

# חלק מספר 1:

בחלק זה של העבודה קיבלנו קטע קול מלא בתהודה (אקו) והתבקשנו לבנות מסנן אשר במוצאו יתקבל סיגנל נקי ללא תהודה.

על מנת לפתור את הבעיה השתמשנו באלגוריתם  $\text{LMS}$  –  $\text{Least Mean Square}$  המשמש כפילטר אדפטיבי כלומר מעדכן את עצמו.

אלגוריתם זה הוא שיטה חישובית לקביעת המקדמים האופטימליים של המסנן ומציאת המינימום של השגיאה בריבוע.

הגדרות:

$x(n)$  - האות המצוי ה"מלוכלך" אשר נכנס למסנן.

$d(n)$  - האות הרצוי ללא הרעש.

$e(n)$  - השגיאה אשר מוגדת  $e(n) = d(n) - y(n)$  שבאמצעותה נעדכן את מקדמי מסנן  $\text{FIR}$ .

$y(n)$  - האות המתקבל במוצא המסנן.

## Load signal to matlab

```
%טעינת קבצי האודיו לתוך המטבל
clc
[x,Fs]=audioread('resona.m4a');%האות המצוי בכניסה למסנן עם האקו
[d,fs]=audioread('imTitrzi2.m4a');%האות הרצוי - ללא אקו או רעש
p=audioplayer(x,Fs);
a=audioplayer(d,fs);
%לאותות הכניסה יש שני ווקטור עמודה ואילו הפונקציה מקבלת ווקטור אחד ולכן נדרש לדרוס את אחת העמודות.
x=x(:,2); d=d(:,1);
%יש לציין כי לאות הרצוי d שני ווקטורי העמודה זהים אחד לשני ולכן אין כל
%חשיבות איזה מהם דרסנו, אבל לאות המצוי שתי העמודות אינם זהות אך עשינו בדיקה
%במישור המזן וראינו כי הן כמעט זהות ולכן בחרנו את העמודה שבאמצעותה קיבלנו
%שגיאה קטנה יותר.
```

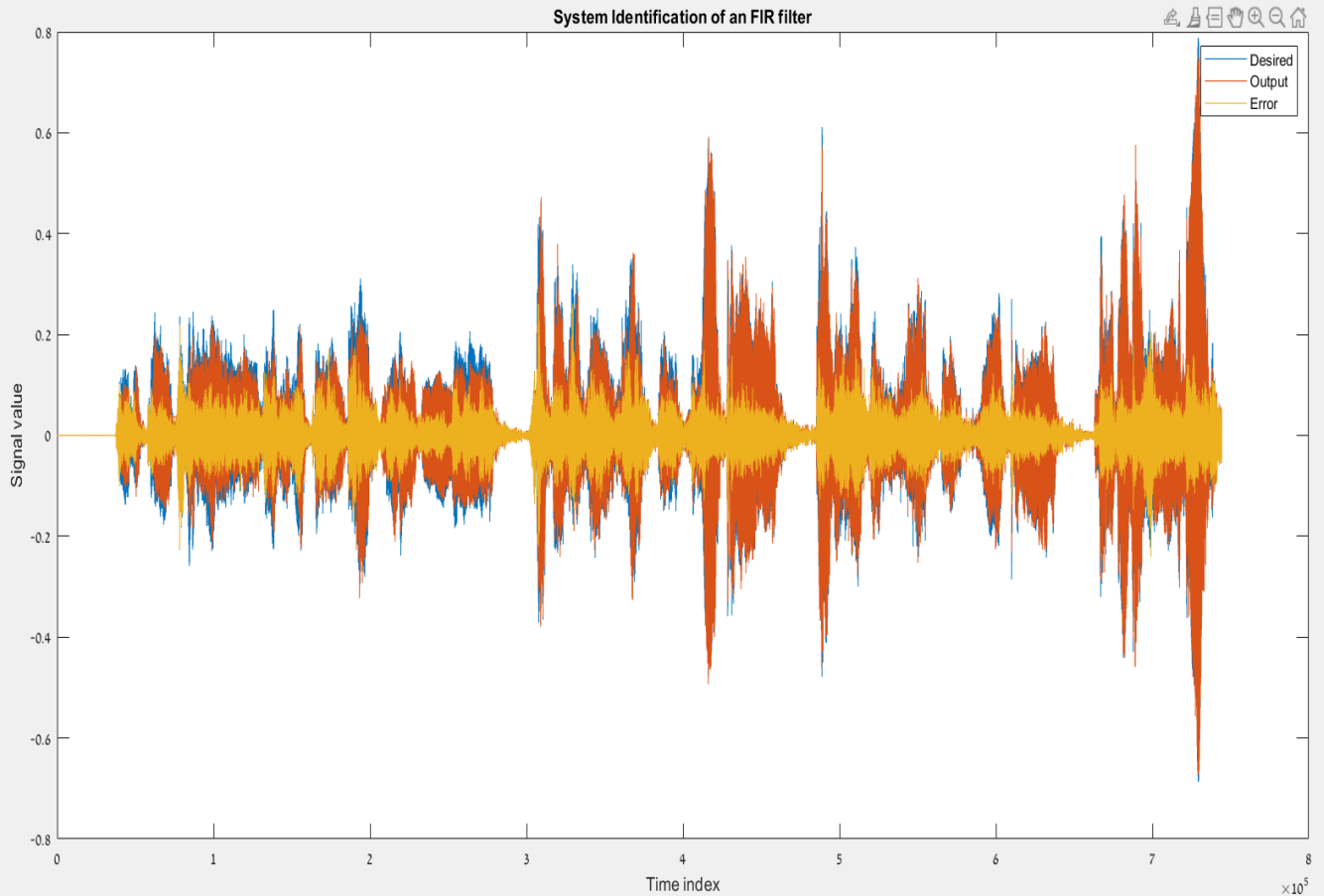
### LMS-Algorithm

```
M=2100;%Filter order
mu=0.01;%Step parameter
lms = dsp.LMSFilter('Length',M,'StepSize',mu);%The LMS algorithm function
[y,e,w] =step(lms,x,d);
plot([d,y,e])%plot Disired,Error,Outpout in the same gaph.
title('System Identification of an FIR filter')
legend('Desired','Output','Error')
xlabel('Time index')
ylabel('Signal value')
%filename = 'OutputNoEco2.m4a';
%audiowrite(filename,y,Fs);
```

