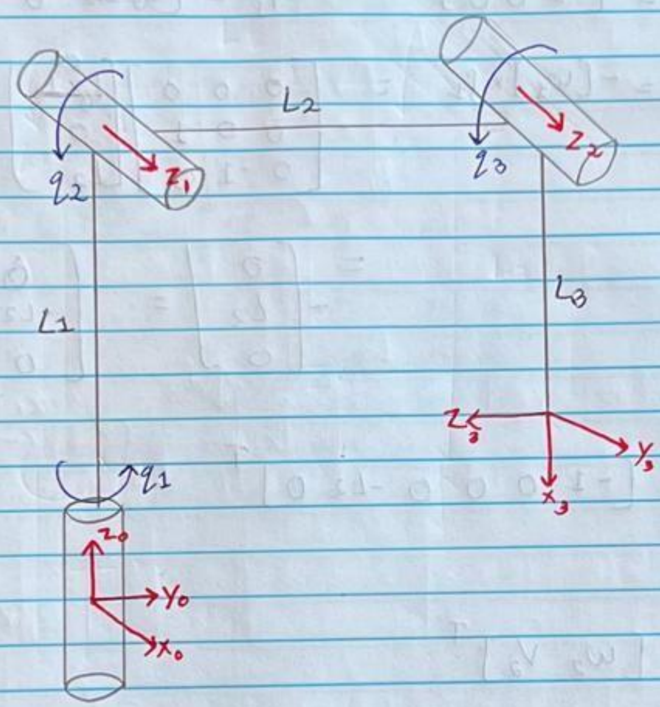
Section2:

Problem-1:

A].

The calculations for defining the home configuration (M) and the screw axis in body (endeffector) of the given 3-DOF robot are as follows:
Please take note that L1 = L2 = L3 = 0.3m.

HW3 - problem - 1



$$M = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \qquad P = \left[0 \;\; L_2 \;\; (L_1 - L_3) \right]^T$$

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & L_2 \\ -1 & 0 & 0 & (L_1 - L_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

→ screw axis calculation in body (end-effector) frame.

→ $S_1 = [w_1 \; V_1]^T$

$w_1 = [-1 \; 0 \; 0]^T$ $\quad P_1 = [\cancel{L_3\, L_1}\;0 \;\; 0 \;\; L_2]^T$

$$v_1 = -[w_1] \times P_1 = -\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \cancel{L_3 L_1}\;0 \\ 0 \\ L_2 \end{bmatrix}$$

$$= -\begin{bmatrix} 0 \\ L_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -L_2 \\ 0 \end{bmatrix}$$

$S_1 = [-1 \; 0 \; 0 \; 0 \; -L_2 \; 0]^T$

→ $S_2 = [w_2 \; V_2]^T$

$w_2 = [0 \; 1 \; 0]^T$ $\quad P_2 = [-L_3 \;\; 0 \;\; L_2]^T$

$$v_2 = -[w_2] \times P_2 = -\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -L_3 \\ 0 \\ L_2 \end{bmatrix} = -\begin{bmatrix} L_2 \\ 0 \\ +L_3 \end{bmatrix} = \begin{bmatrix} -L_2 \\ 0 \\ -L_3 \end{bmatrix}$$

$S_2 = [0 \; 1 \; 0 \; -L_2 \; 0 \; -L_3]^T$

→ $S_3 = [w_3 \; V_3]^T$

$w_3 = [0 \; 1 \; 0]^T$ $\quad P_3 = [-L_3 \;\; 0 \;\; 0]^T$

$$V_3 = -[w_3] \times p_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -L_3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ L_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -L_3 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & -L_3 \end{bmatrix}^T$$

$$S\_body = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & -L_2 & 0 \\ -L_2 & 0 & 0 \\ 0 & -L_3 & -L_3 \end{bmatrix}$$

I have calculated the homogenous transformation matrix of the home configuration (M) and used it to calculate the forward kinematics of the robot along with the

Screw axes in body frame (S_body) and "fkine" function developed for section-1 of this homework assignment.

```
T = fkine(S_body,M,q,'body');
```

Where: S_body is matrix of screw axes represented in body frame,
M is a matrix representing the home configuration, q is the matrix of joint variables.

