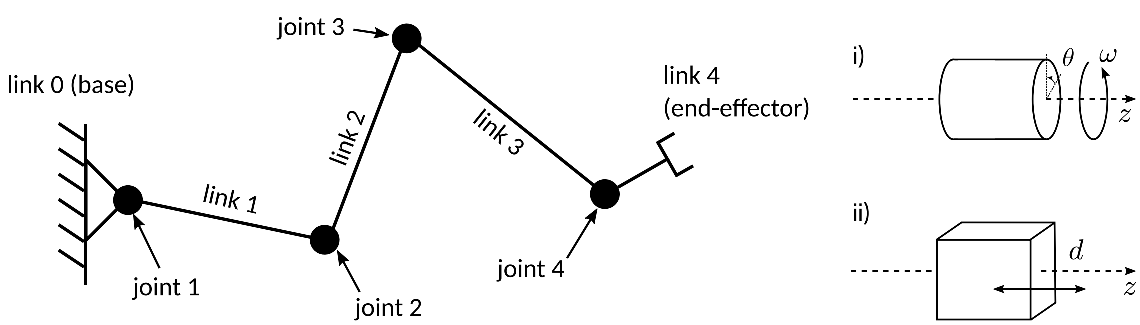
Institute for Cognitive Systems
Technische Universität München
Prof. Gordon Cheng

Multi-sensory Based Robot Dynamic Manipulation
Tutorial 2: Denavit-Hartenberg

1 Kinematic model of a robot manipulator

a Kinematic Chains

Conventional robot manipulators can be seen as a set of **links**, interconnected by **joints**, into a **kinematic chain**, as shown in Fig. 1a. Joints are typically revolute or prismatic (c.f. Fig. 1b). A revolute joint is like a hinge and allows relative rotation between two links. A prismatic joint allows a linear relative motion between two links. In practice, joints can take many different forms, either simple such as prismatic and revolute, or more complex, such as ball or socket joints. In this course, it is assumed that all joints have a single degree of freedom (DOF). We therefore only consider revolute or prismatic joints. It is interesting to notice that this assumption does not involve any real loss of generality, since complex joints such as a 2-DOF ball and socket joint or a 3-DOF spherical wrist can always be thought of as a succession of single DOF joints with links of length zero in between.



(a) Translation vector

(b) Revolute joint (i) vs Prismatic joint (ii)

Figure 1: Illustration of a Kinematic Chain

A serial robot manipulator with n joints will have $n+1$ links, since each joint connects two links. With the i^{th} joint, we associate a *joint variable*, denoted by "q_i". In the case of a revolute joint, q_i is the angle of rotation, and in the case of a prismatic joint, q_i is the joint displacement (c.f. Fig. 1b):

$$q_i = \begin{cases} \theta_i & : \text{ if joint } i \text{ is } \textbf{revolute} \\ d_i & : \text{ if joint } i \text{ is } \textbf{prismatic} \end{cases} \tag{1}$$

The set of all joint parameters form the **joint space** of a robot manipulator.

b Forward kinematics

The objective of **kinematics** is to map the Cartesian variables $\mathbf{x} = [x, y, z, \varphi, \theta, \psi]^T$ related to any point of a robot, to a set of joint variables $\mathbf{q} = [q_1, q_2, ..., q_n]^T$, without consideration of the inertia or external forces. To perform the kinematic analysis of a robot manipulator, we rigidly attach a coordinate frame to each of its links. In particular, we attach the frame $(O_i x_i y_i z_i)$ to link i. By this convention, joint i connects link $i-1$ to link i. We will consider the location of joint i to be fixed with respect to link $i-1$. *When joint i is actuated, link i moves*. The frame $(O_0 x_0 y_0 z_0)$, which is attached to the robot base, is referred to as the **inertial frame**. Now suppose $\mathbf{H}_{i-1}^i, i = 1, 2, ..., n$ is the homogeneous transformation matrix that expresses the position and orientation of $(O_i x_i y_i z_i)$ with respect to $(O_{i-1} x_{i-1} y_{i-1} z_{i-1})$. Taking the above definition of a joint, $\mathbf{H}_{i-1}^i$ is a *function of only a single joint variable*, namely q_i. The resulting homogeneous transformation matrix $\mathbf{H}_0^n(\mathbf{q})$ is given by:

$$\mathbf{H}_0^n(\mathbf{q}) = \begin{bmatrix} \mathbf{R}_0^n(\mathbf{q}) & \mathbf{p}_0^n(\mathbf{q}) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2}$$

