دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی مکانیک
دانشگاه صنعتی امیرکبیر

Lecture 4_2:
Jacobians:
Velocities & Static Forces

Advanced Robotics

Hamed Ghafarirad

Outlines

Jacobians

- The Jacobian is a multidimensional form of the derivative.
- Assume six functions, each of which is a function of six independent variables.

$$\begin{cases} y_1 = f_1(x_1, x_2, x_3, x_4, x_5, x_6) \\ y_2 = f_2(x_1, x_2, x_3, x_4, x_5, x_6) \\ \vdots \\ y_6 = f_6(x_1, x_2, x_3, x_4, x_5, x_6) \end{cases}$$

- Using vector notation:

$$\boldsymbol{Y} = \boldsymbol{F}(\boldsymbol{X})$$

Jacobians

- Calculate the differentials of y_i as a function of differentials of x_j, using the chain rule.

$$\begin{cases} \delta y_1 = \dfrac{\partial f_1}{\partial x_1}\delta x_1 + \dfrac{\partial f_1}{\partial x_2}\delta x_2 + \cdots + \dfrac{\partial f_1}{\partial x_6}\delta x_6 \\ \delta y_2 = \dfrac{\partial f_2}{\partial x_1}\delta x_1 + \dfrac{\partial f_2}{\partial x_2}\delta x_2 + \cdots + \dfrac{\partial f_2}{\partial x_6}\delta x_6 \\ \qquad\qquad\qquad \vdots \\ \delta y_6 = \dfrac{\partial f_6}{\partial x_1}\delta x_1 + \dfrac{\partial f_6}{\partial x_2}\delta x_2 + \cdots + \dfrac{\partial f_6}{\partial x_6}\delta x_6 \end{cases}$$

- In vector notation:

$$\delta \boldsymbol{Y} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{X}}\delta \mathbf{X}$$

- The 6×6 matrix of partial derivatives is the **Jacobian (J)**.

- **Note:** Functions $f_1(\mathbf{X})$ through $f_6(\mathbf{X})$ are generally nonlinear, then the partial derivatives are a function of the x_i, so

$$\delta \boldsymbol{Y} = J(\boldsymbol{X})\,\delta \mathbf{X}$$

Jacobians

$$\delta \boldsymbol{Y} = J(\boldsymbol{X})\,\delta \mathbf{X}$$

- Dividing by the differential time element, Jacobian is a velocities mapping in X to those in Y:

$$\dot{\boldsymbol{Y}} = J(\boldsymbol{X})\,\dot{\boldsymbol{X}}$$

- Jacobians are time-varying linear transformations. (*Why?*)

- In robotics, Jacobians relate joint velocities to Cartesian velocities of the tip of the arm.

$${}^{0}\nu = \nu = {}^{0}J(\Theta)\dot{\Theta}$$

  - Θ is the vector of joint angles of the manipulator .
  - ν is a vector of Cartesian velocities.

- The leading superscript indicates in which frame the resulting Cartesian velocity is <u>expressed</u>.

Jacobians

- For the general case of a six-jointed robot, (the Jacobian is 6×6), ($\dot{\Theta}$ is 6×1), and (${}^{0}\nu$ is 6×1) .

- This 6×1 Cartesian velocity vector (${}^{0}\nu$) includes the 3×1 linear velocity vector (${}^{0}v$) and the 3×1 rotational velocity vector (${}^{0}\omega$).

$$ {}^{0}\nu = \begin{bmatrix} {}^{0}v \\ {}^{0}\omega \end{bmatrix} $$

- Jacobians may be nonsquare:
  - The number of rows = the number of degrees of freedom in the Cartesian space being considered.
  - The number of columns = the number of joints of the manipulator.

- For planar arm, no reason to have more than three rows.

- For redundant planar manipulators, there could be arbitrarily many columns.

Jacobians

$$^0\nu = \begin{bmatrix} ^0v \\ ^0\omega \end{bmatrix} = {}^0J(\Theta)\dot{\Theta}$$

- The vector ν is sometimes called a body velocity.

- Seeking expressions of the form:

$$v = {}^0J_v\dot{\Theta}$$

$$\omega = {}^0J_\omega\dot{\Theta}$$

> where J_v and J_ω are $3 \times n$ matrices.

- Therefore:

$$\nu = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} ^0J_v(\Theta) \\ ^0J_\omega(\Theta) \end{bmatrix}\dot{\Theta}$$

$$^0J(\Theta) = \begin{bmatrix} ^0J_v(\Theta) \\ ^0J_\omega(\Theta) \end{bmatrix}$$

Jacobians

- Jacobian can be achieved by **3 methods**:

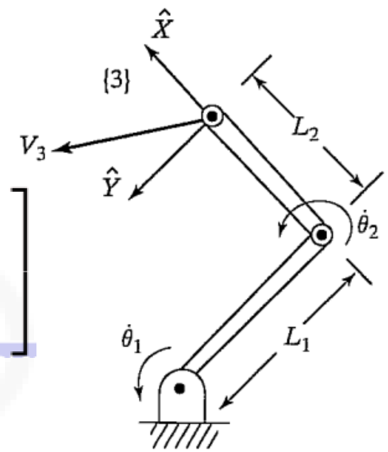
  1) Using velocity propagation (e.g. ${}^{e}v_e$ & ${}^{e}\omega_e$ or ${}^{0}v_e$ & ${}^{0}\omega_e$)

  2) Direct differentiation (e.g. $\frac{d}{dt}\,{}^{0}P_e$ & $S = \dot{R}\,R^T$)

  3) An alternative approach (…)

Jacobians

- **1st Method: Velocity Propagation**
- **Example:** Two-Link RR Manipulator
- $J_v(\Theta)$ is a 3×2 Jacobian that relates joint rates to end-effector linear velocity.



