Introduction to Robotics

Lab Instructor: Miss Sadaf Sheikh
Group Members:
Course Instructor: Dr Basit Memon

LAB 07-FORWARD KINEMATICS

March 29, 2024

Contents

1 Task 5.1 DH Frame Assignment (15 points)

1.1 Using standard DH convention, assign DH frames to the robot arm in Figure 5.1. Make sure to clearly indicate the z and x axes, and the origin of each frame; drawing the y axis is optional. Place the origin of the end-effector frame at the center of gripper motor horn, for convenience of measurements in upcoming tasks. Draw and paste each frame's z and x-axis on the motor or link bodies of the robot. This will help your visualization in later tasks

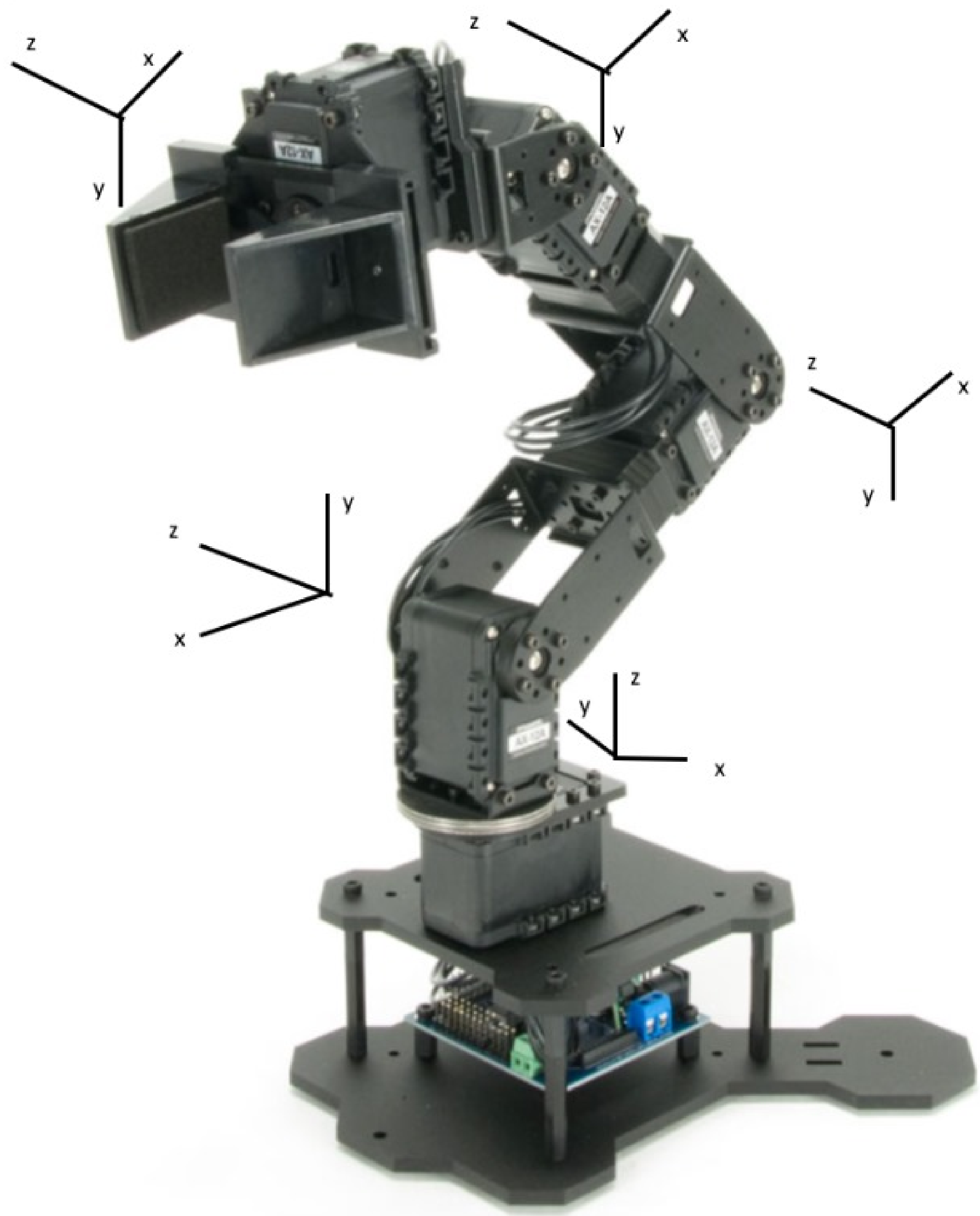

Figure 1: Labelled DH frame assignments

1.2 Task 5.2 DH Parameters

Frame	a_i	α_i	d_i	θ_i
1	0	$\frac{\pi}{2}$	54	θ_1
