ROBOT MECHANICS, MODELLING AND SIMULATION

ROBOTICS 2023

OLE MADSEN
OM@M-TECH.AAU.DK

AALBORG UNIVERSITY
DENMARK

Lecture plan

No	Responsible	Contents
1	Ole Madsen	Introduction to: • the course; • robotics and robot terminology.
2	Ole Madsen	Spatial descriptions and transformation matrices
3 (FIB)	Ole Madsen + More	Practical exercise with the on-line programming (1.5 timer/gruppe).
4	Ole Madsen	Orientation
5	Ole Madsen	Forward Kinematics I
6	Ole Madsen	Forward Kinematics II (go though 6 DOF robot) – exercise, you go though your robot
7	Ole Madsen	Inverse kinematics I
8	Ole Madsen	Inverse kinematics II (go through 6DOF robot) – you start on your robot
9	Ole Madsen	Trajectory generation and control (joint)
10	Ole Madsen	Trajectory generation and control (cartesian)
11	Ole Madsen	Jacobian/Exam preparation

Basic Rotations

- Rotation about X

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

- Rotation about Y

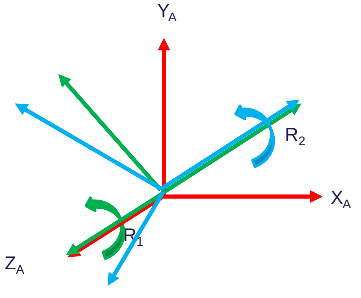
$$R_y(\theta) = \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix}$$

