Nolan Anderson

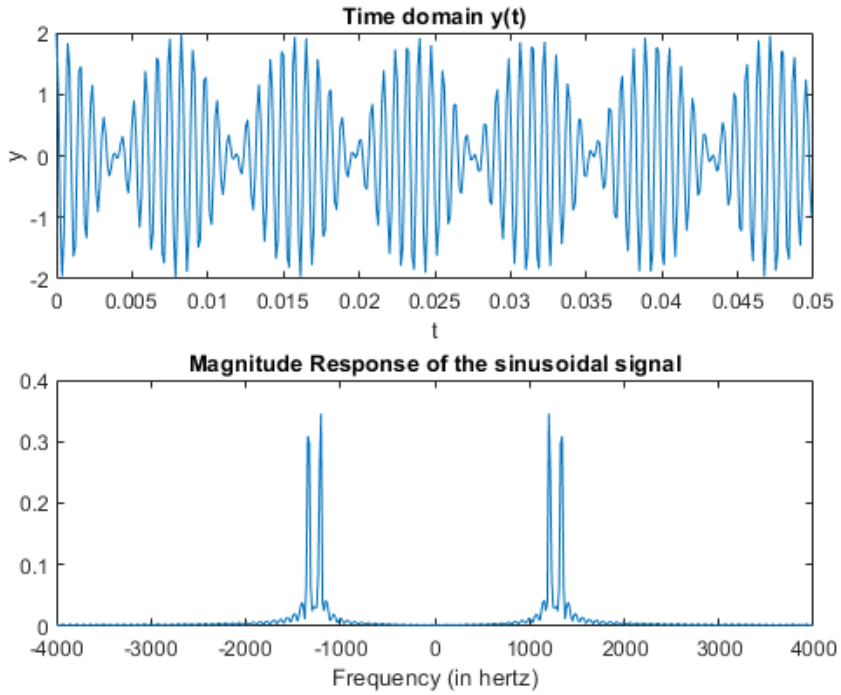
EE 384 Classwork 4 Due 17 September 2021

```
clear; clc; clf; close all;
```

