Project 4 report

Nguyen Vu

ECE-3723

Introduction

In this project, we will be introduced to custom function, it can be considered our own function. In order to make custom function, the good practice is that we should create a separate .m files for the different functions. In this project, we will learn about how to make a custom function and we will see how useful the custom function gives the benefit to us.

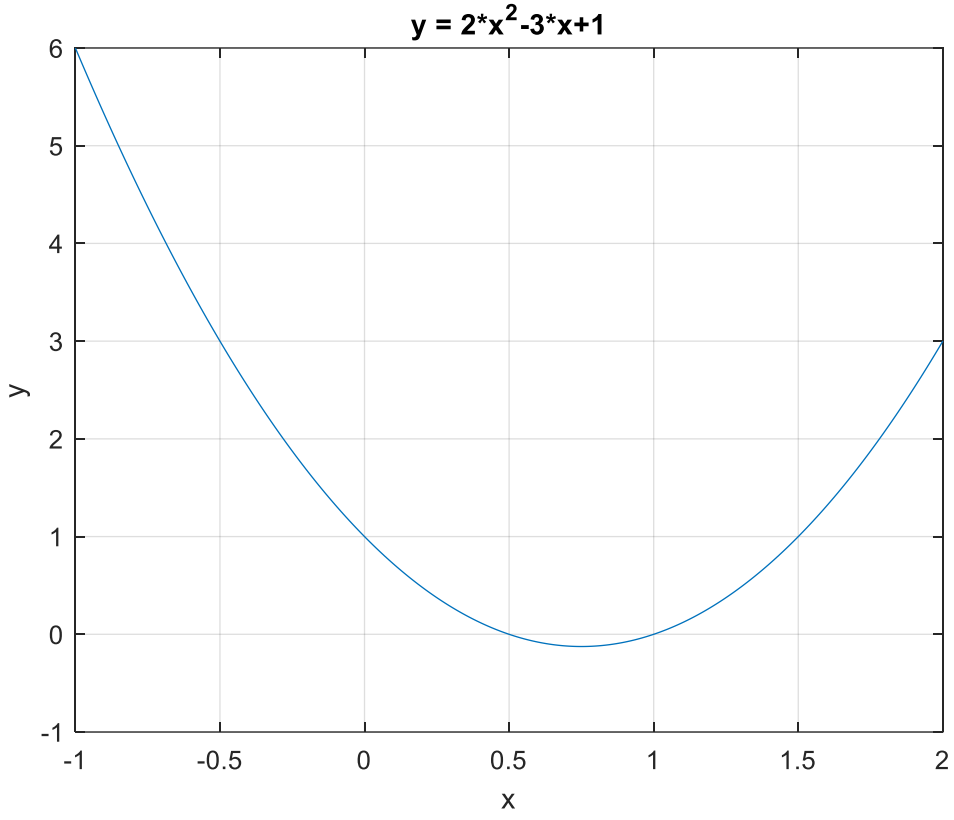
Procedure

1.Function Basics

