Report of Assignment: 01

ASE-9406 2017-01 Robot Manipulators: Modeling, Control and Programming

Submitted by: Prasun Biswas
ID:267948

Contents of the report:

- **Modified DH Parameters with explanation and figure of robot.**
- **Robotic toolbox with code snippets**
- **Forward kinematics and figures**
- **Inverse kinematics and figures**

Modified DH Parameters:

- SCARA robot has a combination of revolute and prismatic joints to achieve access points in three-dimensional workspace. This assignment requires to place the end effector tool in two different position which varies by height and change of orientation change by 45 degrees along z axis. So, to determine the link frame of SCARA robot the most important consideration was to encircle the two destination frames easily when the robot is in zero position and also various configurations within its reach. The length of the prismatic tool was selected in such a way that it can reach the height of the two destinations frame with ease hence selecting length more the necessary to keep a safer threshold. The first revolute joint could be placed on the bottom of the first link just at the adjacent top of the base but, I choose to place on the end of the link. Both works just fine but, my choice is easier for me to calculate DH parameters. The second revolute joint could be placed little higher than the height of the previous revolute joint using offset but, here also I kept the two revolute joints on same level which is also the same level of the base of the prismatic joint placed at the tip of the third link.

- Drawing of the robots in two different position using Tinkercad is below:

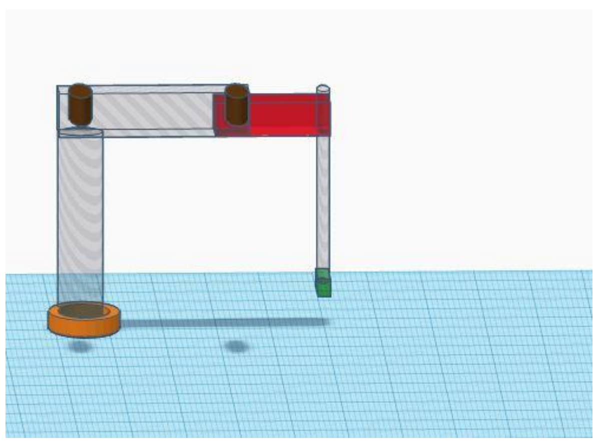


Figure: robot at zero position

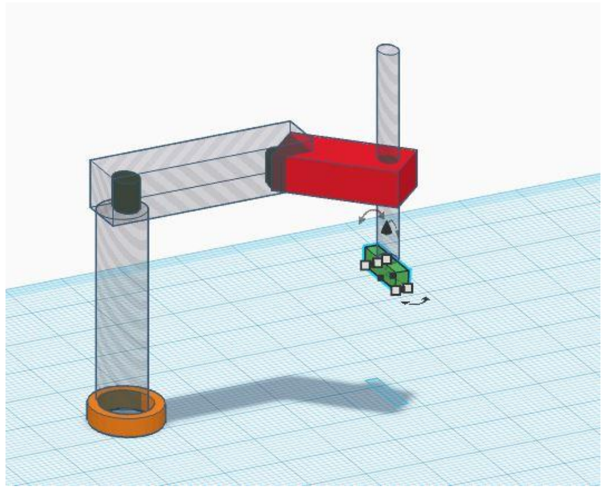


Figure: robot at offset position

- From different options to keep mine a good consideration was the shift of base to a height of 0.2 unit. The length of the first two links connected with the first two revolute joints respectively is 0.5 unit and 0.4, achieving a total length of 0.9 at zero position of the robot. This length can easily reach any of the green and red piece which are located respectively at (0.6,0.6,0) and (0.3,0.8,0.1) easily. Although the length of the links can be varied to acquire the same reach, considering the first link is the strongest to carry out the weight of the following links it is good idea not to select the length of links that follows the first one much larger in proportion than the first one. I kept the height of the first frame that connects the base to first revolute joint as 0.4 unit and kept the height of prismatic arm as 0.8 unit so that it can reach ground from that height effortlessly. I consider that the end effector of the robot is also a revolute joint placed on the tip of the prismatic joint arm so that it can align with any rotation along z axis while z axis of itself is facing downwards along direction of the movement of the prismatic joint.
- DH parameter fits for both case for joints. These parameters depends on relationships of two adjacent joints which keeps the parameter fixed regardless of respective position of each other.

- Modified DH parameter table:
  The following table contains the modified DH parameter for my version of SCARA robot that is demonstrated on the picture above.

