Table of Contents

Problem 1

```
Fs = 8000;
Ts = 1/Fs;
t = 0:Ts:0.0005;    %0.5ms
f1 = 1209;
f2 = 1336;
y = cos(2*pi*f1*t) + cos(2*pi*f2*t);
```

1a Plot in time and freq domain

```
figure(1)
% Plot in time domain
plot(t, y), xlabel('Time (s)'), ylabel('Amplitude');
title('Signal y(t) in the Time Domain');
% Perform FFT and plot in the frequency domain
Y = fft(y);
frequencies = linspace(0, Fs, length(Y));
figure(2)
plot(frequencies, abs(Y)), xlabel('Frequency (Hz)'), ylabel('Magnitude');
title('Signal y(t) in the Frequency Domain');
xlim([0, Fs/2]);    %limit x-axis
```

1b Low Pass Filter

Butterworth, drop off gradual

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

```
[ord,Wn] = buttord(Wp, Ws, Rp, Rs);
[Num, Den] = butter(ord, Wn, 'low');
yf = filter(Num, Den, y);

figure(3);
n = 512;
freqz(Num, Den, n, Fs); % n is evaualation points
% FFT
L = length(yf);                  % Window
n = pow2(nextpow2(L));         % next power of 2 from length of y
yf_dft = fft(yf,n);              % DFT
y_s = fftshift(yf_dft);          % Rearrange y values
f = (-n/2:n/2-1)*(Fs/n);       % 0 centered frequency range

figure(4)
subplot(2, 1, 1);
plot(t,yf), xlabel('time'), ylabel('Magnitude'),title('Signal in time domain')
subplot(2, 1, 2);
plot(f,abs(y_s)/n), xlabel('Frequency (Hz)'), title('Magnitude response of the
 sinusoidal signal')
```

1c High Pass filter

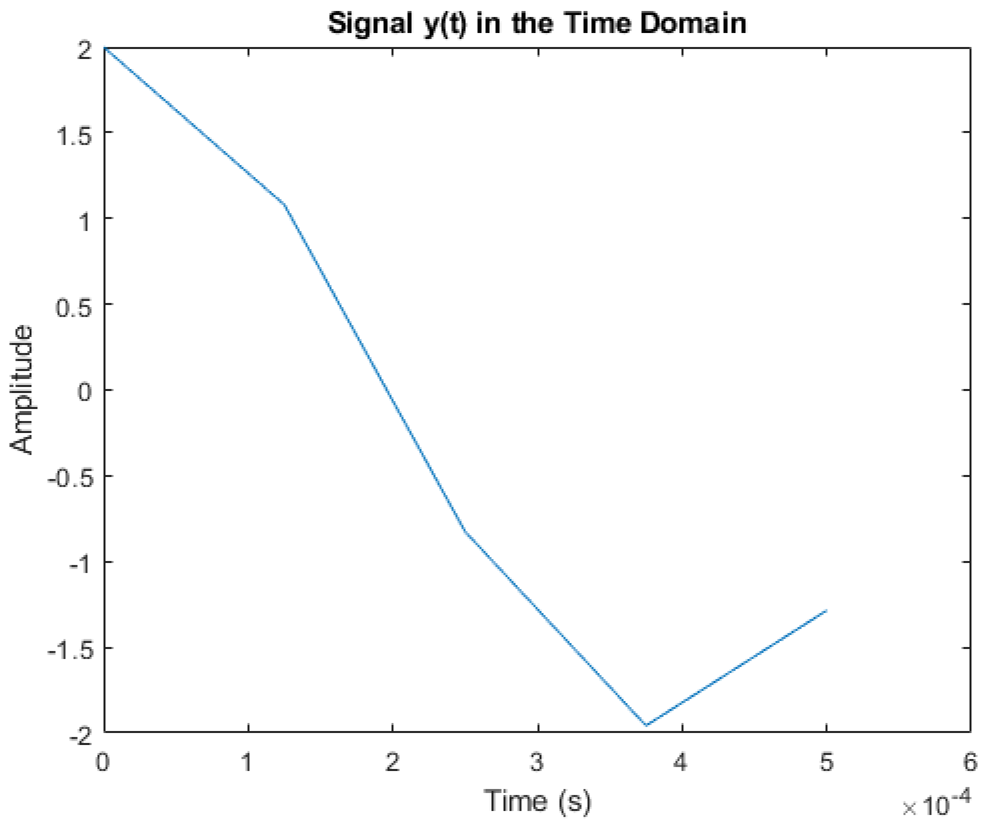
High Pass Elliptic Filter

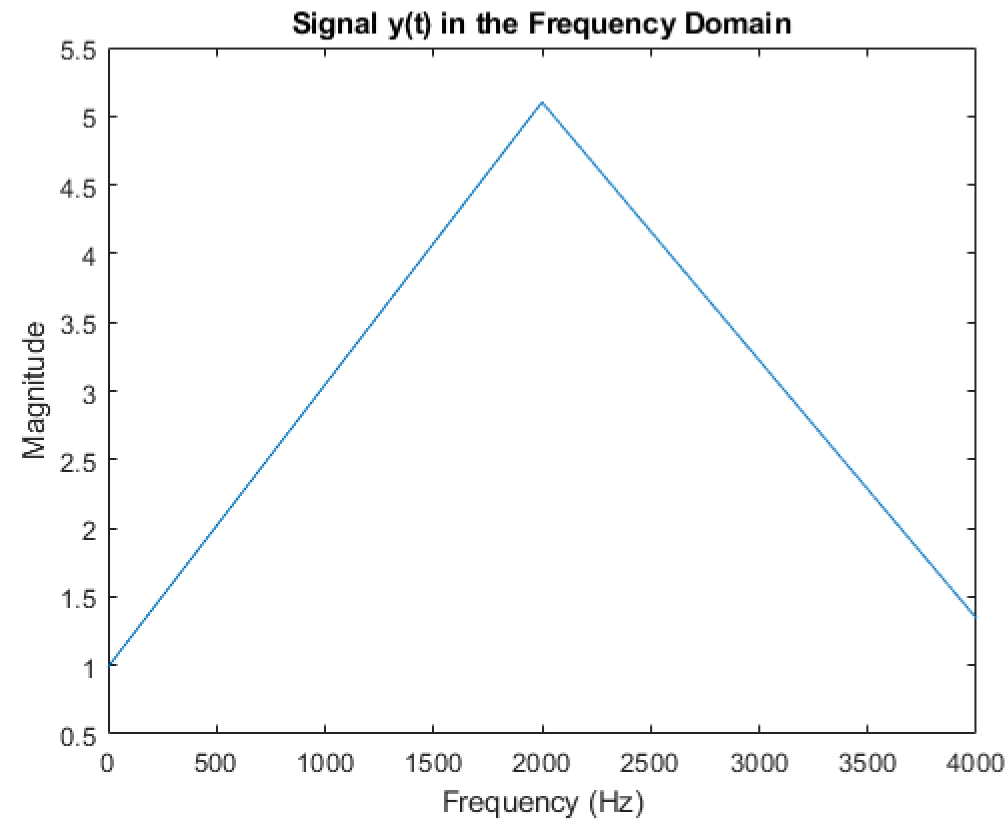
```
Wp2 = (2*f2)/Fs;
Ws2 = (2*f1)/Fs;
Rp2 = 3;
Rs2 = 40;
[n2,Wn2] = ellipord(Wp2, Ws2, Rp2, Rs2);
[Num2,Den2] = ellip(n2, Rp2, Rs2, Wn2,'high');
yf2 = filter(Num2, Den2, y);

figure(5);
freqz(Num2, Den2, n, Fs);
%
L = length(yf2);                  % Window
n = pow2(nextpow2(L));         % next power of 2 from length of y
yf2_dft = fft(yf2,n);              % DFT
y_s2 = fftshift(yf2_dft);          % Rearrange y values
f = (-n/2:n/2-1)*(Fs/n);       % 0 centered frequency range

figure(6)
subplot(2, 1, 1);
plot(t,yf2), xlabel('time'), ylabel('Magnitude'),title('Signal in time
 domain')
subplot(2, 1, 2);
plot(f,abs(y_s2)/n), xlabel('Frequency (Hz)'), title('Magnitude response of
 the sinusoidal signal')
```