`'body'` denotes the frame in which forward kinematics is to be calculated

**The code for the fkine function can be found in the fkine.m file of the submission.

B].

I calculated the body jacobian using the "jacobe" function developed for the section-1 of this homework assignment.

```
J_b = jacobe(S_space,M,q);
```

Where: S_space is matrix of screw axes represented in space frame,
M is a matrix representing the home configuration, q is the matrix of joint variables.
**The code for the jacob0 function can be found in the jacobe.m file of the submission.

C].

I calculated the analytical jacobian using the "jacoba" function developed for the section1 of this homework assignment.

```
J_a = jacoba(S_space,M,q);
```

Where: S_space is matrix of screw axes represented in space frame,
M is a matrix representing the home configuration, q is the matrix of joint variables.
**The code for the jacob0 function can be found in the jacoba.m file of the submission.

D].

I implemented the Damped-Least-Square method to find out the deltaQ and update it iteratively.

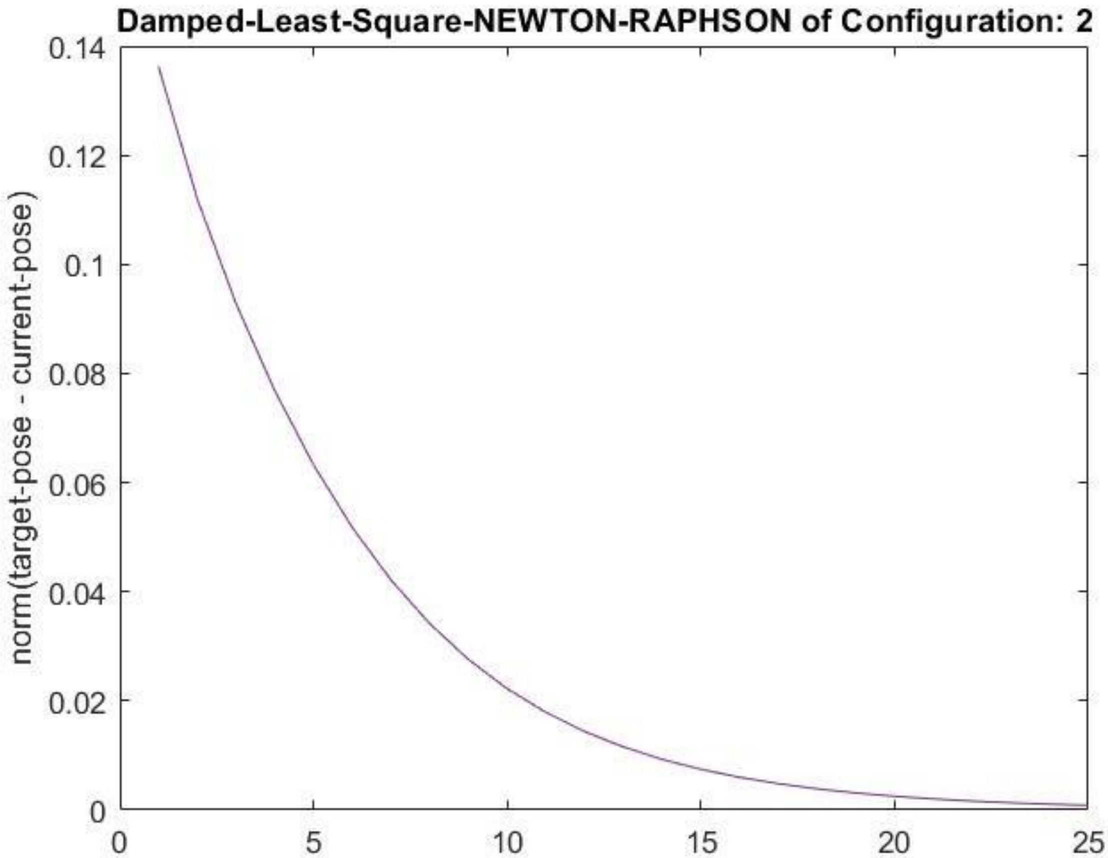
```
J_a = jacoba(S_space,M,currentQ);

J_star = J_a'*pinv(J_a*J_a' + (lambda^2)*eye(3));
```