- From velocity propagation:

$$ {}^3v_3 = \begin{bmatrix} l_1 s_2 \dot{\theta}_1 \\ l_1 c_2 \dot{\theta}_1 + l_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} \qquad {}^0v_3 = \begin{bmatrix} -l_1 s_1 \dot{\theta}_1 - l_2 s_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ l_1 c_1 \dot{\theta}_1 + l_2 c_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} $$

- The Jacobian written in frame {3} :

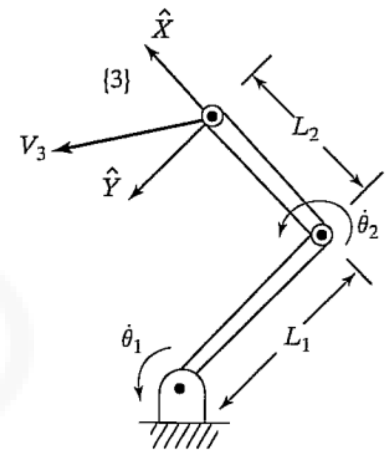
$$ {}^3J_v(\Theta) = \begin{bmatrix} l_1 s2 & 0 \\ l_1 c2 + l_2 & l_2 \\ 0 & 0 \end{bmatrix} $$

- The Jacobian written in frame {0} :

$$ {}^0J_v(\Theta) = \begin{bmatrix} -l_1 s1 - l_2 s12 & -l_2 s12 \\ l_1 c1 + l_2 c12 & l_2 c12 \\ 0 & 0 \end{bmatrix} $$

Jacobians

- **1st Method: Velocity Propagation**
- **Example:** Two-Link RR Manipulator
- $J_\omega(\Theta)$ is a 3×2 Jacobian that relates joint rates to end-effector angular velocity.



- From velocity propagation:

$$^{3}\omega_3 = {}^{0}\omega_3 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}$$

- Therefore,

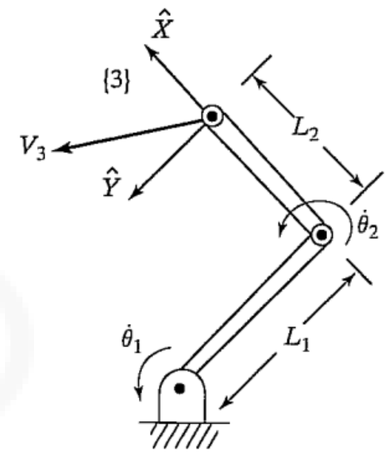
$$^{0}J_\omega(\Theta) = {}^{3}J_\omega(\Theta) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

Jacobians

- **1st Method: Velocity Propagation**
- **Example:** Two-Link RR Manipulator

$$^{0}J(\Theta) = \begin{bmatrix} ^{0}J_v(\Theta) \\ ^{0}J_\omega(\Theta) \end{bmatrix}$$

$$^{0}J(\Theta) = \begin{bmatrix} -l_1 s1 - l_2 s12 & -l_2 s12 \\ l_1 c1 + l_2 c12 & l_2 c12 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

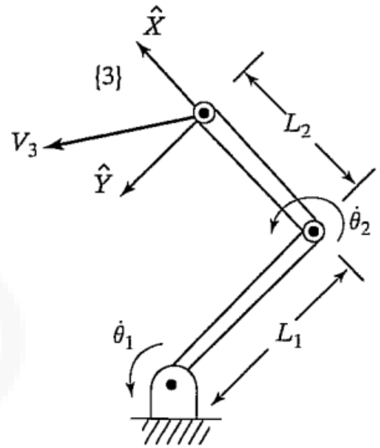


- **Note:**
- For planar manipulators, it is also possible to consider a 3×2 Jacobian including the linear and angular velocity of the end-effector, simultaneously. (*How ?*)

$$^{0}J(\Theta) = \begin{bmatrix} -l_1 s1 - l_2 s12 & -l_2 s12 \\ l_1 c1 + l_2 c12 & l_2 c12 \\ 1 & 1 \end{bmatrix}$$

Jacobians

- **2nd Method: Direct Differentiation**
- **Position & Linear Velocity**
- The Jacobian might also be found by directly differentiating the kinematic equations.



$$^0P_3 = \begin{bmatrix} l_1\, c1 + l_2\, c12 \\ l_1\, s1 + l_2\, s12 \\ 0 \end{bmatrix}$$

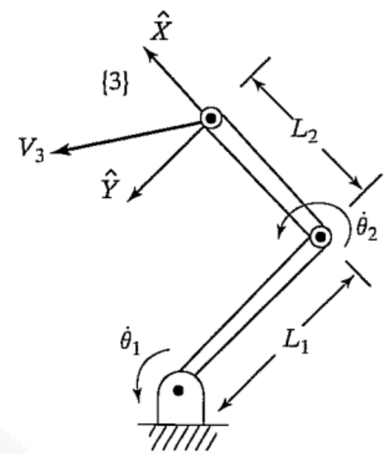
$$\frac{d}{dt}(^0P_3) = \begin{bmatrix} -l_1\dot{\theta}_1 s1 - l_2\dot{\theta}_1 s12 - l_2\dot{\theta}_2 s12 \\ l_1\dot{\theta}_1 c1 + l_2\dot{\theta}_1 c12 + l_2\dot{\theta}_2 c12 \\ 0 \end{bmatrix}$$

$$^0v_3 = \begin{bmatrix} -l_1 s1 - l_2 s12 & -l_2 s12 \\ l_1 c1 + l_2 c12 & l_2 c12 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = {}^0J_v(\Theta)\dot{\Theta}$$

- This is straightforward for linear velocity, but there is **no** 3×1 orientation vector whose derivative is ω.

