TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Lorenzo Etchenique 282974
Jan Dierkes 282910
Christen Blomdahl 282803

Assignment 1
ASE-9407 2018-01

Robot Manipulators:

Modeling, Control and Programming

Table of Contents

LIST OF FIGURES

List of Tables

LIST OF SYMBOLS AND ABBREVIATIONS

Abbreviation	**Meaning**
DH	Denavit-Hartenberg
F	Frame
h	height
HT	Homogenous Transformation
P	Prismatic (joint)
R	Revolute (joint)
T	Tool
α	Alpha
θ	Theta

1. MODIFIED DH PARAMETERS

1.1. Explain clearly any choices you made when placing the link frames. Do not describe here the DH convention, but rather your decisions in some frames where you had freedom to choose their location.

Explanations are done with respect to the frame assignment shown in Figure X.X. We chose to place the frames in order to generate the simplest Denavit-Hartenberg (DH) table we can have and an easy visualization. First off, we decided to design a classic RRPR SCARA robot even though we could also have done a RRRP structure. We put all the frames at the same height except for the tool frame. In this respect we started with Frame Zero (F_0) with an offset of the basis (F_0 = v(0.1,0.1, h). Here h can be arbitrary, but we chose it to be h = 0.1 *[height of basis]* + 0.025 *[height of red piece]* + 0.135 *[tool offset]* = 0.26 so that in the initial position (when d_3=0), the gripper is just at the height of the red piece. Here we could have placed the zero frame F_0 into the basis or everywhere else, but that would have made our transformation much more complicated as we also need to assign one extra frame to transform to the first joint. After that we assigned the frames for the joints with the Z-axis pointing in direction of the joint axis according to DH convention. For F_1 we need x_1 to be in the direction of the common perpendicular of z_1 and z_2 as they are parallel. This means x_1 is pointing in the direction of z_2. But as z_1 and z_2 are parallels we are free to place F_1 anywhere along z_1. We decided to place F_1 in F_0 to keep the transformation simple. In the same way we have the freedom to place F_2 anywhere along z_2. Axis x_3 could point in any direction, it just needs to be perpendicular to Z_3 as Z_3 and Z_4 are coincident. To keep it simple x_3 is pointing in the same direction as x_2. We placed F_4, which is referring to the last revolute joint, in the origin of F_3 as z_4 and z_T are parallels and we can but the origin of frame 4 anywhere along z_4. The axis x_4 then needs to point to F_T. We have the freedom to put the tool frame wherever we want, but the tool frame F_T should be in the Tool Center Point (TCP) and not elsewhere in order to keep it simpler. This point is a translation from the center of F_4 along the prismatic joint considering the length of the gripper and the offset of the gripper fingers. This information can be drawn from Figure 3 in the Assignment 1 Description. The tool frame has the same orientation as F_4 as it is directly affected by the rotation of the revolute joint in F_4.

1.2. Draw the robot in two configurations (zero and with offsets). In each show clearly, the frame assignment.

The frame axes in the pictures are coded as it follows:
Green: X axis
Blue: Y axis
Red: Z axis

Note: The frames assigned to the pieces are just extra information to the pictures, do not to take them in account as in MatLab their pose might differ.

