

Robots Fundamentals: Assignment report

Rodrigo MORENO CARRILLO

November 26, 2018

1 Part 1

1.1 Forward kinematics

According with the diagram of the Lynxmotion robot depicted in figure 1 the correspondig Denavit-Harterverg table using distal convention is:

n	a_n	α_n	d_n	θ_n
1	0	90°	d_1	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3
4	0	90°	0	θ_4
5	0	0	d_5	θ_5

Table 1: Denavit-Hartenverg table using distal convention

In the diagram it is shown that the robot contains five different frames of reference which are described using the parameters in table 1. Using these four parameters the frames can be described by the following equations:

$${}^0T_1 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$${}^1T_2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^2T_3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

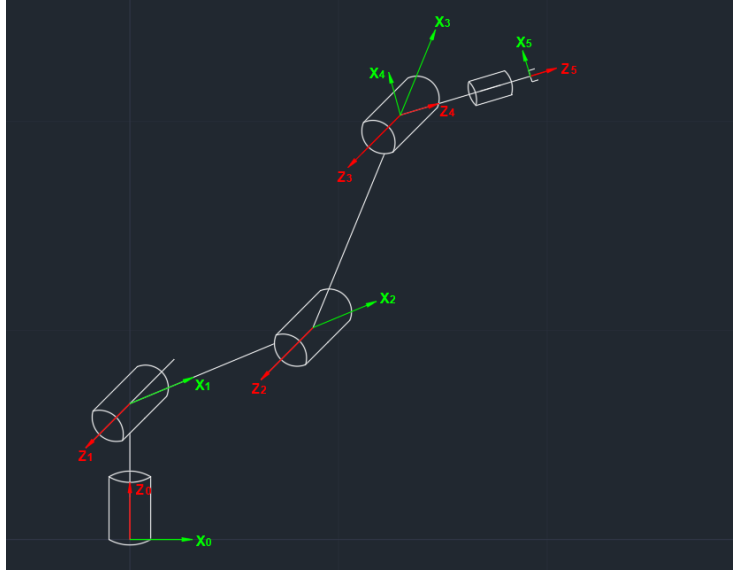


Figure 1: Lynxmotion robot diagram

$${}^3T_4 = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$${}^4T_5 = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where

$$a_2 = 9.5[cm]$$

$$a_3 = 11[cm]$$

$$d_1 = 6.5[cm]$$

$$d_5 = 3.2[cm]$$

Every matrix represents the orientation and displacement of every frame with respect to the previous reference frame. In order to derive the forward kinematics of the robots it is necessary to know the orientation and position of the end effector, the frame at the point O_5 , with respect to the main reference frame which is the frame with origin at point O_0 .

In order to know the position and orientation of the end effector after several rotations has been performed, due to the movement of the joints, it is necessary to use the relationship

$${}^0T_5 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 \quad (6)$$

After performing the matrix product, the resultant matrix will be the homogeneous matrix that represents the orientation and position of the frame attached at the end effector, and it will have the following form:

$${}^0T_5 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & {}^0Ox_5 \\ r_{21} & r_{22} & r_{23} & {}^0Oy_5 \\ r_{31} & r_{32} & r_{33} & {}^0Oz_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7a)$$

$$= \begin{bmatrix} R & {}^0O_5 \\ 0 & 1 \end{bmatrix} \quad (7b)$$

where R is the rotation matrix of the fifth frame with respect to main frame and 0O_5 is the coordinate vector of point O_5 with respect to main frame. This point is also the desired point of the end effector. This matrix is a function of the variables $\theta_1 \dots \theta_5$.

$${}^0O_5 = \begin{bmatrix} {}^0Ox_5 \\ {}^0Oy_5 \\ {}^0Oz_5 \end{bmatrix} = \begin{bmatrix} {}^0P_x \\ {}^0P_y \\ {}^0P_z \end{bmatrix} = {}^0P$$

To perform the computing of the forward kinematics of the Lynxmotion it was created a function in Matlab that receives as input the vector $Q = [\theta_1 \theta_2 \theta_3 \theta_4 \theta_5]^T$ and it gives the position 0P and orientation R of the end effector as output. The code of the function can be found in appendix A.1.