This matrix can be computed as a product of every relative link transformation matrices:

$$\mathbf{H}_0^n(\mathbf{q}) = \mathbf{H}_0^1(q_1)\mathbf{H}_1^2(q_2)...\mathbf{H}_{n-1}^n(q_n) \tag{3}$$

In this context the objective of forward kinematic analysis is to determine the *cumulative effect* of the entire set of joint variables over a **point of interest**. In practice, this *point of interest* can be any point of the robot although in general it is the *end-effector*, where the tool is attached. More formally, the forward kinematics of a serial robot manipulator can be defined a function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, which describes the **position** and **orientation** of the end-effector with respect to the base coordinate frame. Here n is the joint space dimension – i.e. the number of joints of the considered robot – and m is the dimension of Cartesian space. Typically $m = 6$ since three translational parameters and 3 rotational parameters are required to define the pose of a given rigid body in Cartesian space.

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m \tag{4a}$$
$$\mathbf{q} \rightarrow \mathbf{x}_{ef} = f(\mathbf{q}) \tag{4b}$$

When no *kinematic decoupling*[1] has been used to describe the kinematic chain of a robot arm, the *forward kinematic* is computed directly from the translation vector $\mathbf{p}_0^n(\mathbf{q})$ and the rotation matrix $\mathbf{R}_0^n(\mathbf{q})$, obtained from $\mathbf{H}_0^n(\mathbf{q})$ as follows:

$$\boxed{\mathbf{x}_{ef} = \begin{bmatrix} \mathbf{p}_0^n(\mathbf{q}) \\ \mathrm{rpy}(\mathbf{R}_0^n(\mathbf{q})) \end{bmatrix}} \tag{5}$$

where $\mathrm{rpy}() : \mathbb{R}^{3\times 3} \subset \mathbf{SO}(3) \rightarrow \mathbb{R}^3$ is a function computing the orientation of a body in terns of Euler angles (roll-pitch-yaw) from a given rotation matrix $\mathbf{R}_0^n(\mathbf{q})$. Such a function was for example implemented in the last tutorial.

[1] This topic will be covered in Session 3. For now, lets say that *kinematic decoupling* is a special coordinate frame arrangement where the orientation and position of the end-effector are controlled by a set of different joint variables, therefore they are decoupled from each other.

c Denavit Hartenberg Representation

c.1 Motivation

Fully describing the pose of a rigid body in Cartesian space usually requires a set of 6 distinct parameters, describing respectively rotations and translations with respect to each axis of the base frame. This method can rapidly become problematic for both structural and computational reasons. A commonly used alternative for selecting frames of reference at lower computational complexity is the so called *Denavit-Hartenberg (D-H) convention*. The effectiveness of this methodology lies in its simplicity and intuitiveness: the main idea is to set the coordinate frames for each link in a *specific form*. This form will limit the relative motion between consecutive frames to a pair of axes. In other words, motions (rotation and translation) on a specific axis are *constrained*. In particular, D-H constrains the motion on the *y-axis*. Therefore, in order to specify the position and orientation between two consecutive frames, *only 4 parameters are needed*.

