Introduction to Robotics Chapter VI Jacobian Matrix and Motion Planning

By Dr. Song LIU

AMNR Lab Director

liusong@shanghaitech.edu.cn

Jacobian Matrix and Motion Planning

- Content
  - Robot Jacobian matrix
  - Case Study
    - Jacobian Matrix of V-80 Robot
    - Jacobian Matrix of PUMA 560
  - Static Force Transformation
  - Robot Motion Planning
    - Motion Planning in Joint Space
    - Motion Planning in End Cartesian Space

Velocity Control?

- With robot kinematics, we can determine the end-effector's pose by joint angles.
- With inverse kinematics, we can get the joint angles which enables a robot to reach the expected pose.

- The next important question is:

  how to control a robot's velocity?

- The answer is :

  differential kinematics and the corresponding Jacobian matrix.

Robot Jacobian Matrix

- Jacobian Matrix： the robot Jacobian matrix is defined as the transformation matrix from motion velocity in joint space to motion velocity in end-effector’s Cartesian coordinates space.
- Define *x* as the generalized position vector of robot end-effector

q is the robot joint vector, and there are *n* joints (q is *n*-dimensional)

$$x = x(q) \Rightarrow \dot{x} = \left[\sum_{j=1}^{n} \frac{\partial x_1}{\partial q_j} \dot{q}_j \quad \cdots \quad \sum_{j=1}^{n} \frac{\partial x_6}{\partial q_j} \dot{q}_j \right]^T = J(q)\dot{q}$$

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & \vdots & J_{1n} \\ J_{21} & J_{22} & \vdots & J_{2n} \\ J_{31} & J_{32} & \vdots & J_{3n} \\ J_{41} & J_{42} & \vdots & J_{4n} \\ J_{51} & J_{52} & \vdots & J_{5n} \\ J_{61} & J_{62} & \vdots & J_{6n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \vdots \\ \dot{q}_{n-1} \\ \dot{q}_n \end{bmatrix}$$

Robot Jacobian Matrix

- Jacobian Matrix： the robot Jacobian matrix is defined as the transformation matrix from motion velocity in joint space to motion velocity in end-effector's Cartesian coordinates space.
- Define *x* as the generalized position vector of robot end-effector

q is the robot joint vector, and there are *n* joints (q is *n*-dimensional)

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

$J(q)$ size is $6 \times n$

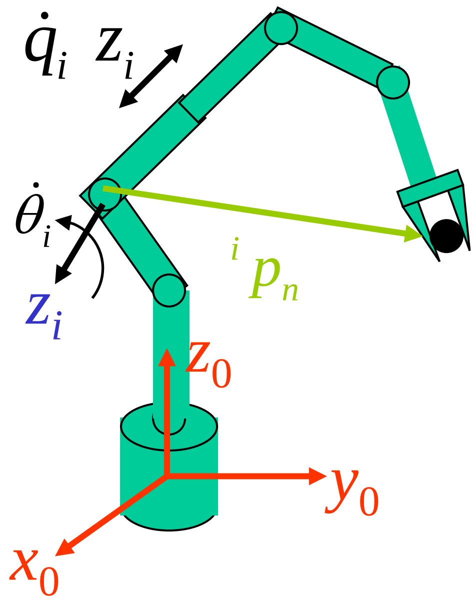
$$J_{ij}(q) = \frac{\partial x_i}{\partial q_j}$$

$$\dot{x} = \begin{bmatrix} v \\ w \end{bmatrix} = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \begin{bmatrix} d \\ \delta \end{bmatrix} \Rightarrow D = \begin{bmatrix} d \\ \delta \end{bmatrix} = \lim_{\Delta t \to 0} \dot{x} \Delta t$$

$$D = \lim_{\Delta t \to 0} J(q)\dot{q}\Delta t = J(q)dq$$

Robot Jacobian Matrix

➤ Calculation of Jacobian Matrix from Differential Kinematics



For joint i, supposing the transformation matrix towards end-effector's coordinates ${}^iT_n$ is known in advance.

$$\begin{bmatrix} {}^Td_x \\ {}^Td_y \\ {}^Td_z \\ {}^T\delta_x \\ {}^T\delta_y \\ {}^T\delta_z \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z & (p\times n)_x & (p\times n)_y & \boxed{(p\times n)_z} \\ o_x & o_y & o_z & (p\times o)_x & (p\times o)_y & (p\times o)_z \\ a_x & a_y & a_z & (p\times a)_x & (p\times a)_z & (p\times a)_z \\ 0 & 0 & 0 & n_x & n_y & n_z \\ 0 & 0 & 0 & o_x & o_y & o_z \\ 0 & 0 & 0 & a_x & a_y & a_z \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \\ \delta_x \\ \delta_y \\ \delta_z \end{bmatrix}$$

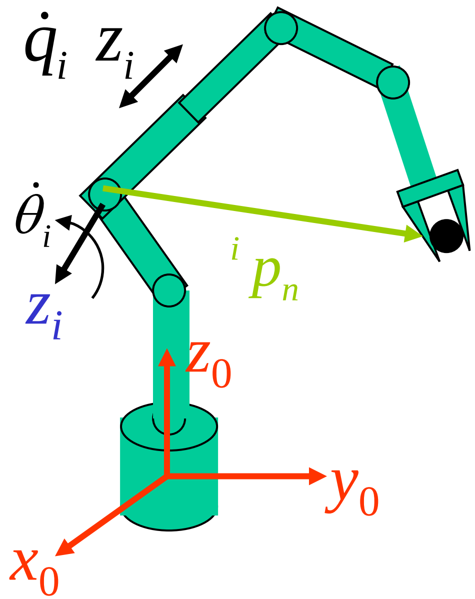
If i is rotatory joint： $d = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \delta = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathrm{d}\theta_i$

$$\begin{bmatrix} {}^Td_x \\ {}^Td_y \\ {}^Td_z \\ {}^T\delta_x \\ {}^T\delta_y \\ {}^T\delta_z \end{bmatrix} = \begin{bmatrix} (p\times n)_z \\ (p\times o)_z \\ (p\times a)_z \\ n_z \\ o_z \\ a_z \end{bmatrix} \mathrm{d}\theta_i$$

Then: $J_i = \begin{bmatrix} (p\times n)_z \\ (p\times o)_z \\ (p\times a)_z \\ n_z \\ o_z \\ a_z \end{bmatrix}$

Robot Jacobian Matrix

➢ Calculation of Jacobian Matrix from Differential Kinematics



$$\begin{bmatrix} {}^{T}d_x \\ {}^{T}d_y \\ {}^{T}d_z \\ {}^{T}\delta_x \\ {}^{T}\delta_y \\ {}^{T}\delta_z \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z & (p\times n)_x & (p\times n)_y & (p\times n)_z \\ o_x & o_y & o_z & (p\times o)_x & (p\times o)_y & (p\times o)_z \\ a_x & a_y & a_z & (p\times a)_x & (p\times a)_z & (p\times a)_z \\ 0 & 0 & 0 & n_x & n_y & n_z \\ 0 & 0 & 0 & o_x & o_y & o_z \\ 0 & 0 & 0 & a_x & a_y & a_z \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \\ \delta_x \\ \delta_y \\ \delta_z \end{bmatrix}$$

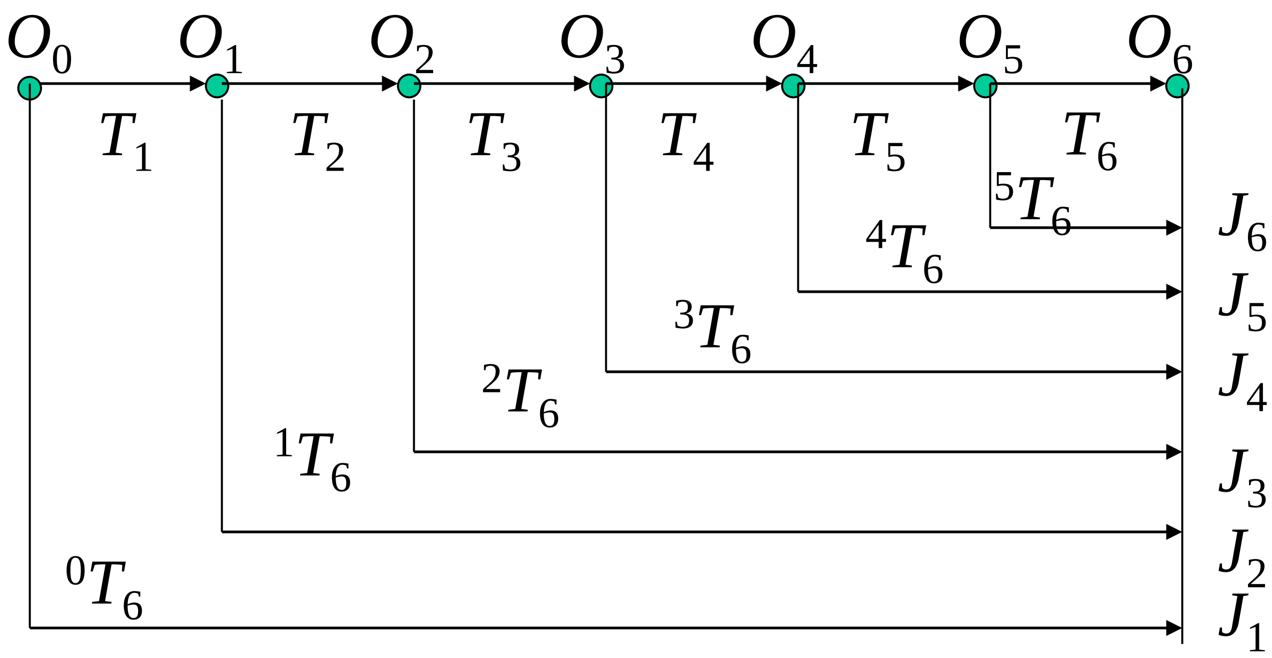
If i is translational joint： $d = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathrm{d}d_i, \delta = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

For joint/link i, supposing the transformation matrix towards end-effector's coordinates ${}^{i}T_n$ is known in advance.

$$\begin{bmatrix} {}^{T}d_x \\ {}^{T}d_y \\ {}^{T}d_z \\ {}^{T}\delta_x \\ {}^{T}\delta_y \\ {}^{T}\delta_z \end{bmatrix} = \begin{bmatrix} n_z \\ o_z \\ a_z \\ 0 \\ 0 \\ 0 \end{bmatrix} \mathrm{d}d_i, \qquad \text{Then: } J_i = \begin{bmatrix} n_z \\ o_z \\ a_z \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Robot Jacobian Matrix

- Calculation of Jacobian Matrix from Differential Kinematics
  - ✓ Calculate Link Transformation Matrices: ${}^{0}T_1, {}^{1}T_2, \ldots, {}^{n-1}T_n$
  - ✓ Calculate Transformation Matrices from Link n towards End-Effector's Coordinates ${}^{n-1}T_n, {}^{n-2}T_n, \ldots, {}^{0}T_n$
  - ✓ Calculate $J(q)$ column by column. Depending on rotatory joint or translational joint, J_i is derived from ${}^{i}T_n$.



