Tell me and I will forget.
Show me and I will remember.
Involve me and I will understand.

Chinese proverb

What is Kinematics

Review

- What is a robot?
  - By general agreement a robot is:
    - A programmable machine that imitates the actions or appearance of an intelligent creature–usually a human.
  - To qualify as a robot, a machine must be able to:

  1) Sensing and perception: get information from its surroundings
  2) Carry out different tasks: Locomotion or manipulation, do something physical–such as move or manipulate objects
  3) Reprogrammable: can do different things
  4) Function autonomously and/or interact with human beings
- Why use robots?
  - Perform 4A tasks in 4D environments

    4A: Automation, Augmentation, Assistance, Autonomous

    4D: Dangerous, Dirty, Dull, Difficult

Kinematics studies the motion of bodies

Joints for Robots

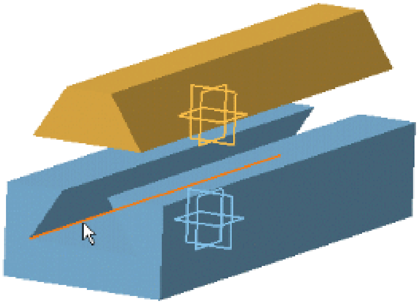
Manipulators

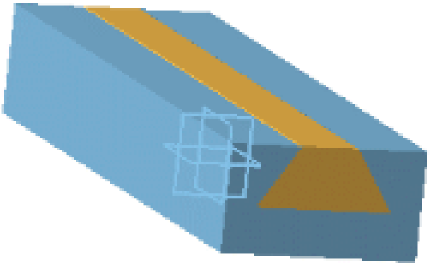
- Robot arms, industrial robot
  - Rigid bodies (links) connected by joints
  - Joints: revolute or prismatic
  - Drive: electric or hydraulic
  - End−effector (tool) mounted on a flange or plate secured to the wrist joint of robot

Robot Joints

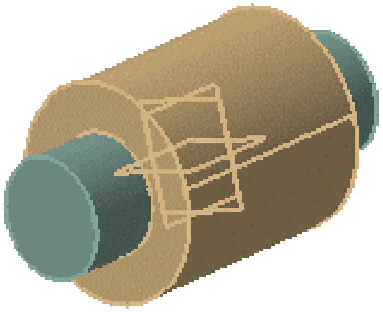
Prismatic Joint: Linear, No rotation involved.

(Hydraulic or pneumatic cylinder)



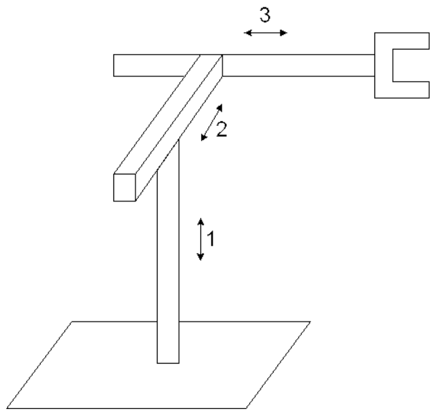


Revolute Joint: Rotary, (electrically driven with stepper motor, servo motor)

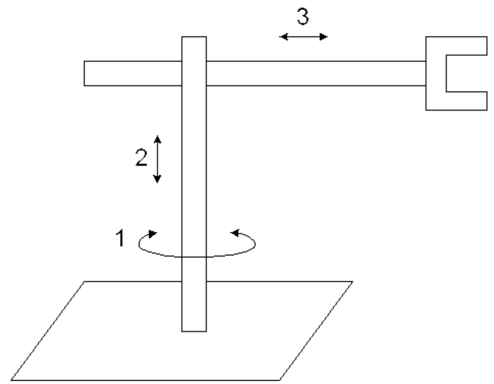


Manipulators

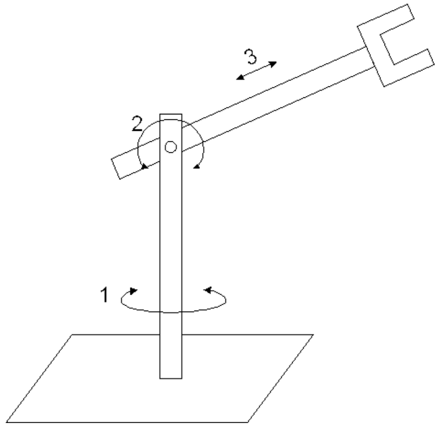
- Robot Configuration:



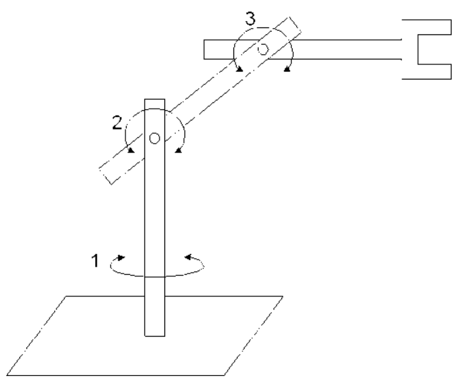
Cartesian: PPP



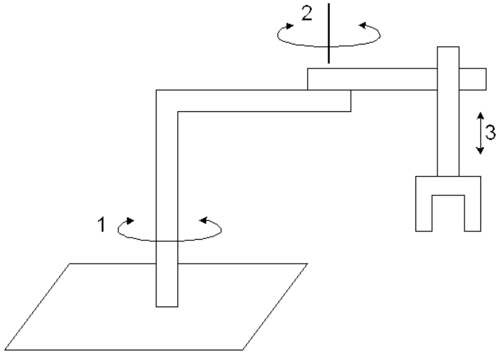
Cylindrical: RPP



Spherical: RRP

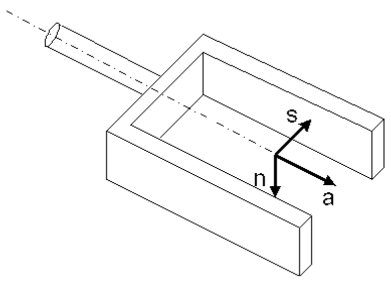


Articulated: RRR



SCARA: RRP

(Selective Compliance Assembly Robot Arm)



Hand coordinate:

n: normal vector; **s**: sliding vector;

a: approach vector, normal to the tool mounting plate

Manipulators

- Motion Control Methods
  - Point to point control
    - a sequence of discrete points
    - spot welding, pick−and−place, loading & unloading
  - Continuous path control
    - follow a prescribed path, controlled−path motion
    - Spray painting, Arc welding, Gluing

Manipulators

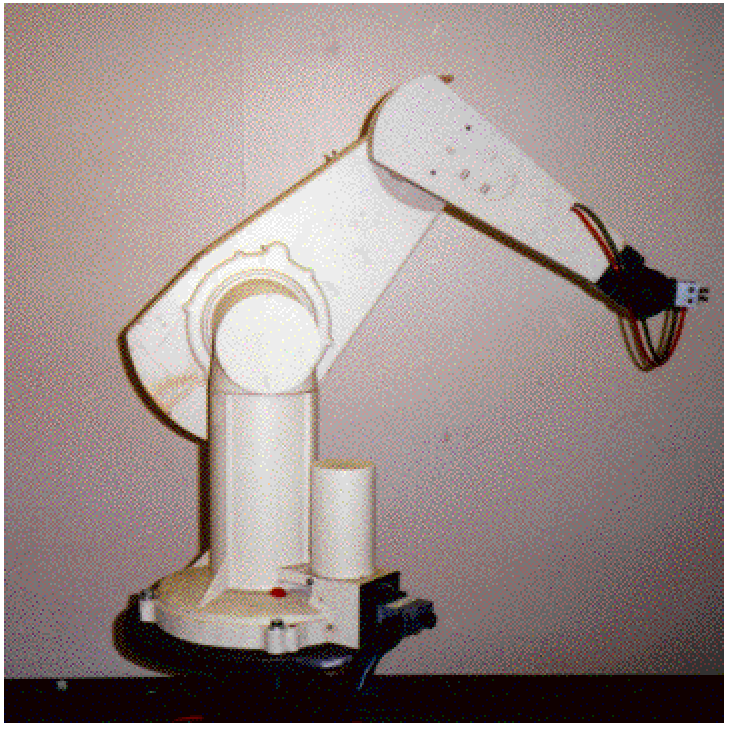
- Robot Specifications
  - Number of Axes
    - Major axes, (1−3) => Position the wrist
    - Minor axes, (4−6) => Orient the tool
    - Redundant, (7−n) => reaching around obstacles, avoiding undesirable configuration
  - Degree of Freedom (DOF)
  - Workspace
  - Payload (load capacity)
  - **Precision** vs. **Repeatability**

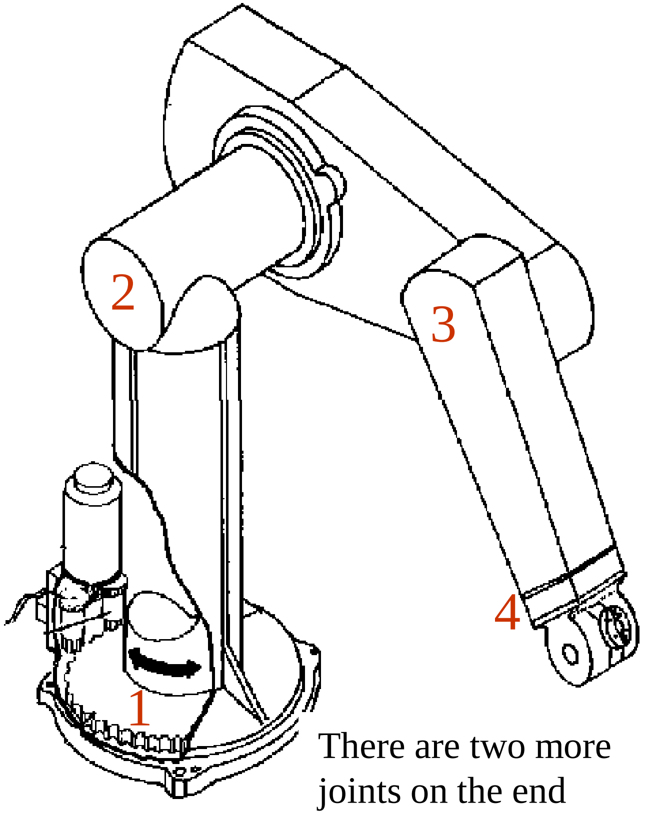
how accurately a specified point can be reached

how accurately the same position can be reached if the motion is repeated many times

Which one is more important?

An Example – The PUMA 560





There are two more joints on the end effector (the gripper)

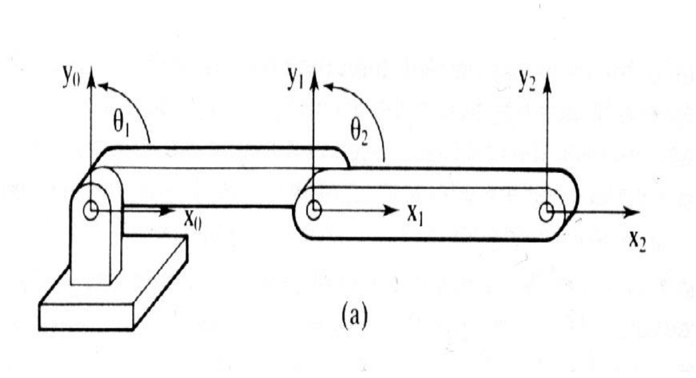
The PUMA 560 has SIX revolute joints

A revolute joint has ONE degree of freedom (1 DOF) that is defined by its angle

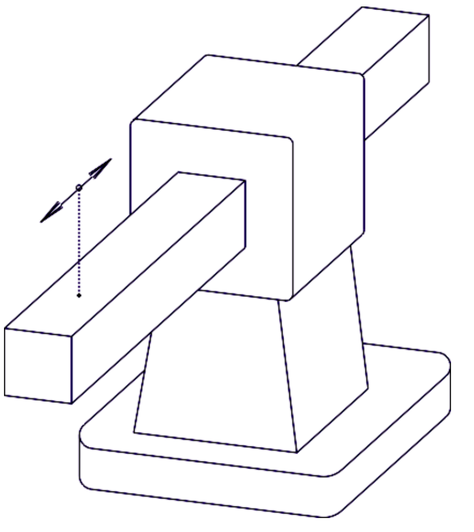
Concepts:

- Revolute joint
- DOF

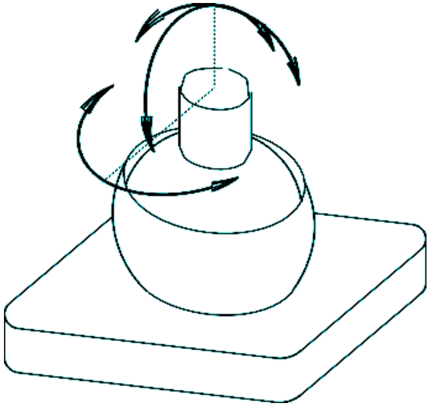
Other basic joints



Revolute Joint
1 DOF (Variable – Y)



Prismatic Joint
1 DOF (linear) (Variables – d)



Spherical Joint
3 DOF (Variables – Y_1, Y_2, Y_3)

Concepts:
- **Prismatic joint**
- **Spherical joint**

We are interested in two kinematics topics

Forward Kinematics (angles to position)

What you are given: The length of each link
The angle of each joint

What you can find: The position of any point
(i.e. its (x, y, z) coordinates

Given the angles, locate the tool tip position

Inverse Kinematics (position to angles)

What you are given: The length of each link
The position of some point on the robot

What you can find: The angles of each joint needed to obtain that position

Given the tool tip position, determine the joints angles

Concepts:
- Forward Kinematics
- Inverse Kinematics

- **Forward Kinematics**:
  to determine where the robot's hand is?
  (If all joint variables are known)

- **Inverse Kinematics:**
  to calculate what each joint variable is?
  (If we desire that the hand be located at a particular point)

Kinematic Problems for Manipulation

- **Reliably <u>position the tip</u> – go from one position to another position**

- **<u>Don't hit</u> anything, <u>avoid obstacles</u>**

- **Make <u>smooth motions</u>**
  - **at <u>reasonable speeds</u> and**
  - **at <u>reasonable accelerations</u>**

- **<u>Adjust to changing</u> conditions –**
  - **i.e. when something is picked up *respond to the change in weight***

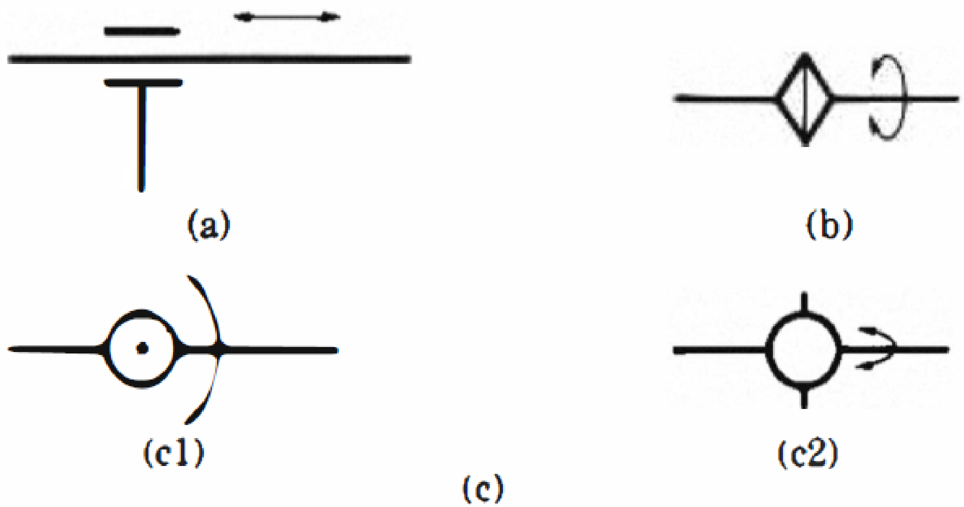
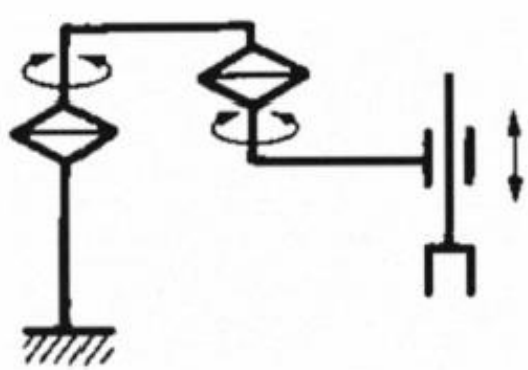


Figure 1.2
Symbols of joints (arrows show direction of motion). (a) Prismatic joint. (b) Revolute joint 1. (c) Revolute joint 2. (c1) Up-and-down rotation. (c2) Back-and-forth rotation.



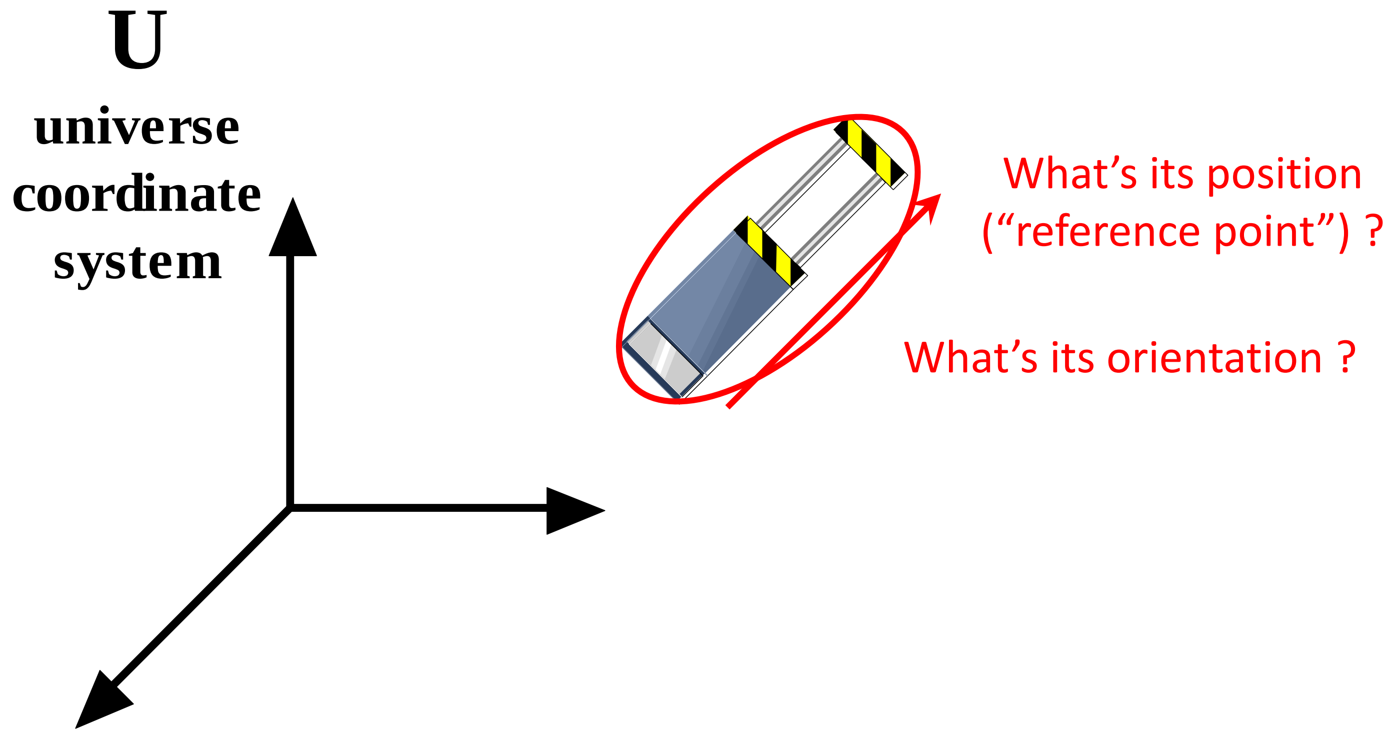
SCARA

Spatial description and transformation

- We need to be able to describe the position and the orientation of the robot's parts
- Suppose there's a *universe coordinate system* to which everything can be referenced.

Spatial description and transformation

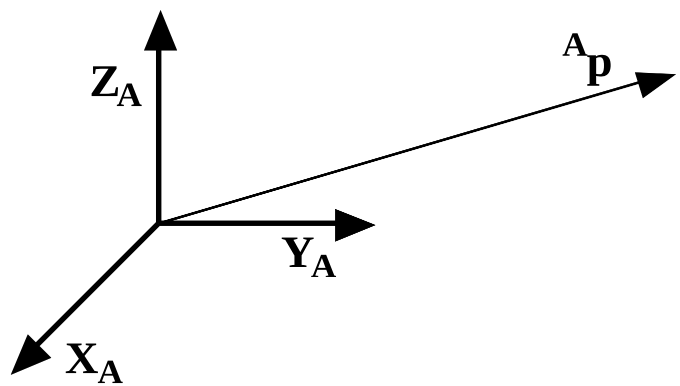
- We need to be able to describe the position and the orientation of the robot's parts (relative to *U*)



Positions, orientations and frames

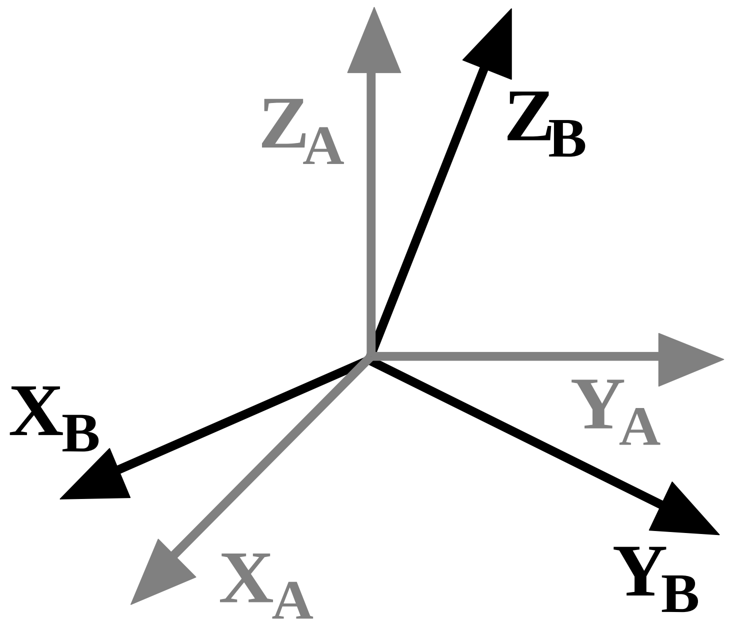
- The *position* of a point p relative to a coordinate system A (^{A}p):

$$^{A}p = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}$$



Positions, orientations and frames

- The *orientation* of a body is described by a coordinate system *B* attached to the body, relative to *A* (a known coordinate system).



Positions, orientations and frames

- The *orientation* of a body is described by a coordinate system *B* attached to the body, relative to *A* (a known coordinate system).

$$
{}^{A}_{B}R = \begin{bmatrix} {}^{A}X_{B} & {}^{A}Y_{B} & {}^{A}Z_{B} \end{bmatrix}
= \begin{bmatrix} X_{B}\cdot X_{A} & Y_{B}\cdot X_{A} & Z_{B}\cdot X_{A} \\ X_{B}\cdot Y_{A} & Y_{B}\cdot Y_{A} & Z_{B}\cdot Y_{A} \\ X_{B}\cdot Z_{A} & Y_{B}\cdot Z_{A} & Z_{B}\cdot Z_{A} \end{bmatrix}
$$

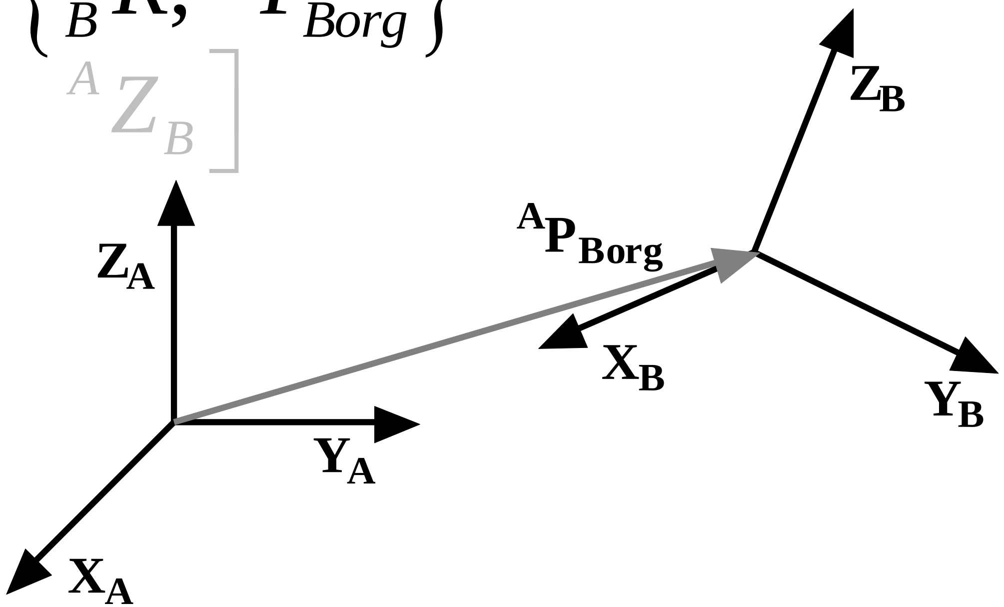
cosine of the angle

Positions, orientations and frames

- A frame is a set of 4 vectors giving the position and orientation.
- Example: frame *B*

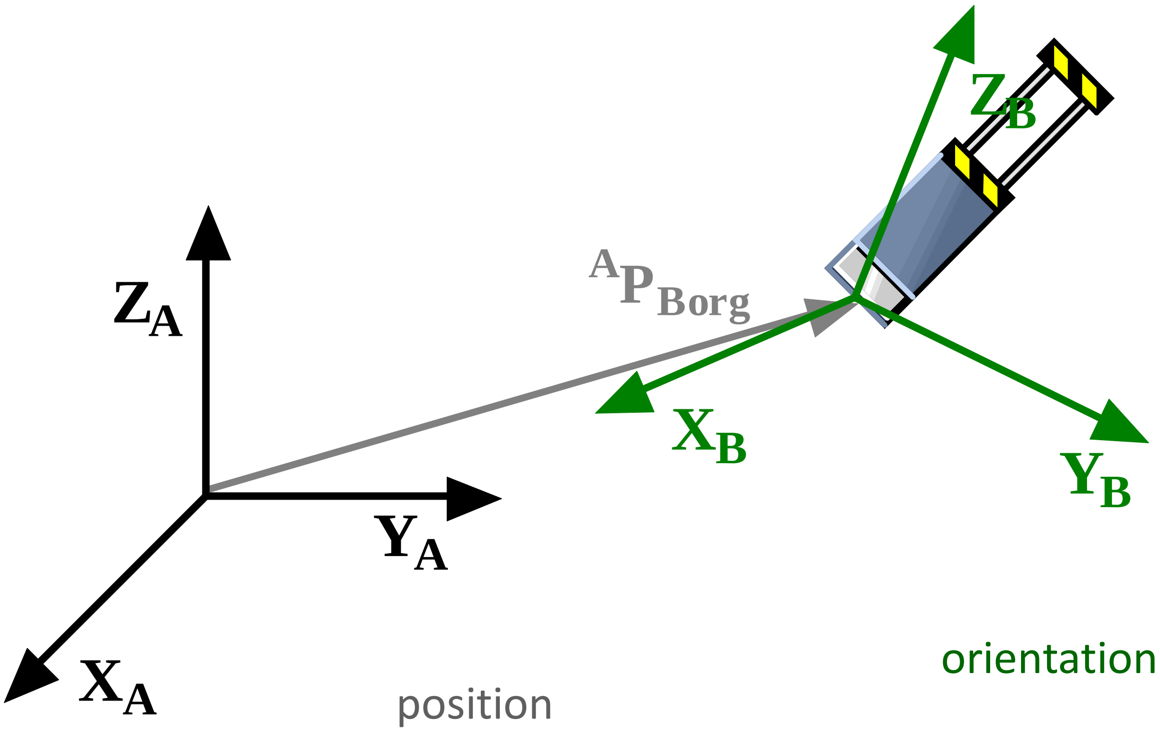
$$\{B\} = \left\{ {}^{A}_{B}R,\ {}^{A}P_{Borg} \right\}$$

$$\begin{bmatrix} {}^{A}X_{B} & {}^{A}Y_{B} & {}^{A}Z_{B} \end{bmatrix}$$

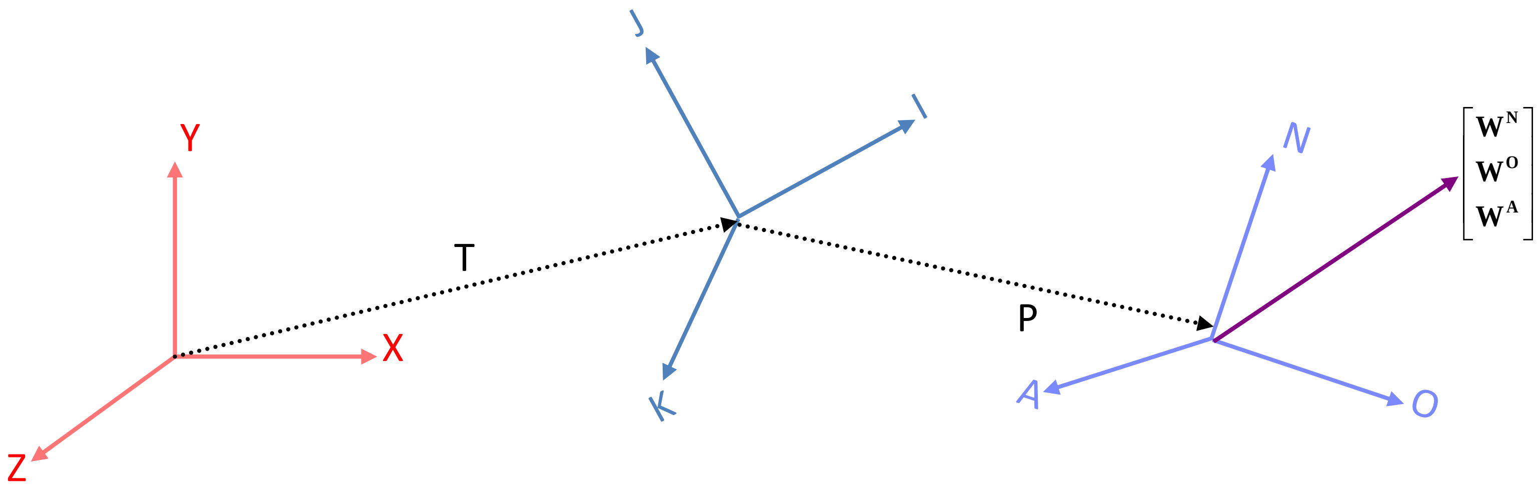


Positions, orientations and frames

- Remember the robot's part:



Concatenation of numerous translations and rotations



H = (Rotate so that the X−axis is aligned with T)

* (Translate along the new t−axis by || T || (magnitude of T))

* (Rotate so that the t−axis is aligned with P)

* (Translate along the p−axis by || P ||)

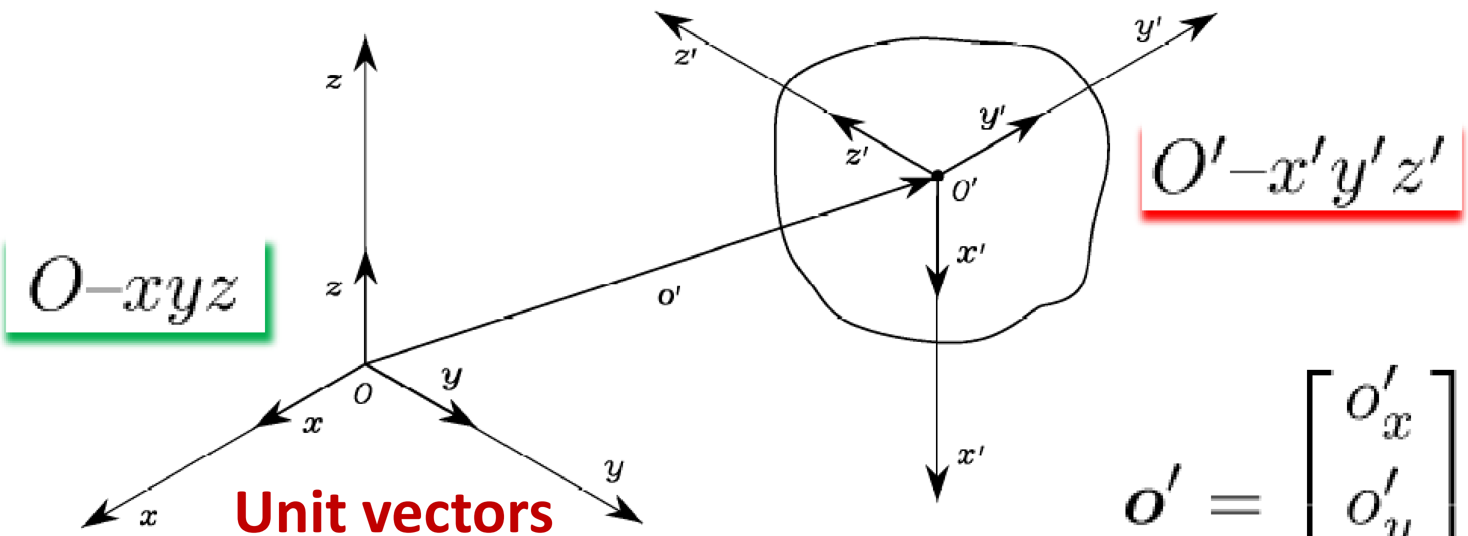
* (Rotate so that the p−axis is aligned with the O−axis)

Mapping

- Until now, we saw how to describe positions, orientations and frames.
- We need to be able to change descriptions from one frame to another: *mapping*.
- Mappings:
  - translated frames
  - rotated frames
  - general frames

A rigid body in space

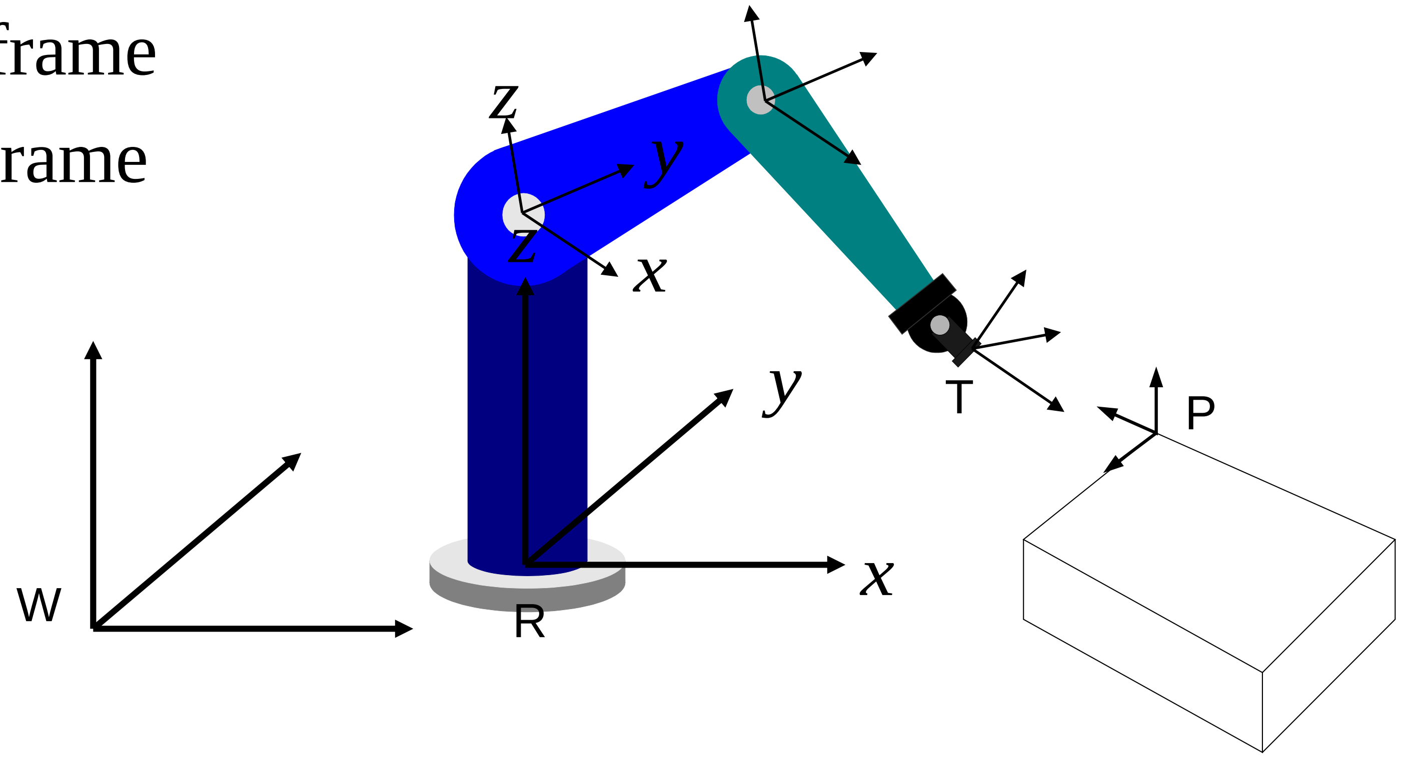
- A *rigid body* is completely described in space by its *position* and *orientation*



$$\boldsymbol{o}' = \begin{bmatrix} o'_x \\ o'_y \\ o'_z \end{bmatrix}$$

Preliminary

- Robot Reference Frames
  - World frame
  - Joint frame
  - Tool frame



Preliminary

- Coordinate Transformation
  - Reference coordinate frame Oxyz
  - Body–attached frame O'uvw