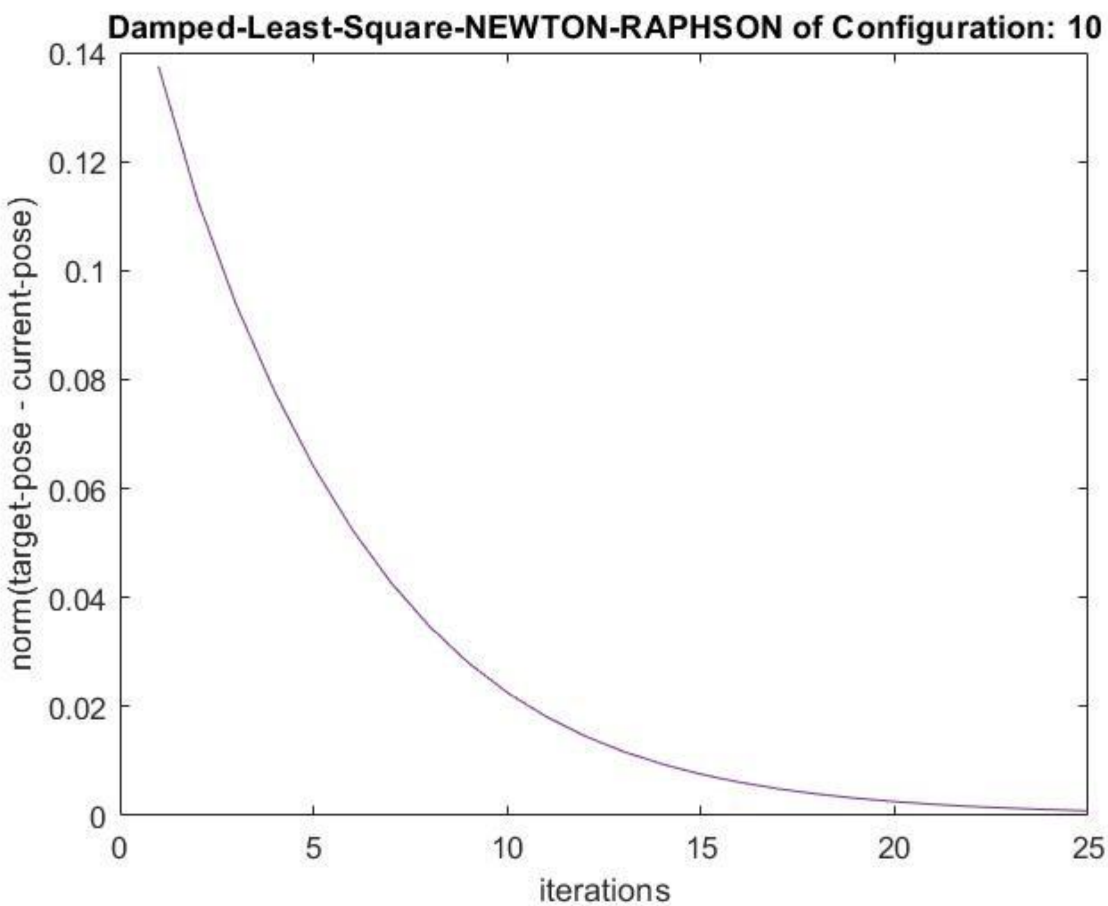
```
deltaQ = J_star*(targetPose - currentPose);
```

here,

- J_a is the analytical Jacobian
- J_star is the damped-least-square step
- deltaQ is a vector whose elements represent the change in joint values from the previous joint values.
- pinv(k) is Moore-Penrose pseudo-inverse of the matrix k.
- targetPose is twist of the final desire pose of the end effector.
- CurrentPose is the twist of the current pose (at current iteration) of the end effector. - lambda is the damping factor.
- eye(3) give 3*3 Identity matrix

After trial-and-error I found lamda = 0.5 to best fit the solutions and the error smoothly converges to zero when we use this lambda for all the configurations of the target pose. This is depicted by the error V/S iterations graphs as shown below.


Damped-Least-Square-NEWTON-RAPHSON of Configuration: 2
norm(target-pose - current-pose)
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0
5
10
15
20
25


Damped-Least-Square-NEWTON-RAPHSON of Configuration: 10
norm(target-pose - current-pose)
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0
5
10
15
20
25
iterations

E].