Robot Jacobian Matrix: Case Study of V80 Robot

V80 Robot

Link	θ	α	a	d
1	θ_1	-90°	0	0
2	θ_2	0°	a_2	0
3	θ_3	90°	a_3	0
4	θ_4	-90°	0	d_4
5	θ_5	90°	0	0
6	θ_6	0°	0	0

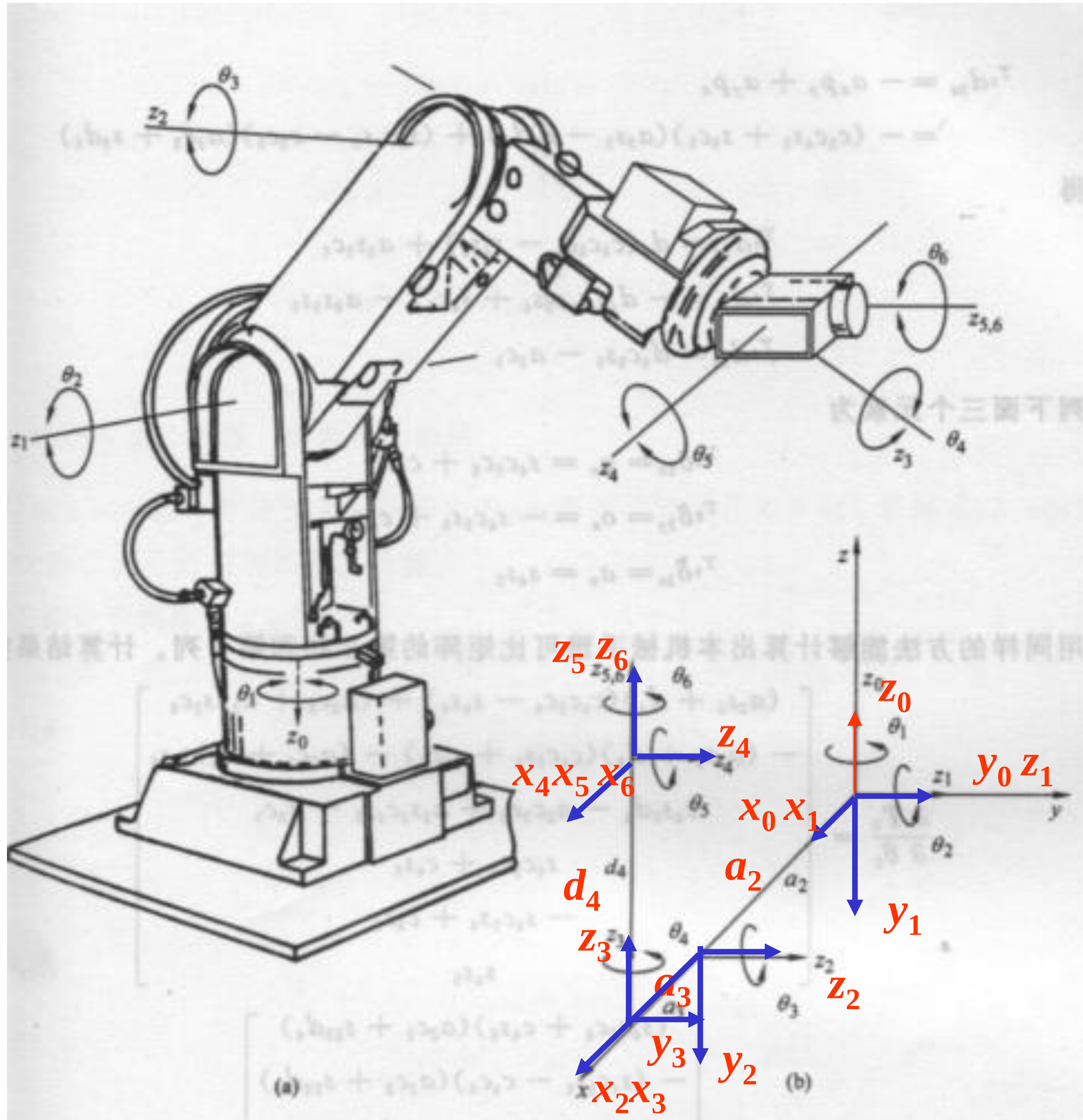


图 3.15 V-80 机械手外形图及停止位置

Robot Jacobian Matrix: Case Study of V80 Robot

- Jacobian Matrix of V-80 Robot
  - ✓ Link Transformation Matrix

$$T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1 = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_3 = \begin{bmatrix} \cos\theta_3 & 0 & \sin\theta_3 & a_3\cos\theta_3 \\ \sin\theta_3 & 0 & -\cos\theta_3 & a_3\sin\theta_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} \cos\theta_4 & 0 & -\sin\theta_4 & 0 \\ \sin\theta_4 & 0 & \cos\theta_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_5 = \begin{bmatrix} \cos\theta_5 & 0 & \sin\theta_5 & 0 \\ \sin\theta_5 & 0 & -\cos\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_6 = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Robot Jacobian Matrix: Case Study of V80 Robot

✓ Transformation from link *i* towards end-effector's coordinates

$$^{5}T_6 = T_6 = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{4}T_6 = T_5 T_6 = \begin{bmatrix} \cos\theta_5 & 0 & \sin\theta_5 & 0 \\ \sin\theta_5 & 0 & -\cos\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_5\cos\theta_6 & -\cos\theta_5\sin\theta_6 & \sin\theta_5 & 0 \\ \sin\theta_5\cos\theta_6 & -\sin\theta_5\sin\theta_6 & -\cos\theta_5 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{3}T_6 = T_4\,{}^{4}T_6 = \begin{bmatrix} \cos\theta_4 & 0 & -\sin\theta_4 & 0 \\ \sin\theta_4 & 0 & \cos\theta_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_5\cos\theta_6 & -\cos\theta_5\sin\theta_6 & \sin\theta_5 & 0 \\ \sin\theta_5\cos\theta_6 & -\sin\theta_5\sin\theta_6 & -\cos\theta_5 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6 & -\cos\theta_4\cos\theta_5\sin\theta_6 - \sin\theta_4\cos\theta_6 & \cos\theta_4\sin\theta_5 & 0 \\ \sin\theta_4\cos\theta_5\cos\theta_6 + \cos\theta_4\sin\theta_6 & -\sin\theta_4\cos\theta_5\sin\theta_6 + \cos\theta_4\cos\theta_6 & -\sin\theta_4\cos\theta_5 & 0 \\ -\sin\theta_5\cos\theta_6 & \sin\theta_5\sin\theta_6 & \cos\theta_5 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Robot Jacobian Matrix: Case Study of V80 Robot

✓Jacobian Matrix

$$d = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \delta = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathrm{d}\theta_i$$

$$J_i = \begin{bmatrix} (p \times n)_z \\ (p \times o)_z \\ (p \times a)_z \\ n_z \\ o_z \\ a_z \end{bmatrix} \quad J_6 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad J_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sin\theta_6 \\ \cos\theta_6 \\ 0 \end{bmatrix} \quad p_4 = \begin{bmatrix} 0 \\ 0 \\ d_4 \end{bmatrix}, \begin{bmatrix} (p_4 \times n)_z \\ (p_4 \times o)_z \\ (p_4 \times a)_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow J_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\sin\theta_5 \cos\theta_6 \\ \sin\theta_5 \sin\theta_6 \\ \cos\theta_5 \end{bmatrix}$$

$$J_3 = \begin{bmatrix} d_4(\cos\theta_4 \cos\theta_5 \cos\theta_6 - \sin\theta_4 \sin\theta_6) + a_3 \sin\theta_5 \cos\theta_6 \\ -d_4(\cos\theta_4 \cos\theta_5 \sin\theta_6 + \sin\theta_4 \cos\theta_6) - a_3 \sin\theta_5 \cos\theta_6 \\ d_4 \cos\theta_4 \sin\theta_5 - a_3 \cos\theta_5 \\ \sin\theta_4 \cos\theta_5 \cos\theta_6 + \cos\theta_4 \sin\theta_6 \\ -(\sin\theta_4 \cos\theta_5 \sin\theta_6 - \cos\theta_4 \cos\theta_6) \\ \sin\theta_4 \sin\theta_5 \end{bmatrix}$$

Robot Jacobian Matrix: Case Study of V80 Robot

✓Jacobian Matrix

$$J_2 = \begin{bmatrix} (a_2 \sin\theta_3 + d_4)(\cos\theta_4 \cos\theta_5 \cos\theta_6 - \sin\theta_4 \sin\theta_6) + (a_2 \cos\theta_3 + a_3)\sin\theta_5 \cos\theta_6 \\ -(a_2 \sin\theta_3 + d_4)(\cos\theta_4 \cos\theta_5 \sin\theta_6 + \sin\theta_4 \cos\theta_6) - (a_2 \cos\theta_3 + a_3)\sin\theta_5 \cos\theta_6 \\ d_4 \cos\theta_4 \sin\theta_5 - a_2 \cos\theta_3 \cos\theta_5 + a_2 \sin\theta_3 \cos\theta_4 \sin\theta_5 - a_3 \cos\theta_5 \\ \sin\theta_4 \cos\theta_5 \cos\theta_6 + \cos\theta_4 \sin\theta_6 \\ -(\sin\theta_4 \cos\theta_5 \sin\theta_6 - \cos\theta_4 \cos\theta_6) \\ \sin\theta_4 \sin\theta_5 \end{bmatrix}$$

$$J_1 = \begin{bmatrix} (\sin\theta_4 \cos\theta_5 \cos\theta_6 + \cos\theta_4 \sin\theta_6)[a_2 \cos\theta_2 + d_4 \sin(\theta_2 + \theta_3)] \\ -(\sin\theta_4 \cos\theta_5 \sin\theta_6 - \cos\theta_4 \cos\theta_6)[a_2 \cos\theta_2 + d_4 \sin(\theta_2 + \theta_3)] \\ \sin\theta_4 \sin\theta_5 [a_2 \cos\theta_2 + d_4 \sin(\theta_2 + \theta_3)] \\ -\sin(\theta_2 + \theta_3)(\cos\theta_4 \cos\theta_5 \cos\theta_6 - \sin\theta_4 \sin\theta_6) + \cos(\theta_2 + \theta_3)\sin\theta_5 \sin\theta_6 \\ \sin(\theta_2 + \theta_3)(\cos\theta_4 \cos\theta_5 \sin\theta_6 + \sin\theta_4 \cos\theta_6) - \cos(\theta_2 + \theta_3)\sin\theta_5 \cos\theta_6 \\ -\sin(\theta_2 + \theta_3)\cos\theta_4 \sin\theta_5 + \cos(\theta_2 + \theta_3)\cos\theta_5 \end{bmatrix}$$

Robot Jacobian Matrix: Case Study of PUMA 560

- Jacobian Matrix of PUMA 560

$$
{}^5T_6 = T_6 = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
J_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T
$$

$$
{}^4T_6 = T_5 T_6 = \begin{bmatrix} \cos\theta_5\cos\theta_6 & -\cos\theta_5\sin\theta_6 & -\sin\theta_5 & 0 \\ \sin\theta_5\cos\theta_6 & -\sin\theta_5\sin\theta_6 & \cos\theta_5 & 0 \\ -\sin\theta_6 & -\cos\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
J_5 = \begin{bmatrix} 0 & 0 & 0 & -\sin\theta_6 & -\cos\theta_6 & 0 \end{bmatrix}^T
$$

$$
{}^3T_6 = T_4 T_5 T_6
$$

$$
= \begin{bmatrix} \cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6 & -\cos\theta_4\cos\theta_5\sin\theta_6 - \sin\theta_4\cos\theta_6 & -\cos\theta_4\sin\theta_5 & 0 \\ \sin\theta_4\cos\theta_5\cos\theta_6 + \cos\theta_4\sin\theta_6 & -\sin\theta_4\cos\theta_5\sin\theta_6 + \cos\theta_4\cos\theta_6 & -\sin\theta_4\sin\theta_5 & 0 \\ \sin\theta_5\cos\theta_6 & -\sin\theta_5\sin\theta_6 & \cos\theta_5 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
J_4 = \begin{bmatrix} 0 & 0 & 0 & \sin\theta_5\cos\theta_6 & -\sin\theta_5\sin\theta_6 & \cos\theta_5 \end{bmatrix}^T
$$

