Control Systems - Lab 1 & Lab 2

Ahmed Bahgat Elsherif 18010078

Moaz Fathy El-Defrawy 18011823

Mohamed Metwalli Noureldin 18011587

Abdallah Yasser Youssef 18015026

Moaz Nabil Khafagy 18011824

Youssef Hesham Akl 18012153

April 17, 2021

Summarized Learning Outcomes

Lab Experiment 1: Using MATLAB for Control Systems:

- Part I: Learned about vectors and matrices in MATLAB, as well as operations on them such as element wise operations, and plotting.

- Part II: Learned how to represent polynomials in MATLAB, find their roots, create polynomials from roots, and partial fractions.

- Part III: Learned how to write M-file scripts and functions, and flow control structures.

Lab Experiment 2: Mathematical Modeling of Physical Systems

- Converting an ODE into a system of ODEs.

- Solving a system of ODEs numerically.

1 Lab Experiment 1

1.1 Part 1

1.1.1 Exercise 1

```
clc;
clear;
A = magic(6);

fprintf('1) A = \n')
disp(A)

fprintf('fourth row = \n')
disp(A(4, :));

fprintf('---------------------------\n');
x = (0:0.1:1.1);
fprintf("x = \n");
disp(x);
y = (10:21);
fprintf("y = \n");
disp(y);
fprintf("2) x*y = \n")
disp(x.*y)
fprintf("y/x = \n")
disp(y./x)

fprintf('---------------------------\n');
r = randi([-8, 9], [4, 5]);
fprintf('3)')
disp(r)
```

1. The code first generates a $6 * 6$ matrix using the magic command

$$A = \begin{bmatrix} 35 & 1 & 6 & 26 & 19 & 24 \\ 3 & 32 & 7 & 21 & 23 & 25 \\ 31 & 9 & 2 & 22 & 27 & 20 \\ 8 & 28 & 33 & 17 & 10 & 15 \\ 30 & 5 & 34 & 12 & 14 & 16 \\ 4 & 36 & 29 & 13 & 18 & 11 \end{bmatrix}$$

2. The A(4, :) extracts the fourth row of the matrix

$$\begin{bmatrix} 8 & 28 & 33 & 17 & 10 & 15 \end{bmatrix}$$

3. The vectors x and y are initialized

$$x = \begin{bmatrix} 0 & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 1.0 & 1.1 \end{bmatrix}$$

$$y = \begin{bmatrix} 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \end{bmatrix}$$

4. We then perform element-wise multiplication and division

$$x * y = \begin{bmatrix} 0 & 1.1000 & 2.4000 & 3.9000 & 5.6000 & 7.5000 & 9.6000 & 11.9000 & 14.4000 & 17.1000 & 20.0000 & 23.1000 \end{bmatrix}$$
$$x/y = \begin{bmatrix} Inf & 110.00 & 60.00 & 43.3333 & 35.00 & 30.00 & 26.6667 & 24.2857 & 22.50 & 21.1111 & 20.00 & 19.0909 \end{bmatrix}$$