	Theta	D	'a'	Alpha	R/P
First link	0	0.4	0	0	0
Second link	0	0	0.5	0	0
Third link	0	0	0.4	pi	1
Fourth link	0	0	0	0	0

- Transformation matrix between the links:

```
Ti2iminus =

[              cos(theta),             -sin(theta),           0,             a]
[ cos(alpha)*sin(theta), cos(alpha)*cos(theta), -sin(alpha), -d*sin(alpha)]
[ sin(alpha)*sin(theta), sin(alpha)*cos(theta),  cos(alpha),  d*cos(alpha)]
[                     0,                     0,           0,             1]
```

- transformation matrixes for different joints:

```
t21 =

[ 1, 0, 0,   0]
[ 0, 1, 0,   0]
[ 0, 0, 1, 2/5]
[ 0, 0, 0,   1]


t32 =

[ 1, 0, 0, 1/2]
[ 0, 1, 0,   0]
[ 0, 0, 1,   0]
[ 0, 0, 0,   1]


t43 =

[ 1,  0,  0, 2/5]
[ 0, -1,  0,   0]
[ 0,  0, -1,   0]
[ 0,  0,  0,   1]
```

- Homogeneous transformation matrix:

```
t14 =

[ 1,  0,  0, 9/10]
[ 0, -1,  0,    0]
[ 0,  0, -1,  2/5]
[ 0,  0,  0,    1]
```

The code that generated the matrixes above is given below.

```
% syms alpha theta d a
%disp('generate transformation matrix')

%  Ti2iminus=trotx(alpha)*transl([a 0 0])*trotz(theta)*transl([0 0 d])
% Ti2iminus=simplify (Ti2iminus)
%
%  t21 = subs(Ti2iminus,[alpha theta d a],[0 0 0.4 0])
% t32 = subs(Ti2iminus,[alpha theta d a],[0 0 0 0.5])
% t43 = subs(Ti2iminus,[alpha theta d a],[pi 0 0 0.4])
%
% t14 =t21*t32*t43

%-----------------------------------
```

Figure: code for transformation matrix.

Robotics Toolbox:

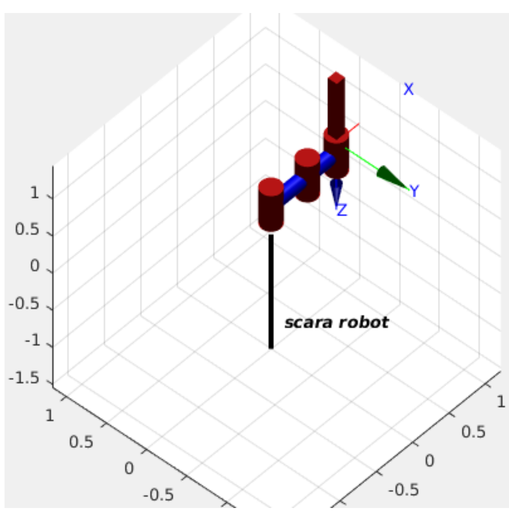
the code used to model the robot is inserted below.

```
L(1)=Link([0 0.4 0 0 0],'modified')
L(2)=Link([0 0 0.5 0 0],'modified')
L(3)=Link([0 0 0.4 pi 1],'modified')
L(4)=Link([0 0 0 0 0],'modified')
%fixing limit for prismatic joint>
L(3).qlim=[0 0.8]


disp('Link frames')

%Joining all the links%
SCARA=SerialLink(L,'name','scara robot');
% % SCARA.base=transl(0.1,0.1,0.2);
```

- plot of the robot in the zero-angle position:



Forward kinematics:

- Code snippet to determine tool transformation:

```
SCARA.tool=transl(0, 0, -0.135)% modified for checking
```

To determine the tool transformation, I assumed that the tip of the tool is extended towards z-axis from the end of the prismatic joint arm. The value I used is arbitrary but could be changed to match any tool definition accordingly. The goal of my assumption was to align the end effector with the alignment of the origin of co-ordinate frame of red and green boxes.

- code snippet to determine base transformation:

```
SCARA.base=transl(0.1,0.1,0.2);%value for z axis 0.2 instead of 0 to represent the heitht of base
```

This transformation has three values for x, y and z axis. The first two value transforms the base in horizontal plane. The base has a height of 0.2 so, I used value of z as 0.2 to include the change of height of the base.

