



دانشگاه سمنان

Semnan University
Faculty of Mechanical Engineering

دانشکده مهندسی مکانیک



دانشکده مهندسی مکانیک

درس رباتیک پیشرفته

ADVANCED ROBOTICS

Chapter 3 – Differential Kinematics and Statics

Class Lecture

□ CONTENTS:

❖ Chapter 1: Introduction

❖ Chapter 2: Kinematics

➔ ❖ Chapter 3: **Differential Kinematics and Statics**

❖ Chapter 4: Trajectory Planning

❖ Chapter 5: Actuators and Sensors

❖ Chapter 6: Control Architecture

❖ Chapter 7: Dynamics

❖ Chapter 8: Motion Control

3. DIFFERENTIAL KINEMATICS AND STATICS

□ Differential Kinematics:

- ❖ The relationship between the joint velocities and the corresponding end-effector linear and angular
- ❖ This mapping is described by a matrix, termed *Geometric Jacobian*, which depends on the manipulator configuration.

□ Analytical Jacobian:

- ❖ The end-effector pose is expressed with reference to a minimal representation in the operational space, it is possible to compute the Jacobian matrix via differentiation of the direct kinematics function with respect to the joint variables.



3. DIFFERENTIAL KINEMATICS AND STATICS

- The *Jacobian* is used for:
 - ❖ Finding singularities
 - ❖ Analyzing redundancy
 - ❖ Determining inverse kinematics algorithms
 - ❖ Describing the mapping between forces applied to the end-effector and resulting torques at the joints (statics)
 - ❖ Deriving dynamic equations of motion
 - ❖ Designing operational space control schemes



3.1 GEOMETRIC JACOBIAN

- The direct kinematics equation for an n-DOF manipulator:

$$T_e(\mathbf{q}) = \begin{bmatrix} \mathbf{R}_e(\mathbf{q}) & \mathbf{p}_e(\mathbf{q}) \\ \mathbf{0}^T & 1 \end{bmatrix} \quad \mathbf{q} = [q_1 \quad \dots \quad q_n]^T$$

- It is desired to express the end-effector linear velocity $\dot{\mathbf{p}}_e$ and angular velocity $\boldsymbol{\omega}_e$ as a function of the joint velocities $\dot{\mathbf{q}}$.

$$\rightarrow \dot{\mathbf{p}}_e = \mathbf{J}_P(\mathbf{q})\dot{\mathbf{q}}$$

$$\rightarrow \boldsymbol{\omega}_e = \mathbf{J}_O(\mathbf{q})\dot{\mathbf{q}}$$



3.1 GEOMETRIC JACOBIAN

- The manipulator differential kinematics equation:

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

- ❖ The $(6 \times n)$ matrix \mathbf{J} is the manipulator *geometric Jacobian*:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_P \\ \mathbf{J}_O \end{bmatrix}$$

- ✓ $\mathbf{J}_P : (3 \times n)$ matrix relating the contribution of joint velocities to end-effector linear velocity
- ✓ $\mathbf{J}_O : (3 \times n)$ matrix relating the contribution of joint velocities to end-effector angular velocity



3.1.1 DERIVATIVE OF A ROTATION MATRIX

□ The derivative of a rotation matrix with respect to time:

❖ A time-varying rotation matrix $R(t)$

$$R = R(t)$$

$$\rightarrow R(t)R^T(t) = I$$

$$\rightarrow \dot{R}(t)R^T(t) + R(t)\dot{R}^T(t) = O$$

❖ Set S:

$$S(t) = \dot{R}(t)R^T(t)$$

✓ The (3×3) matrix S is skew-symmetric since:

$$S(t) + S^T(t) = O$$



3.1.1 DERIVATIVE OF A ROTATION MATRIX

- ❖ Postmultiplying both sides by $R(t)$:

$$S(t) = \dot{R}(t)R^T(t) \quad \longrightarrow \quad \dot{R}(t) = S(t)R(t)$$

- ✓ The time derivative of $R(t)$ is expressed as a function of $R(t)$.

- ❖ Consider a constant vector p' and the vector $p(t) = R(t)p'$:

$$\dot{p}(t) = \dot{R}(t)p' \quad \longrightarrow \quad \dot{p}(t) = S(t)R(t)p'$$

- ✓ It is known from mechanics that ($\omega(t)$ denotes the angular velocity of frame $R(t)$ with respect to the reference frame):

$$\longrightarrow \quad \dot{p}(t) = \omega(t) \times R(t)p'$$



3.1.1 DERIVATIVE OF A ROTATION MATRIX

- Therefore, the matrix operator $S(t)$ describes the vector product between the vector ω and the vector $R(t)p'$.

$$\omega(t) = [\omega_x \quad \omega_y \quad \omega_z]^T$$

$$\rightarrow S = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

$$\rightarrow S(t) = S(\omega(t)) \quad \rightarrow \dot{R} = S(\omega)R$$

❖ It can be shown that:

$$RS(\omega)R^T = S(R\omega)$$



3.1.1 DERIVATIVE OF A ROTATION MATRIX

□ Example 3.1

❖ The elementary rotation matrix about axis z

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \rightarrow S(t) &= \begin{bmatrix} -\dot{\alpha} \sin \alpha & -\dot{\alpha} \cos \alpha & 0 \\ \dot{\alpha} \cos \alpha & -\dot{\alpha} \sin \alpha & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -\dot{\alpha} & 0 \\ \dot{\alpha} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = S(\omega(t)). \end{aligned}$$

$$\omega = [0 \quad 0 \quad \dot{\alpha}]^T$$



3.1.1 DERIVATIVE OF A ROTATION MATRIX

- The coordinate transformation of a point P from Frame 1 to Frame 0:

$$p^0 = o_1^0 + R_1^0 p^1$$

$$\rightarrow \dot{p}^0 = \dot{o}_1^0 + R_1^0 \dot{p}^1 + \dot{R}_1^0 p^1$$

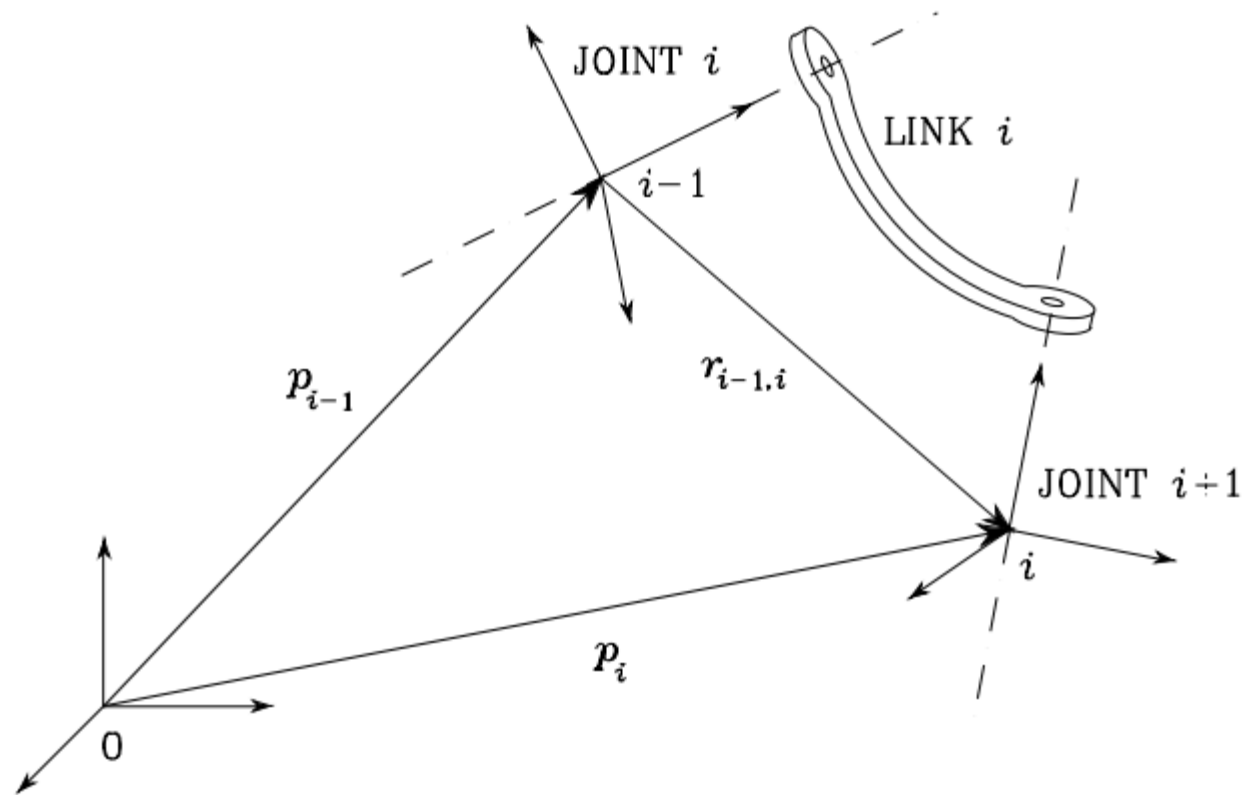
$$\rightarrow \dot{p}^0 = \dot{o}_1^0 + R_1^0 \dot{p}^1 + S(\omega_1^0) R_1^0 p^1$$

$$\rightarrow \dot{p}^0 = \dot{o}_1^0 + R_1^0 \dot{p}^1 + \omega_1^0 \times r_1^0$$

- ❖ which is the known form of the velocity composition rule.