לאחר טעינת הסיגנלים למטלב הפעלנו את ה-Fir LMS Algorithm כאשר לאלגוריתם נכנסים האות המצוי והאות הרצוי אשר בעזרתם מחשבים את השגיאה ומעדכנים את מקדמי המסנן הרצויים. הפונקציה מחזירה את אות השגיאה שהוגדר בתחילת השאלה אות מוצא המסנן ואת מקדמי המסנן האופטימליים.

**\*\* קובץ האודיו של מוצא המסנן  $y(n)$  מצורף בסוף. שם הקובץ: OutputNoEco2.m4a \*\***

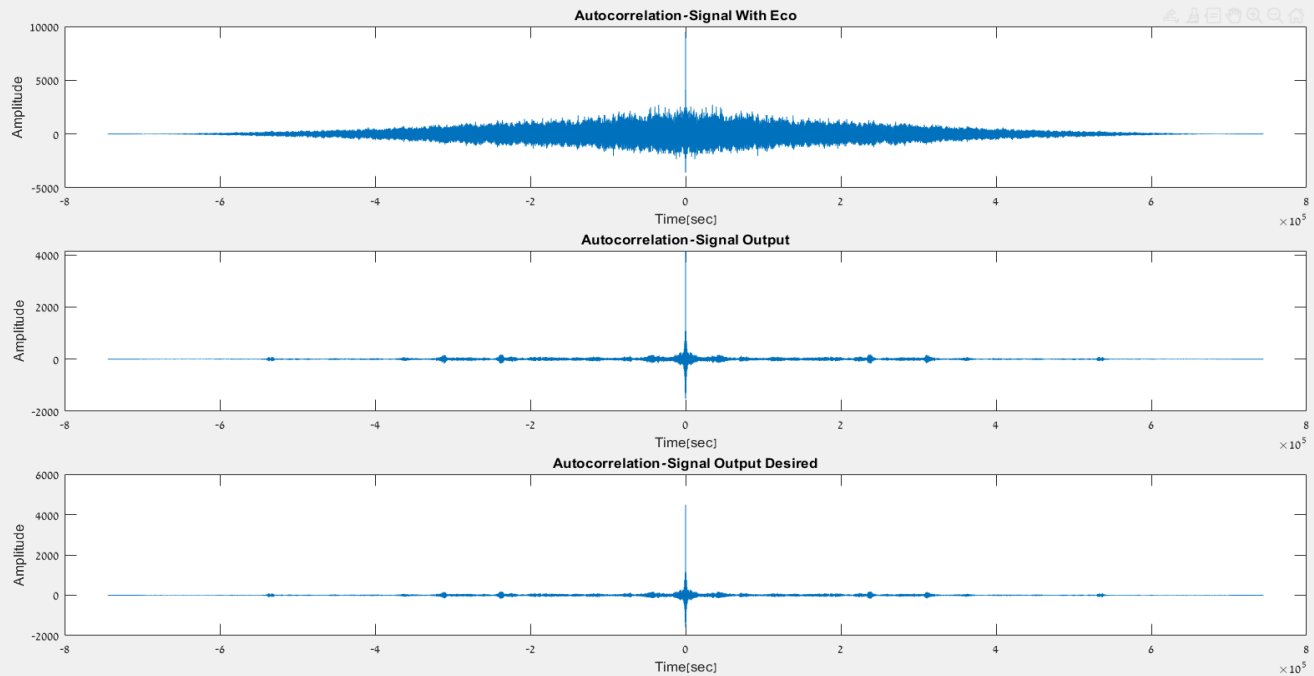
## גרף המראה את מוצא המסנן, האות הרצוי והשגיאה



