{0}J_{N}$	None
Iterative Velocity Eq.	${}^{N}J_{N}$	Transform Method 1: $${}^{0}v_{N} = {}^{0}_{N}R\,{}^{N}v_{N}$$ $${}^{0}\omega_{N} = {}^{0}_{N}R\,{}^{N}\omega_{N}$$ Transform Method 2: $${}^{0}J_{N}(\theta) = \begin{bmatrix} {}^{0}_{N}R & 0 \\ 0 & {}^{0}_{N}R \end{bmatrix} {}^{N}J_{N}(\theta)$$
Iterative Force Eq.	${}^{N}J_{N}^{T}$	Transpose $${}^{N}J_{N} = [{}^{N}J_{N}^{T}]^{T}$$ Transform $${}^{0}J_{N}(\theta) = \begin{bmatrix} {}^{0}_{N}R & 0 \\ 0 & {}^{0}_{N}R \end{bmatrix} {}^{N}J_{N}(\theta)$$

UCLA

Summary

- ✓ Jacobian with SCARA example
  - Velocity propagation
  - Direct differentiation
- ✓ Frame of Representation