3.1.2 LINK VELOCITIES

- Using Denavit–Hartenberg convention:



3.1.2 LINK VELOCITIES

□ p_{i-1} and p_i : Position vectors of the origins of Frames $i-1$ and i

$$p_i = p_{i-1} + R_{i-1} r_{i-1,i}^{i-1}$$

$$\begin{aligned} \rightarrow \dot{p}_i &= \dot{p}_{i-1} + R_{i-1} \dot{r}_{i-1,i}^{i-1} + \omega_{i-1} \times R_{i-1} r_{i-1,i}^{i-1} \\ &= \dot{p}_{i-1} + v_{i-1,i} + \omega_{i-1} \times r_{i-1,i} \end{aligned}$$

❖ The linear velocity of Link i as a function of the translational and rotational velocities of Link $i-1$



3.1.2 LINK VELOCITIES

□ Link angular velocity:

$$R_i = R_{i-1}R_i^{i-1}$$

$$\rightarrow S(\omega_i)R_i = S(\omega_{i-1})R_i + R_{i-1}S(\omega_{i-1,i}^{i-1})R_i^{i-1}$$

$$\rightarrow R_{i-1}S(\omega_{i-1,i}^{i-1})R_i^{i-1} = R_{i-1}S(\omega_{i-1,i}^{i-1})R_{i-1}^T R_{i-1}R_i^{i-1}$$

$$\rightarrow R_{i-1}S(\omega_{i-1,i}^{i-1})R_i^{i-1} = S(R_{i-1}\omega_{i-1,i}^{i-1})R_i$$

$$\rightarrow S(\omega_i)R_i = S(\omega_{i-1})R_i + S(R_{i-1}\omega_{i-1,i}^{i-1})R_i$$

$$\rightarrow \omega_i = \omega_{i-1} + R_{i-1}\omega_{i-1,i}^{i-1} = \omega_{i-1} + \omega_{i-1,i}$$



3.1.2 LINK VELOCITIES

- Prismatic joint

$$\boldsymbol{\omega}_{i-1,i} = \mathbf{0}$$

$$\mathbf{v}_{i-1,i} = \dot{d}_i \mathbf{z}_{i-1}$$

$$\begin{aligned} \rightarrow \quad \boldsymbol{\omega}_i &= \boldsymbol{\omega}_{i-1} \\ \dot{\mathbf{p}}_i &= \dot{\mathbf{p}}_{i-1} + \dot{d}_i \mathbf{z}_{i-1} + \boldsymbol{\omega}_i \times \mathbf{r}_{i-1,i} \end{aligned}$$

3.1.2 LINK VELOCITIES

- Revolute joint

$$\boldsymbol{\omega}_{i-1,i} = \dot{\vartheta}_i \mathbf{z}_{i-1}$$

$$\mathbf{v}_{i-1,i} = \boldsymbol{\omega}_{i-1,i} \times \mathbf{r}_{i-1,i}$$

$$\begin{aligned} \boldsymbol{\omega}_i &= \boldsymbol{\omega}_{i-1} + \dot{\vartheta}_i \mathbf{z}_{i-1} \\ \rightarrow \dot{\mathbf{p}}_i &= \dot{\mathbf{p}}_{i-1} + \boldsymbol{\omega}_i \times \mathbf{r}_{i-1,i} \end{aligned}$$

3.1.3 JACOBIAN COMPUTATION

□ it is convenient to proceed separately for linear velocity and angular velocity.

❖ The *linear velocity*

$$\dot{\mathbf{p}}_e = \sum_{i=1}^n \frac{\partial \mathbf{p}_e}{\partial q_i} \dot{q}_i = \sum_{i=1}^n \mathbf{J}_{P_i} \dot{q}_i$$

✓ A prismatic joint

$$q_i = d_i \quad \longrightarrow \quad \dot{q}_i \mathbf{J}_{P_i} = \dot{d}_i \mathbf{z}_{i-1} \quad \longrightarrow \quad \mathbf{J}_{P_i} = \mathbf{z}_{i-1}$$

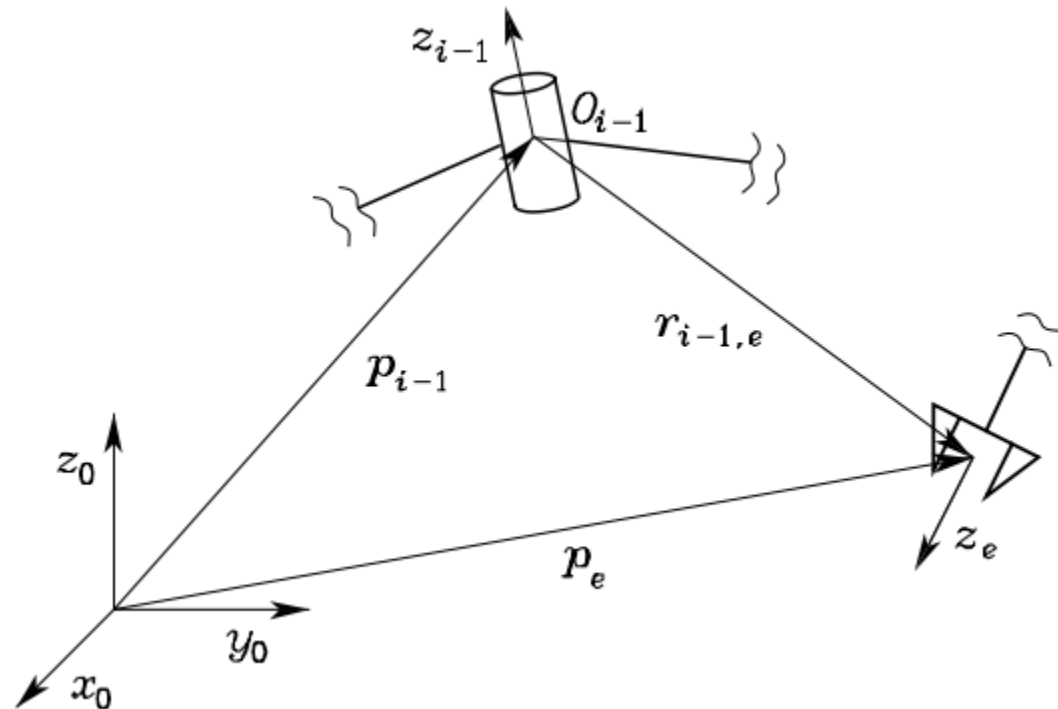
✓ A revolute joint

$$\begin{aligned} \dot{q}_i \mathbf{J}_{P_i} &= \boldsymbol{\omega}_{i-1,i} \times \mathbf{r}_{i-1,e} = \dot{\vartheta}_i \mathbf{z}_{i-1} \times (\mathbf{p}_e - \mathbf{p}_{i-1}) \\ &\longrightarrow \mathbf{J}_{P_i} = \mathbf{z}_{i-1} \times (\mathbf{p}_e - \mathbf{p}_{i-1}) \end{aligned}$$



3.1.3 JACOBIAN COMPUTATION

- Velocity contribution of a revolute joint to the end-effector linear velocity



3.1.2 LINK VELOCITIES

❖ The angular velocity

$$\omega_e = \omega_n = \sum_{i=1}^n \omega_{i-1,i} = \sum_{i=1}^n \mathcal{J}_{O_i} \dot{q}_i$$

✓ A prismatic joint

$$\dot{q}_i \mathcal{J}_{O_i} = \mathbf{0} \quad \rightarrow \quad \mathcal{J}_{O_i} = \mathbf{0}$$

✓ A revolute joint

$$\dot{q}_i \mathcal{J}_{O_i} = \dot{\vartheta}_i \mathbf{z}_{i-1} \quad \rightarrow \quad \mathcal{J}_{O_i} = \mathbf{z}_{i-1}$$



3.1.2 LINK VELOCITIES

□ The Jacobian

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{P1} & \dots & \mathbf{J}_{Pn} \\ \mathbf{J}_{O1} & & \mathbf{J}_{On} \end{bmatrix}$$

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

3.1.2 LINK VELOCITIES

- The vectors z_{i-1} , p_e and p_{i-1} are all functions of the joint variables:

$$\rightarrow z_{i-1} = R_1^0(q_1) \dots R_{i-1}^{i-2}(q_{i-1}) z_0$$

$$z_0 = [0 \ 0 \ 1]^T$$

$$\rightarrow \tilde{p}_e = A_1^0(q_1) \dots A_n^{n-1}(q_n) \tilde{p}_0$$

$$\tilde{p}_0 = [0 \ 0 \ 0 \ 1]^T$$

$$\rightarrow \tilde{p}_{i-1} = A_1^0(q_1) \dots J^u = \begin{bmatrix} R^u & O \\ O & R^u \end{bmatrix} J$$

- ❖ Jacobian in a different Frame u:

$$\begin{bmatrix} \dot{p}_e^u \\ \omega_e^u \end{bmatrix} = \begin{bmatrix} R^u & O \\ O & R^u \end{bmatrix} \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} R^u & O \\ O & R^u \end{bmatrix} J \dot{q}$$

$$\rightarrow J^u = \begin{bmatrix} R^u & O \\ O & R^u \end{bmatrix} J$$



3.2 JACOBIAN OF TYPICAL MANIPULATOR STRUCTURES

□ 3.2.1 Three-link Planar Arm

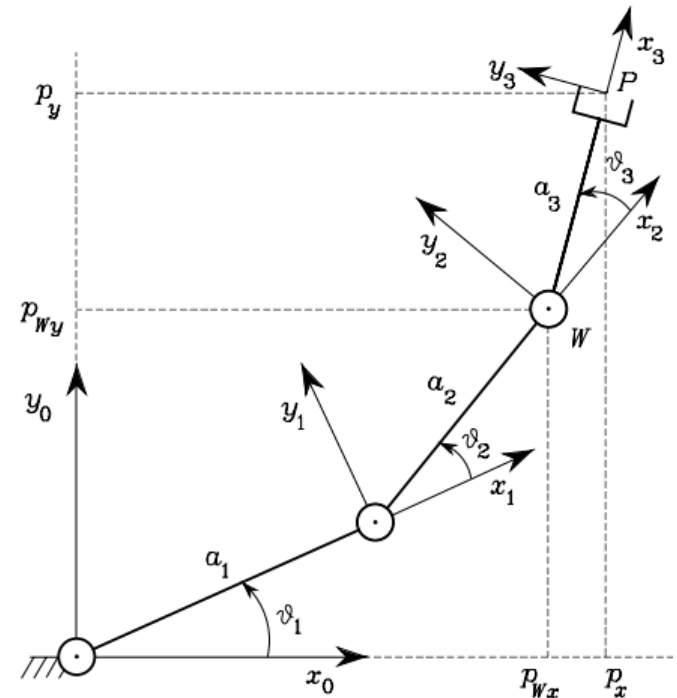
$$\rightarrow J(q) = \begin{bmatrix} z_0 \times (p_3 - p_0) & z_1 \times (p_3 - p_1) & z_2 \times (p_3 - p_2) \\ z_0 & z_1 & z_2 \end{bmatrix}$$

$$p_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad p_1 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix}$$

$$p_2 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ 0 \end{bmatrix}$$

$$p_3 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 \end{bmatrix}$$

$$z_0 = z_1 = z_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

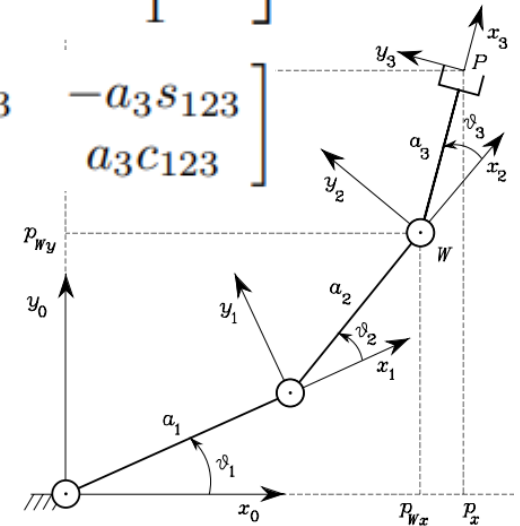


3.2 JACOBIAN OF TYPICAL MANIPULATOR STRUCTURES

□ 3.2.1 Three-link Planar Arm

$$\rightarrow J = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\rightarrow J_P = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \end{bmatrix}$$