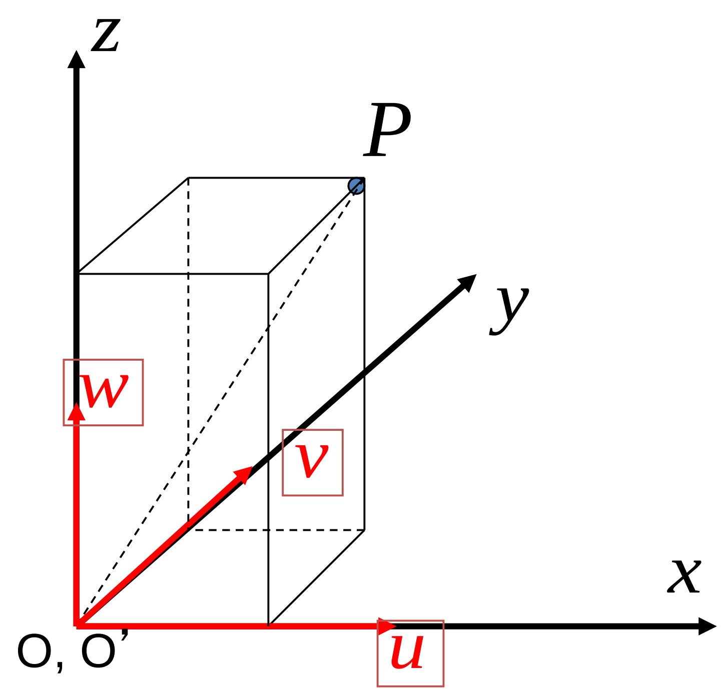
Point represented in Oxyz:

$$P_{xyz} = [p_x, p_y, p_z]^T$$

$$\vec{P}_{xyz} = p_x \mathrm{i_x} + p_y \mathrm{j_y} + p_z \mathrm{k_z}$$

Point represented in O' uvw:

$$\vec{P}_{uvw} = p_u \mathrm{i_u} + p_v \mathrm{j_v} + p_w \mathrm{k_w}$$



Two frames coincide ==> $p_u = p_x \quad p_v = p_y \quad p_w = p_z$

Preliminary

Properties: Dot Product

Let x and y be arbitrary vectors in R^3 and θ be the angle from x to y, then

$$x \cdot y = |x||y| \cos\theta$$

Properties of orthonormal coordinate frame

- Mutually perpendicular

$$\vec{i} \cdot \vec{j} = 0$$

$$\vec{i} \cdot \vec{k} = 0$$

$$\vec{k} \cdot \vec{j} = 0$$

- Unit vectors

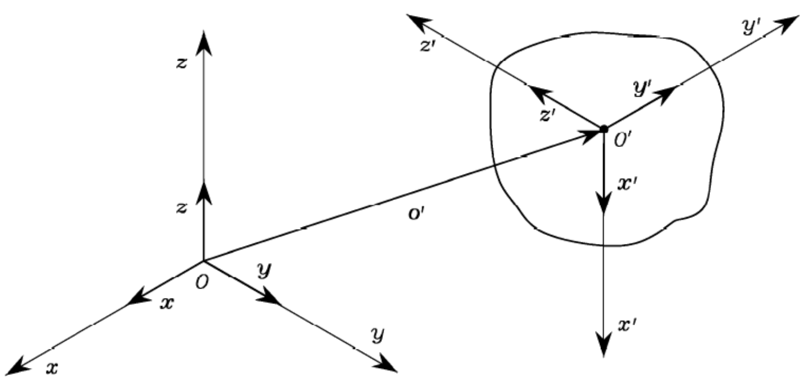
$$|\vec{i}| = 1$$

$$|\vec{j}| = 1$$

$$|\vec{k}| = 1$$

The components of each unit vector are the direction cosines of the axes of frame $O'-x'y'z'$ with respect to the reference frame $O-xyz$.

$$\boldsymbol{x}' = x'_x\boldsymbol{x} + x'_y\boldsymbol{y} + x'_z\boldsymbol{z}$$
$$\boldsymbol{y}' = y'_x\boldsymbol{x} + y'_y\boldsymbol{y} + y'_z\boldsymbol{z}$$
$$\boldsymbol{z}' = z'_x\boldsymbol{x} + z'_y\boldsymbol{y} + z'_z\boldsymbol{z}.$$



Rotation Matrix

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{x}' & \boldsymbol{y}' & \boldsymbol{z}' \end{bmatrix} = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}'^T\boldsymbol{x} & \boldsymbol{y}'^T\boldsymbol{x} & \boldsymbol{z}'^T\boldsymbol{x} \\ \boldsymbol{x}'^T\boldsymbol{y} & \boldsymbol{y}'^T\boldsymbol{y} & \boldsymbol{z}'^T\boldsymbol{y} \\ \boldsymbol{x}'^T\boldsymbol{z} & \boldsymbol{y}'^T\boldsymbol{z} & \boldsymbol{z}'^T\boldsymbol{z} \end{bmatrix}$$

Properties:

$$\boldsymbol{x}'^T\boldsymbol{y}' = 0 \qquad \boldsymbol{y}'^T\boldsymbol{z}' = 0 \qquad \boldsymbol{z}'^T\boldsymbol{x}' = 0.$$

$$\boldsymbol{x}'^T\boldsymbol{x}' = 1 \qquad \boldsymbol{y}'^T\boldsymbol{y}' = 1 \qquad \boldsymbol{z}'^T\boldsymbol{z}' = 1$$

As a consequence, $\boldsymbol{R}$ is an *orthogonal* matrix meaning that

$$\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}_3 \quad \rightarrow \quad \boldsymbol{R}^T = \boldsymbol{R}^{-1}$$

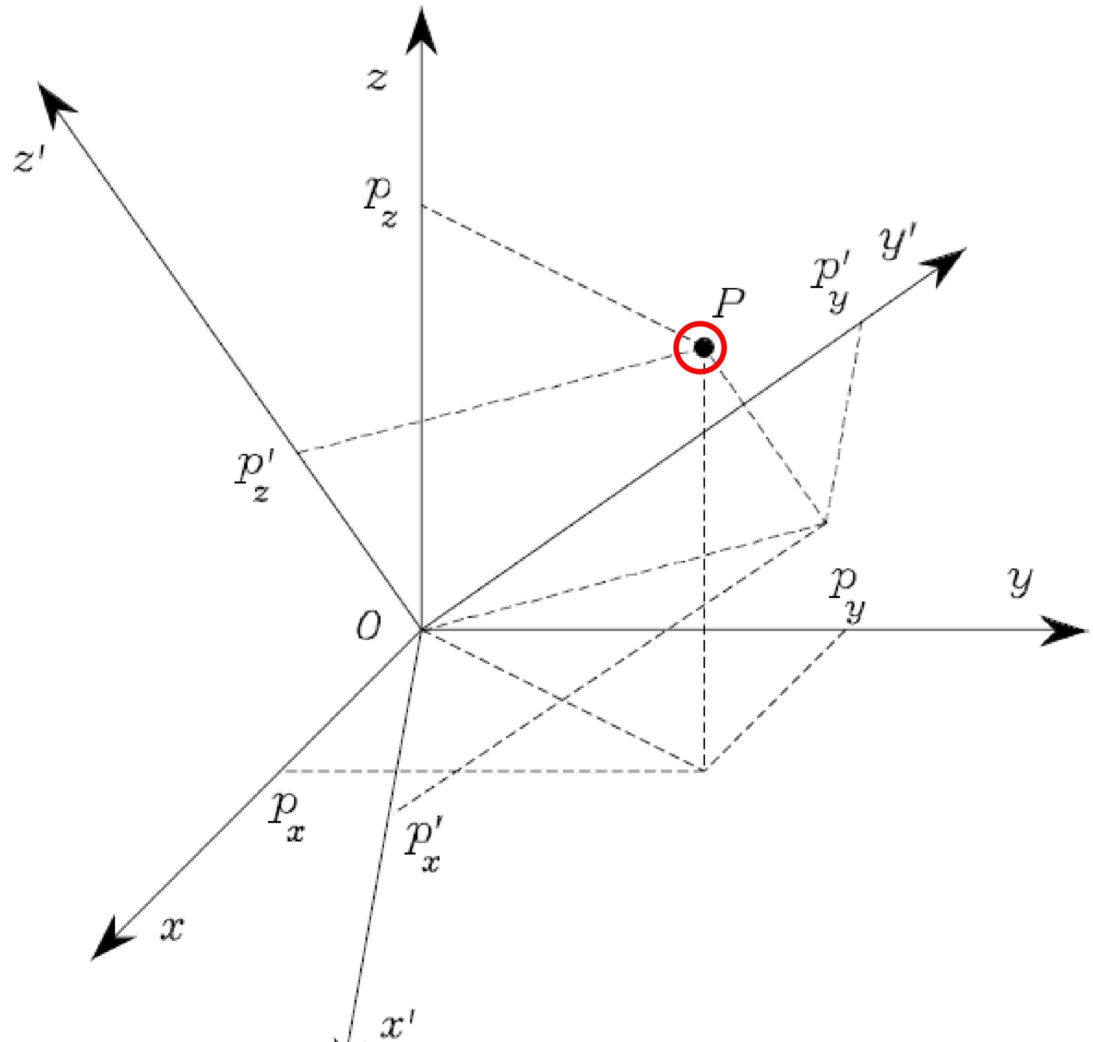
$\det(\boldsymbol{R}) = 1$ if the frame is right-handed

$\det(\boldsymbol{R}) = -1$ if the frame is left-handed

Representation of a Vector

$$\boldsymbol{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

$$\boldsymbol{p}' = \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix}$$



Representation of a Vector

- Since ***p*** and ***p'*** are representations of the same point *P*, it is

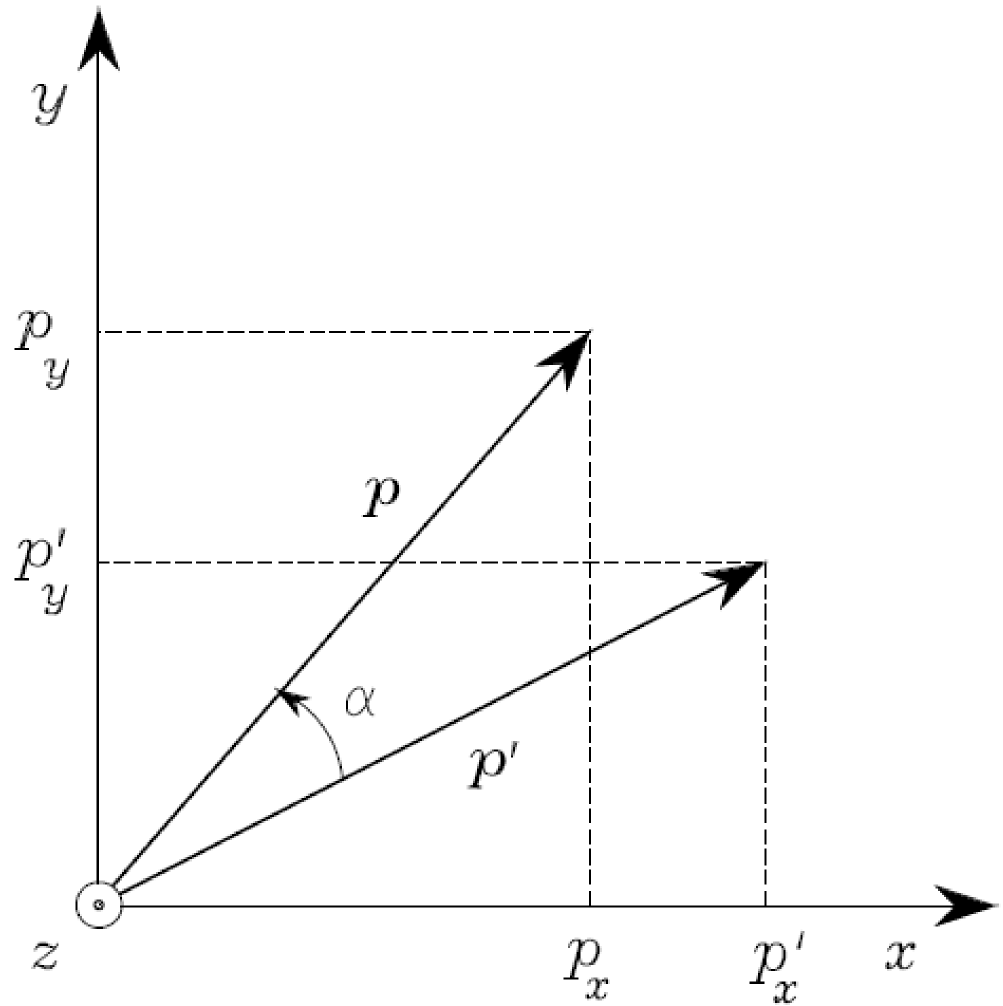
$$\boldsymbol{p} = p'_x \boldsymbol{x}' + p'_y \boldsymbol{y}' + p'_z \boldsymbol{z}' = \begin{bmatrix} \boldsymbol{x}' & \boldsymbol{y}' & \boldsymbol{z}' \end{bmatrix} \boldsymbol{p}'$$

$$\boldsymbol{p} = \boldsymbol{R}\boldsymbol{p}'.$$

$$\boldsymbol{p}' = \boldsymbol{R}^T \boldsymbol{p}.$$

Rotation of a Vector

- Point “=“ Vector



Rotation matrix: *equivalent geometrical meanings*

- It describes the mutual orientation between two coordinate frames; its column vectors are the direction cosines of the axes of the rotated frame with respect to the original frame.
- It represents the coordinate transformation between the coordinates of a point expressed in two different frames (with common origin).
- It is the operator that allows the rotation of a vector in the same coordinate frame.

Rotation Matrices

- $$\boldsymbol{R}_i^j = (\boldsymbol{R}_j^i)^{-1} = (\boldsymbol{R}_j^i)^T$$

- Successive rotations can be also specified by constantly referring them to the initial frame; in this case, the rotations are made with respect to a *fixed frame*.

Vector rotation (fixed frame)

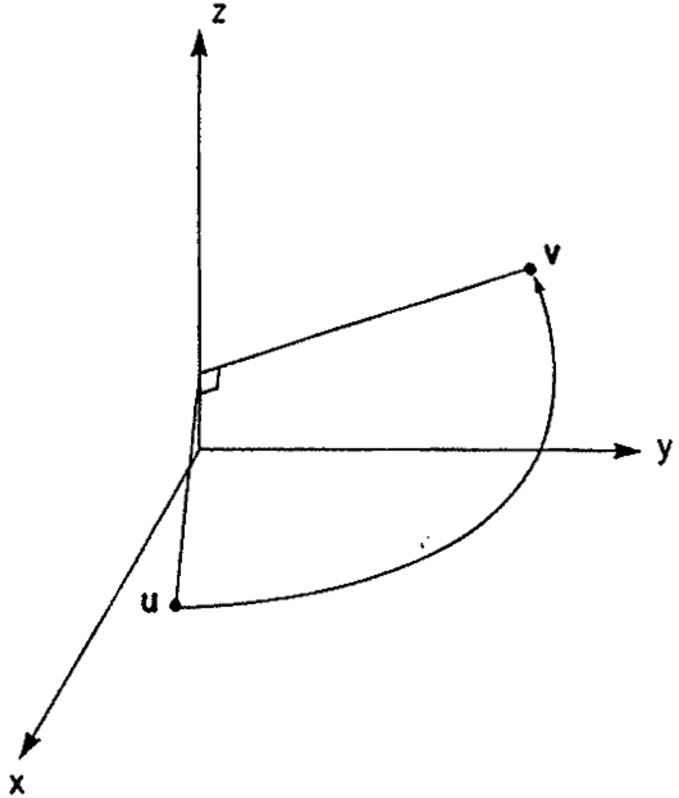


Figure 1.1. Rot(z, 90)

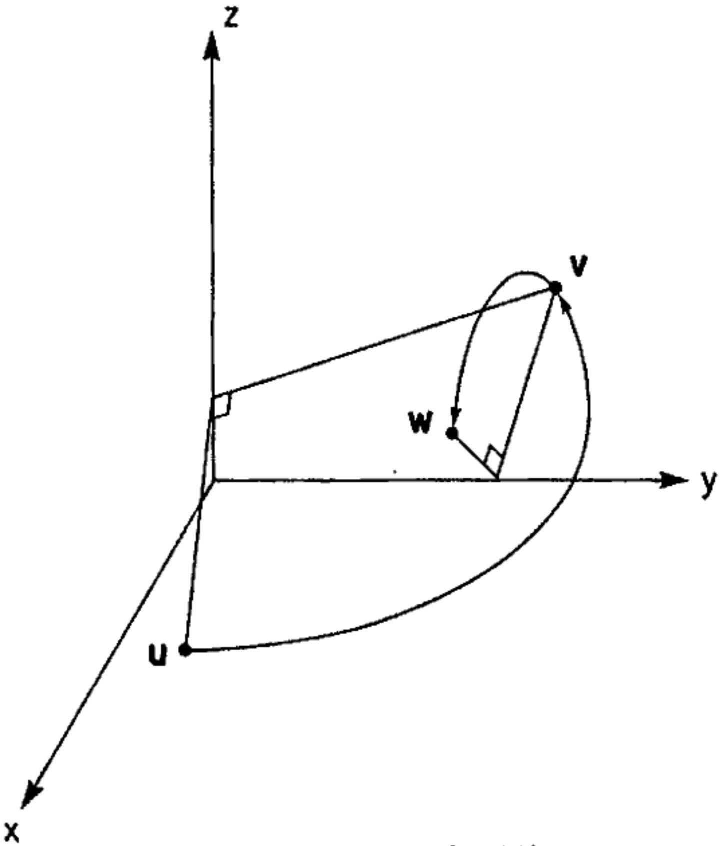


Figure 1.2. Rot(y, 90)

$$\mathbf{v} = \text{Rot}(z, 90)\mathbf{u}$$

$$\mathbf{w} = \text{Rot}(y, 90)\mathbf{v}$$

$$\mathbf{w} = \text{Rot}(y, 90)\ \text{Rot}(z, 90)\ \mathbf{u}$$

Vector rotation: order is important

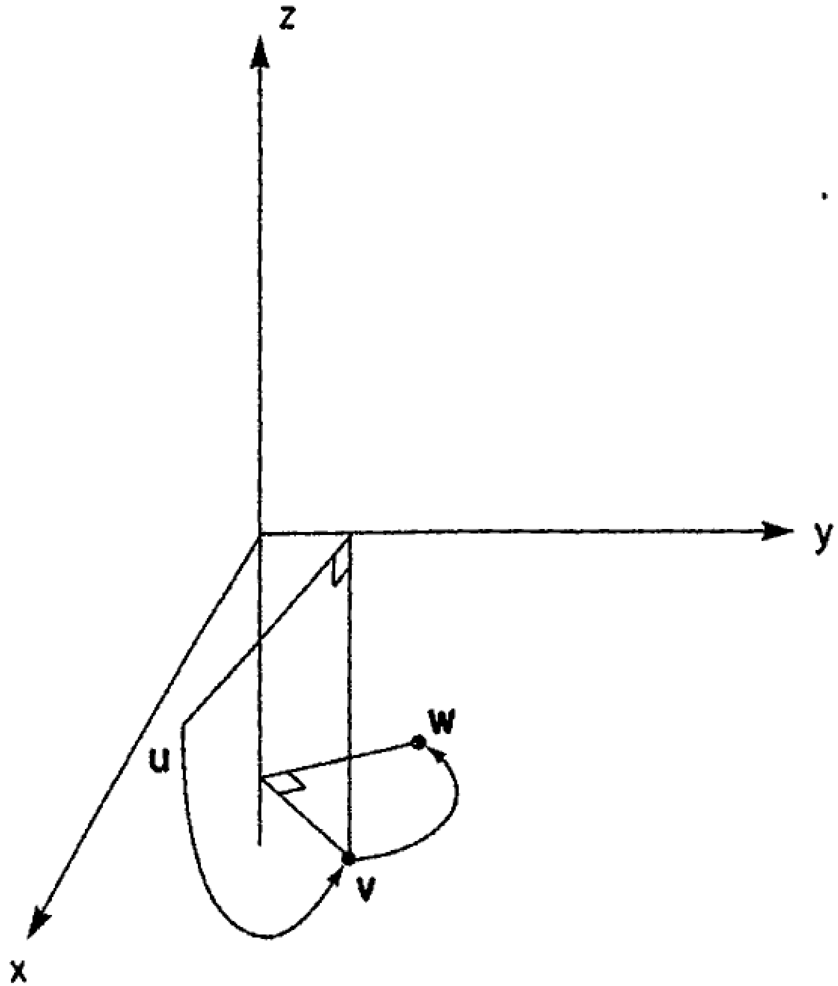


Figure 1.3. Rot(z, 90)Rot(y, 90)

Composition of rotation matrices for current Frames

- First rotate the given frame *A* according to R_B^A so as to align it with frame *B*
- Then rotate the current frame, now aligned with frame *B*, according to R_C^B so as to align it with frame *C*

Current and fixed Frames

- Current:

$$p^A = R_B^A p^B = R_B^A R_C^B p^C = R_B^A R_C^B R_D^C p^D$$

- Fixed: $p^A = R_B^A p^B$

$$\Rightarrow R_A^B p^A = p^B = R_C^B p^C$$

$$\Rightarrow R_B^C R_A^B p^A = p^C = R_D^C p^D$$

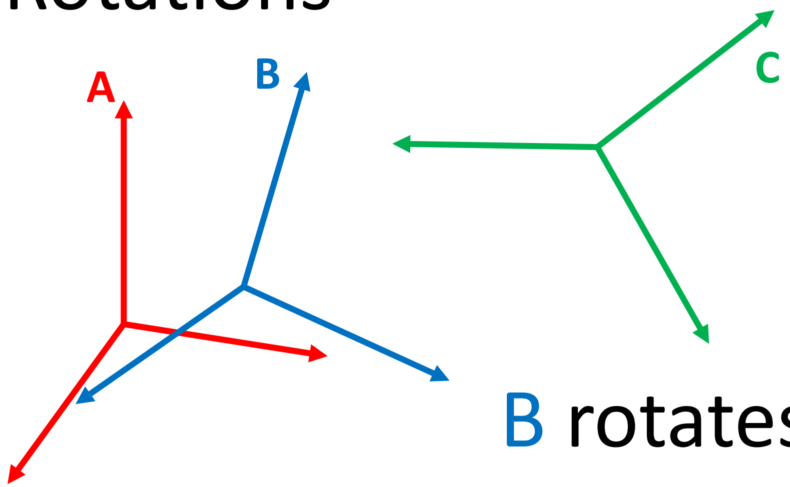
$$\Rightarrow R_C^D R_B^C R_A^B p^A = p^D$$

Composite Rotation Matrix

- A sequence of finite rotations
  - matrix multiplications do not commute!
  - rules:
    - if rotating coordinate O−uvw is rotating about a principal axis of a ***fixed*** O−xyz frame, then ***pre−multiply*** the previous (resultant) rotation matrix with an appropriate basic rotation matrix
    - if rotating coordinate O−uvw is rotating about **its own** principal axes, then ***post−multiply*** the previous (resultant) rotation matrix with an appropriate basic rotation matrix

So far:

Rotations



B rotates wrt A and C rotates wrt B but the rotation is described via A

Multiply on the left

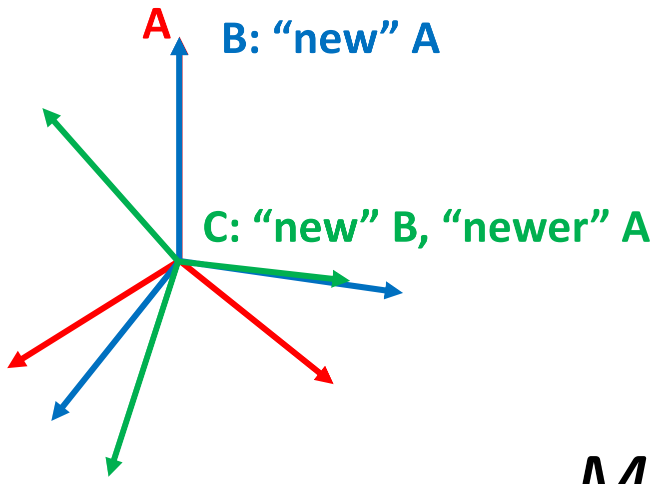
A FIXED:

$$p^{A} \rightarrow R_{A}^{B} p^{A} \rightarrow R_{B}^{C} R_{A}^{B} p^{A}$$

$$p^{B} \qquad\qquad p^{C}$$

So far:

Rotations



Multiply on the right

CURRENT:

$$p^{A} \rightarrow R_{B}^{A} p^{B} \rightarrow R_{B}^{A} R_{C}^{B} p^{C}$$

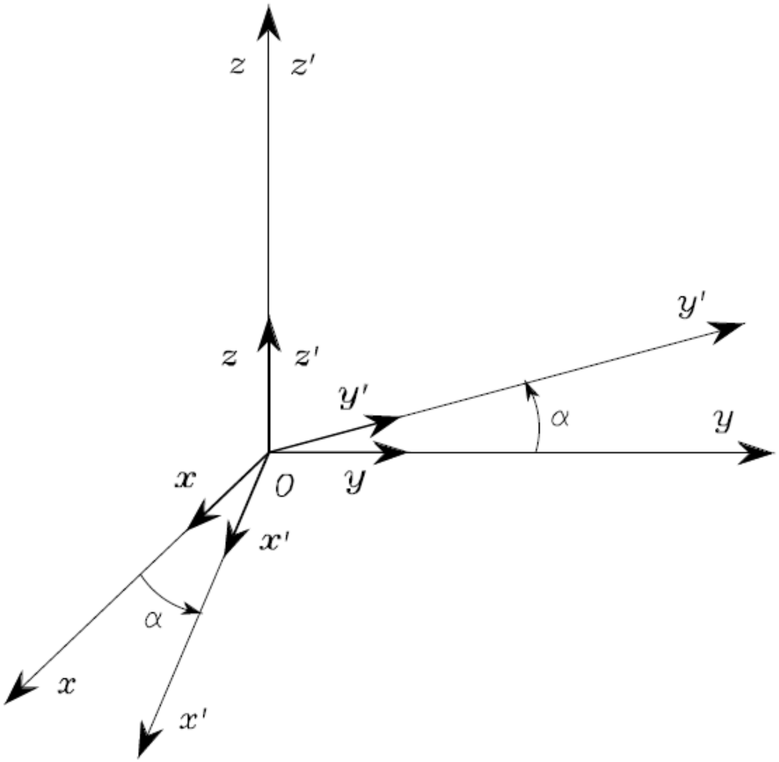
Practical Matters:

How to transform

Elementary Rotations

Frames that can be obtained via *elementary rotations* of the reference frame about one of the coordinate axes

Positive if they are made **counter−clockwise** about the relative axis. **Example: *z***



New unit vectors:

$$x' = \begin{bmatrix} \cos\alpha \\ \sin\alpha \\ 0 \end{bmatrix} \qquad y' = \begin{bmatrix} -\sin\alpha \\ \cos\alpha \\ 0 \end{bmatrix} \qquad z' = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Rotation Matrix 3D

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

$$\boldsymbol{R}_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$\boldsymbol{R}_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

$$\boldsymbol{R}_k(-\vartheta) = \boldsymbol{R}_k^T(\vartheta) \qquad k = x, y, z$$

Example 1

Consider the vector ***p*** which is obtained by rotating a vector ***p'*** in the plane *xy* by an angle α about axis *z* of the reference frame

Coordinates of the vector ***p'*:** (p'_x, p'_y, p'_z)

The vector ***p*** has components

$$p_x = p'_x \cos\alpha - p'_y \sin\alpha$$

$$p_y = p'_x \sin\alpha + p'_y \cos\alpha$$

$$p_z = p'_z.$$

$$\boldsymbol{p} = \boldsymbol{R}_z(\alpha)\boldsymbol{p}'$$

Example 2

- A point $a_{uvw} = (4,3,2)$ is attached to a rotating frame and this frame rotates 60 degree about the Oz axis **of the reference frame**. Find the coordinates of the point relative to the reference frame after the rotation.

$$a_{xyz} = Rot(z,60)a_{uvw}$$

$$= \begin{bmatrix} 0.5 & -0.866 & 0 \\ 0.866 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} -0.598 \\ 4.964 \\ 2 \end{bmatrix}$$

Example 3

- A point $a_{xyz} = (4,3,2)$ is the coordinate w.r.t. the reference coordinate system, find the corresponding point a_{uvw} **w.r.t. the rotated** Ouvw coordinate system if it has been rotated 60 degree about Oz axis.

$$a_{uvw} = Rot(z,60)^T a_{xyz}$$

$$= \begin{bmatrix} 0.5 & 0.866 & 0 \\ -0.866 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 4.598 \\ -1.964 \\ 2 \end{bmatrix}$$

Example 4

- Find the rotation matrix for the following operations:

Rotation φ about O_y axis

Rotation θ about O_w axis

Rotation α about O_u axis

Answer...

$$R = Rot(y,\varphi)Rot(w,\theta)Rot(u,\alpha)$$

$$R = \begin{bmatrix} C\varphi & 0 & S\varphi \\ 0 & 1 & 0 \\ -S\varphi & 0 & C\varphi \end{bmatrix} \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha & C\alpha \end{bmatrix}$$

$$= \begin{bmatrix} C\varphi C\theta & S\varphi S\alpha - C\varphi S\theta C\alpha & C\varphi S\theta S\alpha + S\varphi C\alpha \\ S\theta & C\theta C\alpha & -C\theta S\alpha \\ -S\varphi C\theta & S\varphi S\theta C\alpha + C\varphi S\alpha & C\varphi C\alpha - S\varphi S\theta S\alpha \end{bmatrix}$$

Pre−multiply if rotating about the O_{xyz} (reference) axes

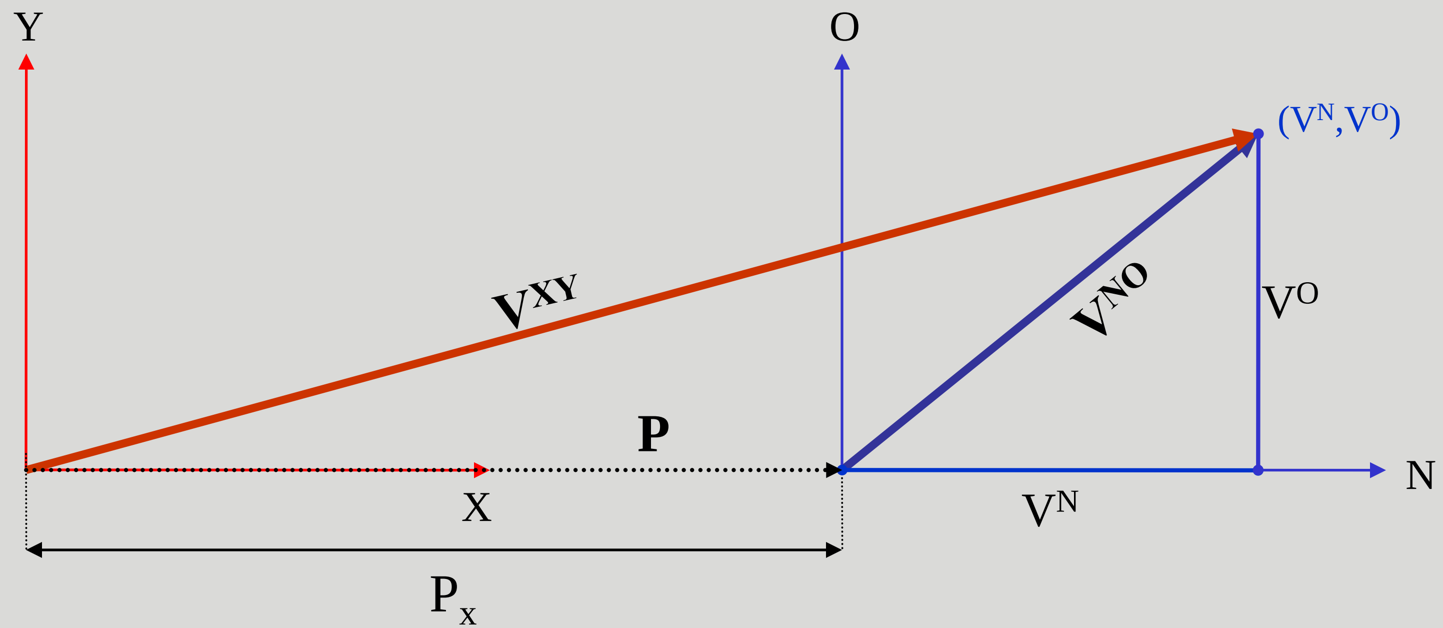
Post−multiply if rotating about the O_{uvw} (current) axes

Trigonometric shorthand

Symbol	Meaning
$C\phi$	$\cos\phi$
$S\phi$	$\sin\phi$
$V\phi$	$1 - \cos\phi$
C_k	$\cos\theta_k$
S_k	$\sin\theta_k$
C_{kj}	$\cos(\theta_k + \theta_j)$
S_{kj}	$\sin(\theta_k + \theta_j)$
C_{k-j}	$\cos(\theta_k - \theta_j)$
S_{k-j}	$\sin(\theta_k - \theta_j)$

Moving Between Coordinate Frames

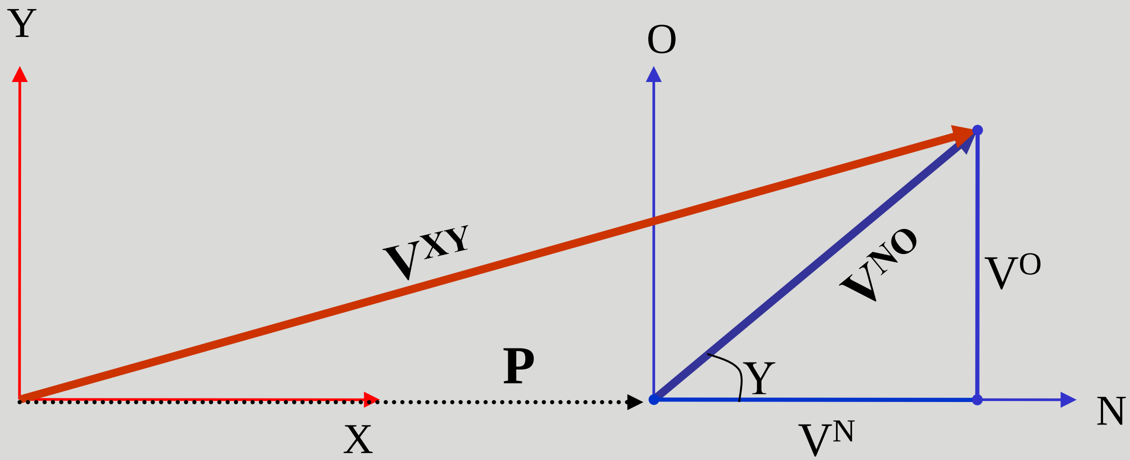
Translation Along the X−Axis



P_x = distance between the XY and NO coordinate planes

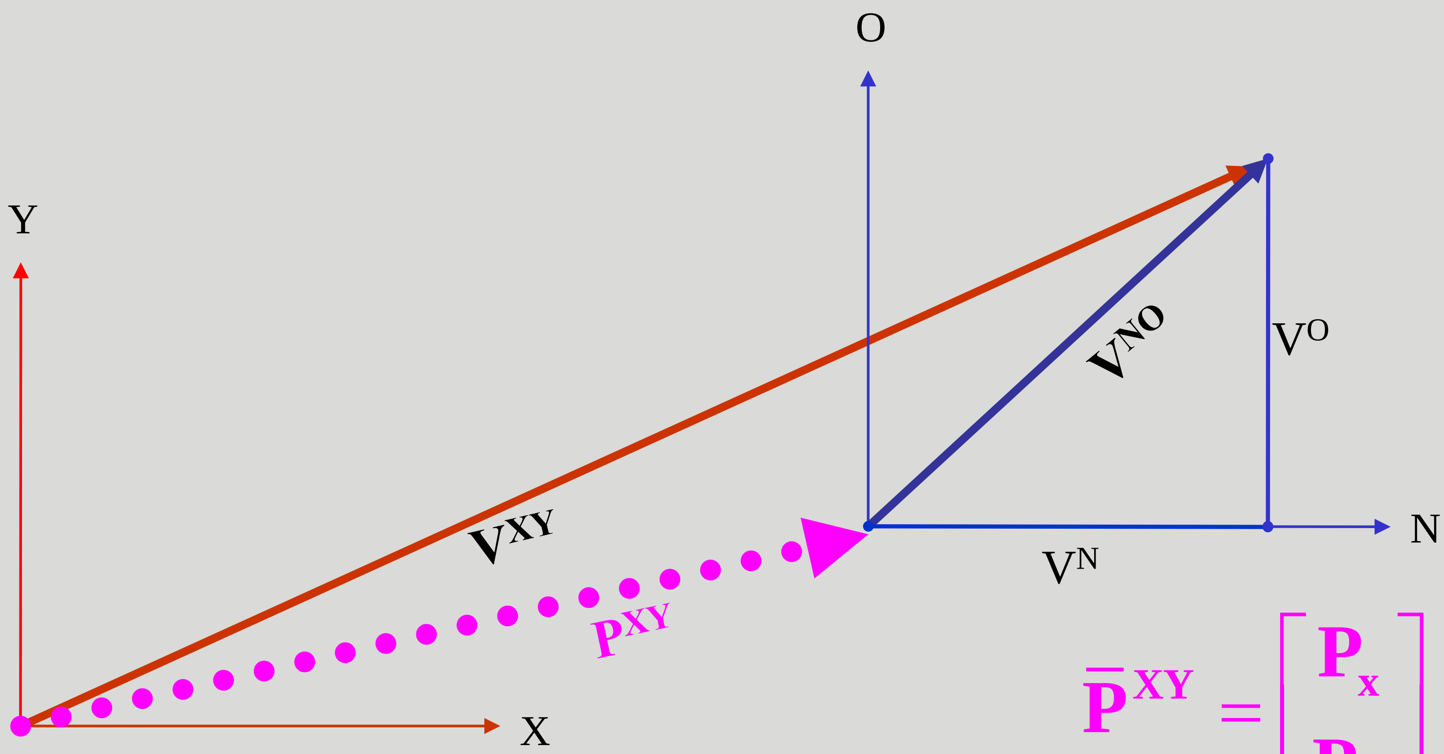
Notation: $\overline{\mathbf{V}}^{XY} = \begin{bmatrix} \mathbf{V}^X \\ \mathbf{V}^Y \end{bmatrix}$ $\quad \overline{\mathbf{V}}^{NO} = \begin{bmatrix} \mathbf{V}^N \\ \mathbf{V}^O \end{bmatrix}$ $\quad \overline{\mathbf{P}} = \begin{bmatrix} \mathbf{P_x} \\ \mathbf{0} \end{bmatrix}$

Writing $\overline{V}^{XY}$ in terms of $\overline{V}^{NO}$



$$\overline{V}^{XY} = \begin{bmatrix} P_X + V^N \\ V^O \end{bmatrix} = \overline{P} + \overline{V}^{NO}$$

