Communication Project

Amplitude Modulation

Name	Sec	BN
Norhan Reda Abdelwahed Ahmed	2	32
Hoda Gamal Hamouda Ismail	2	34

Code and steps explanation:

- Read the signal and get its features (length, bandwidth, time range, frequency range, fft)

- Resampling the signals to a new fs (fs = 650,000 Hz)
  to satisfy the condition (fs >= 2 BW)
  and we have these bandwidths (BW1=1.5319*10^5, BW2=1.5291*10^5, BW3=1.4537*10^5)

- Modulating the signals by multiplying each signal by corresponding carrier:
  Carrier1 = cos (2*pi*fc1*t1) where fc1 = 160,000 Hz
  Carrier2 = cos (2*pi*fc2*t2) where fc2 = 320,000 Hz
  Carrier3 = sin (2*pi*fc2*t3) where fc2 = 320,000 Hz
  Choosing the carriers should satisfy Nyquist condition that:
  the condition (fc < fs / 2)
  The signals are far enough not to overlap with each other

- Make the signals have the same length which is the max length among their lengths.

- Sum the signals to get the required signal s(t).

- Perform synchronous demodulation to s(t) to restore the 3 signals
  Demodulation steps:
  For each carrier from the 3 carriers:
  Multiply s(t) by the carrier.
  Use a low pass filter to cancel undesired frequencies.
  Multiply by 2 to get the original amplitude.

- Perform demodulation with phase shift 10, 30, 90 degrees.
  Carrier1 = cos (2*pi*fc1*t1 + (phase_shift_angle* pi) / 180) where fc1 = 160,000 Hz
  Carrier2 = cos (2*pi*fc2*t2 + (phase_shift_angle* pi) / 180) where fc2 = 320,000 Hz
  Carrier3 = sin (2*pi*fc2*t3 + (phase_shift_angle* pi) / 180) where fc2 = 320,000 Hz

- Perform demodulation with a local carrier frequency that is different by 2 Hz and 10 Hz from its carrier frequency.
  Carrier1 = cos (2*pi*(fc1+2)*t1) where fc1 = 160,000 Hz
  Carrier1 = cos (2*pi*(fc1+10)*t1) where fc1 = 160,000 Hz

Results explanation:

In synchronous demodulation:

The 3 signals are almost returned to the original ones.

In demodulation with a phase shift:

With increasing the phase shift, the attenuation is increased, and the interference between signals with same fc and different carriers function cos & sin is also increased.

In phase shift 10:

Signal 1: there is very little attenuation in it (the sound is slightly lower).

Signal 2 & 3: there is very little interference between them, but the signal with the current corresponding carrier is the one with the higher sound.

In phase shift 30:

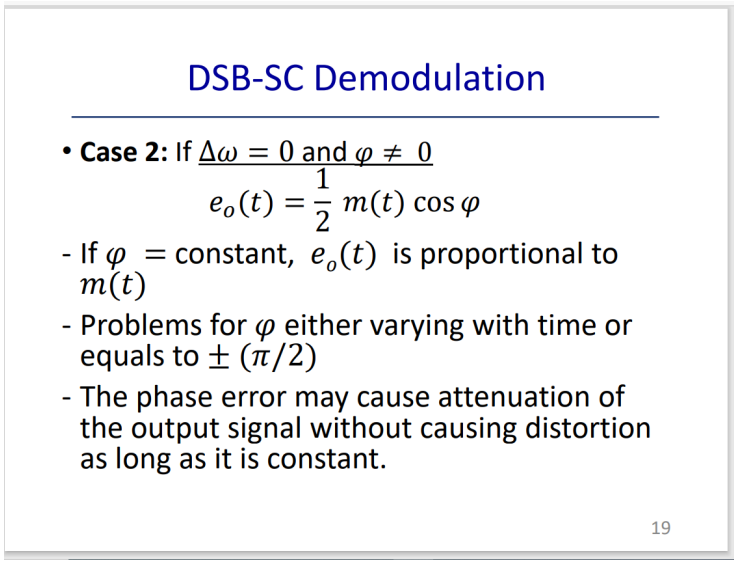
Signal 1: there is more attenuation in it (the sound is lower than the previous case).

Signal 2 & 3: there is more interference between them, and the signal with the current corresponding carrier is still a little bit with higher sound.

In phase shift 90:

Signal 1: it is completely attenuated (there is no sound).

Signal 2 & 3: there is complete interference between them, the signals are interchanged with each other (when we try to get signal 2, we get signal 3 and vice versa).



QAM

- QAM demodulation must be totally synchronous
- If there is a phase error of θ:

$$x_1(t) = [m_1(t)\cos\omega_c t + m_2(t)\sin\omega_c t]\cos(\omega_c t + \theta)$$

$$= \frac{1}{2}[m_1(t)\cos\theta - m_2(t)\sin\theta + m_1(t)\cos(2\omega_c t + \theta) + m_2(t)\sin(2\omega_c t + \theta)]$$

$$x_3(t) = \frac{1}{2}[m_1(t)\cos\theta - m_2(t)\sin\theta]$$

- Error in phase of carrier at the demodulator results in:
  - attenuation
  - interference between the two signals (cochannel interference)

9

In demodulation with different local carrier:

Different by 2Hz:

There is little attenuation and distortion in the output signal.

Different by 10Hz:

There is more attenuation and distortion in the output signal than the previous case.

DSB-SC Demodulation

- **Case 3:** If $\Delta\omega \neq 0,\ \varphi = 0$

$$e_o(t) = \frac{1}{2}\, m(t)\cos\Delta\omega\, t$$

- The output is multiplied by a low frequency sinusoid, this causes attenuation and distortion of the output signal.

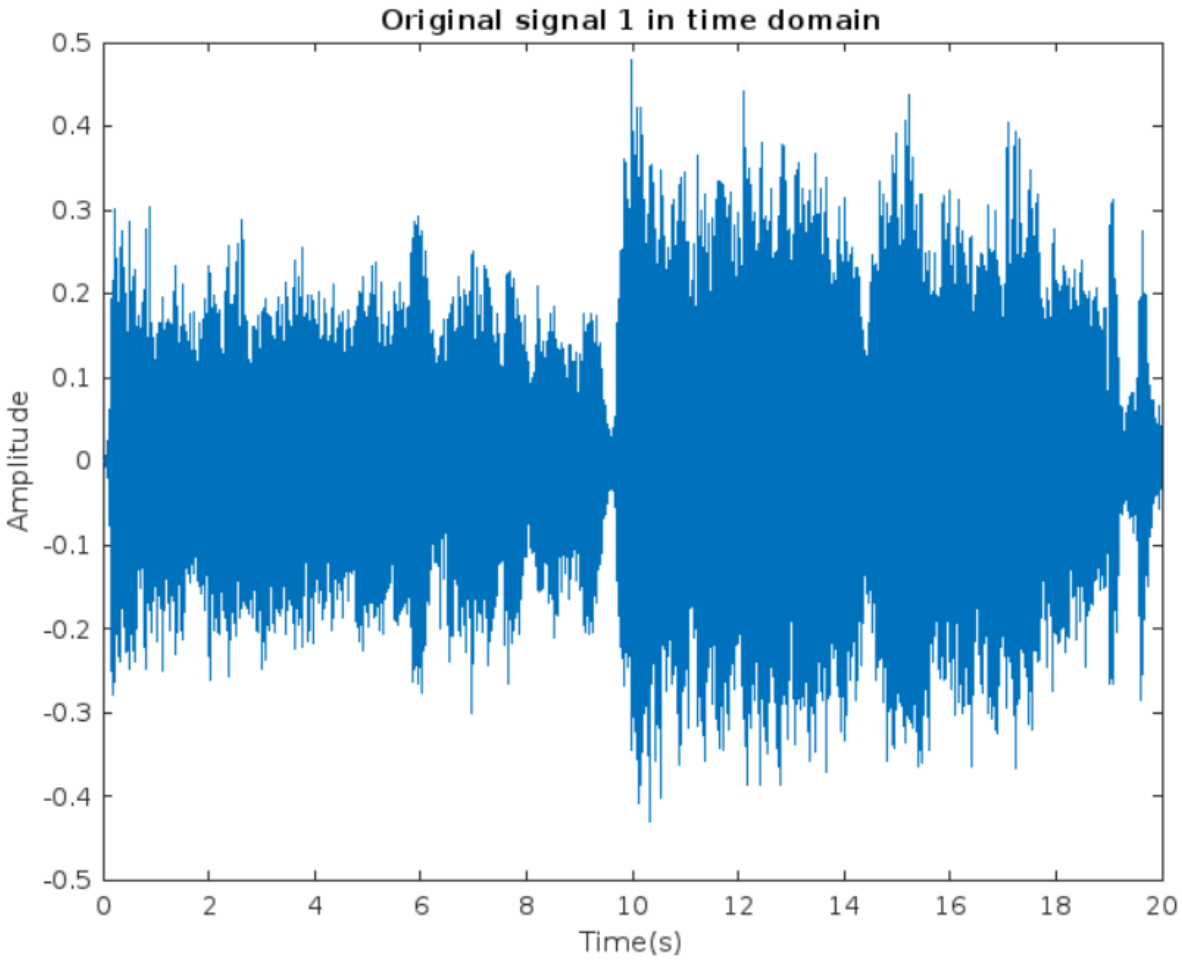
- In a following lecture, we will study methods to synchronize the local carrier with the incoming carrier in the received signal.

