EC 415: Homework 4

Due by Friday 04/09/2021 6:00PM

Professor David Starobinski

Michael Kremer

kremerme@bu.edu

Exercise 4.21

Suppose that the noise in improvesnr.m is replaced with narrowband noise (as discussed in Section 4.1.3). Investigate the improvements in SNR

a. when the narrowband interference occurs outside the 3000 to 4000 Hz passband,

b. when the narrowband interference occurs inside the 3000 to 4000 Hz passband.

For part (a) use $n = 0.1 * (\cos(2 * pi * f1 * t) + \cos(2 * \pi * f_2 * t))$ to model narrowband noise around the frequencies f_1 and f_2. Choose $f_1 = 2000$ Hz and $f_2 = 5000$Hz.
For part (b) use $n = 0.1 * \cos(2 * \pi * f3 * t)$ to model narrowband noise around the frequency f_3. Choose f_3 = 3500 Hz.

Solution

a. using this code:

Listing 1: MATLAB code for Exercise 4.21a

```
% improvesnr.m: using linear filters to improve SNR
time=3; Ts=1/20000;                     % time and sampling interval
freqs=[0 0.29 0.3 0.4 0.41 1];          % filter design, bandlimited
amps=[0 0 1 1 0 0];                     % ...between 3K and 4K
b=firpm(100,freqs,amps);                % BP filter
f1 = 2000; f2 = 5000;
x=filter(b,1,2*randn(1,time/Ts));       % do the filtering
N=length(x);                          % length of the signal x
t=Ts*(1:N);                           % define time vector
n=0.1*(cos(2*pi*f1*t)+cos(2*pi*f2*t)); % generate white noise signal

y=filter(b,1,x+n);                      % (a) filter the signal+noise
yx=filter(b,1,x);                       % or (b) filter signal
yn=filter(b,1,n);                       % ...and noise separately
z=yx+yn;                                % add them
diffzy=max(abs(z-y))                    % and make sure y = z
snrinp=pow(x)/pow(n)                    % SNR at input
snrout=pow(yx)/pow(yn)                  % SNR at output

% check spectra
figure(1),plotspec(n,Ts)
figure(2),plotspec(x,Ts)
figure(3),plotspec(x+n,Ts)
figure(4),plotspec(y,Ts)

%Here's how the figure improvesnr.eps was actually drawn

ssf=(-N/2:N/2-1)/(Ts*N);              % frequency vector
fx=fftshift(fft(x(1:N)+n(1:N)));
figure(5), subplot(2,1,1), plot(ssf,abs(fx))
xlabel('magnitude spectrum of signal + noise')
fy=fftshift(fft(y(1:N)));
subplot(2,1,2), plot(ssf,abs(fy))
xlabel('magnitude spectrum after filtering')
```

the SNR at the input is 47 while the SNR at the output is 1906. Looking at the chart below, this makes sense because the noise falls outside of the desired passband range.