1.2 Workspace

To have a representation of the workspace, it was necessary to measure the physical operational range of every joint in order to establish the domain of every variable θ_i , the results are in table 2.

Variable	Operational range [°]
θ_1	[-90, 90]
θ_2	[0, 135]
θ_3	[-135, 30]
θ_4	[0, 180]
θ_5	[-90, 90]

Table 2: Operational range of joint variables

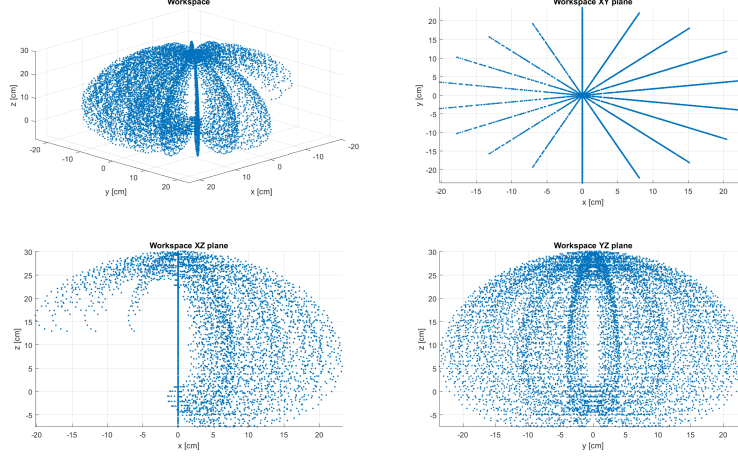


Figure 2: Lynxmotion robot workspace

Once the range of the variables has been determined, it can be simulated the movement of every joint in its corresponding operational range to create a representation of the reachable 3D space. The space generated by all the points that can be reached by the end effector of the robot is called workspace. The used code to perform the simulation can be found in appendix A.2.

The workspace of the Lynxmotion robot is depicted in figure 2, the picture shows a 3D view of the space, while the other three view shows the planes YX, ZX and ZY respectively.

1.3 Inverse kinematics

The concept of inverse kinematics is finding joint valuables given the specific position and orientation of the end effector. Concretely, there are five joints valuables need to be worked out, namely $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$.

1.3.1 Finding θ_1

Since the x and y position of the end effector are only decided by the rotation of joint 1, we can project the manipulator into the horizontal plane of the frame 0. Therefore, we can derive the expression of θ_1 as

$$\theta_1 = \text{atan2}(Py, Px) \quad (8)$$

1.3.2 Finding θ_2, θ_3

In order to simplify the configuration, we can extract link 2 and link 3 from the whole manipulator, hence the link 2 and link 3 would be a planar mechanism. The next step is to find the origin coordinates of frame 3 with respect to frame 1, namely 1O_3 , which can be derived from the matrix of 1T_3 . Furthermore, 1T_3 can be expressed as the product

$$\begin{aligned} {}^0T_1 {}^1T_3 &= {}^0T_3 \\ ({}^0T_1)^{-1} {}^0T_1 {}^1T_3 &= ({}^0T_1)^{-1} {}^0T_3 \\ {}^1T_3 &= {}^1T_0 {}^0T_3 \end{aligned} \quad (9)$$

According to the equation (9), we need to get the matrixes 1T_0 and 0T_3 respectively. 1T_0 is easy to get from the inverse matrix of 0T_1 , which should be

$$({}^0T_1)^{-1} = {}^1T_0 = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ \sin(\theta_1) & -\cos(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

As for 0T_3 , we only focus on the displacement part 0O_3 (the last column of 0T_3).

$${}^0T_3 = \begin{bmatrix} \cdot & \cdot & \cdot & {}^0O_{3x} \\ \cdot & \cdot & \cdot & {}^0O_{3y} \\ \cdot & \cdot & \cdot & {}^0O_{3z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Furthermore, Z_5 is aligned with link 4 whose length is d_5 , so 0O_3 can be calculated by

$$\begin{aligned} {}^0O_3 &= {}^0P - d_5 R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} {}^0O_{3x} \\ {}^0O_{3y} \\ {}^0O_{3z} \end{bmatrix} &= \begin{bmatrix} {}^0P_x \\ {}^0P_y \\ {}^0P_z \end{bmatrix} - d_5 \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} \end{aligned} \quad (12)$$