- Rotation about Z

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

Intrinsic rotations

Rotations relative to the moving coordinate system

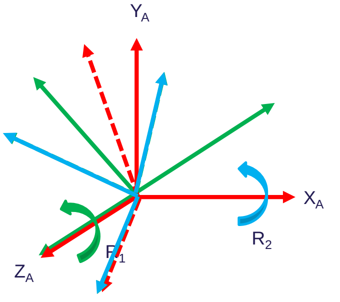


$${}_{Blue}^{A}R = {}_{green}^{A}R \cdot {}_{blue}^{green}R = R_1 \cdot R_2$$

Post multiplications

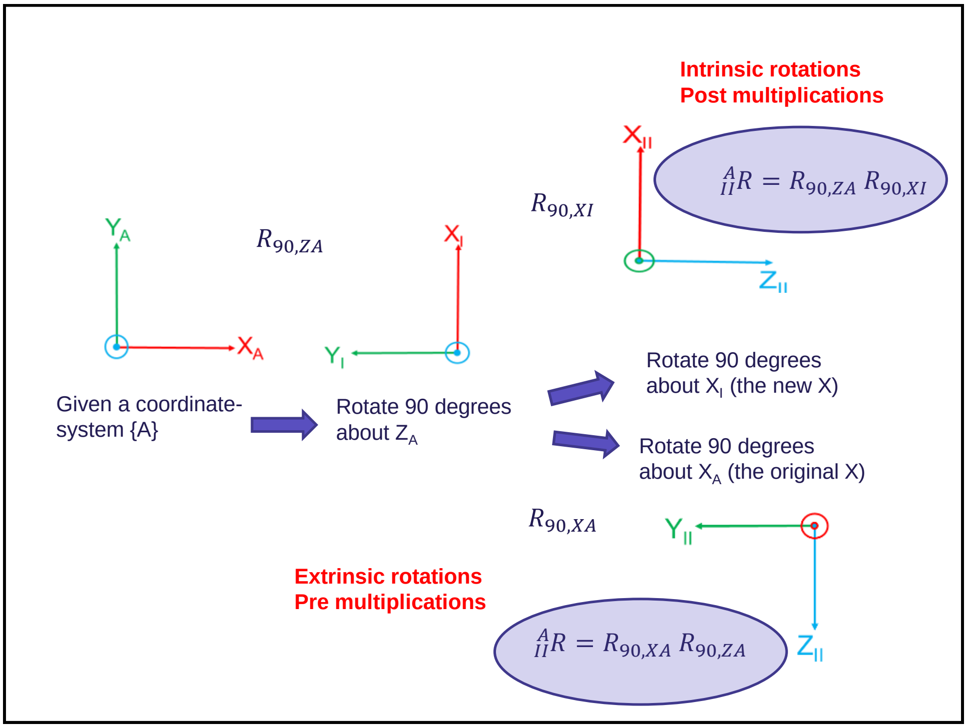
Extrinsic rotations

Rotations relative to fixed coordinate system



$${}_{Blue}^{A}R = {}_{red\ dot}^{A}R \cdot {}_{blue}^{red\ dot}R = R_2 \cdot R_1$$

Pre multiplications



Test exercise

- Start with a frame {B} coincident with a known reference frame {A}. Give the rotation matrix which accomplishes the following rotations of {B}:
  1. rotate about j_A (Y_A) by an angle 30 degrees
  2. rotate about i_A (X_A) by an angle 45 degrees

$$R_{45,X}R_{30,Y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c45 & -s45 \\ 0 & s45 & c45 \end{bmatrix} \begin{bmatrix} c30 & 0 & s30 \\ 0 & 1 & 0 \\ -s30 & 0 & c30 \end{bmatrix}$$

- A frame {B} is located as follows: initially coincident with a frame {A}. We rotate {B} about Z_B by θ degrees and then we rotate the resulting frame about X_B by ϕ degrees

$$R_{\theta,Z}R_{\phi,X} = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}$$

Hand-in solutions for this exercise on Moodle

FORWARD KINEMATICS

ROBOTICS 2023

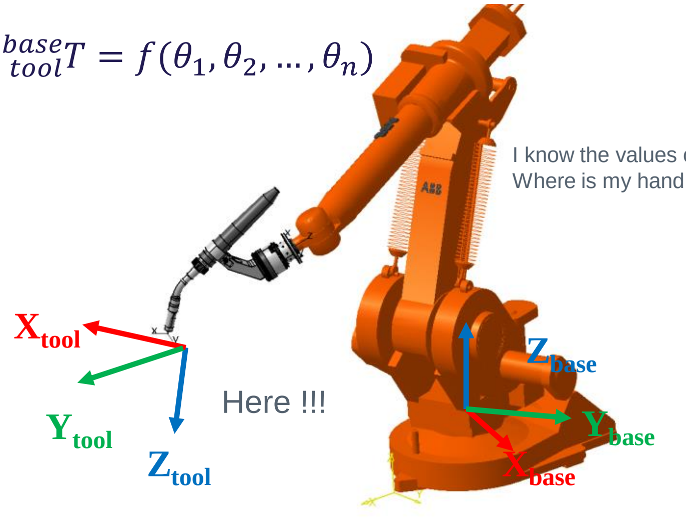
OLE MADSEN
OM@M-TECH.AAU.DK

AALBORG UNIVERSITY
DENMARK

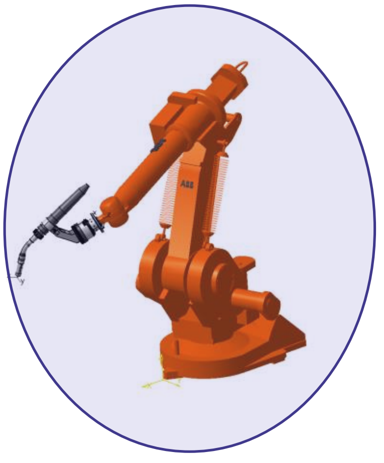
Forward kinematics (Direct kinematics)

$${}^{base}_{tool}T = f(\theta_1, \theta_2, \dots, \theta_n)$$

I know the values of my joint angles
Where is my hand ?