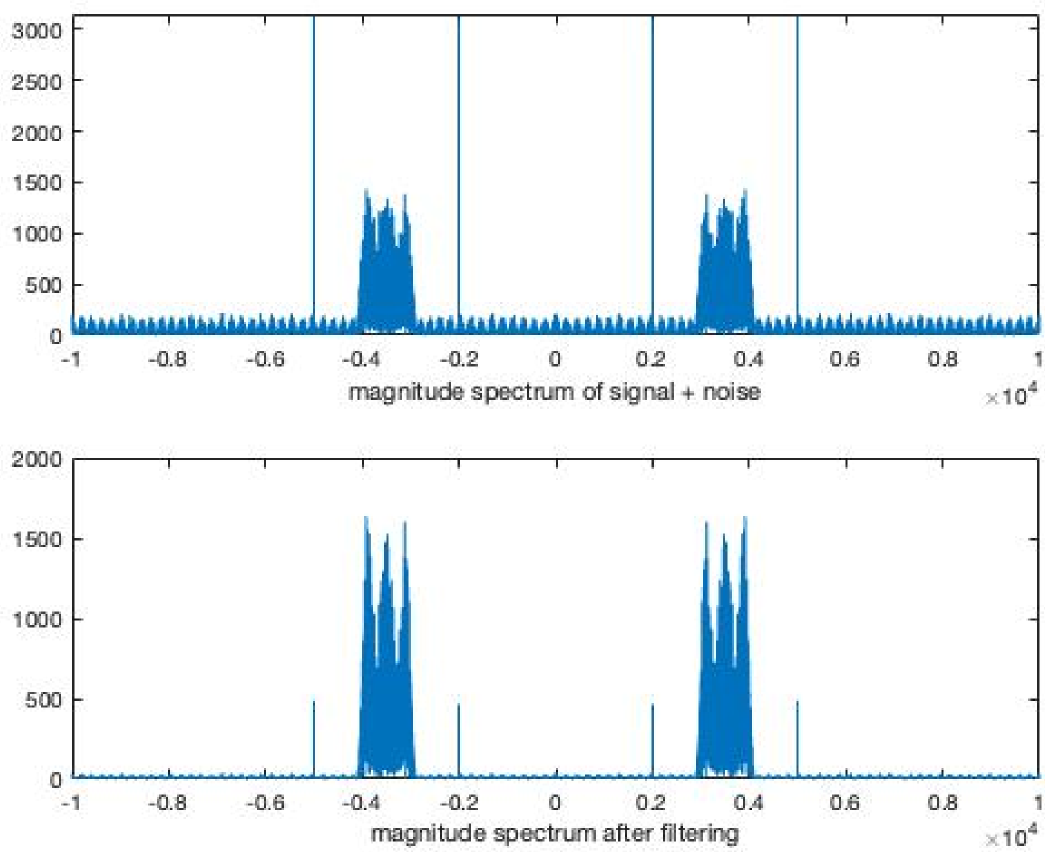


Figure 1: Figure 5.8

b. using this code:

Listing 2: MATLAB code for Exercise 4.21b

```
% improvesnr.m: using linear filters to improve SNR
time=3; Ts=1/20000;                    % time and sampling interval
freqs=[0 0.29 0.3 0.4 0.41 1];         % filter design, bandlimited
amps=[0 0 1 1 0 0];                    % ...between 3K and 4K
b=firpm(100,freqs,amps);               % BP filter
f3=3500;
x=filter(b,1,2*randn(1,time/Ts));      % do the filtering
N=length(x);                           % length of the signal x
t=Ts*(1:N);                            % define time vector
n=0.1*cos(2*pi*f3*t); % generate white noise signal

y=filter(b,1,x+n);                     % (a) filter the signal+noise
yx=filter(b,1,x);                      % or (b) filter signal
yn=filter(b,1,n);                      % ...and noise separately
```

```
z=yx+yn;                                  % add them
diffzy=max(abs(z-y))                      % and make sure y = z
snrinp=pow(x)/pow(n)                      % SNR at input
snrout=pow(yx)/pow(yn)                    % SNR at output

% check spectra
figure(1), plotspec(n,Ts)
figure(2), plotspec(x,Ts)
figure(3), plotspec(x+n,Ts)
figure(4), plotspec(y,Ts)

%Here's how the figure improvesnr.eps was actually drawn

ssf=(-N/2:N/2-1)/(Ts*N);                 % frequency vector
fx=fftshift(fft(x(1:N)+n(1:N)));
figure(5), subplot(2,1,1), plot(ssf,abs(fx))
xlabel('magnitude spectrum of signal + noise')
fy=fftshift(fft(y(1:N)));
subplot(2,1,2), plot(ssf,abs(fy))
xlabel('magnitude spectrum after filtering')
```

the SNR at the input is 97 while the SNR at the output is 68. Looking at the chart below, this makes sense because the noise falls inside of the desired passband range.

