UNIVERSITY OF GRONINGEN

ROBOTICS FOR IEM

Assignment 1

Author:
Jonathan Chandra
(s5161533)

Lecturer:
Dr. Bahar Haghighat

September 28, 2023

Question 1

1. Assigning frame

The frames were assigned based on the right hand rule (RHR). Z axes of each frames is firstly determined and followed with the RHR, X and Y axes can be determined. Figure 1 shows the robot with assigned frame to all the joints. $\odot$ represents the axis pointing vertically outward from the paper (Z axis Frame 2 and 3), while $\otimes$ represents the axis pointing vertically inward to the paper (Y Axis in Frame 0 and 1).

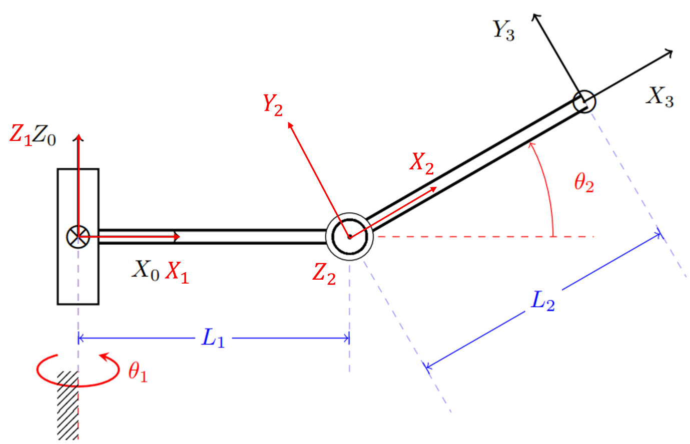


Figure 1: Frame assignment on 2R robot (non-planar)

*note: $[X_i, Y_i, Z_i]^T$ where $i = 1, 2, 3$ are the axis frames assigned.

2. Denavit Hartenberg table

The assigned frame from the previous subsection can be used to create the Denavit Hartenberg (DH) table. This table can be constructed based on the links parameter $a_{i-1}, \alpha_{i-1}, d_i, \theta_i$ [1]. Each row in the table corresponds to a joint describing its relationship with the previous joint and the link connecting them. Table 1 (next page) shows the DH table of the robot in Figure 1.

Link	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	L_1	$\pi/2$	0	θ_2
3	L_2	0	0	0

Table 1: DH Parameters for a 2R robot (non-planar)

3. Homogeneous Transformation Matrix

Homogenous transformation matrix can be defined with a general form ${}^{i-1}_{i}T$ [1]. It is defined as:

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

Substituting the DH parameters from Table 1 into (1) shows the neighboring homogeneous matrices ${}^{i-1}_{i}T$, $i = 1, 2, 3$ of the Robot on Figure 1

$$
\begin{aligned}
{}^{0}_{1}T &= \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{1}_{2}T &= \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_1 \\ 0 & 0 & -1 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{2}_{3}T &= \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{2}
$$

Homogeneous transformation matrix ${}^{0}_{3}T$ from the base frame to the end-effector frame can be computed with:

$$
{}^{0}_{3}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T \tag{3}
$$

Substituting (2) into (3) yields:

$$
\begin{aligned}
{}^0_3T &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & L_1 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & c_2L_2 + L_1 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & s_2L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_1c_2 & -c_1s_2 & s_1 & c_1c_2L_2 + c_1L_1 \\ s_1c_2 & -s_1s_2 & -c_1 & s_1c_2L_2 + s_1L_1 \\ s_2 & c_2 & 0 & s_2L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{4}
$$

*Note: $c_j = \cos\theta_j$ and $s_j = \sin\theta_j$, where $j = 1, 2$

4. End Effector Position

Given a set of angles (in this case $\Theta = [\theta_1, \theta_2]^T$), forward kinematics can be used to compute the position of the end-effector. Recall that (1) consists of 2 set of matrix, that is the rotation and translation matrix. The end effector position can be obtained from the translation matrix computed in (4).