Robot Jacobian Matrix: Case Study of PUMA 560

$$^2T_6 = T_3 T_4 T_5 T_6$$

$$= \begin{bmatrix} \cos\theta_3 & 0 & -\sin\theta_3 & a_3\cos\theta_3 \\ \sin\theta_3 & 0 & \cos\theta_3 & a_3\sin\theta_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6 & -\cos\theta_4\cos\theta_5\sin\theta_6 - \sin\theta_4\cos\theta_6 & -\cos\theta_4\sin\theta_5 & 0 \\ \sin\theta_4\cos\theta_5\cos\theta_6 + \cos\theta_4\sin\theta_6 & -\sin\theta_4\cos\theta_5\sin\theta_6 + \cos\theta_4\cos\theta_6 & -\sin\theta_4\sin\theta_5 & 0 \\ \sin\theta_5\cos\theta_6 & -\sin\theta_5\sin\theta_6 & \cos\theta_5 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \left[\begin{array}{c:c:c:c} \begin{matrix}\cos\theta_3(\cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6) \\ -\sin\theta_3\sin\theta_5\cos\theta_6\end{matrix} & \begin{matrix}-\cos\theta_3(\cos\theta_4\cos\theta_5\sin\theta_6 + \sin\theta_4\cos\theta_6) \\ +\sin\theta_3\sin\theta_5\sin\theta_6\end{matrix} & \begin{matrix}-\cos\theta_3\cos\theta_4\sin\theta_5 \\ -\sin\theta_3\cos\theta_5\end{matrix} & \begin{matrix}-d_4\sin\theta_3 \\ +a_3\cos\theta_3\end{matrix} \\ \hdashline \begin{matrix}\sin\theta_3(\cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6) \\ +\cos\theta_3\sin\theta_5\cos\theta_6\end{matrix} & \begin{matrix}-\sin\theta_3(\cos\theta_4\cos\theta_5\sin\theta_6 + \sin\theta_4\cos\theta_6) \\ -\cos\theta_3\sin\theta_5\sin\theta_6\end{matrix} & \begin{matrix}-\sin\theta_3\cos\theta_4\sin\theta_5 \\ +\cos\theta_3\cos\theta_5\end{matrix} & \begin{matrix}d_4\cos\theta_3 \\ +a_3\sin\theta_3\end{matrix} \\ \hdashline -\sin\theta_4\cos\theta_5\cos\theta_6 - \cos\theta_4\sin\theta_6 & \sin\theta_4\cos\theta_5\sin\theta_6 - \cos\theta_4\cos\theta_6 & \sin\theta_4\sin\theta_5 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

$$\begin{cases} J_{13} = (-d_4\sin\theta_3 + a_3\cos\theta_3)[\sin\theta_3(\cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6) + \cos\theta_3\sin\theta_5\cos\theta_6] \\ \qquad -(d_4\cos\theta_3 + a_3\sin\theta_3)[\cos\theta_3(\cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6) - \sin\theta_3\sin\theta_5\cos\theta_6] \\ J_{23} = (-d_4\sin\theta_3 + a_3\cos\theta_3)[-\sin\theta_3(\cos\theta_4\cos\theta_5\sin\theta_6 + \sin\theta_4\cos\theta_6) - \cos\theta_3\sin\theta_5\sin\theta_6] \\ \qquad -(d_4\cos\theta_3 + a_3\sin\theta_3)[-\cos\theta_3(\cos\theta_4\cos\theta_5\sin\theta_6 + \sin\theta_4\cos\theta_6) + \sin\theta_3\sin\theta_5\sin\theta_6] \\ J_{33} = (-d_4\sin\theta_3 + a_3\cos\theta_3)[-\sin\theta_3\cos\theta_4\sin\theta_5 + \cos\theta_3\cos\theta_5] \\ \qquad -(d_4\cos\theta_3 + a_3\sin\theta_3)[-\cos\theta_3\cos\theta_4\sin\theta_5 - \sin\theta_3\cos\theta_5] \\ J_{43} = -\sin\theta_4\cos\theta_5\cos\theta_6 - \cos\theta_4\sin\theta_6 \\ J_{53} = \sin\theta_4\cos\theta_5\sin\theta_6 - \cos\theta_4\cos\theta_6 \\ J_{63} = \sin\theta_4\sin\theta_5 \end{cases}$$

Robot Jacobian Matrix: Case Study of PUMA 560

$$
{}^1T_6 = T_2 T_3 T_4 T_5 T_6 = \begin{bmatrix} b_{111} & b_{112} & b_{113} & -d_4 \sin(\theta_2+\theta_3) + a_2 \cos\theta_2 + a_3 \cos(\theta_2+\theta_3) \\ b_{121} & b_{122} & b_{123} & d_4 \cos(\theta_2+\theta_3) + a_2 \sin\theta_2 + a_3 \sin(\theta_2+\theta_3) \\ b_{131} & b_{132} & b_{133} & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$b_{111} = \cos(\theta_2+\theta_3)(\cos\theta_4 \cos\theta_5 \cos\theta_6 - \sin\theta_4 \sin\theta_6) - \sin(\theta_2+\theta_3)\sin\theta_5 \cos\theta_6$$

$$b_{121} = \sin(\theta_2+\theta_3)(\cos\theta_4 \cos\theta_5 \cos\theta_6 - \sin\theta_4 \sin\theta_6) + \cos(\theta_2+\theta_3)\sin\theta_5 \cos\theta_6$$

$$b_{131} = -\sin\theta_4 \cos\theta_5 \cos\theta_6 - \cos\theta_4 \sin\theta_6$$

$$b_{112} = -\cos(\theta_2+\theta_3)(\cos\theta_4 \cos\theta_5 \sin\theta_6 + \sin\theta_4 \cos\theta_6) + \sin(\theta_2+\theta_3)\sin\theta_5 \sin\theta_6$$

$$b_{122} = -\sin(\theta_2+\theta_3)(\cos\theta_4 \cos\theta_5 \sin\theta_6 + \sin\theta_4 \cos\theta_6) - \cos(\theta_2+\theta_3)\sin\theta_5 \sin\theta_6$$

$$b_{132} = \sin\theta_4 \cos\theta_5 \sin\theta_6 - \cos\theta_4 \cos\theta_6$$

$$b_{113} = -\cos(\theta_2+\theta_3)\cos\theta_4 \sin\theta_5 - \sin(\theta_2+\theta_3)\cos\theta_5$$

$$b_{123} = -\sin(\theta_2+\theta_3)\cos\theta_4 \sin\theta_5 + \cos(\theta_2+\theta_3)\cos\theta_5$$

$$b_{133} = \sin\theta_4 \sin\theta_5$$

Robot Jacobian Matrix: Case Study of PUMA 560

$$
\begin{cases}
J_{12} = [-d_4 \sin(\theta_2 + \theta_3) + a_2 \cos\theta_2 + a_3 \cos(\theta_2 + \theta_3)] b_{121} \\
\qquad -[d_4 \cos(\theta_2 + \theta_3) + a_2 \sin\theta_2 + a_3 \sin(\theta_2 + \theta_3)] b_{111} \\
J_{22} = [-d_4 \sin(\theta_2 + \theta_3) + a_2 \cos\theta_2 + a_3 \cos(\theta_2 + \theta_3)] b_{122} \\
\qquad -[d_4 \cos(\theta_2 + \theta_3) + a_2 \sin\theta_2 + a_3 \sin(\theta_2 + \theta_3)] b_{112} \\
J_{32} = [-d_4 \sin(\theta_2 + \theta_3) + a_2 \cos\theta_2 + a_3 \cos(\theta_2 + \theta_3)] b_{123} \\
\qquad -[d_4 \cos(\theta_2 + \theta_3) + a_2 \sin\theta_2 + a_3 \sin(\theta_2 + \theta_3)] b_{113} \\
J_{42} = b_{131} \\
J_{52} = b_{132} \\
J_{62} = b_{133}
\end{cases}
$$

Robot Jacobian Matrix: Case Study of PUMA 560

$$
{}^{0}T_6 = T_1 T_2 T_3 T_4 T_5 T_6
$$

$$
= \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_{111} & b_{112} & b_{113} & -d_4 \sin(\theta_2+\theta_3) + a_2\cos\theta_2 + a_3\cos(\theta_2+\theta_3) \\ b_{121} & b_{122} & b_{123} & d_4\cos(\theta_2+\theta_3) + a_2\sin\theta_2 + a_3\sin(\theta_2+\theta_3) \\ b_{131} & b_{132} & b_{133} & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

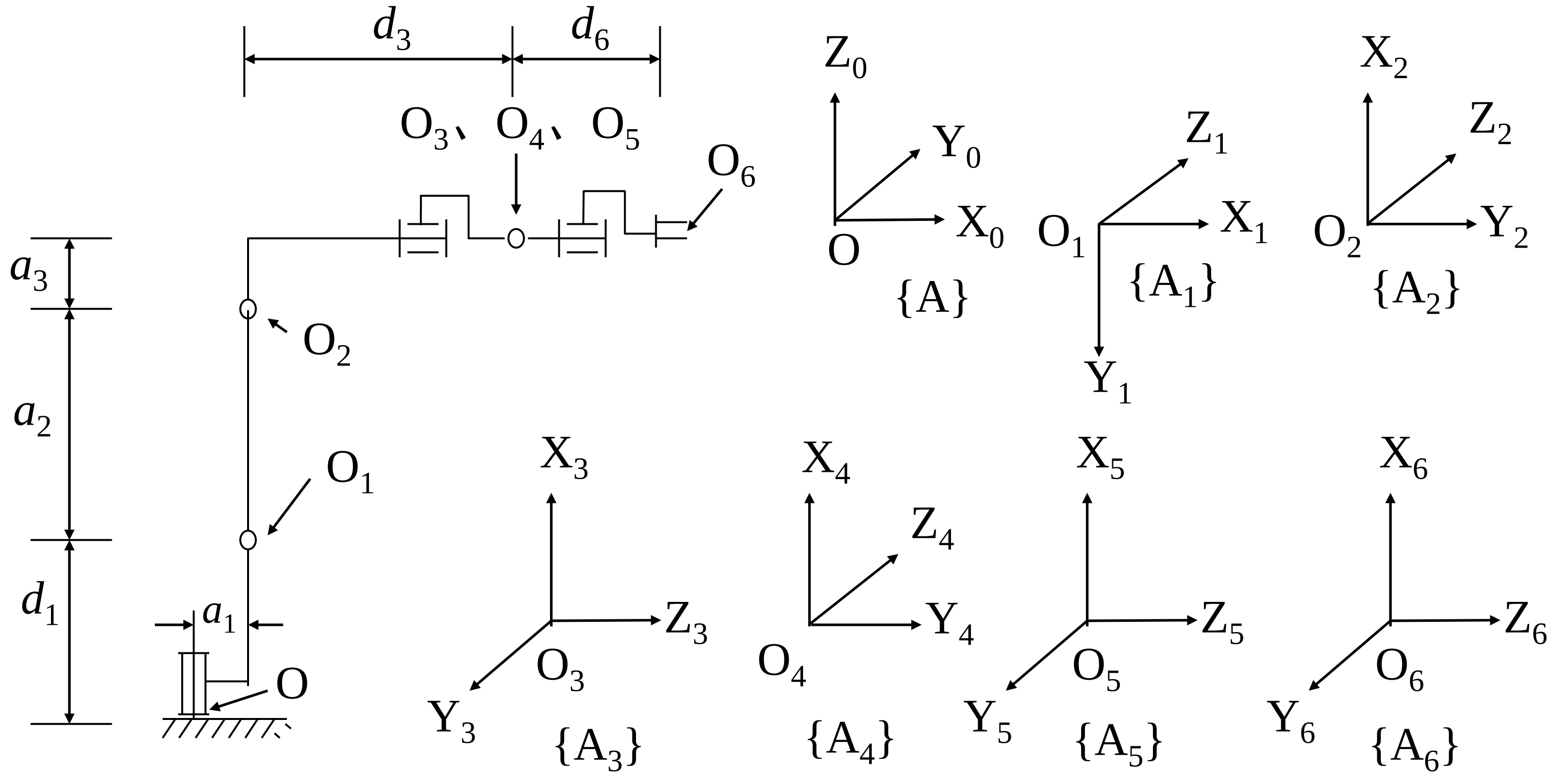
$$
= \begin{bmatrix} b_{111}\cos\theta_1 - b_{131}\sin\theta_1 & b_{112}\cos\theta_1 - b_{132}\sin\theta_1 & b_{113}\cos\theta_1 - b_{133}\sin\theta_1 & b_{114}\cos\theta_1 - d_2\sin\theta_1 \\ b_{111}\sin\theta_1 + b_{131}\cos\theta_1 & b_{112}\sin\theta_1 + b_{132}\cos\theta_1 & b_{113}\sin\theta_1 + b_{133}\cos\theta_1 & b_{114}\sin\theta_1 + d_2\cos\theta_1 \\ -b_{121} & -b_{122} & -b_{123} & -b_{124} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
\begin{cases} J_{11} = (b_{114}\cos\theta_1 - d_2\sin\theta_1)(b_{111}\sin\theta_1 + b_{131}\cos\theta_1) \\ \qquad -(b_{114}\sin\theta_1 + d_2\cos\theta_1)(b_{111}\cos\theta_1 - b_{131}\sin\theta_1) \\ J_{21} = (b_{114}\cos\theta_1 - d_2\sin\theta_1)(b_{112}\sin\theta_1 + b_{132}\cos\theta_1) \\ \qquad -(b_{114}\sin\theta_1 + d_2\cos\theta_1)(b_{112}\cos\theta_1 - b_{132}\sin\theta_1) \\ J_{31} = (b_{114}\cos\theta_1 - d_2\sin\theta_1)(b_{113}\sin\theta_1 + b_{133}\cos\theta_1) \\ \qquad -(b_{114}\sin\theta_1 + d_2\cos\theta_1)(b_{113}\cos\theta_1 - b_{133}\sin\theta_1) \\ J_{41} = -b_{121} \\ J_{51} = -b_{122} \\ J_{61} = -b_{123} \end{cases}
$$