20

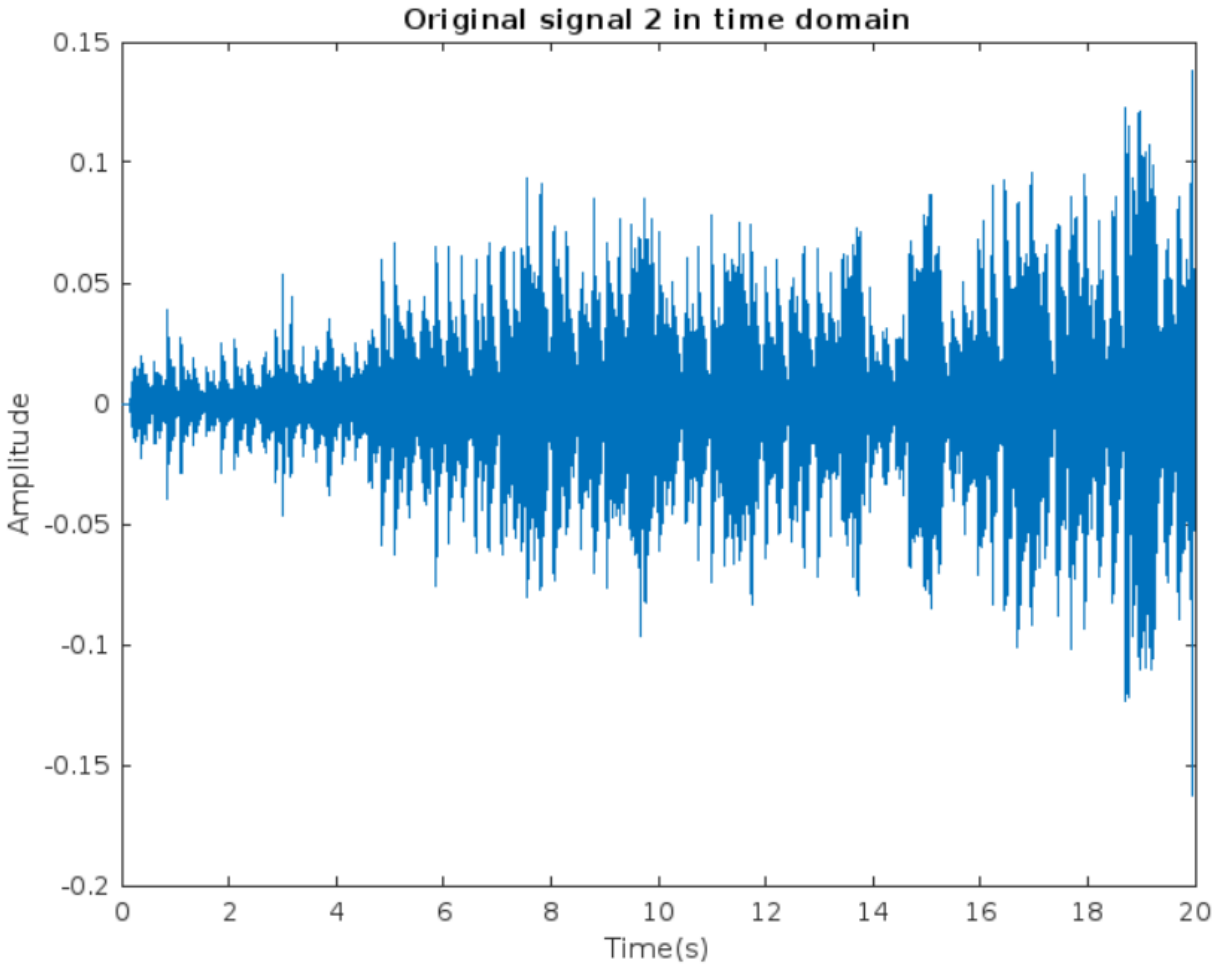
Figures

Original signals in time Domain

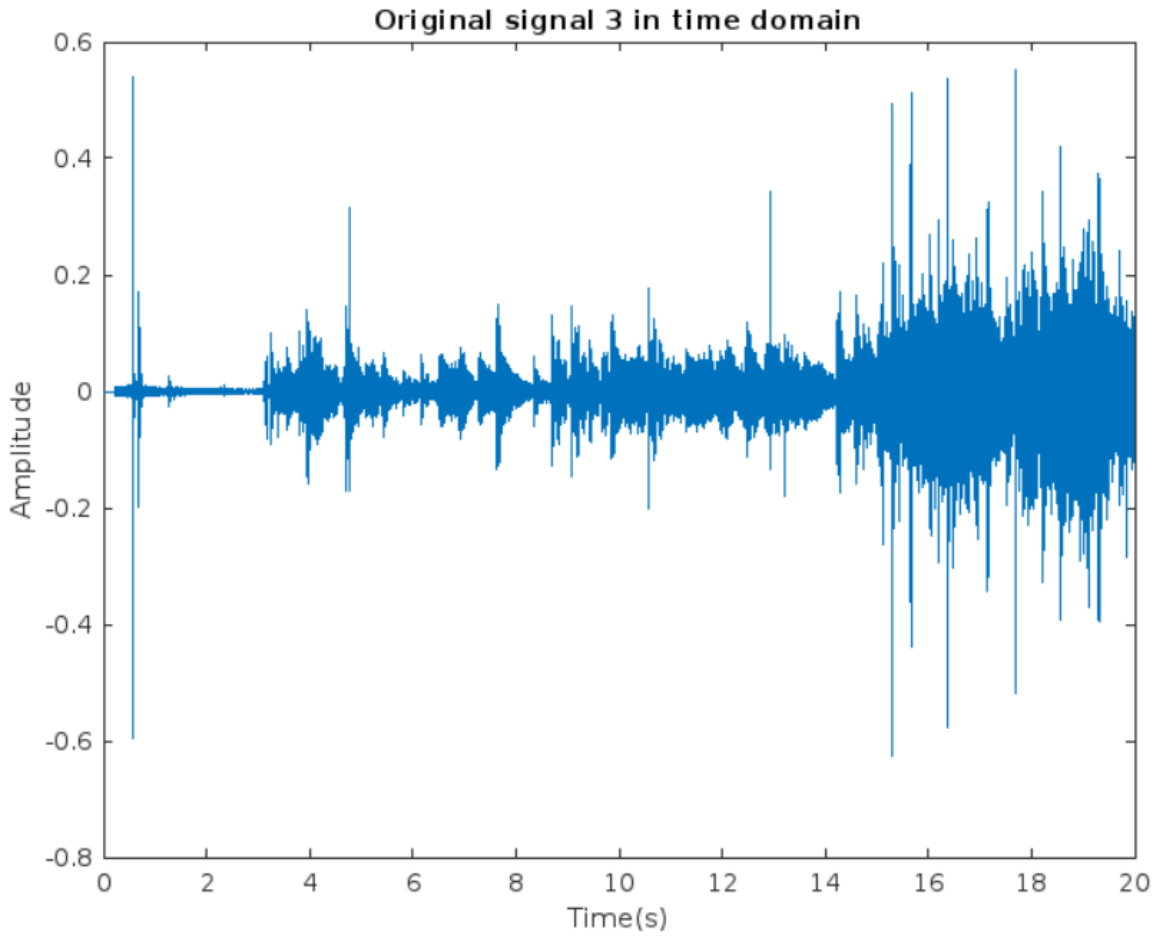
Signal 1



Signal 2

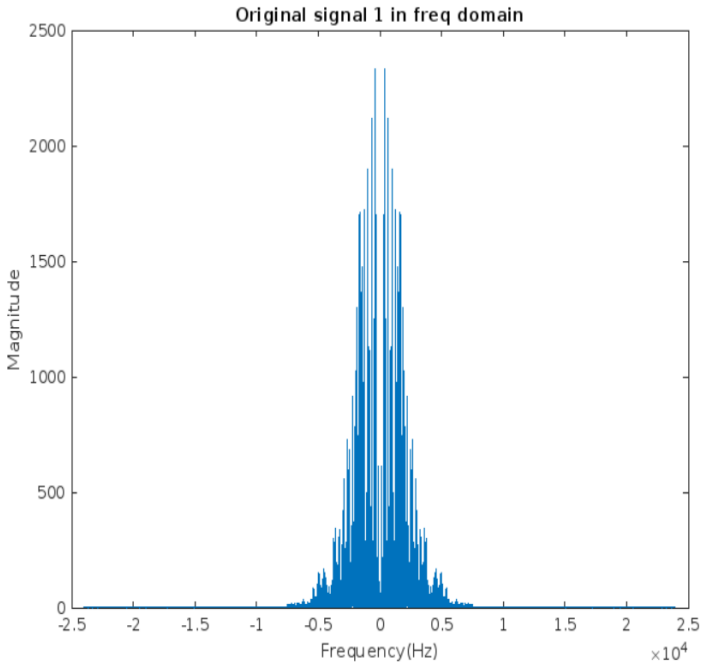

Original signal 2 in time domain
Amplitude
Time(s)
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
2
4
6
8
10
12
14
16
18
20

Signal 3

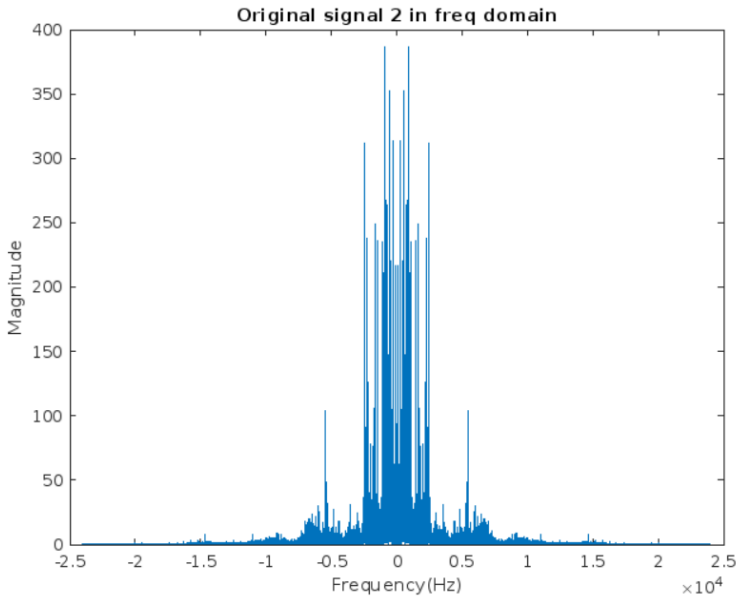

Original signal 3 in time domain
Amplitude
Time(s)
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
2
4
6
8
10
12
14
16
18
20