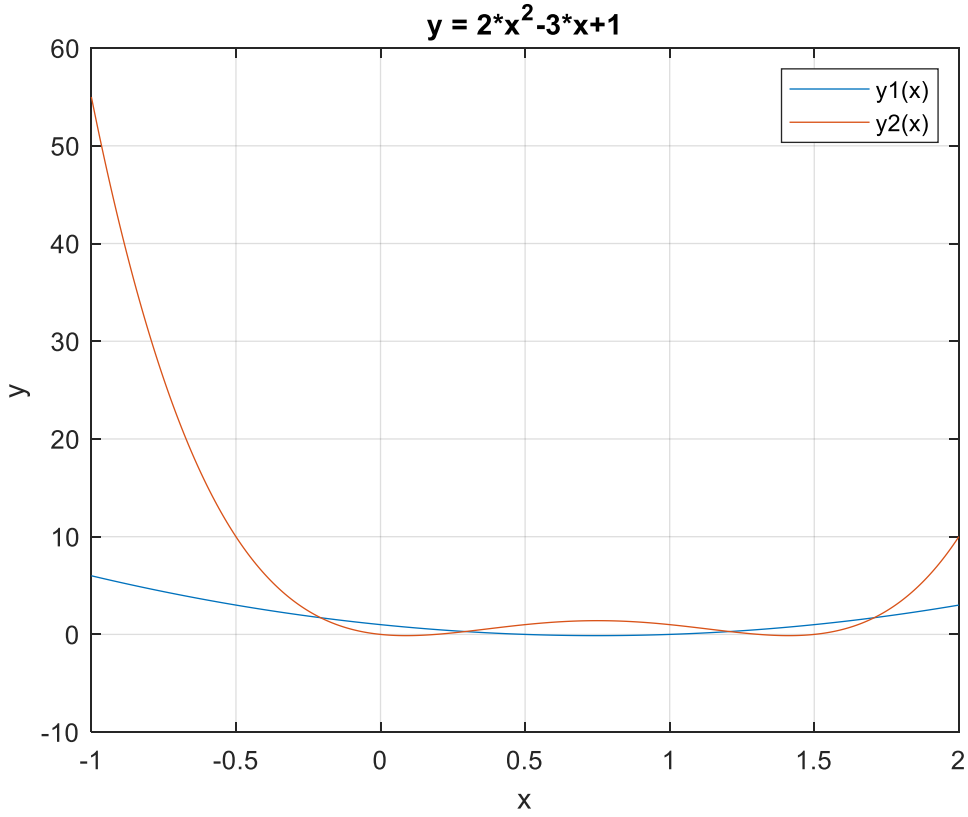
Create the function file above. Then pass the number-1 (pbola (-1)) into the function. What is the output? Next create an x-vector from -1 to 2 and pass into the function. Plot the response (10 points).

Part 1: When the input x = -1 then the output y = 6

Part 2: Plot the response

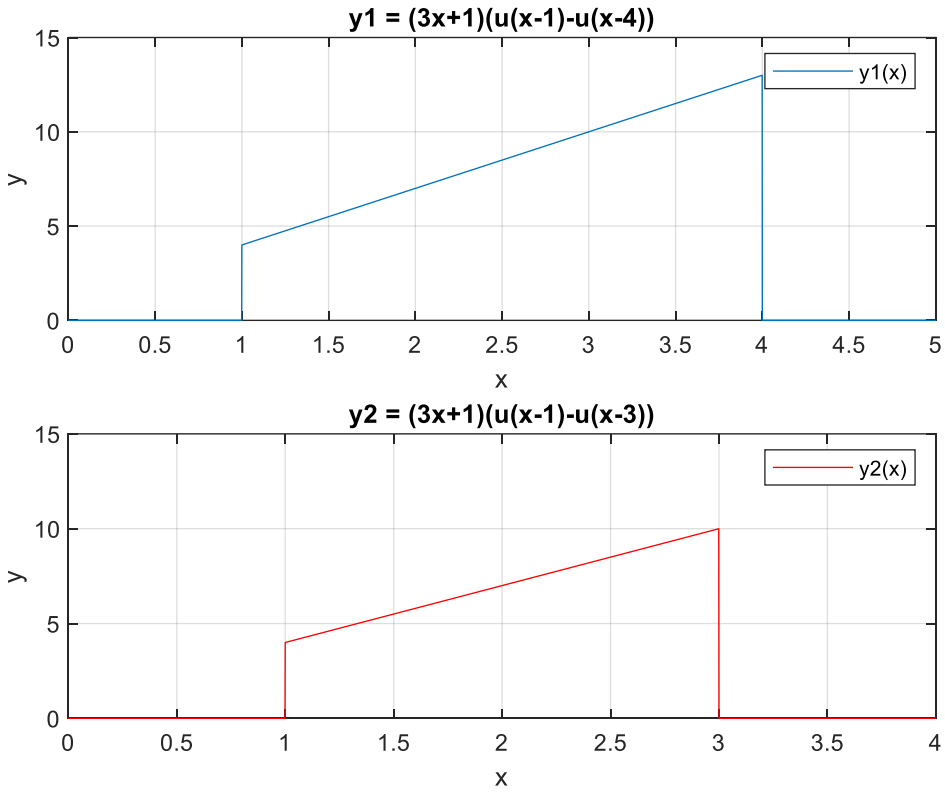


Pass the output of pbola(x) into itself. Plot pbola(x) and pbola(pbola(x)) on the same graph using hold (10 points).

