GEBZE TECHNICAL UNIVERSITY

ELECTRONICS ENGINEERING

ELM 361

Analog Communication Systems

PROJECT #1

Prepared by
1) 1801022037 Ömer Emre POLAT

1. Introduction

In this project we will analyze a communication system with AGF filter. The analyzing will be done on MATLAB by simulation and the results will be compared to hand calculations.

2. Project Design

2.1. Writing the MATLAB code

First, we will write a MATLAB code to calculate and graph the values that we want to graph. The code will be written using the functions given in the project 1 pdf.

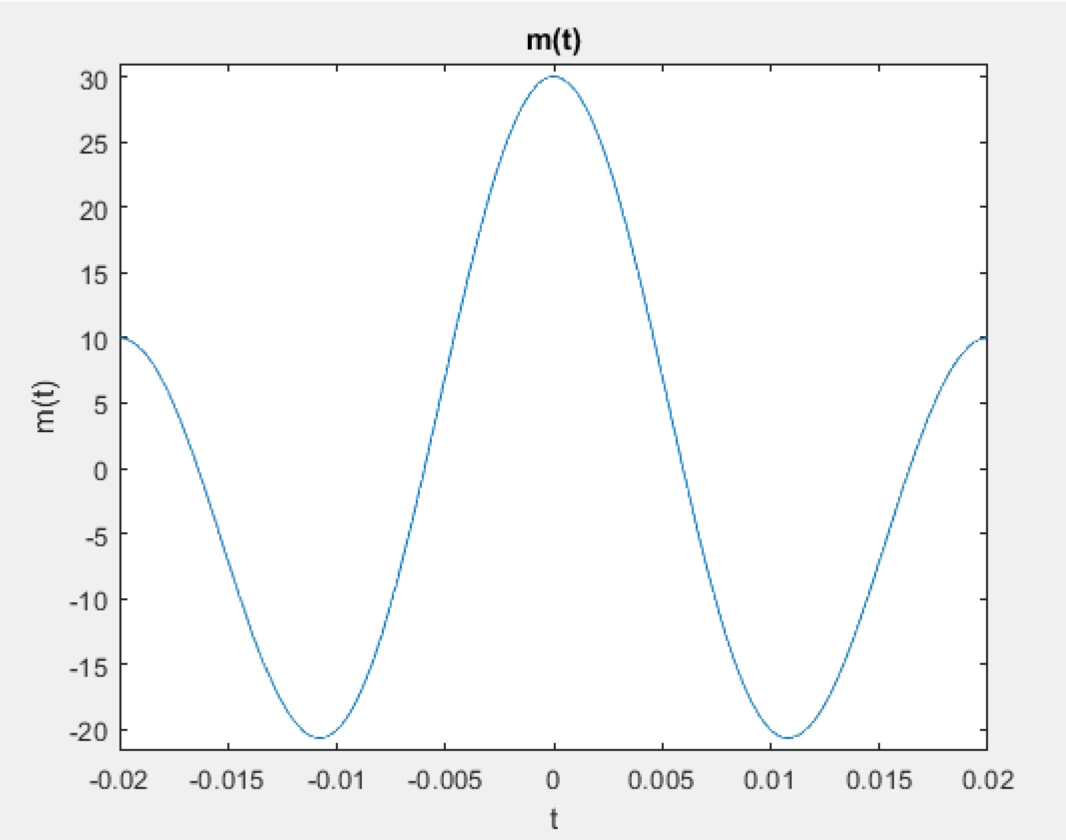
```
%Ömer Emre POLAT
%1801022037
%Analog Proje 1

clear all; close all; clc %clear variables close plots and clear console

%///////////SETTING VARIABLES//////////
fsample = 20000; %sample frequency
fm = 50; %highest frequency of m
fc = 250; %highest frequency of c
tm = 1/fm; %period of m
tc = 1/fc; %period of c

%///////////SETTING X AXIS VARIABLES//////////
t = linspace(-1/2,1/2,fsample); %vector of t
n = length(t); %signal length
f = linspace(-n/2, n/2-1, fsample) * (fsample/n); %frequency vector

%///////////SETTING Y AXIS VARIABLES//////////
m = (10*cos((fm/2)*2*pi*t) + 20*cos(fm*2*pi*t)); %vector of m
c = (100*cos(fc*2*pi*t)); %vector of c
y = m .* c; %modulated vector
d = (1*cos(fm*2*pi*t)); %demodulator vector
yd = y .* d; %demodulated vector

%///////////PLOTTING m(t)//////////
plot(t, m); %plot t and m vectors
hold on; %hold on so that plot does not disappear
title("m(t)");
xlabel("t");
ylabel("m(t)");
axis([-tm tm (min(m)-1) (max(m)+1)]); %graph scaling

%///////////CALCULATING M(f)///////////
mfft = fftshift(fft(m));
mfft = abs(mfft/length(m));

%///////////PLOTTING M(f)//////////
figure(); %open new plot figure
plot(f, mfft); %plot f and mfft vectors
hold on; %hold on so that plot does not disappear
title("M(f)");
xlabel("f");
ylabel("M(f)");
axis([(-fm-10) (fm+10) (min(mfft)-1) (max(mfft)+1)]); %graph scaling
```

```
%///////////PLOTTING Y(f)///////////
figure(); %open new plot figure
plot(t, y); %plot t and y vectors
hold on; %hold on so that plot does not disappear
title("y(t)");
xlabel("t");
ylabel("y(t)");
axis([-tm tm (min(y)-1) (max(y)+1)]); %graph scaling

%///////////CALCULATING Y(f)///////////
yfft = fftshift(fft(y));
yfft = abs(yfft/length(y));

%///////////PLOTTING Y(f)///////////
figure(); %open new plot figure
plot(f, yfft); %plot f and yfft vectors
hold on; %hold on so that plot does not disappear
title("Y(f)");
xlabel("f");
ylabel("Y(f)");
axis([(-(fm+fc)-50) ((fm+fc)+50) (min(yfft)-50) (max(yfft)+50)]); %graph
scaling

%///////////PLOTTING yd(t)///////////
figure(); %open new plot figure
plot(t, yd); %plot t and yd vectors
hold on; %hold on so that plot does not disappear
title("yd(t)");
xlabel("t");
ylabel("yd(t)");
axis([-tm tm (min(yd)-100) (max(yd)+100)]); %graph scaling

%///////////CALCULATING Yd(f)///////////
ydfft = fftshift(fft(yd));
ydfft = abs(ydfft/length(yd));

%///////////PLOTTING Yd(f)///////////
figure(); %open new plot figure
plot(f, ydfft); %plot f and ydfft vectors
hold on; %hold on so that plot does not disappear
title("Yd(f)");
xlabel("f");
ylabel("Yd(f)");
axis([(-(fm+fc)-100) ((fm+fc)+100) (min(ydfft)-50) (max(ydfft)+50)]);
%graph scaling

%///////////CALCULATING AGF///////////
lpfilter = @(x) (1.*((-fm) <= x & x <= (fm))); %low pass filter function
agf = lpfilter(f); %filtered values vector

%///////////USING AGF///////////
zfft = ydfft .* agf; %demodulated signal passing through the AGF
z = ifftshift(ifft(zfft)); %inverse fourier of the zfft to z

%///////////PLOTTING z(t)///////////
figure(); %open new plot figure
plot(t, z); %plot t and z vectors
hold on; %hold on so that plot does not disappear
title("z(t)");
```