Original signals in frequency Domain

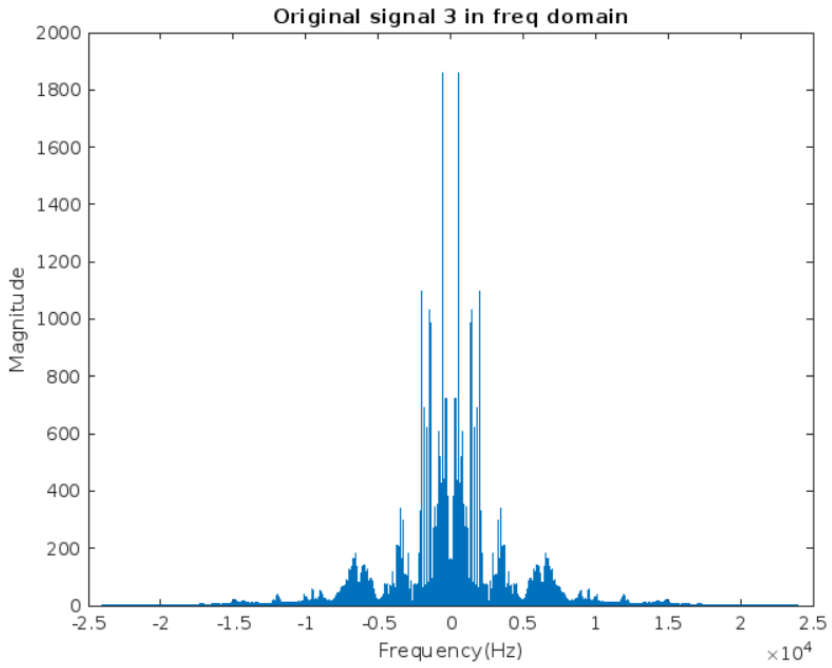
Signal1



Signal 2

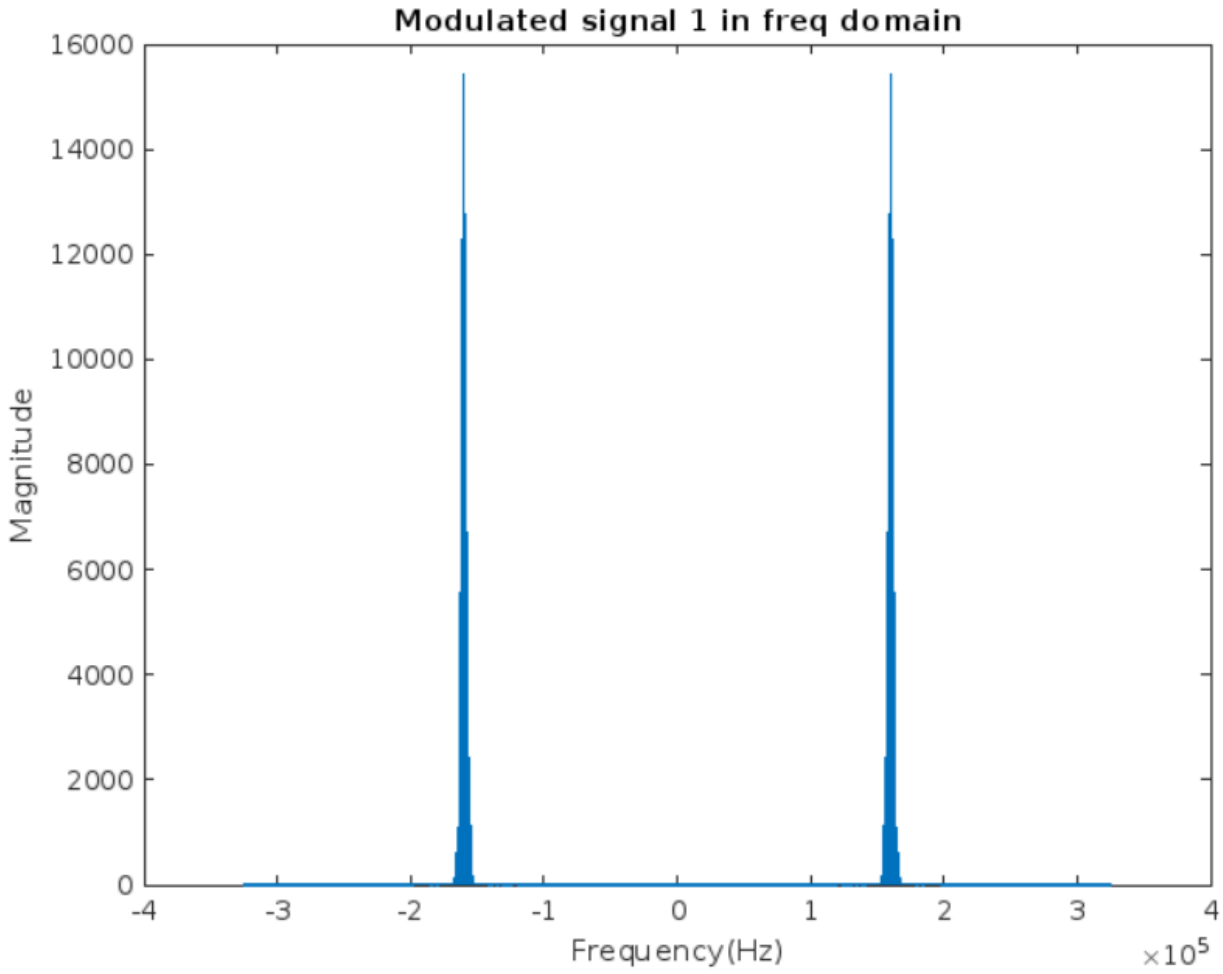


Signal 3

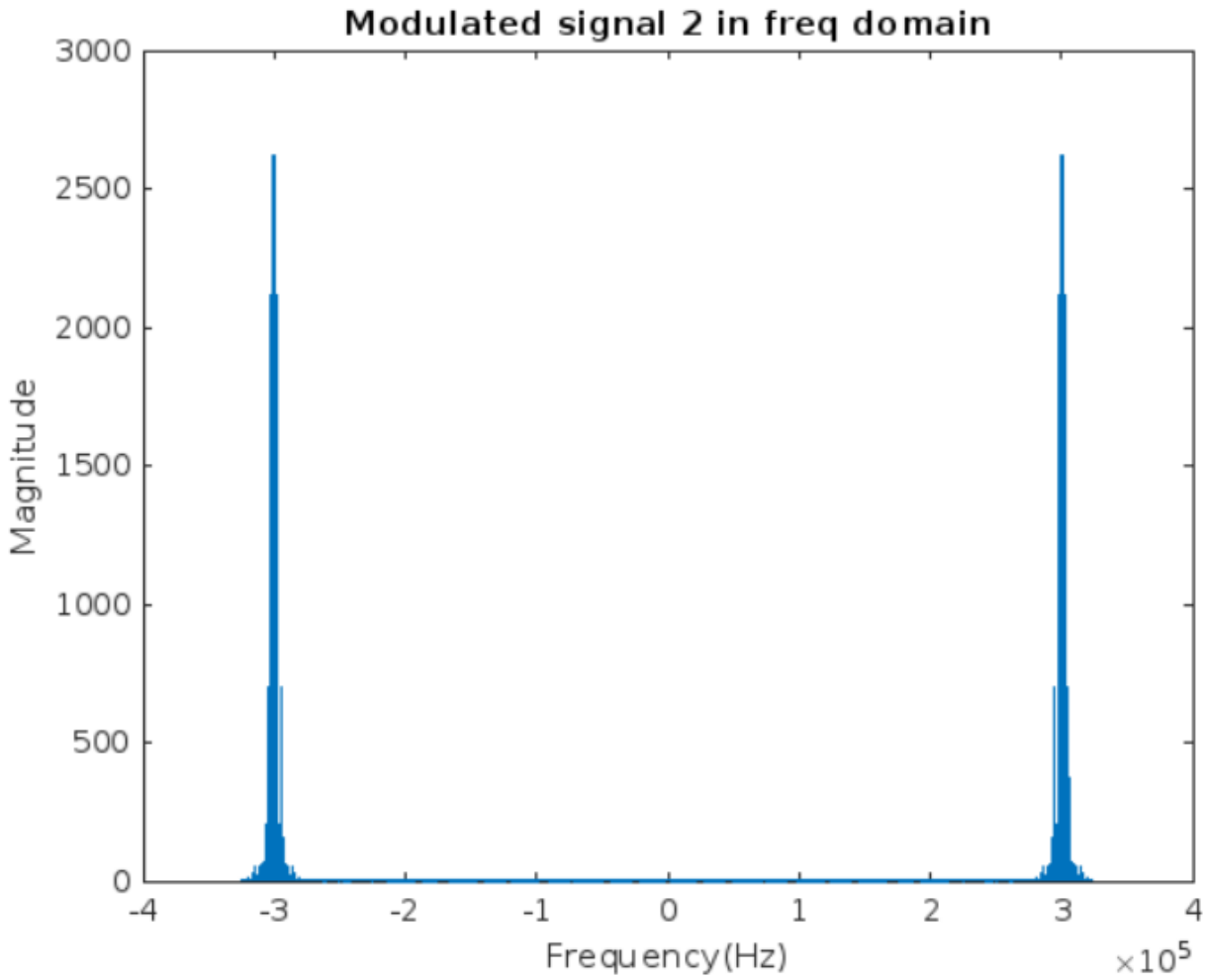

Original signal 3 in freq domain
Magnitude
Frequency(Hz)
×10^4
2000
1800
1600
1400
1200
1000
800
600
400
200
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

Modulated signals in frequency Domain

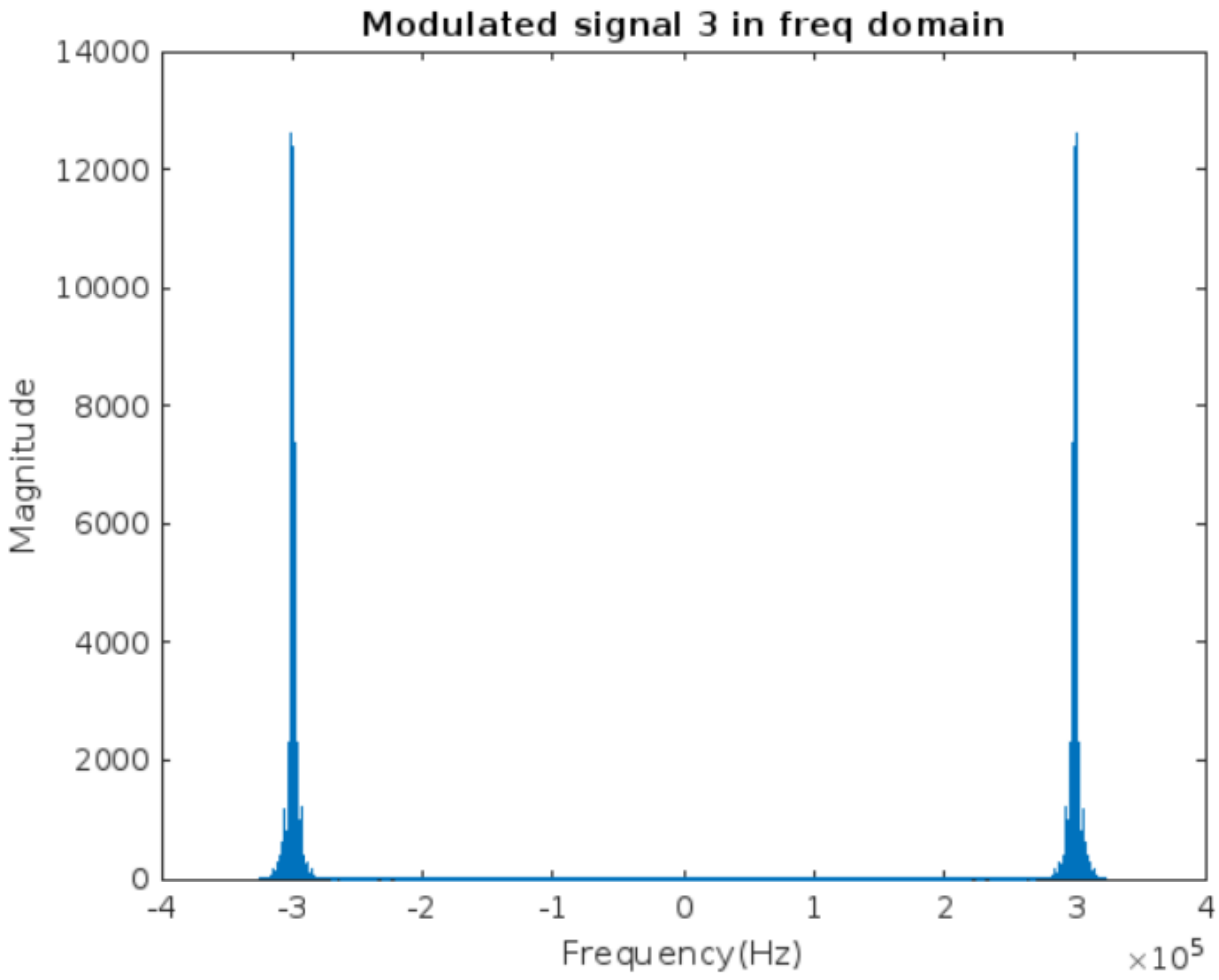
signal1



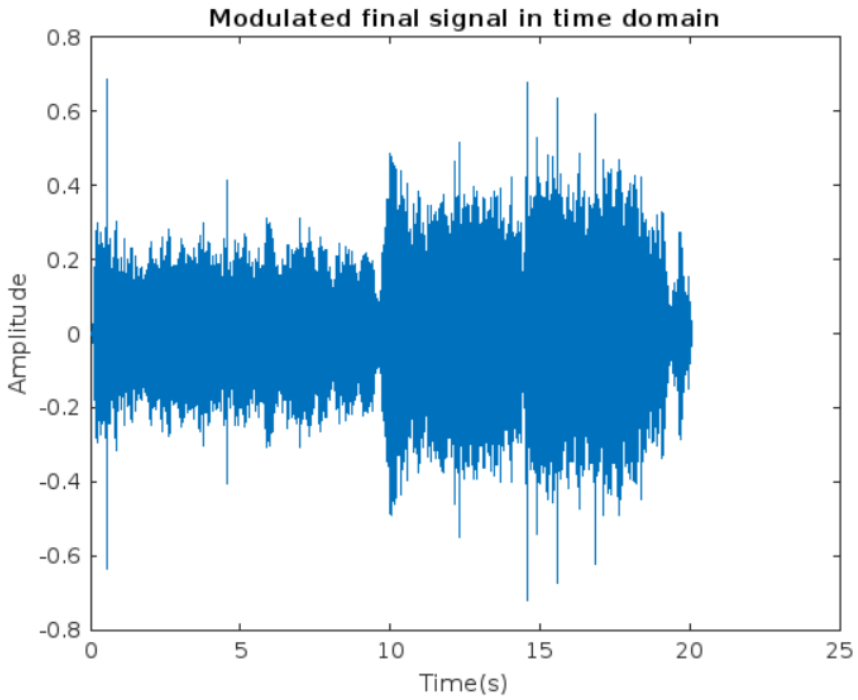
Signal 2


Modulated signal 2 in freq domain
3000
2500
2000
1500
1000
500
0
Magnitude
-4
-3
-2
-1
0
1
2
3
4
Frequency(Hz)
×10^5

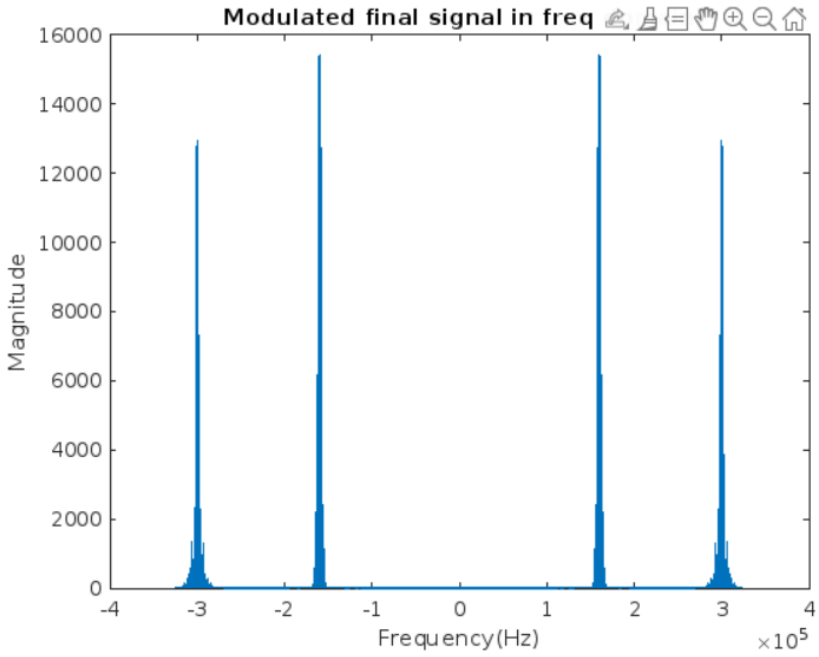
Signal 3


Modulated signal 3 in freq domain
14000
12000
10000
8000
6000
4000
2000
0
Magnitude
-4
-3
-2
-1
0
1
2
3
4
Frequency(Hz)
×10^5

