ABDULLAH ALMOSALAMI

ECE463

MODERN CONTROL

TEST # 2

03/29/2021

MONDAY

We will use the following: (gantry system)

$$s\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 39.2 & 0 & 0 \\ 0 & -49 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}F + \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}d$$

Feedback Controller Design

1) Design a servo compensator to track a sinusoidal setpoint $R(t) = \sin(1.5t)$ and reject a constant disturbance. $d = 10$. Give the resulting CL matrices.

We will need our servo comp to have poles at $0, \pm j1.5$.

$$\Rightarrow A_z = \begin{bmatrix} 0 & 1.5 & 0 \\ -1.5 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_z = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Augmented system will be: $s\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} A & 0 \\ B_zC & A_z \end{bmatrix}\begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix}u + \begin{bmatrix} 0 \\ -B_z \end{bmatrix}R$

Let's set the des_poles $= [-1, -2, -3, -4, -5, -6, -7] \Rightarrow$ place Poles (...)

$\Downarrow$

$$\boxed{\begin{aligned} K_x &= [624.6481,\ 353.8981,\ 80.7256,\ 52.7256] \\ K_z &= [414.6627,\ 462.6552,\ 228.5714] \end{aligned}}$$

$$A_{\text{CLOSED-LOOP}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -624 & -314 & -80 & -52 & -414 & -462 & -228 \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}$$

```
Acl =

         0         0    1.0000         0         0         0         0
         0         0         0    1.0000         0         0         0
 -624.6481 -314.6981  -80.7256  -52.7256 -414.6627 -462.6552 -228.5714
  624.6481  304.8981   80.7256   52.7256  414.6627  462.6552  228.5714
    1.0000         0         0         0         0    1.5000         0
    1.0000         0         0         0   -1.5000         0         0
    1.0000         0         0         0         0         0         0
```

```
>> Bcl

Bcl =

     0
     0
     0
    -1
    -1
    -1
```

2) Verify your control law on the linearized system.

- Plot when $R = \sin(t)$ and $d = 0$.
- " " $R = 0$ $d = 10$
- " " $R = \sin(1.5t)$ $d = 10$

(See attached $\Rightarrow$ Code at end of submission)

3) Verify on the nonlinear sim w/ $R = \sin(1.5t)$ and $d = 10$.

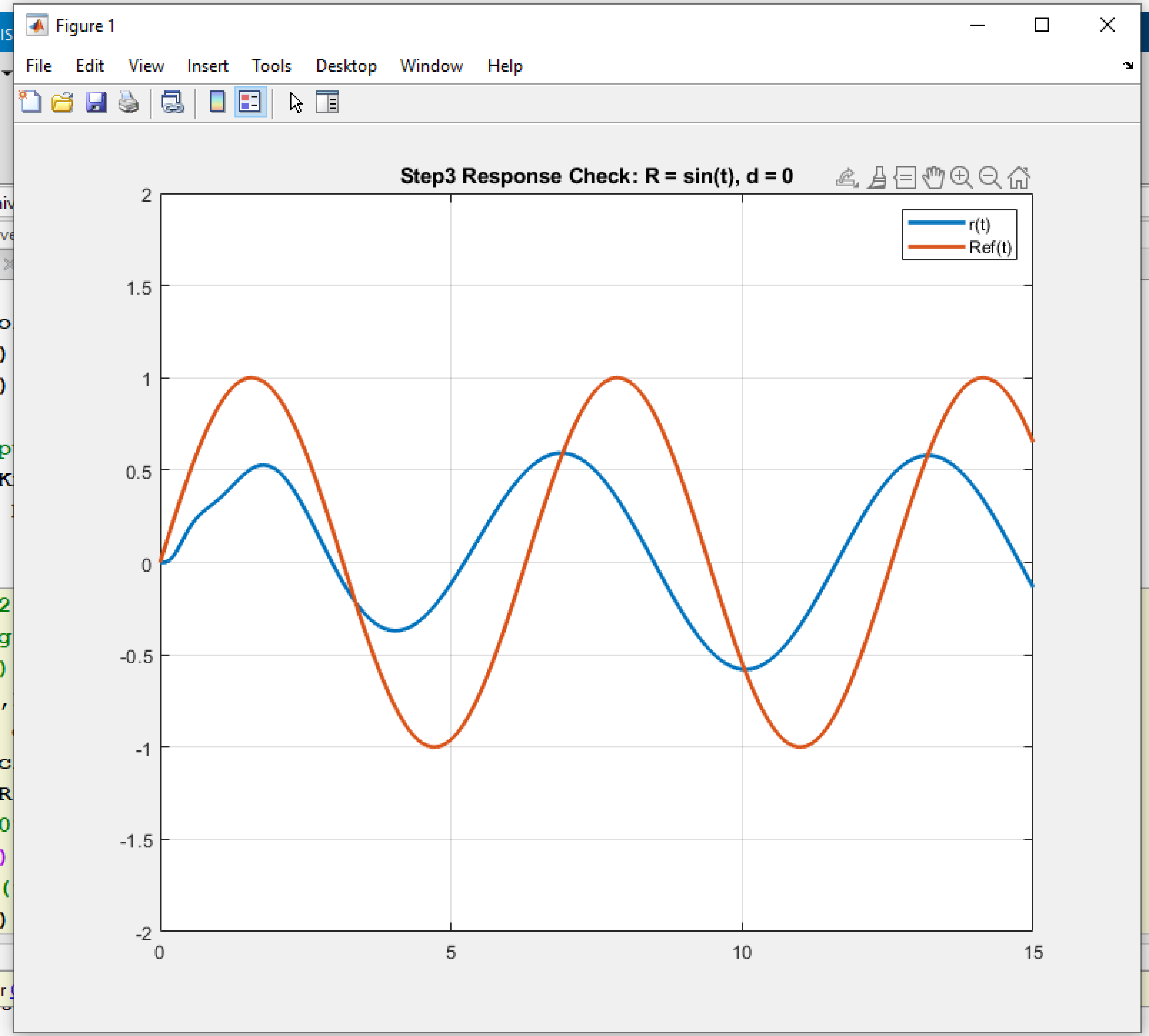
Figure 1
File Edit View Insert Tools Desktop Window Help
Step3 Response Check: R = sin(t), d = 0
r(t)
Ref(t)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
5
10
15

