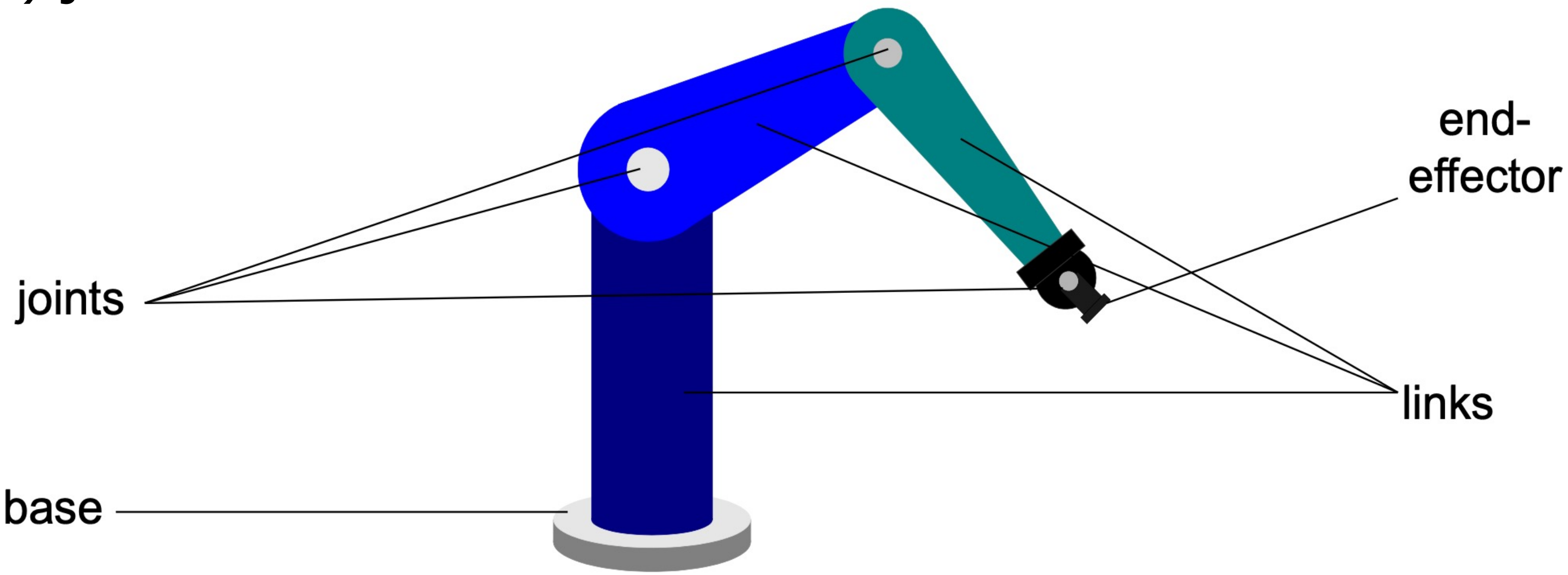
Lecture 4

Introduction to Forward Kinematics (FK)

Alexandros Lioulemes, PhD

Robotic Manipulator

- Kinematic chain of rigid bodies (links)
- Revolute (R) or prismatic (P) joints
- Base
- End-effector

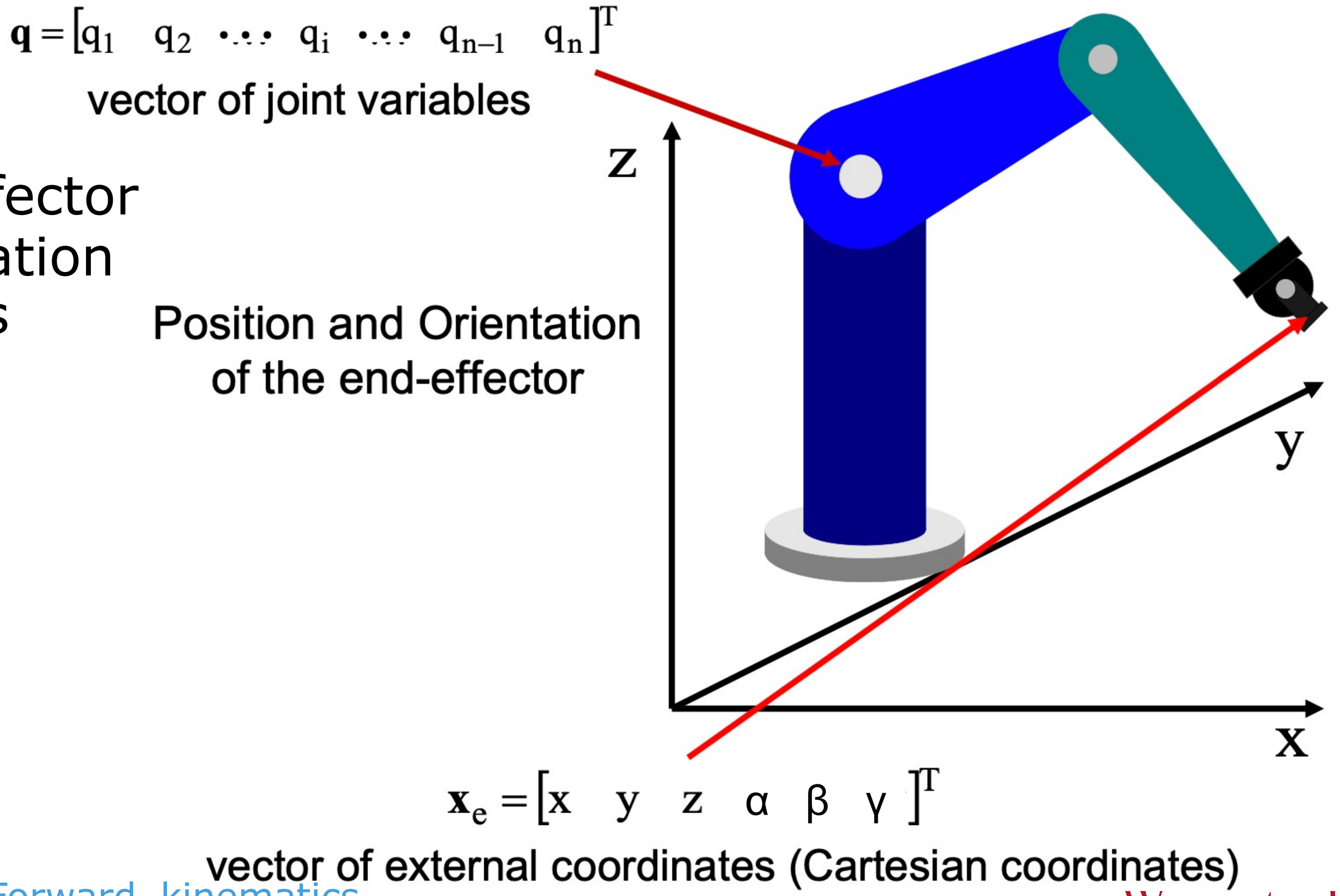


https://en.wikipedia.org/wiki/Manipulator_(device)

Robotic Manipulator - Forward Kinematics

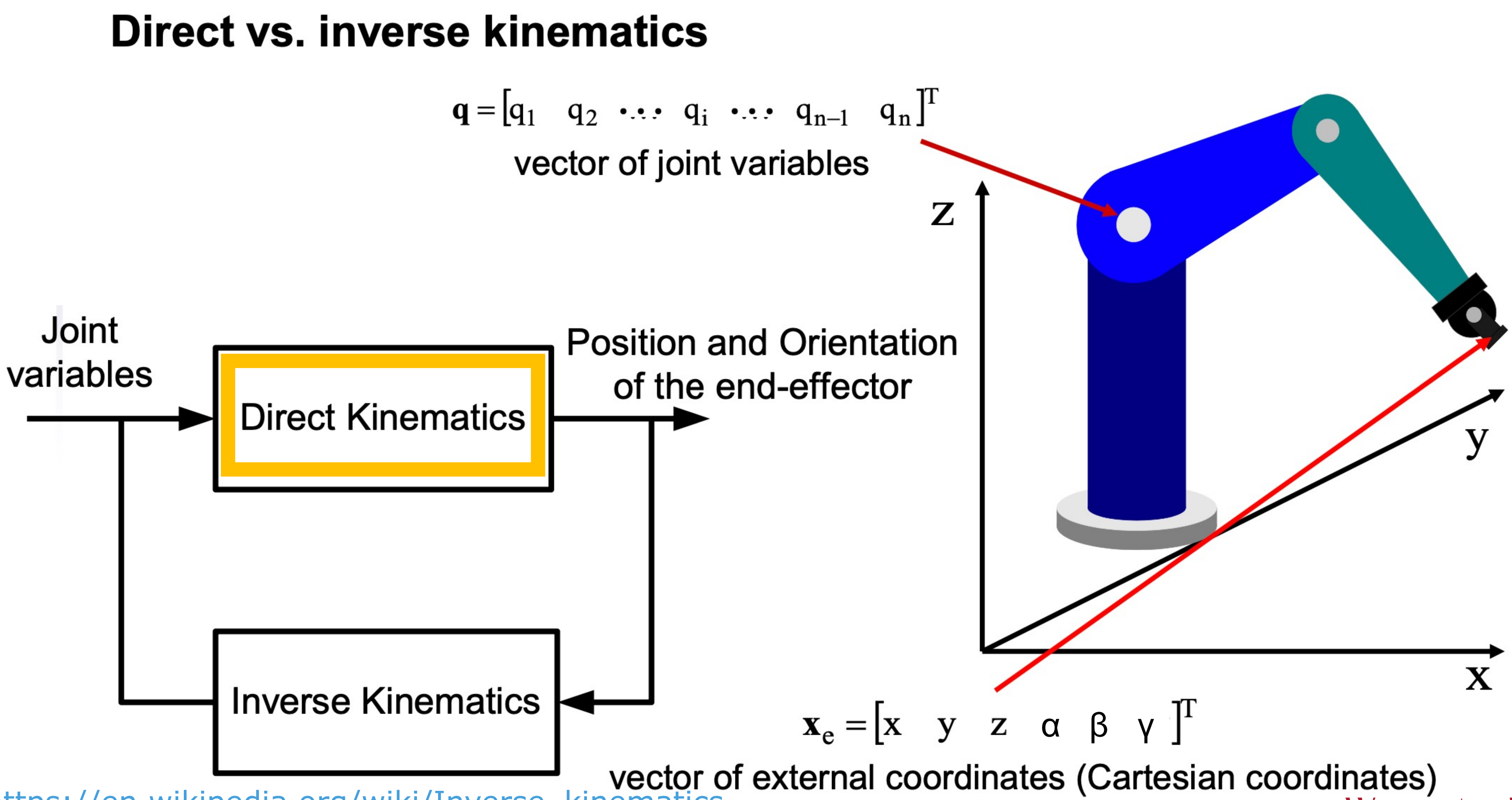
- Derive Forward (Direct) Kinematics using linear algebra

- Express the end-effector position and orientation as function of joints variables



https://en.wikipedia.org/wiki/Forward_kinematics

Forward (Direct) vs. Inverse Kinematics



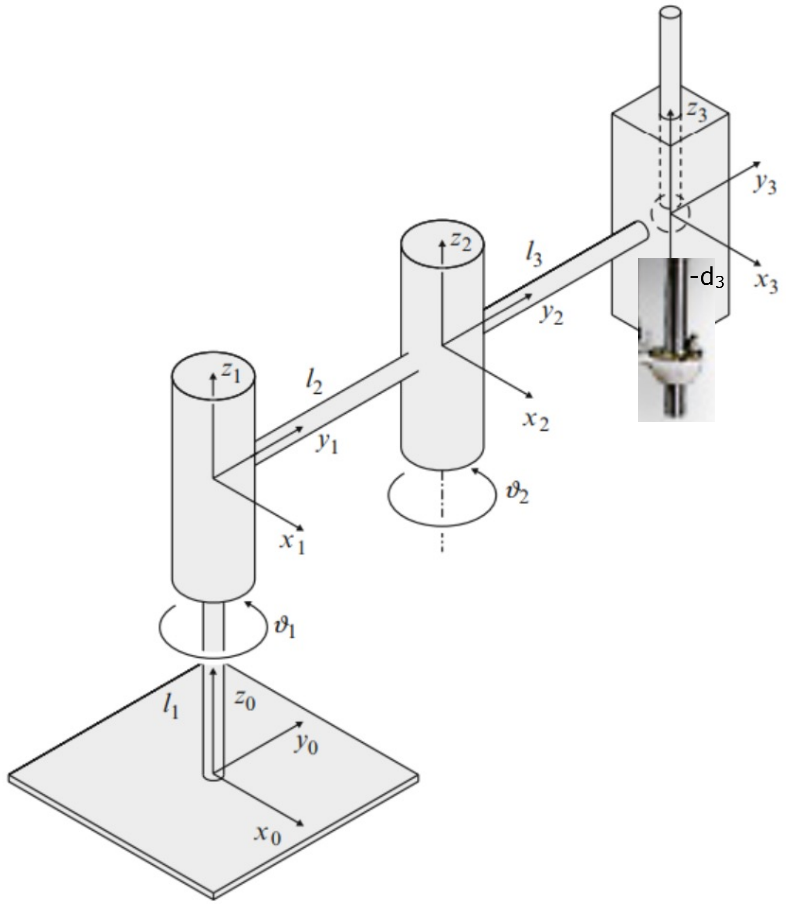
https://en.wikipedia.org/wiki/Inverse_kinematics

SCARA*

- Selective Compliance Assembly Robot Arm (SCARA)
- Two rotational joint in the first two links and one prismatic at the last link (RRP)
- Let’s derive the Forward Kinematics
  – **How?**

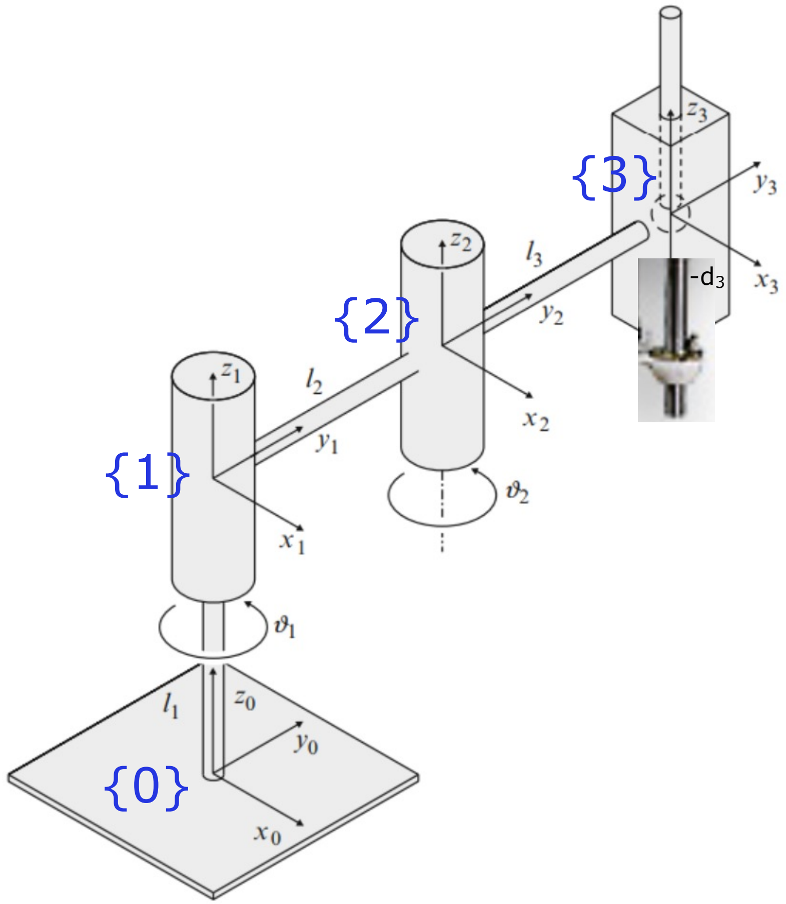
https://en.wikipedia.org/wiki/SCARA

SCARA* - Kinematic diagram



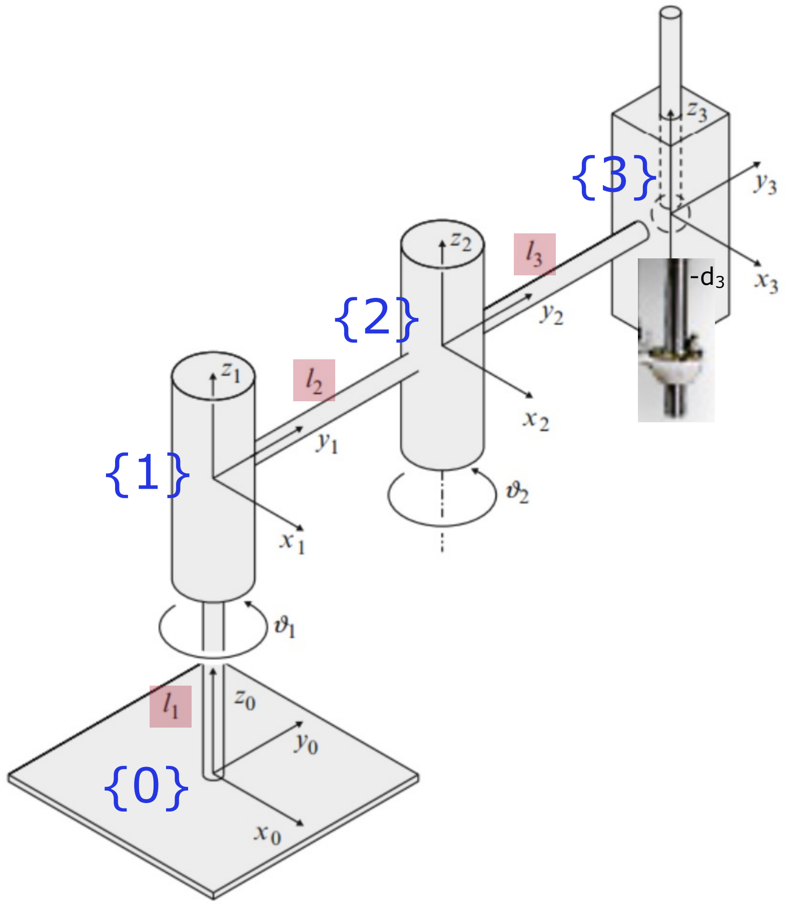
https://en.wikipedia.org/wiki/Kinematic_diagram

