1.

a)

$p_{1x} = l_1c_1, p_{1y} = l_1s_1$

$p_{2x} = p_{1x} + l_2c_{12}, p_{2y} = p_{1y} + l_2s_{12}$

$p_{3x} = p_{1x} + p_{2x} + l_3c_{123}, p_{3y} = p_{1y} + p_{2y} + l_3s_{123}$

$p_{4x} = p_{1x} + p_{2x} + p_{3x} + l_4c_{1234}, p_{4y} = p_{1y} + p_{2y} + p_{3y} + l_4s_{1234}$

$$p_{cmx} = \frac{p_{1x} + p_{2x} + p_{3x} + p_{4x}}{4} = \frac{4\, l_1c_1 + 3l_2c_{12} + 2l_3c_{123} + l_4c_{1234}}{4}$$

$$p_{cmy} = \frac{p_{1y} + p_{2y} + p_{3y} + p_{4y}}{4} = \frac{4\, l_1s_1 + 3l_2s_{12} + 2l_3s_{123} + l_4s_{1234}}{4}$$

$$\sum_{i=1}^{4} m_i(p_i - p_{cm}) = 0$$

$$p_{cm} = \frac{1}{4}\sum_{i=1}^{4} p_i$$

p_{cm} is 4 kg.

b)

$$\frac{\partial p_4}{\partial \theta} = \begin{bmatrix} \frac{\partial P_4}{\partial \theta_1} & \frac{\partial P_4}{\partial \theta_2} & \frac{\partial P_4}{\partial \theta_3} & \frac{\partial P_4}{\partial \theta_4} \end{bmatrix}$$

$$= [z_0 \times (p_4 - p_0)\ \ z_1 \times (p_4 - p_1)\ \ \ z_2 \times (p_4 - p_2)\ \ z_3 \times (p_4 - p_3)]$$

c)

$$(\partial p\_4)/(\partial \theta\_1\ ) = \begin{bmatrix} -l_1s_1 - l_2s_{12} - l_3s_{123} - l_4s_{1234} \\ l_1c_1 + l_2c_{12} + l_3c_{123} + l_4c_{1234} \\ 0 \end{bmatrix}$$

$$\frac{\partial p_4}{\partial \theta_2} = \begin{bmatrix} -l_2s_{12} - l_3s_{123} - l_4s_{1234} \\ l_2c_{12} + l_3c_{123} + l_4c_{1234} \\ 0 \end{bmatrix}$$

$$\frac{\partial p_4}{\partial \theta_3} = \begin{bmatrix} -l_3s_{123} - l_4s_{1234} \\ l_3c_{123} + l_4c_{1234} \\ 0 \end{bmatrix}$$

$$\frac{\partial p_4}{\partial \theta_4} = \begin{bmatrix} -l_4s_{1234} \\ l_4c_{1234} \\ 0 \end{bmatrix}$$

$$\frac{\partial p_4}{\partial \theta} = \begin{bmatrix} -l_1s_1 - l_2s_{12} - l_3s_{123} - l_4s_{1234} & -l_2s_{12} - l_3s_{123} - l_4s_{1234} & -l_3s_{123} - l_4s_{1234} & -l_4s_{1234} \\ l_1c_1 + l_2c_{12} + l_3c_{123} + l_4c_{1234} & l_2c_{12} + l_3c_{123} + l_4c_{1234} & l_3c_{123} + l_4c_{1234} & l_4c_{1234} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$where\ s_{12} = \sin(\theta_1 + \theta_2)\,, c_{12} = \cos(\theta_1 + \theta_2)\,, etc.$

d)

$$\frac{\partial p_3}{\partial \theta} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} & 0 \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial p_2}{\partial \theta} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial p_1}{\partial \theta} = \begin{bmatrix} -l_1 s_1 & 0 & 0 & 0 \\ l_1 c_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

e)

$$\frac{\partial p_{cm}}{\partial \theta} = \frac{\frac{1}{4}\left(\sum_{i=1}^{4} \partial p_i\right)}{\partial \theta} = \frac{1}{4}\left(\frac{\partial p_1}{\partial \theta} + \frac{\partial p_2}{\partial \theta} + \frac{\partial p_3}{\partial \theta} + \frac{\partial p_4}{\partial \theta}\right)$$