1.2.1 Zero position

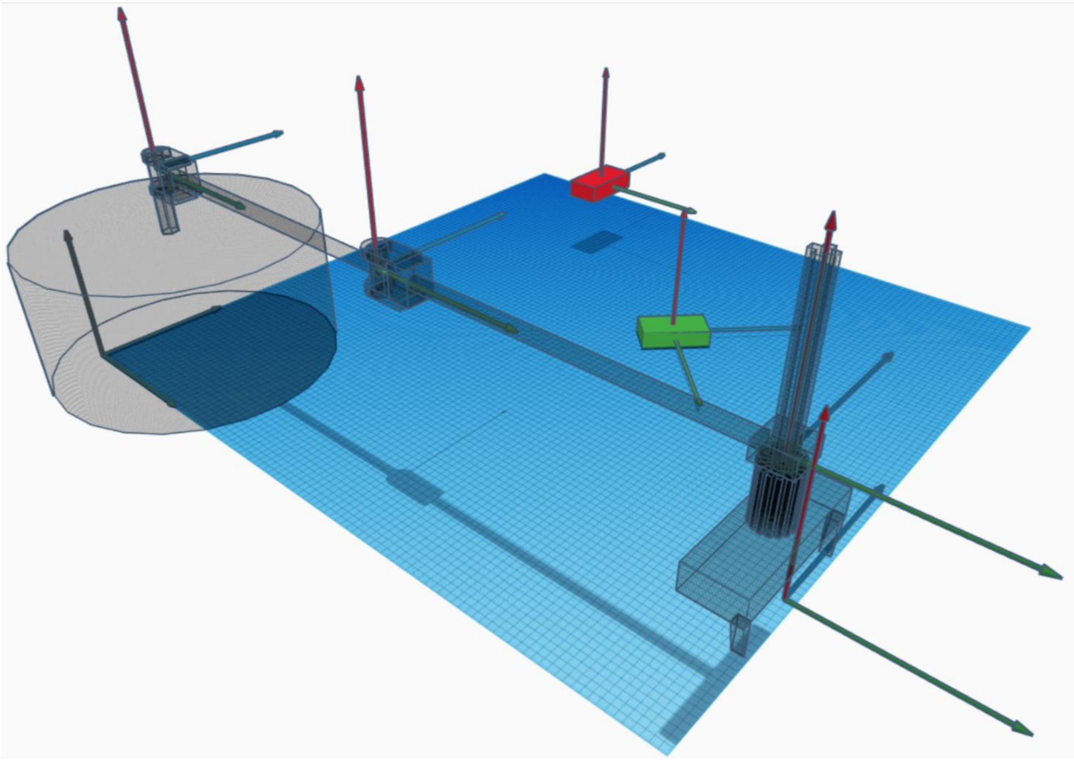

Figure 1: Mechanism with zero value joint parameters. Notice that there are two Frames in the position of Frames zero and one and position of Frames three and four.