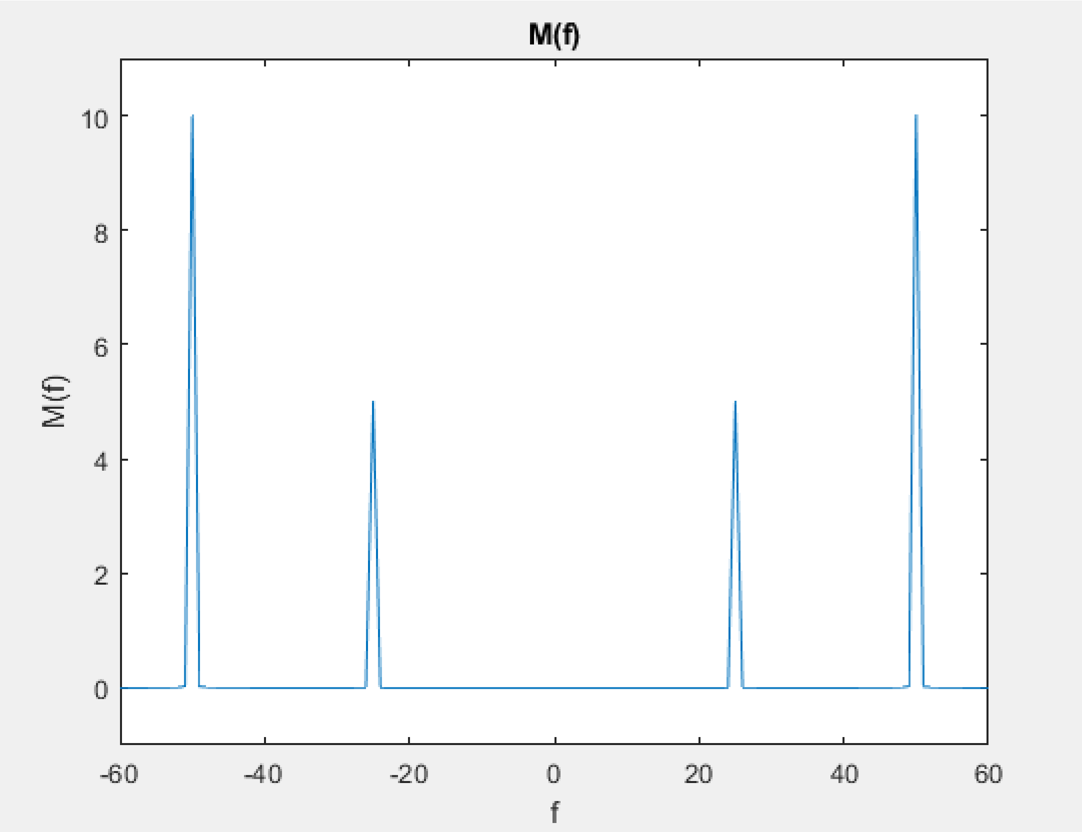
```
xlabel("t");
ylabel("z(t)");
axis([-tm tm (min(z)) (max(z))]); %graph scaling

%//////////PLOTTING Z(f)//////////
figure(); %open new plot figure
plot(f, zfft); %plot f and zfft vectors
hold on; %hold on so that plot does not disappear
title("Z(f)");
xlabel("f");
ylabel("Z(f)");
axis([(-(fm+fc)-100) ((fm+fc)+100) (min(zfft)-50) (max(zfft)+50)]); %graph
scaling
```