5. We use the function randi to generate the random matrix

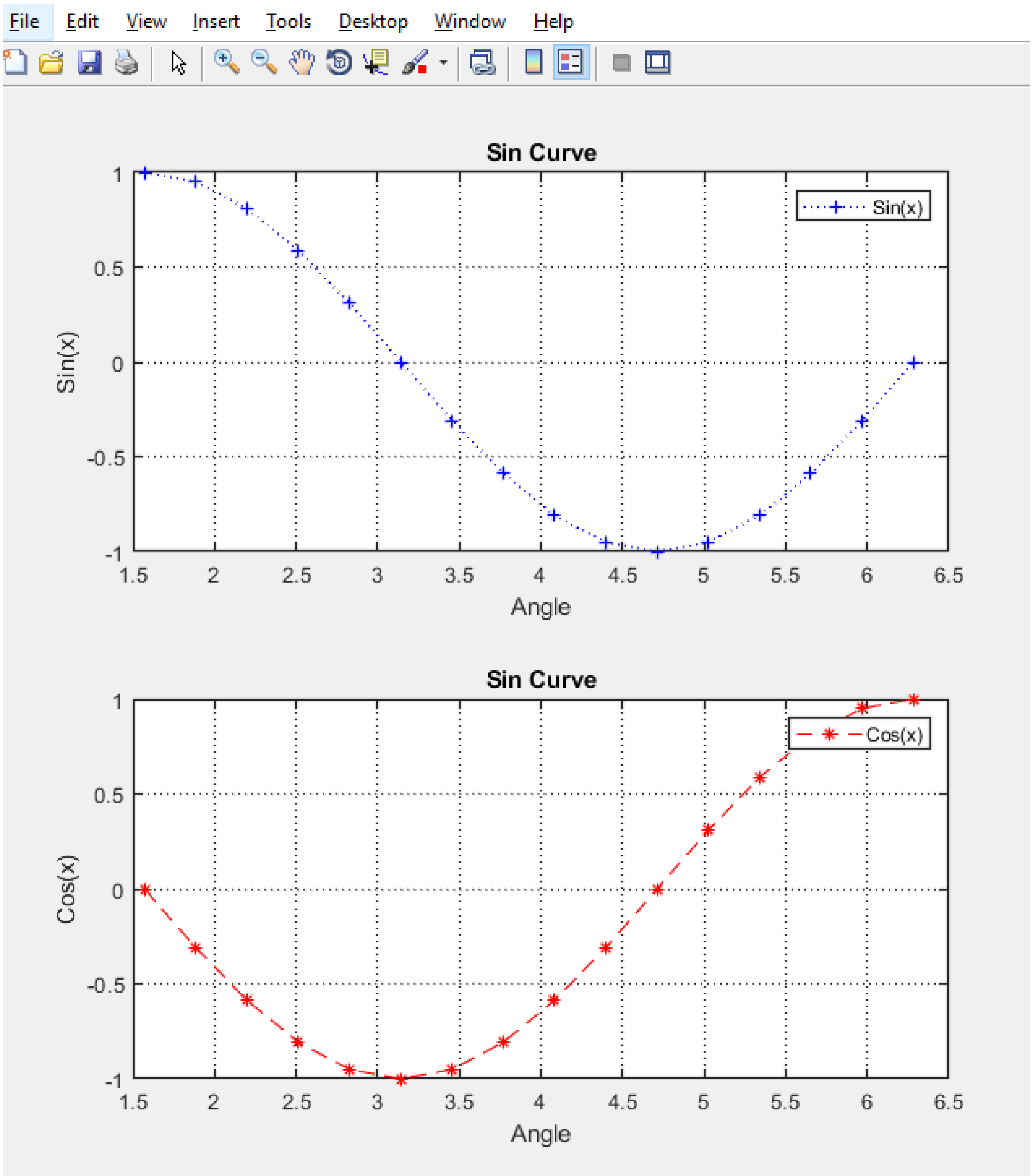
$$\begin{bmatrix} 3 & 4 & 3 & -4 & 4 \\ -8 & 5 & -5 & -8 & -3 \\ 7 & 5 & 4 & -7 & 9 \\ 8 & -1 & -8 & 6 & -8 \end{bmatrix}$$

And here is a screenshot of program execution

```
1) A =
    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11

fourth row =
     8    28    33    17    10    15

-------------------------
x =
         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000    0.7000    0.8000    0.9000    1.0000    1.1000

y =
    10    11    12    13    14    15    16    17    18    19    20    21

2) x*y =
         0    1.1000    2.4000    3.9000    5.6000    7.5000    9.6000   11.9000   14.4000   17.1000   20.0000   23.1000

y/x =
       Inf  110.0000   60.0000   43.3333   35.0000   30.0000   26.6667   24.2857   22.5000   21.1111   20.0000   19.0909

-------------------------
3)     3     4     3    -4     4
    -8     5    -5    -8    -3
     7     5     4    -7     9
     8    -1    -8     6    -8
```

1.1.2 Exercise 2

```
clc;

line_width = 0.9;
grid_line_style = ':';
grid_alpha = 1; %opacity

x=pi/2:pi/10:2*pi;
y=sin(x);
z=cos(x);

subplot(2,1, 1);
plot(x,y, ':+b', 'DisplayName', 'Sin(x)', 'LineWidth', 0.9);
title('Sin Curve')
xlabel('Angle')
ylabel('Sin(x)')
grid on
set(gca,'GridLineStyle',grid_line_style)
set(gca,'GridAlpha',grid_alpha)
set(gca,'LineWidth',line_width)
legend

subplot(2,1, 2);
plot(x,z, '--*r', 'DisplayName', 'Cos(x)', 'LineWidth', 0.8);
title('Sin Curve')
xlabel('Angle')
ylabel('Cos(x)')
grid on
set(gca,'GridLineStyle',grid_line_style)
set(gca,'GridAlpha',grid_alpha)
set(gca,'LineWidth',line_width)

legend
```


File
Edit
View
Insert
Tools
Desktop
Window
Help
Sin Curve
Sin(x)
Angle
Sin Curve
Cos(x)
Angle

