GEBZE TECHNICAL UNIVERSITY

ELECTRONIC ENGINEERING

ELEC361

ANALOG COMMUNICATION SYSTEMS

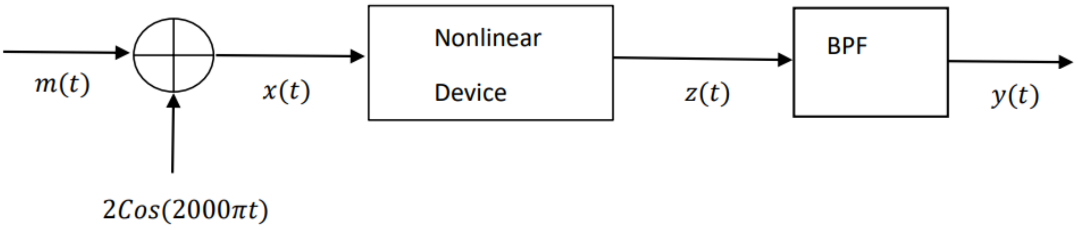
MATLAB PROJECT ASSIGNMENT

Beyzanur Cam – 210102002037

Introduction

In this project, the use of the DSB Large Carrier (LC)-AM modulator and detailed graphic solutions for each step are included in general terms. It also includes the use of filters in the frequency domain and demodulation of the given signal.

The message signal $m(t) = 5Cos(200\pi t) + 10Cos(400\pi t)$ is transmitted by using the DSB Large Carrier (LC)-AM modulator given below. The nonlinear device has the input-output relationship $z(t) = 60x(t)$ + x 2 (t). The bandpass filter (BPF) has a gain of 1, $BW = 2B = 400\ Hz$, center frequency $fC = 1\ kHz$.



a) Plot $m(t)$, $x(t)$, $z(t)$, $y(t)$.

To achieve the m(t) graph in MATLAB first the range in the time domain is defined and as we see from the code below it is a very small amount of time because it is the period of m(t). After that, our function is defined, and it is plotted with our defined time variable.

```
t = linspace(0,0.01,10000); %range in time domain
mt=5*cos(200*pi*t)+10*cos(400*pi*t); % defining message signal m(t)

plot(t,mt); %plotting function
xlabel('time (s)')
ylabel('Amplitude of m(t)')
title('Plot of m(t)’)
```

In order to plot message signal in analytics there are approximate values were given. And with the help of these values m(t) graph is drawn by hand.

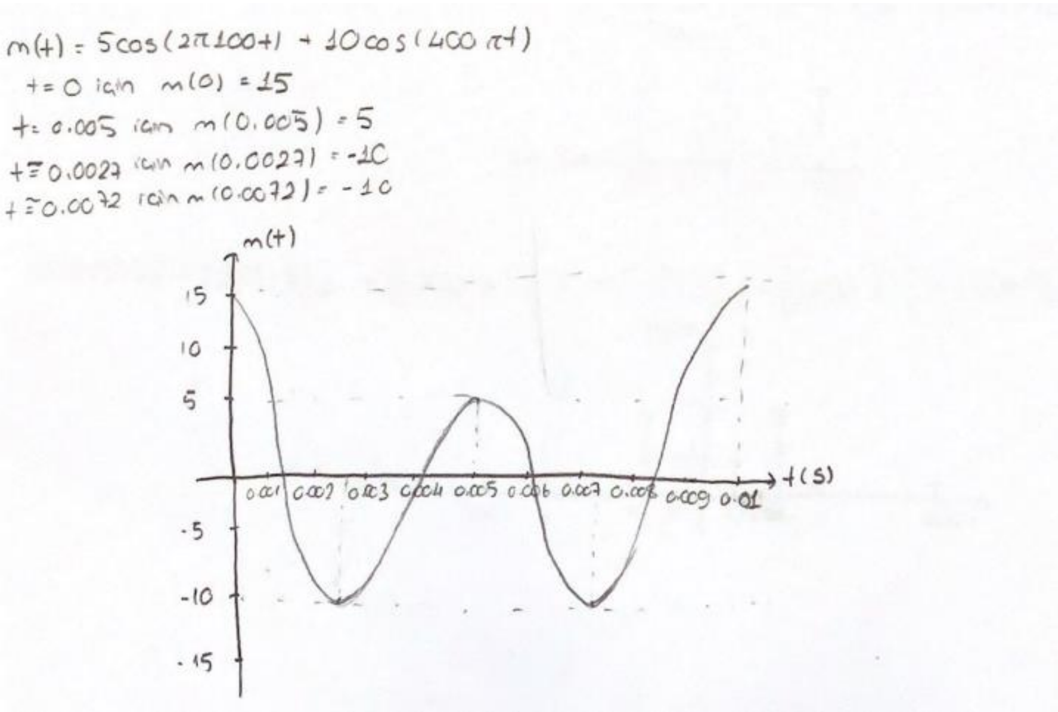


Figure 1 Analytic Graphic Drawn of m(t).

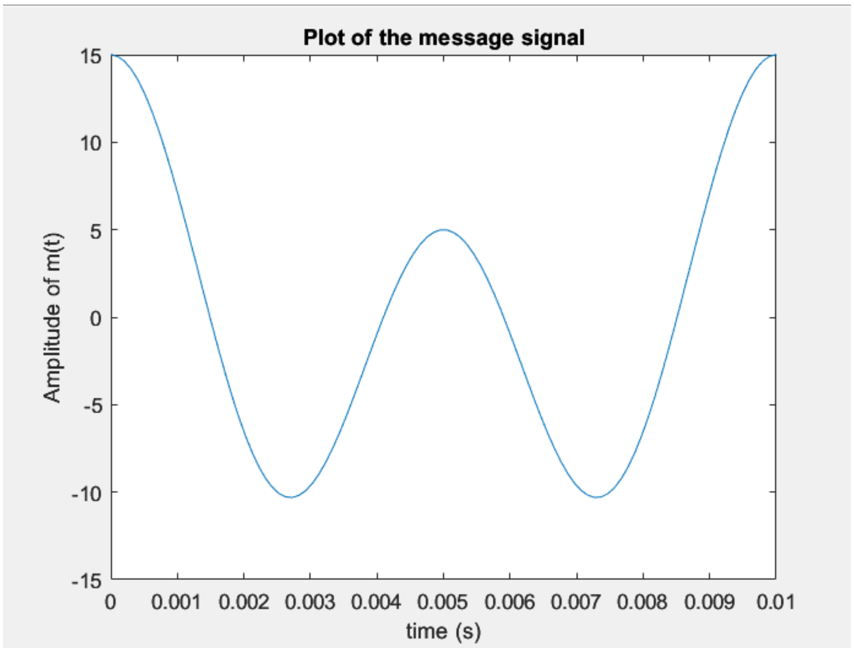


Figure 2 MATLAB Graph of m(t).

