GEBZE TECHNICAL UNIVERSITY

ELECTRONIC ENGINEERING

ELEC361

ANALOG COMMUNICATION SYSTEMS

MATLAB PROJECT ASSIGNMENT-2

Beyzanur Cam – 210102002037

Introduction

In this project, the use of the FM Modulation 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.

1) Plot the message signal. Obtain and plot the magnitude spectrum of the message signal.

Since the message signal is rectangular wave between 0 and 4 it has been defined with basic ranges that changes its value from 1 to -1 within the given ranges. Here in the code below Fs is sampling value which means it generates that much value in given range -2<t<4. To plot message signal in frequency domain its Fourier transform is calculated with (fft) command and (fftshift) command is applied to shift the values so the graph look more accurate. Also, message signals Fourier transform includes two sinc waves added with phase difference. That's the reason why two peaks seen in Figure 2.

```
Fs= 10000; %sampling frequency
t = -2:1/Fs:6;
N = length(t);
f = (-N/2:N/2-1)*(1/(t(2)-t(1)))/N;

%defining message signal
mt = zeros(1,length(t));
mt(0<=t & t<=2) = 1;
mt(2<t & t<=4) = -1;

%fourier transform of m(t)
m = fft(mt);
mf = fftshift(m)./Fs;

%ploting message signal
figure;
subplot(2,1,1);
plot(t,mt);

xlabel('time (s)')
ylabel('|m(t)|')
title('Plot of the message signal')

%ploting message signals magnitude spectrum
subplot(2,1,2);
plot(f,abs(mf));

xlabel('f(Hz)')
ylabel('|M(f)|')
title('Magnitude Spectrum of the message signal')
```

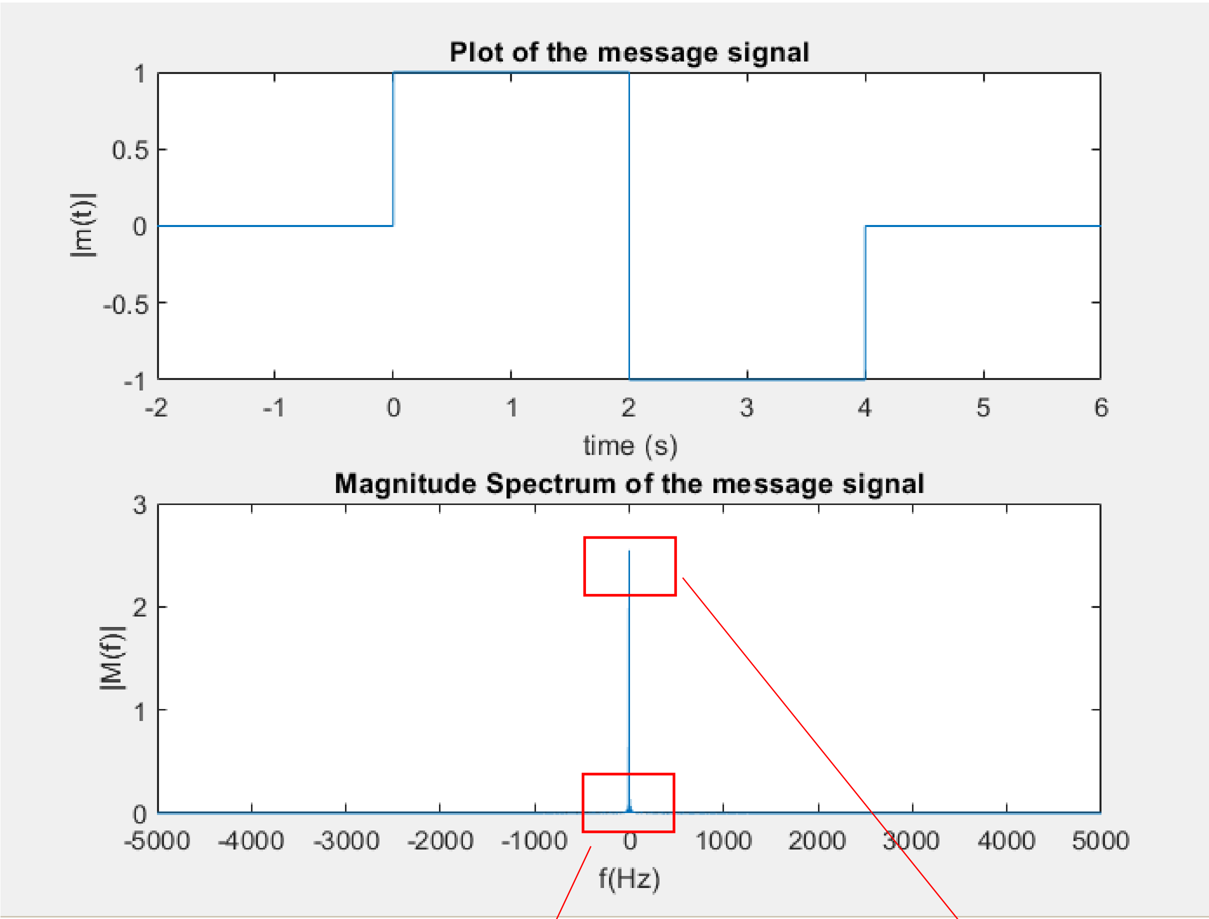

Figure 1. Message Signal Graphs in time and frequency domain.

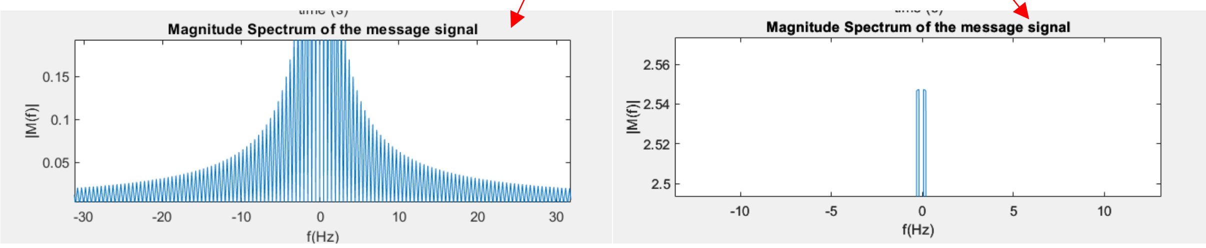

Figure 2. Zoom in to Magnitude Spectrum of the message signal for better ease.

2) Obtain and plot the time-domain representation and magnitude spectrum of the phase $\emptyset(t)$.