1.2 Part 2

1.2.1 Exercise 1

```
clc;
p = [1,2,1];
q = [1,1];

resultOfProduct = conv(p,q) %p(s)*q(s)

fprintf('--------------------------------------------------\n');

rootsOfP = roots(p) %roots of p

fprintf('--------------------------------------------------\n');

rootsOfQ = roots(q) %roots of q

fprintf('--------------------------------------------------\n');

pAtNegativeOne = polyval(p,-1) %p(-1)

fprintf('--------------------------------------------------\n');

qAtSix = polyval(q,6) %q(6)

fprintf('--------------------------------------------------\n');
```

```
resultOfProduct =

     1     3     3     1

--------------------------------------------------

rootsOfP =

    -1
    -1

--------------------------------------------------

rootsOfQ =

    -1

--------------------------------------------------

pAtNegativeOne =

     0

--------------------------------------------------

qAtSix =

     7
```

1. We first define the polynomials p and q by making an array of their coefficients
2. We do the operation $p * q$ which is the convolution of p and q, which results in [1,3,3,1], that is equivalent to the polynomial
$$x^3 + 3x^2 + 3x + 1$$
3. We use the function roots() to compute the roots, whic are -1 with multiplicity 2 for p, and -1 for q
4. We then use polyval() to evaluate polynomials at certain points

1.2.2 Exercise 2

```
clc;

B1 = [2,5,3,6]
A1 = [1,6,11,6]

[r1,p1,k1] = residue(B1,A1) %B1/A1

fprintf('--------------------------------------------------\n');

B2 = [1,2,3]
A2 = [1,1];
A2 = conv(A2,conv(A2,A2))
[r2,p2,k2] = residue(B2, A2) %B2/A2

fprintf('--------------------------------------------------\n');
```

```
B1 =

     2     5     3     6


A1 =

     1     6    11     6


r1 =

   -6.0000
   -4.0000
    3.0000


p1 =

   -3.0000
   -2.0000
   -1.0000


k1 =

     2
```

1. We use the residue() function which returns the partial fraction decomposition in terms of triplets of residues, poles, and remainder term
2. For the first question, the answer is

$$\frac{A(s)}{B(s)} = \frac{2s^3 + 5s^2 + 3s + 6}{s^3 + 6s^2 + 11s + 6} = \frac{-6}{s+3} + \frac{-4}{s+2} + \frac{3}{s+1} + 2$$

```
B2 =

     1     2     3


A2 =

     1     3     3     1


r2 =

    1.0000
    0.0000
    2.0000


p2 =

   -1.0000
   -1.0000
   -1.0000


k2 =

     []
```

For the second question, the decomposition is:

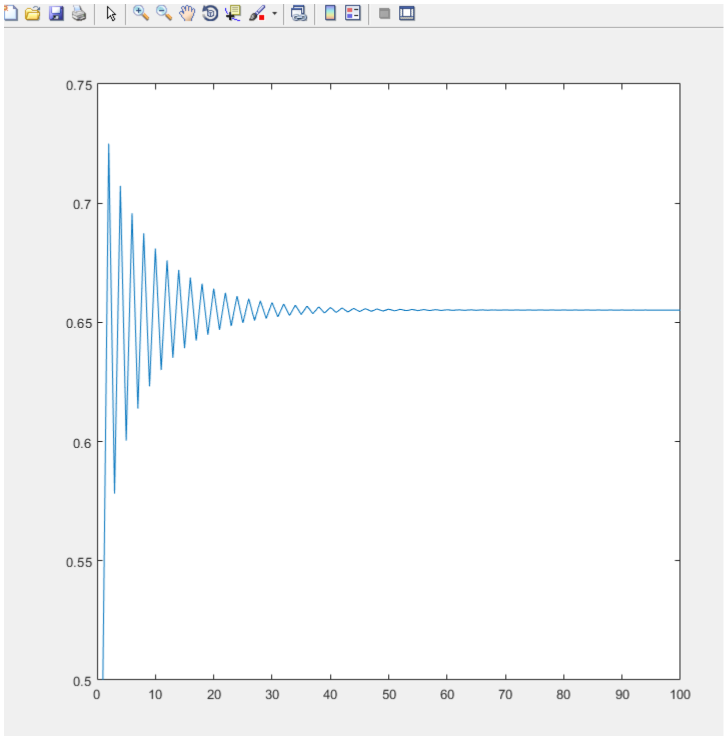
$$\frac{A(s)}{B(s)} = \frac{s^2 + 2s + 3}{(s+1)^3} = \frac{1}{s+1} + \frac{2}{(s+1)^3}$$

