BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

ENGINEERING SIMULATION LAB

(ENEE4104)

Exp3

“Filter Design Using Matlab”

Prepared by:

Name: Mohammad Shehadeh   Number:1201458

Section:1   Date:10/9/2024

Contents

Table of Figures

Theroy

Filters

A filter is a device or process used to remove unwanted components or features from a signal. The main purpose of filtering is to suppress some components of a signal which is most likely noise. While filters are not just used in frequency domain applications, this usually entails eliminating frequencies or frequency ranges.

Among the most commonly used filters are low pass, high pass, and band pass filters. Each type is designed to allow certain frequencies to pass while attenuating others.

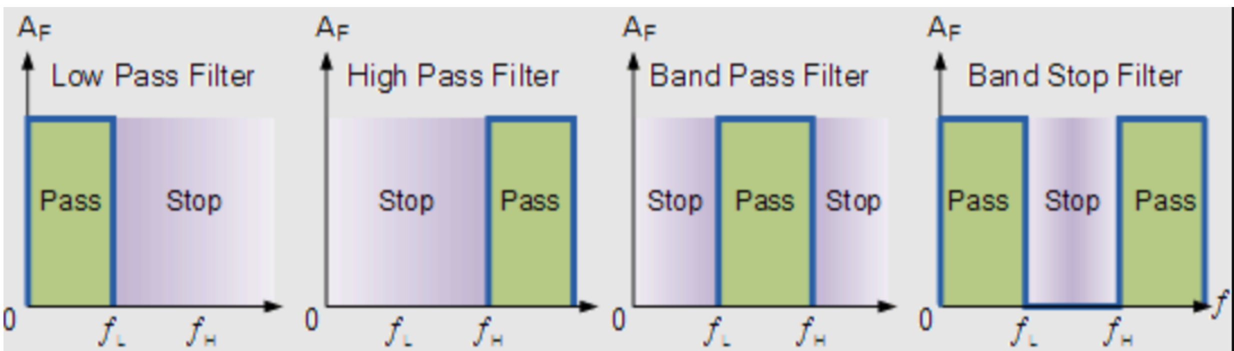

Figure 1 Ideal Filter Response Curves

Lowpass filter:

A low pass filter (LPF) is a device that lets low-frequency signals pass through while reducing or blocking higher-frequency ones. It's frequently applied to eliminate undesired high-frequency noise or to smooth out data. An LPF, for instance, can be used in audio processing to reduce harsh, high-pitched noises and produce a smoother signal and it is also used in communication networks to pass multiple signals at the same band.

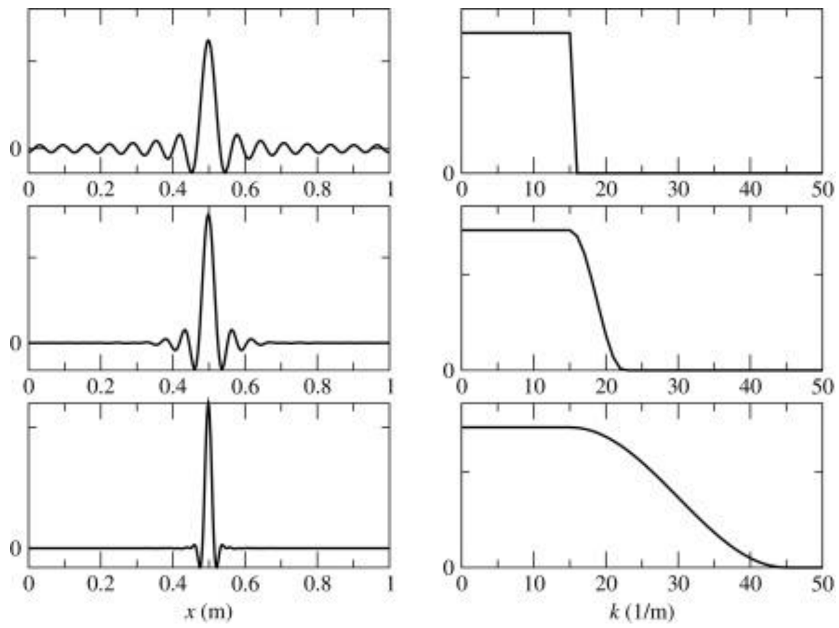

Figure 2 Low pass filter Time and frequence domain

High pass filter

A high pass filter (HPF) does the opposite of an LPF, allowing frequencies above a certain cutoff point to pass while attenuating the lower frequencies. This type of filter is commonly used to remove low-frequency noise, or to emphasize higher frequencies in signals.