Translation along the X–Axis and Y–Axis



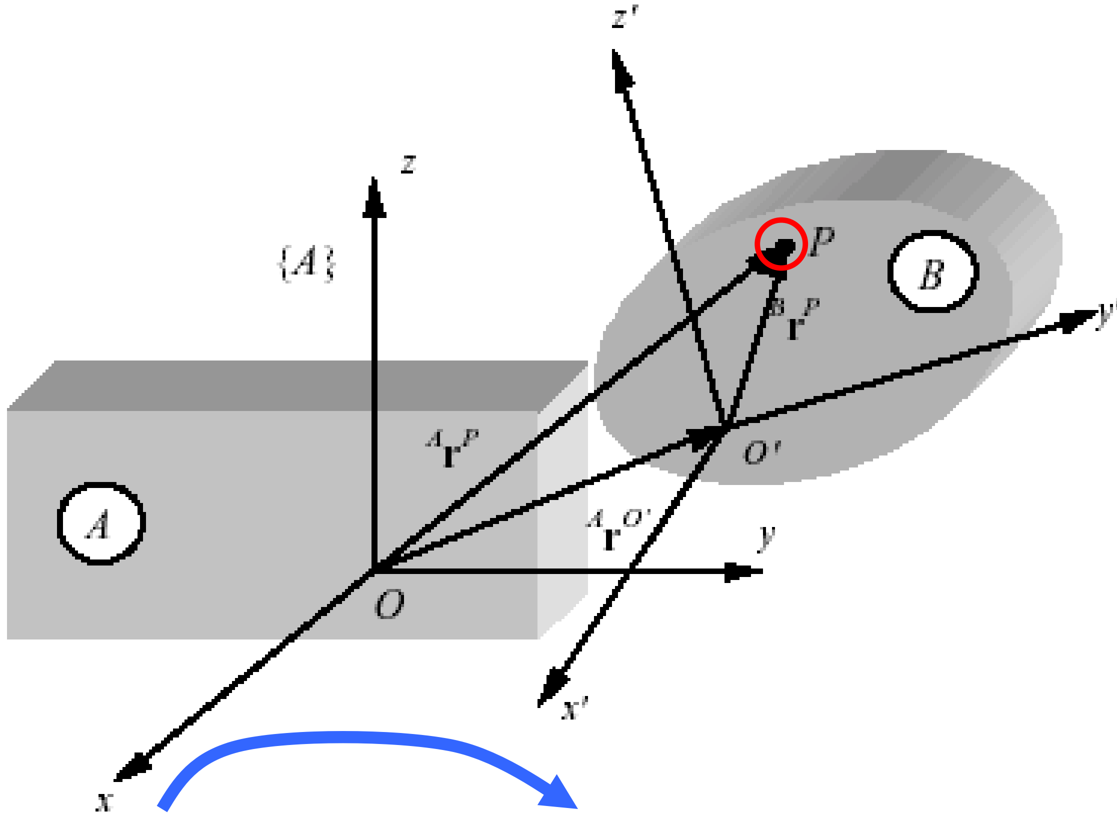
$$\overline{P}^{XY} = \begin{bmatrix} P_x \\ P_Y \end{bmatrix}$$

$$\overline{V}^{XY} = \overline{P} + \overline{V}^{NO} = \begin{bmatrix} P_X + V^N \\ P_Y + V^O \end{bmatrix}$$

Coordinate Transformations

- position vector of P in $\{B\}$ is transformed to position vector of P in $\{A\}$

- description of $\{B\}$ as seen from an observer in $\{A\}$



Rotation of $\{B\}$ with respect to $\{A\}$

$$ {}^{A}\mathbf{r}^{P} = {}^{A}\mathbf{R}_{B}\,{}^{B}\mathbf{r}^{P} + {}^{A}\mathbf{r}^{O'} $$

Translation of the origin of $\{B\}$ with respect to origin of $\{A\}$

Coordinate Transformations

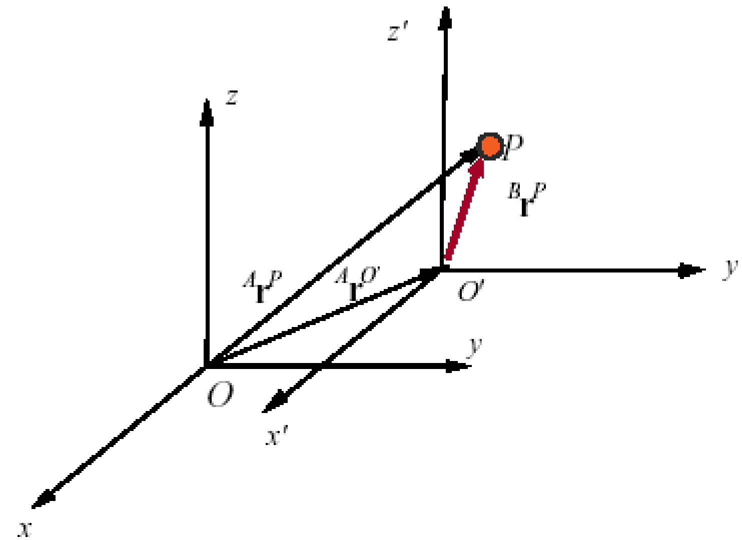
- Two Cases

$$ {}^{A}r^{P} = {}^{A}R_{B}\,{}^{B}r^{P} + {}^{A}r^{o'} $$

1. **Translation only**

– Axes of {B} and {A} are parallel

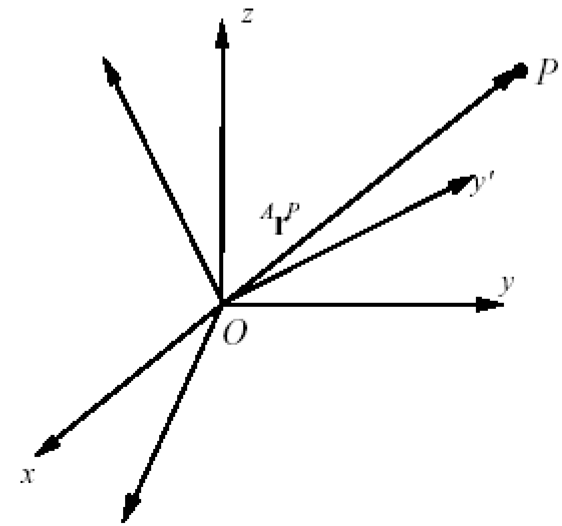
$$ {}^{A}R_{B} = 1 $$



2. **Rotation only**

– Origins of {B} and {A} are coincident

$$ {}^{A}r^{o'} = 0 $$



Homogeneous Representation

- Coordinate transformation from $\{B\}$ to $\{A\}$

$$^{A}r^{P} = {}^{A}R_{B}\,{}^{B}r^{P} + {}^{A}r^{o'}$$

$$\begin{bmatrix} ^{A}r^{P} \\ 1 \end{bmatrix} = \begin{bmatrix} ^{A}R_{B} & ^{A}r^{o'} \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} ^{B}r^{P} \\ 1 \end{bmatrix}$$

- Homogeneous transformation matrix

$$^{A}T_{B} = \begin{bmatrix} ^{A}R_{B} & ^{A}r^{o'} \\ 0_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} R_{3\times 3} & P_{3\times 1} \\ 0 & 1 \end{bmatrix}$$

Rotation matrix

Position vector

Scaling

Homogeneous Transformation

- Special cases

1. Translation

$$^{A}T_{B} = \begin{bmatrix} I_{3\times 3} & ^{A}r^{o'} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

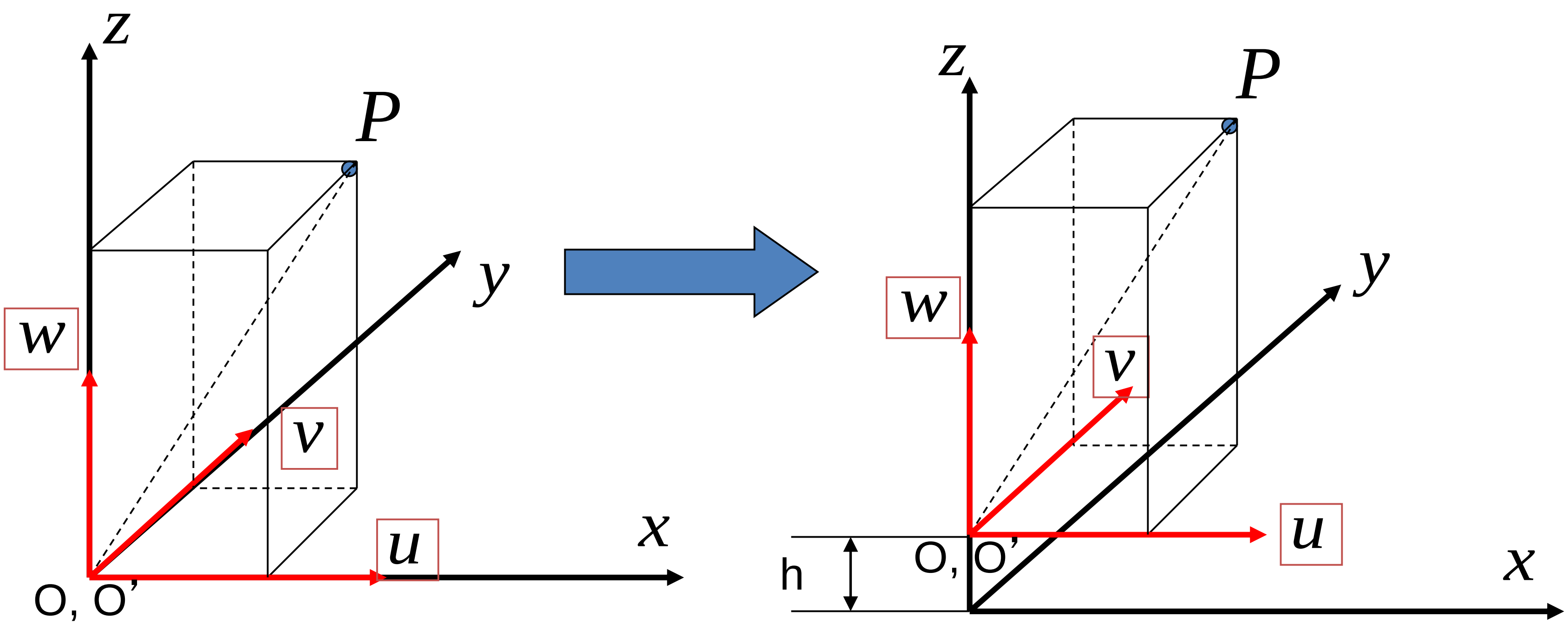
2. Rotation

$$^{A}T_{B} = \begin{bmatrix} ^{A}R_{B} & 0_{3\times 1} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

Example 5

- Translation along z–axis with *h*:

$$Trans(z,h) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_u \\ p_v \\ p_w \\ 1 \end{bmatrix} = \begin{bmatrix} p_u \\ p_v \\ p_w + h \\ 1 \end{bmatrix}$$

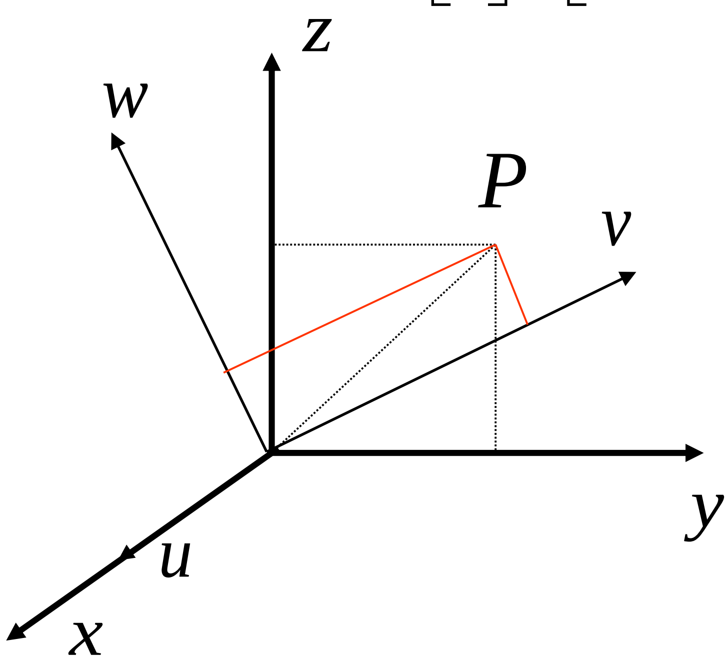


Example 6

- Rotation about the X−axis by θ

$$Rot(x,\theta)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & C\theta & -S\theta & 0\\ 0 & S\theta & C\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

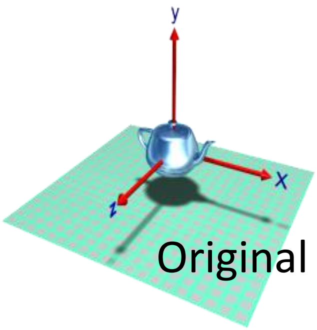
$$\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & C\theta & -S\theta & 0\\ 0 & S\theta & C\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}p_u\\ p_v\\ p_w\\ 1\end{bmatrix}$$

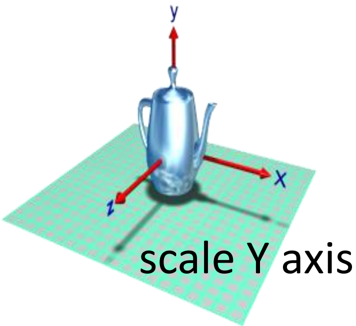


BONUS: Scaling & Stretching

- Scaling

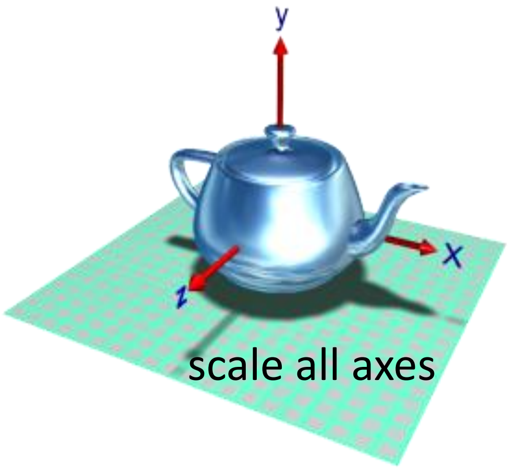
$$S = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & 3s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





- Stretching

$$T = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Recap: Homogeneous Transformation

- Composite Homogeneous Transformation Matrix
- Rules:
  - Transformation (rotation/translation) w.r.t (X,Y,Z) (OLD FRAME), using pre−multiplication
  - Transformation (rotation/translation) w.r.t (U,V,W) (NEW FRAME), using post−multiplication

Recap: Homogeneous Transformation

rotation translation

$$T \triangleq \left[\begin{array}{c|c} R & p \\ \hline \eta^T & \sigma \end{array}\right]$$

perspective scale

- Composite Homogeneous Transformation Matrix
- **Perspective**: to be used when a camera gets involved; now:[0,0,0]
- Homogeneous coordinates for $q \in \mathbb{R}^3$ wrt *F*, coordinate frame in $\mathbb{R}^3$:

$$[q]^F = [\sigma q_1, \sigma q_1, \sigma q_1, \sigma]^T$$

- Then, $q = H_\sigma [q]^F$ where $H_\sigma = \frac{1}{\sigma}[\mathbf{I}_3 \vdots \mathbf{0}_3]^F$ (We take $\sigma = 1$)

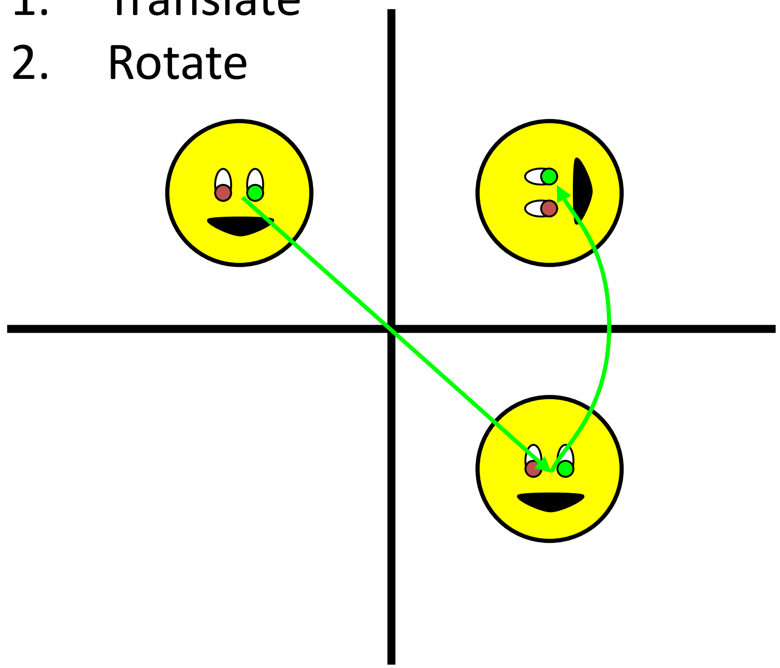
Recap: Homogeneous Coordinates

- Composite Homogeneous Transformation Matrix
- Rules:
  - Transformation (rotation/translation) w.r.t (X,Y,Z) (OLD FRAME), using pre–multiplication
  - Transformation (rotation/translation) w.r.t (U,V,W) (NEW FRAME), using post–multiplication

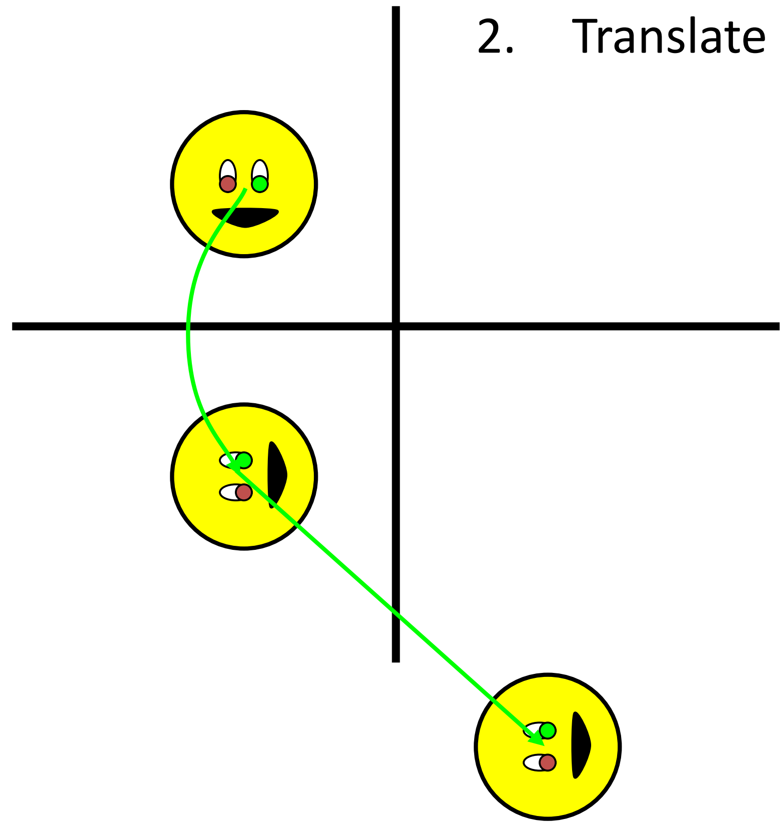
Order of operations…

…does matter. Let’s look at an example:

1. Translate
2. Rotate



1. Rotate
2. Translate



Example 7

- Find the homogeneous transformation matrix (T) for the following operations:

  Rotation α about Ox axis

  Translation of h along Ox axis

  Translation of d along Oz axis

  Rotation of θ about Oz axis

  Answer : $T = T_{z,\theta} T_{z,d} T_{x,h} T_{x,\alpha} I_{4\times 4}$

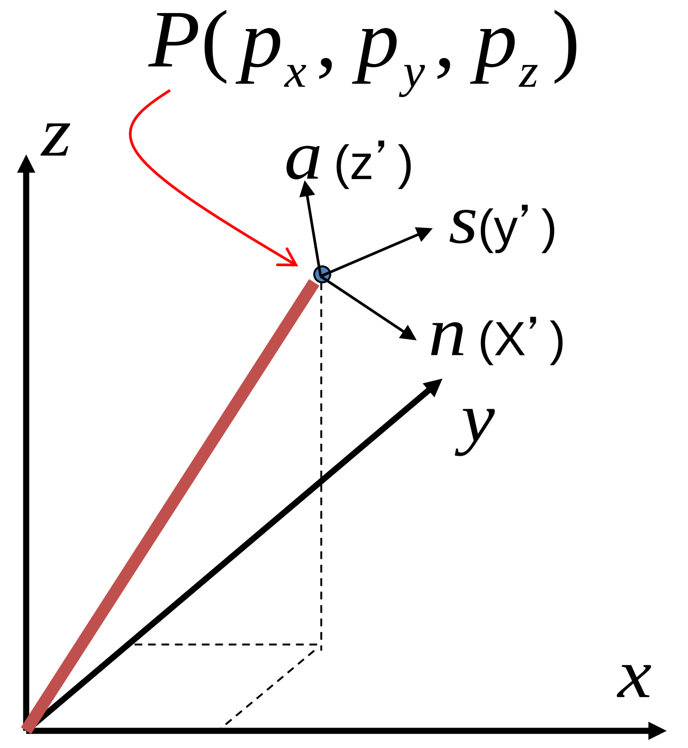
$$= \begin{bmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & h \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha & -S\alpha & 0 \\ 0 & S\alpha & C\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogeneous Representation

- A frame in space (Geometric Interpretation)

$$F = \begin{bmatrix} R_{3\times3} & P_{3\times1} \\ 0 & 1 \end{bmatrix}$$

$$F = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



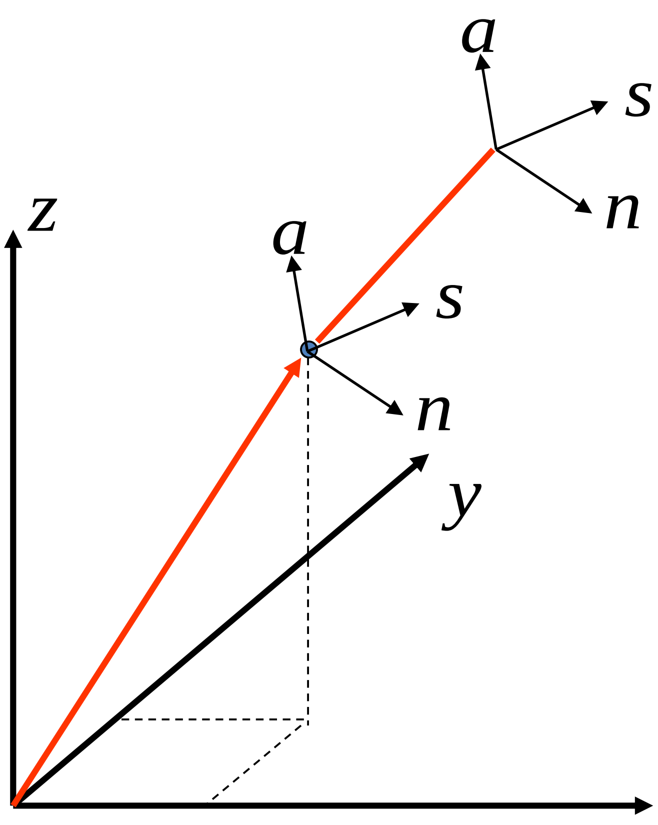
Principal axis *n* w.r.t. the reference coordinate system

Homogeneous Transformation

- Translation

$$F_{new} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

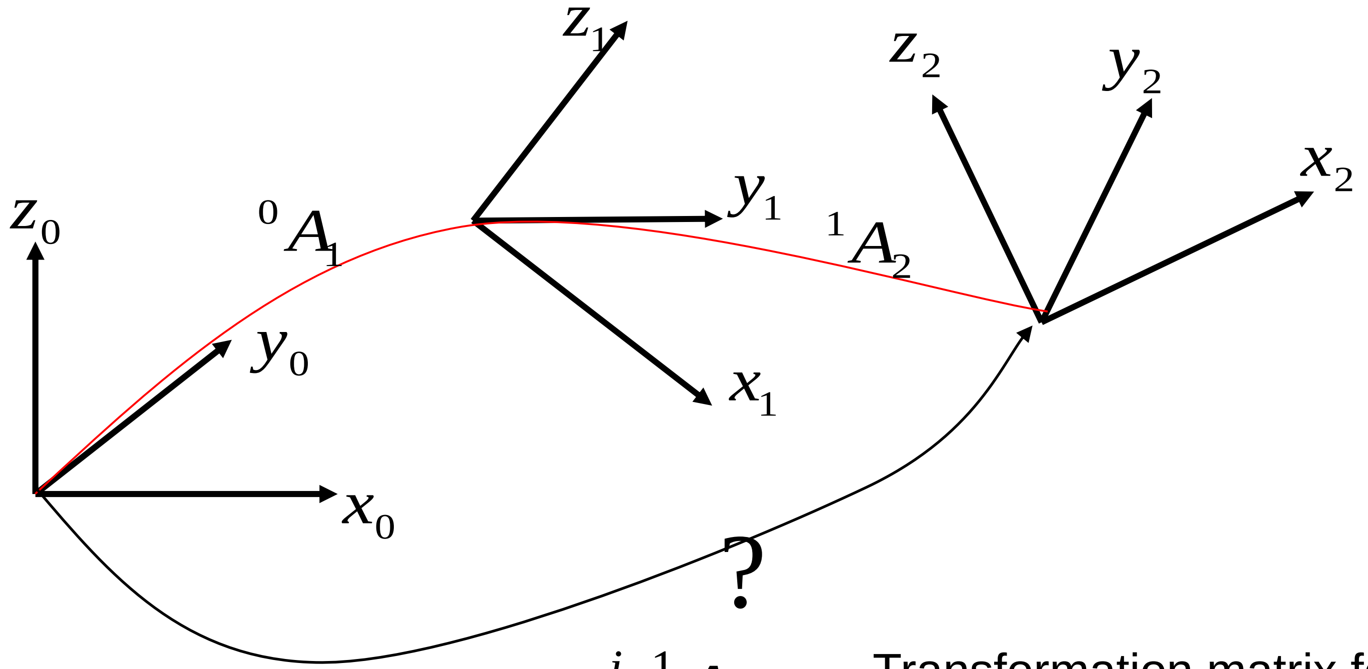
$$= \begin{bmatrix} n_x & s_x & a_x & p_x + d_x \\ n_y & s_y & a_y & p_y + d_y \\ n_z & s_z & a_z & p_z + d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$F_{new} = Trans(d_x, d_y, d_z) \times F_{old}$$

Homogeneous Transformation

Composite Homogeneous Transformation Matrix



$${}^{i-1}A_i$$

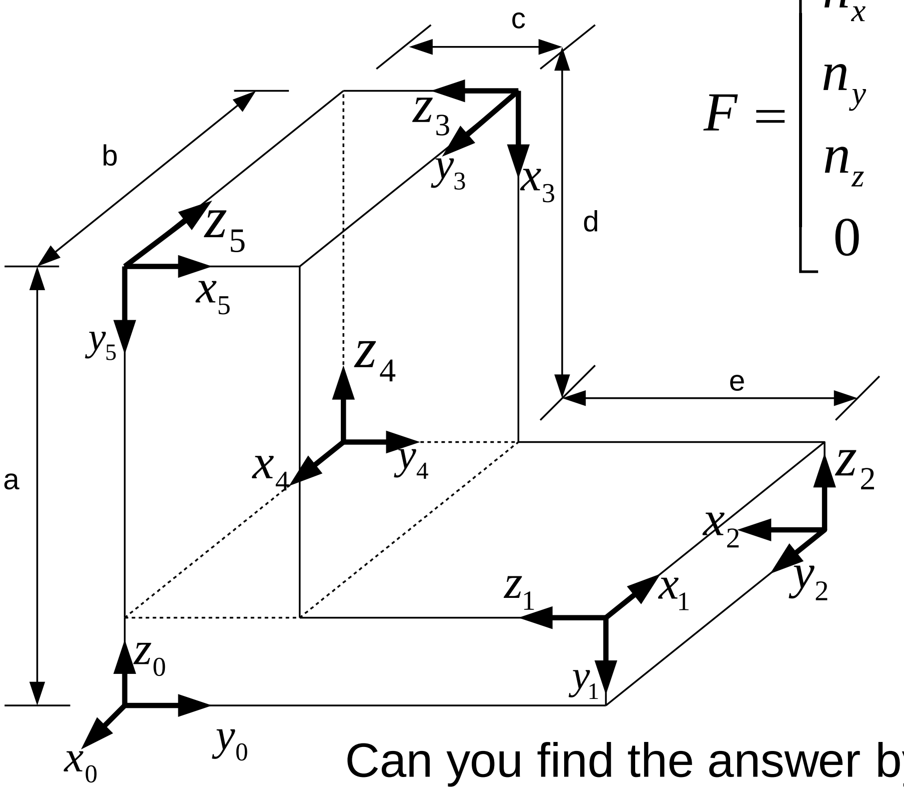
Transformation matrix for adjacent coordinate frames

$${}^0A_2 = {}^0A_1 \, {}^1A_2$$

Chain product of successive coordinate transformation matrices

Example 8

- For the figure shown below, find the 4x4 homogeneous transformation matrices ${}^{0}A_i$ and ${}^{i-1}A_i$ for i = 1, 2, 3, 4, 5



$$F = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{0}A_1 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & e+c \\ 0 & -1 & 0 & a-d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{1}A_2 = \begin{bmatrix} 0 & -1 & 0 & b \\ 0 & 0 & -1 & a-d \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{0}A_2 = \begin{bmatrix} 0 & 1 & 0 & -b \\ -1 & 0 & 0 & e+c \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

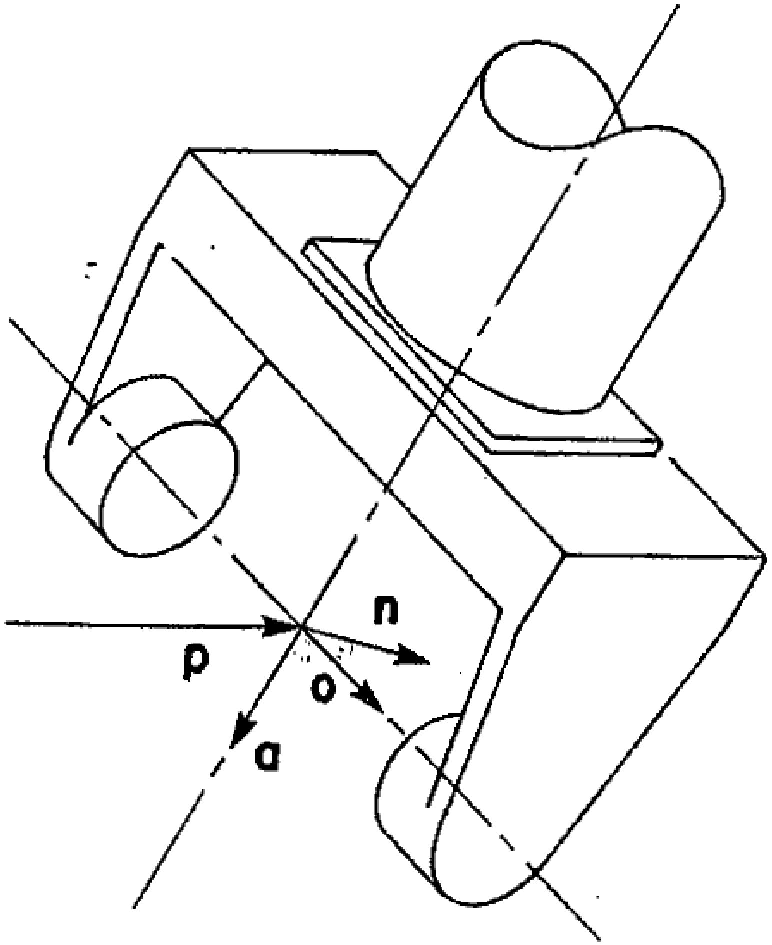
Can you find the answer by observation based on the geometric interpretation of homogeneous transformation matrix?

Positions, orientations and frames

- Remember the robot's end part:

Three unit vectors describing the hand orientation:

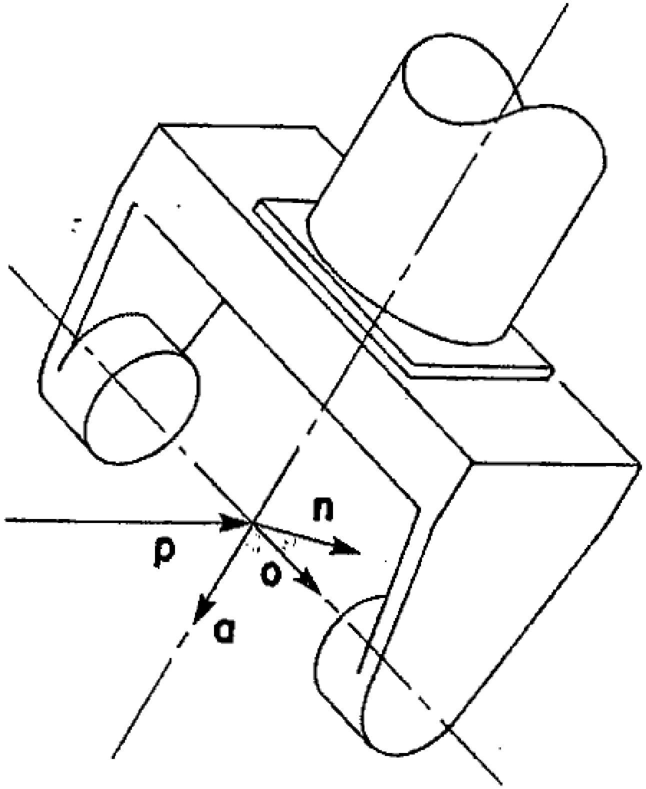
- The z vector lies in the direction from which the hand would approach an object and is known as the **approach vector, *a***.
- The y vector, known as the **orientation vector, o**, is in the direction specifying the orientation of the hand, from fingertip to fingertip.
- The final vector, known as the **normal vector, n**, forms a right−handed set of vectors and is thus specified by the vector cross−product



Positions, orientations and frames

$$n = o \times a$$

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

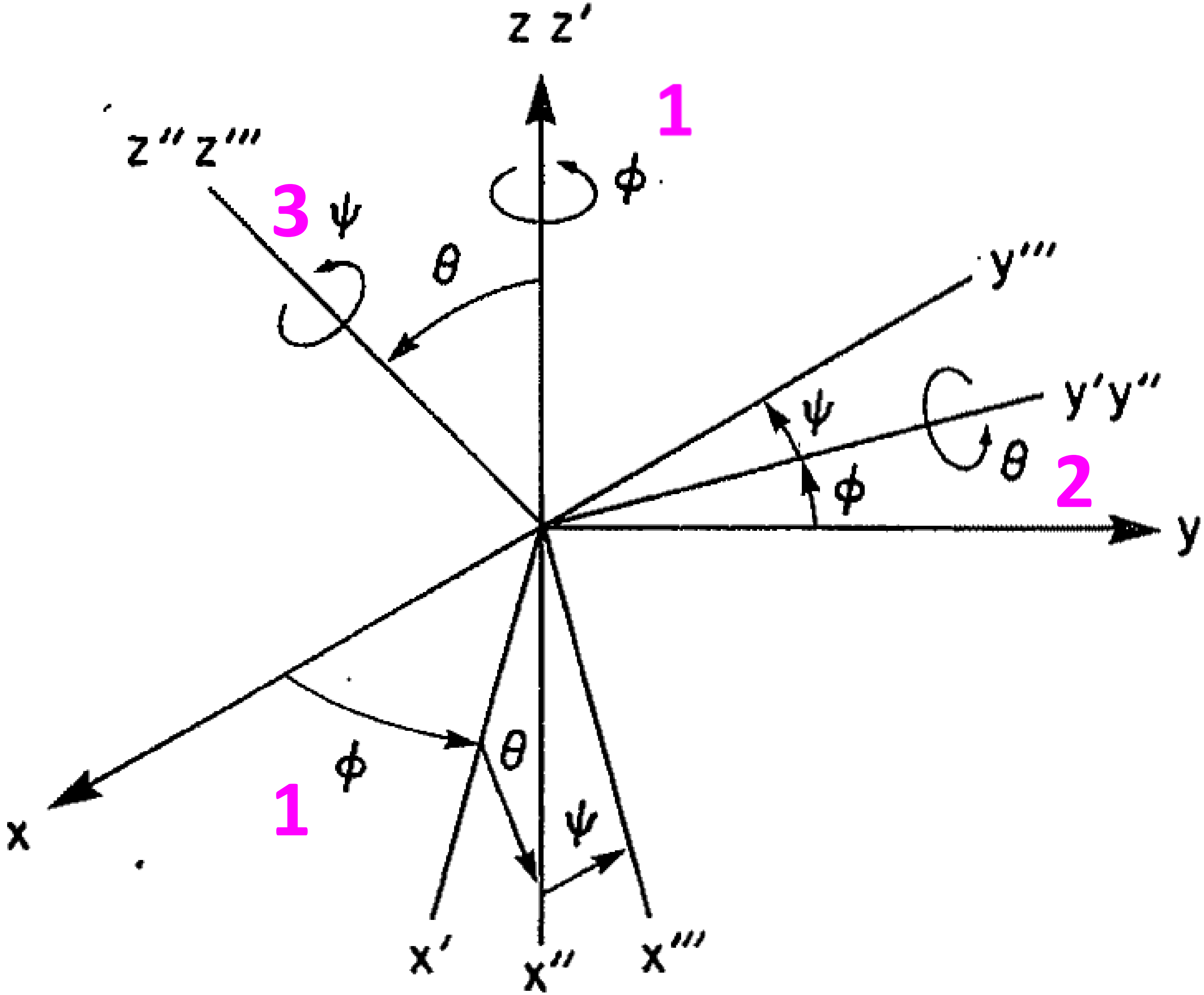


- *12 quantities*

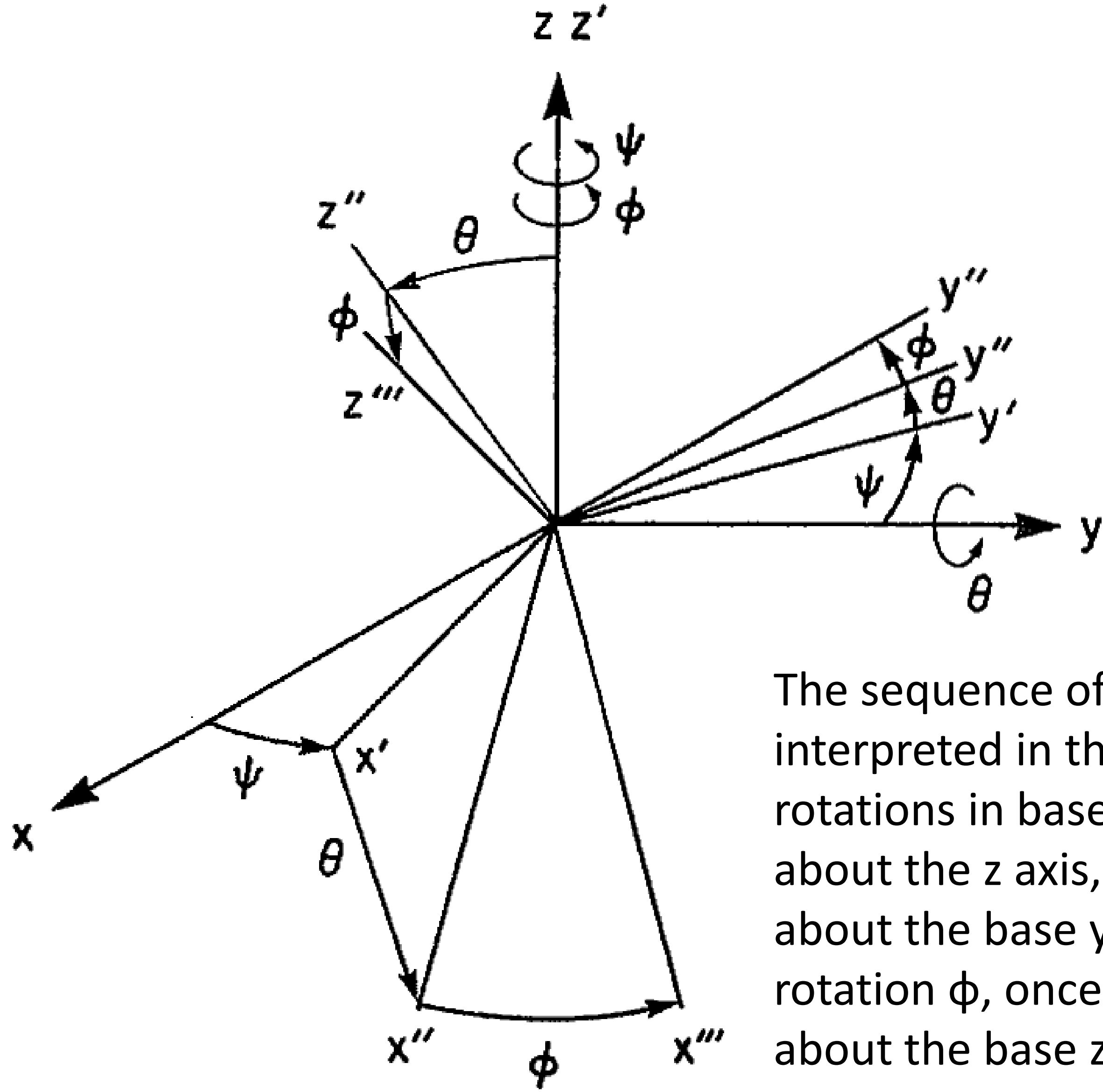
Euler Angles

- Orientation is more frequently specified by a sequence of rotations about the x, y, or z axes.
- **Euler angles** describe any possible orientation in terms of a rotation $\boldsymbol{\varphi}$ about the z axis, then a rotation $\boldsymbol{\theta}$ about the new y axis, y', and finally, a rotation of $\boldsymbol{\psi}$ about the new z axis, z''.