- Tool transformation:

```
SCARA =

scara robot:: 4 axis, RRPR, modDH, slowRNE
+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|        0.4|          0|          0|          0|
|  2|         q2|          0|        0.5|          0|          0|
|  3|          0|         q3|        0.4|    3.14159|          0|
|  4|         q4|          0|          0|          0|          0|
+---+-----------+-----------+-----------+-----------+-----------+
tool:    t = (0, 0, -0.135), RPY/xyz = (0, 0, 0) deg
```

- Base transformation:

```
scara robot:: 4 axis, RRPR, modDH, slowRNE
+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|        0.4|          0|          0|          0|
|  2|         q2|          0|        0.5|          0|          0|
|  3|          0|         q3|        0.4|    3.14159|          0|
|  4|         q4|          0|          0|          0|          0|
+---+-----------+-----------+-----------+-----------+-----------+
base:    t = (0.1, 0.1, 0.2), RPY/xyz = (0, 0, 0) deg
```

- Joint angles and end frame homogeneous transformation matrix:

The following figures show the different position of robot along with joint angle and position of end effector tool generate using robotic toolbox.

At offset position Joint value (-42.2, -68.2, 0.285, 0) and end effector position (0.265, -0.730, 0.247)

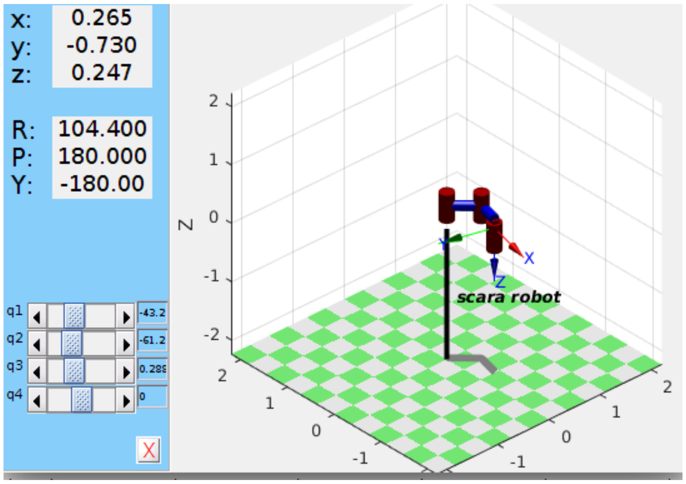


Figure: robot with offset value.

At zero position Joint value (0, 0, 0, 0) and end effector position (0.900, 0.000, 0.535)

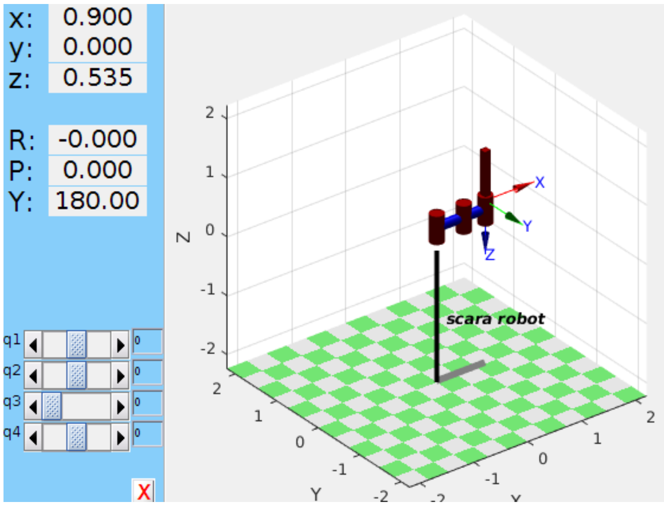


Figure: robot at zero position.

- Forward kinematics for co-ordinate of green box:

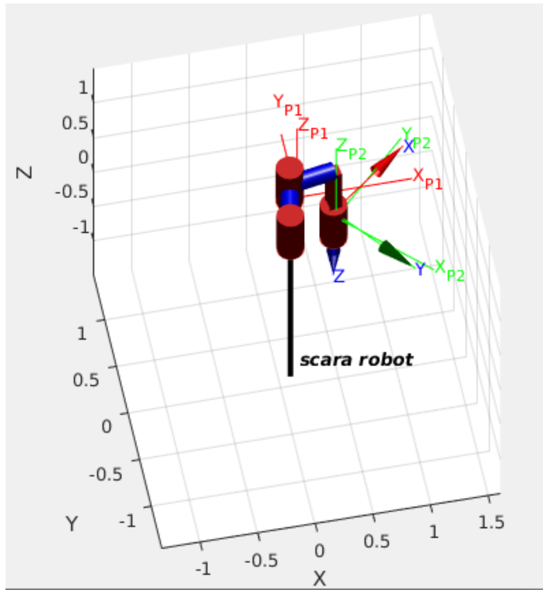


Figure: forward kinematics.