Modulated final signal in time Domain

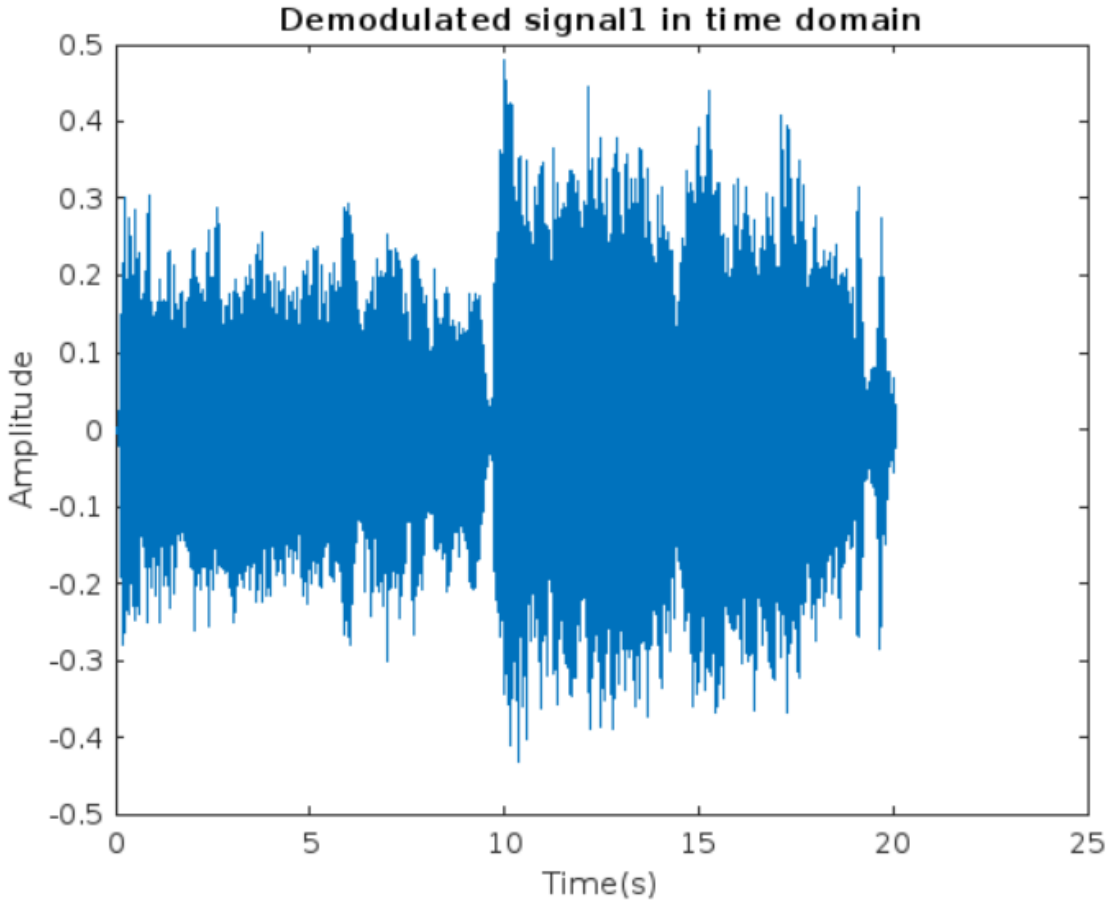


Modulated final signal in frequency Domain

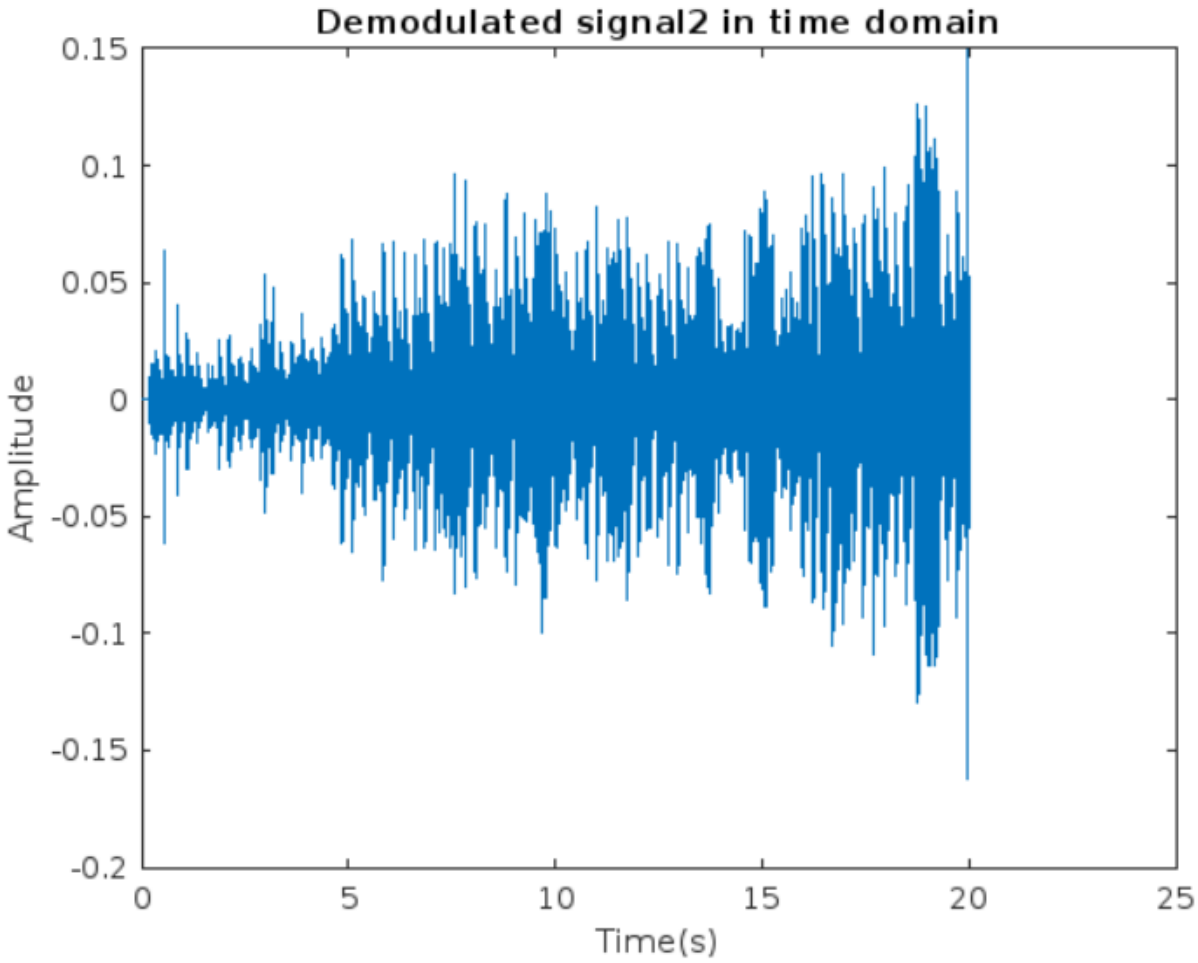

Modulated final signal in freq
16000
14000
12000
10000
8000
6000
4000
2000
0
Magnitude
-4
-3
-2
-1
0
1
2
3
4
Frequency(Hz)
×10^5

Demodulated signals in time Domain

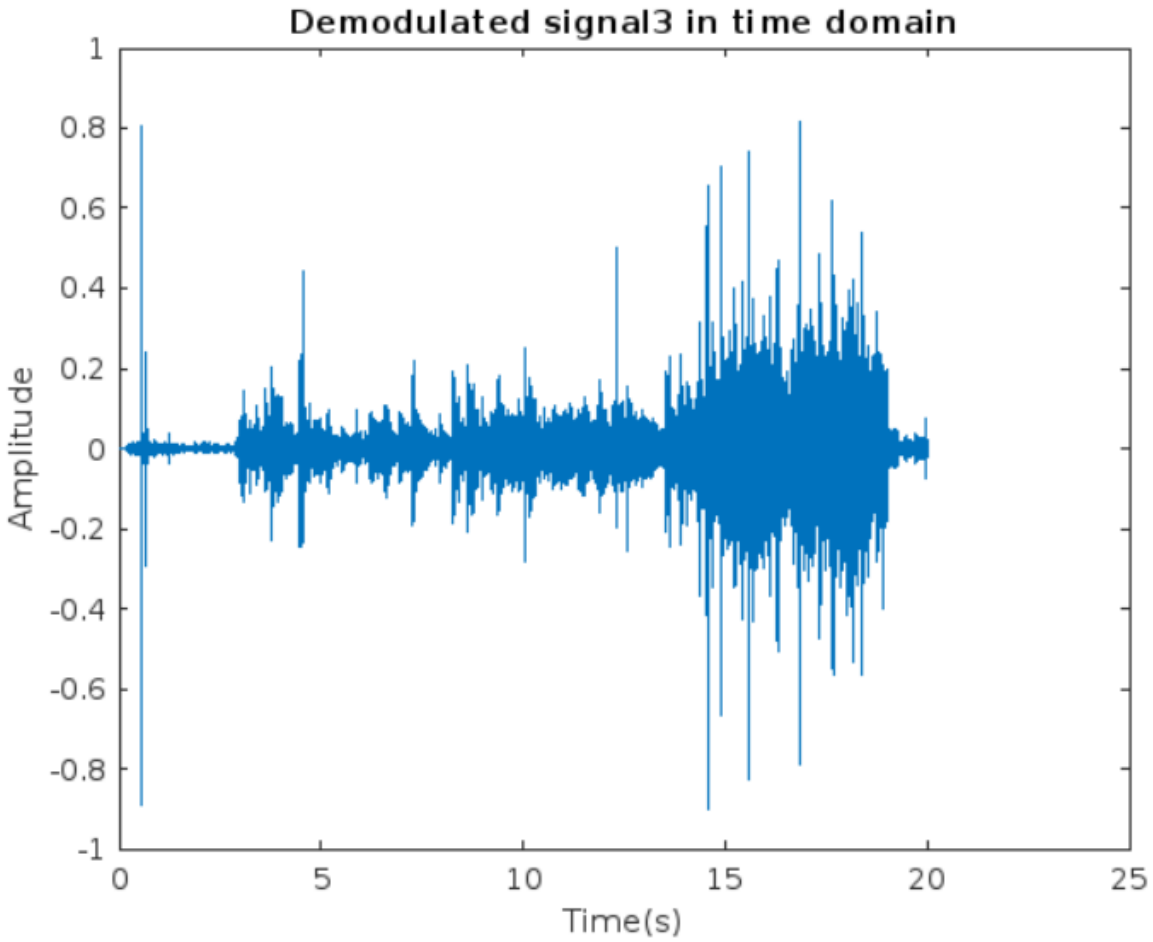
Signal 1



Signal 2

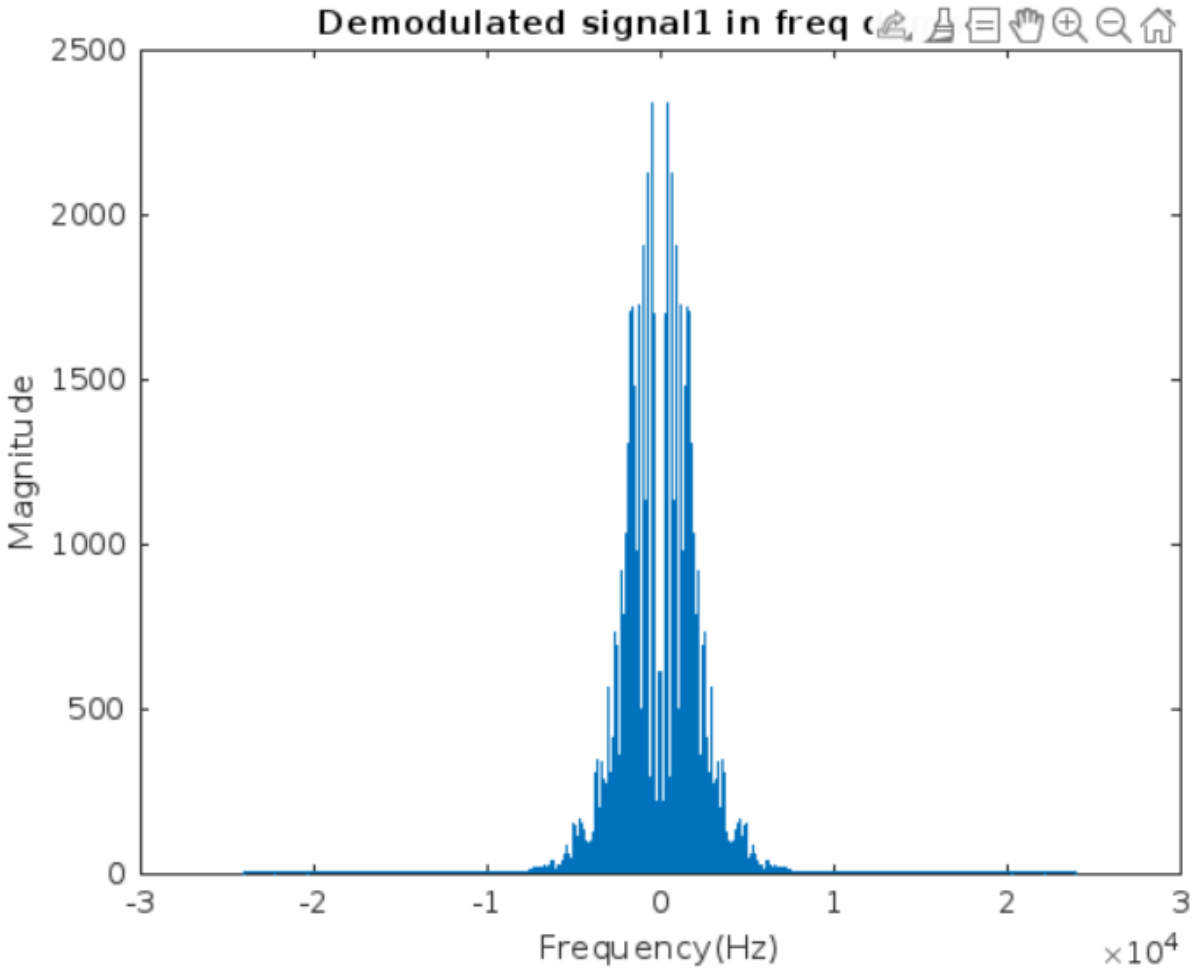

Demodulated signal2 in time domain
Amplitude
Time(s)
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
5
10
15
20
25

Signal 3

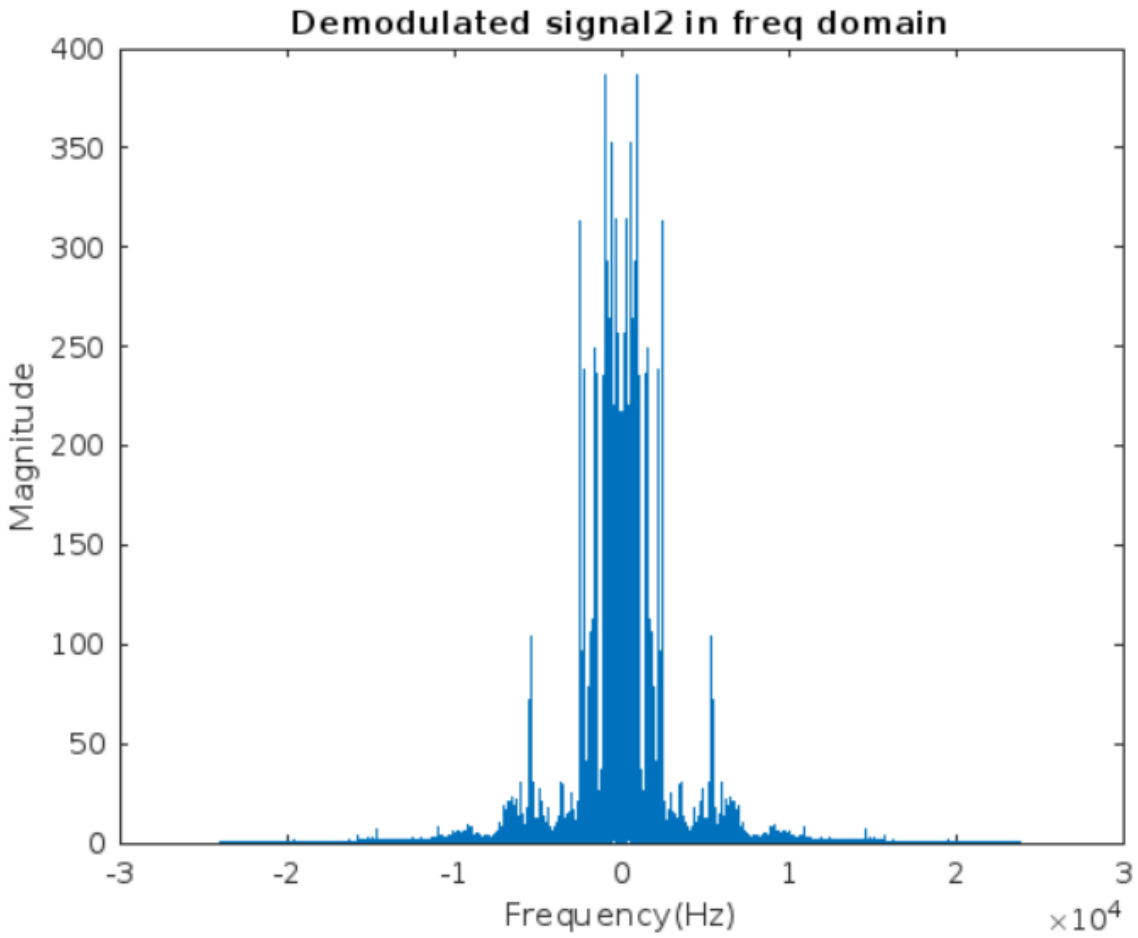

Demodulated signal3 in time domain
Amplitude
Time(s)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
20
25