To achieve the x(t) graph in MATLAB and analytic same steps are followed.

```
t = linspace(0,0.01,10000); %range in time domain

mt=5*cos(200*pi*t)+10*cos(400*pi*t); % defining message signal
ct=2*cos(2000*pi*t); % defining carrier signal
xt=mt+ct; %defining modulated signal
```

```
plot(t,xt); %plotting function
xlabel('time (s)')
ylabel('Amplitude of x(t)')
title('Plot of the modulated signal')
```

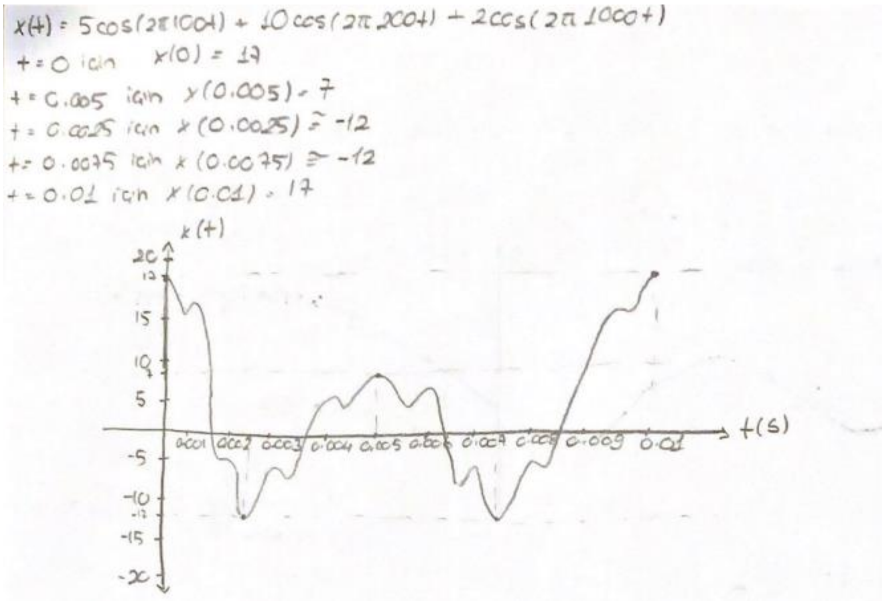


Figure 3 Analytic Graphic Drawn of x(t).

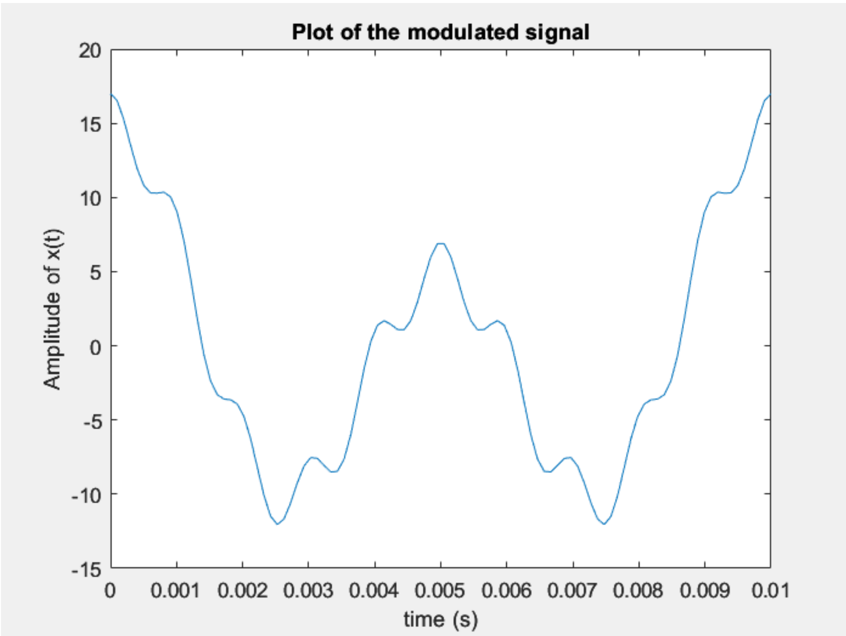


Figure 4 MATLAB Graph of x(t).

To achieve the z(t) graph in MATLAB and analytic same steps are followed.

```
t = linspace(0,0.01,10000); %range in time domain

mt=5*cos(200*pi*t)+10*cos(400*pi*t); % defining message signal
ct=2*cos(2000*pi*t); % defining carrier signal
xt=mt+ct; %defining modulated signal
```

```
zt=60*xt+xt.^2; %defining output signal of non-linear device

plot(t,zt); %plotting function
xlabel('time (s)')
ylabel('Amplitude of z(t)')
title('Plot of the non-linear device signal')
```

