Department of Electrical and Electronic Engineering
Summer 21

EEE 344/ECE344

Digital Signal Processing Laboratory

Lab Report

Section: 3 Group: 22

Experiment = Open ended project

Name of members:

Name	***ID***
Md. Shamsur Rahman shishir	18321020
Nuren Tahsin	18321047
Md. Abid Azad	17121034
Proma Roy	17121036

Date of Submission: 16/9/2021

Objective: The objective of this project is get us familiar with real life implementation of digital signal processing using MATLAB. In this open-ended project, we need to design a radar system which has a pre determind singnal that transmits a short tone sound wave and receives a weak echo from a distant target and we have to calculate the time delay (the time needed to travel the sound which was transmitted from the transmitter to the target and then from the target to the receiver) of a Radar system theoretically.Since the echo will be masked and not detectable we are required to auto correlate the transmitted signal and the received signal to view the time delay of the echo from the auto correlated graph.

To get required output we need to follow few steps-

1.Transmit Signal

2. Delay the Signal

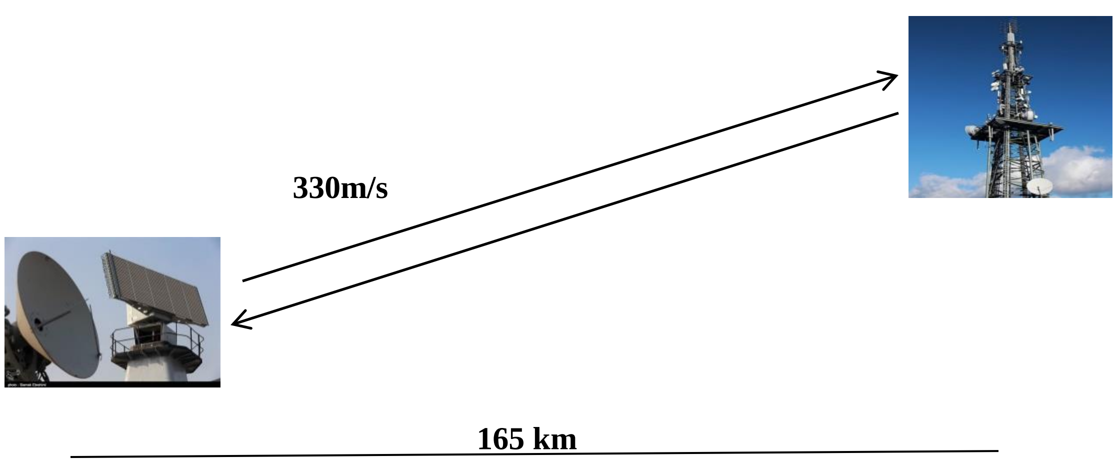
3. Attenuate signal

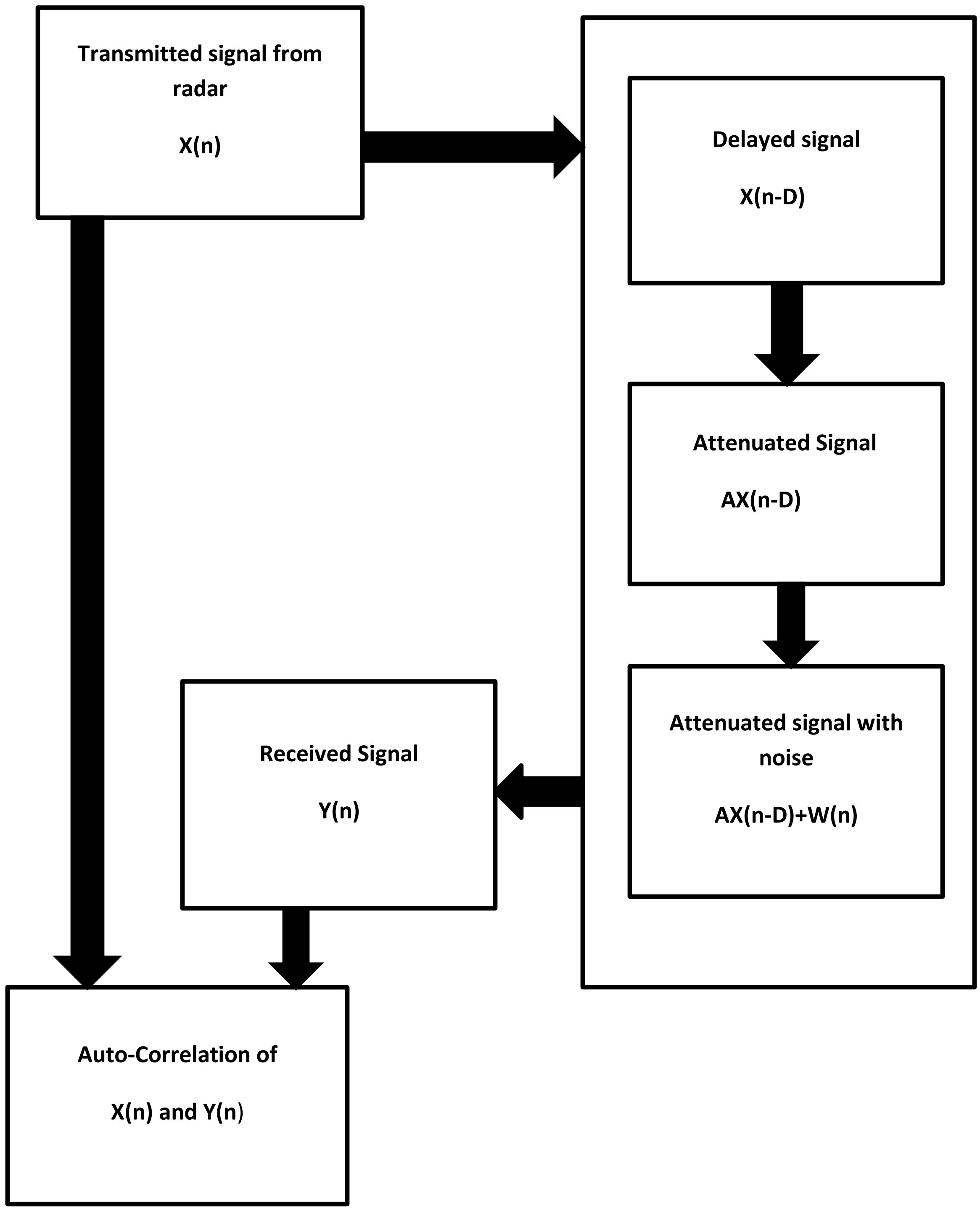
4.Attenuate signal with noise

5.Receive signal

6.Auto correlation

Block Diagram :




Transmitted signal from radar
X(n)
Delayed signal
X(n-D)
Attenuated Signal
AX(n-D)
Attenuated signal with noise
AX(n-D)+W(n)
Received Signal
Y(n)
Auto-Correlation of
X(n) and Y(n)

Methodology:

As mentioned in this project,

Velocity of the sound wave, V = 330 m/s

Distance, d = 165 km = 165000 m

Time delay = (2*d)/V

= (2*165000)/330

= 1000 second

Steps:

1. The delay time between transmitted and received signal is 1000 seconds.
2. From the input graph we can see that ,t he signal completes it's full cycle in 300 seconds . So, To complete 1 cycle it takes(300/2) =150 second . So, T= 150 seconds. The signal from time 0-300 is continuous and then time 301-1000 will lead the signal with zero frequency and zero amplitude. Again, following the same pattern it will be periodic after 1001-2000 and so on.
3. We used auto correlation using " tstep" in MATLAB code. Autocorrelation represents the degree of similarity between a given time series and a lagged version of itself over successive time intervals.
4. Autocorrelation measures the relationship between a variable's current value and its past values.
5. We can write the initial input signal from the equation, x(n) = amplitude*sin ($2\pi * f * t$), where the amplitude of the signal is 5.
6. Now we know that, Y(n) = AX(n-D) + W(n), which is a equation of a received signal where we have to add different level of white noise(W(n)) and attenuation(A)with time delay(D) .
7. At first, we have to add noise using SNR command. In signal processing, noise is a general term for unwanted (and in general, unknown) modifications that a signal may suffer during

capture, storage, transmission, processing, or conversion.

8. Then we have to add attenuation with delay time. Attenuation is a general term that refers to any reduction in the strength of a signal.

9. By implementing code in Matlab , we get our expected result.

Results:

The results of this open-ended lab project required different level of noises and attenuation. There is total 4 levels of noise and each has 4 levels of attenuation. That makes in total 16 combinations.

Input Signal :