### איור מספר - 1

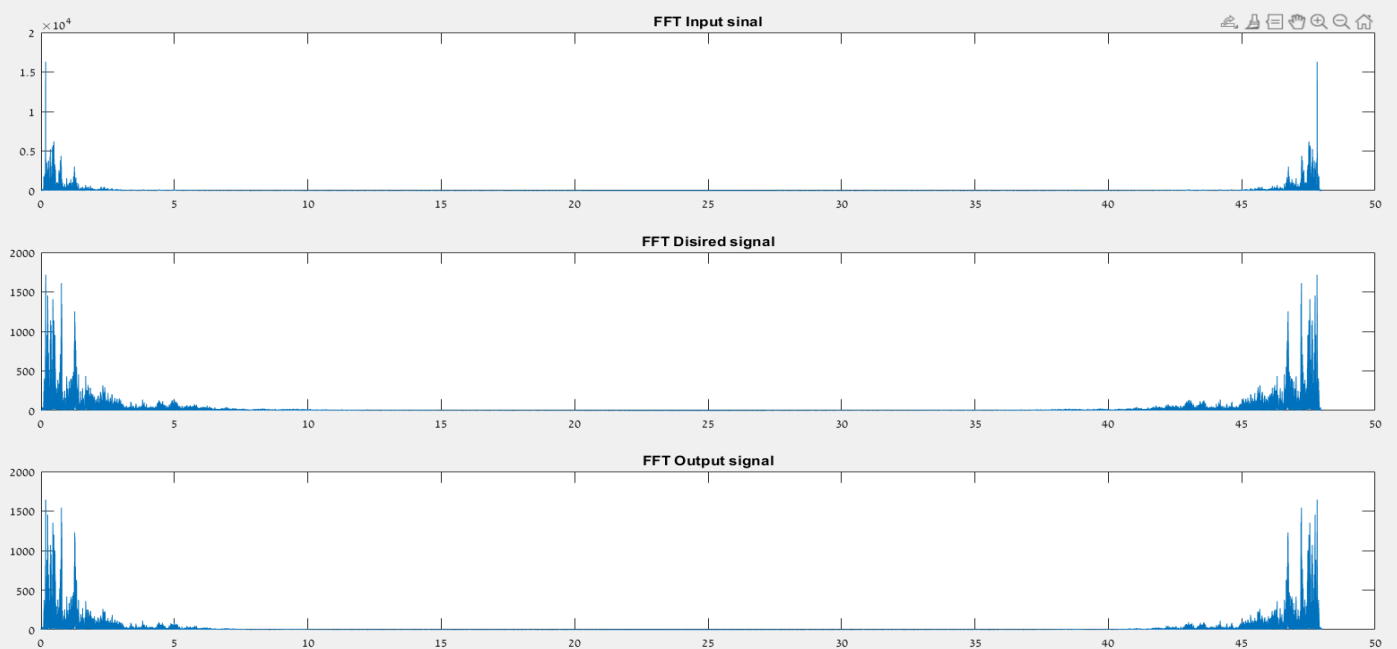
ניתן לראות כי אות המוצא בצבע האדום כמעט עולה כולו על האות הרצוי הכחול כלומר המסנן האדפטיבי פעל הייטיב, ניתן לשמוע בקובץ האודיו כי הכן כמעט וכל האקו נוקה כפי שהתבקשנו לבצע.