SCARA* - Kinematic diagram - Frames



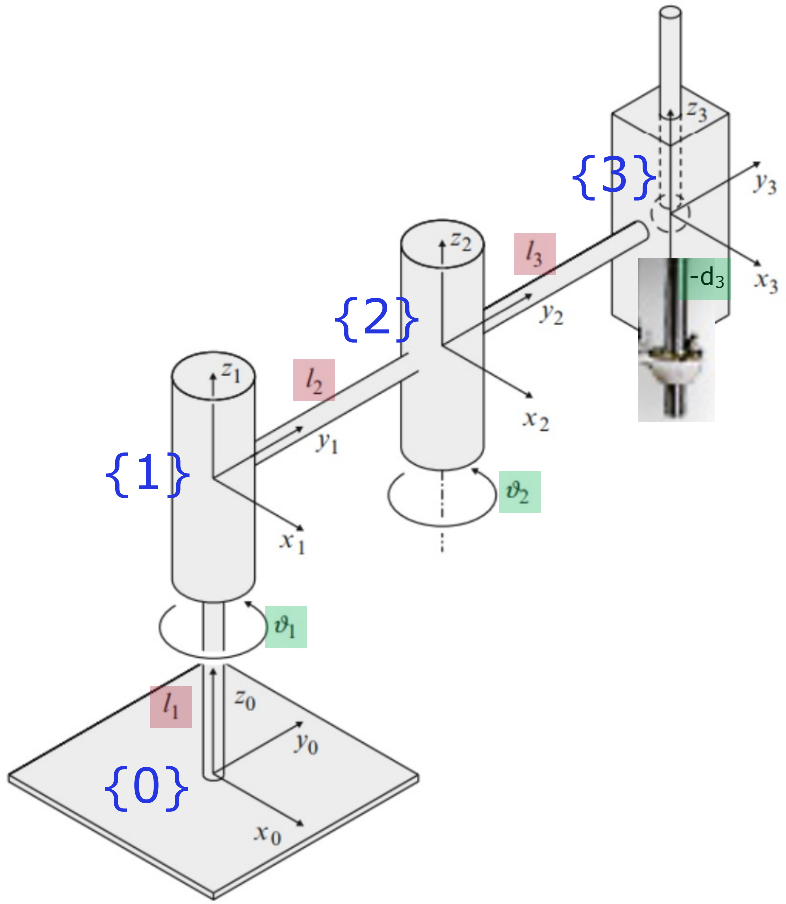
- Homogeneous Transformations {0} → {3}

SCARA* - Forward Kinematics - Link lengths



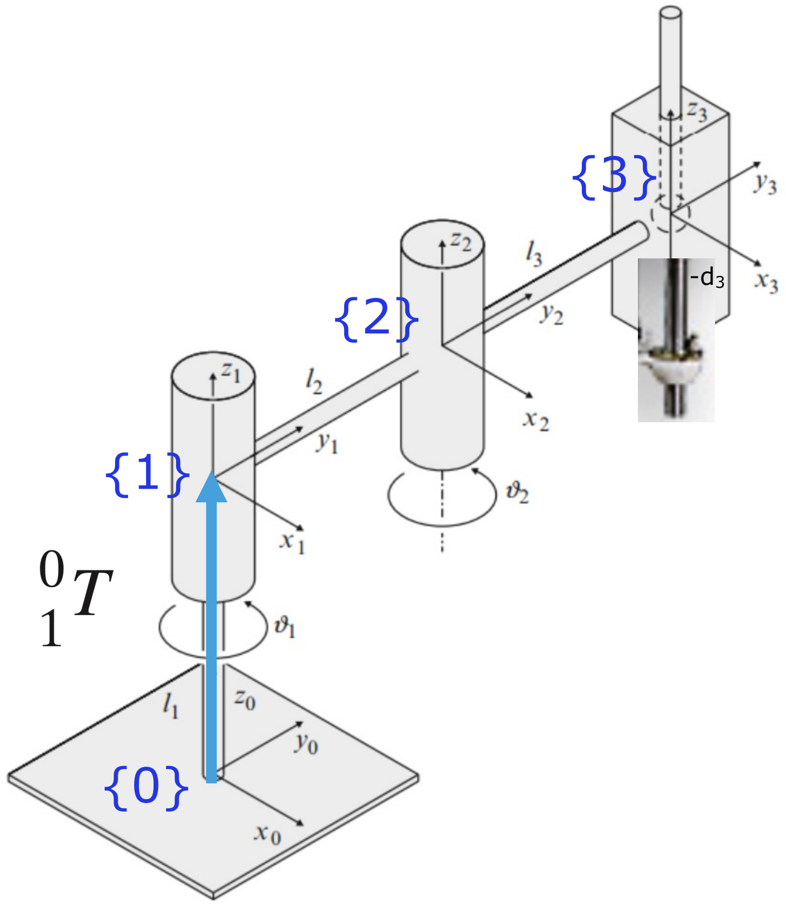
- link 1 → length 1 → *l1*
- link 2 → length 2 → *l2*
- link 3 → length 3 → *l3*

SCARA* - Forward Kinematics - Joint types



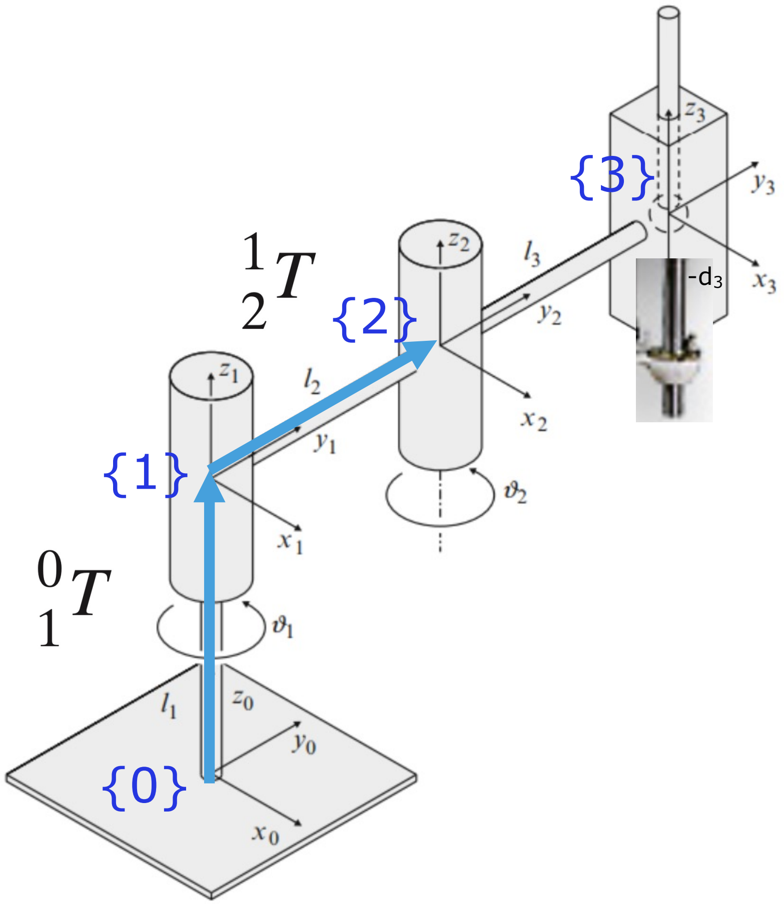
- Revolute joint 1 → θ_1
- Revolute joint 2 → θ_2
- Prismatic joint 3 → $-d_3$

SCARA* - Forward Kinematics - HT



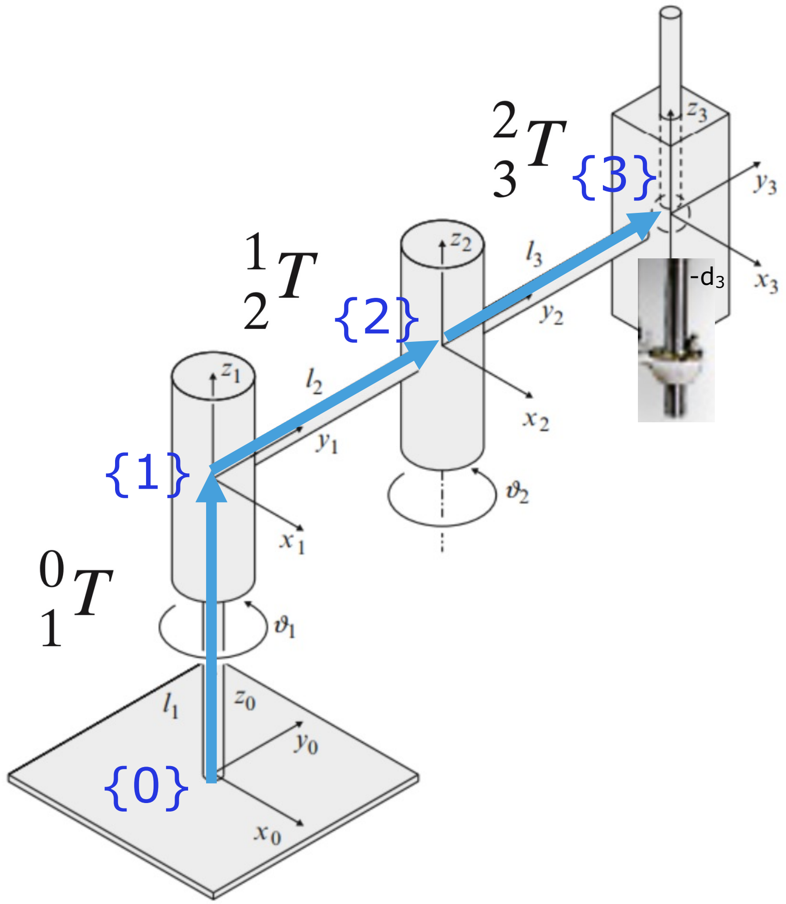
- Homogeneous Transformations:
- ^0_1T : {0} → {1}

SCARA* - Forward Kinematics - HT



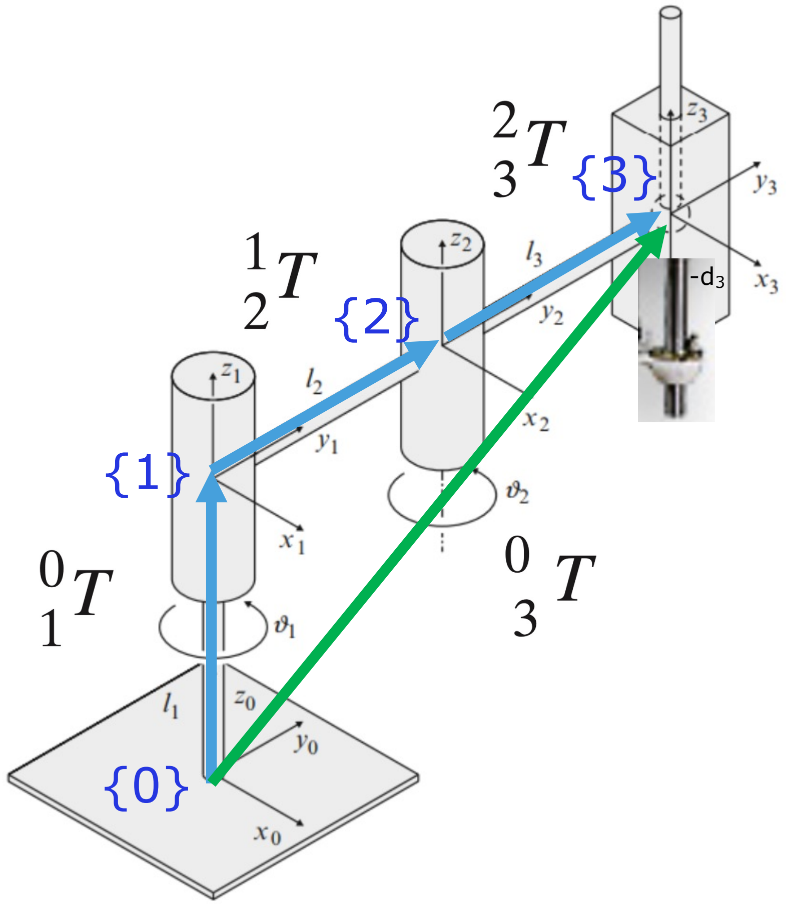
- Homogeneous Transformations:
- ^0_1T : {0} → {1}
- ^1_2T : {1} → {2}

SCARA* - Forward Kinematics - HT



- Homogeneous Transformations:
- ${}^{0}_{1}T$: {0} → {1}
- ${}^{1}_{2}T$: {1} → {2}
- ${}^{2}_{3}T$: {2} → {3}

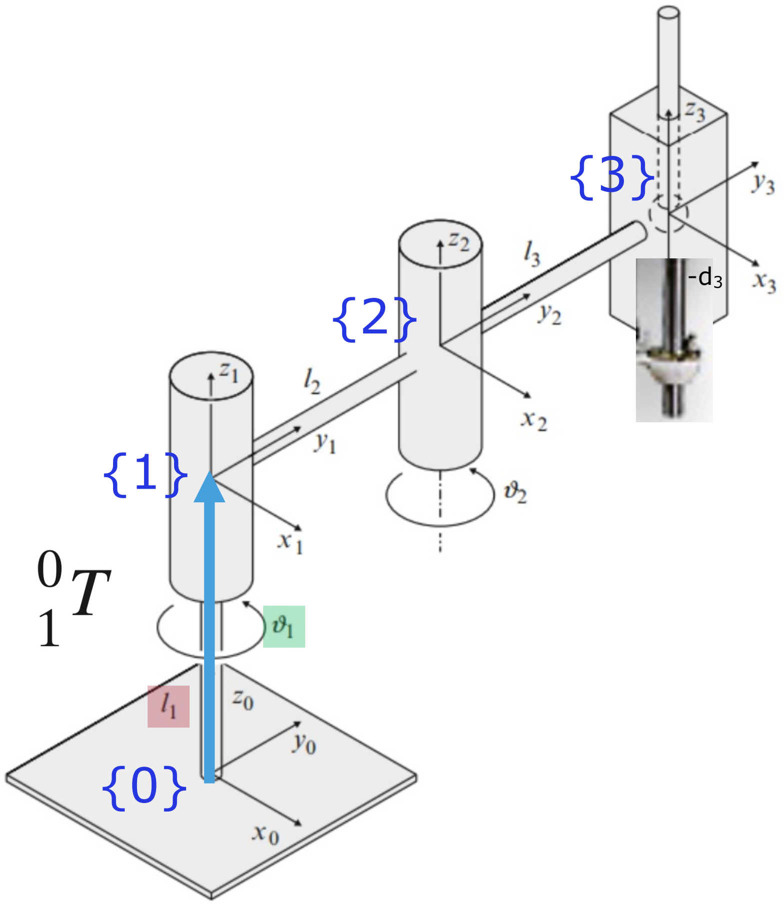
SCARA* - Forward Kinematics - HT



- Homogeneous Transformations:
- $^{0}_{1}T$: {0} → {1}
- $^{1}_{2}T$: {1} → {2}
- $^{2}_{3}T$: {2} → {3}

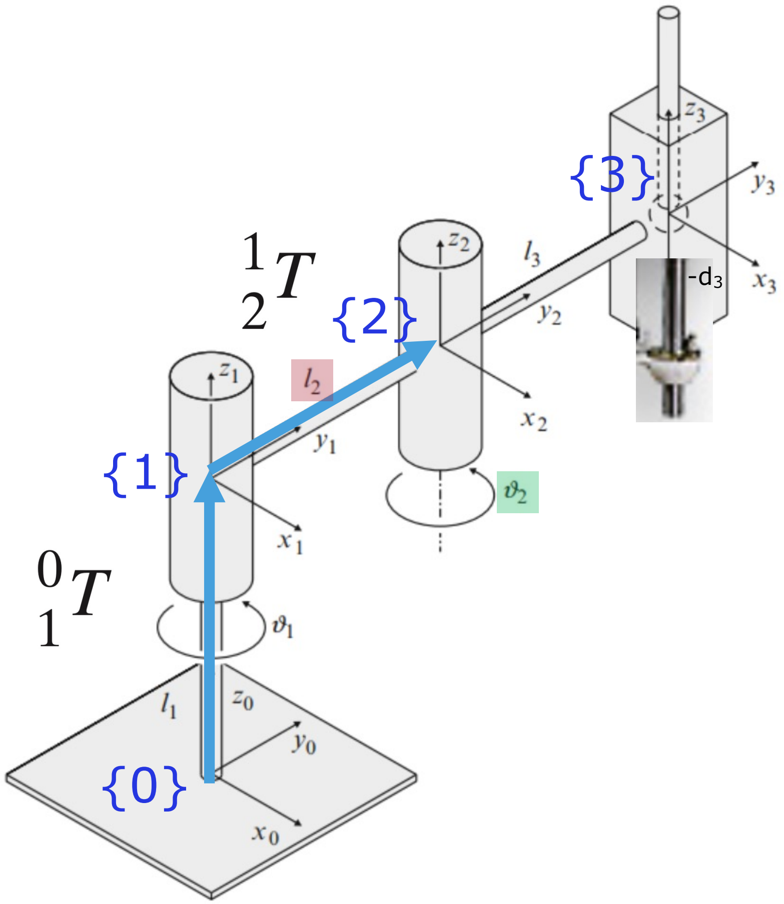
- {0} → {3}: $^{0}_{3}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T$

SCARA* - Forward Kinematics - HT Matrices



$$
{}^0_1T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

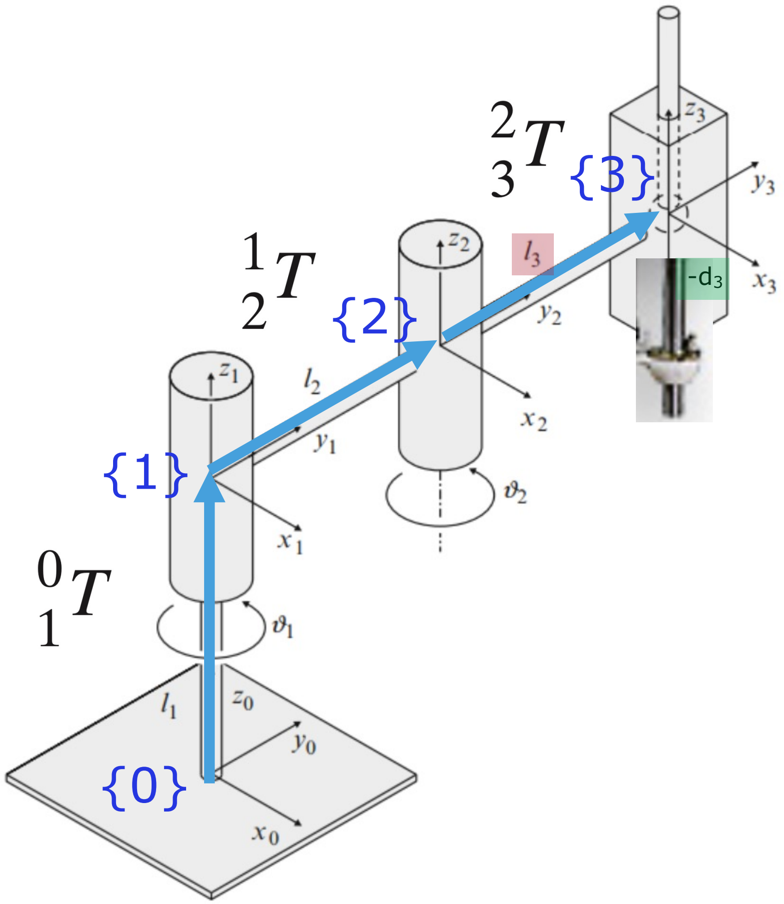
SCARA* - Forward Kinematics - HT Matrices



$$^{0}_{1}T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{1}_{2}T = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

SCARA* - Forward Kinematics - HT Matrices



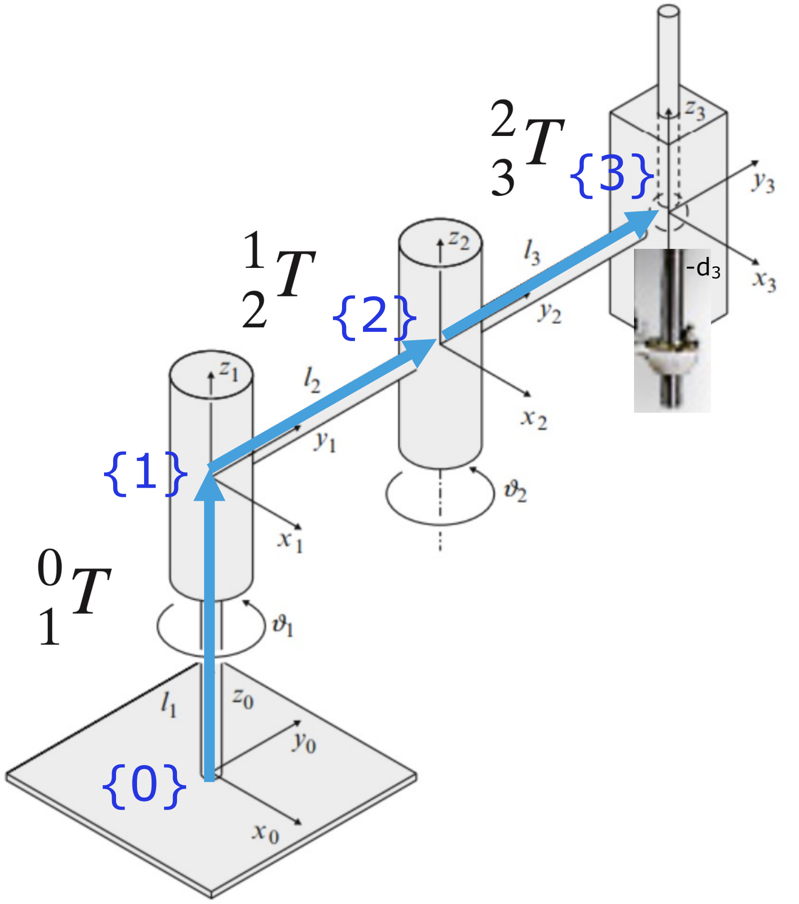
$$ {}^0_1T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^1_2T = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^2_3T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_3 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

SCARA* - Forward Kinematics in MATLAB

```
>> syms th1 th2 l1 l2 l3 d3
>> T_0_1 = [cos(th1) -sin(th1) 0 0; sin(th1) cos(th1) 0 0; 0 0 1 l1; 0 0 0 1]

T_0_1 =

[cos(th1), -sin(th1), 0,  0]
[sin(th1),  cos(th1), 0,  0]
[       0,         0, 1, l1]
[       0,         0, 0,  1]

>> T_1_2 = [cos(th2) -sin(th2) 0 0; sin(th2) cos(th2) 0 l2; 0 0 1 0; 0 0 0 1]

T_1_2 =

[cos(th2), -sin(th2), 0,  0]
[sin(th2),  cos(th2), 0, l2]
[       0,         0, 1,  0]
[       0,         0, 0,  1]

>> T_2_3 = [1 0 0 0; 0 1 0 l3; 0 0 1 -d3; 0 0 0 1]

T_2_3 =

[1, 0, 0,   0]
[0, 1, 0,  l3]
[0, 0, 1, -d3]
[0, 0, 0,   1]
```