I ploted the Manipulability Ellipsoid of the end effector of the given robotic arm using the following line of code.

```
h = plot_ellipse(J_a*J_a',currentPose);
```

This function helps printing the ellipsoid using the analytical Jacibian (J_a) and the current pose of the end effector.

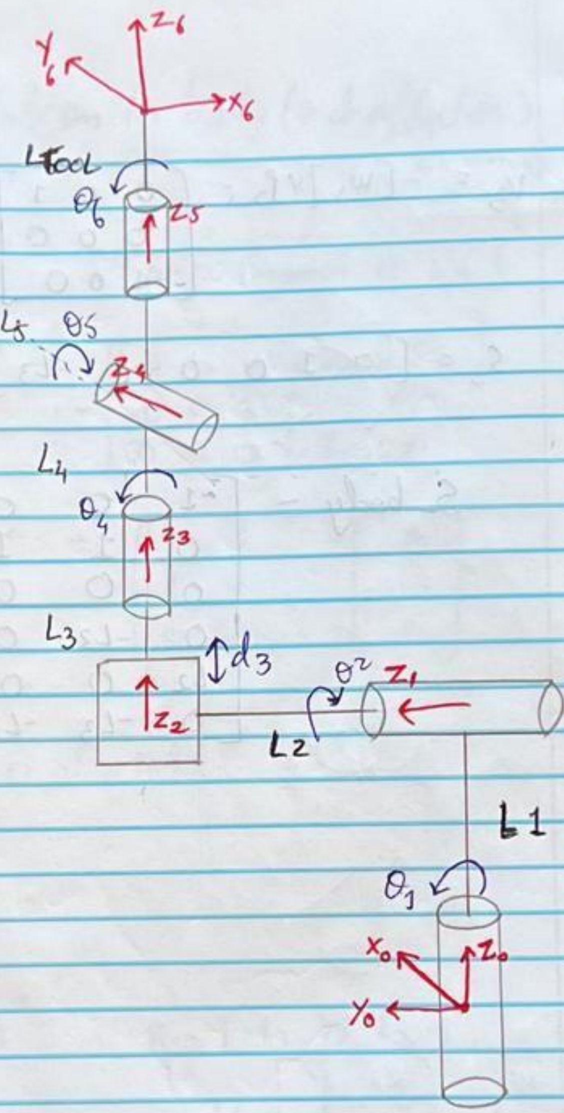
<u>Problem-1:</u>

A].

The calculations for defining the home configuration (M) and the screw axis in body (endeffector) of the given 6-DOF Stanford Manipulatorare as follows: Please take note that

L1 = 0.412m
L2 = 0.154m
L3 = 0m
L4 = 0m
L5 =0m
Ltool = 0.263m

HW3 problem 2



$L_1 = 0.412\ m$

$L_2 = 0.152\ m$

$L_3 = L_4 = L_5 = 0\ m \qquad L_{Tool} = 0.263\ m$

$$M = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad P = [0 \ \ L2 \ \ (L_1 + L_3 + L_4 + L_5 + L_{Tool})]$$

$$\therefore P = [0 \ \ L2 \ \ L1 + L_{Tool}]^T$$

$$\therefore M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & L2 \\ 0 & 0 & 1 & L1 + L_{tool} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

→ Calculating Screw axis in space frame →

→ $S_1 = [w_1 \ \ V_1]^T$

$w_1 = [0 \ 0 \ 1]^T \qquad P_1 = [0 \ 0 \ 0]^T$

$\therefore S_1 = [0 \ 0 \ 1 \ 0 \ 0 \ 0]^T$

→ $S_2 = [w_2 \ \ V_2]^T$

$w_2 = [0 \ 1 \ 0] \qquad P_2 = [0 \ 0 \ L1]^T$

$$V_2 = -[w_2]P_2 = -\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ L_1 \end{bmatrix} = -\begin{bmatrix} L1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -L1 \\ 0 \\ 0 \end{bmatrix}$$

$\therefore S_2 = [0 \ 1 \ 0 \ -L1 \ 0 \ 0]^T$

→ $S_3 = [w_3 \ \ V_3]^T$ (prismatic joint)

