Chapter IV
Forward Kinematics

Dr. Song LIU

Director of AMNR Lab

liusong@shanghaitech.edu.cn

Robot Forward Kinematics

- Content
  - Link Transformation Matrix
  - Robot Kinematics
  - Forward Kinematics for Spherical Coordinates Articulated Robot
  - Forward Kinematics for Cylindrical Coordinates Articulated Robt
  - Forward Kinematics for Cartesian Coordinates Articulated Robot
  - Kinematics based Mobile Robot Localization (Dead Reckoning)

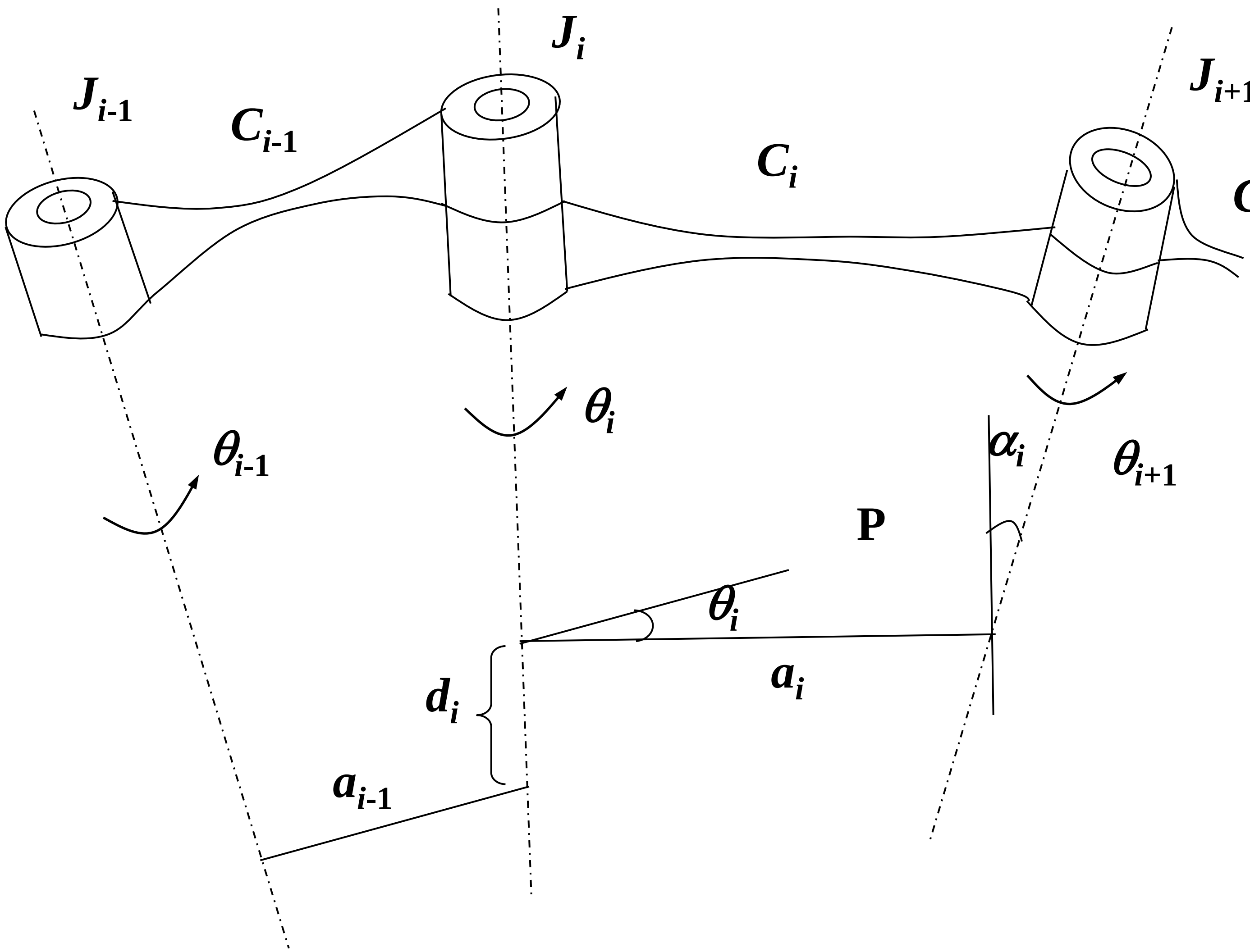
Link Transformation Matrix

- Some Concepts：
  - **Joint and Link**：Industrial robots typically consists of several joints and links (which is usually called manipulator). The joints are connections between links that permit relative motion between them; while the links are rigid bodies that give structure to the robot.
  - Joints composed of multiple DOFs can be modelled as the superposition of several joints with single DOF and separate link with zero link length.
  - A joint with single DOF can be translational joint or rotatory joint.

Link Transformation Matrix

- **Axis of Movement**: the axis of movement for joint i is noted as J_i. A rotatory joint rotates about J_i, while a translational joint translates along J_i.
- **Parameters of Links**: (Defined with respect to J_i)
  - **Length a_i**: the length of the common perpendicular segment of J_i and J_{i+1}.
  - **Link Twist α_i**: for J_i and J_{i+1}, J_i and the common perpendicular line determines a plane P. Twist angle is defined as the angle between J_{i+1} and P.
  - **Link offset d_i**: Apart from the first and the last link, for link C_i, we can find the common perpendiculars of J_{i-1} and J_i, J_i and J_{i+1}, noted as a_{i-1} and a_i. The distance between a_{i-1} and a_i is defined as link offset. (J_i is perpendicular to a_{i-1} and a_i)
  - **Joint Angle θ_i**: the angle between the projections of a_{i-1} and a_i on the surface perpendicular to J_i.
- $(a_i, \alpha_i, d_i, \theta_i)$ is defined as Denavit-Hartenberg (D-H) parameters.

Link Transformation Matrix



Question: Which parameters are constant and which parameters are changeable?

