Aalto University
School of Electrical
Engineering

Robot arm forward kinematics

ELEC-C1320 Robotics

Pekka Forsman

Topics

- Describing a robot arm
- Denavit-Hartenberg robot link parameters
    - Standard Denavit-Hartenberg parameters
    - Modified Denavit-Hartenberg parameters
- Forward kinematics of a serial link manipulator, examples
- Example problems

Describing a robot arm

Kinematics is the branch of mechanics that studies the motion of a body, or a system of bodies, without consideration given to its mass or the forces acting on it (i.e. dynamic effects)

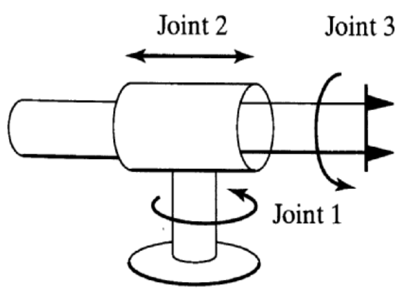
A serial-link manipulator comprises a set of bodies, called links, in a chain and connected by joints.

Each joint has one Degree of Freedom (DoF), either translational (sliding/telescopic/prismatic) joint or rotational (revolute) joint

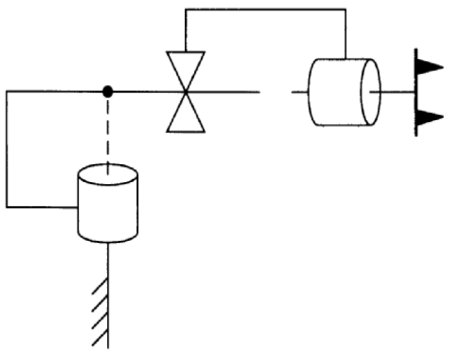
One end of the chain, the base, is generally fixed (to stationary or mobile platform) and the other end is free to move in space and holds the tool or end-effector

Motion of a joint changes the relative angle or position of its neighboring links. The joint structure of a robot can be described by a string such as "RRRRRR" for the Puma (where R stands for Rotational joint). The placement of the character in the string indicates the location of the joint in the kinematic chain of the robot (starting from the base on the left of the string)

A schematic illustration of a "RPR" robot is shown in the figure (P in the string indicates that the second joint is Prismatic)



(a)

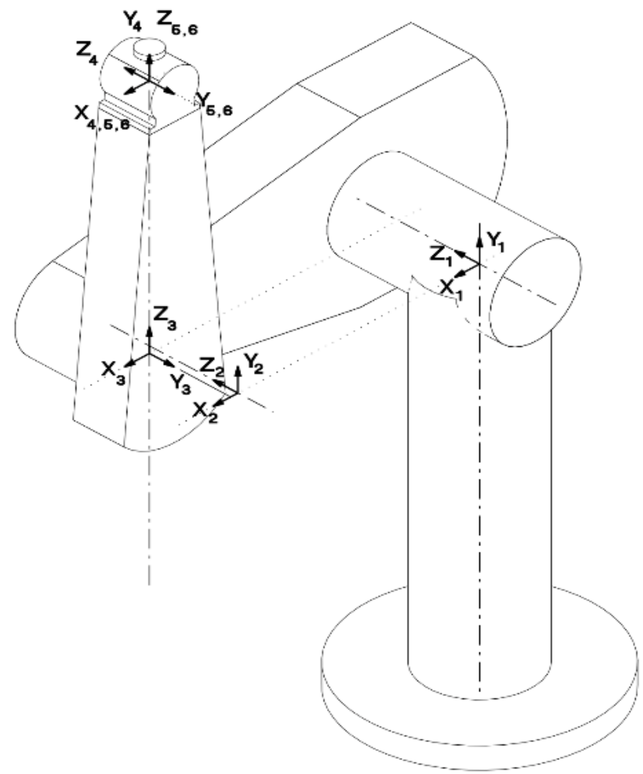


(b)

Denavit-Hartenberg robot link parameters

A systematic way of describing the geometry of a serial chain of links and joints was proposed by Denavit and Hartenberg in 1955 and is known today as Denavit-Hartenberg notation

For a manipulator with N joints numbered from 1 to N, there are $N + 1$ links, numbered from 0 to N. Link 0 is the base of the manipulator and link N carries the end-effector or tool