## גרף המראה את פונקציית האוטו – קורלציה של האותות



## איור מספר - 2

### התמרת פורייה על אות הכניסה, אות המוצא והאות הרצוי

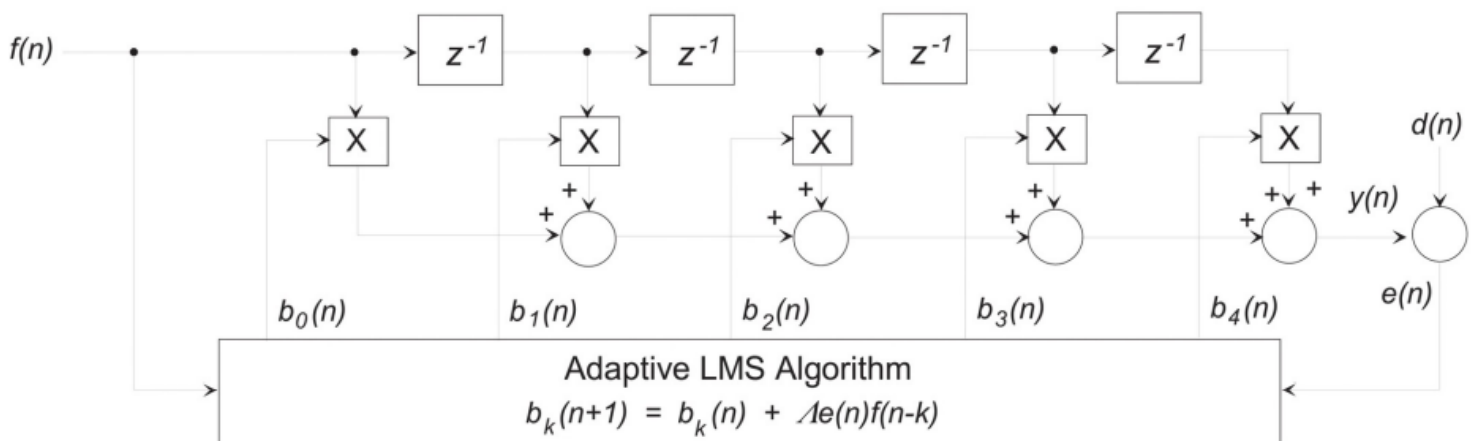


## איור מספר - 3

חזרנו על החלק הראשון רק שבחלק זה כתבנו אנחנו את הפונקציה המתבססת על אלגוריתם LMS.

%כתבנו פונקצייה המתבססת על אלגוריתם lms אשר מקבלת את האות הרצוי והאות המצוי  
%ומחזירה את מקדמי המנסן השגיאה והסיגנל המתקבל במוצא המסנן.

```
function [y,e,h]=LmsAlgorithm(x,d,)%The name of the function
N=250;%Filter order
h=zeros(1,N);%Initial zeros vector for filter coefficients
mu=0.0117;
for k=1:length(d)
    y(k)=h(k)'.*x(k);
    e(k) = d(k)-y(k);%חישוב השגיאה
    h(k+1)= h(k)+ mu*(e(k).*x(k));%עידכון מקדמי המנסן
end
return
```



```
%קריאה לפונקציה שכתבנו ושמירת קובץ השמע
[y,e,h]=LmsAlgorithm(x,d);
B=y';
A=e';
%sound(A,fs)
%clear sound
filename = 'OutputNoEcol.m4a';
audiowrite(filename,A,Fs);
```

**\*\*קובץ האודיו של מוצא המסנן (ח) מצורף בסוף. שם הקובץ: OutputNoEco1.m4a\*\***

## חלק מספר 2:

בחלק זה של העובדה קיבלנו סיגנל עם הפרעת אקו כמו בחלק הראשון אך עם תוספת של הסחת תדר. התבקשנו לתכן מסנן אשר יתקן את הסחת התדר וינקה את האות מהאקו.