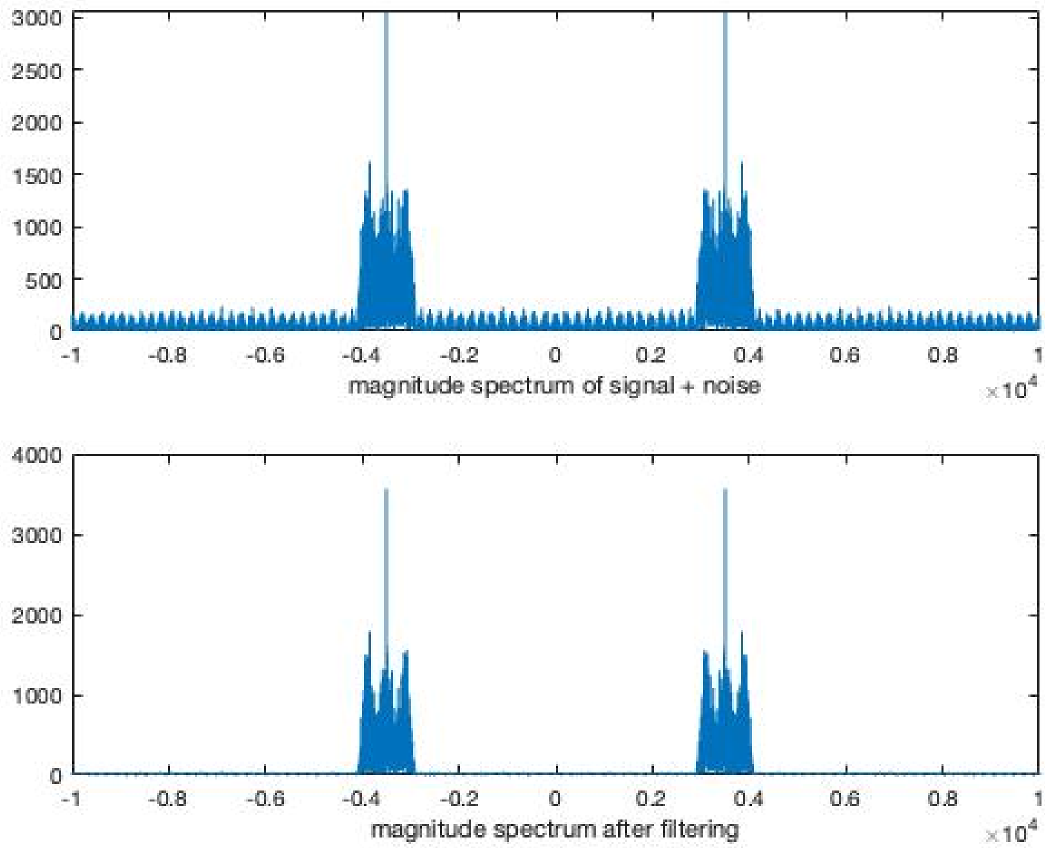


Figure 2: Figure 5.8

Exercise 5.9

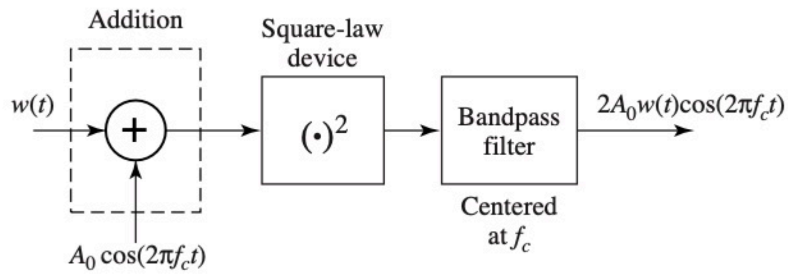


Figure 5.8 The square-law mixing transmitter of Exercises 5.9 through 5.11.

Figure 3: Figure 5.8

Consider the system shown in Figure 5.8. Show that the output of the system is $2A_0w(t)\cos(2\pi f_c t)$, as indicated.

Solution

$[w(t) + A_0\cos(2\pi f_c t)]^2 = w(t)^2 + 2A_0w(t)\cos(2\pi f_c t) + \cos(2\pi f_c t)^2$

$\cos(2\pi f_c t)^2$ is centered at 0 and $2f_c$and filtered out
$w(t)^2$ is centered at 0 and filtered out

output is $2A_0w(t)\cos(2\pi f_c t)$

Exercise 5.12

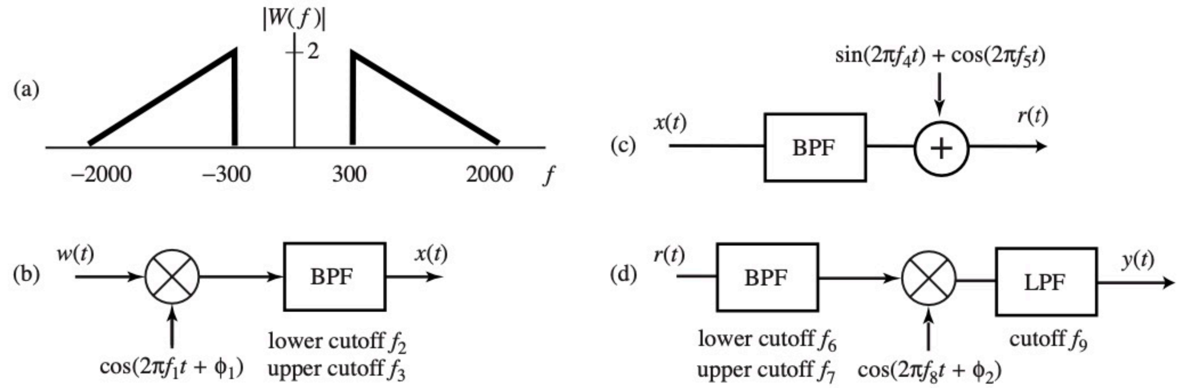


Figure 5.9 The transmission system for Exercise 5.12: (a) the magnitude spectrum of the message, (b) the transmitter, (c) the channel, and (d) the receiver.