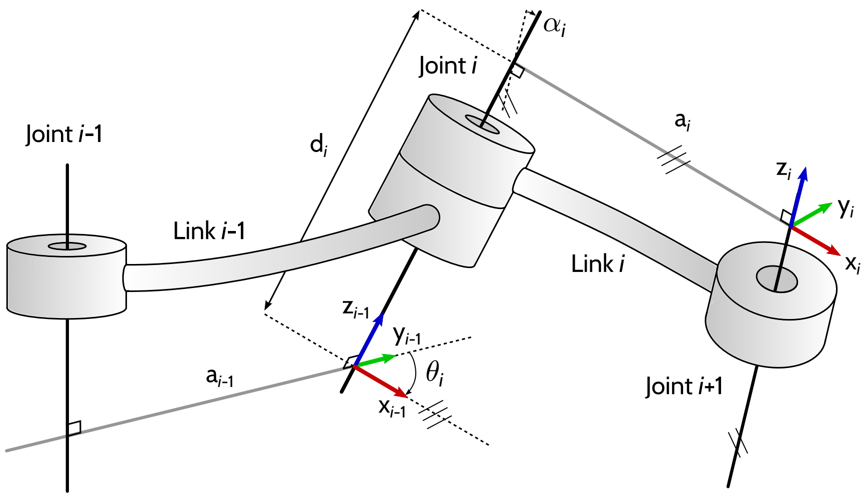


Figure 2: Classic (Distal) Denavit Hartenberg convention

c.2 Classical (Distal) Denavit Hartenberg formalism

In a serial robot, the link i is connected to two other links (i.e. link $i-1$ and link $i+1$). Two joint axes are therefore defined at both ends of each link as shown in Fig.2. By convention, these joint axes are set as the z axes of each coordinate systems. Joints $i-1$ and i are connected by link $i-1$, while joints i and $i+1$ are connected by link i. Since the different links are rigid, they maintain a fixed configuration between the joints which can be characterized by two parameters a_i and α_i, which determine the structure of the link:

- **link length** a_i: distance from O_i to the intersection of z_{i-1} and x_i, along the x_i axis.
- **twist angle** α_i: angle between z_{i-1} and z_i, measured about x_i.

These two parameters do not change with the robot configurations. A joint axis establishes the connection between two links. This joint axis will have two normals connected to it, one for each link. The distance and the angle between these two normals can be characterized by two other parameters, d_i and $\theta_i$determining the relative position of neighboring links:

- **link offset** d_i: distance between O_{i-1} and the intersection of z_{i-1} with the x_i.
- **joint angle** θ_i: angle between x_{i-1} and x_i, measured about z_{i-1}.

For revolute joint, θ_i varies and d_i is a fixed length (i.e. zero or constant). For prismatic joint, d_i varies and θ_i is zero or constant. Using the classical D-H convention, each homogeneous transformation $\mathbf{H}_{i-1}^{i}(q_i)$ is represented as a product of the following four basic transformations:

$$\boxed{\mathbf{H}_{i-1}^{i}(q_i) \Leftrightarrow \underbrace{\mathrm{Rot}_{z_{i-1},\theta_i} \times \mathrm{Trans}_{z_{i-1},d_i}}_{\text{w.r.t frame } i-1} \times \underbrace{\mathrm{Trans}_{x_i,a_i} \times \mathrm{Rot}_{x_i,\alpha_i}}_{\text{w.r.t frame } i}}$$

$$\mathbf{H}_{i-1}^{i}(q_i) = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6a}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6b}$$

We can summarize the distal D-H convention in the following algorithm for deriving the forward kinematics for any serial manipulator:

1. Locate and label the joint axes $z_0, ..., z_n$ as the **axis of motion** of each joint.
2. Establish x_i: $x_i \perp z_{i-1}$, and x_i intersects z_{i-1}, i.e. the minimum distance between axis z_{i-1} and the coordinate frame i. Some times, you have to move the origin point of the coordinate frame O_i to satisfy this condition. **Please note, O_i can be moved only along z_i axis**.
3. Set y_i to complete a right-hand frame.
4. Create a table of link parameters $\theta_i, d_i, a_i, \alpha_i$:

Link i	θ_i	d_i	a_i	α_i
1				
$\vdots$				
n				

5. Compute the relative homogeneous transformation matrices $\mathbf{H}_{i-1}^{i}(q_i)$ by substituting the above parameters into eq.(6).
6. Compute the absolute homogeneous transformation matrix $\mathbf{H}_0^n(\mathbf{q}) = \mathbf{H}_0^1(q_1)\mathbf{H}_1^2(q_2)...\mathbf{H}_{n-1}^n(q_n)$. This matrix provides the position and orientation of the end-effector frame with respect to the base coordinate frame.