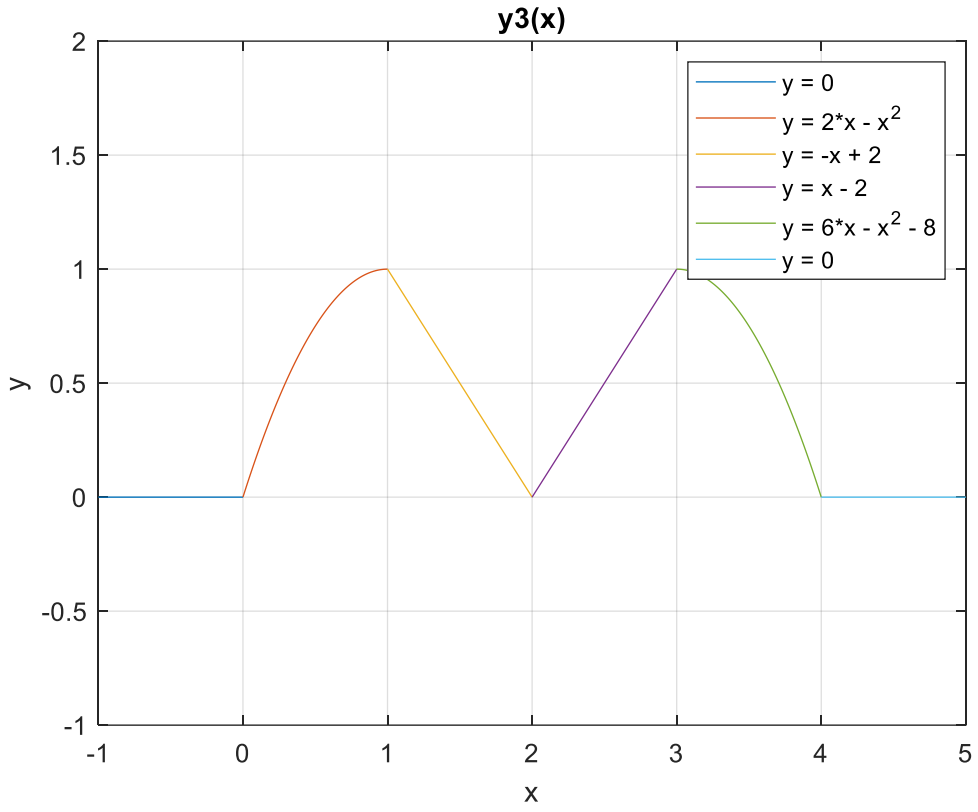


Create a function that can generate lines with the parameters mentioned. Then plot the following lines when x is from 0 to 4 (use different formatting for each line to make the easily distinguishable from each other).

Show the plot in the report (10 points).

