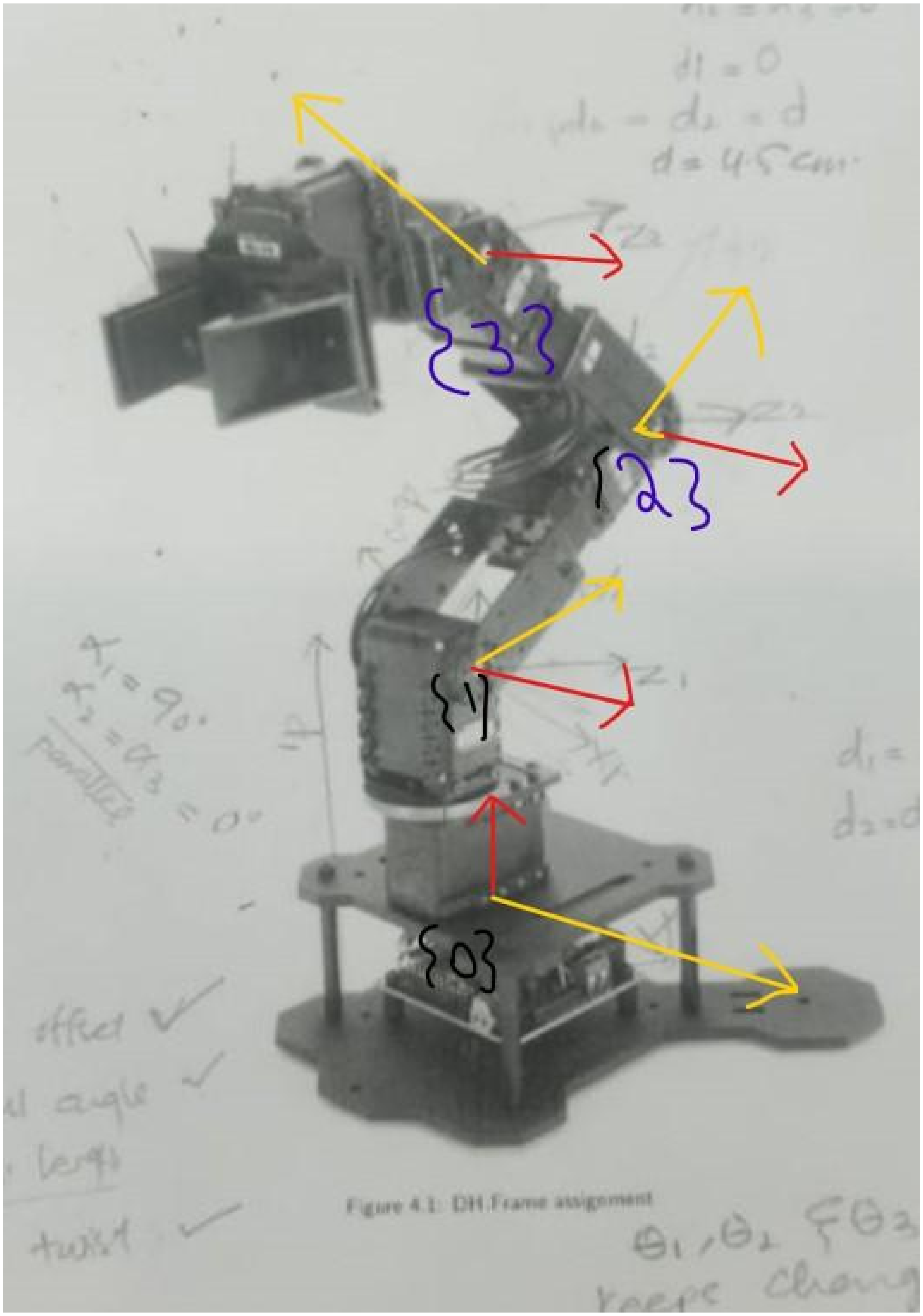
Lab 4

Members: Hazika Farooq & Syed Mustafa

Task 4.1



Yellow = x axis

Red = Z axis

Task 4.2

Link	a_i	α_i	d_i	θ_i
1	0	90	14 cm	θ_1
2	10.5 cm	0	0	θ_2
3	10.5cm	0	0	θ_3
4	7.5 cm	0	0	θ_4

Table 1: Joint DH Parameter Table

Task 4.3

```
syms('theta_1')
syms('theta_2')
syms('theta_3')
syms('theta_4')


alpha_1 =90;
d_1=14;
a_1=0;

alpha_2 =0;
d_2=0;
a_2 = 10.5;

alpha_3 =0;
d_3=0;
a_3=10.5;

alpha_4 =0;
d_4=0;
a_4 =7.5;

T01 = [cosd(theta_1),  -sind(theta_1)*cosd(alpha_1), sind(theta_1)*sind(alpha_1) a_1*cosd(theta
    sind(theta_1), cosd(theta_1)*cosd(alpha_1), -cosd(theta_1)*sind(alpha_1), a_1*sind(theta_1)
    0, sind(alpha_1), cosd(alpha_1), d_1;
    0, 0 , 0, 1];

T02 = [cosd(theta_2),  -sind(theta_2)*cosd(alpha_2), sind(theta_2)*sind(alpha_2) a_2*cosd(theta
    sind(theta_2), cosd(theta_2)*cosd(alpha_2), -cosd(theta_2)*sind(alpha_2), a_2*sind(theta_2)
    0, sind(alpha_2), cosd(alpha_2), d_2;
    0, 0 , 0, 1];

T03 = [cosd(theta_3),  -sind(theta_3)*cosd(alpha_3), sind(theta_3)*sind(alpha_3) a_3*cosd(theta
    sind(theta_3), cosd(theta_3)*cosd(alpha_3), -cosd(theta_3)*sind(alpha_3), a_3*sind(theta_3)
    0, sind(alpha_3), cosd(alpha_3), d_3;
```

```
    0, 0 , 0, 1];

T04= [cosd(theta_4),  -sind(theta_4)*cosd(alpha_4), sind(theta_4)*sind(alpha_4) a_4*cosd(theta_
    sind(theta_4), cosd(theta_4)*cosd(alpha_4), -cosd(theta_4)*sind(alpha_4), a_4*sind(theta_4)
    0, sind(alpha_4), cosd(alpha_4), d_4;
    0, 0 , 0, 1];


T=simplify(T01*T02*T03*T04 )
```

T =

$$\begin{pmatrix} \sigma_3\,\sigma_5 & -\sigma_3\,\sigma_1 & \sigma_2 & \frac{3\,\sigma_3\,\sigma_4}{2} \\ \sigma_2\,\sigma_5 & -\sigma_2\,\sigma_1 & -\sigma_3 & \frac{3\,\sigma_2\,\sigma_4}{2} \\ \sigma_1 & \sigma_5 & 0 & \frac{21\sin\left(\frac{\pi\,(\theta_2+\theta_3)}{180}\right)}{2} + \frac{21\sin\left(\frac{\pi\,\theta_2}{180}\right)}{2} + \frac{15\,\sigma_1}{2} + 14 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = \sin\left(\frac{\pi\,(\theta_2+\theta_3+\theta_4)}{180}\right)$$

$$\sigma_2 = \sin\left(\frac{\pi\,\theta_1}{180}\right)$$

$$\sigma_3 = \cos\left(\frac{\pi\,\theta_1}{180}\right)$$

$$\sigma_4 = 7\cos\left(\frac{\pi\,(\theta_2+\theta_3)}{180}\right) + 7\cos\left(\frac{\pi\,\theta_2}{180}\right) + 5\,\sigma_5$$