Important remarks:

- For a **prismatic joint** i, θ_i is a constant, while d_i is the i^{th} joint variable q_i.
- For a **revolute joint** i, d_i is constant, while θ_i is the i^{th} joint variable q_i.
- The parameters a_i and α_i are ***always constant*** (do not depend on q_i).
- There can be only **one** joint variable q_i in each row of the D-H table.
- The above rules apply for any given point of interest in the robot. This means, that we can compute the relative/absolute position/orientation of any desired point (besides the end-effector). This is particularly important to compute the *special* points known as *centers of mass*.
- For the ***end-effector*** (n^{th}-coordinate frame) the orientation of z_n is **not constrained** and only x_n must follow the D-H convention. This also apply for the ***centers of mass*** of each link, by simply choosing $z_{cm_i} = z_{i-1}$.

c.3 Modified (Proximal) Denavit Hartenberg formalism

A modified (proximal) D-H convention was proposed in J. Craig's book "Introduction to Robotics", 1986. Although this convention is perfectly valid, it is not widely used. In this convention:

- z_i is on joint i in stead of joint $i+1$ (proximal convention)
- a_{i-1} and α_{i-1} are respectively the distance and twist angle from z_{i-1} to z_i, measured along and about x_{i-1} as shown in Fig.3
- d_i and θ_i are the distance and angle from x_{i-1} to x_i, measured along and about z_i

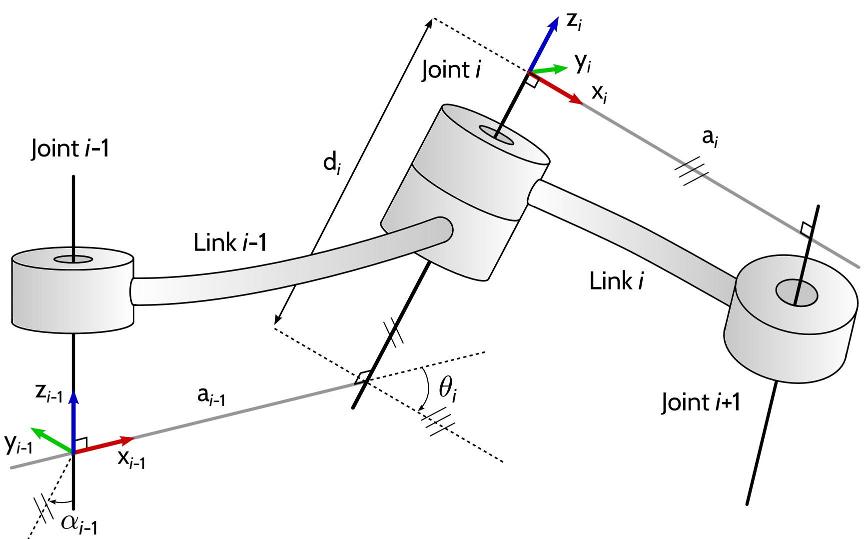


Figure 3: Modified (Proximal) Denavit Hartenberg convention

Using the modified D-H convention, each homogeneous transformation $\mathbf{H}_{i-1}^{i}(q_i)$ is represented as a product of the following four basic transformations:

$$\boxed{\mathbf{H}_{i-1}^{i}(q_i) \Leftrightarrow \underbrace{\mathrm{Rot}_{x_{i-1},\alpha_{i-1}} \times \mathrm{Trans}_{x_{i-1},a_{i-1}}}_{\text{w.r.t frame } i-1} \times \underbrace{\mathrm{Trans}_{z_i,d_i} \times \mathrm{Rot}_{z_i,\theta_i}}_{\text{w.r.t frame } i}}$$