$\emptyset(t)$ is defined as the multiplication of k_f, 2π and convolution integral of the message signal. In order to take convolution integral of the message signal (cumtrapz) command of MATLAB is used. Then multiplication of all variables are given to a variable called fi and it's plotted like it's been before. For the magnitude spectrum its Fourier transform is taken, shifted, and plotted. Because message signal is rectangular its convolution integral will give a rectangular wave in time domain. Magnitude graph will be sinc^2 since the Fourier transform of a rectangular signal is sinc^2.

```
% defining Ø(t)
kf=50; %given in question
integral_mt = cumtrapz(t, mt);
fi=2.*pi.*kf.*integral_mt;

% plotting Ø(t)
figure;
subplot(2,1,1);
plot(t,fi);

xlabel('time (s)')
ylabel('|Ø(t)|')
title('Plot of the Ø(t)')

% Fourier transform of Ø(f)
fi_fourier=fft(fi);
fi_fourier_shifted=fftshift(fi_fourier);

%plotting Ø(f)
subplot(2,1,2);
plot(f,abs(fi_fourier_shifted)./Fs);

xlabel('f(Hz)')
ylabel('|Ø(f)|')
title('Magnitude Spectrum of the Ø(f)')
```

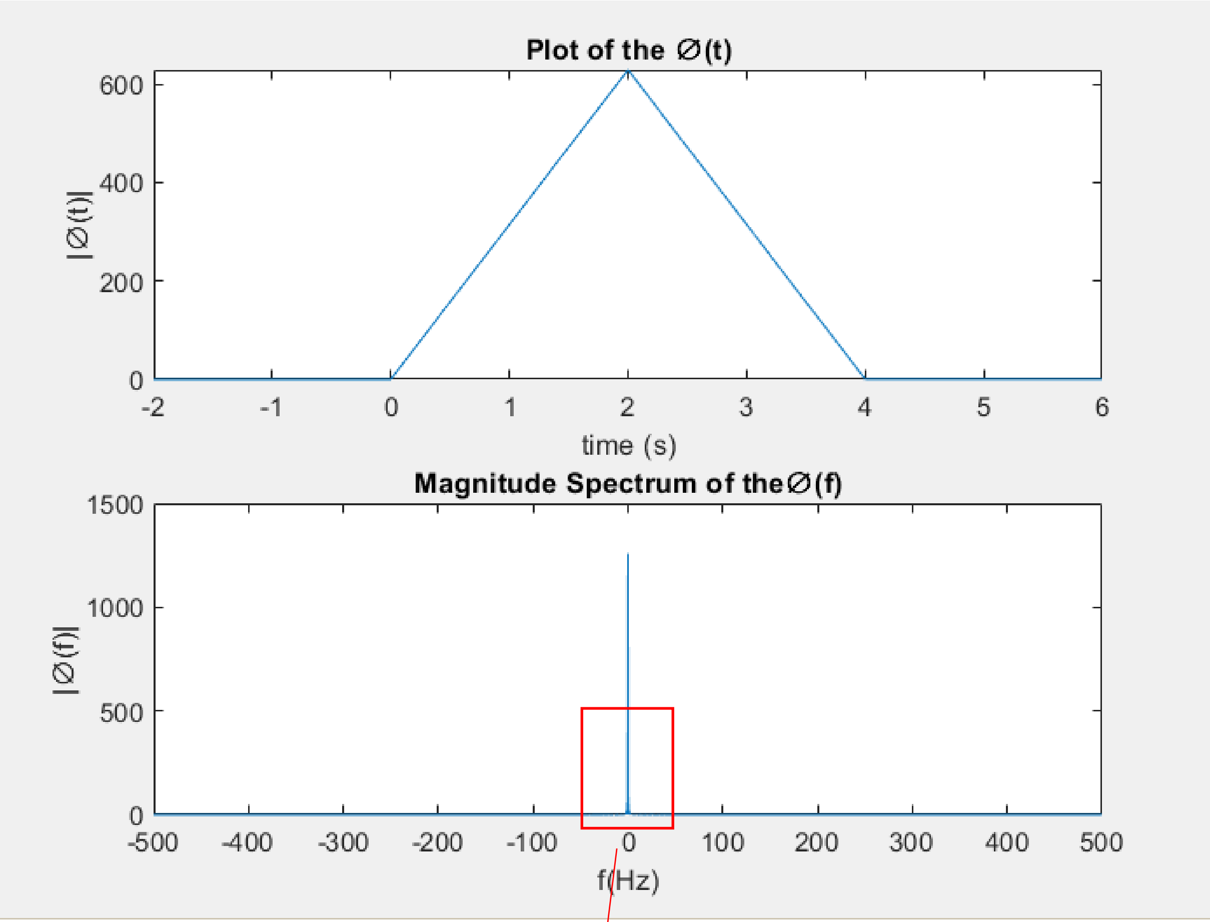

Figure 3. ∅ Graphs in time and frequency domain.

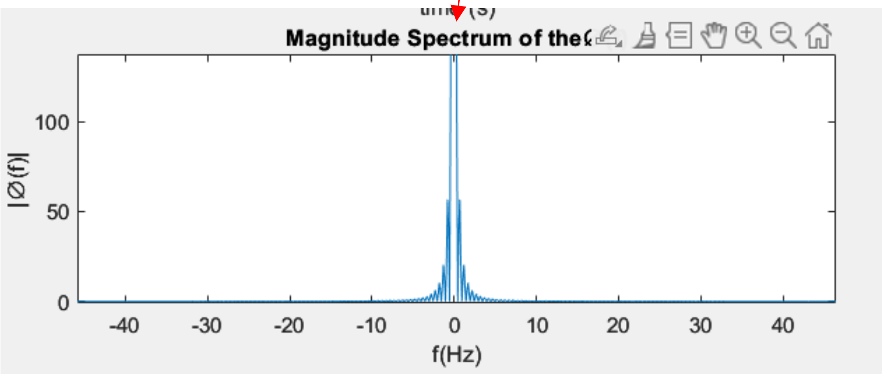

Figure 4. Zoom in to Magnitude Spectrum of the ∅(f) for better ease sight.

3) Obtain and plot the time-domain representation and magnitude spectrum of the modulated signal $y(t)$.

y(t) is defined and written with the given values and to plot it in frequency domain its Fourier transform is obtained and plotted. If the graph on the time domain is zoomed in (Figure 6.) towards to changing point which is 2 changes in frequency is visible. Changing point is 2, because message signals maximum value becomes -1 from 1. So, the instant frequency changes from like the equation below. Between 0 and 2 frequency increases (f_c+k_f), between 2 and 4 frequency decreases (f_c-k_f) according to carrier frequency 250Hz.

$$\boxed{f_i = f_c + k_f \max|m(t)|}$$

```
% Defining y(t)
yt=5.*cos((2.*pi.*250.*t)+fi);

% Fourier Transform of y(f)
y_fourier=fft(yt);
y_fourier_shifted=fftshift(y_fourier)./Fs; %normalizing the signal

%Plotting

figure;
subplot(2,1,1);
plot(t,yt);

xlabel('time (s)')
ylabel('|y(t)|')
title('Plot of the y(t)')

subplot(2,1,2);
plot(f,abs(y_fourier_shifted));

xlabel('f(Hz)')
ylabel('|y(f)|')
title('Magnitude Spectrum of the y(f)')
```