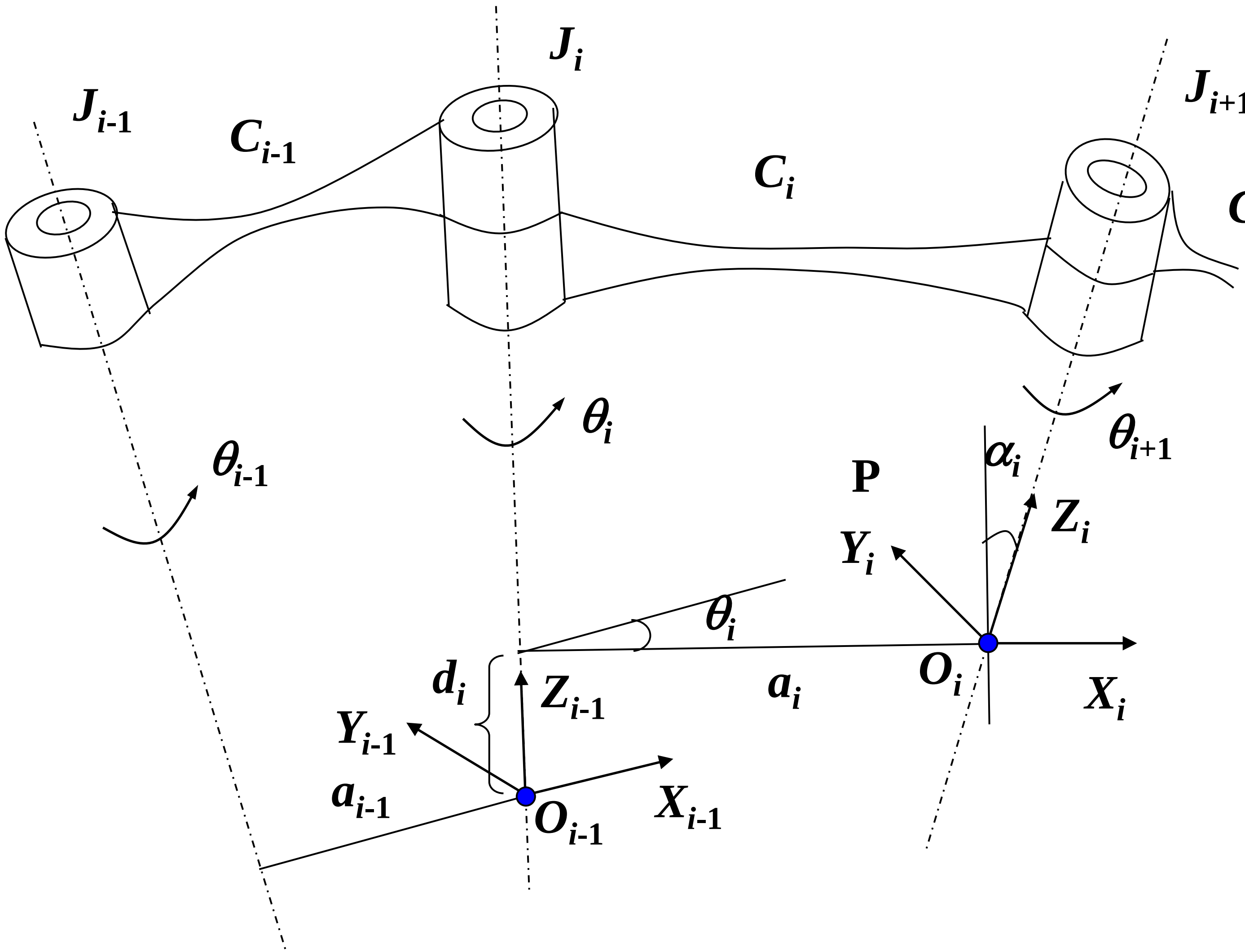
Link Transformation Matrix

1.1 Coordinates establishment case I: Origin O_i is located on J_{i+1}

- **Link Coordinates:** for adjacent links C_i and C_{i+1}, we consider J_{i-1}, J_i and J_{i+1}
  - ✓ For middle link C_i , the coordinates is established as follows:
    - Origin O_i: the intersection point of a_i and J_{i+1}
    - Z_i-axis: along J_{i+1}, also denoted as ${}^ie_z$
    - X_i-axis: along a_i, starting from O_i, also denoted as ${}^ie_x$
    - Y_i-axis: determined by right-hand rule from X_i and Y_i axis, also denoted as ${}^ie_2$
  - ✓ For first link C_1, the coordinates is established as follows:
    - Origin O_1: located on the origin of base coordinates
    - Z_1-axis: along J_1
    - X_1-axis: Arbitrarily determined
    - Y_1-axis: determined by right-hand rule from X_1 and Y_1 axis
  - ✓ For last link C_n , the coordinates is established as follows (the last link is typically the end-effector):
    - Origin O_n: located in the geometrical center of end-effector
    - Z_n-axis: along the direction from end-effector towards object
    - X_n-axis: along the direction from one-finger to another finger
    - Y_n-axis: determined by right-hand rule from X_n and Y_n axis

Link Transformation Matrix

1.1 Coordinates establishment case I: Origin O_i is located on J_{i+1}



Link Transformation Matrix

1.1 Coordinates establishment case I: Origin O_i is located on J_{i+1}

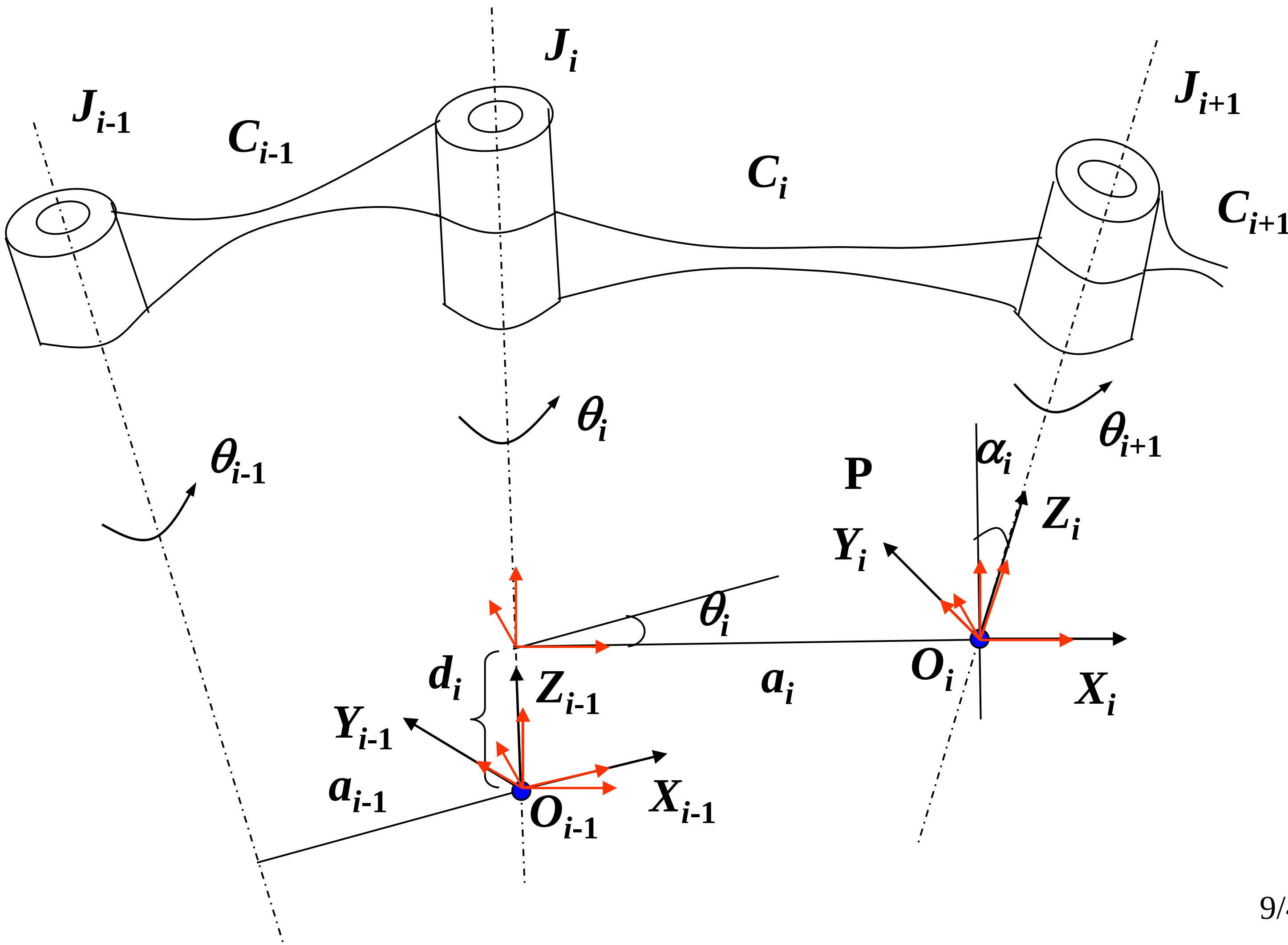
- **Link Transformation Matrix:** coordinates of C_{i-1} is transformed to coordinates of C_i by twice rotation and twice translation.
  - ✓ First: rotate about Z_{i-1}-axis by θ_i to align X_{i-1}-axis to X_i-axis
  - ✓ Then: translate along Z_{i-1}-axis by d_i to align O_{i-1} to the intersection point of J_i and a_i
  - ✓ Thereafter: translate along the new X_{i-1}-axis (now the X_i-axis) by a_i to align O_{i-1}towards O_i
  - ✓ Finally: rotate about X_i-axis by α_i to align Z_{i-1}-axis to Z_i-axis

Now, coordinates $O_{i-1}X_{i-1}Y_{i-1}Z_{i-1}$ and coordinates $O_iX_iY_iZ_i$ completely overlaps each other. The above mentioned procedure transforming coordinates of C_{i-1} to coordinates of C_i can be mathematically described by four times homogeneous transformations.

Link Transformation Matrix

1.1 Coordinates establishment case I: Origin O_i is located on J_{i+1}

Graphical Depiction of the Link Coordinates Transformation Process



Link Transformation Matrix

1.1 Coordinates establishment case I: Origin O_i is located on J_{i+1}

- Link Transformation Matrix (D-H Matrix):

$$T_i = \mathrm{Rot}(z,\theta_i)\mathrm{Trans}(0,0,d_i)\mathrm{Trans}(a_i,0,0)\mathrm{Rot}(x,\alpha_i)$$