1.3 Part 3

1.3.1 Exercise 1

```
clc;
clear;

p = 2.9;
n = 100;

a = zeros(1, n);
a(1) = 0.5;
for i = 2:n
    a(i) = p * a(i-1) * (1 - a(i-1));
end

disp(a);
plot(a);
```

```
  Columns 1 through 15

    0.5000    0.7250    0.5782    0.7073    0.6004    0.6958    0.6139    0.6874    0.6232    0.6810    0.6300    0.6760    0.6352    0.6720    0.6392

  Columns 16 through 30

    0.6688    0.6424    0.6662    0.6449    0.6641    0.6469    0.6624    0.6485    0.6611    0.6498    0.6600    0.6508    0.6591    0.6516    0.6583

  Columns 31 through 45

    0.6523    0.6577    0.6529    0.6572    0.6533    0.6568    0.6537    0.6565    0.6539    0.6563    0.6542    0.6561    0.6544    0.6559    0.6545

  Columns 46 through 60

    0.6558    0.6546    0.6556    0.6547    0.6556    0.6548    0.6555    0.6549    0.6554    0.6549    0.6554    0.6550    0.6553    0.6550    0.6553

  Columns 61 through 75

    0.6551    0.6553    0.6551    0.6553    0.6551    0.6552    0.6551    0.6552    0.6551    0.6552    0.6551    0.6552    0.6551    0.6552    0.6551

  Columns 76 through 90

    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552

  Columns 91 through 100

    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552    0.6552
```



1. In the code, we set n to any number we want
2. We initialize a row vector of n zeroes
3. We do a simple for loop to calculate all the values of the sequence

1.3.2 Exercise 2

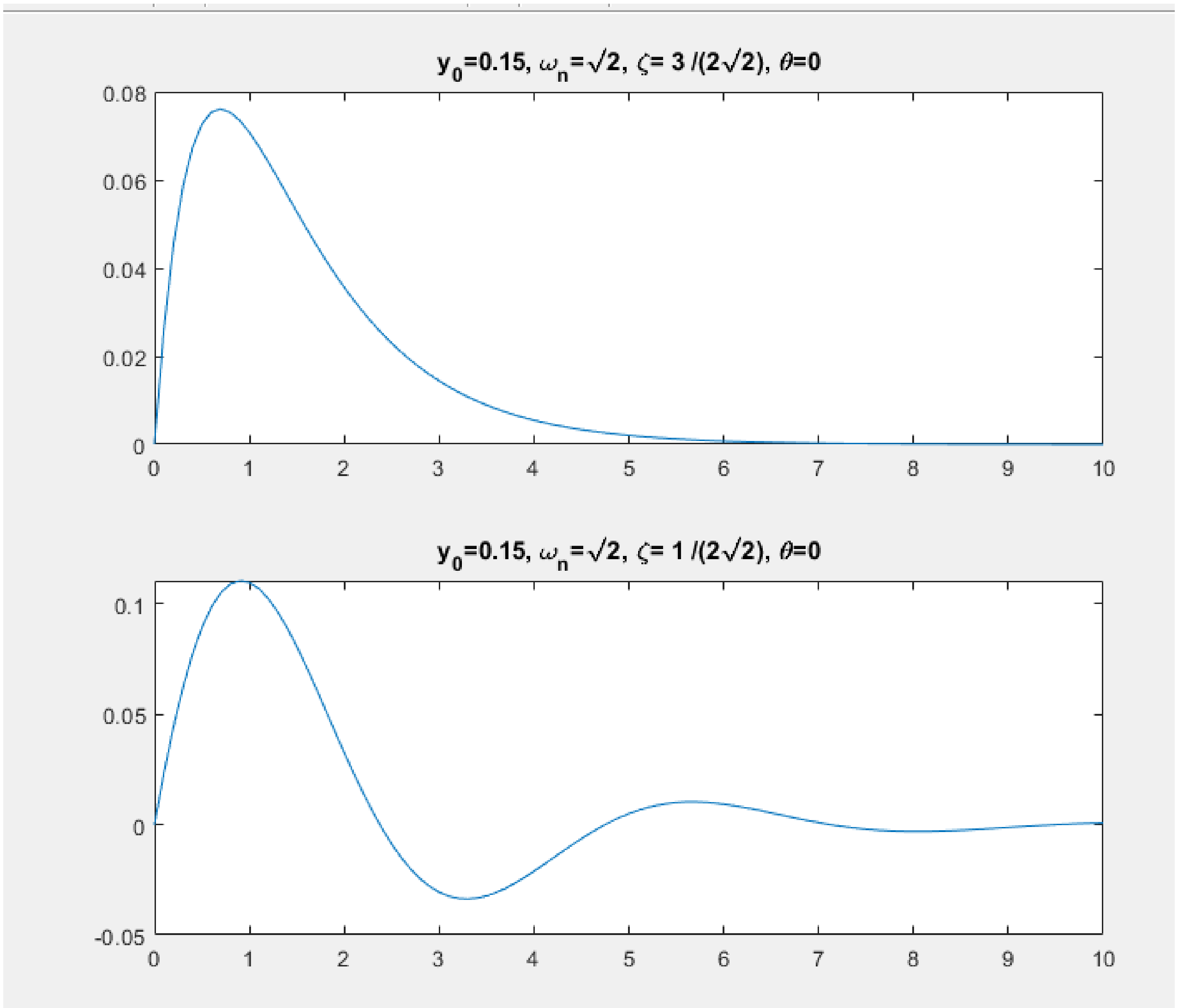
First, we define the y function in `y.m` file

```
function [y_t] = y(y0, zeta, omega, t, theta)
y_t = y0 ./ sqrt(1 - zeta) .* exp(-1 * zeta * omega .* t) .* ...
    sin(omega .* sqrt(1 - zeta^2) .* t + theta);
end
```