1c Bandpass Filter

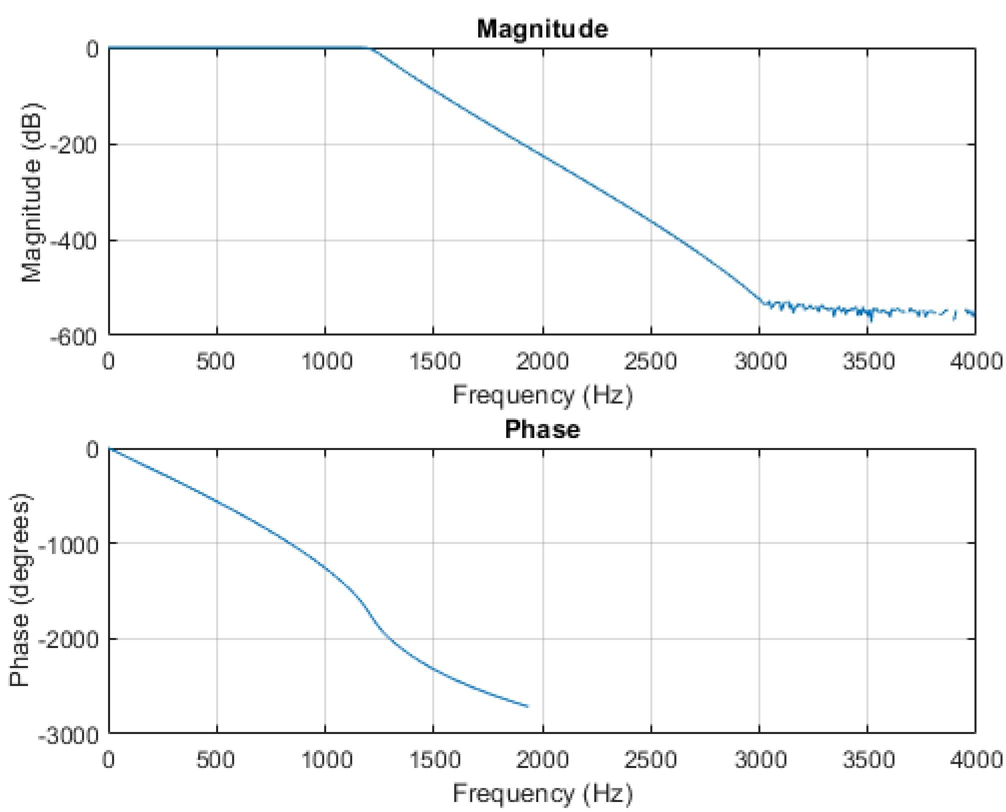
bandpass elliptic filter design

```
Wpb = (2*[1200 1225])/Fs;
Wsb = (2*[1110 1240])/Fs;
%ws1 < wp1 < wp2 < ws2
Rpb = 3;
Rsb = 40;
[nb, Wnb] = ellipord(Wpb, Wsb, Rpb, Rsb);
[Numb, Denb] = ellip(nb, Rpb, Rsb, Wnb, 'bandpass');
yfb = filter(Numb, Denb, y);

figure(7);
freqz(Numb,Denb, n, Fs);

L = length(yfb);                % Window
n = pow2(nextpow2(L));        % next power of 2 from length of y
yfb_dft = fft(yfb,n);             % DFT
y_sb = fftshift(yfb_dft);       % Rearrange y values

figure(8)
subplot(2, 1, 1);
plot(t,yfb), xlabel('time'), ylabel('Magnitude'),title('Signal in time
 domain')
subplot(2, 1, 2);
plot(f,abs(y_sb)/n), xlabel('Frequency (Hz)'), title('Magnitude response of
 the sinusoidal signal')
```