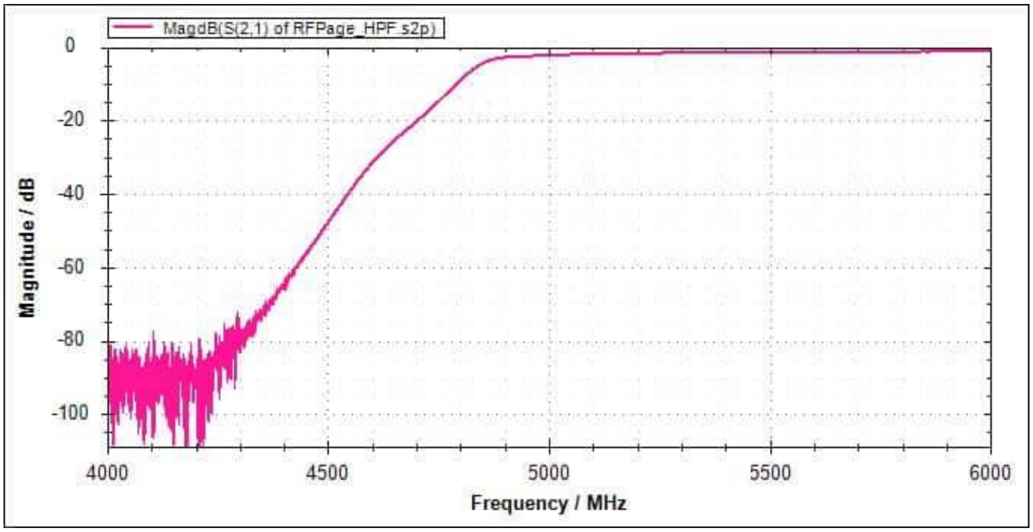

Figure 3 frequency response of HPF

Band pass filter

A band pass filter (BPF) allows frequencies within a specific range, or band, to pass through while attenuating frequencies outside that range. It is used to isolate a narrow range of frequencies from a signal, making it ideal for applications like radio tuners, where only a specific frequency band is of interest.

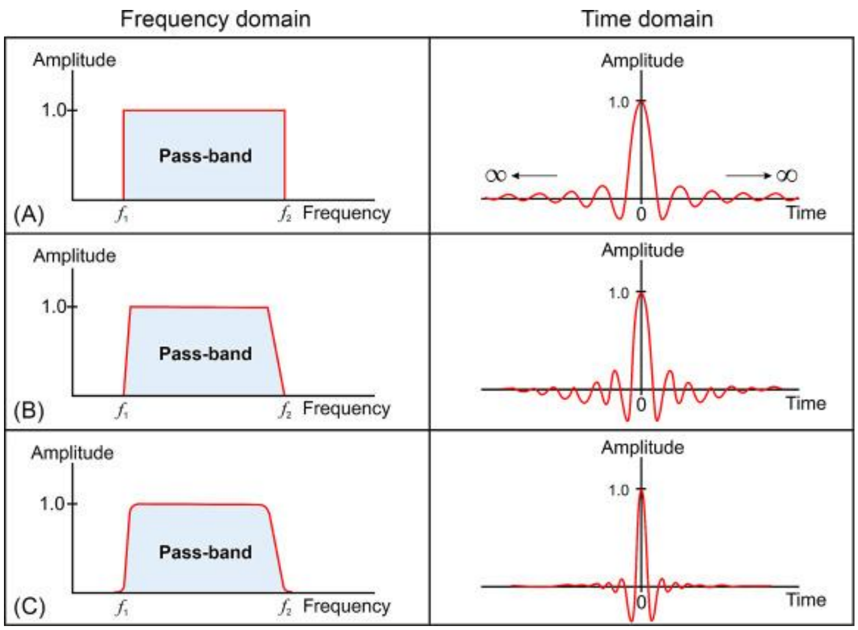

Figure 4 band pass filter Time and frequence domain

Procedure

First we used MATLAB to draw every signal on it's own x1,x2,x3 and the sum of them called as X as you can see in Figure 5.