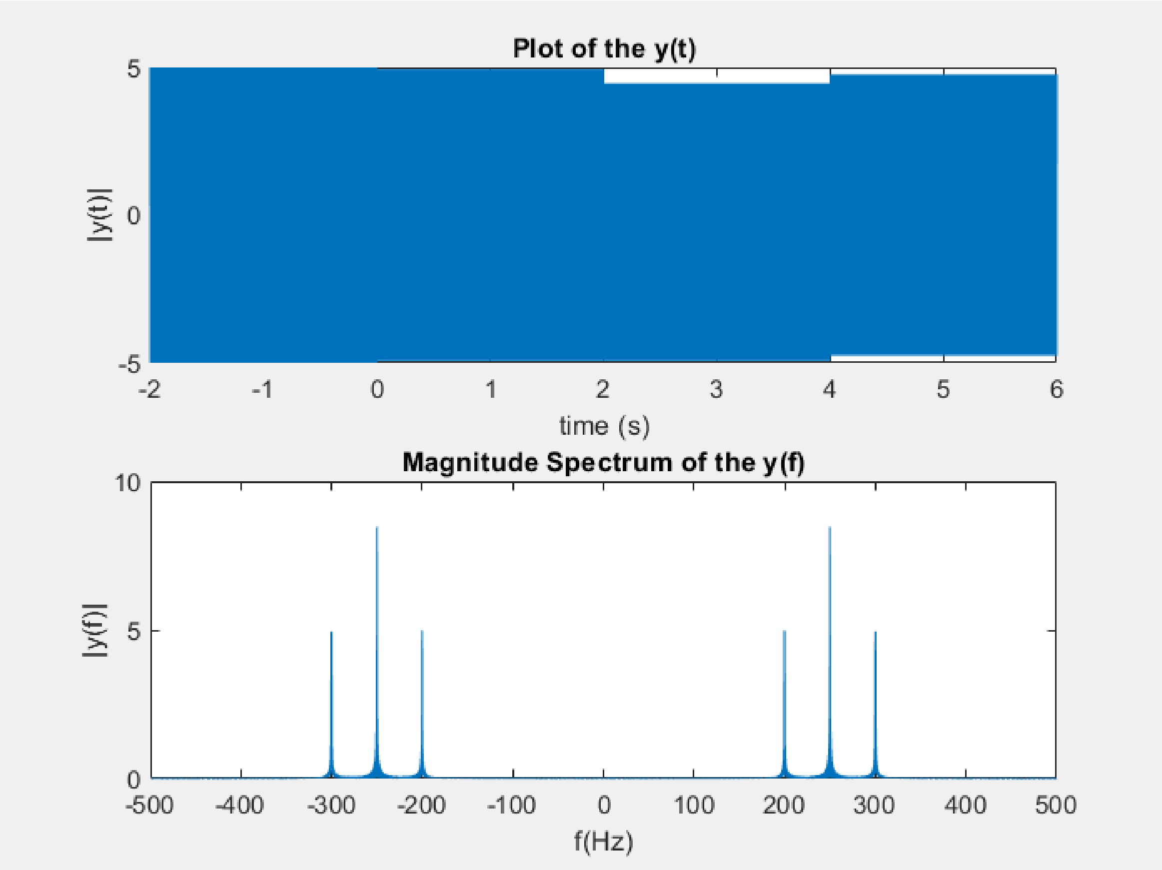

Figure 5. y(t) and y(f) Graphs.

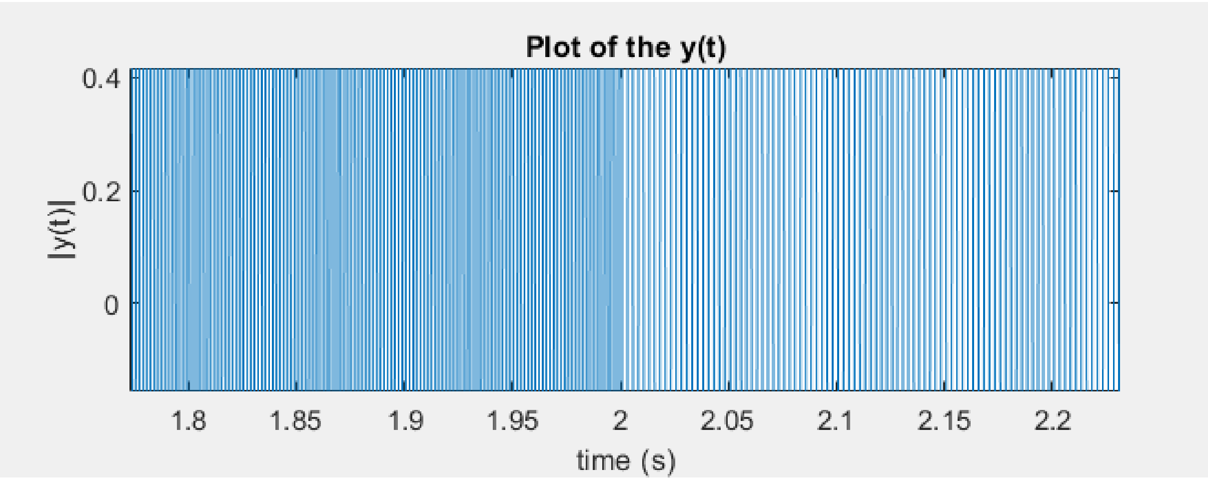

Figure 6. y(t) Graph zoomed in for better ease sight.

4) Obtain and plot the time-domain representation and magnitude spectrum of the signal at each step of the demodulator. (Do the filtering in frequency domain.) Comment on the signal at each step. Compare the message signal and demodulated signal, and comment on them.

   4.a) Demodulation Process Signal After Differentiator

When the signal is differentiated integral of message signal comes to beginning of y(t) with other constant values that comes from differentiation of y(t). And message signal appears at the envelope of differentiated signal. And a sin wave oscillates between this envelope. In code section y(t) is differentiated and plotted. For its Fourier transform frequency ranges are adjusted and plotted like before.

```
%demodulation
y_diff= diff(yt)./(t(2)-t(1));

% differantiator frequency domain
y_diff_f=fft(y_diff);
y_diff_ff=fftshift(y_diff_f)./Fs;

%graph
figure;
subplot(2,2,1);
plot(t(1:end-1),y_diff);

xlabel('time (s)')
ylabel('|ydiff(t)|')
title('Plot of the signal after Differentiator')

% Assuming length(y_diyot_ff) is the correct length for the frequency
domain
f_diff = (-length(y_diff_ff)/2:length(y_diff_ff)/2-1) * (1/(t(2)-
t(1)))/length(y_diff_ff);

% Plotting frequency domain of y_diff_ff
subplot(2,2,3);
plot(f_diff, abs(y_diff_ff));

xlabel('f(Hz)')
ylabel('|ydiff(f)|')
title('Magnitude Plot of the signal after Differantiator ')
```