$$\sigma_5 = \cos\left(\frac{\pi\,(\theta_2+\theta_3+\theta_4)}{180}\right)$$

Task 4.4

```
function [x,y,z,R] = findPincher(theta_1,theta_2,theta_3,theta_4)

    alpha_1 =90;
    d_1=14;
    a_1=0;

    alpha_2 =0;
    d_2=0;
```

```
    a_2 = 10.5;

    alpha_3 =0;
    d_3=0;
    a_3=10.5;

    alpha_4 =0;
    d_4=0;
    a_4 =7.5;

    T01 = [cosd(theta_1),  -sind(theta_1)*cosd(alpha_1), sind(theta_1)*sind(alpha_1) a_1*cosd(theta_1);
        sind(theta_1), cosd(theta_1)*cosd(alpha_1), -cosd(theta_1)*sind(alpha_1), a_1*sind(theta_1);
        0, sind(alpha_1), cosd(alpha_1), d_1;
        0, 0 , 0, 1];

    T02 = [cosd(theta_2),  -sind(theta_2)*cosd(alpha_2), sind(theta_2)*sind(alpha_2) a_2*cosd(theta_2);
        sind(theta_2), cosd(theta_2)*cosd(alpha_2), -cosd(theta_2)*sind(alpha_2), a_2*sind(theta_2);
        0, sind(alpha_2), cosd(alpha_2), d_2;
        0, 0 , 0, 1];

    T03 = [cosd(theta_3),  -sind(theta_3)*cosd(alpha_3), sind(theta_3)*sind(alpha_3) a_3*cosd(theta_3);
        sind(theta_3), cosd(theta_3)*cosd(alpha_3), -cosd(theta_3)*sind(alpha_3), a_3*sind(theta_3);
        0, sind(alpha_3), cosd(alpha_3), d_3;
        0, 0 , 0, 1];

    T04= [cosd(theta_4),  -sind(theta_4)*cosd(alpha_4), sind(theta_4)*sind(alpha_4) a_4*cosd(theta_4);
        sind(theta_4), cosd(theta_4)*cosd(alpha_4), -cosd(theta_4)*sind(alpha_4), a_4*sind(theta_4);
        0, sind(alpha_4), cosd(alpha_4), d_4;
        0, 0 , 0, 1];

    T=simplify(T01*T02*T03*T04);
    R= 0;
    q = T(1:3,4);
    x= q(1); y = q(2); z = q(3);

end
```

Note: The angle in this function is taken strictly in degrees

Task 4.5

The given code was made into a function to allow for reussability

```
function pincherModel(joint_angles)
% This is a script to build the Phantom X Pincher in MATLAB based on its DH
    % parameters. It is based on MATLAB example available at
    % https://www.mathworks.com/help/robotics/ug/build-manipulator-robot-using-kinematic-dh-parameters.html
    % MATLAB creates a rigid body tree


    % Provide the DH parameters for the robot. The parameters are arranged in
    % the order [a, alpha, d, theta], and going from link 1 to link n. The
    % entry in the matrix corresponding to the joint variable is ignored.
    dhparams = [0       pi/2    14       0;
                10.5        0       0       0;
                10.5        0        0        0;
                7.5        0        0        0];
```

```
numJoints = size(dhparams,1);

% Create a rigid body tree object.
robot = rigidBodyTree;

% Create a model of the robot using DH parameters.
% Create a cell array for the rigid body object, and another for the joint
% objects. Iterate through the DH parameters performing this process:
% 1. Create a rigidBody object with a unique name.
% 2. Create and name a revolute rigidBodyJoint object.
% 3. Use setFixedTransform to specify the body-to-body transformation of the
%    joint using DH parameters.
% 4. Use addBody to attach the body to the rigid body tree.
bodies = cell(numJoints,1);
joints = cell(numJoints,1);
for i = 1:numJoints
    bodies{i} = rigidBody(['body' num2str(i)]);
    joints{i} = rigidBodyJoint(['jnt' num2str(i)],"revolute");
    setFixedTransform(joints{i},dhparams(i,:),"dh");
    bodies{i}.Joint = joints{i};
    if i == 1 % Add first body to base
        addBody(robot,bodies{i},"base")
    else % Add current body to previous body by name
        addBody(robot,bodies{i},bodies{i-1}.Name)
    end
end

% Verify that your robot has been built properly by using the showdetails or
% show function. The showdetails function lists all the bodies of the robot
% in the MATLAB® command window. The show function displays the robot with
% a specified configuration (home by default).
showdetails(robot)
figure(Name="Phantom X Pincher")
show(robot);

% Forward Kinematics for different configurations
% Enter joint angles in the matrix below in radians
configNow = joint_angles;%[pi/2, 0, 0, 0 ];

% Display robot in provided configuration
config = homeConfiguration(robot);
for i = 1:numJoints
    config(i).JointPosition = configNow(i);
end
show(robot,config);

% Determine the pose of end-effector in provided configuration
poseNow = getTransform(robot,config,"body4");

% Display position and orientation of end-effector
clc;
disp('The position of end-effector is:');
disp('');
disp(['X: ', num2str(poseNow(1,4))]);
disp('');
disp(['Y: ', num2str(poseNow(2,4))]);
```

```
        disp('');
        disp(['Z: ', num2str(poseNow(3,4))]);
        disp(' ');
        disp(['R: ']);
        poseNow(1:3,1:3)
        disp(' ');
        disp('The orientation angle is given with respect to the x-axis of joint 2:');
        disp('');
        poseNow01 = getTransform(robot,config,"body1");
        R14 = poseNow01(1:3,1:3)'*poseNow(1:3,1:3);
        angle = rad2deg(atan2(R14(2,1),R14(1,1)));
        disp(['Angle: ',num2str(angle), ' degrees.']);

    end
```

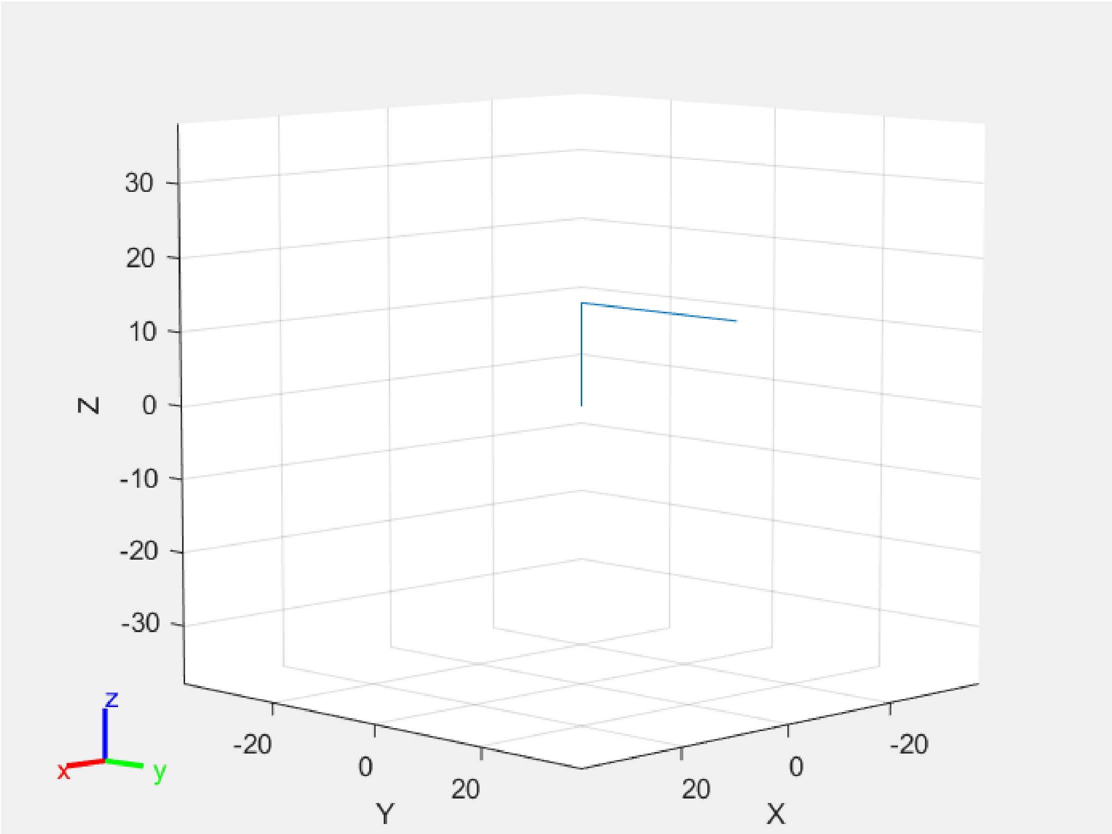
```
joint_angles = [pi/2, 0, 0, 0]
```

```
joint_angles = 1×4
    1.5708         0         0         0
```

```
pincherModel(joint_angles)
```

```
--------------------
Robot: (4 bodies)

 Idx     Body Name    Joint Name    Joint Type     Parent Name(Idx)    Children Name(s)
 ---     ---------    ----------    ----------     ----------------    ----------------
   1         body1          jnt1      revolute              base(0)    body2(2)
   2         body2          jnt2      revolute             body1(1)    body3(3)
   3         body3          jnt3      revolute             body2(2)    body4(4)
   4         body4          jnt4      revolute             body3(3)
--------------------
```




```
The position of end-effector is:
X: 1.7451e-15
Y: 28.5
Z: 14

R:
ans = 3×3
    0.0000   -0.0000    1.0000
    1.0000    0.0000   -0.0000
         0    1.0000    0.0000

The orientation angle is given with respect to the x-axis of joint 2:
Angle: 0 degrees.
```

```
joint_angles = joint_angles * 180/pi
```

```
joint_angles = 1×4
    90     0     0     0
```

```
findPincher(joint_angles(1),joint_angles(2),joint_angles(3),joint_angles(4))
```

```
ans = 0
```

