22I-2327 MOHAMMAD ROHAAN SEC-A

REPORT OF ASSIGNMENT# 01

❾ ROBOTIC ARM SIMULATION

DH Table:

JOINTS	LINK LENGTH	ALPHA	D	THETA
1	A1	-90	D1	THETA1
2	A2	0	0	THETA2
3	A3	0	0	THETA3
4	0	90	D4	THETA4

Code For Transformation Matrix (Pose)

- `prompt = {'Enter theta1 (degrees):', 'Enter theta2 (degrees):', 'Enter theta3 (degrees):', 'Enter theta4 (degrees):'};`
- `theta1 = deg2rad(theta1);`
- `theta2 = deg2rad(theta2);`
- `theta3 = deg2rad(theta3);`
- `theta4 = deg2rad(theta4);`
- `pose = app.forwardKinematics(theta1, theta2, theta3, theta4);`
- `endEffectorPos = pose(1:3, 4);`
- `disp(endEffectorPos);`

- var = sprintf('[%.1f %.1f %.1f %.1f\n %.1f %.1f %.1f %.1f\n %.1f %.1f %.1f %.1f\n %.1f %.1f %.1f %.1f]', ...
- pose(1,1), pose(1,2), pose(1,3), pose(1,4), ...
- pose(2,1), pose(2,2), pose(2,3), pose(2,4), ...
- pose(3,1), pose(3,2), pose(3,3), pose(3,4), ...
- pose(4,1), pose(4,2), pose(4,3), pose(4,4));
- if theta(i) < qmin || theta(i) > qmax
- sprintf('Joint %d exceeds its limits! Valid range: [%.1f, %.1f]’)
- figure('Name', 'SUMO_ROBO', 'NumberTitle', 'off', 'WindowState', 'maximized');
- show(puma560, theta, 'Frames', 'off’);
- app.TextArea.Value = {output_String};
- app.TextArea.Visible = 'on';
- pause(5);
- app.TextArea.Visible = 'off';
- puma560 = loadrobot('puma560', 'DataFormat', 'row', 'Gravity', [0 0 -9.81]);

Code For Forward Kinematics

- prompt = {'Enter theta1 (degrees):', 'Enter theta2 (degrees):', 'Enter theta3 (degrees):', 'Enter theta4 (degrees):'};
- theta1 = deg2rad(theta1);
- theta2 = deg2rad(theta2);
- theta3 = deg2rad(theta3);
- theta4 = deg2rad(theta4);

```
JN = [theta1, theta2, theta3, theta4, 0, 0];
for i = 1:length(JN)
qmin = JN(i, 1);
if JN(i) < qmin
sprintf('Joint %d exceeds its limits!’);
return;
end
EN = 'link4’;
position = transl(EN);
if app.isSingular(JN)
sprintf(‘LOST ONE ANGLE', position(1), position(2), position(3));
end
output_String = sprintf('\n\n End Effector Position:\n\nX = %.2f\nY = %.2f\nZ = %.2f', position(1), position(2), position(3));
app.TextArea.Value = {output_String};
app.TextArea.Visible = 'on';
app.TextArea.Visible = 'off';
figure('Name', 'SOMO', 'NumberTitle', 'off', 'WindowState', 'maximized');
show(p560, JN);
```

Code For Inverse Kinematics

- prompt = {'Enter X position:', 'Enter Y position:', 'Enter Z position:’};
- r = sqrt(x^2 + y^2);
- if r < 0.5 || r > 1
- sprint('Position is outside the radial limits ');
- return;
- elseif theta < -90 || theta > 90
- sprint('Position is outside the angular limits ');
- return;
- elseif z < 0
- sprint('Z-position must be non-negative.', 'Workspace Error');
- return;
- end
- JN = ik('link6’, EN, weights, zeros(1, 6));
- output_String = sprintf(rad2deg(jointAngles(1)), rad2deg(jointAngles(2)), rad2deg(jointAngles(3)), rad2deg(jointAngles(4)), rad2deg(jointAngles(5)), rad2deg(jointAngles(6)));
- app.TextArea.Visible = 'on';
- pause(5);
- app.TextArea.Visible = 'off';
- figure('Name', ‘SUMO_ROBO', 'NumberTitle', 'off', 'WindowState', 'maximized');
- show(p560, jointAngles);
-