1a Plot time and freqnency domain

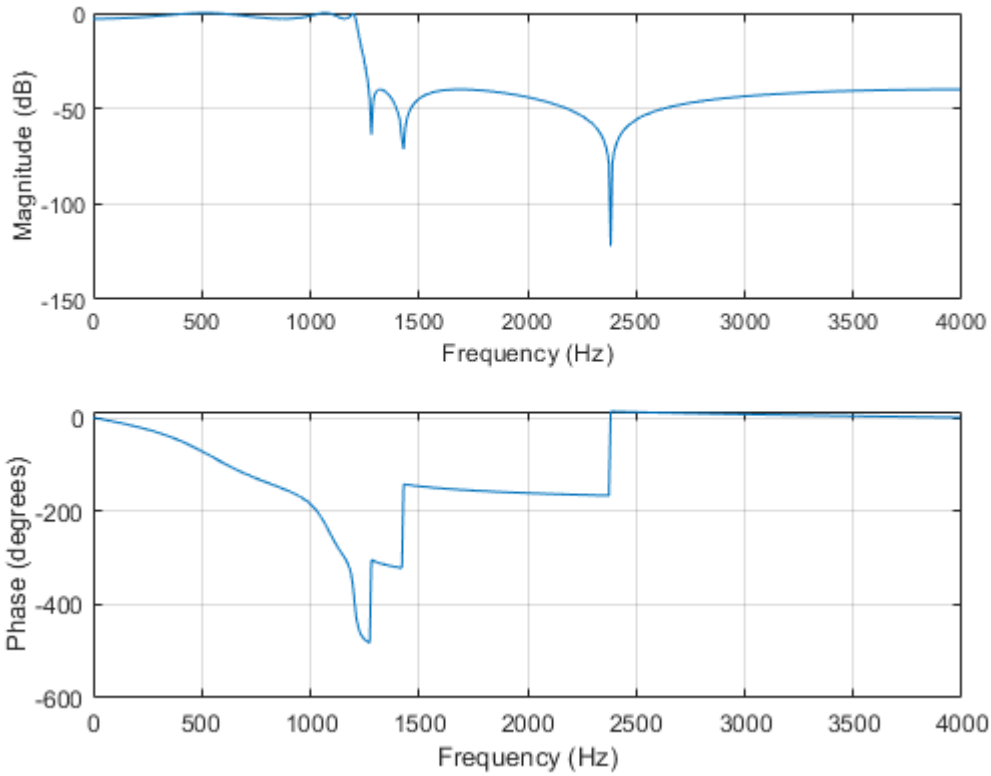
Initial signal

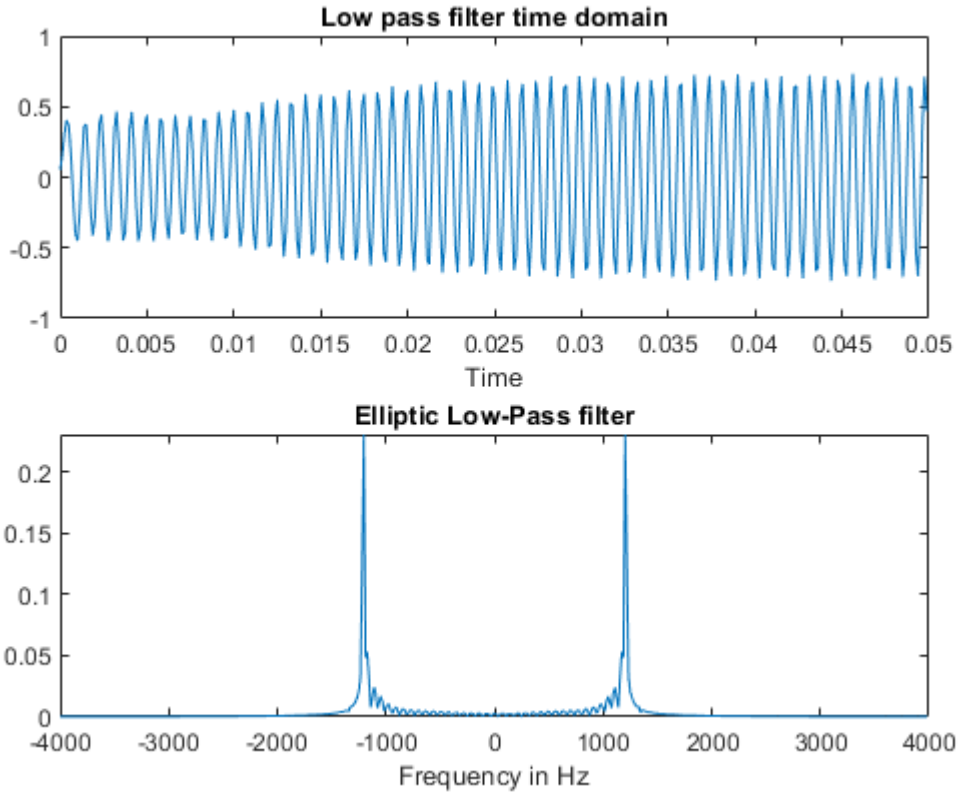
```
Fs = 8000; dt = 1/Fs; StopTime = 0.05; f1 = 1209; f2 = 1336;
t = 0 : dt : 0.05 - dt;
y = cos(2*pi*f1*t) + cos(2*pi*f2*t);

% Convert to freqency domain
L = length(y);
n = pow2(nextpow2(L));
y_dft = fft(y,n);
y_s = fftshift(y_dft);
f = (-n/2 : n/2 -1) * (Fs/n);

% Plot
figure
subplot(2, 1, 2)
plot(f,abs(y_s) / n), xlabel('Frequency (in hertz)');
title('Magnitude Response of the sinusoidal signal');
subplot(2, 1, 1)
plot(t,y), title('Time domain y(t)'), xlabel('t'), ylabel('y');
```



1b Low-Pass filter design

```
Wp = (2*f1) / Fs;
Ws = (2*f2) / Fs;
Rp = 3;
Rs = 40;
[Ord, Wn] = ellipord(Wp, Ws, Rp, Rs);
[b, a] = ellip(Ord, Rp, Rs, Wn);
yf = filter(b, a, y);

% Convert to Frequency Domain
n = 512;
freqz(b, a, n, Fs);
L = length(yf);
n = pow2(nextpow2(L));
y_dft = fft(yf, n);
y_s = fftshift(y_dft);
f = (-n/2 : n/2 - 1)*(Fs / n);

% Plot time and frequency domain.
figure
subplot(2, 1, 1)
plot(t, yf), title('Low pass filter time domain'), xlabel('Time');
subplot(2, 1, 2)
plot(f,abs(y_s) /n ), xlabel('Frequency in Hz'), title('Elliptic Low-Pass filter');
```