Jacobians

- **2nd Method: Direct Differentiation**
  - **Orientation & Angular Velocity**



$$^0S_3 = {}^0\dot{R}_3\, {}^0{R_3}^T$$

$$^0R_3 = \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$^0\dot{R}_3 = (\dot{\theta}_1 + \dot{\theta}_2)\begin{bmatrix} -s_{12} & -c_{12} & 0 \\ c_{12} & -s_{12} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$^0S_3 = {}^0\dot{R}_3\, {}^0{R_3}^T = (\dot{\theta}_1 + \dot{\theta}_2)\begin{bmatrix} -s_{12} & -c_{12} & 0 \\ c_{12} & -s_{12} & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} = (\dot{\theta}_1 + \dot{\theta}_2)\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$S = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix} \quad \rightarrow \quad {}^0\Omega_3 = {}^0\omega_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}(\dot{\theta}_1 + \dot{\theta}_2)$$

$$^0\omega_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = {}^0J_\omega(\Theta)\dot{\Theta}$$

Jacobian: An Alternative Approach

3rd Method: An Alternative Approach

$$\nu = \begin{bmatrix} v \\ \omega \end{bmatrix} = {}^{0}J(\Theta)\dot{\Theta}$$

- The vector ν is sometimes called a body velocity.

- Seeking expressions of the form:

$$v = {}^{0}J_v\dot{\Theta}$$

$$\omega = {}^{0}J_\omega\dot{\Theta}$$

➢ where J_v and J_ω are $3 \times n$ matrices.

- Therefore:

$$\nu = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} {}^{0}J_v \\ {}^{0}J_\omega \end{bmatrix} \dot{\Theta}$$

- Now, find J_v and J_ω ?

Jacobian: An Alternative Approach

Angular Velocity

- Angular velocities can be added as free vectors, if expressed relative to a common frame.

- Angular velocity of the end-effector relative to the base ($^{0}\omega_e = {}^{0}\omega_n$):
  - Summing the expressed angular velocities of all joints in the orientation of the base frame .

- **Remember:** The angular velocity of link $i+1$ is the same as that of link i plus a new component caused by rotational velocity at joint $i+1$.

$$^{0}\omega_{i+1} = {}^{0}\omega_i + {}^{0}R_{i+1}\,\dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \quad , \quad \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix}$$

- $\dot{\theta}_{i+1}$ is scalar parameter, so

$$^{0}\omega_{i+1} = {}^{0}\omega_i + \dot{\theta}_{i+1}\,{}^{0}R_{i+1}\,{}^{i+1}\hat{Z}_{i+1} = {}^{0}\omega_i + \dot{\theta}_{i+1}\,{}^{0}\hat{Z}_{i+1}$$

← Outline

Jacobian: An Alternative Approach

- **Angular Velocity**
- **Note:** If joint i is prismatic, the angular velocity of the end-effector **does not depend** on d_i .

- The overall angular velocity of the end-effector, ${}^{0}\omega_e = {}^{0}\omega_n$, in the base frame is as follow:

$$
{}^{0}\omega_n = \rho_1 \dot{\theta}_1 {}^{0}\hat{Z}_1 + \rho_2 \dot{\theta}_2 {}^{0}\hat{Z}_2 + \ \dots + \rho_n \dot{\theta}_n {}^{0}\hat{Z}_n = \sum_{i=1}^{n} \rho_i \dot{\theta}_i {}^{0}\hat{Z}_i
$$

- ρ_i is equal to 1 if joint i is revolute and 0 if joint i is prismatic.

- Since

$$
{}^{0}Z_i = z_i
$$

- The lower half of the Jacobian, i.e., ${}^{0}J_\omega$:

$$
{}^{0}J_\omega = [\rho_1 z_1 \quad \cdots \quad \rho_n z_n]
$$

Jacobian: An Alternative Approach

- **Linear Velocity**
- The linear velocity of the end-effector is just ${}^0\dot{P}_e$.
- By the chain rule for differentiation:

$$v_e = {}^0\dot{P}_e = \sum_{i=1}^{n} \frac{\partial\, {}^0P_e}{\partial \theta_i}\, \dot{\theta}_i$$

- the i-th column of ${}^0J_v$, which we denote as ${}^0J_{vi}$ is given by:

$$J_{vi} = \frac{\partial\, {}^0P_e}{\partial \theta_i}$$

- Now consider the two cases (prismatic and revolute joints) separately.

Jacobian: An Alternative Approach

- **Linear Velocity**
- **Prismatic Joints:**
- If all joints are fixed except the **prismatic joint *i***, then it imparts a pure translation to the end-effector.

- Direction of translation is parallel to the axis of joint i .

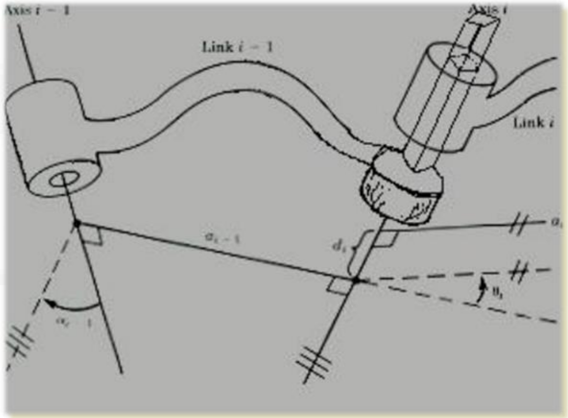
- If only joint i is allowed to move ($\dot{d}_i$ is scalar):

$$v_e = {}^0R_i\,\dot{d}_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \dot{d}_i\,{}^0R_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \dot{d}_i\,{}^0Z_i$$

- d_i is the joint variable for prismatic joint i .

- Thus for the case of prismatic joints:

$${}^0J_{vi} = {}^0Z_i = z_i$$



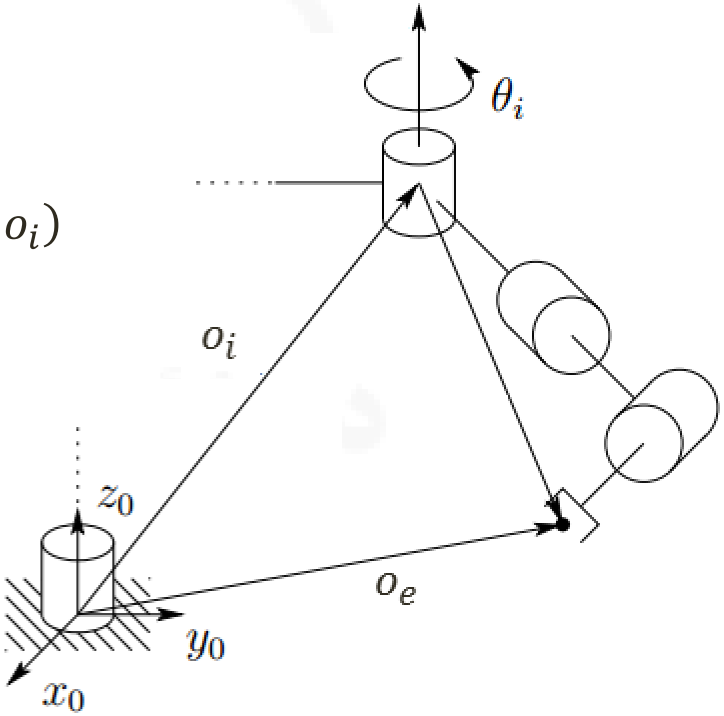
Jacobian: An Alternative Approach

- **Linear Velocity**
- **Revolute Joints:**
- If all joints are fixed except the **revolute joint *i***, the end-effector linear velocity caused by this joint :
  - Cross product of angular velocity of joint i ($\dot{\theta}_i$) and the position vector connecting the origin of frame $\{i\}$ to $\{e\}$

$$v_e = \dot{\theta}_i {}^0Z_i \times (o_e - o_i)$$

- Hence, for the revolute joint i

$${}^0J_{vi} = z_i \times (o_e - o_i)$$



Jacobian: An Alternative Approach

- **Combining the Angular and Linear Jacobians**

- The upper half of the Jacobian ${}^0J_v$ is given as
  ${}^0J_v = [{}^0J_{v1} \quad \dots \quad {}^0J_{vn}]$

$$ {}^0J_{vi} = \begin{cases} z_i \times (o_e - o_i) & for\ revolute\ joint\ i \\ z_i & for\ prismatic\ joint\ i \end{cases} $$

- The lower half of the Jacobian ${}^0J_\omega$ is given as
  ${}^0J_\omega = [{}^0J_{\omega 1} \quad \dots \quad {}^0J_{\omega n}]$

$$ {}^0J_{\omega i} = \begin{cases} z_i & for\ revolute\ joint\ i \\ 0 & for\ prismatic\ joint\ i \end{cases} $$

- Putting the upper and lower halves of the Jacobian together:

$$ {}^0J = [{}^0J_1 \quad \dots \quad {}^0J_n] $$

- If joint i is revolute
  ${}^0J_i = \begin{bmatrix} z_i \times (o_e - o_i) \\ z_i \end{bmatrix}$

- If joint i is prismatic
  ${}^0J_i = \begin{bmatrix} z_i \\ 0 \end{bmatrix}$

← Outline

Jacobian: An Alternative Approach

Combining the Angular and Linear Jacobians

- All of the quantities needed are available once the forward kinematics are worked out.

- The only quantities needed are the unit vectors z_i and the coordinates of the origins $o_1, \ldots, o_n$.

  - z_i : the first three elements in the third column of ${}^0T_i$.
  - o_i : the first three elements of the fourth column of ${}^0T_i$.

- This method is not only for computing the velocity of the end-effector but also *any point* on the manipulator (necessary for dynamic).

Jacobian: An Alternative Approach

❖ **Example 1:** Two-Link RR Manipulator

- The Jacobian matrix, which in this case is 6 × 2, is of the form

$$^0J(\Theta) = \begin{bmatrix} z_1 \times (o_3 - o_1) & z_2 \times (o_3 - o_2) \\ z_1 & z_2 \end{bmatrix}$$

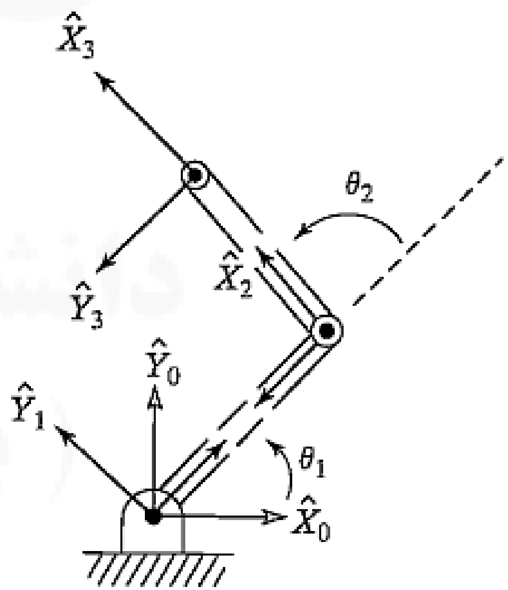
- where

$$o_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad o_2 = \begin{bmatrix} l_1 c1 \\ l_1 s1 \\ 0 \end{bmatrix} \quad o_3 = \begin{bmatrix} l_1 c1 + l_2 c12 \\ l_1 s1 + l_2 s12 \\ 0 \end{bmatrix}$$

$$z_1 = z_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

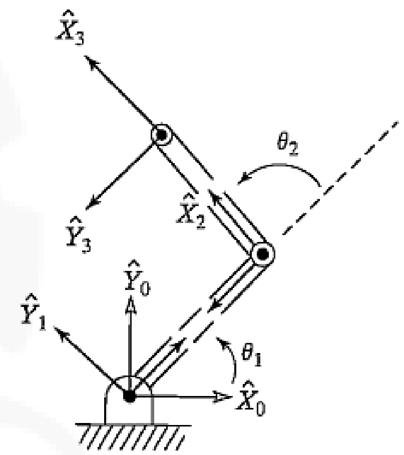
- Therefore

- $$^0J(\Theta) = \begin{bmatrix} -l_1 s1 - l_2 s12 & -l_2 s12 \\ l_1 c1 + l_2 c12 & l_2 c12 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$