Then, we write the m file that will generate the 2 plots for the required cases

```
clc;
clear;
t = 0:0.1:10;
y0 = 0.15;
omega = sqrt(2);
theta = 0;

% Case 1:
zeta = 3 / (2 * sqrt(2));
y_1 = y(y0, zeta, omega, t, theta);

subplot(2,1,1);
plot(t,y_1);
title('y_0=0.15, \omega_n=\surd{2}, \zeta= 3 /(2\surd{2}), \theta=0');

% Case 2:
zeta = 1 / (2 * sqrt(2));
y_2 = y(y0, zeta, omega, t, theta);

subplot(2,1,2)
plot(t,y_2);
title('y_0=0.15, \omega_n=\surd{2}, \zeta= 1 /(2\surd{2}), \theta=0');
```

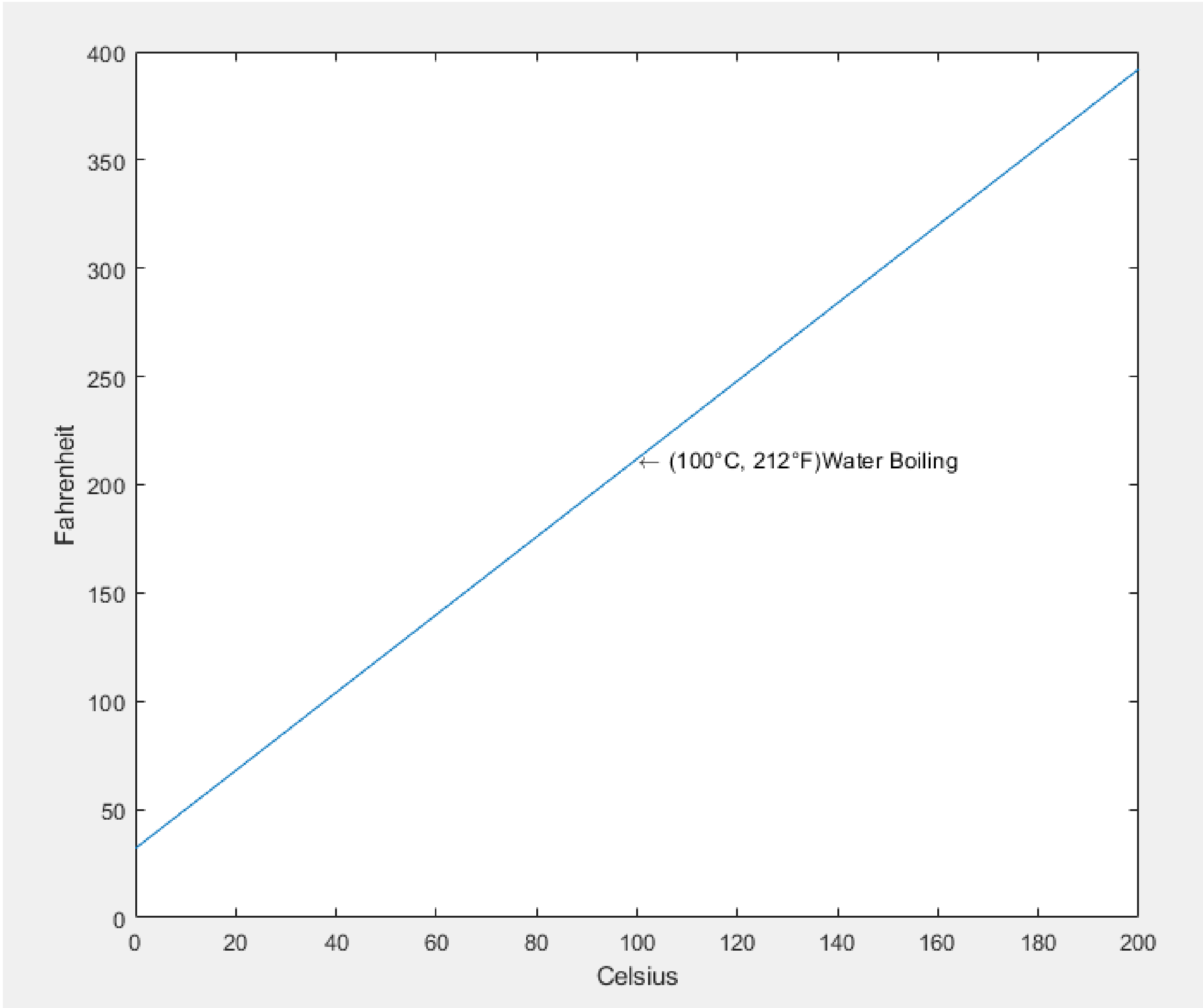
We get those 2 graphs


y0=0.15, ωn=√2, ζ= 3 /(2√2), θ=0
0.08
0.06
0.04
0.02
0
0 1 2 3 4 5 6 7 8 9 10
y0=0.15, ωn=√2, ζ= 1 /(2√2), θ=0
0.1
0.05
0
-0.05
0 1 2 3 4 5 6 7 8 9 10

1.3.3 Exercise 3

```
clc;
clear;


tf = 0:10:200;
tc = zeros(size(tf,2),1);
for i = 1:size(tf,2)
    tc(i) = c_to_f(tf(i));
end

disp(tc)
plot(tf,tc)
xlabel("Celsius")
ylabel("Fahrenheit")
text(100,212,'\leftarrow (100 C , 212 F )Water Boiling')


function [f] = c_to_f (c)
f = 9/5 * c + 32;
end
```

Then we get the graph for the function


400
350
300
250
200
150
100
50
0
Fahrenheit
← (100°C, 212°F)Water Boiling
0
20
40
60
80
100
120
140
160
180
200
Celsius

2 Lab Experiment 2

Proofs:

1. To prove the superposition and homogeneity of $c_1x_1 + c_2x_2$

$$L(x) = Ma + F_f(t) + F_s(t) = F_a(t)$$
$$L(x) = M\frac{d^2x}{dt^2} + B\frac{dx}{dt} + kx(t) = F_a(t)$$
$$L(cx) = Mcx'' + Bcx_1' + kcx = F_a(t)$$

where

$$F_a(t) = 0$$

then

$$L(cx) = c(Mx'' + Bx_1' + kx) = cL(x)$$

which proves the homogeneity and to prove the superposition property:

$$L(x_1) = Mx_1'' + Bx_1' + kx_1(t) = F_a(t)$$
$$L(x_2) = Mx_2'' + Bx_2' + kx_2(t) = F_a(t)$$

summing them:

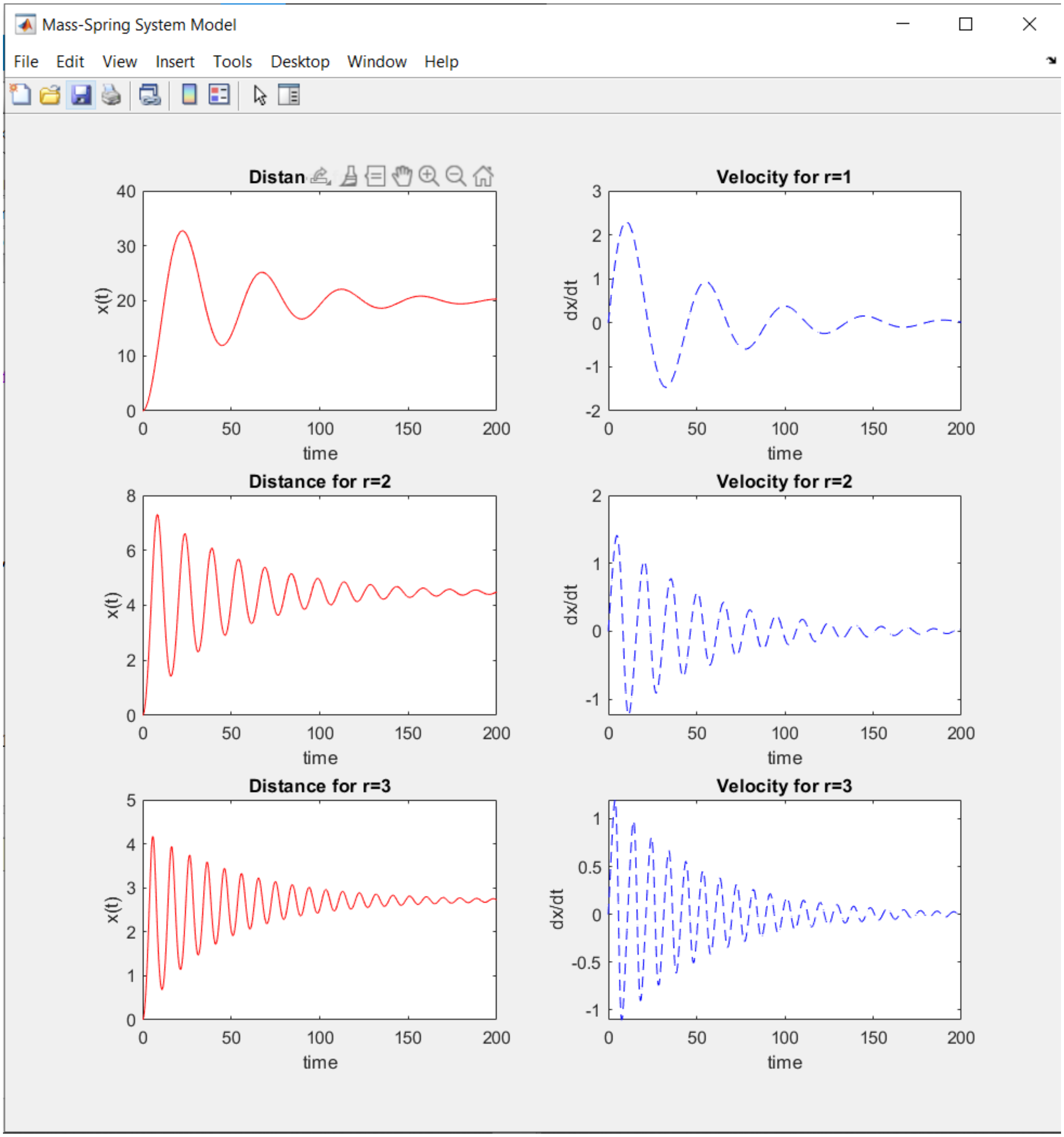
$$L(x_1 + x_2) = M(x_1'' + x_2'') + B(x_1' + x_2') + k(x_1 + x_2) = L(x_1) + L(x_2)$$

2. To disprove the linearity for $Mx'' + Bx' + kx^r(t) = F_a(t)$:

$$L(x) = Mx'' + Bx' + kx^r(t) = F_a(t)$$
$$L(cx) = Mcx'' + Bcx' + k(cx)^r(t)$$
$$L(cx) = cMx'' + Bx' + c^r k(x)^r(t) \neq cF_a(t) \neq cL(x)$$