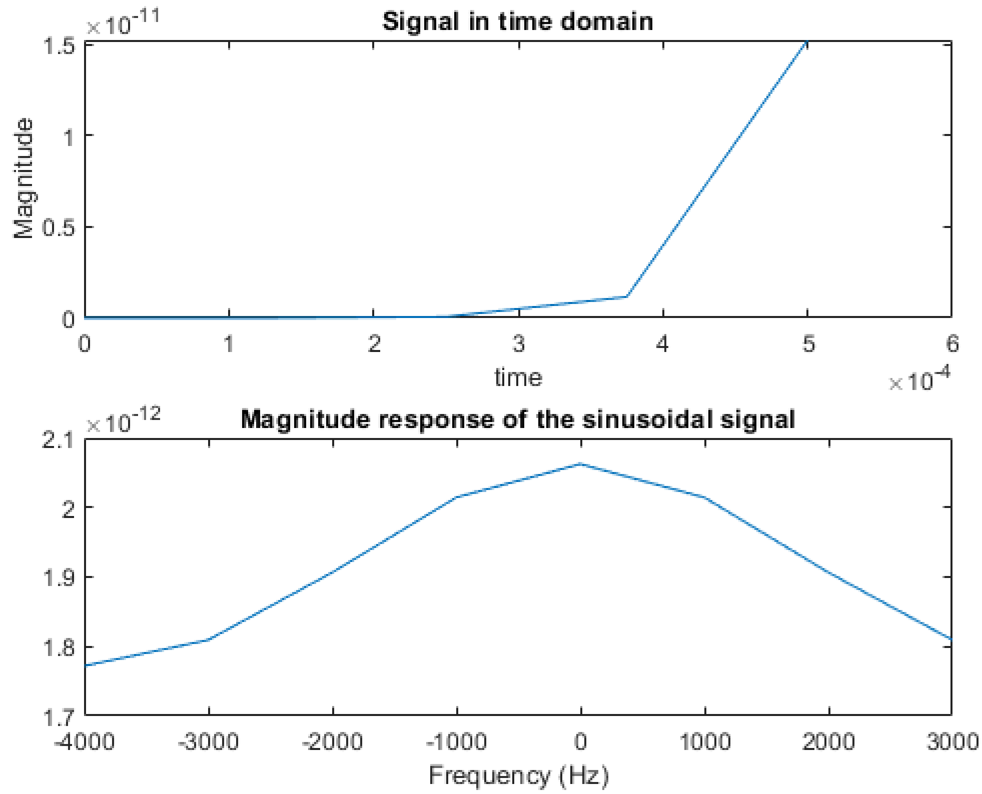


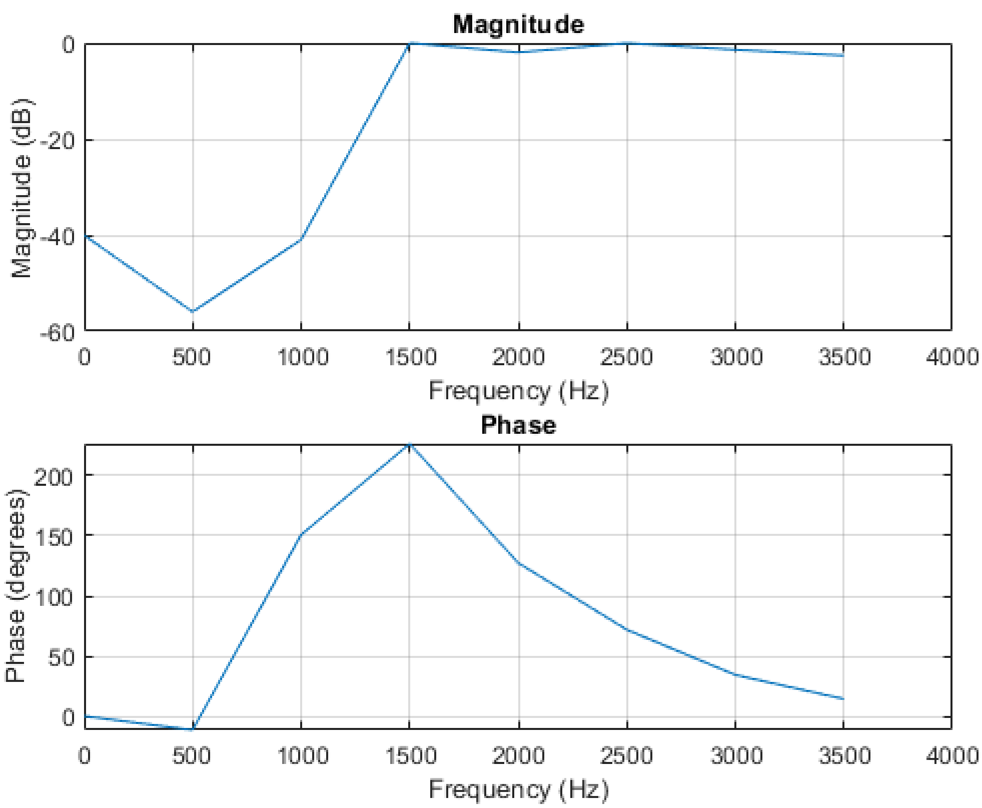


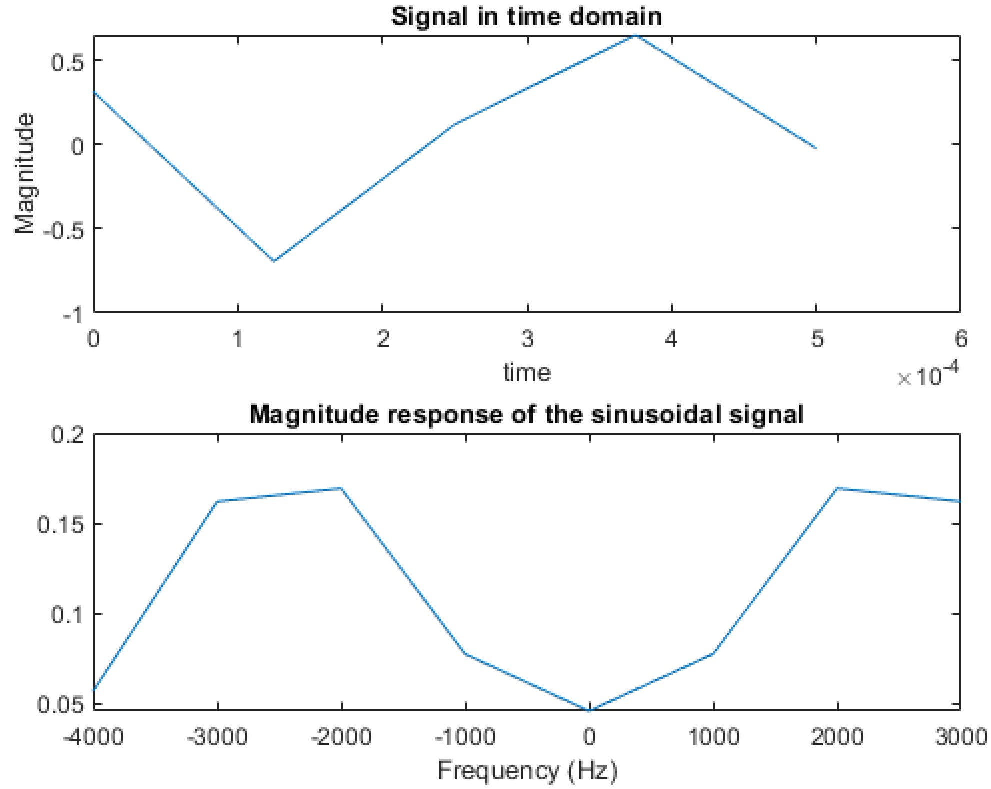
Signal y(t) in the Frequency Domain
Magnitude
Frequency (Hz)