$$
\begin{cases} b_{114} = -d_4\sin(\theta_2+\theta_3) + a_2\cos\theta_2 + a_3\cos(\theta_2+\theta_3) \\ b_{124} = d_4\cos(\theta_2+\theta_3) + a_2\sin\theta_2 + a_3\sin(\theta_2+\theta_3) \end{cases}
$$

HOMEWORK

- List the D-H Parameters of each link?
- Calculate the Jacobian Matrix of YASKAWA K10 Robot?



Static Force Transformation

- Static force and torque are unitedly represented by：

$$F = \begin{bmatrix} f_x & f_y & f_z & m_x & m_y & m_z \end{bmatrix}^T$$

Generalized force

static force， torque

- Static force and torque transformation between coordinates

In based coordinates, define virtual work:

$$\delta W = F^T D$$

$$F = \begin{bmatrix} f_x & f_y & f_z & m_x & m_y & m_z \end{bmatrix}^T$$

$$D = \begin{bmatrix} d_x & d_y & d_z & \delta_x & \delta_y & \delta_z \end{bmatrix}^T$$

In end-effector coordinates, define

Concept: ${}^{C}\boldsymbol{F}$ is the equivalent generalized force of $\boldsymbol{F}$ in {C}.

$$\delta W = F^T D = {}^{c}F^T \, {}^{c}D$$

$${}^{c}F = \begin{bmatrix} {}^{c}f_x & {}^{c}f_y & {}^{c}f_z & {}^{c}m_x & {}^{c}m_y & {}^{c}m_z \end{bmatrix}^T$$

$${}^{c}D = \begin{bmatrix} {}^{c}d_x & {}^{c}d_y & {}^{c}d_z & {}^{c}\delta_x & {}^{c}\delta_y & {}^{c}\delta_z \end{bmatrix}^T$$

Static Force Transformation

Since

$$\begin{bmatrix} {}^T d_x \\ {}^T d_y \\ {}^T d_z \\ {}^T \delta_x \\ {}^T \delta_y \\ {}^T \delta_z \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z & (p\times n)_x & (p\times n)_y & (p\times n)_z \\ o_x & o_y & o_z & (p\times o)_x & (p\times o)_y & (p\times o)_z \\ a_x & a_y & a_z & (p\times a)_x & (p\times a)_y & (p\times a)_z \\ 0 & 0 & 0 & n_x & n_y & n_z \\ 0 & 0 & 0 & o_x & o_y & o_z \\ 0 & 0 & 0 & a_x & a_y & a_z \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \\ \delta_x \\ \delta_y \\ \delta_z \end{bmatrix} \Rightarrow {}^c D = J_d D$$

$$F^T D = {}^c F^T\, {}^c D = {}^c F^T J_d D \Rightarrow F^T = {}^c F^T J_d \Rightarrow F = J_d^T\, {}^c F \Rightarrow {}^c F = (J_d^T)^{-1} F$$

$$\begin{bmatrix} {}^c f_x \\ {}^c f_y \\ {}^c f_z \\ {}^c m_x \\ {}^c m_y \\ {}^c m_z \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & 0 & 0 & 0 \\ n_y & o_y & a_y & 0 & 0 & 0 \\ n_z & o_z & a_z & 0 & 0 & 0 \\ (p\times n)_x & (p\times o)_x & (p\times a)_x & n_x & o_x & a_x \\ (p\times n)_y & (p\times o)_y & (p\times a)_y & n_y & o_y & a_y \\ (p\times n)_z & (p\times o)_z & (p\times a)_z & n_z & o_z & a_z \end{bmatrix}^{-1} \begin{bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{bmatrix}$$

Static Force Transformation

Further:

$$\begin{bmatrix} {}^c f_x \\ {}^c f_y \\ {}^c f_z \\ {}^c m_x \\ {}^c m_y \\ {}^c m_z \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z & 0 & 0 & 0 \\ o_x & o_y & o_z & 0 & 0 & 0 \\ a_x & a_y & a_z & 0 & 0 & 0 \\ (p\times n)_x & (p\times n)_y & (p\times n)_z & n_x & n_y & n_z \\ (p\times o)_x & (p\times o)_y & (p\times o)_z & o_x & o_y & o_z \\ (p\times a)_x & (p\times a)_y & (p\times a)_z & a_x & a_y & a_z \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{bmatrix}$$

Finally:

$$\begin{bmatrix} {}^c m_x \\ {}^c m_y \\ {}^c m_z \\ {}^c f_x \\ {}^c f_y \\ {}^c f_z \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z & (p\times n)_x & (p\times n)_y & (p\times n)_z \\ o_x & o_y & o_z & (p\times o)_x & (p\times o)_y & (p\times o)_z \\ a_x & a_y & a_z & (p\times a)_x & (p\times a)_y & (p\times a)_z \\ 0 & 0 & 0 & n_x & n_y & n_z \\ 0 & 0 & 0 & o_x & o_y & o_z \\ 0 & 0 & 0 & a_x & a_y & a_z \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \\ f_x \\ f_y \\ f_z \end{bmatrix} \Rightarrow \begin{bmatrix} {}^c m_x \\ {}^c m_y \\ {}^c m_z \\ {}^c f_x \\ {}^c f_y \\ {}^c f_z \end{bmatrix} = J_d \begin{bmatrix} m_x \\ m_y \\ m_z \\ f_x \\ f_y \\ f_z \end{bmatrix}$$

Steady State load

Steady state load concerns with determination the joint force and torque to support an end-effector manipulating some object with some mass.

Determination of Joint Torque and Force

$$\left.\begin{aligned} \delta W &= {}^{T_6}F^T\ {}^{T_6}D = \tau^T Q \\ {}^{T_6}D &= JQ \end{aligned}\right\} \Rightarrow \quad \tau^T = {}^{T_6}F^T\ J, \quad \underline{\tau = J^T\ {}^{T_6}F}$$

Q: Joint Vector Increments

${}^{T_6}D$ is end-effector's motion increments

τ is joint force (for translation joint) and torque (for rotation joint) vector

$$\boxed{\dot{q} = J^{-1}\dot{x} \qquad \tau = J^T\ {}^{T_6}F}$$

Robot Motion Planning: Trajectory Planning

1 Joint Trajectory Planning

Given the initial and target joint vectors,

Calculate the middle joint vectors

along the path by interpolation

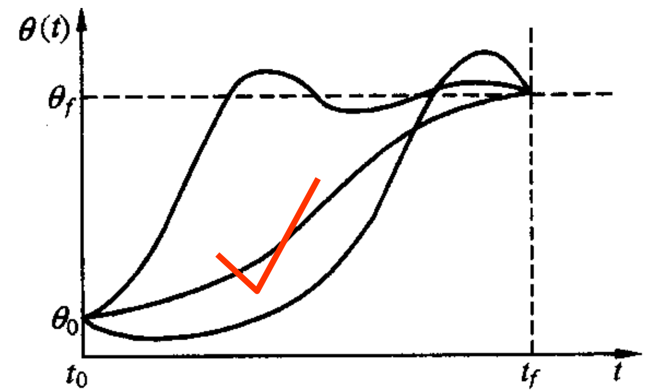


图 7.33 单个关节的不同轨迹曲线

- 1.1 Cubic Polynomial Interpolation Method

Boundary Conditions: $\begin{cases} \theta(0)=\theta_0 & \theta(t_f)=\theta_f \\ \dot{\theta}(0)=0 & \dot{\theta}(t_f)=0 \end{cases}$

Define: $\theta(t)=a_0+a_1t+a_2t^2+a_3t^3 \Rightarrow \dot{\theta}(t)=a_1+2a_2t+3a_3t^2$

We have:

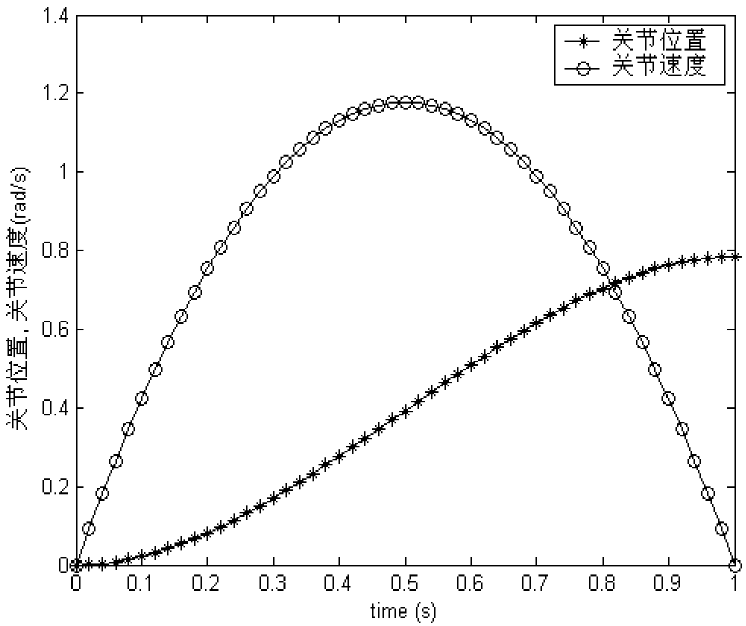
$$\begin{cases} a_0=\theta_0 \\ a_0+a_1t_f+a_2t_f{}^2+a_3t_f{}^3=\theta_f \\ a_1=0 \\ a_1+a_2t_f+a_3t_f{}^2=0 \end{cases} \Rightarrow \begin{cases} a_0=\theta_0 \\ a_1=0 \\ a_2=\dfrac{3}{t_f^2}(\theta_f-\theta_0) \\ a_3=-\dfrac{2}{t_f^3}(\theta_f-\theta_0) \end{cases} \Rightarrow \begin{cases} \theta(t)=\theta_0+\dfrac{3}{t_f^2}(\theta_f-\theta_0)t^2-\dfrac{2}{t_f^3}(\theta_f-\theta_0)t^3 \\ \dot{\theta}(t)=\dfrac{6}{t_f^2}(\theta_f-\theta_0)t-\dfrac{6}{t_f^3}(\theta_f-\theta_0)t^2 \\ \qquad =\dfrac{6}{t_f^2}(\theta_f-\theta_0)(1-\dfrac{t}{t_f})t \quad \text{Monotone} \\ sig(\dot{\theta})=sig(\theta_f-\theta_0) \end{cases}$$