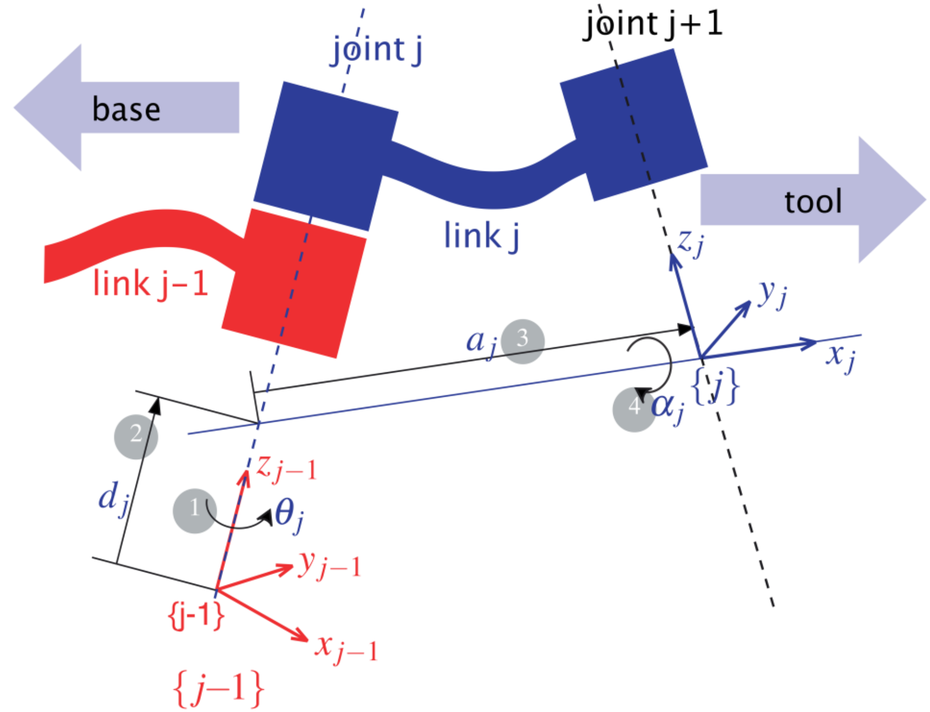
Joint j connects link $j - 1$ to link j and therefore joint j moves link j

In the figure to the right, Puma 560 robot link coordinate frames 1 to 6 are shown in the zero-angle pose of the mechanism (*according to the standard Denavit-Hartenberg convention*)

Definition of **standard** Denavit and Hartenberg link parameters

A **link** is specified by two parameters, its ***length*** a_j and its ***twist*** α_j

Joints are also described by two parameters. The ***link offset*** d_j is the distance from one link coordinate frame to the next along the axis of the joint. The ***joint angle*** θ_j *corresponds to* the rotation of link *j* with respect link *j-1* about the joint (motor) axis *j* which is paralled to frame axis z_{j-1}



Note*: the link frames do not have to be inside the physical structures of links as is demonstrated by the figure*

Fixing the coordinate frames to the robot kinematic chain
(rules for the **standard** DH-convention):

1. The coordinate frame $\{j\}$ is attached to the far (distal) end of link j. The axis of joint $j+1$ is aligned with the z_j-axis (*i.e. joint* $j+1$ *rotates/moves around/along* z_j*-axis*)

2. Find the line (*corresponding to the shortest distance between the lines drawn along the neighboring joint axes j and j+1*) to connect the joint axes. Or if the joint axes intersect, find the intersection point. Place the origin of the link frame to this point

3. Set the x_j-axis to point along the line connecting the joint axes. Or if the joint axes intersect, set the x_j-axis to be the normal of the plane defined by the two joint axes

4. Set the y_j axis according to the right hand rule

The link and joint parameters known as Denavit-Hartenberg parameters, are summarized in the Table (note: indices correspond to Standard DH-notation):