לביצוע המשימה השתמשנו במסנן FIR RLS Algorithm אשר פועל בשיטה דומה למסנן המבוטס LMS בחלק הראשון..

```
function [e,w,delta] = RLSFilterI232(x,d)
    p      = 30;                % filter order
    lambda  = 0.999;            % forgetting factor
    laminv  = 1/lambda;
    delta   = 1.0;              % initialization parameter
    % Filter Initialization
    w       = zeros(p,1);       % filter coefficients
    P       = delta*eye(p);     % inverse correlation matrix
    for m = p:length(d)
        y = x(m:-1:m-p+1); % Acquire chunk of data
        e(m) = d(m)-w'*y;% Error signal equation
        Pi = P*y;% Parameters for efficiency
        k = (Pi)/(lambda+y'*Pi); % Klman Filter gain vector update
        P = (P - k*y'*P)*laminv;% Inverse correlation matrix update
        w = w + k*e(m);% Filter coefficients adaption

        clc
    end
    x=audioread('sampeling.m4a');
    return [d,fs]=audioread('imTitrzi2.m4a');
    x=x(:,1);d=d(:,1);
    [y,w,delta] = RLSFilterI232(x,d);
    y=y';
    plot([d,y])
    title('System Identification of an FIR filter')
    legend('Desired','Output')
    xlabel('Time index')
    ylabel('Signal value')

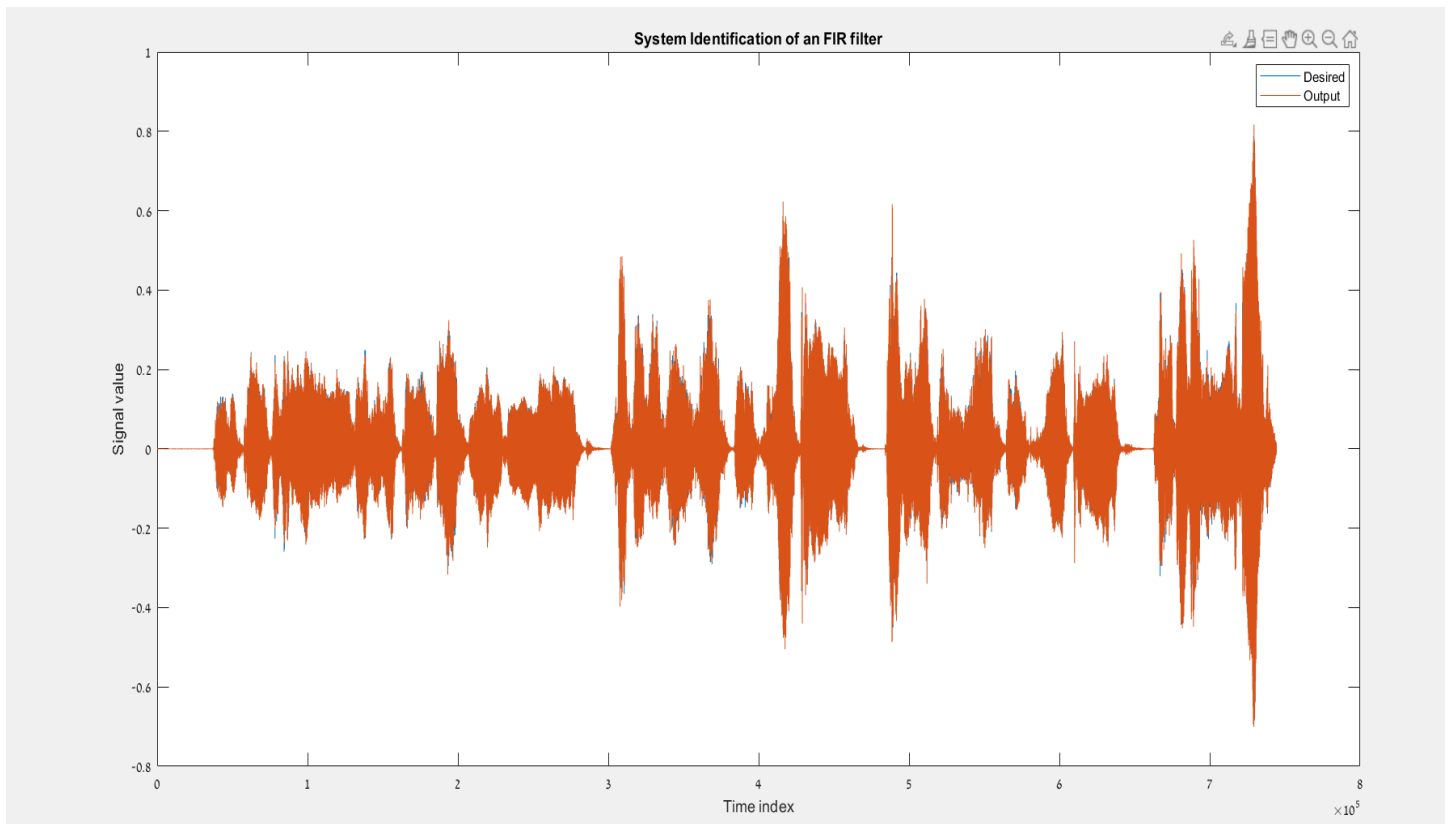
    %%sound(x,fs)
    %clear sound

    %%filename = 'OutputNoEcoNoDiv.m4a';
    %%audiowrite(filename,e,fs);
```