The code is written with comments on almost every line and the comments explain what each line does. The fftshift function was used to show the negative part of the spectra too.

2.2 Plotting of the m(t) and M(f) Functions



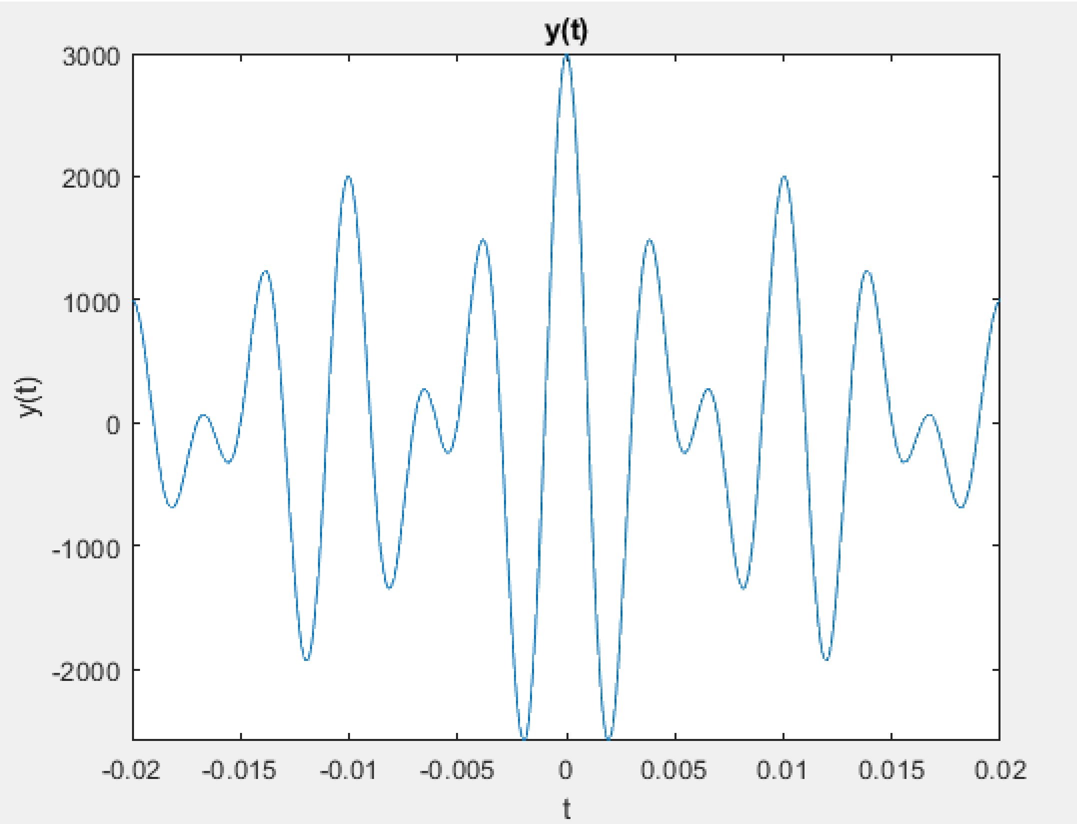
The m(t) graph is shown above. The graph is expected since the m(t) function consists of two cosine waves with different frequencies