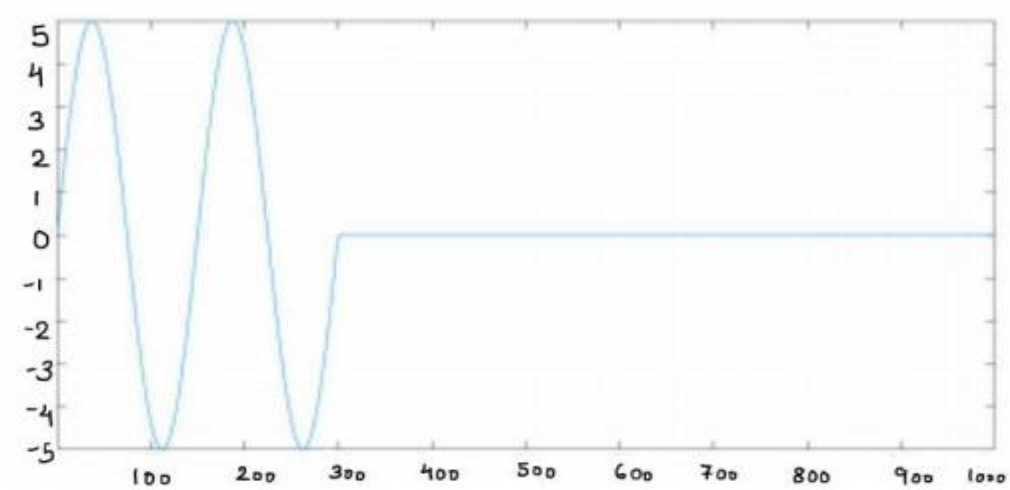


Figure 1: Transmitted sound wave

1. Attenuation=0.2 and SNR=-5db

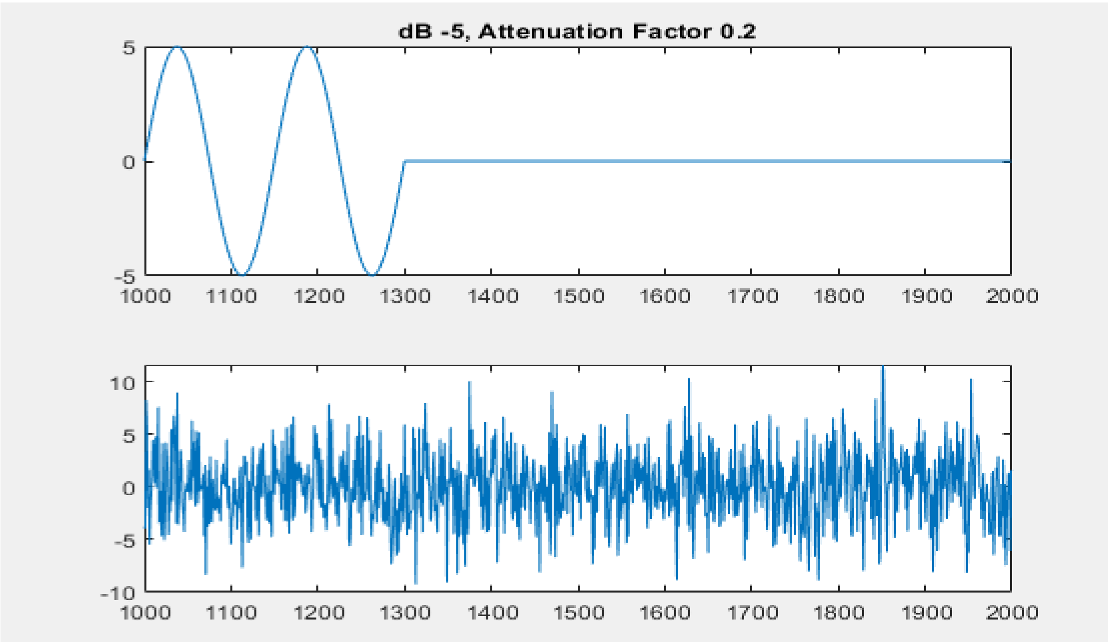


Figure 01: Receive Output Graph at -5 dB and 0.2 Attenuation

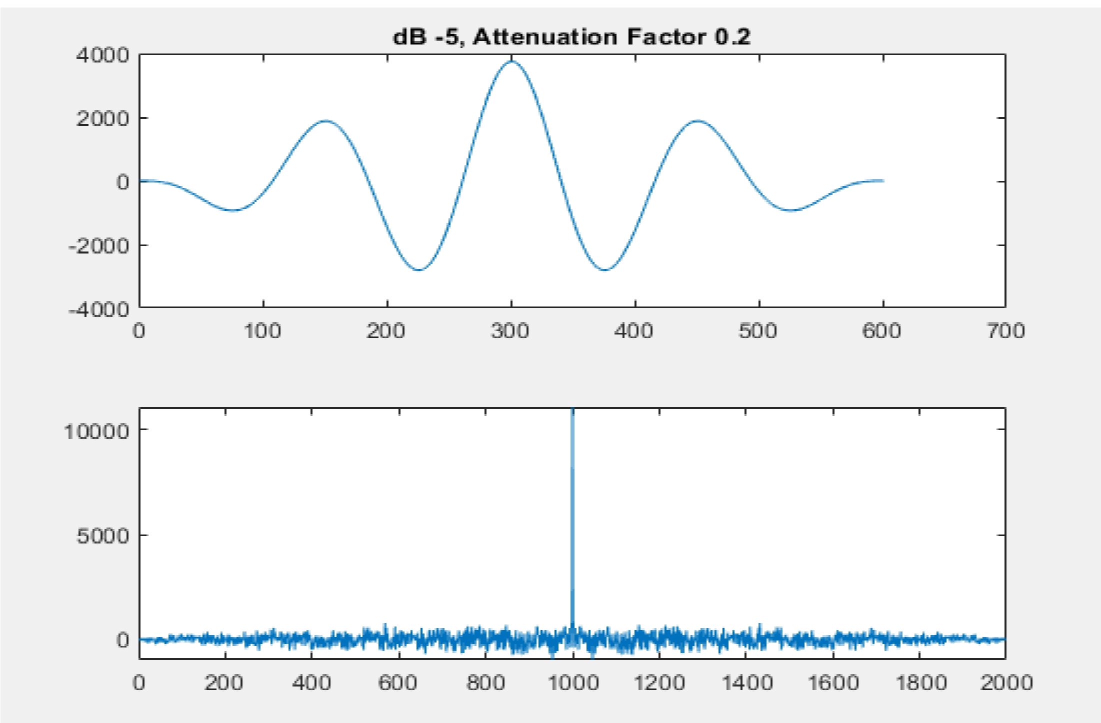


Figure 02: ACF graph at -5 dB and 0.2 attenuation

2. Attenuation =0.5 and SNR= -5db

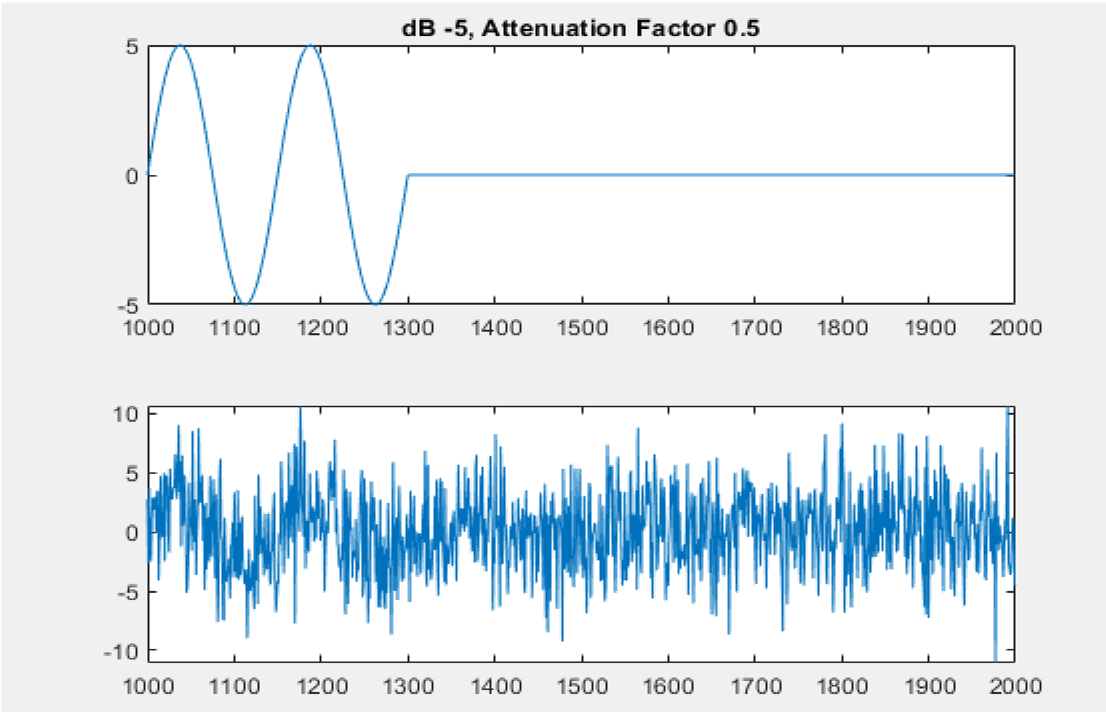


Figure 03: Receive Output Graph at -5 dB and 0.5 Attenuation

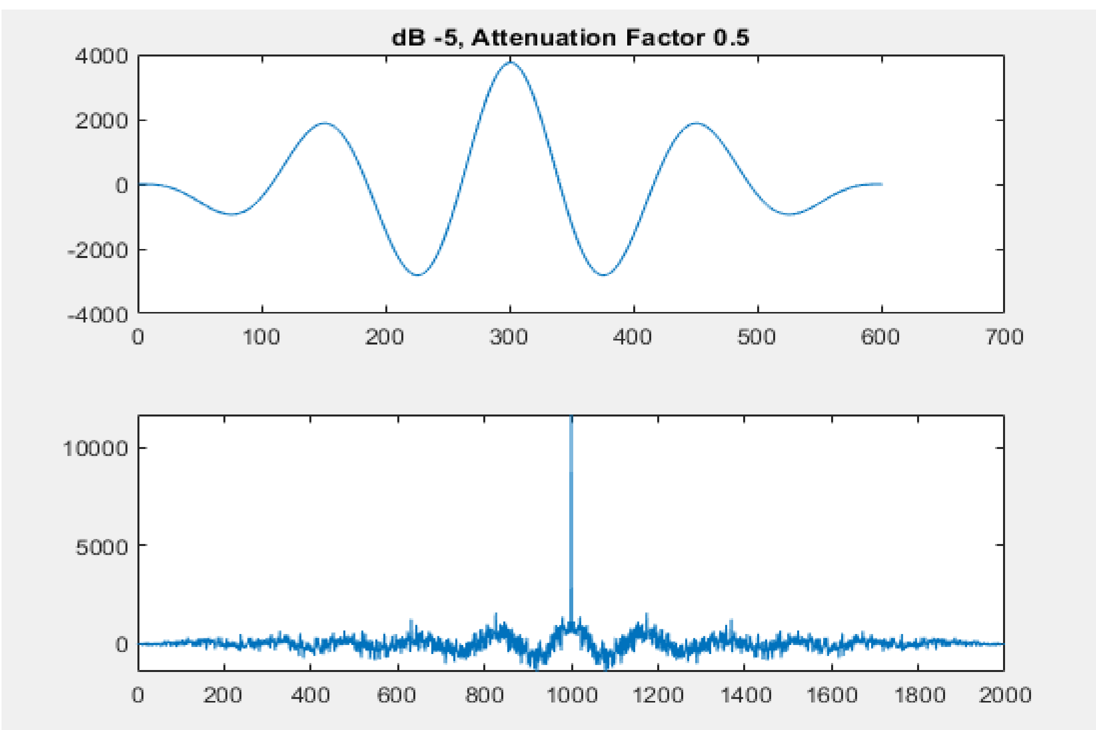


Figure 04: ACF graph at -5 dB and 0.5 attenuation

3. Attenuation=0.7 and SNR=-5db

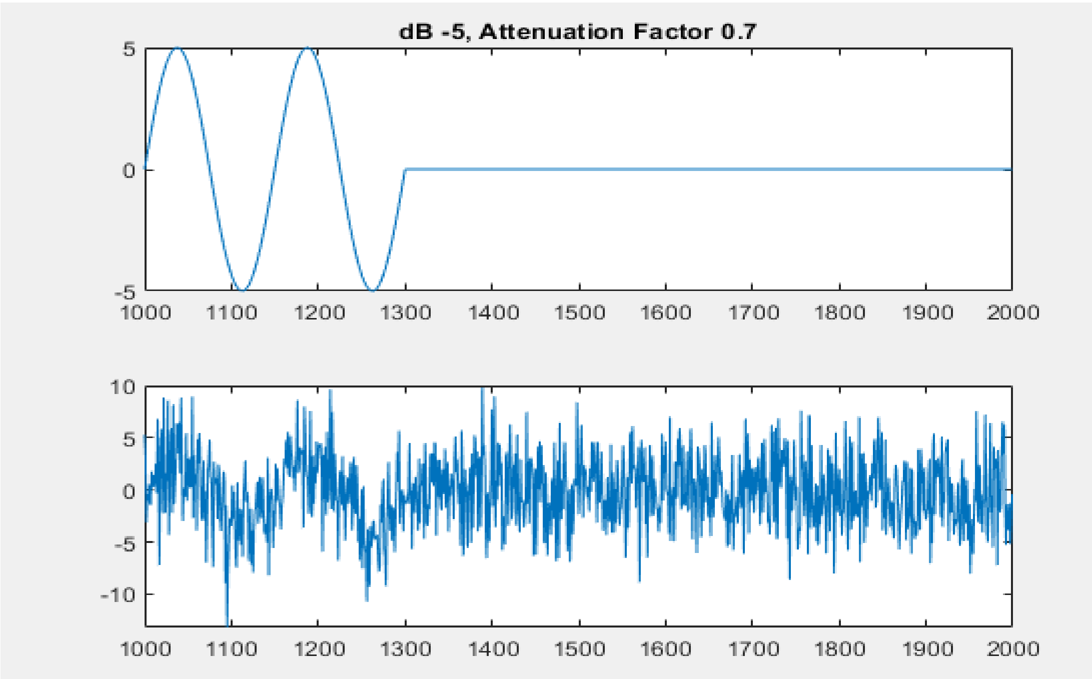


Figure 05: Receive Output Graph at -5 dB and 0.7 Attenuation

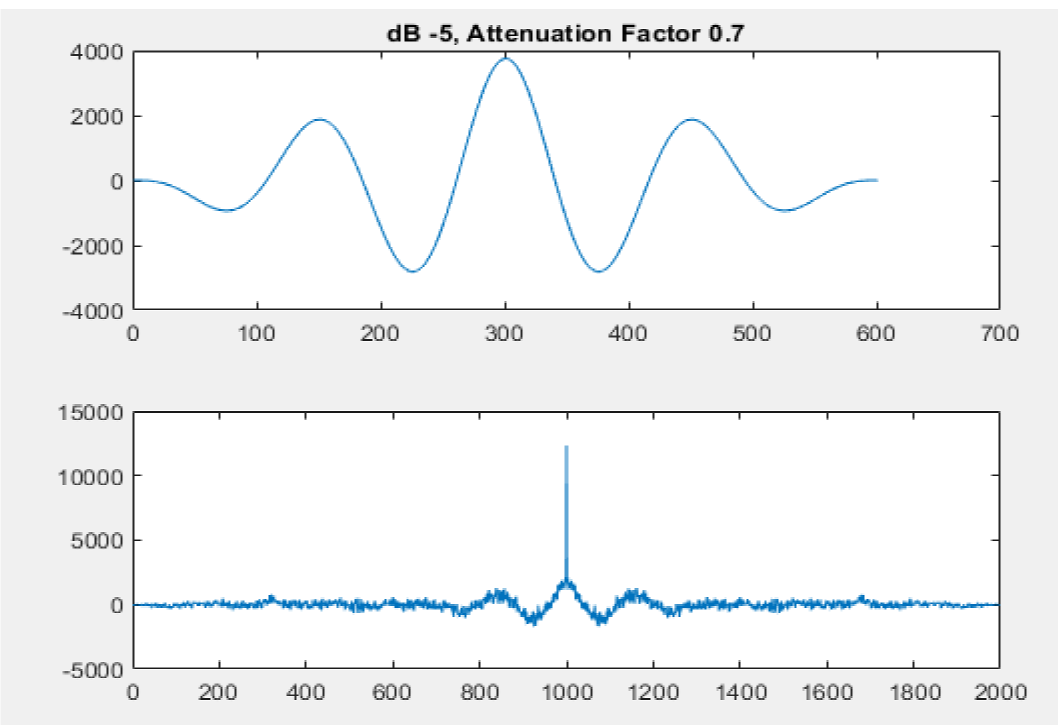


Figure 06: ACF graph at -5 dB and 0.7 attenuation

4. Attenuation=0.9 and SNR= -5db