A MATLAB function file `fk.m` is used to compute the Homogeneous Transformation Matrix in (4) with the input Θ. In this report, two sets of $\Theta = [\theta_1, \theta_2]^T = [\frac{\pi}{4}, \frac{\pi}{5}]^T$ and $[\theta_1, \theta_2]^T = [\frac{\pi}{6}, \frac{\pi}{2}]^T$ is used as an input for `fk.m`. The listing of `fk.m` is available in Appendix B.

The MATLAB function file basically computes ${}^0_3T$ with the given Θ and takes the elements of translation matrix inside ${}^0_3T$ as an output that describes the end-effector position.

Assuming that $L_1 = L_2 = 1$, and the given angle $\Theta = [\frac{\pi}{4}, \frac{\pi}{5}]^T$, we have:

$$
\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1.2792 \\ 1.2792 \\ 0.5878 \end{bmatrix} \tag{5}
$$

Now, using the second $\Theta = [\frac{\pi}{6}, \frac{\pi}{2}]^T$, we have:

$$
\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.8660 \\ 0.5000 \\ 1.0000 \end{bmatrix} \tag{6}
$$

This concludes the answer to question 1.

Question 2

This question involves on deriving the forward kinematics solution of a low-cost robotic manipulator AX-18A Smart Robotic Arm.

1. Assigning frame

The first step involves the assignment of the joint frames. Recall the assignment worksheet states that offsets exist from the zero configuration, that is $\theta_2 = \pi/2, \theta_3 = -\pi/2$. Figure 4 in Appendix A. depicts frame assignment of the robot zero configuration, which leads to Figure 2 that depicts the frame assignment with offsets given.

2. Denavit Hartenberg table

Given the frame assignment in Figure 2, DH table can be generated as follows:

Link	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	L_1	$\pi + \theta_1$
2	0	$-\pi/2$	0	θ_2
3	L_2	0	0	θ_3
4	L_4	$\pi/2$	$L_3 + L_5$	$\pi/2 + \theta_4$
5	0	$-\pi/2$	0	θ_5
6	0	$\pi/2$	L_6	$\pi/2$

Table 2: DH Parameters of AX-18A Smart Robotic Arm

Note that Table 2 shows the DH parameter of AX-18A with the given offset of $\theta_2 = \pi/2, \theta_3 = -\pi/2$ (Not in zero configuration).