$$\mathbf{H}_{i-1}^{i}(q_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & 0 \\ 0 & s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7a}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & a_{i-1} \\ s_{\theta_i} c_{\alpha_{i-1}} & c_{\theta_i} c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & -d_i s_{\alpha_{i-1}} \\ s_{\theta_i} s_{\alpha_{i-1}} & c_{\theta_i} s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & d_i c_{\alpha_{i-1}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7b}$$

c.4 Example 1: Three-Link Cylindrical Robot

Consider the three-link cylindrical robot represented in the following figure:

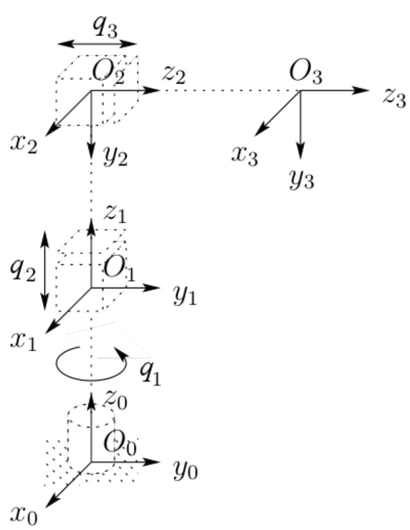


Figure 4: Three-Link Cylindrical Robot

The D-H parameters for this robot are shown in following table:

Link i	θ_i	d_i	a_i	α_i
1	q_1+0	d_1	0	0
2	0	$q_2 + 0$	0	-90^o
3	0	$q_3 + 0$	0	0

Following the classical D-H convention, we use eq.(6) in order to find the relative transforms between each frame of the robot:

$$\mathbf{H}_0^1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{H}_1^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & q_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}_2^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, the corresponding $\mathbf{H}_0^n(\mathbf{q})$ matrix is:

$$\mathbf{H}_0^3(\mathbf{q}) = \mathbf{H}_0^1(q_1)\mathbf{H}_1^2(q_2)\mathbf{H}_2^3(q_3) = \begin{bmatrix} c_1 & 0 & -s_1 & -s_1 q_3 \\ s_1 & 0 & c_1 & c_1 q_3 \\ 0 & -1 & 0 & d_1 + q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

Finally, we can obtain end-effector position using the definition of *forward kinematics*:

$$\mathbf{x}_{ef} = \begin{bmatrix} \mathbf{p}_0^n(\mathbf{q}) \\ \mathrm{rpy}(\mathbf{R}_0^n(\mathbf{q})) \end{bmatrix} \tag{9}$$

c.5 Example 2: Cylindrical robot with spherical wrist

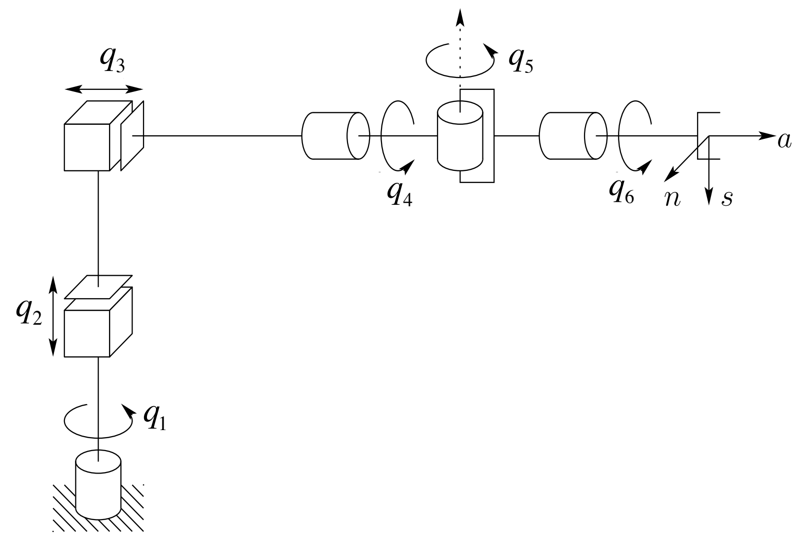


Figure 5: Cylindrical robot with spherical wrist.