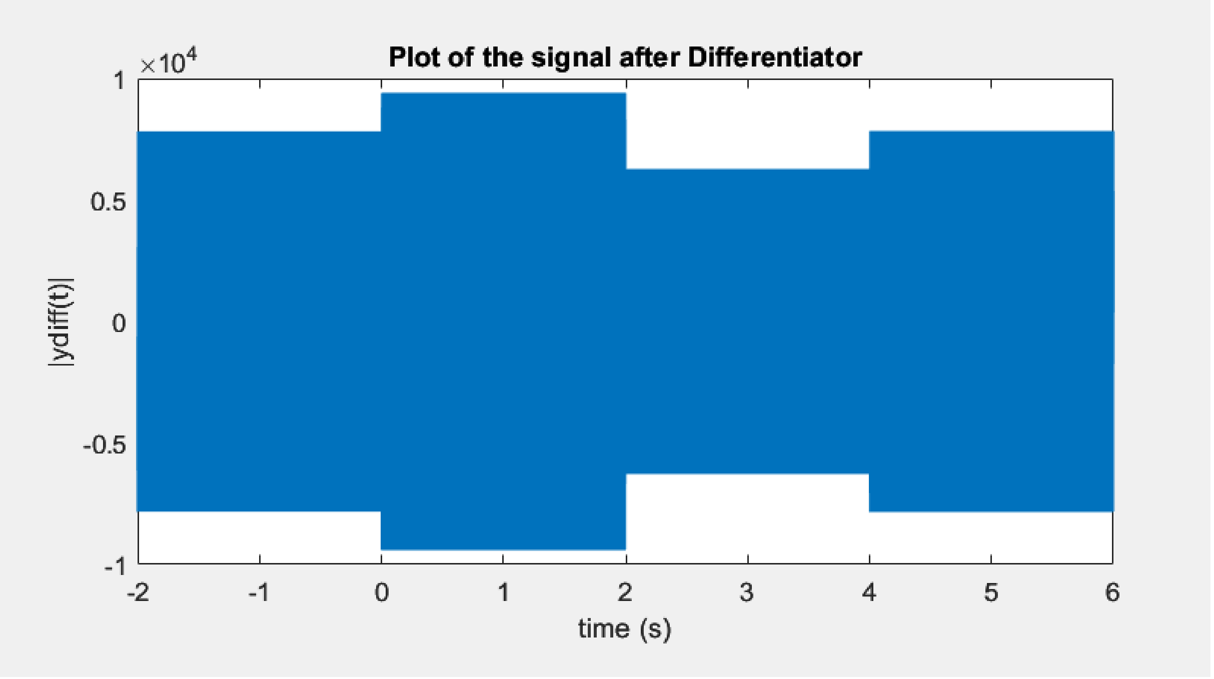

Figure 7.Graph of Signal after Differentiator in time Domain.

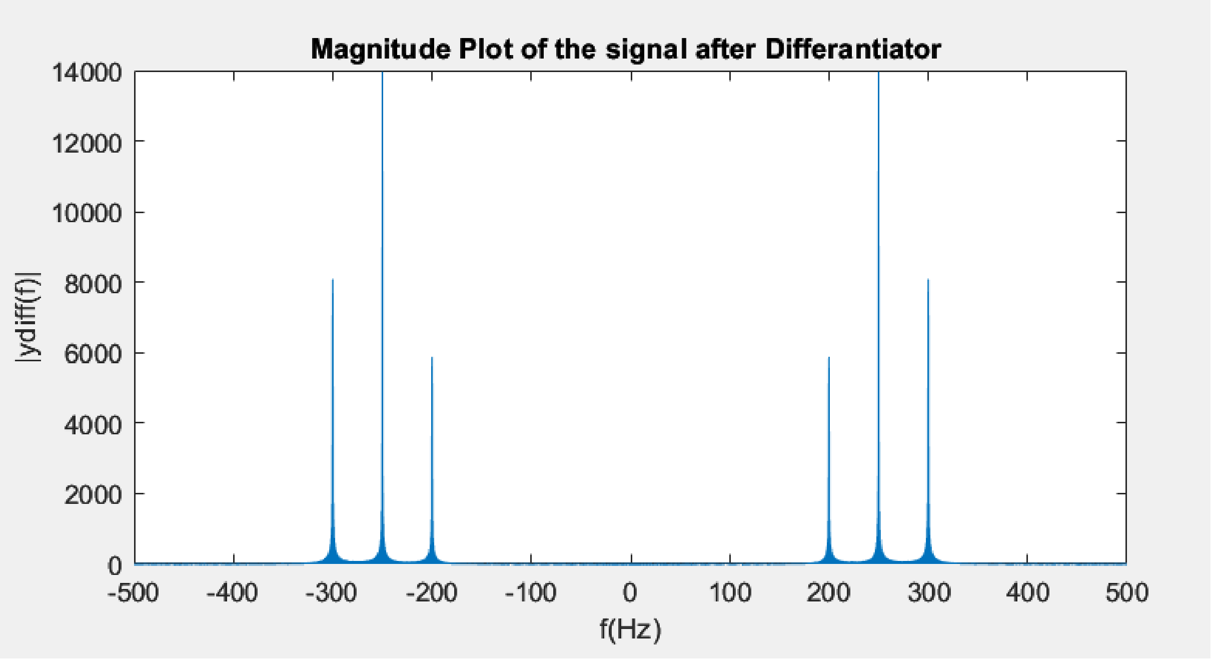

Figure 8. Graph of Signal after Differentiator in frequency Domain.

4.b) Demodulation Process Signal After Diode

Adding a diode cut of the negative parts of signal. Code creates a variable called y_diyot whose value is the maximum of 0 and the value of another variable y_diff. After that to plot it in the frequency domain it Fourier transform is obtained and plotted like before.

```
%diyottan geçirme
y_diyot = max(0, y_diff);

%y_diyot frekans domaininde
y_diyot_f= fft(y_diyot);
y_diyot_ff= fftshift(y_diyot_f)./Fs;

subplot(2,2,2);
plot(t(1:end-1),y_diyot);
xlabel('time (s)')
ylabel('|ydiyot(t)|')
title('Plot of the signal after Diyot')

% Assuming length(y_diyot_ff) is the correct length for the frequency
domain
f_diyot = (-length(y_diyot_ff)/2:length(y_diyot_ff)/2-1) * (1/(t(2)-
t(1)))/length(y_diyot_ff);

% Plotting frequency domain of y_diyot_ff
subplot(2,2,4);
plot(f_diyot, abs(y_diyot_ff));

xlabel('f(Hz)')
ylabel('|ydiyot(f)|')
title('Magnitude Plot of the signal after Diyot ')
```