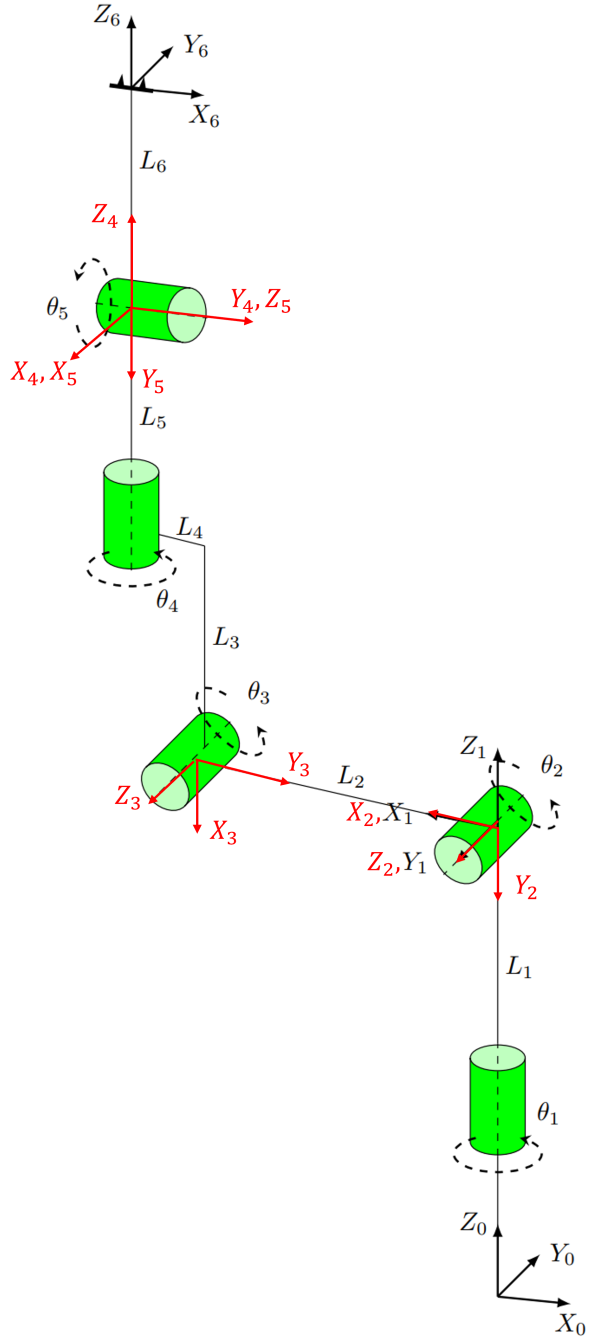


Figure 2: Frame assignment on AX-18A Smart Robotic

3. Homogeneous Transformation Matrix

Substituting the DH parameters from Table 2 into (1) shows the neighboring homogeneous matrices ${}^{i-1}_{\ \ i}T$, $i = 1, \ldots, 6$ of the robot on Figure 2.

$$
\begin{aligned}
{}^0_1T &= \begin{bmatrix} \cos(\theta_1+\pi) & -\sin(\theta_1+\pi) & 0 & 0 \\ -\sin(\theta_1+\pi) & \cos(\theta_1+\pi) & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^1_2T &= \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin\theta_2 & -\cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^2_3T &= \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^3_4T &= \begin{bmatrix} \cos(\theta_4+\frac{\pi}{2}) & -\sin(\theta_4+\frac{\pi}{2}) & 0 & L_4 \\ 0 & 0 & -1 & -(L_3+L_5) \\ \sin(\theta_4+\frac{\pi}{2}) & \cos(\theta_4+\frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^4_5T &= \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin\theta_5 & \cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^5_6T &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & L_6 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{7}
$$

4. DH parameters change on robot movement and its significance

A robot may maneuvers through its designated workspace, the changes in its configuration are mainly driven by the variations in the DH parameters. There are 2 joints to be accounted for. For revolute joints, the joint angle θ_i is the main variable that changes the rotational position of one link w.r.t its predecessor. As for prismatic joints, it's the link offset d_i that adjusts, denoting the link linear translation.

Despite those variations, the two remaining parameters maintain their value constant. The link length a_{i-1}, tells the fixed distance between two successive joint axes, and the link twist α_{i-1}, which defines the fixed angular difference between these axes, remain unchanged regardless of the robot's movement or the type of joint. These parameter is then combined to maneuver the robot's and its end effector's pose within its workspace.

In summary, as the robot moves within its workspace, the joint angle θ_i varies for revolute joints, the link offset d_i varies for prismatic joints, while the link length a_{i-1} and link twist α_{i-1} stays constant for both joint types.

5. Implementation of Forward Kinematics on MATLAB

Forward Kinematics can be implemented inside MATLAB with the use of Peter Corke's Robotics Toolbox. In this report, the toolbox is used to create the model of the robotic arm based on the DH parameter. A script named `arm_creation_val.m` is used to generate the robot model figure. The listing of `arm_creation_val.m` is available in Appendix C.

Generating the robot model is quite straight forward. Five links with an end effector is defined based on the DH parameter derived in Table 2. It is to be noted that a modified DH-parameter is used, an option of 'modified' needed to included in the link definition function (Revolute).

Figure 3 in the next page depicts AX-18A model in its zero configuration created using the Robotics Toolbox.

This concludes the answer to Question 2.

References

[1] Craig, J. (2021). Introduction to Robotics, Global Edition. United Kingdom: Pearson Education Limited.