Demodulated signals in frequency Domain

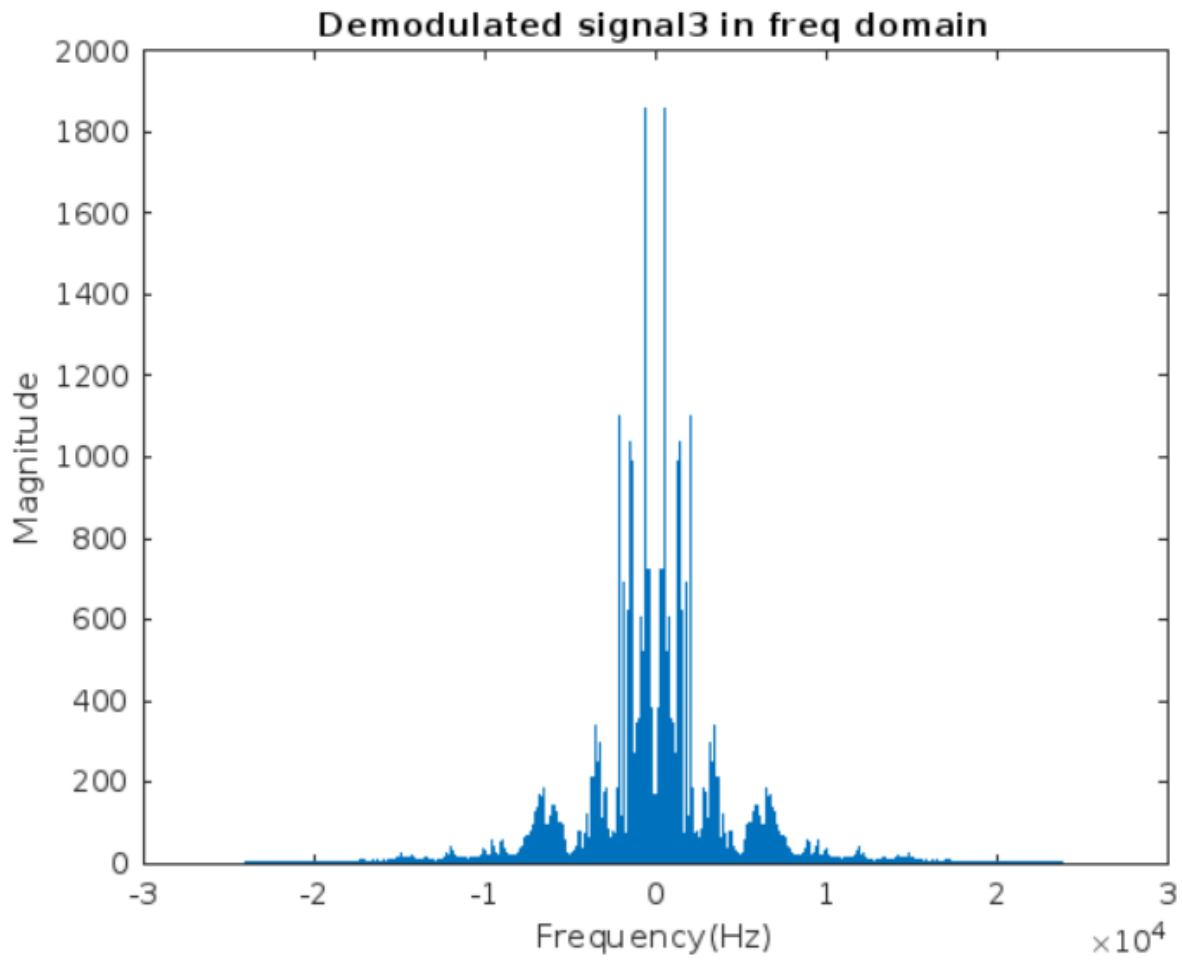
Signal1



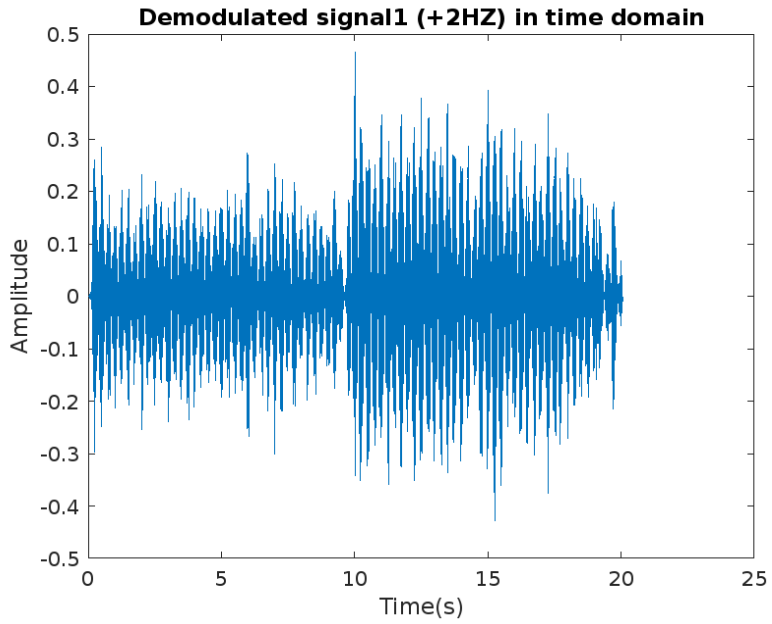
Signal2


Demodulated signal2 in freq domain
Magnitude
400
350
300
250
200
150
100
50
0
-3
-2
-1
0
1
2
3
Frequency(Hz)
×10^4

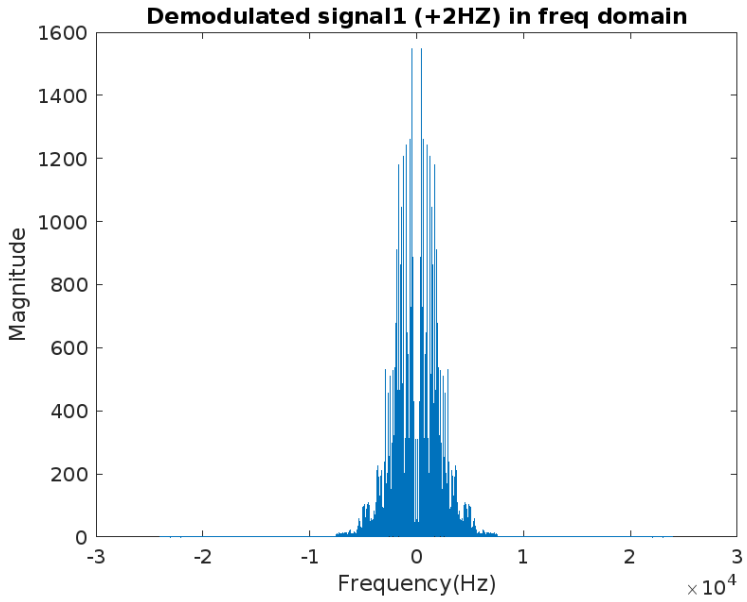
Signal3


Demodulated signal3 in freq domain
Magnitude
2000
1800
1600
1400
1200
1000
800
600
400
200
0
-3
-2
-1
0
1
2
3
Frequency(Hz)
×10^4

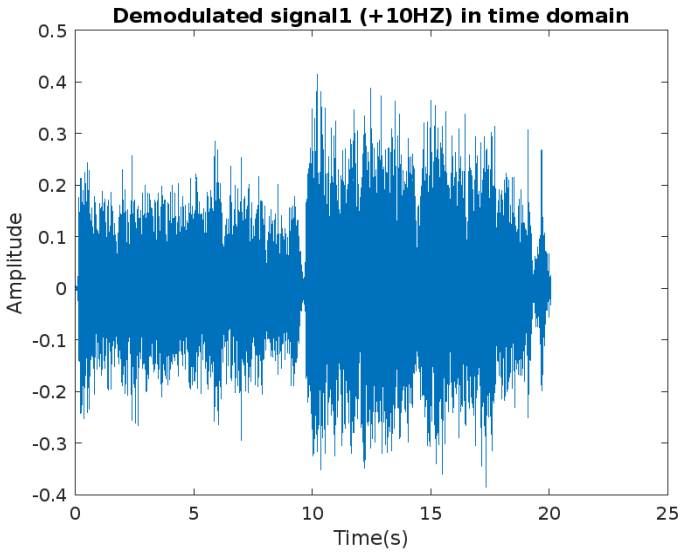
Demodulated signal with local carrier frequency that is different by 2 Hz in time Domain



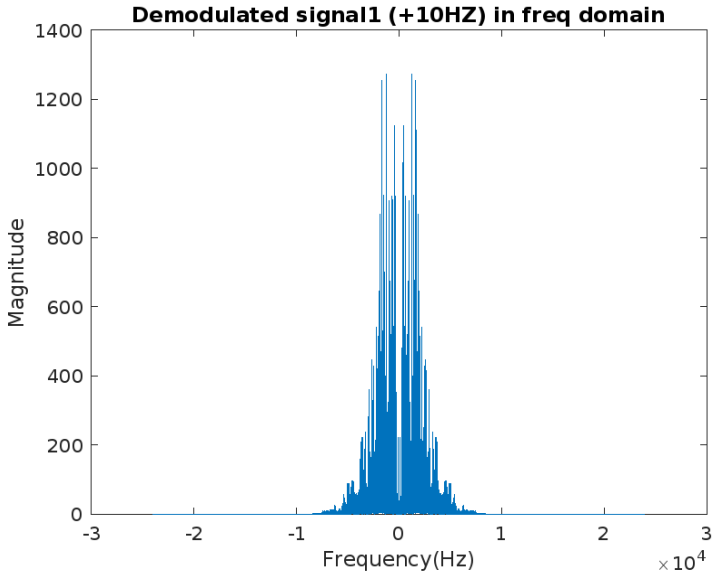
Demodulated signal with local carrier frequency that is different by 2 Hz in frequency Domain



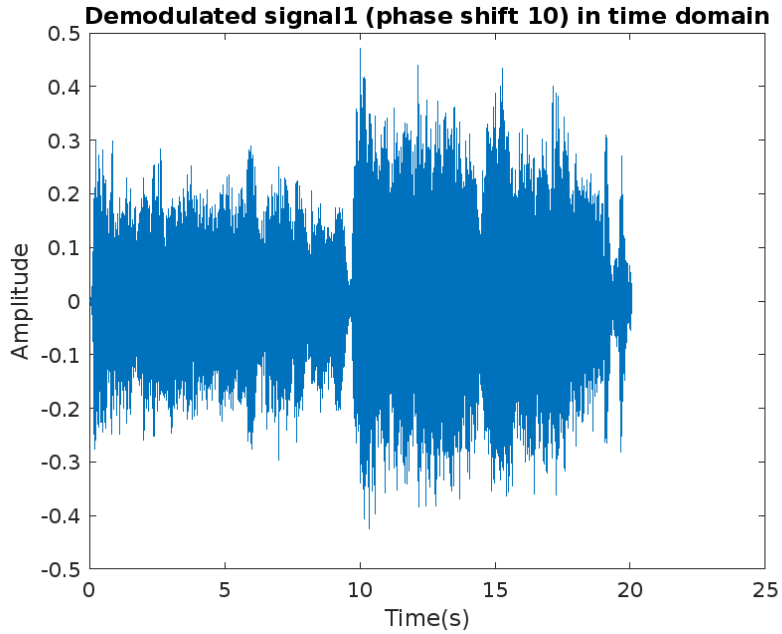
Demodulated signal with local carrier frequency that is different by 10 Hz in time Domain