- Forward kinematics for co-ordinate of red box:

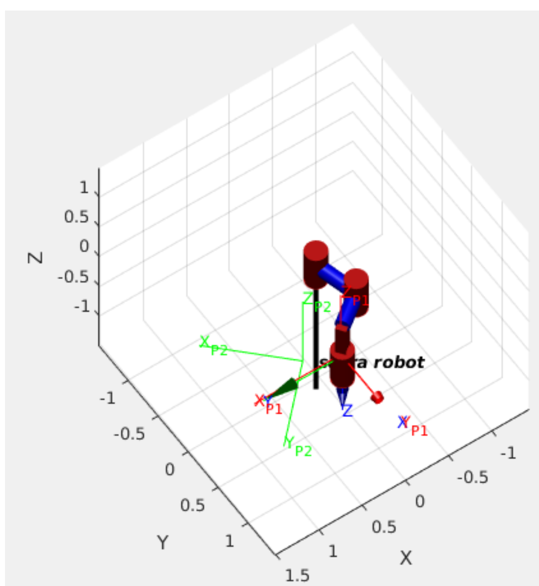


Figure: Inverse kinematics.

Inverse kinematics:

The following two figures shows the result of the inverse kinematics that obtained the end position to align the origin of co-ordinate of frames of red and green boxes

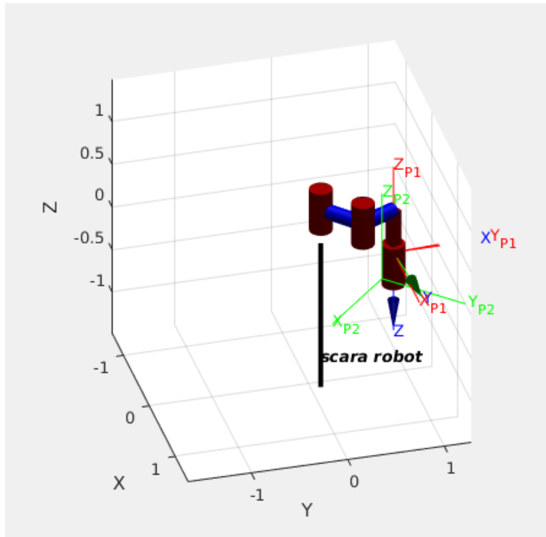


Figure: plot of inverse kinematics for red box.

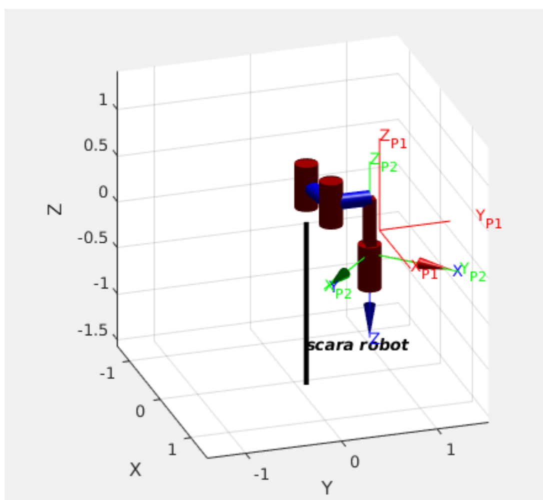


Figure: plot of inverse kinematics for green box.

- Code snippet for inverse kinematics:

```
% %inverse kinematics for red box
q1=SCARA.ikine(transl([0.3, 0.8, 0.1]),[2*pi, pi, pi],'mask',[1 1 1 0 0 0.01]);
hold on
SCARA.plot(q1,'workspace',[-2 2 -2 2 -2 2],'reach',1)
% w=waitforbuttonpress
%disp('inverse kinematics for green box')

% %inverse kinematics for green box
q2=SCARA.ikine(transl([0.6, 0.6, 0])*trotz(-pi/4),[2*pi, pi, pi],'mask',[1 1 1 0 0 0.033]);
hold on
SCARA.plot(q2,'workspace',[-3 3 -3 3 -3 3],'reach',1)
% w=waitforbuttonpress
%disp('run animation')
```

- The process of verification:

Using inverse kinematics, the joint angles are generated. These values can be used to generate transformation matrix and also for forward kinematics. If the value matches then the process is verified as a working model.

Verification of robotic toolbox forward kinematics:

transformation matrix after using forward kinematics:

```
TgFK =
    0.7223    0.6916         0    0.5298
    0.6916   -0.7223         0    0.7027
         0         0        -1     0.007
         0         0         0         1
```

Location of end effector: (x, y, z,) = (1.0, 0.1, 0.735)

It is possible to verify from the generated plot of kinematics which adjusts its arms to get in the right position and alignment.