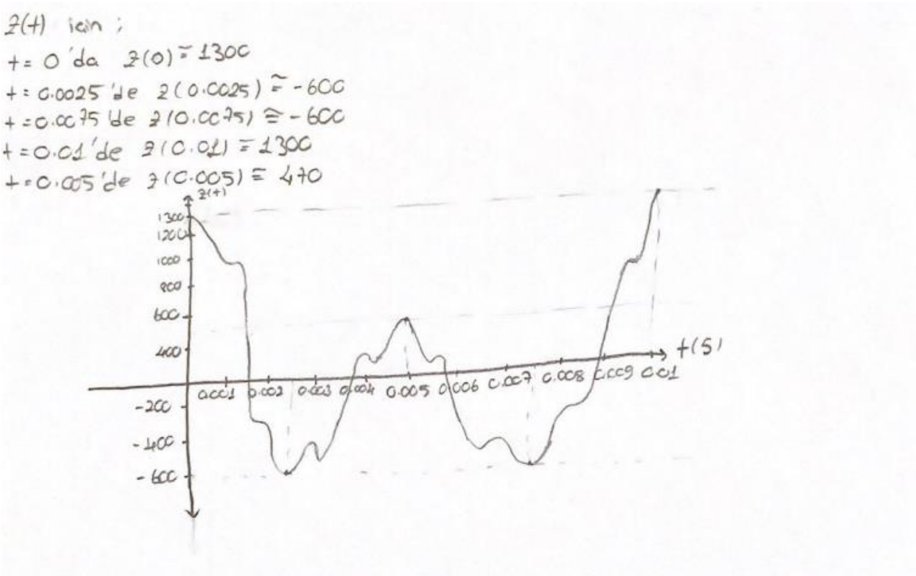


Figure 5 Analytic Graphic Drawn of z(t).

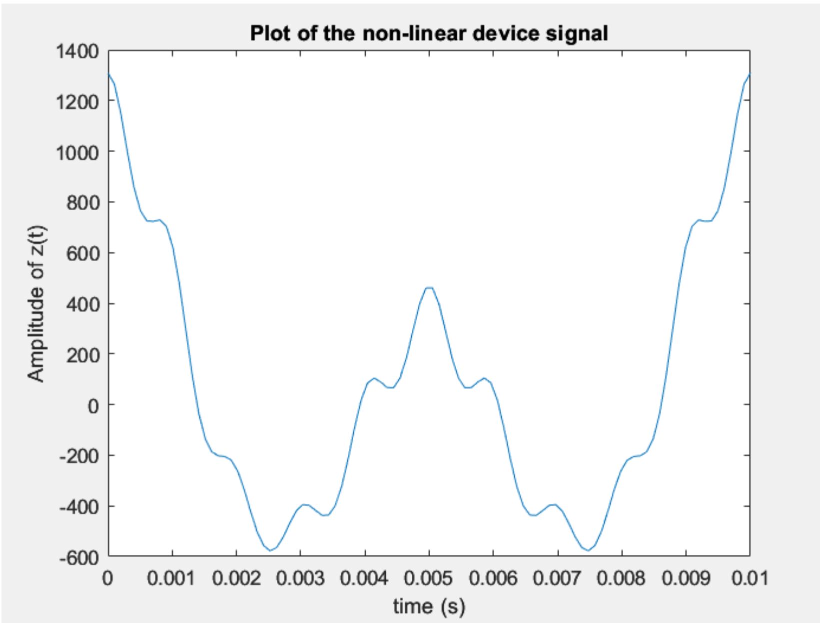


Figure 6 MATLAB Graph of z(t).

b) Find and plot $M(f)$, $X(f)$, $Z(f)$, $Y(f)$. (Do the filtering in frequency domain. Do not use filter command!)

```
Ts = 1/10000; %period of signal
t = linspace(0,0.01,10000); %range in time domain
```

```
mf = fft(mt); %taking Fourier transform to message signal with fft command

fs = 1/Ts; %frequency of signal
N = length(t); %length of time

fshift = (-N/2:N/2-1)*(1/(t(2)-t(1)))/N; %arranging frequency domain
ff1shift = fftshift(mf); %applyinf a shift to our signal

stem(fshift,abs(ff1shift)) %stem command is used for an easy presentation of
diracs
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('Plot of the M(f)')
xlim([-1000,1000]);
```

Because we know the x(t) function and it only contains cosinus signals from the Euler equations it can easily written as below.

$$\cos\omega_0 t = \left[\frac{e^{j\omega_0 t} + e^{-j\omega_0 t}}{2}\right]$$

Equation 1 Euler formula

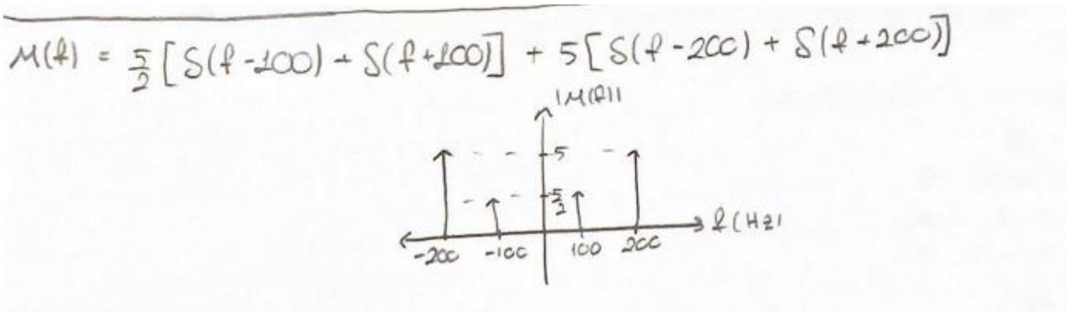


Figure 7 Analytic Graphic Drawn of M(f).