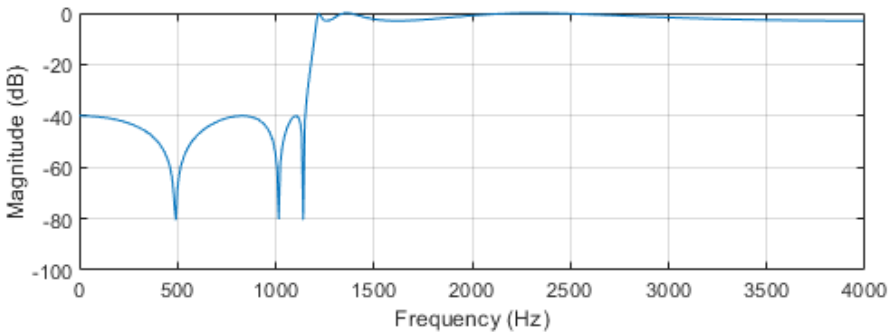


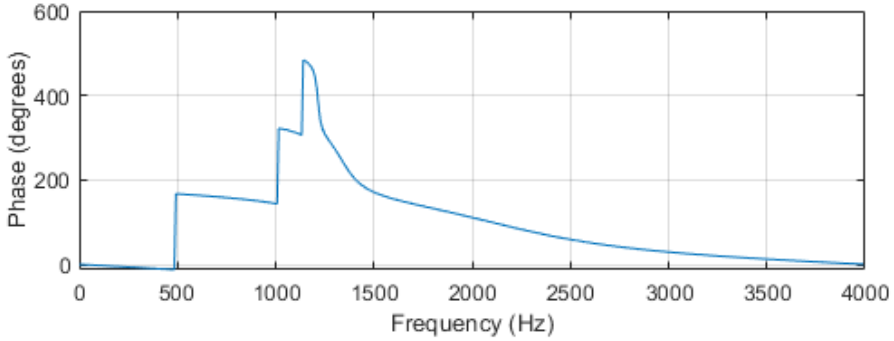
1c High-pass filter design

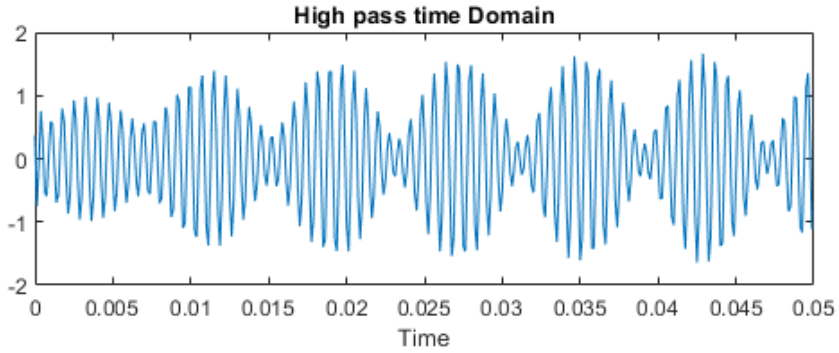
Initialize values

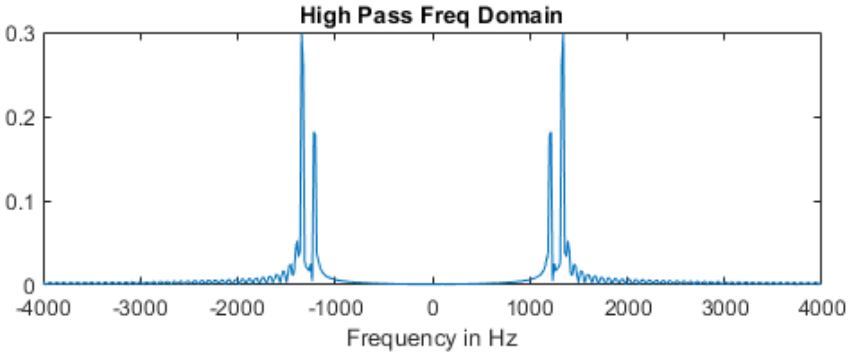
```
Wp = (2*f1) / Fs;
Ws = (2*f2) / Fs;
Rp = 3;
Rs = 40;

% High Pass Filter Design
[Ord, Wn] = ellipord(Wp, Ws, Rp, Rs);
[b, a] = ellip(Ord, Rp, Rs, Wn, 'high');
yf = filter(b, a, y);

% Convert to Frequency Domain
n = 512;
freqz(b, a, n, Fs);
L = length(yf);
n = pow2(nextpow2(L));
y_dft = fft(yf, n);
y_s = fftshift(y_dft);
f = (-n/2 : n/2 - 1)*(Fs / n);

% Plot time and frequency domain.
figure
subplot(2, 1, 1)
```

```
plot(t, yf), title('High pass time Domain'), xlabel('Time');

subplot(2, 1, 2)
plot(f,abs(y_s) /n ), xlabel('Frequency in Hz'), title('High Pass Freq Domain');
```