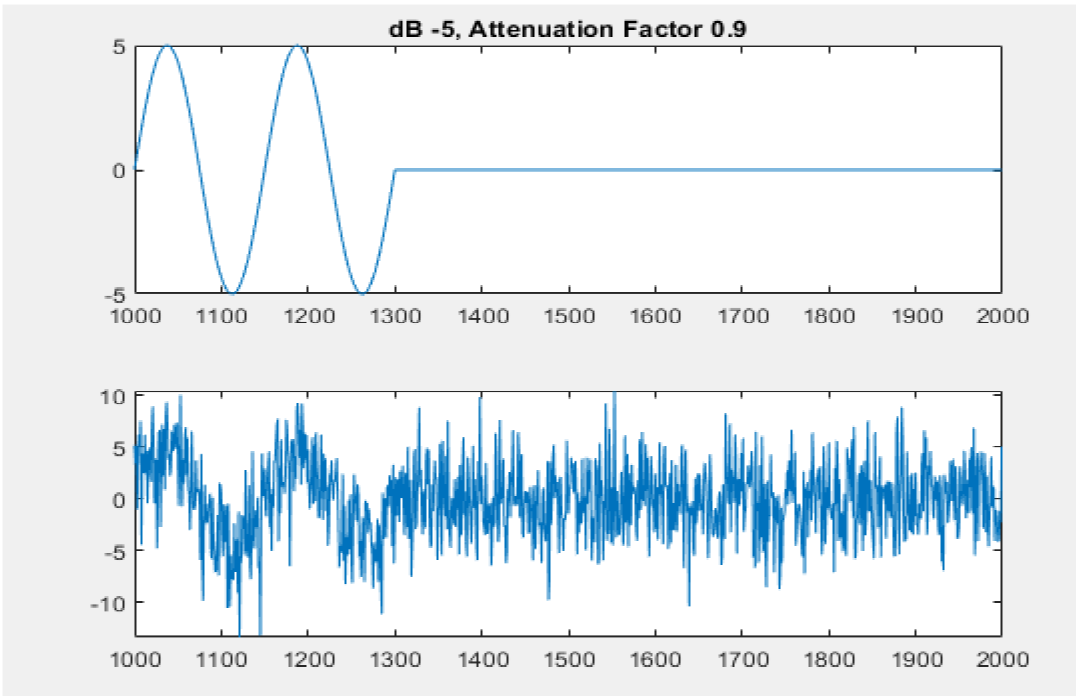


Figure 07: Receive Output Graph at -5 dB and 0.9 Attenuation

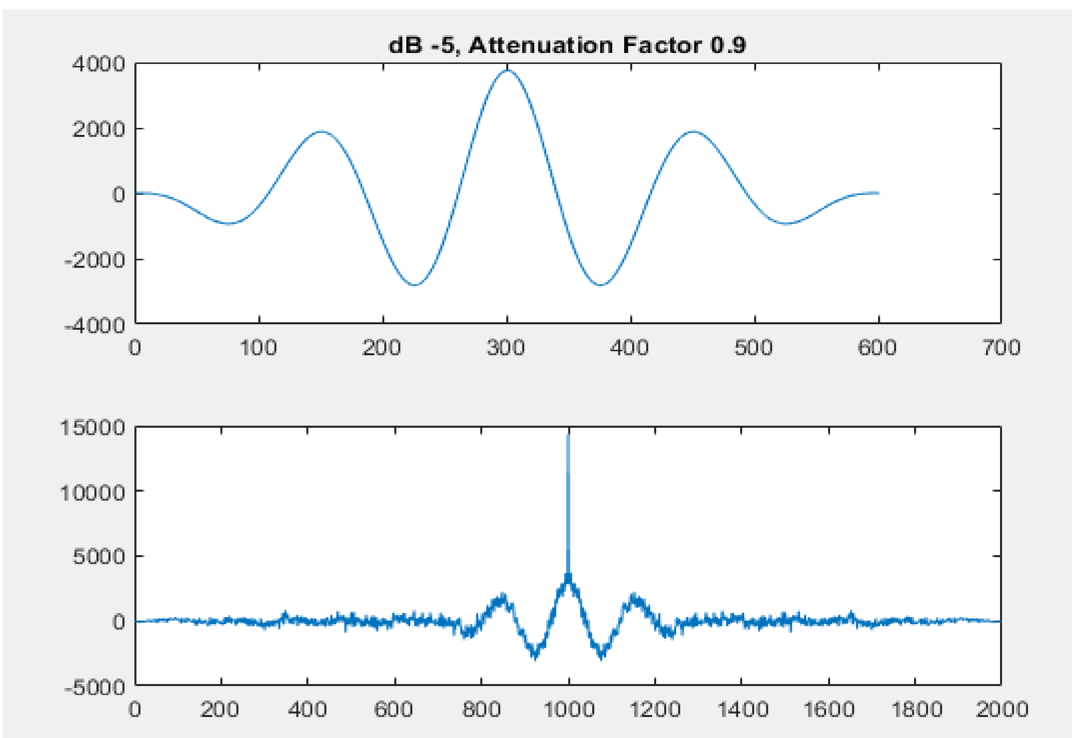


Figure 08: ACF graph at -5 dB and 0.9 attenuation

5. Attenuation=0.2 and SNR=-15db

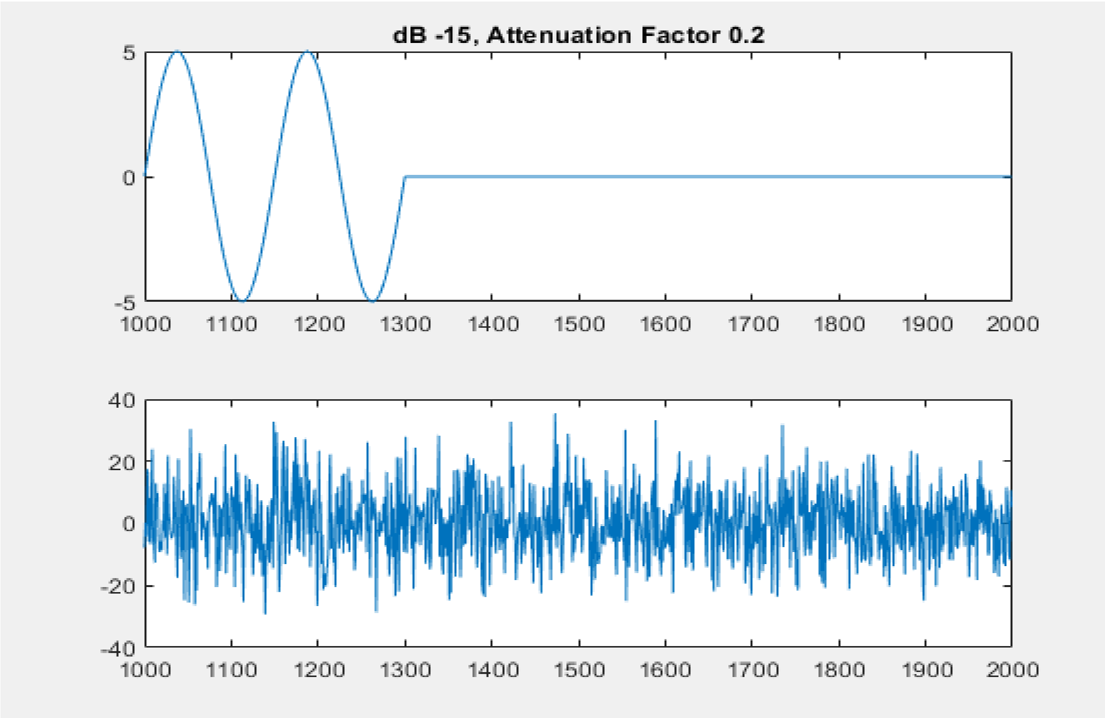


Figure 09: Receive Output Graph at -15 dB and 0.2 Attenuation

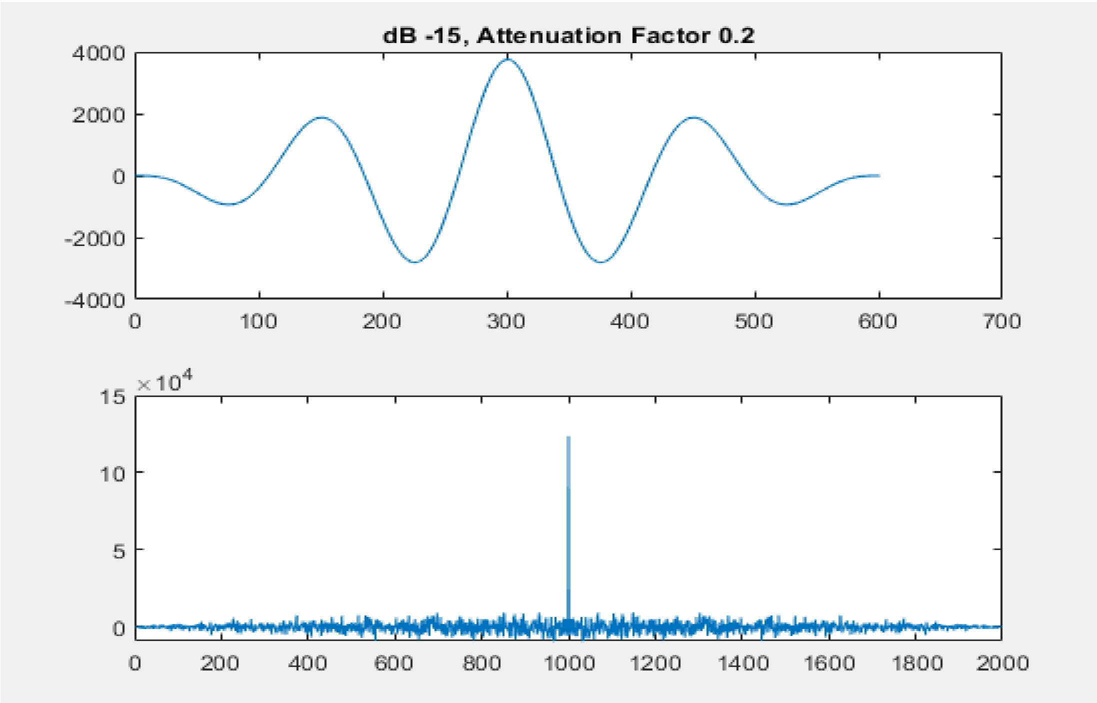


Figure 10: ACF graph at -15 dB and 0.2 attenuation

6. Attenuation=0.5 and SNR=-15db

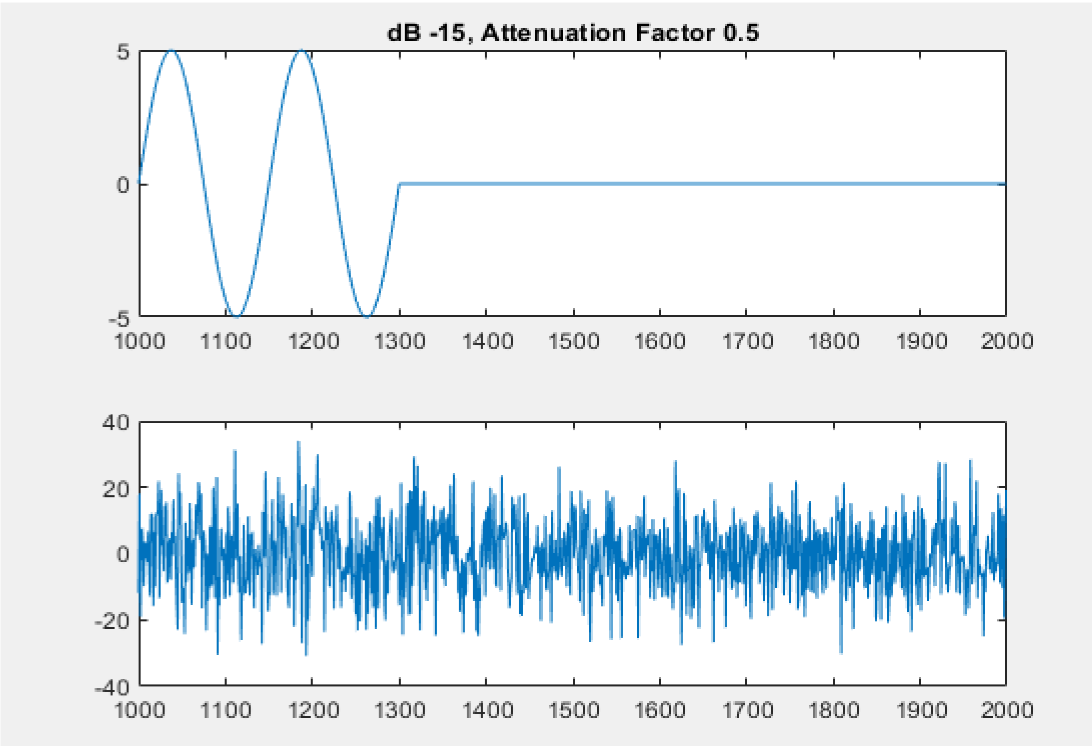


Figure 11: Receive Output Graph at -15 dB and 0.5 Attenuation

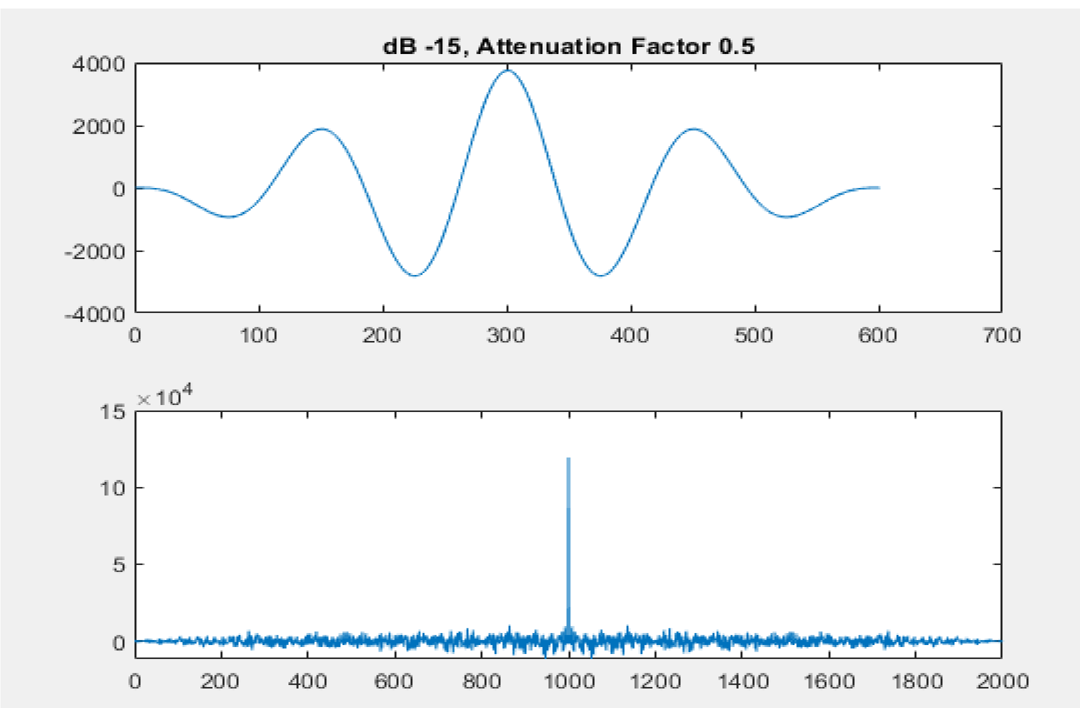


Figure 12: ACF graph at -15 dB and 0.5 attenuation

7. Attenuation=0.7 and SNR=-15db

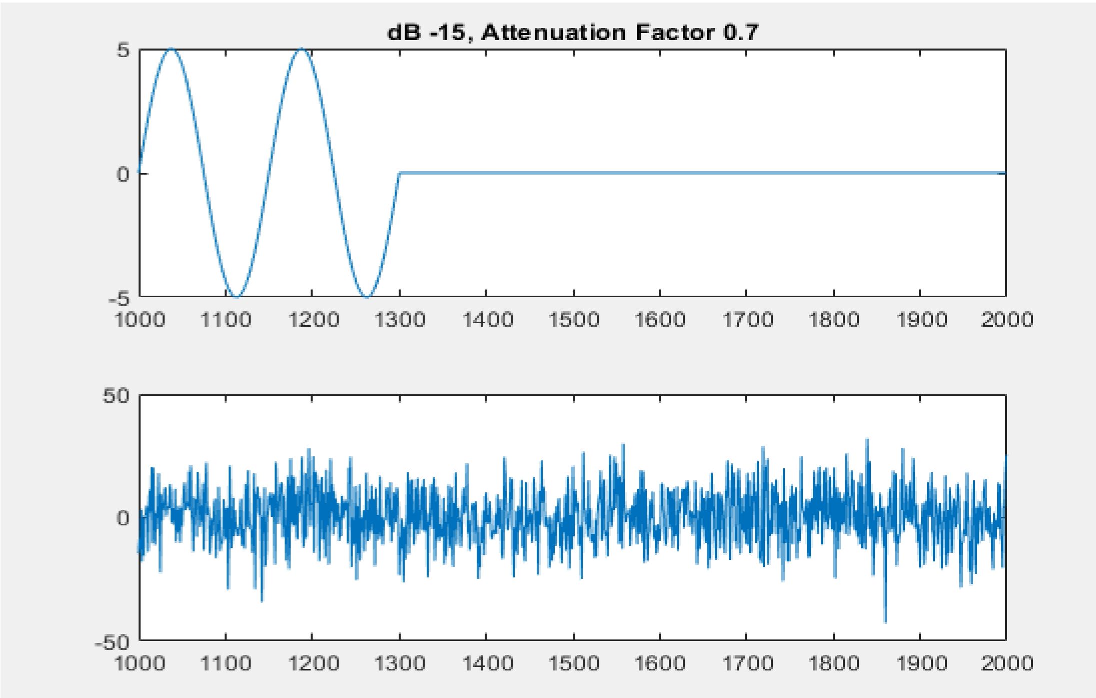


Figure 13: Receive Output Graph at -15 dB and 0.7 Attenuation

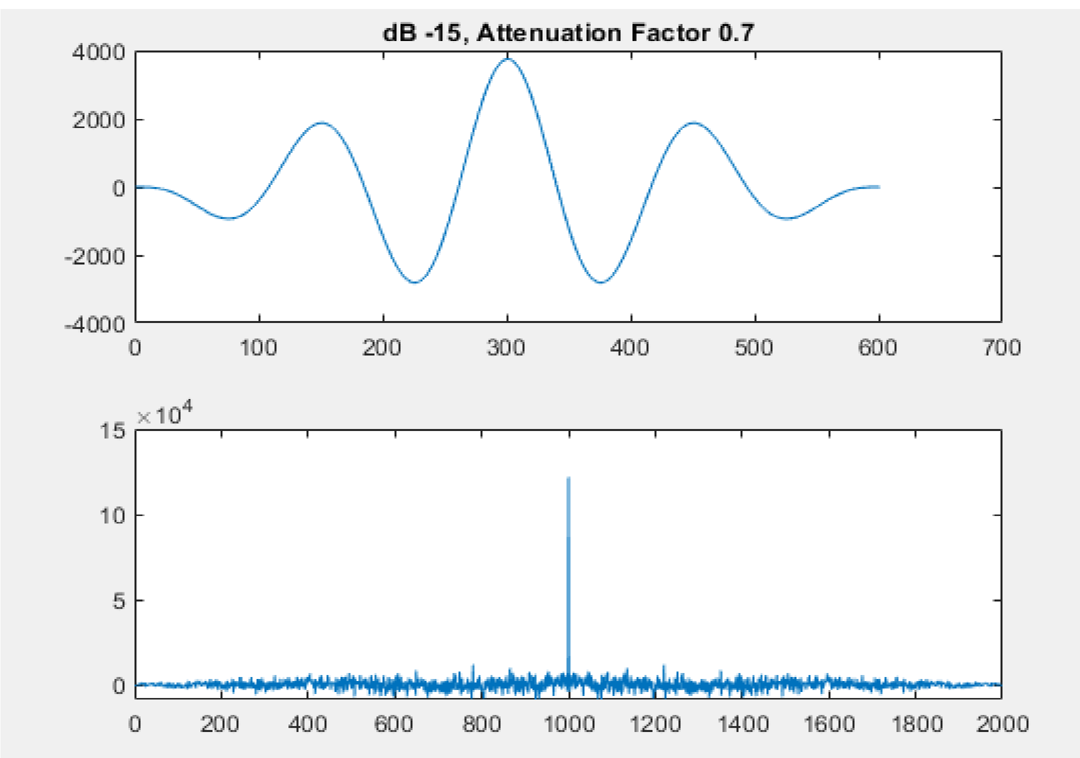