$w_3 = [0 \ 0 \ 0] \qquad V_3 = [0 \ 0 \ 1]$

$\therefore S_3 = [0 \ 0 \ 0 \ \ 0 \ 0 \ 1]^T$

$\rightarrow S_4 = [w_4 \;\; v_4]^T$

$w_4 = [0 \; 0 \; 1]^T \quad p_4 = [0 \; L_2 \; 0]^T$

$\therefore v_4 = -[w_4] p_4 = -\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ L_2 \\ 0 \end{bmatrix} = -\begin{bmatrix} -L_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix}$

$\therefore S_4 = [0 \; 0 \; 1 \; L_2 \; 0 \; 0]^T$

$\rightarrow S_5 = [w_5 \;\; v_5]^T$

$w_5 = [1 \; 0 \; 0]^T \quad p_5 = [0 \; L_2 \; L_1]^T$

$v_5 = -[w_5] p_5 = -\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ L_2 \\ L_1 \end{bmatrix} = -\begin{bmatrix} 0 \\ -L_1 \\ L_2 \end{bmatrix} = \begin{bmatrix} 0 \\ L_1 \\ -L_2 \end{bmatrix}$

$\therefore S_5 = [1 \; 0 \; 0 \; 0 \; L_1 \; -L_2]^T$

$\rightarrow S_6 = [w_6 \;\; v_6]^T$

$w_6 = [0 \; 0 \; 1] \quad p_6 = [0 \; L_2 \; 0]^T$

$v_6 = -[w_6] p_6 = -\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ L_2 \\ 0 \end{bmatrix} = -\begin{bmatrix} -L_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix}$

$\therefore S_6 = [0 \; 0 \; 1 \; L_2 \; 0 \; 0]^T$

→ $S_4 = [0\ 0\ 1]^T \quad P_4 = [0\ 0\ 0]^T$

$\therefore S_4 = [0\ 0\ 1\ 0\ 0\ 0]^T$

→ $S_5 = [0\ 1\ 0]^T \quad P_5 = [0\ 0\ L_{Tool}]^T$

$S_5 = [0\ 1\ 0\ L_{Tool}\ 0\ 0]^T \qquad V_5 = -[\omega_5]P_5 = \begin{bmatrix}0&0&-1\\0&0&0\\1&0&0\end{bmatrix}\begin{bmatrix}0\\0\\-L_{Tool}\end{bmatrix}$

$V_5 = [L_{Tool}\ 0\ 0]^T$

→ $S_6 = [0\ 0\ 1]^T \quad P_6 = [0\ 0\ 0]^T$

$S_6 = [0\ 0\ 1\ 0\ 0\ 0]^T$

∴ S_body =

$$\begin{bmatrix}0&-1&0&0&0&0\\0&0&0&0&1&0\\1&0&0&1&0&1\\0&0&0&0&L_{Tool}&0\\-L2&L_{Tool}&0&0&0&0\\0&0&1&0&0&0\end{bmatrix}$$

S_body =

$$\begin{bmatrix}0&-1&0&0&0&0\\0&0&0&0&1&0\\1&0&0&1&0&1\\0&0&0&0&Ltool&0\\-L2&Ltool&0&0&0&0\\0&0&1&0&0&0\end{bmatrix}$$

I have calculated the homogenous transformation matrix of the home configuration (M) and used it to calculate the forward kinematics of the robot along with the

Screw axes in body frame (S_body) and “fkine” function developed for section-1 of this homework assignment.

```
T = fkine(S_body,M,q,'body');
```

Where: S_body is matrix of screw axes represented in body frame,
M is a matrix representing the home configuration, q is the matrix of joint variables.

`'body'` denotes the frame in which forward kinematics is to be calculated

**The code for the fkine function can be found in the fkine.m file of the submission.

B].

I calculated the body jacobian using the “jacobe” function developed for the section-1 of this homework assignment.

```
J_b = jacobe(S_space,M,q);
```

Where: S_space is matrix of screw axes represented in space frame,
M is a matrix representing the home configuration, q is the matrix of joint variables.

**The code for the jacob0 function can be found in the jacobe.m file of the submission.

C].

I calculated the analytical jacobian using the “jacoba” function developed for the section1 of this homework assignment.

```
J_a = jacoba(S_space,M,q);
```

Where: S_space is matrix of screw axes represented in space frame,
M is a matrix representing the home configuration, q is the matrix of joint variables.