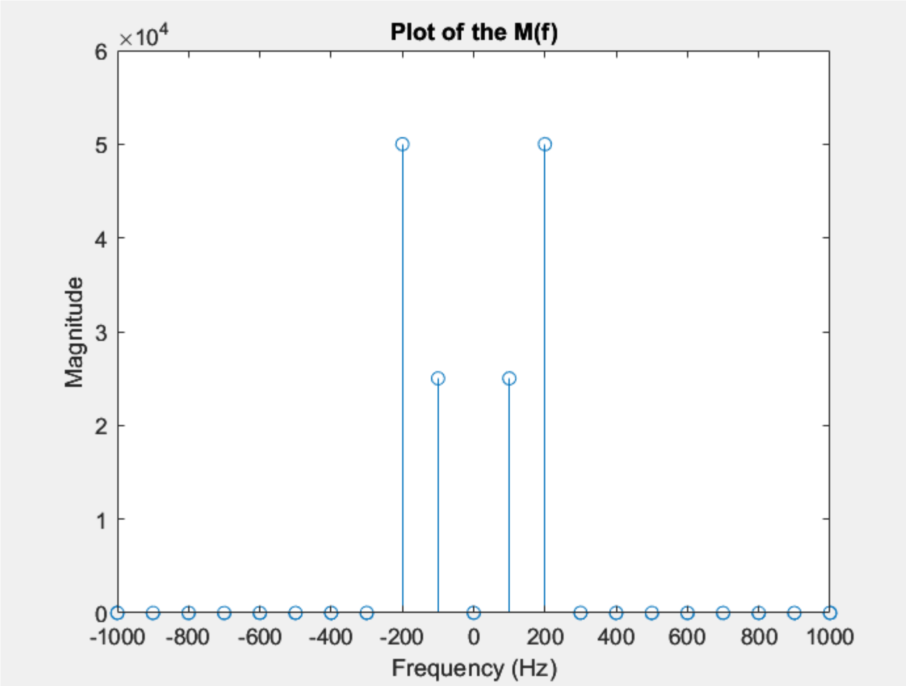


Figure 8 MATLAB Graph of M(f).

```
Ts = 1/10000; %period of signal
t = linspace(0,0.01,10000); %range in time domain

xf = fft(xt); %taking Fourier transform to message signal with fft command

fs = 1/Ts; %frequency of signal
N = length(t); %length of time

fshift = (-N/2:N/2-1)*(1/(t(2)-t(1)))/N; %arranging frequency domain
ff2shift = fftshift(xf); %applyinf a shift to our signal

stem(fshift,abs(ff2shift)) %stem command is used for an easy presentation of
diracs
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('Plot of the X(f)')
xlim([-1000,1000]);
```

$$X(f) = \frac{5}{2}\left[\delta(f-100) + \delta(f+100)\right] + 5\left[\delta(f-200) + \delta(f+200)\right] + \delta(f-1000) + \delta(f+1000)$$

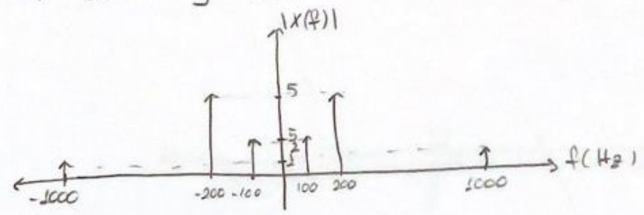


Figure 9 Analytic Graphic Drawn of X(f).

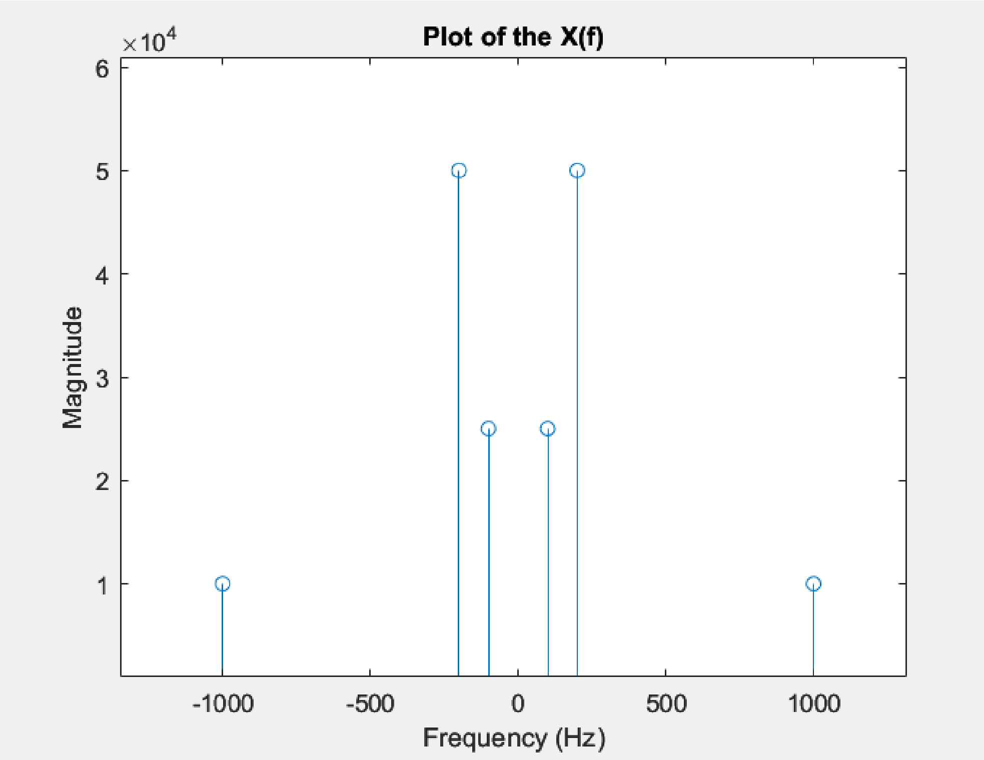


Figure 10 MATLAB Graph of X(f).

```
Ts = 1/10000; %period of signal
t = linspace(0,0.01,10000); %range in time domain

zf = fft(zt); %taking Fourier transform to message signal with fft
command
fs = 1/Ts; %frequency of signal
N = length(t); %length of time

fshift = (-N/2:N/2-1)*(1/(t(2)-t(1)))/N; %arranging frequency domain
ff3shift = fftshift(zf); %applyinf a shift to our signal

stem(fshift,abs(ff3shift)) %stem command is used for an easy presentation
of diracs
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('Plot of the Z(f)')
xlim([-1000,1000]);
```

$$Z(f) = 64.5\,\delta(f) + 150[\delta(f-100) + \delta(f+100)] + 300[\delta(f-200) + \delta(f+200)] + 60[\delta(f-1000) + \delta(f+1000)] + 6.25[\delta(f-200) + \delta(f+200)] + 25[\delta(f-400) + \delta(f+400)] + \delta(f-2000) + \delta(f+2000) + 25[\delta(f-300) + \delta(f+300)] + 25[\delta(f-100) + \delta(f+100)] + 5[\delta(f-1100) + \delta(f+1100)] + 5[\delta(f-900) + \delta(f+900)] + 10[\delta(f-1200) + \delta(f-1200)] + 10[\delta(f-800) + \delta(f+800)]$$