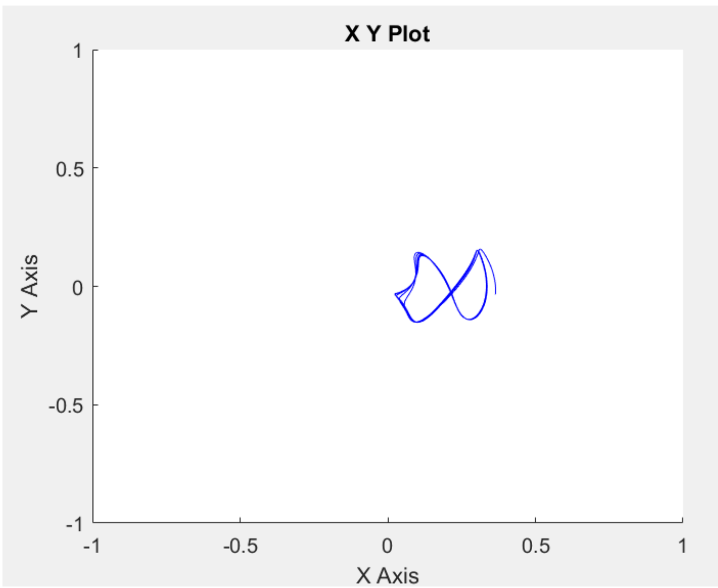
f)

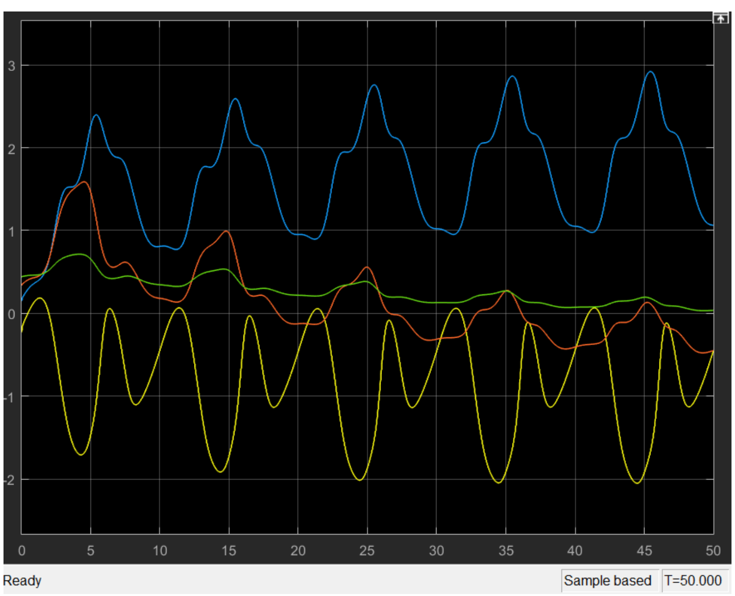
```
p0=[0;0;0];

p1=[links(1)*cos(theta(1));links(1)*sin(theta(1));0];
p2=[links(1)*cos(theta(1))+links(2)*cos(theta(1)+theta(2));links(1)*sin(theta(1))+links(2)*sin(theta(1)+theta(2));0];
p3=[links(1)*cos(theta(1))+links(2)*cos(theta(1)+theta(2))+links(3)*cos(theta(1)+theta(2)+theta(3));
    links(1)*sin(theta(1))+links(2)*sin(theta(1)+theta(2))+links(3)*sin(theta(1)+theta(2)+theta(3));
    0];
p4=[links(1)*cos(theta(1))+links(2)*cos(theta(1)+theta(2))+links(3)*cos(theta(1)+theta(2)+theta(3))...
    +links(4)*cos(theta(1)+theta(2)+theta(3)+theta(4));
    links(1)*sin(theta(1))+links(2)*sin(theta(1)+theta(2))+links(3)*sin(theta(1)+theta(2)+theta(3))...
    +links(4)*sin(theta(1)+theta(2)+theta(3)+theta(4));
    0];

Z=[0;0;1];

J4=[cross(Z,p4-p0) cross(Z,p4-p1) cross(Z,p4-p2) cross(Z,p4-p3)];
J3=[cross(Z,p3-p0) cross(Z,p3-p1) cross(Z,p3-p2) zeros(3,1)];
J2=[cross(Z,p2-p0) cross(Z,p2-p1) zeros(3,1) zeros(3,1)];
J1=[cross(Z,p1-p0) zeros(3,1) zeros(3,1) zeros(3,1)];

JP=0.25*(J4+J3+J2+J1);
JP=JP(1:2,:);
```

g)

$$\Delta\theta = \alpha J^T(\theta)\Delta x$$

```
alpha=2;
thetad = alpha*JP'*xd;
```