Joint angle	θ_j	the angle between the x_{j-1} and x_j axes about the z_{j-1} axis	revolute joint variable
Link offset	d_j	the distance from the origin of frame $j-1$ to the x_j axis along the z_{j-1} axis	prismatic joint variable
Link length	a_j	the distance between the z_{j-1} and z_j axes along the x_j axis; for intersecting axes is parallel to $\hat{\boldsymbol{z}}_{j-1} \times \hat{\boldsymbol{z}}_j$	constant
Link twist	α_j	the angle from the z_{j-1} axis to the z_j axis about the x_j axis	constant
Joint type	σ_j	$\sigma = R$ for a revolute joint, $\sigma = P$ for a prismatic joint	constant

Note:

- Often in practice x_{j-1} is aligned with link *j-1* and x_j is aligned with link *j* etc.
- For intersecting joint axes the *a*-parameter gets value 0 (*because distance btw. Intersecting axes is zero*)

The parameters α_j and a_j are always constant. For a revolute joint θ_j is the joint variable and d_j is constant, while for a prismatic/telescopic joint d_j is variable and θ_j is constant

The final joint, joint N connects link $N-1$ to link N. Usually we will also need a (**constant**) "tool transformation matrix", ${}^{N}\boldsymbol{T}_{T}$, that describes relative position and orientation of the robot tool frame {T} with respect to the last link frame of the mechanism {N}

The transformation from link coordinate frame {j − 1} to frame {j} is defined in terms of elementary rotations and translations as

$$ {}^{j-1}\xi_j\left(\theta_j, d_j, a_j, \alpha_j\right) = \mathscr{R}_z\left(\theta_j\right) \oplus \mathscr{T}_z\left(d_j\right) \oplus \mathscr{T}_x\left(a_j\right) \oplus \mathscr{R}_x\left(\alpha_j\right) \qquad (7.2)$$

which can be expanded into a symbolic form link transformation matrix ${}^{j-1}\boldsymbol{T}_j$ for the **standard** DH-parameter method:

$$ {}^{j-1}\boldsymbol{T}_j = {}^{j-1}A_j = \begin{pmatrix} \cos\theta_j & -\sin\theta_j\cos\alpha_j & \sin\theta_j\sin\alpha_j & a_j\cos\theta_j \\ \sin\theta_j & \cos\theta_j\cos\alpha_j & -\cos\theta_j\sin\alpha_j & a_j\sin\theta_i \\ 0 & \sin\alpha_j & \cos\alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (7.3)$$

In some of the formulations of Corke's text book we use generalized joint coordinates

$$\text{if } \sigma_j = \begin{cases} R\colon & \theta_j \leftarrow q_j \\ P\colon & d_j \leftarrow q_j \end{cases}$$

For an *N*-axis robot the generalized joint coordinates $q \in C$ where $C \subset R^N$ is called the joint space or configuration space

Within the Toolbox a robot revolute joint and link can be created by

```
>> L = Revolute('a', 1)
L =
Revolute(std): theta=q, d=0, a=1, alpha=0, offset=0
```

which is a revolute-joint object of type Revolute which is a subclass of the generic Link object. The displayed value of the object shows the kinematic parameters (most of which have defaulted to zero), the joint type and that standard Denavit-Hartenberg convention is used (the tag std)

Definition of **modified** Denavit and Hartenberg link parameters

In some text books (e.g. Craig, Introduction to robotics) **another grouping** of the elementary translations and rotations is used to define the *link matrix*. In Peter Corke's book this definition is called the **modified** Denavit and Hartenberg link parameters