Jacobian: An Alternative Approach

❖ **Example 1:** Two-Link RR Manipulator

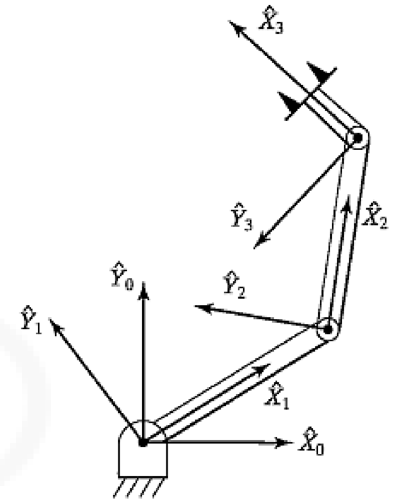


➢ $^0J(\Theta) = \begin{bmatrix} -l_1 s1 - l_2 s12 & -l_2 s12 \\ l_1 c1 + l_2 c12 & l_2 c12 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$

- The first two rows give the linear velocity of the origin o_3 relative to the base.
- The third row is the linear velocity in the direction of z_0 (always zero).
- The last three rows represent the angular velocity of the final frame (a rotation about the vertical axis at the rate $\dot{\theta}_1 + \dot{\theta}_2$).

Jacobian: An Alternative Approach

- **Example 2:** Three-link RRR planar manipulator
- Compute the linear & angular velocity of the **center of link 2** .

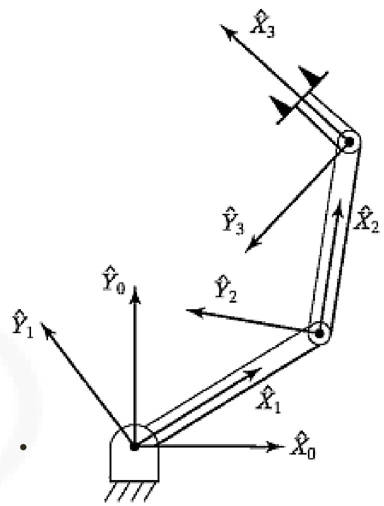


i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3

Jacobian: An Alternative Approach

- **Example 2:** Three-link RRR planar manipulator
- Compute the linear & angular velocity of the **center of link 2** .

- $^0J(\Theta) = \begin{bmatrix} z_1 \times (o_c - o_1) & z_2 \times (o_c - o_2) & 0 \\ z_1 & z_2 & 0 \end{bmatrix}$



- It is merely the usual the Jacobian with o_c in place of o_e .

- The third column of the Jacobin is zero, since the velocity of the second link is unaffected by motion of the third link.

- The vector o_c must be computed as it is not given directly by the T matrices.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3

Analytical Jacobian

- **Remember:**
  - Assume the orientation of the end-effector frame relative to the base frame is described by the set of Z-Y-Z Euler angles (α, β, γ).

  - **Objective:** Express the angular velocity (Ω) of the end-effector as rates of the set of Z-Y-Z Euler angles $(\dot{\alpha}, \dot{\beta}, \dot{\gamma})$.

$$\Omega = f(\Theta, \dot{\Theta}) \qquad \Theta_{Z'Y'Z'} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \qquad \dot{\Theta}_{Z'Y'Z'} = \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}$$

$${}^{A}R_{B\,Z'Y'Z'}(\alpha,\beta,\gamma) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}$$

$$\dot{R}R^T = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix} \qquad \begin{aligned} \Omega_x &= \dot{r}_{31}r_{21} + \dot{r}_{32}r_{22} + \dot{r}_{33}r_{23} \\ \Omega_y &= \dot{r}_{11}r_{31} + \dot{r}_{12}r_{32} + \dot{r}_{13}r_{33} \\ \Omega_z &= \dot{r}_{21}r_{11} + \dot{r}_{22}r_{12} + \dot{r}_{23}r_{13} \end{aligned}$$

$$\Omega = E_{Z'Y'Z'}(\Theta_{Z'Y'Z'})\dot{\Theta}_{Z'Y'Z'} \qquad E_{Z'Y'Z'} = \begin{bmatrix} 0 & -s\alpha & c\alpha\, s\beta \\ 0 & c\alpha & s\alpha\, s\beta \\ 1 & 0 & c\beta \end{bmatrix}$$

Analytical Jacobian

- The conventional Jacobian matrix is sometimes called the *Geometric Jacobian.*

- The Jacobian might also be found by directly differentiating the kinematic equations.

- This is straightforward for linear velocity, but there is **no** 3×1 orientation vector whose derivative is ω .

- Denote the end-effector pose:

$$X = \begin{bmatrix} d(\Theta) \\ \Theta(\Theta) \end{bmatrix}$$

  - $d(\Theta)$ = Position of the end-effector frame relative to the base frame.

  - $\Theta(\Theta)$ = **Minimal representation** for the orientation of the end-effector frame relative to the base frame, e.g, the set of Z-Y-Z Euler angles (α, β, γ).

$$\Theta_{Z'Y'Z'} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

Analytical Jacobian

- Look for an expression of the form:

$$\dot{X} = \begin{bmatrix} \dot{d}(\Theta) \\ \dot{\Theta}(\Theta) \end{bmatrix} = J_a(\Theta)\dot{\Theta}$$

- Analytical Jacobian, $J_a(\Theta)$, is based on a minimal representation for the orientation of the end-effector frame.

- The analytical Jacobian can be found by the kinematic problem.

Analytical Jacobian

Analytical & Geometric Jacobians

- Combining the above relationship with the previous definition of the Jacobian

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{d} \\ \omega \end{bmatrix} = J(\Theta)\dot{\Theta}$$

$$J(\Theta)\dot{\Theta} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{d} \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{d} \\ E(\Theta)\,\dot{\Theta} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & E(\Theta) \end{bmatrix} \begin{bmatrix} \dot{d} \\ \dot{\Theta} \end{bmatrix}$$
$$= \begin{bmatrix} I & 0 \\ 0 & E(\Theta) \end{bmatrix} J_a(\Theta)\dot{\Theta}$$

- Yields

$$J(\Theta) = \begin{bmatrix} I & 0 \\ 0 & E(\Theta) \end{bmatrix} J_a(\Theta)$$

- Then

$$J_a(\Theta) = \begin{bmatrix} I & 0 \\ 0 & E^{-1}(\Theta) \end{bmatrix} J(\Theta) \qquad E_{Z'Y'Z'} = \begin{bmatrix} 0 & -s\alpha & c\alpha\, s\beta \\ 0 & c\alpha & s\alpha\, s\beta \\ 1 & 0 & c\beta \end{bmatrix}$$

- The analytical Jacobian, $J_a(\theta)$, may be computed from the geometric Jacobian provided that $det\, E(\Theta) \neq 0$.