Euler Angles



Euler Angles Interpreted in Base Coordinates



The sequence of rotations can be interpreted in the reverse order as rotations in base coordinates: a rotation ψ about the z axis, followed by a rotation θ about the base y axis, and finally a rotation ϕ, once again about the base z axis

Euler Angles

$$Euler\left(\varphi,\theta,\psi\right)=Rot\left(z,\varphi\right)Rot\left(y,\theta\right)Rot\left(z,\psi\right)$$

$$R_{z\varphi}=\begin{pmatrix}\cos\varphi & -\sin\varphi & 0\\ \sin\varphi & \cos\varphi & 0\\ 0 & 0 & 1\end{pmatrix},R_{u'\theta}=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos\theta & -\sin\theta\\ 0 & \sin\theta & \cos\theta\end{pmatrix},$$

$$R_{w''\psi}=\begin{pmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{pmatrix}$$

Euler Angle

Resultant Eulerian rotation matrix:

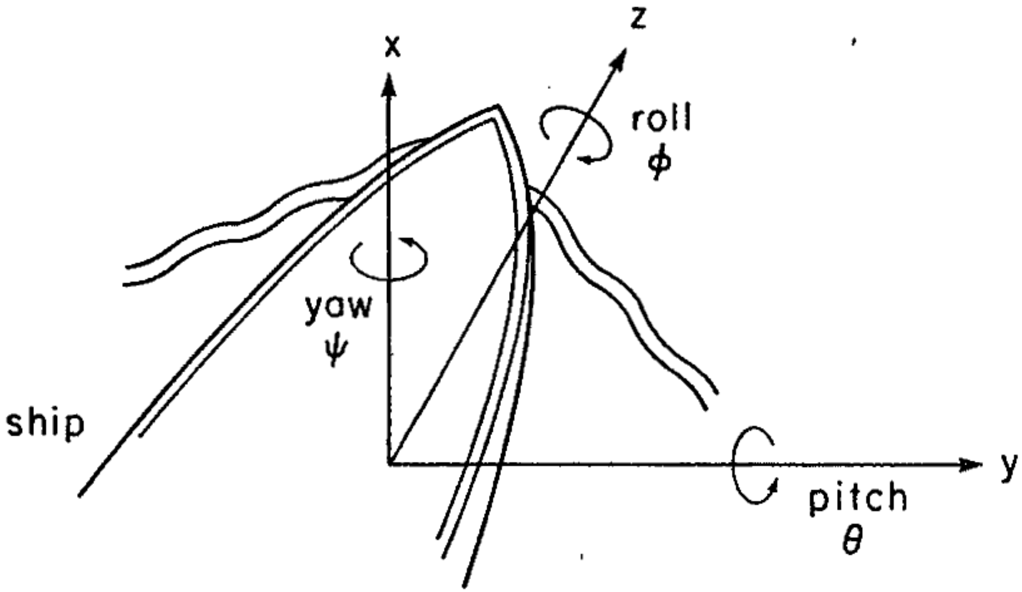
$$R_{\phi,\theta,\psi} = R_{z,\phi} R_{y,\theta} R_{z,\psi} = \begin{bmatrix} C\phi & -S\phi & 0 \\ S\phi & C\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C\phi C\theta C\psi - S\phi S\psi & -C\phi C\theta S\psi - S\phi C\psi & C\phi S\theta \\ S\phi C\theta C\psi + C\phi S\psi & -S\phi C\theta S\psi + C\phi C\psi & S\phi S\theta \\ -S\theta C\psi & S\theta S\psi & C\theta \end{bmatrix}$$

Roll, Pitch, Yaw

Περιστροφή, Πρόνευση, Εκτροπή

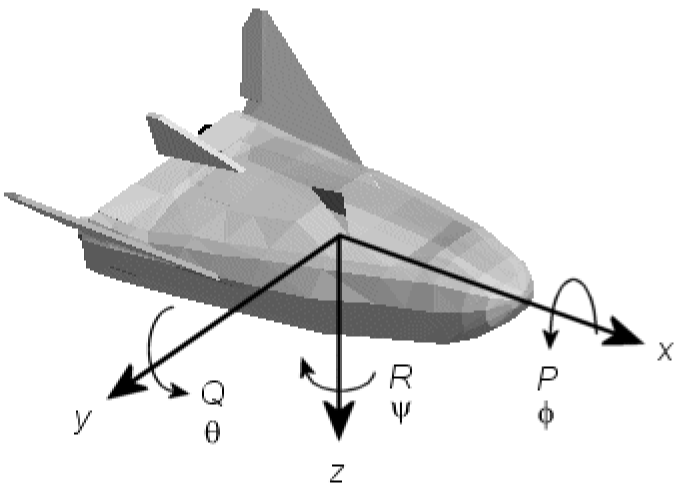
For a ship moving along the z axis, then roll corresponds to a rotation φ about the z axis, pitch corresponds to a rotation θ about the y axis, and yaw corresponds to a rotation ψ about the x axis



Roll, Pitch, Yaw

Περιστροφή, Πρόνευση, Εκτροπή

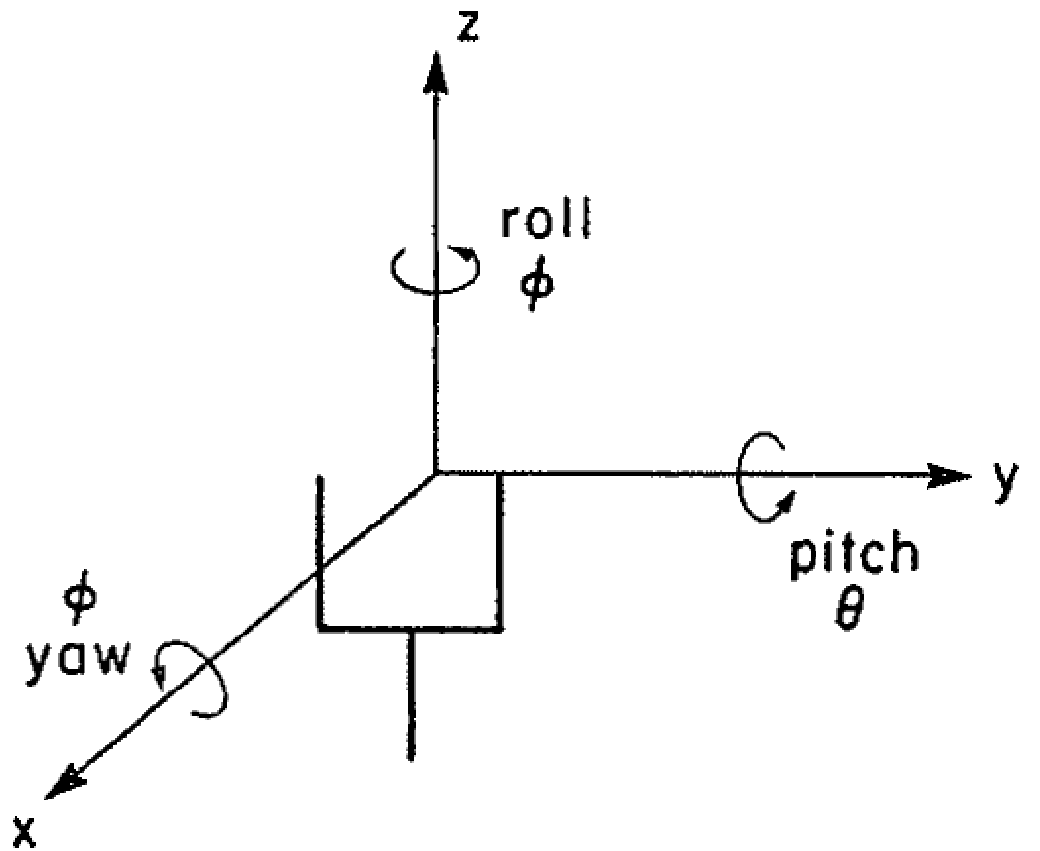
For an airplane moving along the x axis, then roll corresponds to a rotation φ about the x axis, pitch corresponds to a rotation θ about the y axis, and yaw corresponds to a rotation ψ about the z axis



Roll, Pitch, Yaw

Περιστροφή, Πρόνευση, Εκτροπή

- Robot manipulator, Robot hand



Roll, Pitch, Yaw

$$R_{\phi,\theta,\psi} = R_{z,\phi} R_{y,\theta} R_{x,\psi} = \begin{bmatrix} C\phi & -S\phi & 0 \\ S\phi & C\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\psi & -S\psi \\ 0 & S\psi & C\psi \end{bmatrix}$$

$$= \begin{bmatrix} C\phi C\theta & C\phi S\theta S\psi - S\phi C\psi & C\phi S\theta C\psi + S\phi S\psi \\ S\phi C\theta & S\phi S\theta S\psi + C\phi C\psi & S\phi S\theta C\psi - C\phi S\psi \\ -S\theta & C\theta S\psi & C\theta C\psi \end{bmatrix}$$

Orientation Representation

- Euler Angles Representation (ϕ, θ, ψ)
  - Many different types
  - Description of Euler angle representations

	Euler Angle I	Euler Angle II	Roll−Pitch−Yaw
Sequence	ϕ about OZ axis	ϕ about OZ axis	ψ about OX axis
of	θ about OU axis	θ about OV axis	θ about OY axis
Rotations	ψ about OW axis	ψ about OW axis	ϕ about OZ axis

Can be useful

- **Cylindrical** Coordinates :

$$Cyl(z,\alpha,r) \;=\; Trans(0,0,z)\; Rot(z,\alpha)\; Trans(r,0,0)$$

- **Spherical** Coordinates:

$$Sph(\alpha,\beta,r) \;=\; Rot(z,\alpha)\; Rot(y,\beta)\; Trans(0,0,r)$$

Homogeneous transformations for robot manipulators

- A serial link manipulator consists of a sequence of links connected together by actuated joints.
- For an *n* degree of freedom manipulator, there will be *n* links and *n* joints.
- The base of the manipulator is link 0 and is not considered one of the *n* links.
- Link 1 is connected to the base link by joint 1.
- There is no joint at the end of the final link.

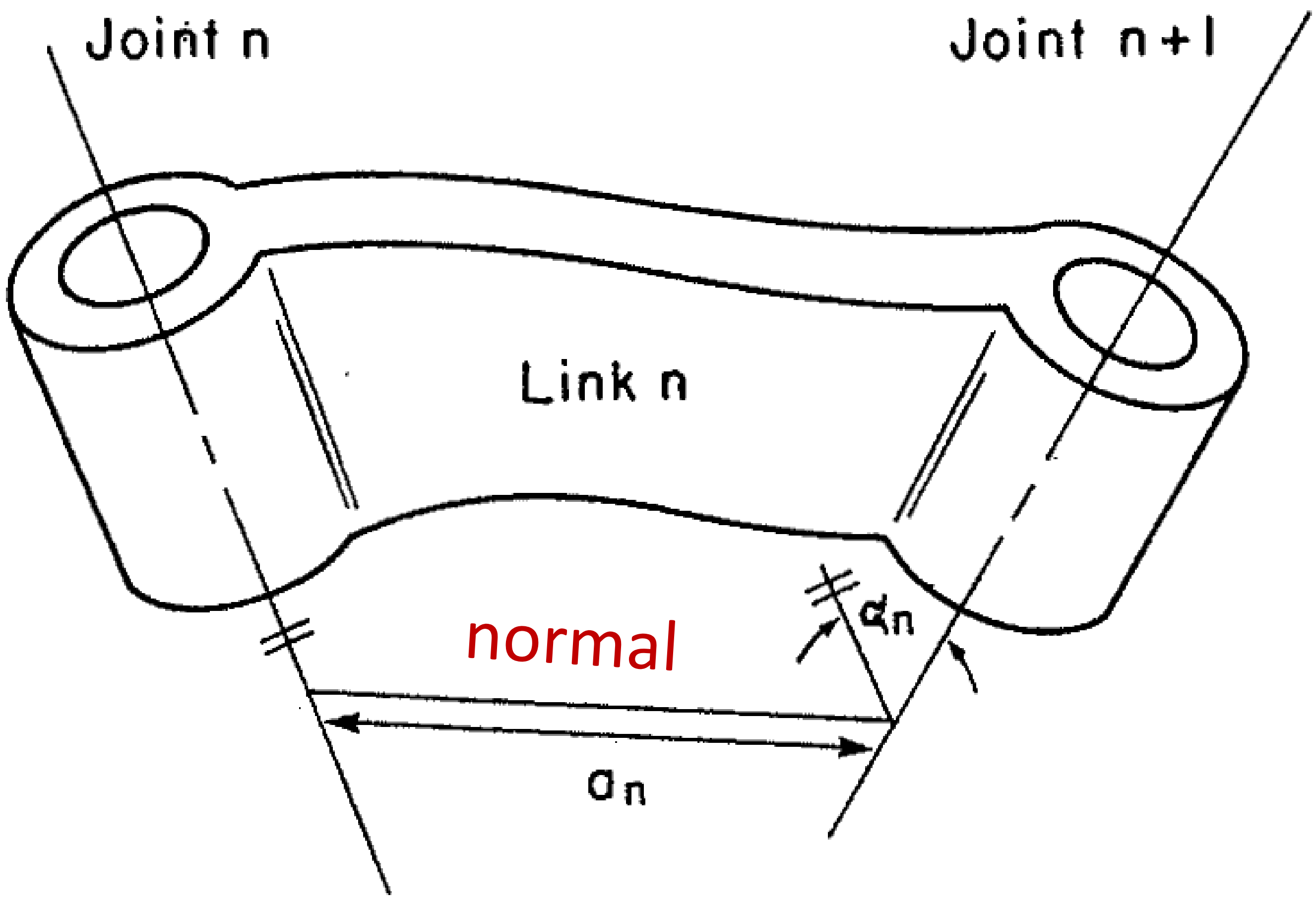
The Length a and Twist a of a Link

- Any link can be characterized by two dimensions:

1. the common normal distance a_n and
2. the angle α_n between the axes in a plane perpendicular to a_n.

- It is customary to call a_n the **length** and α_n the **twist** of the link

The Length a and Twist a of a Link



Important variables

- Angle θ between the normals
- Relative position d between links

Denavit–Hartenberg Parameters

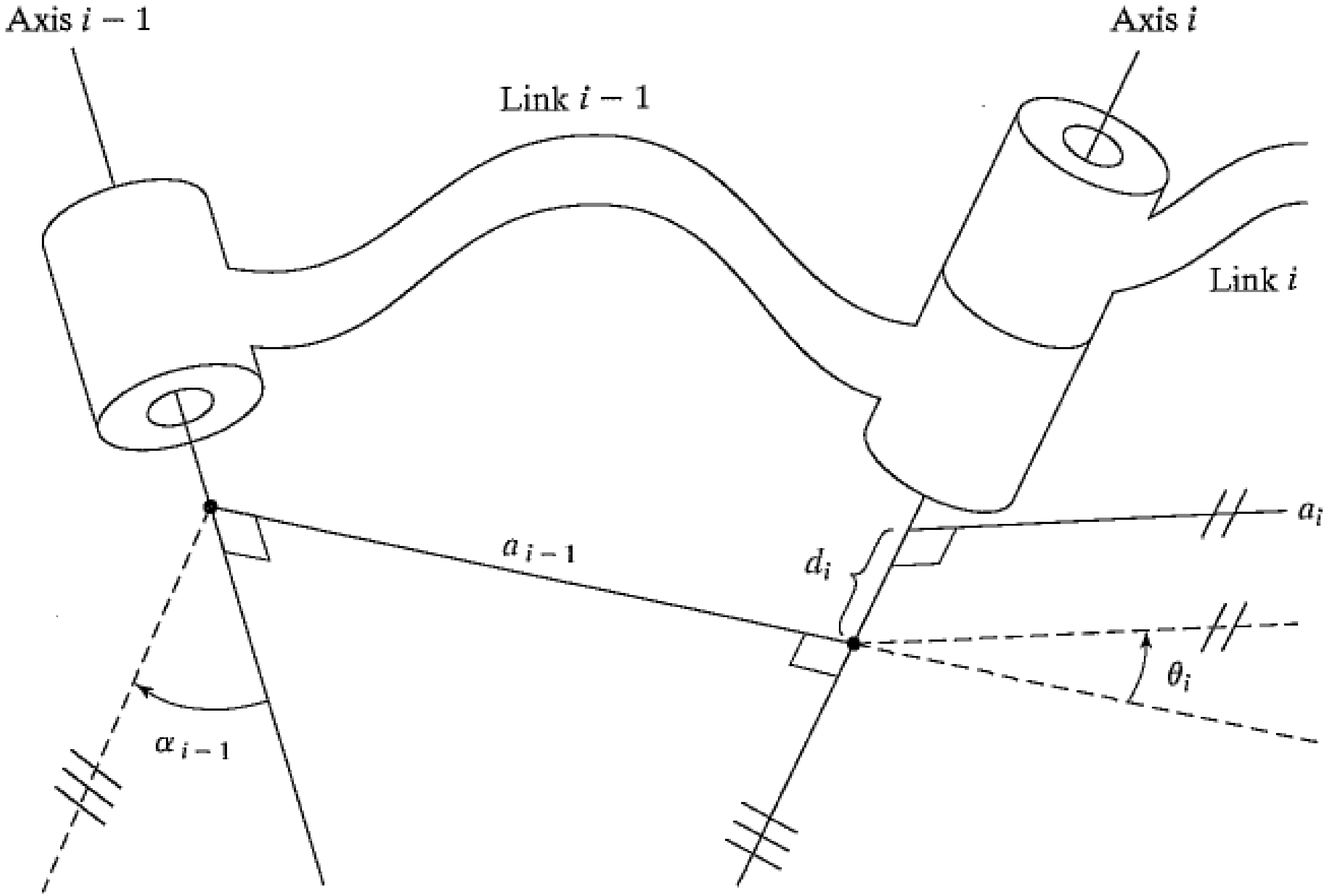
- 4 in total D–H parameters: $\alpha_i,\ a_i,\ d_i,\ \theta_i$

- 3 fixed link parameters

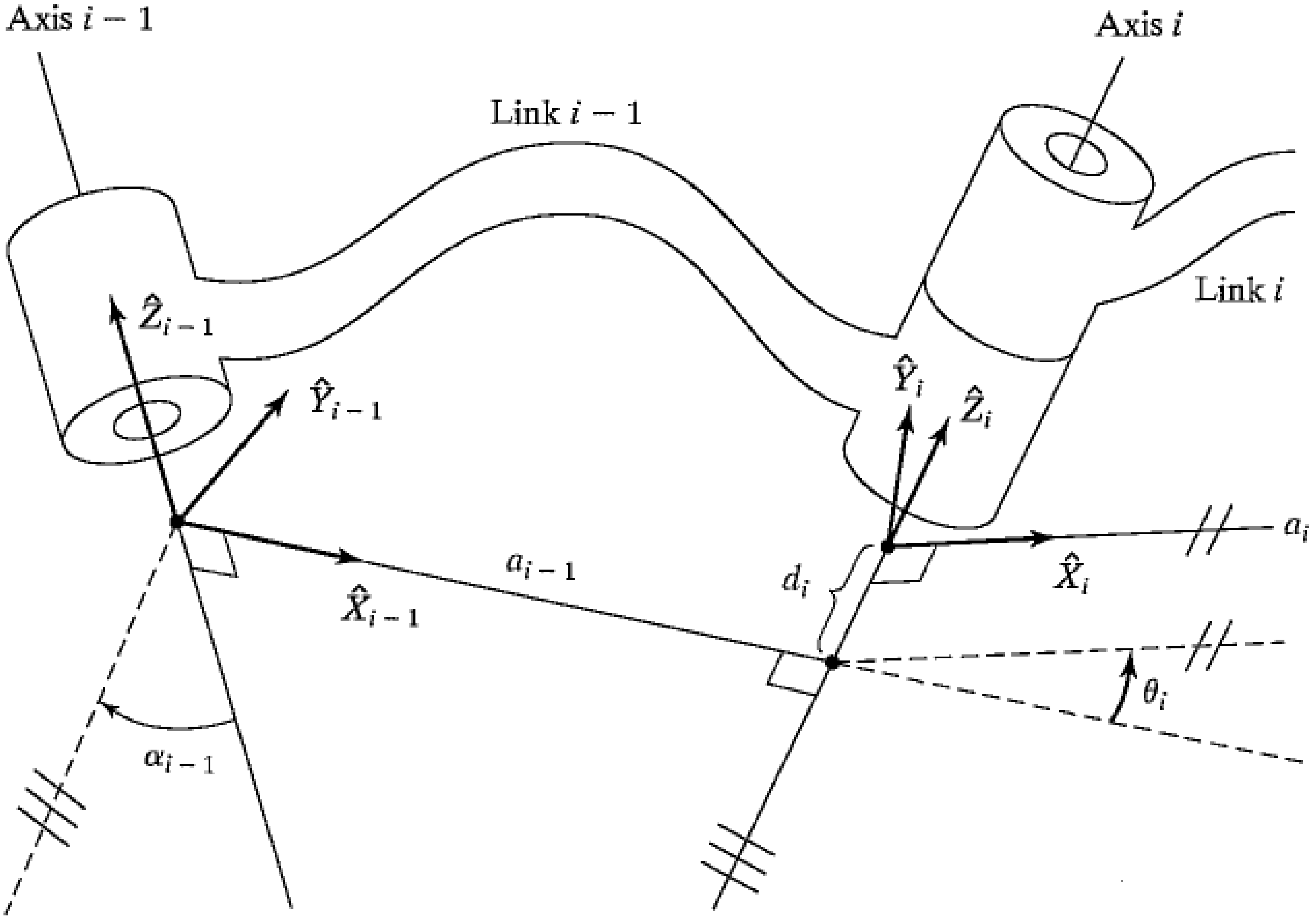
- 1 joint variable $\begin{cases} \theta_i \text{ variable if revolute joint} \\ d_i \text{ variable if prismatic joint} \end{cases}$

- α_i and a_i describe the Link i
- d_i and θ_i describe the Link's connection

Denavit–Hartenberg Parameters



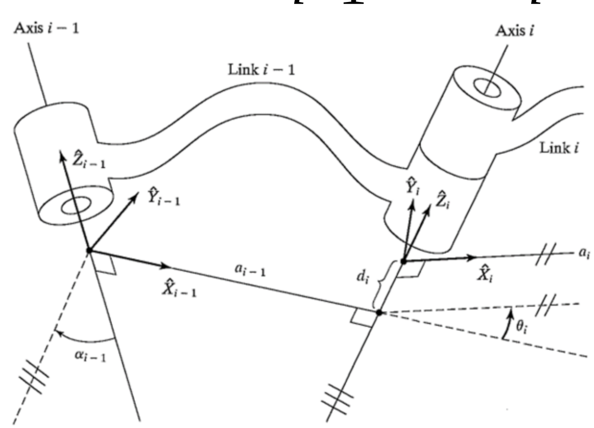
AFFIXING FRAMES TO LINKS



Summary for link parameters and link frames

If the attachment convention has been followed, then define:

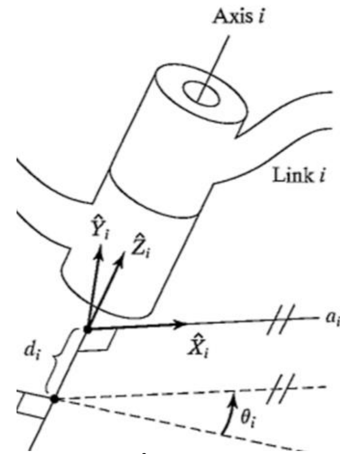
- a_i = the distance from Z_i to Z_{i+1} measured along X_i
- α_i = the angle from Z_i to Z_{i+1} measured about X_i
- d_i = the distance from X_{i-1} to X_i measured along Z_i
- θ_i = the angle from X_{i-1} to X_i measured about Z_i



Summary for link parameters and link frames

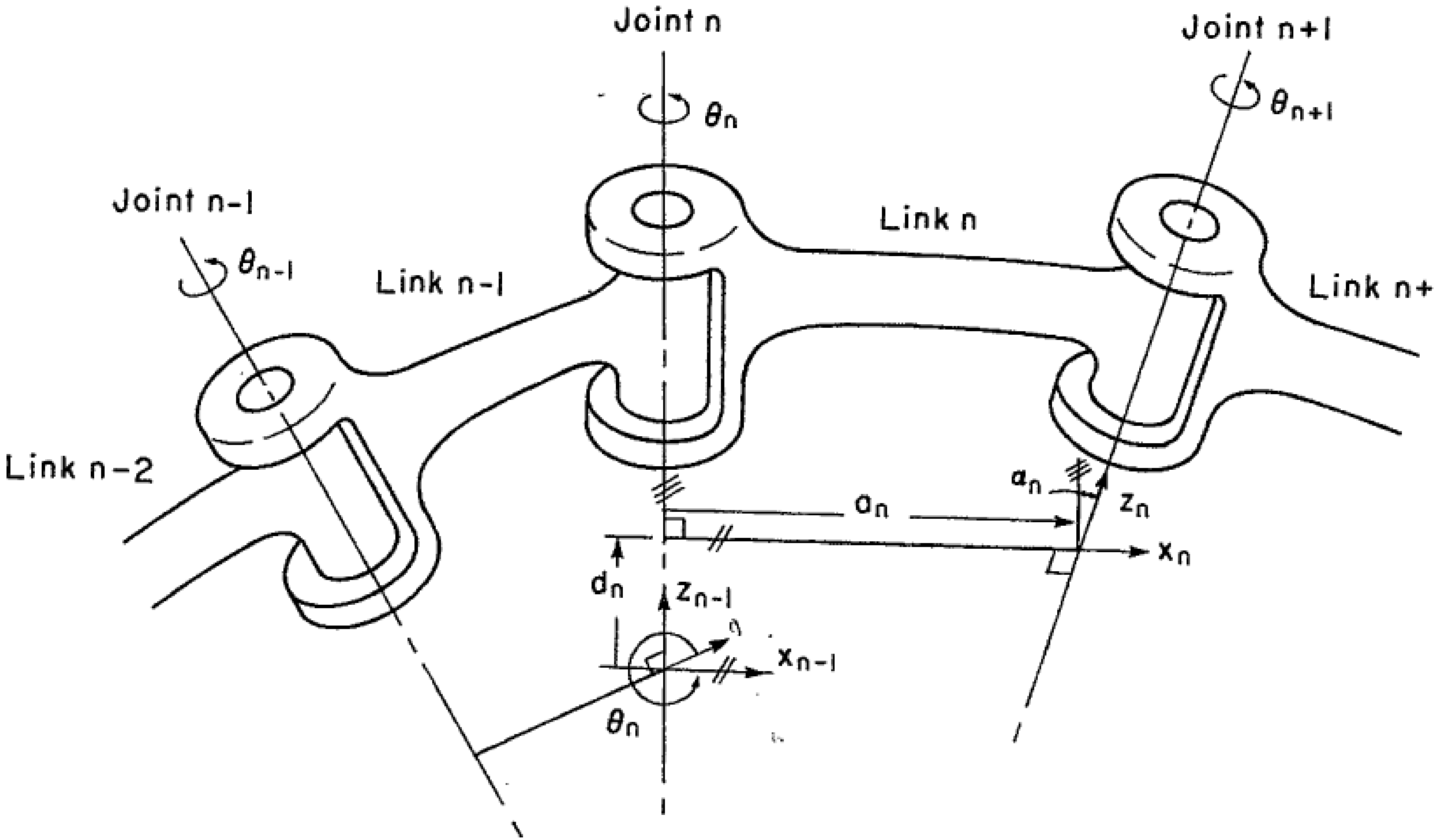
- We usually choose $a_i > 0$, because it corresponds to a distance; however, α_i, d_i, and θ_i are signed quantities
- Attachment of frames to links **NOT UNIQUE** !!

RULES: Revolute joints



1. Each link requires a coordinate frame assigned to it.
2. In revolute joints θ_n is the joint variable.
3. The origin of the coordinate frame of link *n* is set to be at the intersection of the common normal between the axes of joints *n* and *n* + 1 and the axis of joint *n*.
4. In the case of intersecting joint axes, the origin is at the point of intersection of the joint axes.
5. If the axes are parallel, the origin is chosen to make the joint distance zero for the next link whose coordinate origin is defined.
6. The *z* axis for link *n* will be aligned with the axis of joint *n* + 1.
7. The *x* axis will be aligned with any common normal which exists and is directed along the normal from joint *n* to joint *n* + 1.
8. In the case of intersecting joints, the direction of the *x* axis is parallel or antiparallel to the vector cross product $z_{n-1} \times z_n$.
9. Notice that this condition is also satisfied for the *x* axis directed along the normal between joints *n* and *n* + 1.
10. θ_n is zero for the *n*th revolute joint when x_{n-1} and x_n are parallel and have the same direction.

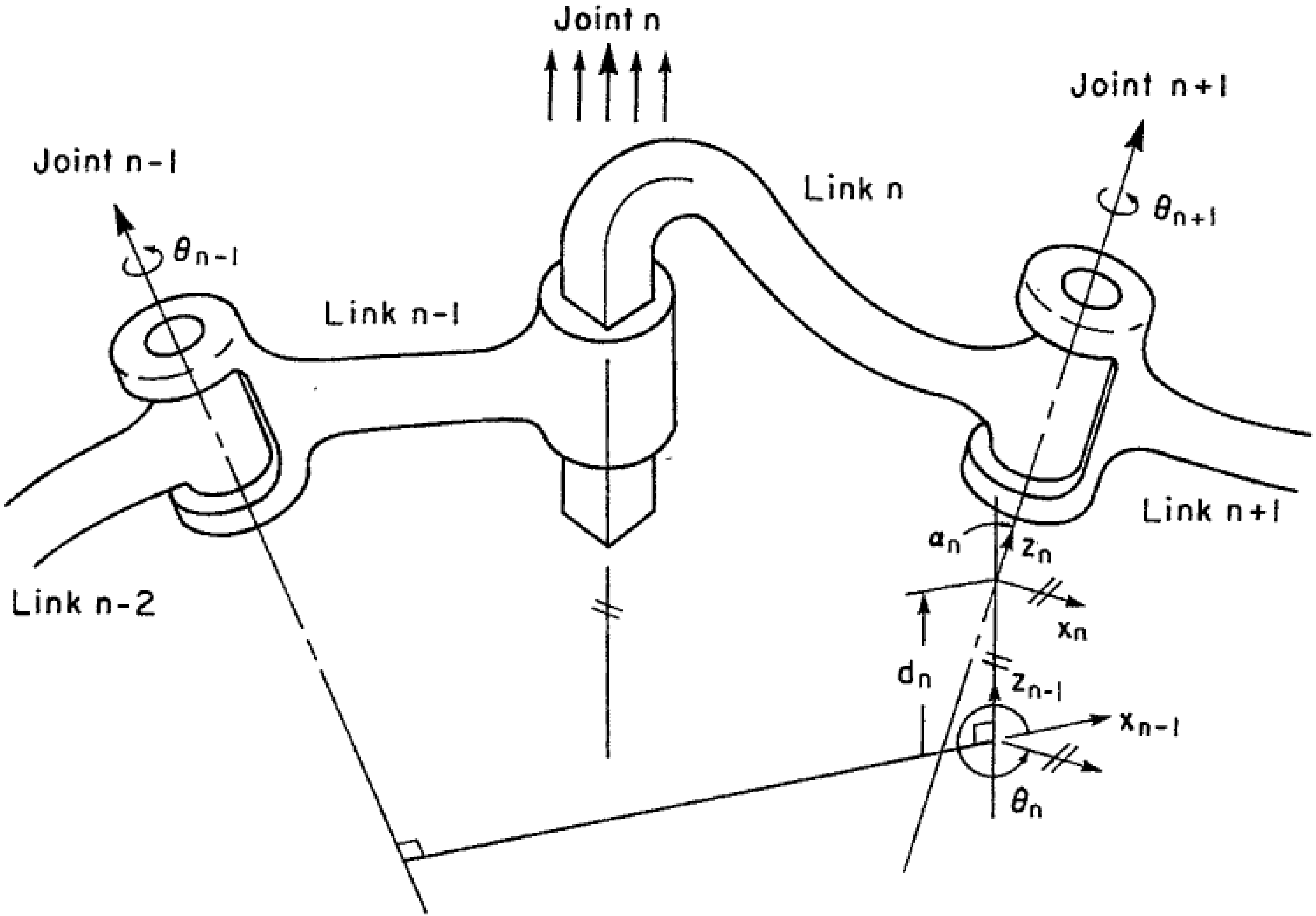
RULES: Revolute joints



RULES: Prismatic joints

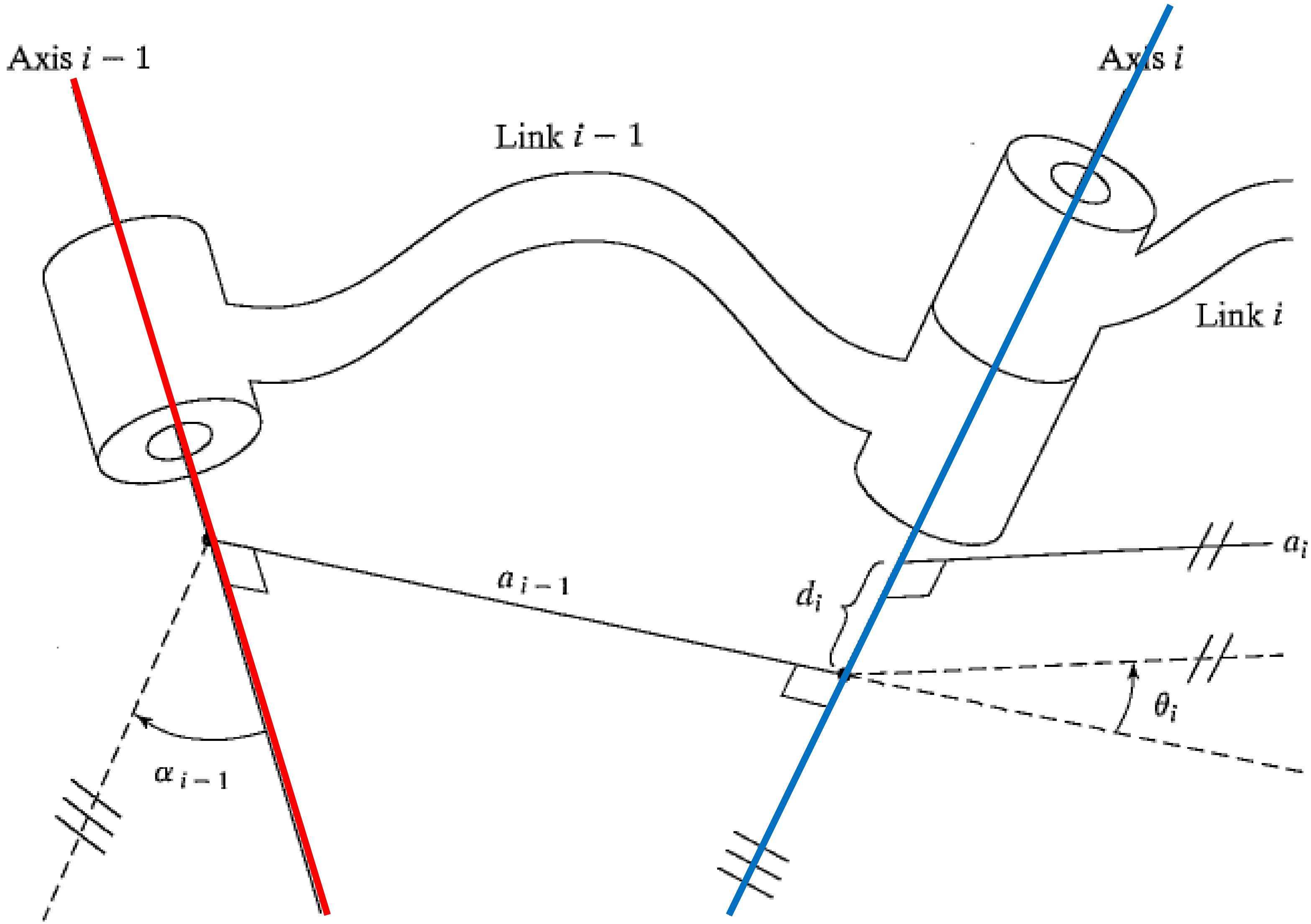
1. In the case of a prismatic joint, the distance d_n is the joint variable.
2. The direction of the joint axis is the direction in which the joint moves.
3. The direction of the axis is defined but, unlike a revolute joint, the position in space is not defined.
4. Length a_n has no meaning and is set to zero.
5. **The origin of the coordinate frame for a prismatic joint is coincident with the next defined link origin.**
6. The z axis of the prismatic link is aligned with the axis of joint $n+1$.
7. The x_n axis is parallel or antiparallel to the vector cross product of the direction of the prismatic joint and z_n.
8. For a prismatic joint, we will define the zero position when $d_n = 0$.