Figure 14: ACF graph at -15 dB and 0.7 attenuation

8. Attenuation=0.9 and SNR=-15db

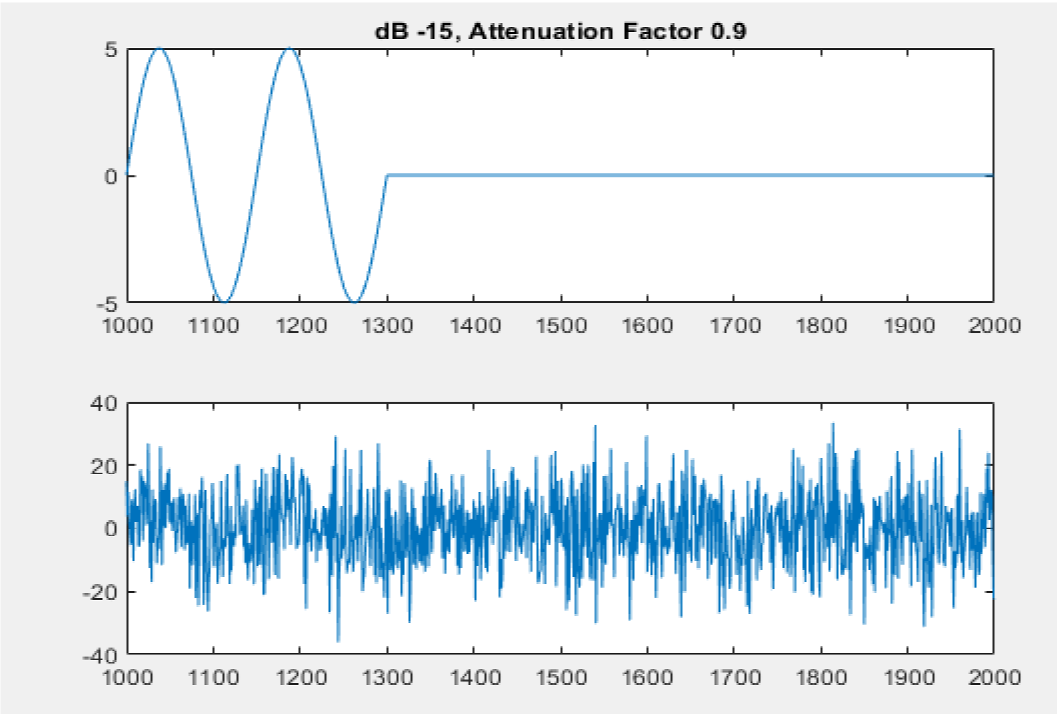


Figure 15: Receive Output Graph at -15 dB and 0.9 Attenuation

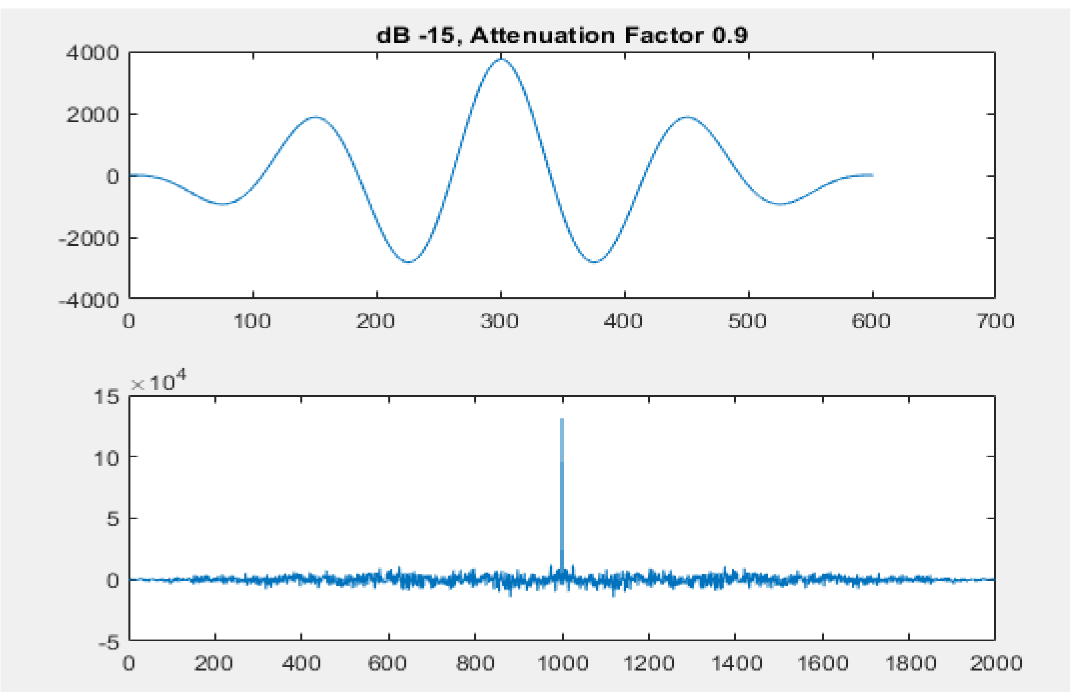


Figure 16: ACF graph at -15 dB and 0.9 attenuation

9. Attenuation=0.2 and SNR=-25db

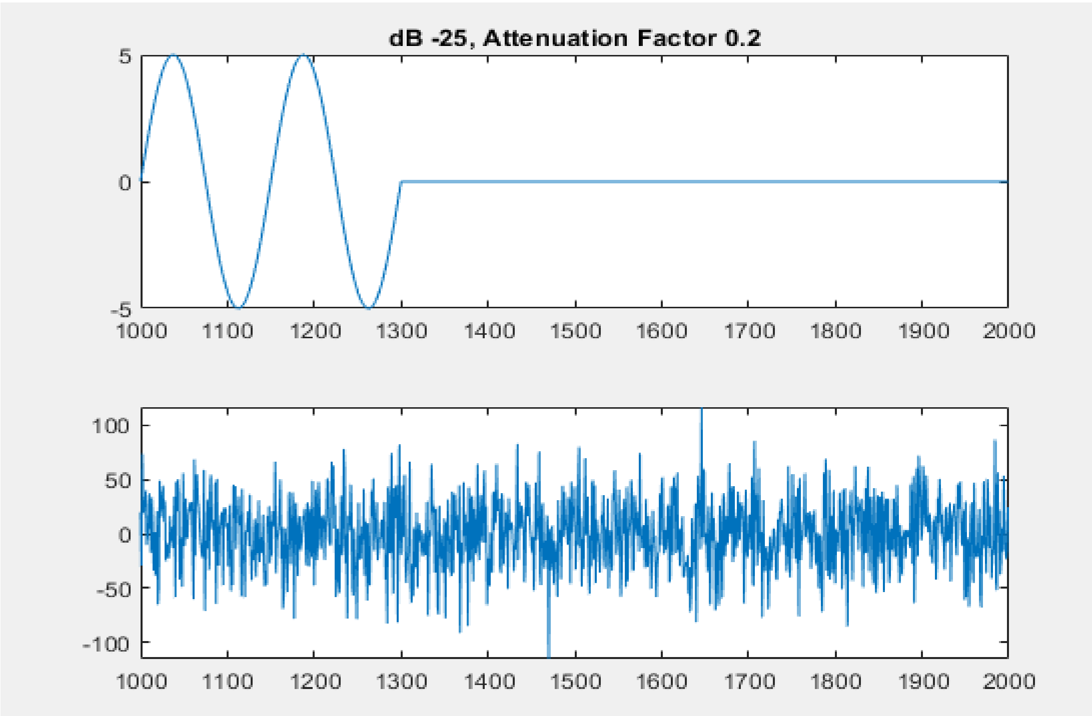


Figure 17: Receive Output Graph at -25 dB and 0.2 Attenuation

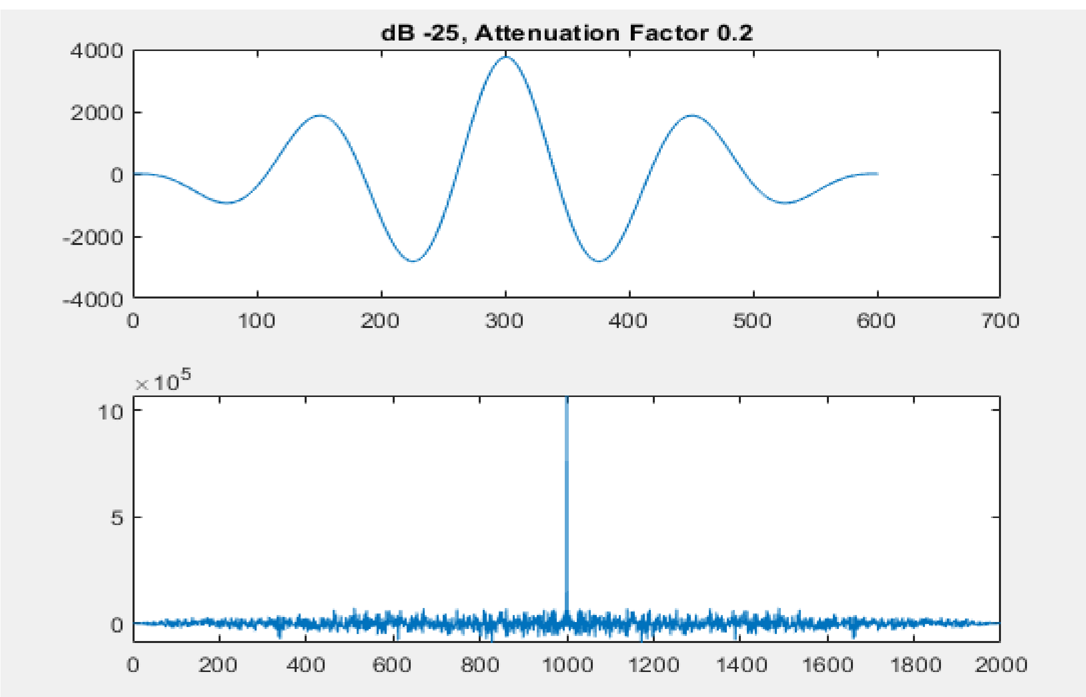


Figure 18: ACF graph at -25 dB and 0.2 attenuation

10. Attenuation=0.5 and SNR=-25db

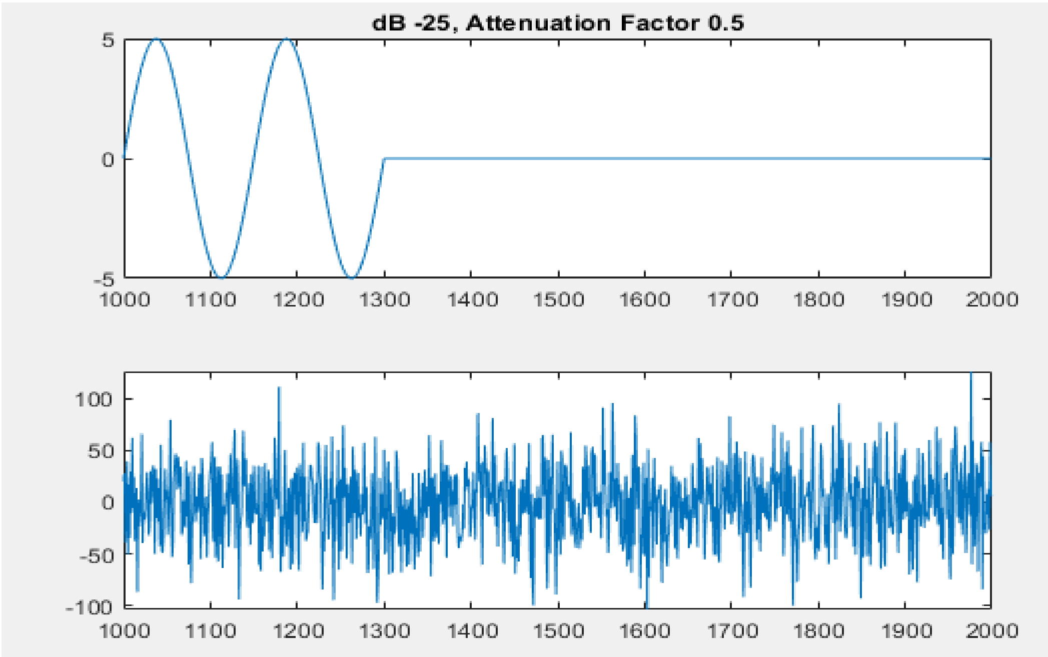


Figure 17: Receive Output Graph at -25 dB and 0.5 Attenuation

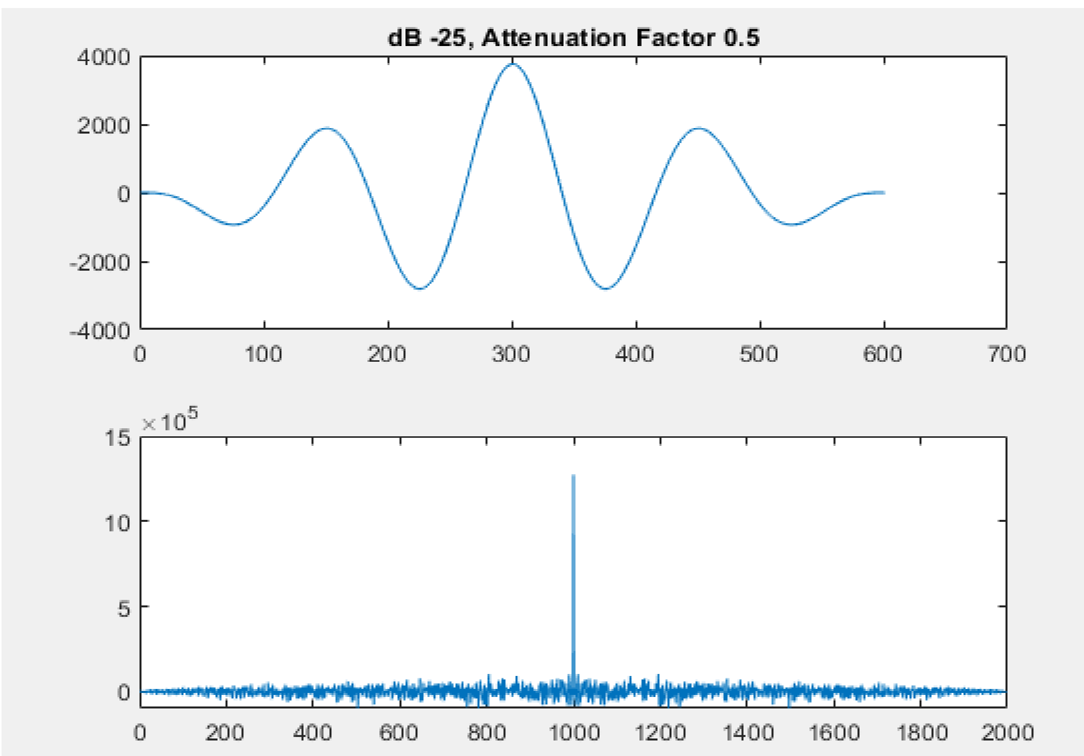


Figure 20: ACF graph at -25 dB and 0.5 attenuation

11. Attenuation=0.7 and SNR=-25db