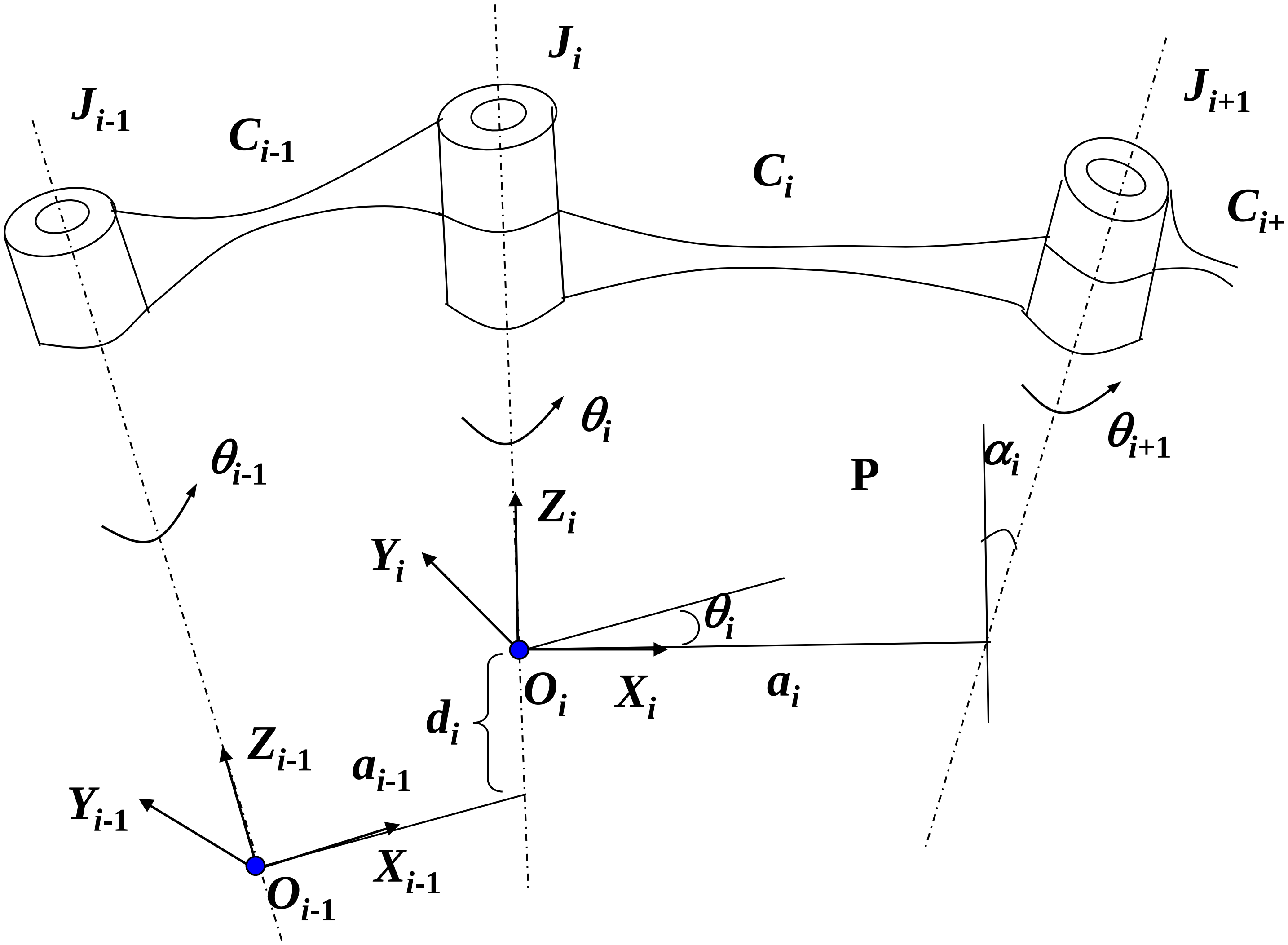
$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 &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

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

The Link Transformation Matrix T_i of link C_i fully depends on the D-H parameters of C_i, which actually describes how end-effector is affected by the link C_i.

Link Transformation Matrix

1.2 Coordinates establishment case II: Origin O_i is located on J_i



Link Transformation Matrix

1.2 Coordinates establishment case II: Origin O_i is located on J_i

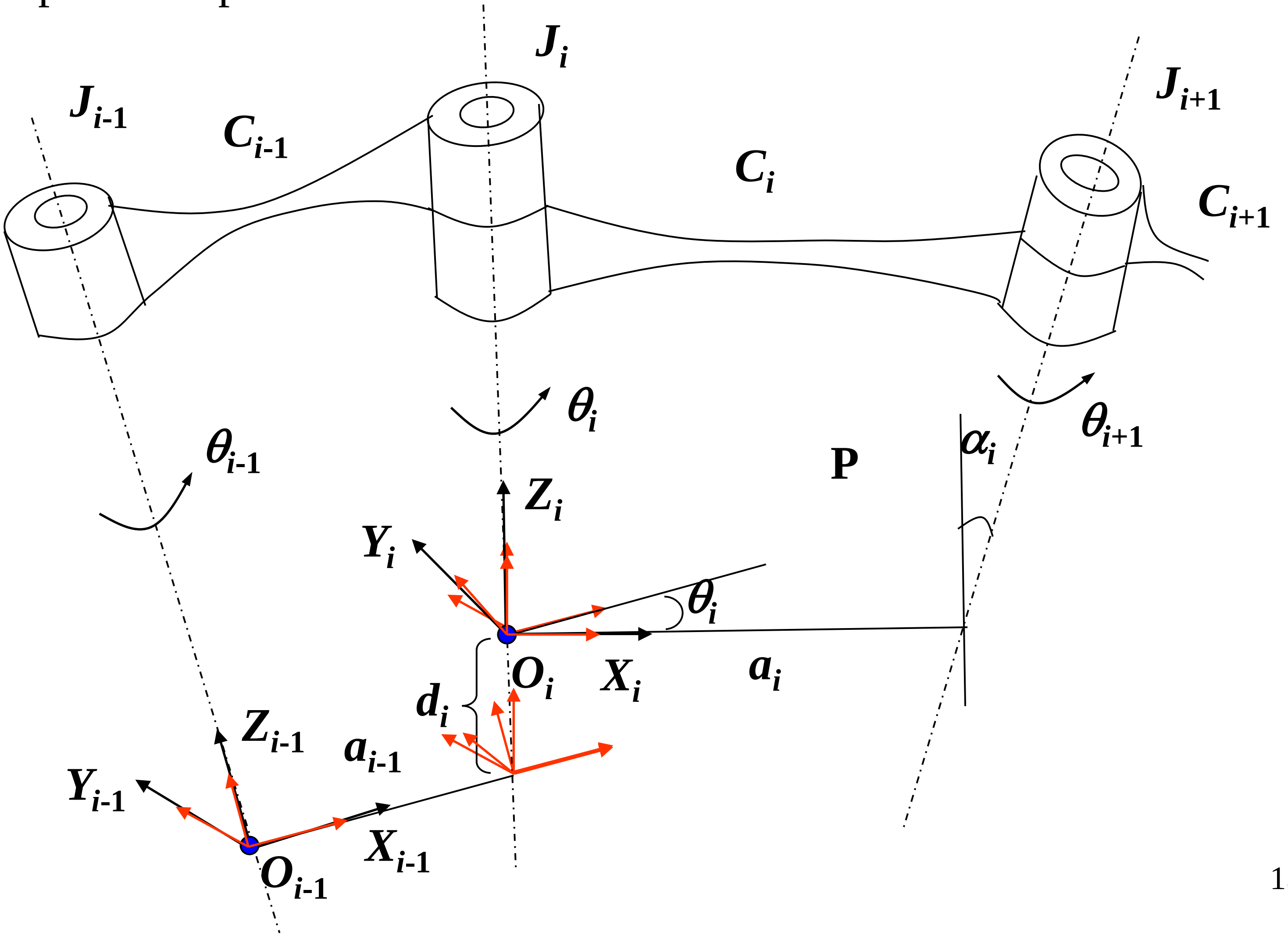
- **Link Transformation Matrix**： coordinate of Ci-1 is transformed to coordinates of Ci by twice rotation and twice translation.
  - ✓ First： translate along X_{i-1}-axis by a_{i-1} to align O_{i-1}towards the intersection point of J_{i-1} and a_i
  - ✓ Then： rotate about X_i-axis by α_{i-1} to align Z_{i-1}-axis to Z_i-axis
  - ✓ Thereafter： translate along Z_i-axis by d_i to align the new O_{i-1} to O_i
  - ✓ Finally： rotate about Z_i-axis by θ_i to align the new X_{i-1}-axis to X_i-axis

  Now, coordinates $O_{i-1}X_{i-1}Y_{i-1}Z_{i-1}$ and coordinates $O_iX_iY_iZ_i$ completely overlaps each other. The above mentioned procedure transforming coordinates of C_{i-1} to coordinates of C_i can be mathematically described by four times homogeneous transformations.