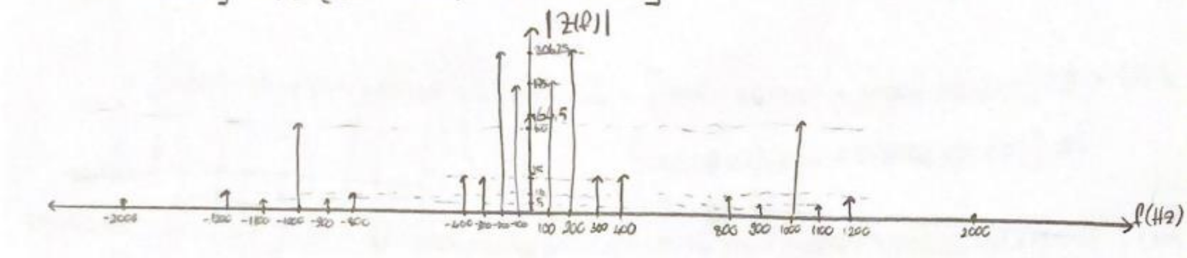


Figure 11 Analytic Graphic Drawn of Z(f).

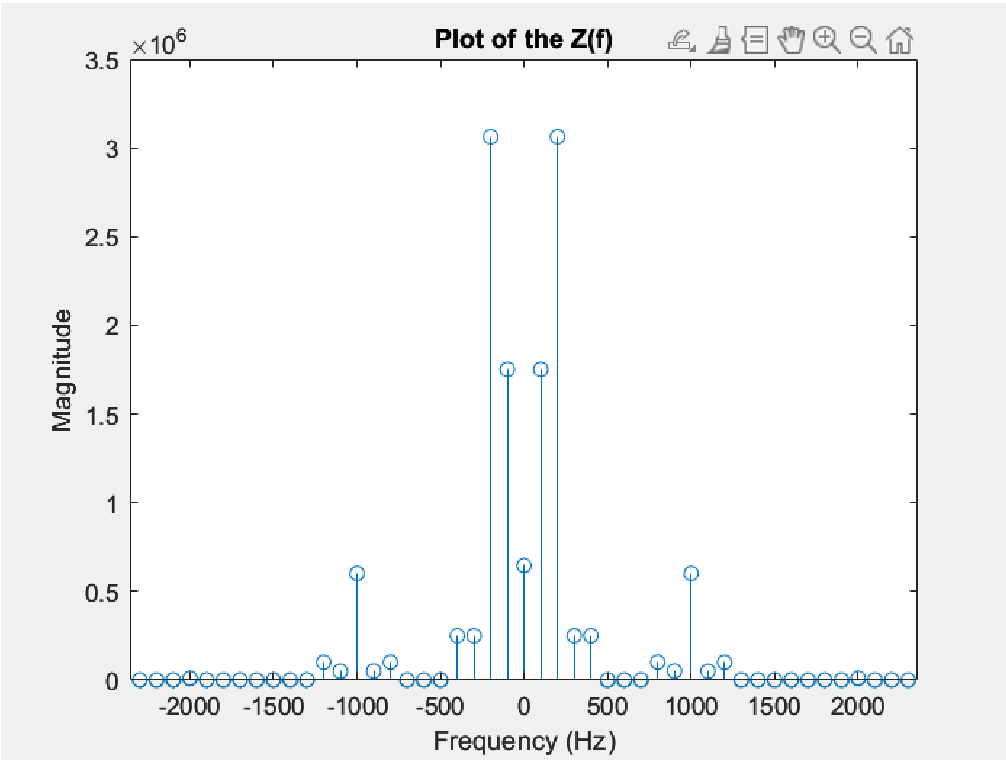


Figure 12 MATLAB Graph of Z(f).

In order to find y(t), first we need to achieve Y(f) and to do that we apply a band pass filter to Z(f) with the given properties. Our filter's center frequency is 1000 Hz, and its bandwidth is 400 Hz. Which means it will pass the signals in between 800, 1200Hz (-800, -1200Hz).

```
center_frequency = 1000; % Center frequency in Hz
bandwidth = 401; % Bandwidth in Hz

% Find indices corresponding to the bandpass filter
lower_cutoff = center_frequency - bandwidth/2; %800Hz
upper_cutoff = center_frequency + bandwidth/2; %1200Hz
```

To apply the filter there is a logic code line and it basically says if frequencies(fshift) are in between low and upper_cutoff pass them and they aren't in the range fill them with zeros. After this part both signals are plotted together for an easy comparison.

```
% Apply bandpass filter
ff3shift_filtered = zeros(size(ff3shift));
ff3shift_filtered(abs(fshift) >= lower_cutoff & abs(fshift) <= upper_cutoff) =
ff3shift(abs(fshift) >= lower_cutoff & abs(fshift) <= upper_cutoff);

% Plot the original and filtered signals
figure;
subplot(2,1,1);
stem(fshift,abs(ff3shift));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Z(f)');
xlim([-1000,1000]);

subplot(2,1,2);
stem(fshift,abs(ff3shift_filtered));
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Y(f)');
xlim([-1000,1000]);
```