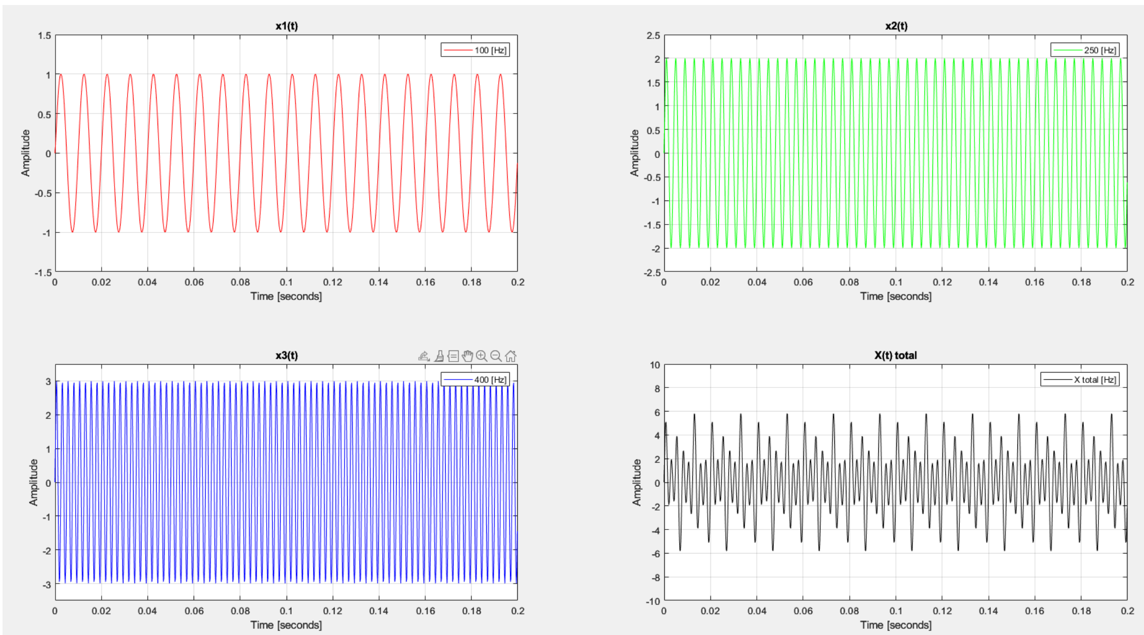

Figure 5 plot of each signal

Then we introduced AWGN (Additive white Gaussian noise) to the sum of them and we called this signal Y then we compared it to the signal without noise as you can see in Figure 6

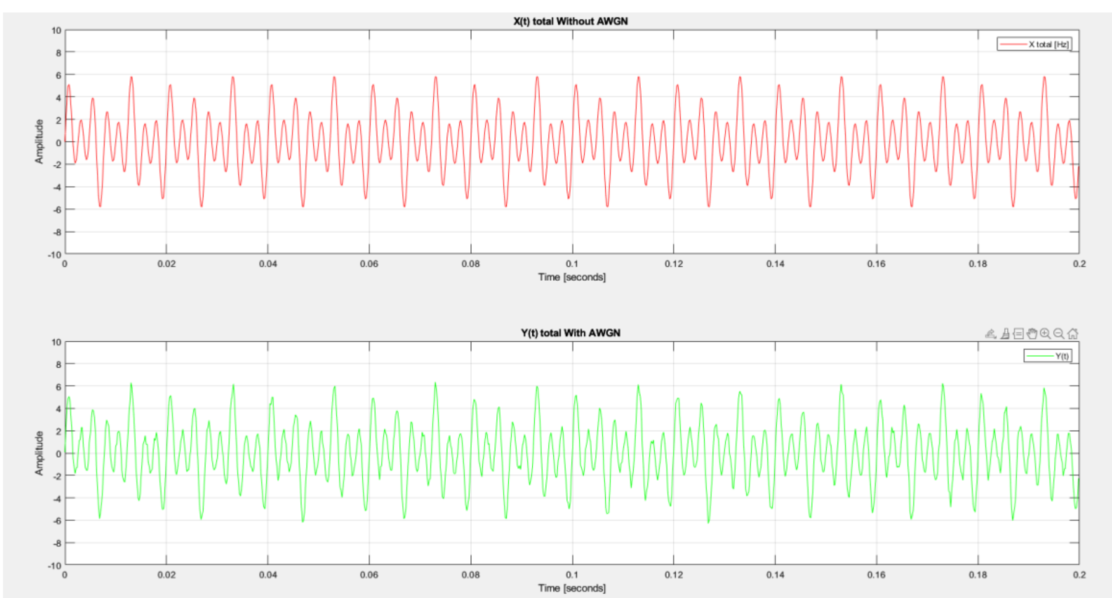

Figure 6 plot with AWGN and without it

Then we used the signal Analyzer tool in MATLAB and we added Y the signal with AWGN and we used the filters to extract x1,x2,x3 as you can see in Figure 7