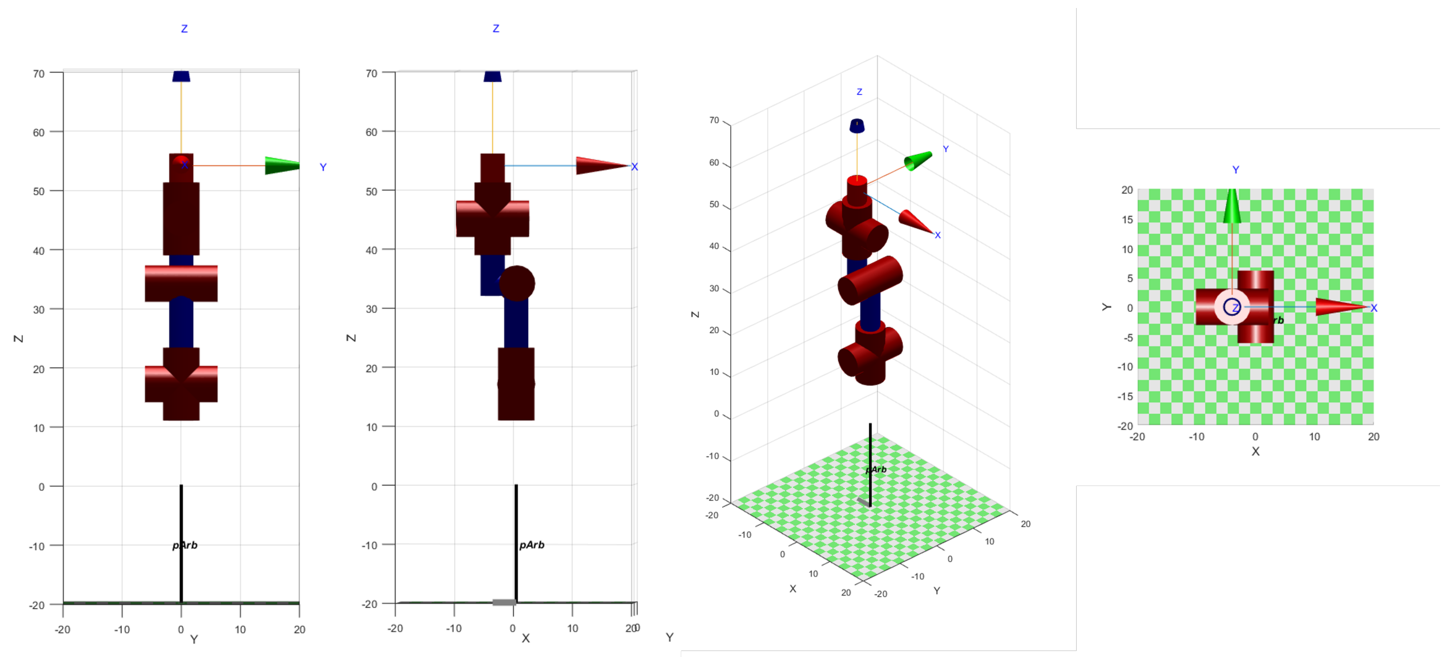
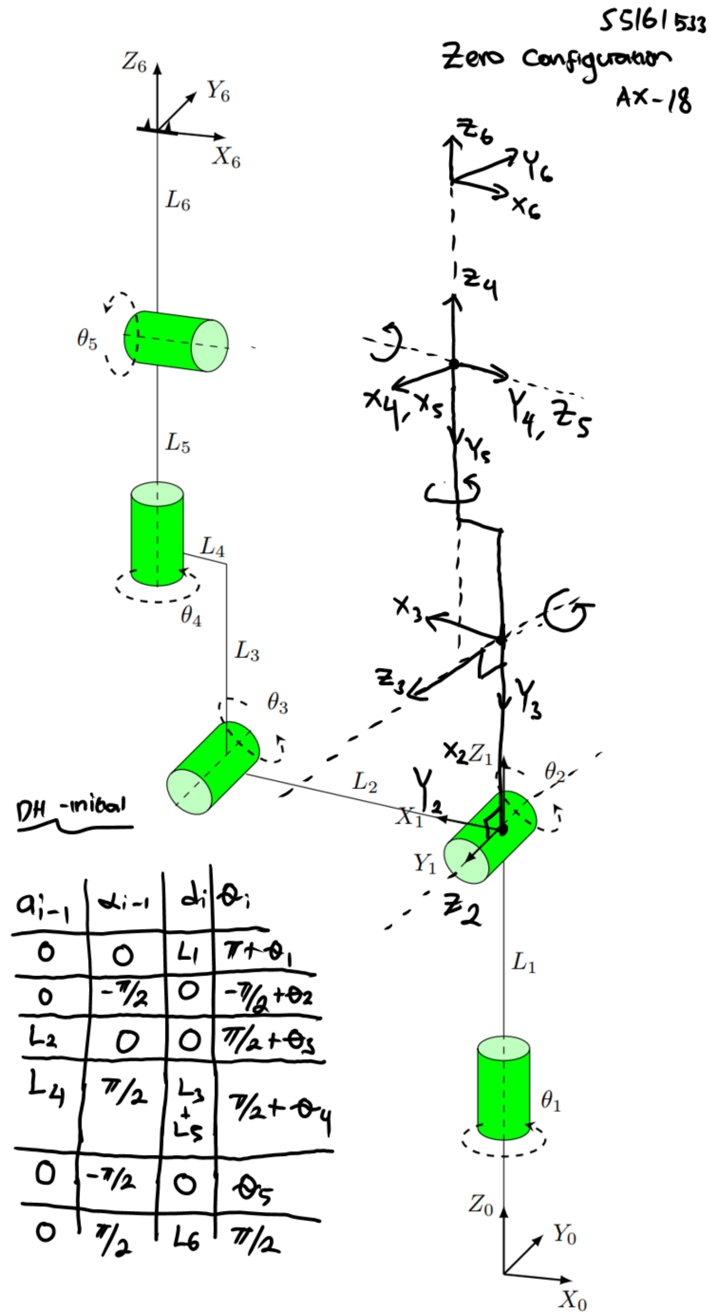