**\*\*קובץ האודיו של מוצא המסנן (ח) מצורף בסוף. שם הקובץ:**

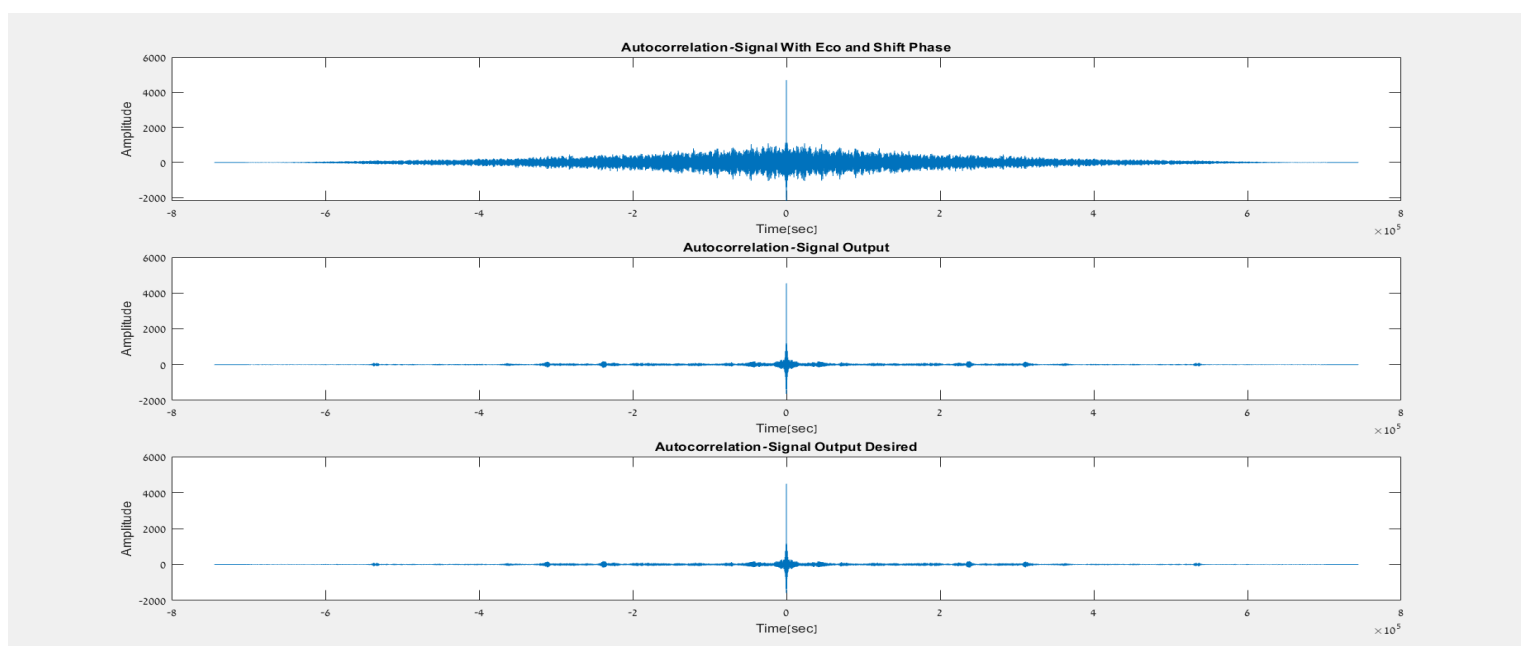
**\*\*OutputNoEcoNoDiv.m4**

**גרף המראה את אות המוצא והאות הרצוי- חפיפה כמעט מלאה**



#### איור מספר - 4

**פונקציה אוטו-קורולציה של אות הכניסה, אות המוצא והאות הרצוי**



## איור מספר - 5

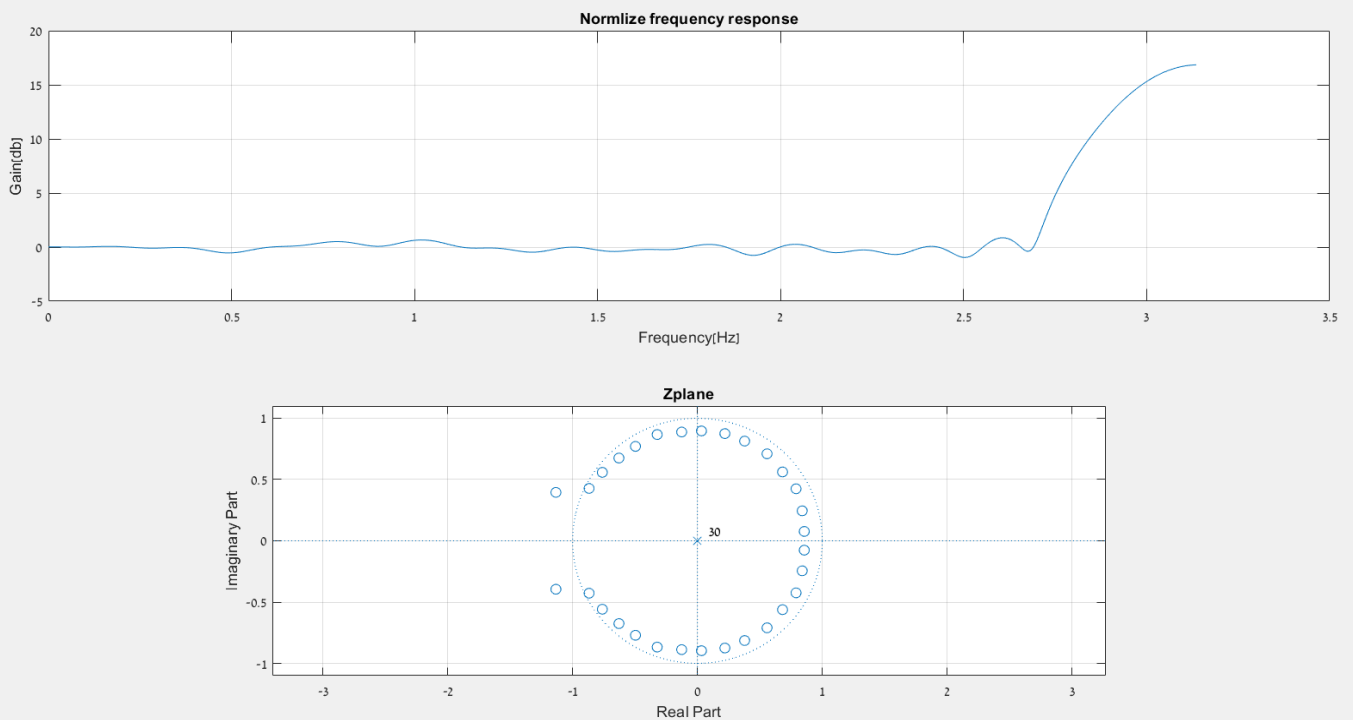
```
%בחלק זה בחרנו להראות את תגובת התדר של המסנן שנוצר באמצעות האלגוריתם.
h_overall = [ 1 w'];