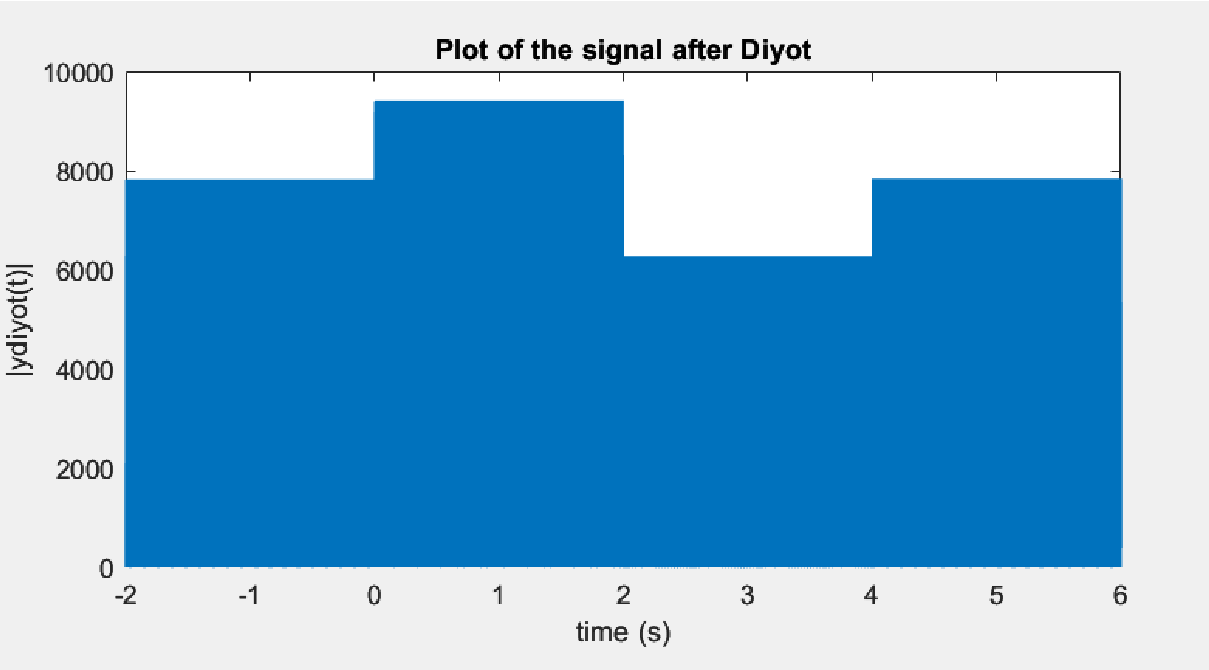

Figure 9. Graph of Signal after Diode in time Domain.

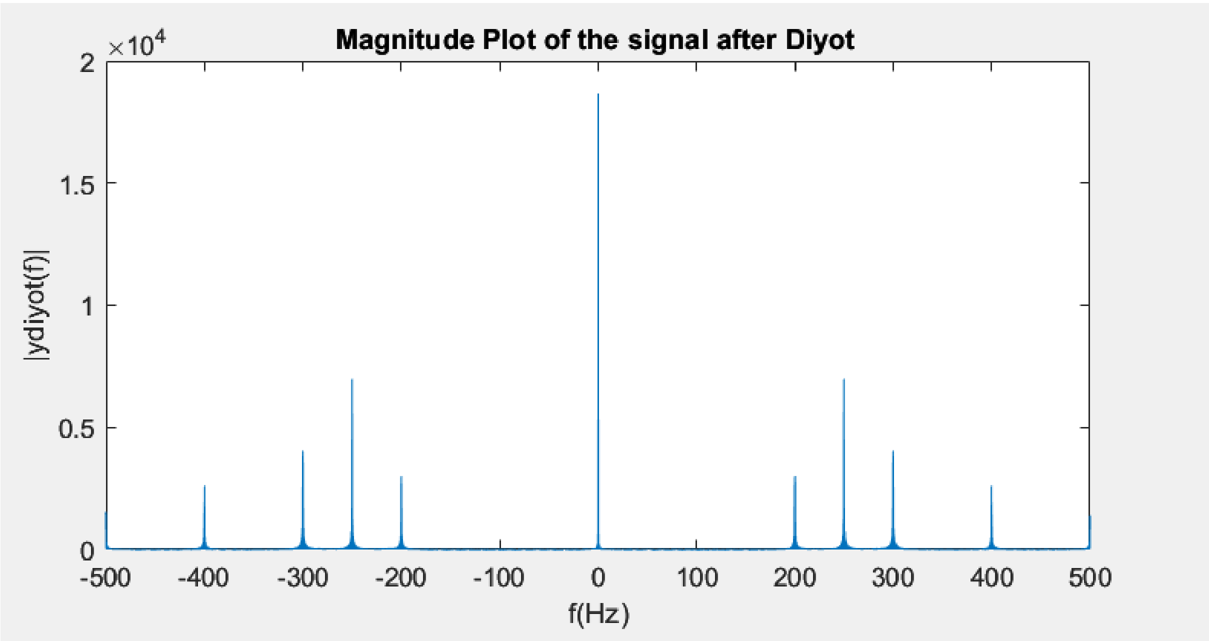

Figure 10.Graph of Signal after Diode in frequency Domain.

4.c) Demodulation Process Signal After Low Pass Filter

In order to obtain an envelope detector, detector function must follow the y_diode function graph if its value is smaller. Else it will follow the drop_off function. An array with filled one value is created named y_envelope and its changed according to if else condition. RC values are calculated in analytical solutions. After the function and its Fourier transformed version is plotted like before.

Drop_off = $e^{(-1*t/RC)}$

```
%RC Filter
R=200000;
C=1e-6;
tao=R.*C;
drop_off = exp(-((t(2)-t(1))./tao));

y_envelope=ones(size(y_diyot));

for i= 2: length(y_diyot)
    if y_envelope(i-1) < y_diyot(i)

        y_envelope(i) = y_diyot(i);

    else
        y_envelope(i)=drop_off.*y_envelope(i-1);
    end
end

%y(t)türevinde eğer envelopdaki değer y(tin türevinden(y_diyot) küçükse envelopa
%eşit değilse ve büyükse exponansiyel formul kadar azalcak (mantığım)

% plot y_envelope
figure;
plot(t(1:end-1), y_envelope);
xlabel('time (s)')
ylabel('|yenvelope(t)|')
title('Plot of the Envelope of y(t)')

%y_envelope fourier

y_envelope_f=fft(y_envelope);
y_envelope_fft= fftshift(y_envelope_f)./Fs;

figure
plot(f_diyot, abs(y_envelope_fft));
xlabel('f(Hz)')
ylabel('|yenvelope(f)|')
title('Magnitude Spectrum Plot of the y envelope')
```

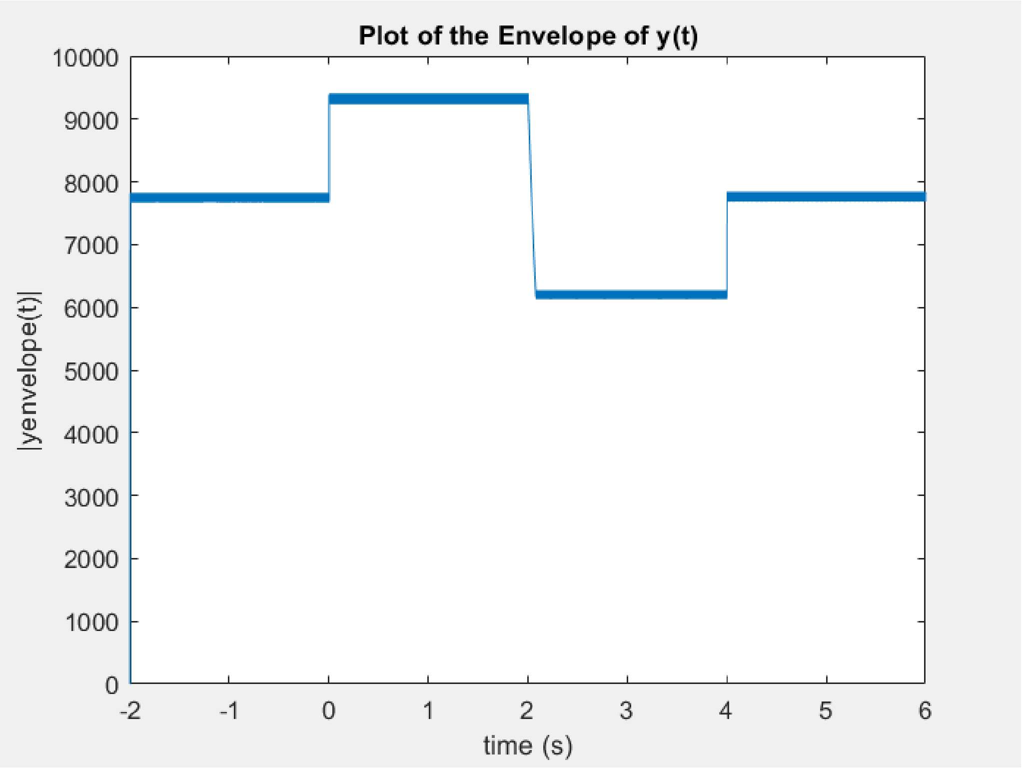

Figure 11. Envelope Signal Graph.