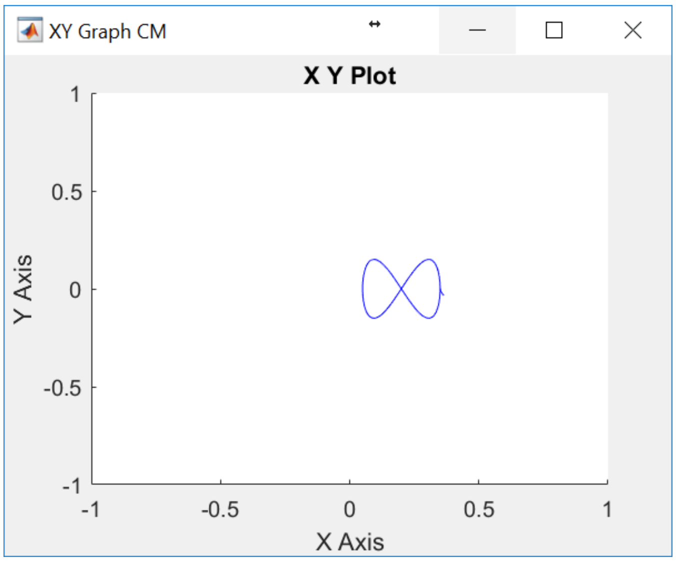


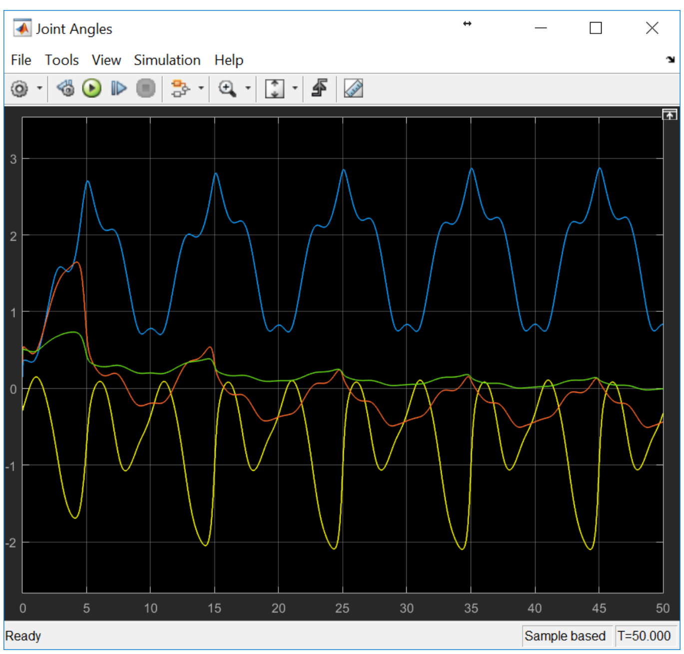
The Jacobian Transpose performs poorly, partly because the coefficients in Jacobian matrix is small.

h)

$$\Delta\theta = \alpha J^T(\theta)\left(J(\theta)J^T(\theta)\right)^{-1}\Delta x = J^{\#}\Delta x$$

```
alpha=0.5;
thetad = alpha*JP'*pinv(JP*JP')*xd;
```