**The code for the jacob0 function can be found in the jacoba.m file of the submission.

D].

I implemented the Damped-Least-Square method to find out the deltaQ and update it iteratively.

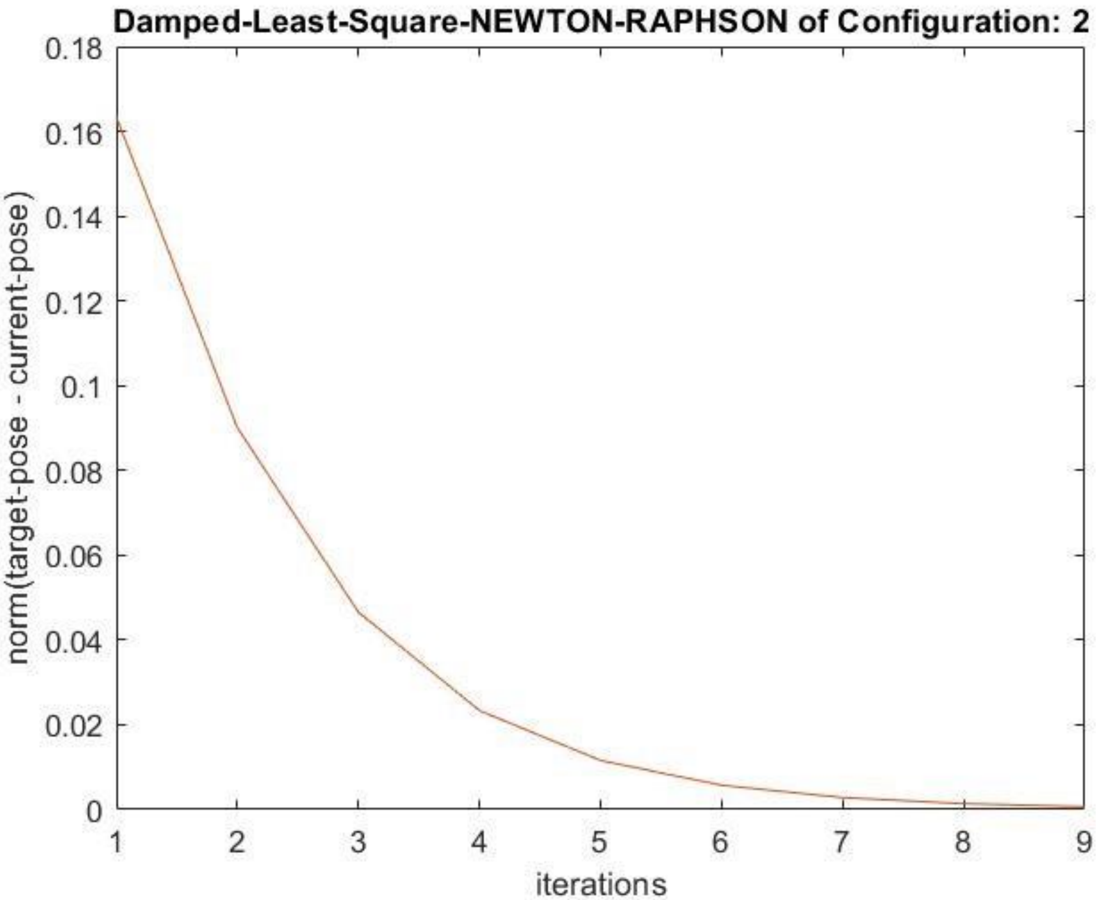
```
J_a = jacoba(S_space,M,currentQ);

J_star = J_a'*pinv(J_a*J_a' + (lambda^2)*eye(3));
```

```
deltaQ = J_star*(targetPose - currentPose);
```