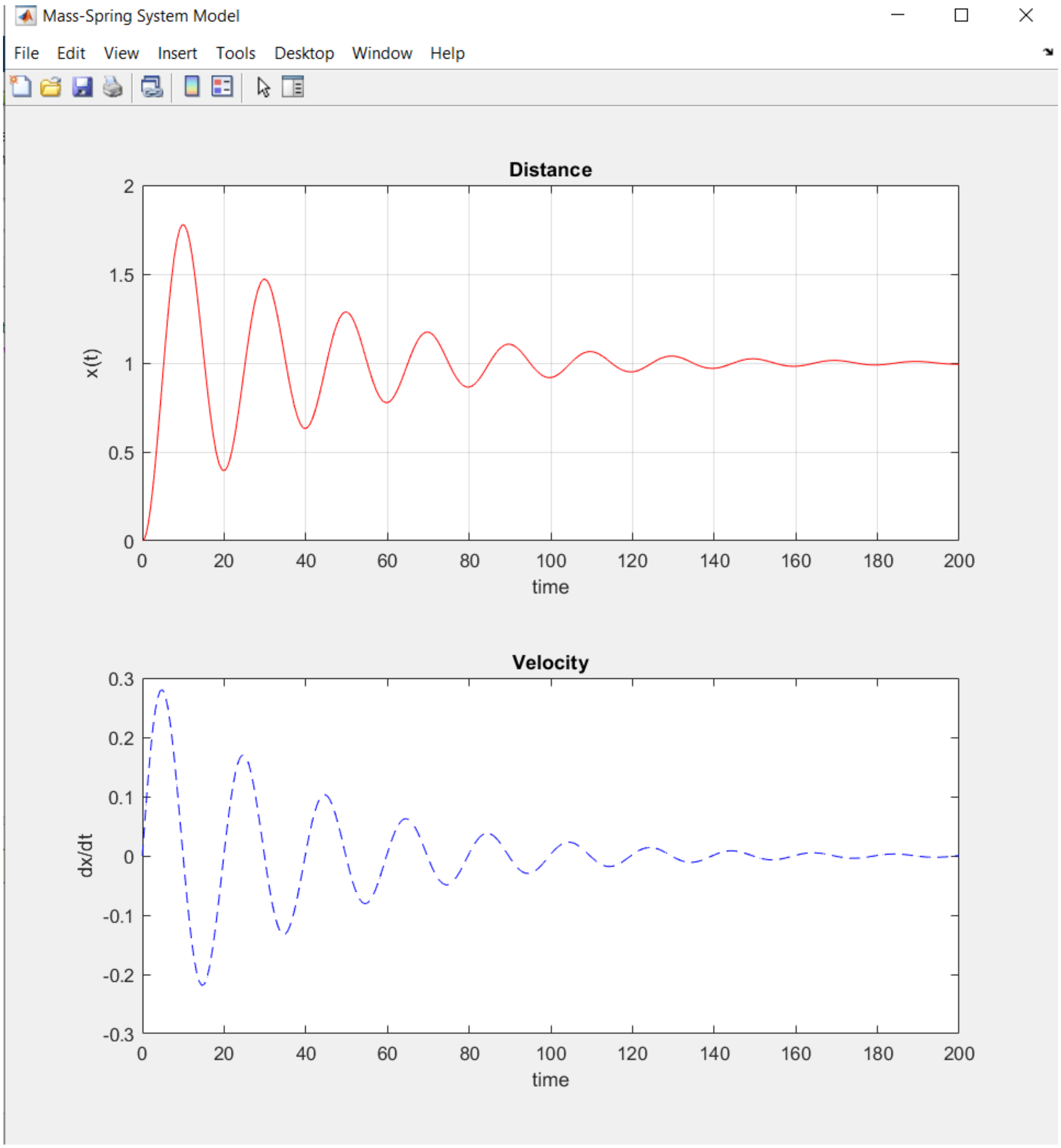
2.1 Exercise 1

```
figure("Name", "Mass-Spring System Model", 'NumberTitle', 'off');

for r = 1:3

    X0=[0;0];
    [t,v]=ode45(@(t,y) mass_spring(t,y,r), [0 200],X0);

    %plost distance
    subplot(3,2,r*2-1);
    plot(t, v(:,1), '-r');
    title(strcat("Distance for r=", num2str(r)));
    xlabel("time");
    ylabel("x(t)");

    %plot velocity
    subplot(3,2,r*2);
    plot(t, v(:,2), '--b');
    title(strcat("Velocity for r=", num2str(r)));
    xlabel("time");
    ylabel("dx/dt");

end


function dXdt=mass_spring(t, X,r)
    %flow rate
    M=750; %(Kg)
    B=30; %( Nsec/m)
    Fa=300; %N
    K=15; %(N/m)
    %r=1; % dX/dt
    dXdt(1,1)=X(2); dXdt(2,1)=-B/M*X(2)-K/M*X(1)^r+Fa/M;
end
```


Mass-Spring System Model
File Edit View Insert Tools Desktop Window Help
Distan
x(t)
time
Velocity for r=1
dx/dt
time
Distance for r=2
x(t)
time
Velocity for r=2
dx/dt
time
Distance for r=3