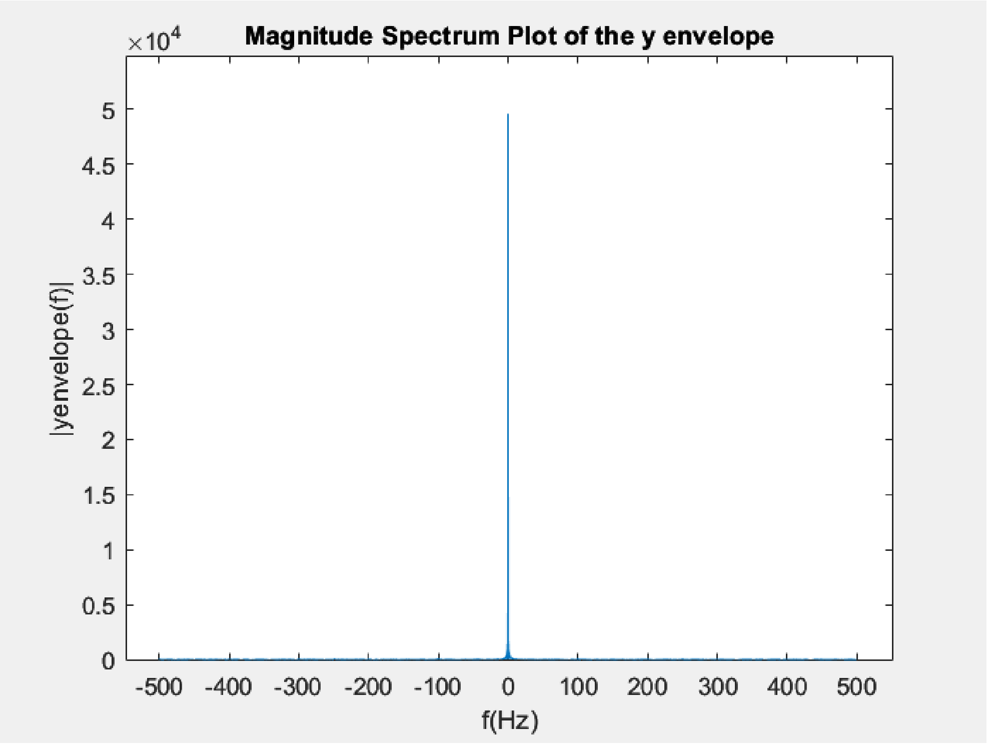

Figure 12. Magnitude Spectrum of Envelope Signal.

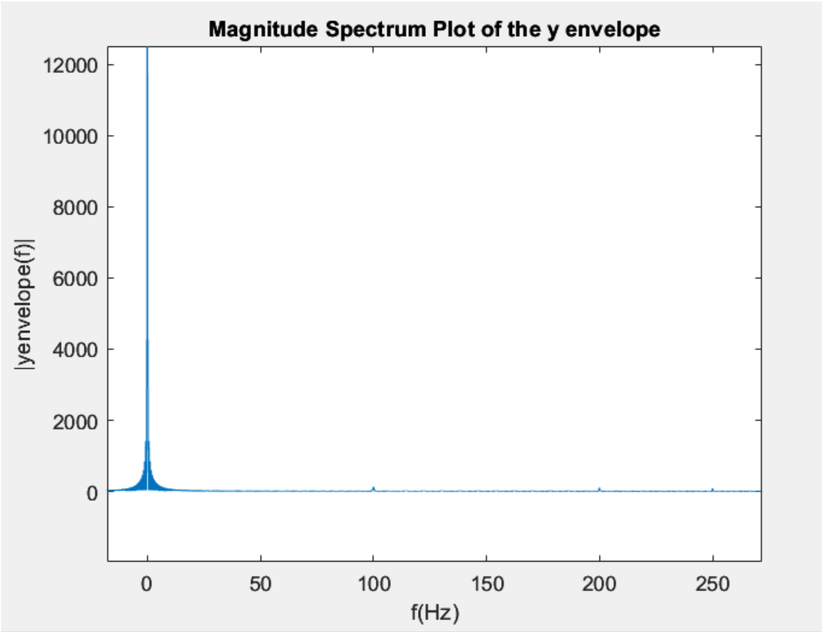

Figure 13. Close-up Look to Figure12.

Lowpass filter is used from the previous assignment. Because the type of the filter was bandpass to adapt it center frequency is given as zero. So, it acts as a lowpass filter. After filtering the signal its inverse Fourier is taken to be represented in time domain. If the graph in the frequency domain is zoomed in it can be seen that the values after 100Hz are gone.

```
%Low pass filter

center_frequency = 0; % Center frequency in Hz
bandwidth = 101; % Bandwidth in Hz

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

% Apply bandpass filter
y_envelope_filtered = zeros(size(y_envelope_fft));
y_envelope_filtered(abs(f) >= lower_cutoff & abs(f) <= upper_cutoff) =
y_envelope_fft(abs(f) >= lower_cutoff & abs(f) <= upper_cutoff);

% inverse fourier
y_envelope_filtered_time=ifft(ifftshift(y_envelope_filtered));

%plotting
figure
subplot(2,1,1)
plot(t(1:end-1),y_envelope_filtered_time);
xlabel('time (s)')
ylabel('|y envelope- filtered(t)|')
title('Plot of the Filtered Envelope of y(t)')

subplot(2,1,2)
plot(f_diyot,y_envelope_filtered);
xlabel('f(Hz)')
ylabel('|y envelope filtered(f)|')
title('Magnitude Spectrum Plot of the y envelope filtered')
```