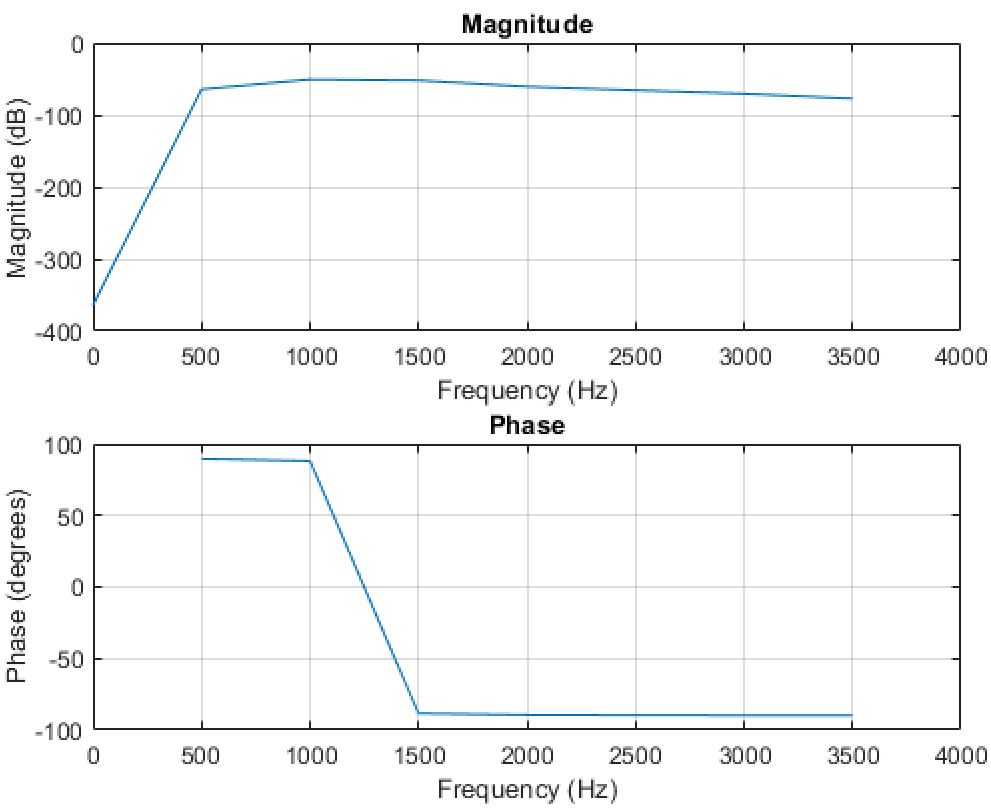


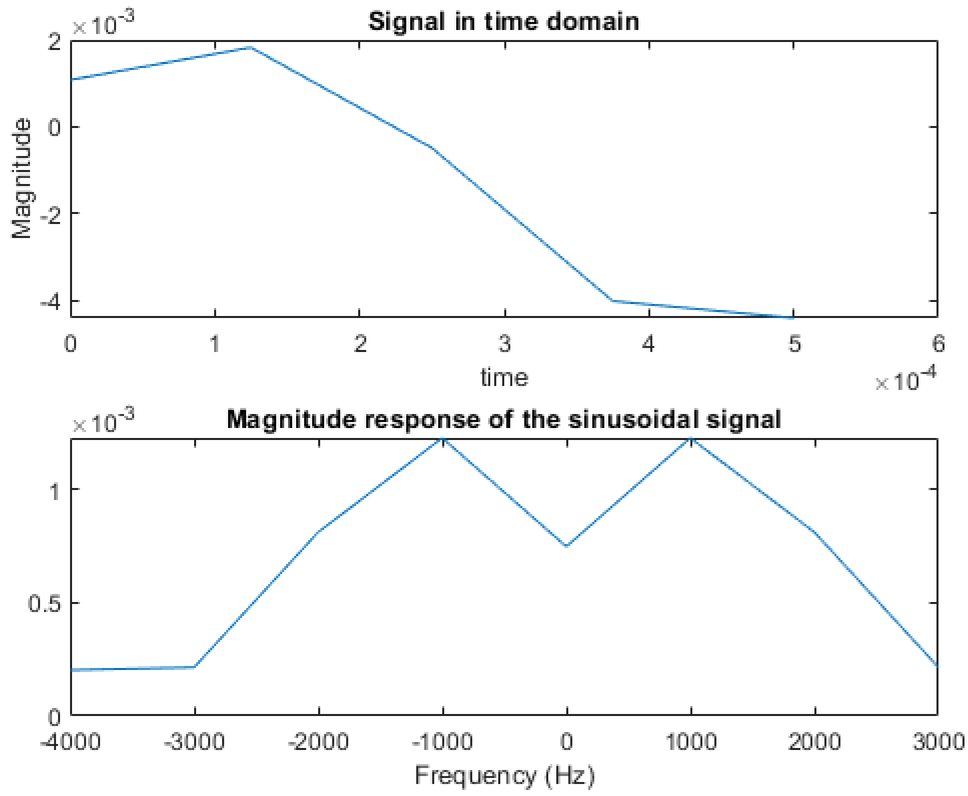
Magnitude
Magnitude (dB)
Frequency (Hz)
Phase
Phase (degrees)
Frequency (Hz)


Signal in time domain
×10^-11
Magnitude
time
×10^-4
Magnitude response of the sinusoidal signal
×10^-12
Frequency (Hz)


Magnitude
Magnitude (dB)
Frequency (Hz)
Phase
Phase (degrees)
Frequency (Hz)


Signal in time domain
Magnitude
time
×10^-4
Magnitude response of the sinusoidal signal
Frequency (Hz)


Magnitude
Magnitude (dB)
Frequency (Hz)
Phase
Phase (degrees)
Frequency (Hz)



Problem 2

```
load SAMPLE_ECG.mat
x = ECG_Data;
Fs = 512;
T = length(x)/Fs;
Ts = 1/Fs;
t = 0:Ts:T-Ts;
```

2a Plot signal vs time and estimate heart rate

```
figure(9)
plot(t,x), xlabel('Time'), ylabel('ECG Signal'), title('ECG Data')
%Estimate Heart Rate (go back and check)
heart_rate = 60 * length(findpeaks(x)) / (t(end) - t(1));  % Assuming
 findpeaks function is available
disp(['Estimated Heart Rate: ' num2str(heart_rate) ' beats per minute']);
```