2	108	0	0	θ_2
3	108	0	0	θ_3
4	76	0	0	θ_4

Table 1: DH Parameters

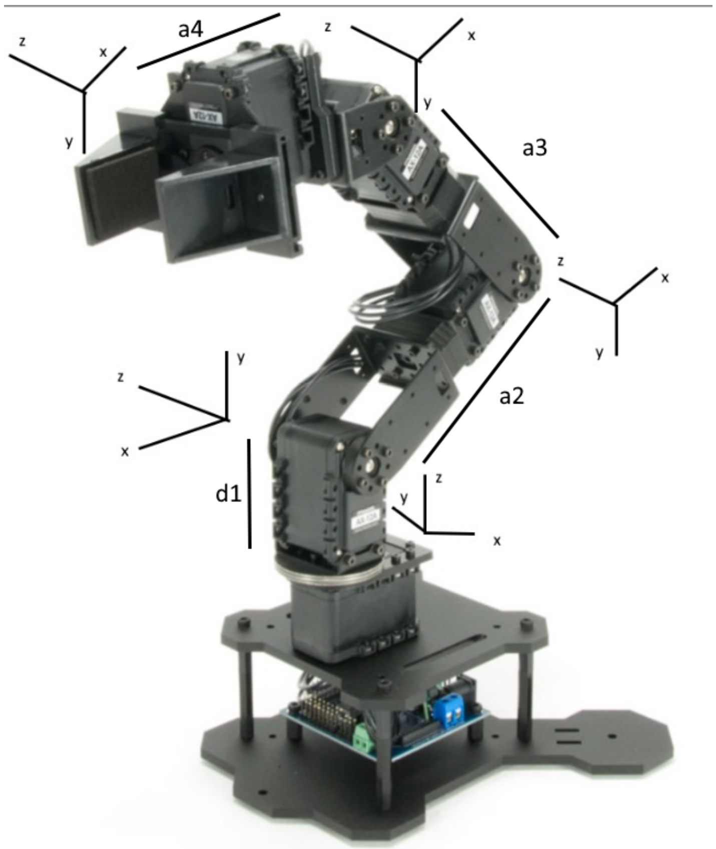

Figure 2: Labelled parameters

1.3 Task 5.3 Homogeneous Transformations (5+5+3 points)

```
syms theta;
syms alpha;
syms a;
syms d;
syms theta1 theta2 theta3 theta4;
syms d1 d2 d3 d4;
syms a1 a2 a3 a4
theta=[theta1,theta2,theta3,theta4]
alpha=[sym(pi/2),0,0,0]
d=[54,0,0,0]
a=[0,108,108,76]
T04=eye(4,4);
for i=1:4;
    T=[cos(theta(i)), -sin(theta(i))*cos(alpha(i)), sin(theta(i))*sin(alpha(i)), a(i)*cos(theta(i));
    sin(theta(i)), cos(theta(i))*cos(alpha(i)), -cos(theta(i))*sin(alpha(i)), a(i)*sin(theta(i));
    0, sin(alpha(i)), cos(alpha(i)), d(i);
    0, 0, 0, 1]
    T=vpa(T,3)
    T04=vpa(T04*T,3)
end

|
expand(T04)
simplify(T04)
T04=vpa(T04,3)

position=T04(1:3, 4)
orientation=T04(1:3, 1:3)
```