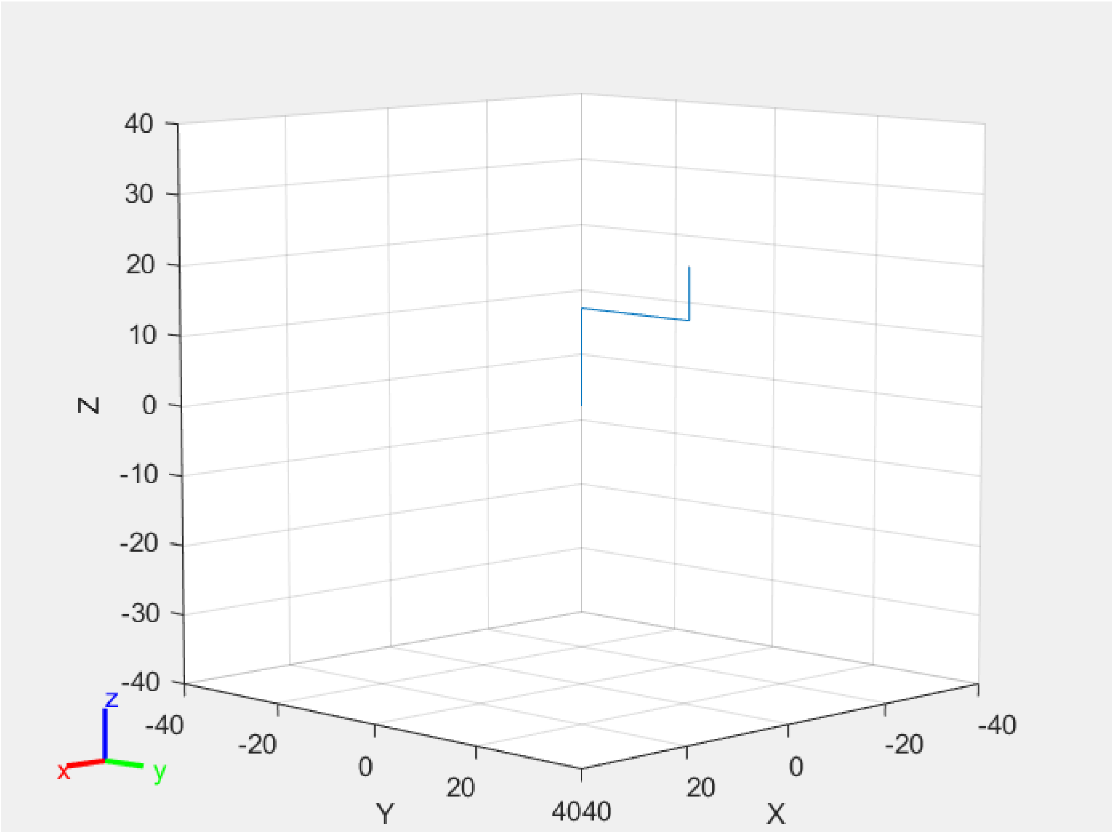
```
joint_angles = [pi/2, 0, 0, pi/2]
```

```
joint_angles = 1×4
    1.5708         0         0    1.5708
```

```
pincherModel(joint_angles)
```

```
--------------------
Robot: (4 bodies)
```

```
 Idx     Body Name    Joint Name   Joint Type     Parent Name(Idx)   Children Name(s)
 ---     ---------    ----------   ----------     ----------------   ----------------
   1         body1          jnt1     revolute              base(0)   body2(2)
   2         body2          jnt2     revolute             body1(1)   body3(3)
   3         body3          jnt3     revolute             body2(2)   body4(4)
   4         body4          jnt4     revolute             body3(3)
--------------------
```




```
The position of end-effector is:
X: 8.2664e-16
Y: 21
Z: 21.5

R:
ans = 3×3
   -0.0000   -0.0000    1.0000
    0.0000   -1.0000   -0.0000
    1.0000    0.0000    0.0000

The orientation angle is given with respect to the x-axis of joint 2:
Angle: 90 degrees.
```

```
joint_angles = joint_angles * 180/pi
```

```
joint_angles = 1×4
    90     0     0    90
```

```
findPincher(joint_angles(1),joint_angles(2),joint_angles(3),joint_angles(4))
```

```
ans = 0
```

```
joint_angles = [pi/2, 0, pi/4, pi/2]
```