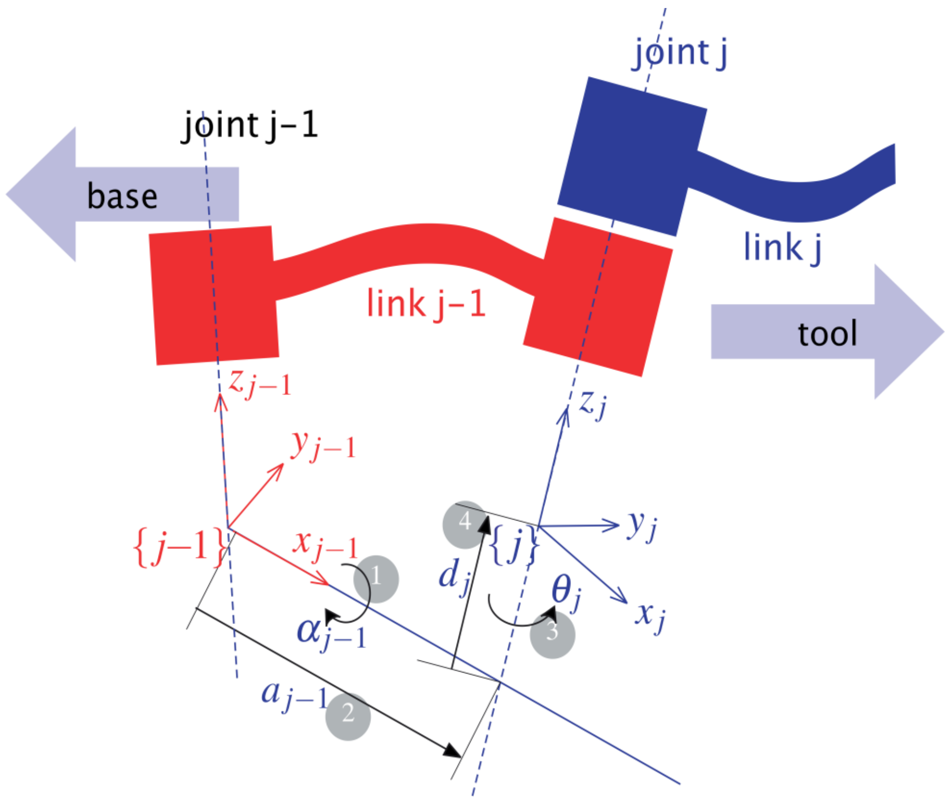
$$ {}^{j-1}\xi_j\left(\alpha_{j-1}, a_{j-1}, d_j, \theta_j\right) = \mathscr{R}_x\left(\alpha_{j-1}\right) \oplus \mathscr{T}_x\left(a_{j-1}\right) \oplus \mathscr{T}_z\left(d_j\right) \oplus \mathscr{R}_z\left(\theta_j\right) \qquad (7.8) $$

which can be expanded to yield the symbolic form link transformation matrix ${}^{j-1}\boldsymbol{T}_j$ for the **modified** DH-parameter method: (source: Craig, 2005)

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

In the **modified** Denavit-Hartenberg notation the link coordinate frames are attached to the near (proximal), rather than the far end of each link

Note that $\boldsymbol{a_j}$ **corresponds always to the length of link** $\boldsymbol{j}$**,** but it is the displacement between the origins of frame $\{j\}$ and frame $\{j + 1\}$ in the modified Denavit-Hartenberg convention, and frame $\{j - 1\}$ and frame $\{j\}$ in the standard one.



The two approaches can be compared by considering the link transforms as a sequence of elementary rotations and translations represented by Eq. 7.2 and Eq. 7.8.

Consider the transformation chain for Standard Denavit-Hartenberg notation:

$$\underbrace{\mathcal{R}_z(\theta_1)\oplus\mathcal{T}_z(d_1)\oplus\mathcal{T}_x(a_1)\oplus\mathcal{R}_x(\alpha_1)}_{\mathrm{DH}_1}\oplus\underbrace{\mathcal{R}_z(\theta_2)\oplus\mathcal{T}_z(d_2)\oplus\mathcal{T}_x(a_2)\oplus\mathcal{R}_x(\alpha_2)}_{\mathrm{DH2}}\cdots$$

Elementary transformations to fix link frame 1 And to get $^0\boldsymbol{T}_1$

Elementary transformations to fix link frame 2 And to get $^1\boldsymbol{T}_2$

Sidenote: **concecutive** translation along and rotation about the **same axis** is **commutative**

which we can **regroup** for Modified Denavit Hartenberg notation as:

$$\underbrace{\mathcal{R}_z(\theta_1)\oplus\mathcal{T}_z(d_1)}_{\text{base}}\oplus\underbrace{\mathcal{T}_x(a_1)\mathcal{R}_x(\alpha_1)\oplus\mathcal{R}_z(\theta_2)\oplus\mathcal{T}_z(d_2)}_{\mathrm{MDH}_1}\oplus\underbrace{\mathcal{T}_x(a_2)\oplus\mathcal{R}_x(\alpha_2)}_{\mathrm{MDH}_2}\cdots$$

Elementary transformations to fix link frame 1 And to get $^0\boldsymbol{T}_1$

Elementary transformations to fix link frame 2 And to get $^1\boldsymbol{T}_2$

where the terms marked as MDH_j (Modified Denavit Hartenberg) correspond to Eq. 7.8 and terms marked DH_j (Standard Denavit Hartenberg) correspond to Eq. 7.2.