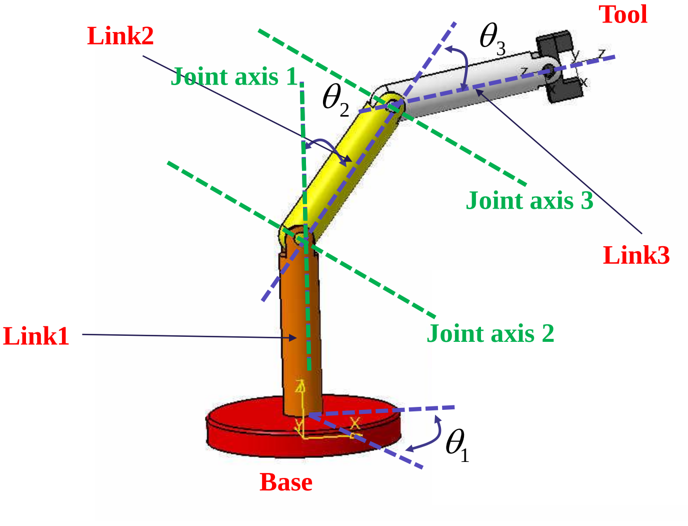


Serial and Parallel Manipulators



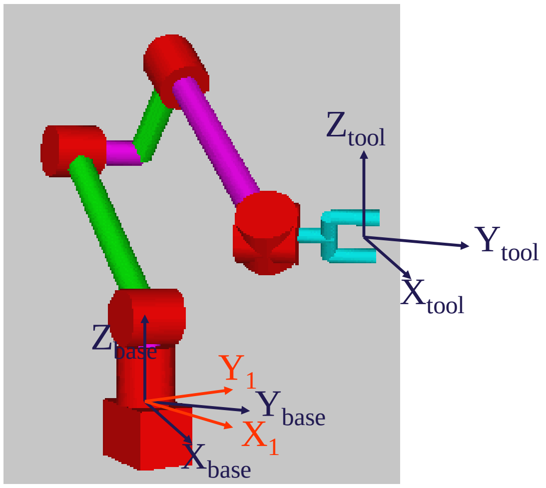
Focus here

Joints axes and links



Basic principl

$$ {}^{base}_{1}T = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

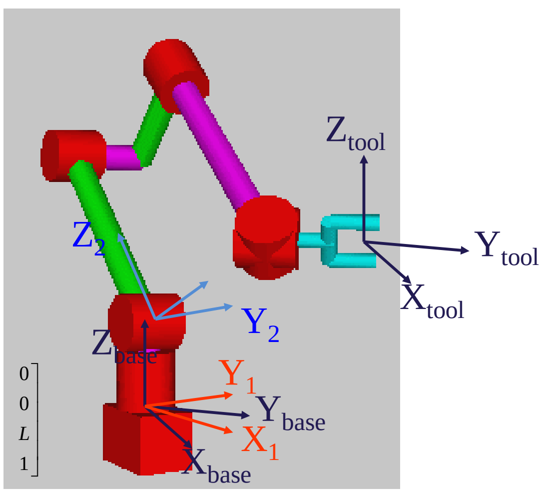


Basic principle

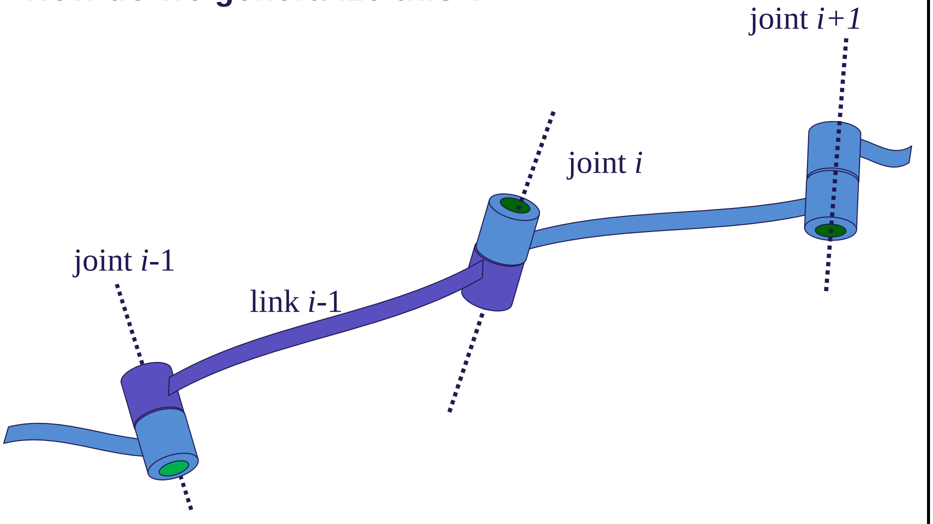
$$ {}^{base}_{1}T = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}_{2}T = \begin{bmatrix} C\theta_2 & 0 & -S\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ S\theta_2 & 0 & C\theta_2 & L \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{base}_{2}T = {}^{base}_{1}T \cdot {}^{1}_{2}T $$