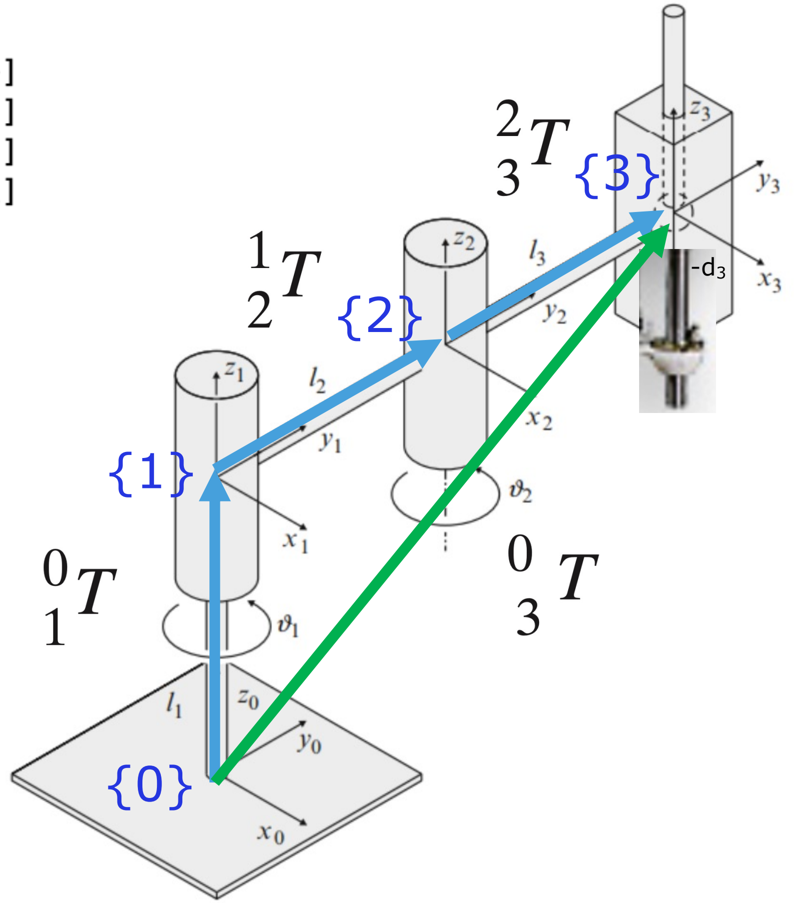


SCARA* - Forward Kinematics in MATLAB

```
T_0_1 =

[cos(th1), -sin(th1), 0,  0]
[sin(th1),  cos(th1), 0,  0]
[       0,         0, 1, l1]
[       0,         0, 0,  1]
```

```
T_1_2 =

[cos(th2), -sin(th2), 0,  0]
[sin(th2),  cos(th2), 0, l2]
[       0,         0, 1,  0]
[       0,         0, 0,  1]
```

```
T_2_3 =

[1, 0, 0,   0]
[0, 1, 0,  l3]
[0, 0, 1, -d3]
[0, 0, 0,   1]
```




```
>> T_0_3 = T_0_1 * T_1_2 * T_2_3

T_0_3 =

[cos(th1)*cos(th2) - sin(th1)*sin(th2), - cos(th1)*sin(th2) - cos(th2)*sin(th1), 0, - l3*(cos(th1)*sin(th2) + cos(th2)*sin(th1)) - l2*sin(th1)]
[cos(th1)*sin(th2) + cos(th2)*sin(th1),   cos(th1)*cos(th2) - sin(th1)*sin(th2), 0,   l3*(cos(th1)*cos(th2) - sin(th1)*sin(th2)) + l2*cos(th1)]
[                                    0,                                       0, 1,                                                     l1 - d3]
[                                    0,                                       0, 0,                                                           1]
```

SCARA* - Forward Kinematics in MATLAB

- T_0_1 =

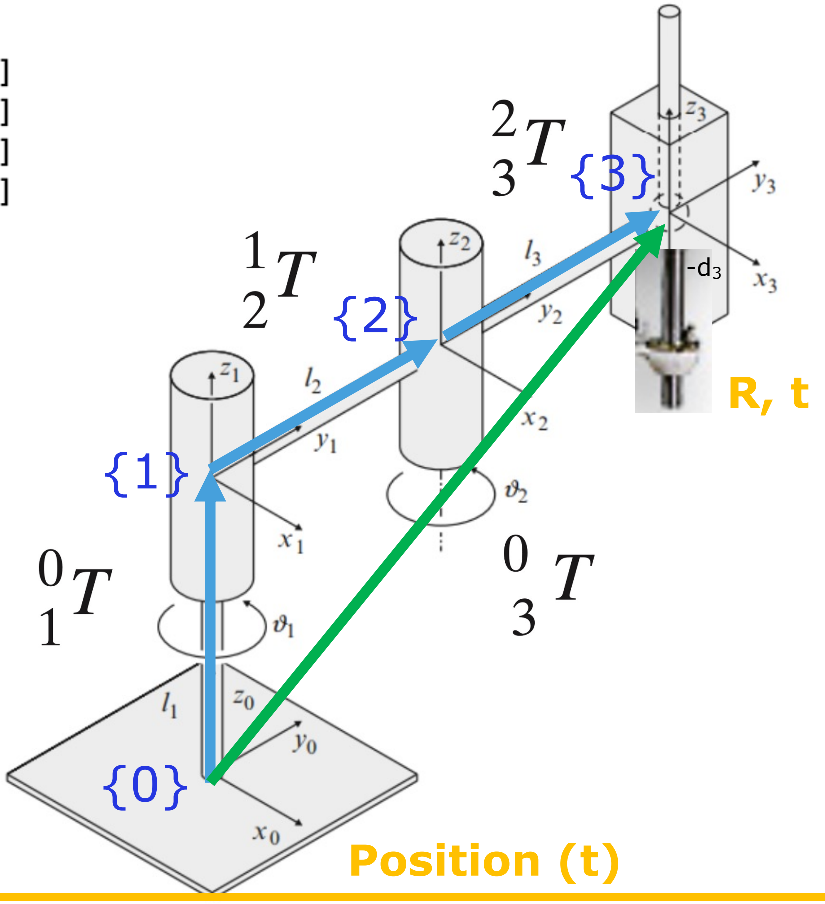
```
[cos(th1), -sin(th1), 0,  0]
[sin(th1),  cos(th1), 0,  0]
[       0,         0, 1, l1]
[       0,         0, 0,  1]
```

- T_1_2 =

```
[cos(th2), -sin(th2), 0,  0]
[sin(th2),  cos(th2), 0, l2]
[       0,         0, 1,  0]
[       0,         0, 0,  1]
```

- T_2_3 =

```
[1, 0, 0,   0]
[0, 1, 0,  l3]
[0, 0, 1, -d3]
[0, 0, 0,   1]
```



- \>> T_0_3 = T_0_1 * T_1_2 * T_2_3

Rotation (R) | Position (t)

```
T_0_3 =

[cos(th1)*cos(th2) - sin(th1)*sin(th2), - cos(th1)*sin(th2) - cos(th2)*sin(th1), 0, - l3*(cos(th1)*sin(th2) + cos(th2)*sin(th1)) - l2*sin(th1)]
[cos(th1)*sin(th2) + cos(th2)*sin(th1),   cos(th1)*cos(th2) - sin(th1)*sin(th2), 0,   l3*(cos(th1)*cos(th2) - sin(th1)*sin(th2)) + l2*cos(th1)]
[                                     0,                                       0, 1,                                                    l1 - d3]
[                                     0,                                       0, 0,                                                          1]
```

SCARA* - Forward Kinematics - Example 1

Assume that each link of the SCARA robot has length 1m and $\theta_1 = 30^o$, $\theta_2 = 30^o$ and $d_3 = 0.3m$, **what is the end-effector's rotation and position?**

SCARA* - Forward Kinematics - Example 1

Solution in MATLAB

```
>> l1 = 1

l1 =

     1

>> l2 = 1

l2 =

     1

>> l3 = 1

l3 =

     1

>> d3 = 0.3

d3 =

    0.3000
```

SCARA* - Forward Kinematics - Example 1

Solution in MATLAB

```
>> l1 = 1

l1 =

     1
```

```
>> l2 = 1

l2 =

     1
```

```
>> l3 = 1

l3 =

     1
```

```
>> d3 = 0.3

d3 =

    0.3000
```

```
>> th1 = deg2rad(30)

th1 =

    0.5236
```

```
>> th2 = deg2rad(30)

th2 =

    0.5236
```

SCARA* - Forward Kinematics - Example 1

Solution in MATLAB

```
>> l1 = 1

l1 =

     1
```

```
>> l2 = 1

l2 =

     1
```

```
>> l3 = 1

l3 =

     1
```

```
>> d3 = 0.3

d3 =

    0.3000
```

```
>> th1 = deg2rad(30)

th1 =

    0.5236
```

```
>> th2 = deg2rad(30)

th2 =

    0.5236
```

```
>> T_0_1 = [cos(th1) -sin(th1) 0 0; sin(th1) cos(th1) 0 0; 0 0 1 l1; 0 0 0 1]

T_0_1 =

    0.8660   -0.5000         0         0
    0.5000    0.8660         0         0
         0         0    1.0000    1.0000
         0         0         0    1.0000
```

```
>> T_1_2 = [cos(th2) -sin(th2) 0 0; sin(th2) cos(th2) 0 l2; 0 0 1 0; 0 0 0 1]

T_1_2 =

    0.8660   -0.5000         0         0
    0.5000    0.8660         0    1.0000
         0         0    1.0000         0
         0         0         0    1.0000
```

```
>> T_2_3 = [1 0 0 0; 0 1 0 l3; 0 0 1 -d3; 0 0 0 1]

T_2_3 =

    1.0000         0         0         0
         0    1.0000         0    1.0000
         0         0    1.0000   -0.3000
         0         0         0    1.0000
```

SCARA* - Forward Kinematics - Example 1

Solution in MATLAB

```
>> l1 = 1

l1 =

     1
```

```
>> th1 = deg2rad(30)

th1 =

    0.5236
```

```
>> l2 = 1

l2 =

     1
```

```
>> th2 = deg2rad(30)

th2 =

    0.5236
```

```
>> l3 = 1

l3 =

     1
```

```
>> d3 = 0.3

d3 =

    0.3000
```

```
>> T_0_1 = [cos(th1) -sin(th1) 0 0; sin(th1) cos(th1) 0 0; 0 0 1 l1; 0 0 0 1]

T_0_1 =

    0.8660   -0.5000         0         0
    0.5000    0.8660         0         0
         0         0    1.0000    1.0000
         0         0         0    1.0000
```

```
>> T_1_2 = [cos(th2) -sin(th2) 0 0; sin(th2) cos(th2) 0 l2; 0 0 1 0; 0 0 0 1]

T_1_2 =

    0.8660   -0.5000         0         0
    0.5000    0.8660         0    1.0000
         0         0    1.0000         0
         0         0         0    1.0000
```

```
>> T_2_3 = [1 0 0 0; 0 1 0 l3; 0 0 1 -d3; 0 0 0 1]

T_2_3 =

    1.0000         0         0         0
         0    1.0000         0    1.0000
         0         0    1.0000   -0.3000
         0         0         0    1.0000
```

```
>> T_0_3 = T_0_1 * T_1_2 * T_2_3

T_0_3 =

    0.5000   -0.8660         0   -1.3660
    0.8660    0.5000         0    1.3660
         0         0    1.0000    0.7000
         0         0         0    1.0000
```

SCARA* - Forward Kinematics - Example 1

Solution in MATLAB

```
>> l1 = 1

l1 =

     1
```

```
>> th1 = deg2rad(30)

th1 =

    0.5236
```

```
>> l2 = 1

l2 =

     1
```

```
>> th2 = deg2rad(30)

th2 =

    0.5236
```

```
>> l3 = 1

l3 =

     1
```

```
>> d3 = 0.3

d3 =

    0.3000
```

```
>> T_0_1 = [cos(th1) -sin(th1) 0 0; sin(th1) cos(th1) 0 0; 0 0 1 l1; 0 0 0 1]

T_0_1 =

    0.8660   -0.5000         0         0
    0.5000    0.8660         0         0
         0         0    1.0000    1.0000
         0         0         0    1.0000
```

```
>> T_1_2 = [cos(th2) -sin(th2) 0 0; sin(th2) cos(th2) 0 l2; 0 0 1 0; 0 0 0 1]

T_1_2 =

    0.8660   -0.5000         0         0
    0.5000    0.8660         0    1.0000
         0         0    1.0000         0
         0         0         0    1.0000
```

```
>> T_2_3 = [1 0 0 0; 0 1 0 l3; 0 0 1 -d3; 0 0 0 1]

T_2_3 =

    1.0000         0         0         0
         0    1.0000         0    1.0000
         0         0    1.0000   -0.3000
         0         0         0    1.0000
```

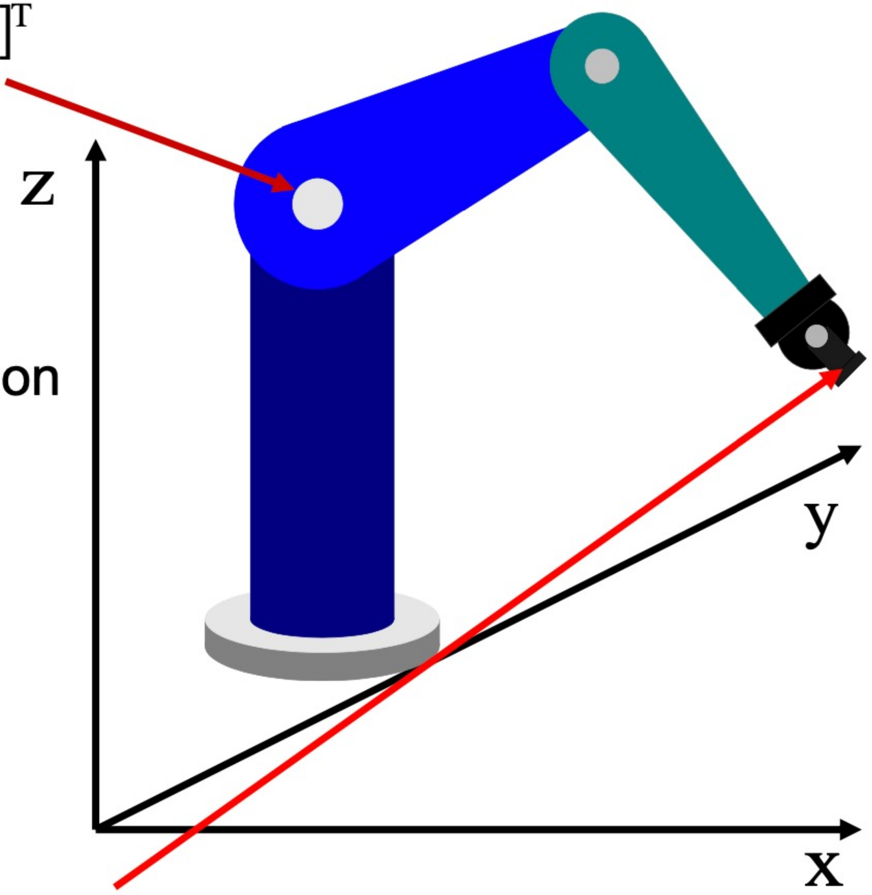
```
>> T_0_3 = T_0_1 * T_1_2 * T_2_3

T_0_3 =
                 Rotation                  Position
    0.5000   -0.8660         0            -1.3660
    0.8660    0.5000         0             1.3660
         0         0    1.0000             0.7000
         0         0         0             1.0000
```

Forward (Direct) Kinematics (FK)

$$\mathbf{q} = [q_1 \quad q_2 \quad \cdots \quad q_i \quad \cdots \quad q_{n-1} \quad q_n]^T$$

vector of joint variables

Position and Orientation of the end-effector



$$\mathbf{x}_e = [x \quad y \quad z \quad \alpha \quad \beta \quad \gamma \;]^T$$

vector of external coordinates (Cartesian coordinates)

Forward (Direct) Kinematics (FK)

- Given vector of joint variables $\mathbf{q}$
- **Task** is to determine the position and orientation of the end-effector frame (coordinate system) {n} with respect to base frame {0}

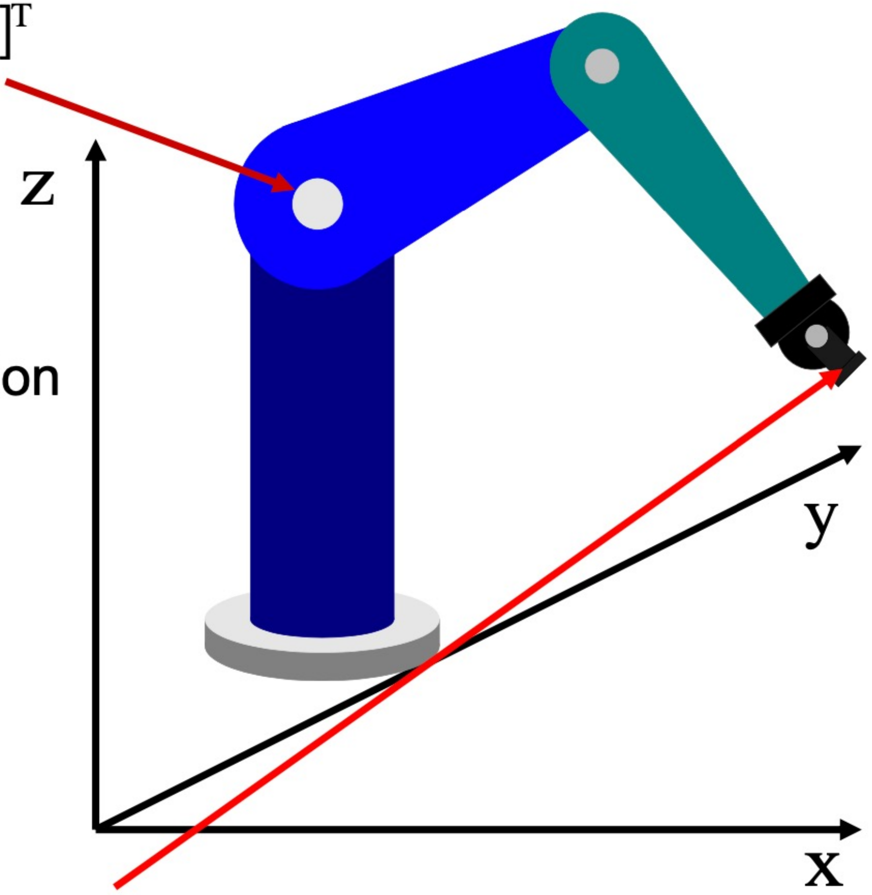
$$ {}^{0}_{n}\mathbf{T}(\mathbf{q}) = ? $$

- Define a function $\mathbf{f}()$ that maps $\mathbf{q}$ onto $\mathbf{x}_e$:

$$ \mathbf{x}_e = \mathbf{f}(\mathbf{q}) $$

$$ \mathbf{q} = [q_1 \quad q_2 \quad \cdots \quad q_i \quad \cdots \quad q_{n-1} \quad q_n]^T $$

vector of joint variables



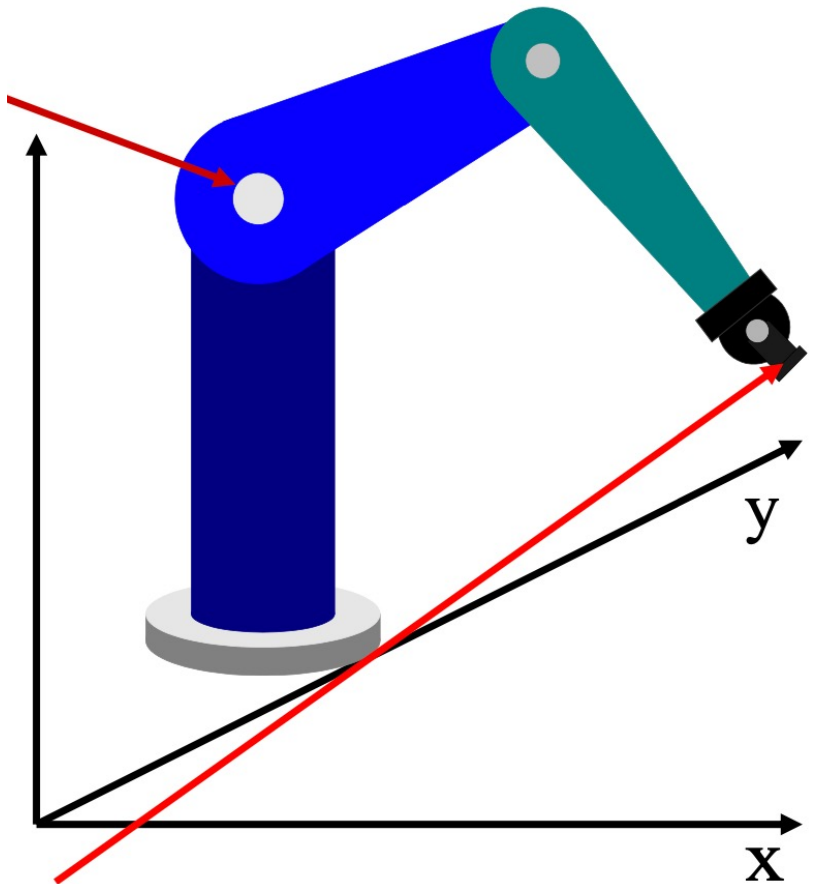
Position and Orientation of the end-effector

$$ \mathbf{x}_e = [x \quad y \quad z \quad \alpha \quad \beta \quad \gamma \;]^T $$

vector of external coordinates (Cartesian coordinates)

Forward (Direct) Kinematics (FK) - Main Tasks

- **Task #1**: Attach frames to the end of each link where the joints are located. We also attach a frame to the end of the end effector link right at the tip of the robot and we attach a frame to the base of the robot.
- **Task #2**: Find transformation matrix between each two consecutive frames starting from the base going to the end effector frame successively.
- **Task #3**: post multiply transformation matrices successively to derive the intended 4x4 homogeneous transformation matrix from the base to the end effector frame.