Joint Trajectory Planning

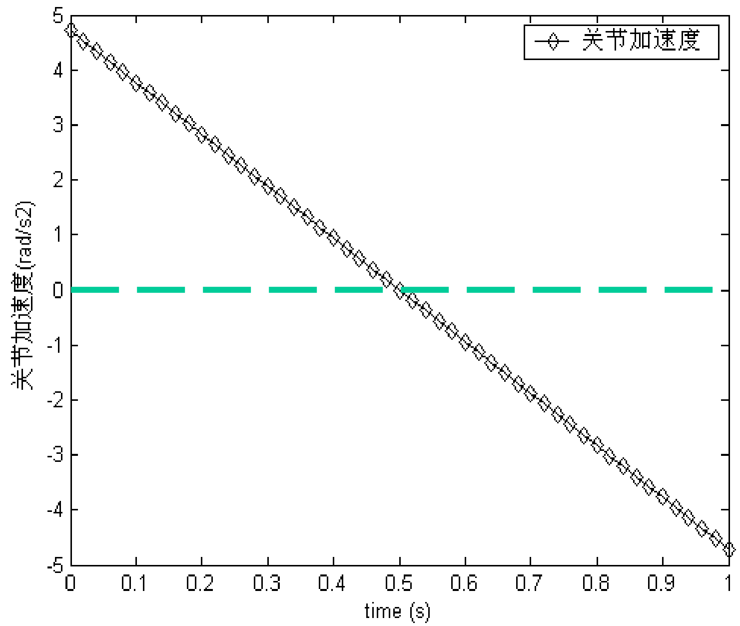
➢ 1.1 Cubic Polynomial Interpolation Method

Case Study: for a rotatory joint, supposing q_0=0 for t_0=0, q_f=π/4 for t_f=1s, sampling period is 0.02s, joint motion trajectory under cubic polynomial interpolation method is as follows:

$a_0=0$, $a_1=0$, $a_2=2.3562$, $a_3=-1.5708$



(a)



(b)

Joint Trajectory Planning

➢ 1.2 Cubic Polynomial Interpolation Method through Path Point

Define $\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \Rightarrow \dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2$

$$\begin{cases} \theta(0) = \theta_0 \\ \theta(t_f) = \theta_f \\ \dot{\theta}(0) = \dot{\theta}_0 \\ \dot{\theta}(t_f) = \dot{\theta}_f \end{cases} \qquad \begin{cases} a_0 = \theta_0 \\ a_1 = \dot{\theta}_0 \\ a_2 = \dfrac{3}{t_f^2}(\theta_f - \theta_0) - \dfrac{2}{t_f}\dot{\theta}_0 - \dfrac{1}{t_f}\dot{\theta}_f \\ a_3 = -\dfrac{2}{t_f^3}(\theta_f - \theta_0) + \dfrac{1}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0) \end{cases}$$

$$\dot{\theta}(t) = \dot{\theta}_0 + 2[\frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f]t + 3[-\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0)]t^2$$

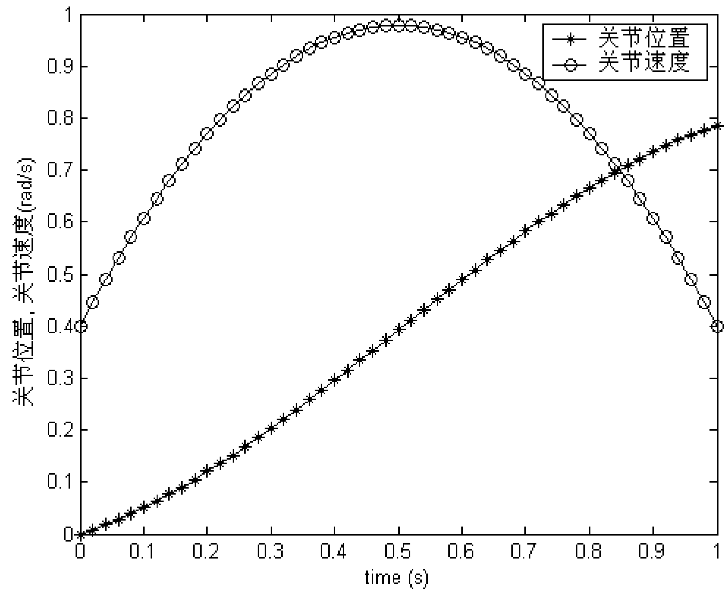
Note that the Path Point can only be velocity rather than position under cubic polynomial interpolation method.

Joint Trajectory Planning

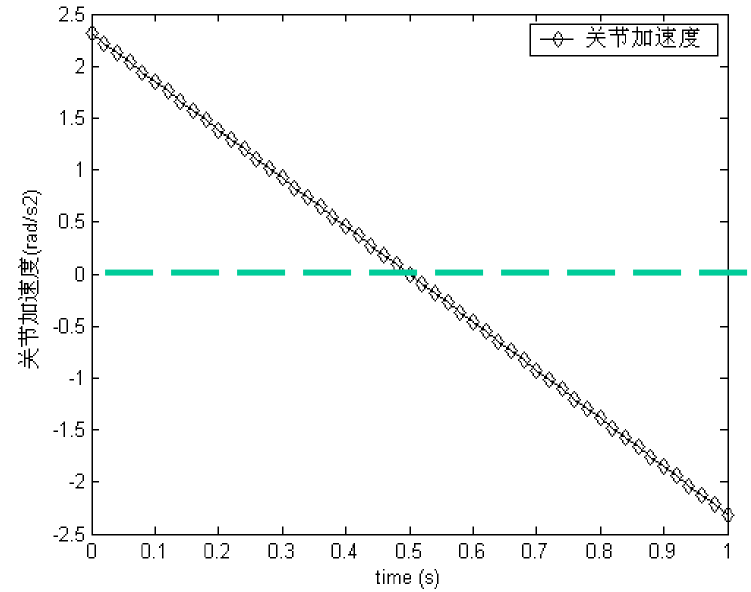
- 1.2 Cubic Polynomial Interpolation Method through Path Point

Case Study: for a rotatory joint, supposing q_0=0, t_f=1s, q_f=π/4, $\dot{q}_0 = \dot{q}_f$ =0.4 rad/s when t_0=0, sampling period is 0.02s, joint motion trajectory under cubic polynomial interpolation method is as follows:

a_0=0， a_1=0.4000， a_2=1.1562， a_3=-0.7708



(a)



(b)

Joint Trajectory Planning

➢ 2 High Order Polynomial Interpolation Method

Define $\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \Rightarrow \dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4$

$$\begin{cases} \theta(0) = q_0 \\ \theta(t_f) = q_f \\ \dot{\theta}(0) = \dot{q}_0 \\ \dot{\theta}(t_f) = \dot{q}_f \\ \ddot{\theta}(0) = \ddot{q}_0 \\ \ddot{\theta}(t_f) = \ddot{q}_f \end{cases} \qquad \begin{cases} a_0 = q_0 \\ a_1 = \dot{q}_0 \\ a_2 = \dfrac{\ddot{q}_0}{2} \\ a_3 = \dfrac{20q_f - 20q_0 - (8\dot{q}_f + 12\dot{q}_0)t_f - (3\ddot{q}_0 - \ddot{q}_f)t_f^2}{2t_f^3} \\ a_4 = \dfrac{-30q_f + 30q_0 + (14\dot{q}_f + 16\dot{q}_0)t_f + (3\ddot{q}_0 - 2\ddot{q}_f)t_f^2}{2t_f^4} \\ a_5 = \dfrac{12q_f - 12q_0 - (6\dot{q}_f + 6\dot{q}_0)t_f - (\ddot{q}_0 - \ddot{q}_f)t_f^2}{2t_f^5} \end{cases}$$

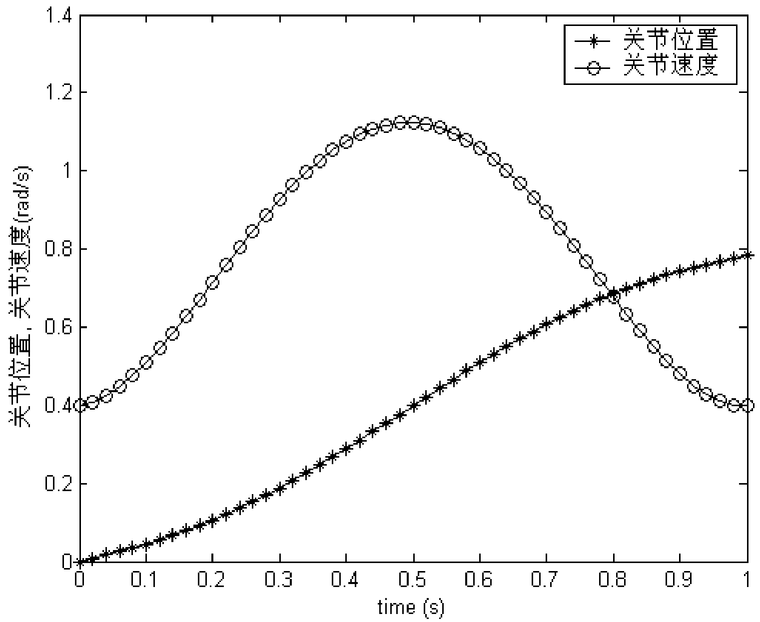
We can and have to regulate position, velocity and acceleration simultaneously under high order polynomial interpolation method.

Joint Trajectory Planning

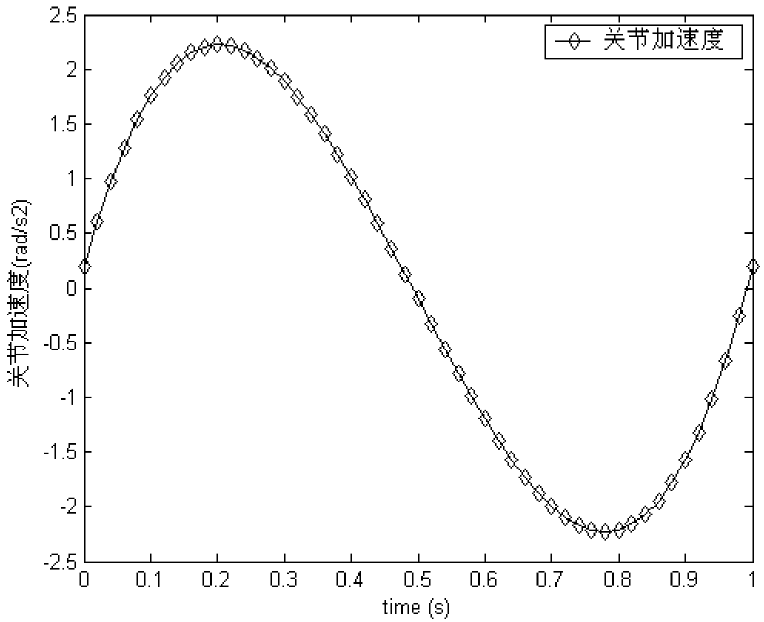
➢ 2 High Order Polynomial Interpolation Method

Case Study: for a rotatory joint, q_0=0, t_f=1s, q_f=π/4, $\dot{q}_0 = \dot{q}_f$ =0.4 rad/s, $\ddot{q}_0 = \ddot{q}_f$ =0.2 rad /s^2 when t_0=0, sampling period is 0.02s, joint motion trajectory under cubic polynomial interpolation method is as follows:

a_0=0, a_1=0.4000, a_2=0.1000, a_3=3.6540, a_4=-5.6810, a_5= 2.3124



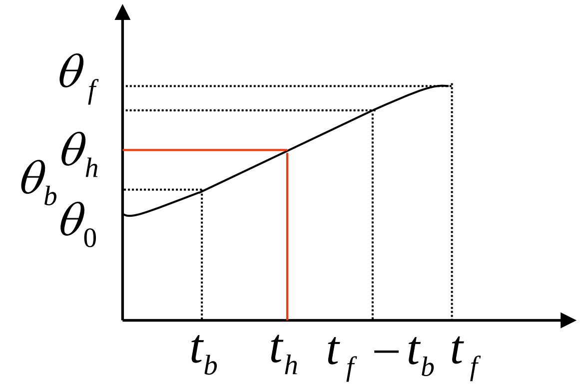
(a)



(b)

Joint Trajectory Planning

➢ 3 Parabolic Interpolation Method



From $t_b - t_f$ - t_b: linear interpolation.