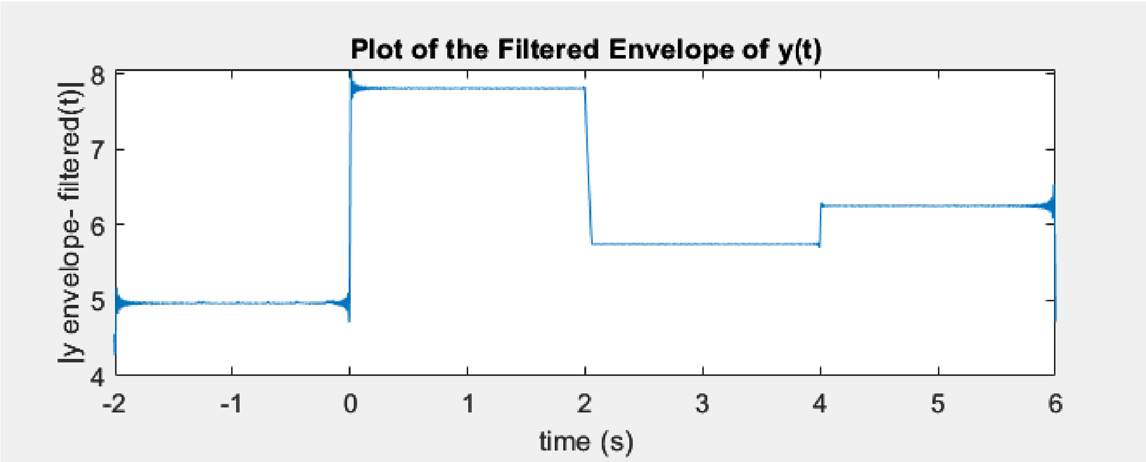

Figure 14. Filtered Envelope Signal Graph.

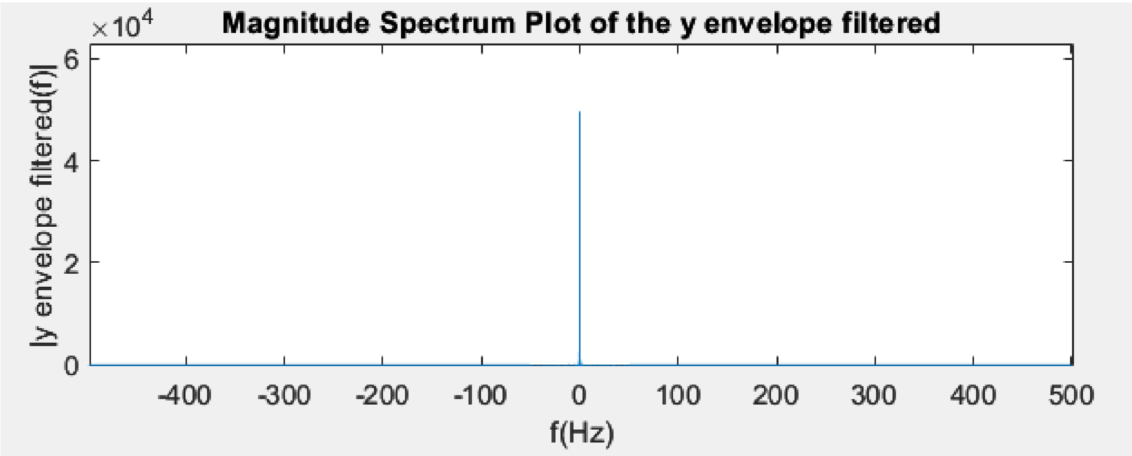

Figure 15. Filtered Signal in Frequency Domain

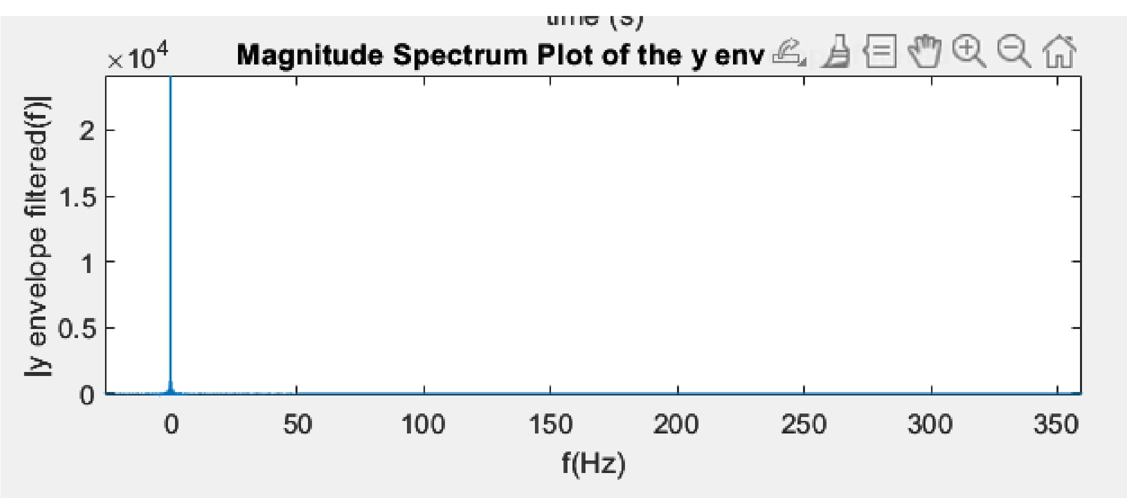

Figure 16. Close-up Look to Figure8.

4.d) Demodulation Process Signal After DC blocking

In order to dc block enveloped signal its index which holds the dc value is found and it's equaled to zero. Later its inverse Fourier transform is taken, and message signal is obtained back. And dc blocked signal is plotted both time and frequency domains like before.

```
% DC Blocking
y_envelope_filtered_dcblocked = y_envelope_filtered;
dc_index = find(f_diyot == 0); % Find the index corresponding to DC component

% Set the DC component to zero
y_envelope_filtered_dcblocked(dc_index) = 0;

% inverse fourier after DC blocking
y_envelope_filtered_dcblocked_time =
ifft(ifftshift(y_envelope_filtered_dcblocked));

% Plotting
figure
subplot(2,1,1)
plot(t(1:end-1), y_envelope_filtered_dcblocked_time);
xlabel('time (s)')
ylabel('|y-DC blocked(t)|')
title('Plot of the Filtered and DC Blocked Envelope of y(t)')

subplot(2,1,2)
plot(f_diyot, abs(y_envelope_filtered_dcblocked));
xlabel('f(Hz)')
ylabel('|y-DC blocked(f)|')
title('Magnitude Spectrum Plot of the Filtered and DC Blocked y envelope')
```

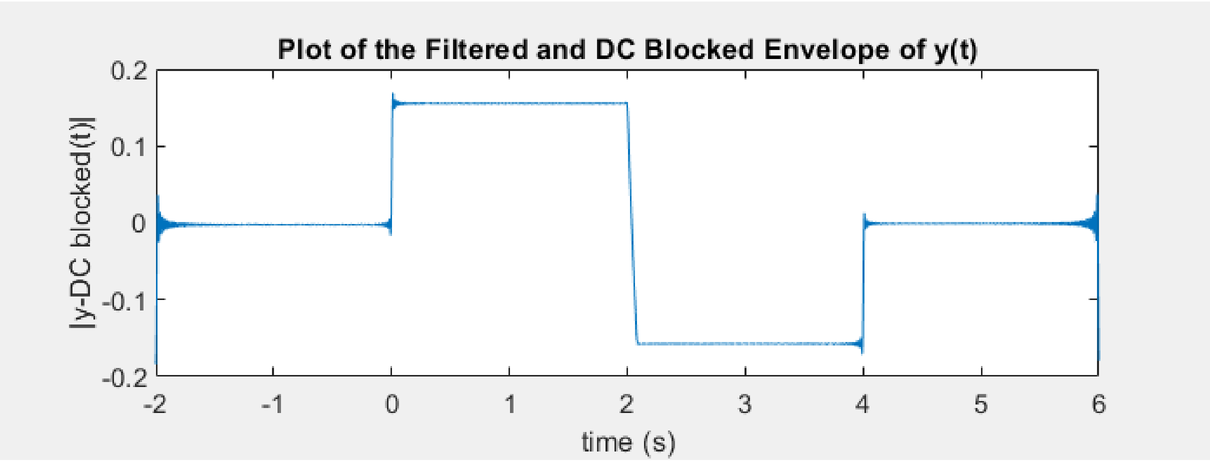

Figure 17. Dc Blocked Signal.