2b Plot Freq Spectrum

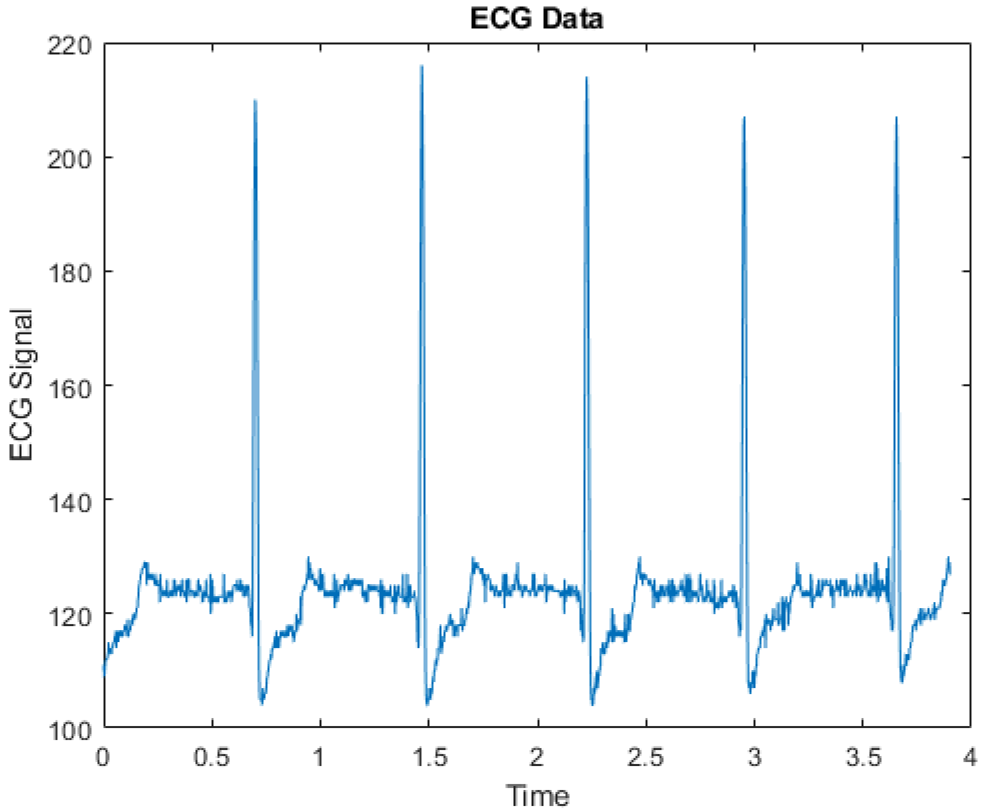
```
L = length(x);
n = pow2(nextpow2(L));
Y = fft(x, n);  % Compute the magnitude of the Fourier transform
y_s = fftshift(Y);
```

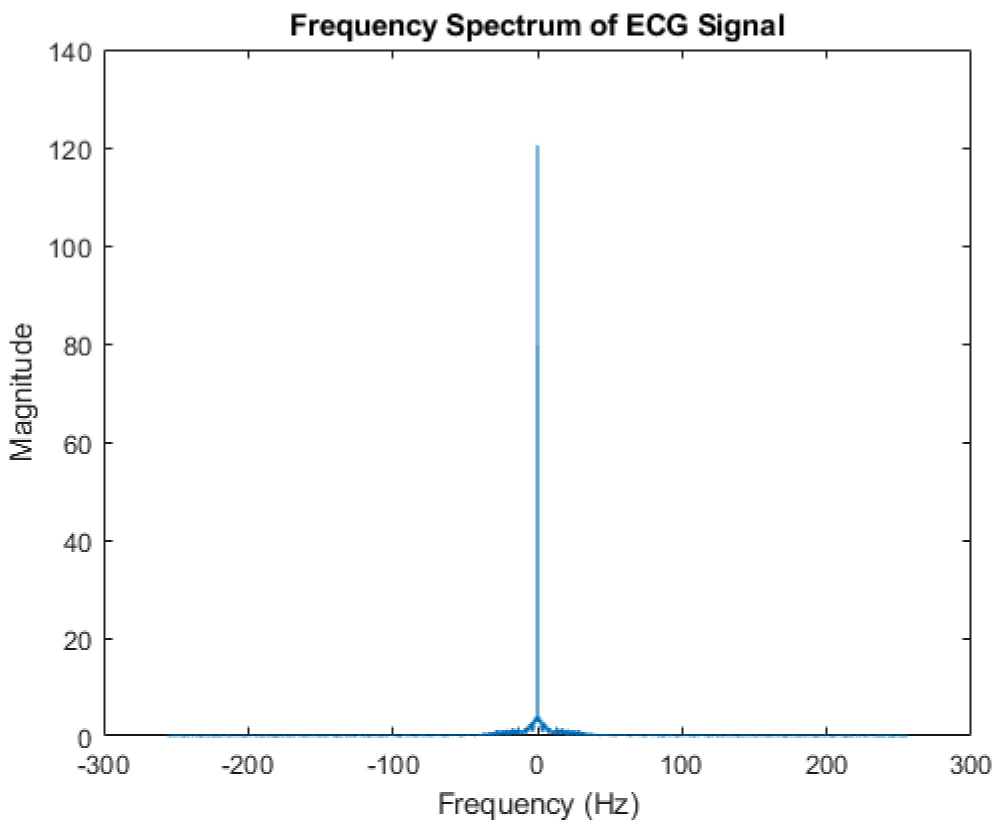
```
f = (-n/2:n/2-1) * (Fs/n);  % Frequency vector based on the sampling rate
figure(10)
plot(f, abs(y_s)/n), xlabel('Frequency (Hz)'), ylabel('Magnitude');
title('Frequency Spectrum of ECG Signal');
```

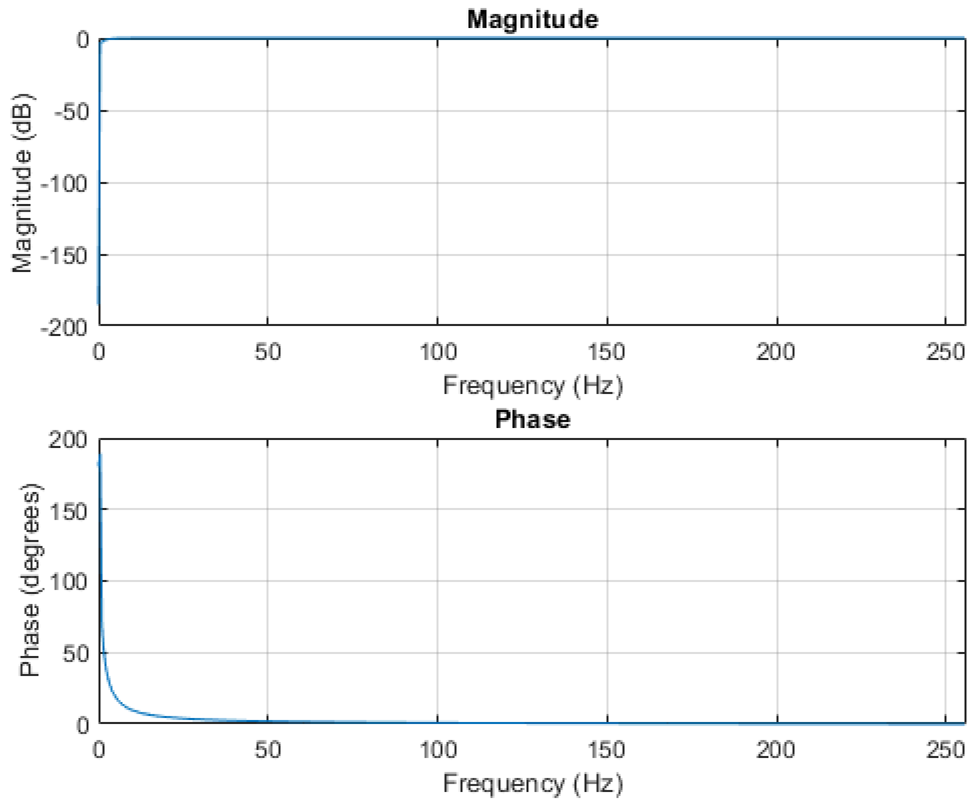
2c Baseline Wander Removal (High Pass filter design)