Figure 4: Figure 5.9

Consider the transmission system of Figure 5.9. The message signal $w(t)$ has the magnitude spectrum shown in part (a). The transmitter in part (b) produces the transmitted signal x(t), which passes through the channel in part (c). The channel scales the signal and adds narrowband interferers to create the received signal r(t). The transmitter and channel parameters are $\phi_1 = 0.3$ radians, f1 = 24.1 kHz, f2 = 23.9 kHz, f3 = 27.5 kHz, f4 = 29.3 kHz, and f5 = 22.6 kHz. The receiver processing r(t) is shown in Figure 5.9(d). All bandpass and lowpass filters are considered ideal, with a gain of unity in the passband and zero in the stopband.

a. Sketch $|R(f)|$ for -30 kHz $\leq$ f $\leq$ 30 kHz. Clearly indicate the amplitudes and frequencies of key points in the sketch.

b. Assume that ϕ_2 is chosen to maximize the magnitude of $y(t)$ and reflects the value of ϕ_1 and the delays imposed by the two ideal bandpass filters that form the received signal $r(t)$. Select the receiver parameters f_6, f_7, f_8, and f_9, so the receiver output $y(t)$ is a scaled version of $w(t)$.

Solution

a. Sketch below

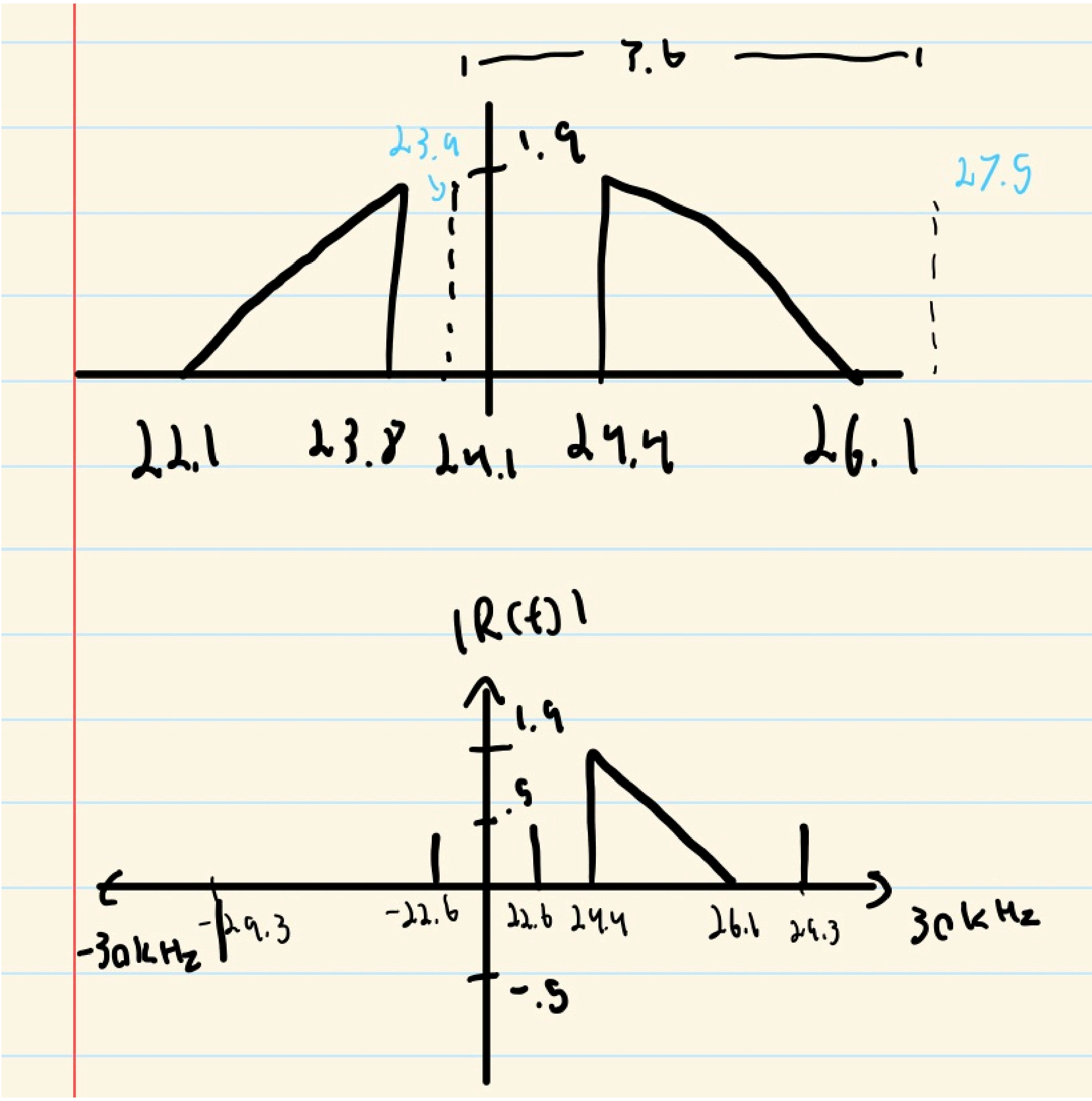


Figure 5: $|R(f)|$ for -30 kHz $\leq$ f $\leq$ 30 kHz

b. $f_6 = 24$kHz $f_7 = 27$kHz $f_8 = 0$kHz $f_9 = 30$kHz

Exercise 5.16

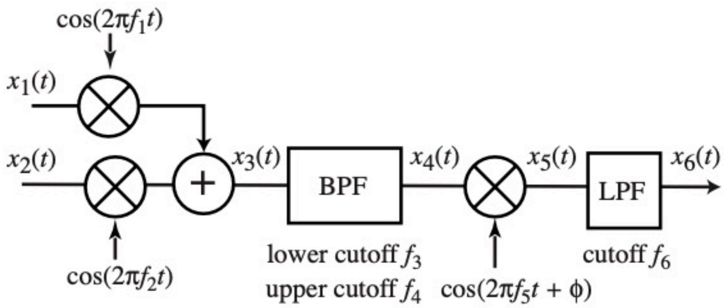


Figure 5.11 The transmission system of Exercise 5.16.