$$ = \begin{bmatrix} C\theta_1 C\theta_2 & -S\theta_1 & -C\theta_1 S\theta_2 & 0 \\ S\theta_1 C\theta_2 & C\theta_1 & -S\theta_1 S\theta_2 & 0 \\ S\theta_2 & 0 & C\theta_2 & L \\ 0 & 0 & 0 & 1 \end{bmatrix} $$



How do we generalize this ?

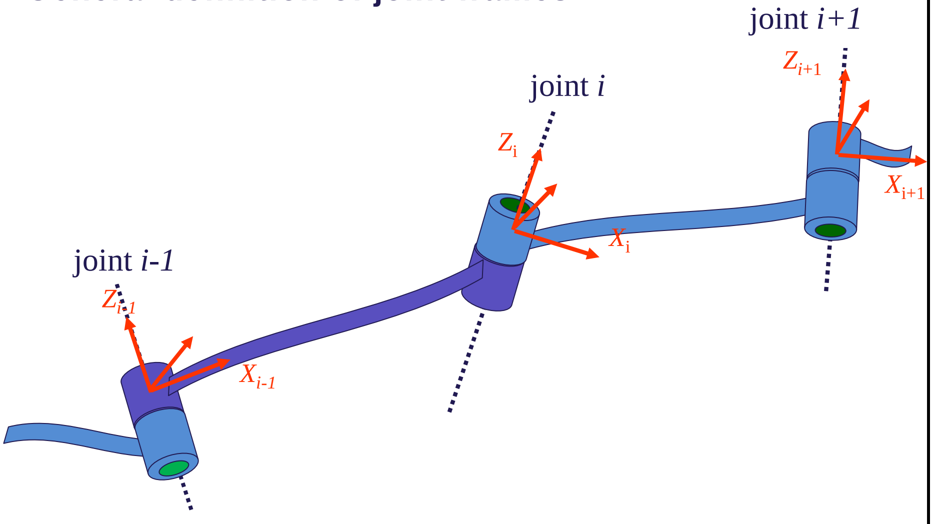


How do we generalize this ?

Answer: Use the Denavit-Hartenberg formalism

Three steps:
- Put out joint coordinate systems (using a certain procedure)
- Determine DH parameters:
    - Link parameters (link lenght, twist angle)
    - Link offset
    - Joint angle
- Insert into DH-equation