If you intend to build a Toolbox robot model from a table of kinematic parameters provided in a paper it is really important to know which convention the author of the table used. Too often this important fact is not mentioned. An important clue lies in the column headings. If they all have the same subscript, i.e. θ_j, d_j, a_j and α_j then this is standard Denavit-Hartenberg notation. If half the subscripts are different, i.e. θ_j, d_j, a_{j-1} and α_{j-1} then you are dealing with modified Denavit-Hartenberg notation. In short, you must know which kinematic convention your Denavit-Hartenberg parameters conform to.

You can also help the cause when publishing by stating clearly which kinematic convention is used for your parameters.

The Toolbox can handle either form, it only needs to be specified, and this is achieved using variant classes when creating a link object

```
>> L1 = RevoluteMDH('d', 1)
L1 =
Revolute(mod): theta=q, d=1, a=0, alpha=0, offset=0
```

Forward kinematics of a serial link manipulator, examples

The **forward kinematics** is a function of the joint coordinates and is simply the composition of the relative pose due to each link.

To complete the chain of transformations from the robot base, *W-frame*, to the tip of its tool, *E-frame*, we add two extra transforms (**base** and **tool** transformations) to the equation

$$ {}^{W}\xi_{E} = \underbrace{{}^{W}\xi_{0}}_{\xi_B} \oplus {}^{0}\xi_{1} \oplus {}^{1}\xi_{2} \cdots {}^{N-1}\xi_{N} \oplus \underbrace{{}^{N}\xi_{E}}_{\xi_T} $$

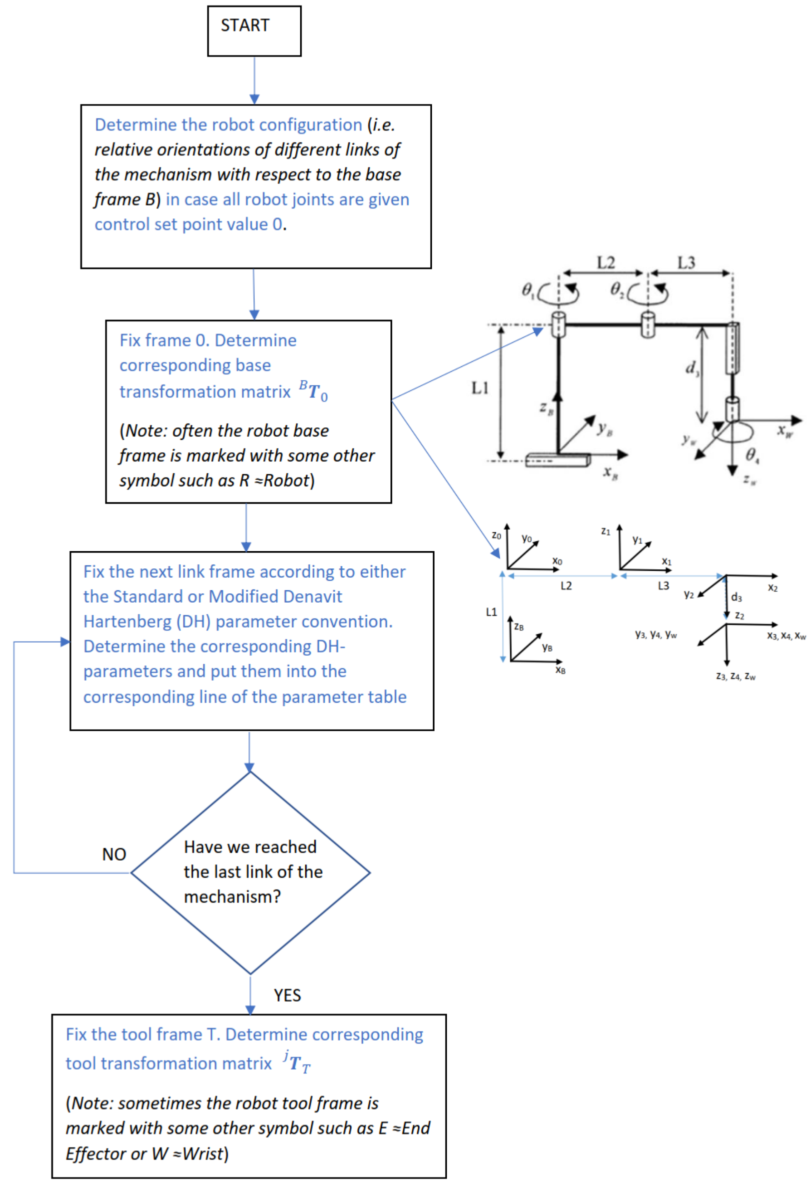
In this course, instead of *W*-frame, we use for the fixed robot frame symbol *B* in order to refer to the robot Base. And for the robot tool frame we will use symbol *T instead of E (End effector) shown in the equation above.*