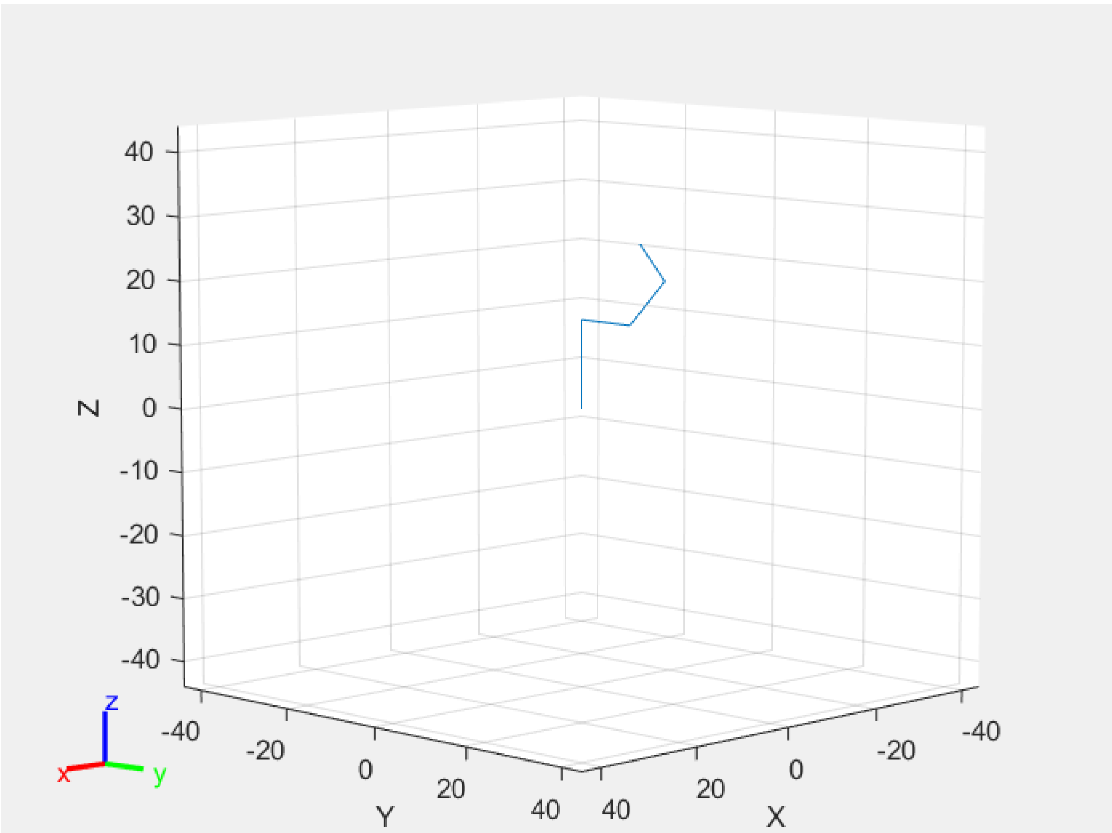
```
joint_angles = 1×4
```

```
    1.5708         0    0.7854    1.5708
```

```
pincherModel(joint_angles)
```

```
--------------------
Robot: (4 bodies)

 Idx     Body Name   Joint Name   Joint Type    Parent Name(Idx)   Children Name(s)
 ---     ---------   ----------   ----------    ----------------   ----------------
   1         body1         jnt1     revolute             base(0)   body2(2)
   2         body2         jnt2     revolute            body1(1)   body3(3)
   3         body3         jnt3     revolute            body2(2)   body4(4)
   4         body4         jnt4     revolute            body3(3)
--------------------
```




```
The position of end-effector is:
X: -6.5275e-18
Y: 12.6213
Z: 26.7279

R:
ans = 3×3
   -0.0000   -0.0000    1.0000
   -0.7071   -0.7071   -0.0000
    0.7071   -0.7071    0.0000

The orientation angle is given with respect to the x-axis of joint 2:
Angle: 135 degrees.
```

```
joint_angles = joint_angles * 180/pi
```

```
joint_angles = 1×4
    90     0    45    90
```

```
findPincher(joint_angles(1),joint_angles(2),joint_angles(3),joint_angles(4))
```

```
ans = 0
```

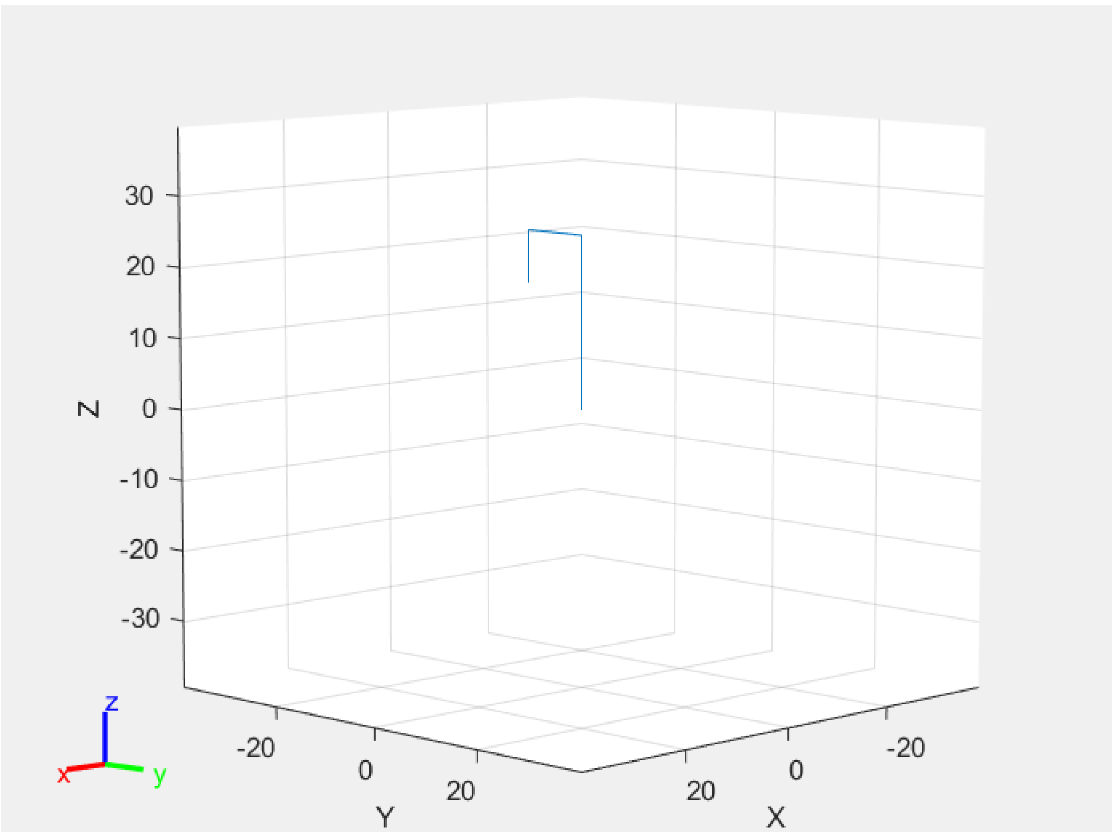
```
joint_angles = [pi/2, pi/2, pi/2, pi/2]
```

```
joint_angles = 1×4
    1.5708    1.5708    1.5708    1.5708
```

```
pincherModel(joint_angles)
```

```
--------------------
Robot: (4 bodies)

 Idx     Body Name   Joint Name   Joint Type    Parent Name(Idx)   Children Name(s)
 ---     ---------   ----------   ----------    ----------------   ----------------
   1         body1         jnt1     revolute             base(0)   body2(2)
   2         body2         jnt2     revolute            body1(1)   body3(3)
   3         body3         jnt3     revolute            body2(2)   body4(4)
   4         body4         jnt4     revolute            body3(3)
--------------------
```




```
The position of end-effector is:
X: -8.2664e-16
Y: -10.5
Z: 17

R:
ans = 3×3
    0.0000    0.0000    1.0000
   -0.0000    1.0000   -0.0000
   -1.0000   -0.0000    0.0000
```

```
The orientation angle is given with respect to the x-axis of joint 2:
Angle: -90 degrees.
```

```
joint_angles = joint_angles * 180/pi
```

```
joint_angles = 1×4
    90    90    90    90
```

```
findPincher(joint_angles(1),joint_angles(2),joint_angles(3),joint_angles(4))
```

```
ans = 0
```

Task 4.6

Joint ID	DH Joint Angles	Servo Angles	Aligned in direction
1	0	90	Yes
2	0	-90	Yes
3	0	0	Yes
4	0	0	Yes

Table 2: Linear Mapping between servo angles and DH angles

Joint ID	Minimum Joint Angle		Maximum Joint Angles	
	Servo Angle	DH Joint Angles	Servo Angle	DH Joint Angles
1	-150	-240	150	60
2	-150	-60	150	240
3	-150	-150	150	150
4	-150	-150	150	150