1.2.2 Offset position

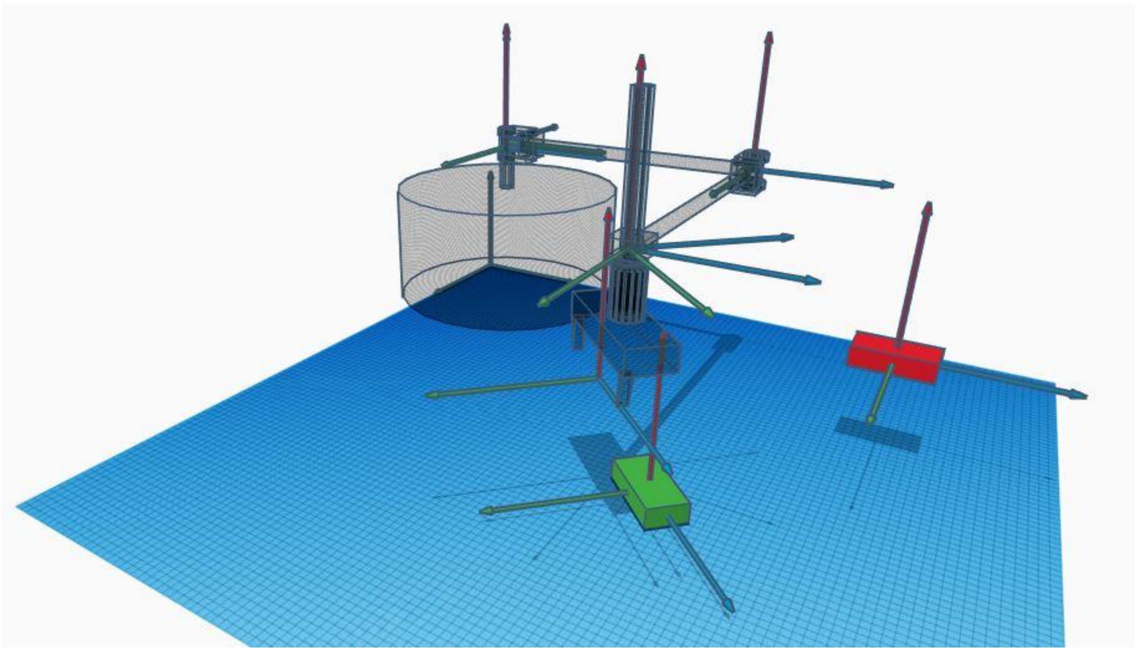

Figure 2: Mechanism with random (offset) value joint parameters. Difference between Frames Zero and one only to point out that there are two Frames.

1.2.3 Explain if the DH rules fit for both of the configurations

As all the frames are attached to the pose of the links of the mechanism, regardless of the position they get, the relative pose of every frame to its correspondent link will remain the same. The features that are modified after a movement of the mechanism are the global pose of the frames and the relative pose among those. This means that, for example, the axis X_1 will always be parallel to the link #1 regardless of the joint values. So the DH rules remain valid.

Note: Pose is the combination of position and orientation.

1.3. Provide the modified DH parameter table of the SCARA manipulator you designed

Table 1: DH Table of the SCARA robot

i	α_{i-1}	a_{i-1}	d_i	Θ_i
1	0	0	0	Θ_1
2	0	L_1	0	Θ_2
3	0	L_2	d_3	0
4	0	0	0	Θ_4
T	0	0.03	0.135	0

We chose after some iterations L_1=L_2= 0.4m

1.4. Using the Symbolic Math toolbox, give the transformation matrix between link frames. (R1: Transformations between links)

Using the Math toolbox, we compute all the transformation matrices between the frames of the links (F_0, F_1, F_2, F_3, F_4 and F_T) and by multiplying them we obtain the transformation matrix from F_T to F_0:

```
TT0 =

[ cos(theta1 + theta2 + theta4), -sin(theta1 + theta2 + theta4), 0, (3*cos(theta1 + theta2 + theta4))/100 + (2*cos(theta1 + theta2))/5 + (2*cos(theta1))/5]
[ sin(theta1 + theta2 + theta4),  cos(theta1 + theta2 + theta4), 0, (3*sin(theta1 + theta2 + theta4))/100 + (2*sin(theta1 + theta2))/5 + (2*sin(theta1))/5]
[                              0,                               0, 1,                                                                            d3 - 27/200]
[                              0,                               0, 0,                                                                                      1]
```

Figure 3: Transformation matrix of the link frames in variable terms

1.5. Give the homogeneous transformation matrix for the robot when all the joint angles are set to zero. What is the position in X, Y and Z of the manipulator? (R2: HT with joints values =0)

R 2.1:
When $\Theta_1=\Theta_2=d_3=\Theta_4=0$ the homogeneous transformation matrix is:

```
HTinitial =

    1.0000         0         0    0.9300
         0    1.0000         0    0.1000
         0         0    1.0000    0.2250
         0         0         0    1.0000
```

Figure 4: initial homogenous transformation matrix

(Here we also added the displacement of the base)

R 2.2:
So we can conclude the position (x, y, z) of the manipulator when all the joint angles are set to zero:

```
Pm =

    0.9300
    0.1000
    0.2250
```