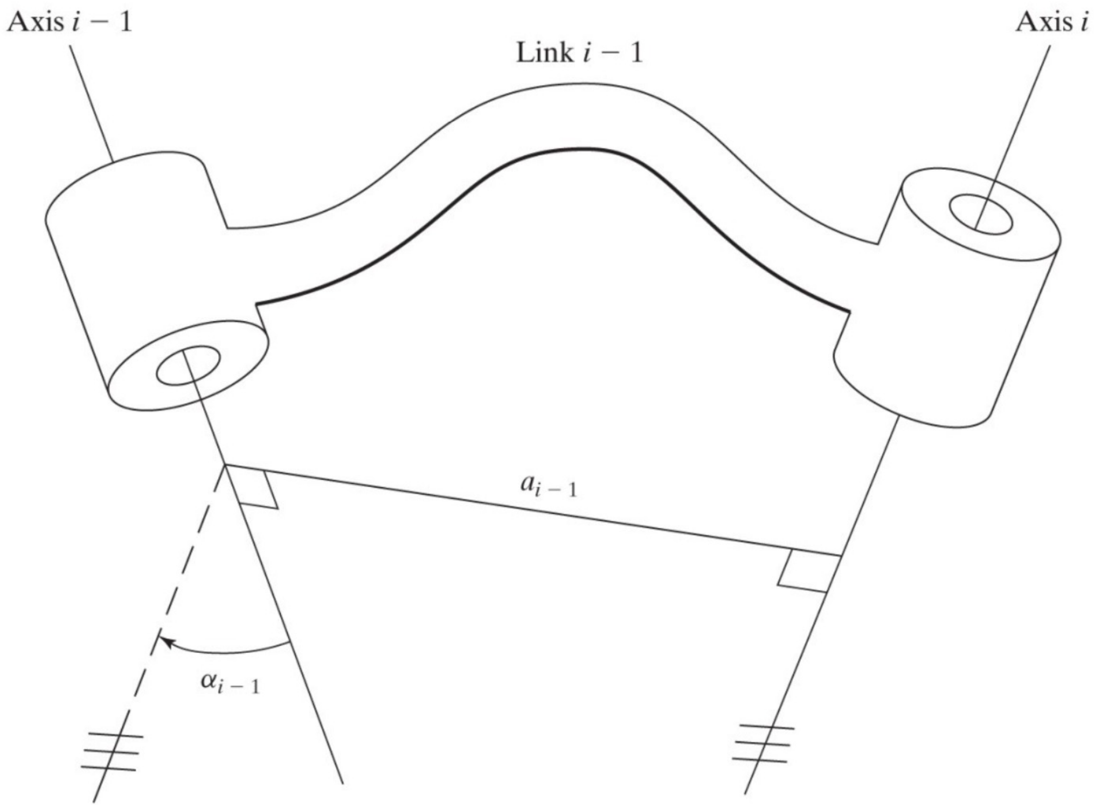
Link description

- Most robot manipulators have revolute or prismatic joints
- We will consider only manipulators that have joints with a single degree of freedom
- If a joint has > 1 DoF, the joint it can be modeled as n joints of one degree of freedom connected with $(n - 1)$ links of zero length
- The links are numbered starting from the base of the arm, which will be called link 0
- The first moving body is the <u>link 1</u>
- The end of the arm is the <u>link n</u>

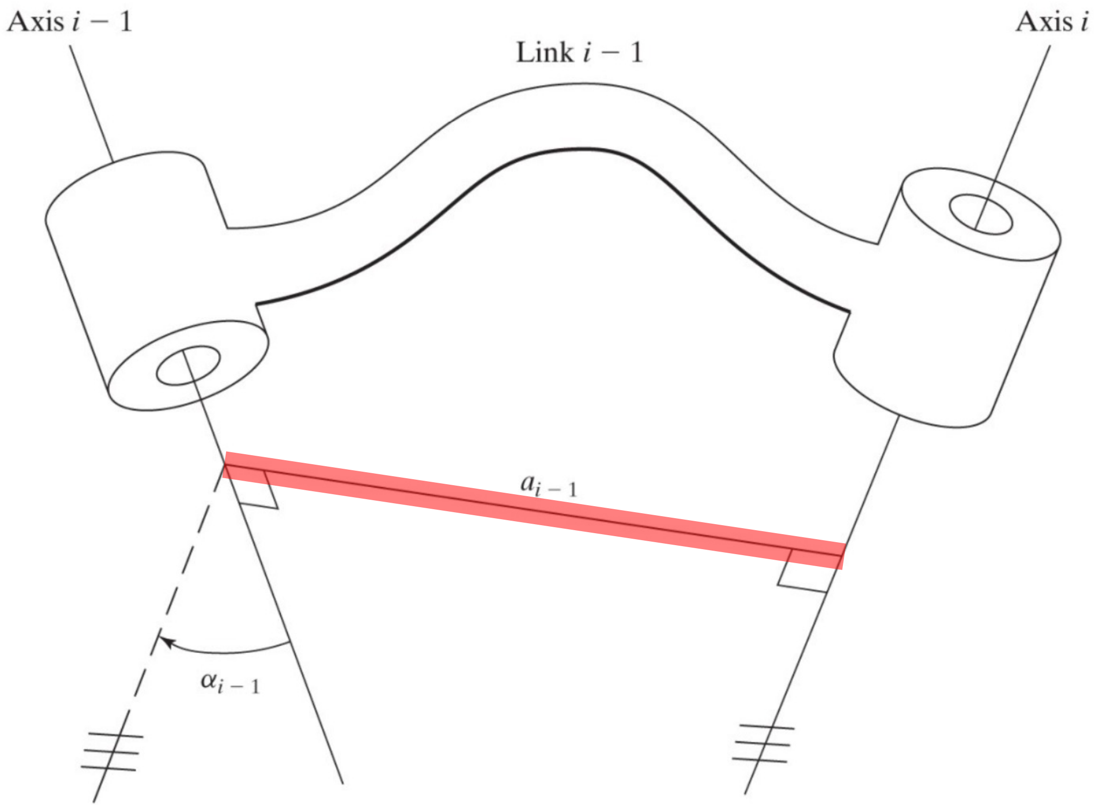
Link description

- To reach any position with any possible orientation, a minimum of six joints are required
- Link attributes:
  - the type of material used
  - the strength and stiffness of the link
  - the location and type of the joint bearings
  - the external shape
  - the weight and inertia
- A link is a rigid body that defines the relationship between two neighboring joint axes

Link description - between two joints

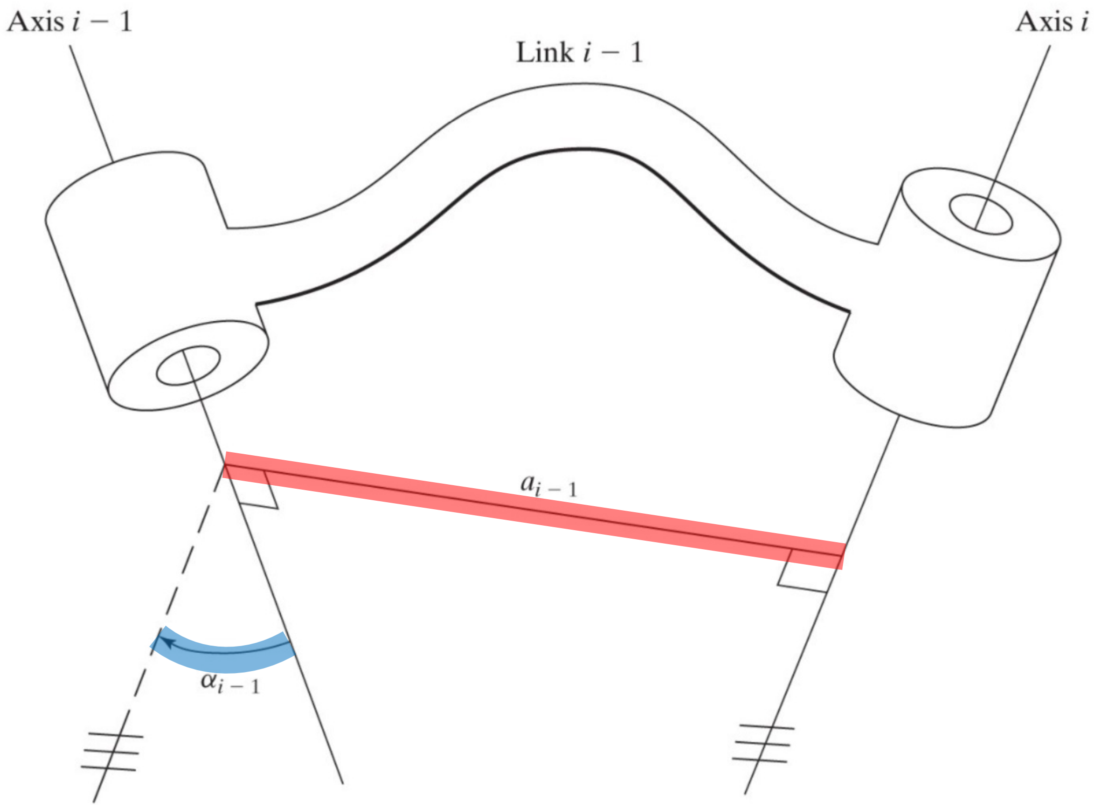


Link description - between two joints



a_{i-1} – **link length-** the length of the line perpendicular to the axis i of the joint i and the axis i - 1 of the joint i - 1

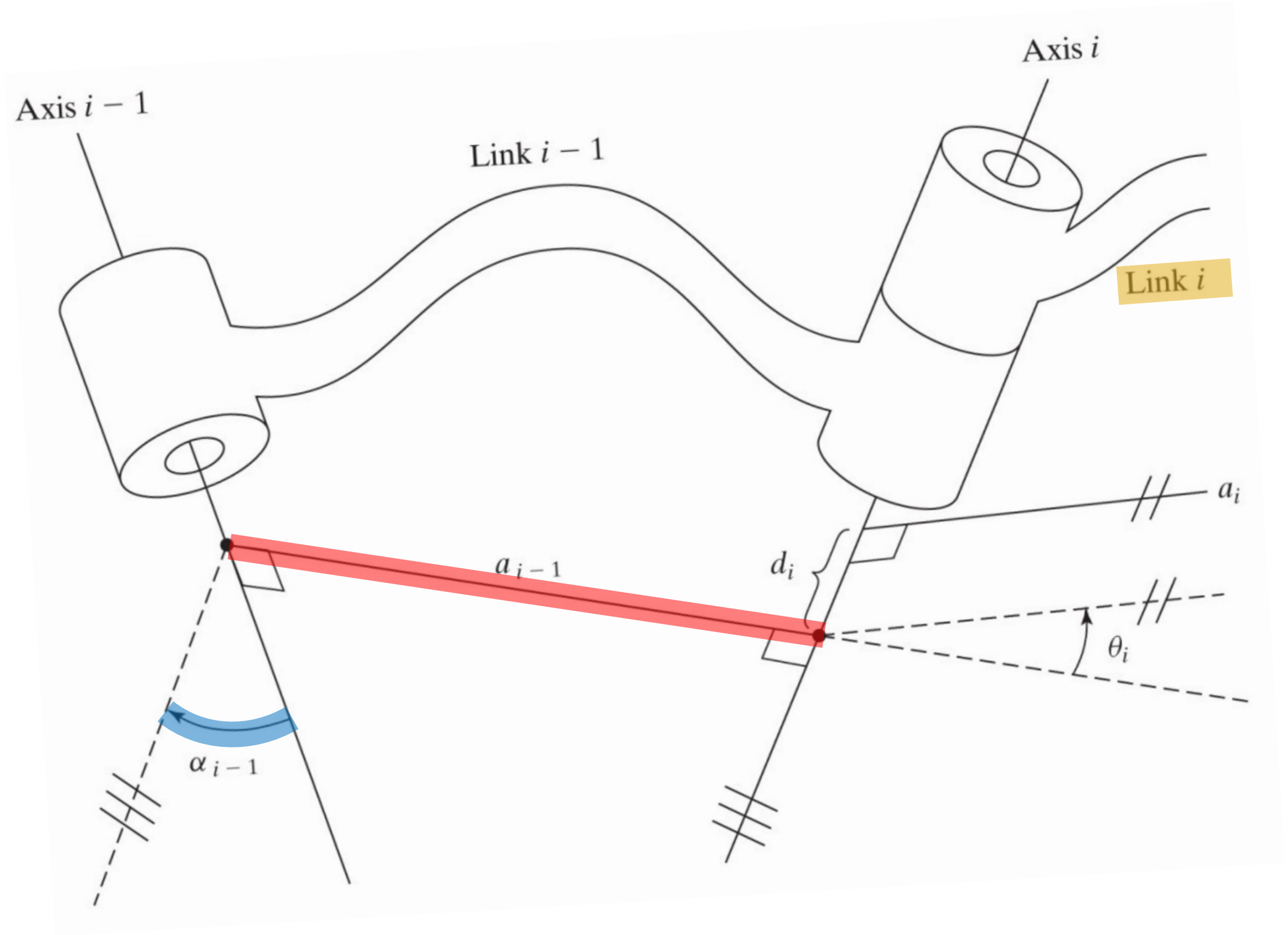
Link description - between two joints



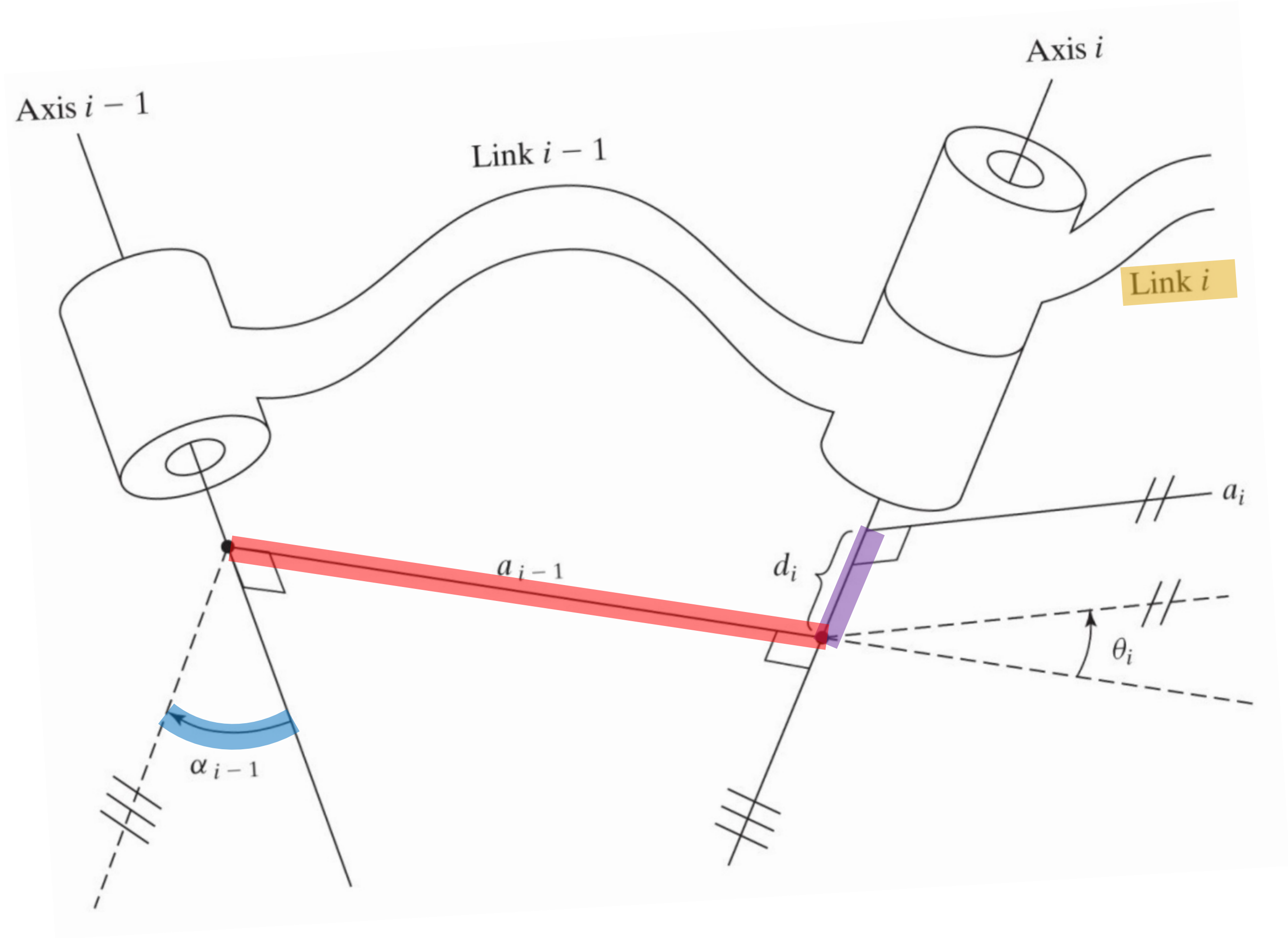
a_{i-1} – **link length-** the length of the line perpendicular to the axis i of the joint i and the axis i - 1 of the joint i - 1

α_{i-1} – **link twist-** the angle between the axes in a plane perpendicular to a_{i-1}