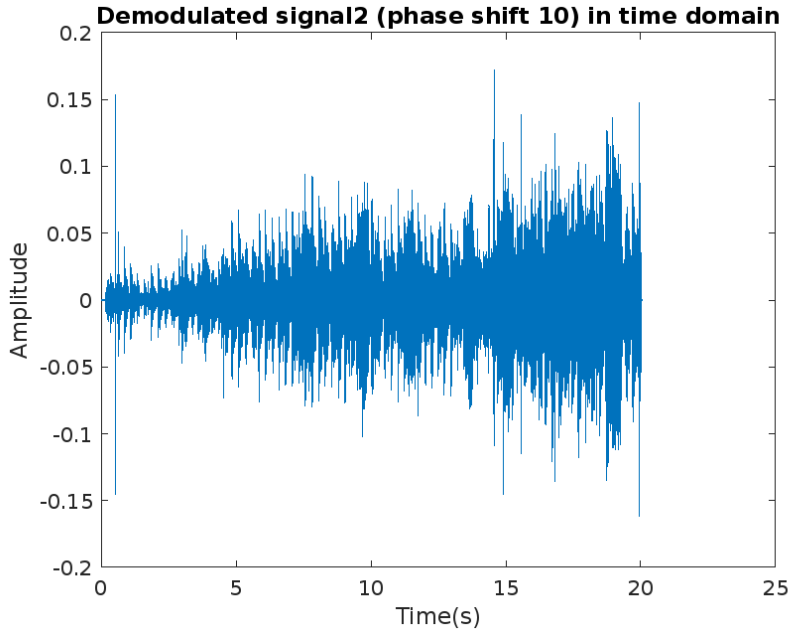


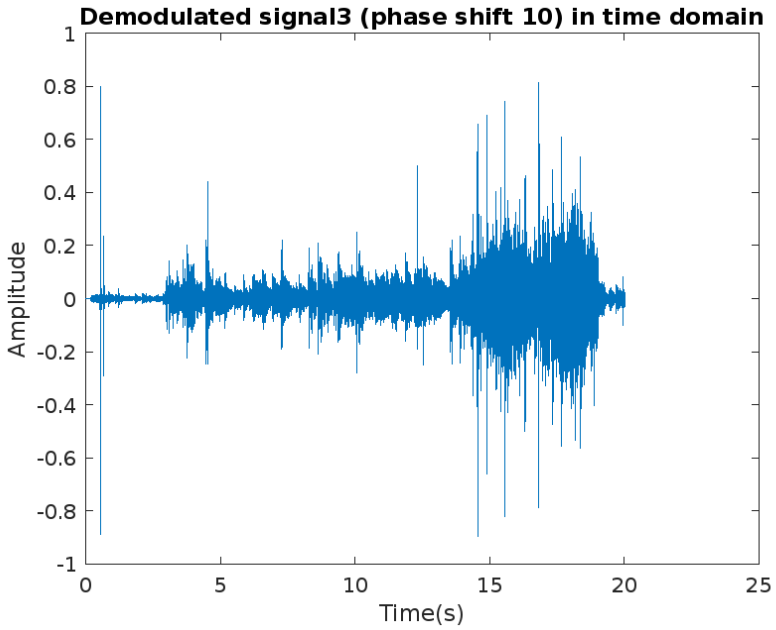
Demodulated signal with local carrier frequency that is different by 10 Hz in frequency Domain



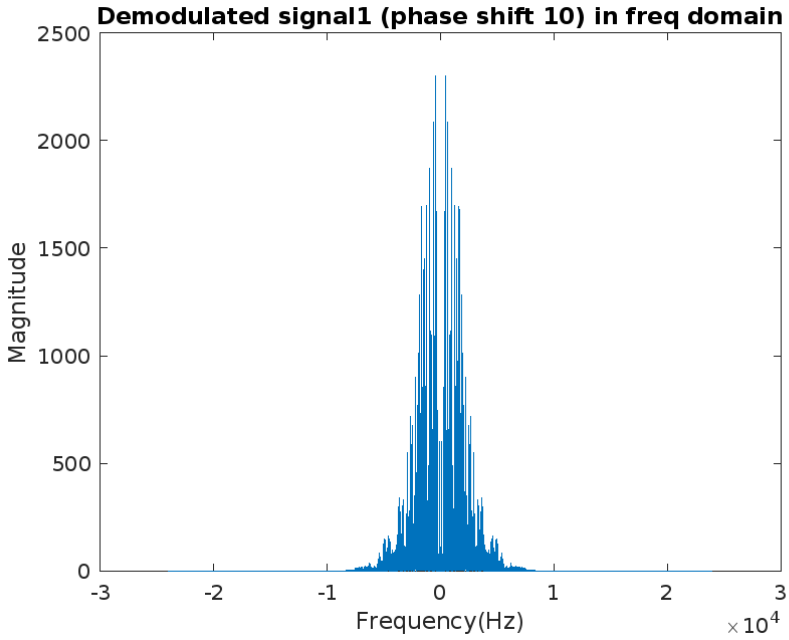
Demodulated signal with phase shift 10 in time Domain

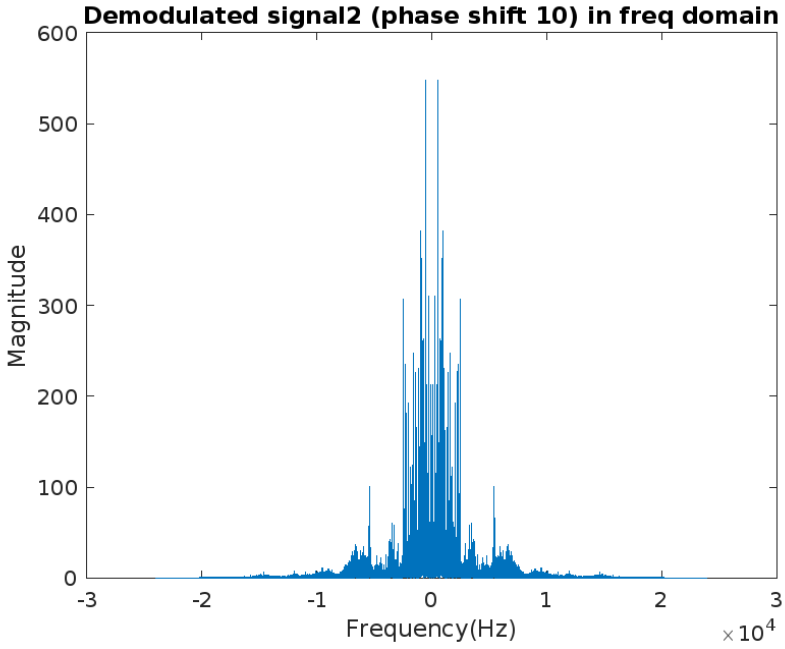


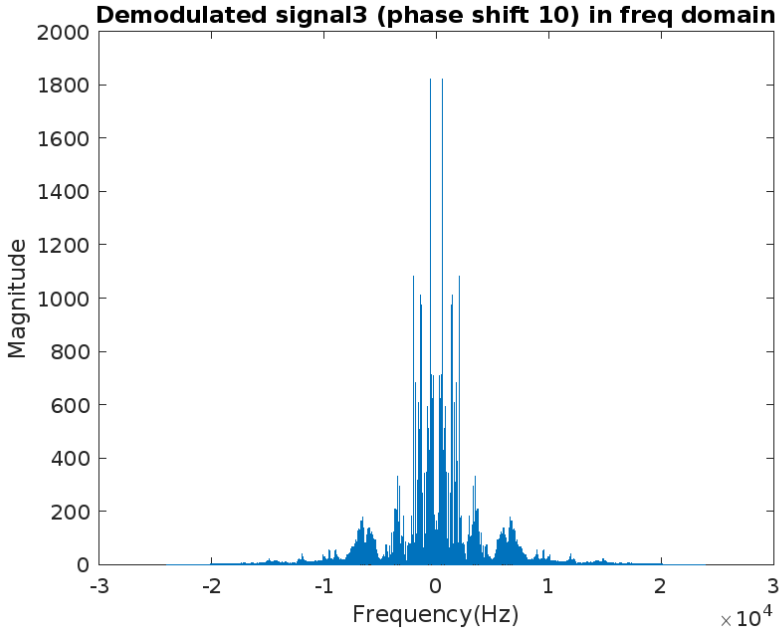




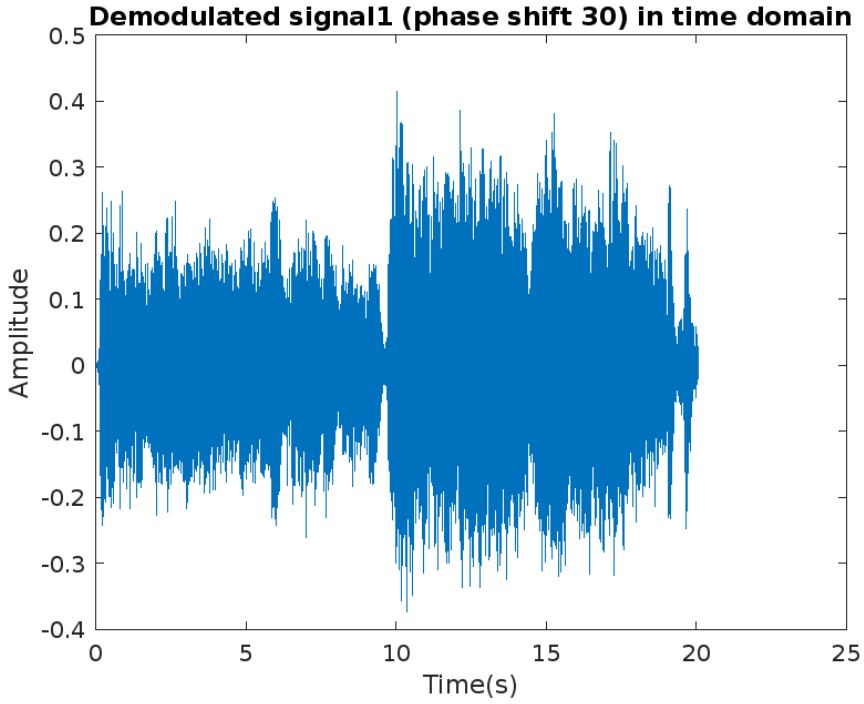
Demodulated signal with phase shift 10 in frequency Domain