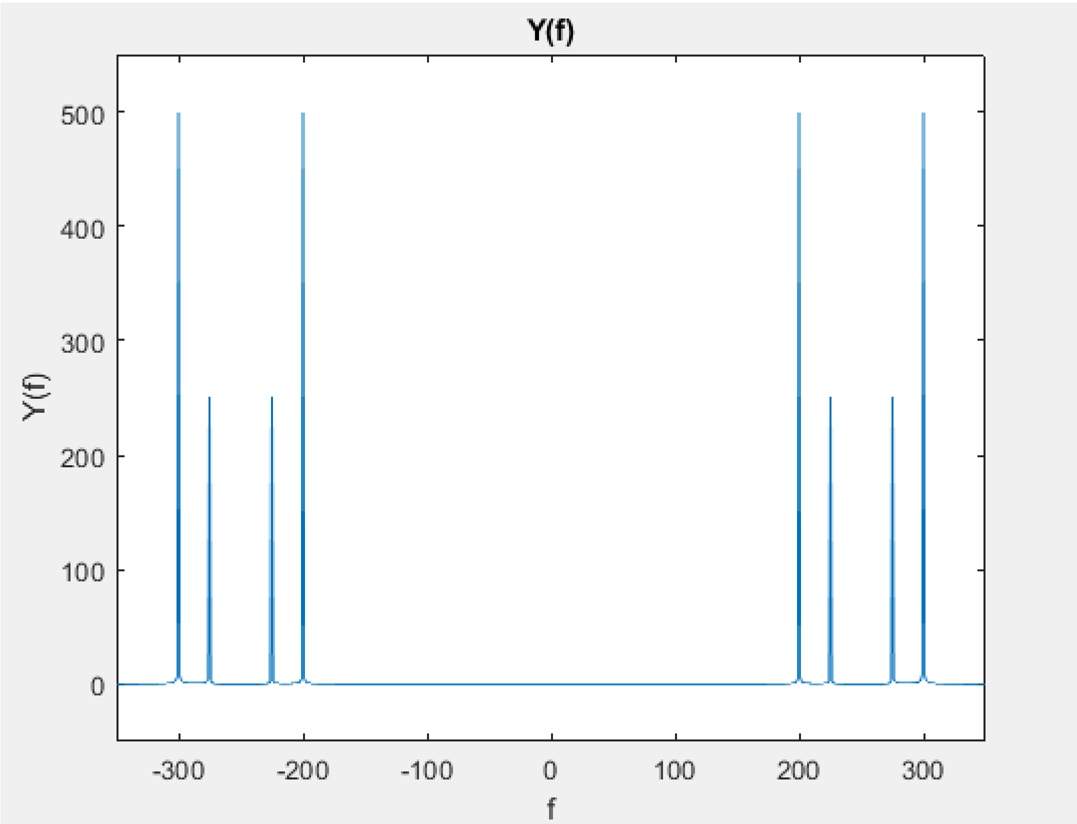
The M(f) fast fourier transform of m(t) graph is given above. The graph is expected because there are two cosine waves with frequencies 25 and 50. The spikes on -+25 and -+50 are the signs that there are signals that have those frequencies inside the m(t) signal.

2.3 Plotting of the y(t) and Y(f) Functions

The y(t) function is the modulated m(t) signal which is m(t) * c(t).