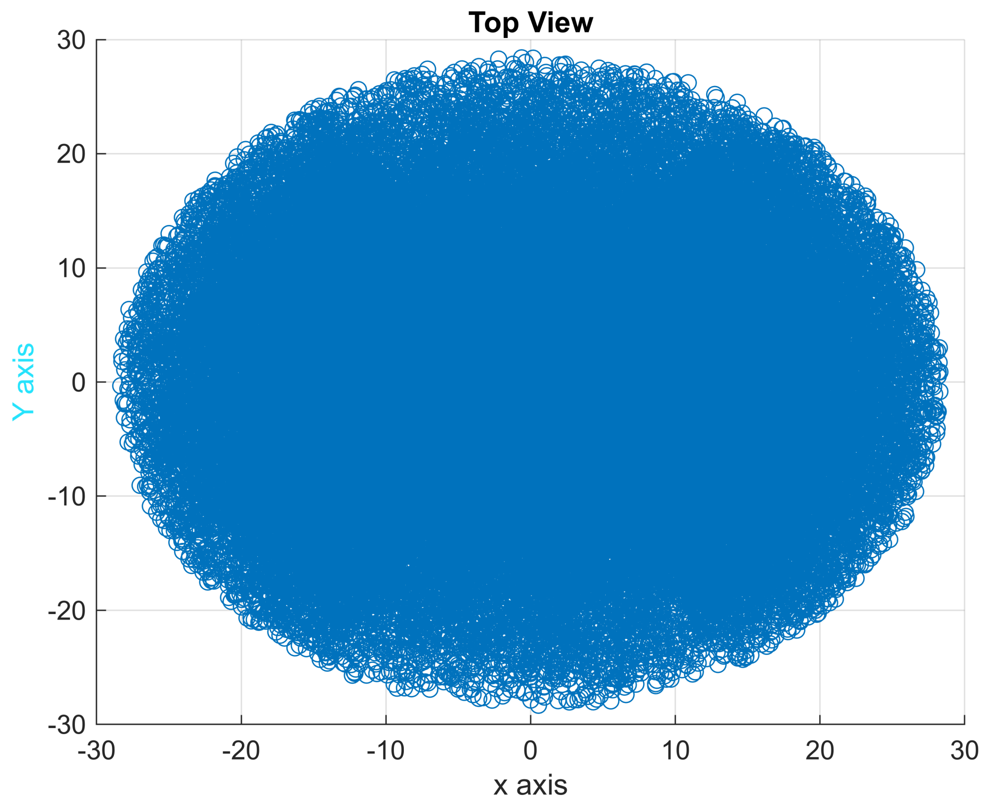
Table 3: Joint Limits

Task 4.7

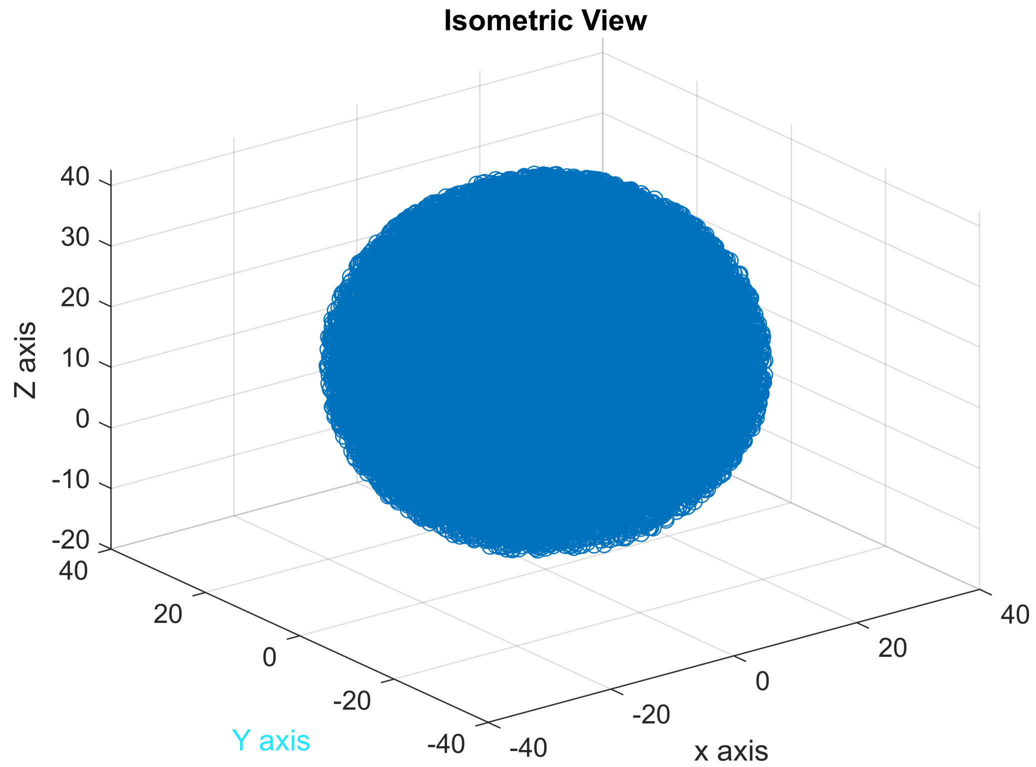
```
%% Task 4.7
N = 100000;
theta = zeros(4,N);
Angles = [-240 60; -60 240; -150 150; -150 150];
theta = Angles(:,1) + (Angles(:,2)-Angles(:,1)).* rand(4, N);pos = zeros(3, N);
index = 1;
for thet = theta
  [x,y,z,R] = findPincher(thet(1),thet(2),thet(3),thet(4));
  pos(:,index) = [x,y,z];
```

```
  index = index + 1;

end
figure;
scatter3(pos(1,:),pos(2,:),pos(3,:))
xlabel('x axis'); ylabel('Y axis', 'Color',[0.15 0.89 0.99]);zlabel('Z axis');
view(0,90)
title('Top View')
```




```
figure;
scatter3(pos(1,:),pos(2,:),pos(3,:))
title('Isometric View')
xlabel('x axis'); ylabel('Y axis', 'Color',[0.15 0.89 0.99]);zlabel('Z axis');
```



Task 4.8

```
function errorCode = setPosition(jointAngles)
    jointAngles = jointAngles * pi/180;
    disp('The joint angles as expressed in radians are')
    jointAngles
    new_theta = zeros(1,4); % Array for JointAngles Mapped to [-pi, pi]
    errorCode = 0; % 0 -> No error yet
    for i=1:4
        % Mapping each jointangle to [-pi,pi]
        new_theta(i) = mod(jointAngles(i)+pi, 2*pi) - pi;
        % Condition for invalid input angle
        if (new_theta(i) <= -pi*15/18 && new_theta(i) >= +pi*15/18)
            disp(strcat('Angle ', num2str(i), ' Out of range = ', num2str(new_theta(i))));
            % Error code = i -> ith Joint is out of range
            errorCode = i;

        end
    end
    disp('the modded joint angles are')
    new_theta
    % Passing the Joint Angles to Robot is no Error Occured
    if errorCode == 0
        % Mapping DH angles to Servo Angles
        map_theta = zeros(1,5);
        map_theta(1) = -(jointAngles(1) - pi/2);
        map_theta(2) = -(jointAngles(2) + pi/2);
        map_theta(3) = jointAngles(3);
        map_theta(4) = jointAngles(4);
```

```
            % Connecting to Robot and passing the theta information
            % to Robot for execution with a certain speed (64 for every joint
            % in this case
            arb = Arbotix('port','COM10', 'nservos',5);
            arb.setpos(map_theta, [64, 64, 64, 64, 64]);
        end
    end
```

```
setPosition([0,90,20,-10])
```