Link Transformation Matrix

1.2 Coordinates establishment case II: Origin O_i is located on J_i

Graphical Depiction of the Link Coordinates Transformation Process



Link Transformation Matrix

1.2 Coordinates establishment case II: Origin O_i is located on J_i

D-H Matrix：

$$T_i = \mathrm{Trans}(a_{i-1},0,0)\mathrm{Rot}(x,\alpha_{i-1})\mathrm{Trans}(0,0,d_i)\mathrm{Rot}(z,\theta_i)$$

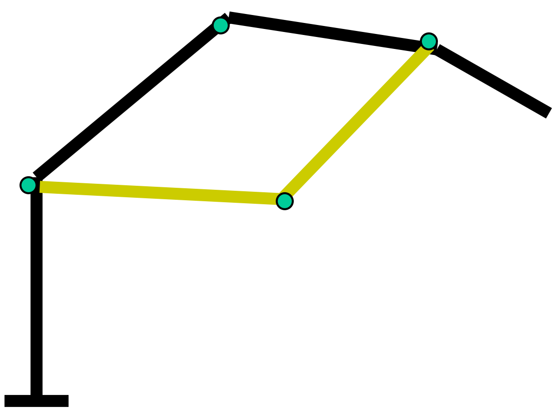
$$= \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_{i-1} & -\sin\alpha_{i-1} & 0 \\ 0 & \sin\alpha_{i-1} & \cos\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i \cos\alpha_{i-1} & \cos\theta_i \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i \sin\alpha_{i-1} \\ \sin\theta_i \sin\alpha_{i-1} & \cos\theta_i \sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i \cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

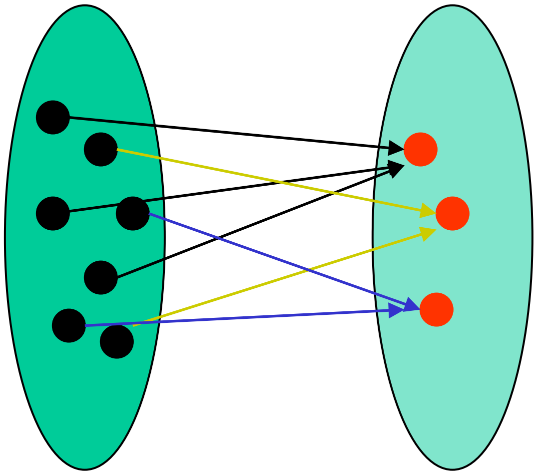
In this circumstance, the Link Transformation Matrix T_i of link C_i depends on the D-H parameters of both C_{i-1} and C_i,

Robot Forward Kinematics

- **Joint Space：** for a industrial robot with n DOFs, the position and orientation of all links can be described by a group of joint variables (d_i or θ_i) , which are called joint vector or joint coordinates. The space represented by the joint vector is called the joint space.
- **Forward Kinematics：** Joint Space→End Cartesian Space, one-to-one mapping
- **Inverse Kinematics：** End Cartesian Space → Joint Space, one-to-many mapping



Different joint vectors may yield the same end coordinate.



The mapping relationship between joint space to end coordinates

Robot Forward Kinematics

- **Forward Kinematics:** For an industrial robot having n DOFs, the D-H matrices of each link are T_1, T_2,…,T_n, then the pose of the robot end (a robot end is typically mounted with an end-effector) is:

$$T = T_1\, T_2\, T_3 \ldots\, T_n$$

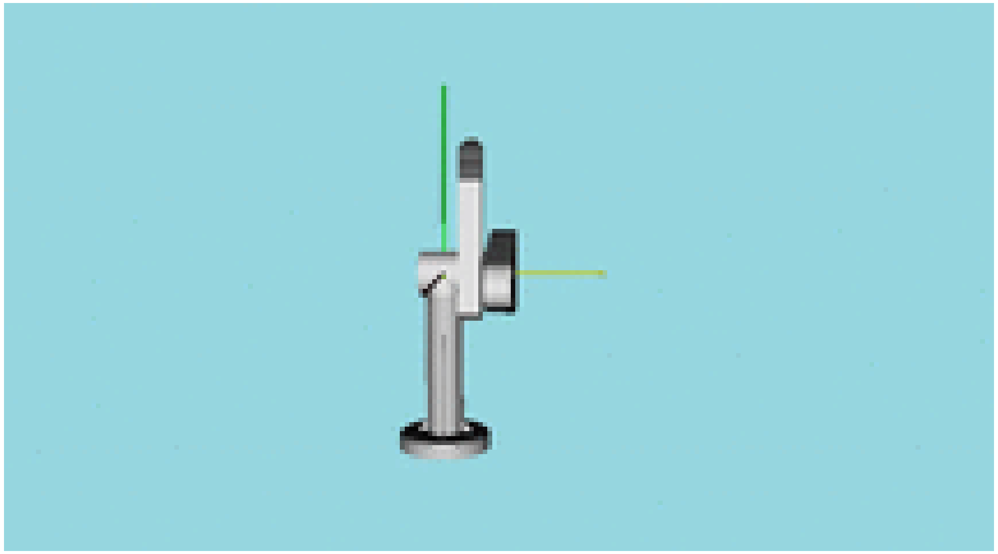
Though we have the two approaches to establish link coordinates and get different D-H matrices, we will always get the same robot end pose if the base coordinates are the same.

The pose of robot end with respect to $C_{i\text{-}1}$ is:

$${}^{i\text{-}1}T_n = T_i\, T_{i+1} \ldots\, T_n$$

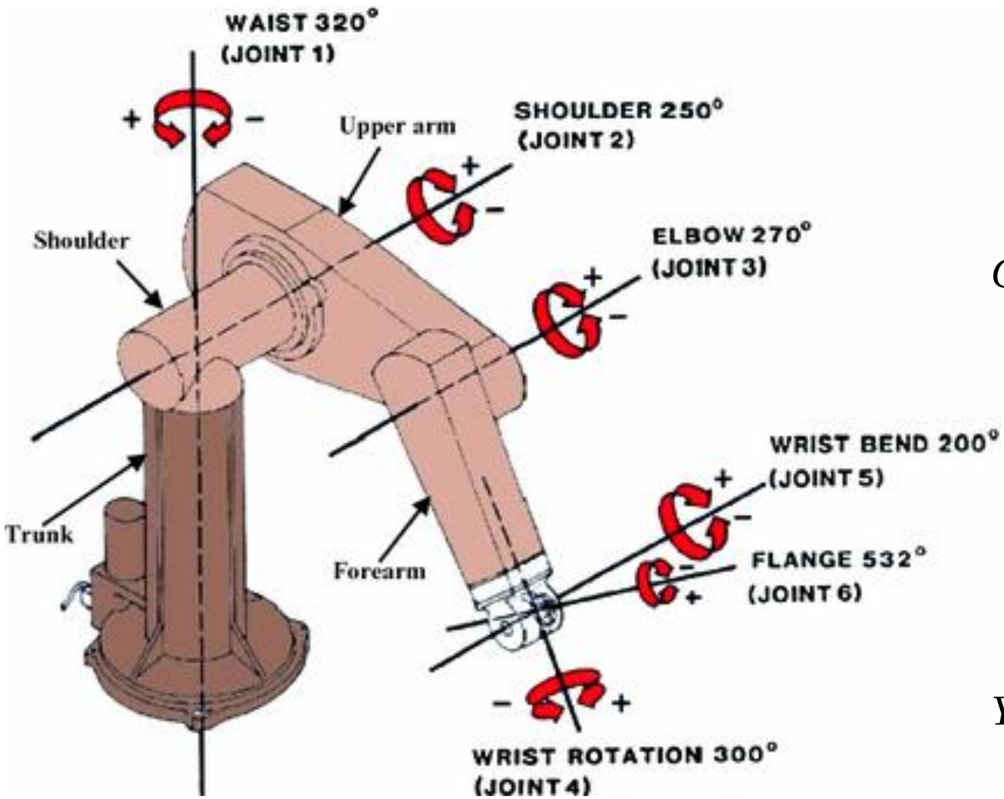
Regarding PUMA560 Robot

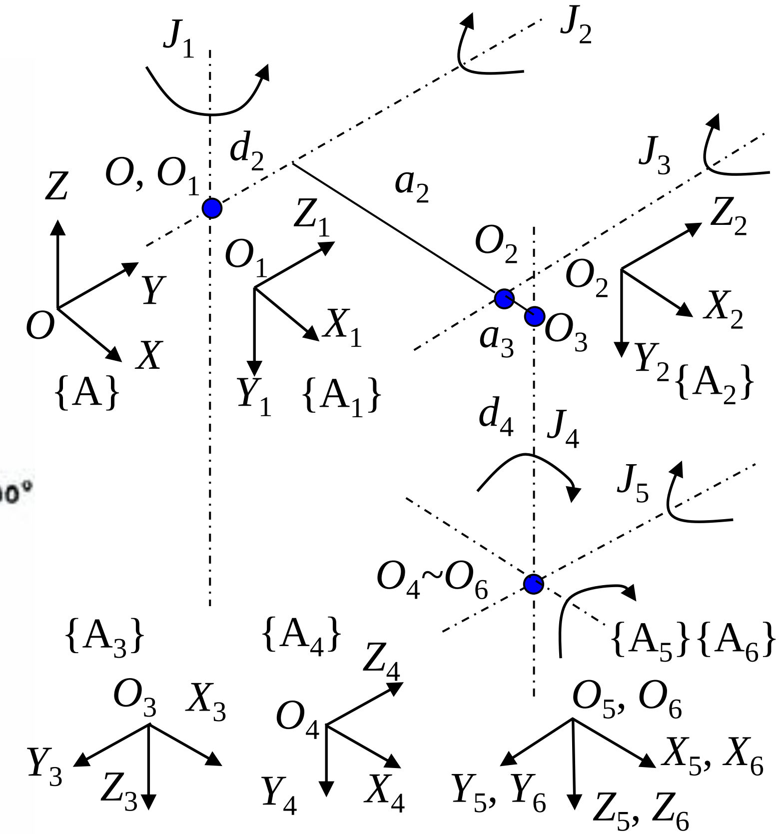
The PUMA 560 (Programmable Universal Machine for Assembly) is the industrial robot arm developed by Victor Scheinman at pioneering robot company Unimation and released in 1978. It is perhaps one of best known robots ever manufactured and was subject of the researches both in the academic institutions and industry.