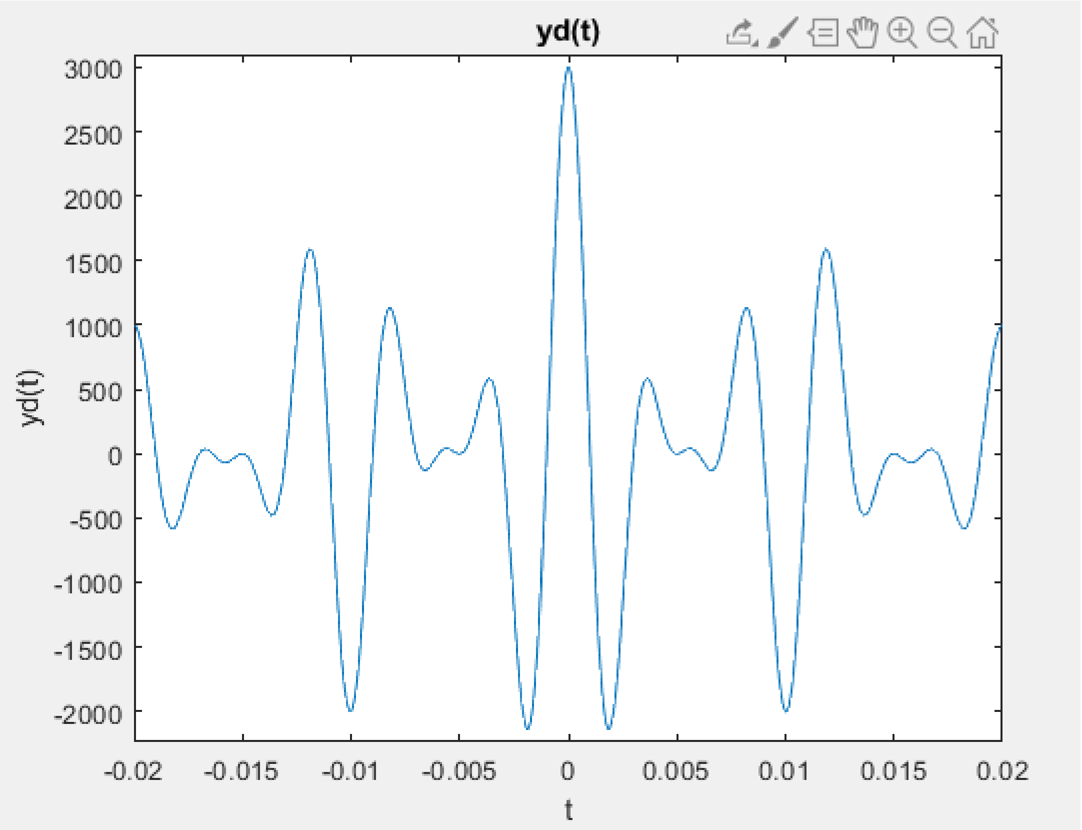
The y(t) graph is given above. As we can tell from the graph, the m(t) signal is modulated with the c(t) signal.



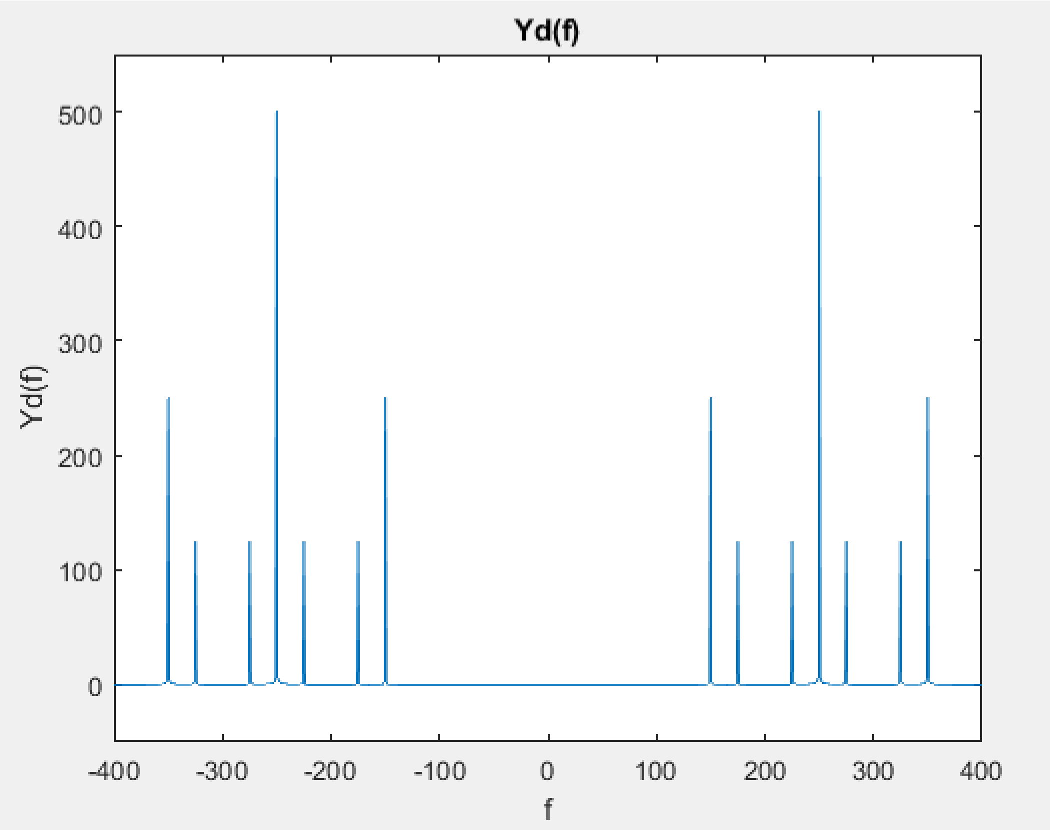
The Y(f) graph is given above. As it can be seen from the graph the m(f) is shifted left and right with the modulators frequency value. This can be expected from the fourier modulation theorem.

