EE 383 Communication Systems 1 – Spring 2018

Matlab Assignment 6 – Noise in Continuous Wave Modulation

Due Date 5/13/18

By: Hussam Abdellatif

#50164204

1. ***Exercise 1: Noise Suppression in band-limited signals***
    a. ***Code***

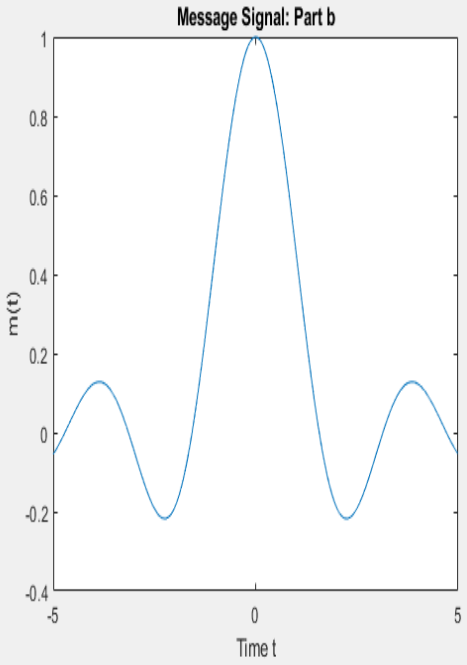
Calculation For Bandwidth

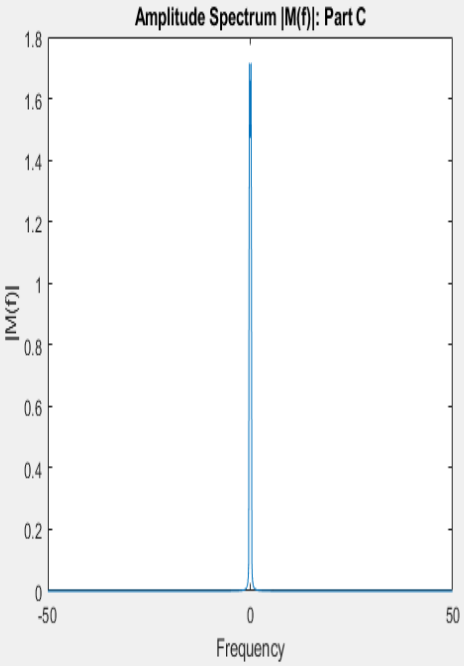
$$Bandwidth = \frac{2}{\pi}. Thus, the\ rect\ pulse\ stretches\ from -0.31\ up\ to\ 0.31\ Hz.$$

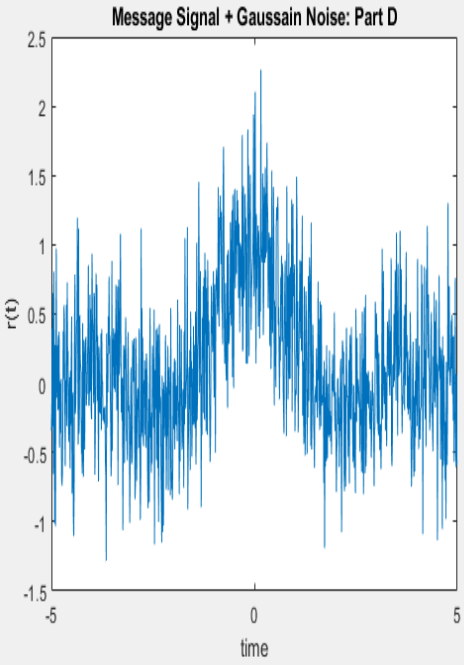
```
%part b
t = -5:0.01:4.99;
m_t = sinc( (2*t) / pi);
subplot(2,3,1);
plot(t,m_t);
title("Message Signal: Part b");
xlabel("Time t");
ylabel("m(t)");

%part c

[m_f,f] = fouriert(m_t,t);
subplot(2,3,2);
plot(f,abs(m_f));
title("Amplitude Spectrum |M(f)|: Part C");
xlabel("Frequency");
ylabel("|M(f)|");

%part d
s_dev = sqrt(0.2);
n_t = s_dev.*randn(1,length(t));
r_t = m_t + n_t;
subplot(2,3,3);
plot(t,r_t);
title("Message Signal + Gaussain Noise: Part D");
xlabel("time");
ylabel("r(t)");

%part e

[R_f,f_parte] = fouriert(r_t,t);
```