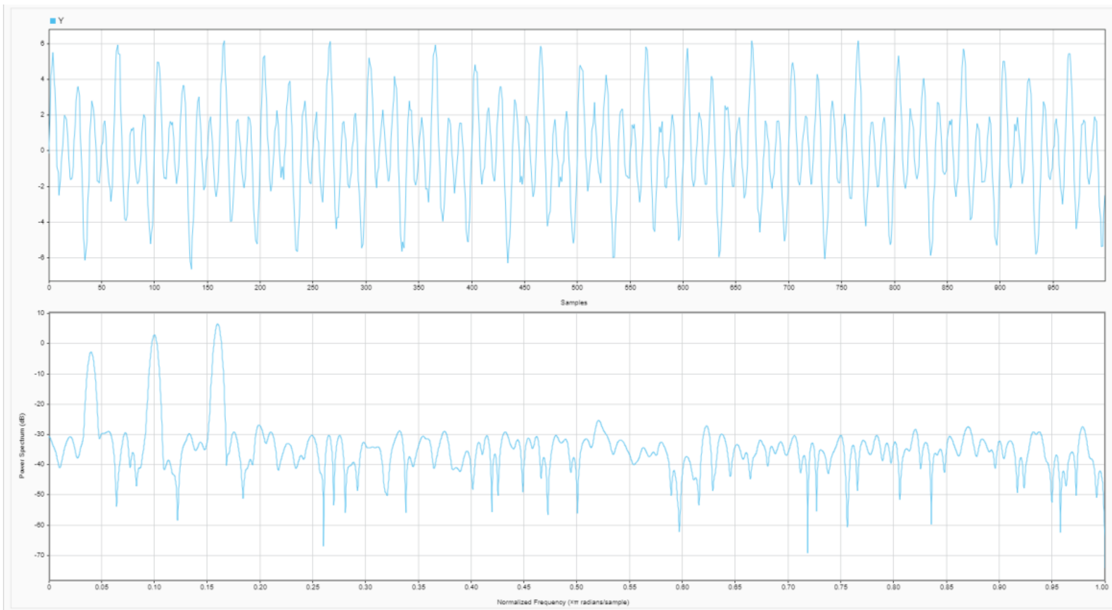

Figure 7 Y in both time and frequence domain

First we used a low pass filter set with a 110 Hz as it's maximum passing limit and we got the signal in Figure 8