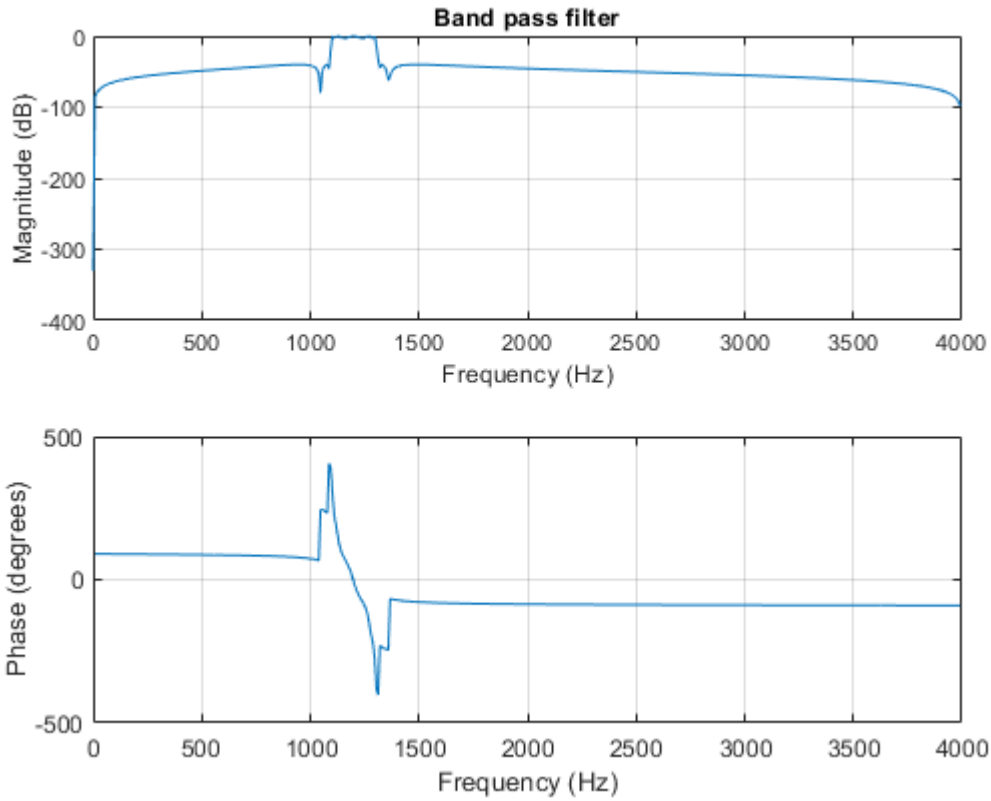


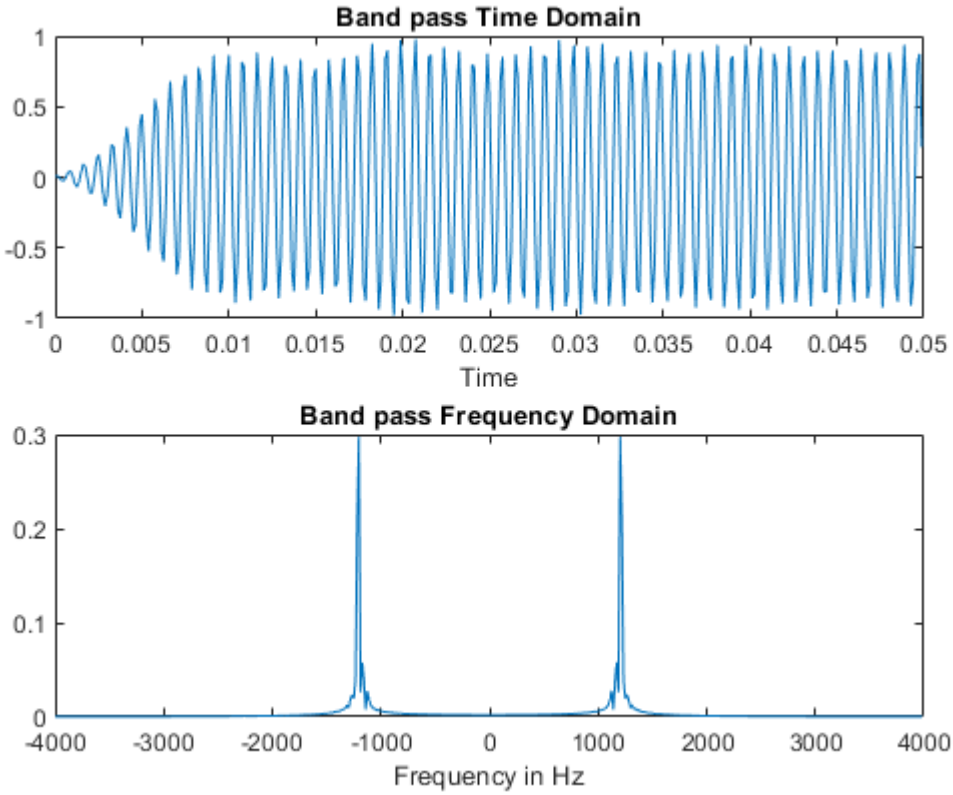




1d Band-pass filter design

```
y = cos(2*pi*f1*t) + cos(2*pi*f2*t);
Rp = 3; Rs = 40;
Wp = [2*1100 2*1300]/Fs; Ws = [2*1000 2*1336]/Fs;
[n,Wn] = ellipord(Wp,Ws,Rp,Rs);
[b,a] = ellip(n,Rp,Rs,Wn);
freqz(b,a, 512, 8000);
title('Band pass filter');
yf = filter(b, a, y);

n = 512;
L = length(yf);
n = pow2(nextpow2(L));
y_dft = fft(yf, n);
y_s = fftshift(y_dft);
f = (-n/2 : n/2 - 1)*(Fs / n);

figure
subplot(2, 1, 1)
plot(t, yf), title('Band pass Time Domain'), xlabel('Time');

subplot(2, 1, 2)
plot(f,abs(y_s) /n ), xlabel('Frequency in Hz'), title('Band pass Frequency Domain');
```