Static Forces

- Consider how forces and moments "propagate" from one link to the next.

- The robot is pushing on something in the environment with the end-effector or is supporting a load at the hand (*Static Structure*).

- Solve for the joint torques that must be acting to keep the system in static equilibrium.

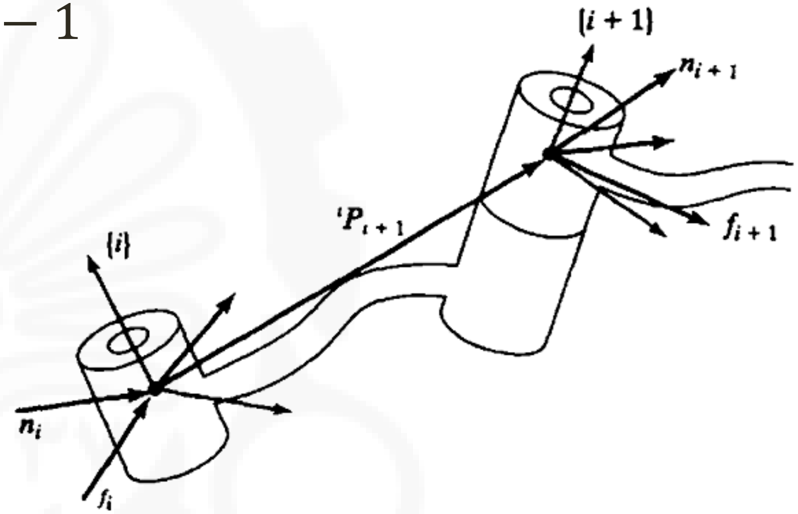
- In considering static forces:
  - ➢ First lock all the joints so that the manipulator becomes a structure.
  - ➢ Write a force-moment balance relationship in terms of the link frames.
  - ➢ Compute what static torque must be acting about the joint axis in order for the manipulator to be in static equilibrium.

- Ignore the force on the links due to gravity (Postpone it!).

- The static forces and torques we are considering at the joints are those caused by a static force or torque (or both) acting on the last link.

Static Forces

- f_i = force exerted on link i by link $i-1$
- n_i = torque exerted on link i by link $i-1$



- Forces balance:
- ${}^if_i - {}^if_{i+1} = 0$

- Torques balance wrt $\{i\}$:
- ${}^in_i - {}^in_{i+1} - {}^iP_{i+1} \times {}^if_{i+1} = 0$

- So

$$ {}^if_i = {}^if_{i+1} $$
$$ {}^in_i = {}^in_{i+1} + {}^iP_{i+1} \times {}^if_{i+1} $$

- Use the rotation matrix to change the describing frame:

$$ {}^if_i = {}^iR_{i+1}\,{}^{i+1}f_{i+1} $$
$$ {}^in_i = {}^iR_{i+1}\,{}^{i+1}n_{i+1} + {}^iP_{i+1} \times {}^if_i $$

Static Forces

$${}^{i}f_i = {}^{i}R_{i+1}\,{}^{i+1}f_{i+1}$$

$${}^{i}n_i = {}^{i}R_{i+1}\,{}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i$$

- Static Force "**Propagation**":
  - calculate the force and moment applied by each link, working from the last link down to the base (link 0).

- What joint torques are needed in order to balance the reaction forces and moments acting on the links?

- All components of the force and moment vectors are resisted by the structure of the mechanism itself, **except** for those about the joint axis.

- **The required joint torque** = the *dot product* of the joint-axis vector with the moment vector acting on the link

$$\tau_i = {}^{i}n_i^T \hat{Z}_i$$

- If joint i is prismatic

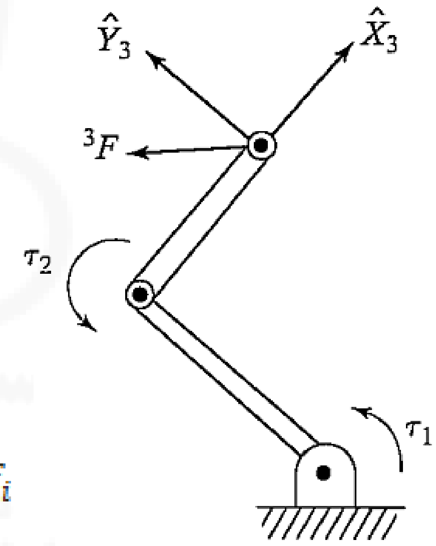
$$\tau_i = {}^{i}f_i^T \hat{Z}_i$$

- **Note:** It is using the symbol τ even for a linear joint force.

Static Forces

- **Example:** The two-link RR manipulator
- Manipulator is applying a force vector ${}^{3}F$ with its end-effector (acting at the origin of {3})

- Find the required joint torques as a function of configuration and of the applied force.



$$ {}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & l_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{i}f_i = {}^{i}R_{i+1}\,{}^{i+1}f_{i+1} $$

$$ {}^{i}n_i = {}^{i}R_{i+1}\,{}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i $$

- Start from the last link and going toward the base of the robot:

$$ {}^{2}f_2 = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix}, \qquad {}^{2}n_2 = l_2\hat{X}_2 \times \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix} $$

Static Forces

❖ **Example:** The two-link RR manipulator

$$
{}^{1}f_1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} c_2 f_x - s_2 f_y \\ s_2 f_x + c_2 f_y \\ 0 \end{bmatrix},
$$

$$
{}^{1}n_1 = \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix} + l_1 \hat{X}_1 \times {}^{1}f_1 = \begin{bmatrix} 0 \\ 0 \\ l_1 s_2 f_x + l_1 c_2 f_y + l_2 f_y \end{bmatrix}
$$