Code For Animation

```
• dh_table = {
• '1', '0', '0.2', '0.5', '-90';
• '2', '0', '0', '0.5', '0';
• '3', '0', '0', '0.5', '0';
• '4', '0', '0.1', '0', '-90';
• '5', '0', '0', '0', '90';
• '6', '0', '0', '0', '0'
• };
• dh = uifigure('Name', 'DH Table', 'NumberTitle', 'off');
• dh.WindowState = 'maximized';
• T0 = eye(4);
• T1 = transl(1, 2, 3) * rpy2tr(0.6, 0.8, 1.4);
• figure;
• trplot(T0);
• hold on;
• trplot(T1);
• T_current = transl(S(idx, :)) * rpy2tr(0.6, 0.8, 1.4);
• trplot(T_current);
```

Code For Exit

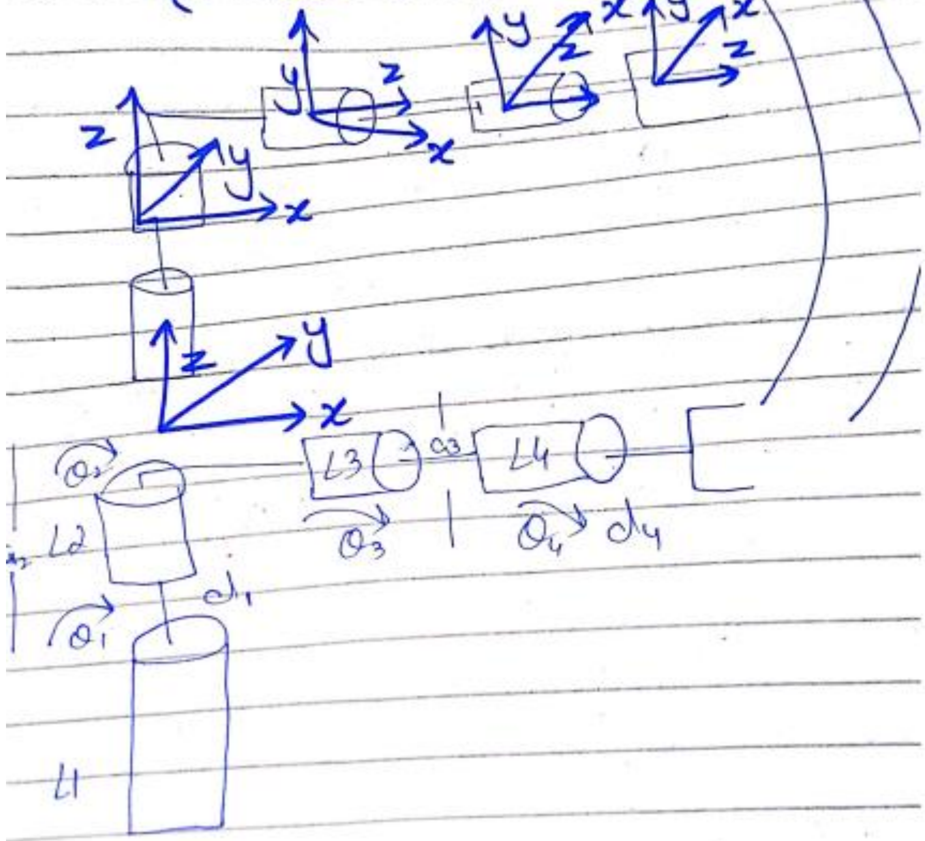
```
function ExitButtonPushed(app, event)
delete(app.UIFigure);

end
```

DH Table:-

Joints	link length(a)	α	d	θ
1)	a_1	-90	d_1	θ_1
2)	a_2	0	0	θ_2
3)	a_3	0	0	θ_3
4)	0	-90	d_4	θ_4
5)	0	-90	0	θ_5
6)	0	0	d_6	θ_6

⁎ Hand Written Report



Valid Test Cases For Transformation Matrix(Pose):

Test case	Theta 1	Theta 2	Theta 3	Theta 4
1	0	30	0	0
2	-45	45	0	0
3	45	0	0	0
4	30	30	30	0
5	-30	15	10	0

Invalid Test Cases For Transformation Matrix(Pose):

Test case	Theta 1	Theta 2	Theta 3	Theta 4
1	100	100	0	0
2	95	85	88	0
3	45	110	0	0
4	0	50	-100	0
5	-120	150	90	0

Valid Test Cases For Forward Kinematics:

Test case	Theta 1	Theta 2	Theta 3	Theta 4
1	30	-45	20	10
2	0	45	45	45
3	50	0	50	50
4	-30	-60	10	0
5	15	45	15	45

Invalid Test Cases For Forward Kinematics:

Test case	Theta 1	Theta 2	Theta 3	Theta 4
1	180	-50	40	20
2	-170	-100	10	50
3	30	-250	20	40
4	10	-60	230	60
5	90	-45	10	200