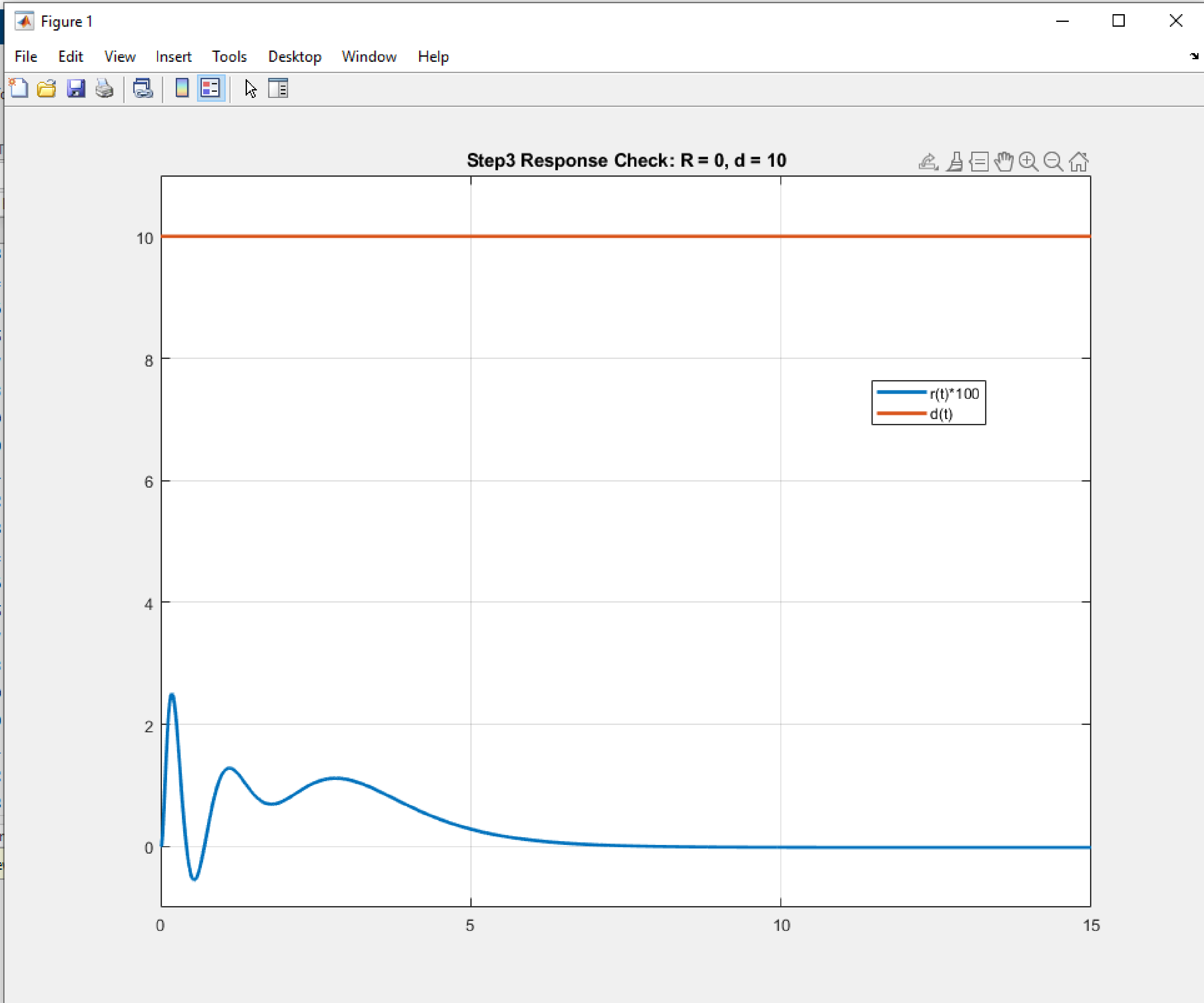
Figure 1
File Edit View Insert Tools Desktop Window Help
Step3 Response Check: R = 0, d = 10
r(t)*100
d(t)
10
8
6
4
2
0
0
5
10
15