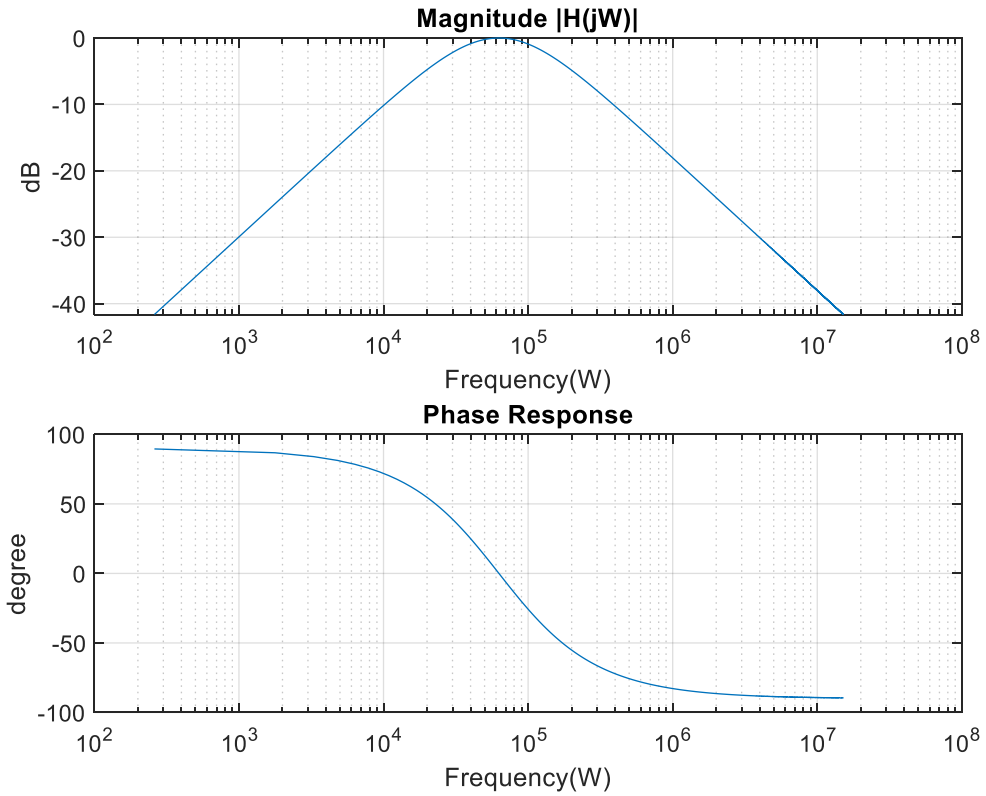


Plot the resulting vector (y3(x)). Scale the plot to be from $-1 \leq x \leq 5$ and $-1 \leq y \leq 2y3,max$ (20 points).



2.Filter Functions

Show the magnitude plot (in dB) and phase plot (in degrees) of the filter (20 points).



Conclusion

Through this project, besides the functions that were created by MATLAB, we also can create our own function and use it easily. We can just write one function and pass any input to get the output result. After this project, we are able to create the parabola function, the line function, the function which can give us more than one output from one or more inputs. Then we can apply the custom function to make filter function and get the various results from it.

Appendix

1.Function Basics

Create the function file above. Then pass the number-1 (pbola (-1)) into the function. What is the output? Next create an x-vector from -1 to 2 and pass into the function. Plot the response (10 points).

pbola.m file

```
function y = pbola(x)
y = 2*x.^2-3*x+1;
end
```

project4Circuit.m file

```
clear;
clc;
close all;

% calculate output y at input x = -1 by calling the function pbola
y = pbola(-1)
% Create the x-vector from -1 to 2
x = -1:0.001:2;
% Call the function pbola with input x-vector
y1 = pbola(x)
plot(x,y1)
grid on
xlabel('x')
ylabel('y')
title('y = 2*x^2-3*x+1')
```

Pass the output of pbola(x) into itself. Plot pbola(x) and pbola(pbola(x)) on the same graph using hold (10 points).

project4Circuit.m file

```
% Call the function inside the function pbola(pbola(x))
y2 = pbola(y1)
hold on
plot(x,y2)
legend('y1(x)','y2(x)')
```

Create a function that can generate lines with the parameters mentioned. Then plot the following lines when x is from 0 to 4 (use different formatting for each line to make the easily distinguishable from each other).