$\ddot{\theta}$ is known in advance. t_b?

$$\dot{\theta}_{tb} = \frac{\theta_h - \theta_b}{t_h - t_b}, \quad \theta_b = \theta_0 + \frac{1}{2}\ddot{\theta}t_b^2$$

$$\dot{\theta}_{tb} = \ddot{\theta}t_b \Rightarrow \ddot{\theta}t_b = \frac{\theta_h - \theta_b}{t_h - t_b} = \frac{(\theta_f + \theta_0)/2 - \theta_b}{t_f/2 - t_b} = \frac{\theta_f + \theta_0 - 2\theta_b}{t_f - 2t_b} = \frac{\theta_f - \theta_0 - \ddot{\theta}t_b^2}{t_f - 2t_b}$$

$$\ddot{\theta}t_b^2 - \ddot{\theta}t_f t_b + (\theta_f - \theta_0) = 0$$

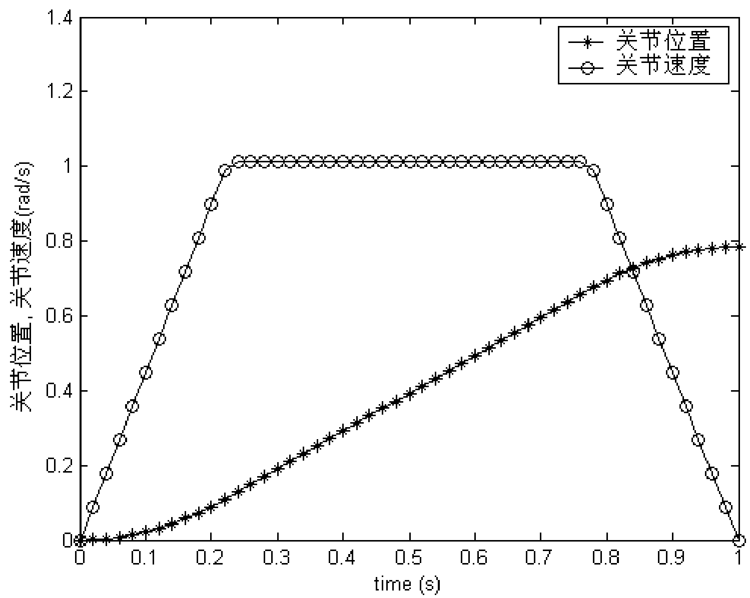
$$t_b = \frac{t_f}{2} - \frac{\sqrt{\ddot{\theta}^2 t_f^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}} \qquad \ddot{\theta} \ge \frac{4(\theta_f - \theta_0)}{t_f^2}$$

If $\ddot{\theta} = \frac{4(\theta_f - \theta_0)}{t_f^2}$, there will be no linear interpolation. Larger acceleration, the shorter for linear interpolation.

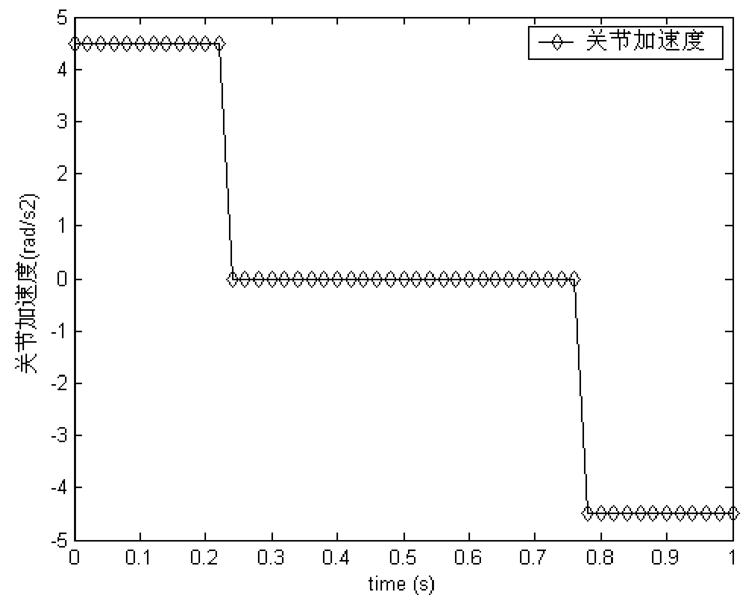
Joint Trajectory Planning

➢ 3 Parabolic Interpolation Method

Case Study: for a rotatory joint $q_0=0$, $t_f=1s$, $q_f=\pi/4$, $\dot{q}_0 = \dot{q}_f = 0$, $\ddot{q} = 4.5 rad/s^2$ when $t_0=0$, then for parabolic interpolation method, $t_b = 0.2253s$, $\theta_b=0.1142$. Joint motion trajectory under cubic polynomial interpolation method is as follows:



(a)



(b)

Joint Trajectory Planning

➢ B-Spline Interpolation

0th order B-Spline $N_{i,0}(x)=\begin{cases}1, x\in[x_i,x_{i+1}) \\ 0, x\notin[x_i,x_{i+1})\end{cases}$

mth order B-Spline $N_{i,m}(x)=\dfrac{x-x_i}{x_{i+m}-x_i}N_{i,m-1}(x)+\dfrac{x_{i+m+1}-x}{x_{i+m+1}-x_{i+1}}N_{i+1,m-1}(x)$

1st order B-Spline

$$N_{i,1}(x)=\begin{cases}\dfrac{x-x_i}{x_{i+1}-x_i}, & x\in[x_i,x_{i+1}) \\ \dfrac{x_{i+2}-x}{x_{i+2}-x_{i+1}}, & x\in[x_{i+1},x_{i+2})\end{cases}$$

2nd order B-Spline

$$N_{i,2}(x)=\begin{cases}\dfrac{(x-x_i)^2}{(x_{i+1}-x_i)(x_{i+2}-x_i)}, & x\in[x_i,x_{i+1}) \\ \dfrac{(x-x_i)(x_{i+2}-x)}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})}+\dfrac{(x-x_{i+1})(x_{i+3}-x)}{(x_{i+2}-x_{i+1})(x_{i+3}-x_{i+1})}, & x\in[x_{i+1},x_{i+2}) \\ \dfrac{(x_{i+3}-x)^2}{(x_{i+3}-x_{i+1})(x_{i+3}-x_{i+2})}, & x\in[x_{i+2},x_{i+3})\end{cases}$$

Joint Trajectory Planning

➢ B-Spline Interpolation

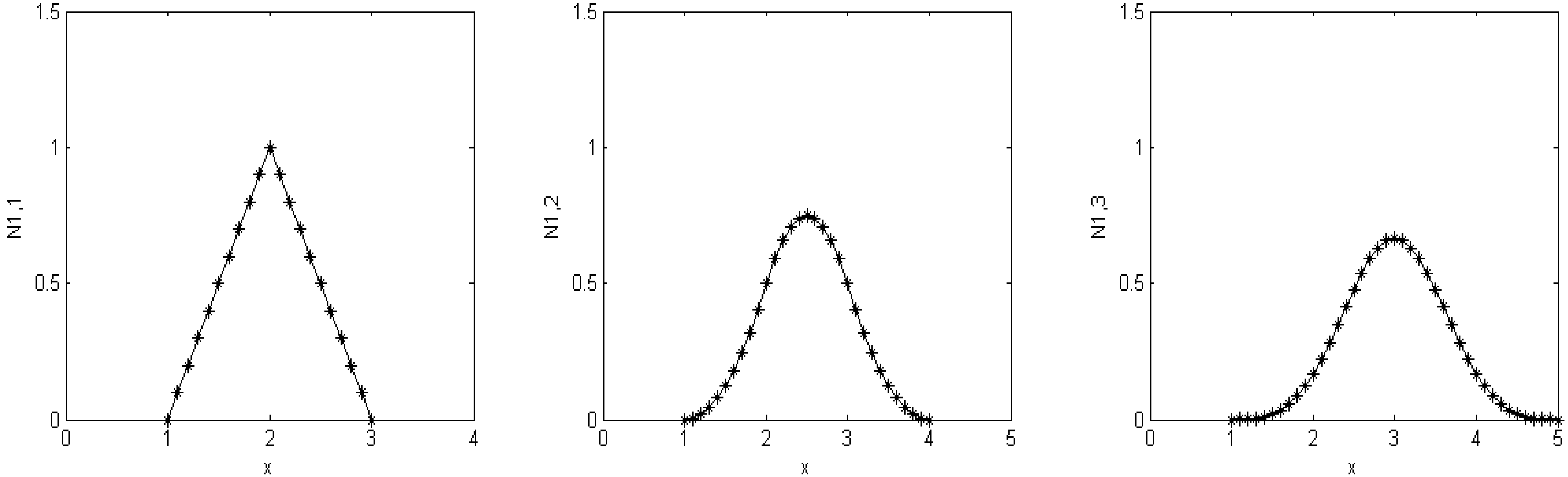
3rd order B-Spline

$$N_{i,3}(x)=\begin{cases}\dfrac{(x-x_i)^3}{(x_{i+1}-x_i)(x_{i+2}-x_i)(x_{i+3}-x_i)}, & x\in[x_i,x_{i+1})\\[2ex] \dfrac{(x-x_i)^2(x_{i+2}-x)}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})(x_{i+3}-x_i)}+\dfrac{(x-x_i)(x-x_{i+1})(x_{i+3}-x)}{(x_{i+2}-x_{i+1})(x_{i+3}-x_{i+1})(x_{i+3}-x_i)} & \\ +\dfrac{(x-x_{i+1})^2(x_{i+4}-x)}{(x_{i+2}-x_{i+1})(x_{i+3}-x_{i+1})(x_{i+4}-x_{i+1})}, x\in[x_{i+1},x_{i+2}) & \\[2ex] \dfrac{(x-x_i)(x_{i+3}-x)^2}{(x_{i+3}-x_i)(x_{i+3}-x_{i+1})(x_{i+3}-x_{i+2})}+\dfrac{(x-x_{i+1})(x_{i+3}-x)(x_{i+4}-x)}{(x_{i+3}-x_{i+1})(x_{i+3}-x_{i+2})(x_{i+4}-x_{i+1})} & \\ +\dfrac{(x-x_{i+2})(x_{i+4}-x)^2}{(x_{i+3}-x_{i+2})(x_{i+4}-x_{i+1})(x_{i+4}-x_{i+2})}, x\in[x_{i+2},x_{i+3}) & \\[2ex] \dfrac{(x_{i+4}-x)^3}{(x_{i+4}-x_{i+1})(x_{i+4}-x_{i+2})(x_{i+4}-x_{i+3})}, & x\in[x_{i+3},x_{i+4})\end{cases}$$