-7.0000

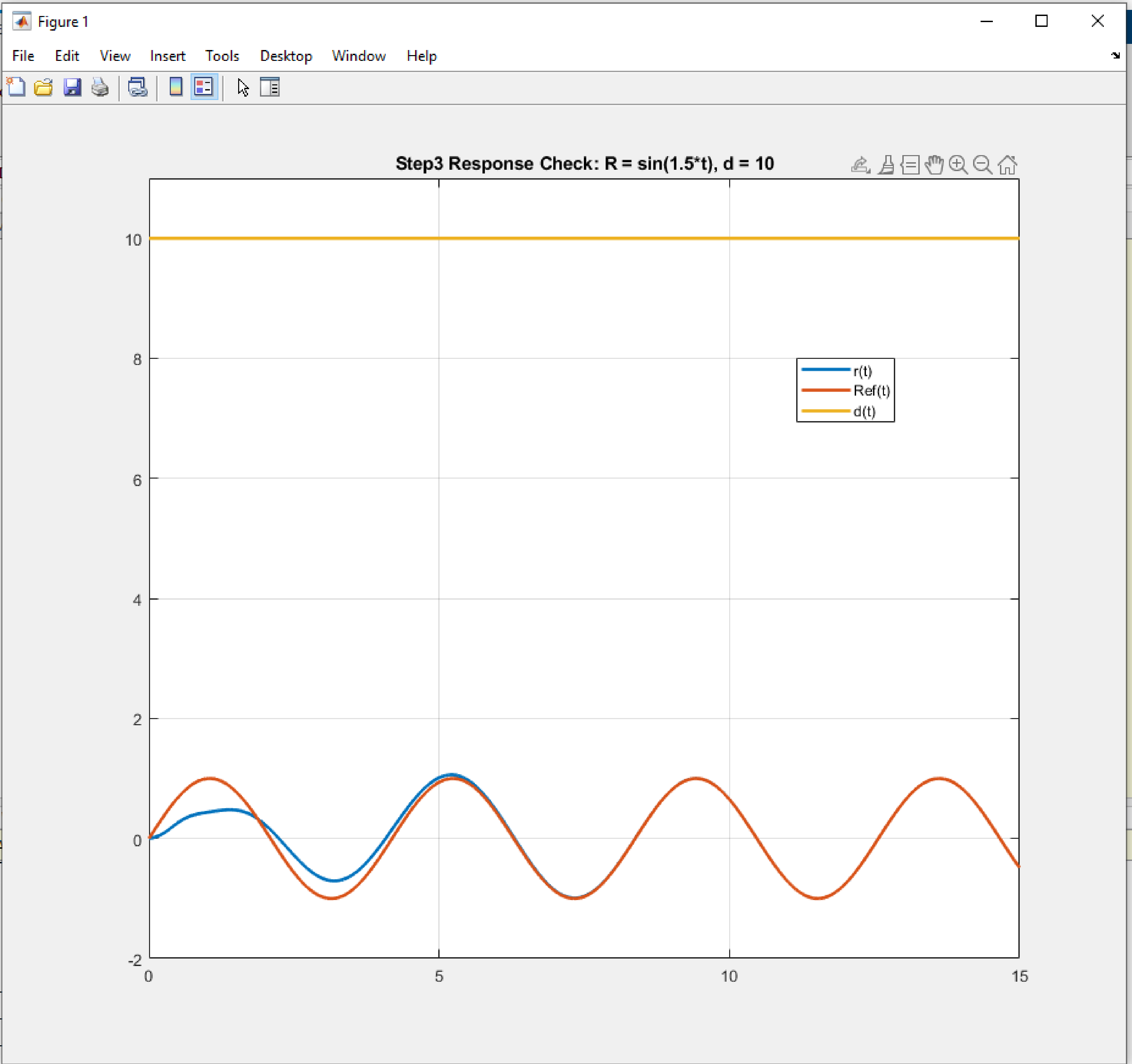
Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Step3 Response Check: R = sin(1.5*t), d = 10
r(t)
Ref(t)
d(t)
10
8
6
4
2
0
-2
0
5
10
15

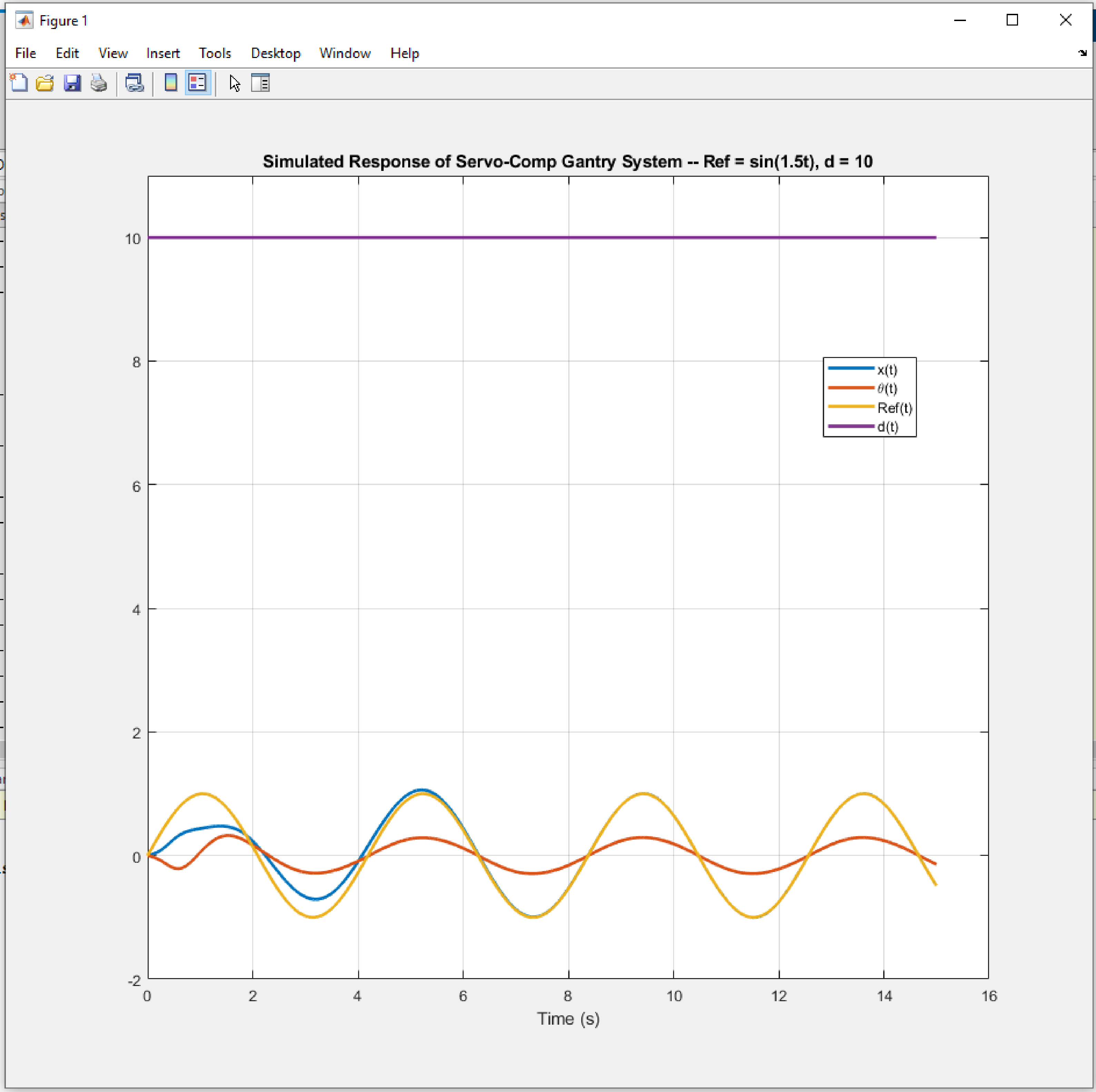
Figure 1
File Edit View Insert Tools Desktop Window Help
Simulated Response of Servo-Comp Gantry System -- Ref = sin(1.5t), d = 10
x(t)
θ(t)
Ref(t)
d(t)
-2
0
2
4
6
8
10
0
2
4
6
8
10
12
14
16
Time (s)