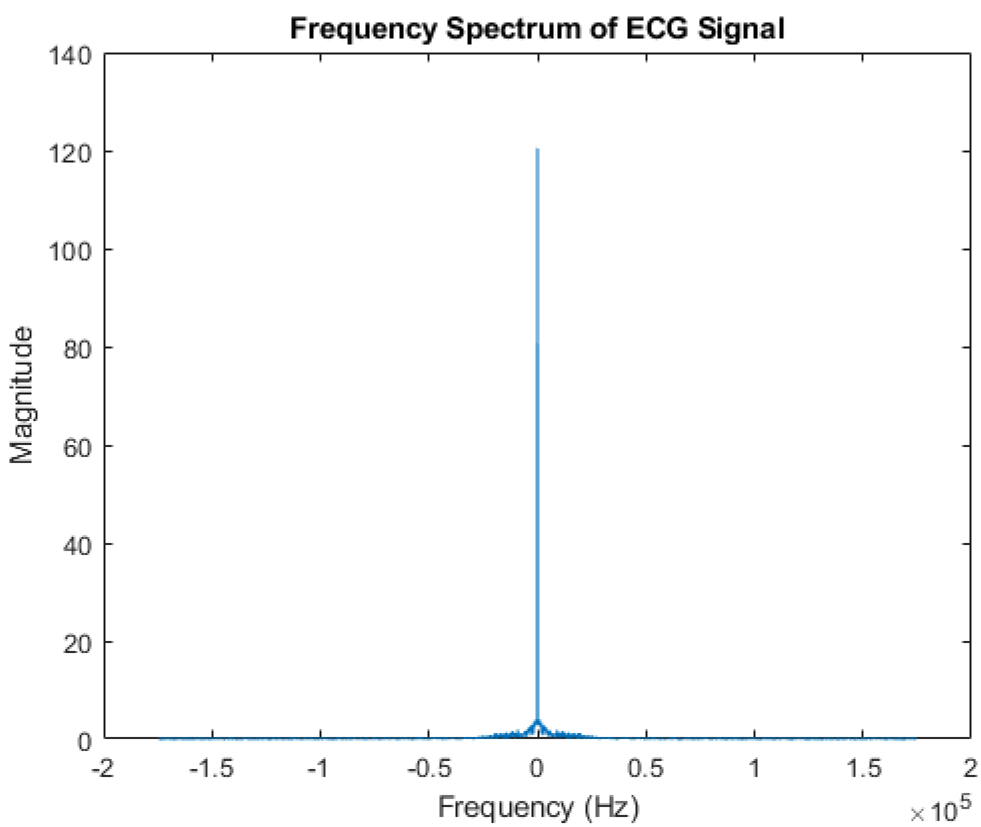
```
fp = 0.5;    %passband
fs = 0.2;    %stopband
Wp = (2*fp)/Fs;
Ws = (2*fs)/Fs;
Rp = 3;
Rs = 40;
[n2,Wn] = ellipord(Wp, Ws, Rp, Rs);
[Num, Den] = ellip(n2, Rp, Rs, Wn, 'high');
yf = filter(Num, Den, x);

figure(11);
nd = 512;
freqz(Num, Den, nd, Fs);

L = length(x);
n3 = pow2(nextpow2(L));
Y2 = fft(x, n3);  % Compute the magnitude of the Fourier transform
y_s2 = fftshift(Y2);
f2 = (-n3/2:n3/2-1) * (Fs/3);  % Frequency vector based on the sampling rate
figure(12)
plot(f2, abs(y_s2)/n3), xlabel('Frequency (Hz)'), ylabel('Magnitude');
title('Frequency Spectrum of ECG Signal');
```

```
Estimated Heart Rate: 5839.7199 beats per minute
```


ECG Data
ECG Signal
Time
220
200
180
160
140
120
100
0
0.5
1
1.5
2
2.5
3
3.5
4


Frequency Spectrum of ECG Signal
Magnitude
Frequency (Hz)
140
120
100
80
60
40
20
0
-300
-200
-100
0
100
200
300


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


Frequency Spectrum of ECG Signal
Magnitude
140
120
100
80
60
40
20
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
Frequency (Hz)
$\times 10^5$

Problem 3

```
Fs = 500;
Ts = 1/Fs;
t = 0:Ts:1;
f = 10;     % 10 hz
x = sin(2*pi*f*t);   % message

figure(13)
plot(t,x), title(['3a. Message signal with f = ', num2str(f), ' Hz']);
xlabel('Time, t'), ylabel('Amplitude')
```

3b Modulate Signal

```
fc = 100;
y = sin(2*pi*f*t);  % message
mod_signal = my_ammod(y,fc,Fs);
```