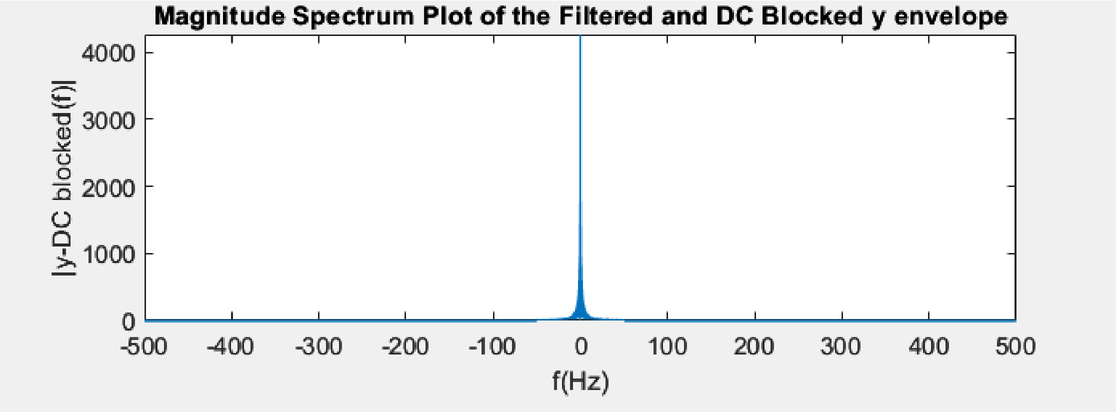

Figure 18. Magnitude Spectrum of DC Blocked Signal.

Comparison

First message signal and output of the demodulation of message signal is similar in shape. But amplitudes are different. In some places there are ripples observed reason of it is while applying a low pass filter to our signal some values in higher frequencies are lost. We know that Fourier transform of rectangular wave is sinc function and sinc function has values goes to infinity. Because these values are lost. Exact signal can't be obtained back. But the essence of the signal is obtained back successfully.

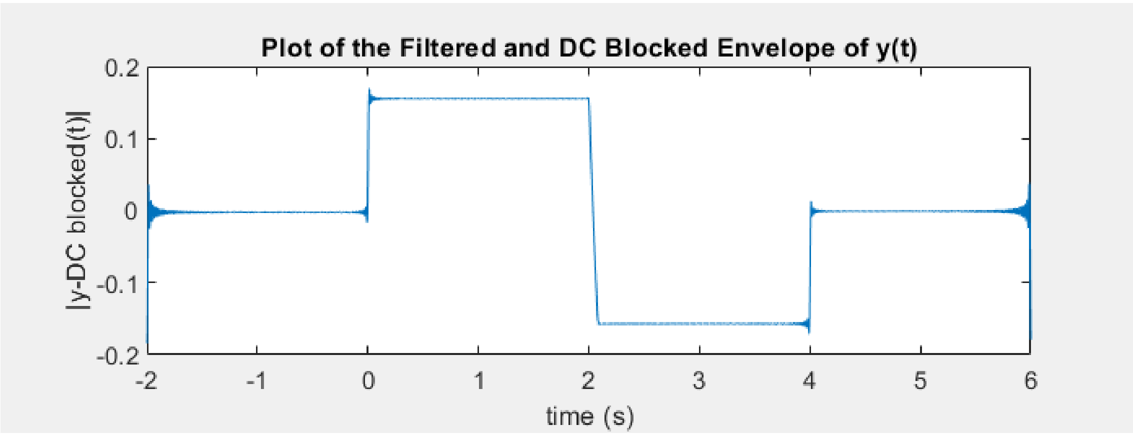

Figure 19. m(t) Signal Obtained from Demodulation.

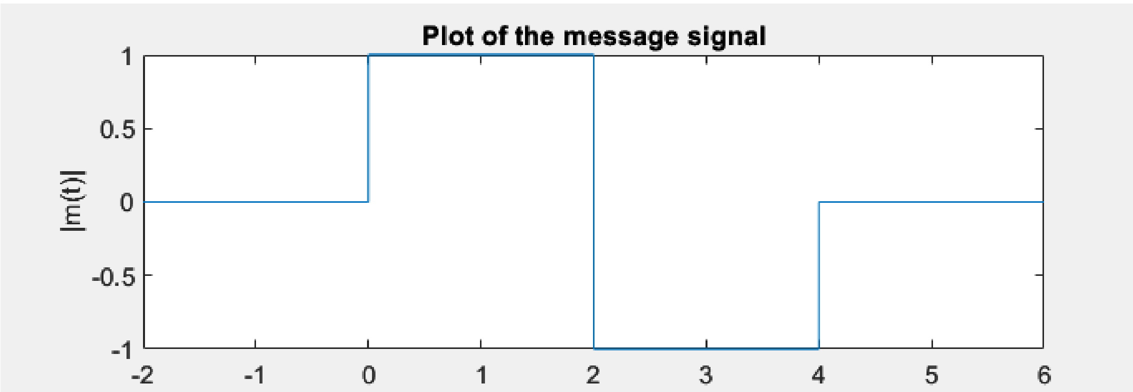

Figure 20. Original Message Signal m(t).

Conclusion

In conclusion, this report delved into the analysis and simulation of a frequency modulation (FM) system using a message signal m(t) and a carrier signal c(t). The modulation process generated the FM signal y(t), which was subsequently demodulated through a series of steps including differentiation, diode demodulation, low-pass filtering, and DC blocking.

The message signal m(t) was successfully plotted in the time domain, revealing its intricate waveform. The magnitude spectrum of the message signal was also obtained through Fourier transform, providing insights into its frequency components.

The phase function φ(t) was derived from the message signal and examined in both the time domain and frequency domain. This allowed for a comprehensive understanding of the phase modulation imposed on the carrier signal.

The modulated signal y(t) was formed by combining the carrier signal with the phase information. The time-domain representation and magnitude spectrum of y(t) were plotted, illustrating the impact of the modulation process on the signal.

In conclusion, this study has shed light on the intricate dynamics of FM modulation and demodulation. The simulation results aligning with analytical calculations validate the accuracy of the process. The demodulator, while successfully recovering the essence of the message signal, introduces some distortions and losses, highlighting the importance of careful consideration in demodulation system design. This report serves as a valuable resource for understanding and optimizing FM communication systems.