Joint Trajectory Planning

- B-Spline Interpolation

$$f(x)=\sum_{i=-m}^{k} a_i N_{i,m}(x) \quad [x_0, x_{k+1})$$



(a) 1^{st} (b) 2^{nd} (c) 3^{rd}

$$N_{1,2}(1)=0, N_{1,2}(2)=\frac{1}{2}, N_{1,2}(3)=\frac{1}{2}, N_{1,2}(4)=0$$

$$N_{1,3}(1)=0, N_{1,3}(2)=\frac{1}{6}, N_{1,3}(3)=\frac{2}{3}$$

$$N_{1,3}(4)=\frac{1}{6}, N_{1,3}(5)=0$$

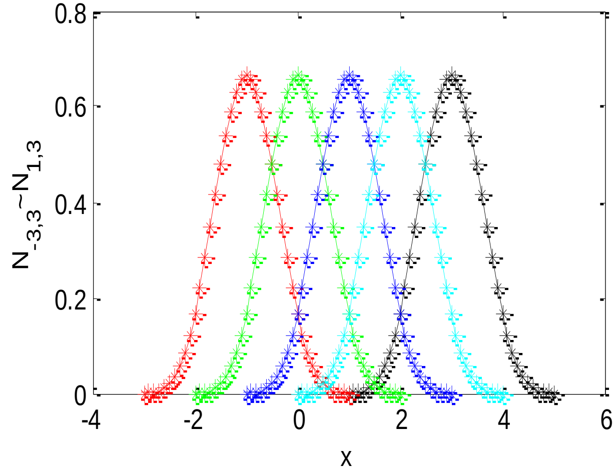
Joint Trajectory Planning

- B-Spline Interpolation

Case Study: t ∈[0， 4], for a rotatory joint, $q(0)=2$, $q(1)=2.8$, $q(2)=1.2$, $q(3)=2.2$, $q(4)=0.9$, How is the B-spline interpolation by 3rd order B-spine?

Here supposing $a_{-3}=a_{-2}=0$, we get

$$\begin{cases} a_{-1}N_{-1,3}(0)+a_0N_{0,3}(0)=q(0) \\ a_{-1}N_{-1,3}(1)+a_0N_{0,3}(1)+a_1N_{1,3}(1)=q(1) \\ a_{-1}N_{-1,3}(2)+a_0N_{0,3}(2)+a_1N_{1,3}(2)+a_2N_{2,3}(2)=q(2) \\ a_0N_{0,3}(3)+a_1N_{1,3}(3)+a_2N_{2,3}(3)+a_3N_{3,3}(3)=q(3) \\ a_1N_{1,3}(4)+a_2N_{2,3}(4)+a_3N_{3,3}(4)+a_4N_{4,3}(4)=q(4) \end{cases}$$

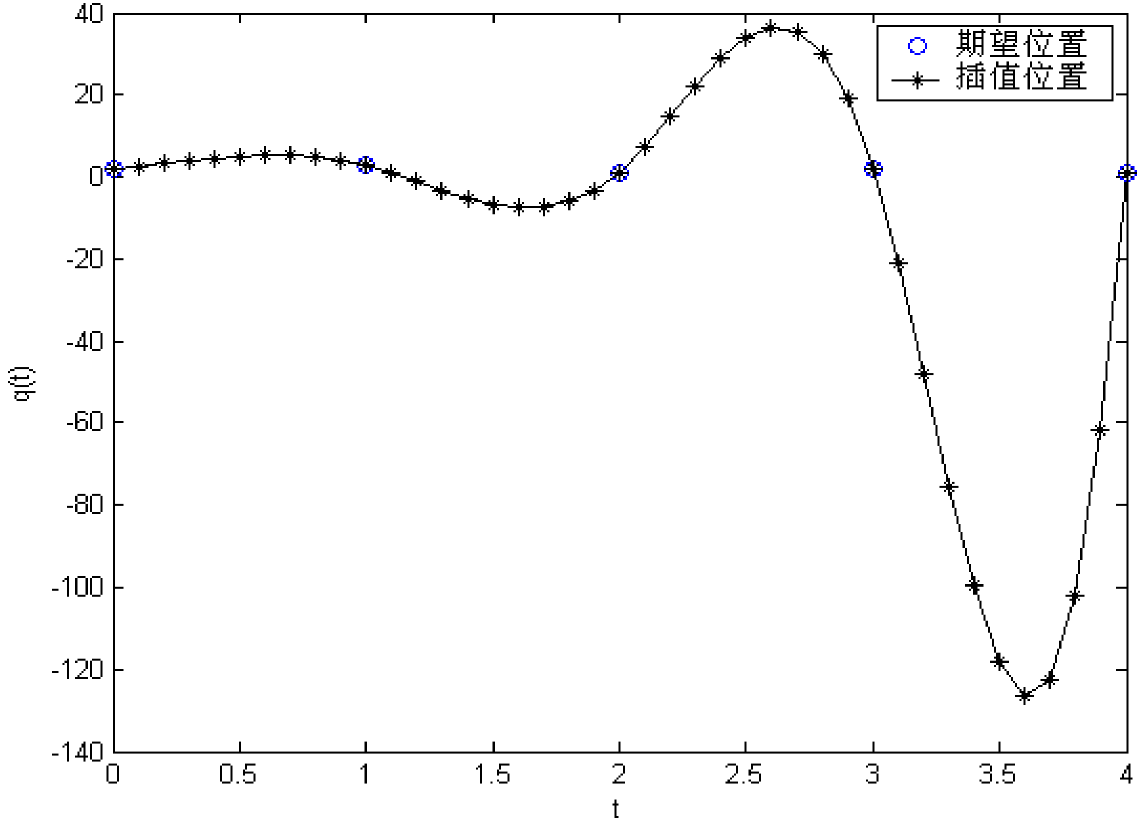


Finally: $a_{-1}=12$， $a_0=-31.2$， $a_1=120$， $a_2=-435.6$， $a_3=1627.8$.
The interpolation function is now:

$$f(x)=12N_{-1,3}(x)-31.2N_{0,3}(x)+120N_{1,3}(x)-435.6N_{2,3}(x)+1627.8N_{3,3}(x)$$

Joint Trajectory Planning

- B-Spline Interpolation



a_{-1}=12， a_0=-31.2， a_1=120， a_2=-435.6， a_3=1627.8

Joint Trajectory Planning

➢ B-Spline Interpolation

To eliminate vibration, we added some control points to calculate a_{-3}~a_3

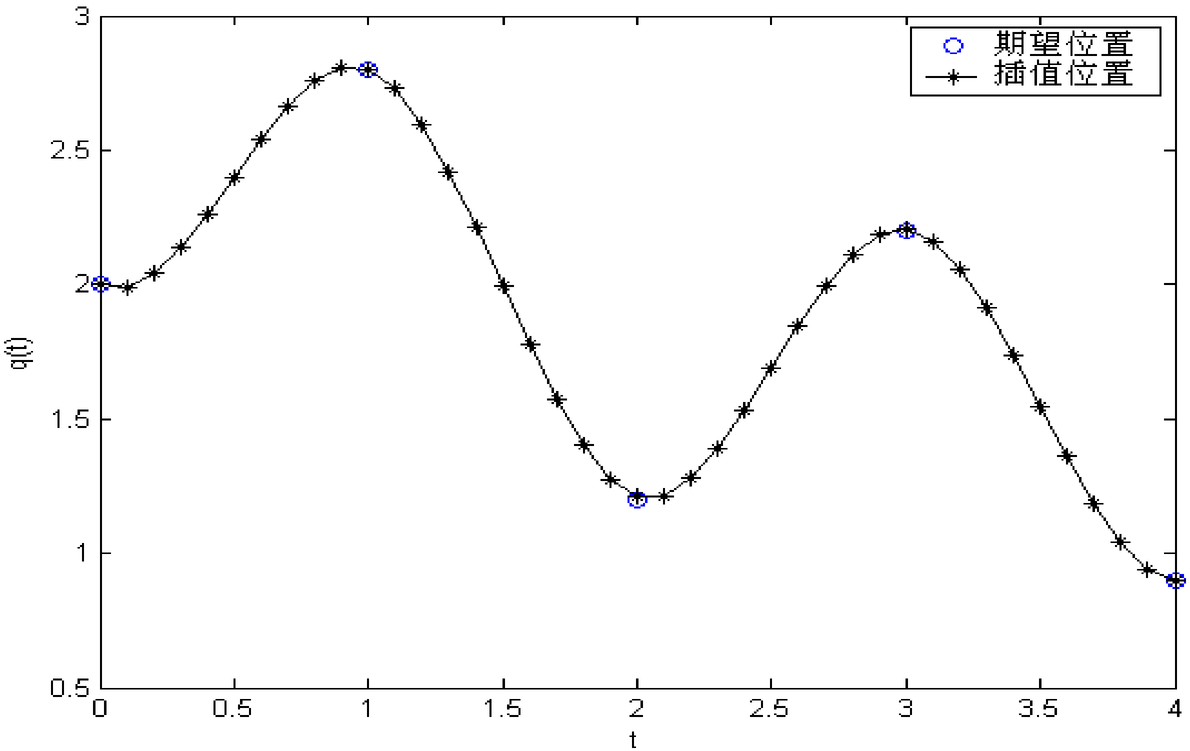
$$\begin{cases}
a_{-3}N_{-3,3}(0)+a_{-2}N_{-2,3}(0)+a_{-1}N_{-1,3}(0)+a_0N_{0,3}(0)=q(0)\\
a_{-3}N_{-3,3}(0.5)+a_{-2}N_{-2,3}(0.5)+a_{-1}N_{-1,3}(0.5)+a_0N_{0,3}(0.5)=[q(0)+q(1)]/2\\
a_{-2}N_{-2,3}(1)+a_{-1}N_{-1,3}(1)+a_0N_{0,3}(1)+a_1N_{1,3}(1)=q(1)\\
a_{-2}N_{-2,3}(1.5)+a_{-1}N_{-1,3}(1.5)+a_0N_{0,3}(1.5)+a_1N_{1,3}(1.5)=[q(1)+q(2)]/2\\
a_{-1}N_{-1,3}(2)+a_0N_{0,3}(2)+a_1N_{1,3}(2)+a_2N_{2,3}(2)=q(2)\\
a_{-1}N_{-1,3}(2.5)+a_0N_{0,3}(2.5)+a_1N_{1,3}(2.5)+a_2N_{2,3}(2.5)=[q(2)+q(3)]/2\\
a_0N_{0,3}(3)+a_1N_{1,3}(3)+a_2N_{2,3}(3)+a_3N_{3,3}(3)=q(3)\\
a_0N_{0,3}(3.5)+a_1N_{1,3}(3.5)+a_2N_{2,3}(3.5)+a_3N_{3,3}(3.5)=[q(3)+q(4)]/2\\
a_1N_{1,3}(4)+a_2N_{2,3}(4)+a_3N_{3,3}(4)+a_4N_{4,3}(4)=q(4)
\end{cases}$$

So: a_{-3}=4.8666， a_{-2}=0.7783， a_{-1}=4.0189， a_0=-0.0392， a_1=3.3999， a_2=-0.3105， a_3=3.2430 。

Joint Trajectory Planning

- B-Spline Interpolation

$$f(x) = 4.8666N_{-3,3}(x) + 0.7783N_{-2,3}(x) + 4.0189N_{-1,3}(x) - 0.0392N_{0,3}(x)$$
$$+ 3.3999N_{1,3}(x) - 0.3105N_{2,3}(x) + 3.2430N_{3,3}(x)$$