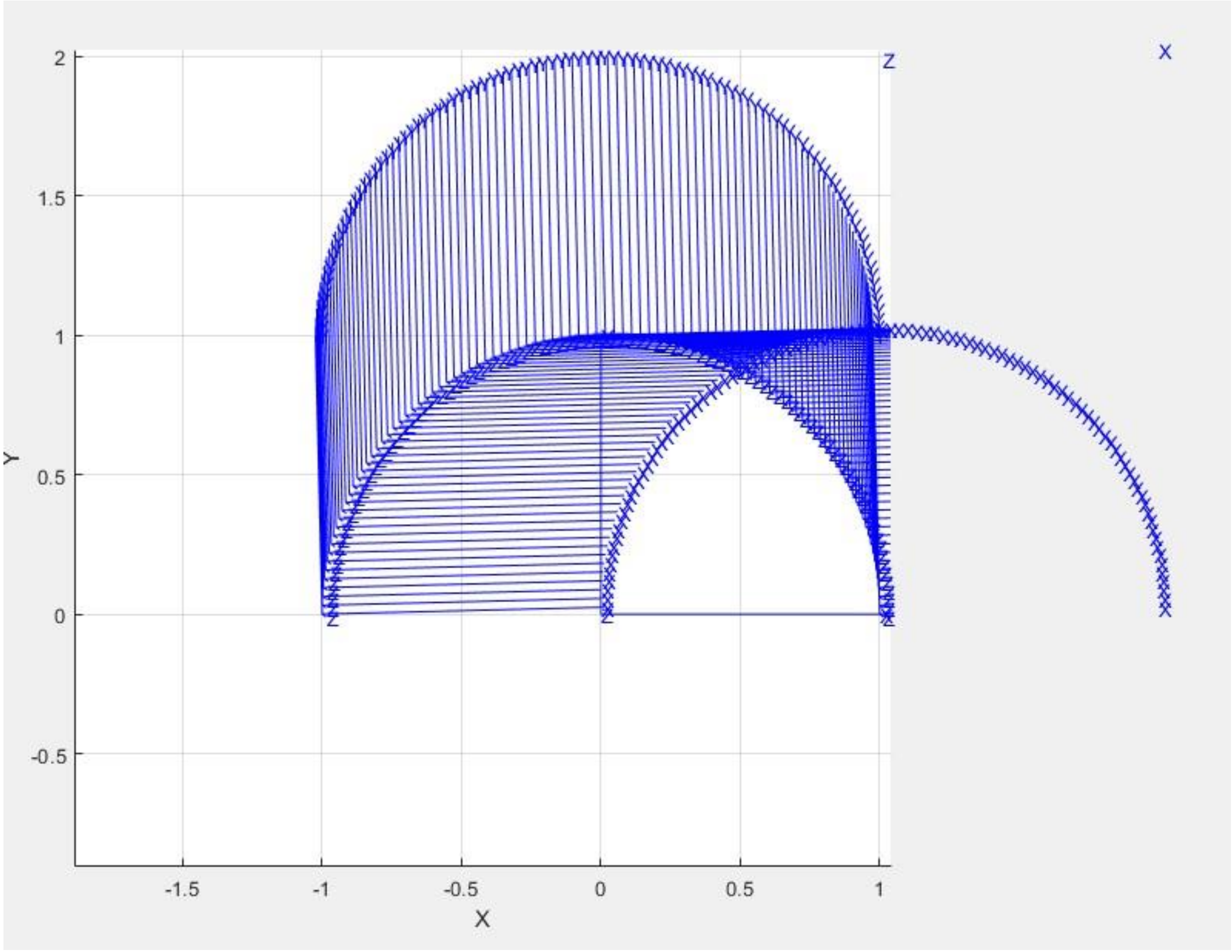
Valid Test Cases For Inverse Kinematics:

Test case	X	Y	Z
1	0.7	0.5	0.2
2	1.0	0.0	0.5
3	0.5	0.5	0.0
4	0.3	0.4	0.4
5	0.6	-0.8	0.3

Invalid Test Cases For Inverse Kinematics:

Test case	X	Y	Z
1	0.4	0.2	0.0
2	1.1	0.0	0.5
3	0.5	0.5	-0.1
4	0.8	1.0	0.0
5	0.6	-0.8	-0.1

Animation (Reach Every point Of Work Space)



2
1.5
1
0.5
0
-0.5
Y
-1.5
-1
-0.5
0
0.5
1
X
Z
X
Z
Z
Z