Link description - between neighboring links

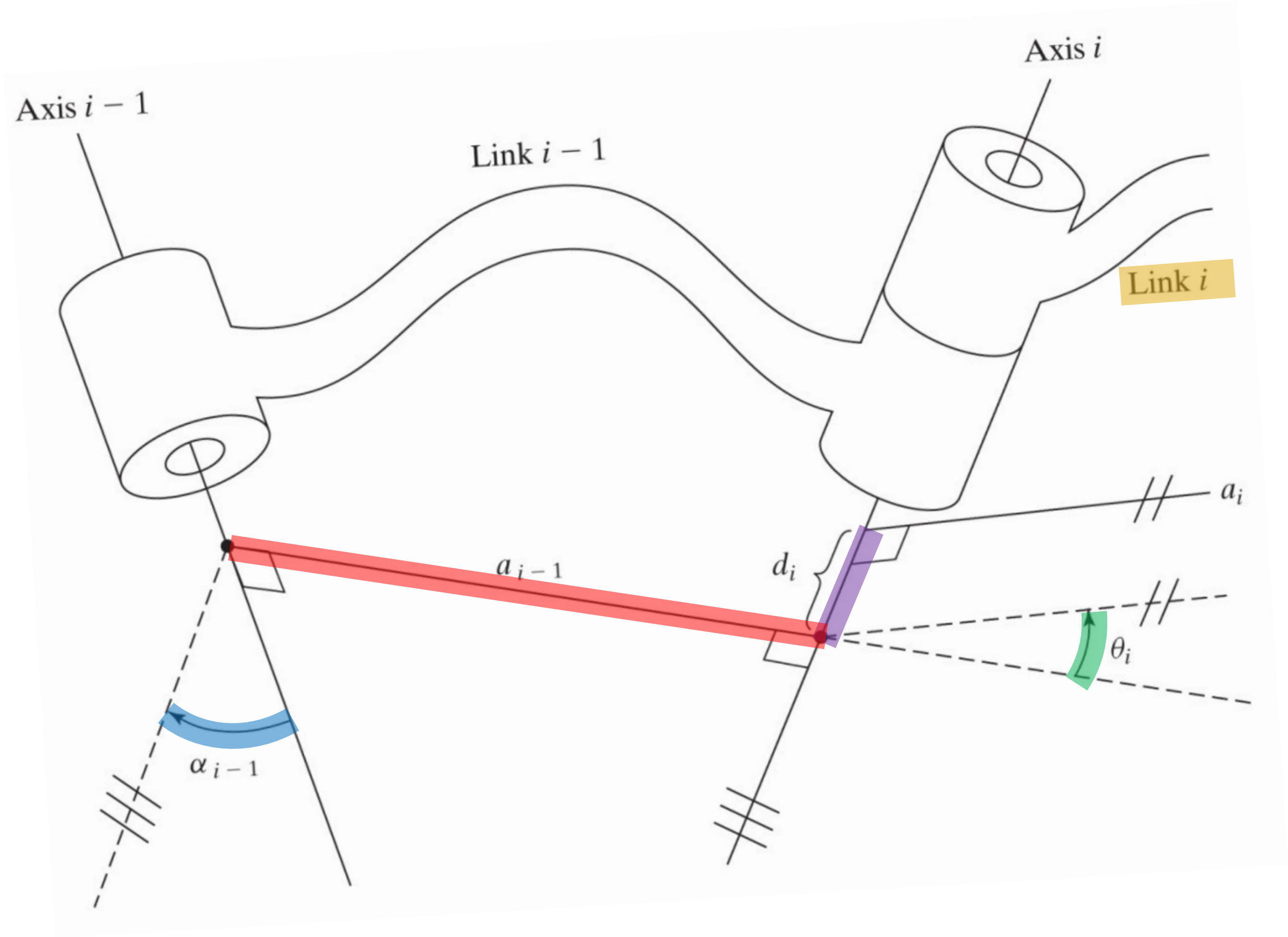


Link description - between neighboring links



d_i - **link offset** - the distance along this common axis from one link to the next

Link description - between neighboring links



d_i - **link offset** - the distance along this common axis from one link to the next

θ_i - **joint angle** - the amount of rotation about this common axis between one link and its neighbor

Link description - Denavit-Hartenberg (DH)

- Describe kinematically any robot using four parameters (variables)
  - Two describe the link itself

    (Link variables)
  - Two describe the link's connection to a neighboring link

    (Joint variables)

a_{i-1} – link length

α_{i-1} – link twist

θ_i – joint angle

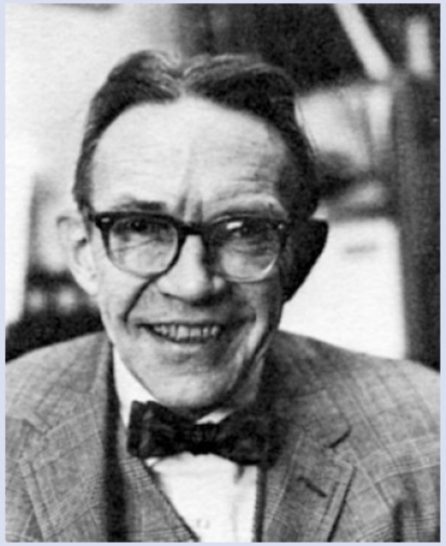
d_i – link offset

- *Revolute joint* → $\boldsymbol{\theta_i}$ → joint variable (the other three fixed link parameters).
- *Prismatic joint* → $\boldsymbol{d_i}$ → joint variable (the other three fixed link parameters)

Denavit-Hartenberg (DH) Convention

Jacques Denavit and **Richard Hartenberg** introduced many of the key concepts of kinematics for serial-link manipulators in a 1955 paper (Denavit and Hartenberg 1955) and their later classic text *Kinematic Synthesis of Linkages* (Hartenberg and Denavit 1964).

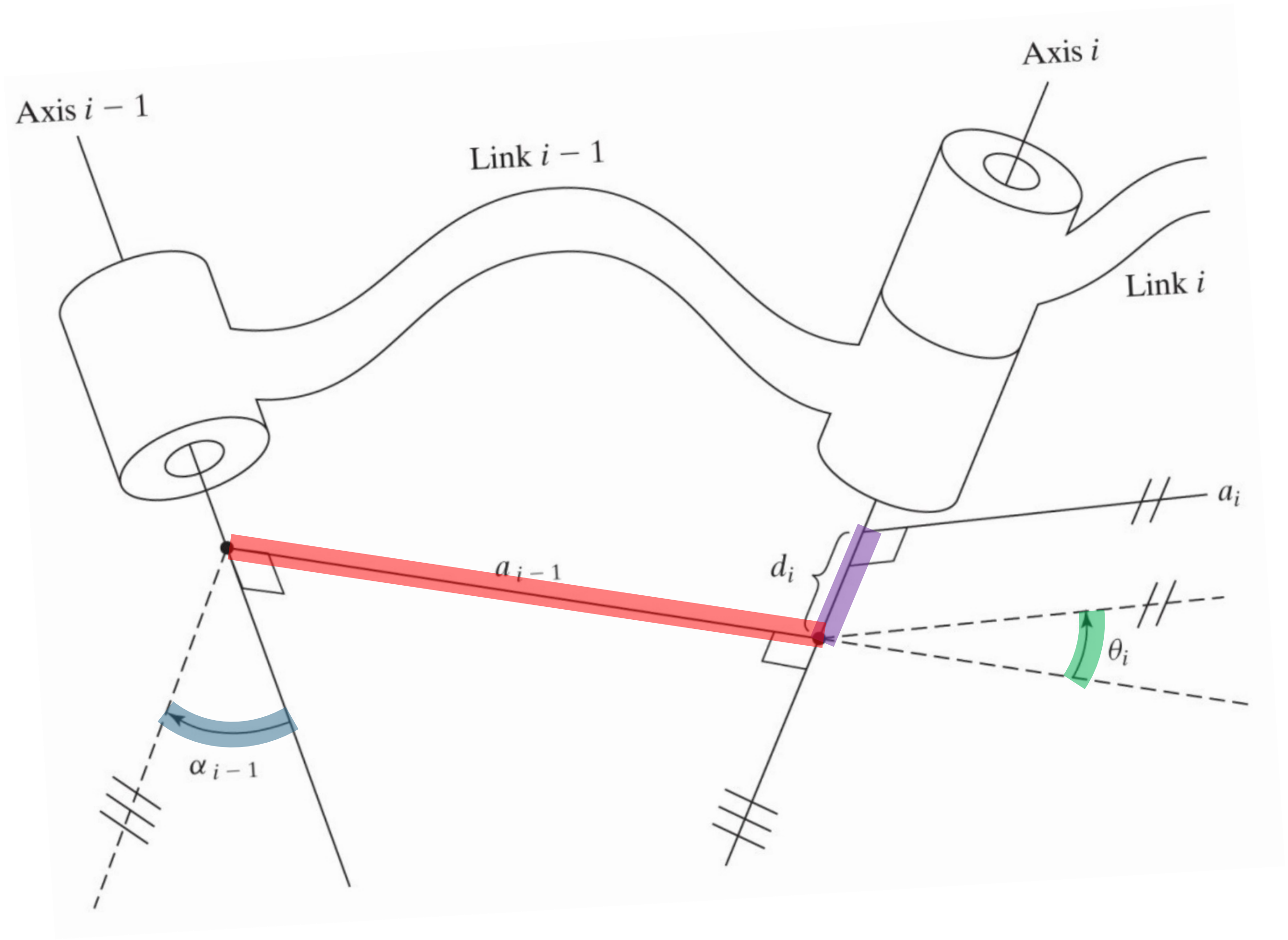
Jacques Denavit (1930–2012) was born in Paris where he studied for his Bachelor degree before pursuing his masters and doctoral degrees in mechanical engineering at Northwestern University, Illinois. In 1958 he joined the Department of Mechanical Engineering and Astronautical Science at Northwestern where the collaboration with Hartenberg was formed. In addition to his interest in dynamics and kinematics Denavit was also interested in plasma physics and kinetics. After the publication of the book he moved to Lawrence Livermore National Lab, Livermore, California, where he undertook research on computer anal-



Richard Hartenberg (1907–1997) was born in Chicago and studied for his degrees at the University of Wisconsin, Madison. He served in the merchant marine and studied aeronautics for two years at the University of Göttingen with space-flight pioneer Theodore von Kármán. He was Professor of mechanical engineering at Northwestern University where he taught for 56 years. His research in kinematics led to a revival of interest in this field in the 1960s, and his efforts helped put kinematics on a scientific basis for use in computer applications in the analysis and design of complex mechanisms. He also wrote

https://asmedigitalcollection.asme.org/appliedmechanics/article-abstract/22/2/215/1110292/A-Kinematic-Notation-for-Lower-Pair-Mechanisms?redirectedFrom=fulltext

Link description - DH parameters



a_{i-1} – link length

α_{i-1} – link twist

θ_i – joint angle

d_i – link offset

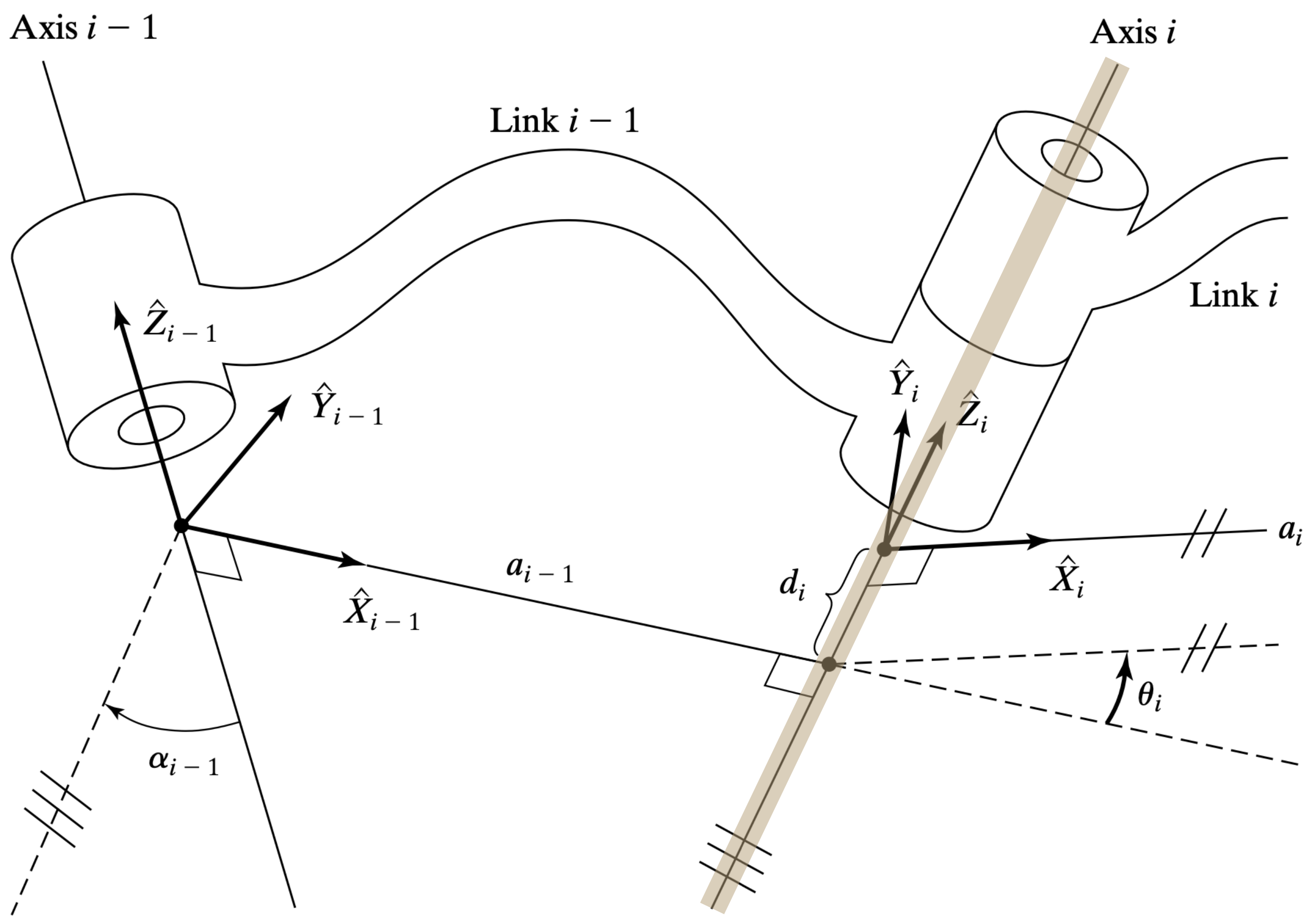
Link-frame attachment procedure

1. Identify the joint axes and imagine (or draw) infinite lines along them. For steps 2 through 5 below, consider two of these neighboring lines (at axes i and $i+1$).
2. Identify the common perpendicular between them, or point of intersection. At the point of intersection, or at the point where the common perpendicular meets the ith axis, assign the link-frame origin.
3. Assign the $\hat{Z}_i$ axis pointing along the ith joint axis.
4. Assign the $\hat{X}_i$ axis pointing along the common perpendicular, or, if the axes intersect, assign $\hat{X}_i$ to be normal to the plane containing the two axes.
5. Assign the $\hat{Y}_i$ axis to complete a right-hand coordinate system.
6. Assign $\{0\}$ to match $\{1\}$ when the first joint variable is zero. For $\{N\}$, choose an origin location and $\hat{X}_N$ direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

Link-frame attachment procedure

1. Identify the joint axes and imagine (or draw) infinite lines along them. For steps 2 through 5 below, consider two of these neighboring lines (at axes i and $i+1$).
2. Identify the common perpendicular between them, or point of intersection. At the point of intersection, or at the point where the common perpendicular meets the ith axis, assign the link-frame origin.
3. Assign the $\hat{Z}_i$ axis pointing along the ith joint axis.
4. Assign the $\hat{X}_i$ axis pointing along the common perpendicular, or, if the axes intersect, assign $\hat{X}_i$ to be normal to the plane containing the two axes.
5. Assign the $\hat{Y}_i$ axis to complete a right-hand coordinate system.
6. Assign $\{0\}$ to match $\{1\}$ when the first joint variable is zero. For $\{N\}$, choose an origin location and $\hat{X}_N$ direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

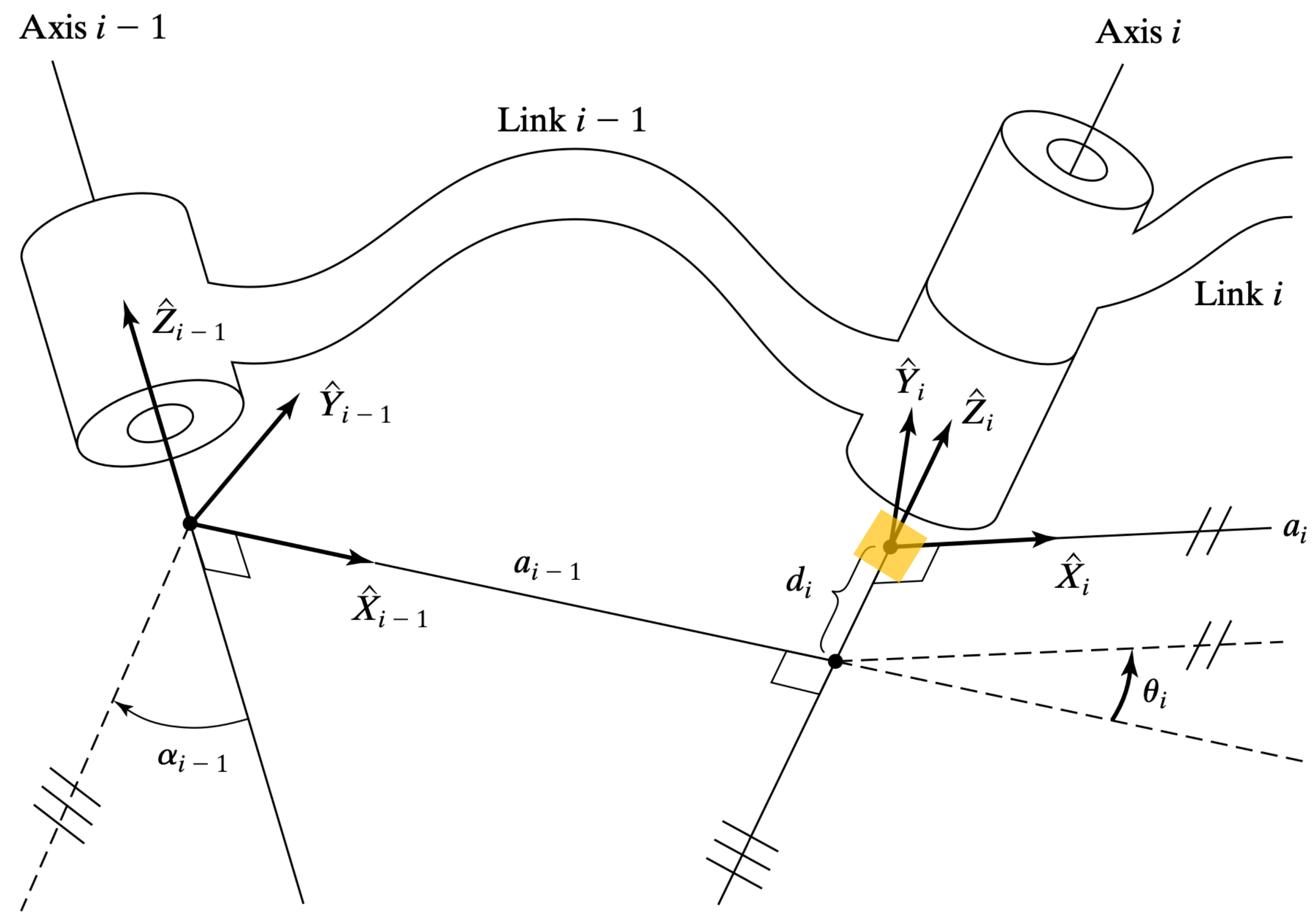
Link-frame attachment procedure



Link-frame attachment procedure

1. Identify the joint axes and imagine (or draw) infinite lines along them. For steps 2 through 5 below, consider two of these neighboring lines (at axes i and $i+1$).
2. Identify the common perpendicular between them, or point of intersection. At the point of intersection, or at the point where the common perpendicular meets the ith axis, assign the link-frame origin.
3. Assign the $\hat{Z}_i$ axis pointing along the ith joint axis.
4. Assign the $\hat{X}_i$ axis pointing along the common perpendicular, or, if the axes intersect, assign $\hat{X}_i$ to be normal to the plane containing the two axes.
5. Assign the $\hat{Y}_i$ axis to complete a right-hand coordinate system.
6. Assign $\{0\}$ to match $\{1\}$ when the first joint variable is zero. For $\{N\}$, choose an origin location and $\hat{X}_N$ direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