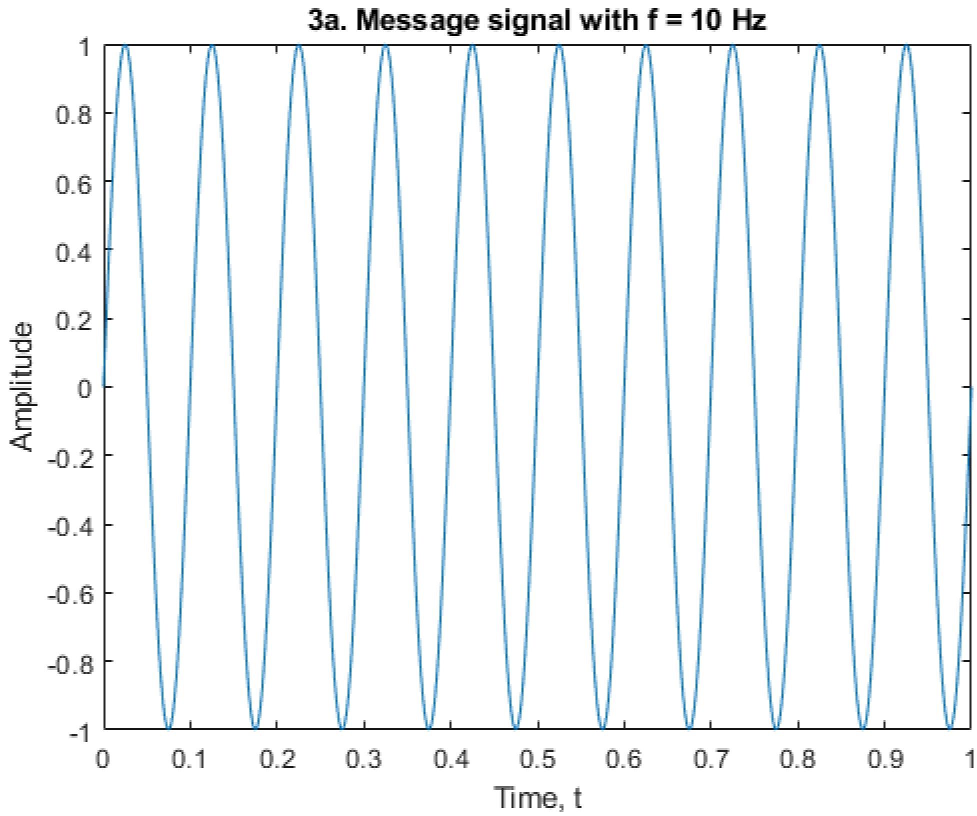
3c freq specturm of plots

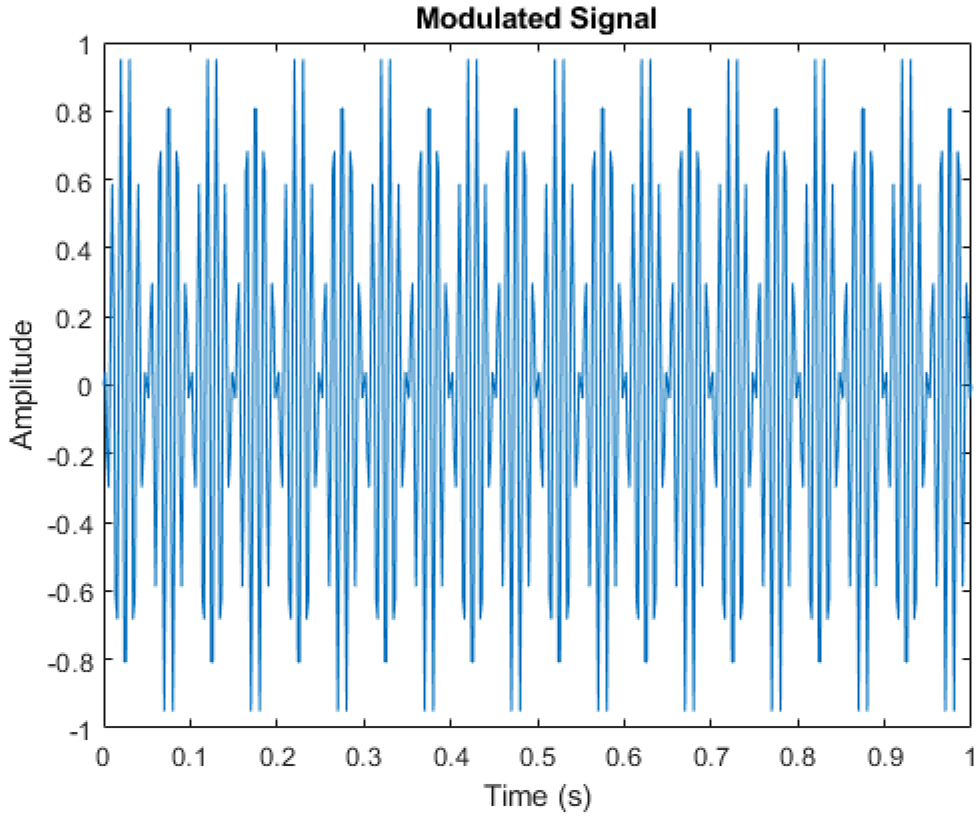
```
y_psd = periodogram(x);
mod_psd = periodogram(mod_signal);
figure(14);
subplot(2,1,1);
plot(10*log10(y_psd));
title('Frequency Spectrum of Message Signal');
xlabel('Frequency (Hz)'), ylabel('Magnitude');
subplot(2,1,2);
plot(10*log10(mod_psd));
title('Frequency Spectrum of Modulated Signal');
xlabel('Frequency (Hz)'), ylabel('Magnitude');

% The difference between the two plots is that the message signal has a
% small peak at 10 Hz whereas the modulatated signal has one at 100 Hz
```

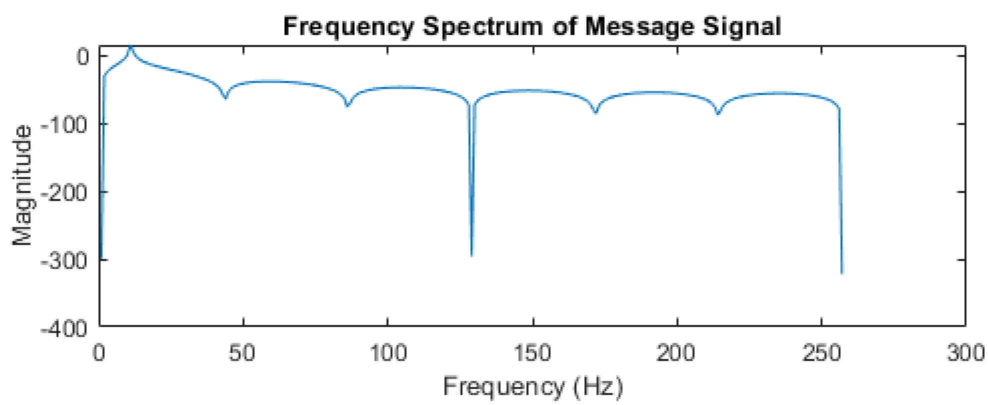
3d Demodulate Signal

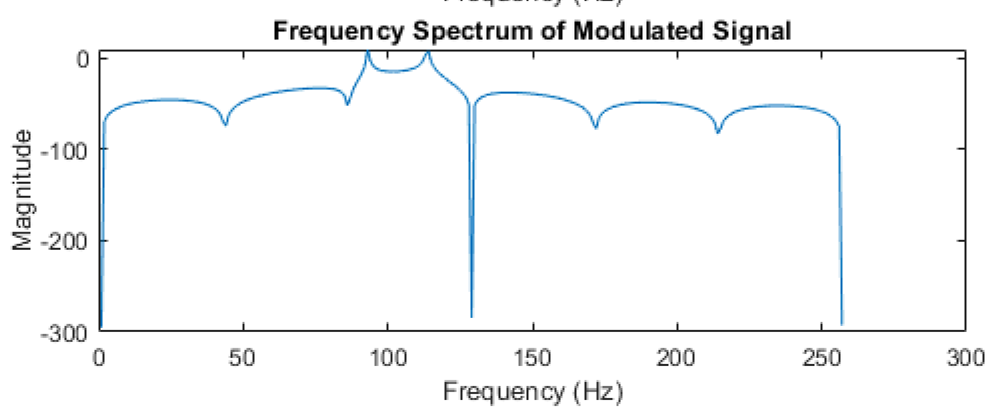
```
demod_signal = my_amdemod(mod_signal,fc,Fs);
% Plot inside function
% Freq specturm plot
figure(15)
demod_psd = periodogram(demod_signal);
plot(10*log10(demod_psd));
title('Frequency Spectrum of Demodulated Signal');
xlabel('Frequency'), ylabel('Magnitude');

% Compared to the message signal, you can see the message signal a lot more
% clearly than the modulated and demodulated. After demodulating the
% message there is less noise the closer you get to 0 versus the modulated
% one this is because the demodulated signal uses a low pass filter on the
```