3.2 JACOBIAN OF TYPICAL MANIPULATOR STRUCTURES

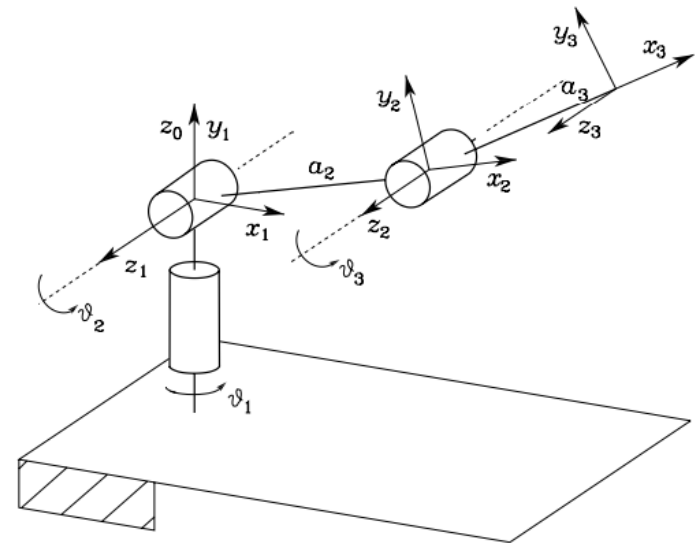
□ 3.2.2 Anthropomorphic Arm

$$\rightarrow J = \begin{bmatrix} z_0 \times (p_3 - p_0) & z_1 \times (p_3 - p_1) & z_2 \times (p_3 - p_2) \\ z_0 & z_1 & z_2 \end{bmatrix}$$

$$p_0 = p_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad p_2 = \begin{bmatrix} a_2 c_1 c_2 \\ a_2 s_1 c_2 \\ a_2 s_2 \end{bmatrix}$$

$$p_3 = \begin{bmatrix} c_1 (a_2 c_2 + a_3 c_{23}) \\ s_1 (a_2 c_2 + a_3 c_{23}) \\ a_2 s_2 + a_3 s_{23} \end{bmatrix}$$

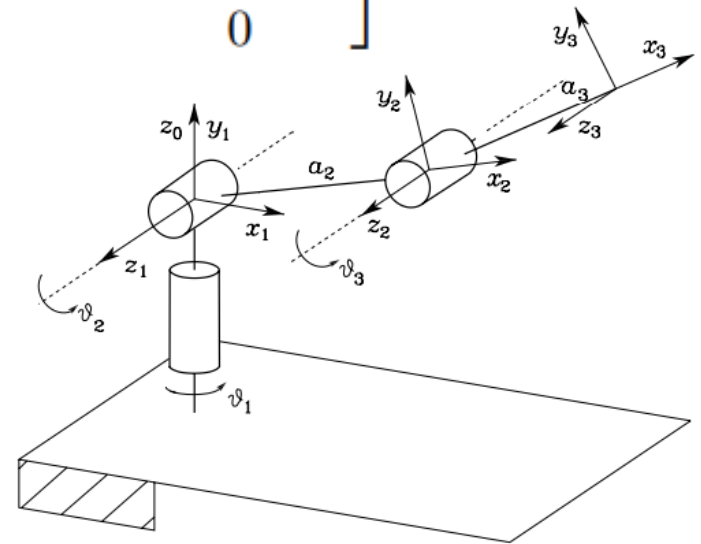
$$z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad z_1 = z_2 = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}$$



3.2 JACOBIAN OF TYPICAL MANIPULATOR STRUCTURES

□ 3.2.2 Anthropomorphic Arm

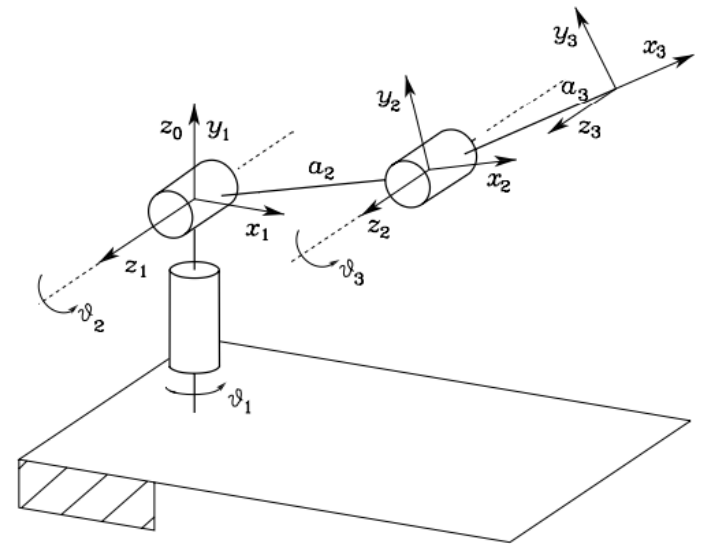
$$\rightarrow J = \begin{bmatrix} -s_1(a_2c_2 + a_3c_{23}) & -c_1(a_2s_2 + a_3s_{23}) & -a_3c_1s_{23} \\ c_1(a_2c_2 + a_3c_{23}) & -s_1(a_2s_2 + a_3s_{23}) & -a_3s_1s_{23} \\ 0 & a_2c_2 + a_3c_{23} & a_3c_{23} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix}$$



3.2 JACOBIAN OF TYPICAL MANIPULATOR STRUCTURES

□ 3.2.2 Anthropomorphic Arm

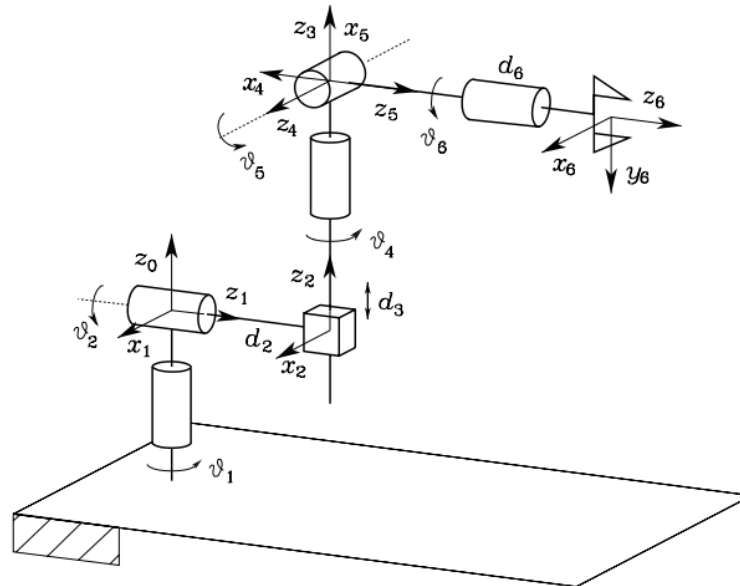
$$\rightarrow \mathbf{J}_P = \begin{bmatrix} -s_1(a_2c_2 + a_3c_{23}) & -c_1(a_2s_2 + a_3s_{23}) & -a_3c_1s_{23} \\ c_1(a_2c_2 + a_3c_{23}) & -s_1(a_2s_2 + a_3s_{23}) & -a_3s_1s_{23} \\ 0 & a_2c_2 + a_3c_{23} & a_3c_{23} \end{bmatrix}$$



3.2 JACOBIAN OF TYPICAL MANIPULATOR STRUCTURES

□ 3.2.3 Stanford Manipulator

$$\rightarrow J = \begin{bmatrix} z_0 \times (p_6 - p_0) & z_1 \times (p_6 - p_1) & z_2 & z_3 \times (p_6 - p_3) & z_4 \times (p_6 - p_4) & z_5 \times (p_6 - p_5) \\ z_0 & z_1 & \mathbf{0} & z_3 & z_4 & z_5 \end{bmatrix}$$

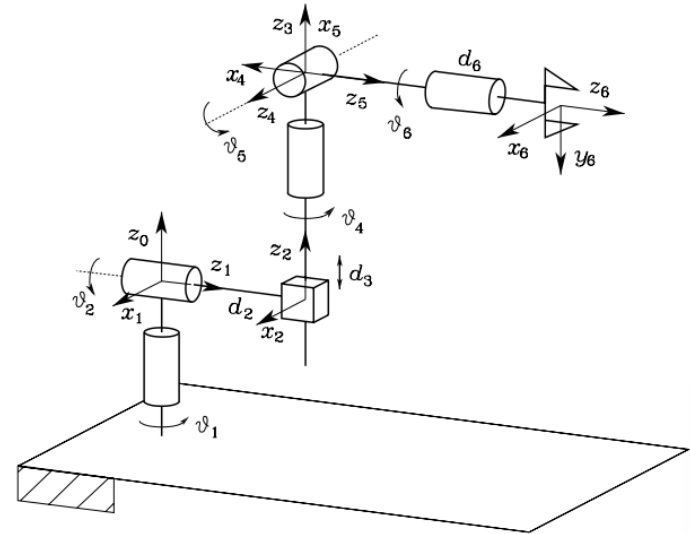


3.2 JACOBIAN OF TYPICAL MANIPULATOR STRUCTURES

□ 3.2.3 Stanford Manipulator

$$\mathbf{p}_0 = \mathbf{p}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{p}_3 = \mathbf{p}_4 = \mathbf{p}_5 = \begin{bmatrix} c_1 s_2 d_3 - s_1 d_2 \\ s_1 s_2 d_3 + c_1 d_2 \\ c_2 d_3 \end{bmatrix}$$



$$\mathbf{p}_6 = \begin{bmatrix} c_1 s_2 d_3 - s_1 d_2 + (c_1(c_2 c_4 s_5 + s_2 c_5) - s_1 s_4 s_5) d_6 \\ s_1 s_2 d_3 + c_1 d_2 + (s_1(c_2 c_4 s_5 + s_2 c_5) + c_1 s_4 s_5) d_6 \\ c_2 d_3 + (-s_2 c_4 s_5 + c_2 c_5) d_6 \end{bmatrix}$$