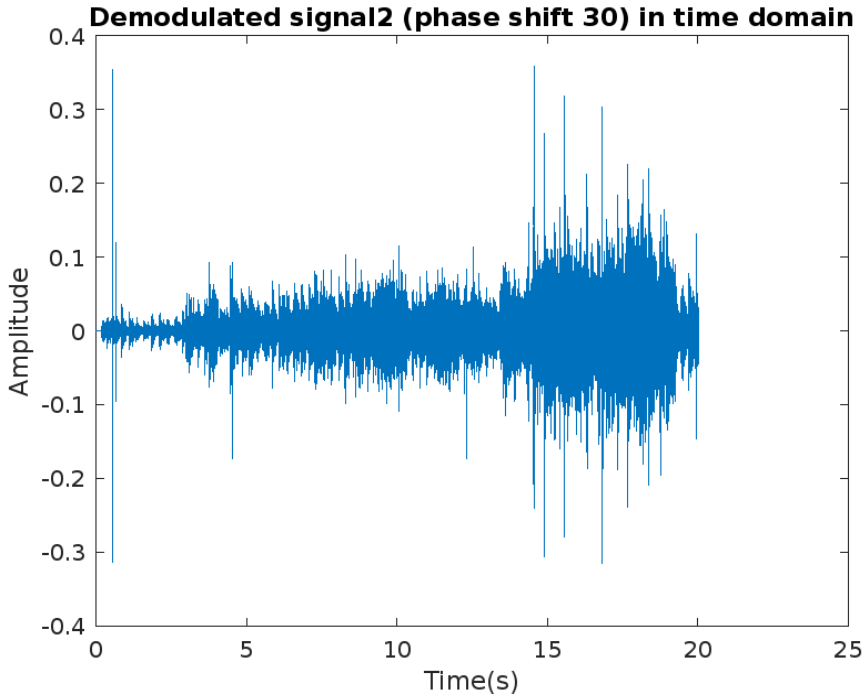


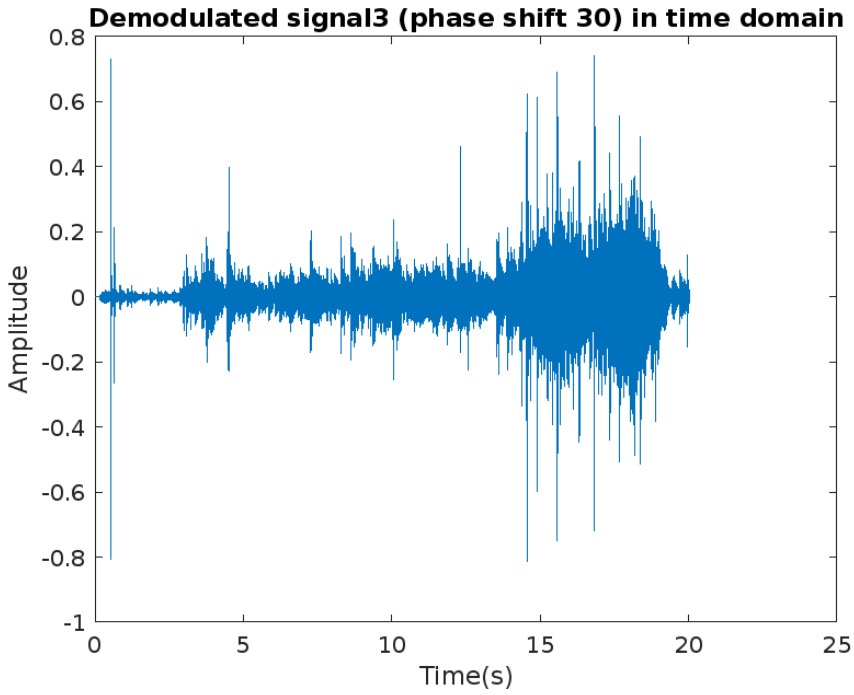




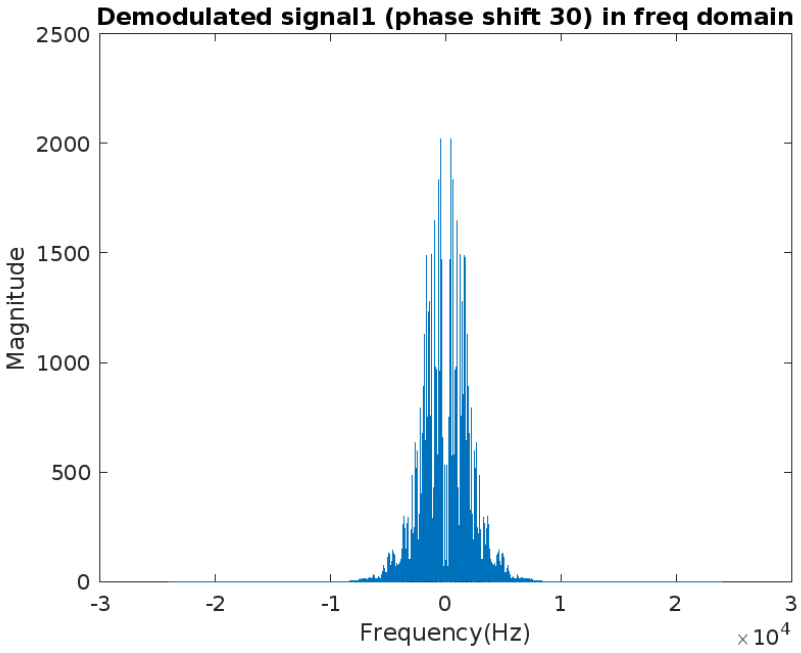
Demodulated signal with phase shift 30 in time Domain

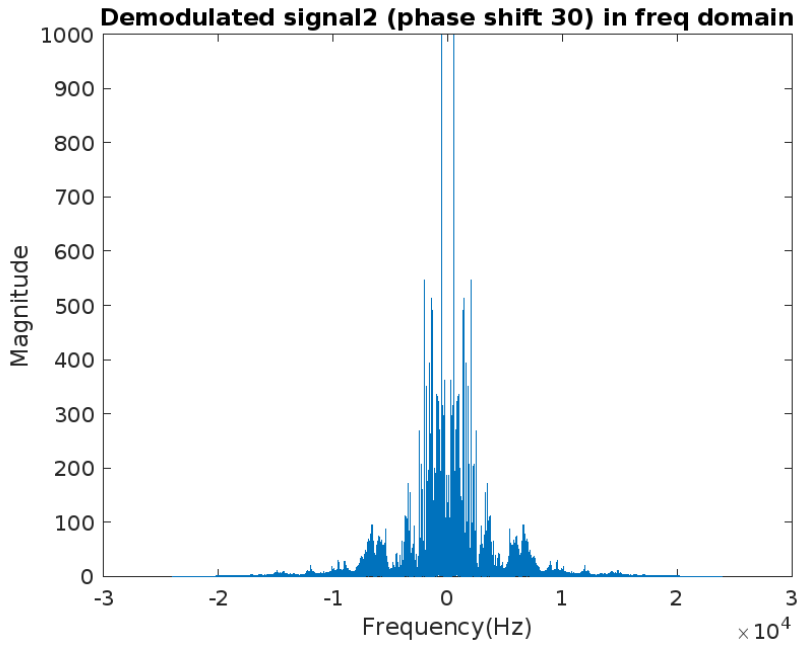

Demodulated signal1 (phase shift 30) in time domain
Amplitude
Time(s)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
5
10
15
20
25

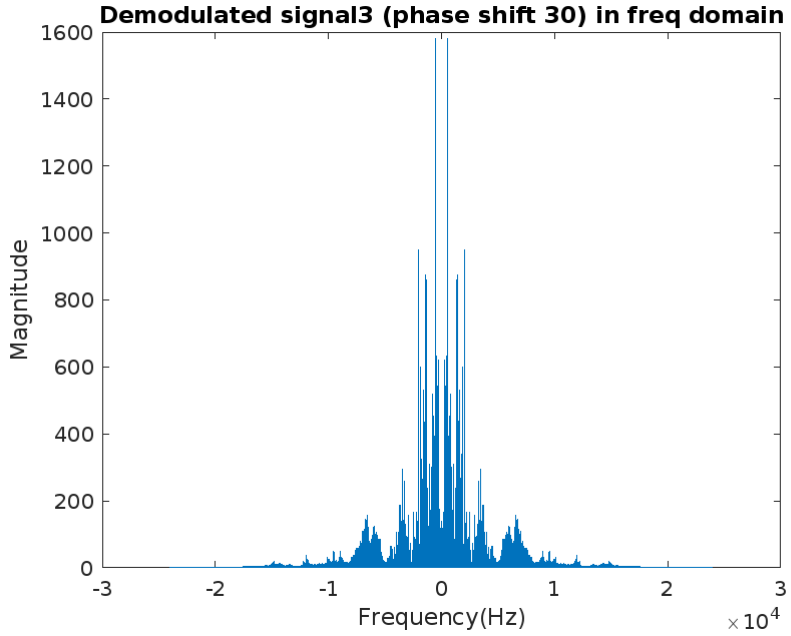

Demodulated signal2 (phase shift 30) in time domain
Amplitude
Time(s)
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
5
10
15
20
25



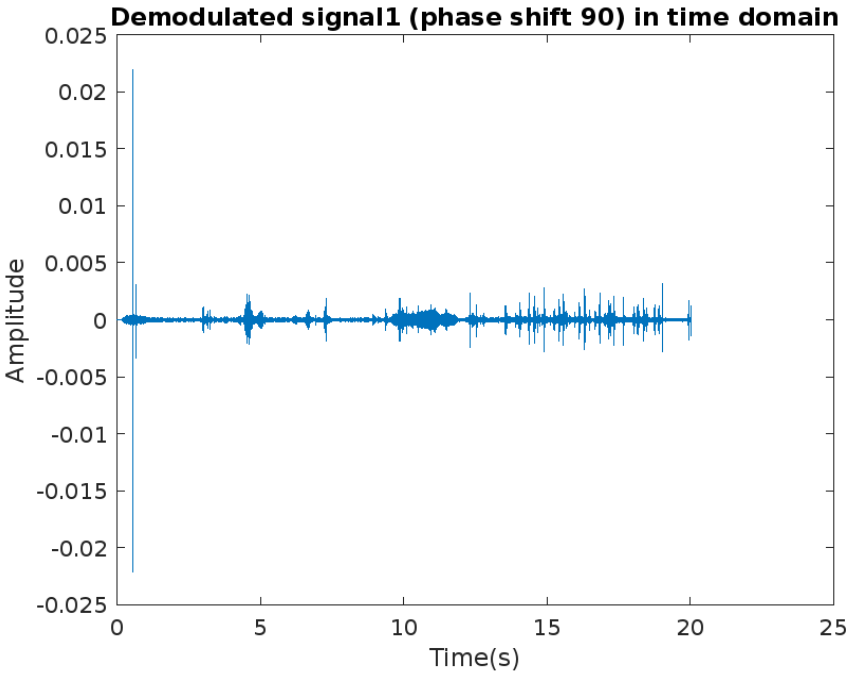
Demodulated signal with phase shift 30 in frequency Domain

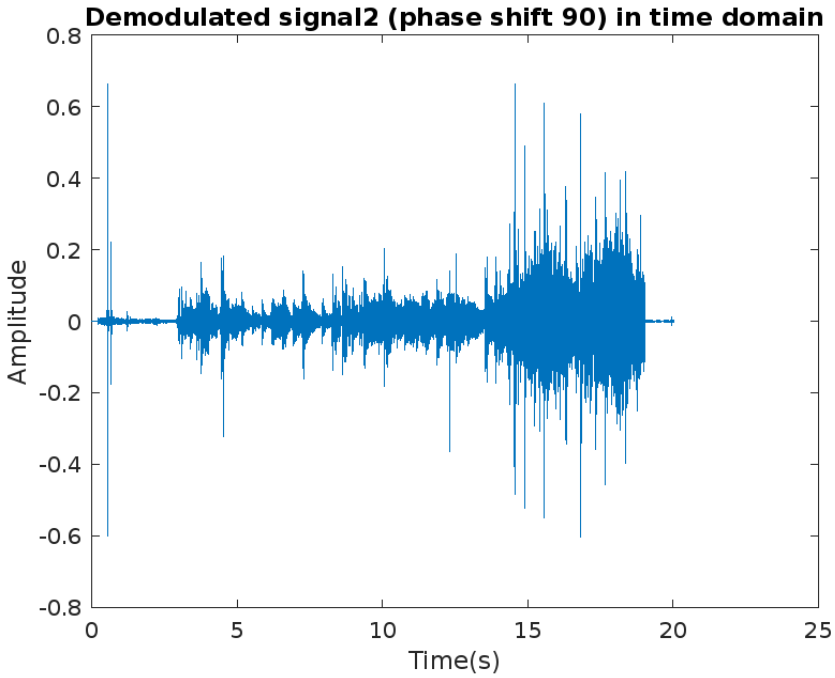


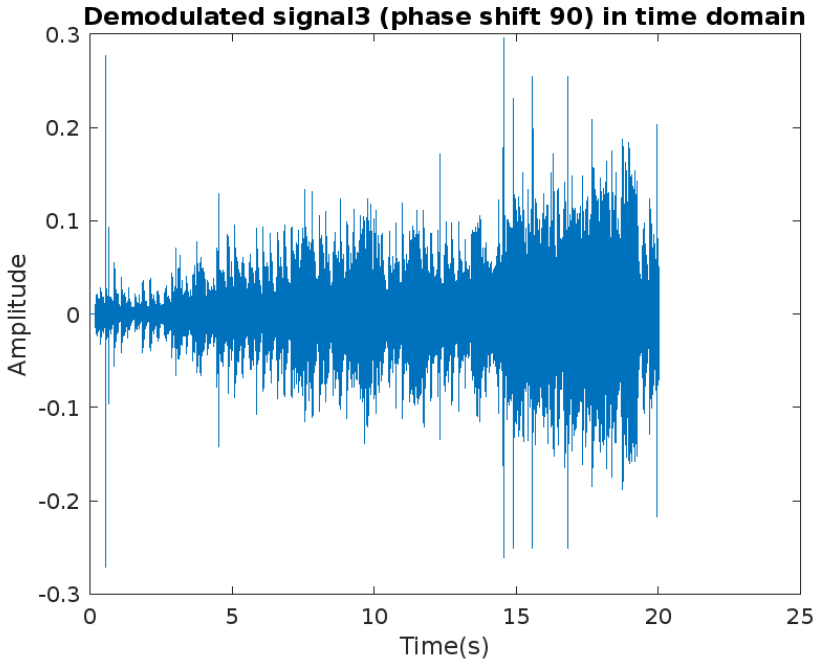




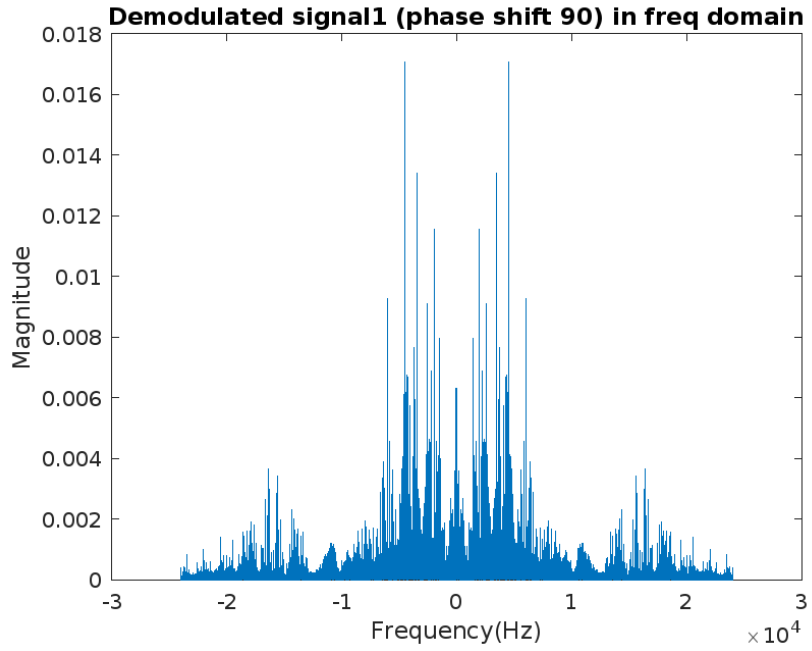
Demodulated signal with phase shift 90 in time Domain