RULES: Prismatic joints

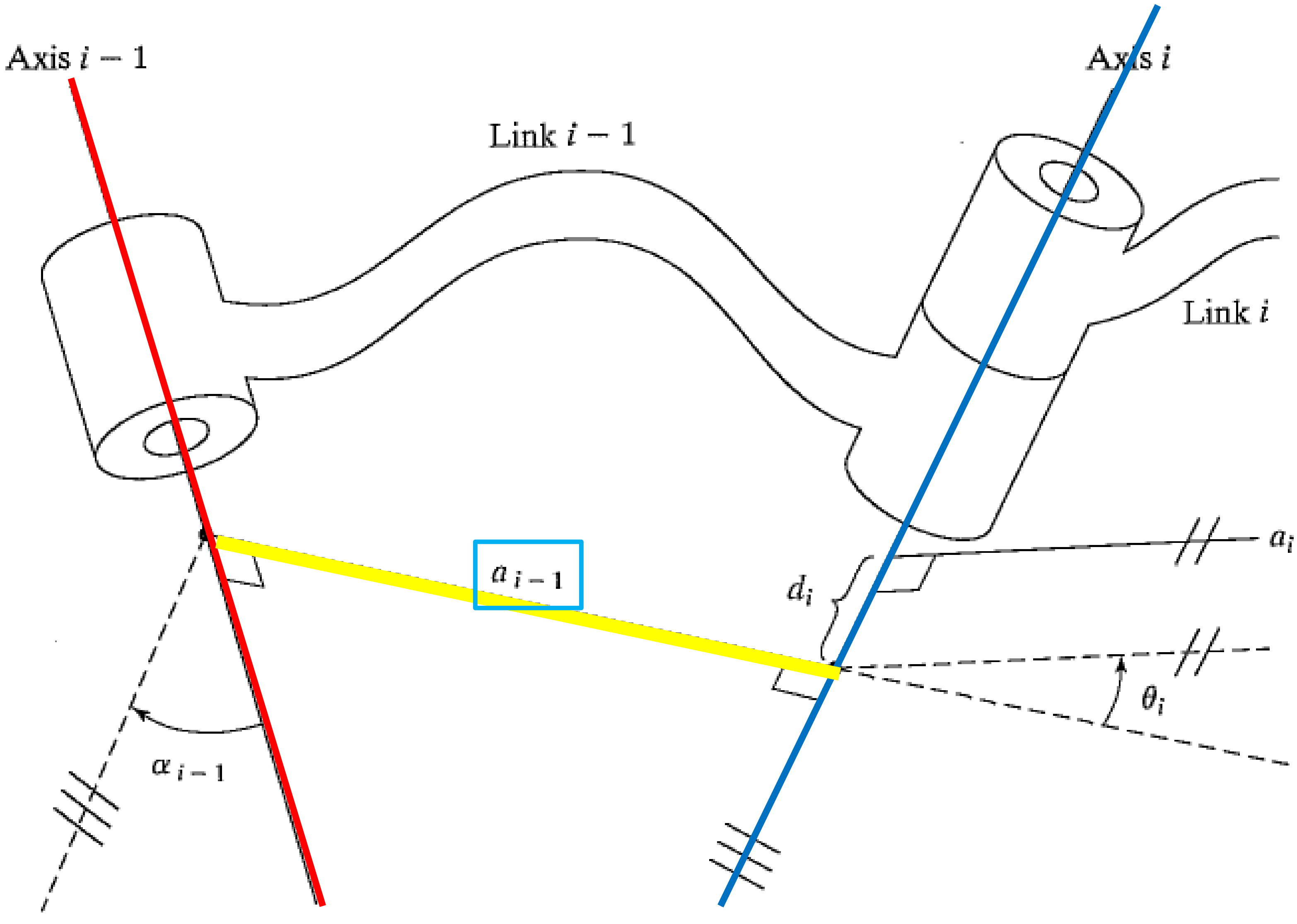


The Denavit-Hartenberg Method

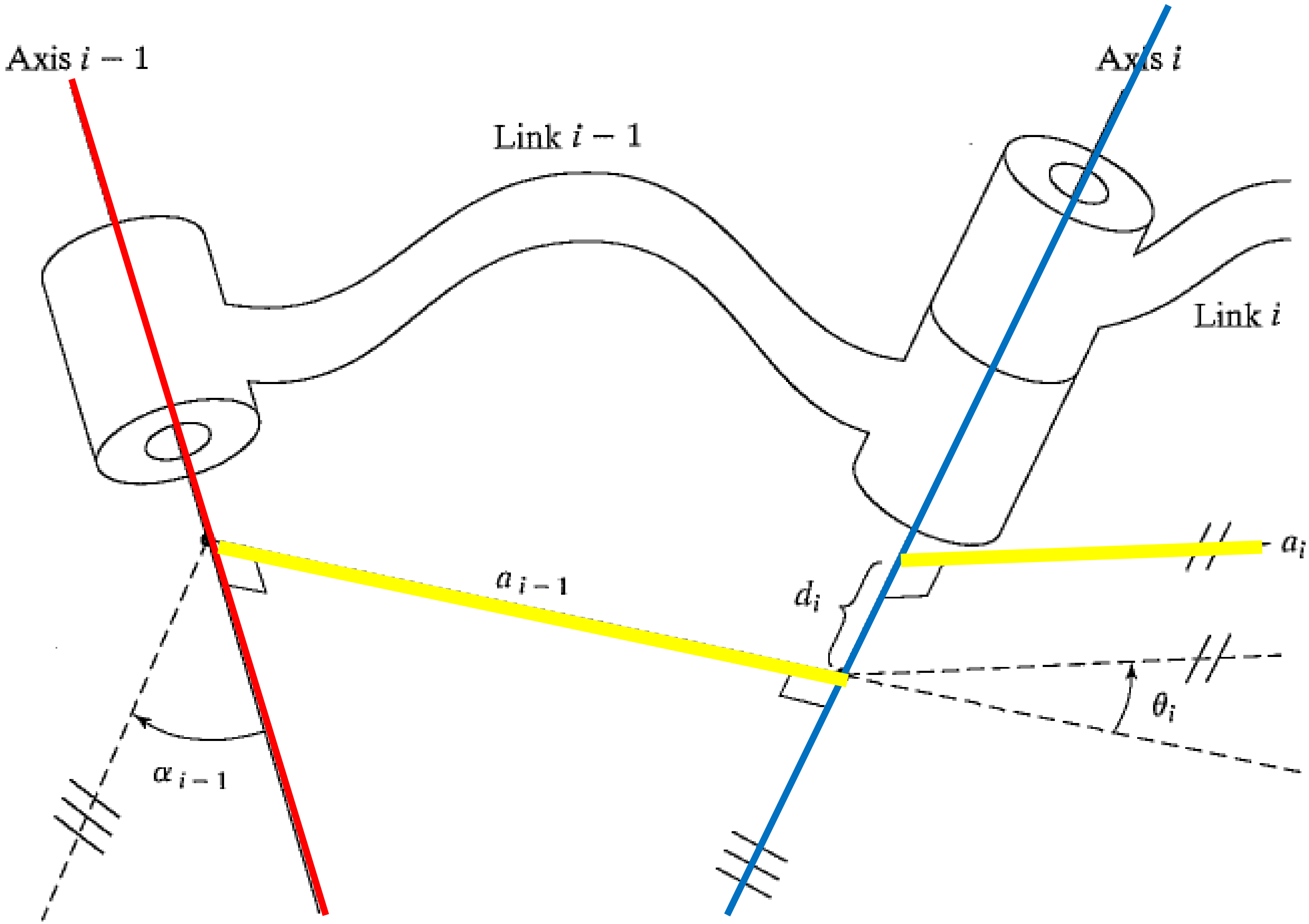
Denavit–Hartenberg Parameters



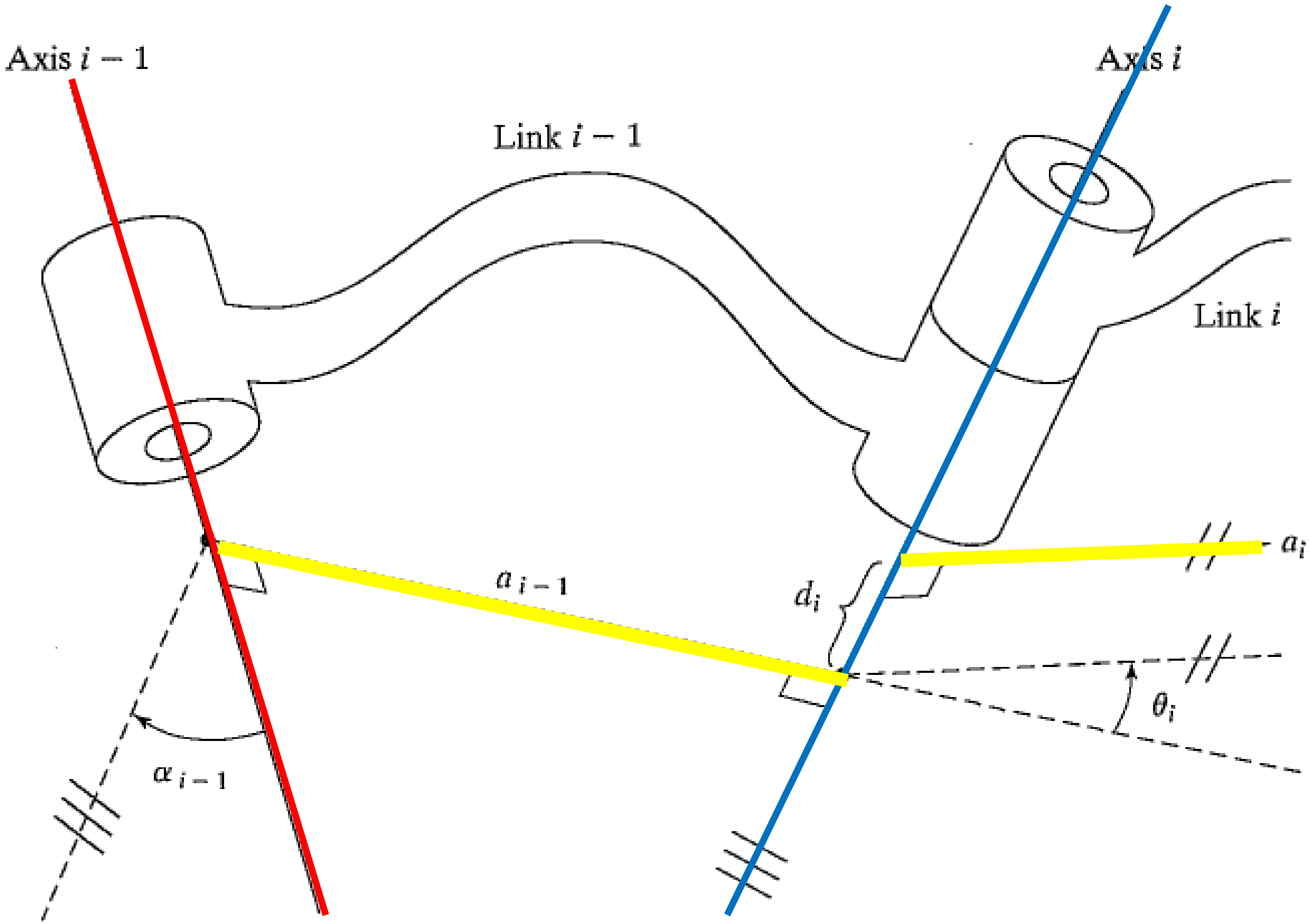
Denavit–Hartenberg Parameters



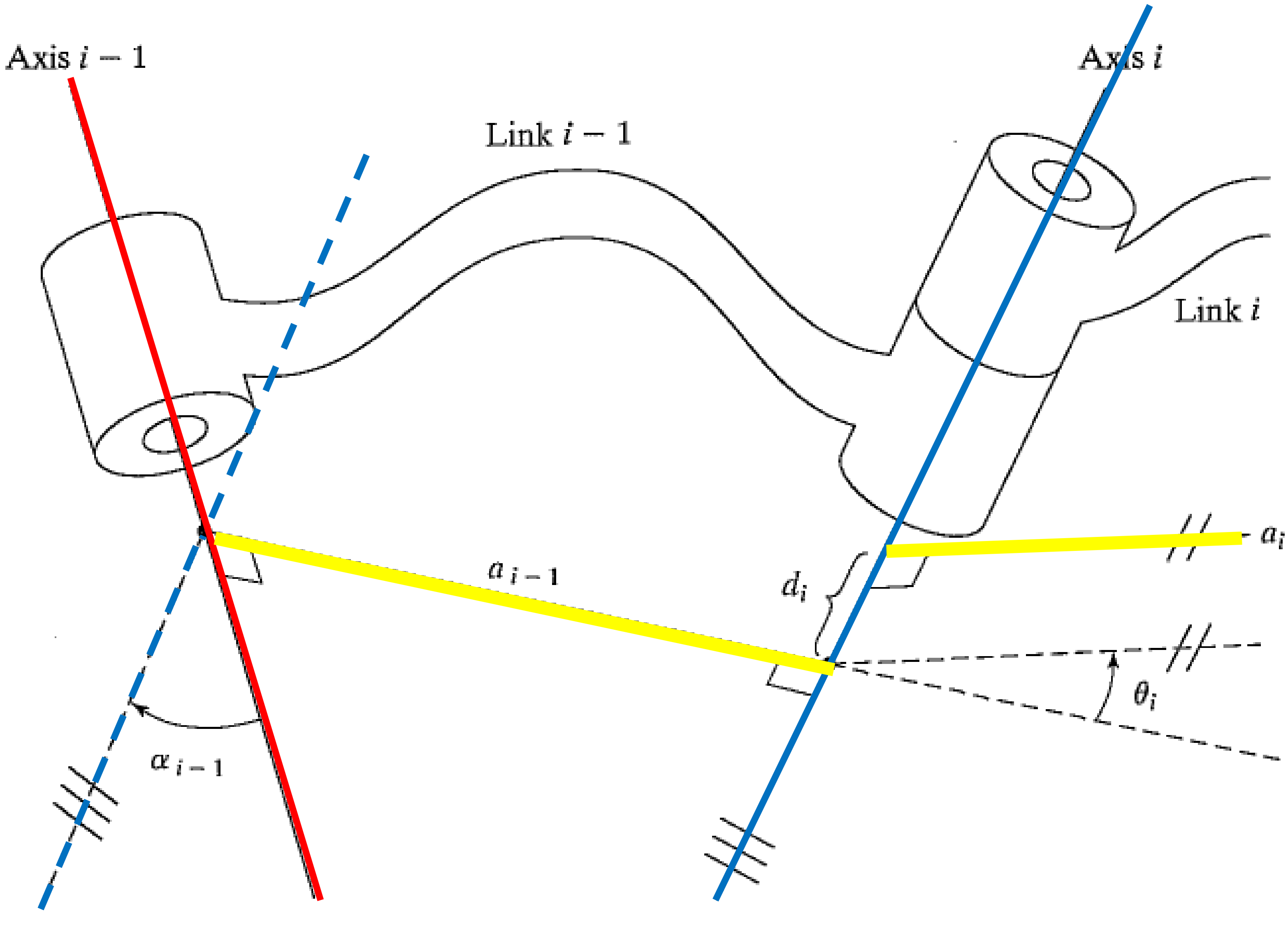
Denavit–Hartenberg Parameters



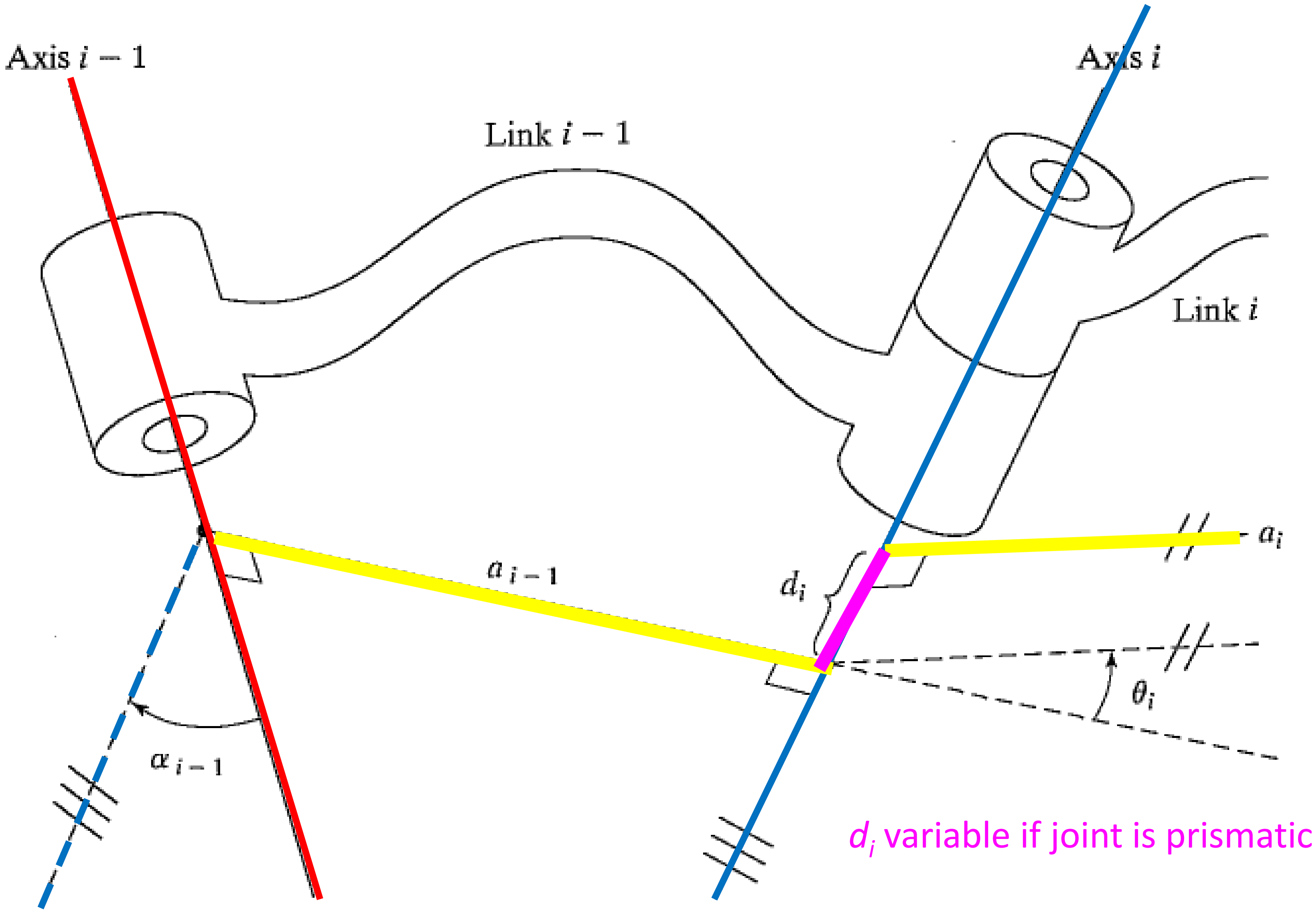
Denavit–Hartenberg Parameters



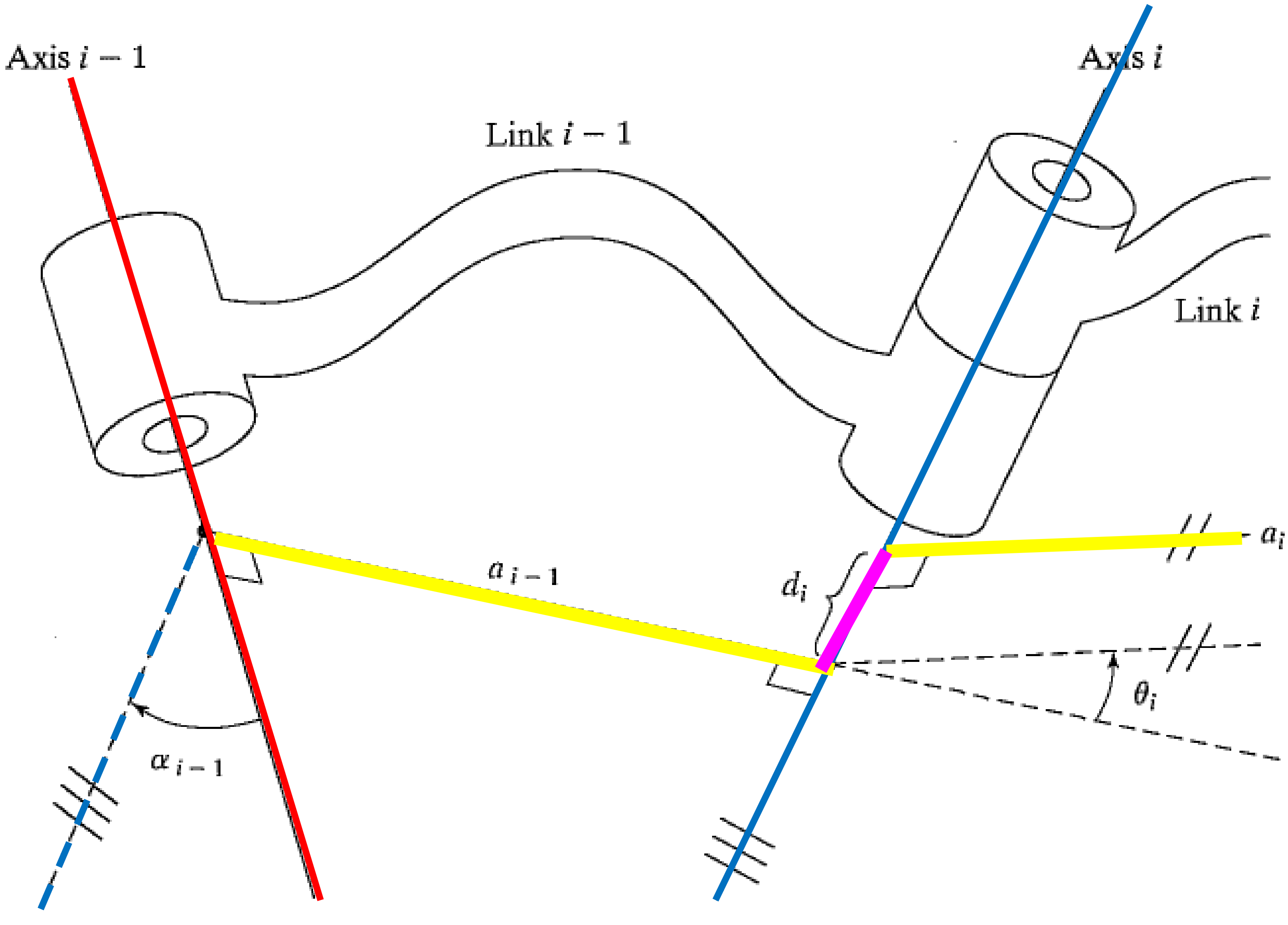
Denavit–Hartenberg Parameters



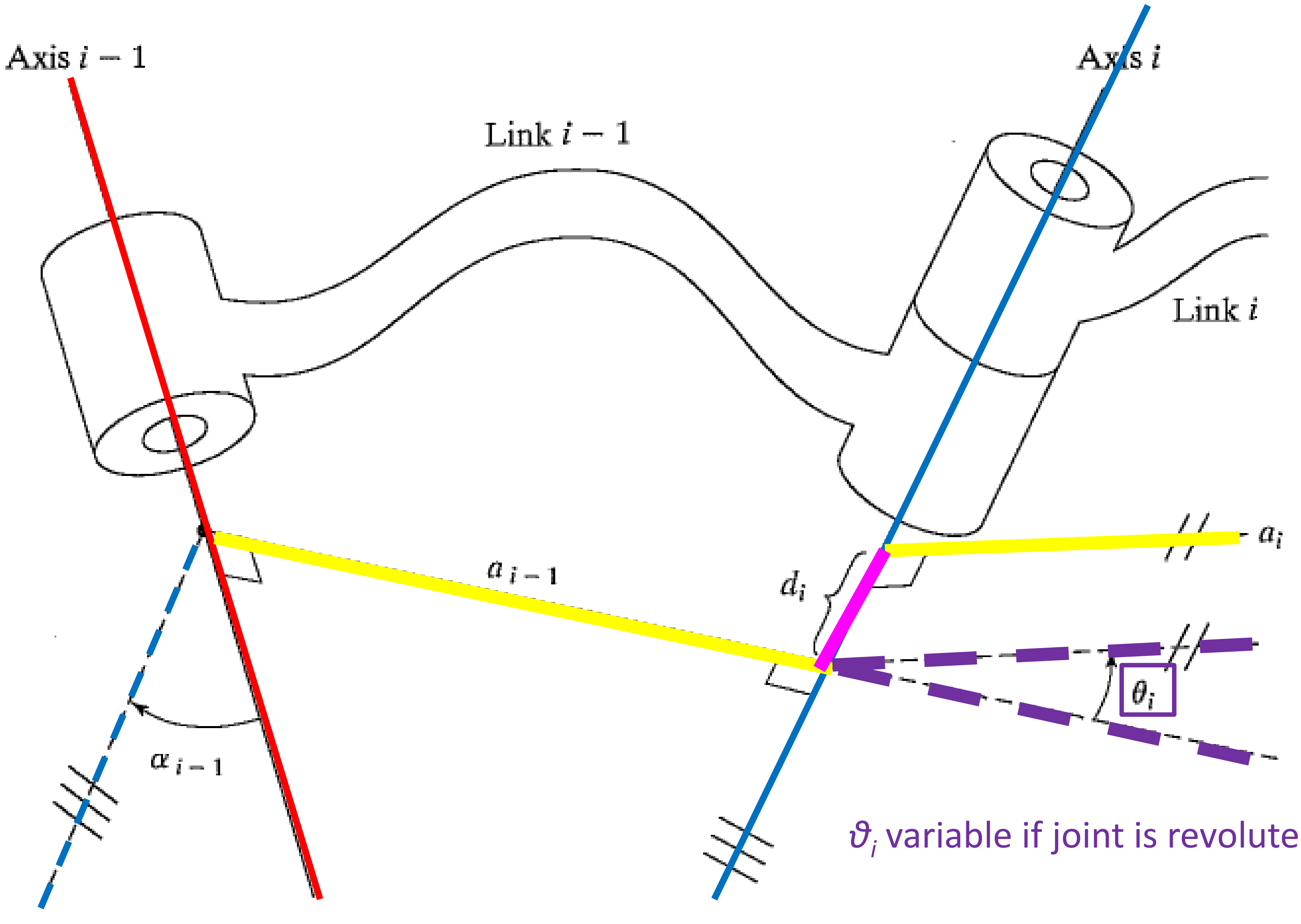
Denavit–Hartenberg Parameters



Denavit–Hartenberg Parameters



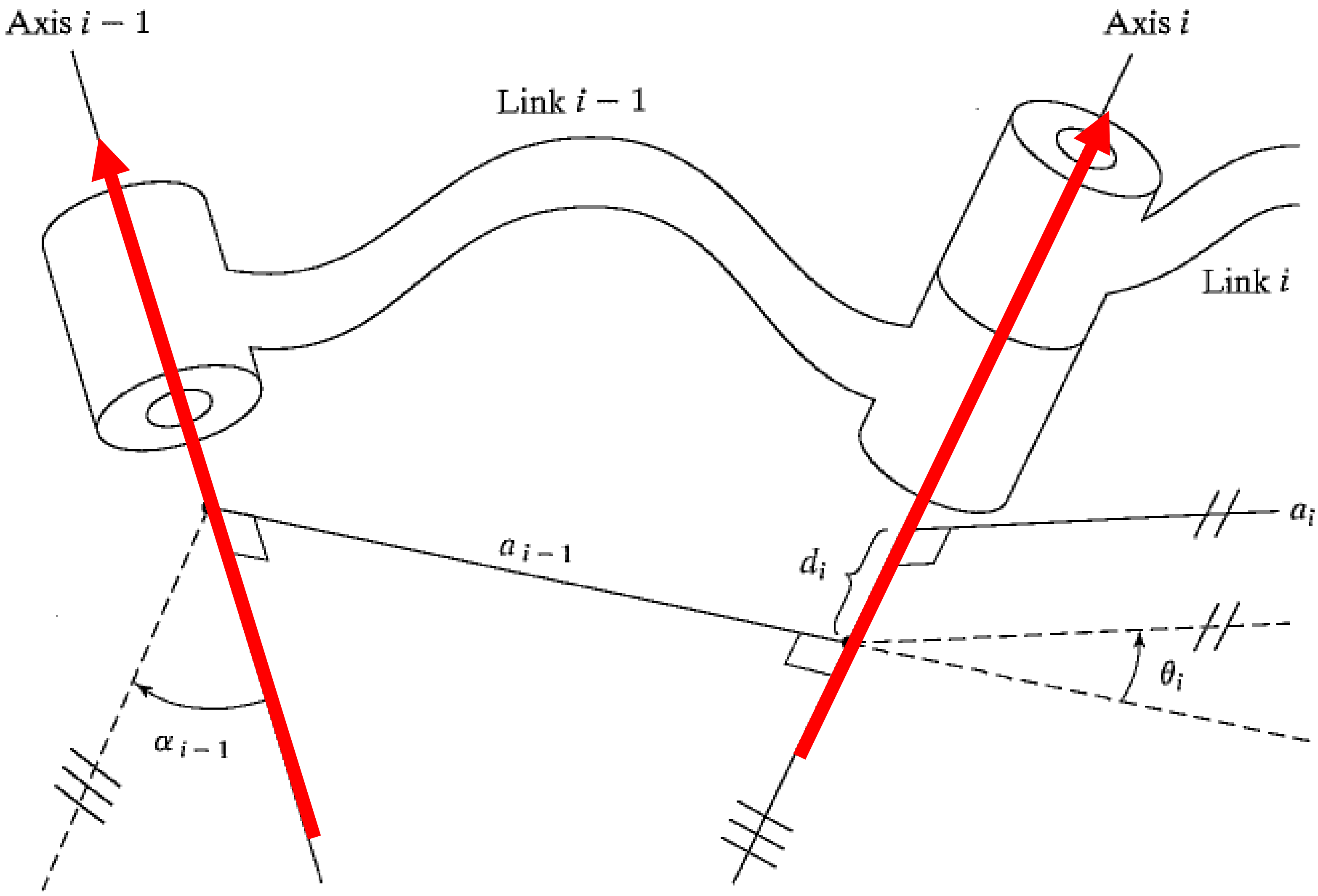
Denavit–Hartenberg Parameters



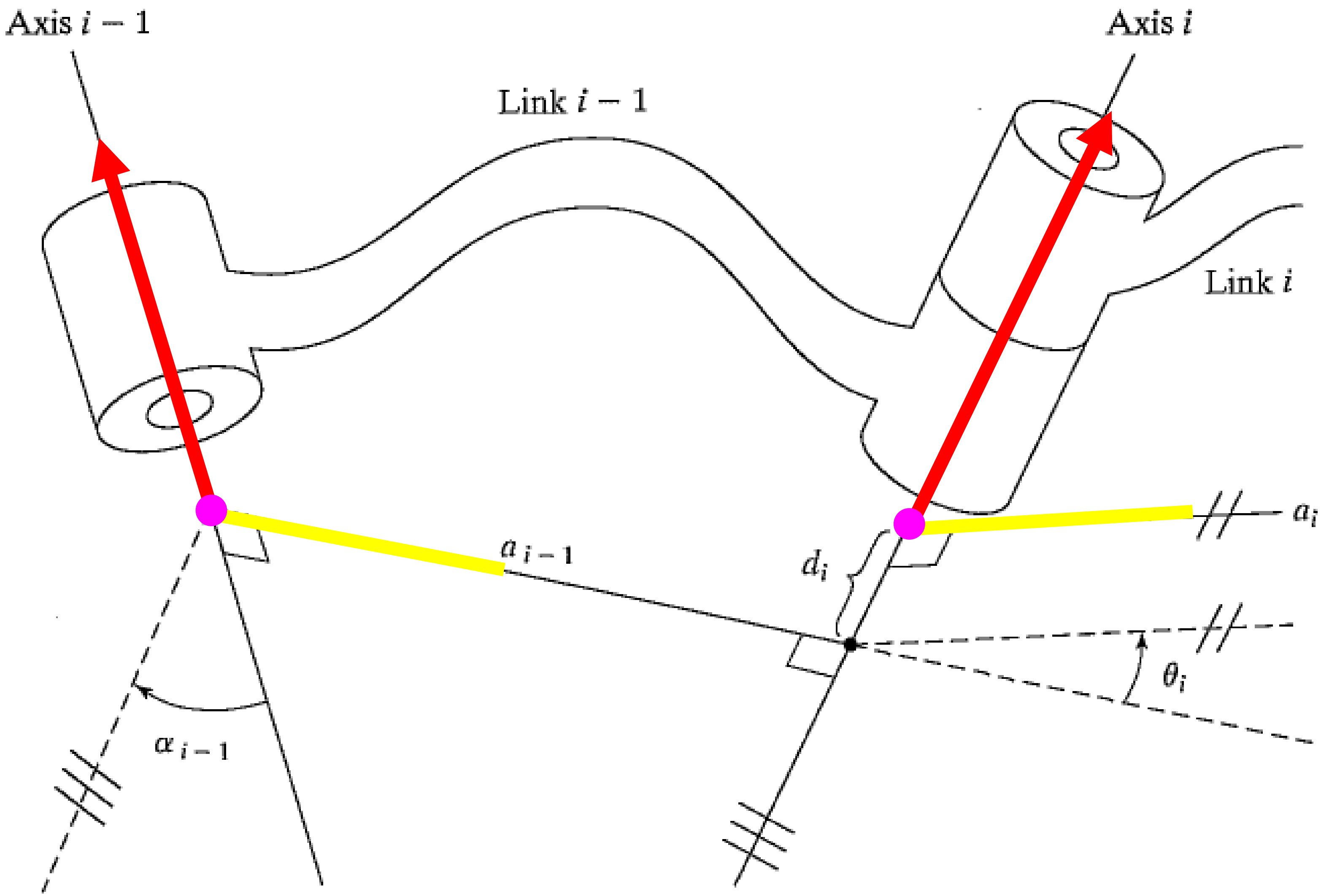
Next: Assign frames

- Two Design Principles prevail in this modeling approach
  - **Principle 1**: The Axis X_i *must be designed* to intersect Z_{i-1}
  - **Principle 2**: The Axis X_i *must be designed* to be perpendicular to Z_{i-1}

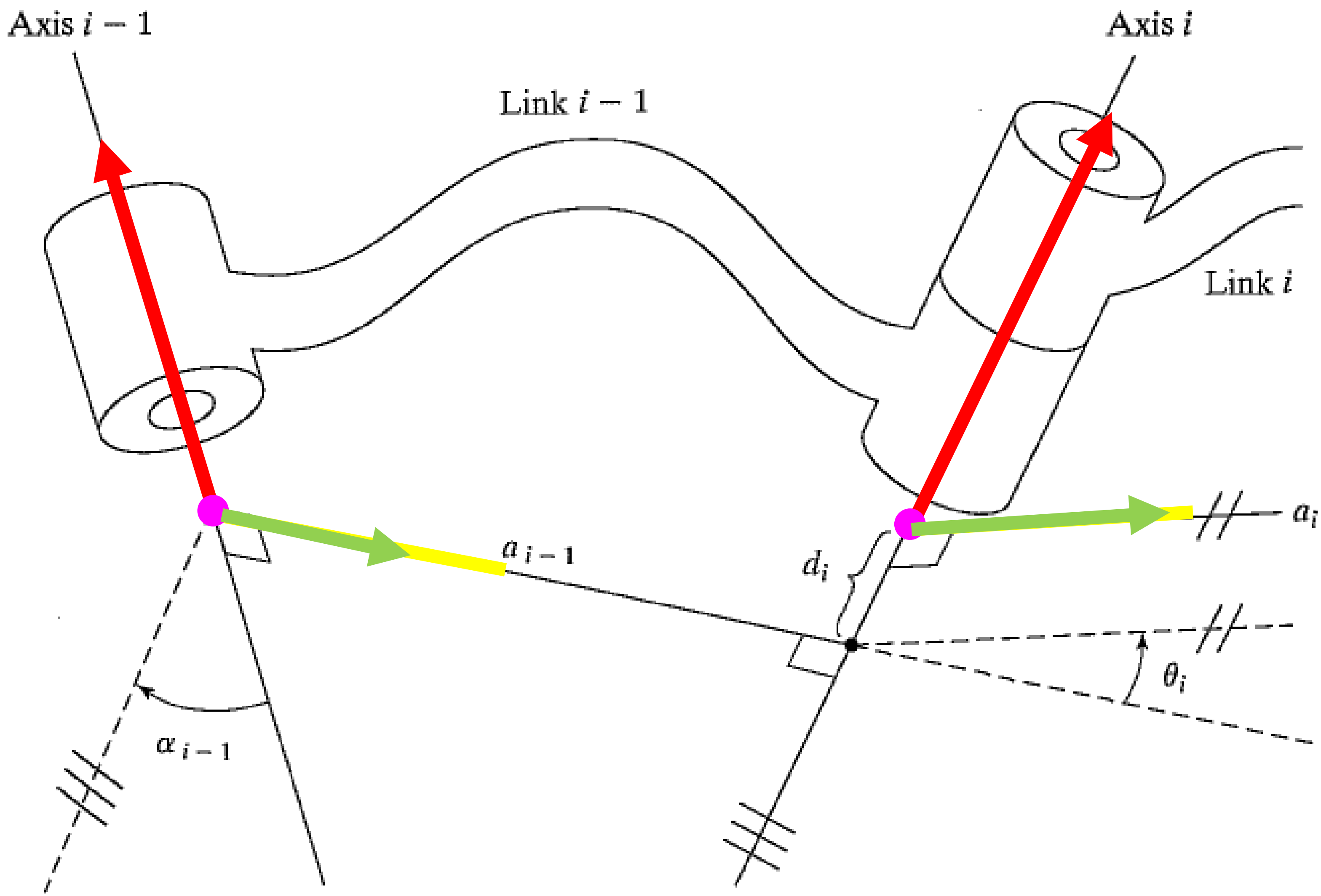
AFFIXING FRAMES TO LINKS: Z_i



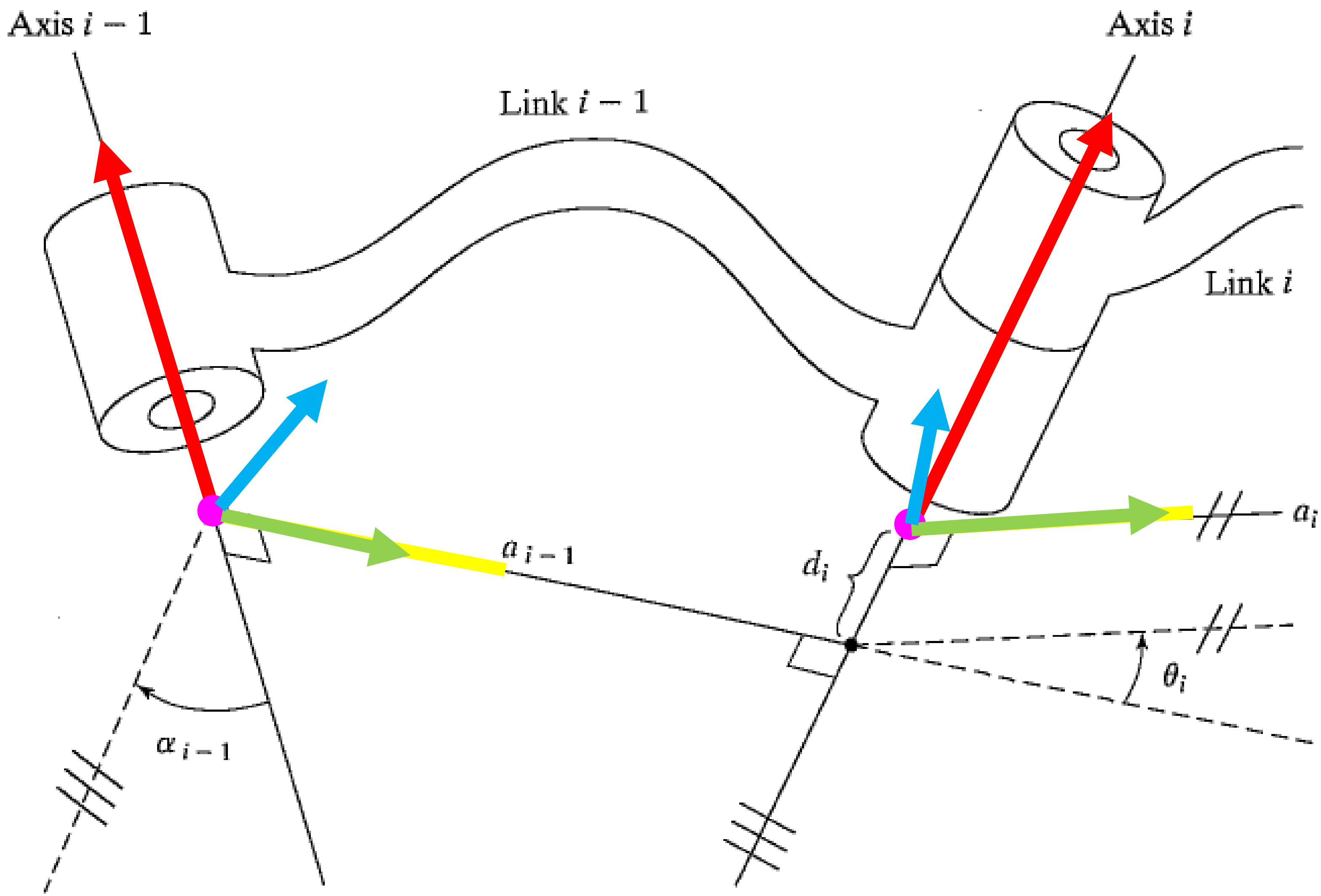
AFFIXING FRAMES TO LINKS: Locate origins