Then, 1T_3 can be computed by substituting (10), (11) and (12) in (9). Lastly, as only the fourth column is needed at this moment, equation (13) only shows the last column for simplicity.

$${}^1T_3 = \begin{bmatrix} \cdot & \cdot & \cdot & {}^0O_{3x} \cos(\theta_1) + {}^0O_{3y} \sin(\theta_1) \\ \cdot & \cdot & \cdot & {}^0O_{3z} - d_1 \\ \cdot & \cdot & \cdot & {}^0O_{3x} \sin(\theta_1) - {}^0O_{3y} \cos(\theta_1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Therefore, we found the coordinates of frame 3 with respect to frame 1 (namely 1O_3), as denoted in equation (14). Then the solutions of θ_2 and θ_3 can be

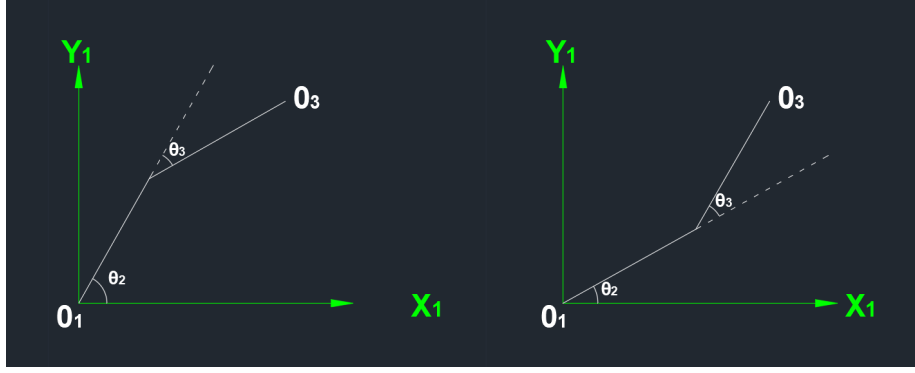


Figure 3: Graphical representation of the two solutions for θ_2 and θ_3 , namely elbow left and elbow right

worked out by the geometrical method mentioned in the lecture. There are 2 solutions as a result of redundancy, which is shown below in figure 3.

$${}^1O_3 = \begin{bmatrix} {}^1O_{3x} \\ {}^1O_{3y} \\ {}^1O_{3z} \end{bmatrix} = \begin{bmatrix} {}^0O_{3x} \cos(\theta_1) + {}^0O_{3y} \sin(\theta_1) \\ {}^0O_{3z} - d_1 \\ {}^0O_{3x} \sin(\theta_1) - {}^0O_{3y} \cos(\theta_1) \end{bmatrix} \quad (14)$$

Applying the law of cosines and using the coordinates of point O_3 with respect to O_1 , we can derive the expression of θ_3 that

$$\cos(\theta_3) = \frac{{}^1O_{3x}^2 + {}^1O_{3y}^2 - a_2^2 - a_3^2}{2a_2a_3}$$

while sine will have two possible values, a positive and a negative value, which will lead to the two solutions depicted above.

$$\sin(\theta_3) = \pm \sqrt{1 - \cos(\theta_3)^2}$$

$$\theta_3 = \begin{cases} \theta_{3_1} & \text{for } \sin(\theta_3)^+ \\ \theta_{3_2} & \text{for } \sin(\theta_3)^- \end{cases}$$

$$\theta_3 = \text{atan2}(\sin(\theta_3), \cos(\theta_3)) \quad (15)$$

Finally, θ_2 can be computed from the angle of 1O_3 once that θ_3 is known.

$$\theta_2 = \text{atan2}({}^1O_{3y}, {}^1O_{3x}) - \text{atan2}(\sin(b), \cos(b)) \quad (16)$$

where

$$\sin(b) = \frac{a_3 \sin(\theta_3)}{\sqrt{{}^1O_{3x}^2 + {}^1O_{3y}^2}}$$

Accordingly with the dependency of θ_2 on the value of θ_3 , this variable will also have two solutions. The set of two solutions for both variables, creates the two possible configurations, elbow right for θ_3^+ and elbow left for θ_3^- , as depicted in figure 3.