3 Forward Kinematics for Spherical Coordinates Articulated Robot

3.1 PUMA560





Unimation PUMA560 Coordinates Establishment

Note no end-effector is attached to robot end; O overlaps O_1

3.1 Forward Kinematics for Spherical Coordinates Articulated Robot

- D-H Parameters：

Link	θ_i	α_i	a_i	d_i
1	θ_1	-90°	0	0
2	θ_2	0°	a_2	d_2
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

3.1 Forward Kinematics for Spherical Coordinates Articulated Robot

- D-H Matrix：

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

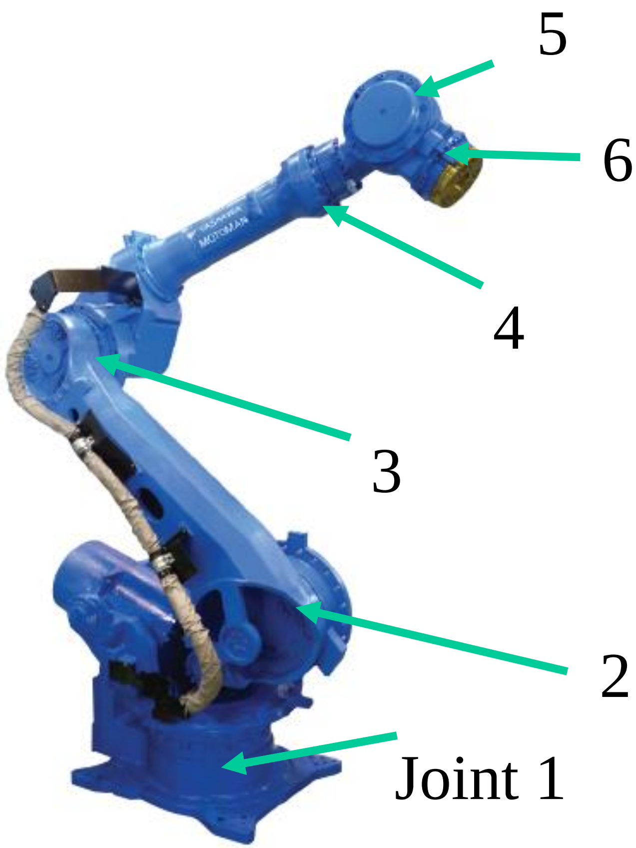
$$T_1 = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_3 = \begin{bmatrix} \cos\theta_3 & 0 & -\sin\theta_3 & a_3\cos\theta_3 \\ \sin\theta_3 & 0 & \cos\theta_3 & a_3\sin\theta_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Robot End Pose/Position: $T = T_1T_2T_3T_4T_5T_6$

3.2 Forward Kinematics for Yaskawa K10

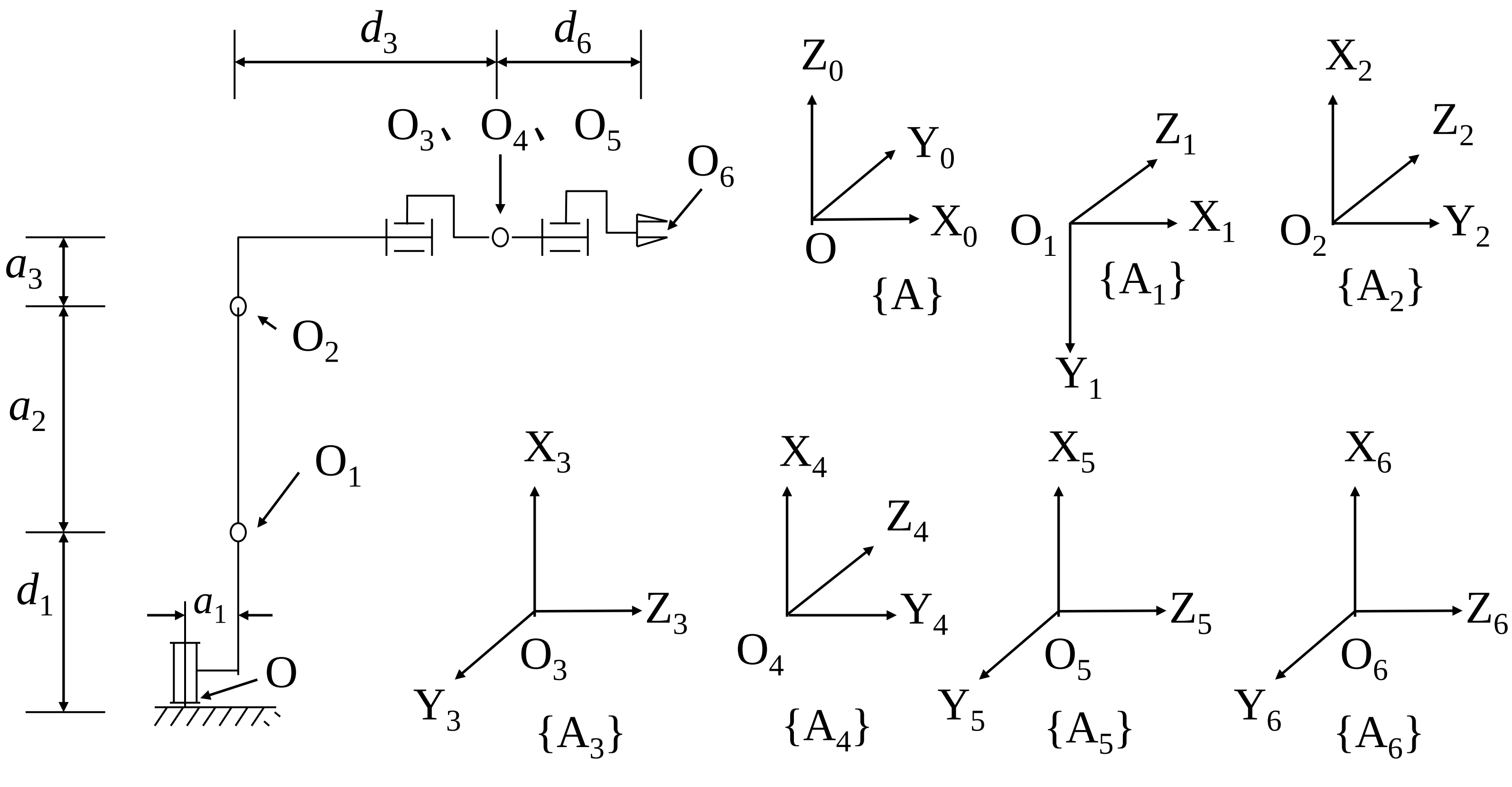
- Joint Configuration of Yaskawa K10



Yaskawa K10 in industry

3.2 Forward Kinematics for Yaskawa K10

- Coordinates Establishment



Note an end-effector is attached to robot end; *O* does not overlap O_1

3.2 Forward Kinematics for Yaskawa K10

- D-H Matrix

$$T_1 = \mathrm{Rot}(Z,\theta_1)\mathrm{Trans}(a_1,0,d_1)\mathrm{Rot}(X,-90°) = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & a_1\cos\theta_1 \\ \sin\theta_1 & 0 & \cos\theta_1 & a_1\sin\theta_1 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = \mathrm{Rot}(Z_1,-90°+\theta_2)\mathrm{Trans}(a_2,0,0) = \begin{bmatrix} \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ -\cos\theta_2 & \sin\theta_2 & 0 & -a_2\cos\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3 = \mathrm{Rot}(Z_2,\theta_3)\mathrm{Trans}(a_3,0,0)\mathrm{Rot}(X_2,-90°)\mathrm{Trans}(0,0,d_3) = \begin{bmatrix} \cos\theta_3 & 0 & -\sin\theta_3 & a_3\cos\theta_3 - d_3\sin\theta_3 \\ \sin\theta_3 & 0 & \cos\theta_3 & a_3\sin\theta_3 + d_3\cos\theta_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4 = \mathrm{Rot}(Z_3,\theta_4)\mathrm{Rot}(X_3,90°) = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 & 0 \\ \sin\theta_4 & 0 & -\cos\theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