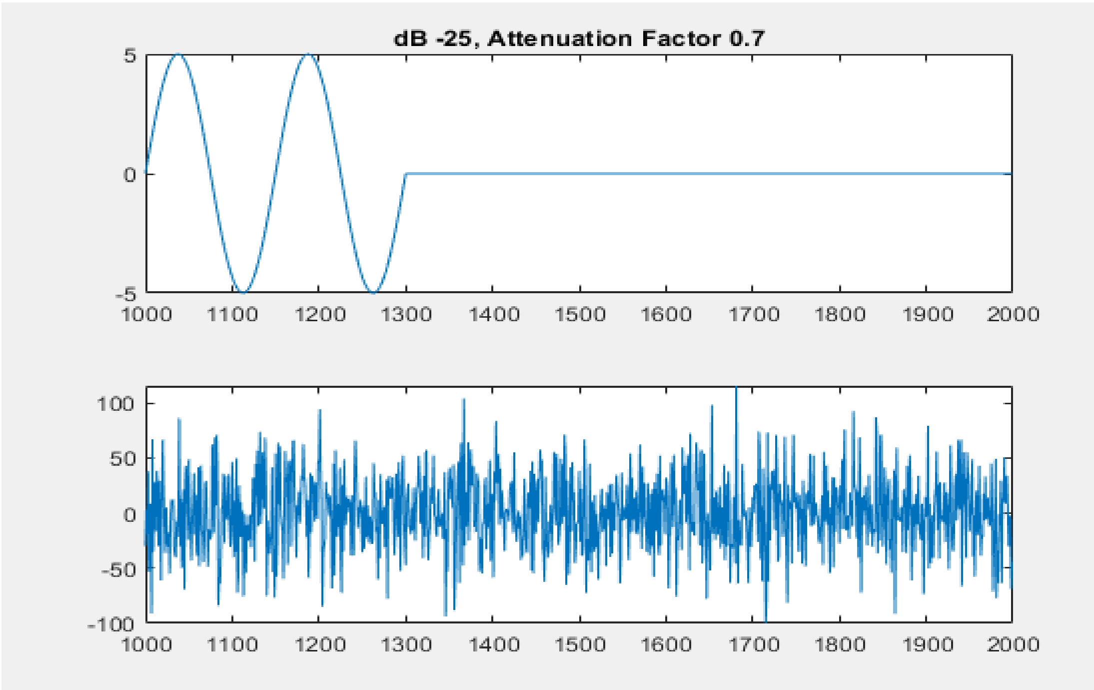


Figure 21: Receive Output Graph at -25 dB and 0.7 Attenuation

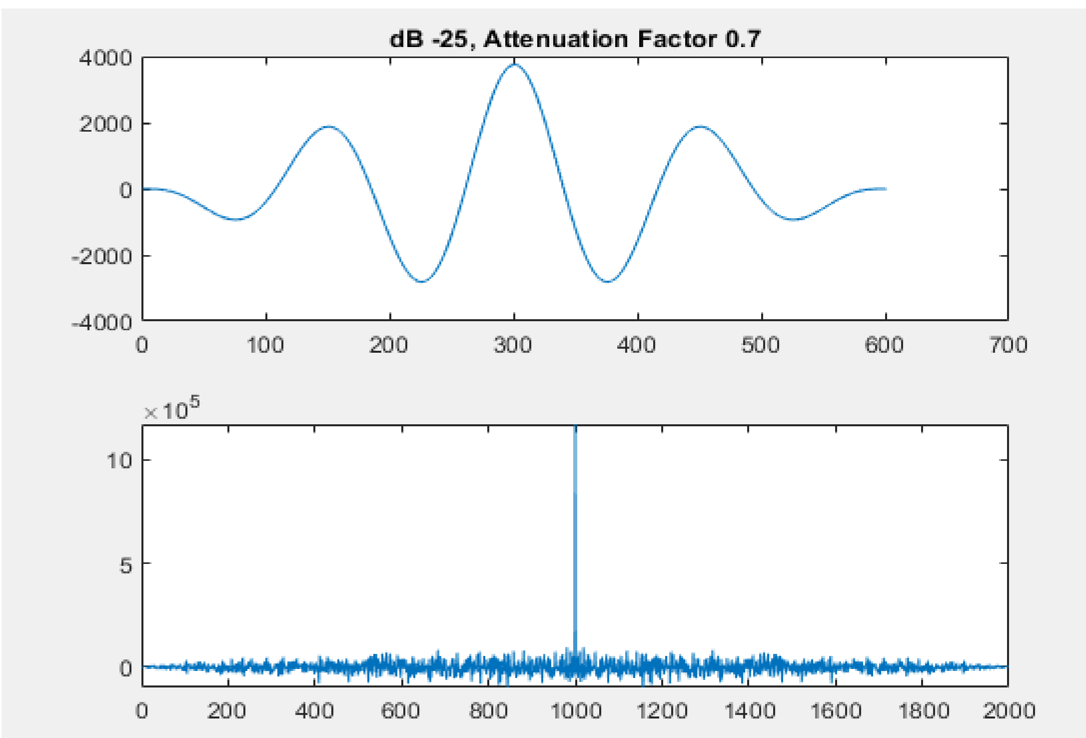


Figure 22: ACF graph at -25 dB and 0.7 attenuation

12. Attenuation=0.9 and SNR=-25db

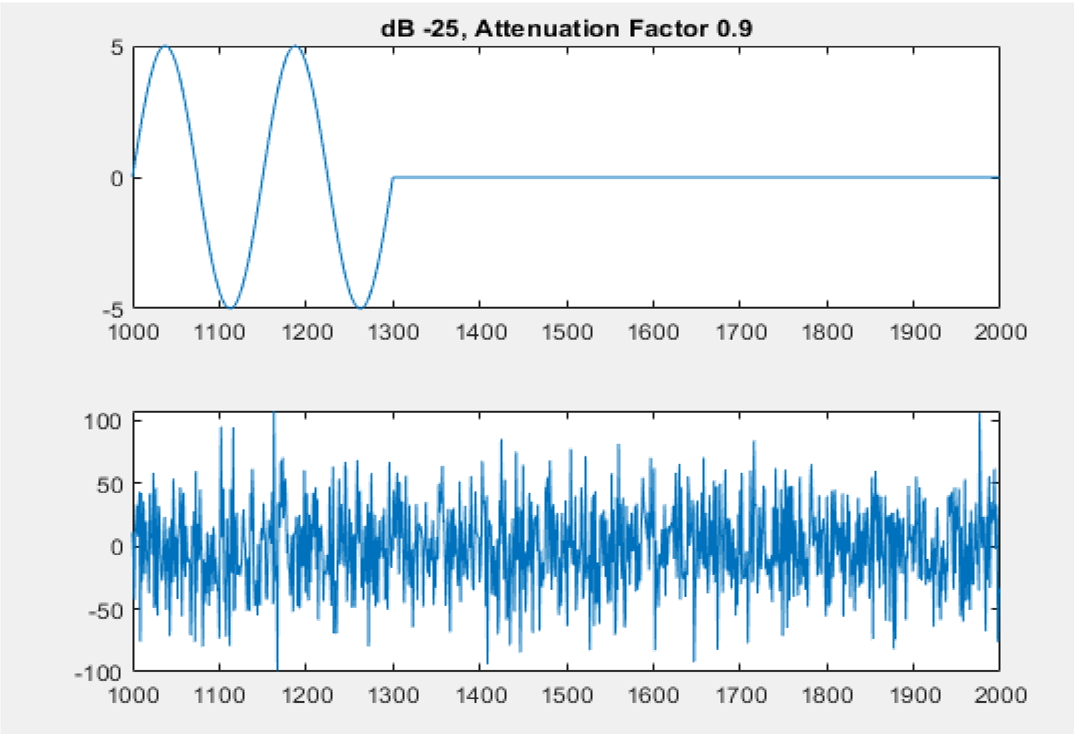


Figure 23: Receive Output Graph at -25 dB and 0.9 Attenuation

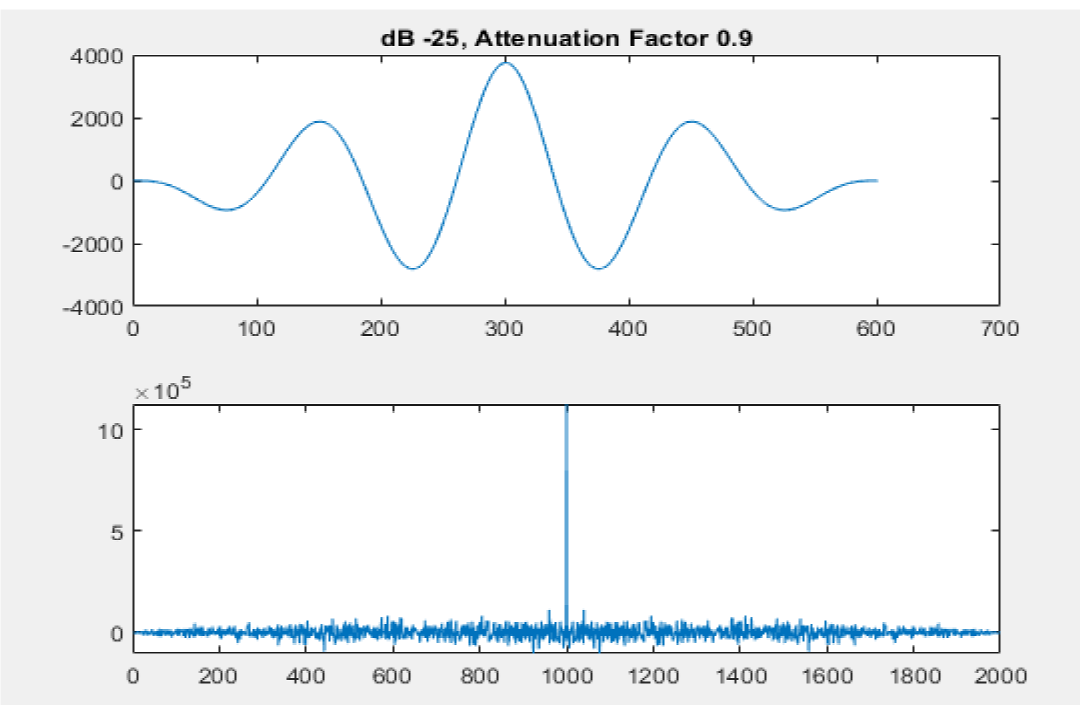


Figure 24: ACF graph at -25 dB and 0.9 attenuation

13. Attenuation=0.2 and SNR=-35db

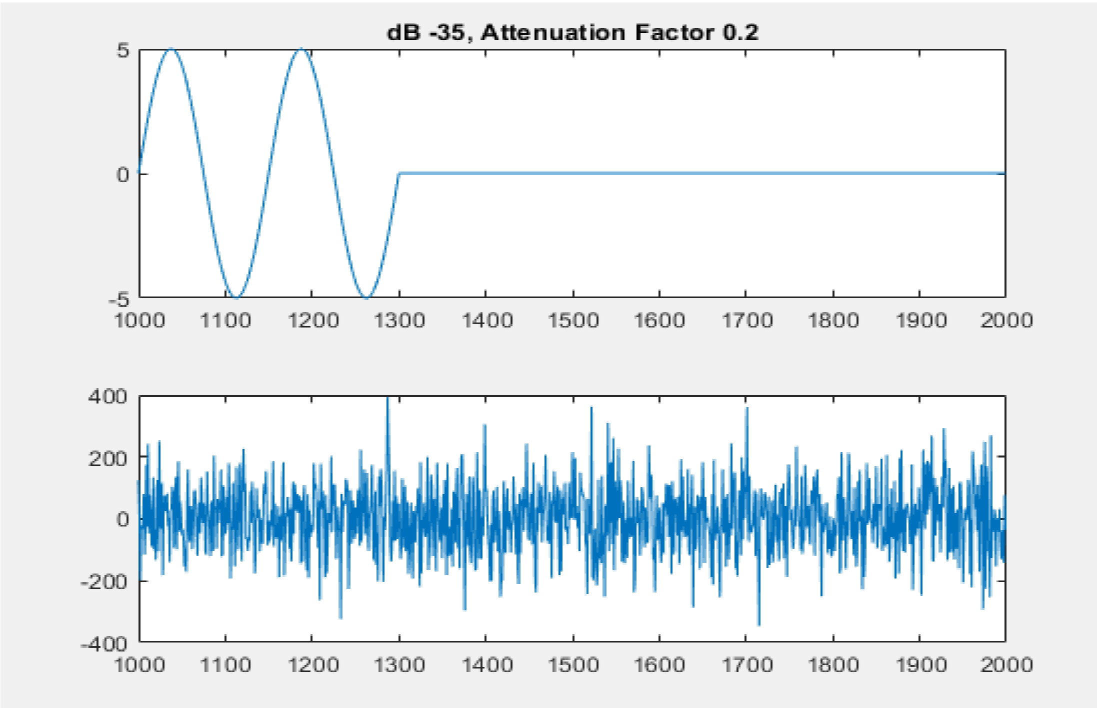


Figure 25: Receive Output Graph at -35 dB and 0.2 Attenuation

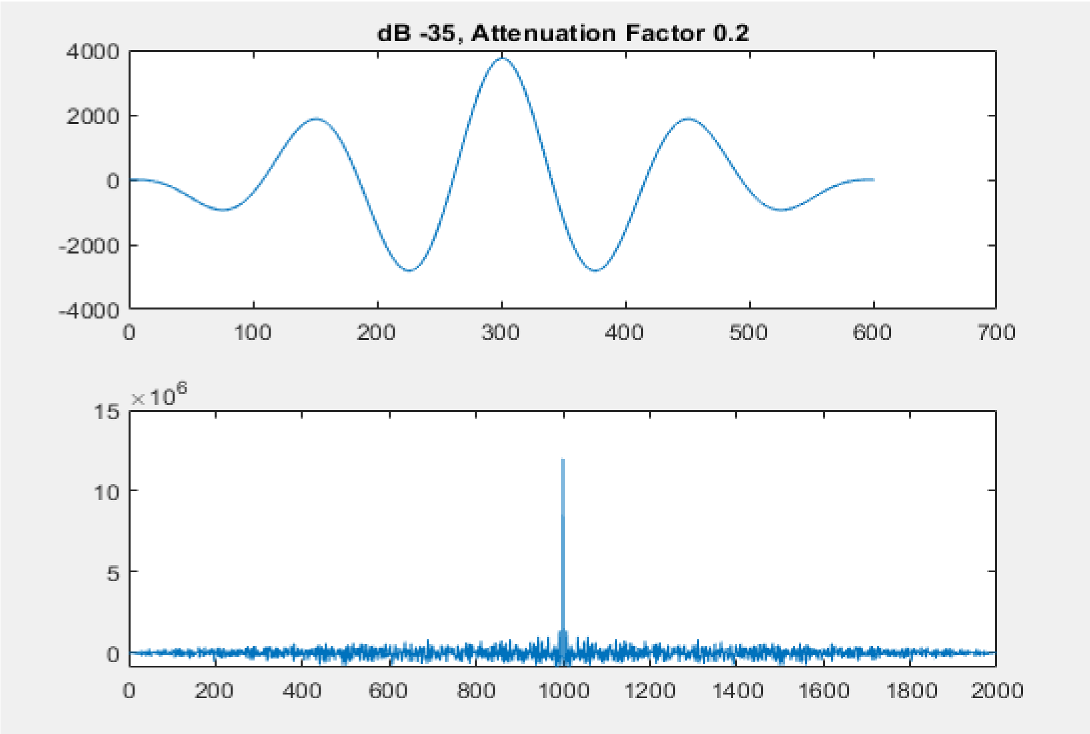


Figure 26: ACF graph at -35 dB and 0.2 attenuation

14. Attenuation=0.5 and SNR=-35db

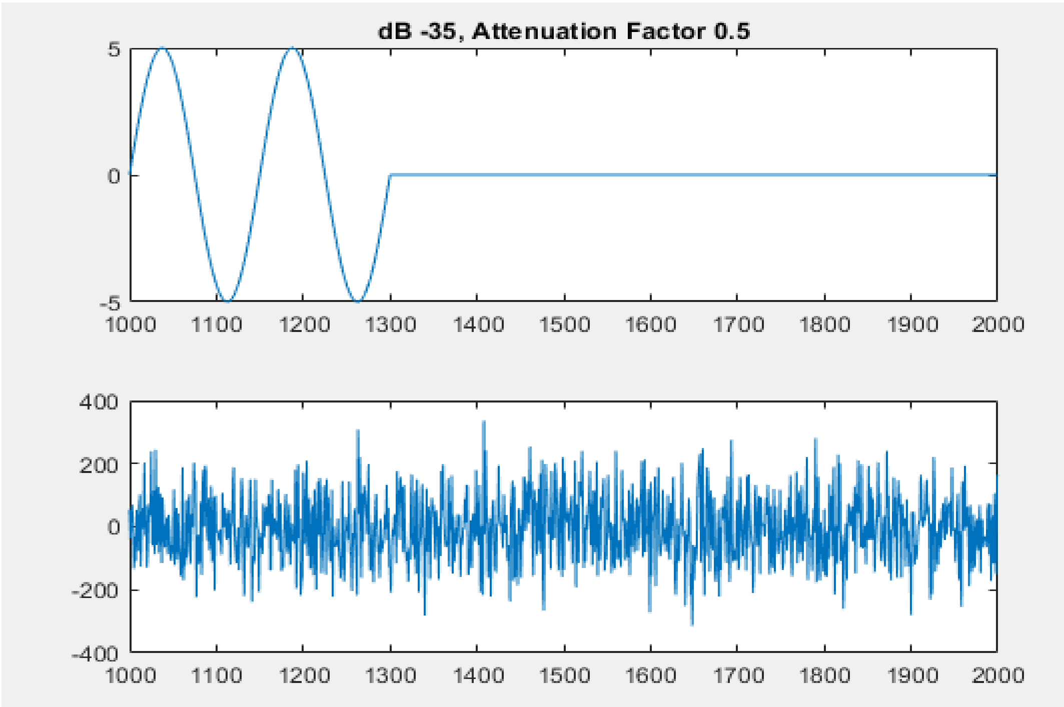


Figure 27: Receive Output Graph at -35 dB and 0.5 Attenuation