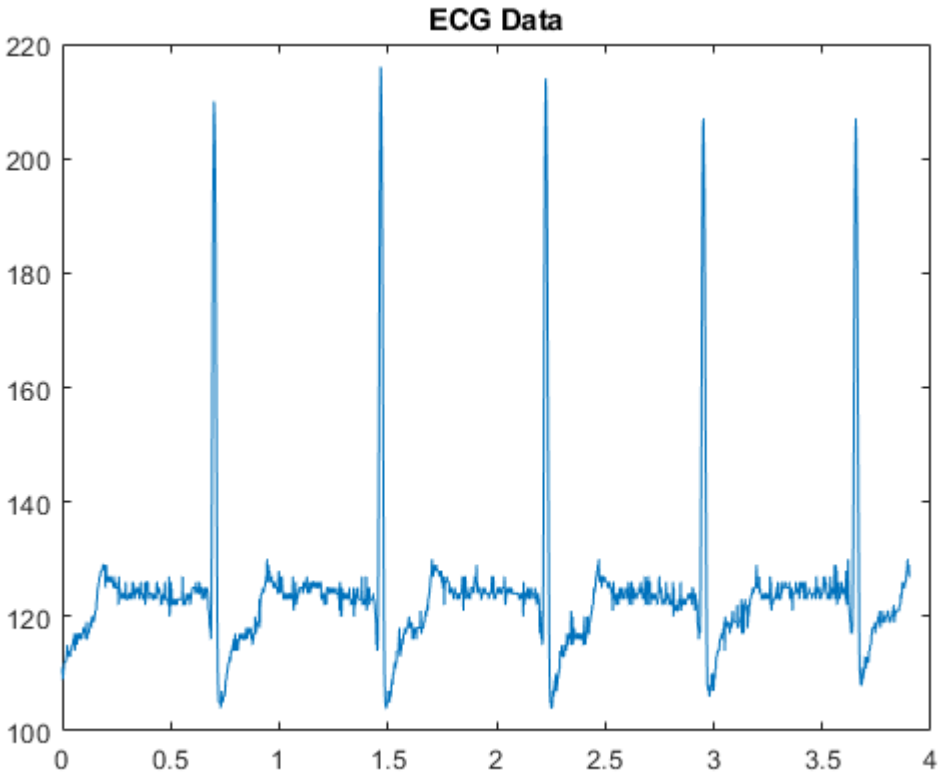


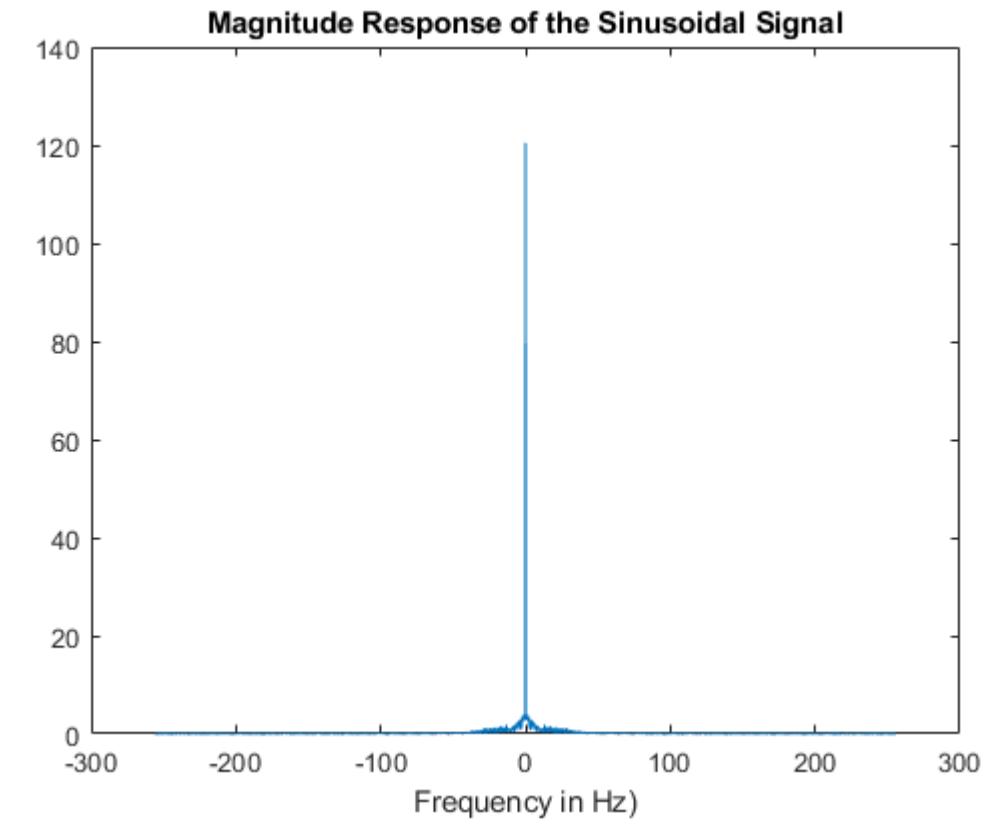


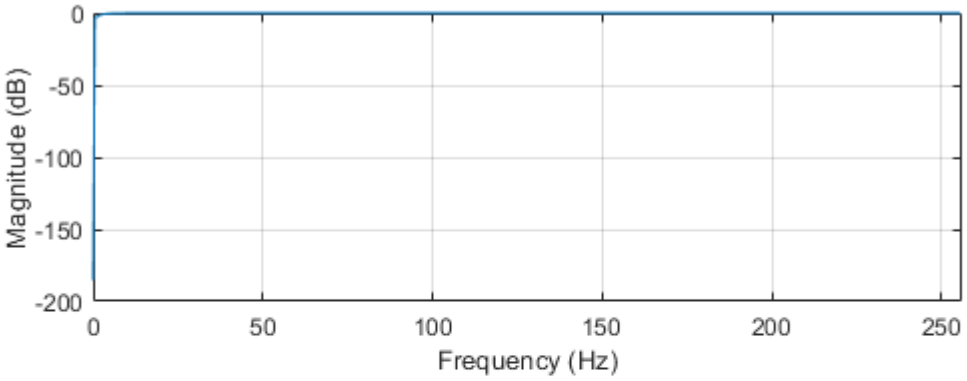
Problem 2

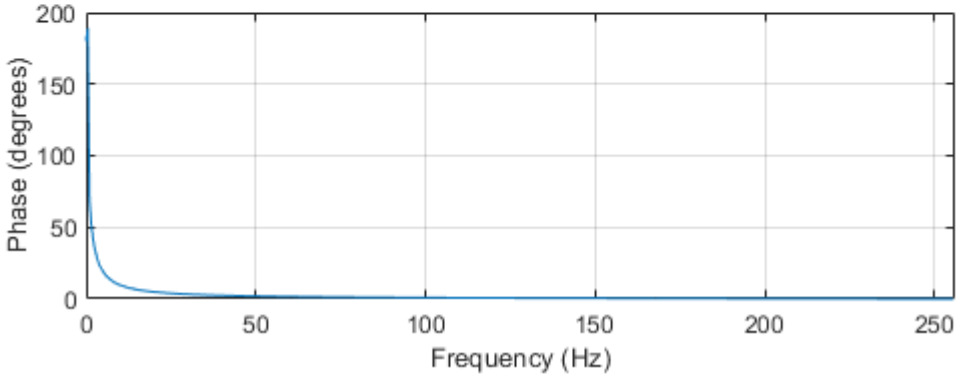
```
% a
x = importdata('SAMPLE_ECG.mat');
Fs = 512;
dt = 1 / Fs;
T = length(x) / Fs;
t = 0 : dt : T - dt;
figure
plot(t,x); title('ECG Data');

% b
L = length(x);
n = pow2(nextpow2(L));
y_dft = fft(x, n);
y_s = fftshift(y_dft);
f = (-n/2 : n/2 - 1)*(Fs / n);

figure
plot(f, abs(y_s) / n);
xlabel('Frequency in Hz)');
title('Magnitude Response of the Sinusoidal Signal');

% c
F1 = 0.5; F2 = 0.2;
Wp = (2*F1) / Fs;
```