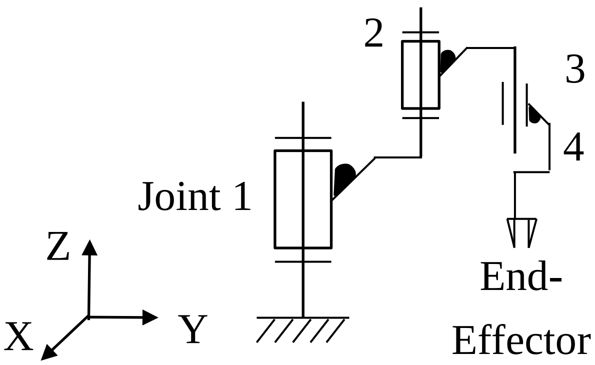
$$T_5 = \mathrm{Rot}(Z_4,\theta_5)\mathrm{Rot}(X_4,-90°) = \begin{bmatrix} \cos\theta_5 & 0 & -\sin\theta_5 & 0 \\ \sin\theta_5 & 0 & \cos\theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6 = \mathrm{Rot}(Z_5,\theta_6)\mathrm{Trans}(0,0,d_6) = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

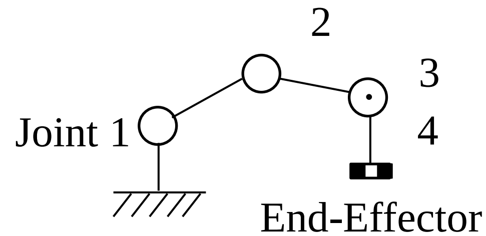
Robot End Pose/Position： $T = T_1T_2T_3T_4T_5T_6$

SCARA is an acronym for **S**elective **C**ompliance **A**rticulated **R**obot **A**rm

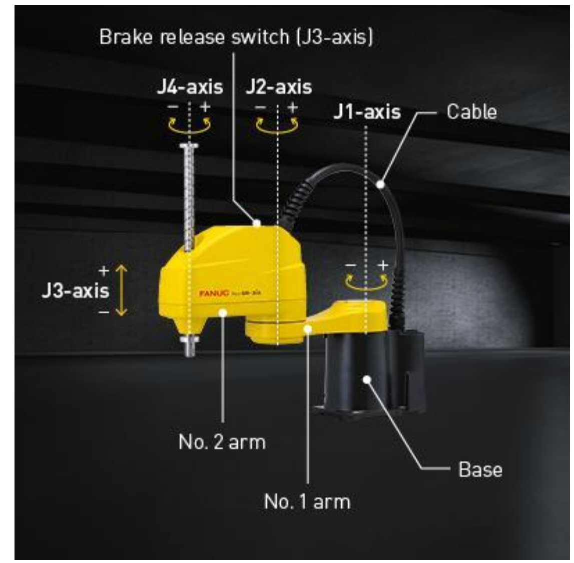
Cylindrical Coordinates Robot



Side View



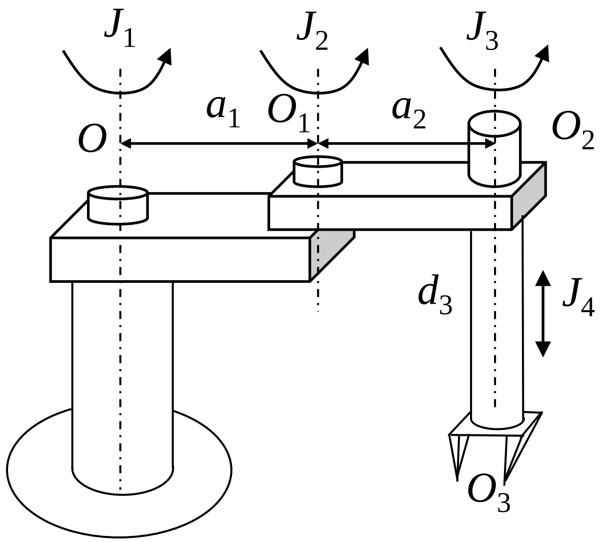
Top View

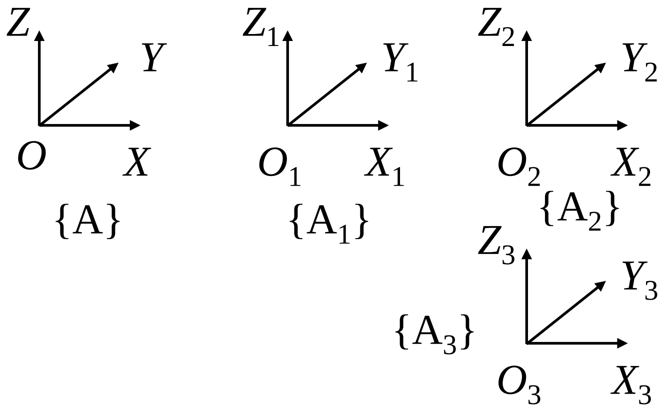


(1) All four joint axes are parallel; (2) Joint 1-3 can be viewed to be in the same surface; (3) End-effector's orientation and position is weakly correlated; Joint 4 determines the position along **Z**; Joint 3 Determines the orientation; Joint 1 and 2 determines the end-effectors position in **X-Y** plane.

4 Forward Kinematics for Cylindrical Coordinates Articulated Robot

SCARA Cylindrical Coordinates Robot

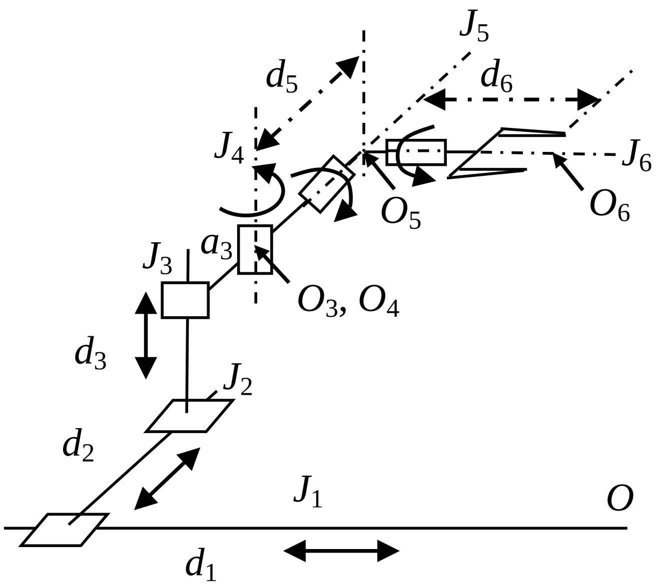


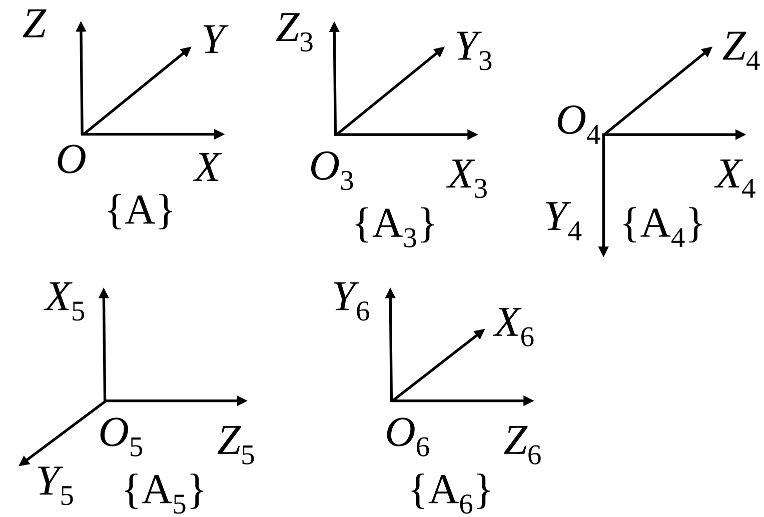


$$T_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & a_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = T_1T_2T_3 = \begin{bmatrix} \cos(\theta_1+\theta_2+\theta_3) & -\sin(\theta_1+\theta_2+\theta_3) & 0 & a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2+\theta_3) & \cos(\theta_1+\theta_2+\theta_3) & 0 & a_1\sin\theta_1 + a_2\sin(\theta_1+\theta_2) \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5 Forward Kinematics for Cartesian Coordinates Articulated Robots





$$T_3 = \begin{bmatrix} 1 & 0 & 0 & d_1 \\ 0 & 1 & 0 & d_2+a_3 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4 = \mathrm{Rot}(Z_3, \theta_4)\mathrm{Rot}(X_3, -90°) = \begin{bmatrix} \cos\theta_4 & 0 & -\sin\theta_4 & 0 \\ \sin\theta_4 & 0 & \cos\theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5 = \mathrm{Rot}(Z_4, -90°+\theta_5)\mathrm{Trans}(0,0,d_5)\mathrm{Rot}(X_4, -90°) = \begin{bmatrix} \sin\theta_5 & 0 & \cos\theta_5 & 0 \\ -\cos\theta_5 & 0 & \sin\theta_5 & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6 = \mathrm{Rot}(Z_5, -90°+\theta_6)\mathrm{Trans}(0,0,d_6) = \begin{bmatrix} \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ -\cos\theta_6 & \sin\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5 Forward Kinematics for Cartesian Coordinates Articulated Robots

$$T = T_3 T_4 T_5 T_6$$

$$= \begin{bmatrix} 1 & 0 & 0 & d_1 \\ 0 & 1 & 0 & d_2 + a_3 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_4 & 0 & -\sin\theta_4 & 0 \\ \sin\theta_4 & 0 & \cos\theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sin\theta_5 & 0 & \cos\theta_5 & 0 \\ -\cos\theta_5 & 0 & \sin\theta_5 & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ -\cos\theta_6 & \sin\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