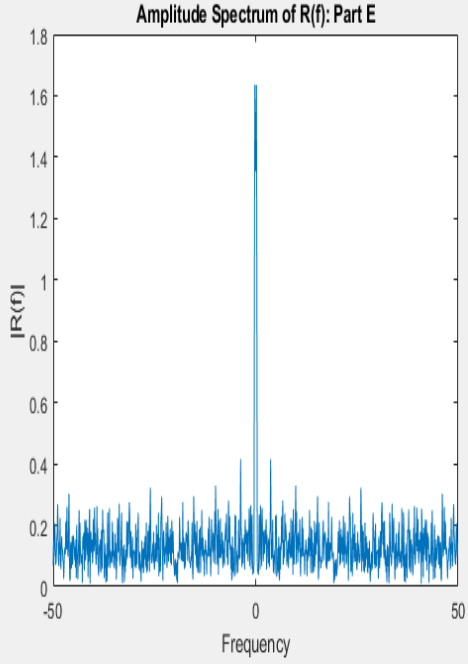
```
subplot(2,3,4);
plot(f_parte,abs(R_f));
title("Amplitude Spectrum of R(f): Part E");
xlabel("Frequency");
ylabel("|R(f)|");

%part f
[slp, tslp] = lowpass(r_t, t, 0.35);
subplot(2,3,5);
plot(tslp,slp);
title("Filtered Signal r(t): Part f");
xlabel("time");
ylabel("r_filt(t)");



%part g

[slp_f,tslp_f] = fouriert(slp,tslp);
subplot(2,3,6);
plot(tslp_f,abs(slp_f));
title("Amplitude Spectrum of Filtered signal r(t): Part g");
xlabel("Frequency");
ylabel("R_Filt(f)");
```

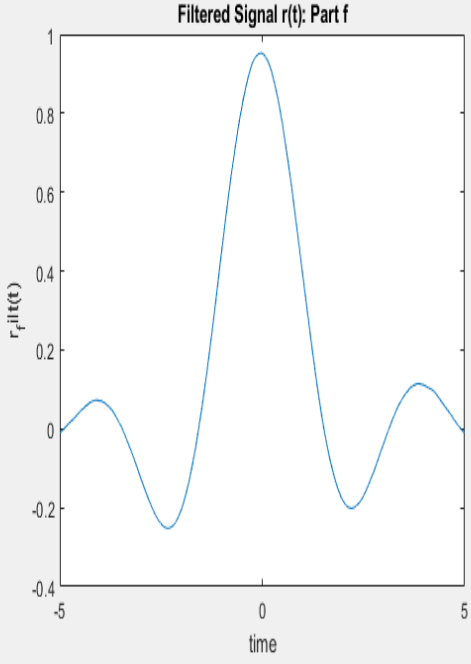
b. ***Plot***

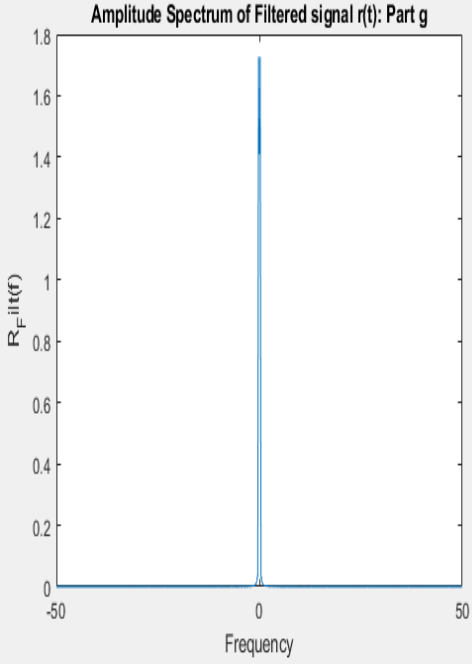

Message Signal: Part b
m(t)
Time t


Amplitude Spectrum |M(f)|: Part C
|M(f)|
Frequency


Message Signal + Gaussain Noise: Part D
r(t)
time


Amplitude Spectrum of R(f): Part E
|R(f)|
Frequency


Filtered Signal r(t): Part f
time


Amplitude Spectrum of Filtered signal r(t): Part g
Frequency

c. Conclusion

The signal was recovered completely, since the bandwidth of the message signal is 2/pi and stretches from -1/pi to 1/pi we were able to filter the noise using a low pass filter with cut-off frequency of 0.35 Hz. The bandwidth can also be seen from the amplitude spectrum of the message signal in the frequency domain. Anything above that will be cleaned out. If you zoom in, you will see some minor distortion in the bottom of the signal, however it is insignificant. The amplitude spectrum of the recovered signal was recovered completely.