Figure 6: Figure 5.11

Consider the scheme shown in Figure 5.11. The absolute bandwidth of the baseband signal x_1 is 6 MHz and that of the baseband signal $x_2(t)$ is 4MHz, f_1 = 164MHz, f_2 = 154MHz, f_3 = 148MHz, f_4 = 160MHz, f_5 = 80MHz, $\phi = \pi/2$,and f_6 = 82MHz.

a. What is the absolute bandwidth of $x_3(t)$?

b. What is the absolute bandwidth of $x_5(t)$?

c. What is the absolute bandwidth of $x_6(t)$?

d. What is the maximum frequency in $x_3(t)$?

e. What is the maximum frequency in $x_5(t)$?

Solution

a. $x_3(t) = x_1(t)\cos(2\pi f_1 t) + x_2(t)\cos(2\pi f_2 t)$
min frequency = 150MHz and max frequency = 170MHz
absolute bandwidth = 20MHz

b. $x_5(t) = x_4(t)\cos(2\pi f_5 t + \phi)$
$x_4(t) = x_3(t)$ from f_3 to f_4 and 0 otherwise
min frequency = 150MHz and max frequency = 160MHz, abs bandwidth of $x_4(t)$=10MHz
absolute bandwidth of $x_5(t)$ = 2 x 10MHz = 20MHz

c. $x_6(t) = 0$ because $\phi = \pi/2$ means the output signal is attenuated by $\cos(\pi/2) = 0$
absolute bandwidth = 0

d. The max frequency of $x_3(t)$ is 170MHz

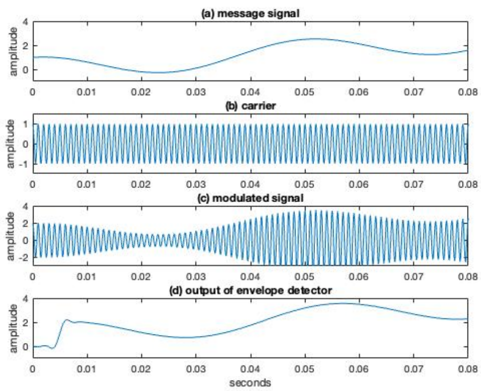
e. The max frequency of $x_5(t)$ is 102MHz

Question 5

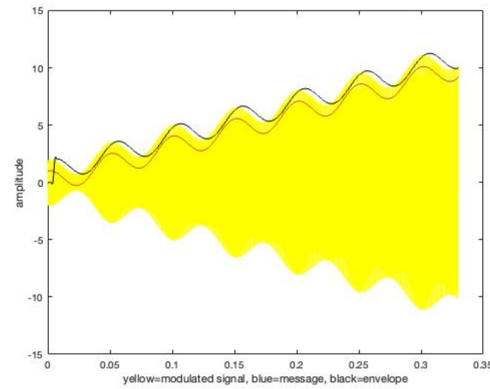
Consider the last line of AMlarge.m (see Listing 5.1):
$envv = (\pi/2) * filter(b, 1, abs(v));$
Why is the output of the filter multiplied by the constant $\pi/2$? Justify your answer.