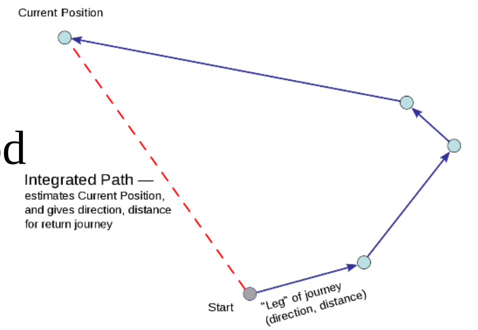
$$= \begin{bmatrix} \cos\theta_4 \sin\theta_5 & \sin\theta_4 & \cos\theta_4 \cos\theta_5 & d_1 - d_5 \sin\theta_4 \\ \sin\theta_4 \sin\theta_5 & -\cos\theta_4 & \sin\theta_4 \cos\theta_5 & d_2 + a_3 + d_5 \cos\theta_4 \\ \cos\theta_5 & 0 & -\sin\theta_5 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ -\cos\theta_6 & \sin\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_4 \sin\theta_5 \sin\theta_6 - \sin\theta_4 \cos\theta_6 & \cos\theta_4 \sin\theta_5 \cos\theta_6 + \sin\theta_4 \sin\theta_6 & \cos\theta_4 \cos\theta_5 & d_1 - d_5 \sin\theta_4 + d_6 \cos\theta_4 \cos\theta_5 \\ \sin\theta_4 \sin\theta_5 \sin\theta_6 + \cos\theta_4 \cos\theta_6 & \sin\theta_4 \sin\theta_5 \cos\theta_6 - \cos\theta_4 \sin\theta_6 & \sin\theta_4 \cos\theta_5 & d_2 + a_3 + d_5 \cos\theta_4 + d_6 \sin\theta_4 \cos\theta_5 \\ \cos\theta_5 \sin\theta_6 & \cos\theta_5 \cos\theta_6 & -\sin\theta_5 & d_3 - d_6 \sin\theta_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6 Kinematics based Localization for Mobile Robots

- For mobile robot, localization is fundamental for navigation.

- Two categories of mobile robot localization method:
  ✓ Absolute method: GPS, Landmark method, Triangulation Method
  ✓ Relative methods: Dead Reckoning



- Dead reckoning, a.k.a. pass integration, is the process of calculating current position of some moving object by using a previously determined position, and then incorporating estimations of speed, heading direction, and course over elapsed time.

- Characteristics:
  ✓ Dead reckoning is a kinematics based mobile robot localization method.
  ✓ Dead reckoning suffers from accumulated errors.
  ✓ Dead reckoning is suitable short term localization, like indoor localization.
  ✓ Dead reckoning is widely applied, like in **inertial navigation** system, to provide very accurate directional information.

6 Kinematics based Localization for Mobile Robots

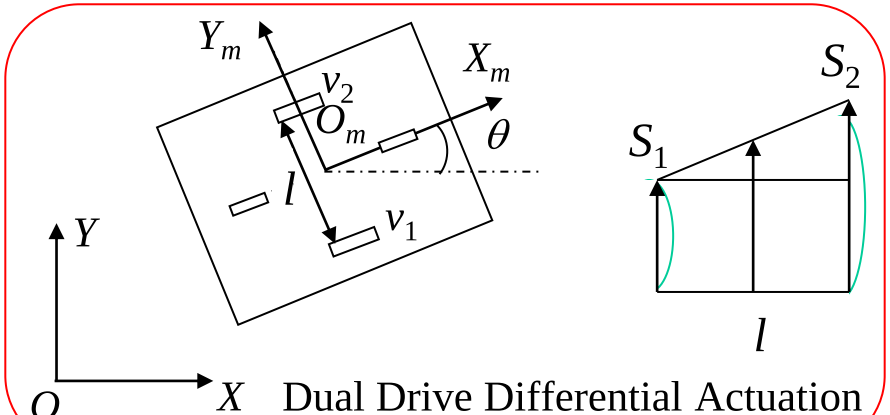
- Kinematics of a mobile robot is to **determine the pose**, including position and orientation of a mobile robot, along moving, which inherently localize a mobile robot.
- Kinematics of mobile robot is defined with respect to **Velocity** (*vs*. kinematics is defined with respect to joint space for articulated robot).
- Dead Reckoning Approach:

$$\begin{cases} \dot{x} = \dfrac{v_1 + v_2}{2}\cos\theta \\ \dot{y} = \dfrac{v_1 + v_2}{2}\sin\theta \\ \dot{\theta} = \dfrac{v_1 - v_2}{L} \end{cases}$$

Representation 1

$$\begin{cases} S = \dfrac{S_1 + S_2}{2} \\ \tan\theta = \dfrac{S_2 - S_1}{l} \end{cases}$$

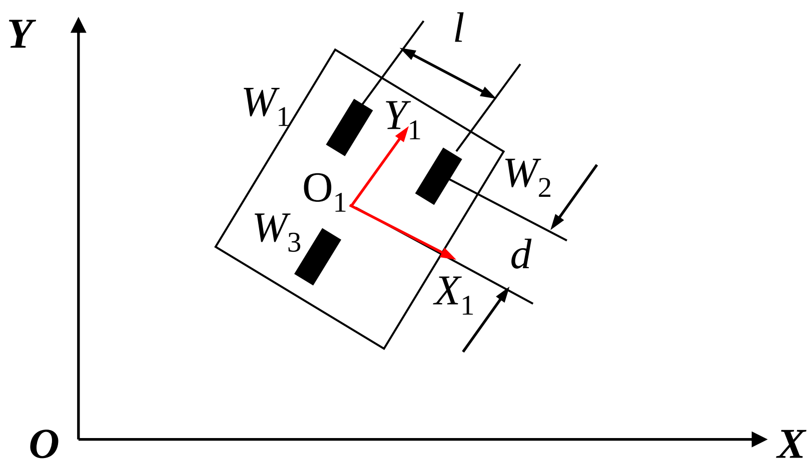
Representation 2



Dual Drive Differential Actuation

6 Kinematics based Localization for Mobile Robots

Dead Reckoning represented by Coordinate Transformation



$$T_1 = \begin{bmatrix} m_x & n_x & p_x \\ m_y & n_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

Initial Pose and Parameter Definition

6 Kinematics based Localization for Mobile Robots

Dead Reckoning represented by Coordinate Transformation

- Linear Motion

✓ The robot cannot move along X_1-axis.

✓ We are talking about differential motion.

$$T_{1M} = T_1 \ Trans(0, \Delta S)$$

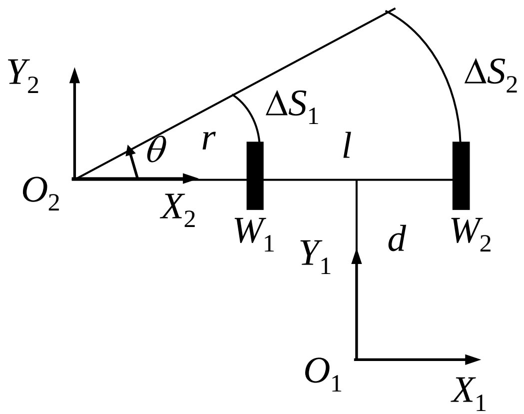
$$= \begin{bmatrix} m_x & n_x & p_x \\ m_y & n_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \Delta S \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} m_x & n_x & p_x + \Delta S n_x \\ m_y & n_y & p_y + \Delta S n_y \\ 0 & 0 & 1 \end{bmatrix}$$