Observer Design Assume you can measure both position $x(t)$ and angle $\theta(t)$.

4) Design a full-order observer to estimate the states and the disturbance d. Feed back the state estimates rather than the actual states. Give the matrices for the resulting plant, servo compensator, observer, and FSF.

Ok clearly we have an input disturbance. We add the disturbance as a state resulting in the plant + disturbance system:

$$s\begin{bmatrix} X \\ X_i \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}\begin{bmatrix} X \\ X_i \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix}u$$

$$Y = \begin{bmatrix} C & 0 \end{bmatrix}\begin{bmatrix} X \\ X_i \end{bmatrix}$$

Just using position

Putting this into MATLAB, we can get our full-order observer:

Since I made the system earlier (w dom pole at -1, I'll make the obs. poles [-3, -5, -5, -5, -5]

H_x

$$H = [23, -9.3961, 161, -147.0408, 191.3265]$$

H_d

Now we design the servo-comp separate from the state estimate ⇒ we just use the first design: $K_x = [624.6481, \ldots, 52.7256]$, $K_z = [414.6627, \ldots, 228.5714]$

This leaves us with:

$$\begin{cases} sX = AX + Bu \\ sZ = A_z Z + B_z(CX - R) = (B_z C)X + (A_z)Z - B_z R \\ s\hat{X} = A\hat{X} + Bu + HC(X - \hat{X}) = (HC)X + (A - HC)\hat{X} + Bu \end{cases}$$

Open-loop:

$$s\begin{bmatrix} X \\ \hat{X} \\ Z \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ HC & A_s - HC_s & 0 \\ B_z C & 0 & A_z \end{bmatrix}\begin{bmatrix} X \\ \hat{X} \\ Z \end{bmatrix} + \begin{bmatrix} B \\ B \\ 0 \end{bmatrix}u + \begin{bmatrix} 0 \\ 0 \\ -B_z \end{bmatrix}R$$

Screenshot next page

$$K_{\hat{X}} = [K_x, 0]$$

Closed-Loop:
(separate paper) for working

$$s\begin{bmatrix} X \\ \hat{X} \\ Z \end{bmatrix} = \begin{bmatrix} A & -BK_{\hat{X}} & -BK_z \\ HC & A_s - HC_s - B_s K_{\hat{X}} & -BK_z \\ B_z C & 0 & A_z \end{bmatrix}\begin{bmatrix} X \\ \hat{X} \\ Z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -B_z \end{bmatrix}R$$

$K_{\hat{x}} = [K_x \quad 0]$

$u = -K_{\hat{x}}\hat{x} - K_z z$

$sx = Ax + B(-K_{\hat{x}}\hat{x} - K_z z) = Ax + (-BK_{\hat{x}})\hat{x} + (-BK_z)z$