Figure 5: Initial postion of manipulator

2. ROBOTICS TOOLBOX

2.1. Model the robot using MATLAB robotics Toolbox. Give the code you used to create your model. (R3: Robot model in robotic toolbox)

```
L(1)= Link([0 0 0 0],'modified');
L(2)= Link([0 0 L1 0],'modified');
L(3)= Link('theta', 0, 'a', L2, 'alpha', 0,'modified');
L(4)= Link([0 0 0 0],'modified');
L(3).qlim=[-0.225,0];
%Defining serial SCARA manipulator
SCARA=SerialLink(L,'name','links')
configuration=SCARA.config
```

Figure 6: code for robot modelling

2.2. Plot the robot in the zero-angle position. (R4: Robot plot in zero position)

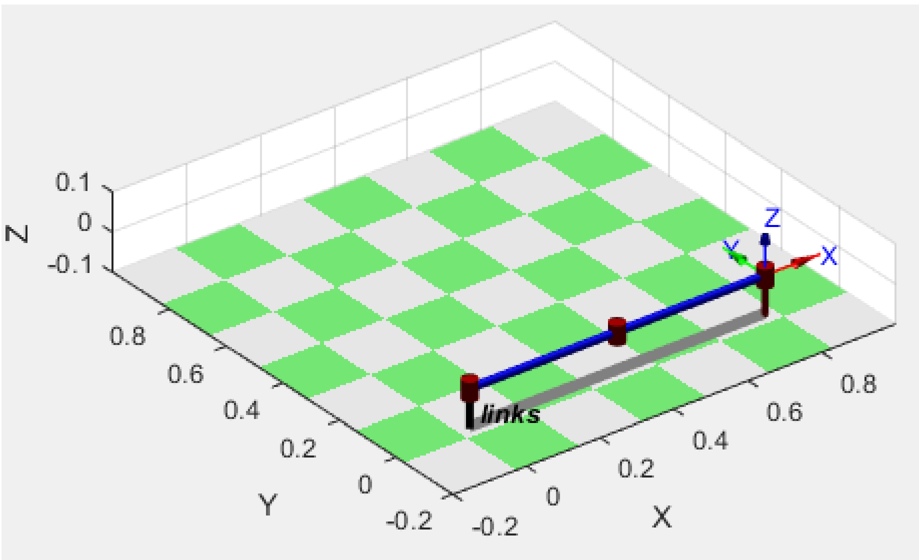

Figure 7: Robot in zero position without considering the base and the tool

3. FORWARD KINEMATICS

3.1. Determine tool transformation and apply it to the model you created.

Basically, it is applied by a translation regarding F_4. It starts from the origin of the common frames F_3 and F_4, the prismatic and last revolute joints, to the centered point of the gripper at the lowest point of its fingers (TCP).

3.2. Determine the base transformation and apply to the model

It consists of a translation from the bottom surface of the base centered in the specified coordinates to the shared origin of the frames F_0 and F_1.

3.3. Provide the tool transformation. (R5: Tool transformation)

```
gripper =

    1.0000         0         0    0.0300
         0    1.0000         0         0
         0         0    1.0000   -0.1350
         0         0         0    1.0000
```

Figure 8: Tool transformation matrix

3.4. Provide the base transformation (R6: Base transformation)

```
base =

    1.0000         0         0    0.1000
         0    1.0000         0    0.1000
         0         0    1.0000    0.3600
         0         0         0    1.0000
```

Figure 9: Base transformation matrix

3.5. Plot the robot in different configurations and provide: joint angles, end frame homogeneous transformation matrix and position (x, y, z) of the tool tip