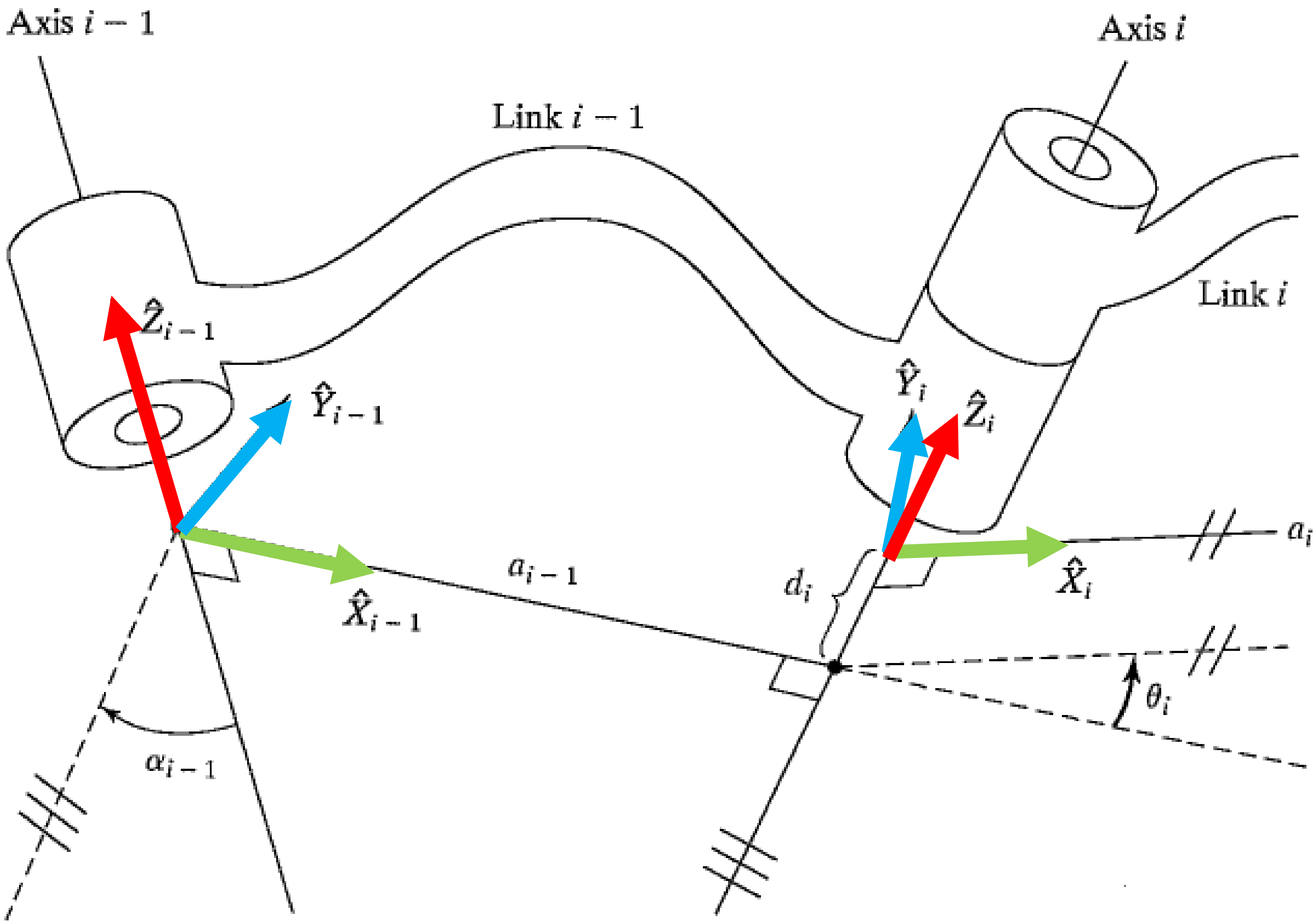
AFFIXING FRAMES TO LINKS: X_i



AFFIXING FRAMES TO LINKS: Y_i

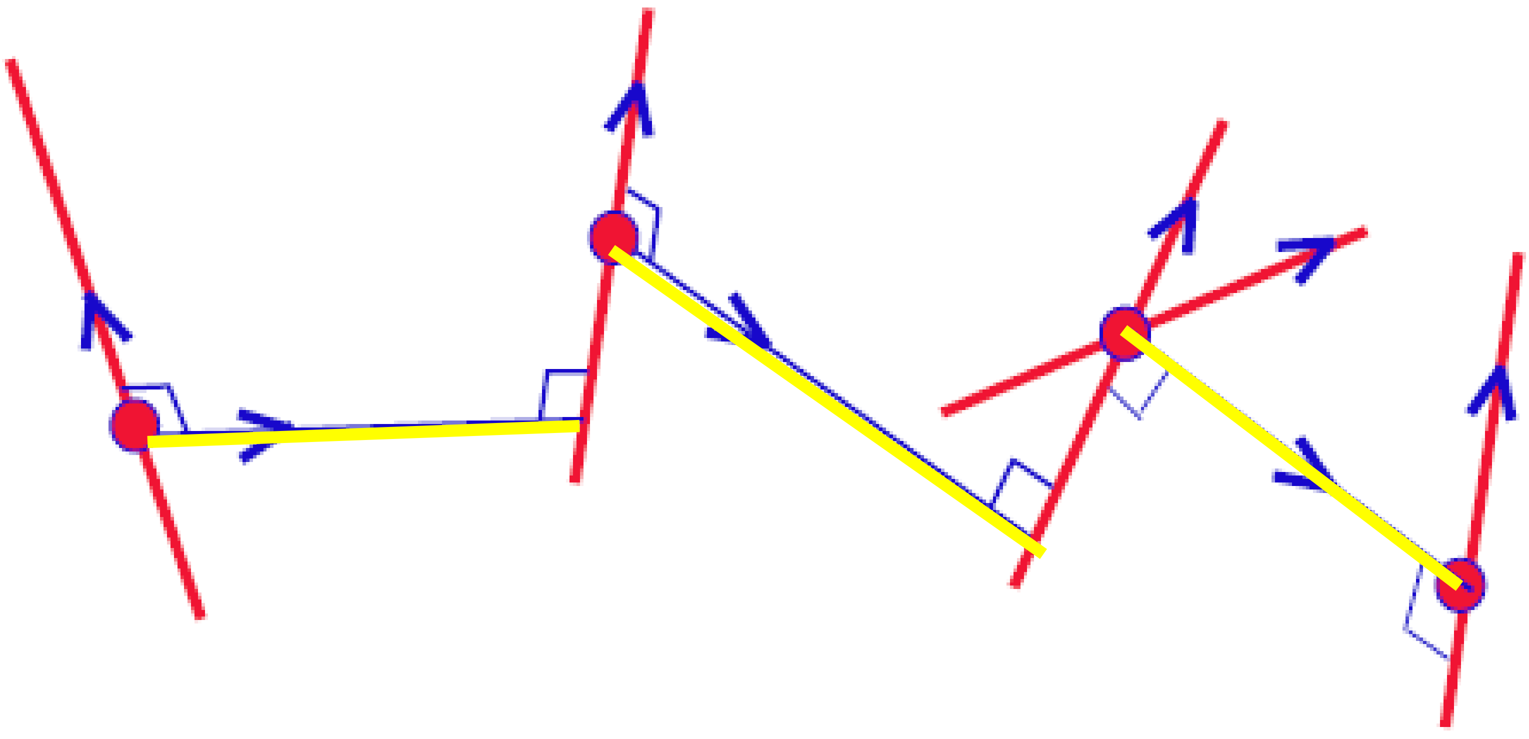


FINAL



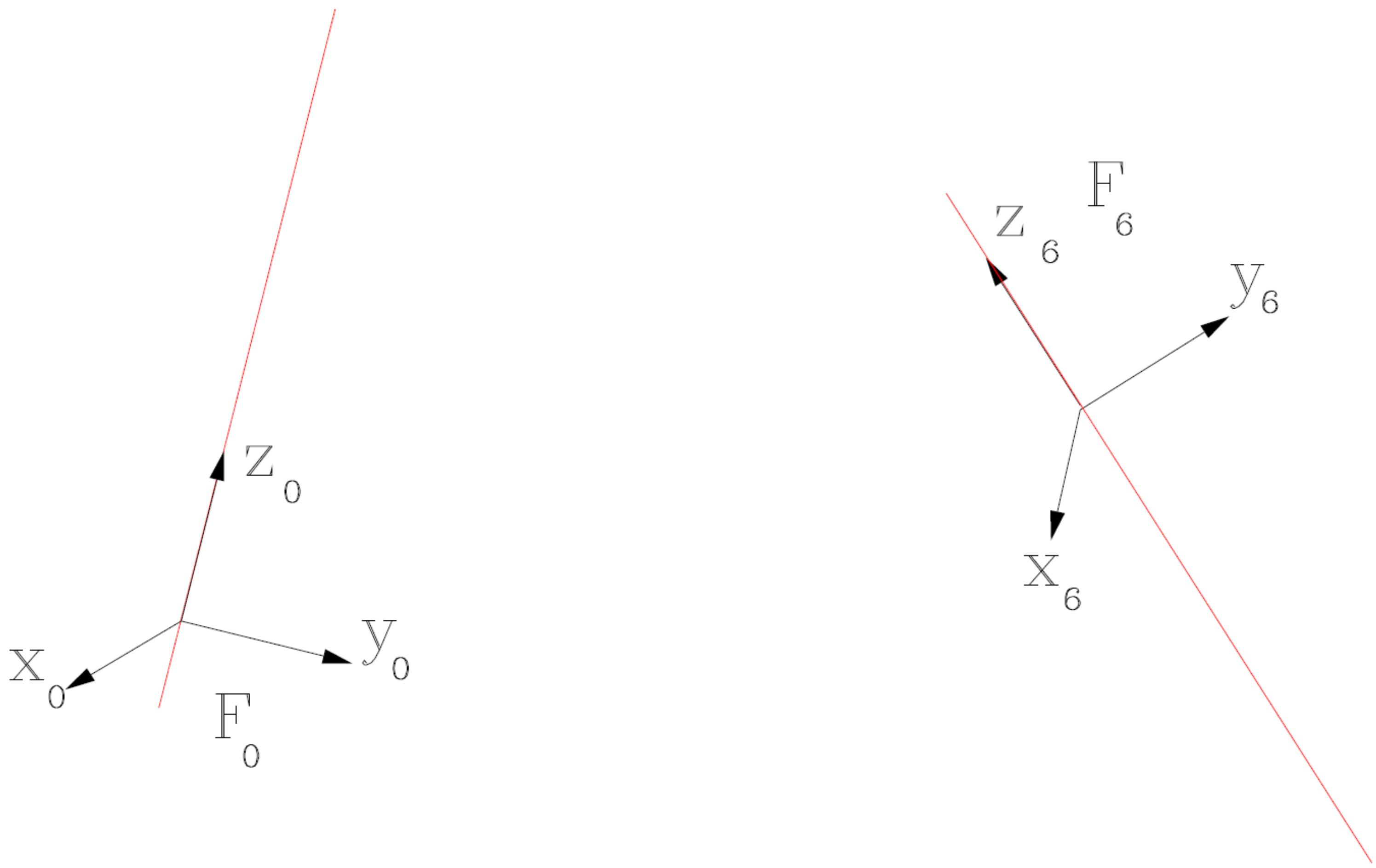
Axis $i-1$
Axis i
Link $i-1$
Link i
$\hat{Z}_{i-1}$
$\hat{Y}_{i-1}$
$\hat{X}_{i-1}$
$\hat{Y}_i$
$\hat{Z}_i$
$\hat{X}_i$
a_i
a_{i-1}
d_i
θ_i
α_{i-1}

Summary – Frame Attachment

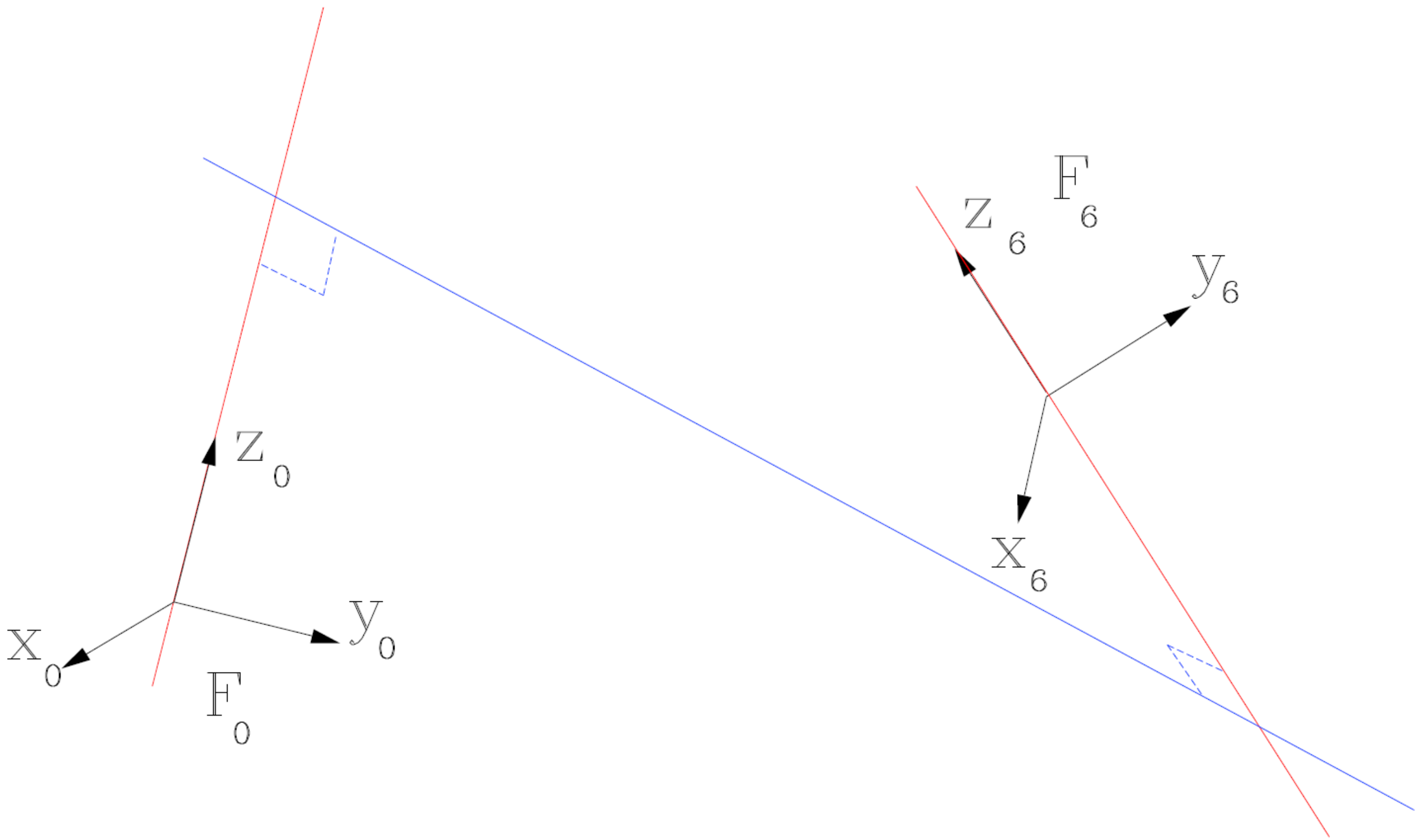


1. Normals
2. Origins
3. Z-axes
4. X-axes

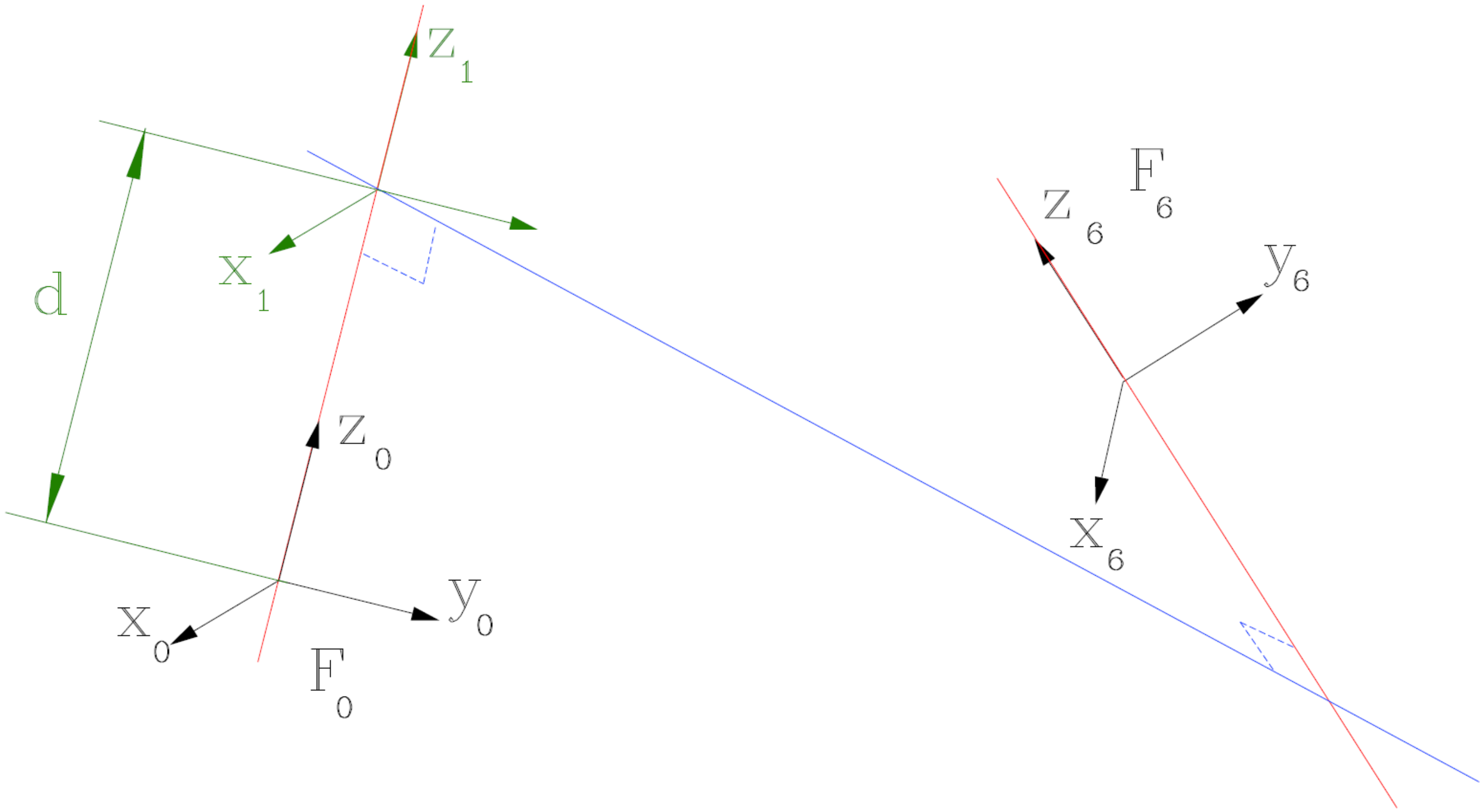
Denavit–Hartenberg Method



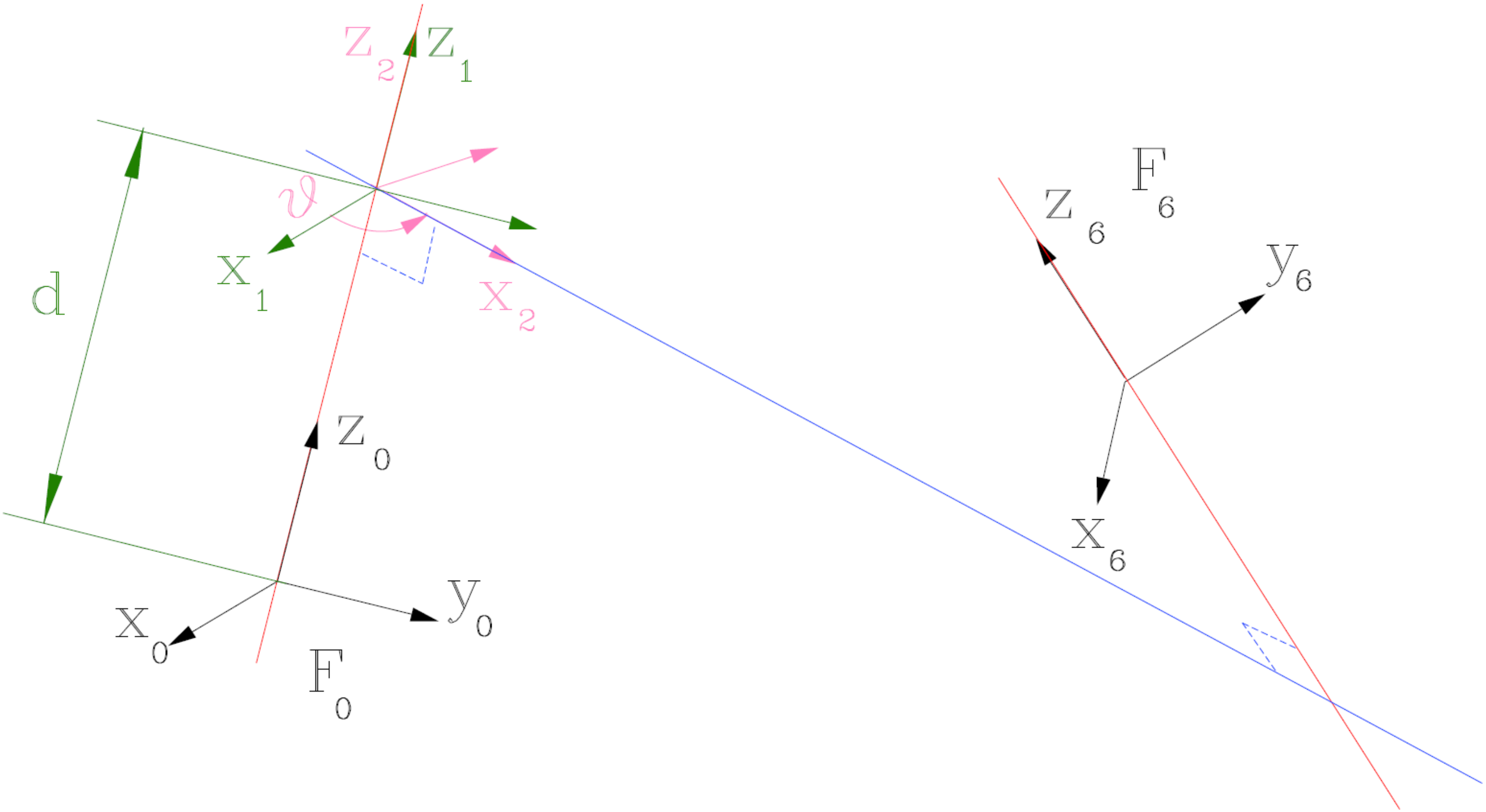
Denavit–Hartenberg Method



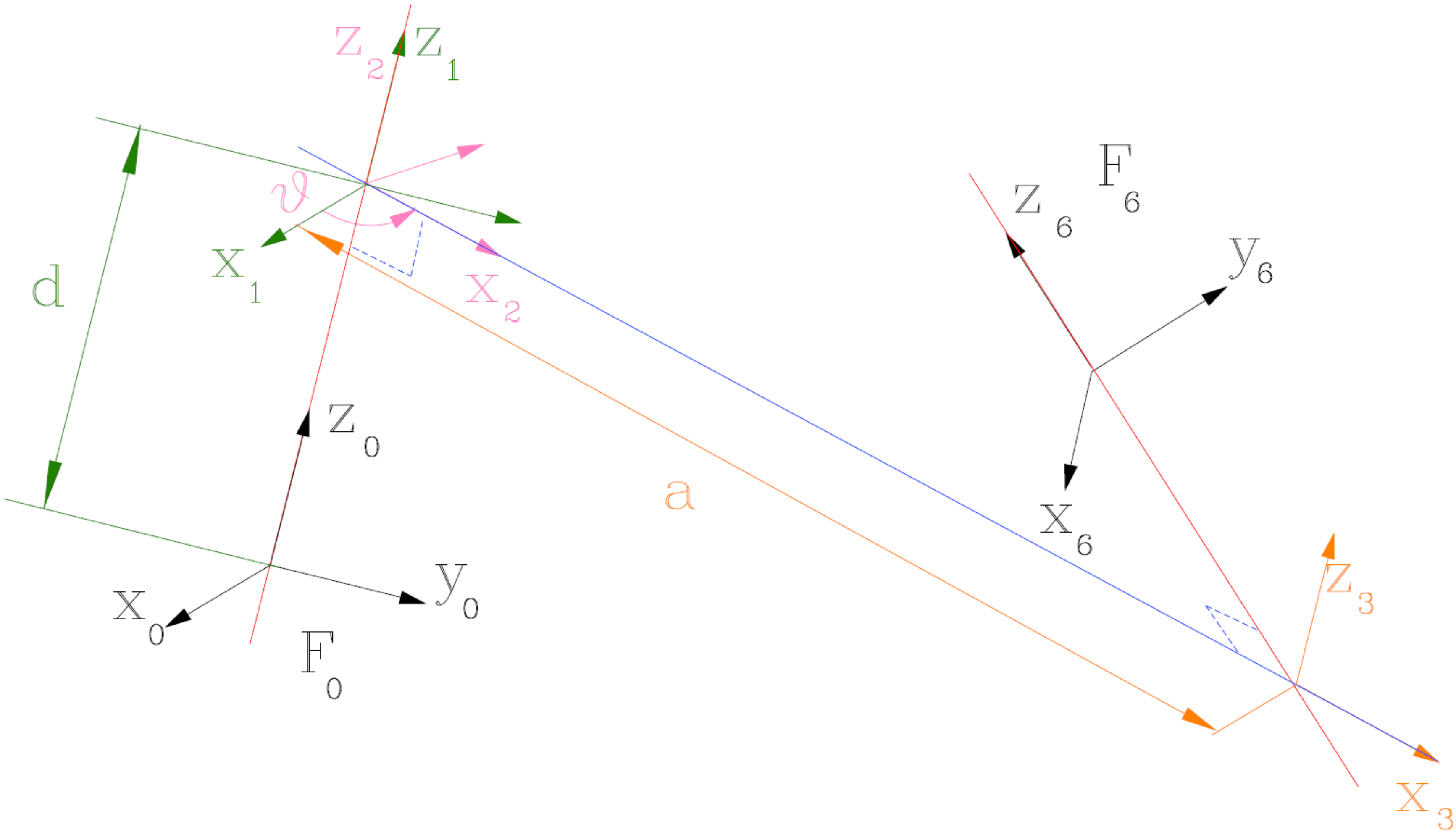
Denavit–Hartenberg Method



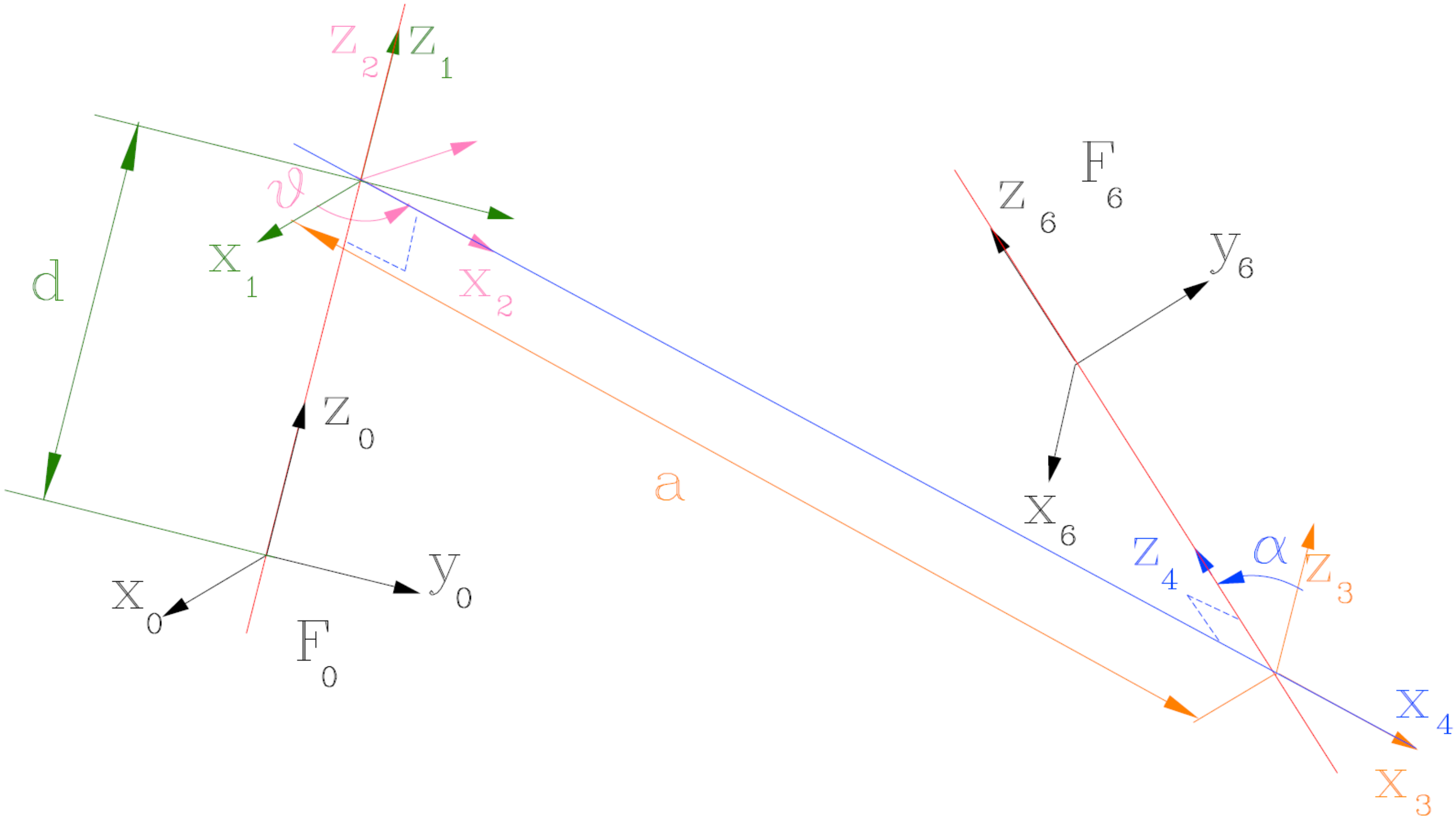
Denavit–Hartenberg Method



Denavit–Hartenberg Method



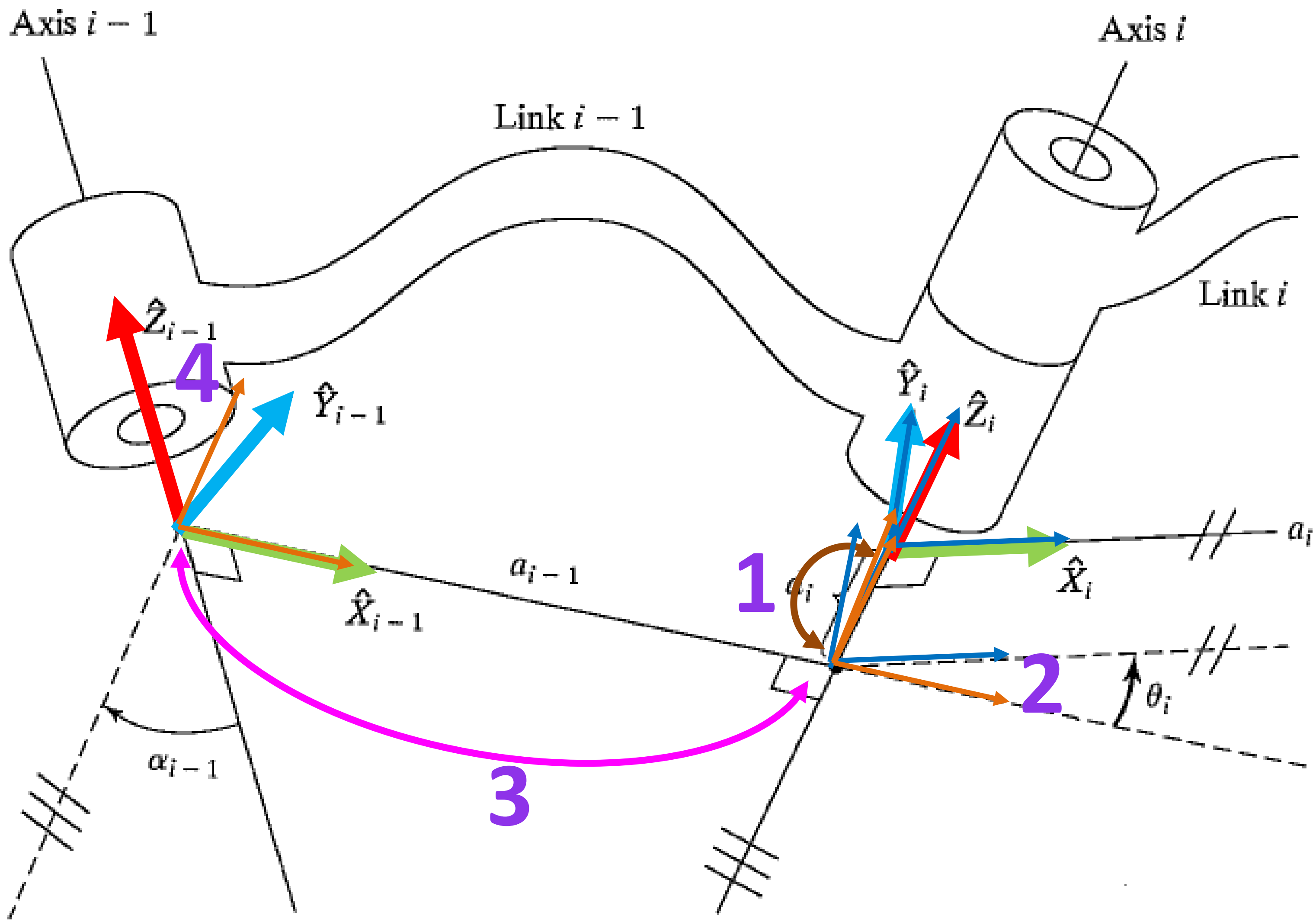
Denavit–Hartenberg Method



MOVIE 1


DH Parameter and Coordinate system assignment
Prof. Jatin Dave
Mechanical Engineering Department
Institute of Technology
Nirma University
Ahmedabad, Gujarat, India
fppt.com
Screencast-O-Matic.com

TRANSFORMATIONS



$$T(z_{i-1}, d_i) R(z_{i-1}, \theta_i) T(x_i, a_i) R(x_i, \alpha_i)$$

Overall transformation fixed $i\!-\!1$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_{i-1} & -S\alpha_{i-1} & 0 \\ 0 & S\alpha_{i-1} & C\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i\, C\alpha_{i-1} & C\theta_i\, C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1}\, d_i \\ S\theta_i\, S\alpha_{i-1} & C\theta_i\, S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Overall transformation fixed *i*

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Overall transformation

The position and orientation of the i–th frame coordinate can be expressed in the (i–1)th frame by the following homogeneous transformation matrix:

Source coordinate

Reference Coordinate

$$T_{i-1}^{i} = T(z_{i-1}, d_i)R(z_{i-1}, \theta_i)T(x_i, a_i)R(x_i, \alpha_i)$$

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

Overall transformation

Each matrix T_{i-1}^{i} is a function of the i-th joint variable, d_i or θ_i depending on the joint type. For notational ease, the joint variable is generically indicated as q_i, i.e.:

$q_i = d_i$ for prismatic joints

$q_i = \theta_i$ for rotational joints

Therefore: $T_{i-1}^{i} = T_{i-1}^{i}\left(q_i\right)$

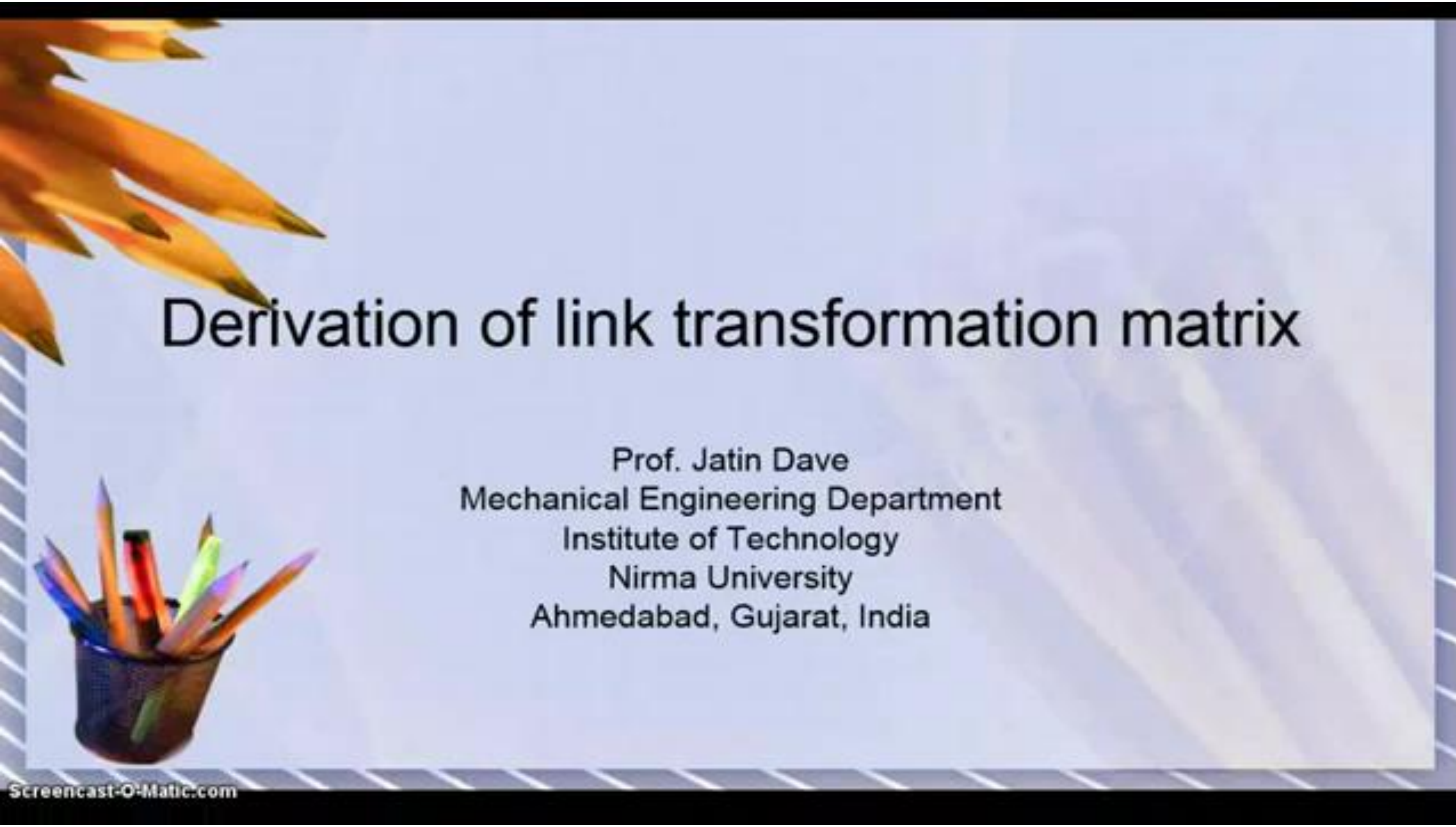
In case of a manipulator with n joints, the relationship between frame F_0 and frame F_n is:

$$T_0^n = T_0^1\left(q_1\right)T_1^2\left(q_2\right)\cdots T_{n-1}^{n}\left(q_1\right)$$

This equation expresses the position and orientation of the last link wrt the base frame, once the joint variables $q_1, q_2, \ldots, q_n$ are known.