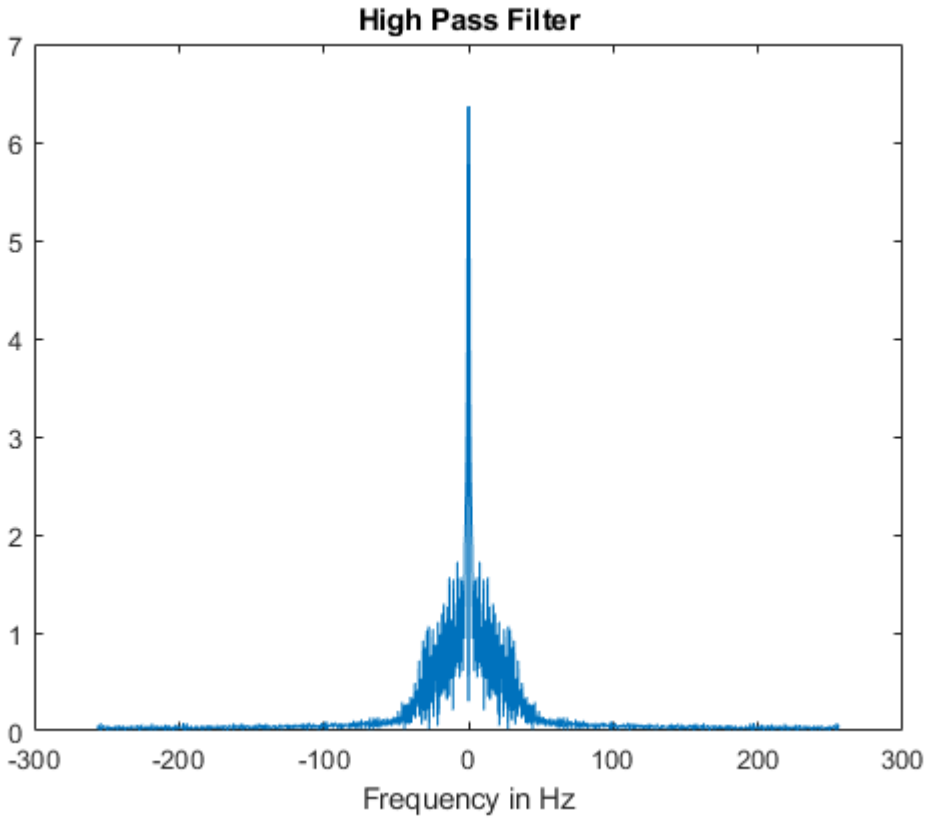
```
Ws = (2*F2) / Fs;
Rp = 3;
Rs = 40;

[Ord, Wn] = ellipord(Wp, Ws, Rp, Rs);
[b, a] = ellip(Ord, Rp, Rs, Wn, 'high');
yf = filter(b, a, x);

figure
n = 512;
freqz(b, a, n, Fs);

L = length(yf);
n = pow2(nextpow2(L));
y_dft = fft(yf, n);
y_s = fftshift(y_dft);
f = (-n/2 : n/2 - 1)*(Fs / n);

figure
plot(f,abs(y_s) /n );

xlabel('Frequency in Hz');
title('High Pass Filter');
```




Magnitude Response of the Sinusoidal Signal
140
120
100
80
60
40
20
0
-300
-200
-100
0
100
200
300
Frequency in Hz)


0
-50
-100
-150
-200
Magnitude (dB)
0
50
100
150
200
250
Frequency (Hz)

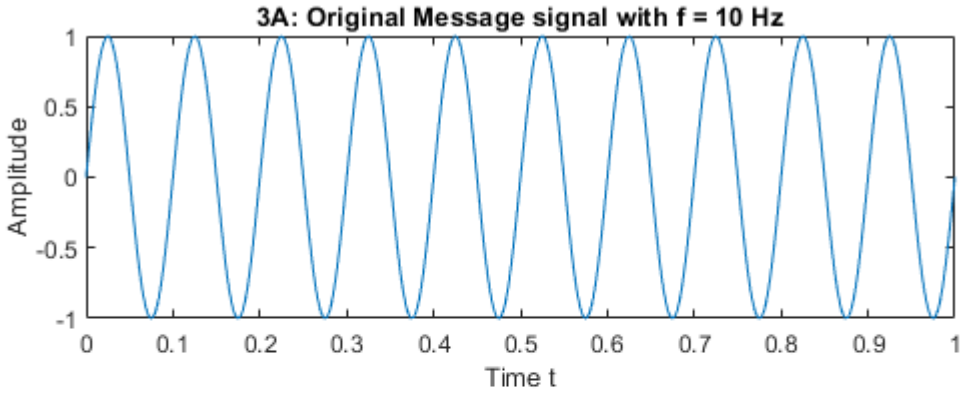

200
150
100
50
0
Phase (degrees)
0
50
100
150
200
250
Frequency (Hz)