End-effector Trajectory Planning

- End-effector Trajectory Planning in End-effector Cartesian Space

➢ Linear End-effector Motion Trajectory: The combination of linear translation and rotation about an spatial axis

Start Pose: $T_1 = \left[\begin{array}{c:c} R_1 & P_1 \\ \hdashline 0 & 1 \end{array}\right]$ End Pose: $T_2 = \left[\begin{array}{c:c} R_2 & P_2 \\ \hdashline 0 & 1 \end{array}\right]$

❖ Translation Vector for i^{th} step

❖ Rotation Transformation Matrix for i^{th} step, (Equivalent rotation axis and angle for i^{th} step)

❖ Pose for i^{th} step

The translation vector form T_2 to T_1: $P = P_2 - P_1$

Translation Vector for i^{th} step : $P(i) = \alpha_i P$

Rotation Transformation Matrix:

$$R = R_1^T R_2 = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}$$

End-effector Trajectory Planning

➢ Linear End-effector Motion Trajectory

Rotation Transformation Matrix for i^th^ step:

$$R(i) = \mathrm{Rot}(f, \theta_i) = \begin{bmatrix} f_x f_x \mathrm{vers}\theta_i + c\theta_i & f_y f_x \mathrm{vers}\theta_i - f_z s\theta_i & f_z f_x \mathrm{vers}\theta_i + f_y s\theta_i & 0 \\ f_x f_y \mathrm{vers}\theta_i + f_z s\theta_i & f_y f_y \mathrm{vers}\theta_i + c\theta_i & f_z f_y \mathrm{vers}\theta_i - f_x s\theta_i & 0 \\ f_x f_z \mathrm{vers}\theta_i - f_y s\theta_i & f_y f_z \mathrm{vers}\theta_i + f_x s\theta_i & f_z f_z \mathrm{vers}\theta_i + c\theta_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\theta_i = \alpha_i \theta$$

Pose of i^th^ step:

$$T(i) = \left[\begin{array}{c:c} I & P_1 \\ \hdashline 0 & 1 \end{array}\right] \left[\begin{array}{c:c} I & P(i) \\ \hdashline 0 & 1 \end{array}\right] \left[\begin{array}{c:c} R_1 & 0 \\ \hdashline 0 & 1 \end{array}\right] \left[\begin{array}{c:c} R(i) & 0 \\ \hdashline 0 & 1 \end{array}\right] = \left[\begin{array}{c:c} R_1 R(i) & P_1 + P(i) \\ \hdashline 0 & 1 \end{array}\right]$$

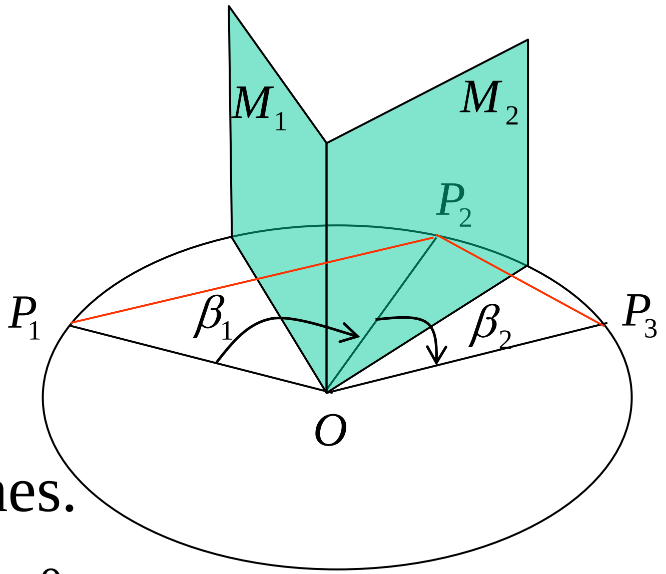
End-effector Trajectory Planning

➢ Circular Arc End-effector Motion Trajectory

Start, middle and end pose of Circular Arc Trajectory are：

$$T_1 = \begin{bmatrix} R_1 & P_1 \\ 0 & 1 \end{bmatrix}, T_2 = \begin{bmatrix} R_2 & P_2 \\ 0 & 1 \end{bmatrix}, T_3 = \begin{bmatrix} R_3 & P_3 \\ 0 & 1 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T, P_2 = \begin{bmatrix} x_2 & y_2 & z_2 \end{bmatrix}^T, P_3 = \begin{bmatrix} x_3 & y_3 & z_3 \end{bmatrix}^T$$



✓ Translation Vector for i^th Step

❖ Circle Center

Circle center is the intersection of three planes.

P_1, P_2, P_3 determine a plan： $A_1 x + B_1 y + C_1 z - D_1 = 0$

$$A_1 = \begin{vmatrix} y_1 & z_1 & 1 \\ y_2 & z_2 & 1 \\ y_3 & z_3 & 1 \end{vmatrix}, B_1 = -\begin{vmatrix} x_1 & z_1 & 1 \\ x_2 & z_2 & 1 \\ x_3 & z_3 & 1 \end{vmatrix}, C_1 = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}, D_1 = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

End-effector Trajectory Planning

➢ Circular Arc End-effector Motion Trajectory

Plane M_1:

$$A_2 x + B_2 y + C_2 z - D_2 = 0$$

$$A_2 = x_2 - x_1, B_2 = y_2 - y_1, C_2 = z_2 - z_1$$

$$D_2 = \frac{1}{2}(x_2^2 + y_2^2 + z_2^2 - x_1^2 - y_1^2 - z_1^2)$$

Plane M_2:

$$A_3 x + B_3 y + C_3 z - D_3 = 0$$

$$A_3 = x_2 - x_3, B_3 = y_2 - y_3, C_3 = z_2 - z_3$$

$$D_3 = \frac{1}{2}(x_2^2 + y_2^2 + z_2^2 - x_3^2 - y_3^2 - z_3^2)$$

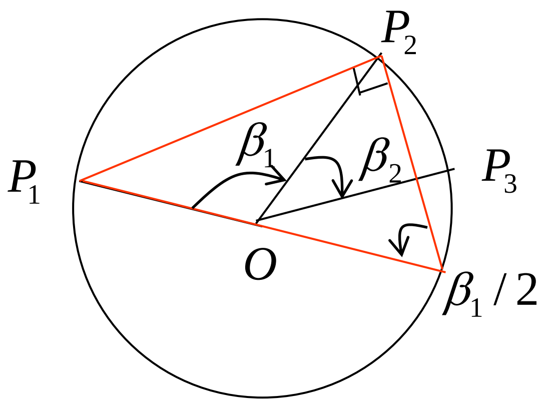
Center Coordinates: $x_0 = \frac{\Delta x}{\Delta}, y_0 = \frac{\Delta y}{\Delta}, z_0 = \frac{\Delta z}{\Delta}$

$$\Delta = \begin{vmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{vmatrix}, \Delta x = -\begin{vmatrix} D_1 & B_1 & C_1 \\ D_2 & B_2 & C_2 \\ D_3 & B_3 & C_3 \end{vmatrix}, \Delta y = \begin{vmatrix} A_1 & D_1 & C_1 \\ A_2 & D_2 & C_2 \\ A_3 & D_3 & C_3 \end{vmatrix}, \Delta z = \begin{vmatrix} A_1 & B_1 & D_1 \\ A_2 & B_2 & D_2 \\ A_3 & B_3 & D_3 \end{vmatrix}$$

End-effector Trajectory Planning

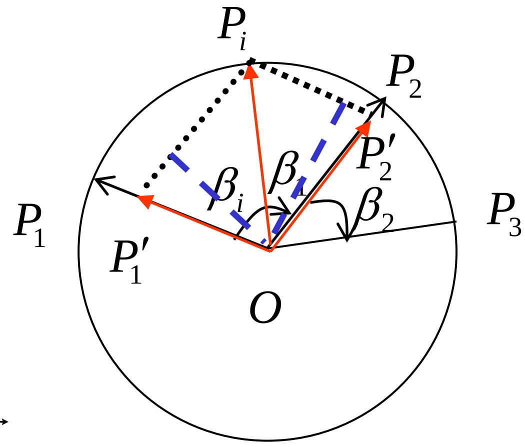
➢ Circular Arc End-effector Motion Trajectory

Radius： $R = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}$



$$\sin\frac{\beta_1}{2} = \frac{P_1P_2}{2R} \Rightarrow \beta_1 = 2\arcsin\frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}}{2R}$$

$$\sin\frac{\beta_2}{2} = \frac{P_2P_3}{2R} \Rightarrow \beta_2 = 2\arcsin\frac{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2}}{2R}$$



Rotation Axis： $\beta = \beta_1 + \beta_2$

❖ Translation Vector for i[th] Step

$$\overrightarrow{OP_i} = \overrightarrow{OP_1'} + \overrightarrow{OP_2'} \qquad \beta_i = \alpha_i \beta_1$$

$$\overrightarrow{OP_1'} = \frac{R\sin(\beta_1 - \beta_i)}{\sin\beta_1}\frac{\overrightarrow{OP_1}}{\left|\overrightarrow{OP_1}\right|} = \frac{\sin(\beta_1 - \beta_i)}{\sin\beta_1}\overrightarrow{OP_1}, \quad \overrightarrow{OP_2'} = \frac{\sin\beta_i}{\sin\beta_1}\overrightarrow{OP_2}$$

$$\overrightarrow{OP_i} = \frac{\sin(\beta_1 - \beta_i)}{\sin\beta_1}\overrightarrow{OP_1} + \frac{\sin\beta_i}{\sin\beta_1}\overrightarrow{OP_2} = \lambda_1\overrightarrow{OP_1} + \delta_1\overrightarrow{OP_2}, \quad \lambda_1 = \frac{\sin(\beta_1 - \beta_i)}{\sin\beta_1}, \quad \delta_1 = \frac{\sin\beta_i}{\sin\beta_1}$$

End-effector Trajectory Planning

➢ Circular Arc End-effector Motion Trajectory

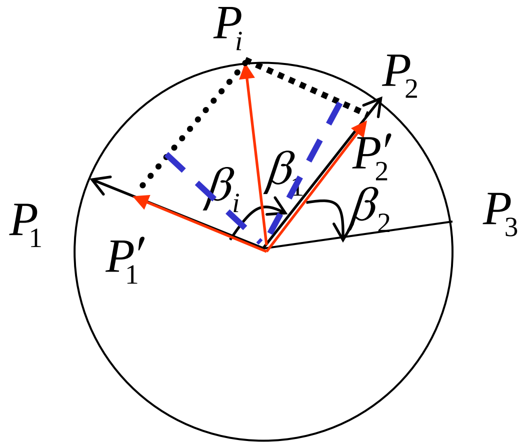
$$\overrightarrow{OP_i} = (x_i - x_0)\vec{i} + (y_i - y_0)\vec{j} + (z_i - z_0)\vec{k}$$

Coordinate of P_i from P_1 to P_2:

$$\begin{cases} x_i = x_0 + \lambda_i(x_1 - x_0) + \delta(x_2 - x_0) \\ y_i = y_0 + \lambda_i(y_1 - y_0) + \delta(y_2 - y_0), i = 0,1,2,\cdots n \\ z_i = z_0 + \lambda_i(z_1 - z_0) + \delta(z_2 - z_0) \end{cases}$$

Coordinate of P_i from P_2 to P_3:

$$\begin{cases} x_i = x_0 + \lambda(x_2 - x_0) + \delta(x_3 - x_0) \\ y_i = y_0 + \lambda(y_2 - y_0) + \delta(y_3 - y_0), i = 0,1,2,\cdots n \\ z_i = z_0 + \lambda(z_2 - z_0) + \delta(z_3 - z_0) \end{cases}$$



✓ Now we go to linear end-effector motion trajectory planning for each step.

THANK YOU