2. Noise in DSB-SC

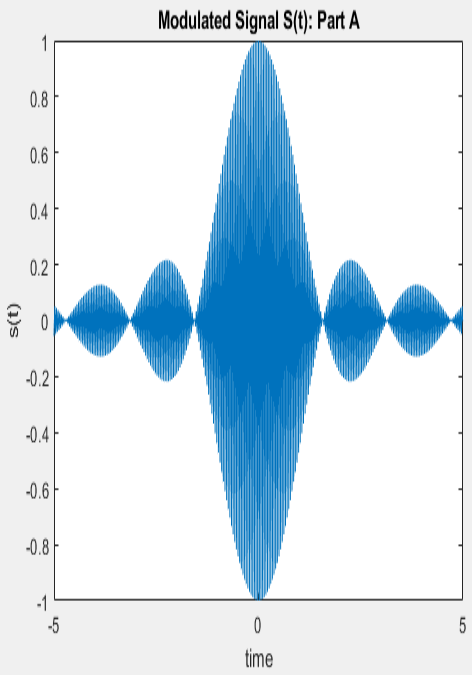
a. Code

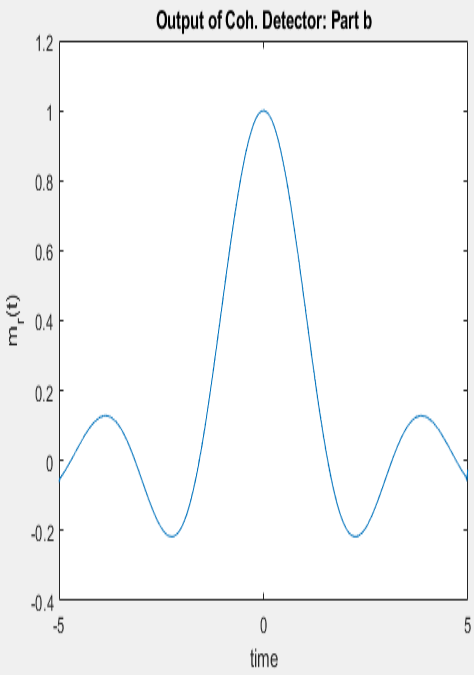
```
t = -5:0.01:4.99;
m_t = sinc( (2*t) / pi);
A_c = 1;
f_c = 25;
c_t = A_c * cos(2*pi*f_c*t);
s_t = m_t .* c_t;

%part a

subplot(2,3,1);
plot(t,s_t);
title("Modulated Signal S(t): Part A");
xlabel("time");
ylabel("s(t)");

%part b
r_t = s_t;
f_cut = 25;
r_t2 = r_t .* (2*cos(2*pi*f_c*t));
[r_t3, t_lp] = lowpass(r_t2,t,f_cut);
subplot(2,3,2);
plot(t_lp,r_t3);
title("Output of Coh. Detector: Part b");
xlabel("time");
ylabel("m_r(t)");
```