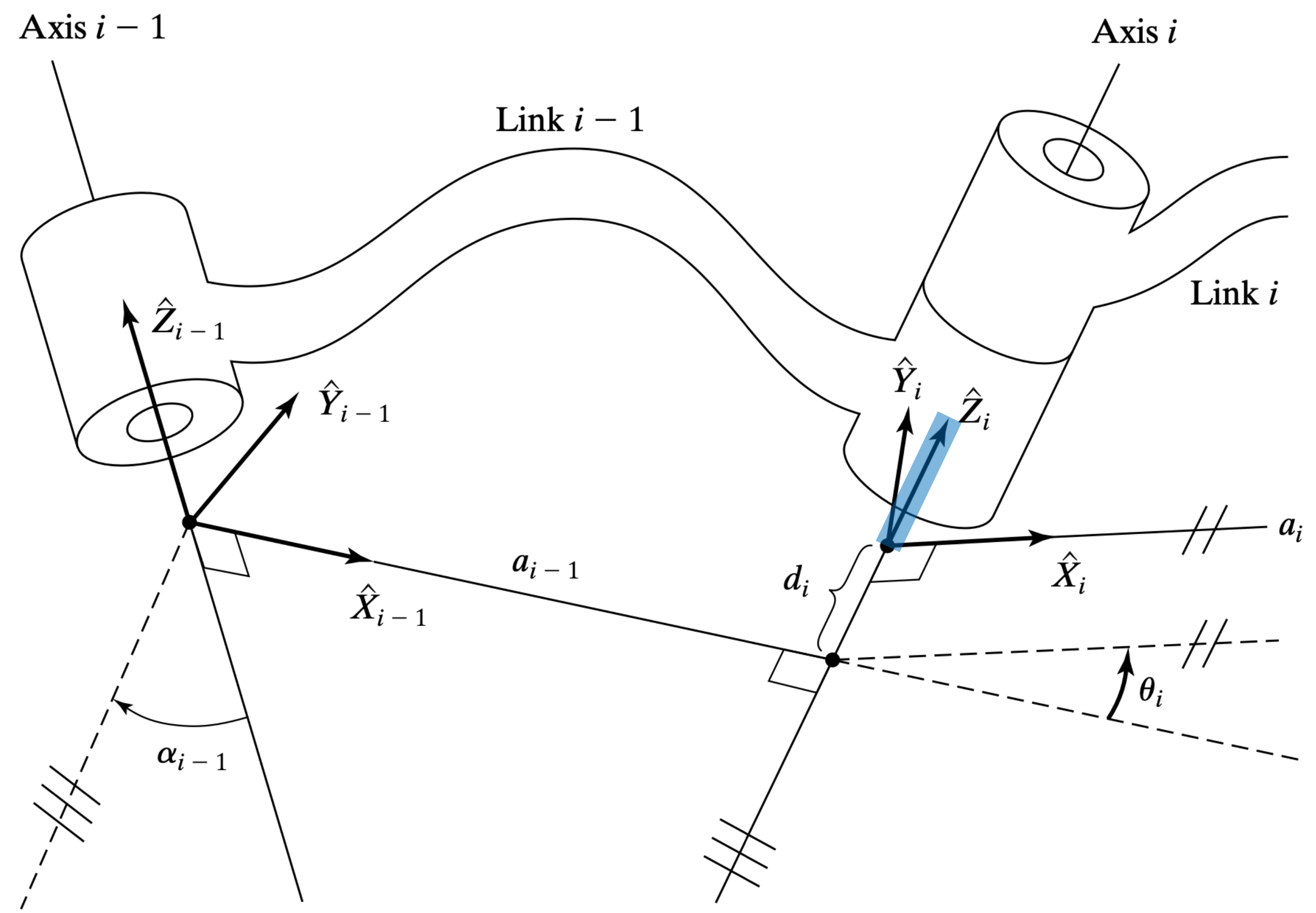
Link-frame attachment procedure



Link-frame attachment procedure

1. Identify the joint axes and imagine (or draw) infinite lines along them. For steps 2 through 5 below, consider two of these neighboring lines (at axes i and $i+1$).
2. Identify the common perpendicular between them, or point of intersection. At the point of intersection, or at the point where the common perpendicular meets the ith axis, assign the link-frame origin.
3. Assign the $\hat{Z}_i$ axis pointing along the ith joint axis.
4. Assign the $\hat{X}_i$ axis pointing along the common perpendicular, or, if the axes intersect, assign $\hat{X}_i$ to be normal to the plane containing the two axes.
5. Assign the $\hat{Y}_i$ axis to complete a right-hand coordinate system.
6. Assign $\{0\}$ to match $\{1\}$ when the first joint variable is zero. For $\{N\}$, choose an origin location and $\hat{X}_N$ direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

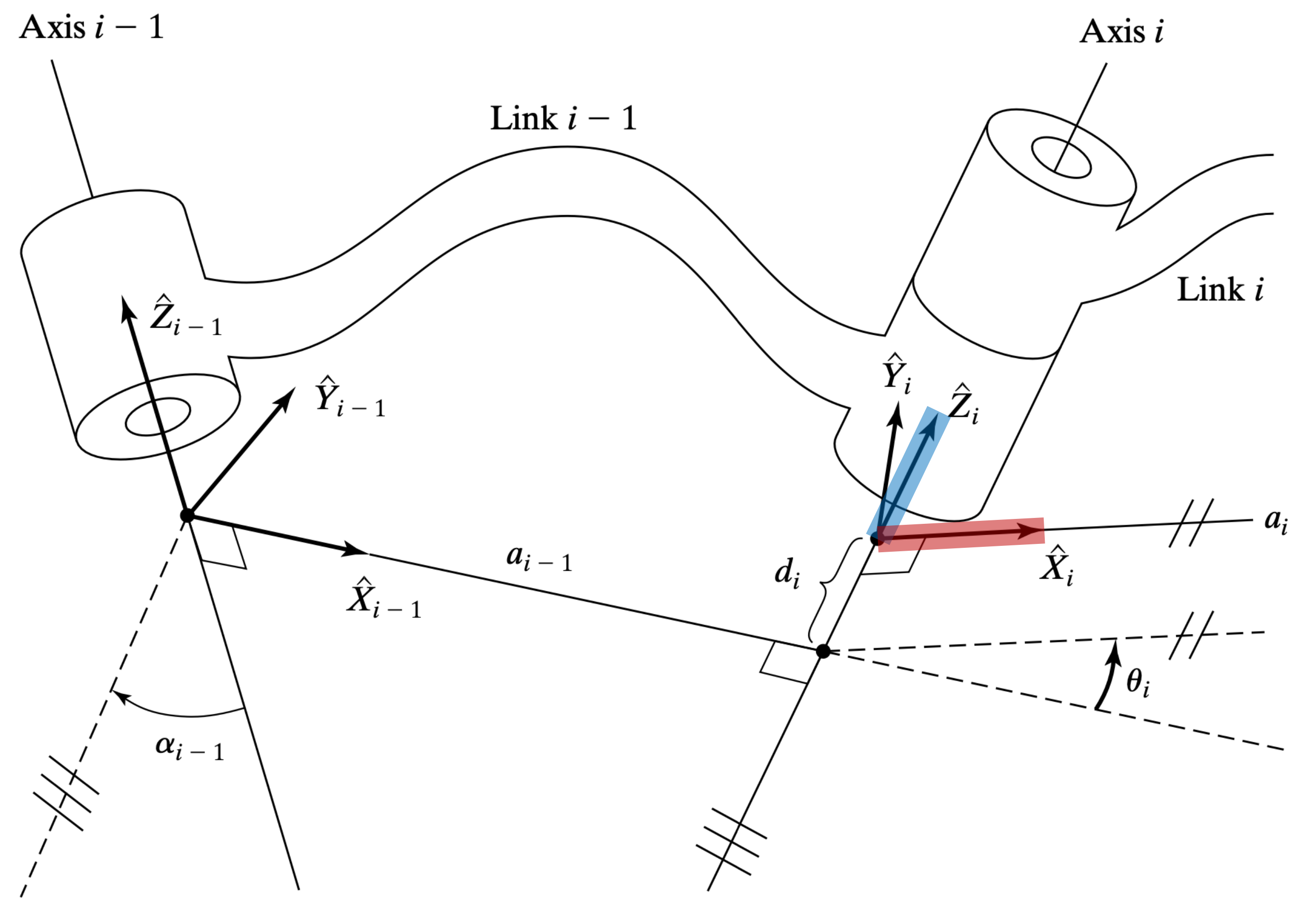
Link-frame attachment procedure



Link-frame attachment procedure

1. Identify the joint axes and imagine (or draw) infinite lines along them. For steps 2 through 5 below, consider two of these neighboring lines (at axes i and $i+1$).
2. Identify the common perpendicular between them, or point of intersection. At the point of intersection, or at the point where the common perpendicular meets the ith axis, assign the link-frame origin.
3. Assign the $\hat{Z}_i$ axis pointing along the ith joint axis.
4. Assign the $\hat{X}_i$ axis pointing along the common perpendicular, or, if the axes intersect, assign $\hat{X}_i$ to be normal to the plane containing the two axes.
5. Assign the $\hat{Y}_i$ axis to complete a right-hand coordinate system.
6. Assign $\{0\}$ to match $\{1\}$ when the first joint variable is zero. For $\{N\}$, choose an origin location and $\hat{X}_N$ direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

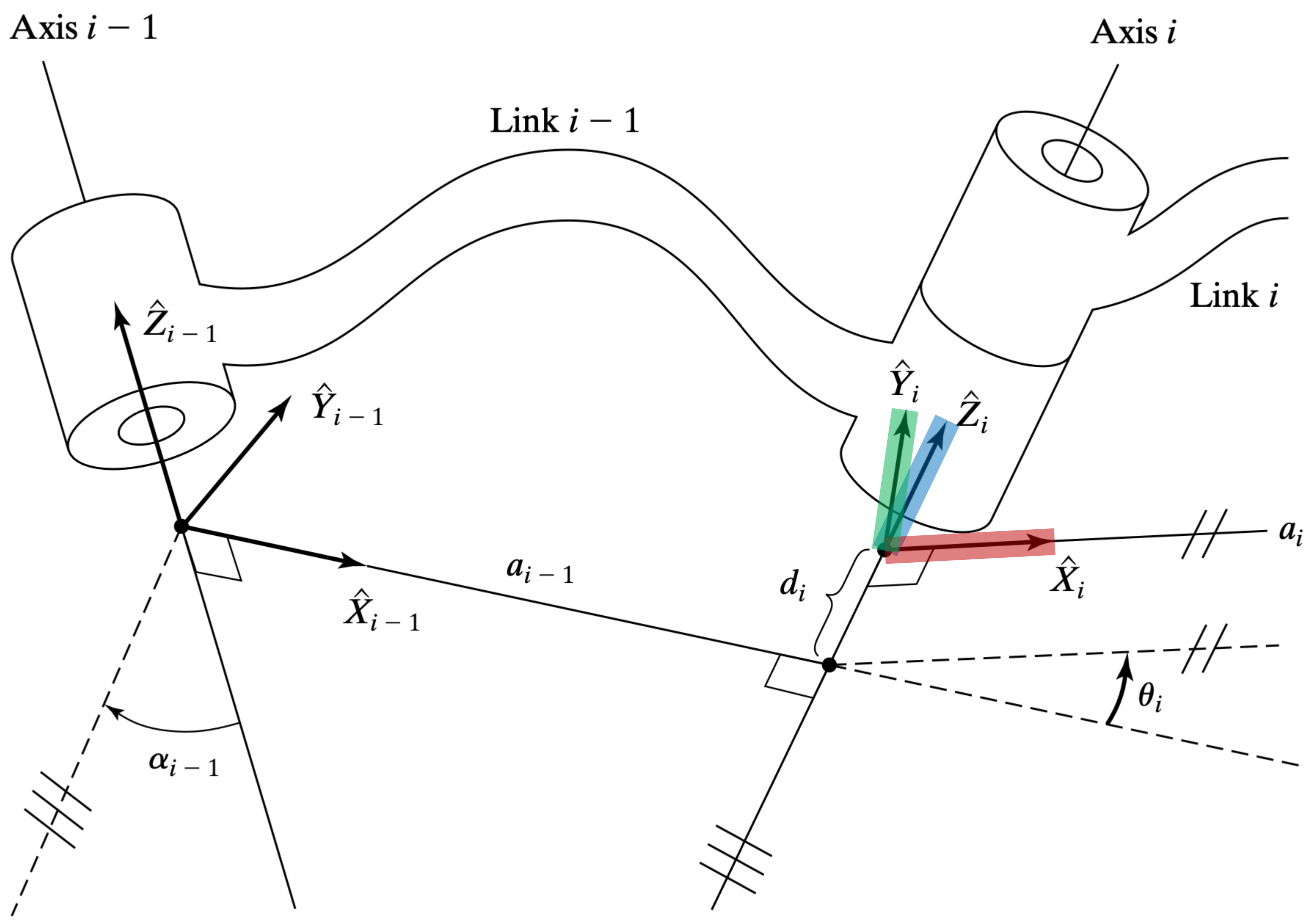
Link-frame attachment procedure

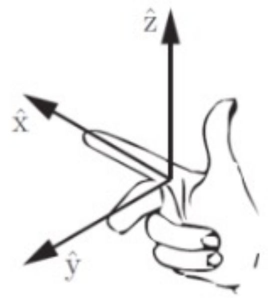


Link-frame attachment procedure

1. Identify the joint axes and imagine (or draw) infinite lines along them. For steps 2 through 5 below, consider two of these neighboring lines (at axes i and $i+1$).
2. Identify the common perpendicular between them, or point of intersection. At the point of intersection, or at the point where the common perpendicular meets the ith axis, assign the link-frame origin.
3. Assign the $\hat{Z}_i$ axis pointing along the ith joint axis.
4. Assign the $\hat{X}_i$ axis pointing along the common perpendicular, or, if the axes intersect, assign $\hat{X}_i$ to be normal to the plane containing the two axes.
5. Assign the $\hat{Y}_i$ axis to complete a right-hand coordinate system.
6. Assign $\{0\}$ to match $\{1\}$ when the first joint variable is zero. For $\{N\}$, choose an origin location and $\hat{X}_N$ direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

Link-frame attachment procedure





Link-frame attachment procedure

1. Identify the joint axes and imagine (or draw) infinite lines along them. For steps 2 through 5 below, consider two of these neighboring lines (at axes i and $i+1$).
2. Identify the common perpendicular between them, or point of intersection. At the point of intersection, or at the point where the common perpendicular meets the ith axis, assign the link-frame origin.
3. Assign the $\hat{Z}_i$ axis pointing along the ith joint axis.
4. Assign the $\hat{X}_i$ axis pointing along the common perpendicular, or, if the axes intersect, assign $\hat{X}_i$ to be normal to the plane containing the two axes.
5. Assign the $\hat{Y}_i$ axis to complete a right-hand coordinate system.
6. Assign $\{0\}$ to match $\{1\}$ when the first joint variable is zero. For $\{N\}$, choose an origin location and $\hat{X}_N$ direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

Link-frame attachment procedure

1. Identify the joint axes and imagine (or draw) infinite lines along them. For steps 2 through 5 below, consider two of these neighboring lines (at axes i and $i+1$).
2. Identify the common perpendicular between them, or point of intersection. At the point of intersection, or at the point where the common perpendicular meets the ith axis, assign the link-frame origin.
3. Assign the $\hat{Z}_i$ axis pointing along the ith joint axis.
4. Assign the $\hat{X}_i$ axis pointing along the common perpendicular, or, if the axes intersect, assign $\hat{X}_i$ to be normal to the plane containing the two axes.
5. Assign the $\hat{Y}_i$ axis to complete a right-hand coordinate system.
6. Assign $\{0\}$ to match $\{1\}$ when the first joint variable is zero. For $\{N\}$, choose an origin location and $\hat{X}_N$ direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

Summary of the link parameters in terms of the link frames

$$\alpha_i = the\ angle\ from\ \hat{Z}_i\ to\ \hat{Z}_{i+1}\ measured\ about\ \hat{X}_i;$$

$\alpha_{i\text{-}1}$ – link twist

$$a_i = the\ distance\ from\ \hat{Z}_i\ to\ \hat{Z}_{i+1}\ measured\ along\ \hat{X}_i;$$

$a_{i\text{-}1}$ – link length

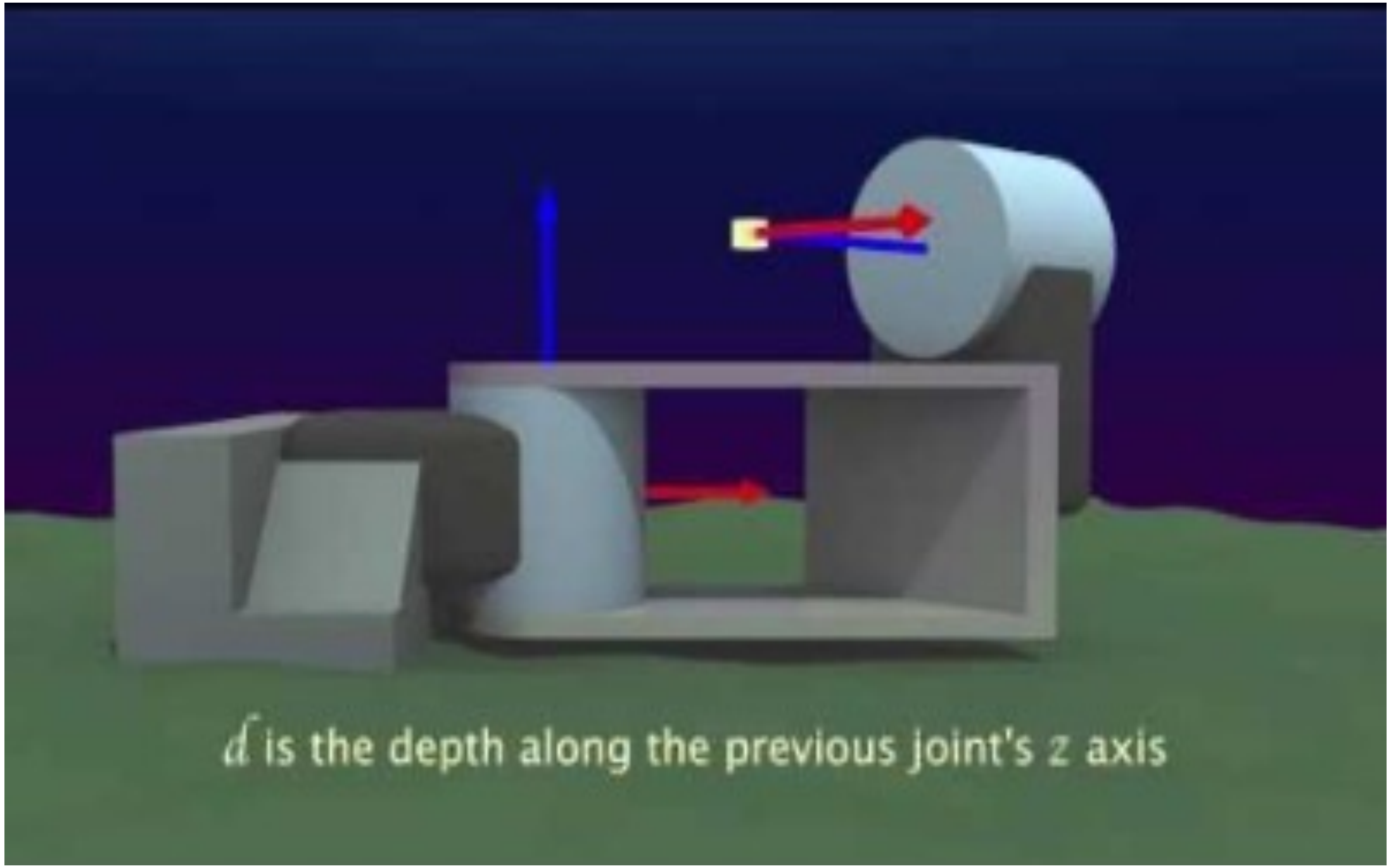
$$d_i = the\ distance\ from\ \hat{X}_{i-1}\ to\ \hat{X}_i\ measured\ along\ \hat{Z}_i;$$

d_i – link offset

$$\theta_i = the\ angle\ from\ \hat{X}_{i-1}\ to\ \hat{X}_i\ measured\ about\ \hat{Z}_i.$$