3.2 JACOBIAN OF TYPICAL MANIPULATOR STRUCTURES

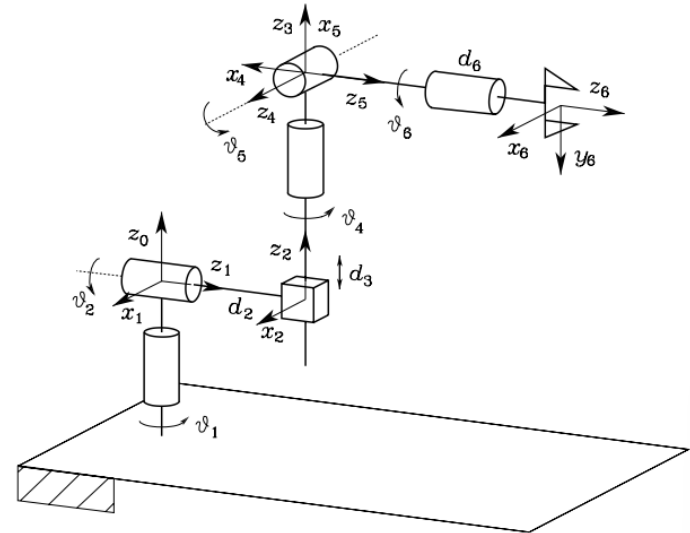
□ 3.2.3 Stanford Manipulator

$$z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad z_1 = \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix}$$

$$z_2 = z_3 = \begin{bmatrix} c_1 s_2 \\ s_1 s_2 \\ c_2 \end{bmatrix}$$

$$z_4 = \begin{bmatrix} -c_1 c_2 s_4 - s_1 c_4 \\ -s_1 c_2 s_4 + c_1 c_4 \\ s_2 s_4 \end{bmatrix}$$

$$z_5 = \begin{bmatrix} c_1 (c_2 c_4 s_5 + s_2 c_5) - s_1 s_4 s_5 \\ s_1 (c_2 c_4 s_5 + s_2 c_5) + c_1 s_4 s_5 \\ -s_2 c_4 s_5 + c_2 c_5 \end{bmatrix}$$



3.3 KINEMATIC SINGULARITIES

- The Jacobian in the differential kinematics equation of a manipulator defines a linear mapping:

$$\rightarrow \mathbf{v}_e = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad \mathbf{v}_e = [\dot{\mathbf{p}}_e^T \quad \boldsymbol{\omega}_e^T]^T$$

- ***Kinematic Singularities***: Configurations that \mathbf{J} is rank-deficient
 - ❖ Singularities represent configurations at which mobility of the structure is reduced, i.e., it is not possible to impose an arbitrary motion to the end-effector.
 - ❖ When the structure is at a singularity, infinite solutions to the inverse kinematics problem may exist.
 - ❖ In the neighborhood of a singularity, small velocities in the operational space may cause large velocities in the joint space.



3.3 KINEMATIC SINGULARITIES

□ Boundary singularities:

- ❖ Occur when the manipulator is either outstretched or retracted.
- ❖ Do not represent a true drawback, since they can be avoided on condition that the manipulator is not driven to the boundaries of its reachable workspace.

□ Internal singularities:

- ❖ Occur inside the reachable workspace and are generally caused by the alignment of two or more axes of motion, or else by the attainment of particular end-effector configurations.
- ❖ Constitute a serious problem, as they can be encountered anywhere in the reachable workspace for a planned path in the operational space.



3.3 KINEMATIC SINGULARITIES

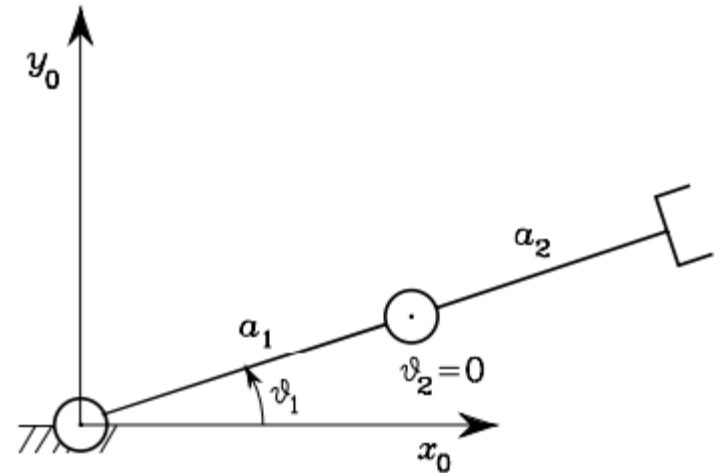
□ Example 3.2

❖ A two-link planar arm

$$\mathbf{J} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \end{bmatrix}$$

➔ $\det(\mathbf{J}) = a_1 a_2 s_2$

➔ $\vartheta_2 = 0 \quad \vartheta_2 = \pi$



- ❖ When the arm tip is located either on the outer ($\vartheta_2 = 0$) or on the inner ($\vartheta_2 = \pi$) boundary of the reachable workspace.
- ❖ Two column vectors of the Jacobian become parallel, and thus the Jacobian rank becomes one.

3.3.1 SINGULARITY DECOUPLING

- ❑ Computation of internal singularities via the Jacobian determinant may be tedious and of no easy solution for complex structures.

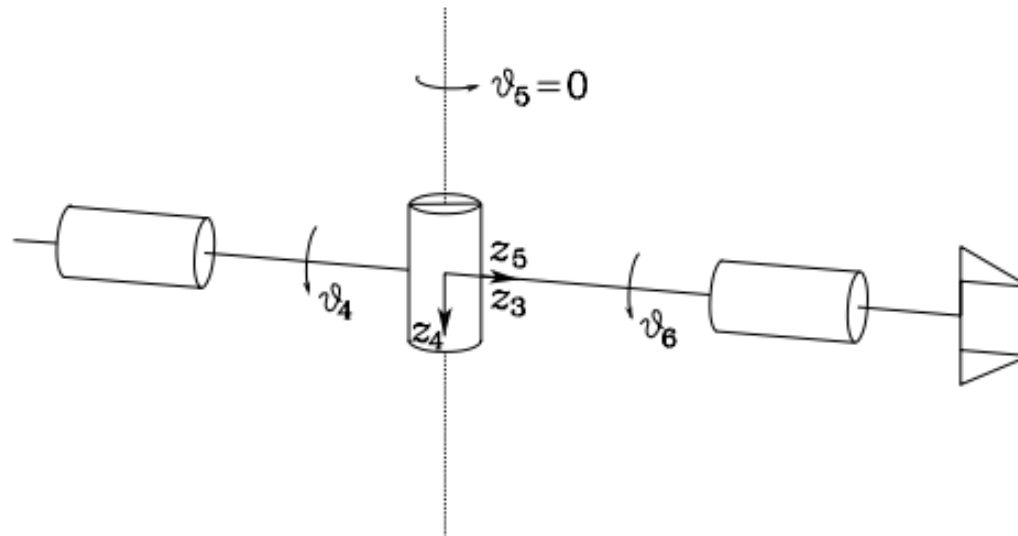
- ❑ For manipulators having a spherical wrist, by analogy with what has already been seen for inverse kinematics, it is possible to split the problem of singularity computation into two separate problems:
 - ❖ Computation of arm singularities resulting from the motion of first 3 or more links
 - ❖ Computation of wrist singularities resulting from the motion of the wrist joints.

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

3.3.2 WRIST SINGULARITIES

- On the basis of singularity decoupling

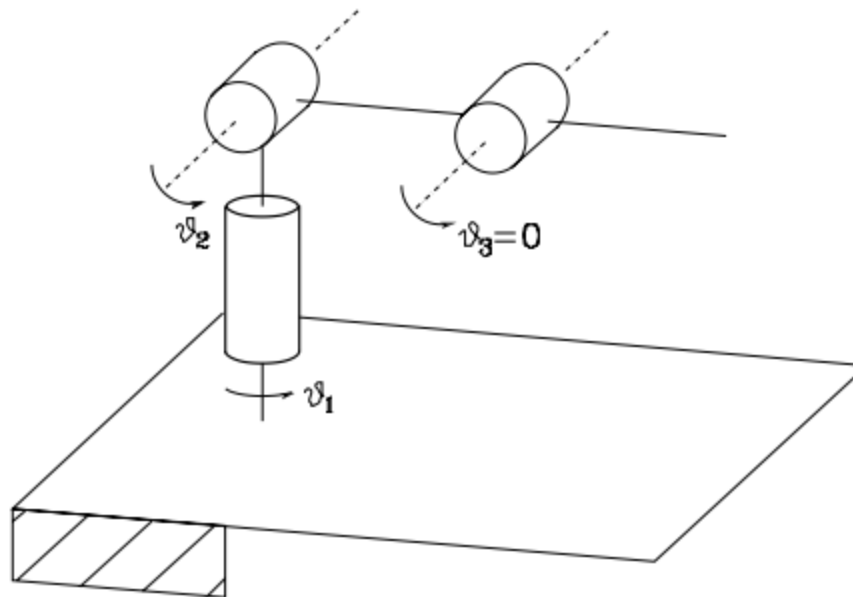
$$\rightarrow \vartheta_5 = 0 \quad \vartheta_5 = \pi$$



3.3.3 ARM SINGULARITIES

- On the basis of singularity decoupling

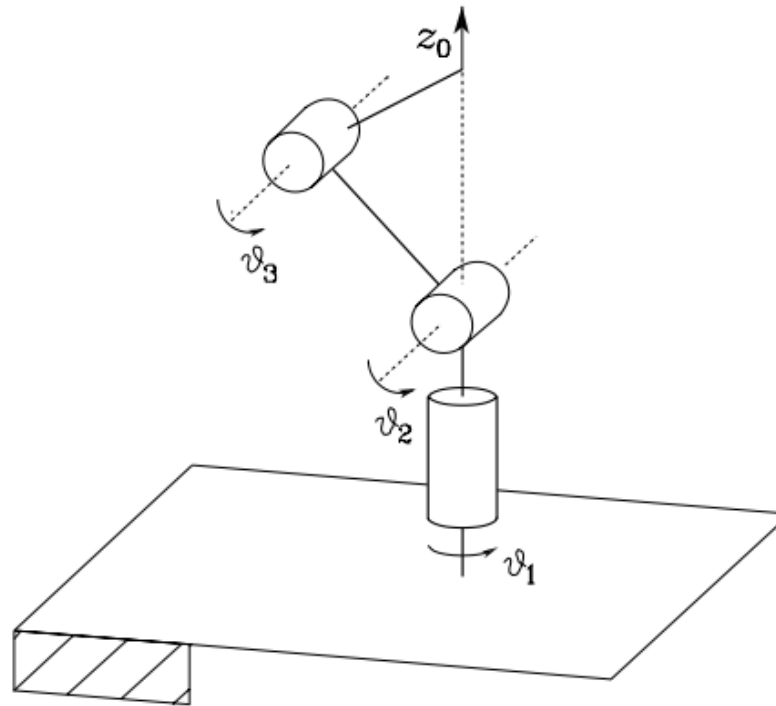
$$\det(\mathbf{J}_P) = -a_2 a_3 s_3 (a_2 c_2 + a_3 c_{23}) \quad \rightarrow \quad \vartheta_3 = 0 \quad \vartheta_3 = \pi$$