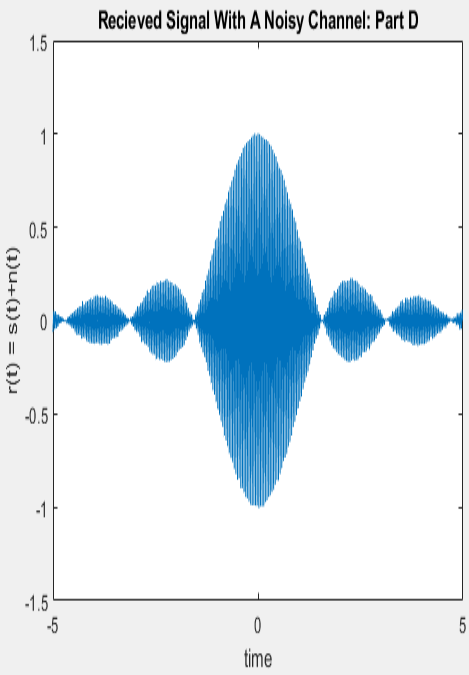
```
%part c (P_noise = 0.006 watts)
p_signal = norm(r_t3)^2/length(r_t3);
SNR = db2pow(14);
P_noise = p_signal / SNR;

%part d

 n_t = P_noise.*randn(1,length(t));
 r_t4 = s_t + n_t ;
 subplot(2,3,3);
 plot(t,r_t4);
 title("Recieved Signal With A Noisy Channel: Part D");
 xlabel("time");
 ylabel("r(t) = s(t)+n(t)");

 %part e

 m_rec_noisy = r_t4 .* (2 * cos(2 * pi * 25 * t));
 [mr, time] = lowpass(m_rec_noisy,t,f_cut);
 %[mr2, time2] = lowpass(mr,time,0.35);
 subplot(2,3,4);
 plot(time,mr);
 title("Recovered Signal With A Noisy Channel: Part E");
 xlabel("time");
 ylabel("m_r(t)");

 %part f

 my_detector_s1 = r_t4 .* (2 * cos(2 * pi * 25 * t));
 [my_det_s2,time_s2] = lowpass(my_detector_s1, t, 25);
 [my_det_s3,time_s3] = lowpass(my_det_s2,time_s2,0.32);

 subplot(2,3,5);
 plot(time_s3,my_det_s3);
 title("Proposed Detector Recovered Signal With A Noisy
Channel: Part f");
 xlabel("time");
```

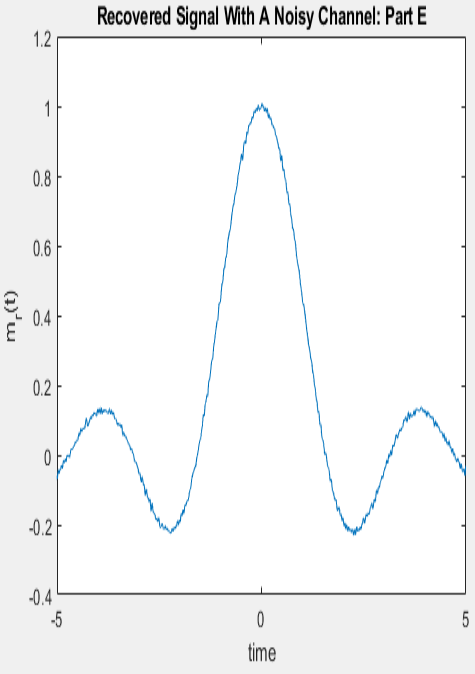
```
ylabel("m_r(t)");
```

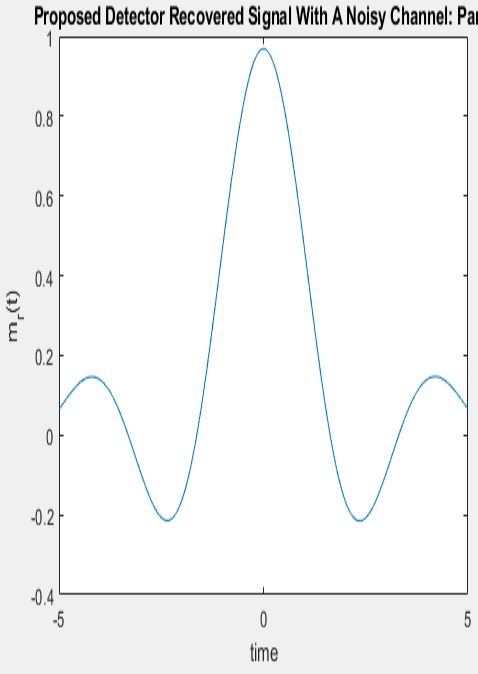
b. ***Plot***











c. Conclusion

In part b, I recovered a modulated signal (without noise), and the recovered signal was in fact fully recovered using a coherent detector. Then I calculated the average noise power, which turned out to be 0.006 watts. Then we applied additive noise to the modulated signal, and attempted to recover the message signal using a coherent detector. The signal was recovered, however it contained some distortion. Since I know that the bandwidth of the signal stretches from 1/pi to -1/pi, I recognized that I can add another low pass filter at the output of the coherent detector, but instead of a cut off frequency of 25, I used 0.32 Hz. Since the first low pass filter, cleans out carrier signal that has a carrier frequency of 25 Hz, it does not really clean out the noise completely. However, we can fully recover the message signal with another low pass filter.