Figure 3: Calculating T04

```
T04 =
```

$$\begin{pmatrix} -1.0\cos(\theta_4)\,\sigma_3 - 1.0\sin(\theta_4)\,\sigma_4 & 1.0\sin(\theta_4)\,\sigma_3 - 1.0\cos(\theta_4)\,\sigma_4 & 1.0\sin(\theta_1) & 108.0\cos(\theta_1)\cos(\theta_2) - 76.0\cos(\theta_4)\,\sigma_3 - 76.0\sin(\theta_4)\,\sigma_4 - 108.0\cos(\theta_1)\sin(\theta_2)\sin(\theta_3) + 108.0\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) \\ -1.0\cos(\theta_4)\,\sigma_1 - 1.0\sin(\theta_4)\,\sigma_2 & 1.0\sin(\theta_4)\,\sigma_1 - 1.0\cos(\theta_4)\,\sigma_2 & -1.0\cos(\theta_1) & 108.0\cos(\theta_2)\sin(\theta_1) - 76.0\cos(\theta_4)\,\sigma_1 - 76.0\sin(\theta_4)\,\sigma_2 - 108.0\sin(\theta_1)\sin(\theta_2)\sin(\theta_3) + 108.0\cos(\theta_2)\cos(\theta_3)\sin(\theta_1) \\ \cos(\theta_4)\,\sigma_6 + \sin(\theta_4)\,\sigma_5 & \cos(\theta_4)\,\sigma_5 - 1.0\sin(\theta_4)\,\sigma_6 & 0 & 108.0\sin(\theta_2) + 108.0\cos(\theta_2)\sin(\theta_3) + 108.0\cos(\theta_3)\sin(\theta_2) + 76.0\cos(\theta_4)\,\sigma_6 + 76.0\sin(\theta_4)\,\sigma_5 + 54.0 \\ 0 & 0 & 0 & 1.0 \end{pmatrix}$$

where

$$\sigma_1 = 1.0\sin(\theta_1)\sin(\theta_2)\sin(\theta_3) - 1.0\cos(\theta_2)\cos(\theta_3)\sin(\theta_1)$$

$$\sigma_2 = 1.0\cos(\theta_2)\sin(\theta_1)\sin(\theta_3) + 1.0\cos(\theta_3)\sin(\theta_1)\sin(\theta_2)$$

$$\sigma_3 = 1.0\cos(\theta_1)\sin(\theta_2)\sin(\theta_3) - 1.0\cos(\theta_1)\cos(\theta_2)\cos(\theta_3)$$

$$\sigma_4 = 1.0\cos(\theta_1)\cos(\theta_2)\sin(\theta_3) + 1.0\cos(\theta_1)\cos(\theta_3)\sin(\theta_2)$$