3.5.1. Zero position (R7: Robot in zero position)

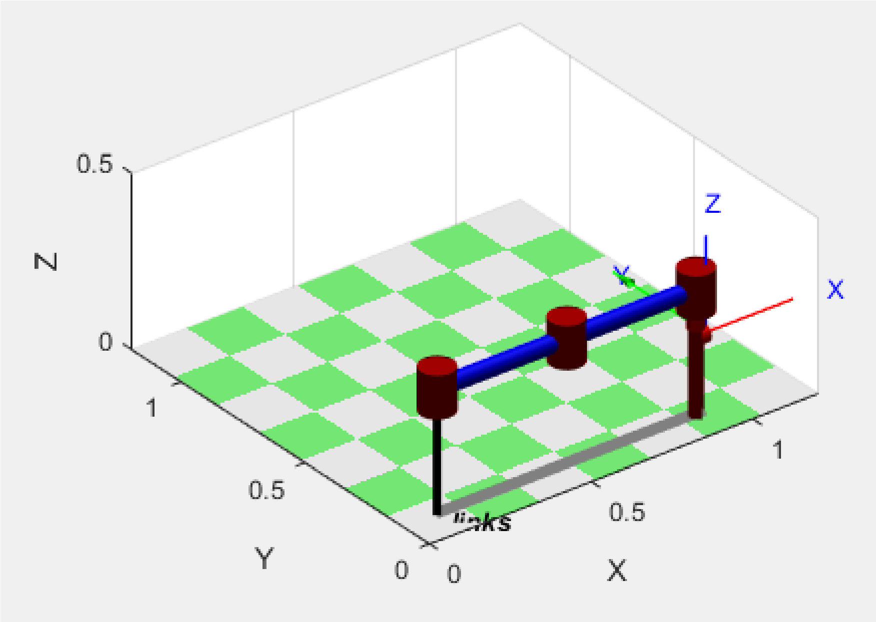

Figure 10: Robot in zero position

3.5.2. Offset position (R8: Robot in offset position)

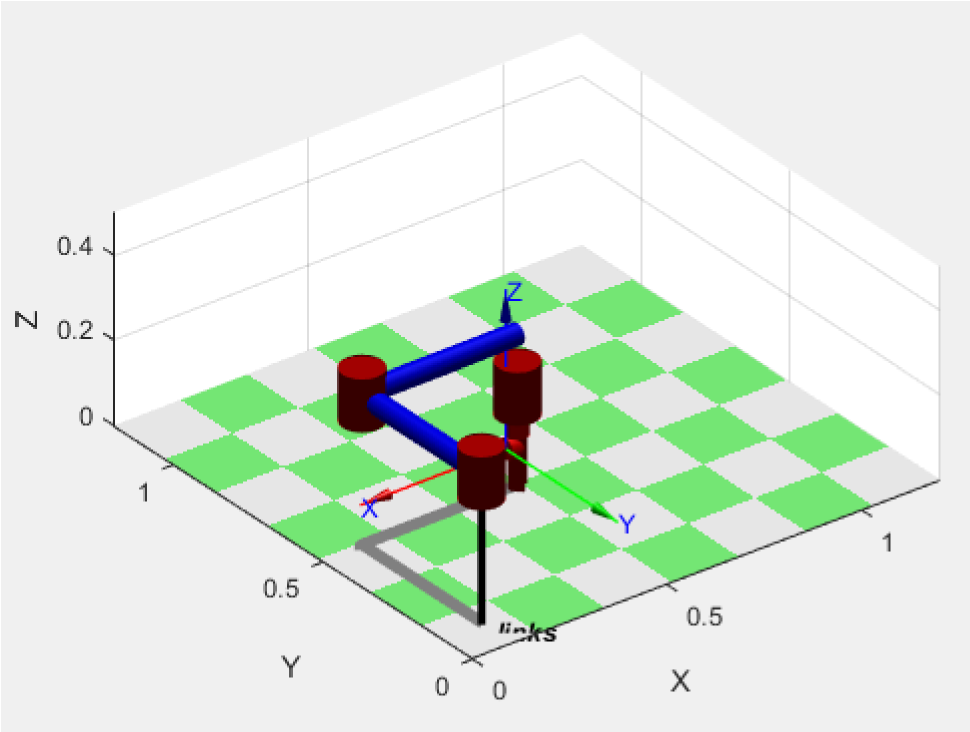

Figure 11: Robot in offset position

4. INVERSE KINEMATICS

4.1. Solve the inverse kinematics for the two work pieces (red and green)

```
%R9: Inverse kinematics for red and green pieces

disp('R9: Inverse kinematics for red and green pieces')
%Creating a target
%Green piece
greenp=transl(.6,.6,0)*trotz(pi/4);
trplot(greenp,'frame','1','color','g','length', 0.2)
hold on
waitforbuttonpress
%red piece
redp=transl(.3,.8,.1)*trotz(pi/2);
trplot(redp,'frame','2','color','r','length', 0.2)
hold on
waitforbuttonpress
%Solve inverse kinematic of green position
Qgreen=SCARA.ikine(greenp,[0,0,0,0],[1,1,1,0,0,1])
%check