Figure 3: AX-18A model Virtual Model

Appendix

A. AX-18A frame assignment derivation



a_{i-1}	α_{i-1}	d_i	θ_i
0	0	L_1	$\pi+\theta_1$
0	$-\pi/2$	0	$-\pi/2+\theta_2$
L_2	0	0	$\pi/2+\theta_3$
L_4	$\pi/2$	L_3+L_5	$\pi/2+\theta_4$
0	$-\pi/2$	0	θ_5
0	$\pi/2$	L_6	$\pi/2$

Figure 4: Frame assignment on AX-18A Smart Robotic Derivation

B. Listing for `fk.m`

```
function end_pos = fk(theta1, theta2)
% The inputs of this function are two angles: theta1
    and theta2, the output
% of this function is the position of the end-
   effector: end_pos, which is
% a 3D column vector

L1 = 1; % link lengths
L2 = 1;

%% Angles input and initialization
st1 = sin(theta1);
ct1 = cos(theta1);
st2 = sin(theta2);
ct2 = cos(theta2);

%% Homogeneous transformation matrix
T_03 = [ct1*ct2 -ct1*st2 st1 ct1*ct2*L2+ct1*L1;
        st1*ct2 -st1*st2 -ct1 st1*ct2*L2+st1*L1;
        st2 ct2 0 st2*L2;
        0 0 0 1
        ];
end_pos =  [T_03(1,4);
            T_03(2,4);
            T_03(3,4);
            ];

% Made by J. Chandra

end
```

C. Listing for `arm_creation_val.m`

```
L1 = 17; % note: all lengths are given in cm.
L2 = 17;
L3 = 7;
L4 = 4;
L5 = 4;
L6 = 9;

%%%%%% Write your codes below this line. DO NOT
   change other parts. %%%%%%

% Assigning values to the links using the robotic
   toolbox

L(1) = Revolute('d', L1, 'a', 0, 'alpha', 0, 'offset
   ', pi, 'modified');
L(2) = Revolute('d', 0, 'a', 0, 'alpha', -pi/2, '
   offset',-pi/2, 'modified');
L(3) = Revolute('d', 0, 'a', L2, 'alpha', 0, 'offset
   ', pi/2, 'modified');
L(4) = Revolute('d', (L3+L5), 'a', L4, 'alpha', pi
   /2, 'offset', pi/2,'modified');
L(5) = Revolute('d', 0, 'a', 0, 'alpha', -pi/2, '
   offset',0,'modified');
pArb = SerialLink(L, 'name', 'pArb');

% adding end effector
pArb.plotopt={'workspace', [-20 20 -20 20 -20 70]};
pArb.tool =  trotx(90) * trotz(90) * transl(0, 0, L6
   );

% plot the arm in the zero configuration using the
   robotic toolbox
figure(1234);
pArb.plot([0, 0, 0, 0, 0])
```