$$\sigma_5 = 1.0\cos(\theta_2)\cos(\theta_3) - 1.0\sin(\theta_2)\sin(\theta_3)$$

$$\sigma_6 = 1.0\cos(\theta_2)\sin(\theta_3) + 1.0\cos(\theta_3)\sin(\theta_2)$$

Figure 4: Final T04

position =

$$\begin{pmatrix} 108.0\cos(\theta_1)\cos(\theta_2) - 76.0\cos(\theta_4)\,(1.0\cos(\theta_1)\sin(\theta_2)\sin(\theta_3) - 1.0\cos(\theta_1)\cos(\theta_2)\cos(\theta_3)) - 76.0\sin(\theta_4)\,(1.0\cos(\theta_1)\cos(\theta_2)\sin(\theta_3) + 1.0\cos(\theta_1)\cos(\theta_3)\sin(\theta_2)) - 108.0\cos(\theta_1)\sin(\theta_2)\sin(\theta_3) + 108.0\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) \\ 108.0\cos(\theta_2)\sin(\theta_1) - 76.0\cos(\theta_4)\,(1.0\sin(\theta_1)\sin(\theta_2)\sin(\theta_3) - 1.0\cos(\theta_2)\cos(\theta_3)\sin(\theta_1)) - 76.0\sin(\theta_4)\,(1.0\cos(\theta_2)\sin(\theta_1)\sin(\theta_3) + 1.0\cos(\theta_3)\sin(\theta_1)\sin(\theta_2)) - 108.0\sin(\theta_1)\sin(\theta_2)\sin(\theta_3) + 108.0\cos(\theta_2)\cos(\theta_3)\sin(\theta_1) \\ 108.0\sin(\theta_2) + 108.0\cos(\theta_2)\sin(\theta_3) + 108.0\cos(\theta_3)\sin(\theta_2) + 76.0\cos(\theta_4)\,(1.0\cos(\theta_2)\sin(\theta_3) + 1.0\cos(\theta_3)\sin(\theta_2)) + 76.0\sin(\theta_4)\,(1.0\cos(\theta_2)\cos(\theta_3) - 1.0\sin(\theta_2)\sin(\theta_3)) + 54.0 \end{pmatrix}$$

orientation =

$$\begin{pmatrix} -1.0\cos(\theta_4)\,\sigma_3 - 1.0\sin(\theta_4)\,\sigma_4 & 1.0\sin(\theta_4)\,\sigma_3 - 1.0\cos(\theta_4)\,\sigma_4 & 1.0\sin(\theta_1) \\ -1.0\cos(\theta_4)\,\sigma_1 - 1.0\sin(\theta_4)\,\sigma_2 & 1.0\sin(\theta_4)\,\sigma_1 - 1.0\cos(\theta_4)\,\sigma_2 & -1.0\cos(\theta_1) \\ \cos(\theta_4)\,\sigma_6 + \sin(\theta_4)\,\sigma_5 & \cos(\theta_4)\,\sigma_5 - 1.0\sin(\theta_4)\,\sigma_6 & 0 \end{pmatrix}$$

where

$$\sigma_1 = 1.0\sin(\theta_1)\sin(\theta_2)\sin(\theta_3) - 1.0\cos(\theta_2)\cos(\theta_3)\sin(\theta_1)$$

$$\sigma_2 = 1.0\cos(\theta_2)\sin(\theta_1)\sin(\theta_3) + 1.0\cos(\theta_3)\sin(\theta_1)\sin(\theta_2)$$

$$\sigma_3 = 1.0\cos(\theta_1)\sin(\theta_2)\sin(\theta_3) - 1.0\cos(\theta_1)\cos(\theta_2)\cos(\theta_3)$$

$$\sigma_4 = 1.0\cos(\theta_1)\cos(\theta_2)\sin(\theta_3) + 1.0\cos(\theta_1)\cos(\theta_3)\sin(\theta_2)$$