Now, suppose that we attach a spherical wrist to the cylindrical manipulator of the previous example as shown in Figure 5. Set the coordinate frames for all the Links and obtain the D-H table for this robot.

Link i	θ_i	d_i	a_i	α_i
1	q_1+0	d_1	0	0
2	0	$q_2 + 0$	0	-90^o
3	?	?	?	?
4	?	?	?	?
5	?	?	?	?
6	?	?	?	?

Table 1: D-H Parameters for the Cylindrical robot with Spherical Wrist.

2 Homework

a Exercise 1: D-H Representation & Forward Kinematics

Consider the robot depicted in Fig. (6). This robot architecture is used for ankle rehabilitation purposes[2]. It has 4 degrees of freedom, namely *revolute-prismatic-prismatic-revolute* (RPPR) joints. The patient's foot is attached to the platform and moves with the robot. This allows to quantify the progress of a rehabilitation process. The end-effector is located at the center of the plate (in the same place as the cm_4).

$[mm]$
$L_1 = 300$
$L_2 = 340$
$L_3 = 583$
$L_4 = 80$
$L_5 = 80$
$L_6 = 115$
$L_7 = 82$
$L_8 = 108$
$L_9 = 123$
$L_{10} = 167$
$L_{11} = 28$
$L_{12} = 50$
$c = d = 320$
$\theta_1 = 45°$
$\theta_2 = 30°$

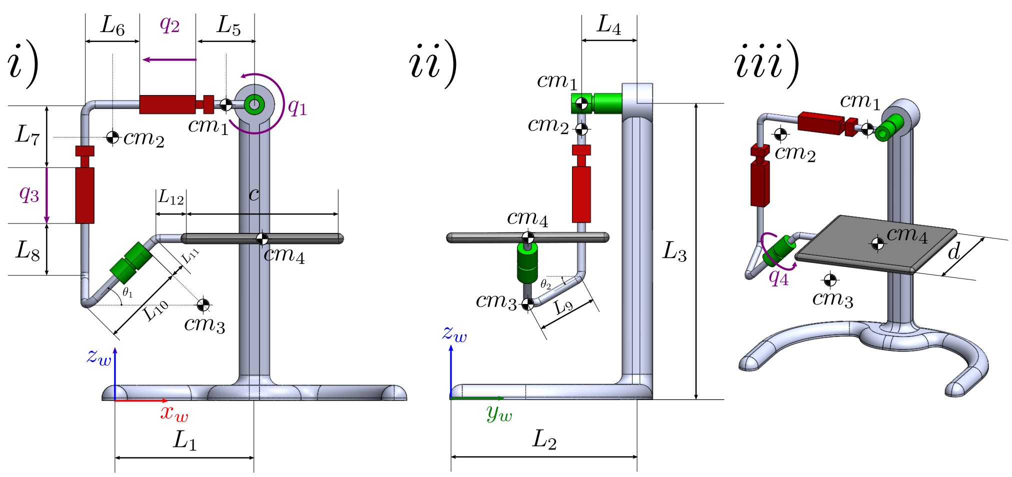


Figure 6: 4 DOF Rehabilitation Robot. $i)$ Front view, $ii)$ Right view, and $iii)$ Isometric view. The figure also shows the *center of mass* of each link cm.