[H,j] = freqz(h_overall,1); % יצירת ווקטור של N+1 איברים כאשר N הוא סדר המסנן
zplane(h_overall,1)
V=10*log10((abs(H))); % יצירת מערך של הערך המוחלט של תגובת התדר

subplot (2,1,1);
plot(j,V)
title 'Normlize frequency response'
xlabel ('Frequency[Hz]')
ylabel ('Gain[db]')
grid on
hold on

subplot (2,1,2);
zplane(h_overall,1)
title 'Zplane'
grid on

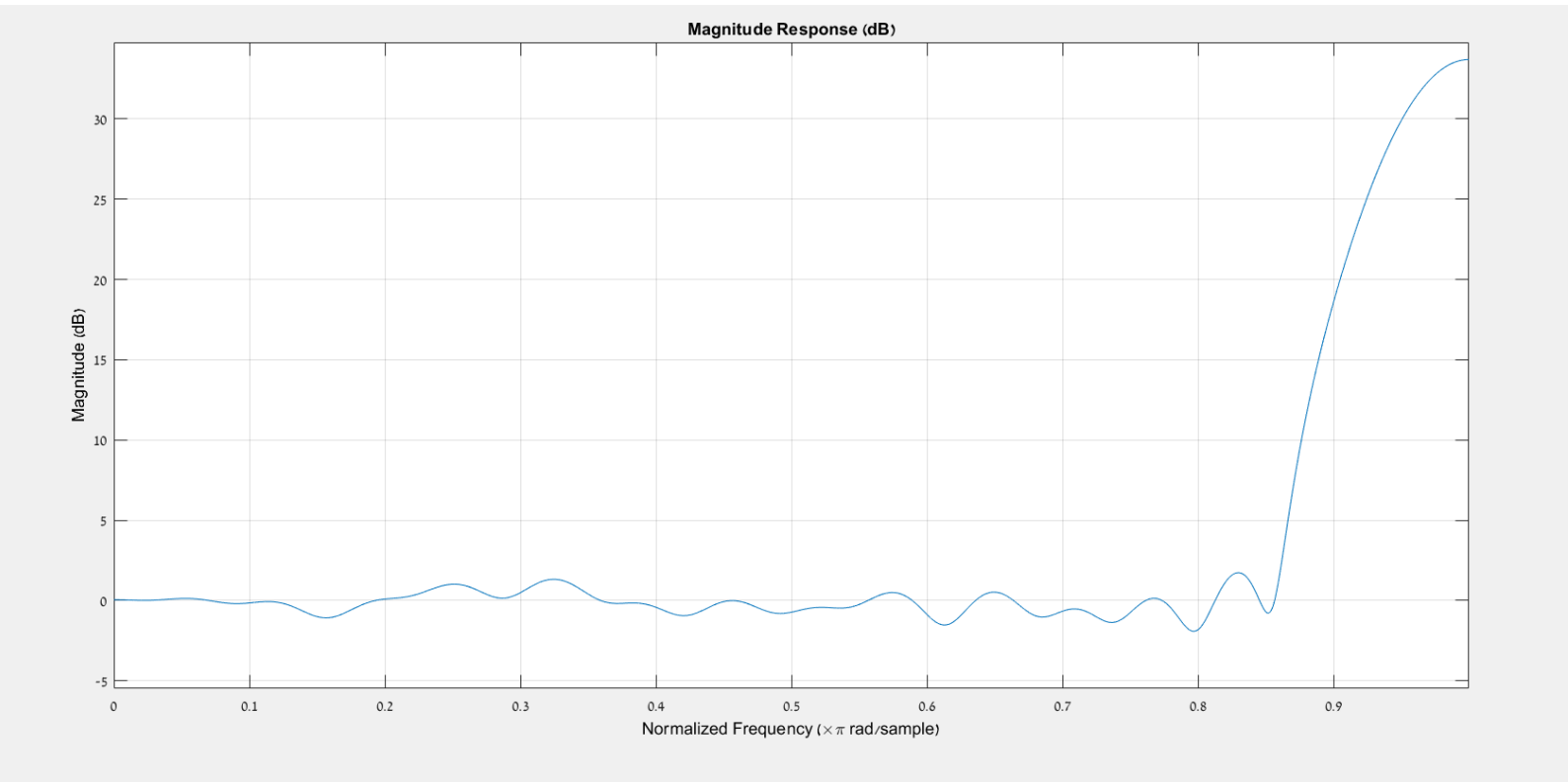
fvtool(h_overall)
```