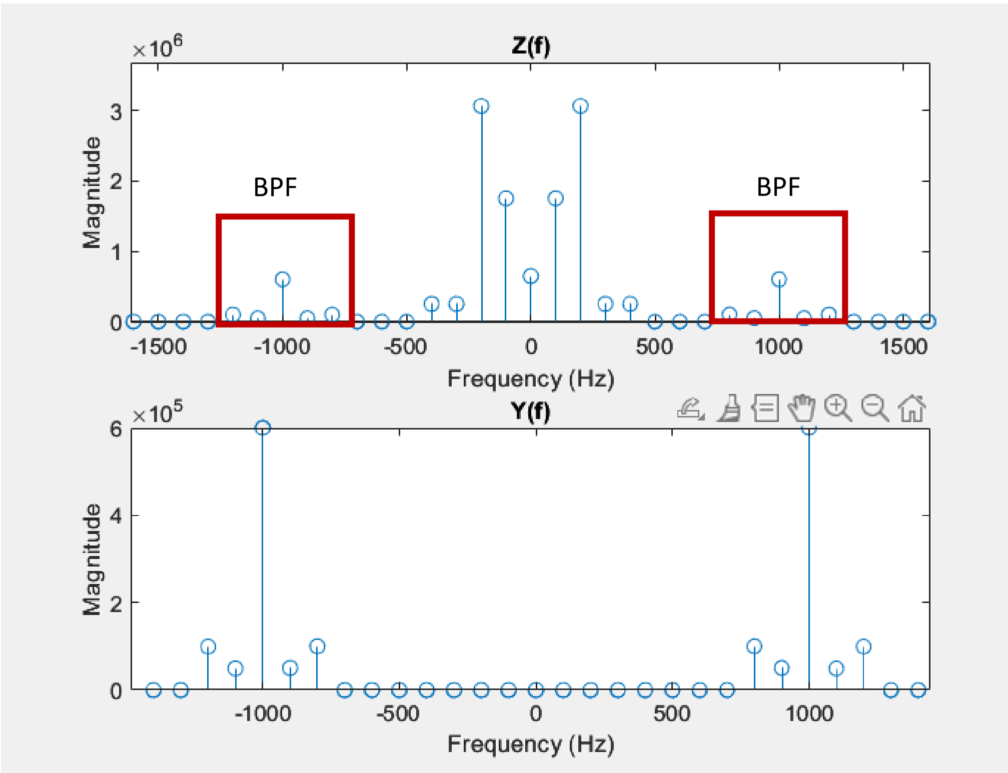


Figure 13 MATLAB Graph of Z(f) and Y(f).

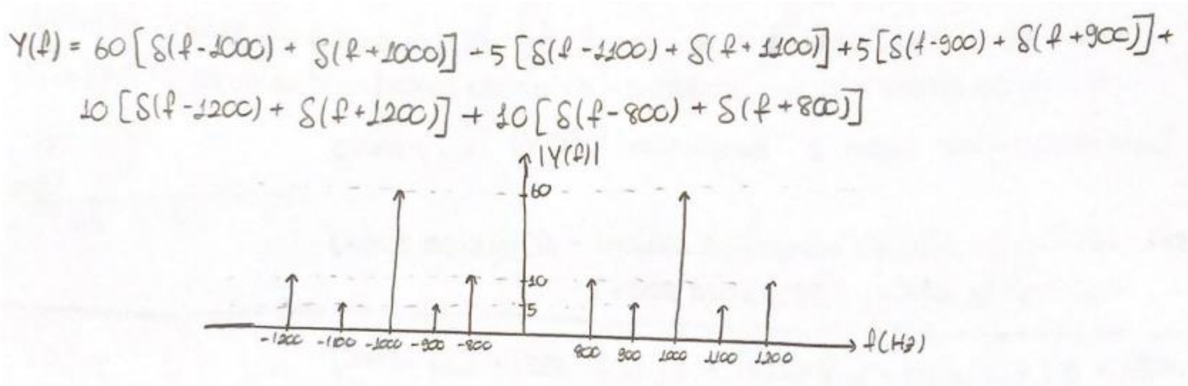


Figure 14 Analytic Graphic Drawn of Y(f).

In section a) we were not able to draw y(t) signals graph because a bandpass filter is only usable in frequency domain. And now that we obtain Y(f) in section b we can simply apply inverse Fourier transform and achieve the y(t) signal.

```
% Inverse Fourier transform to obtain time-domain signal y(t)
yt = ifft(ifftshift(ff3shift_filtered));

% Plot the time-domain signal
plot(t, yt);
xlabel('Time (s)');
ylabel('Amplitude');
title('Filtered signal y(t)');
```

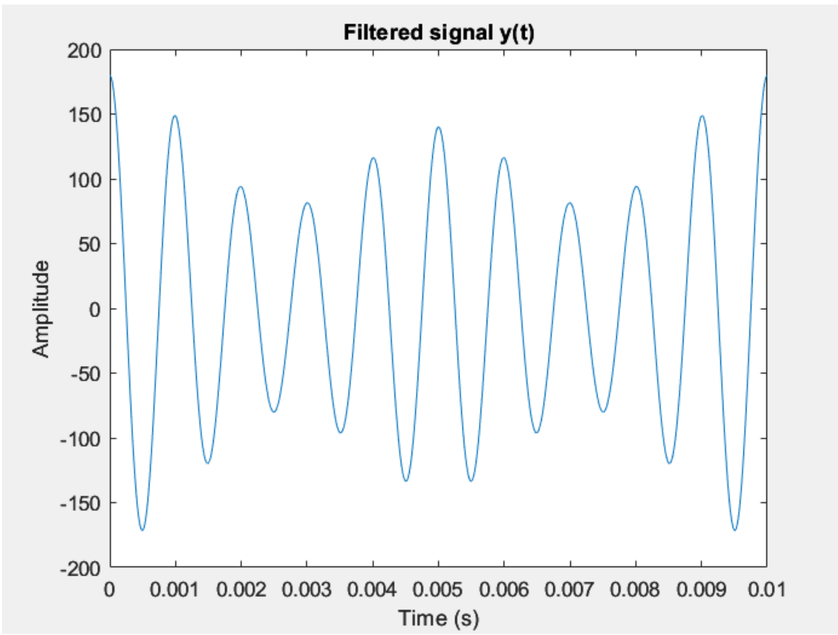


Figure 15 MATLAB Graph of y(t).

c) Demodulate $y(t)$ by using envelope detector. Plot the demodulated signal $\tilde{m}(t)$ and its spectra. Compare your result to the original message signal $m(t)$.