Show the plot in the report (10 points).

theLine.m file

```
function y = theLine(h,q,x)

eqn = h*x+q
```

```
y = eqn.*((x>= min(x) + 1)- (x>=max(x)-1))

end
```

project4Circuit.m file

```
% Create the x-vector from 0 to 5
x = 0:0.001:5;
% Call the function theLine
y1 = theLine(3,1,x);

subplot(211);
plot(x,y1)
xlabel('x');
ylabel('y');
title('y1 = (3x+1)(u(x-1)-u(x-4))')
grid on;
legend('y1(x)')


subplot(212);
% Create the x-vector from 0 to 4
x1 = 0:0.001:4;
% Call the function theLine
y2 = theLine(3,1,x1)

plot(x1,y2,'r')
grid on;
xlabel('x');
ylabel('y');
title('y2 = (3x+1)(u(x-1)-u(x-3))');
legend('y2(x)')
ylim([0 15])
hold off;
```

Plot the resulting vector (y3(x)). Scale the plot to be from $-1 \le x \le 5$ and $-1 \le y \le 2y3,max$ (20 points).

parabola.m file

```
function y = parabola(a2,a1,a0,x)
y = a2*x.^2 + a1*x + a0
end
```

project4Circuit.m file

```
% x is in range from -1 to 0
x0 = -1:0.001:0
% y = 0
outRange1 = parabola(0,0,0,x0)
plot(x0,outRange1)
hold on

% x is in range from 0 to 1
x1 = 0:0.001:1
```

```
% y0 = 2*x - x^2
y0 = parabola(-1,2,0,x1)
plot(x1,y0)
hold on



% x is in range from 1 to 2
x2 = 1:0.001:2
% y1 = y2(x) = -x + 2
y1 = parabola(0,-1,2,x2)
plot(x2,y1)
hold on

% x is in range from 2 to 3
x3 = 2:0.001:3
% y2 = y1(x) = x - 2
y2 = parabola(0,1,-2,x3)
plot(x3,y2)
hold on

% x is in range from 3 to 4
x4 = 3:0.001:4
% y3 = 6*x - x^2 - 8
y3 = parabola(-1,6,-8,x4)
plot(x4,y3)
hold on

% x is in range 4 to 5
x5 = 4:0.001:5
% y = 0
outRange2 = parabola(0,0,0,x5)
plot (x5,outRange2)
grid on
hold off

legend('y = 0', 'y = 2*x - x^2', 'y = -x + 2', 'y = x - 2', 'y = 6*x - x^2 - 8', 'y = 0')
xlabel('x')
ylabel('y')
title('y3(x)')

% Scale from -1 <= x <= 5 and -1 <= y <= 2
xlim([-1 5]);
ylim([-1 2]);
```

2.Filter Functions

Show the magnitude plot (in dB) and phase plot (in degrees) of the filter (20 points).

bandPass.m file

```
function [magH, Angle] = bandPass(f0, BW, C, pt)

% Calculate W0
W0 = 2*pi*f0;
% Calculate inductor
```

```
L = 1/((W0^2)*C);
% Calculate quality
Q = f0/BW;
% Calculate resistor
R = (1/Q) * sqrt(L/C);
% Calculate cutoff Frequecy
Wc1 = -1/(2*R*C) + sqrt((1/(2*R*C))^2 + 1/(L*C));
Wc2 = 1/(2*R*C) + sqrt((1/(2*R*C))^2 + 1/(L*C));

W = linspace(Wc1/100, Wc2*100, 10000);


% Transfer function
H = ((1/(R*C))*1i*W)./((1i*W).^2 + (1/(R*C))*(1i*W) + 1/(L*C));

% find the magnitude in dB
magH = 20*log10(abs(H));
subplot(211);
semilogx(W,magH)
xlabel('Frequency(W)')
ylabel('dB')
title('Magnitude |H(jW)|')
grid on

% find the angle in degree
Angle = angle(H) *(180/pi);
subplot(212);
semilogx(W,Angle)
xlabel('Frequency(W)')
ylabel('degree')
title('Phase Response')
grid on

end
```

project4Circuit.m file

```
% Given inputs
f0 = 10000;
BW = 5000;
C = 10^-6;
pt = 1;

% if the plot toggle value is 1 then make the plot
if pt == 1
[mag, phase] = bandPass(f0,BW,C,pt);
end
```