The equation from above can be rewritten by using our mathematical tool i.e. the homogenous transformation matrix as:

$$ {}^{\mathrm{B}}\boldsymbol{T}_{\mathrm{T}} = {}^{\mathrm{B}}\boldsymbol{T}_{0}\, {}^{0}\boldsymbol{T}_{1}\, {}^{1}\boldsymbol{T}_{2} \cdots {}^{\mathrm{N-1}}\boldsymbol{T}_{\mathrm{N}}\, {}^{\mathrm{N}}\boldsymbol{T}_{\mathrm{T}} $$

<u>A block diagram to illustrate the steps for the determination of the forward kinematics model of a serial link robot mechanism is described with the drawing to the right:</u>

A description of the different steps (*for example what things to consider when fixing, i.e. determining the position and orientation, of the frame 0*) are summarized in the file “a summary of the DH parameter method.pdf” which can be found in the same MyCourses folder as these lecture slides.



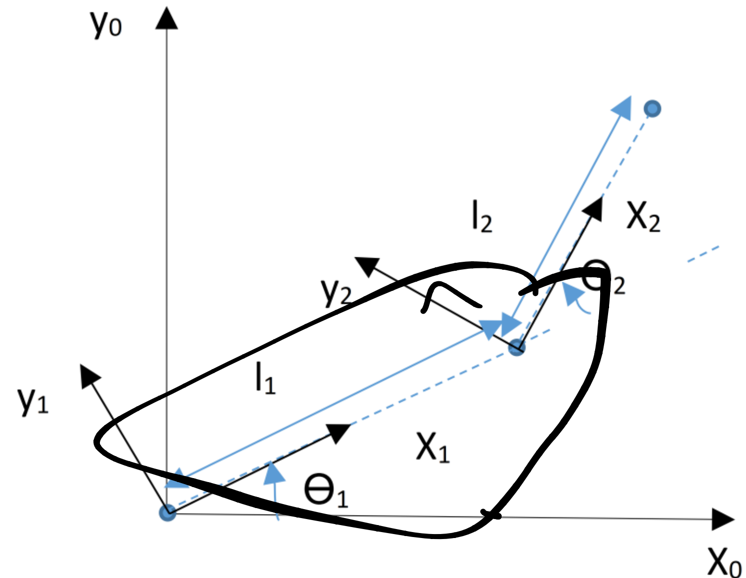
The forward kinematic solution can be computed for *any* serial-link manipulator irrespective of the number of joints or the types of joints

The pose of the end-effector $\xi_E \sim \boldsymbol{T}_E \in SE(3)$ has six degrees of freedom – three in translation and three in rotation. Therefore, robot manipulators commonly have six joints or degrees of freedom to allow them to reach an arbitrary end-effector pose within the reachable operation envelope of the mechanism

Example: A 2-Link Robot

(moving on the xy-plane)

Forward kinematics of the two-axes robot, shown in the figure to the right, according to the **modified** DH-parameter convention.