Solution

The output of the filter is multiplied by the constant $\pi/2$ to scale the output so that it rides on the crest of the wave. Because the output of the LPF is the average of the absolute value of a cos wave, $\pi/2$, the output must be scaled by this factor.
$envv = (\pi/2) * filter(b, 1, abs(v))$

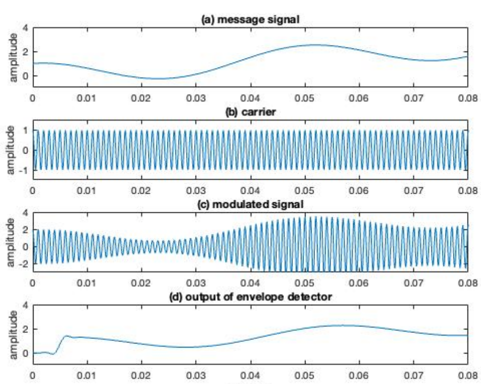


(a) amplitude of signals

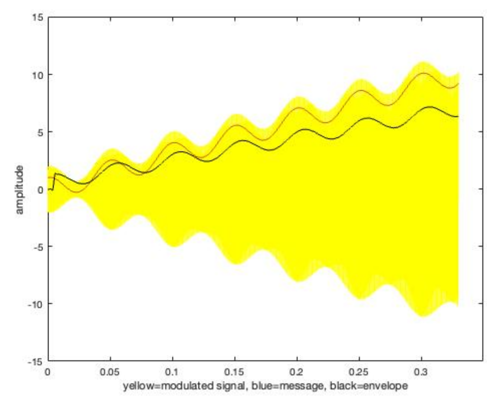


(b) overlay

This becomes more clear when comparing the results to a non-multiplied filter output. $envv = filter(b, 1, abs(v))$



(a) amplitude of signals



(b) overlay

In these plots you can see the envelope (in black) drift below the message (in blue) such that it does not ride the crest of the modulated signal.

Question 6

The attached qam_hw.mat file is a QAM passband signal v that is the sum of two modulated messages w_1 and w_2. These messages were respectively modulated using cosine and sine functions, with carrier signal f = 1000 Hz. The sampling period is T = 1/10000s and the total duration of the signal is 0.3s. Note that these parameters are all the same as in the file AM.m (listing 5.2 in the textbook).
Plot the following:

(a) The modulated signal v.

(b) The demodulated signals (before the LPF) x_1 and x_2.

(c) The recovered signals (after the LPF) m_1 and m_2.

Hints:

(a) To load the QAM signal v, use the command: load($'qam\_hw.mat','v'$);

(b) Use the same LPF parameters as in AM.m.

(c) The x-axis should be [0, 0.3] for all the plots.

(d) For the signal v, the y-axis should be [11,11].

(e) For the signal x_1 and m_1, the y-axis should be [5,10].

(f) For the signals x_2 and m_2, the y-axis should be [10,1].

Solution

Using this code:

Listing 3: MATLAB code for Exercise 4.21a

```
load('qam_hw(1).mat','v')

time=0.3; Ts=1/10000;                    % sampling interval & time
t=Ts:Ts:time; lent=length(t);            % define a time vector

fc=1000;
gam=0; phi=0;                            % freq & phase offset
c1=cos(2*pi*(fc+gam)*t+phi);             % create cosine for demod
c2=sin(2*pi*(fc+gam)*t+phi);             % create sine for demod

x1=v.*c1;                                 % demod received signal
x2=v.*c2;                                 % demod received signal

fbe=[0 0.1 0.2 1]; damps=[1 1 0 0];      % LPF design
fl=100; b=firpm(fl,fbe,damps);           % impulse response of LPF
m1=2*filter(b,1,x1);                     % LPF the demodulated signal
m2=2*filter(b,1,x2);                     % LPF the demodulated signal

% used to plot figure
figure(1)
axis([0,0.3, 11,11.1])
ylabel('amplitude');
title('message after modulation');
subplot(1,1,1), plot(t,v)
```

```
figure(2)
subplot(2,1,1), plot(t,x1)
axis([0,0.3, -1,10])
ylabel('amplitude');
title('demodulated signal x1');
subplot(2,1,2), plot(t,m1)
axis([0,0.3, -1,10])
ylabel('amplitude'); title('recovered message is a LPF applied to x1');

figure(3)
subplot(2,1,1), plot(t,x2)
axis([0,0.3, -10,1])
ylabel('amplitude');
title('demodulated signal x2');
subplot(2,1,2), plot(t,m2)
axis([0,0.3, -10,1])
ylabel('amplitude'); title('recovered message is a LPF applied to x2');
```