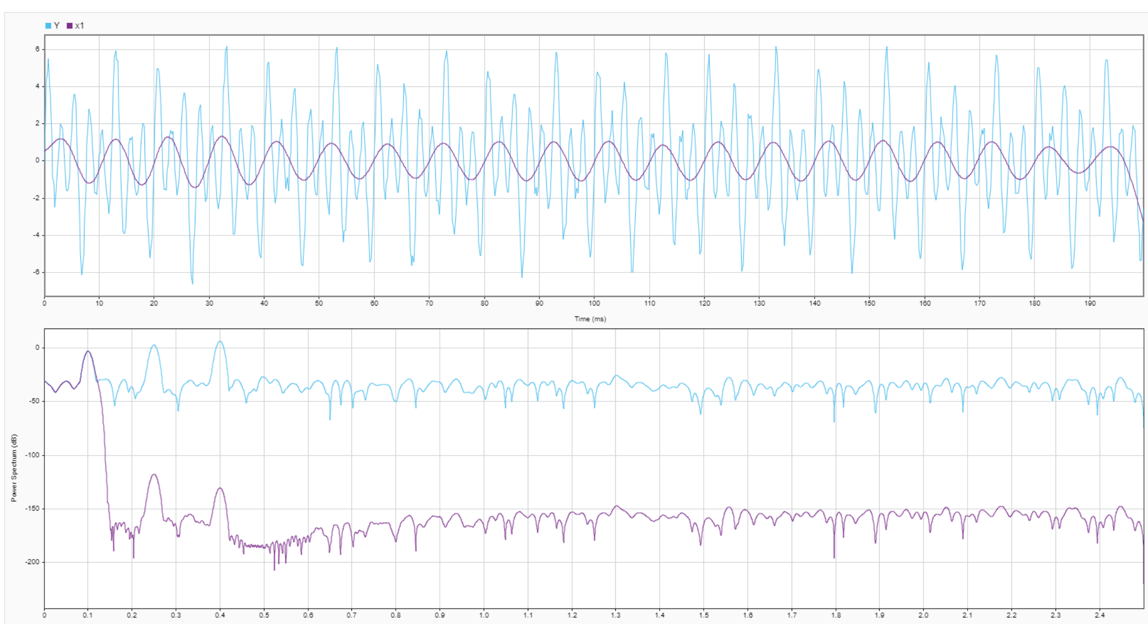

Figure 8 x1 and Y in Time and frequency domain

Then we used a band pass filter with a minimum threshold of 200Hz and a maximum of 300Hz and we got the plot in Figure 9.

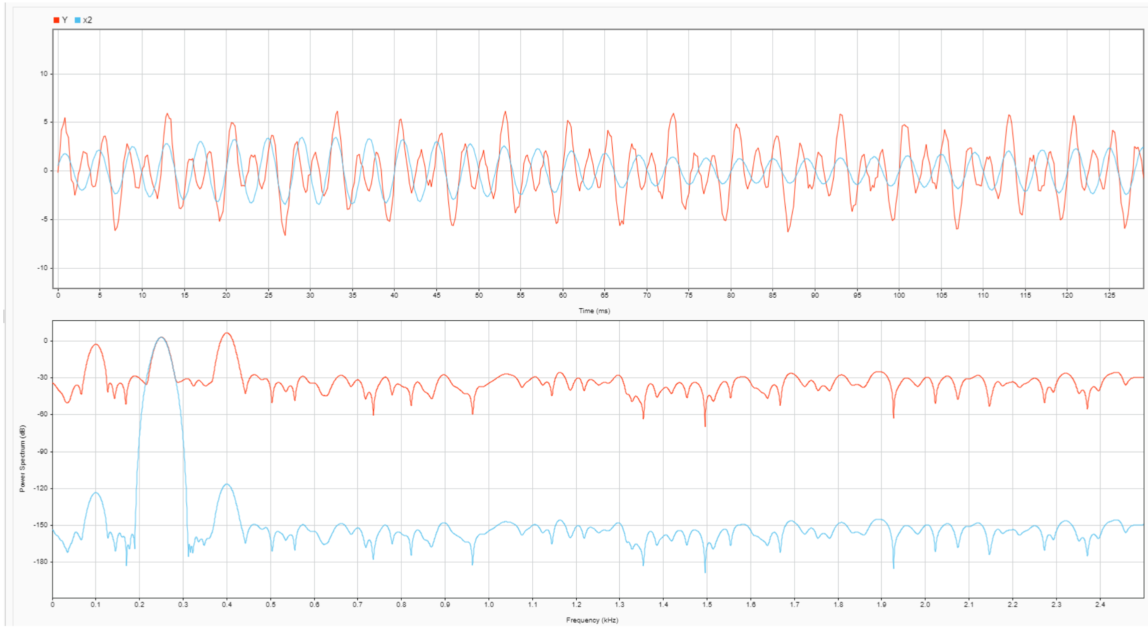

Figure 9 x2 and Y in Time and frequency domain

Then we used a High pass filter with a minimum passing frequency of 350Hz as you can see in Figure 10 below