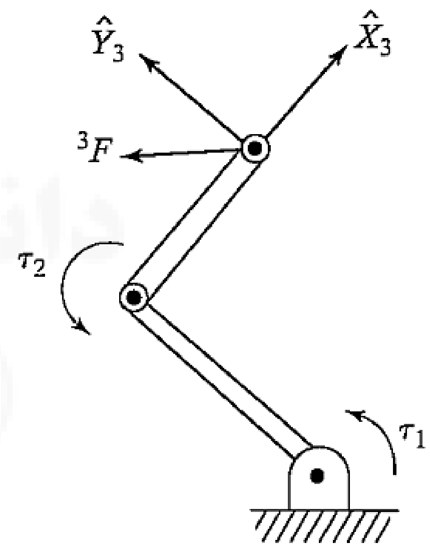
- Therefore

$$
\tau_1 = l_1 s_2 f_x + (l_2 + l_1 c_2) f_y
$$

$$
\tau_2 = l_2 f_y .
$$

- As a matrix operator

$$
\tau = \begin{bmatrix} l_1 s_2 & l_2 + l_1 c_2 \\ 0 & l_2 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix}
$$



- **Note:** This matrix is the transpose of the Jacobian !!!

Jacobians in the Force Domain

- When forces act on a mechanism, work is done if the mechanism moves through a displacement.
- **Work** = a force acting through a distance and is a scalar with units of energy.
- The principle of virtual work allows us to make certain statements about the *static case.*
- Work has the units of energy, so it must be the same measured in any set of generalized coordinates.
- Equate the work done in Cartesian terms with the work done in joint-space terms.

$$\mathcal{F} \cdot \delta\chi = \tau \cdot \delta\Theta$$

$$\mathcal{F}^T \, \delta\chi = \tau^T \, \delta\Theta$$

Jacobians in the Force Domain

$$\mathcal{F}^T\,\delta\chi = \tau^T\,\delta\Theta$$

- $\mathcal{F}$ is a 6×1 Cartesian force-moment vector acting at the end-effector
- $\delta\chi$ is a 6×1 infinitesimal Cartesian displacement of the end-effector
- τ is a 6×1 vector of torques (forces) at the joints
- $\delta\Theta$ is a 6×1 vector of infinitesimal joint displacements

- The definition of the Jacobian

$$\delta\chi = J\,\delta\Theta$$

- So

$$\mathcal{F}^T J\,\delta\Theta = \tau^T\,\delta\Theta$$

- which must hold for all $\delta\Theta$; hence,

$$\mathcal{F}^T J = \tau^T$$

- Transposing both sides yields

$$\tau = J^T\,\mathcal{F}$$

Jacobians in the Force Domain

$$\tau = J^T \mathcal{F}$$

- The Jacobian transpose maps Cartesian forces acting at the hand into equivalent joint torques.

- **Note:** It allows us to convert a Cartesian quantity into a joint-space quantity without calculating any inverse kinematic functions.

- When the Jacobian is written with respect to frame {0}, then force vectors written in {0} can be transformed:

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

Changing the Jacobian's Frame of Expression

- **Note:** Why ${}^{0}J$ & ${}^{3}J$?

Changing the Jacobian's Frame of Expression

- **Note:** Why ${}^{0}J$ & ${}^{3}J$?

  - ${}^{0}J$ is usually used in **Position Control** problems
  - ${}^{3}J$ is usually used in **Force Control** problems

Changing the Jacobian's Frame of Expression

❑ Changing the frame

- A Jacobian written in frame {B} is given.

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

- **Objective:** Find its expression in another frame {A} (i.e. ${}^A J$).

$$\begin{bmatrix} {}^A v \\ {}^A \omega \end{bmatrix} = \left[\begin{array}{c|c} {}^A R_B & 0 \\ \hline 0 & {}^A R_B \end{array}\right] \begin{bmatrix} {}^B v \\ {}^B \omega \end{bmatrix}$$

$$\begin{bmatrix} {}^A v \\ {}^A \omega \end{bmatrix} = \left[\begin{array}{c|c} {}^A R_B & 0 \\ \hline 0 & {}^A R_B \end{array}\right] {}^B J(\Theta)\dot{\Theta}$$

$${}^A J(\Theta) = \left[\begin{array}{c|c} {}^A R_B & 0 \\ \hline 0 & {}^A R_B \end{array}\right] {}^B J(\Theta)$$

Cartesian Transformations of Velocity & Static Forces

- **Remark**
- The general velocity of a body:

$$\mathcal{V} = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

- v is a 3×1 linear velocity vector and ω is a 3×1 rotational velocity vector.

- The general force vectors:

$$\mathcal{F} = \begin{bmatrix} F \\ N \end{bmatrix}$$

- F is a 3×1 force vector and N is a 3×1 moment vector.

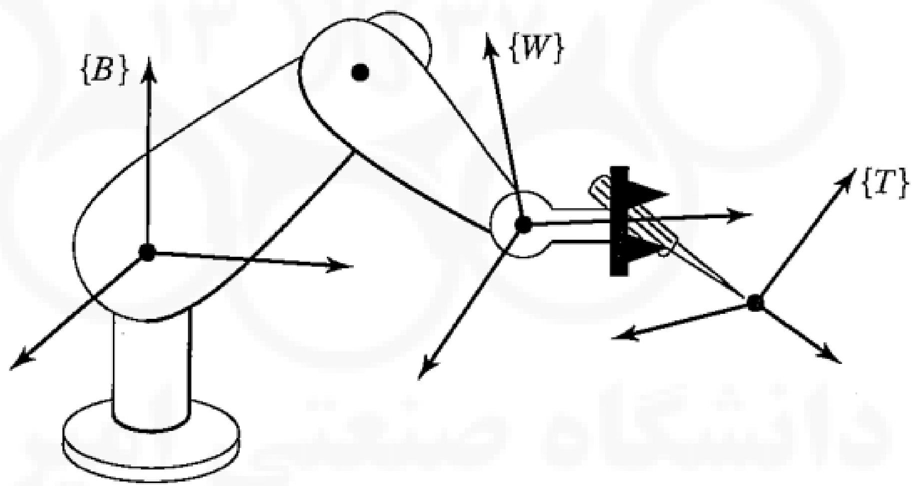
← Outline

Cartesian Transformations of Velocity & Static Forces

Objective:

- Matrix operator to transform general Cartesian vectors in frame $\{A\}$ to their description in frame $\{B\}$ when the two frames are rigidly connected.