6 Kinematics based Localization for Mobile Robots

Dead Reckoning represented by Coordinate Transformation

- Circular Motion case 1

$$sig\Delta S_1 = sig\Delta S_2 \quad |\Delta S_2| > |\Delta S_1|$$



Rotatory Center O_2 locates on the left side of W_1

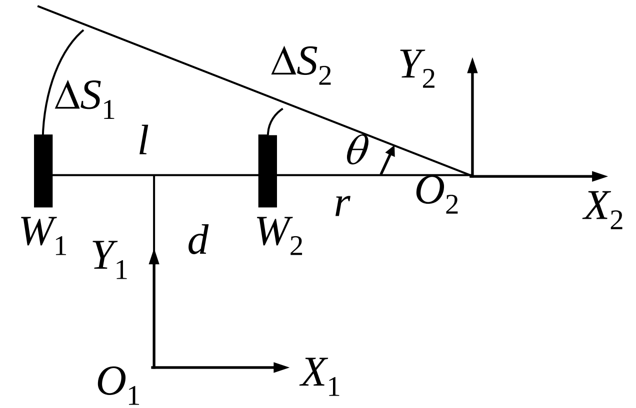
$$T_{1M} = T_2 \text{ Trans}(r + l/2, -d)$$
$$= T_1 \text{ Trans}(-r - l/2, d) \text{ Rot}(\theta) \text{ Trans}(r + l/2, -d)$$
$$= \begin{bmatrix} m_x & n_x & p_x \\ m_y & n_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -r - l/2 \\ 0 & 1 & d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & r + l/2 \\ 0 & 1 & -d \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} m_x\cos\theta + n_x\sin\theta & -m_x\sin\theta + n_x\cos\theta & (m_x\cos\theta + n_x\sin\theta - m_x)(r + l/2) + (m_x\sin\theta - n_x\cos\theta + n_x)d + p_x \\ m_y\cos\theta + n_y\sin\theta & -m_y\sin\theta + n_y\cos\theta & (m_y\cos\theta + n_y\sin\theta - m_y)(r + l/2) + (m_y\sin\theta - n_y\cos\theta + n_y)d + p_y \\ 0 & 0 & 1 \end{bmatrix}$$

6 Kinematics based Localization for Mobile Robots

Dead Reckoning represented by Coordinate Transformation

- Circular Motion case 2

$$sig\Delta S_1 = sig\Delta S_2 \qquad |\Delta S_2| < |\Delta S_1|$$



Rotatory center O_2 locates on right side of W_2

$$T_{1M} = T_2 \text{ Trans}(-r - l/2, -d)$$

$$= T_1 \text{ Trans}(r + l/2, d) \text{ Rot}(\theta) \text{ Trans}(-r - l/2, -d)$$

$$= \begin{bmatrix} m_x & n_x & p_x \\ m_y & n_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & r + l/2 \\ 0 & 1 & d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -r - l/2 \\ 0 & 1 & -d \\ 0 & 0 & 1 \end{bmatrix}$$

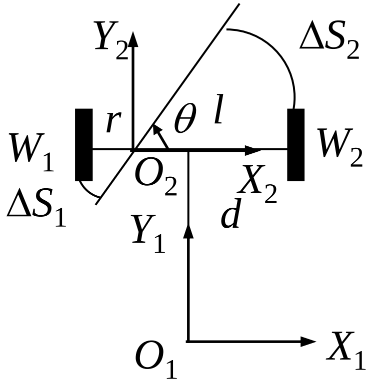
$$= \begin{bmatrix} m_x\cos\theta + n_x\sin\theta & -m_x\sin\theta + n_x\cos\theta & (m_x\cos\theta + n_x\sin\theta - m_x)(-r - l/2) + (m_x\sin\theta - n_x\cos\theta + n_x)d + p_x \\ m_y\cos\theta + n_y\sin\theta & -m_y\sin\theta + n_y\cos\theta & (m_y\cos\theta + n_y\sin\theta - m_y)(-r - l/2) + (m_y\sin\theta - n_y\cos\theta + n_y)d + p_y \\ 0 & 0 & 1 \end{bmatrix}$$

6 Kinematics based Localization for Mobile Robots

Dead Reckoning represented by Coordinate Transformation

- Circular Motion case 3

$$sig\Delta S_1 = -sig\Delta S_2$$



Rotatory center O_2 locates between W_1 and W_2

$$T_{1M} = T_2 \text{ Trans}(-r + l/2, -d)$$

$$= T_1 \text{ Trans}(r - l/2, d) \text{ Rot}(\theta) \text{ Trans}(-r + l/2, -d)$$

$$= \begin{bmatrix} m_x & n_x & p_x \\ m_y & n_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & r - l/2 \\ 0 & 1 & d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -r + l/2 \\ 0 & 1 & -d \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} m_x \cos\theta + n_x \sin\theta & -m_x \sin\theta + n_x \cos\theta & (m_x \cos\theta + n_x \sin\theta - m_x)(-r + l/2) + (m_x \sin\theta - n_x \cos\theta + n_x)d + p_x \\ m_y \cos\theta + n_y \sin\theta & -m_y \sin\theta + n_y \cos\theta & (m_y \cos\theta + n_y \sin\theta - m_y)(-r + l/2) + (m_y \sin\theta - n_y \cos\theta + n_y)d + p_y \\ 0 & 0 & 1 \end{bmatrix}$$

Appendix

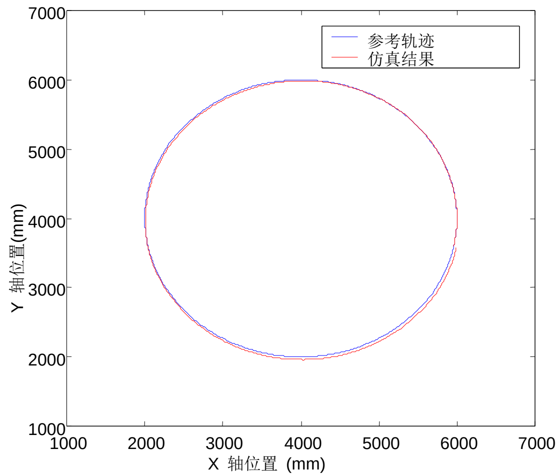
An Improved Dead Reckoning Method for Mobile Robot with Redundant Odometry Information

- De Xu, Min Tan, Gang Chen
- Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, P.R.C.
- By visiting the following URL through campus wifi

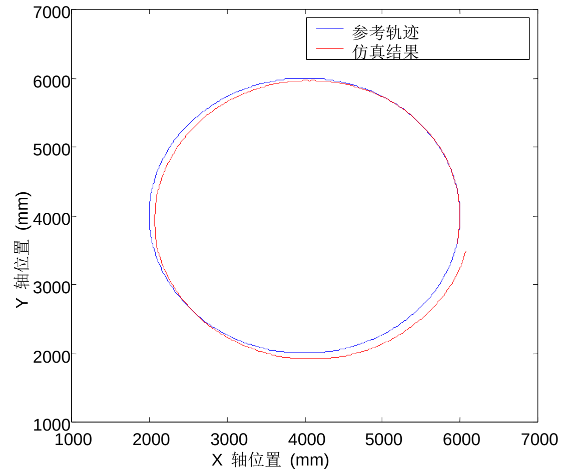
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1238497

Appendix-Simulation Results

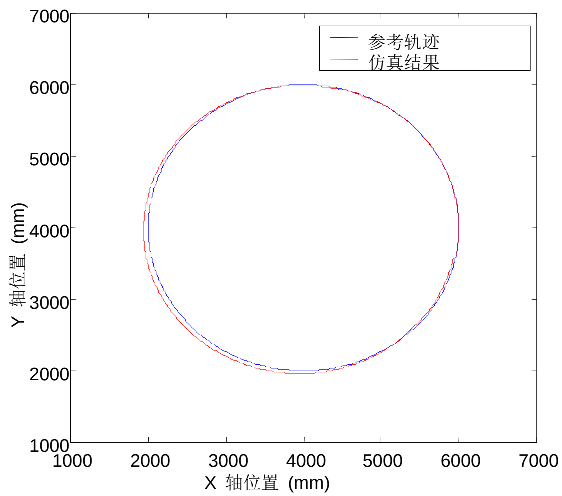
Triple wheel mobile robot parameters： *l*=600mm, *d*=400mm, *h*=800mm. Circular movement radius: 2000mm. Noise are added in simulation.



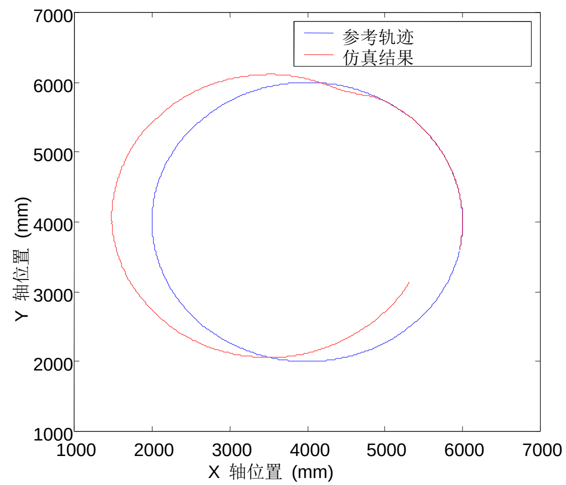
(a) 利用 W1~W3 数据获得的实验结果



(b) 只利用 W1 和 W2 信息的仿真结果



(c) W1 遇到凸起时利用 W1~W3 数据的仿真结果



(d) W1 遇到凸起时利用 W1 和 W2 数据的仿真结果

THANK YOU