SCARA.plot(Qgreen,'workspace',W,'scale', 2)
waitforbuttonpress
%Solve inverse kinematic of red position
Qred=SCARA.ikine(redp,[0,0,0,0],[1,1,1,0,0,1])
%check
SCARA.plot(Qred,'workspace',W,'scale', 2)
waitforbuttonpress
```

Figure 12: Code of inverse kinematics

4.2. For each case, provide the code you used to calculate them, the joint angles values and the robot plotted in those positions. *(R9: Inverse kinematics for red and green pieces)*

Please see Figure 12 for the MATLAB code.

Table 2: Joint values for inverse kinematics

Q=	Θ_1	Θ_2	d_3	Θ_4
Qgreen=	0.2238	1.1233	-0.2250	-0.5616
Qred=	0.7733	1.0148	-0.1250	-0.2173

Plotting the inverse kinematics position for the green piece:

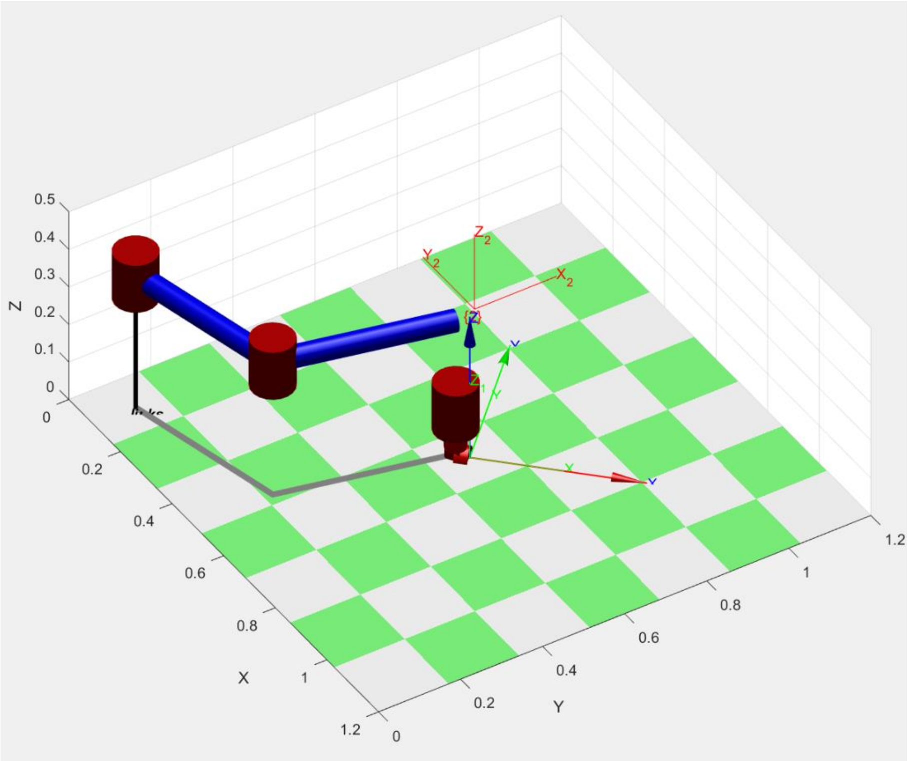

Figure 13: Plotting inverse kinematics for the green piece

Plotting the inverse kinematics position for the red piece:

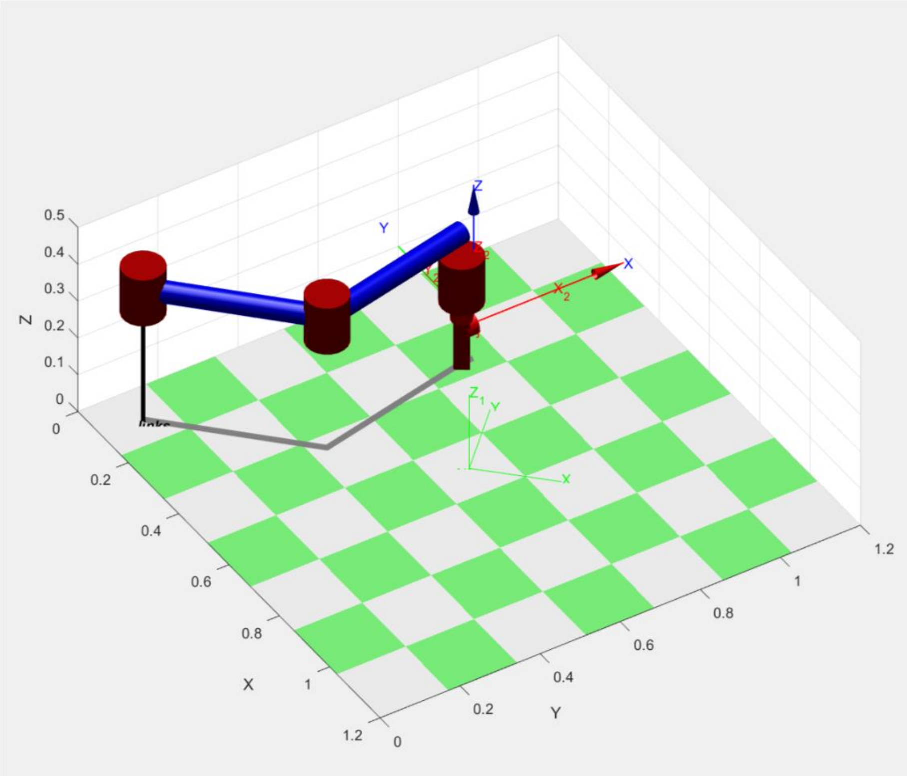

Figure 14: Plotting inverse kinematics for the red piece

4.3. How can you verify that the joints angles that you computed, actually will drive the robot tool to the desired position and orientation?

Using those joint parameters in the forward kinematic computation and check if the cartesian position result is similar to the positions of the pieces.

5. VERIFYING THE ROBOTIC TOOLBOX FORWARD KINEMATICS

5.1. Compute the forward kinematics with the Robotic Toolbox using fkine, for the green work piece. Provide the HT matrix with the location of the end effector. *(R10: Forward kinematics using fkine)*

```
HTgreenpiece_fkine =

    0.7071   -0.7071         0    0.6000
    0.7071    0.7071         0    0.6000
         0         0    1.0000    0.0000
         0         0         0    1.0000
```

Figure 15: HT matrix using fkine

5.2. Using the Symbolic toolbox and the transformation matrix between neighbor links, compute the forward kinematics of the manipulator when it is evaluated with the same joint angles, used in the previous point. *(R11: Forward kinematics using symbolic toolbox)*

```
HTgreenpiece2_symbolic_toolbox =

    0.7071   -0.7071         0    0.6000
    0.7071    0.7071         0    0.6000
         0         0    1.0000    0.0000
         0         0         0    1.0000
```

Figure 16: HT matrix using symbolic toolbox

5.3. Compare the results of both approaches and verify that they are equivalent.

Both of the above-mentioned approaches deliver identical results so that they are equivalent.