$$\sigma_5 = 1.0\cos(\theta_2)\cos(\theta_3) - 1.0\sin(\theta_2)\sin(\theta_3)$$

$$\sigma_6 = 1.0\cos(\theta_2)\sin(\theta_3) + 1.0\cos(\theta_3)\sin(\theta_2)$$

Figure 5: Position & Orientation

1.4 Task 5.4: FK Function

The expressions of position and expressions obtained in the previous part, is substituted in as x,y,z and R in the following function.

```
jointAngles=[sym(pi/3) sym(pi/3) sym(pi/3) sym(pi/3)]
[x,y,z,R]=pincherFK(jointAngles)
```

```
function [x,y,z,R] = pincherFK(jointAngles)
    syms theta1 theta2 theta3 theta4
    theta1_val=jointAngles(1); theta2_val=jointAngles(2); theta3_val=jointAngles(3); theta4_val=jointAngles(4);
    x=108.0*cos(theta1)*cos(theta2) - 76.0*cos(theta4)*(1.0*cos(theta1)*sin(theta2)*sin(theta3) - 1.0*cos(theta1)*cos(theta2)*
    y=108.0*cos(theta2)*sin(theta1) - 76.0*cos(theta4)*(1.0*sin(theta1)*sin(theta2)*sin(theta3) - 1.0*cos(theta2)*cos(theta3)*
    z=108.0*sin(theta2) + 108.0*cos(theta2)*sin(theta3) + 108.0*cos(theta3)*sin(theta2) + 76.0*cos(theta4)*(1.0*cos(theta2)*si
    R=[- 1.0*cos(theta4)*(1.0*cos(theta1)*sin(theta2)*sin(theta3) - 1.0*cos(theta1)*cos(theta2)*cos(theta3)) - 1.0*sin(theta4)

    x=simplify(subs(x,[theta1 theta2 theta3 theta4], [jointAngles(1) theta2_val theta3_val theta4_val]));
    y=simplify(subs(y,[theta1 theta2 theta3 theta4], [theta1_val theta2_val theta3_val theta4_val]));
    z=simplify(subs(z,[theta1 theta2 theta3 theta4], [theta1_val theta2_val theta3_val theta4_val]));
    R=simplify(subs(R,[theta1 theta2 theta3 theta4], [theta1_val theta2_val theta3_val theta4_val]));

end
```

Figure 6: FK Function

```
jointAngles =
```

$$\left(\frac{\pi}{3} \quad \frac{\pi}{3} \quad \frac{\pi}{3} \quad \frac{\pi}{3}\right)$$

x = -38

y = $-38\sqrt{3}$

z = $108\sqrt{3}+54$

R =

$$\begin{pmatrix} -\frac{1}{2} & 0 & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & 0 & -\frac{1}{2} \\ 0 & -1 & 0 \end{pmatrix}$$

Figure 7: Result of the fk function

1.5 Task 5.5: Verification of Forward Kinematic Mapping

Following is the result obtain from the PincherModel code provided, as seen: we get the same results as our FK function.

```
The position of end-effector is:
X: -38
Y: -65.8179
Z: 241.0615

R:

ans =

   -0.5000   -0.0000    0.8660
   -0.8660   -0.0000   -0.5000
    0.0000   -1.0000    0.0000


The orientation angle is given with respect to the x-axis of joint 2:
Angle: 180 degrees.
```