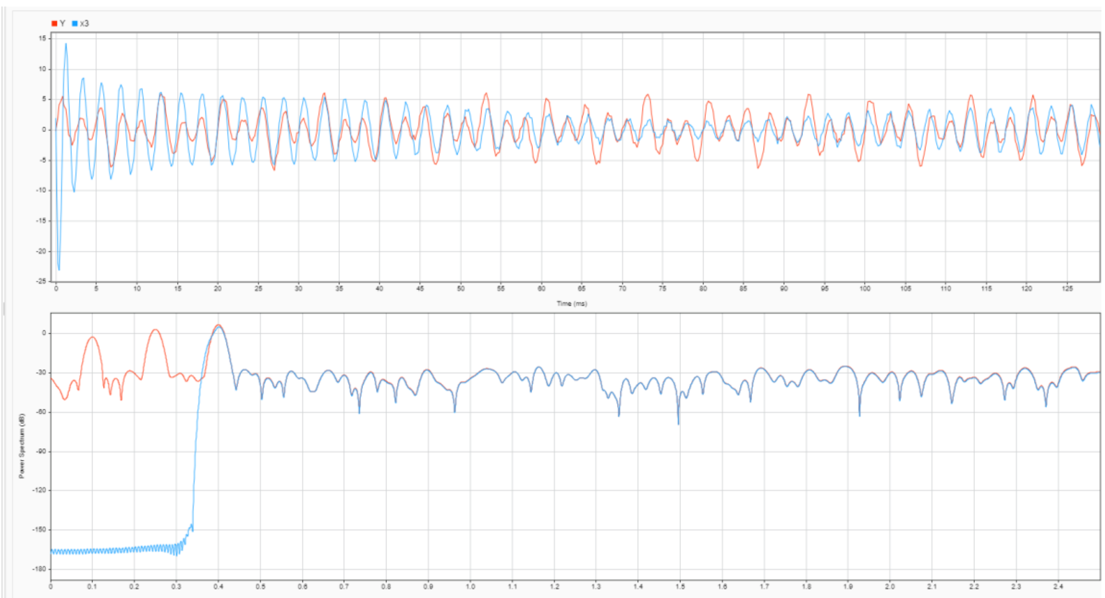

Figure 10 x3 and Y in Time and frequency domain

Conclusion

Finally we learned in this experiment how to use MATLAB to create signals and how to introduce AWGN to the signals and how it affects the signals we also learned how to use signal analyzer tool and the procedure of applying filters to the signals.

References

[1] ENEE3309 - Communication Systems Lecture notes

[2] https://www.electronics-tutorials.ws/filter/filter_2.html 10/10/2024 7:06 PM

[3]https://www.rfpage.com/low-pass-high-pass-and-band-pass-filters-simple-explanation/ 10/10/2024 7:06 PM

[4]https://www.sciencedirect.com/topics/earth-and-planetary-sciences/low-pass-filter 10/10/2024 7:07 PM

Appendix

MATLAB Code

```
clc;

clear all;

%Mohammad shehadeh 1201458
%Iyad Alkhateeb 1190824

fs=5000;

n=1000;

t=(0:n-1)/fs;

r_db = 20;



x1=1*sin(2*pi*100*t);

x2=2*sin(2*pi*250*t);

x3=3*sin(2*pi*400*t);

X=x1+x2+x3;

Y = awgn(X, r_db, 'measured');
```

```
figure(1);

%%%%%%X1

subplot(2,2,1);

plot(t, x1, 'r');

title('x1(t)');

xlabel('Time [seconds]');

ylabel('Amplitude');

ylim([-1.5 1.5]);

legend('100 [Hz]');

grid on

%%%%%%X2

subplot(2,2,2);

plot(t, x2, 'g');

title('x2(t)');

xlabel('Time [seconds]');

ylabel('Amplitude');

ylim([-2.5 2.5]);
```

```
legend('250 [Hz]');

grid on

%%%%%%X3

subplot(2,2,3);

plot(t, x3, 'bl');

title('x3(t)');

xlabel('Time [seconds]');

ylabel('Amplitude');

ylim([-3.5 3.5]);

legend('400 [Hz]');

grid on

%%%%%%X total

subplot(2,2,4);

plot(t, X, 'k');

title('X(t) total');

xlabel('Time [seconds]');

ylabel('Amplitude');
```

```
ylim([-10 10]);

legend('X total [Hz]');

grid on



figure(2);

%Without AWGN;

subplot(2,1,1);

plot(t, X, 'r');

title('X(t) total Without AWGN ');

xlabel('Time [seconds]');

ylabel('Amplitude');

ylim([-10 10]);

legend('X total [Hz]');

grid on

%With AWGN;

subplot(2,1,2);

plot(t, Y, 'g');
```

```
title('Y(t) total With AWGN ');

xlabel('Time [seconds]');

ylabel('Amplitude');

ylim([-10 10]);

legend('Y(t)');

grid on
```