This equation is the kinematic model of the manipulator.

MOVIE 2


Derivation of link transformation matrix
Prof. Jatin Dave
Mechanical Engineering Department
Institute of Technology
Nirma University
Ahmedabad, Gujarat, India
Screencast-O-Matic.com

Algorithm 2-5-1: D-H Representation

0. Number the joints from 1 to n starting with the base and ending with the tool yaw, pitch, and roll, in that order.
1. Assign a right-handed orthonormal coordinate frame L_0 to the robot base, making sure that z^0 aligns with the axis of joint 1. Set $k = 1$.
2. Align z^k with the axis of joint $k + 1$.
3. Locate the origin of L_k at the intersection of the z^k and z^{k-1} axes. If they do not intersect, use the intersection of z^k with a common normal between z^k and z^{k-1}.
4. Select x^k to be orthogonal to both z^k and z^{k-1}. If z^k and z^{k-1} are parallel, point x^k away from z^{k-1}.
5. Select y^k to form a right-handed orthonormal coordinate frame L_k.
6. Set $k = k + 1$. If $k < n$, go to step 2; else, continue.
7. Set the origin of L_n at the tool tip. Align z^n with the approach vector, y^n with the sliding vector, and x^n with the normal vector of the tool. Set $k = 1$.
8. Locate point b^k at the intersection of the x^k and z^{k-1} axes. If they do not intersect, use the intersection of x^k with a common normal between x^k and z^{k-1}.
9. Compute θ_k as the angle of rotation from x^{k-1} to x^k measured about z^{k-1}.
10. Compute d_k as the distance from the origin of frame L_{k-1} to point b^k measured along z^{k-1}.
11. Compute a_k as the distance from point b^k to the origin of frame L_k measured along x^k.
12. Compute α_k as the angle of rotation from z^{k-1} to z^k measured about x^k.
13. Set $k = k + 1$. If $k \leq n$, go to step 8; else, stop.

Denavit–Hartenberg Algorithm v2

1: Numerate links beginning with *1* (first mobile link of link's chain) and ending with n (last mobile link). The fixed base reference coordinate system will be numbered as link *0*.
2: Numerate each articulation beginning with *1* (that is the first DOF for a joint) and ending with n.
3: Locate axis of each articulation. If this is *revolving*, the axis will be its own turn axis. If it is *prismatic*, it will be the axis along which the displacement takes place.
4: For $n+1$ of link *0* to n locate Z_{n+1} axis on the axis of articulation n.
5: Place the origin of the base reference coordinate system in any point of z_0 axis. Axes z_0 and y_0 will be located so that they form a right-handed system with z_0.
6: For $n+1$ of link *1* to n, place the $\{S_j\}$ system with regard to the link $n+1$) in the intersection of Z_{n+1} axis with the normal line common to Z_n and Z_j. If both axes cuts, $\{S_j\}$ would be located in the cut point. If they were parallel then $\{S_j\}$ would be located in the articulation n.

Denavit–Hartenberg Algorithm v3

Step 1: Locate and label the joint axes $z_0, \cdots z_{n-1}$.

Step 2: Establish the base frame. Set the origin anywhere on the z_0 axis. The x_0 and y_0 axes are chosen conveniently to form the right-hand frame.

For $i = 1, \cdots n-1$, perform steps 3 to 5

Step 3: Locate the origin o_i where the common normal to z_i and z_{i-1} intersects z_i. If z_i intersects z_{i-1} locate o_i at this intersection. If z_i and z_{i-1} are parallel, locate o_i at joint i.

Step 4: Establish x_i along the common normal between z_{i-1} and z_i through o_i, or in the direction normal to the $z_{i-1} \times z_i$ plane if z_{i-1} and z_i intersect.

Step 5: Establish y_i to complete a right hand frame.

Step 6: Establish the end-effector frame $o_n x_n y_n z_n$. Set z_n along the direction z_{n-1}. Establish the origin o_n conveniently along z_n preferably at the center of the gripper or at the tip of any tool that the manipulator may be carrying. Establish x_n along the common normal between z_{n-1} and z_n through o_n.

Step 7: Create a table of link parameters $\theta_i, d_i, a_i, \alpha_i$.

θ_i : the angle between x_{i-1} and x_i measured about z_{i-1}. θ_i is variable if joint i is revolute.

d_i : distance along z_{i-1} from o_{i-1} to the intersection of the x_i and z_{i-1} axes. d_i is variable if joint i is prismatic.

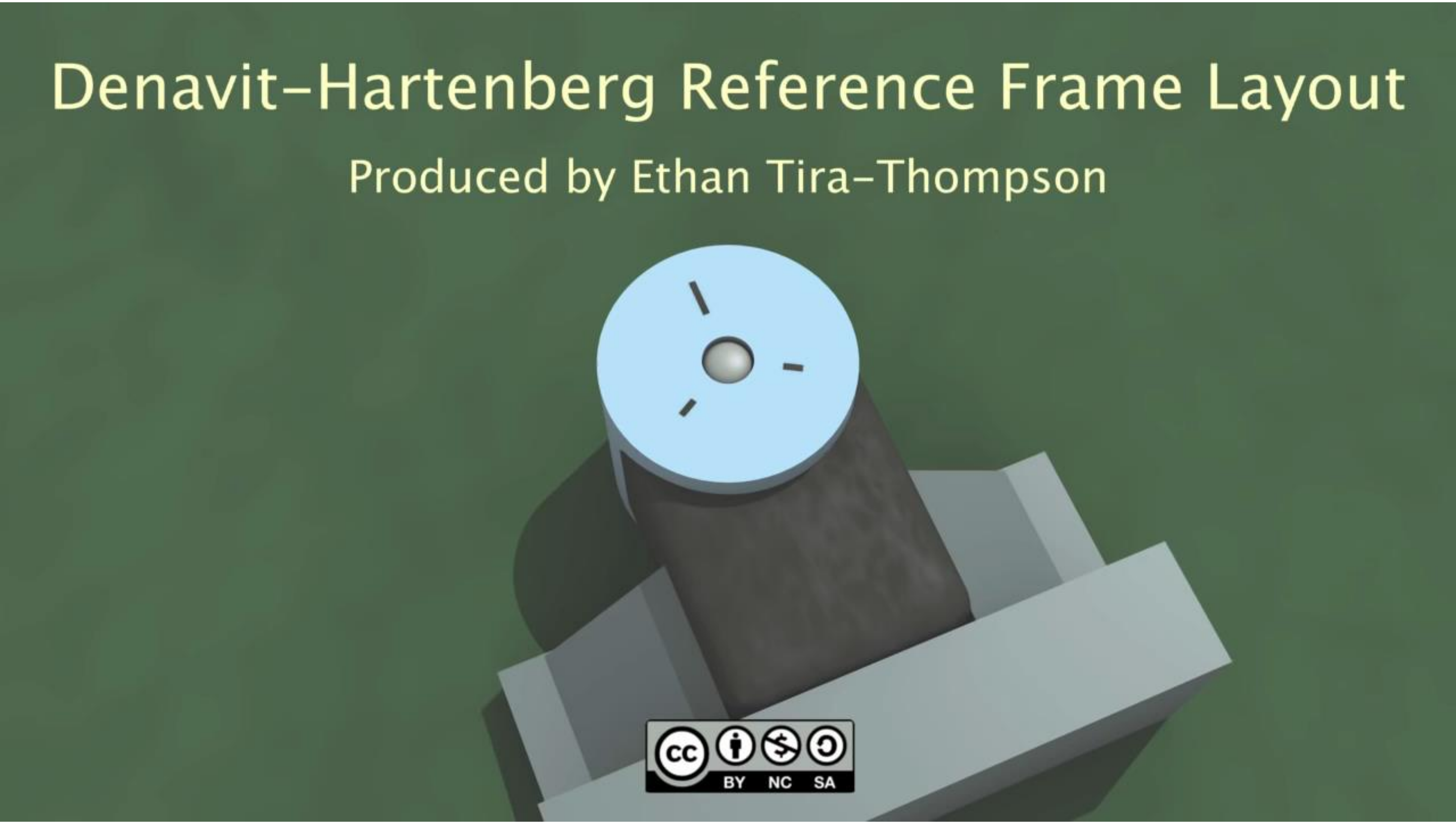
a_i : distance along x_i from o_i to the intersection of the x_i and z_{i-1} axes.

α_i : the angle between z_{i-1} and z_i measured about x_i.

Step 8: Form the homogeneous transformation matrices A_i by substituting the above parameters.

Step 9: Form ${}^0T_n = A_1 A_2 \cdots A_n$. This then gives the position and orientation of the tool frame expressed in base coordinates.

MOVIE


Denavit–Hartenberg Reference Frame Layout
Produced by Ethan Tira–Thompson
cc BY NC SA

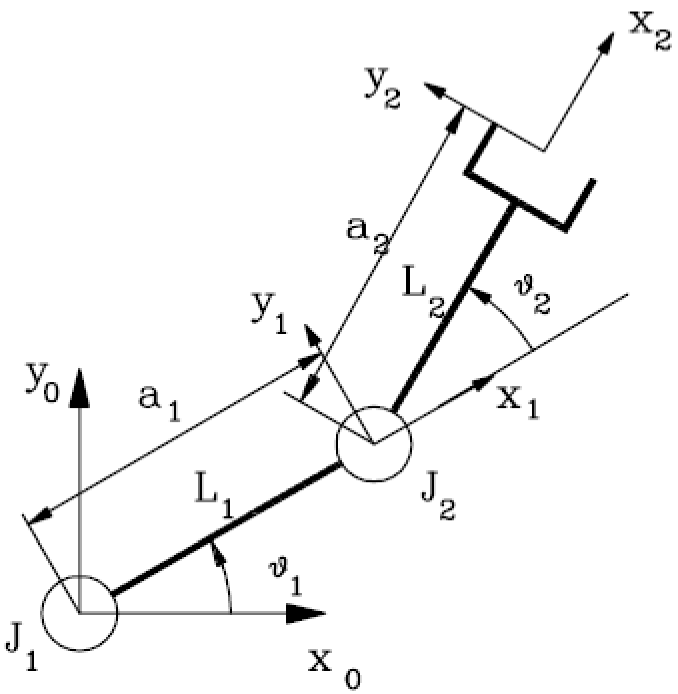
The Denavit−Hartenberg Matrix

$$\begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i\, C\alpha_{i-1} & C\theta_i\, C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1}\, d_i \\ S\theta_i\, S\alpha_{i-1} & C\theta_i\, S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The Denavit−Hartenberg Matrix is an homogeneous transformation matrix from one coordinate frame to the next.
- Using a series of D−H Matrix multiplications and the **D−H Parameter table**, the final result is a transformation matrix from some frame to your initial frame.

The Denavit–Hartenberg Parameter table

- **Two–link manipulator arm**

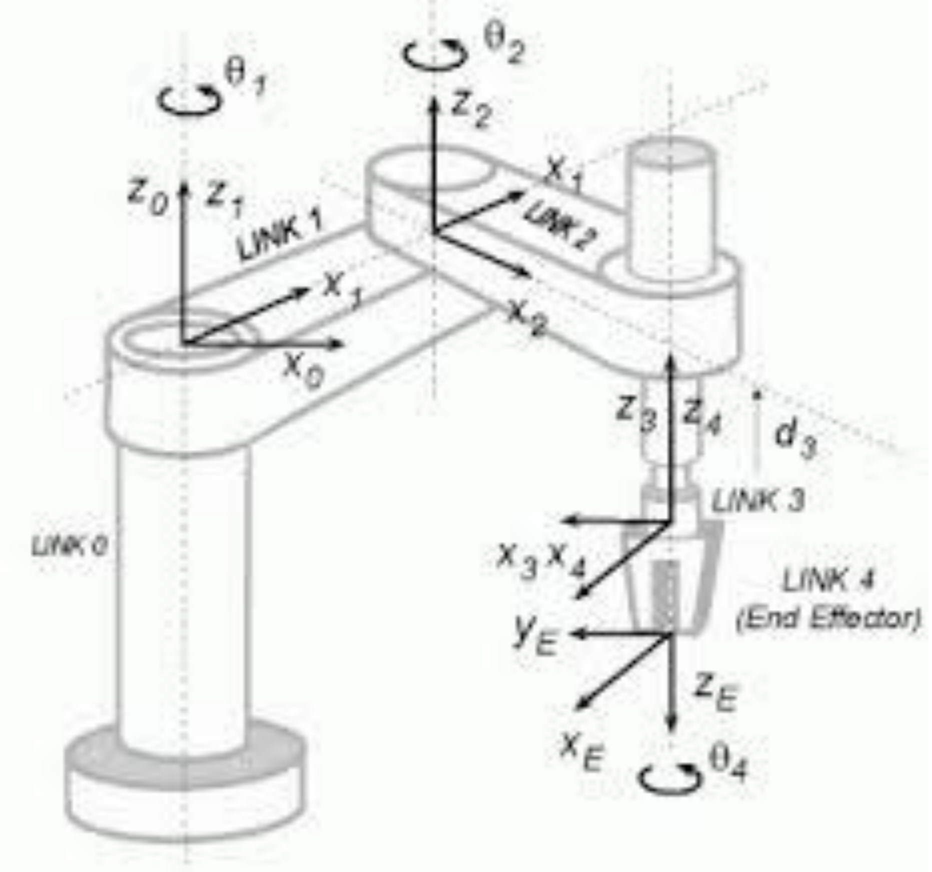


	d	θ	a	α
L1	0	θ_1	a_1	0^o
L2	0	θ_2	a_2	0^o

$$T_0^1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_1^2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The Denavit–Hartenberg Parameter table

- **SCARA arm, the Selective Compliant Articulated Robot for Assembly**
- two revolute joints (elbow, “wrist”) and one prismatic joint



The Denavit−Hartenberg Parameter table

- **SCARA arm, the Selective Compliant Articulated Robot for Assembly**
- two revolute joints (elbow, “wrist”) and one prismatic joint

DH parameter	θ	d	a	α
Joint 1	θ_1*	d_1	0	0
Joint 2	θ_2*	0	r_2	π
Joint 3	0	d_3*	0	0

- * indicates the moving joint variable
- *the end-tool is not included

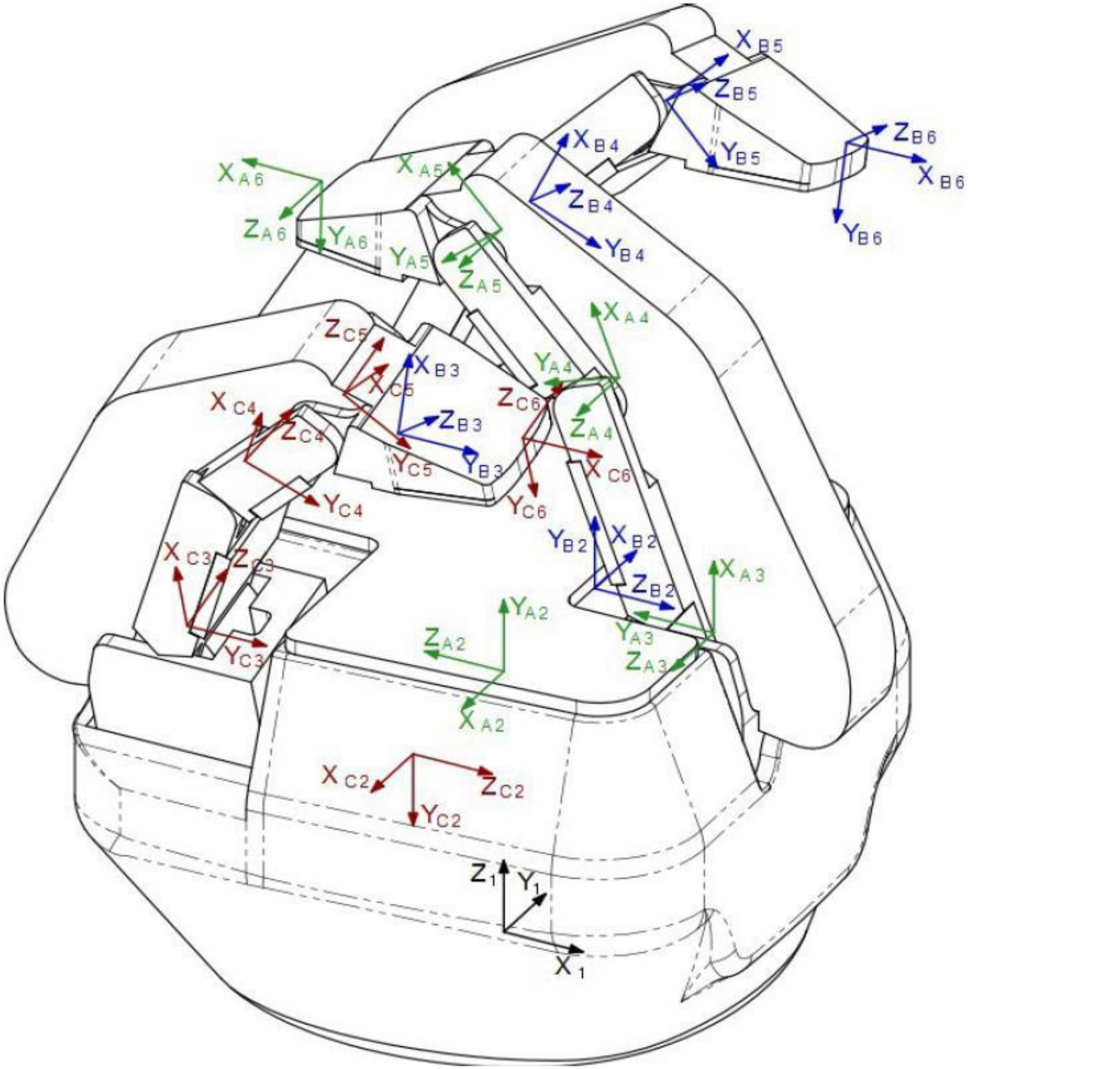
The Denavit–Hartenberg Parameter table

- **Spherical wrist**
- can be used to achieve any desired orientation of the end effector

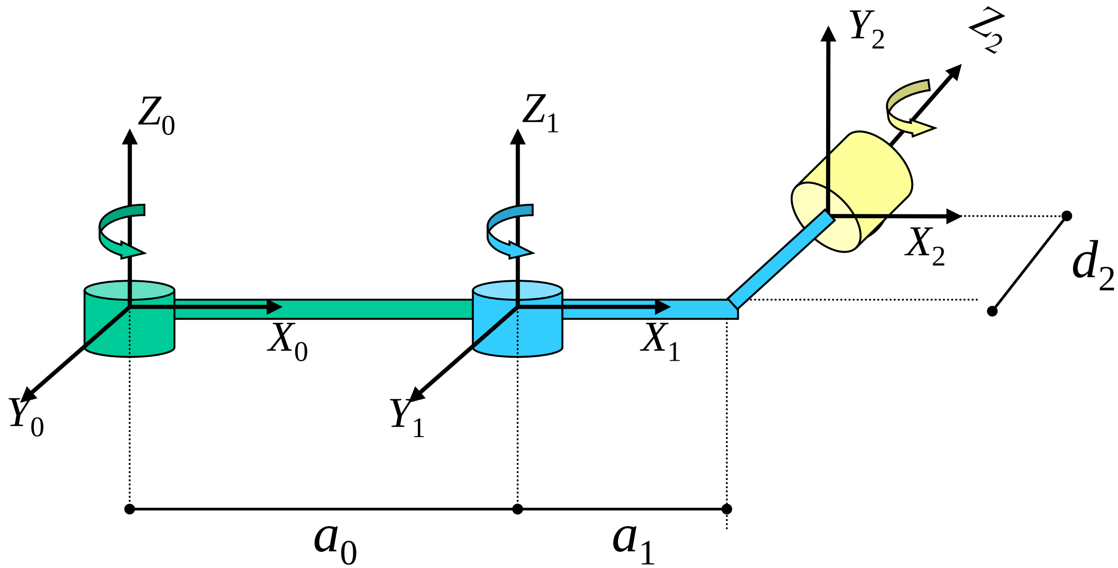
DH parameter	θ	d	a	α
Joint 1	$\theta_1{}^*$	d_1	0	$-\pi/2$
Joint 2	$\theta_2{}^*$	0	0	$\pi/2$
Joint 3	$\theta_3{}^*$	d_3	0	0

- * indicates the moving joint variable

APPLICATION



3 Revolute Joints



i	$\alpha_{(i-1)}$	$a_{(i-1)}$	d_i	θ_i
0	0	0	0	θ_0
1	0	a_0	0	θ_1
2	-90	a_1	d_2	θ_2

$$T = ({}_{0}T)({}^{0}_{1}T)({}^{1}_{2}T)$$

$$V^{X_0 Y_0 Z_0} = T \begin{bmatrix} V^{X_2} \\ V^{Y_2} \\ V^{Z_2} \\ 1 \end{bmatrix}$$

Note: T is the D−H matrix with $(i-1) = 0$ and $i = 1$.

i	$\alpha_{(i-1)}$	$a_{(i-1)}$	d_i	θ_i
0	0	0	0	θ_0
1	0	a_0	0	θ_1
2	-90	a_1	d_2	θ_2

$$ {}_{0}T = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 & 0 \\ \sin\theta_0 & \cos\theta_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

This is just a rotation around the Z_0 axis

$$ {}^{0}_{1}T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & a_0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

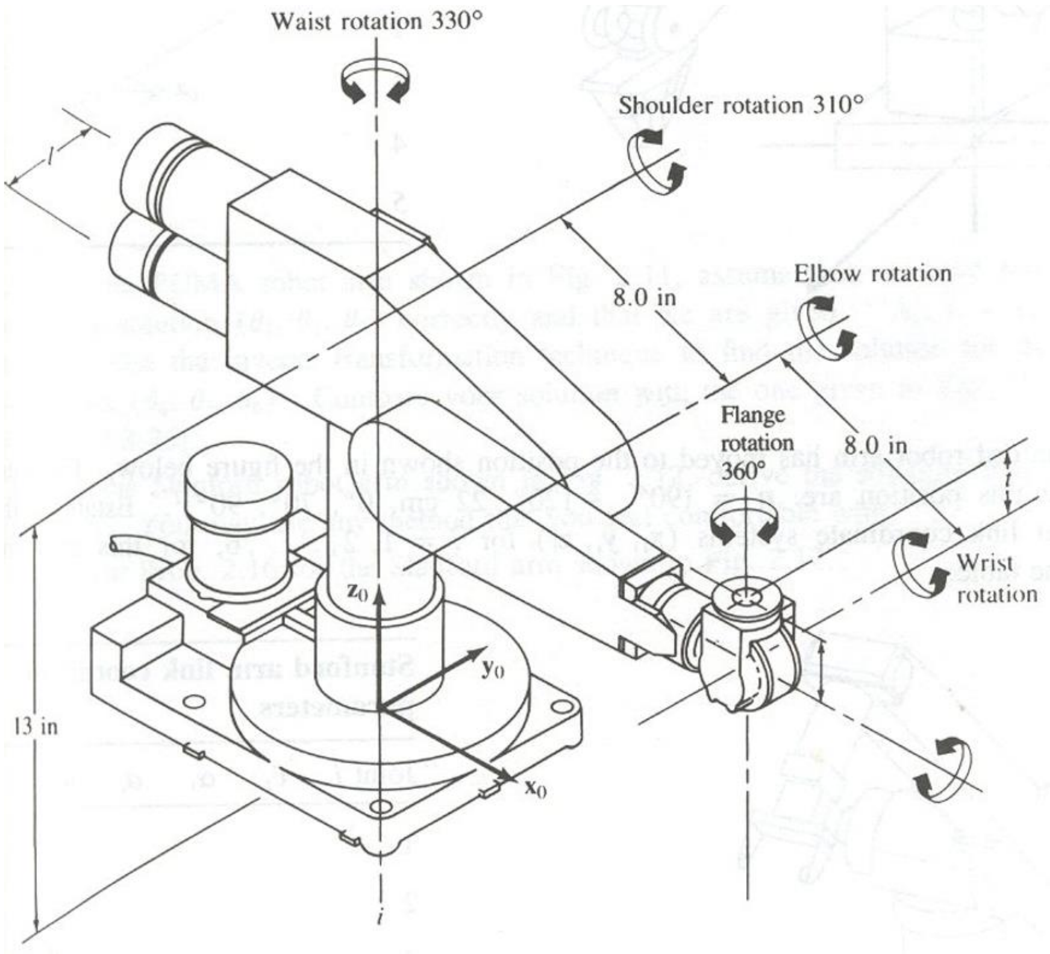
This is a translation by a_0 followed by a rotation around the Z_1 axis

$$ {}^{1}_{2}T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_1 \\ 0 & 0 & 1 & d_2 \\ -\sin\theta_2 & -\cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

This is a translation by a_1 and then d_2 followed by a rotation around the X_2 and Z_2 axis

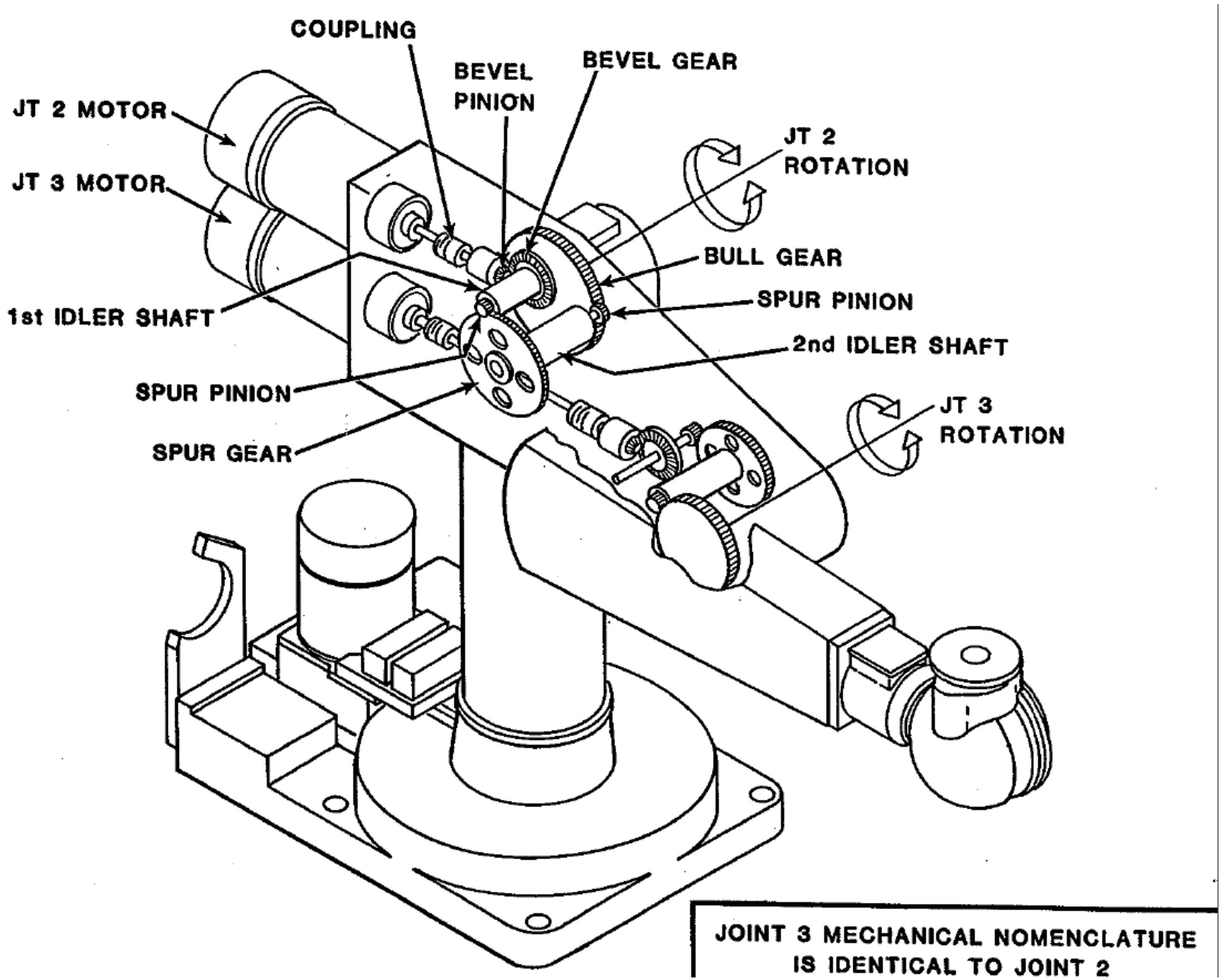
$$ T = ({}_{0}T)({}^{0}_{1}T)({}^{1}_{2}T) $$

PUMA 260



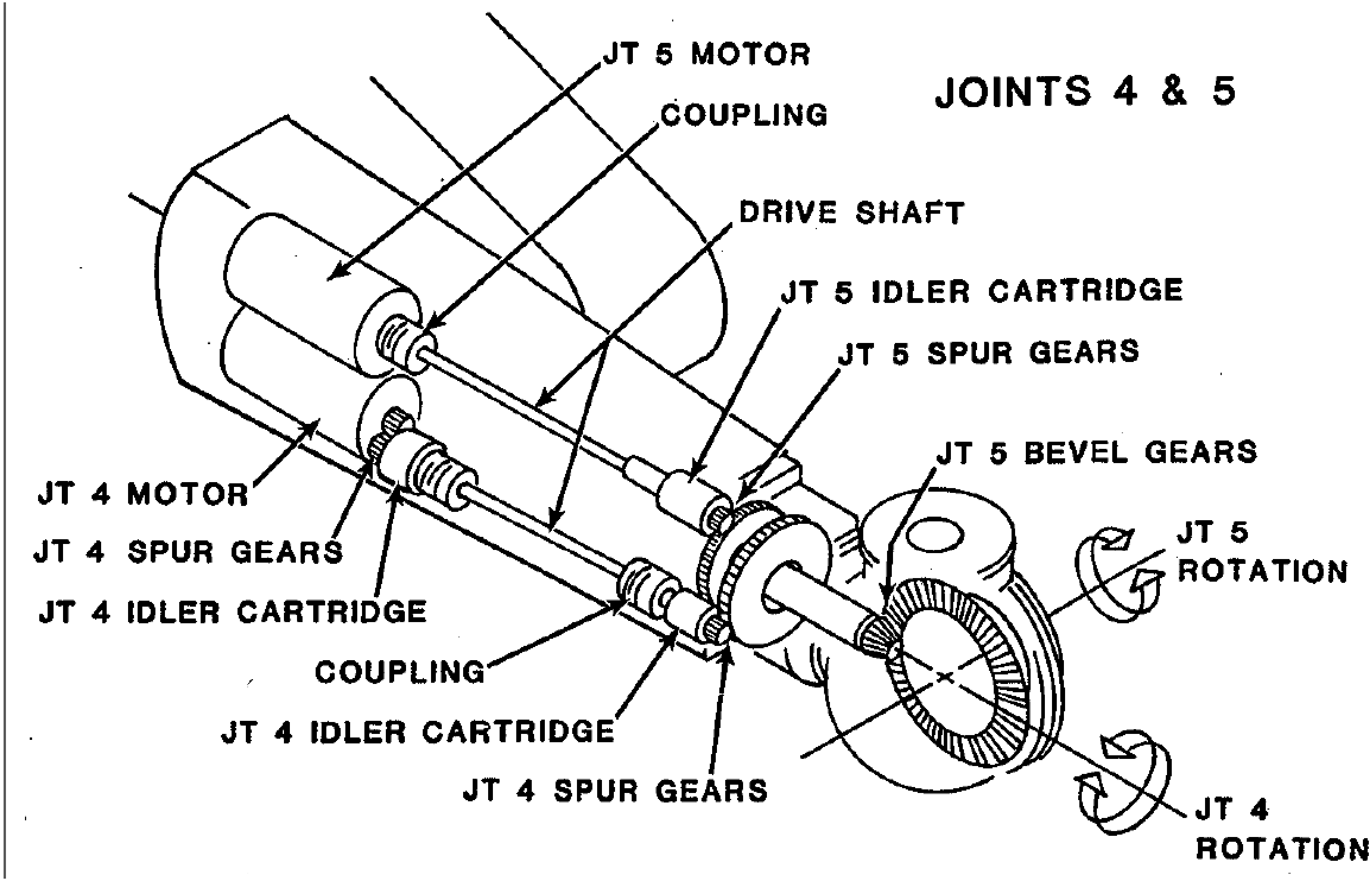
Waist rotation 330°
Shoulder rotation 310°
Elbow rotation
8.0 in
Flange rotation 360°
8.0 in
Wrist rotation
13 in
z_0
y_0
x_0

PUMA 260



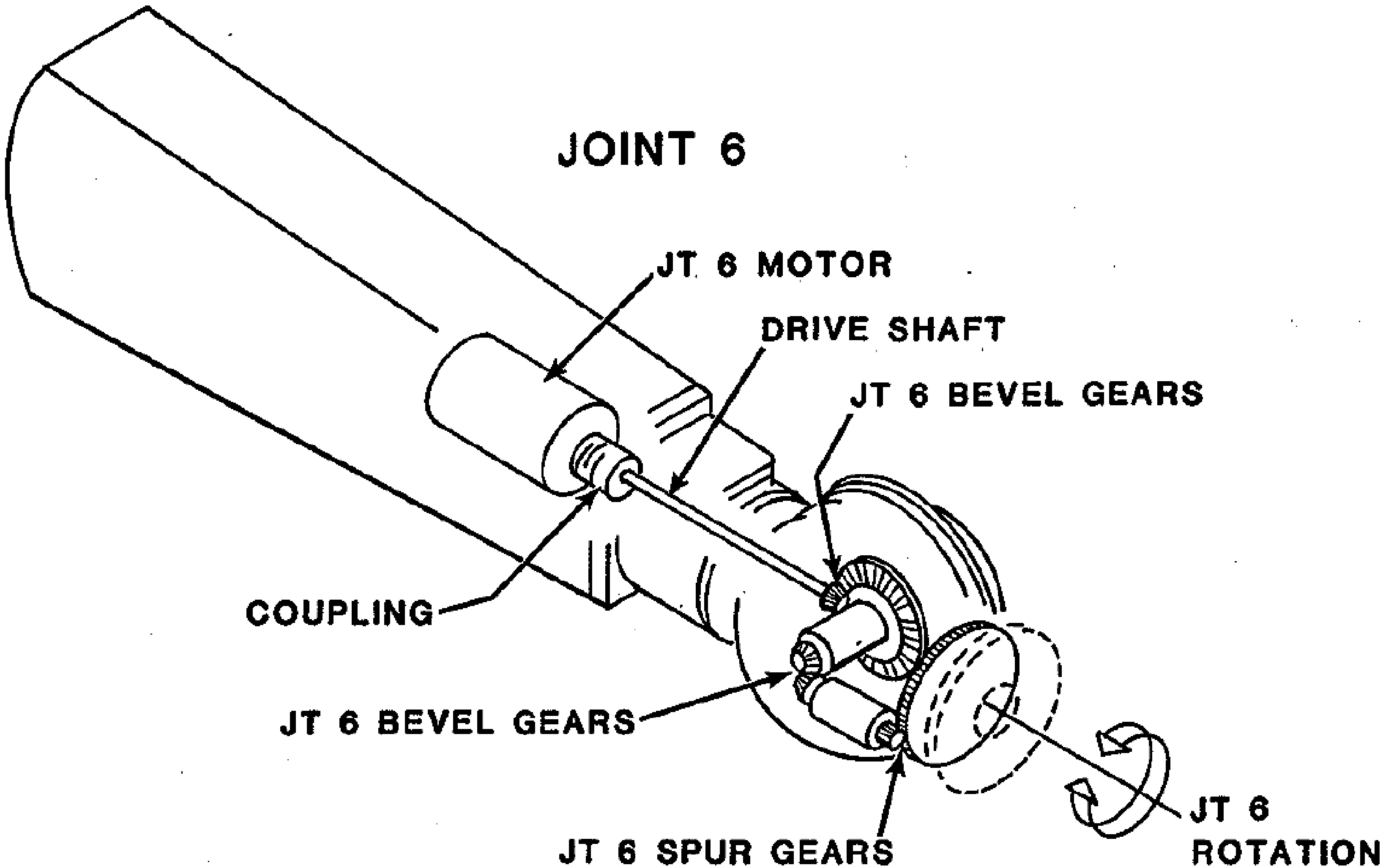
COUPLING
BEVEL PINION
BEVEL GEAR
JT 2 MOTOR
JT 3 MOTOR
JT 2 ROTATION
BULL GEAR
SPUR PINION
1st IDLER SHAFT
2nd IDLER SHAFT
SPUR PINION
SPUR GEAR
JT 3 ROTATION
JOINT 3 MECHANICAL NOMENCLATURE IS IDENTICAL TO JOINT 2