Figure 8: Result of the PincherModel

1.6 5.6: DH and Servo Joint Angles alignment

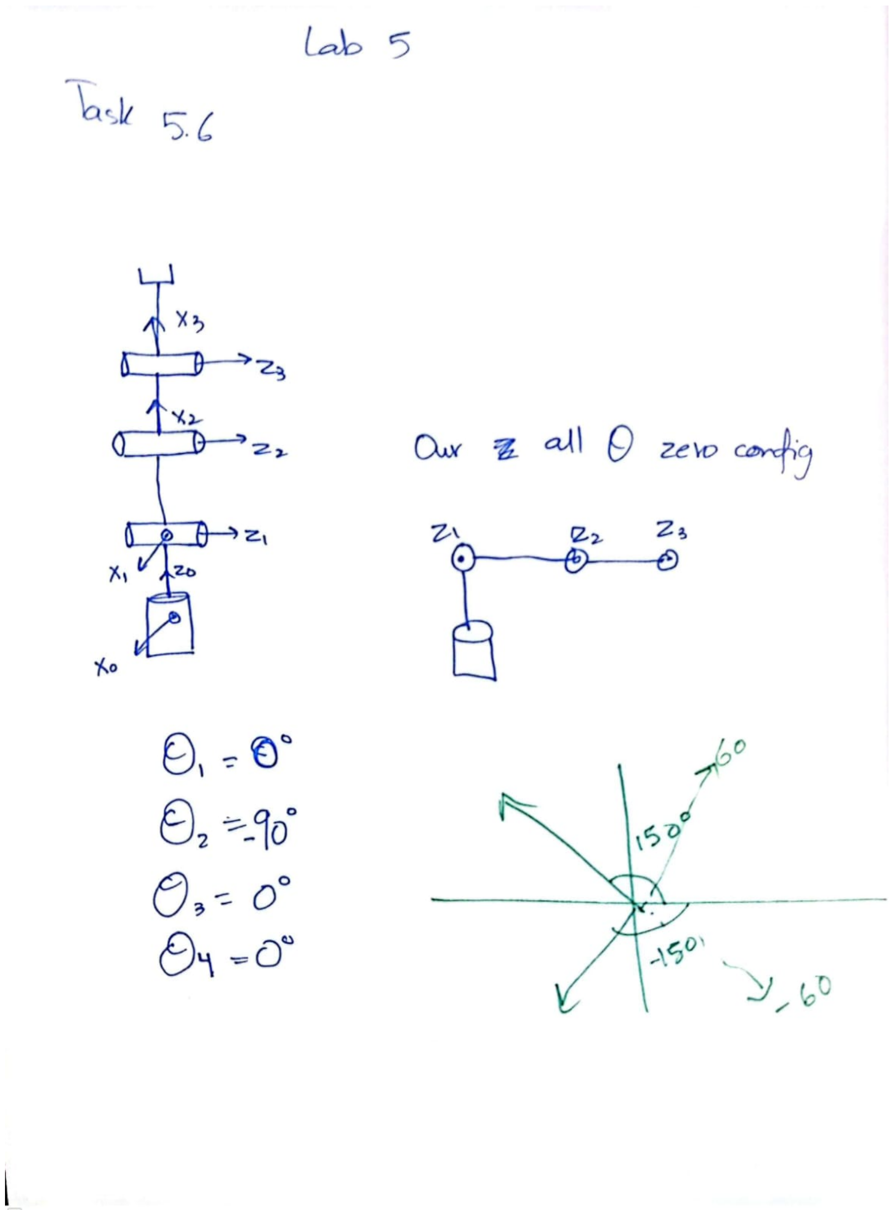

Figure 9: Enter Caption

When we set all our θ_i equal to zero, we get the configuration on the right side of the image. We have set our xo same as x1, which makes our theta 1 aligned.

The all theta i =0 position of the

Table 2: Linear mapping between servo angles and DH angles

Joint ID	$\boldsymbol{\theta_i}$	$\boldsymbol{\psi_i}$	**Aligned Directions of Rotation (Yes/No)**
1	$0°$	0	No
2	$0°$	-90	No
3	$0°$	0	No
4	$0°$	0	No

Table 3: Joint Limits

Joint ID	Minimum Joint Angle		Maximum Joint Angle	
	Servo angle	DH Joint Angle	Servo angle	DH Joint Angle
1	$-150°$	$-150°$	$150°$	$150°$
2	$-150°$	$-60°$	$150°$	$240°$
3	$-150°$	$-150°$	$150°$	$150°$
4	$-150°$	$-150°$	$150°$	$150°$

1.7 Task 5.7: Mapping servo angles to DH angles

```
function dhJointAngles= servo2dh(jointAngles)
jointAngles(2)=jointAngles(2)+1.5708;
end
```

1.8 Task 5.8: Identifying reachable workspace

```
syms theta1 theta2 theta3 theta4 theta5 theta6 % Define symbolic variables

% DH parameters
DH_params = [0, -pi/2, 162.5, theta1; ...
             425, 0, 0, theta2; ...
             392.2, 0, 0, theta3; ...
             0, pi/2, 133, theta4; ...
             0, -pi/2, 99.7, theta5; ...
             0, 0, 60, theta6];

% Parameter ranges
step_theta1 = linspace(0, 360, 50) * pi/180;
step_theta2 = linspace(0, 360, 40) * pi/180;
step_theta3 = linspace(0, 360, 30) * pi/180;
step_theta4 = linspace(0, 360, 20) * pi/180;
step_theta5 = linspace(0, 360, 10) * pi/180;
step_theta6 = linspace(0, 360, 5) * pi/180;

% Create a map of symbolic variables to their respective step ranges
mapping = containers.Map({'theta1', 'theta2', 'theta3', 'theta4', 'theta5', 'theta6'}, ...
                         {step_theta1, step_theta2, step_theta3, step_theta4, step_theta5, step_theta6});

% Workspace plotting function + timing
plot3dworkspace(DH_params, mapping, @get_alternative_dh_transform, true);

% Function to compute alternative DH transform


function out = arr2Rad(A)
    out = arrayfun(@(angle) deg2rad(angle), A);
end

function T = get_alternative_dh_transform(a, alpha, d, theta)
    T = [cos(theta), -cos(alpha)*sin(theta), sin(alpha)*sin(theta), a*cos(theta);
         sin(theta), cos(alpha)*cos(theta), -sin(alpha)*cos(theta), a*sin(theta);
         0, sin(alpha), cos(alpha), d;
         0,0,0,1];
end
```