2.4 Plotting of the e(t) and E(f) Functions

The e(t) function is the demodulated signal y(t) * d(t). we could call this demodulated y signal ydemodulated (yd) which is equal to e(t).

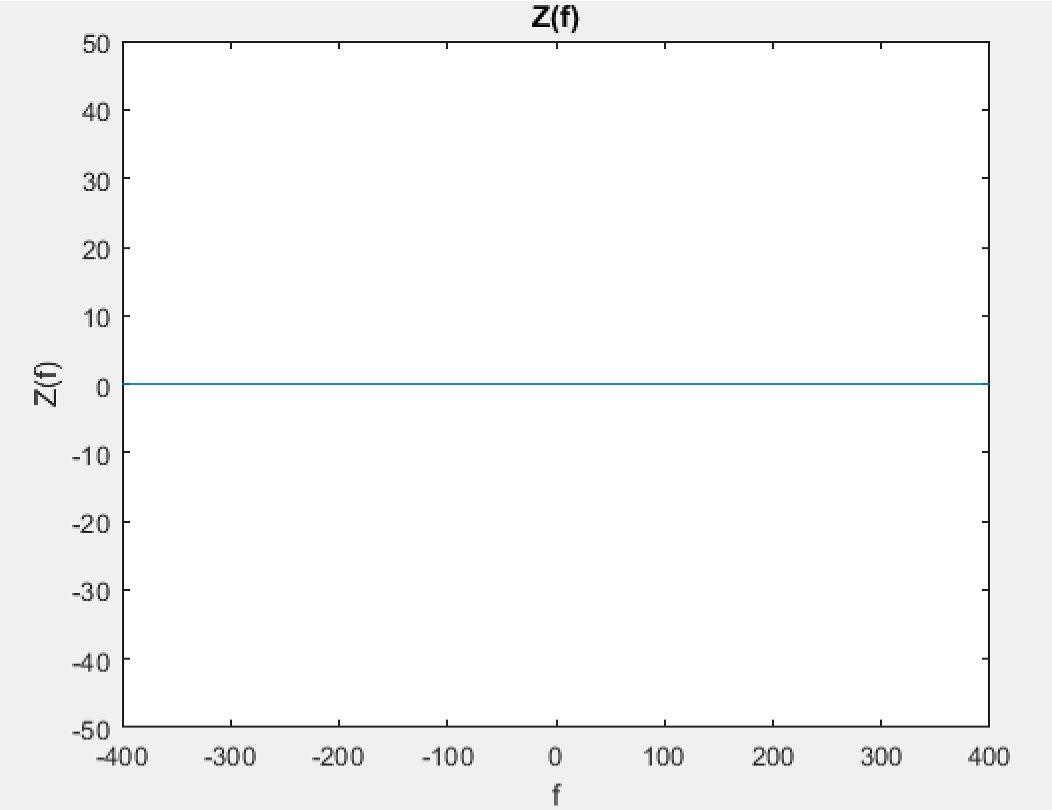


The yd(t) (or simply e(t)) graph is given above. The graph is the demodulated y(t) signal.