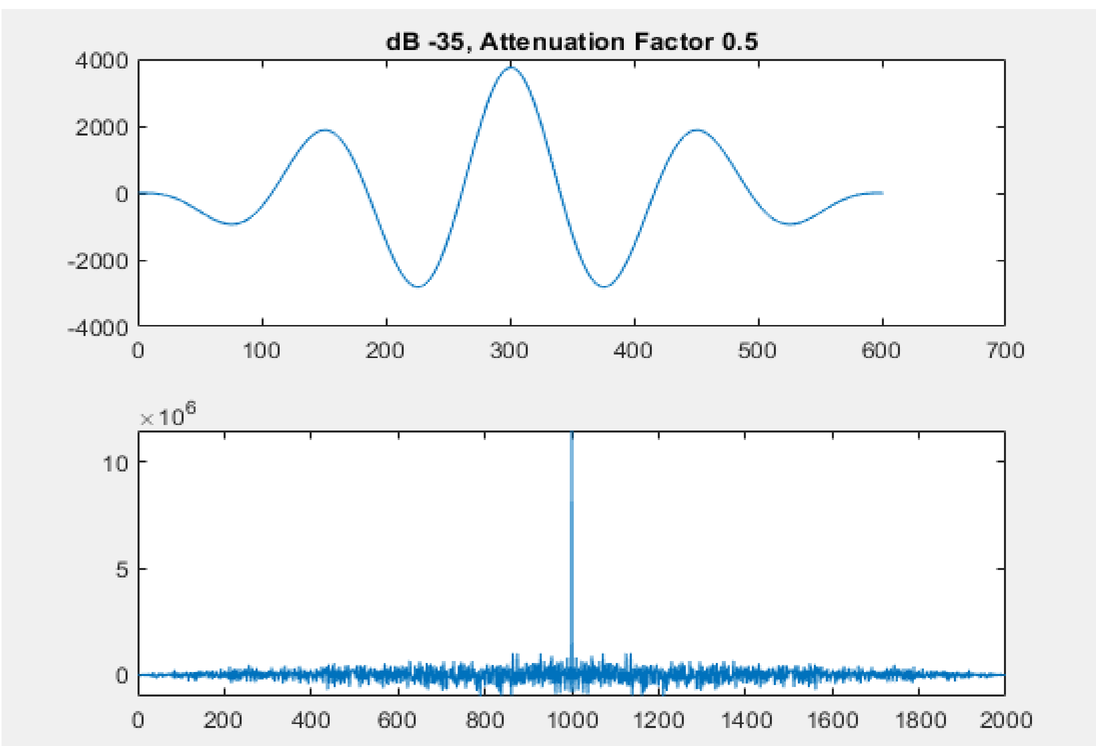


Figure 28: ACF graph at -35 dB and 0.5 attenuation

15. Attenuation=0.7 and SNR=-35db

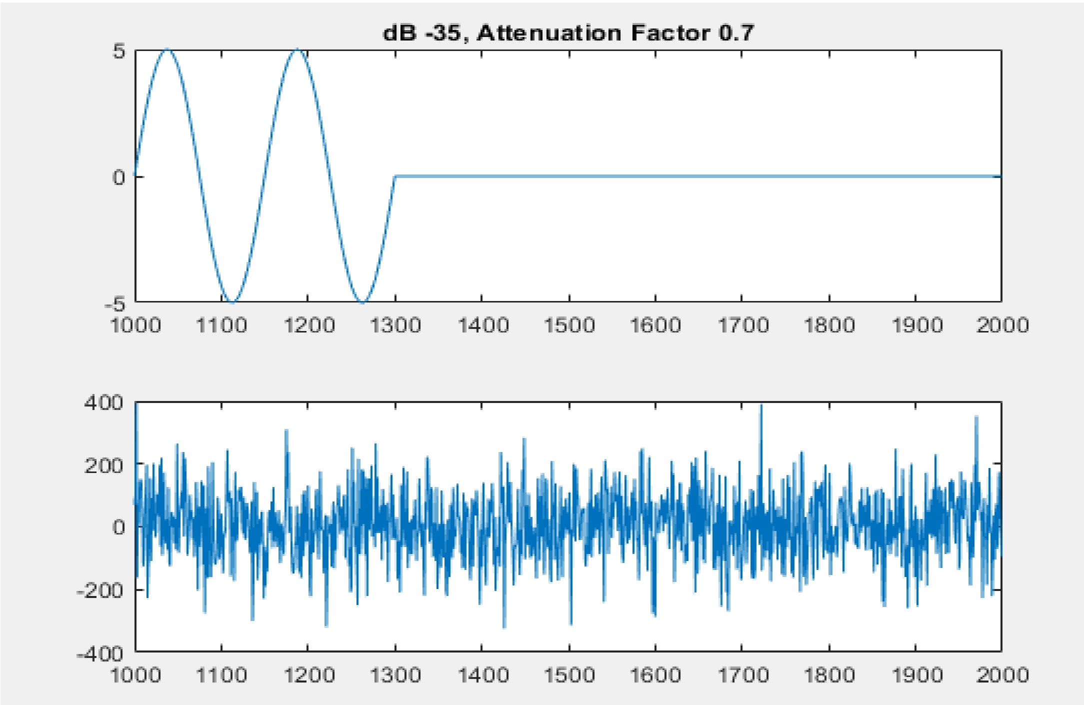


Figure 29: Receive Output Graph at -35 dB and 0.7 Attenuation

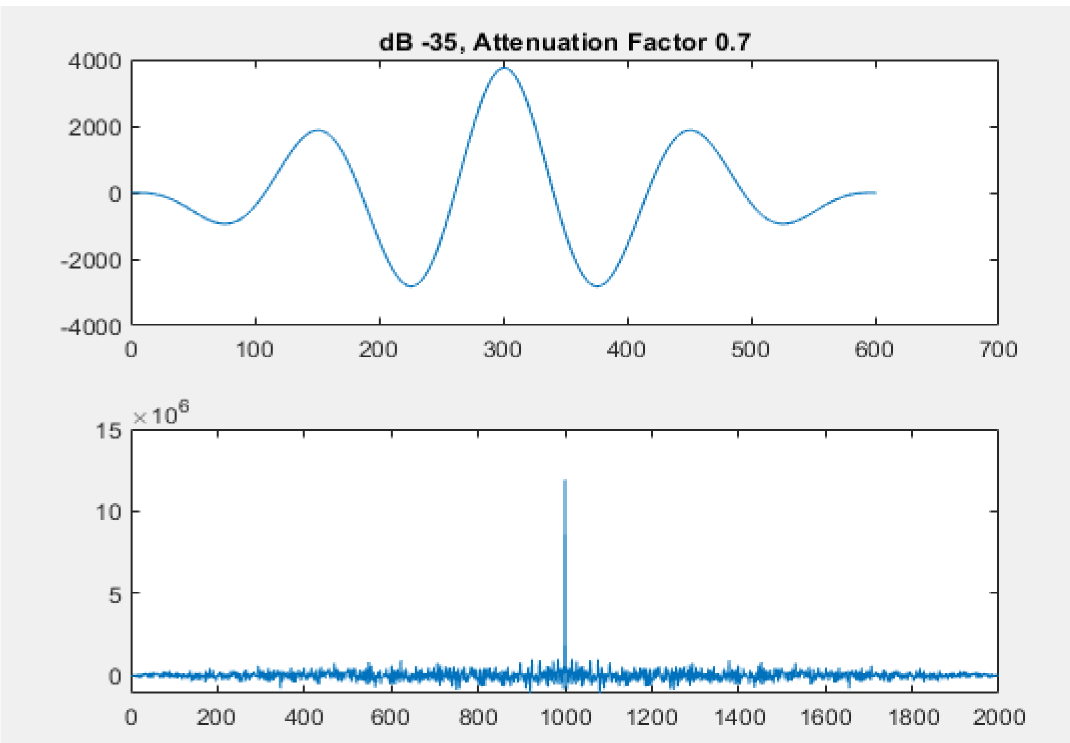


Figure 30: ACF graph at -35 dB and 0.7 attenuation

16. Attenuation=0.9 and SNR=-35db

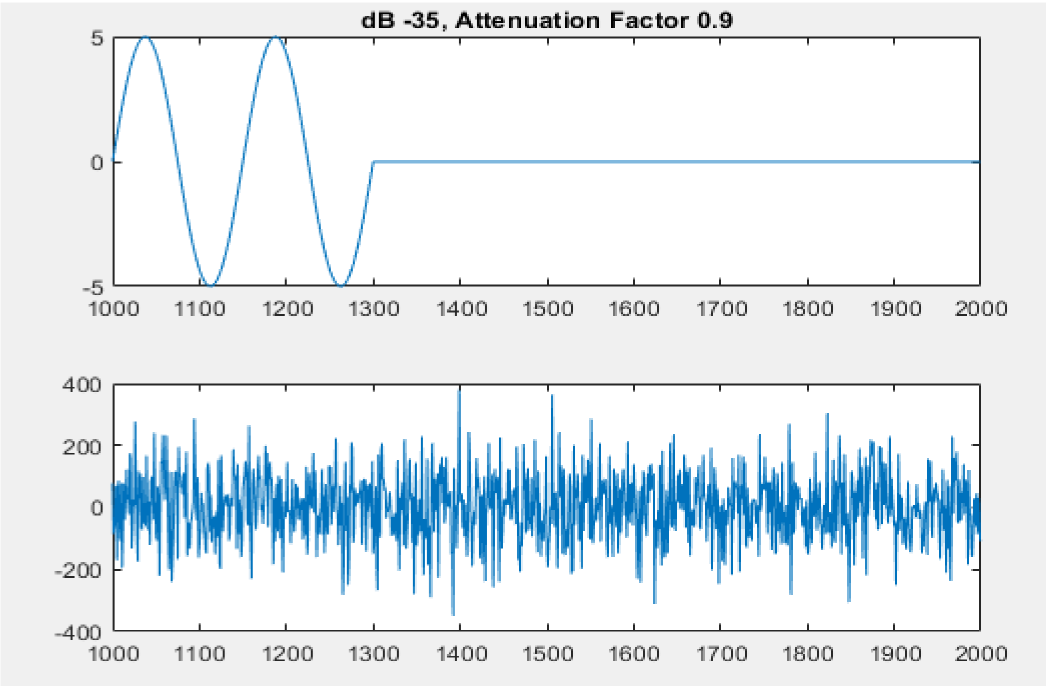


Figure 31: Receive Output Graph at -35 dB and 0.9 Attenuation

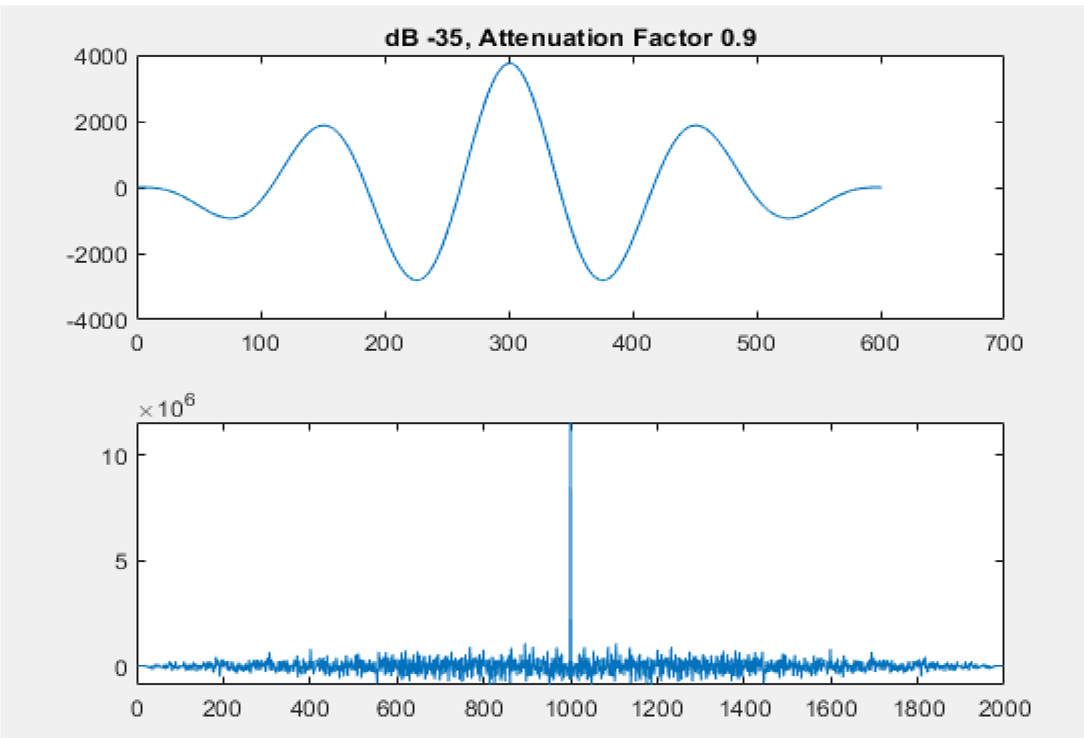


Figure 32: ACF graph at -35 dB and 0.9 attenuation

Conclusion :

From the graphs, we can say that the receiving signal is more clear in -5 dB noise than -15db,-25db and -35 dB noise level. Though in -5 dB ,the signal is clear but also it's a weak signal. As analyzing the period of a white noise signal is so problematic