1.3.3 Finding θ_4

The algebraic was used method to work out θ_4 . The core idea is building an equation of θ_4 from the element of transformation matrix 3T_5 . Now, $\theta_1, \theta_2, \theta_3$ are already known. Combined with 0T_5 , 3T_5 can be derived from

$$\begin{aligned} {}^0T_3 {}^3T_5 &= {}^0T_5 \\ ({}^0T_3)^{-1} {}^0T_3 {}^3T_5 &= ({}^0T_3)^{-1} {}^0T_5 \\ {}^3T_5 &= {}^3T_0 {}^0T_5 \end{aligned} \quad (17)$$

where

$${}^3T_0 = {}^3T_2 {}^2T_1 {}^1T_0 \quad (18)$$

Each element on the right side of equation 17 depends on θ_1, θ_2 and θ_3 which are already known. It is important to remark that there exists two versions of 3T_2 and 2T_1 as there are two solutions. Therefore, 3T_5 will have two versions, both with a form like this:

$${}^3T_5 = \begin{bmatrix} \cdot & \cdot & \cdot & {}^3O_{5x} \\ \cdot & \cdot & \cdot & {}^3O_{5y} \\ \cdot & \cdot & \cdot & {}^3O_{5z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

On the other hand, another way to get 3T_5 with variable θ_4 is as a product of matrix (4) and matrix (5), which is

$$\begin{aligned} {}^3T_5 &= {}^3T_4 {}^4T_5 = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cdot & \cdot & \cdot & d_5 \sin(\theta_4) \\ \cdot & \cdot & \cdot & -d_5 \cos(\theta_4) \\ \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (20)$$

Thus, 3T_5 has been derived in two ways, which can be used to create an equation system to solve for θ_4 . By equalising the last column of (19) and (20) we obtain:

$$\sin(\theta_4) = \frac{{}^3O_{5x}}{d_5} \quad \cos(\theta_4) = -\frac{{}^3O_{5y}}{d_5} \quad (21)$$

therefore

$$\theta_4 = \text{atan2}(\sin(\theta_4), \cos(\theta_4)) \quad (22)$$

As a result of the two versions of (19), θ_4 will also have two solutions that must be computed in the same way for both configurations.

1.3.4 Finding θ_5

The method of calculating θ_5 is the similar to finding θ_4 . we try to get the matrix 4T_5 from two ways. The one with variable θ_5 is obviously expressed in equation 5.

Another means of representing 4T_5 is

$${}^4T_5 = {}^4T_0 {}^0T_5 \quad (23)$$

where

$${}^4T_0 = {}^4T_3 {}^3T_0$$

Therefore, the equation (23) depends on the inverse of (4), (11) and 0T_5 which is given as the desired position and orientation.

It seems that θ_5 should also have two solutions since $\sin(\theta_4)$ and $\cos(\theta_4)$ have 2 pairs of solutions and the expression of 4T_0 contains $\sin(\theta_4)$ and $\cos(\theta_4)$. Whereas, it turns out that 4T_0 only have one value due to the same computing result of 2 pairs of $\sin(\theta_4)$ and $\cos(\theta_4)$. For simplicity, we use (a,b) to represent each element of 4T_5 (a,b are the number of rows and columns). Thus, $\sin(\theta_5)$ and $\cos(\theta_5)$ can be written as

$$\sin(\theta_5) = {}^4T_5(2, 1) \quad \cos(\theta_5) = {}^4T_5(1, 1) \quad (24)$$

Hence, the solutions of θ_5 is

$$\theta_5 = \text{atan2}(\sin(\theta_5), \cos(\theta_5)) \quad (25)$$

1.4 Task plan

The chosen task was to imagine the problem of taking three objects and stick them to a wall, this might be considered as a pick and place task. In this case the robot should take an object and take it to its final position, then take the second object and place it in the second position and finally perform the same process for the last object.

The points were chosen by inspecting the workspace and placing them in the workspace of the robot. They can be seen in the chart of figure 4.