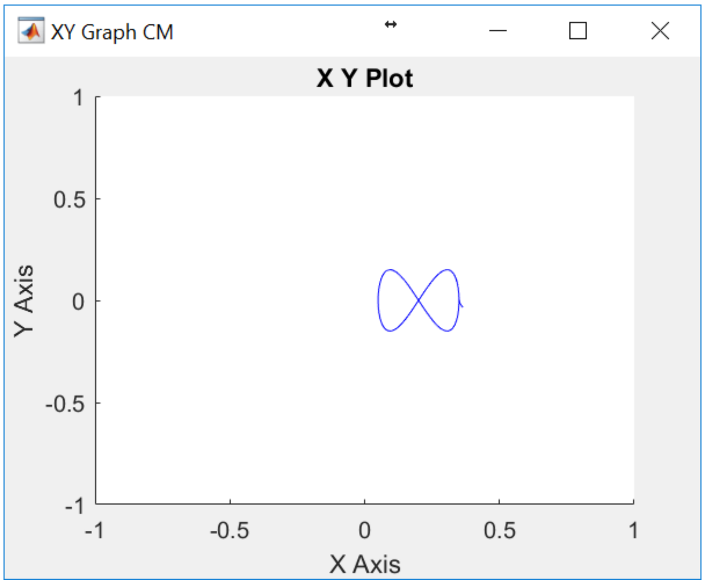


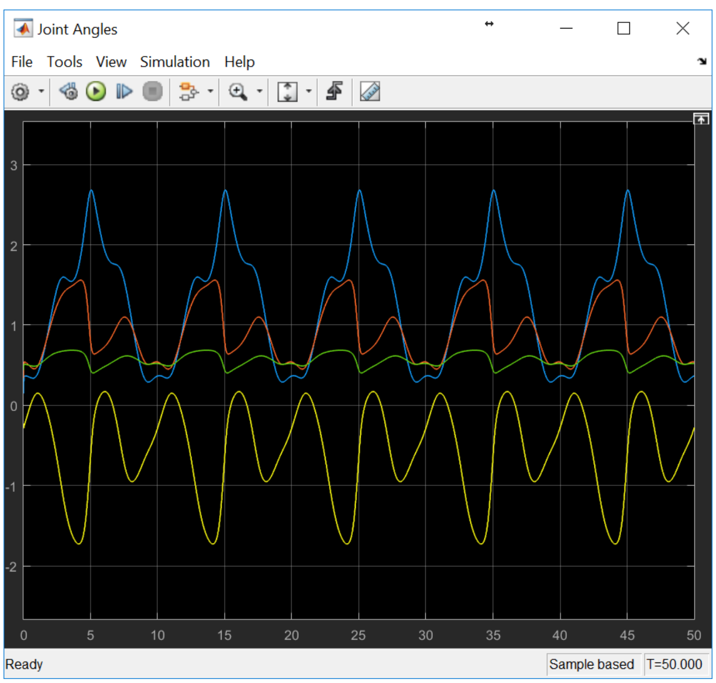
The pseudo-inverse for inverse kinematics performs better than the Jacobian Transpose. It forms a good figure-8 shape. However, it still takes roughly 20 seconds to form a stable figure-8 shape.

i)

$$\Delta\theta = \alpha J^{\#}\Delta x + (I - J^{\#}J)(\theta_O - \theta)$$

```
alpha=0.5;
J_sharp=JP'*pinv(JP*JP');
thetad = alpha*J_sharp*xd+(eye(4)-J_sharp*JP)*([0.5;0.5;0.5;0.5]-theta);
```





The pseudo-inverse with Null-space optimization for inverse kinematics performs very well. It forms a stable figure-8 very soon.

j)

$$F = \frac{1}{2}\Delta\theta^T W \Delta\theta + \lambda^T(\Delta x - J(\theta)\Delta\theta)$$

(1) $\frac{\partial F}{\partial \lambda} = 0 => \Delta x = J\Delta\theta$

(2) $\frac{\partial F}{\partial \Delta\theta} = 0 => \ W^T\Delta\theta = W\Delta\theta = J^T\lambda => \Delta\theta = W^{-1}J^T\lambda => J\Delta\theta = JW^{-1}J^T\lambda \quad => \lambda = (JW^{-1}J^T)^{-1}J\Delta\theta$

$$W^T = W$$

Insert (1) into (2):