therefore we need to do auto-correlation of the signals. Now, if we look at the attenuation, we can say that the loss of the signal strength in 0.9 attenuation will be great as its losses less signal strength than 0.2, 0.5 and 0.7 attenuation. The more attenuation, the more is the strength of the signal. After analyzing all of the graphs we can conclude that, if we design our radar signal in basis of -5 dB and 0.9 attenuation, we can receive the best frequency reflected by the radar.

Appendix (Matlab code):

```
clc;
close all;
clear all;
T=150;
f=1/T;
tstep=T/150;
t=0:tstep:2*T;
SNR=-35;
atten=0.9;
a=5; %amplitude of the signal
x=a*sin(2*pi*f*t); %input sine wave

xx= zeros(1,1000);
Adelt=0:999;
Adelty=Adelt+1000;
    for i=1:length(x)
    xx(i)=x(i);
    end
     for i=length(x):(length(xx)-length(x))
    xx(i)=0;
    end
length (xx)
P_x=sum(xx.^2)/length(xx);
P_n=P_x/10^(SNR/10); %noise power
n=sqrt(P_n)*randn(1,length(xx)); %generated noise sequence
y=xx+n; %adding signal with noise
yy=atten*xx+n; %adding attenuation with noise signal

c_x=xcorr(x); %finding auto correlation of input signal
```

```
c_n=xcorr(n); %finding auto correlation of noise
c_y=xcorr(y); %finding auto correlation of noisy signal
c_yy=xcorr(yy); %finding auto correlation of attenuation signal


figure
subplot(211),plot(Adelty,xx)
title('dB -35, Attenuation Factor 0.9')
subplot(212),plot(Adelty,yy)



figure
subplot(211),plot(tstep*(1:length(c_x)),c_x)
title('dB -35, Attenuation Factor 0.9')
subplot(212),plot(tstep*(1:length(c_yy)),c_yy)
```

Matlab link:

https://drive.google.com/file/d/1lJY8nzamu46-IkxvZ5DJ4hEkjuDoPjKU/view?usp=sharing