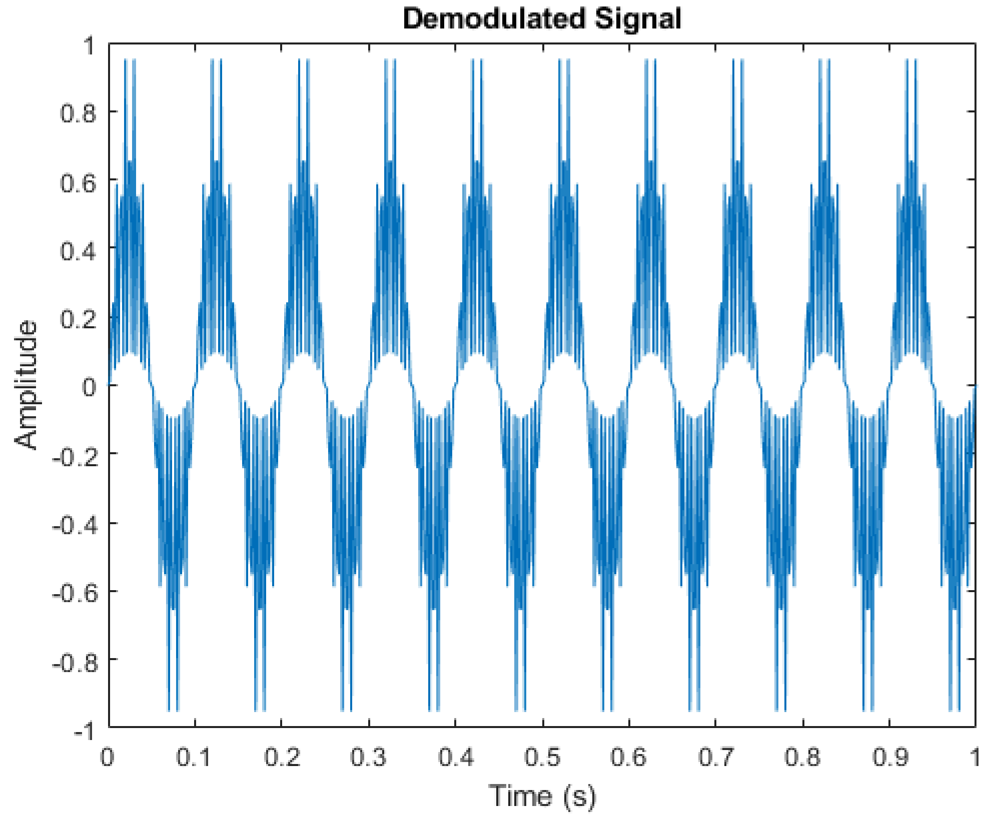
```
% result.
```

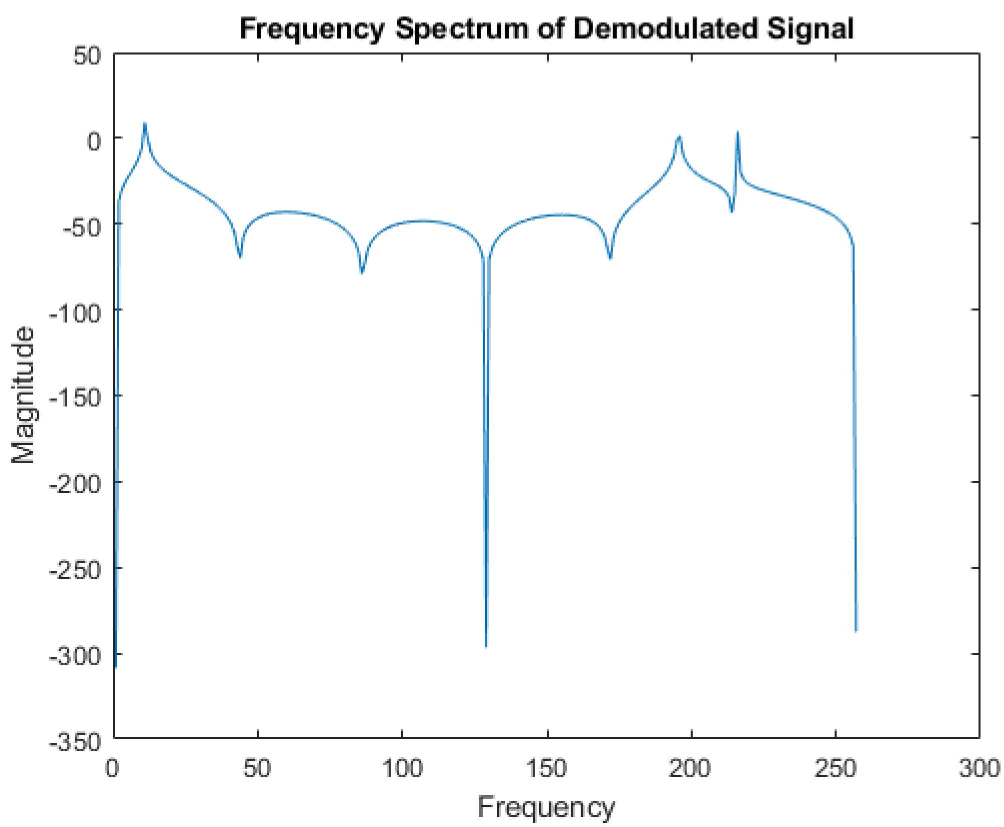

3a. Message signal with f = 10 Hz
Amplitude
Time, t
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Modulated Signal
Amplitude
Time (s)


Frequency Spectrum of Message Signal
Magnitude
Frequency (Hz)


Frequency Spectrum of Modulated Signal
Magnitude
Frequency (Hz)


Demodulated Signal
Amplitude
Time (s)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Frequency Spectrum of Demodulated Signal
Magnitude
Frequency
50
0
-50
-100
-150
-200
-250
-300
-350
0
50
100
150
200
250
300

Modulated Signal Funciton

```
function mod_signal = my_ammod(mesg,carFreq, sampFreq)
    t = (0:length(mesg) - 1)/sampFreq;
    carSig = cos(2*pi*carFreq*t);
    mod_signal = mesg .* carSig;

    % Plot
    figure(16)
    plot(t, mod_signal), xlabel('Time (s)'), ylabel('Amplitude'),
 title('Modulated Signal');
end
```

Demodulated Signal Function

```
function demod_signal = my_amdemod(mod_signal,carFreq, Fs)
    t = (0:length(mod_signal) -1) / Fs;
    carSignal = cos(2*pi*carFreq*t);
    demod_signal = mod_signal .* carSignal;

    figure(17)
    plot(t, demod_signal);
    xlabel('Time (s)'), ylabel('Amplitude'), title('Demodulated Signal');
end
```

Published with MATLAB® R2022b