Task 4.9

```
thetas = zeros(5,4);
thetas(1,:) = [pi/2, 0, 0, 0] ;
thetas(2,:) = [pi/2, pi, 0, 0] ;
thetas(3,:) = [-pi/2, -pi/2, -3*pi/2, 0] ;
thetas(4,:) = [pi/2, 0, 0, +pi/2] ;
thetas(5,:) = [pi/2, 0, 0, -pi/2] ;
calculated = zeros(5,3)
```

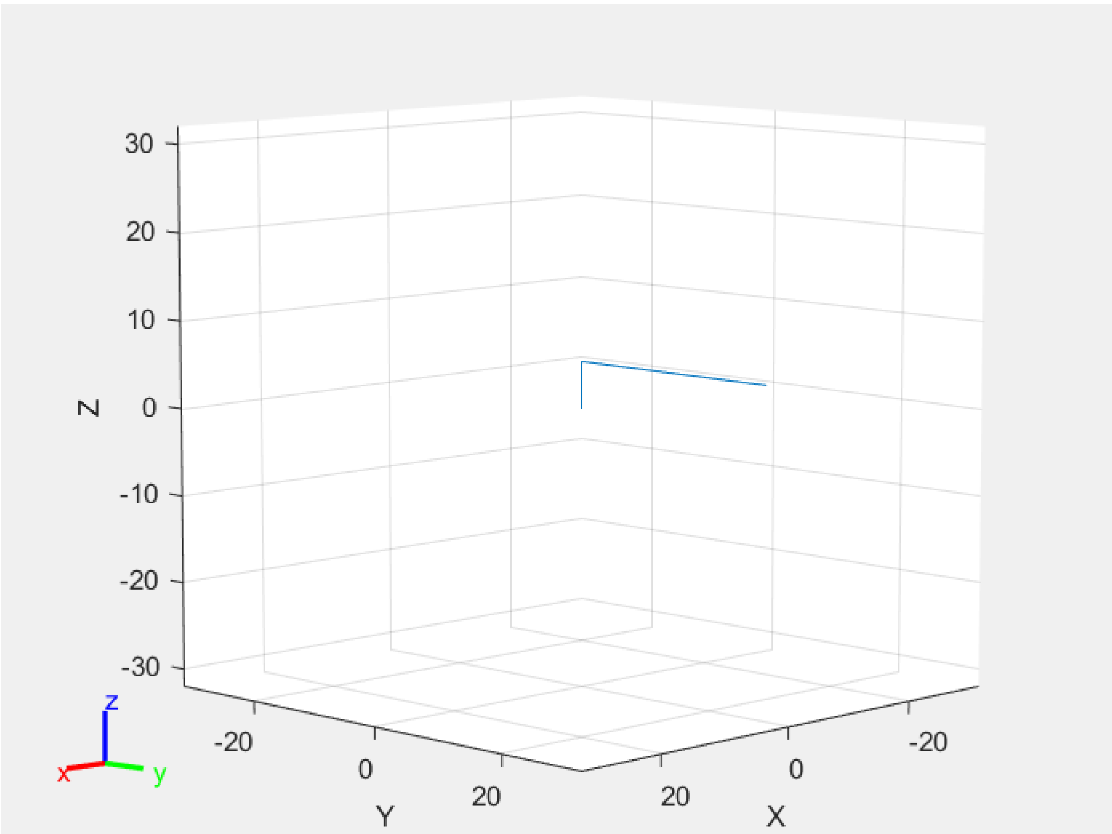
```
calculated = 5×3
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
```

```
for i=1:5
    [x, y, z, R] = findPincher(thetas(i,1), ...
    thetas(i,2), ...
    thetas(i,3), ...
    thetas(i,4))
    caluclated(i,:) = [x y z]
    pincherModel([thetas(i,1), ...
    thetas(i,2), ...
    thetas(i,3), ...
    thetas(i,4)])
    disp(num2str(thetas(i,:)))
    disp('run')
end
```

```
x = 28.4893
y = 0.7812
z = 14
R = 3×3
    0.9996         0    0.0274
    0.0274         0   -0.9996
         0    1.0000         0
caluclated = 1×3
   28.4893    0.7812   14.0000
```

```
--------------------
Robot: (4 bodies)

 Idx     Body Name   Joint Name   Joint Type    Parent Name(Idx)   Children Name(s)
 ---     ---------   ----------   ----------    ----------------   ----------------
   1         body1         jnt1     revolute             base(0)   body2(2)
   2         body2         jnt2     revolute            body1(1)   body3(3)
   3         body3         jnt3     revolute            body2(2)   body4(4)
   4         body4         jnt4     revolute            body3(3)
--------------------
```



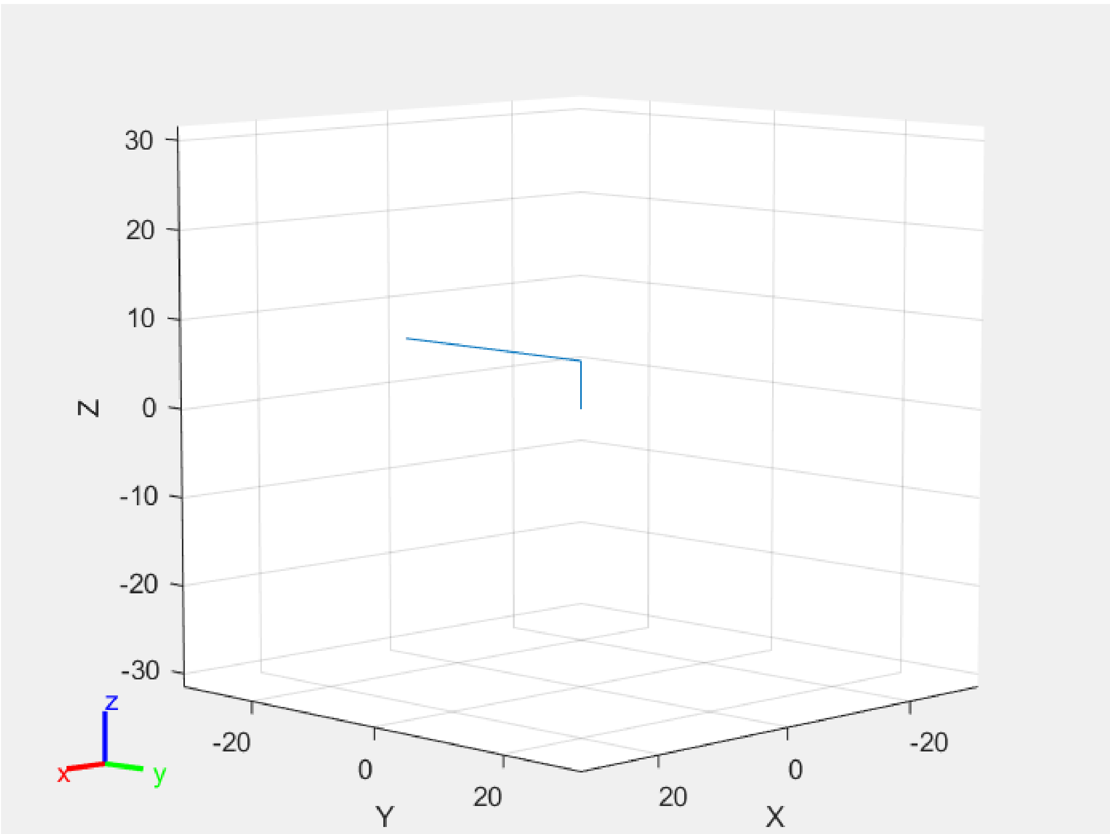

```
The position of end-effector is:
X: 1.7451e-15
Y: 28.5
Z: 5.4

R:
ans = 3×3
    0.0000   -0.0000    1.0000
    1.0000    0.0000   -0.0000
         0    1.0000    0.0000

The orientation angle is given with respect to the x-axis of joint 2:
Angle: 0 degrees.
1.5708          0           0           0
run
x = 28.4465
y = 0.7801
z = 15.5619
R = 3×3
    0.9981   -0.0548    0.0274
    0.0274   -0.0015   -0.9996
    0.0548    0.9985         0
caluclated = 2×3
   28.4893    0.7812   14.0000
```

```
   28.4465    0.7801   15.5619
--------------------
Robot: (4 bodies)

 Idx     Body Name   Joint Name   Joint Type    Parent Name(Idx)   Children Name(s)
 ---     ---------   ----------   ----------    ----------------   ----------------
   1         body1         jnt1     revolute             base(0)   body2(2)
   2         body2         jnt2     revolute            body1(1)   body3(3)
   3         body3         jnt3     revolute            body2(2)   body4(4)
   4         body4         jnt4     revolute            body3(3)
--------------------
```