1.5 Forward kinematics test

In order to test the forward kinematics po

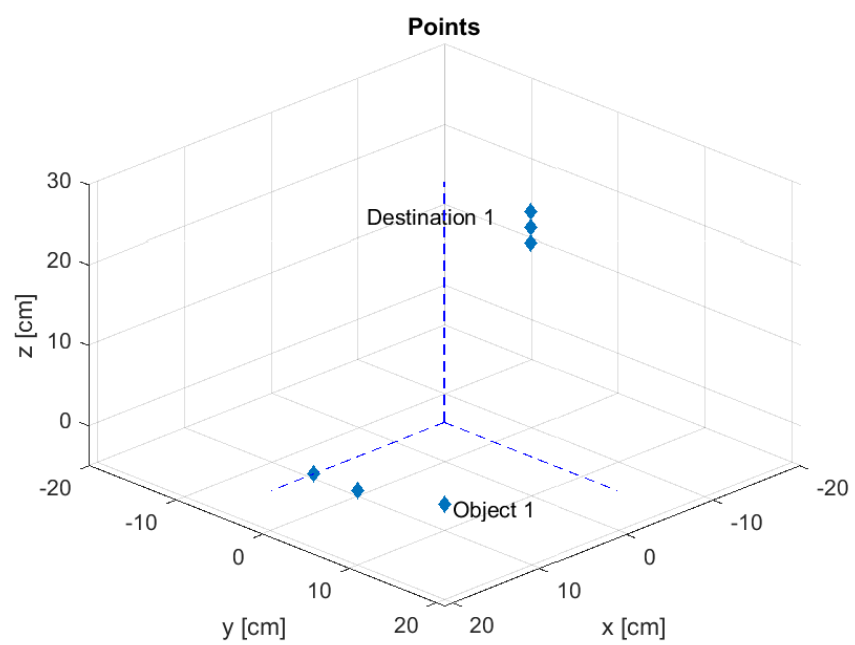
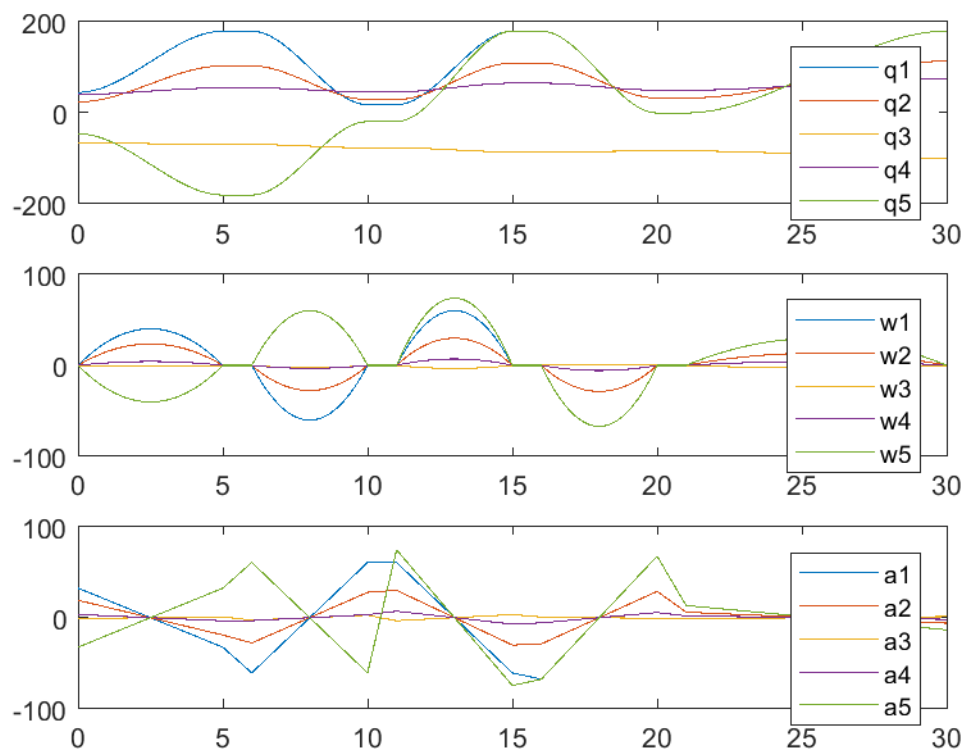