General definition of joint frames

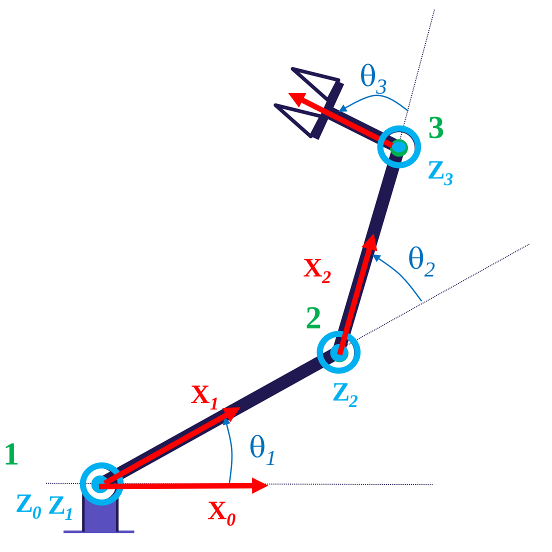


Affixing frames to joints

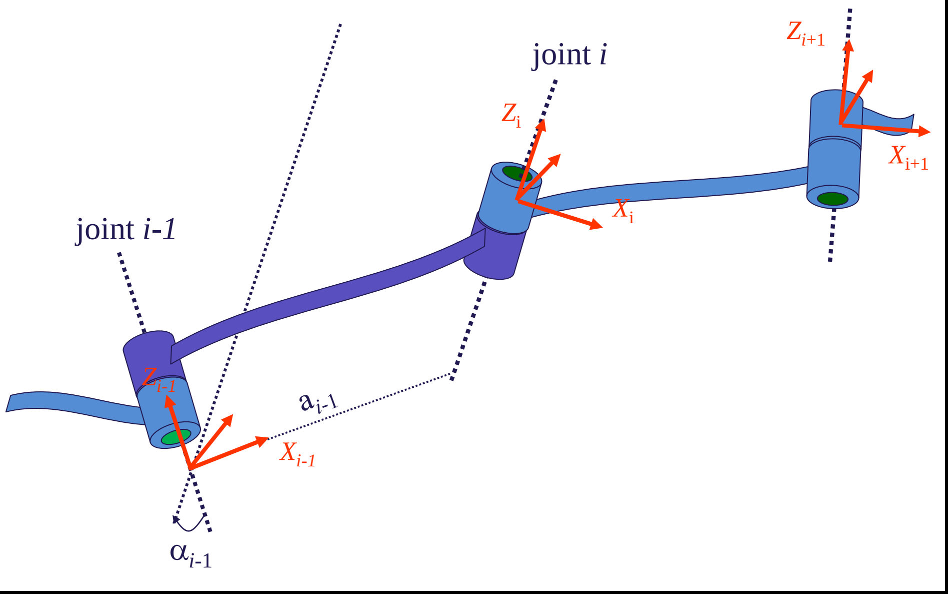
1. Identify the joint axes and draw lines along them.

For step 2 through 5 below consider two of these neighbouring lines (at axes i-1 and i):

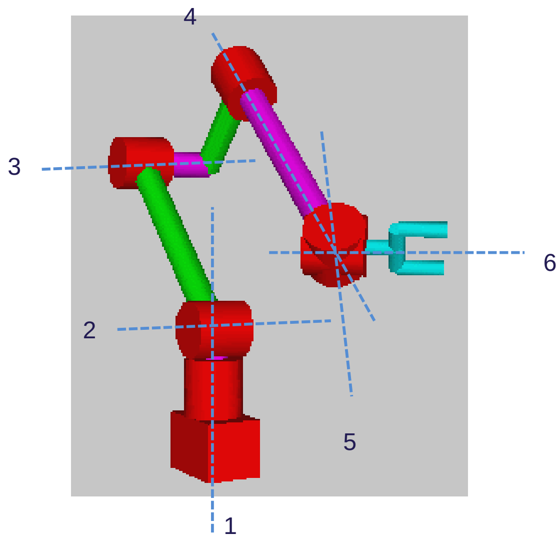
2. Identify the common perpendicular between them, or point of intersection. At the point of intersection, or at the point where the common perpendicular meets the i-1th axis, assign the link frame origin.
3. Assign Z_{i-1} pointing along the direction of axis i-1.
4. Assign X_{i-1} pointing along the common perpendicular, or if the axes intersect, assign X_{i-1} to be normal to the plane containing the two axes.
5. Assign Y_{i-1} to complete a right-hand coordinate system.
6. Assign {0} to match {1} when the joint variable is zero. For {N} choose an origin location and X_N direction freely, but generally so as to cause as many link parameters as possible to become zero.