θ_i – joint angle

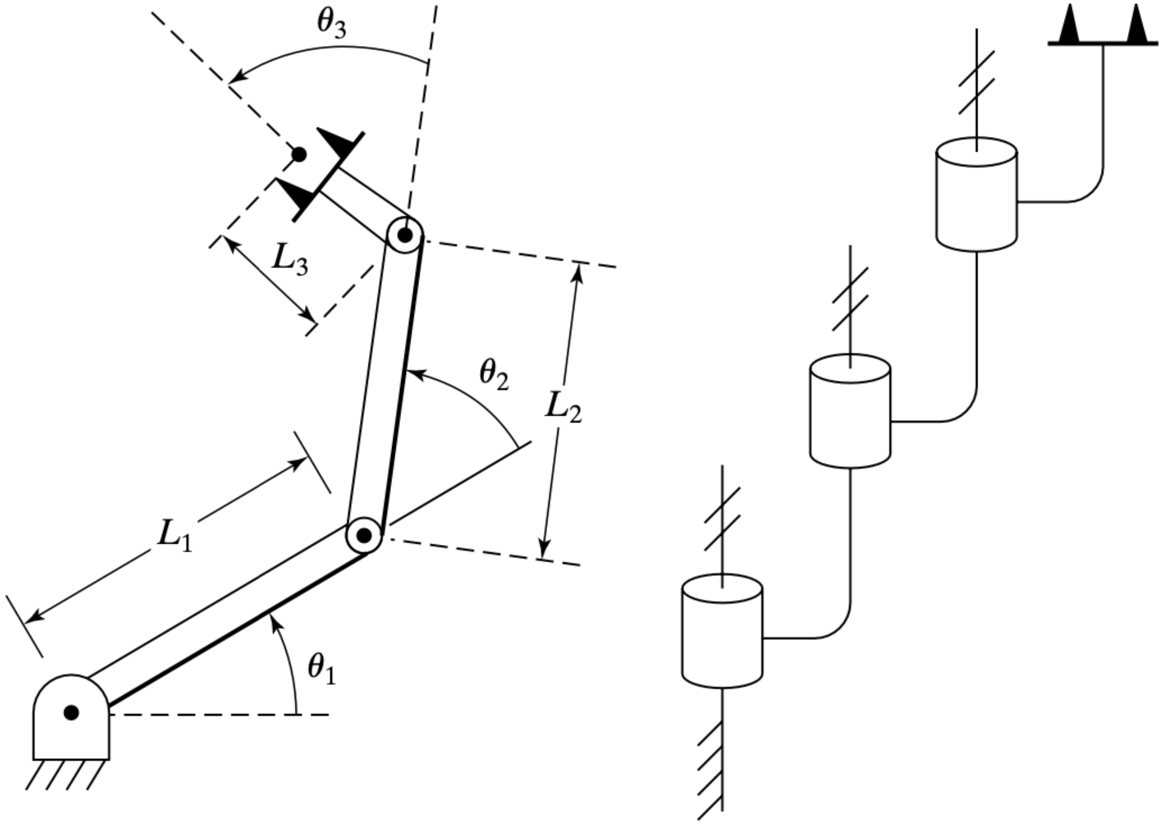
Attach frames to links - Tutorial



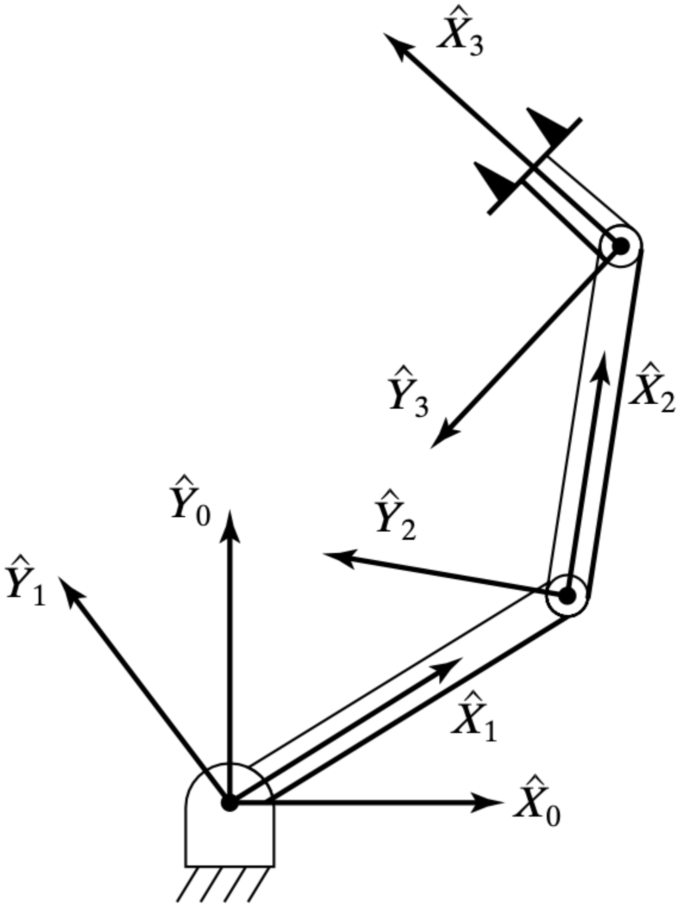
https://www.youtube.com/watch?v=rA9tm0gTln8

RRR manipulator - Example 2

- Consider a three-link planar arm with all revolute joints (RRR):
  - <u>Attach link frames to the mechanism</u>
  - <u>Find the DH parameters</u>

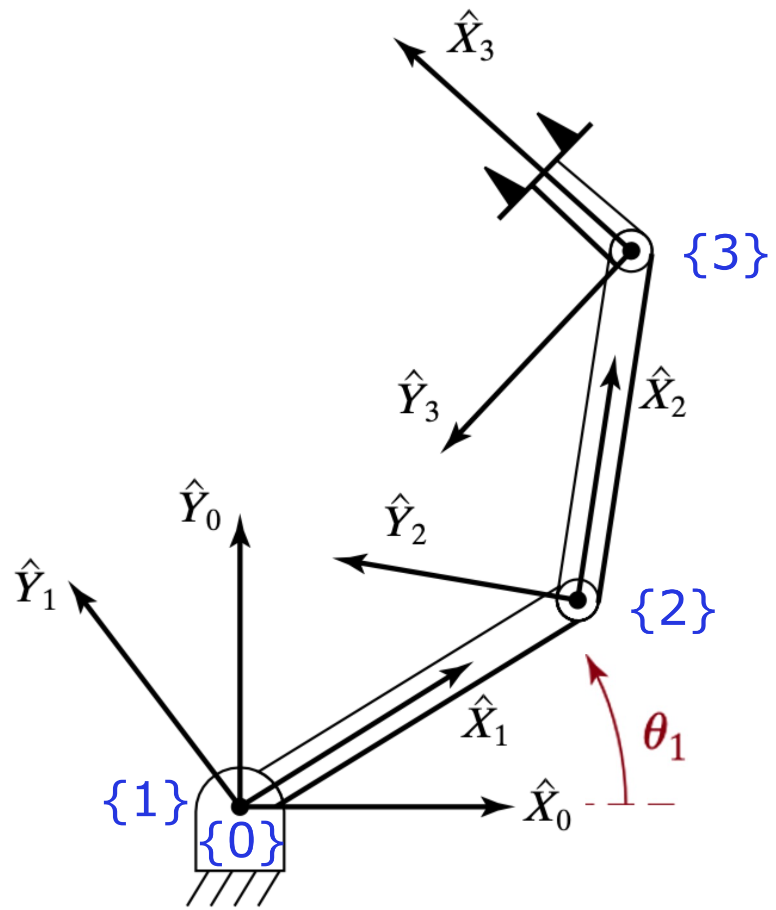


RRR manipulator - Example 2 - Attach frames



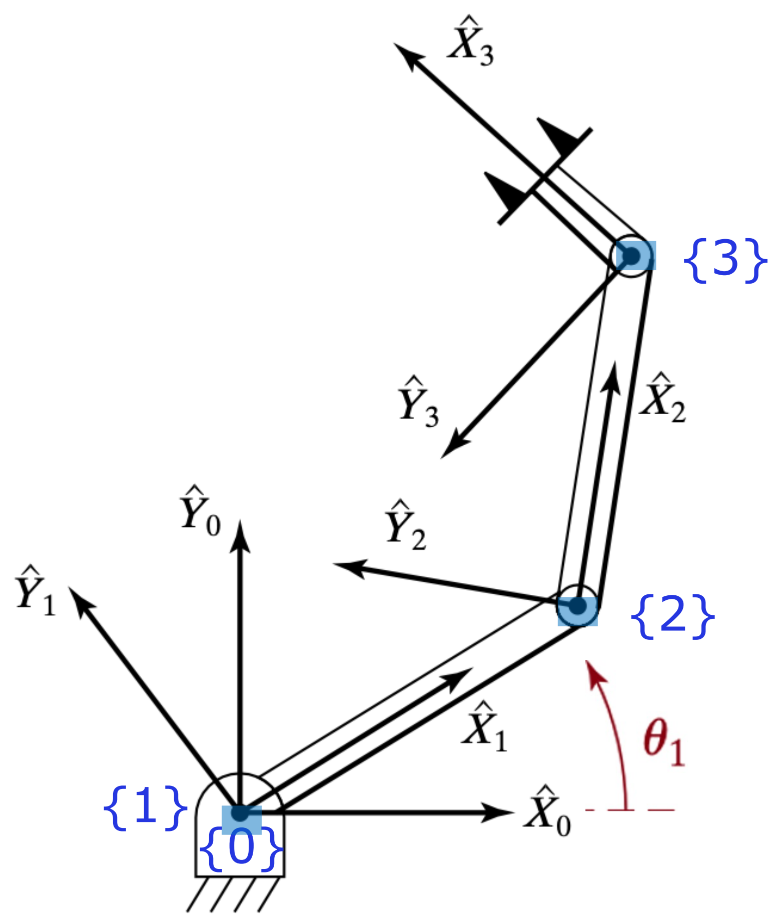
RRR manipulator - Example 2 - Attach frames

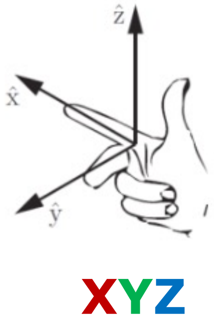
- Frame {0} is fixed to the base and aligns with frame {1} when the first joint variable θ_1 is zero



RRR manipulator - Example 2 - Attach frames

- Frame {0} is fixed to the base and aligns with frame {1} when the first joint variable θ_1 is zero
- All ***Z axes*** parallel pointing out to of the screen/paper, there are no link offsets and twists



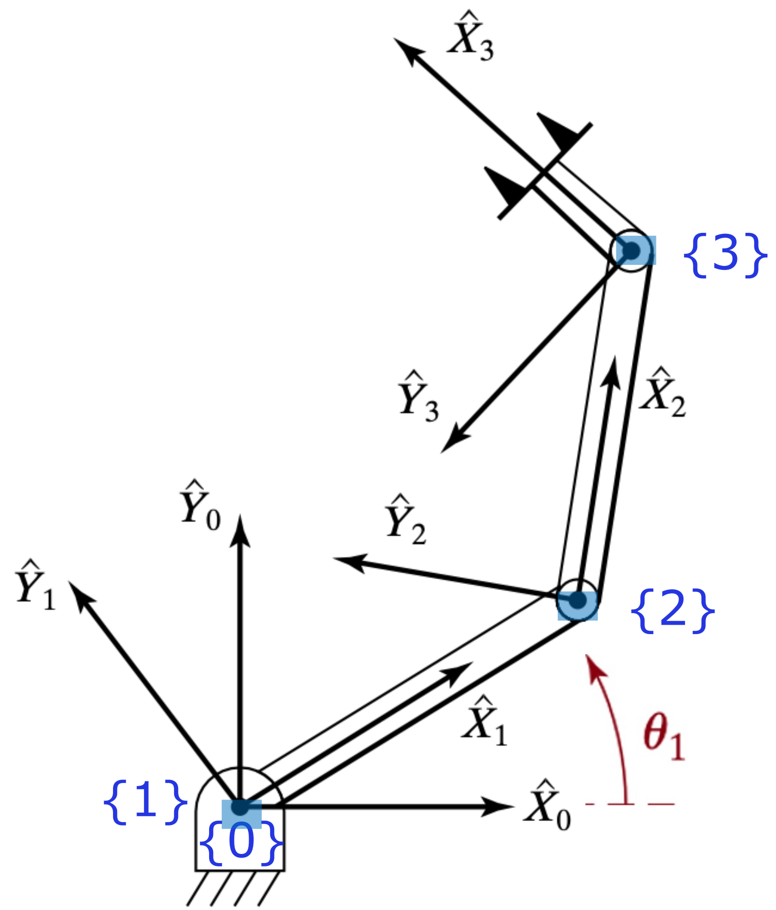


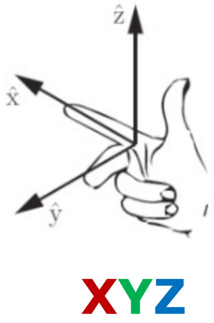
RRR manipulator - Example 2 - Attach frames

- Frame {0} is fixed to the base and aligns with frame {1} when the first joint variable θ_1 is zero
- All ***Z axes*** parallel pointing out to of the screen/paper, there are no link offsets and twists
  - All d_i are zero
  - All α_i are zero

d_i – link offset

α_{i-1} – link twist





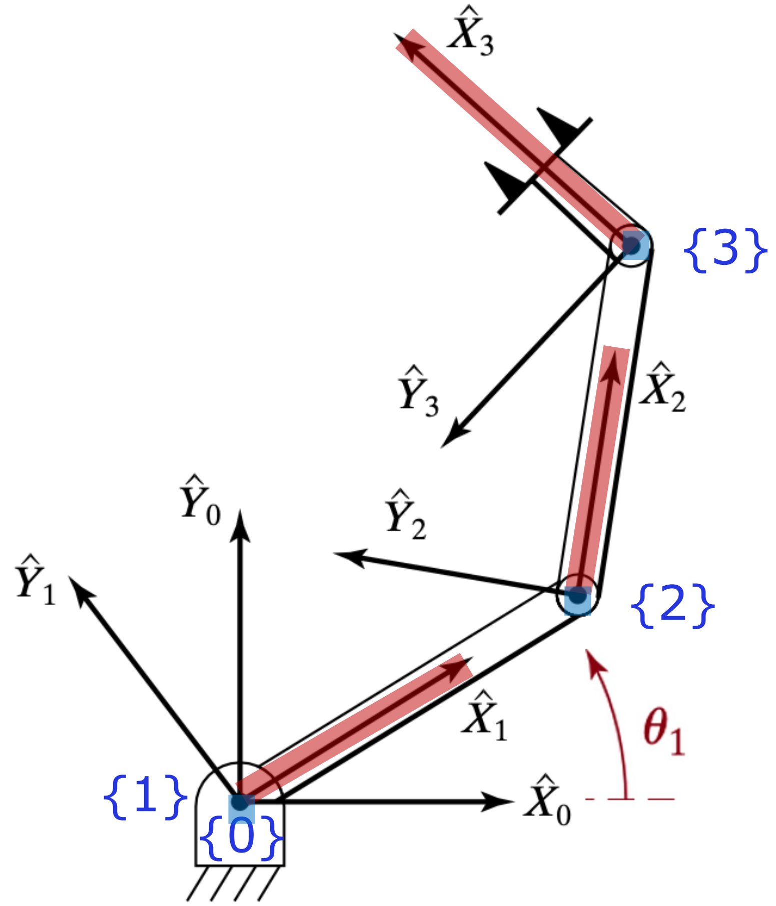
RRR manipulator - Example 2 - Attach frames

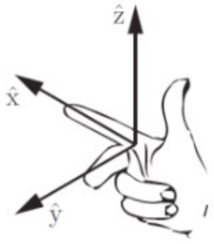
- Frame {0} is fixed to the base and aligns with frame {1} when the first joint variable θ_1 is zero
- All ***Z axes*** parallel pointing out to of the screen/paper, there are no link offsets and twists
  - All d_i are zero
  - All α_i are zero

d_i – **link offset**

α_{i-1} – **link twist**

- All joints are rotational, so when they are at zero degrees, all ***X axes*** must align





XYZ

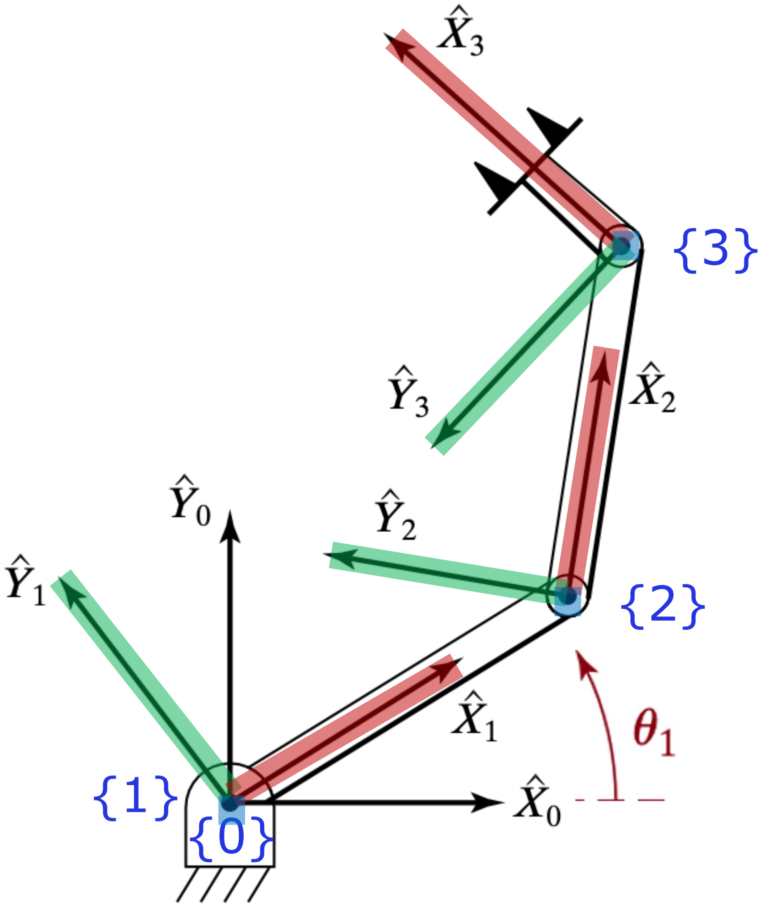
RRR manipulator - Example 2 - Attach frames

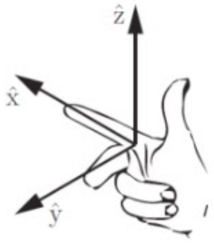
- Frame {0} is fixed to the base and aligns with frame {1} when the first joint variable θ_1 is zero
- All ***Z axes*** parallel pointing out to of the screen/paper, there are no link offsets and twists
  - All d_i are zero
  - All α_i are zero

d_i – **link offset**

α_{i-1} – **link twist**

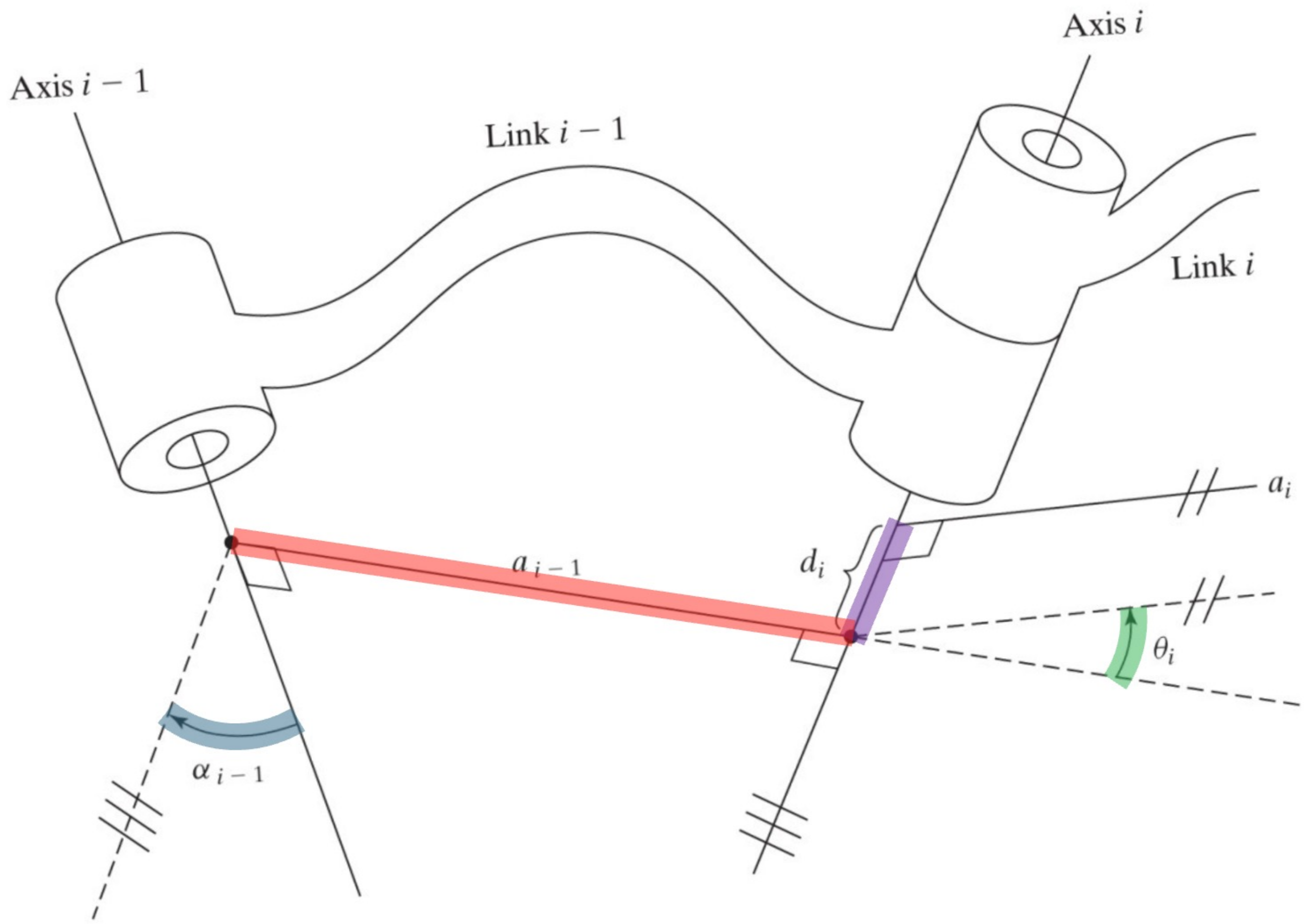
- All joints are rotational, so when they are at zero degrees, all ***X axes*** must align