Figure 4: Chosen points for the pick and place task

1.6 Inverse kinematics test and animation

1.7 Trajectories



2 Part 2

2.1 This is a subsection

A Matlab codes

A.1 Forward kinematics (LynxFK function)

```
function [ P , R , O ] = LynxFK( Q )
%LYNXFK Forward kinematics of Lynxmotion robot
% This function computes the forward kinematics of the Lynxmotion robot
% given a row vector Q containing the values of the five joints of the
% robot in degrees.
% The return value is the Position (P) and Orientation (R) of the end
```

```

% effector, also a vector O(6,3) is returned. Every row (i) represent the
% coordinates of point Oi=[Oxi Oyi Ozi] with respecto to main frame.

a=[0 9.5 11 0 0]; %Parameter a of DH table for Lynxmotion robot
alpha=[90 0 0 90 0]; %Parameter alpha of DH table for Lynxmotion robot
d=[6.5 0 0 0 3.2]; %Parameter d of DH table for Lynxmotion robot
theta=[Q(1) Q(2) Q(3) Q(4) Q(5)]; %Joint values Lynxmotion robot

N=5; %Number of joints
H=eye(4); %Initialize homogeneous resultant matrix
O=zeros(N+1, 3);
for n=1:N
    table(n,:)=[
        a(n) alpha(n) d(n) theta(n)
    ];
    T(:, :, n)=dh2td(table(n,:)); %Compute the T matrix of row n
    H=H*T(:, :, n); %Multiply the T matrix to create H_On

    O(n+1,1)=H(1,4); %Store Oxi
    O(n+1,2)=H(2,4); %Store Oyi
    O(n+1,3)=H(3,4); %Store Ozi
end

P=H(1:3, 4); %Store Position of end effector
R=H(1:3, 1:3); %Store Orientation of end effector

end

```

A.2 Workspace (workspace script)

```

clear all
clc

%% Variables initialization
samples=10;
WSx=[];
WSy=[];
WSz=[];
Q1=linspace(-90,90,samples); %[-90, 90]
Q2=linspace(0,135,samples); %[0, 135]
Q3=linspace(-135,30,samples); %[-135, 30]
Q4=linspace(0,180,samples); %[0, 180]
%Q5=linspace(-90,90,samples); %[-90, 90]
Q5=0;

```

```

%% For-loop iteration for every variable
for q1=Q1
    for q2=Q2
        for q3=Q3
            for q4=Q4
                for q5=Q5
                    theta=[q1 q2 q3 q4 q5];

                    [P R O]=LynxFK(theta);

                    WSx=cat(1,WSx,P(1));
                    WSy=cat(1,WSy,P(2));
                    WSz=cat(1,WSz,P(3));
                end
            end
        end
    end
end

%% Plot the workspace, in 4 different views
fig = figure (1)
%3D view
subplot(2,2,1)
scatter3(WSx, WSy, WSz, '.');
xlabel('x [cm]') ; ylabel('y [cm]') ; zlabel('z [cm]') ;
axis([min(WSx) max(WSx) min(WSy) max(WSy) min(WSz) max(WSz)])
view([1, 1, 1])
title('Workspace')
%2D XY
subplot(2,2,2)
scatter3(WSx, WSy, WSz, '.');
xlabel('x [cm]') ; ylabel('y [cm]') ; zlabel('z [cm]') ;
axis([min(WSx) max(WSx) min(WSy) max(WSy) min(WSz) max(WSz)])
view([0, 0, 1])
title('Workspace XY plane')
%2D XZ
subplot(2,2,3)
scatter3(WSx, WSy, WSz, '.');
xlabel('x [cm]') ; ylabel('y [cm]') ; zlabel('z [cm]') ;
axis([min(WSx) max(WSx) min(WSy) max(WSy) min(WSz) max(WSz)])
view([0, -1, 0])
title('Workspace XZ plane')
%2D YZ
subplot(2,2,4)
scatter3(WSx, WSy, WSz, '.');
xlabel('x [cm]') ; ylabel('y [cm]') ; zlabel('z [cm]') ;

```

```
axis([min(WSx) max(WSx) min(WSy) max(WSy) min(WSz) max(WSz)])  
view([1, 0, 0])  
title('Workspace YZ plane')  
  
print(fig, 'Workspace', '-dpng')  
print(fig, 'Workspace', '-depsc')
```