We know that the modulated signal is being carried in our output signals envelope. If we take y(t) signals envelope with envelope command it will give us the modulated signal m(t). With that our output signal will be demodulated because we achieved the message signal m(t).

```
% Envelope detection
env = envelope(yt);

% Plot the envelope in the time domain
plot(t,env);
hold on
xlabel('Time (s)');
plot(t,yt);
hold off
ylabel('Amplitude');
title('demodulated signal m(t)');
```

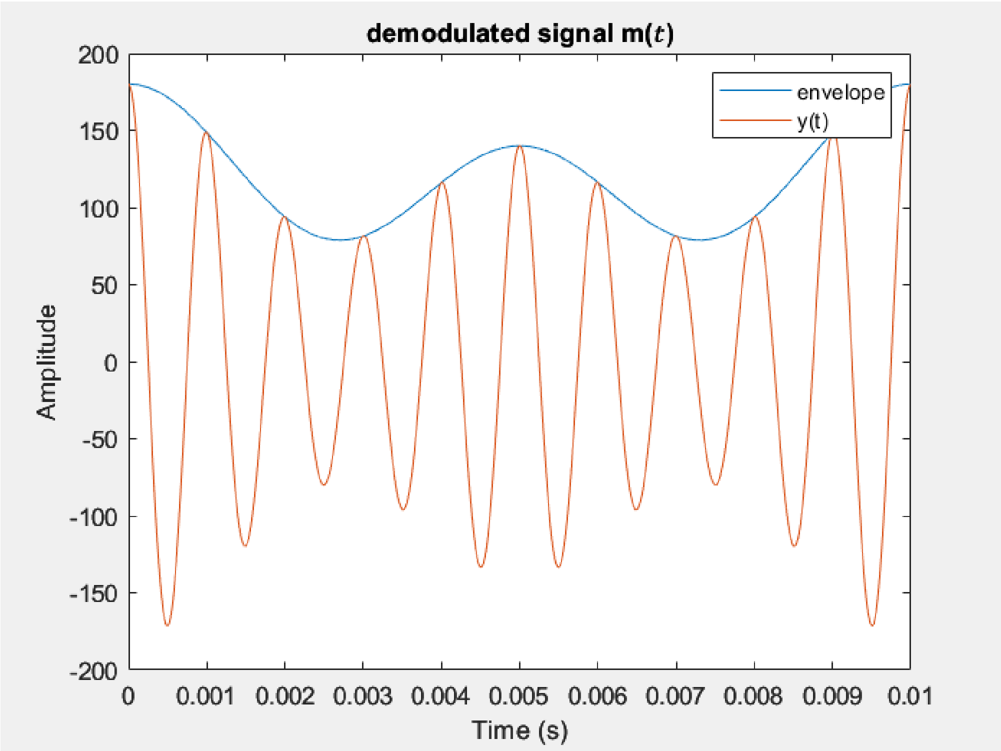


Figure 16 MATLAB Graph of envelope and y(t).

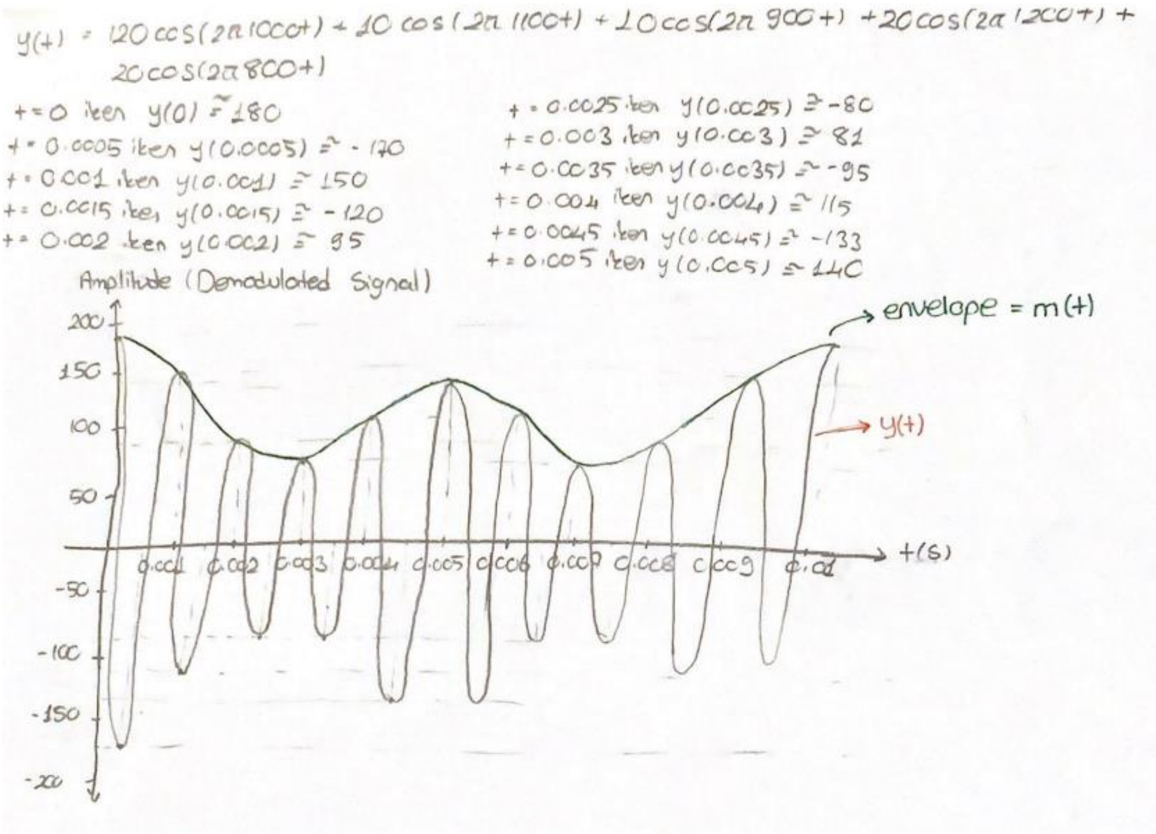


Figure 17 Analytic Graphic Drawn of y(t) and envelope.