Link	α_{i-1}	a_{i-1}	d_i	Θ_i	σ_i
1	0	0	0	Θ_1	R
2	0	l_1	0	Θ_2	R

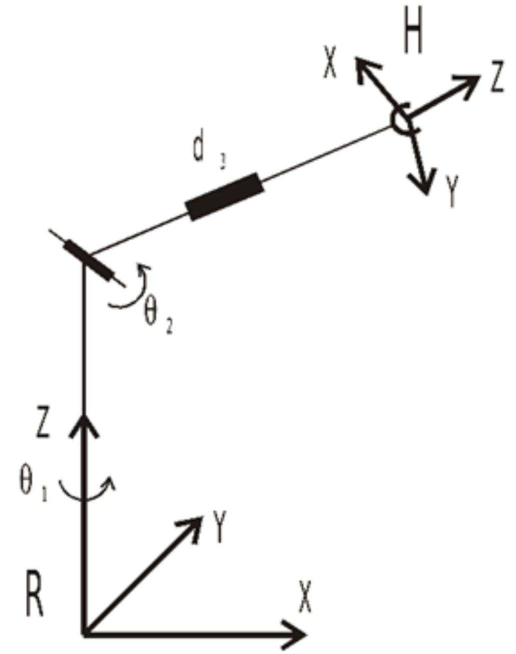
Now, for the modified DH-parameter representation, we have to consider the displacement from the origin of the 2nd link frame up to the far end of the "elbow" link by a separate <u>tool transformation matrix</u>:

tool: $$\begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example: A three-axis **RRP**-manipulator

Make the forward kinematic model for the 3-degree of freedom manipulator shown in the figure to the right. The first rotational joint, Θ_1, rotates the mechanism around a vertical axis. The second, "shoulder" joint, Θ_2, rotates the upper arm around a horizontal axis. The third joint is prismatic, d_3, for adjusting the length of the upper arm. The forward kinematic model should describe the Hand frame position and orientation w.r.t the Robot frame, ${}^{R}\mathbf{T}_{H}$. Apply here the **modified** DH-parameter representation (pay attention on the placement of the coordinate frames on each link). As an answer to the problem, give a drawing showing the link frames for each of the three links. Also, give the DH-parameters in a table.

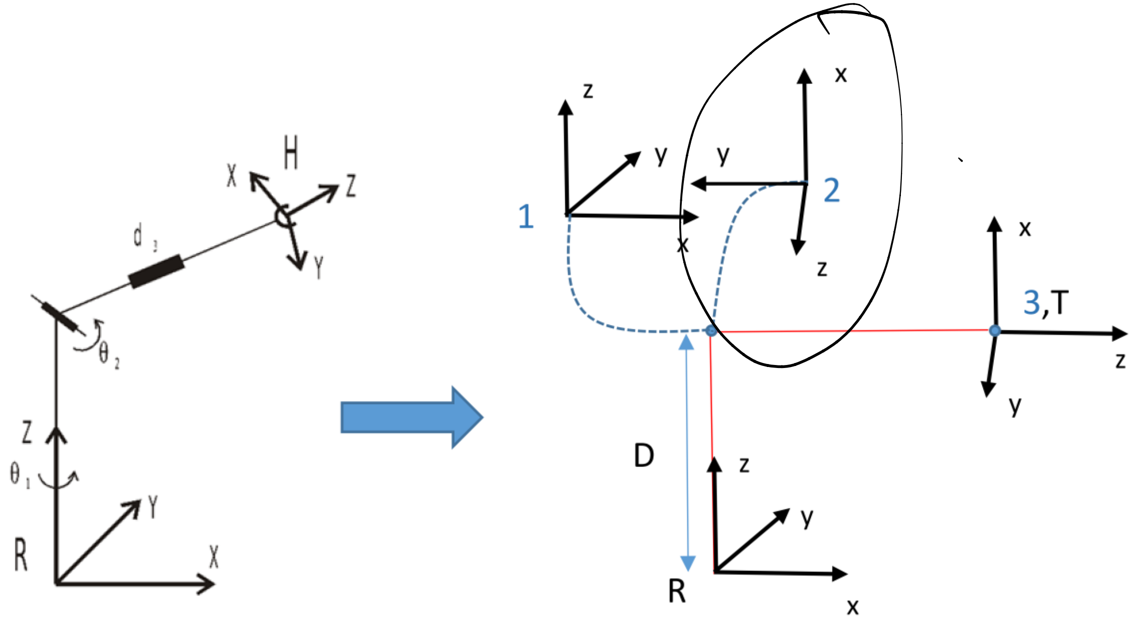
Note: remember to give also the additional base transformation matrix and the tool transformation matrix for adjusting the Hand frame orientation to be the one shown in the figure.