3.3.3 ARM SINGULARITIES

- On the basis of singularity decoupling

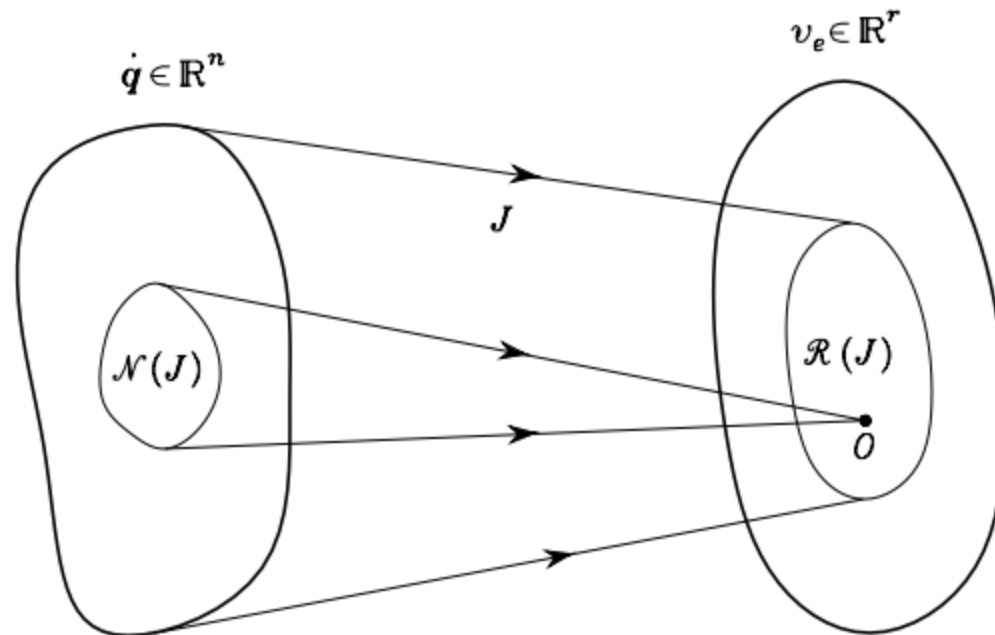
$$\det(\mathbf{J}_P) = -a_2 a_3 s_3 (a_2 c_2 + a_3 c_{23}) \rightarrow p_x = p_y = 0$$



3.4 ANALYSIS OF REDUNDANCY

□ *Redundancy* is related to:

- ❖ The number n of DOFs of the structure
- ❖ The number m of operational space variables
- ❖ The number r of operational space variables necessary to specify a given task



3.5 INVERSE DIFFERENTIAL KINEMATICS

- ❑ The inverse kinematics problem admits closed form solutions only for manipulators having a simple kinematic structure.
- ❑ Problems arise whenever the end-effector attains a particular position and/or orientation in the operational space, or the structure is complex and it is not possible to relate the end-effector pose to different sets of joint variables, or else the manipulator is redundant.
- ❑ These limitations are caused by the highly nonlinear relationship between joint space variables and operational space variables.
- ❑ On the other hand, the differential kinematics equation represents a linear mapping between the joint velocity space and the operational velocity space, although it varies with the current configuration.
- ❑ This fact suggests the possibility to utilize the differential kinematics equation to tackle the inverse kinematics problem.



3.6 ANALYTICAL JACOBIAN

- If the end-effector pose is specified in terms of a minimal number of parameters, it is possible to compute the Jacobian via differentiation of the direct kinematics function with respect to the joint variables.

- The *Analytical Technique*:

$$\rightarrow \dot{\mathbf{p}}_e = \frac{\partial \mathbf{p}_e}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_P(\mathbf{q}) \dot{\mathbf{q}}$$

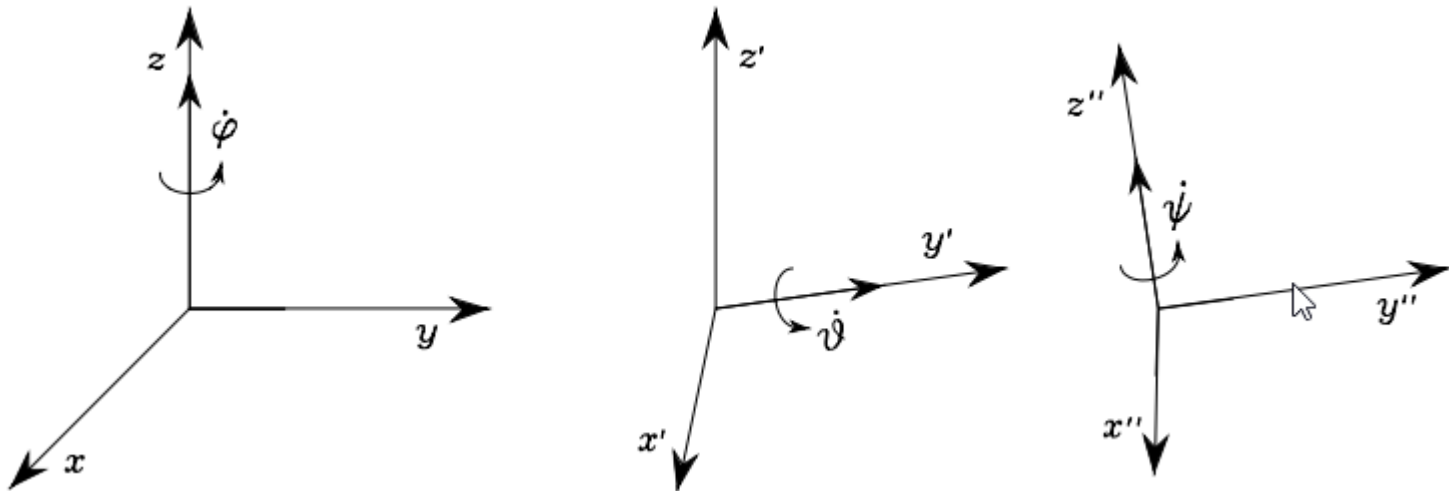
$$\rightarrow \dot{\phi}_e = \frac{\partial \phi_e}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_\phi(\mathbf{q}) \dot{\mathbf{q}}$$

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \dot{\phi}_e \end{bmatrix} = \begin{bmatrix} \mathbf{J}_P(\mathbf{q}) \\ \mathbf{J}_\phi(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_A(\mathbf{q}) \dot{\mathbf{q}} \quad \rightarrow \quad \mathbf{J}_A(\mathbf{q}) = \frac{\partial \mathbf{k}(\mathbf{q})}{\partial \mathbf{q}}$$



3.6 ANALYTICAL JACOBIAN

- The Analytical Jacobian J_A is different from the Geometric Jacobian J , since the end-effector angular velocity ω_e with respect to the base frame is not given by $\dot{\varphi}_e$
- Using ZYZ Euler angles:



3.6 ANALYTICAL JACOBIAN

□ Using ZYZ Euler angles:

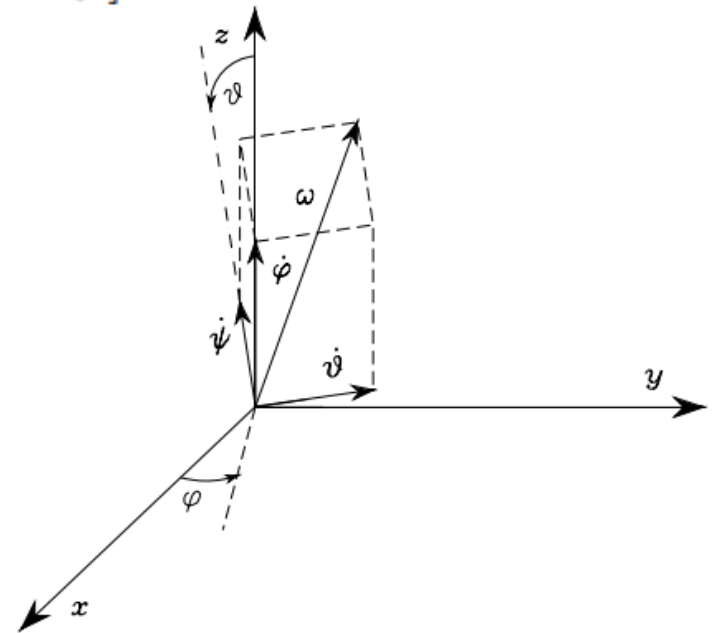
$$\dot{\varphi}: [\omega_x \ \omega_y \ \omega_z]^T = \dot{\varphi} [0 \ 0 \ 1]^T$$

$$\dot{\vartheta}: [\omega_x \ \omega_y \ \omega_z]^T = \dot{\vartheta} [-s_\varphi \ c_\varphi \ 0]^T$$

$$\dot{\psi}: [\omega_x \ \omega_y \ \omega_z]^T = \dot{\psi} [c_\varphi s_\vartheta \ s_\varphi s_\vartheta \ c_\vartheta]^T$$

$$\rightarrow \omega_e = \mathbf{T}(\phi_e) \dot{\phi}_e$$

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



3.6 ANALYTICAL JACOBIAN

- The three components of ω_e represent the components of angular velocity with respect to the base frame. Instead, the three elements of $\dot{\varphi}_e$ represent nonorthogonal components of angular velocity defined with respect to the axes of a frame that varies as the end-effector orientation varies.
- $\det(T) = -s\mathcal{G} \rightarrow$ the relationship cannot be inverted for $\mathcal{G} = 0, \pi$.
- This means that, even though all rotational velocities of the end-effector frame can be expressed by means of a suitable angular velocity vector ω_e , there exist angular velocities which cannot be expressed by means of $\dot{\varphi}_e$ when the orientation of the end-effector frame causes $s\mathcal{G} = 0$.



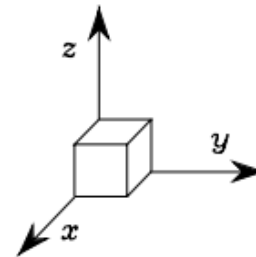
3.6 ANALYTICAL JACOBIAN

□ Example 3.3

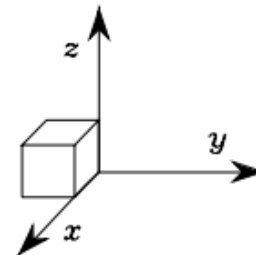
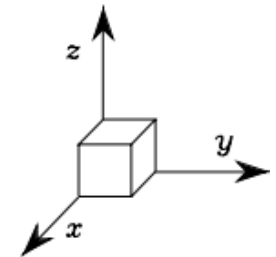
- $\omega = [\pi/2 \ 0 \ 0]^T \quad 0 \leq t \leq 1$
 $\omega = [0 \ \pi/2 \ 0]^T \quad 1 < t \leq 2,$

- $\omega = [0 \ \pi/2 \ 0]^T \quad 0 \leq t \leq 1$
 $\omega = [\pi/2 \ 0 \ 0]^T \quad 1 < t \leq 2.$

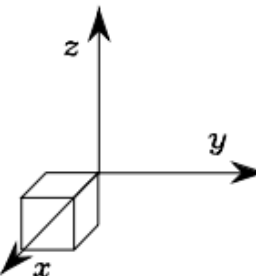
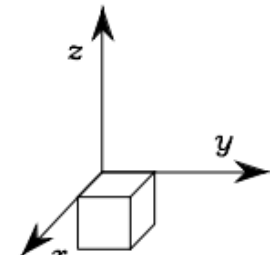
→ $\int_0^2 \omega dt = [\pi/2 \ \pi/2 \ 0]^T$