Link Transformation: Link length and link twist

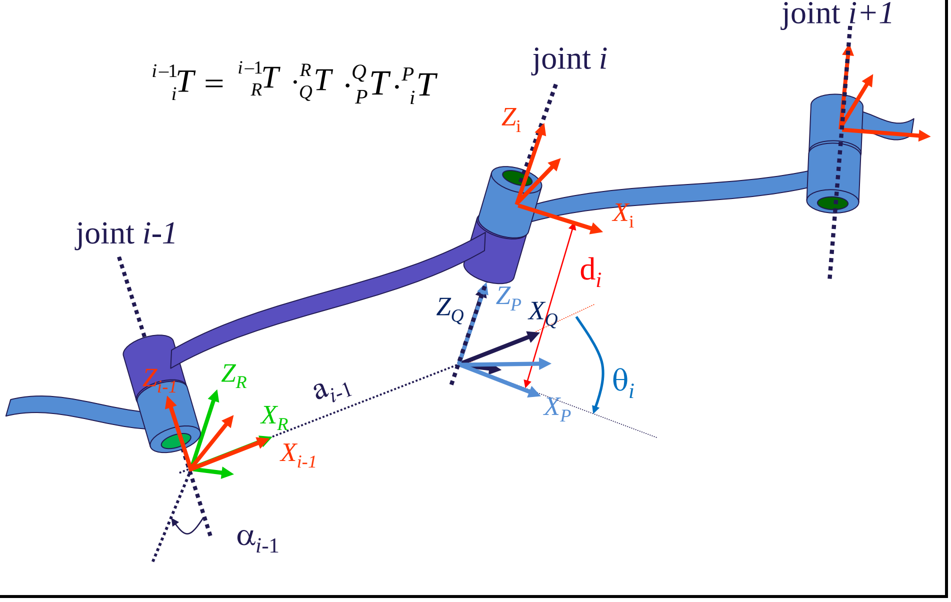


Link Transformation: Link length and link twist



Link Transformation: Link offset, joint angle

$${}^{i-1}_{i}T = {}^{i-1}_{R}T \cdot {}^{R}_{Q}T \cdot {}^{Q}_{P}T \cdot {}^{P}_{i}T$$



Link transformation

$$
{}^{i-1}_{i}T = {}^{i-1}_{R}T \cdot {}^{R}_{Q}T \cdot {}^{Q}_{P}T \cdot {}^{P}_{i}T
$$

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

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

Denavit Hartenberg Parameters

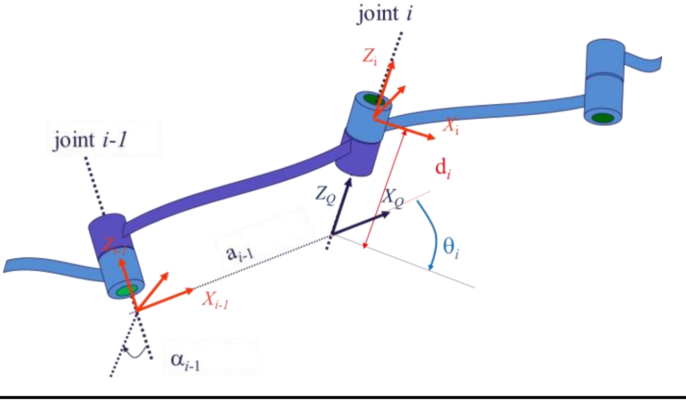
α_{i-1} = the angle between Z_{i-1} to Z_i measured about X_{i-1}