Complete the following tasks:

1. Define and draw the coordinate frames for each Link, according to the classical (distal) D-H convention. Submit the "`DHs.pdf`" file with the DH coordinate frames. Use the drawing provided at the end of this document as template.
2. Define special coordinate frames for each center of mass. Each center of mass is depicted in the figures by a cm_i label. Treat each center of mass in the same form as the end-effector. Draw the corresponding frames on a different file named "`DHcms.pdf`".
3. Generate the D-H Table. In this case, the D-H Table is comprised of eight rows, four for the links and four others for the centers of mass. Put your D-H table in the "`DHs.pdf`" file you used for the frames.
4. Using Matlab, code a m-file to compute the relative $\mathbf{H}_{i-1}^{i}(q_i)$, absolute $\mathbf{H}_0^i(\mathbf{q})$ and $\mathbf{H}_W^i(\mathbf{q})$ homogeneous transformations for each Link, and each center of mass, in symbolic form (use the Matlab's *Symbolic Math Toolbox* and the provided template). Use the absolute transformations to plot the position of the cm_2.

[2] The robot design is taken from the paper: Emiris et al., "Design of a simple and modular 2-DOF ankle physiotherapy device relying on a hybrid serial-parallel robotic architecture", published in the Journal of Applied Bionics and Biomechanics, in 2011

5. Generate the Forward Kinematic function. The obtained function should be a symbolic closed form equation for the forward kinematics and not a numeric solution, see function `FK_robot(u)`, provided in the template folder. Use this function to plot the position of the end-effector.
6. Visualize the robot links using the provided Matlab/Simulink templates for this session. In Fig.7 is depicted the expected output from the Simulink simulation after computing and coding the correct Matlab functions. Use a sinusoidal function for each joint q_i in order to animate the robot in different poses.