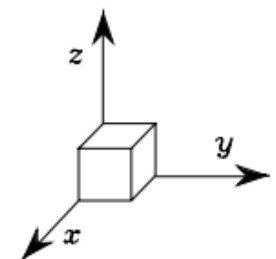
t = 0



t = 1



t = 2



3.6 ANALYTICAL JACOBIAN

- The *Analytical Jacobian* and *Geometric Jacobian* relation:

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

$$\rightarrow \mathbf{v}_e = \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{T}(\phi_e) \end{bmatrix} \dot{\mathbf{x}}_e = \mathbf{T}_A(\phi_e) \dot{\mathbf{x}}_e$$

$$\rightarrow \mathbf{J} = \mathbf{T}_A(\phi) \mathbf{J}_A$$

3.7 INVERSE KINEMATICS ALGORITHMS

□ Numerical implementation:

❖ For $n = r$ case

$$\rightarrow \dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\mathbf{v}_e$$

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \dot{\mathbf{q}}(t_k)\Delta t$$

$$\rightarrow \mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \mathbf{J}^{-1}(\mathbf{q}(t_k))\mathbf{v}_e(t_k)\Delta t.$$

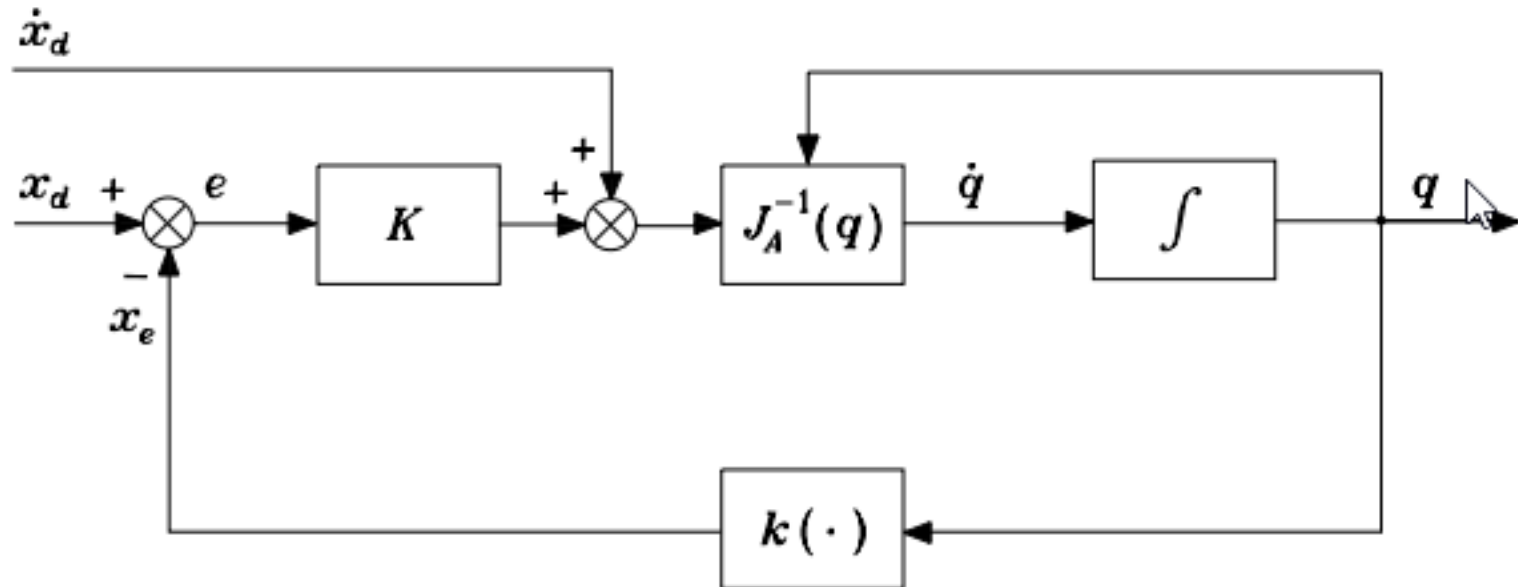
- ❖ The computed joint velocities do not coincide with these in the continuous time. Therefore, reconstruction of joint variables is entrusted to a numerical integration which involves *drift phenomena* of the solution.
- ❖ This inconvenience can be overcome by resorting to a solution scheme that accounts for the *operational space error*:

$$\rightarrow \mathbf{e} = \mathbf{x}_d - \mathbf{x}_e$$



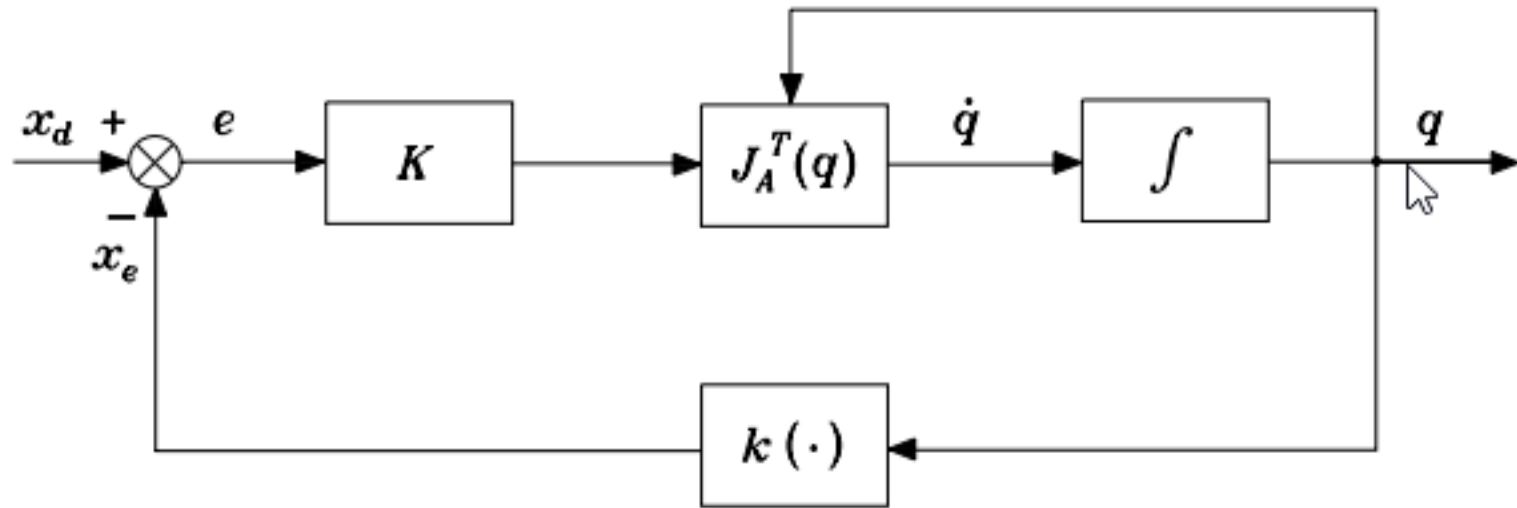
3.7 INVERSE KINEMATICS ALGORITHMS

- Inverse kinematics algorithm with Jacobian inverse



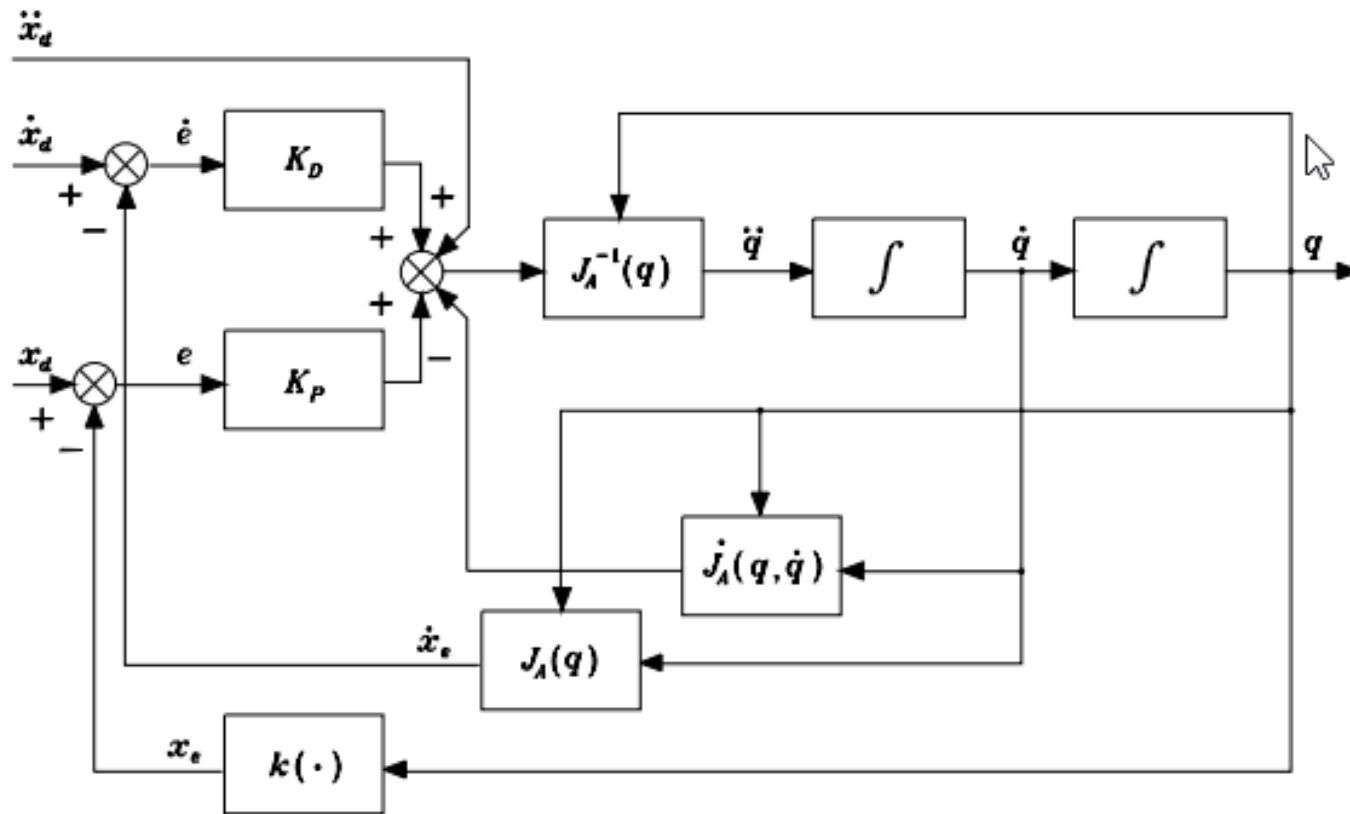
3.7 INVERSE KINEMATICS ALGORITHMS

- Inverse kinematics algorithm with Jacobian transpose



3.7 INVERSE KINEMATICS ALGORITHMS

- Second-order inverse kinematics algorithm with Jacobian inverse



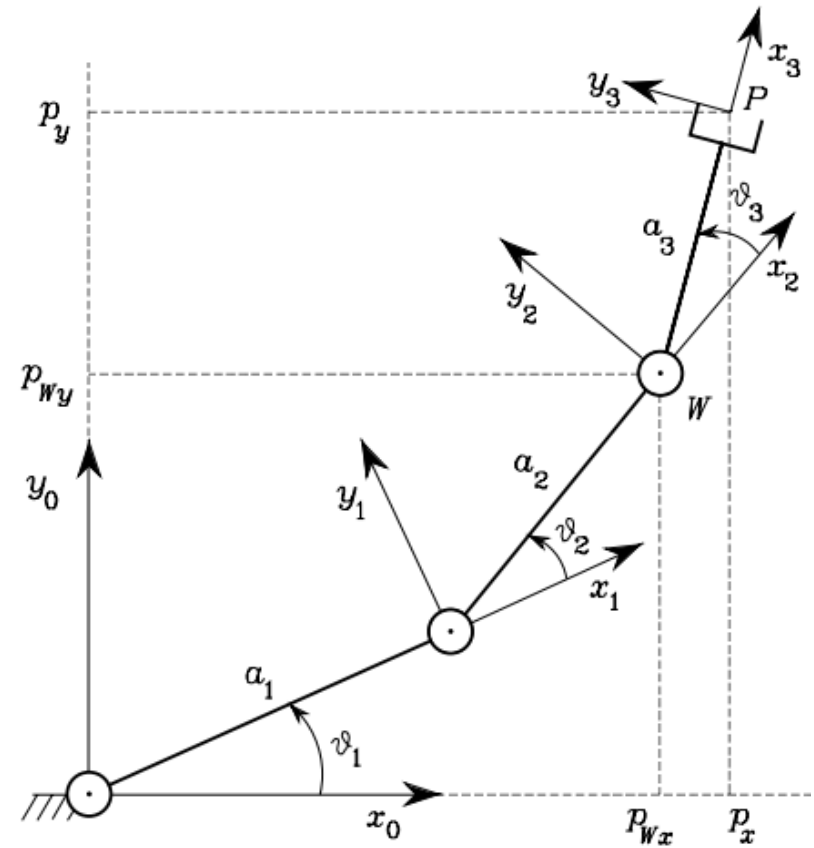
3.7 INVERSE KINEMATICS ALGORITHMS

□ Comparison Among Inverse Kinematics Algorithms

- ❖ 3-link planar arm
- ❖ $a_1 = a_2 = a_3 = 0.5 \text{ m}$
- ❖ Initial posture:

$$\mathbf{q} = [\underline{\pi} \quad -\pi/2 \quad -\pi/2]^T \text{ rad}$$

$$\mathbf{p} = [0 \quad 0.5]^T \text{ m}$$



3.7 INVERSE KINEMATICS ALGORITHMS

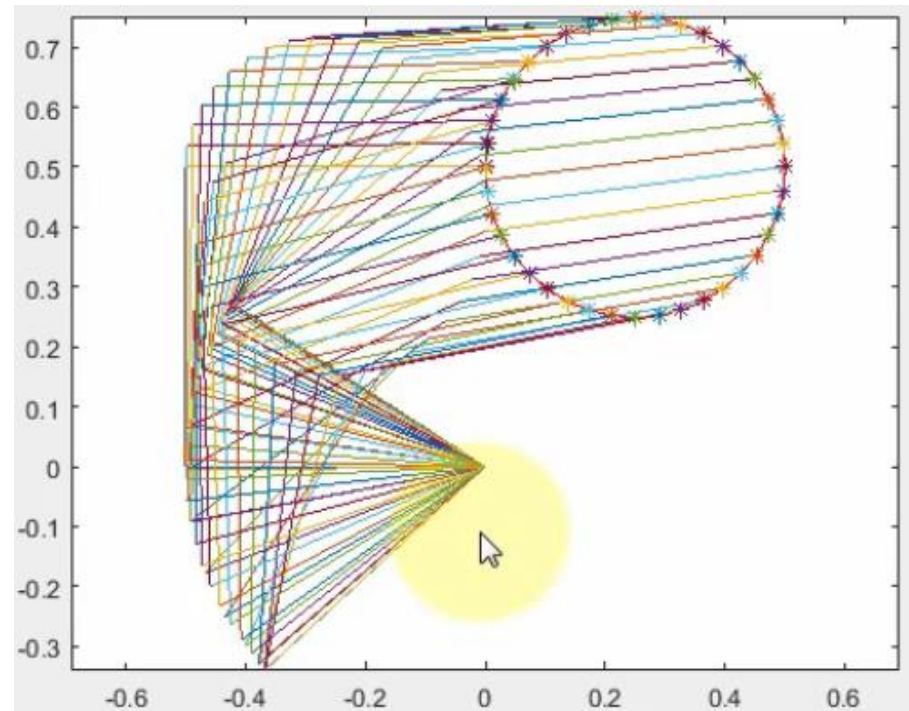
□ Comparison Among Inverse Kinematics Algorithms

❖ Motion trajectory:

$$\mathbf{p}_d(t) = \begin{bmatrix} 0.25(1 - \cos \pi t) \\ 0.25(2 + \sin \pi t) \end{bmatrix}$$

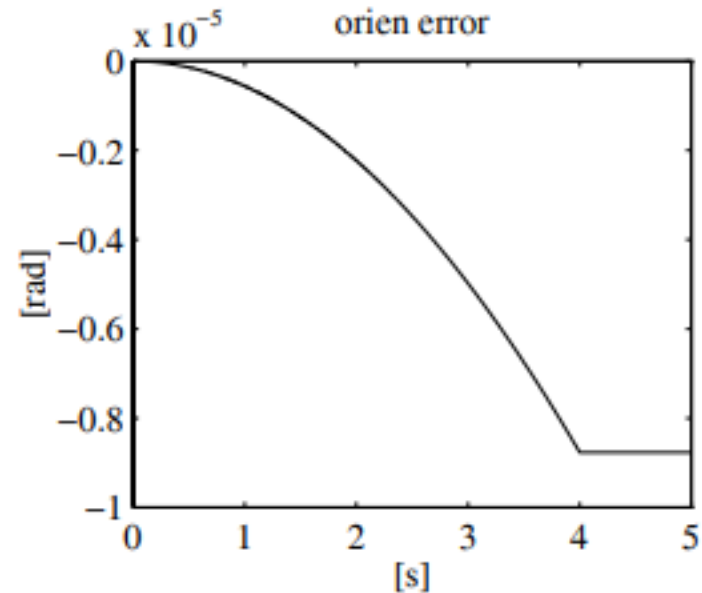
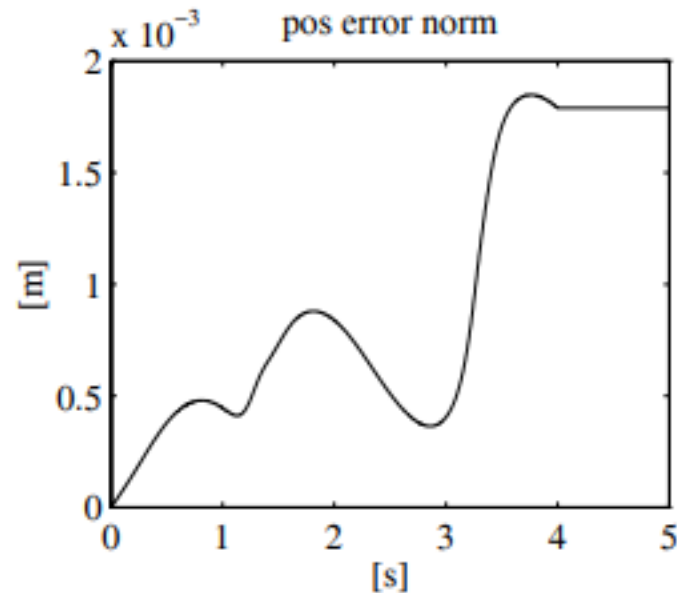
$$0 \leq t \leq 4;$$

$$\phi_d(t) = \sin \frac{\pi}{24} t \quad 0 \leq t \leq 4;$$



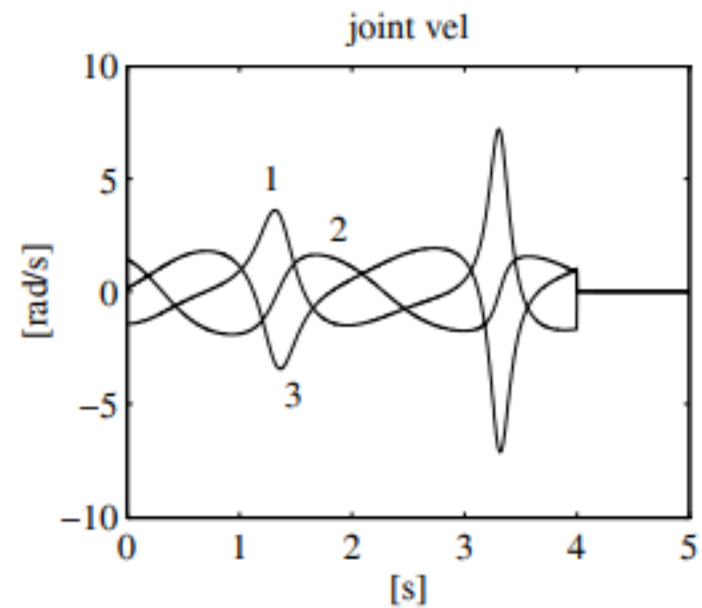
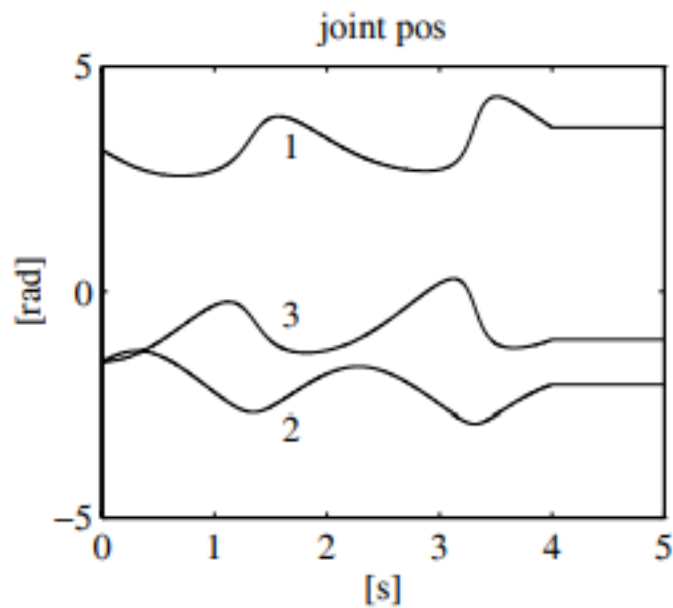
3.7 INVERSE KINEMATICS ALGORITHMS