a_{i-1} = the distance from Z_{i-1} to Z_i measured along X_{i-1}

d_i = the distance from X_{i-1} to X_i measured along Z_i

θ_i = the angle between X_{i-1} to X_i measured about Z_i

i	α_{i-1}	a_{i-1}	d_i	θ_i
1				
2				
3				



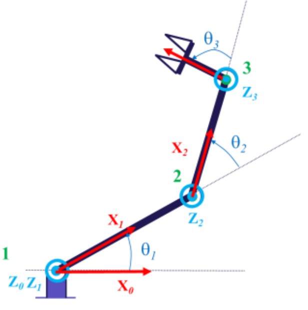
Denavit Hartenberg Parameters

α_{i-1} = the angle between Z_{i-1} to Z_i measured about X_{i-1}

a_{i-1} = the distance from Z_{i-1} to Z_i measured along X_{i-1}

d_i = the distance from X_{i-1} to X_i measured along Z_i

θ_i = the angle between X_{i-1} to X_i measured about Z_i



i	α_i	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L1	0	θ_2
3	0	L2	0	θ_3

Link transformation

$$ {}^{i-1}_{i}T = {}^{i-1}_{R}T \cdot {}^{R}_{Q}T \cdot {}^{Q}_{P}T \cdot {}^{P}_{i}T $$

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

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

$$ {}^{0}_{N}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T \ldots\ldots {}^{N-1}_{N}T $$

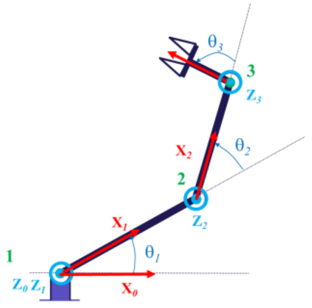
Denavit Hartenberg Parameters

α_{i-1} = the angle between Z_{i-1} to Z_i measured about X_{i-1}

a_{i-1} = the distance from Z_{i-1} to Z_i measured along X_{i-1}

d_i = the distance from X_{i-1} to X_i measured along Z_i

θ_i = the angle between X_{i-1} to X_i measured about Z_i



i	α_i	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L1	0	θ_2
3	0	L2	0	θ_3

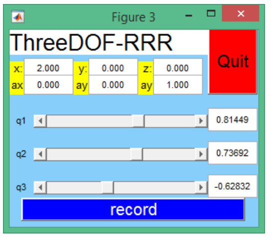
$$ {}^0_3T = {}^0_1T\,{}^1_2T\,{}^2_3T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & L1 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & L2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

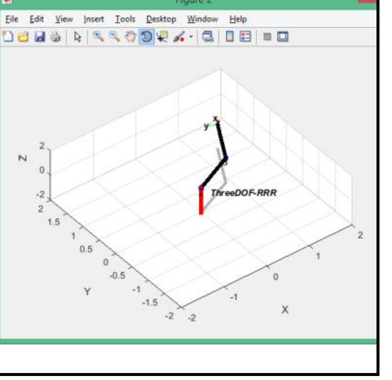
$$ = \begin{bmatrix} c(\theta_1+\theta_2+\theta_3) & -s(\theta_1+\theta_2+\theta_3) & 0 & L2\,c(\theta_1+\theta_2) + L2\,c\theta_1 \\ s(\theta_1+\theta_2+\theta_3) & c(\theta_1+\theta_2+\theta_3) & 0 & L2\,s(\theta_1+\theta_2) + L2\,s\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

In robotic toolbox:

```
L1 = 1
L2 = 1

%          alpha    a      d    modified D&H
%                              Because we use Craig
L(1) = Link('alpha', 0,'a',  0,'d', 0,'modified')    %rotational
L(2) = Link('alpha', 0,'a', L1,'d', 0,'modified')    %rotational
L(3) = Link('alpha', 0,'a', L2,'d', 0,'modified')    %rotational

threeDOF=SerialLink(L, 'name', 'ThreeDOF-RRR');

%Lets test the robot
threeDOF.teach():
```