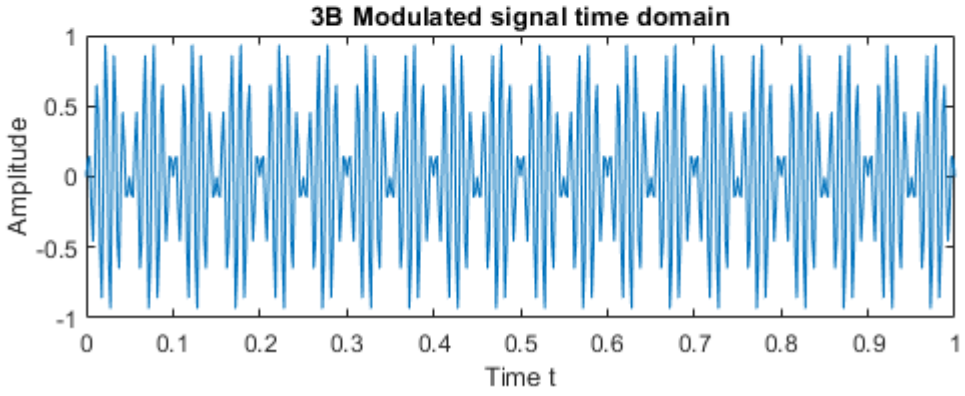


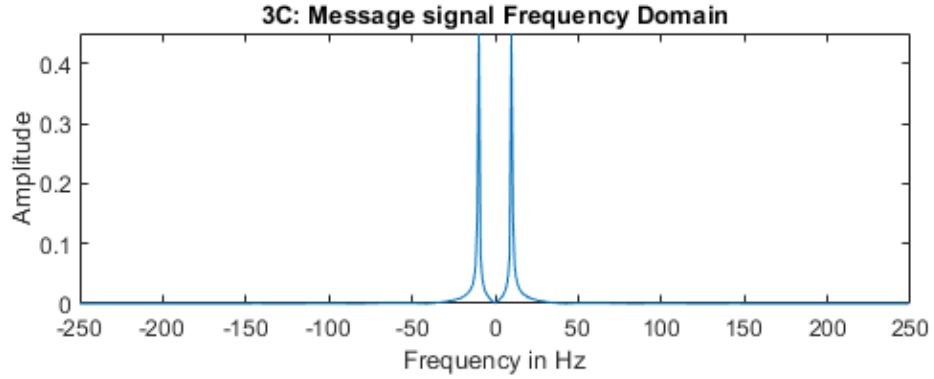
Problem 3

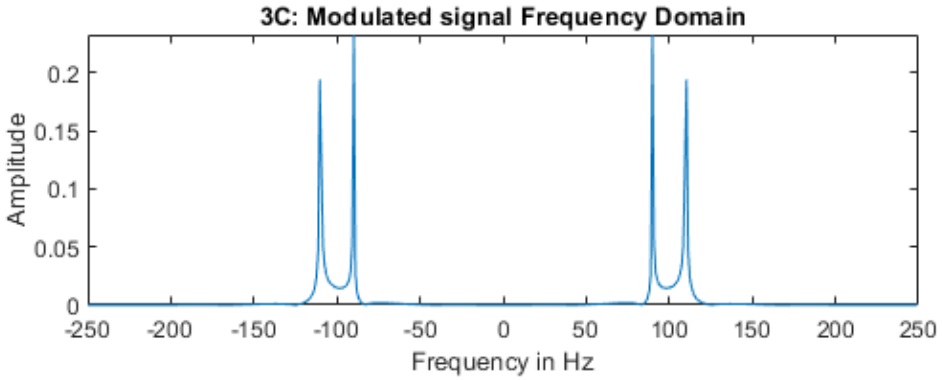
```
% a Plot original message signa;
figure
Fs = 500; f = 10;
dt = 1/Fs;
t = 0 : dt : 1;
x = sin(2*pi*f*t); % Message signal
subplot(2,1,1)
plot(t, x), title('3A: Original Message signal with f = 10 Hz');
xlabel('Time t'), ylabel('Amplitude');

% b
fc = 100;
c = sin(2*pi*fc*t); % Carrier Signal
am_mod = my_ammod(Fs, t, x, c);

subplot(2,1,2)
plot(t, am_mod), title('3B Modulated signal time domain');
xlabel('Time t'), ylabel('Amplitude');


% c Compare Frequency spectrum of message and modulated signal.
% Message Signal
L = length(x);
n = pow2(nextpow2(L));
```

```
x_dft = fft(x,n);
x_s = fftshift(x_dft);
f = (-n/2 : n/2 -1) * (Fs/n);
figure
subplot(2,1,1)
plot(f, abs(x_s) /n ), title('3C: Message signal Frequency Domain');
xlabel('Frequency in Hz'), ylabel('Amplitude');

% Modulate signal
L = length(am_mod);
n = pow2(nextpow2(L));
x_dft = fft(am_mod,n);
x_s = fftshift(x_dft);
f = (-n/2 : n/2 -1) * (Fs/n);
subplot(2,1,2)
plot(f, abs(x_s) /n ), title('3C: Modulated signal Frequency Domain');
xlabel('Frequency in Hz'), ylabel('Amplitude');

% d Demodulate Signal to achieve message signal
am_demod = am_mod.*c;
f1 = 10; f2 = 100;
Wp = (2*f1) / Fs;
Ws = (2*f2) / Fs;
Rp = 3;
Rs = 40;
[Ord, Wn] = ellipord(Wp, Ws, Rp, Rs);
[b, a] = ellip(Ord, Rp, Rs, Wn);
yf = filter(b, a, am_demod);

% Convert to Frequency Domain
n = 512;
figure
freqz(b, a, n, Fs);
L = length(yf);
n = pow2(nextpow2(L));
y_dft = fft(yf, n);
y_s = fftshift(y_dft);
f = (-n/2 : n/2 - 1)*(Fs / n);

% Plot time and frequency domain.
figure
subplot(2, 1, 1)
plot(t, yf), title('Low pass filter time domain'), xlabel('Time');
subplot(2, 1, 2)
plot(f,abs(y_s) /n ), xlabel('Frequency in Hz'), title('Elliptic Low-Pass filter');
```