PUMA 260

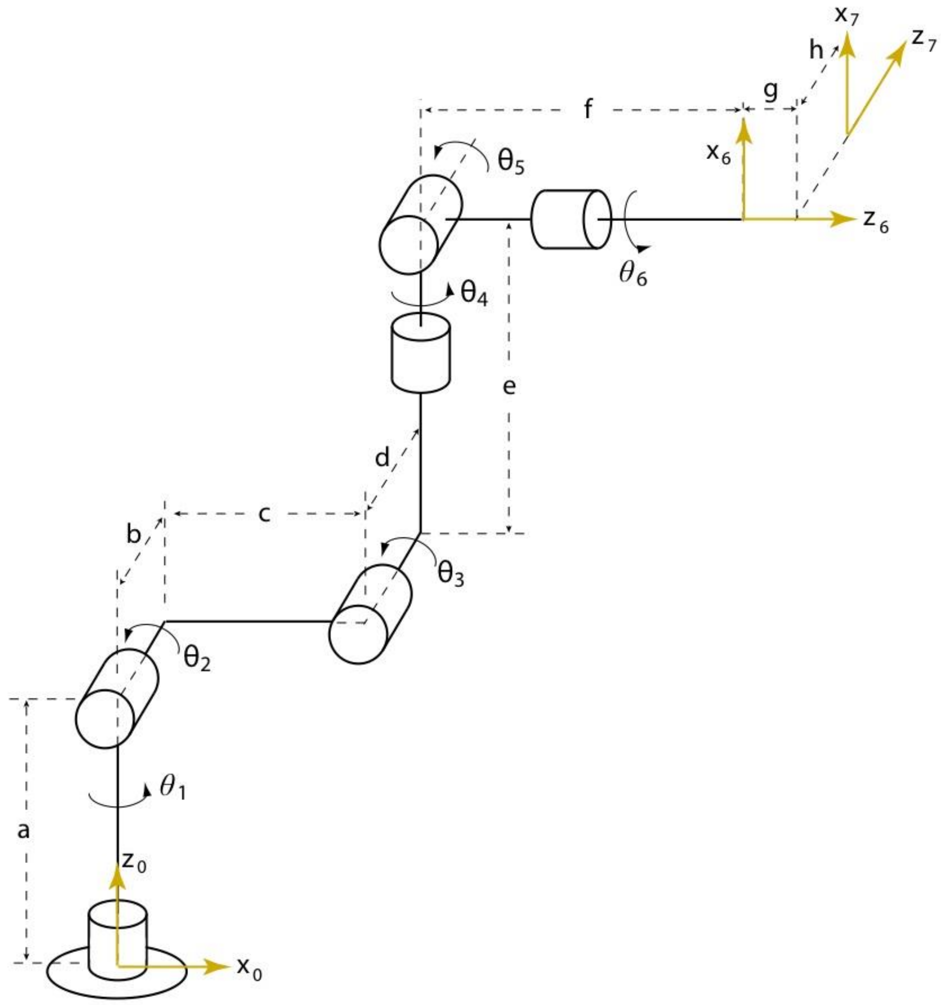


JT 5 MOTOR
JOINTS 4 & 5
COUPLING
DRIVE SHAFT
JT 5 IDLER CARTRIDGE
JT 5 SPUR GEARS
JT 5 BEVEL GEARS
JT 5
ROTATION
JT 4 MOTOR
JT 4 SPUR GEARS
JT 4 IDLER CARTRIDGE
COUPLING
JT 4 IDLER CARTRIDGE
JT 4 SPUR GEARS
JT 4
ROTATION

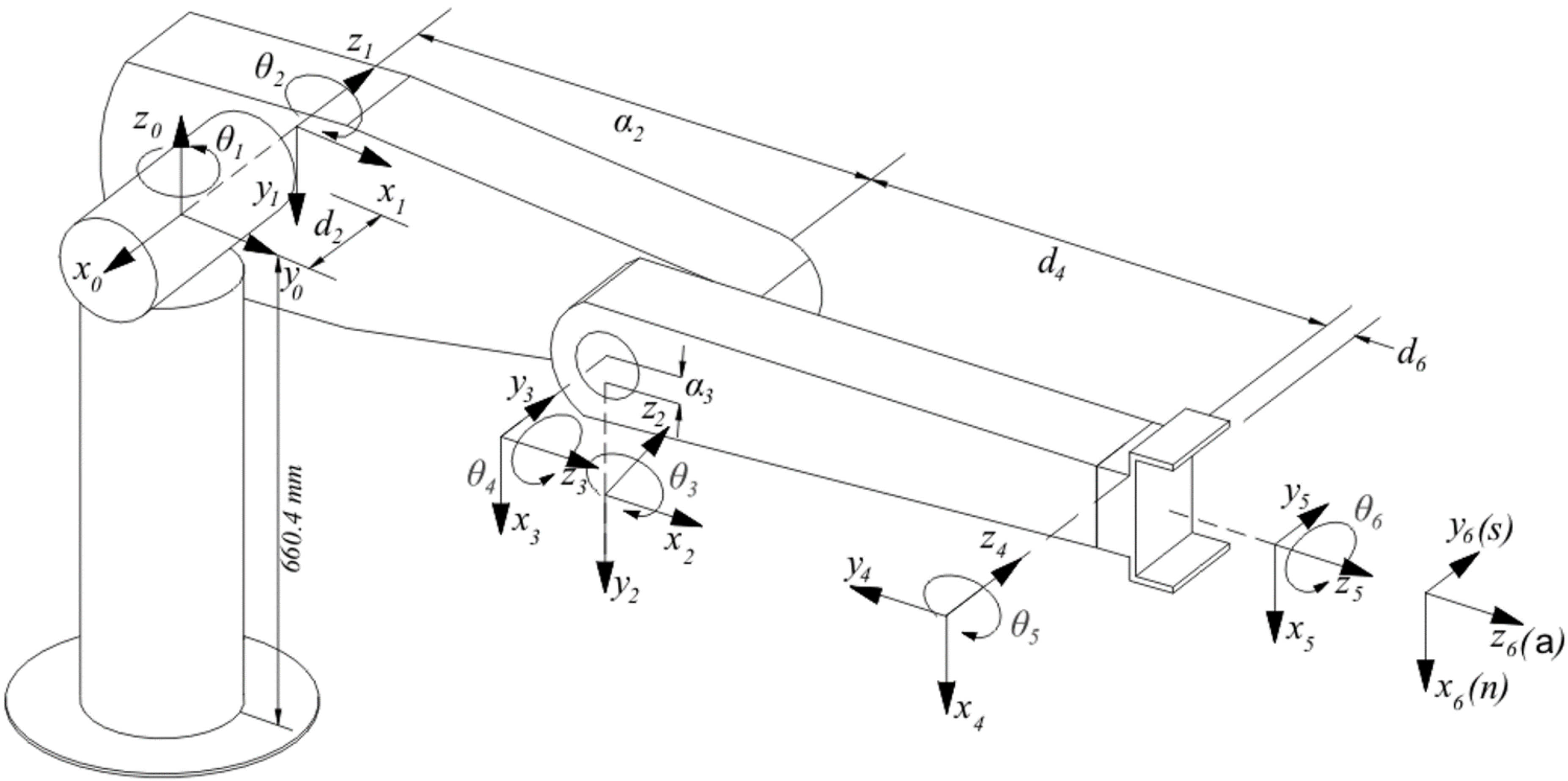
PUMA 260


JOINT 6
JT 6 MOTOR
DRIVE SHAFT
JT 6 BEVEL GEARS
COUPLING
JT 6 BEVEL GEARS
JT 6 SPUR GEARS
JT 6
ROTATION

PUMA 260



PUMA 260

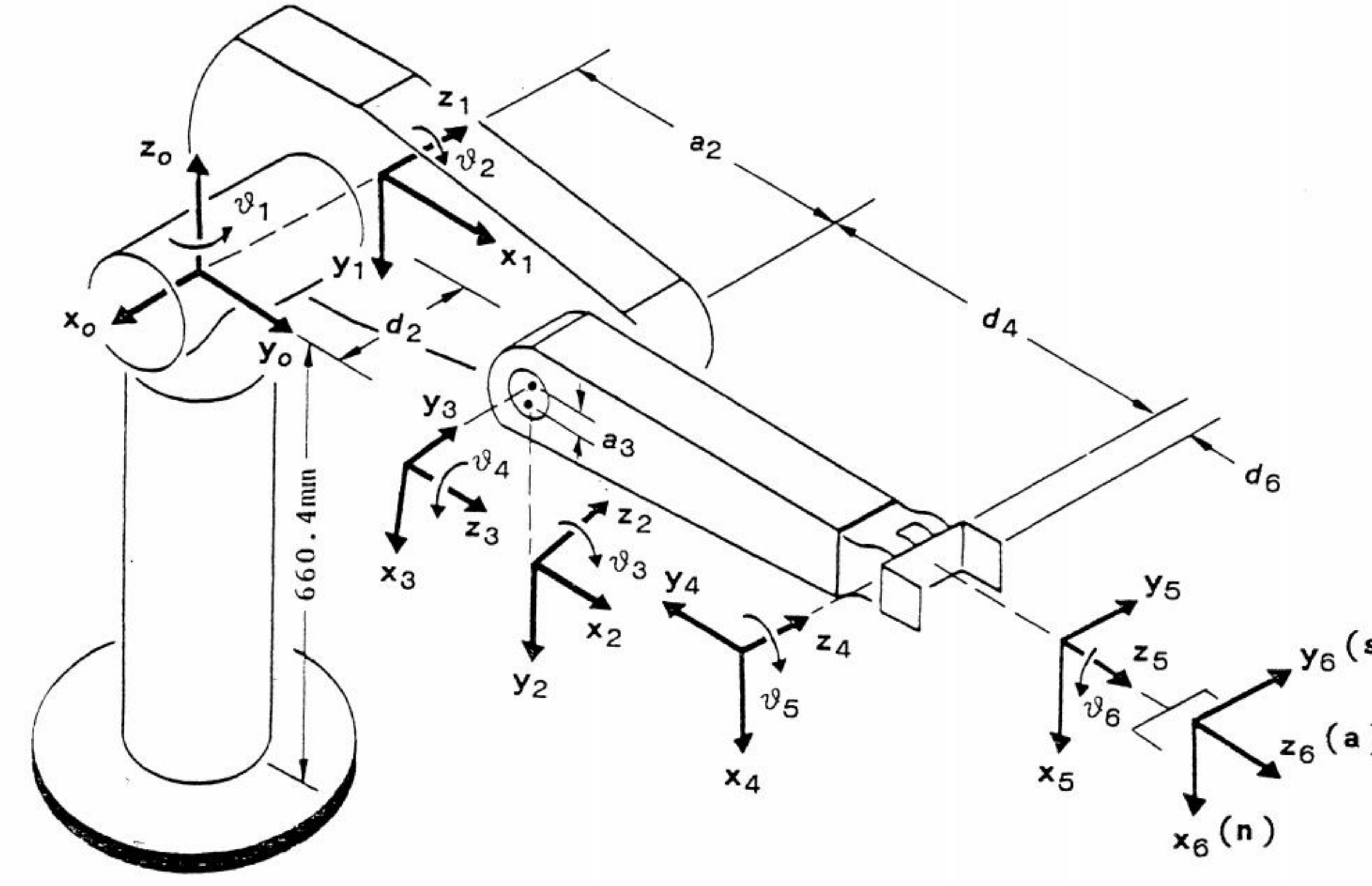


z_1
θ_2
α_2
z_0
θ_1
y_1
x_1
d_2
x_0
y_0
d_4
d_6
α_3
y_3
z_2
θ_4
z_3
θ_3
x_3
x_2
y_2
660.4 mm
y_5
θ_6
$y_6(s)$
z_4
y_4
θ_5
z_5
x_5
z_6(a)
$x_6(n)$
x_4

Denavit–Hartenberg Parameters

PUMA					
Link *i*	θ_i	α_i	a_i	d_i	**Εύρος κίνησης**
1	**90**	**-90**	**0**	**0**	**-160 to +160**
2	**0**	**0**	**431.8mm**	**149.09mm**	**-225 to 45**
3	**90**	**90**	**-20.32mm**	**0**	**-45 to 225**
4	**0**	**-90**	**0**	**433.07mm**	**-110 to 170**
5	**0**	**90**	**0**	**0**	**-100 to 100**
6	**0**	**0**	**0**	**56.25mm**	**-266 to 266**

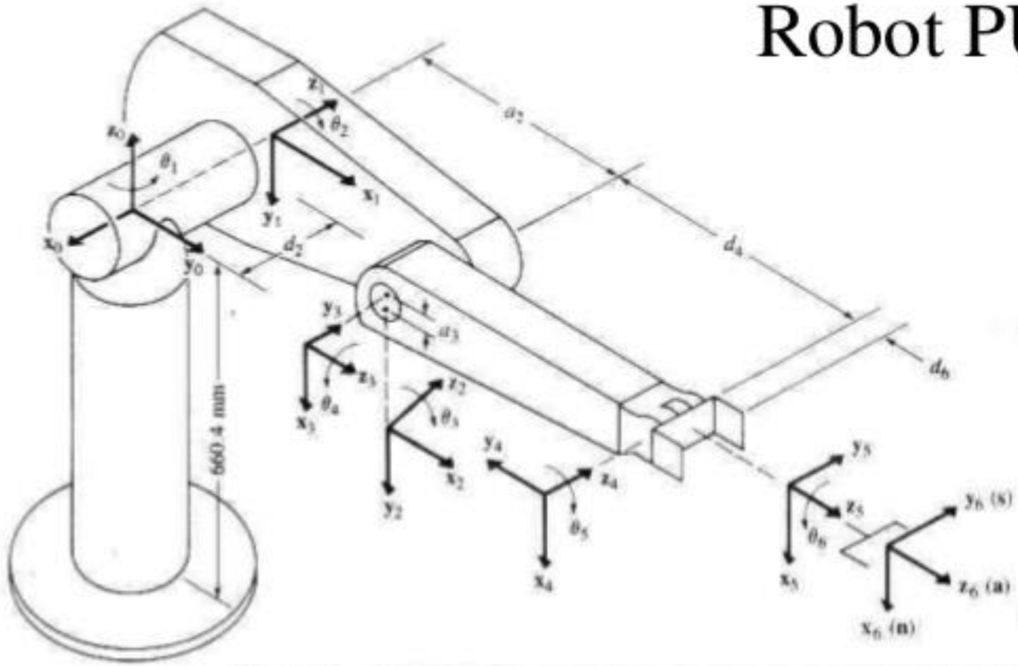
PUMA 560



z_1
a_2
ϑ_2
z_o
ϑ_1
x_1
y_1
x_o
d_2
d_4
y_o
y_3
a_3
ϑ_4
d_6
660 . 4mm
z_3
z_2
x_3
ϑ_3
y_4
y_5
x_2
z_4
y_2
z_5
y_6 (s
ϑ_5
ϑ_6
z_6 (a
x_4
x_5
x_6 (n)

PUMA 560

Robot PUMA 560



PUMA robot arm link coordinate parameters

Joint i	θ_i	α_i	a_i	d_i	Joint range
1	90	−90	0	0	−160 to +160
2	0	0	431.8 mm	149.09 mm	−225 to 45
3	90	90	−20.32 mm	0	−45 to 225
4	0	−90	0	433.07 mm	−110 to 170
5	0	90	0	0	−100 to 100
6	0	0	0	56.25 mm	−266 to 266

40

RECAP: Denavit–Hartenberg

1. Find and number consecutively the joint axes; set the directions of axes $z_0, \dots, z_{n-1}$.
2. Choose Frame 0 by locating the origin on axis z_0; axes x_0 and y_0 are chosen so as to obtain a right-handed frame. If feasible, it is worth choosing Frame 0 to coincide with the base frame.

Execute steps from **3** to **5** for $i = 1, \dots, n-1$:

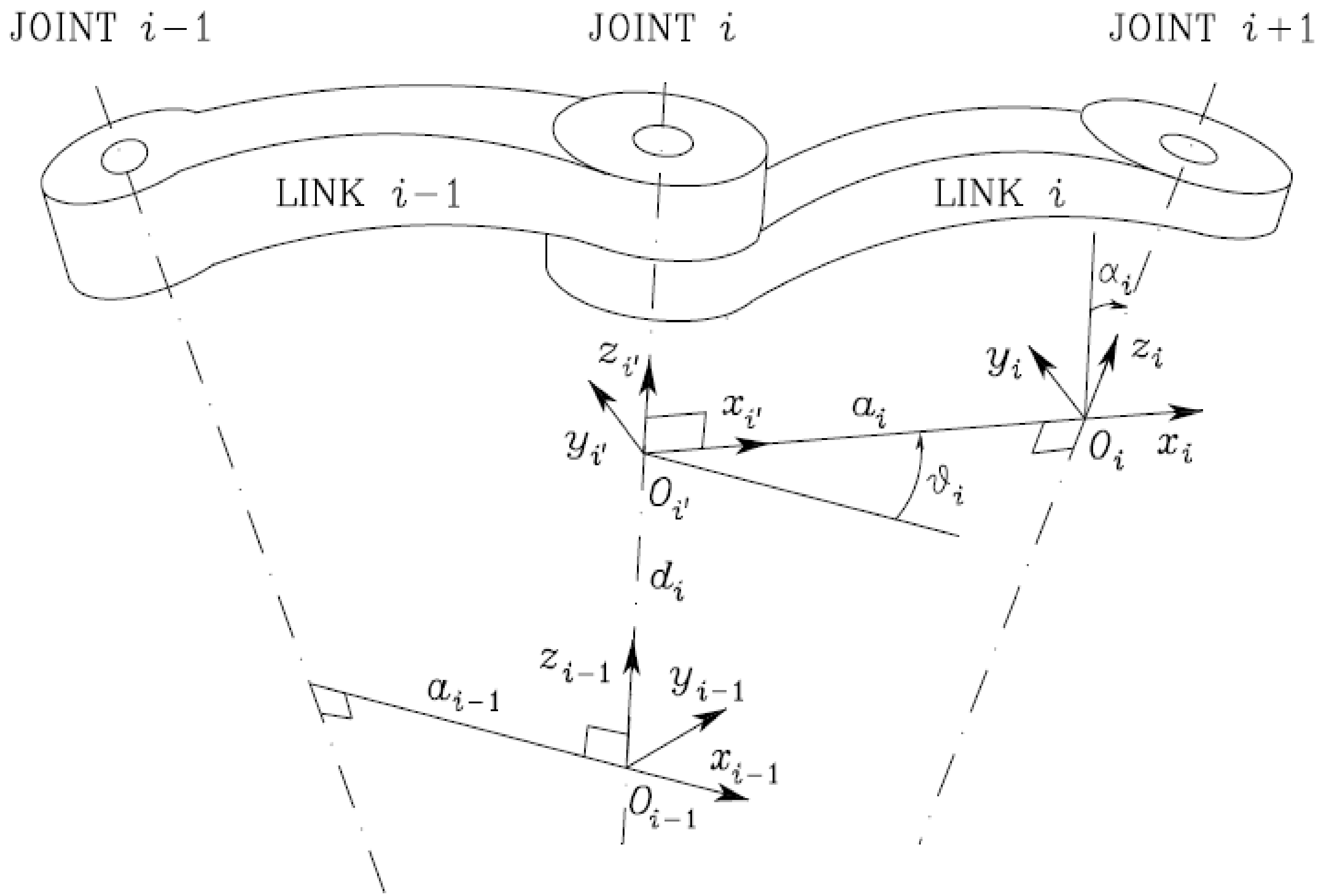
3. Locate the origin O_i at the intersection of z_i with the common normal to axes z_{i-1} and z_i. If axes z_{i-1} and z_i are parallel and Joint i is revolute, then locate O_i so that $d_i = 0$; if Joint i is prismatic, locate O_i at a reference position for the joint range, e.g., a mechanical limit.
4. Choose axis x_i along the common normal to axes z_{i-1} and z_i with direction from Joint i to Joint $i+1$.
5. Choose axis y_i so as to obtain a right-handed frame.

To complete:

6. Choose Frame n; if Joint n is revolute, then align z_n with z_{n-1}, otherwise, if Joint n is prismatic, then choose z_n arbitrarily. Axis x_n is set according to step **4**.
7. For $i = 1, \dots, n$, form the table of parameters $a_i, d_i, \alpha_i, \vartheta_i$.
8. On the basis of the parameters in **7**, compute the homogeneous transformation matrices $\boldsymbol{A}_i^{i-1}(q_i)$ for $i = 1, \dots, n$.
9. Compute the homogeneous transformation $\boldsymbol{T}_n^0(\boldsymbol{q}) = \boldsymbol{A}_1^0 \dots \boldsymbol{A}_n^{n-1}$ that yields the position and orientation of Frame n with respect to Frame 0.
10. Given $\boldsymbol{T}_0^b$ and $\boldsymbol{T}_e^n$, compute the direct kinematics function as $\boldsymbol{T}_e^b(\boldsymbol{q}) = \boldsymbol{T}_0^b \boldsymbol{T}_n^0 \boldsymbol{T}_e^n$ that yields the position and orientation of the end-effector frame with respect to the base frame.

a_i = the distance from z_{i-1} to z_i measured along x_i
α_i = the angle from z_{i-1} to z_i measured about x_i
d_i = the distance from x_{i-1} to x_i measured along z_i
θ_i = the angle from x_{i-1} to x_i measured about z_{i-1}

RECAP: Denavit–Hartenberg



RECAP: Denavit–Hartenberg

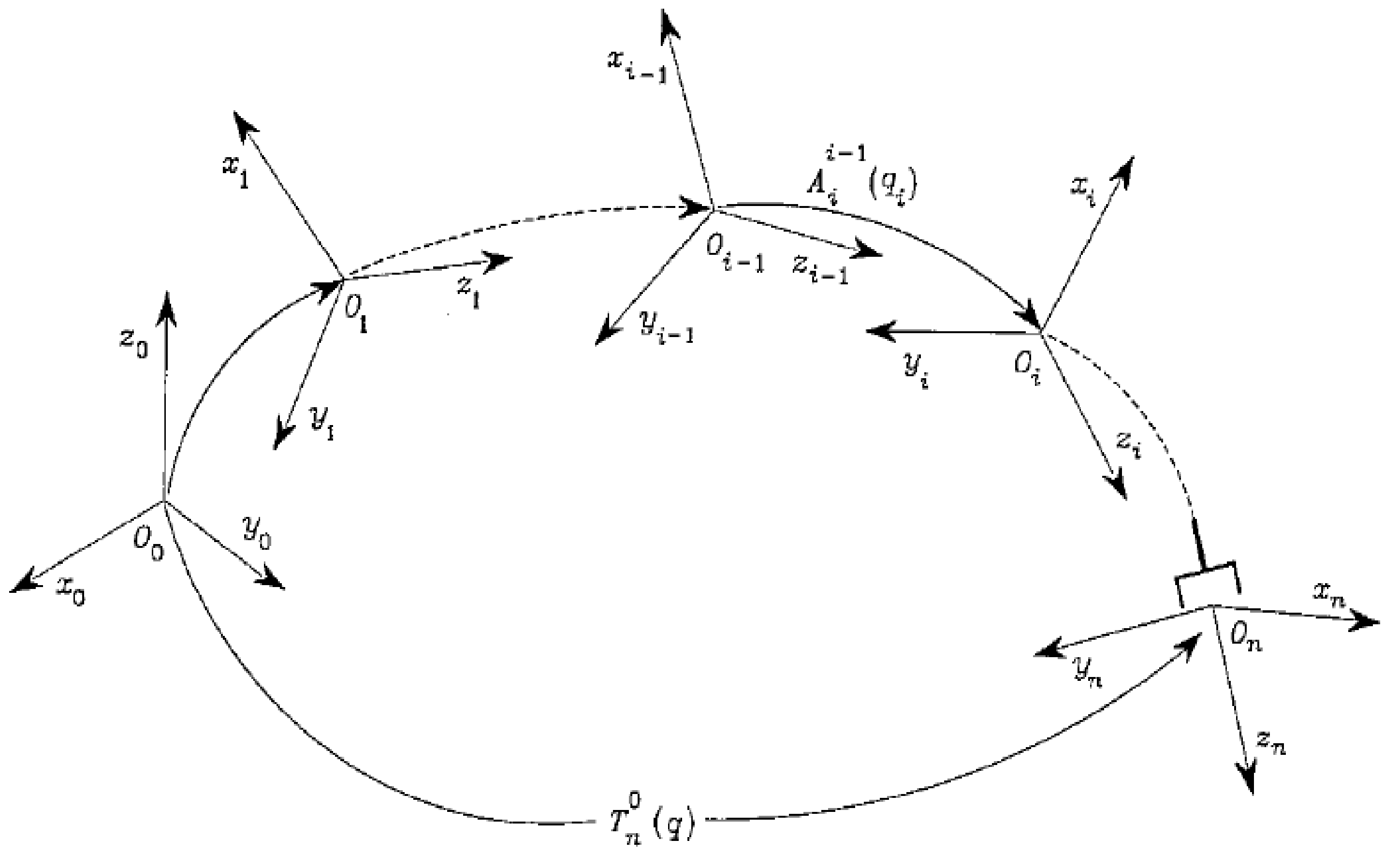
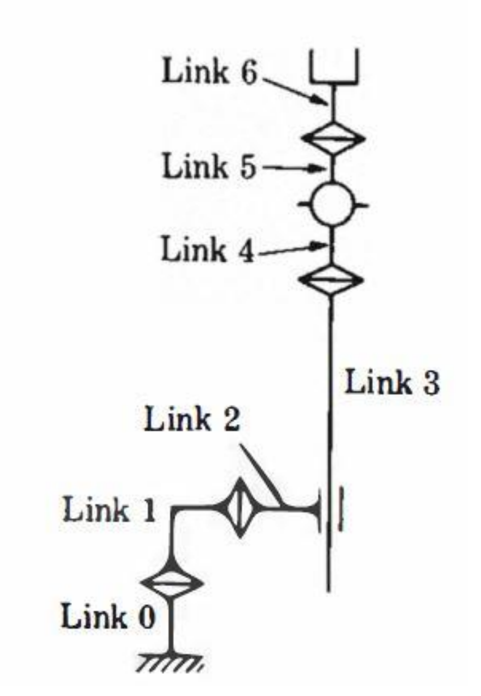


FIGURE 2.16
Coordinate transformation of the end-effector frame with respect to the base frame.

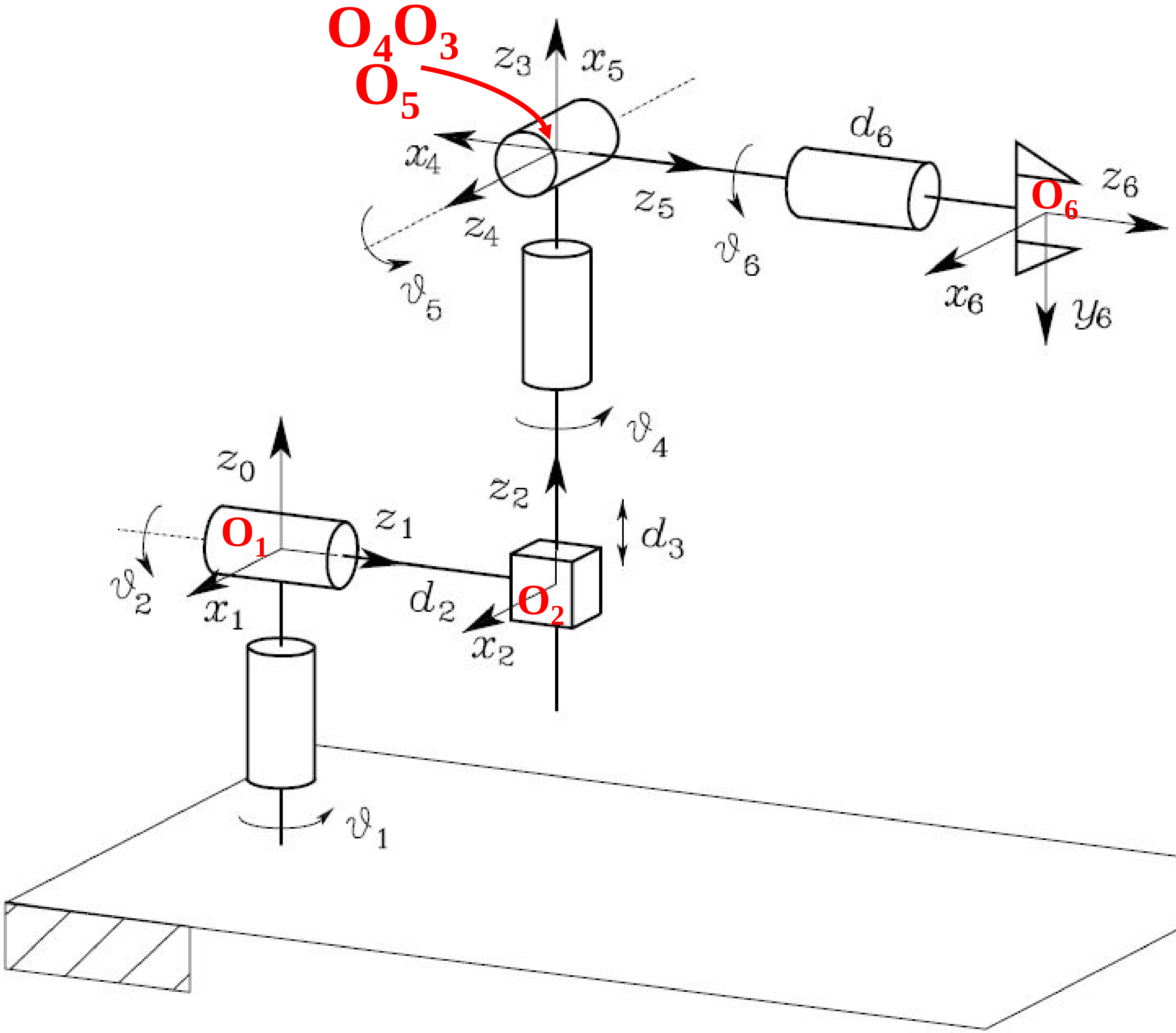
RECAP:
Worked out example
THE STANFORD (Scheinman) Arm



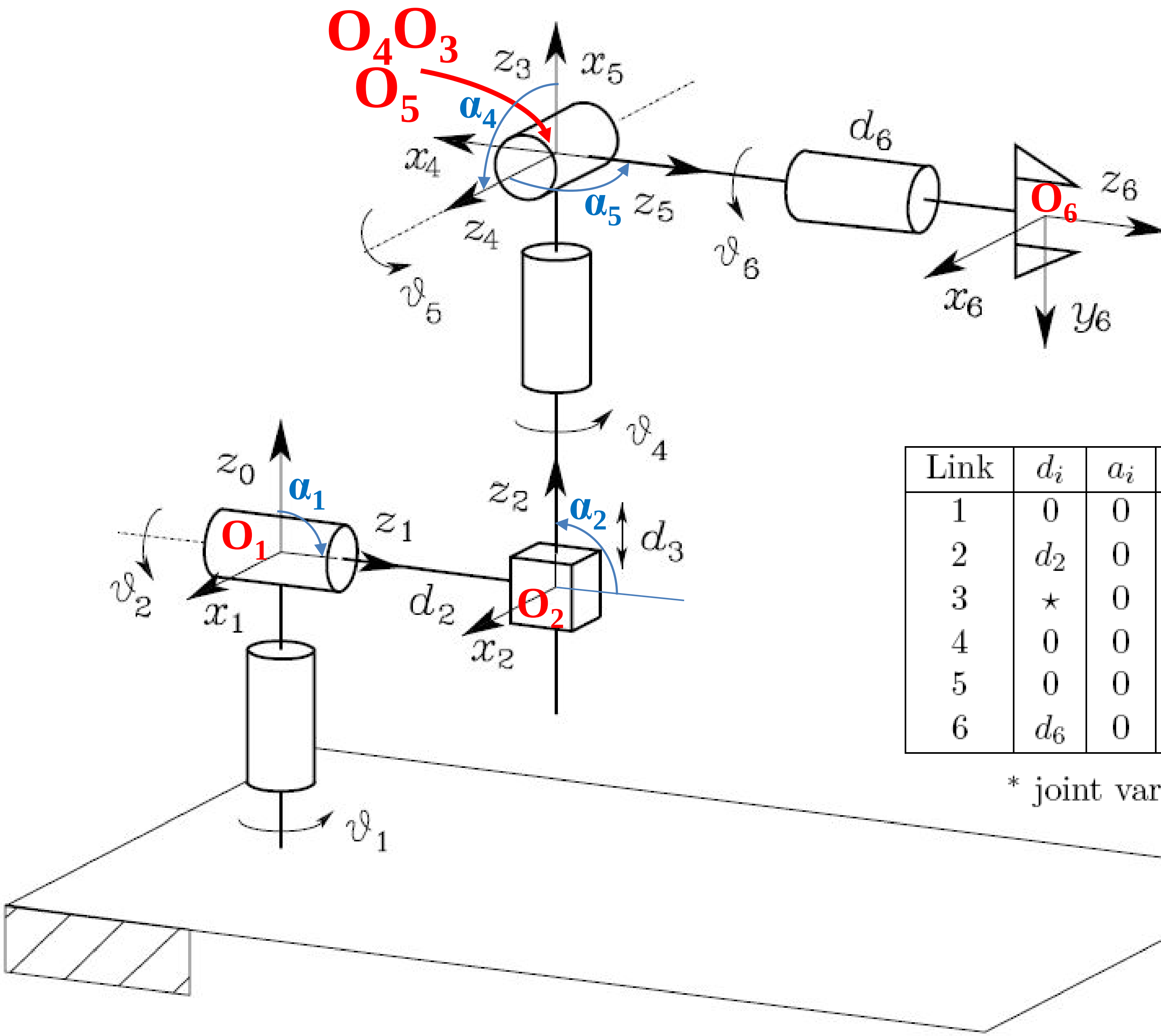
Stanford Manipulator

The original ~1968

STANFORD MANIPULATOR



STANFORD MANIPULATOR



Link	d_i	a_i	α_i	θ_i
1	0	0	-90	$\star$
2	d_2	0	$+90$	$\star$
3	$\star$	0	0	0
4	0	0	-90	$\star$
5	0	0	$+90$	$\star$
6	d_6	0	0	$\star$

* joint variable

STANFORD MANIPULATOR

The DH parameters are:

Link	d_i	a_i	α_i	θ_i
1	0	0	-90	$\star$
2	d_2	0	$+90$	$\star$
3	$\star$	0	0	0
4	0	0	-90	$\star$
5	0	0	$+90$	$\star$
6	d_6	0	0	$\star$

* joint variable

$$A_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

STANFORD MANIPULATOR

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_0^1 = A_1$$

$$T_0^2 = A_1A_2 = \begin{bmatrix} c_1c_2 & -s_1 & c_1s_2 & -d_2s_1 \\ s_1c_2 & c_1 & s_1s_2 & d_2c_1 \\ -s_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad z_1 = \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix} \quad z_2 = \begin{bmatrix} c_1s_2 \\ s_1s_2 \\ c_2 \end{bmatrix} \quad z_3 = \begin{bmatrix} c_1s_2 \\ s_1s_2 \\ c_2 \end{bmatrix}$$

$$T_0^3 = A_1A_2A_3 = \begin{bmatrix} c_1c_2 & -s_1 & c_1s_2 & d_3c_1s_2 - d_2s_1 \\ s_1c_2 & c_1 & s_1s_2 & d_3s_1s_2 + d_2c_1 \\ -s_2 & 0 & c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$O_0 = O_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad O_2 = \begin{bmatrix} -d_2s_1 \\ d_2c_1 \\ 0 \end{bmatrix}$$

$$O_3 = \begin{bmatrix} d_3c_1s_2 - d_2s_1 \\ d_3s_1s_2 + d_2c_1 \\ d_3c_2 \end{bmatrix}$$

STANFORD MANIPULATOR

$$T_0^4 = A_1 A_2 A_3 A_4$$

$$T_0^5 = A_1 A_2 A_3 A_4 A_5$$

$$T_0^6 = A_1 A_2 A_3 A_4 A_5 A_6$$

T4 =

[c1c2c4−s1s4,   −c1s2, −c1c2s4−s1*c4, c1s2d3−sin1d2]
[s1c2c4+c1s4,   −s1s2, −s1c2s4+c1c4, s1s2d3+c1*d2]
[−s2c4, −c2, s2s4, c2*d3]
[0,   0, 0, 1]

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

$$O_4 = \begin{bmatrix} d_3 c_1 s_2 - d_2 s_1 \\ d_3 s_1 s_2 + d_2 c_1 \\ d_3 c_2 \end{bmatrix}$$

STANFORD MANIPULATOR

T5 =
[(c1c2c4−s1s4)c5−c1s2s5, c1c2s4+s1c4, (c1c2c4−s1s4)s5+c1s2c5,c1s2d3−s1d2]
[(s1c2c4+c1s4)c5−s1s2s5, s1c2s4−c1c4, (s1c2c4+c1s4)s5+s1s2c5,s1s2d3+c1d2]
[−s2c4c5−c2s5, −s2s4, −s2c4s5+c2c5, c2d3]
[0, 0, 0, 1]

$$z_5 = \begin{bmatrix} c_1c_2c_4s_5 - s_1s_4s_5 + c_1s_2c_5 \\ s_1c_2c_4s_5 + c_1s_4s_5 + s_1s_2c_5 \\ -s_2c_4s_5 + c_2c_5 \end{bmatrix}$$

STANFORD MANIPULATOR

$T5 =$

$$[\ (c1c2c4 - s1s4)c5 - c1s2s5,\ c1c2s4 + s1c4,\ (c1c2c4 - s1s4)s5 + c1s2c5, c1s2d3 - s1d2]$$

$$[\ (s1c2c4 + c1s4)c5 - s1s2s5, s1c2s4 - c1c4,\ \left(s1c2c4 + c1s4\right)s5 + s1s2c5, s1s2d3 + c1d2]$$

$$[-s2c4c5 - c2s5, -s2s4, -s2c4s5 + c2c5, c2d3]$$

$$\left[\ 0,\ 0,\ 0,\ 1\right]$$

$$z_5 = \begin{bmatrix} c_1c_2c_4s_5 - s_1s_4s_5 + c_1s_2c_5 \\ s_1c_2c_4s_5 + c_1s_4s_5 + s_1s_2c_5 \\ -s_2c_4s_5 + c_2c_5 \end{bmatrix}$$

$$O_5 = \begin{bmatrix} d_3c_1s_2 - d_2s_1 \\ d_3s_1s_2 + d_2c_1 \\ d_3c_2 \end{bmatrix}$$

STANFORD MANIPULATOR
ALTERNATIVE EDITION/REPRESENTATION