I was able to obtain these graphs

(a) graph below

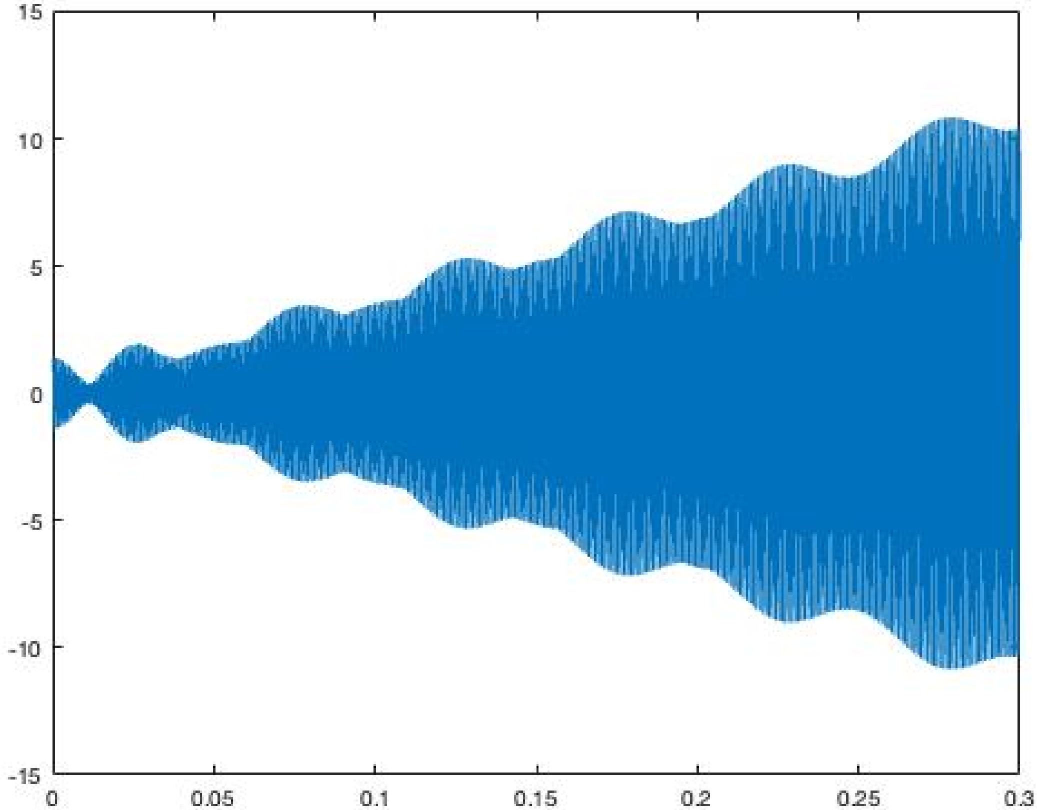


Figure 9: the modulated signal v

(b) graph below

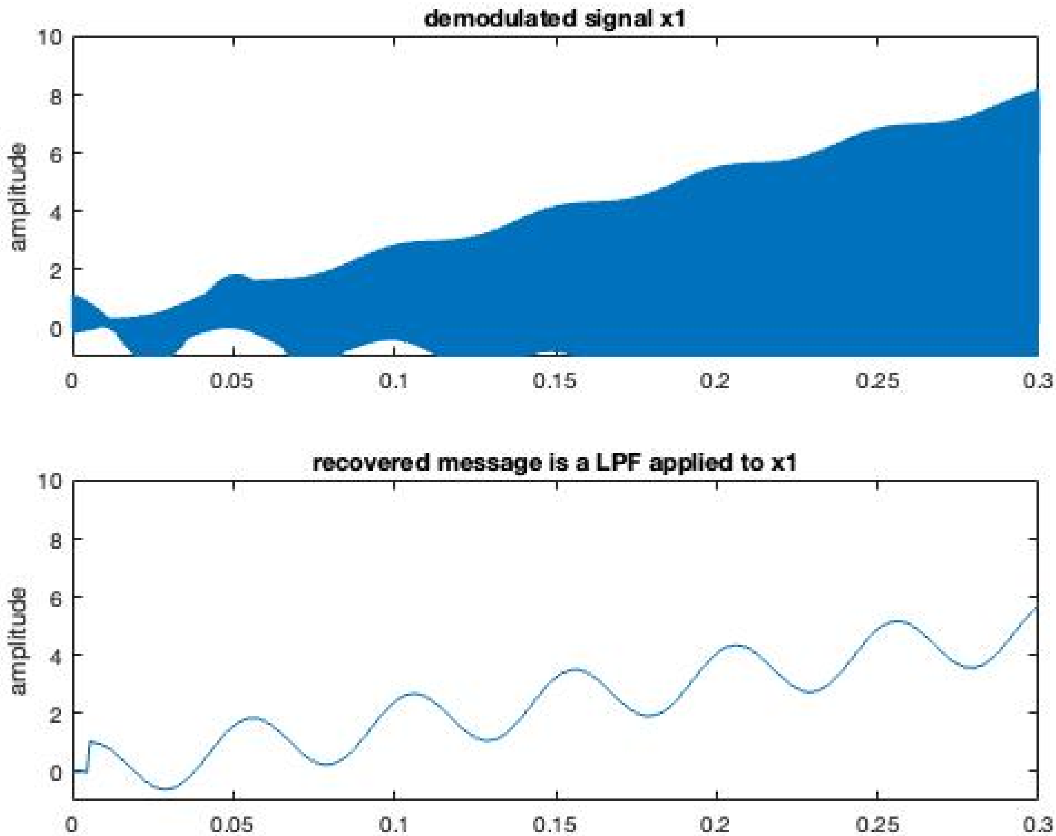