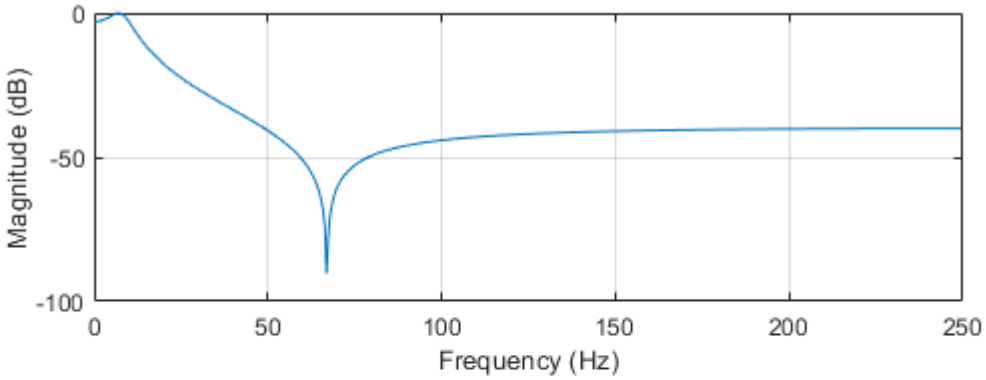


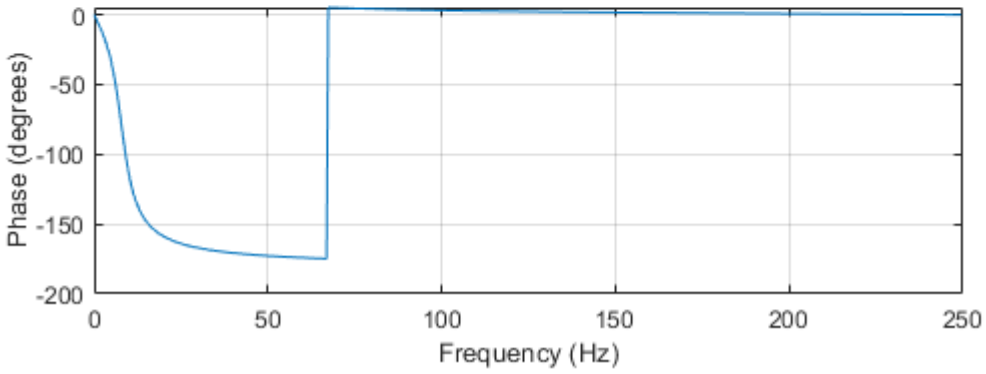


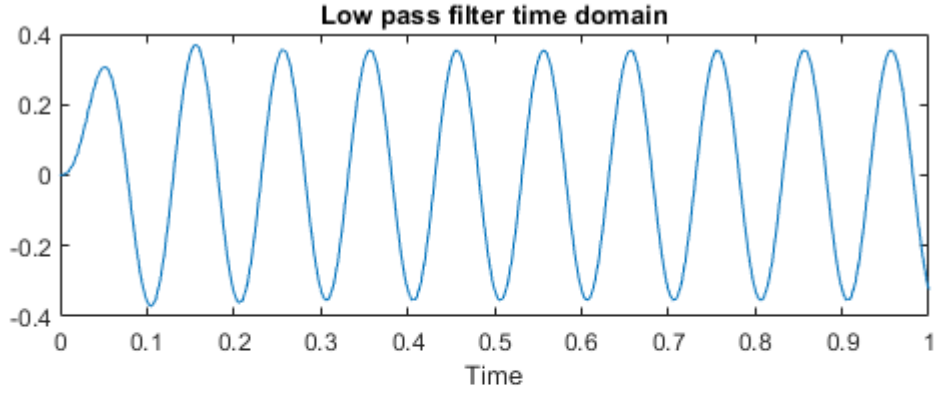


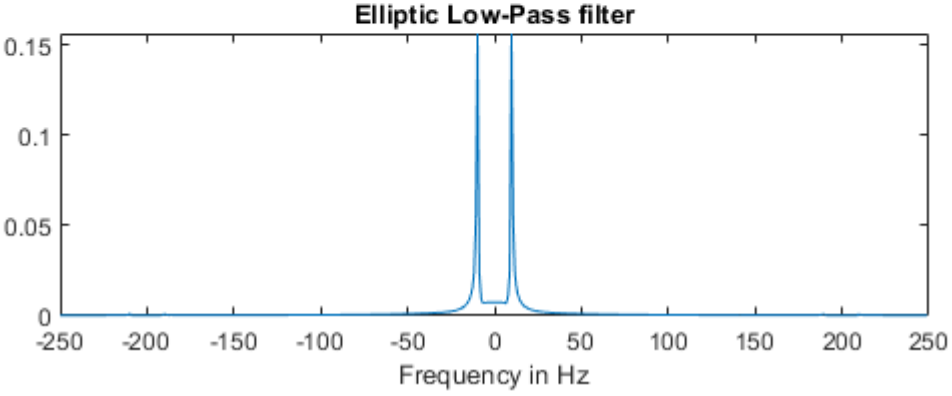


Modulation around the -100 and 100 hz, original only has around 25hz.


Magnitude (dB)
0
-50
-100
0
50
100
150
200
250
Frequency (Hz)


Phase (degrees)
0
-50
-100
-150
-200
0
50
100
150
200
250
Frequency (Hz)


Low pass filter time domain
0.4
0.2
0
-0.2
-0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time


Elliptic Low-Pass filter
0.15
0.1
0.05
0
-250
-200
-150
-100
-50
0
50
100
150
200
250
Frequency in Hz

Functions

```
% Amplitude modulation
function y = my_ammod(Fs, t, x, c)
    y = x .* c;
end

% Convert to frequency domain
function [f, y_s, n] = my_freq_domain(Fs, y)
    L = length(y);
    n = pow2(nextpow2(L));
    y_dft = fft(y,n);
    y_s = fftshift(y_dft);
    f = (-n/2 : n/2 -1) * (Fs/n);
end
```

Published with MATLAB® R2020b