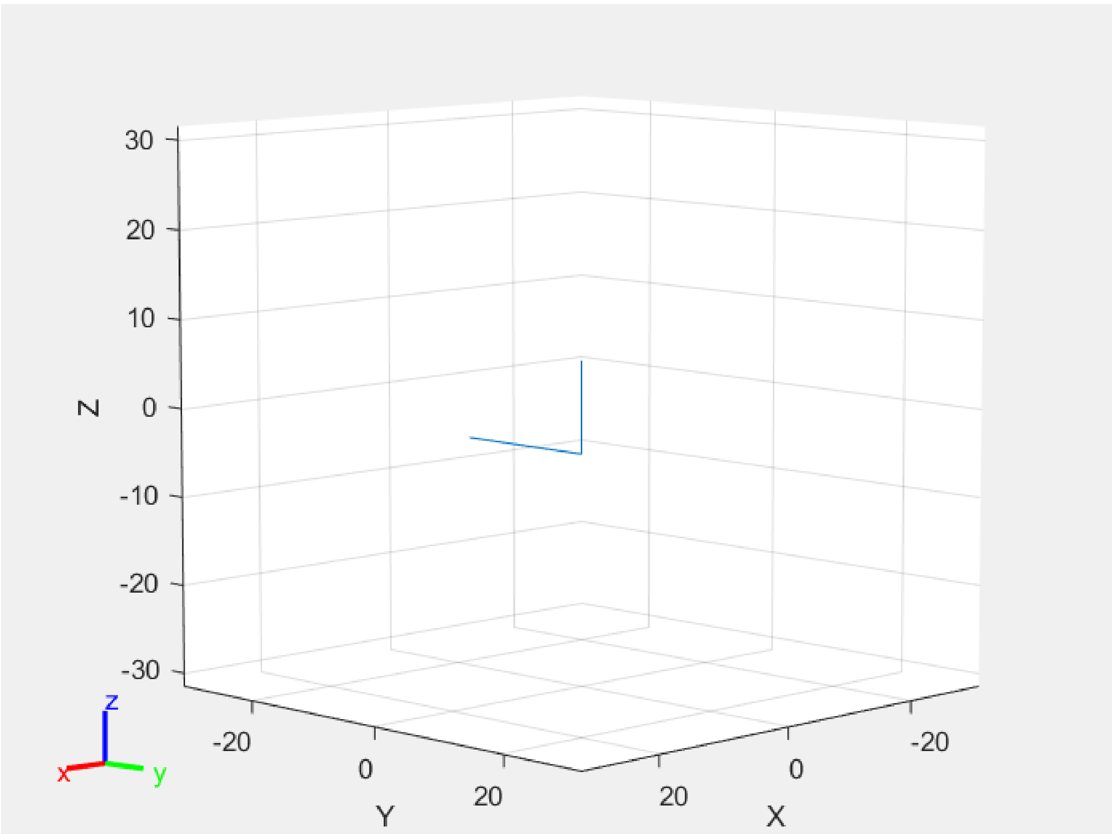

```
The position of end-effector is:
X: -1.7451e-15
Y: -28.5
Z: 5.4

R:
ans = 3×3
   -0.0000    0.0000    1.0000
   -1.0000   -0.0000   -0.0000
    0.0000   -1.0000    0.0000

The orientation angle is given with respect to the x-axis of joint 2:
Angle: 180 degrees.
1.5708      3.1416           0           0
run
x = 28.3773
y = -0.7782
z = 11.7422
R = 3×3
    0.9936    0.1094   -0.0274
   -0.0272   -0.0030   -0.9996
   -0.1094    0.9940         0
caluclated = 3×3
```

```
   28.4893    0.7812   14.0000
   28.4465    0.7801   15.5619
   28.3773   -0.7782   11.7422
--------------------
Robot: (4 bodies)

 Idx    Body Name   Joint Name   Joint Type    Parent Name(Idx)   Children Name(s)
 ---    ---------   ----------   ----------    ----------------   ----------------
   1        body1         jnt1     revolute             base(0)   body2(2)
   2        body2         jnt2     revolute            body1(1)   body3(3)
   3        body3         jnt3     revolute            body2(2)   body4(4)
   4        body4         jnt4     revolute            body3(3)
--------------------
```



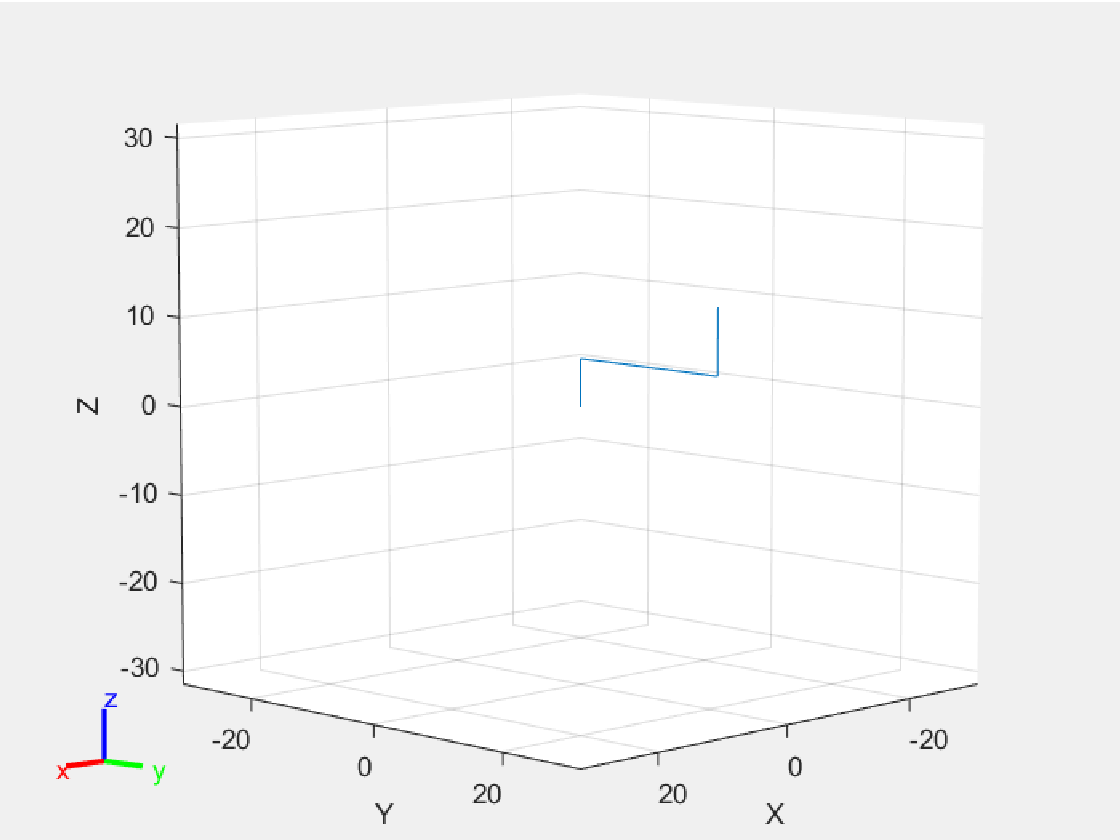

```
The position of end-effector is:
X: 4.5924e-16
Y: -18
Z: -5.1

R:
ans = 3×3
    0.0000    0.0000   -1.0000
   -1.0000    0.0000   -0.0000
    0.0000    1.0000    0.0000

The orientation angle is given with respect to the x-axis of joint 2:
Angle: 1.4033e-14 degrees.
-1.5708     -1.5708     -4.7124           0
run
x = 28.4865
y = 0.7812
z = 14.2056
R = 3×3
    0.9992   -0.0274    0.0274
    0.0274   -0.0008   -0.9996
```

```
    0.0274    0.9996         0
caluclated = 4×3
   28.4893    0.7812   14.0000
   28.4465    0.7801   15.5619
   28.3773   -0.7782   11.7422
   28.4865    0.7812   14.2056
--------------------
Robot: (4 bodies)

 Idx    Body Name   Joint Name   Joint Type    Parent Name(Idx)   Children Name(s)
 ---    ---------   ----------   ----------    ----------------   ----------------
   1        body1         jnt1     revolute             base(0)   body2(2)
   2        body2         jnt2     revolute            body1(1)   body3(3)
   3        body3         jnt3     revolute            body2(2)   body4(4)
   4        body4         jnt4     revolute            body3(3)
--------------------
```