$sz = (B_z C)x + (0)\hat{x} + (A_z)z + (-B_z)R$

$s\hat{x} = \cancel{(HC)x + (A-HC)\hat{x} + B(-K_{\hat{x}}\hat{x} - K_z z)}$

$s\hat{x} = (HC)x + (A - HC - BK_{\hat{x}})\hat{x} + (-BK_z)z$

```
Aol =

         0         0    1.0000         0         0         0         0         0         0         0         0         0
         0         0         0    1.0000         0         0         0         0         0         0         0         0
         0   39.2000         0         0         0         0         0         0         0         0         0         0
         0  -49.0000         0         0         0         0         0         0         0         0         0         0
   23.0000         0         0         0  -23.0000         0    1.0000         0         0         0         0         0
   -9.3961         0         0         0    9.3961         0         0    1.0000         0         0         0         0
  161.0000         0         0         0 -161.0000   39.2000         0         0    1.0000         0         0         0
 -147.0408         0         0         0  147.0408  -49.0000         0         0   -1.0000         0         0         0
  191.3265         0         0         0 -191.3265         0         0         0         0         0         0         0
    1.0000         0         0         0         0         0         0         0         0         0    1.5000         0
    1.0000         0         0         0         0         0         0         0         0   -1.5000         0         0
    1.0000         0         0         0         0         0         0         0         0         0         0         0

>> Bol

Bol =

     0
     0
     1
    -1
     0
     0
     1
    -1
     0
     0
     0
```

```
>> Col

Col =

     1     0     0     0     0     0     0     0     0     0     0     0

>> Dol

Dol =

     0

>>
```

```
Acl =

         0         0    1.0000         0         0         0         0         0         0         0         0         0
         0         0         0    1.0000         0         0         0         0         0         0         0         0
         0   39.2000         0         0 -624.6481 -353.8981  -80.7256  -52.7256         0 -414.6627 -462.6552 -228.5714
         0  -49.0000         0         0  624.6481  353.8981   80.7256   52.7256         0  414.6627  462.6552  228.5714
   23.0000         0         0         0  -23.0000         0    1.0000         0         0         0         0         0
   -9.3961         0         0         0    9.3961         0         0    1.0000         0         0         0         0
  161.0000         0         0         0 -785.6481 -314.6981  -80.7256  -52.7256    1.0000 -414.6627 -462.6552 -228.5714
 -147.0408         0         0         0  771.6889  304.8981   80.7256   52.7256   -1.0000  414.6627  462.6552  228.5714
  191.3265         0         0         0 -191.3265         0         0         0         0         0         0         0
    1.0000         0         0         0         0         0         0         0         0         0    1.5000         0
    1.0000         0         0         0         0         0         0         0         0   -1.5000         0         0
    1.0000         0         0         0         0         0         0         0         0         0         0         0


Bcl =

     0
     0
     0
     0
     0
     0
     0
     0
     0
    -1
    -1
    -1
```

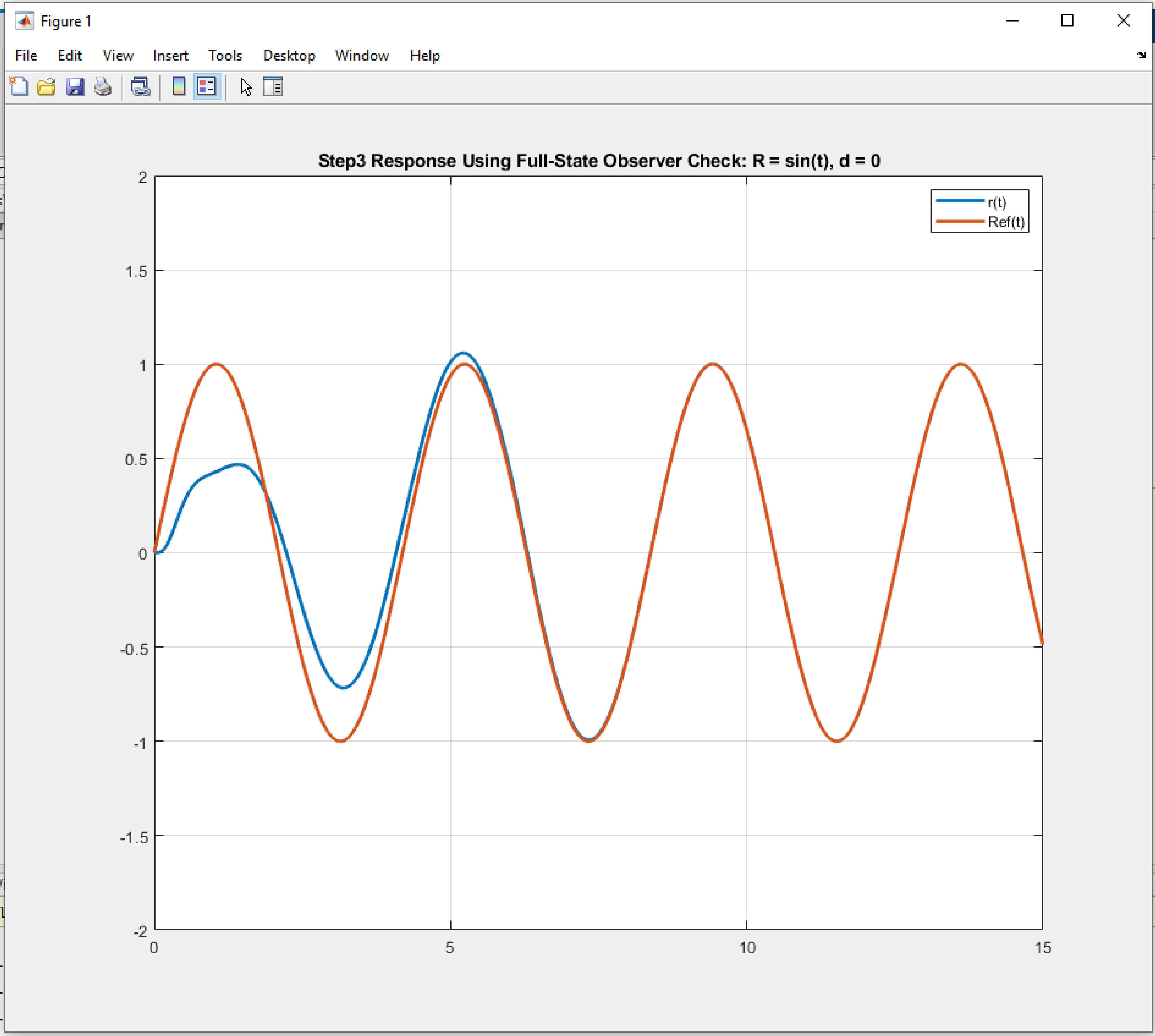
Figure 1
File Edit View Insert Tools Desktop Window Help
Step3 Response Using Full-State Observer Check: R = sin(t), d = 0
r(t)
Ref(t)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
5
10
15