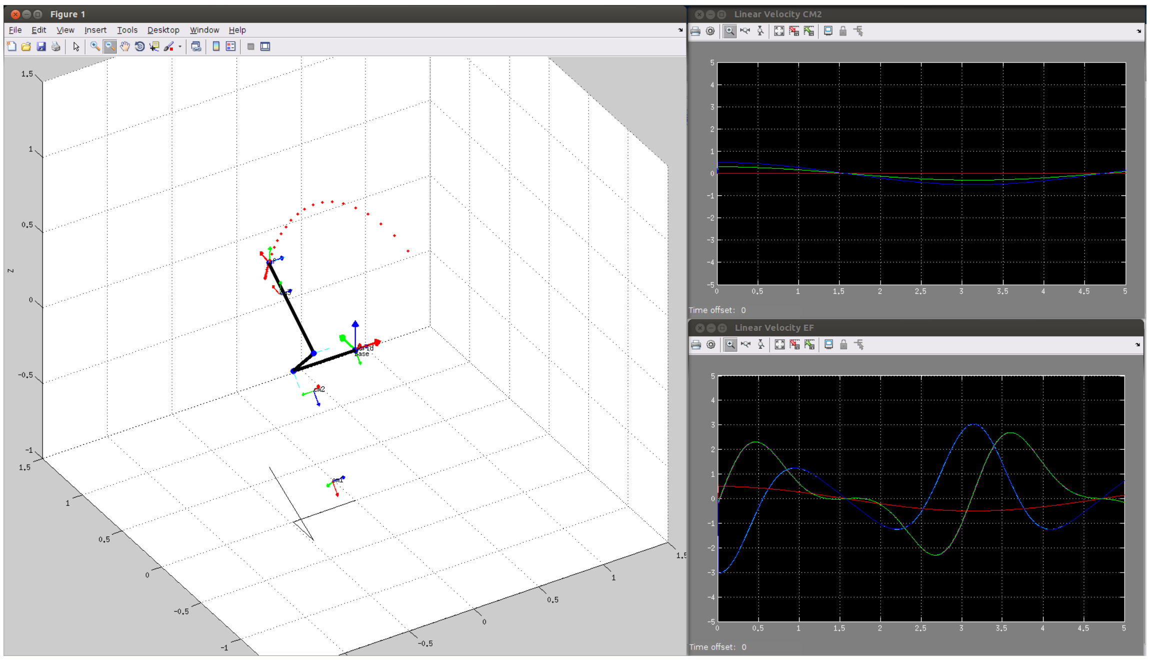
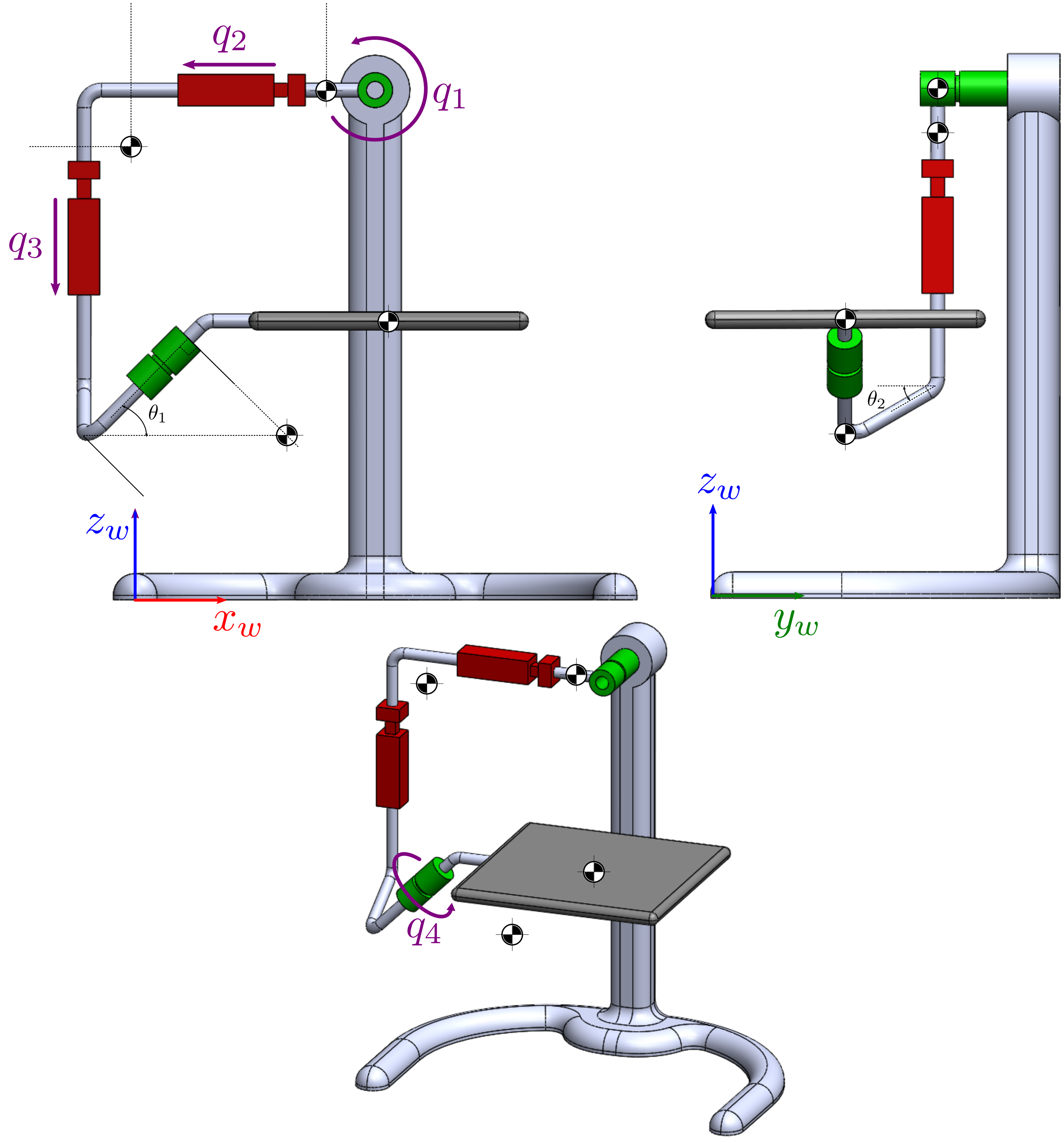


Figure 7: Example result after computing all the models in Matlab/Simulink for a 3DOF robot.



q_2
q_1
q_3
θ_1
θ_2
z_w
x_w
z_w
y_w
q_4