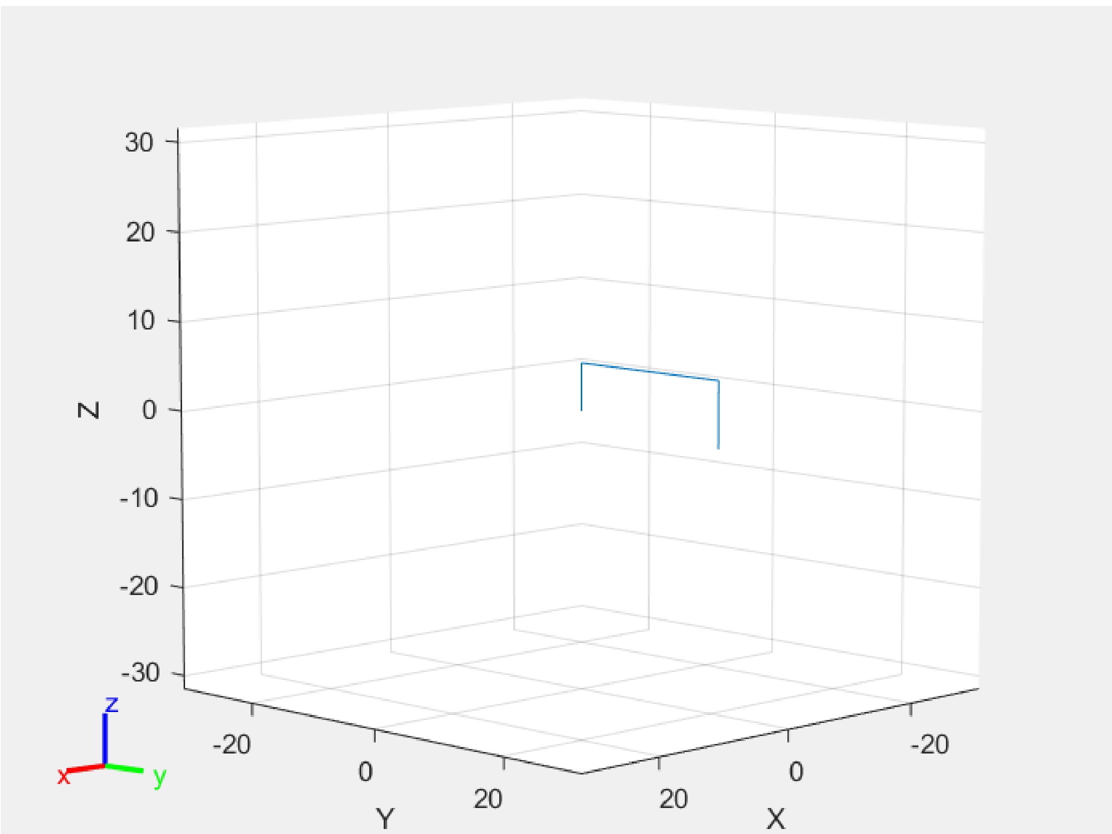

```
The position of end-effector is:
X: 8.2664e-16
Y: 21
Z: 12.9

R:
ans = 3×3
   -0.0000   -0.0000    1.0000
    0.0000   -1.0000   -0.0000
    1.0000    0.0000    0.0000

The orientation angle is given with respect to the x-axis of joint 2:
Angle: 90 degrees.
1.5708           0           0      1.5708
run
x = 28.4865
y = 0.7812
z = 13.7944
```

```
R = 3×3
    0.9992    0.0274    0.0274
    0.0274    0.0008   -0.9996
   -0.0274    0.9996         0
caluclated = 5×3
   28.4893    0.7812   14.0000
   28.4465    0.7801   15.5619
   28.3773   -0.7782   11.7422
   28.4865    0.7812   14.2056
   28.4865    0.7812   13.7944
--------------------
Robot: (4 bodies)

 Idx     Body Name   Joint Name   Joint Type     Parent Name(Idx)   Children Name(s)
 ---     ---------   ----------   ----------     ----------------   ----------------
   1         body1         jnt1     revolute              base(0)   body2(2)
   2         body2         jnt2     revolute             body1(1)   body3(3)
   3         body3         jnt3     revolute             body2(2)   body4(4)
   4         body4         jnt4     revolute             body3(3)
--------------------
```




```
The position of end-effector is:
X: 1.7451e-15
Y: 21
Z: -2.1

R:
ans = 3×3
    0.0000    0.0000    1.0000
    0.0000    1.0000   -0.0000
   -1.0000    0.0000    0.0000

The orientation angle is given with respect to the x-axis of joint 2:
Angle: -90 degrees.
1.5708          0           0     -1.5708
```

run

```
i = 3;
% pincherModel([thetas(i,1), ...
%     thetas(i,2), ...
%     thetas(i,3), ...
%     thetas(i,4)])
% thetas = thetas * 180/pi
% setPosition([thetas(i,1), ...
%     thetas(i,2), ...
%     thetas(i,3), ...
%     thetas(i,4)] ...
%     )
```

```
caluclated
```

```
caluclated = 5×3
   28.4893    0.7812   14.0000
   28.4465    0.7801   15.5619
   28.3773   -0.7782   11.7422
   28.4865    0.7812   14.2056
   28.4865    0.7812   13.7944
```

```
Measured
```

```
Measured = 5×3
   29.1000    1.5000   14.4000
   29.0000    0.8000   16.3000
   29.1000   -0.2000   12.7000
   28.7000    0.8000   15.0000
   29.0000    1.7000   13.9000
```

```
Error = abs(caluclated - Measured)
```

```
Error = 5×3
    0.6107    0.7188    0.4000
    0.5535    0.0199    0.7381
    0.7227    0.5782    0.9578
    0.2135    0.0188    0.7944
    0.5135    0.9188    0.1056
```

```
euclidean_error = sqrt(Error(:,1).^2+Error(:,2).^2+Error(:,3).^2)
```

```
euclidean_error = 5×1
    1.0245
    0.9228
    1.3319
    0.8228
    1.0579
```