To write the demodulated m(t) signal first we have to take it's Fourier transform than apply a shifting and plot it.

```
env = envelope(yt); % Envelope of y(t) which is demodulated m(t) signal
mtt = fft(env); % Fourier transform of m(t)


ff4shift = fftshift(mtt); % shifting m(f)

stem(fshift,abs(ff4shift)) % plotting m(f)

xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('Plot of the demodulated signal m(f)')
xlim([-1000,1000]);
```

```
plot(t,env); % plotting demodulated signal m(t)

xlabel('Time (s)')
ylabel('Magnitude')
title('Plot of the demodulated signal m(t)')
```

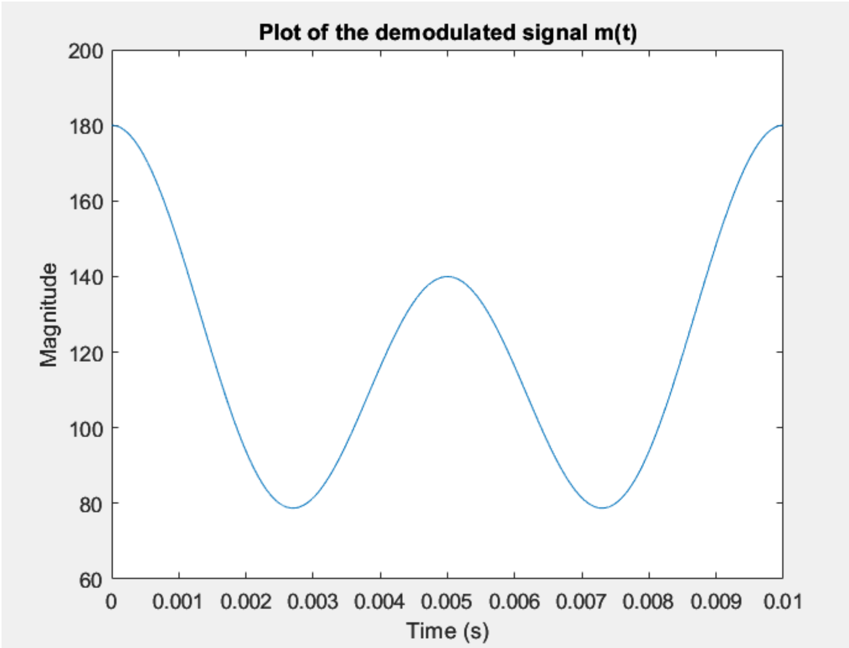


Figure 18 Demodulated signal m(t)

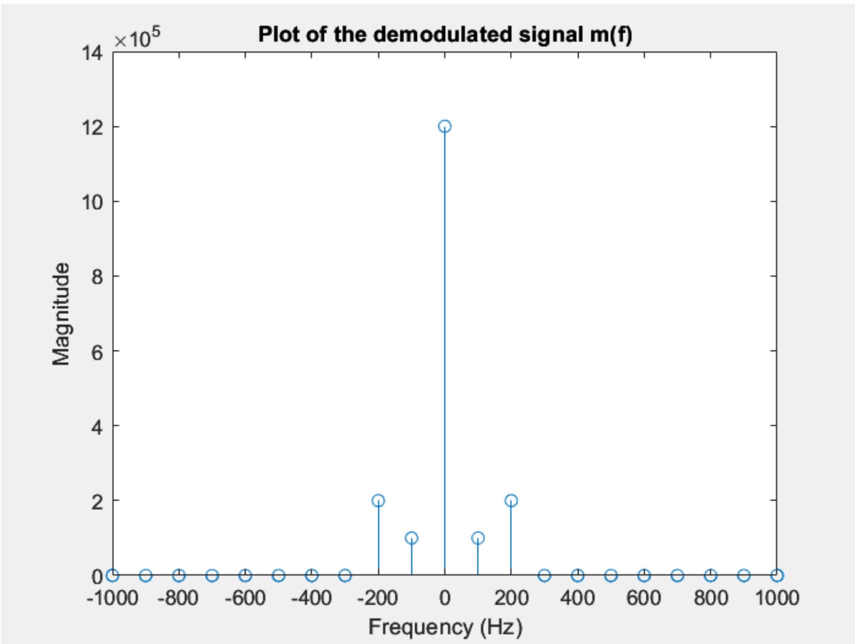


Figure 19 Demodulated signal m(f)

At frequency zero there is an DC component. If we ignore that original message signal will be achieved with only an amplitude difference.

Conclusion

In conclusion, the investigation into the Double-Sideband Large Carrier-Amplitude Modulation (DSB-LC-AM) system with a non-linear device and bandpass filter has provided valuable insights into signal processing and communication. The project's objectives, ranging from signal plotting to frequency domain analysis and demodulation, have been successfully addressed.

In this project, it was concluded that there was only an amplitude difference between the demodulated signal and the original signal. Original message signal is actually hidden in the envelope of output signal. That can be observed from the time domain graph of output signal. This makes demodulation particularly easy. Bandpass filter applied in the frequency domain and shown. Detailed graphics of the signals obtained for each step were draw in the time and frequency domain. All analytics and programming solutions are compatible with each other.

References

https://www.mathworks.com/help/matlab/math/fourier-transforms.html

https://www.mathworks.com/help/matlab/creating_plots/using-high-level-plotting-functions.html

https://www.mathworks.com/help/matlab/ref/fft.html

https://www.mathworks.com/help/signal/ref/envelope.html

https://www.mathworks.com/help/signal/ug/envelope-extraction-using-the-analytic-signal.html