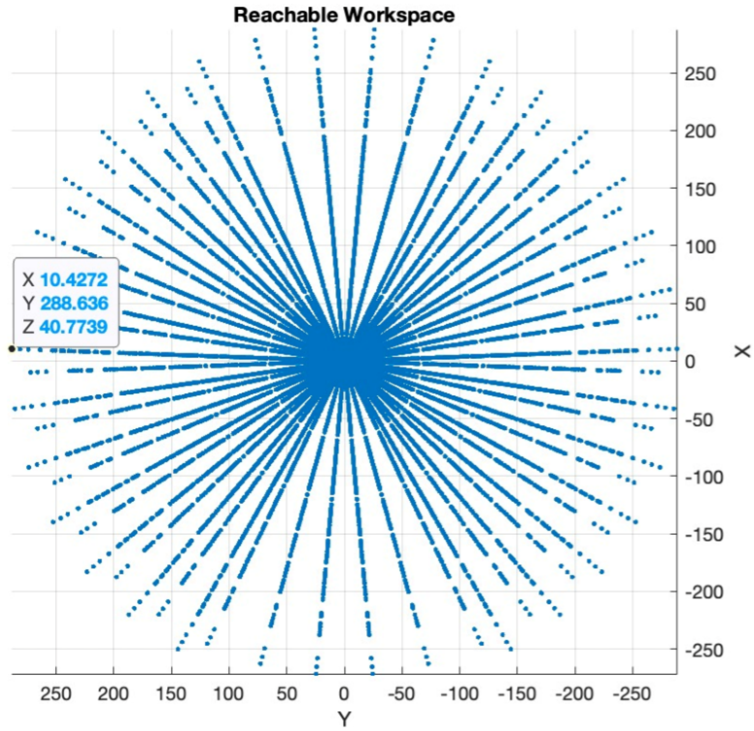

Figure 10: Reachable Workspace

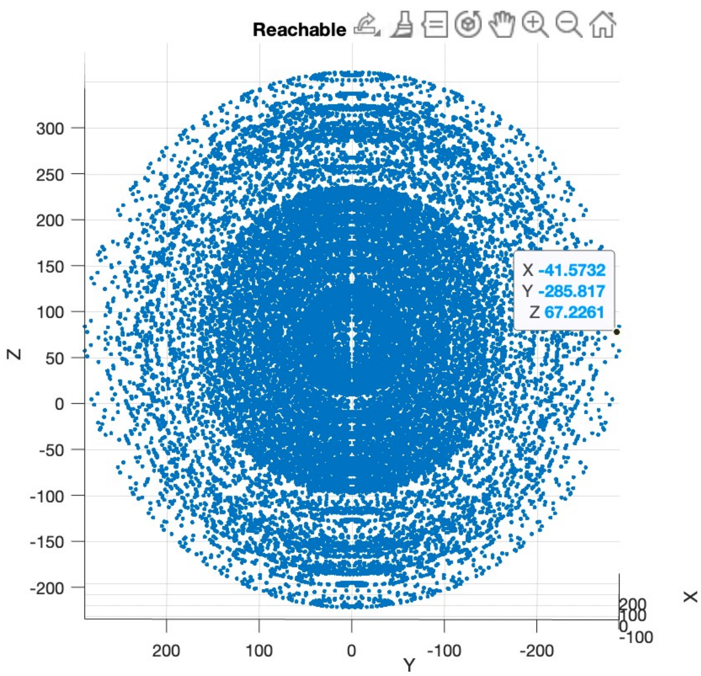

Figure 11: Reachable Workspace

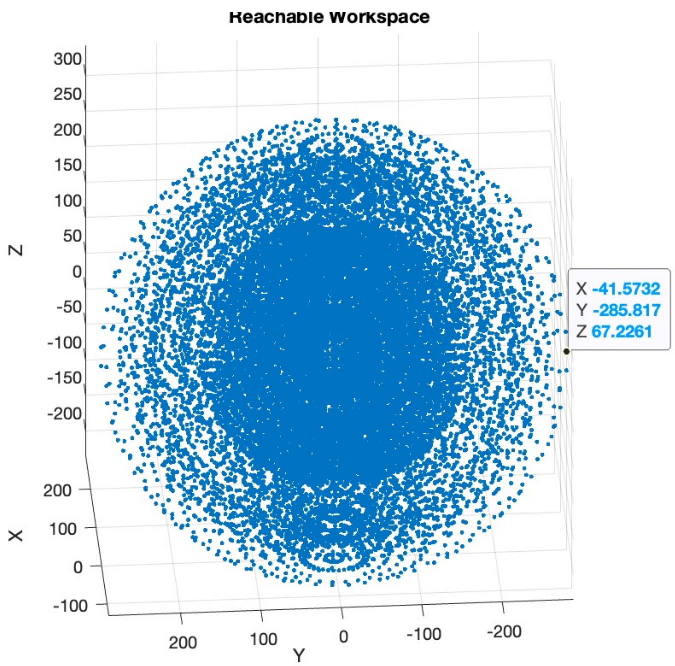

Figure 12: Reachable Workspace

1.9 Task 5.9: Communicating with motors

```
JointAngles=arb.getpos()
servo2dh(JointAngles)
syms theta;
syms alpha;
syms a;
syms d;
syms theta1 theta2 theta3 theta4;
syms d1 d2 d3 d4;
syms a1 a2 a3 a4
theta=[JointAngles(1),JointAngles(2)+1.5708,JointAngles(3),JointAngles(4)]
alpha=[sym(pi/2),0,0,0]
d=[54,0,0,0]
a=[0,108,108,76]
T04=eye(4,4);
for i=1:4;
    T=[cos(theta(i)), -sin(theta(i))*cos(alpha(i)), sin(theta(i))*sin(alpha(i)), a(i)*cos(theta(i));
    sin(theta(i)), cos(theta(i))*cos(alpha(i)), -cos(theta(i))*sin(alpha(i)), a(i)*sin(theta(i));
    0, sin(alpha(i)), cos(alpha(i)), d(i);
    0, 0, 0, 1]
    T=vpa(T,3)
    T04=vpa(T04*T,3)
end
```

```
expand(T04)
simplify(T04)
T04=vpa(T04,3)

position=T04(1:3, 4)
orientation=T04(1:3, 1:3)
% endeff=[-38 -38*sqrt(3) 241.06 sym(pi)]
% q=findJointAngles(endeff)
```

1.10 Task 5.10: Mapping DH angles to servo angles

```
function [servoJointAngles, errorCode] = dh2servo(jointAngles)
  theta1 = jointAngles(1);
  theta2 = jointAngles(2);
  theta3 = jointAngles(3);
  theta4 = jointAngles(4);

  errorCode = 1;
  servoJointAngles = [];

  % Check joint limits (modify limits as needed)
  if -150 < theta1 && theta1 < 150 && ...
      -150 < (theta2 - 90) && (theta2 - 90) < 150 && ...
      -150 < theta3 && theta3 < 150 && -150 < theta4 && theta4 < 150
    servoJointAngles = [theta1, theta2 - 90, theta3, theta4];
    errorCode = 0;
  end
end
```