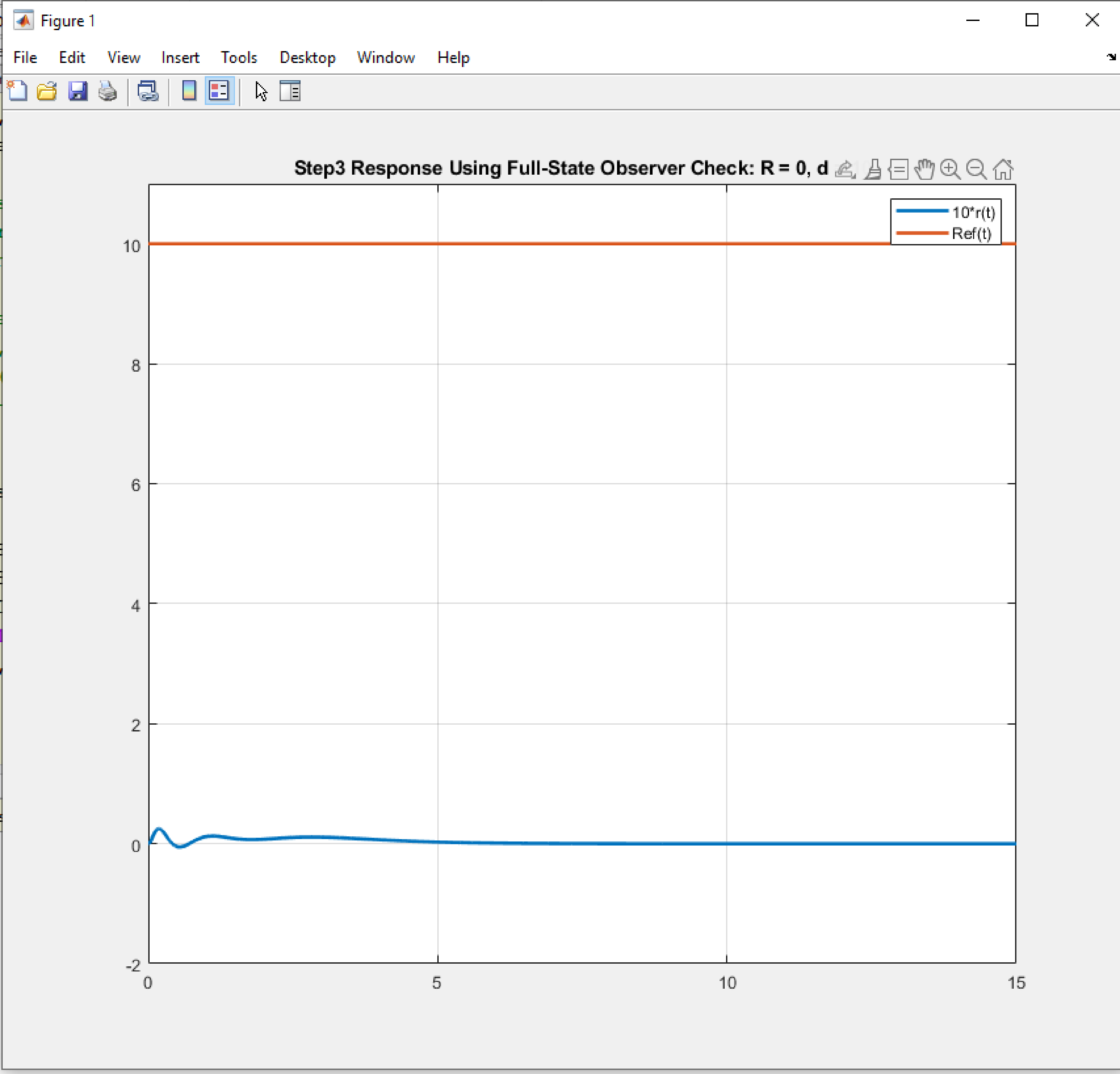
Figure 1
File Edit View Insert Tools Desktop Window Help
Step3 Response Using Full-State Observer Check: R = 0, d
10*r(t)
Ref(t)
10
8
6
4
2
0
-2
0
5
10
15

5) simulate linearized system /w

- $R = \sin(1.5t)$, $d = 0$
- $R = 0$, $d = 10$

(Attached)