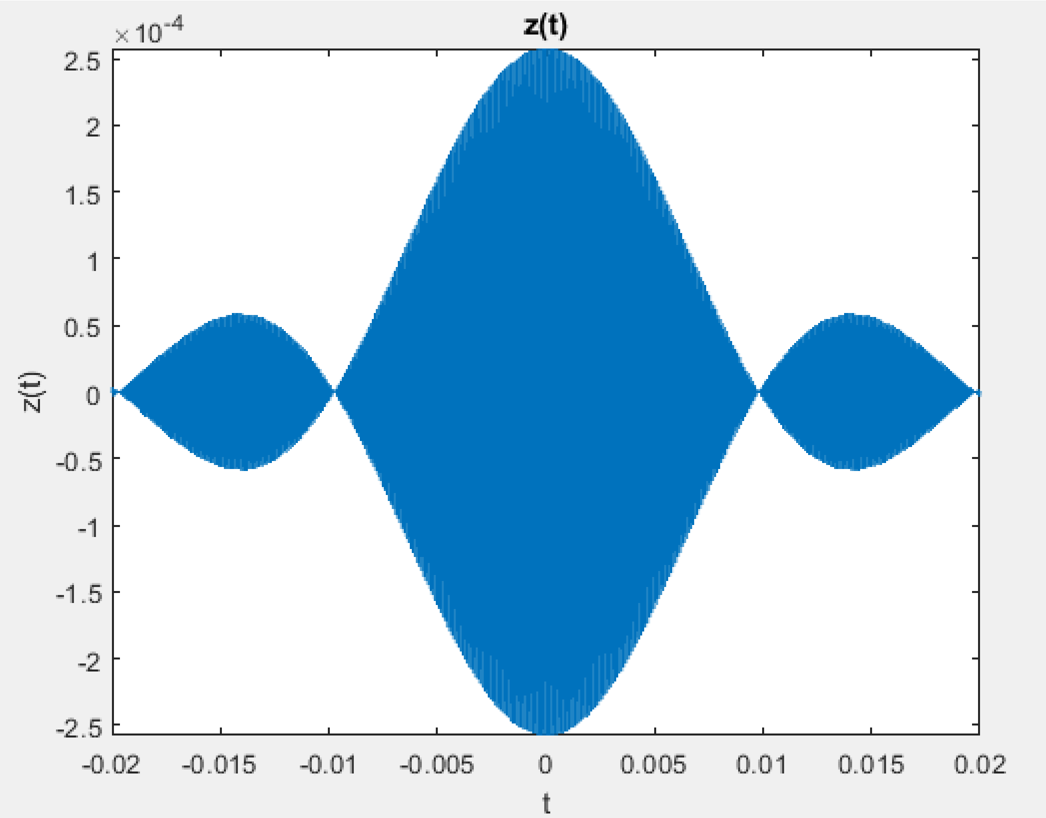


The Yd(f) (or simply E(f)) graph is given above. The graph is as expected since the output is the demodulated signal y(f) which is shifted left and right by the frequency value of the demodulator signal. This can be said using the fourier modulation theorem.

2.5 Plotting of the z(t) and Z(f) Functions



The Z(f) graph is given above. The Z(f) is basically the Yd(f) function passed through a low pass filter with the cutoff frequency at 50 (bandwith of the message signal). This only makes is so that only the part up to 50 and -50 are taken from the Yd(f) graph. Since there are no dirac delat functions on or before 50Hz nothing passes the filter except the noise signals.



The z(t) graph is given above. The z(t) function iş the inverse fourier transform of the Z(f) function. The m(t) message signal could not be recovered. Because the demodulation wave frequency is given wrongly. If we fix the demodulator frequency back to the modulation frequency we could recover the message signal.

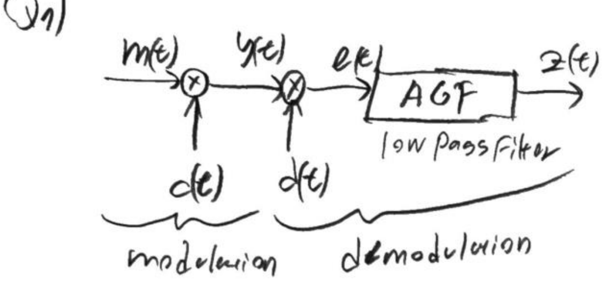
2.6 Analytic Calculation of the Simulations

Below is the analytic calculations of the signals.

ELM 361 Project#1

Ömer Emre Polat
1801022037

Q1)



$$m(t) = 10 \cdot \cos(25 \cdot 2\pi \cdot t) + 20 \cdot \cos(50 \cdot 2\pi \cdot t)$$

$\downarrow \mathcal{F}$ modulation theorem

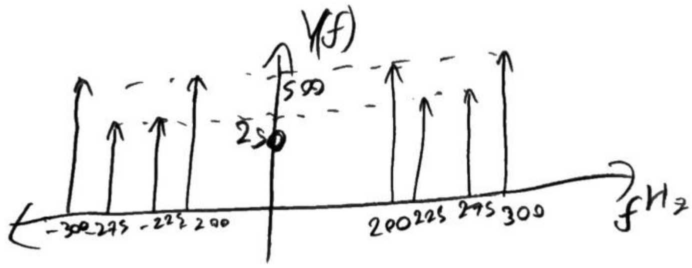
$$M(f) = 5\big(\delta(f-25) + \delta(f+25)\big) + 10\big(\delta(f-50) + \delta(f+50)\big)$$