- $({}^{A}\mathrm{v}_{A},\ {}^{A}\mathrm{F}_{A})$ is known, Find $({}^{B}\mathrm{v}_{B},\ {}^{B}\mathrm{F}_{B})$?



Cartesian Transformations of Velocity & Static Forces

Velocity Transformation

- Remember:

$$^{i+1}v_{i+1} = {}^{i+1}R_i\left({}^iv_i + {}^i\omega_i \times {}^iP_{i+1}\right)$$

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

- Frames are rigidly connected, so $\dot{\theta}_{i+1}$ is set to zero.

$$^{i+1}v_{i+1} = {}^{i+1}R_i\left({}^iv_i - {}^iP_{i+1} \times {}^i\omega_i\right) = {}^{i+1}R_i\left({}^iv_i - S\left({}^iP_{i+1}\right){}^i\omega_i\right)$$

$$^{i+1}\omega_{i+1} = {}^{i+1}R_i\,{}^i\omega_i$$

- where

$$S(P) = \begin{bmatrix} 0 & -p_z & p_y \\ p_x & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}$$

- so

$$^B\mathrm{v}_B = \begin{bmatrix} ^Bv_B \\ ^B\omega_B \end{bmatrix} = \begin{bmatrix} ^BR_A & -{}^BR_A\,S({}^AP_{BORG}) \\ 0 & ^BR_A \end{bmatrix} \begin{bmatrix} ^Av_A \\ ^A\omega_A \end{bmatrix} = {}^BT_{v_A}\,{}^A\mathrm{v}_A$$

- The 6×6 operator $^BT_{v_A}$ will be called a **velocity transformation.**

Cartesian Transformations of Velocity & Static Forces

Force-Moment Transformation

- Remember:

$$^{i}f_i = {}^{i}R_{i+1}\,{}^{i+1}f_{i+1}$$

$$^{i}n_i = {}^{i}R_{i+1}\,{}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i$$

- Rearrange it as:

$$^{i+1}f_{i+1} = {}^{i+1}R_i\,{}^{i}f_i$$

$$^{i+1}n_{i+1} = {}^{i+1}R_i\left({}^{i}n_i - {}^{i}P_{i+1} \times {}^{i}f_i\right) = {}^{i+1}R_i\left({}^{i}n_i - S\left({}^{i}P_{i+1}\right){}^{i}f_i\right)$$

- Therefore:

$$^{B}\mathrm{F}_B = \begin{bmatrix} {}^{B}F_B \\ {}^{B}N_B \end{bmatrix} = \begin{bmatrix} {}^{B}R_A & 0 \\ -{}^{B}R_A\,S({}^{A}P_{BORG}) & {}^{B}R_A \end{bmatrix} \begin{bmatrix} {}^{A}F_A \\ {}^{A}N_A \end{bmatrix} = {}^{B}T_{f\,A}\,{}^{A}\mathrm{F}_A$$

- The 6×6 operator ${}^{B}T_{f\,A}$ will be called **force-moment transformation.**

Cartesian Transformations of Velocity & Static Forces

- **Note:**
- Velocity and force transformations are similar to Jacobians in that they relate velocities and forces in different coordinate systems.

- ${}^{B}\mathrm{v}_{B} = {}^{B}T_{v\,A}\;{}^{A}\mathrm{v}_{A}$

$$ {}^{B}T_{v\,A} = \begin{bmatrix} {}^{B}R_{A} & -{}^{B}R_{A}\,S({}^{A}P_{BORG}) \\ 0 & {}^{B}R_{A} \end{bmatrix} $$

- ${}^{B}\mathrm{F}_{B} = {}^{B}T_{f\,A}\;{}^{A}\mathrm{F}_{A}$

$$ {}^{B}T_{f\,A} = \begin{bmatrix} {}^{B}R_{A} & 0 \\ -{}^{B}R_{A}\,S({}^{A}P_{BORG}) & {}^{B}R_{A} \end{bmatrix} $$

- As with Jacobians, we have

$$ {}^{B}T_{f\,A} = {}^{B}T_{v\,A}{}^{T} $$

Cartesian Transformations of Velocity & Static Forces

❖ **Example:**

- Suppose the tool is rigidly attached to the end-effector and the constant transformation matrix is given by:

$$ {}^{e}T_{t} = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} $$

- where $R = {}^{e}R_{t}$ & $d = {}^{e}P_{tORG}$

- Given EE velocity in its own frame (${}^{e}\nu_{e}$), find tool velocity (${}^{t}\nu_{t}$)?

- We have

$$ {}^{B}T_{\nu_A} = \begin{bmatrix} {}^{B}R_{A} & -{}^{B}R_{A}\, S({}^{A}P_{BORG}) \\ 0 & {}^{B}R_{A} \end{bmatrix} $$

- Assume $A = e$ & $B = t$, so

$$ {}^{t}T_{\nu_e} = \begin{bmatrix} {}^{t}R_{e} & -{}^{t}R_{e}\, S(d) \\ 0 & {}^{t}R_{e} \end{bmatrix} = \begin{bmatrix} R^{T} & -R^{T}\, S(d) \\ 0 & R^{T} \end{bmatrix} $$

$$ {}^{t}\nu_{t} = \begin{bmatrix} R^{T} & -R^{T}\, S(d) \\ 0 & R^{T} \end{bmatrix} {}^{e}\nu_{e} $$

The END

- **References:**

1) …