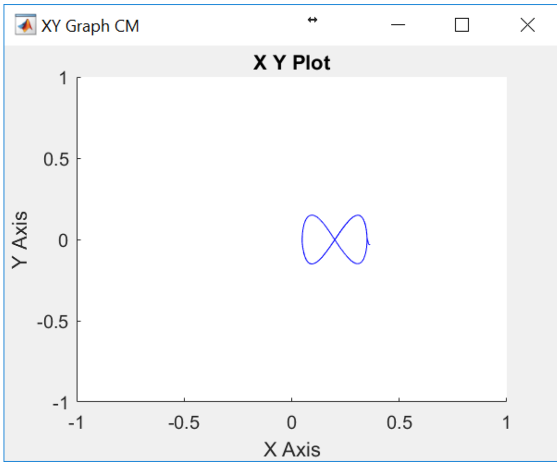
$$(3)\lambda = (JW^{-1}J^T)^{-1}\Delta x$$

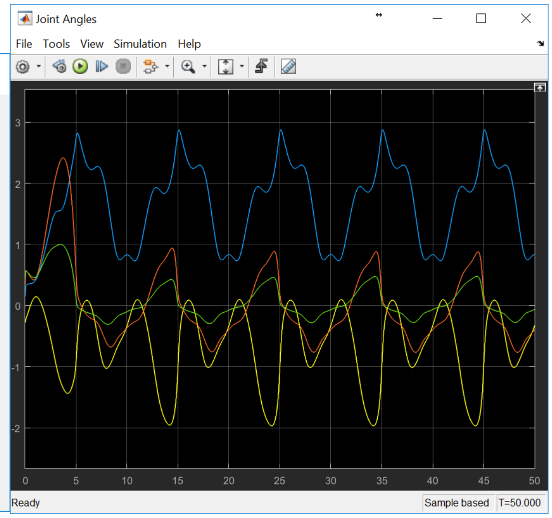
$$\Delta\theta = W^{-1}J^T\lambda$$

$$\Delta\theta = W^{-1}J^T(JW^{-1}J^T)^{-1}\Delta x$$

$$w = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$$

```
% 1-j)


% alpha=0.1;
% weight=[0.1,0.2,0.3,0.5];
% wMat=diag(weight,0);
% assignin('base','JP',JP);
% assignin('base','xd',xd);
% assignin('base','alpha',alpha);
%
% thetad = alpha* pinv(wMat)*JP'*pinv(JP*JP')*xd;
```





The weighted pseudo-inverse performs better than the original pseudo-inverse method. It forms a stable figure-8 quicker.

k)

Following similar procedure in j)

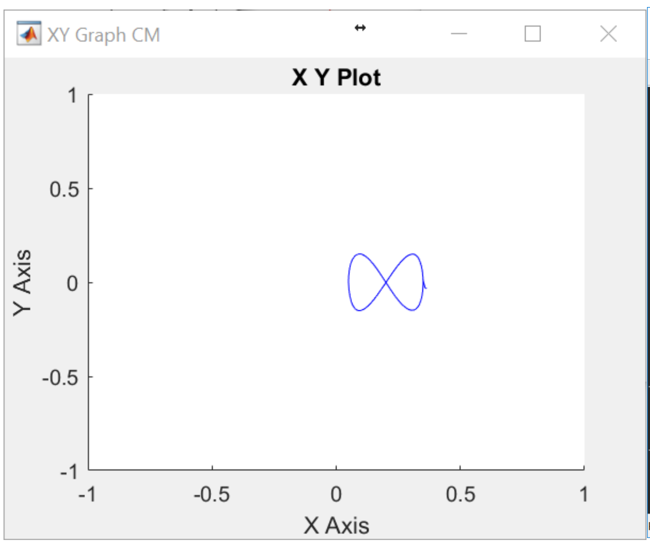
$$\Delta\theta = \alpha J^{\#}\Delta x + (I - J^{\#}J)(\theta_O - \theta)$$

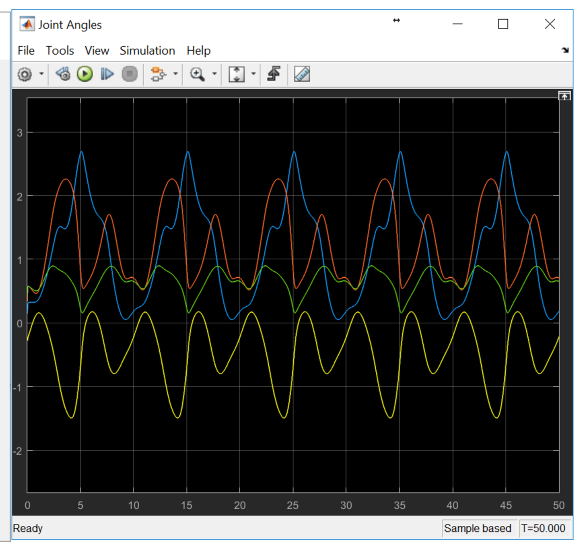
Where :