Q2) $y(t) = m(t) \cdot c(t) = m(t) \cdot 100\cos(250 \cdot 2\pi \cdot t)$

$\downarrow \mathcal{F}$ modulation theorem

$$Y(f) = 50 \cdot \big(M(f-250) + M(f+250)\big)$$

$$Y(f) = 500\big(\delta(f-275) + \delta(f-225) + \delta(f+225) + \delta(f+275)\big) + 250\big(\delta(f-300) + \delta(f-200) + \delta(f+200) + \delta(f+300)\big)$$



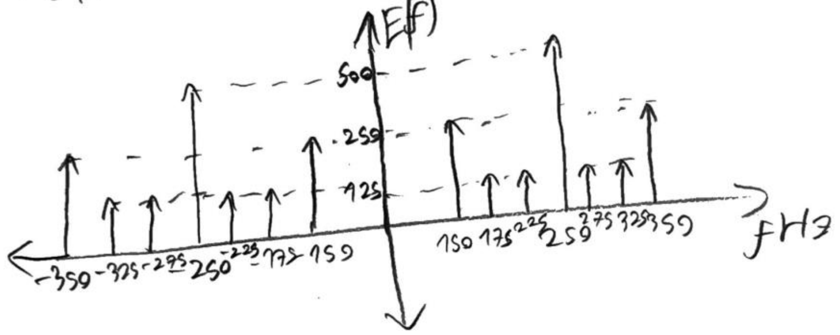
Q3) ELM 361 Proje 1

Ömer Emre Polat
1801022037

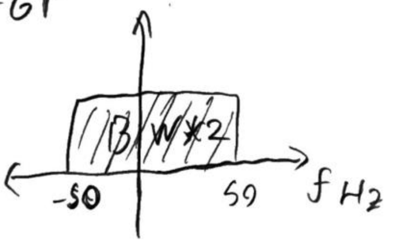
$e(t) = y(t) \cdot \cos(50 \cdot 2\pi \cdot t)$

$\downarrow \mathcal{F}$ modulation theorem

$$E(f) = 125\big(\delta(f-175) + \delta(f-225) + \delta(f-275) + \delta(f-325) + \delta(f+175) + \delta(f+225) + \delta(f+275) + \delta(f+325)\big) + 250\big(\delta(f-150) + \delta(f-250) + \delta(f-250) + \delta(f-350) + \delta(f+150) + \delta(f+250) + \delta(f+250) + \delta(f+350)\big)$$



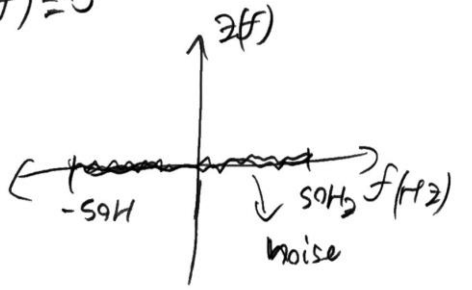
Q4) AGF



$E(f) \cdot AGF = Z(f)$

there are no dirac's on or before 50Hz so this multiplication is zero. only the noise between 0Hz and 50Hz will pass the Filter.

$Z(f) = 0$



The results show the same thing which is that the demodulator is chosen as the wrong frequency. This makes it so that the message signal couldn't be recovered from the low pass filters output.

3. Results and General Comments

As result we have seen that the demodulator frequency was not given right. Because of the demodulator frequency being wrong the message signal couldn’t be recovered from the low pass filters output. In simulation low pass filters output showed a noise that was created from the non-ideal calculations creating noise. Because the modulators and demodulators had certain gain values and the low pass filter was an ideal low pass filter with gain of a 1, the output noise signal was amplified by a big margin. This noise amplification is a problem that can be fixed be lowering the gain of the ideal low pass filter to a certain calculated value.

We could interpret the “wrong demodulator” as another demodulator that communicates with another device that picked up our signal which was supposed to go to our demodulator. Because the carrier signal and demodulator were different, we can assume the receiver end of the signal is a different communication pair than ours. Thankfully having different carrier and demodulator pairs protected them from picking up another signal that was supposed to go to our communication pair.