x(t)
time
Velocity for r=3
dx/dt
time

2.2 Exercise 2

```
%(initial speed and position)
%options = odeset('RelTol',[1e-4 1e-4],'AbsTol',[1e-5 1e-5],
'Stats','on');

figure("Name", "Mass-Spring System Model", 'NumberTitle', 'off');

X0=[0;0];
[t,v]=ode45(@(t,y) mass_spring(t,y), [0 200],X0);

%plost distance
subplot(2,1,1);
plot(t, v(:,1), '-r');
title("Distance");
xlabel("time");
ylabel("x(t)");
grid on;

%plot velocity
subplot(2,1,2);
plot(t, v(:,2), '--b');
title("Velocity");
xlabel("time");
ylabel("dx/dt");

disp("max ditance:\n");
disp(max(v(:,1)));

function dXdt=mass_spring(t, X)
    %flow rate
    M=10; %(Kg)
    B=0.5; %( Nsec/m)
    Fa=1; %N
    K=1; %(N/m)
    %r=1; % dX/dt
    dXdt(1,1)=X(2);
    dXdt(2,1)=-B/M*X(2)-K/M*X(1)+Fa/M;
end
```


Mass-Spring System Model
File Edit View Insert Tools Desktop Window Help
Distance
x(t)
time
Velocity
dx/dt
time

```
>> part_2
max amplitude:\n
     1.7773

>>
```