- Open-loop inverse Jacobian algorithm



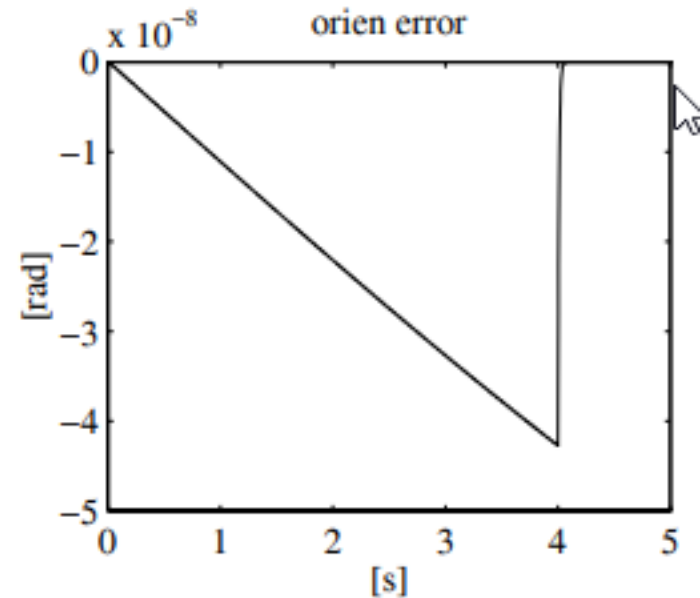
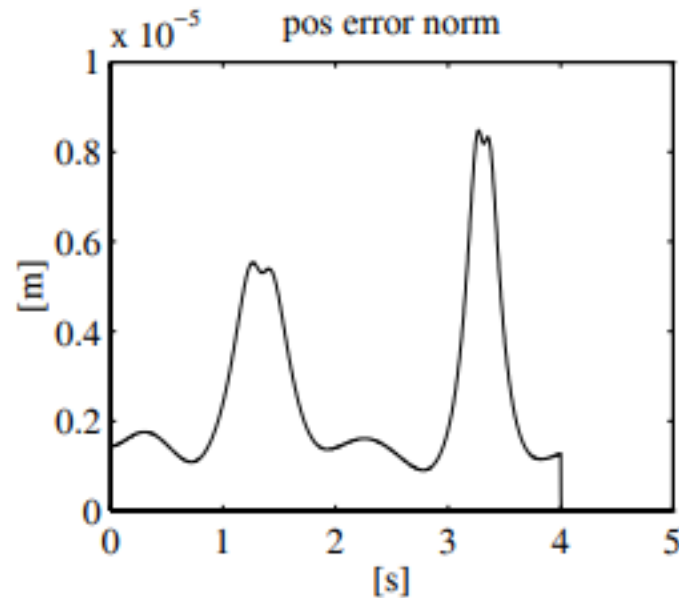
3.7 INVERSE KINEMATICS ALGORITHMS

- Closed-loop inverse Jacobian algorithm



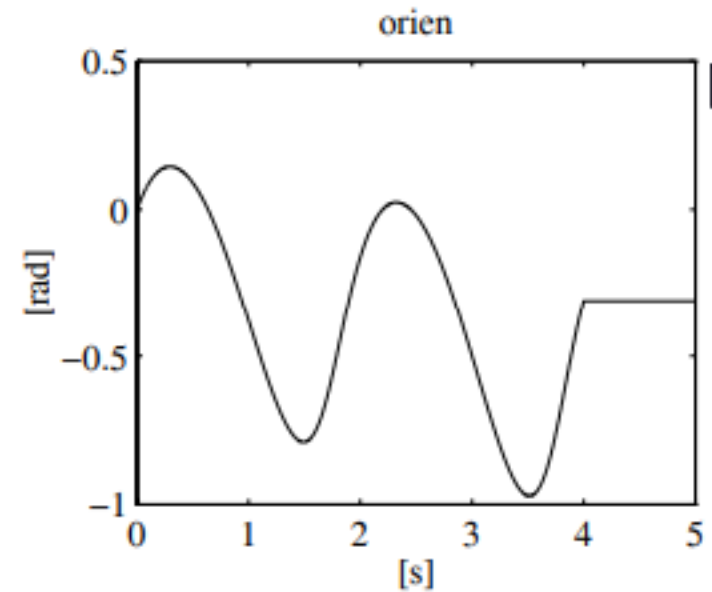
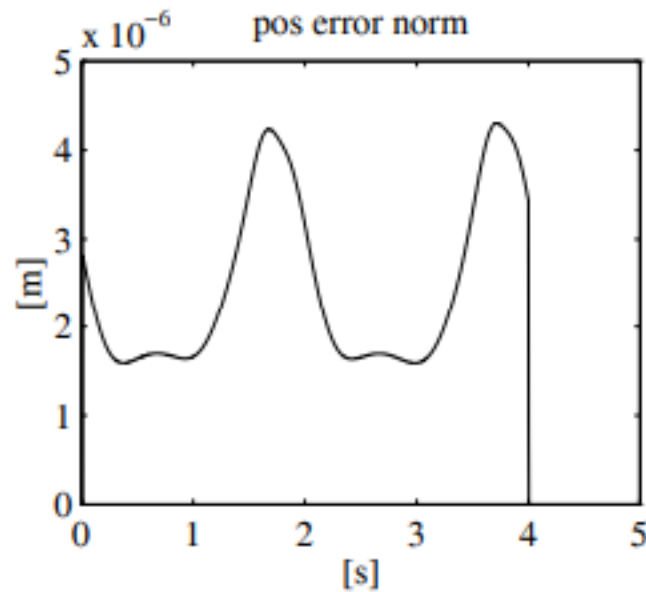
3.7 INVERSE KINEMATICS ALGORITHMS

- Closed-loop inverse Jacobian algorithm



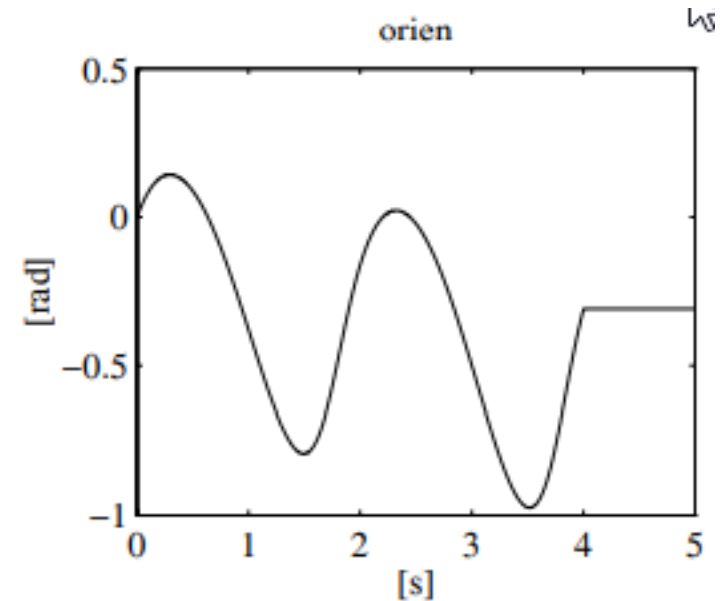
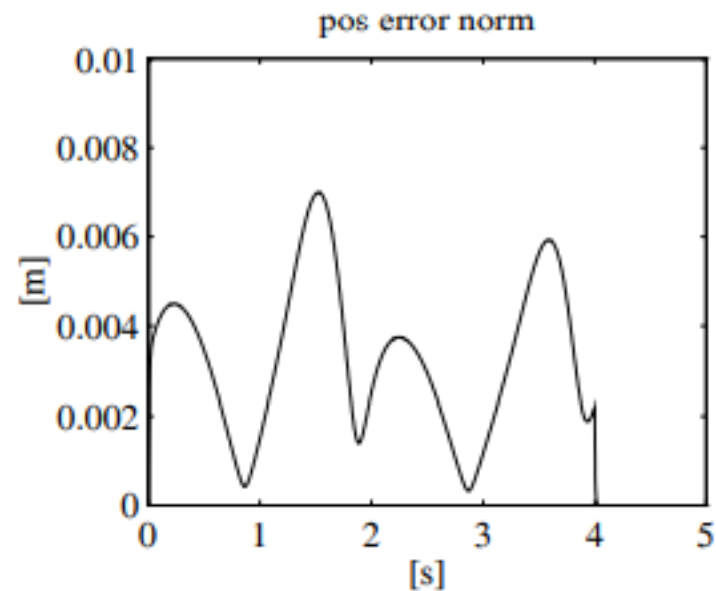
3.7 INVERSE KINEMATICS ALGORITHMS

- Jacobian pseudo-inverse algorithm



3.7 INVERSE KINEMATICS ALGORITHMS

□ Jacobian transpose algorithm



3.8 STATICS

- The goal of statics is to determine the relationship between the generalized forces applied to the end-effector and the generalized forces applied to the joints (forces for prismatic joints, torques for revolute joints) with the manipulator at an equilibrium configuration.
- The application of the *principle of virtual work*:

- ❖ *Joint Torques*

$$\rightarrow dW_\tau = \boldsymbol{\tau}^T d\mathbf{q}$$

- ❖ End-effector forces

$$\rightarrow dW_\gamma = \mathbf{f}_e^T d\mathbf{p}_e + \boldsymbol{\mu}_e^T \boldsymbol{\omega}_e dt$$

3.8 STATICS

- By accounting for the differential kinematics relationship:

$$dW_\gamma = \mathbf{f}_e^T d\mathbf{p}_e + \boldsymbol{\mu}_e^T \boldsymbol{\omega}_e dt$$

$$\boldsymbol{\gamma}_e = [\mathbf{f}_e^T \quad \boldsymbol{\mu}_e^T]^T \quad \rightarrow \quad dW_\gamma = \mathbf{f}_e^T \mathbf{J}_P(\mathbf{q})d\mathbf{q} + \boldsymbol{\mu}_e^T \mathbf{J}_O(\mathbf{q})d\mathbf{q} \\ = \boldsymbol{\gamma}_e^T \mathbf{J}(\mathbf{q})d\mathbf{q}$$

$$\rightarrow \quad \begin{aligned} \delta W_\tau &= \boldsymbol{\tau}^T \delta \mathbf{q} \\ \delta W_\gamma &= \boldsymbol{\gamma}_e^T \mathbf{J}(\mathbf{q})\delta \mathbf{q} \end{aligned}$$

- ❖ The manipulator is at static equilibrium if and only if:

$$\delta W_\tau = \delta W_\gamma \quad \forall \delta \mathbf{q} \quad \rightarrow \quad \boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\boldsymbol{\gamma}_e$$

3.9 MANIPULABILITY ELLIPSOIDS

□ Definition of indices for the evaluation of manipulator performance:

❖ The *velocity manipulability ellipsoid*

✓ Set of joint velocities of constant (unit) norm $\dot{\mathbf{q}}^T \dot{\mathbf{q}} = 1$

✓ Capability of manipulator to change end-effector position and orientation