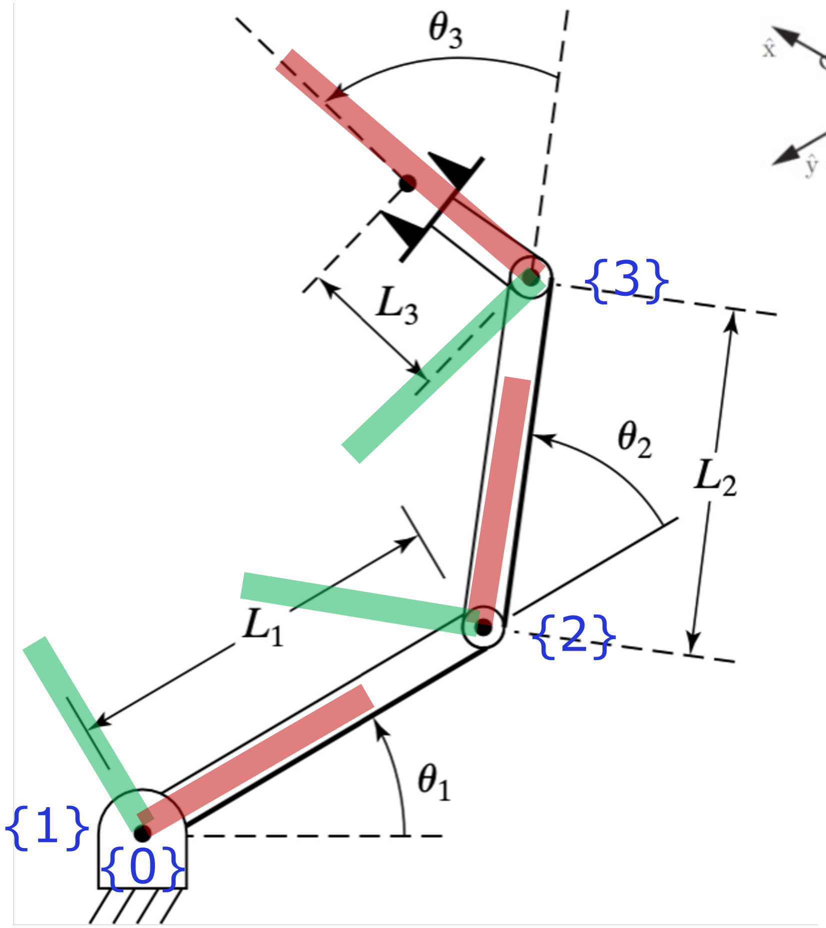


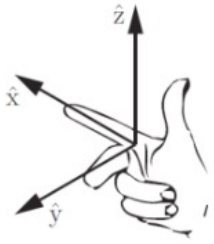
XYZ

RRR manipulator - Example 2 - DH parameters



RRR manipulator - Example 2 - <u>DH parameters</u>





XYZ

RRR manipulator - Example 2 - DH parameters

$\alpha_i = the\ angle\ from\ \hat{Z}_i\ to\ \hat{Z}_{i+1}\ measured\ about\ \hat{X}_i;$

$a_i = the\ distance\ from\ \hat{Z}_i\ to\ \hat{Z}_{i+1}\ measured\ along\ \hat{X}_i;$

$d_i = the\ distance\ from\ \hat{X}_{i-1}\ to\ \hat{X}_i\ measured\ along\ \hat{Z}_i;$

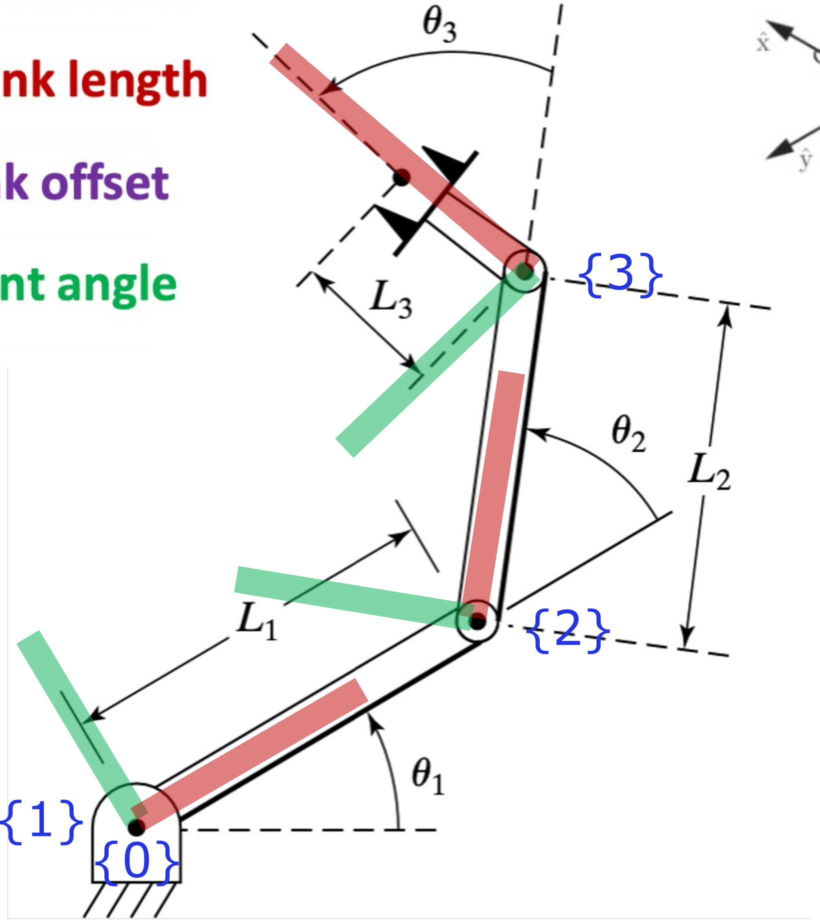
$\theta_i = the\ angle\ from\ \hat{X}_{i-1}\ to\ \hat{X}_i\ measured\ about\ \hat{Z}_i.$

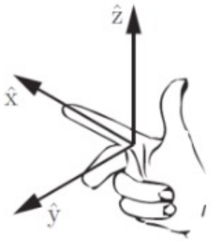
α_{i-1} – link twist

a_{i-1} – link length

d_i – link offset

θ_i – joint angle





XYZ

RRR manipulator - Example 2 - DH parameters

α_i = the angle from $\hat{Z}_i$ to $\hat{Z}_{i+1}$ measured about $\hat{X}_i$;

a_i = the distance from $\hat{Z}_i$ to $\hat{Z}_{i+1}$ measured along $\hat{X}_i$;

d_i = the distance from $\hat{X}_{i-1}$ to $\hat{X}_i$ measured along $\hat{Z}_i$;

θ_i = the angle from $\hat{X}_{i-1}$ to $\hat{X}_i$ measured about $\hat{Z}_i$.

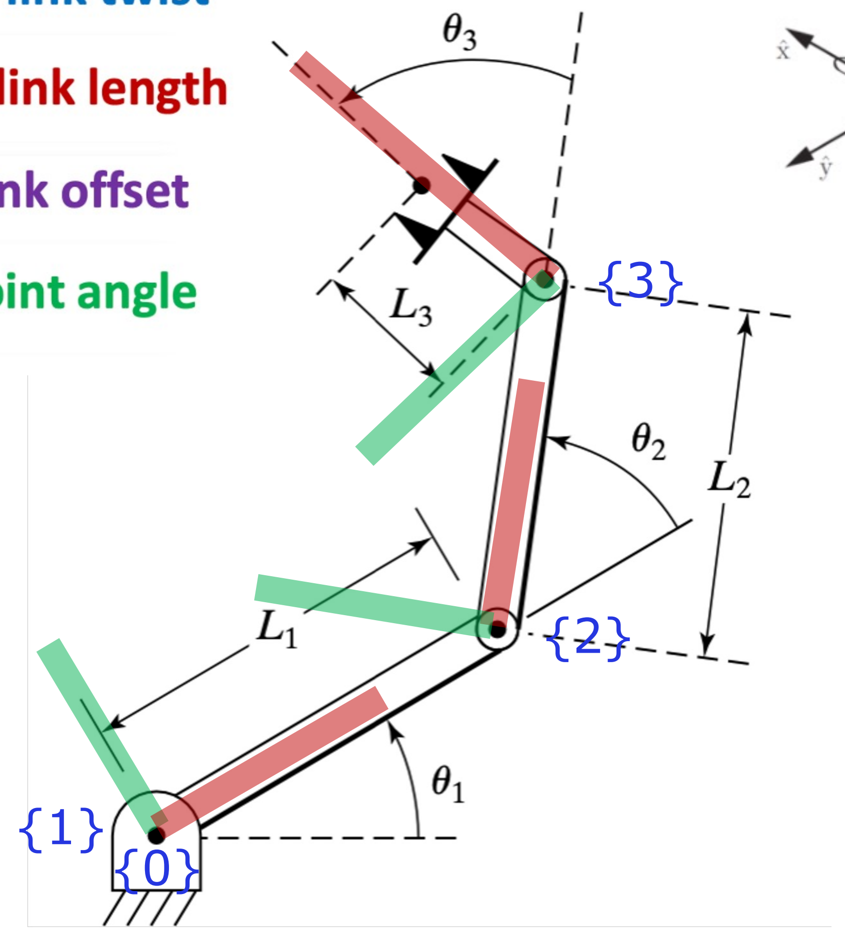
α_{i-1} – link twist

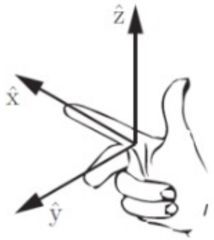
a_{i-1} – link length

d_i – link offset

θ_i – joint angle

i	α_{i-1}	a_{i-1}	d_i	θ_i





XYZ

RRR manipulator - Example 2 - DH parameters

$$\alpha_i = \text{the angle from } \hat{Z}_i \text{ to } \hat{Z}_{i+1} \text{ measured about } \hat{X}_i;$$
$$a_i = \text{the distance from } \hat{Z}_i \text{ to } \hat{Z}_{i+1} \text{ measured along } \hat{X}_i;$$
$$d_i = \text{the distance from } \hat{X}_{i-1} \text{ to } \hat{X}_i \text{ measured along } \hat{Z}_i;$$
$$\theta_i = \text{the angle from } \hat{X}_{i-1} \text{ to } \hat{X}_i \text{ measured about } \hat{Z}_i.$$

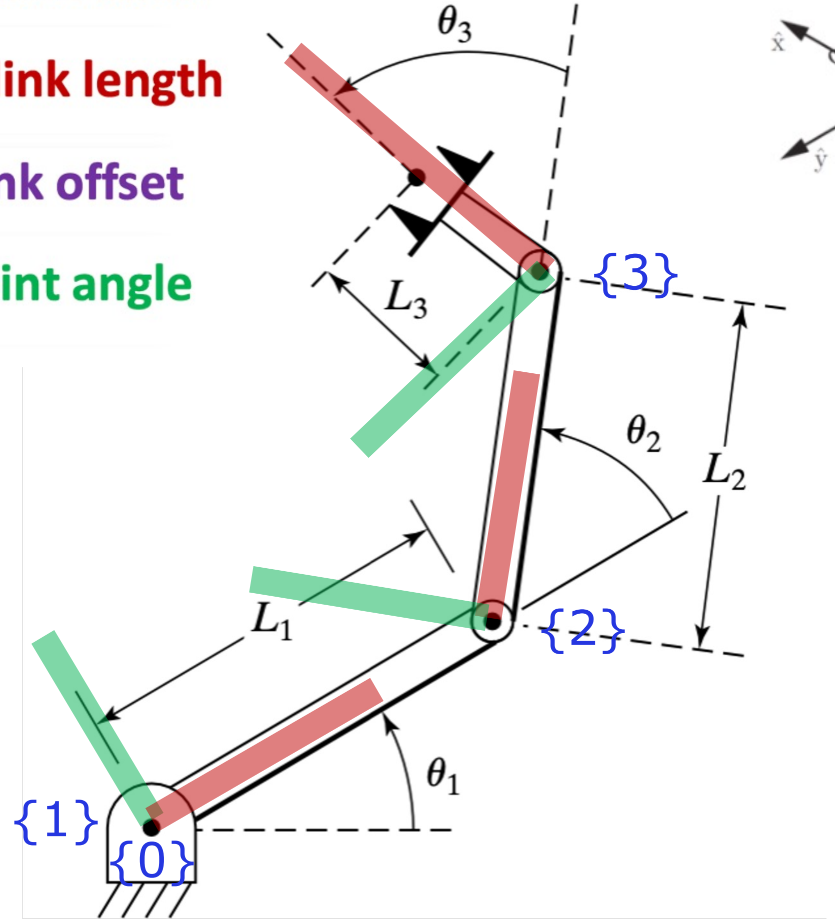
α_{i-1} – **link twist**

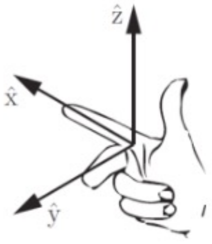
a_{i-1} – **link length**

d_i – **link offset**

θ_i – **joint angle**

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1





XYZ

RRR manipulator - Example 2 - DH parameters

α_i = the angle from $\hat{Z}_i$ to $\hat{Z}_{i+1}$ measured about $\hat{X}_i$;

a_i = the distance from $\hat{Z}_i$ to $\hat{Z}_{i+1}$ measured along $\hat{X}_i$;

d_i = the distance from $\hat{X}_{i-1}$ to $\hat{X}_i$ measured along $\hat{Z}_i$;

θ_i = the angle from $\hat{X}_{i-1}$ to $\hat{X}_i$ measured about $\hat{Z}_i$.

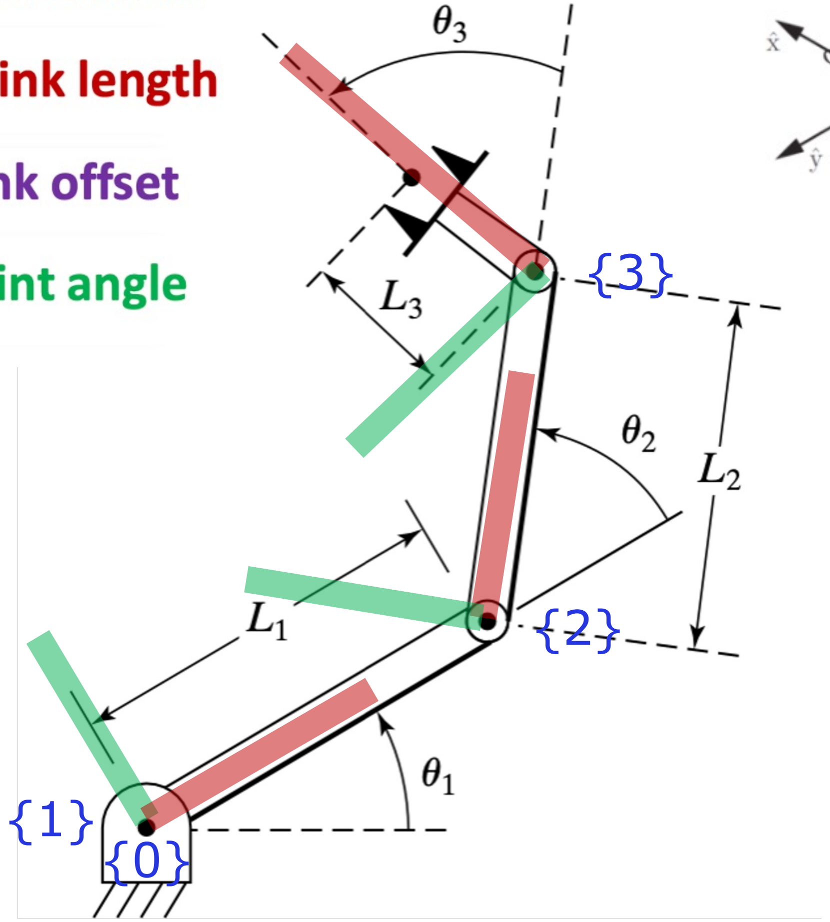
α_{i-1} – link twist

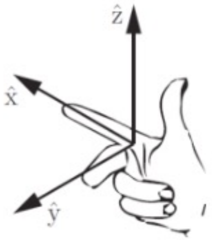
a_{i-1} – link length

d_i – link offset

θ_i – joint angle

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





RRR manipulator - Example 2 - DH parameters

$\alpha_i =$ *the angle from* $\hat{Z}_i$ *to* $\hat{Z}_{i+1}$ *measured about* $\hat{X}_i$;

$a_i =$ *the distance from* $\hat{Z}_i$ *to* $\hat{Z}_{i+1}$ *measured along* $\hat{X}_i$;

$d_i =$ *the distance from* $\hat{X}_{i-1}$ *to* $\hat{X}_i$ *measured along* $\hat{Z}_i$;

$\theta_i =$ *the angle from* $\hat{X}_{i-1}$ *to* $\hat{X}_i$ *measured about* $\hat{Z}_i$.

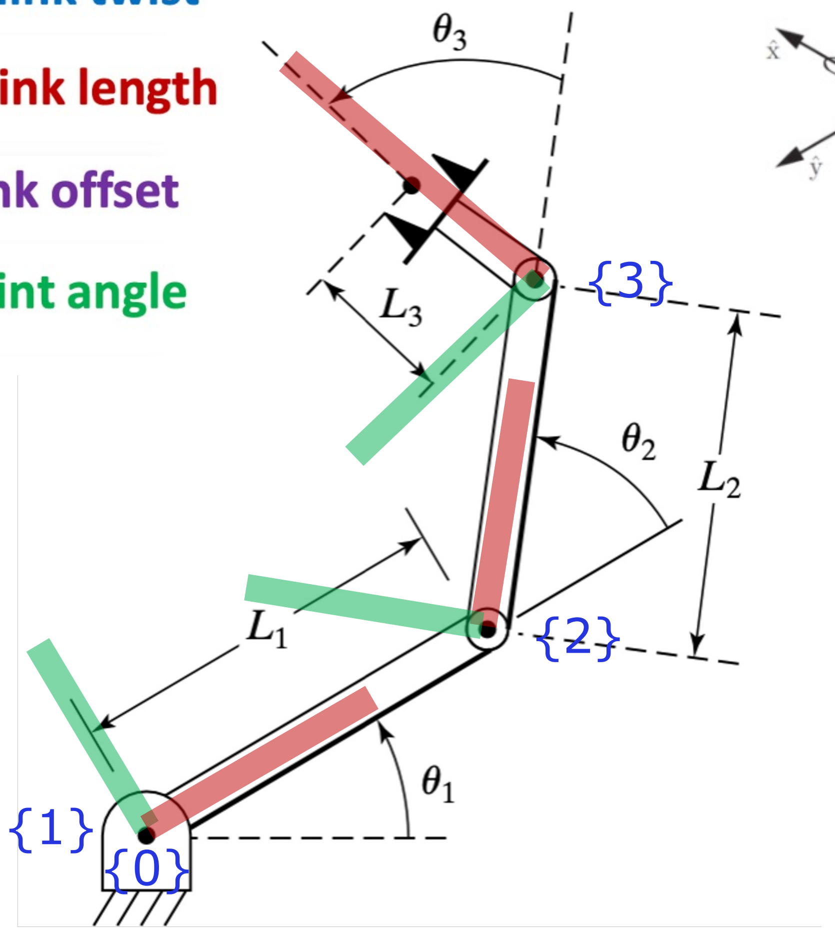
α_{i-1} – **link twist**

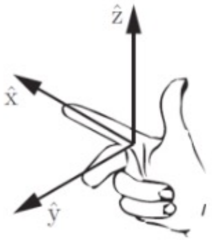
a_{i-1} – **link length**

d_i – **link offset**

θ_i – **joint angle**

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





XYZ

… end of Lecture 4