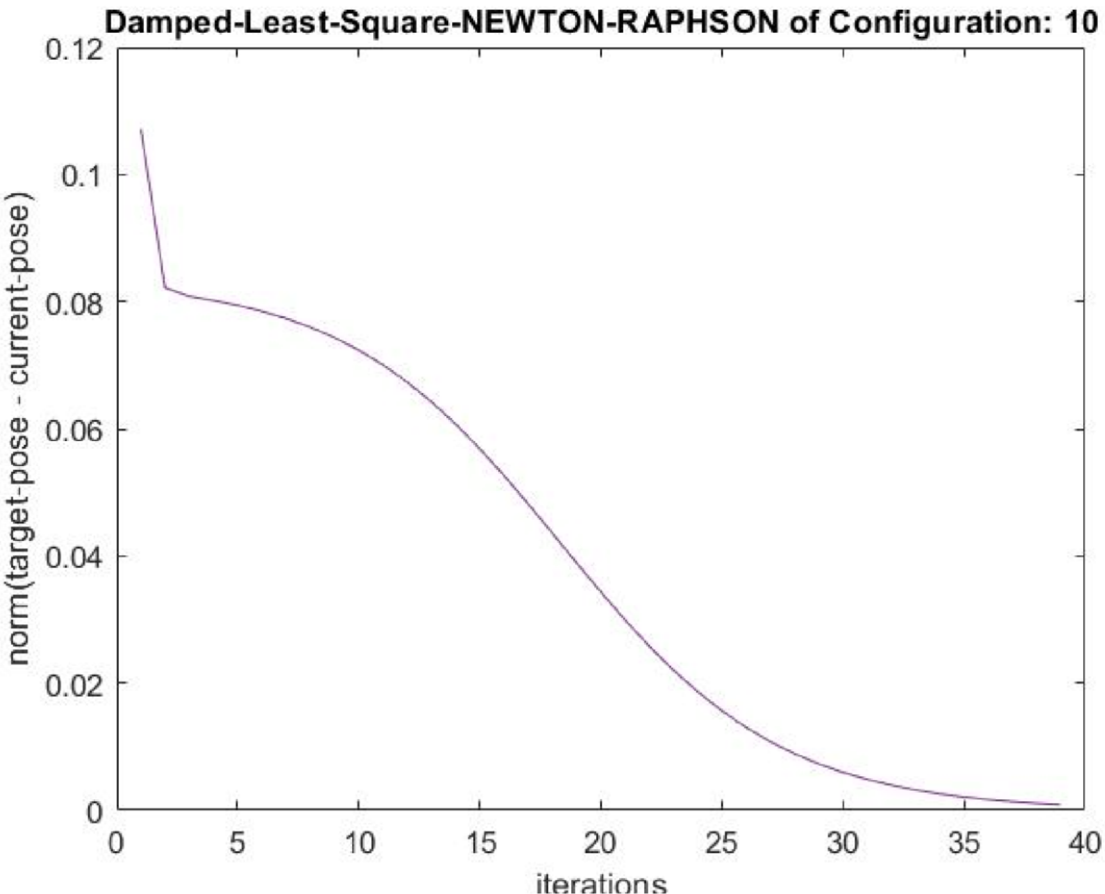
here,

- J_a is the analytical Jacobian
- J_star is the damped-least-square step
- deltaQ is a vector whose elements represent the change in joint values from the previous joint values.
- pinv(k) is Moore-Penrose pseudo-inverse of the matrix k.
- targetPose is twist of the final desire pose of the end effector.
- CurrentPose is the twist of the current pose (at current iteration) of the end effector. - lambda is the damping factor.
- eye(3) give 3*3 Identity matrix

After trial-and-error I found lamda = 0.5 to best fit the solutions and the error smoothly converges to zero when we use this lambda for all the configurations of the target pose. This is depicted by the error V/S iterations graphs as shown below.



Damped-Least-Square-NEWTON-RAPHSON of Configuration: 2
norm(target-pose - current-pose)
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
2
3
4
5
6
7
8
9
iterations



Damped-Least-Square-NEWTON-RAPHSON of Configuration: 10
norm(target-pose - current-pose)
0.12
0.1
0.08
0.06
0.04
0.02
0
0
5
10
15
20
25
30
35
40
iterations

**The path I traced to check the correctness of the inverse-kinematics solution was obtained by the following lines of code-

```
t = linspace(0, 2*pi, nTests);
x = 0.5 * cos(t); y = 0.5 *
sin(t);
z = 0.2 * ones(1,nTests) + 1.0; path
= [x; y; z];
```

E].

I ploted the Manipulability Ellipsoid of the end effector of the given robotic arm using the following line of code.

```
h = plot_ellipse(J_a*J_a',currentPose);
```

This function helps printing the ellipsoid using the analytical Jacibian (J_a) and the current pose of the end effector.