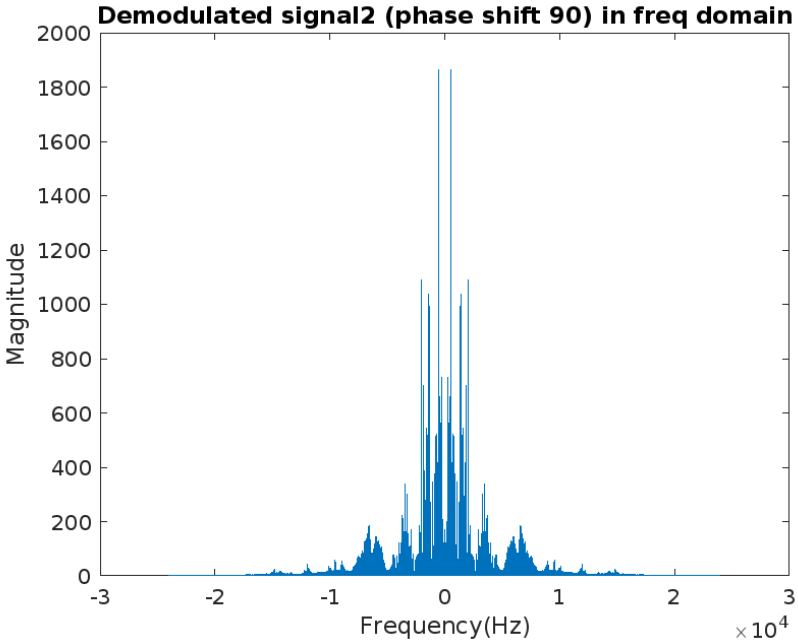

Demodulated signal1 (phase shift 90) in time domain
Amplitude
Time(s)
0.025
0.02
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
-0.025
0
5
10
15
20
25

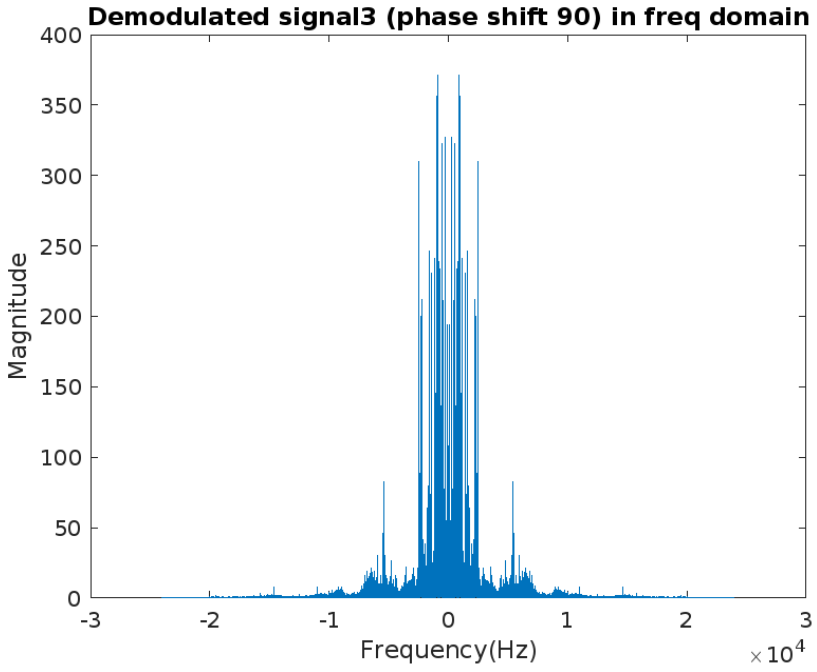

Demodulated signal2 (phase shift 90) in time domain
Amplitude
Time(s)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
5
10
15
20
25



Demodulated signal with phase shift 90 in frequency Domain







Code:

```
% Read signal 1 and pre-process it
[s1, fs1] = audioread("signal1.wav");
s1 = s1(:,1) + s1(:,2);
[len1, BW1, t1, f1, ffts1]=preprocessing(s1,fs1);
% Read signal 2 and pre-process it
```

```
[s2, fs2] = audioread("signal2.wav");
s2 = s2(:,1) + s2(:,2);
[len2, BW2, t2, f2, ffts2]=preprocessing(s2,fs2);
% Read signal 3 and pre-process it
[s3, fs3] = audioread("signal3.wav");
s3 = s3(:,1) + s3(:,2);
[len3, BW3, t3, f3, ffts3]=preprocessing(s3,fs3);
% Plot original signals in time domain
plotSignal(t1,s1,'Original signal 1 in time domain', 'Time(s)',
'Amplitude');
plotSignal(t2,s2,'Original signal 2 in time domain', 'Time(s)',
'Amplitude');
plotSignal(t3,s3,'Original signal 3 in time domain', 'Time(s)',
'Amplitude');
% Plot original signals in freq domain
plotSignal(f1,fftshift(ffts1),'Original signal 1 in freq domain',
'Frequency(Hz)', 'Magnitude');
plotSignal(f2,fftshift(ffts2),'Original signal 2 in freq domain',
'Frequency(Hz)', 'Magnitude');
plotSignal(f3,fftshift(ffts3),'Original signal 3 in freq domain',
'Frequency(Hz)', 'Magnitude');
% Resample original signals
fs_new=650000;
[s1_resampled,len_res1,t_res1]=resampleSignal(s1,fs_new,fs1);
[s2_resampled,len_res2,t_res2]=resampleSignal(s2,fs_new,fs2);
[s3_resampled,len_res3,t_res3]=resampleSignal(s3,fs_new,fs3);
% Modulate signals
fc1=160000;
fc2=350000;
carrier1 = cos(2*pi*fc1* t_res1);
[s1_modulated,
len_ms1,fm1,fftms1]=Modulation(carrier1,s1_resampled,fs_new);
carrier2 = cos(2*pi*fc2* t_res2);
[s2_modulated,
len_ms2,fm2,fftms2]=Modulation(carrier2,s2_resampled,fs_new);
carrier3 = sin(2*pi*fc2* t_res3);
[s3_modulated,
len_ms3,fm3,fftms3]=Modulation(carrier3,s3_resampled,fs_new);
% Plot modulated signals in freq domain
plotSignal(fm1,fftshift(fftms1),'Modulated signal 1 in freq domain',
'Frequency(Hz)', 'Magnitude');
plotSignal(fm2,fftshift(fftms2),'Modulated signal 2 in freq domain',
'Frequency(Hz)', 'Magnitude');
plotSignal(fm3,fftshift(fftms3),'Modulated signal 3 in freq domain',
'Frequency(Hz)', 'Magnitude');
% Sum modulated signals
max_len = max(len_ms1, max(len_ms2, len_ms3));
ss1 = [s1_modulated;zeros(max_len-len_ms1, 1)];
ss2 = [s2_modulated;zeros(max_len-len_ms2, 1)];
```

```
ss3 = [s3_modulated;zeros(max_len-len_ms3, 1)];
modulated_signal_sum = ss1 + ss2 + ss3;
t_sum = (0: max_len - 1) * (1 / fs_new);
f_sum = (-max_len/2 : max_len/2 - 1) * (fs_new / max_len);
fft_sum = abs(fft(modulated_signal_sum));
% plot the final signal
plotSignal(t_sum,modulated_signal_sum,'Modulated final signal in time
domain', 'Time(s)', 'Amplitude');
plotSignal(f_sum,fftshift(fft_sum),'Modulated final signal in freq
domain', 'Frequency(Hz)', 'Magnitude');
% Demodulate final signal to restore the 3 signals
sync_carrier1 = cos(2*pi*fc1 * t_sum);
[fftds1]=Demodulation(modulated_signal_sum, sync_carrier1, len1, fs1,
fs_new ,100000, 'Demodulated signal 1',t_sum,f1,'1');
sync_carrier2 = cos(2*pi*fc2 * t_sum);
[fftds2]=Demodulation(modulated_signal_sum, sync_carrier2, len2, fs2,
fs_new ,7000, 'Demodulated signal 2',t_sum,f2,'2');
sync_carrier3 = sin(2*pi*fc2 * t_sum);
[fftds3]=Demodulation(modulated_signal_sum, sync_carrier3, len3, fs3,
fs_new ,25000, 'Demodulated signal 3',t_sum,f3,'3');
% Demodulate the final signal with phase shift 10
phase10_carrier1 = cos(2*pi*fc1*t_sum + (10 * pi) / 180);
[fftds1_10]=Demodulation(modulated_signal_sum, phase10_carrier1, len1,
fs1, fs_new ,100000, 'Demodulated 10 signal 1',t_sum,f1,'1 (phase shift
10)');
phase10_carrier2 = cos(2*pi*fc2*t_sum + (10 * pi) / 180);
[fftds2_10]=Demodulation(modulated_signal_sum, phase10_carrier2, len2,
fs2, fs_new ,7000, 'Demodulated 10 signal 2',t_sum,f2,'2 (phase shift
10)');
phase10_carrier3 = sin(2*pi*fc2*t_sum + (10 * pi) / 180);
[fftds3_10]=Demodulation(modulated_signal_sum, phase10_carrier3, len3,
fs3, fs_new ,25000, 'Demodulated 10 signal 3',t_sum,f3,'3 (phase shift
10)');
% Demodulate the final signal with phase shift 30
phase30_carrier1 = cos(2*pi*fc1*t_sum + (30 * pi) / 180);
[fftds1_30]=Demodulation(modulated_signal_sum, phase30_carrier1, len1,
fs1, fs_new ,100000, 'Demodulated 30 signal 1',t_sum,f1,'1 (phase shift
30)');
phase30_carrier2 = cos(2*pi*fc2*t_sum + (30 * pi) / 180);
[fftds2_30]=Demodulation(modulated_signal_sum, phase30_carrier2, len2,
fs2, fs_new ,7000, 'Demodulated 30 signal 2',t_sum,f2,'2 (phase shift
30)');
phase30_carrier3 = sin(2*pi*fc2*t_sum + (30 * pi) / 180);
[fftds3_30]=Demodulation(modulated_signal_sum, phase30_carrier3, len3,
fs3, fs_new ,25000, 'Demodulated 30 signal 3',t_sum,f3,'3 (phase shift
30)');
% Demodulate the final signal with phase shift 90
phase90_carrier1 = cos(2*pi*fc1*t_sum + (90 * pi) / 180);
```

```
[fftds1_90]=Demodulation(modulated_signal_sum, phase90_carrier1, len1,
fs1, fs_new ,100000, 'Demodulated 90 signal 1',t_sum,f1,'1 (phase shift
90)');
phase90_carrier2 = cos(2*pi*fc2*t_sum + (90 * pi) / 180);
[fftds2_90]=Demodulation(modulated_signal_sum, phase90_carrier2, len2,
fs2, fs_new ,7000, 'Demodulated 90 signal 2',t_sum,f2,'2 (phase shift
90)');
phase90_carrier3 = sin(2*pi*fc2*t_sum + (90 * pi) / 180);
[fftds3_90]=Demodulation(modulated_signal_sum, phase90_carrier3, len3,
fs3, fs_new ,25000, 'Demodulated 90 signal 3',t_sum,f3,'3 (phase shift
90)');
% Demodulate the final signal with fc+difference
Local2_carrier1 = cos(2*pi*(fc1+2)*t_sum);
[fftds1_diff2]=Demodulation(modulated_signal_sum, Local2_carrier1, len1,
fs1, fs_new , 100000, 'Demodulated +2 signal 1',t_sum,f1,'1 (+2HZ)');
Local10_carrier1 = cos(2*pi*(fc1+10)*t_sum);
[fftds1_diff10]=Demodulation(modulated_signal_sum, Local10_carrier1,
len1, fs1, fs_new , 100000, 'Demodulated +10 signal 1',t_sum,f1,'1
(+10HZ)');
function [len, BW, t, f,ffts] = preprocessing (s,fs)
   % get the signal length
   len = length(s);
   % get its bandwidth
   BW=bandwidth(s)./(2.*pi);
   % get its time range
   t = (0:len - 1)*20/len;
   % get its freq range
   f=(-fs/2:fs/len:fs/2-fs/len);
   % get its fft
   ffts = abs(fft(s));
end
function plotSignal (range, signal, label, x, y)
 fig=figure();
 plot(range,signal)
 title(strcat(label, ""))
 xlabel(strcat(x, ""))
 ylabel(strcat(y, ""))
 saveas(fig,label,'png')
end
function [re_s,len,t] = resampleSignal (signal,fs_all,fs_signal)
% resample the signal
[x, y] = rat(fs_all/fs_signal);
re_s = resample(signal, x, y);
% get its new length
len = length(re_s);
% get its time range
t = (0:len - 1) * (1/fs_all);
end
function [mod_s, len_ms,fms,fftms] = Modulation (c,s,fs)
```

```
% multiply the signal by the carrier
mod_s = s .* c';
% get its length
len_ms = length(mod_s);
% get its freq range
fms = (fs/len_ms)  * (-len_ms/2: len_ms/2 - 1);
% get its fft
fftms = abs(fft(mod_s));
end
function [fftds] = Demodulation(s, c, old_len, fs_old, fs_new,
f_pass,filename,t,f,index)
   % multiply the signal by the carrier
   s1 = s .* c';
   % use a low pass filter
   s2 = lowpass(s1, f_pass , fs_new);
   % raise it to its original magnitude
   s2=s2*2;
   % plot it in time domain
   plotSignal(t,s2,strcat('Demodulated signal',index,' in time domain'),
'Time(s)', 'Amplitude');
   % get its fft
   fftds = abs(fft(s2));
   % resampling
   [x, y] = rat(fs_old/fs_new);
   s3 = resample(s2, x, y);
   % return it to original length
   s3 = s3(1: old_len);
   %plot it in freq domain
   plotSignal(f,fftshift(abs(fft(s3))),strcat('Demodulated
signal',index,' in freq domain'), 'Frequency(Hz)', 'Magnitude');
   % write the sound in a file
   audiowrite(strcat(filename, '.wav'), s3,fs_old)
end
```