6) Nonlinear sim /w $R = \sin(1.5t)$ and $d = 10$

- Use actual states
- Use estimated states

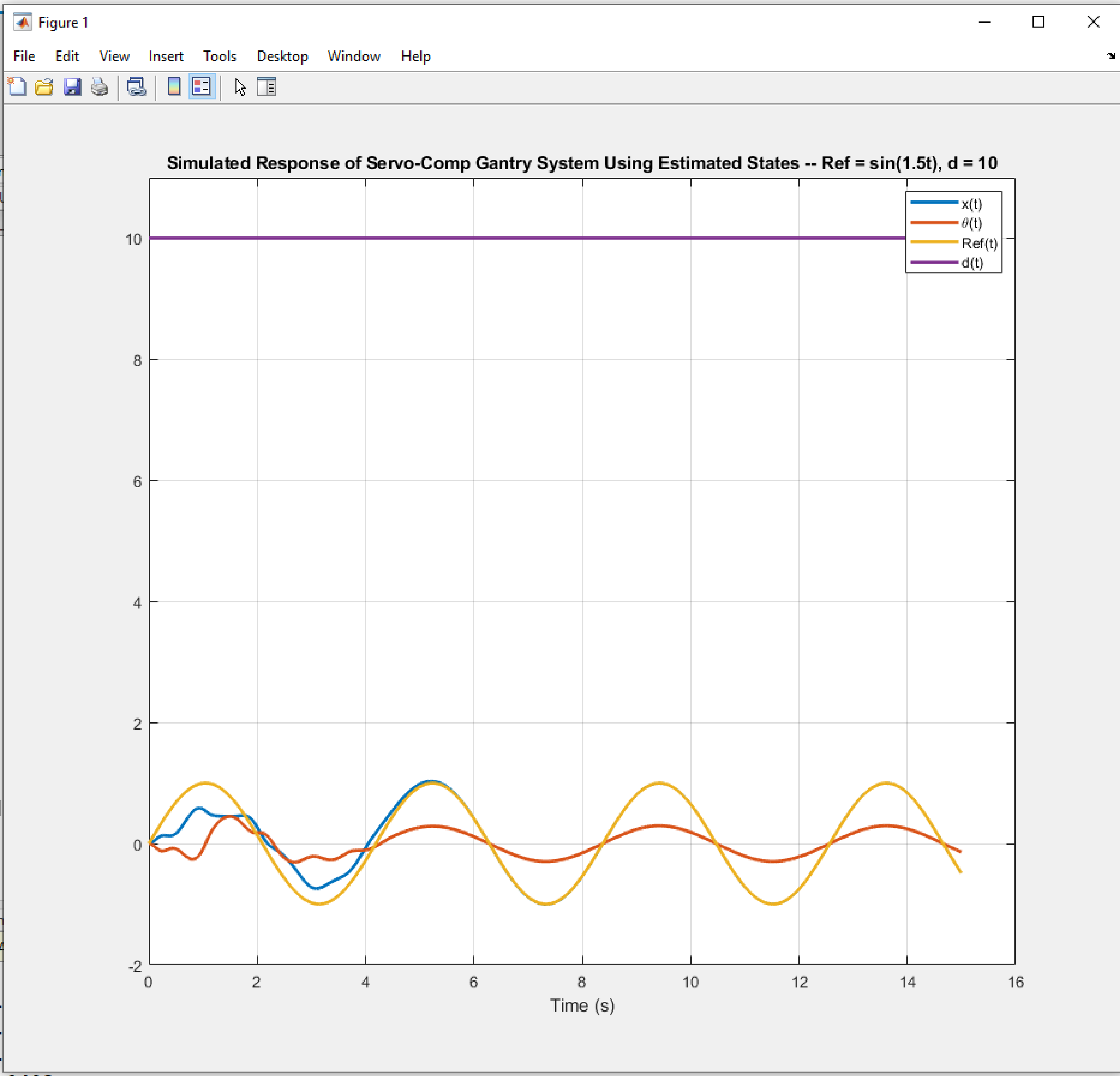
Figure 1
File Edit View Insert Tools Desktop Window Help
Simulated Response of Servo-Comp Gantry System Using Estimated States -- Ref = sin(1.5t), d = 10
x(t)
θ(t)
Ref(t)
d(t)
10
8
6
4
2
0
-2
0
2
4
6
8
10
12
14
16
Time (s)

```matlab
%% Problem 1 Design servo-comp
% Input System
A = [0 0 1 0; 0 0 0 1; 0 39.2 0 0; 0 -49 0 0];
B = [0;0;1;-1];
C = [1 0 0 0]; D = 0;

% Get Az, Bz
Az = [0 1.5 0; -1.5 0 0; 0 0 0]; Bz = ones(3,1);
Aaug = [A zeros(4,3); Bz*C Az]; Baug = [B; 0;0;0]; Caug = [C 0 0 0];
des_poles = [-1 -2 -3 -4 -5 -6 -7];
KK = placePoles(Aaug, Baug, Caug, des_poles);
Kx = KK(1:4)
Kz = KK(5:7)

% Check computed gains
Acl = [A-B*Kx, -B*Kz; Bz*C, Az]; Bcl = [0;0;0;0 ;-Bz];
Ccl = Caug; Dcl = D;
eig(Acl)

%% Problem 2 Linear Simulation
% % Plot using step3
% % R = sin(t), d = 0
% X = zeros(7,1); t = transpose(linspace(0,15,1001));
% R = sin(t); d = 0;
% BclR = Bcl;
% Y = step3(Acl,BclR,Ccl,Dcl,t,X,R);
% plot(t,Y,t,R,'LineWidth',2);
% legend('r(t)','Ref(t)');
% ylim([-2,2]); grid on; title('Step3 Response Check: R = sin(t), d = 0');
% pause

% % R = 0, d = 10
% clf;
% BclD = [B; 0*Bz];
% d = 0*t + 10;
% Y = step3(Acl,BclD,Ccl,Dcl,t,X,d);
% plot(t,Y*100,t,d,'LineWidth',2);
% legend('r(t)*100','d(t)');
% ylim([-1,11]); grid on; title('Step3 Response Check: R = 0, d = 10');
% % pause

% % R = sin(t), d = 10
% clf;
% BclR = Bcl; BclD = [B; 0*Bz];
% BclRD = [BclR, BclD];
% DclRD = [0, 0];
% Ref = sin(1.5*t);
% d = 0*t + 10;
% Y = step3(Acl,BclRD,Ccl,DclRD,t,X,[Ref,d]);
% plot(t,Y, t,Ref, t,d, 'LineWidth',2);
% legend('r(t)','Ref(t)','d(t)');
% ylim([-2,11]); grid on; title('Step3 Response Check: R = sin(1.5*t), d = 10');
% % pause

%% Problem 3 Nonlinear Simulation
% m1 = 1.0kg
% m2 = 4.0kg
% L = 1.0m

X = zeros(4,1);  % [x q dx dq]
Z = zeros(3,1);
dt = 100e-6; T_end = 15;
t = 0;
```

```
d = 10;
N = (T_end / dt) + 1;
DATA = zeros(N,4);

i = 1;
tic
while(t < T_end)

    Ref = sin(1.5*t);

    U = -Kz*Z - Kx*X;

    dX = GantryDynamics(X, U + d);
    dZ = Bz*(C*X - Ref) + Az*Z;

    X = X + dX * dt;
    Z = Z + dZ * dt;
    t = t + dt;

%     if(mod(i,100)==0)
%         GantryDisplay(X, Ref);
%     end

    DATA(i,:) = [X(1), X(2), Ref, d];

    i = i+1;

end
toc

% t = [1:length(DATA)]' * dt; %#ok<NBRAK>
% DATAds = downsample(DATA,10); tds = downsample(t,10);
% plot(t,DATA, 'LineWidth',2);
% ylim([-2,11]);
% grid on;
% legend('x(t)','\theta(t)','Ref(t)','d(t)');
% title('Simulated Response of Servo-Comp Gantry System -- Ref = sin(1.5t), d = 10');
xlabel('Time (s)');


%% Problem 4 Design full-order observer
% Since we have input disturbance,
A5 = [A, B; zeros(1,5)]; B5 = [B;0]; C5 = [C 0];
des_poles = [-3 -5 -5 -5 -5];
H = transpose(placePoles(A5',C5',C5,des_poles))

% Open loop system
Aol = [A, zeros(4,5), zeros(4,3); H*C, A5-H*C5, zeros(5,3); Bz*C, zeros(3,5), Az];