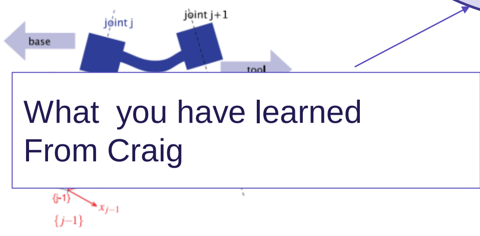
i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3



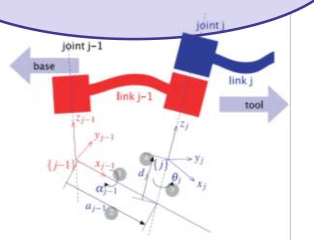


Standard DH

Modified DH



What you have learned From Craig

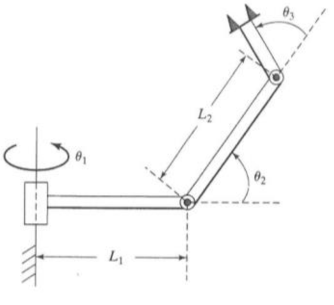


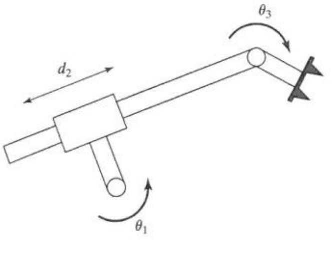

```
% ABB IBR140, DH parameters (standard)
% theta_j, d_j, a_j, alpha_j, joint_type, D&H
L(1) = Link([0      352  70    -pi/2   0 ]);
L(2) = Link([-pi/2  0    360   0       0 ]);
L(3) = Link([pi     0    0     pi/2    0 ]);
L(4) = Link([0      380  0     -pi/2   0 ]);
L(5) = Link([0      0    0     pi/2    0 ]);
L(6) = Link([-pi/2  65   0     pi/2    0 ]);

abb=SerialLink(L, 'name', 'ABB IBR 140');
```

```
% ABB IBR140, DH parameters (modified)
% theta_j, d_j, a_j-1, alpha_j-1, joint_type, D&H
Lm(1) = Link([0      352  0     0     0], 'modified');
Lm(2) = Link([-pi/2  0    70    pi/2  0], 'modified');
Lm(3) = Link([pi     0    360   0     0], 'modified');
Lm(4) = Link([0      380  0     pi/2  0], 'modified');
Lm(5) = Link([0      0    0     -pi/2 0], 'modified');
Lm(6) = Link([-pi/2  65   0     pi/2  0], 'modified');

abb=SerialLink(Lm, 'name', 'ABB IBR 140');
```

Exercises I

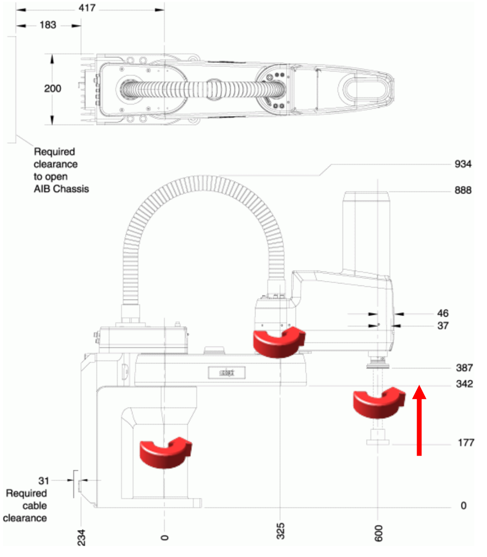
1. For the two robots in the figures below, derive Denavit-Hartenberg parameters, generate the forward kinematics and test with MatLab.





Exercises II

- For the robot in the figure, derive Denavit-Hartenberg parameters and implement in Matlab.



AALBORG UNIVERSITY
DENMARK