*A three-axis **RRP**-manipulator. The positive directions of the first two rotational joints are shown in the figure. The distance of the rotational axis of the shoulder joint, Θ_2, from the origin of the R-frame (along the z_R-axis) is D=0.45 meters. In the zero configuration/position (i.e. all the joint control values equal zero) the upper arm is parallel to the X-axis of the R-frame (X_H is parallel to Z_R, Y_H to negative Y_R and Z_H to X_R)*

Solution

Forward kinematic model of the RRP-robot by applying the **modified** DH-parameter representation.



Link	α_{i-1}	a_{i-1}	d_i	Θ_i	σ_i
1	0	0	0	Θ_1	R
2	90	0	0	Θ_2+90	R
3	90	0	d_3	0	P

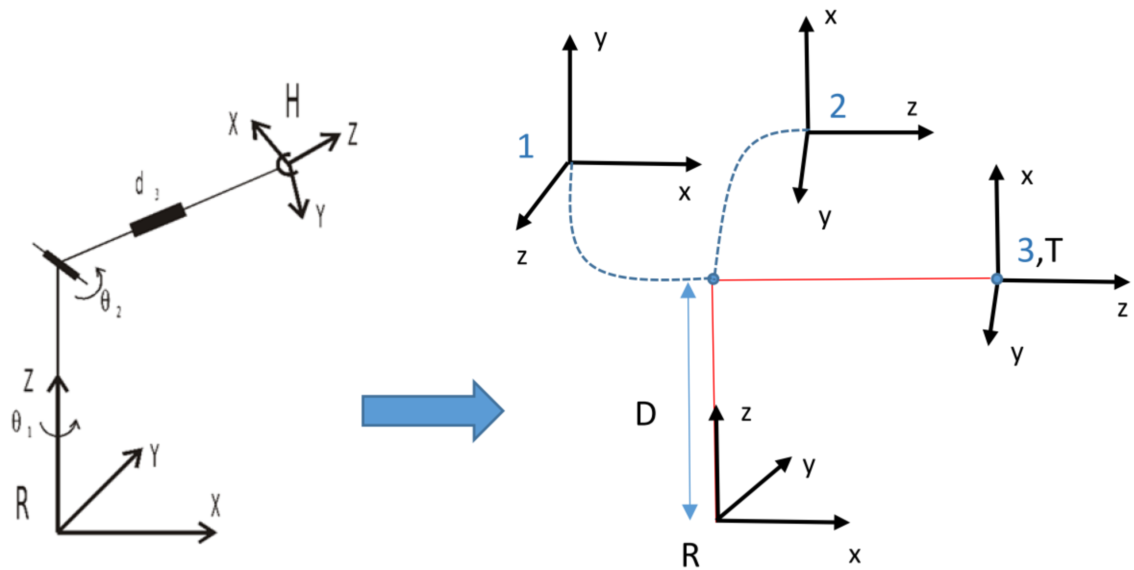
Base transformation: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & D \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Tool transformation: No tool transformation needed (we end up at the H-frame if we assume that the length of the shoulder link is considered entirely by the prismatic joint d_3).

Let's make the forward kinematics model for the same RRP-robot mechanism introduced in the previous problem, but now using the **standard** **DH-parameters**. Otherwise the task description is the same.

Solution

Forward kinematic model of the RRP-robot by applying the **standard** DH-parameter representation.



Link	Θ_i	d_i	a_i	α_i	σ_i
1	Θ_1	0	0	90	R
2	Θ_2+90	0	0	90	R
3	0	d_3	0	0	P

Base transformation: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & D \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Tool transformation: No tool transformation needed (we end up at the H-frame if we assume that the length of the shoulder link is considered entirely by the prismatic joint d_3).

Recommended reading:

Peter Corke, Robotics, Vision and Control, Fundamental Algorithms in MATLAB, Second Edition, Springer, 2017, pages 191-204 (not ch. 7.1.2.2), pp. 218-219 (i.e. ch. 7.4.3)

Craig, J.J, *Introduction to Robotics: Mechanics and Control,* Third Edition, Prentice Hall, 2005, pages 62-83