Bol = [B;B;zeros(3,1)];
Col = [C, zeros(1,5), zeros(1,3)];
Dol = 0;


%% Problem 5 Simulate
Kxe = [Kx, 0];
Acl = [A, -B*Kxe, -B*Kz; H*C, A5-H*C5-B5*Kxe, -B5*Kz; Bz*C, zeros(3,5), Az];
Bcl = [zeros(4,1); zeros(5,1); -Bz]; Ccl = Col; Dcl = Dol;

% % R = sin(1.5t), d = 0
% X = zeros(12,1); t = transpose(linspace(0,15,1001));
% R = sin(1.5*t); d = 0;
% BclR = Bcl;
% Y = step3(Acl,BclR,Ccl,Dcl,t,X,R);
```

```
% plot(t,Y,t,R,'LineWidth',2);
% legend('r(t)','Ref(t)');
% ylim([-2,2]); grid on; title('Step3 Response Using Full-State Observer Check: R =
sin(t), d = 0');

% R = 0, d = 10
% X = zeros(12,1); t = transpose(linspace(0,15,1001));
% R = 0; d = 0*t + 10;
% BclD = [B;B5;zeros(3,1)];
% Y = step3(Acl,BclD,Ccl,Dcl,t,X,d);
% plot(t,10*Y,t,d,'LineWidth',2);
% legend('10*r(t)','Ref(t)');
% ylim([-2,11]); grid on; title('Step3 Response Using Full-State Observer Check: R =
0, d = 10');


%% Problem 6 Nonlinear sim
% % First using actual states
% X = zeros(4,1);  % [x q dx dq]
% Xe = zeros(5,1);
% Z = zeros(3,1);
% dt = 100e-6; T_end = 15;
% t = 0;
% d = 10;
% N = (T_end / dt) + 1;
% DATA = zeros(N,4);
%
% i = 1;
% tic
% while(t < T_end)
%
%     Ref = sin(1.5*t);
%
%     U = -Kz*Z - Kx*X;
%
%     dX = GantryDynamics(X, U + d);
%     dXe = A5*Xe + B5*U + H*(C*X - C5*Xe);
%     dZ = Bz*(C*X - Ref) + Az*Z;
%
%     X = X + dX * dt;
%     Z = Z + dZ * dt;
%     t = t + dt;
%
% %     if(mod(i,100)==0)
% %         GantryDisplay(X, Ref);
% %     end
%
%     DATA(i,:) = [X(1), X(2), Ref, d];
%
%     i = i+1;
%
% end
% toc
%
% t = [1:length(DATA)]' * dt; %#ok<NBRAK>
% DATAds = downsample(DATA,10); tds = downsample(t,10);
% plot(t,DATA, 'LineWidth',2);
% ylim([-2,11]);
% grid on;
% legend('x(t)','\theta(t)','Ref(t)','d(t)');
% title('Simulated Response of Servo-Comp Gantry System Using Actual States -- Ref =
sin(1.5t), d = 10'); xlabel('Time (s)');
```

```matlab
% Now using estimated states
X = zeros(4,1);  % [x q dx dq]
Xe = zeros(5,1);
Z = zeros(3,1);
dt = 100e-6; T_end = 15;
t = 0;
d = 10;
N = (T_end / dt) + 1;
DATA = zeros(N,4);

i = 1;
tic
while(t < T_end)

    Ref = sin(1.5*t);

%     if t<1.5
%         U = -Kz*Z - Kx*X;
%     else
%         U = -Kz*Z - Kxe*Xe;
%     end

    U = -Kz*Z - Kxe*Xe;

    dX = GantryDynamics(X, U + d);
    dXe = A5*Xe + B5*U + H*(C*X - C5*Xe);
    dZ = Bz*(C*X - Ref) + Az*Z;

    X = X + dX * dt;
    Xe = Xe + dXe * dt;
    Z = Z + dZ * dt;
    t = t + dt;

%     if(mod(i,100)==0)
%         GantryDisplay(X, Ref);
%     end

    DATA(i,:) = [X(1), X(2), Ref, d];

    i = i+1;

end
toc

t = [1:length(DATA)]' * dt; %#ok<NBRAK>
DATAds = downsample(DATA,10); tds = downsample(t,10);
plot(t,DATA, 'LineWidth',2);
ylim([-2,11]);
grid on;
legend('x(t)','\theta(t)','Ref(t)','d(t)');
title('Simulated Response of Servo-Comp Gantry System Using Estimated States -- Ref =
sin(1.5t), d = 10'); xlabel('Time (s)');
```