Figure 10: the demodulated signals (before the LPF) x_1 and m_1

(c) graph below

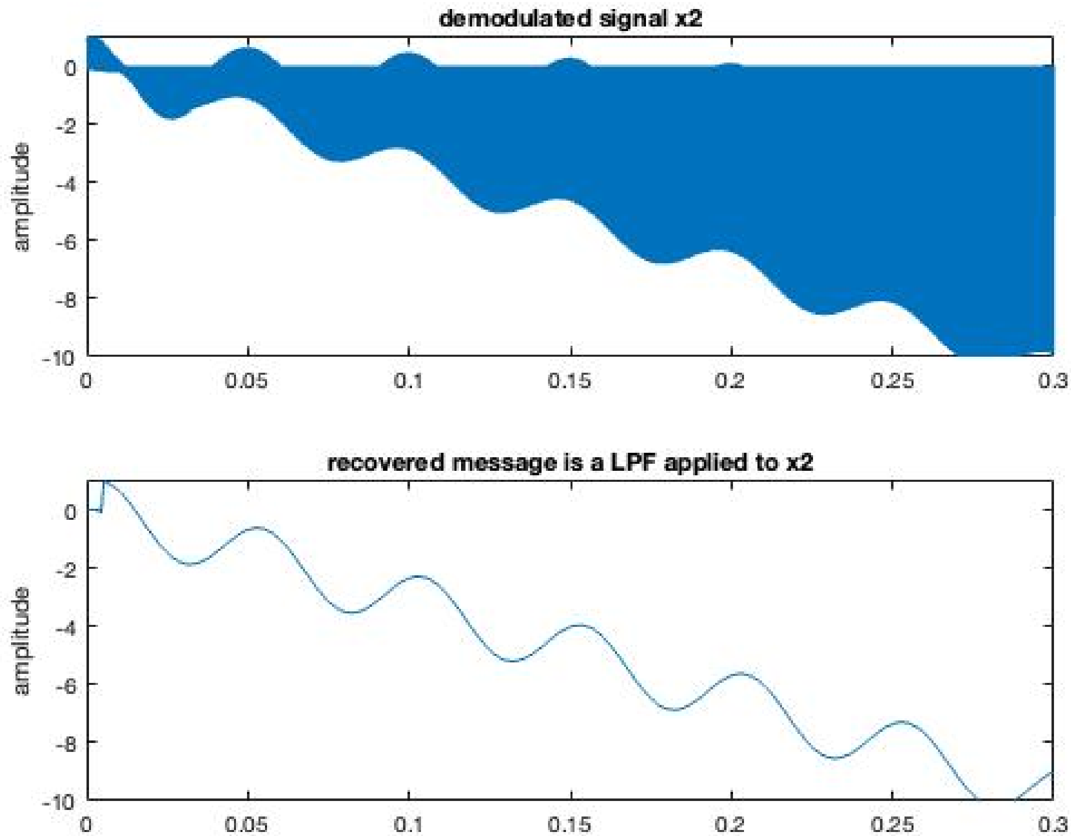


Figure 11: the recovered signals (after the LPF) x_2 and m_2