$$J^{\#} = W^{-1}J^{T}(\theta)\big(J(\theta)W^{-1}J^{T}(\theta)\big)^{-1}$$

$$w = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$$

```
% 1-k)

alpha=0.1;
weight=[0.1,0.2,0.3,0.5];
wMat=diag(weight,0);
assignin('base','JP',JP);
assignin('base','xd',xd);
assignin('base','alpha',alpha);

J_sharp=JP'*pinv(JP*JP');
thetad = alpha* pinv(wMat)* J_sharp*xd+pinv(wMat)*(eye(4)-J_sharp*JP)*([0.5;0.5;0.5;0.5]-theta);
```

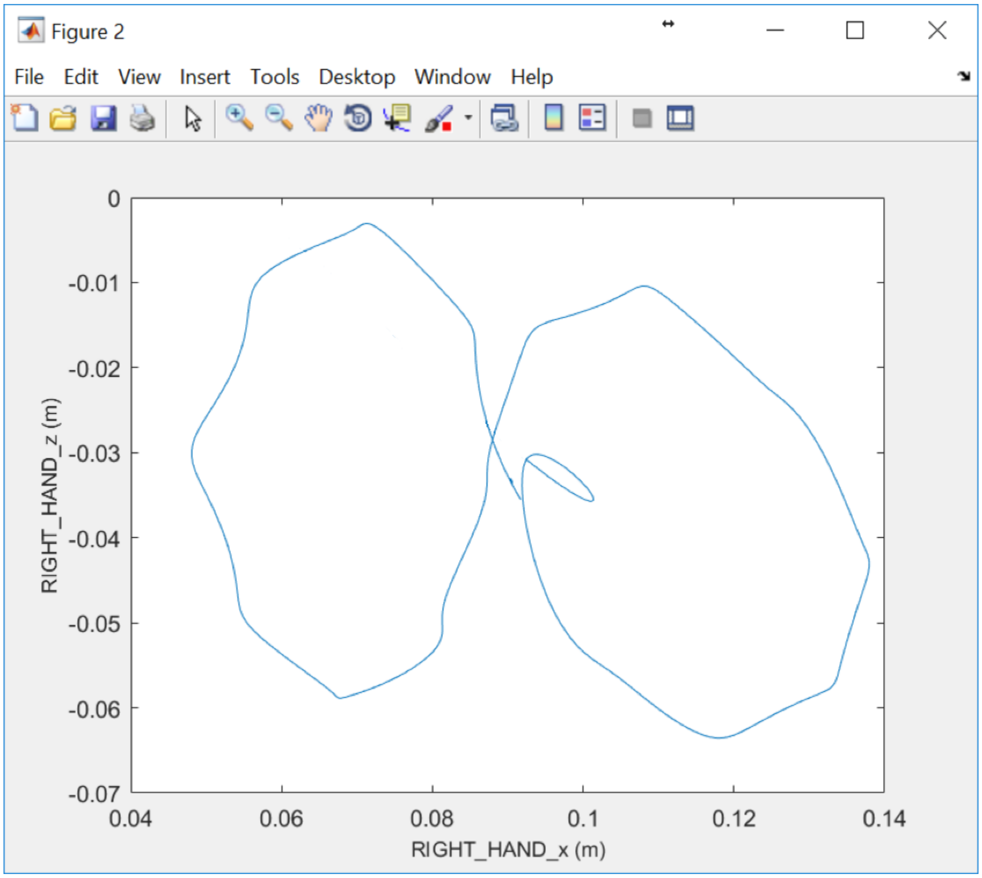




The weighted null-space optimization criterion method performs very well. It is because it puts more weight on the end effector and explicitly provide an optimization criterion on arm configurations.

2.

Phase plot:



Comment:

The performance is moderately good as from the phase plot. A figure-8 can be seen from the phase plot and the simulator. Since the kinematics model ignores the inertia and damp, we can error from the phase plot. Nonetheless, a clear figure-8 can be seen.