### תגובת התדר ומפת הקטבים והאפסים של מסנן FIR





איור מספר - 6



איור מספר - 7

## הספקים:

חלק ראשון:

```
--  
k=xcorr(x,x);%Autocorrelation function  
L=length(x);  
%plot(-(L-1):(L-1),k)  
power_Input=max(k)%max value of Autocorrelation function(this is the power of the input signal)  
  
Output_Corla=xcorr(y,y);%Autocorrelation function  
power_Output=max(Output_Corla)  
  
Error_Coral=xcorr(e,e);  
power_Error=max(Error_Coral)  
  
SNR=10*log10(power_Output/power_Error) % יחס אות לרעש הוא יחס ההפסקים של אות המוצא ואת הכניסה.  
% נרצה תמיד כי יחס זה יהיה גדול מ 1 כלומר הספק המוצא גדול מהספק הרעש.
```

```
power_Input =  
  
9.4858e+03  
  
power_Output =  
  
4.1572e+03  
  
power_Error =  
  
453.2532  
  
SNR =  
  
9.6246
```

## חלק שני:

```
Input_Corla=xcorr(x,x);%Autocorrelation function
L=length(x);
%plot(-(L-1):(L-1),k)
power_Input=max(Input_Corla)%max value of Autocorrelation function(this is the power of the input signal)

Output_Corla=xcorr(y,y);%Autocorrelation function
power_Output=max(Output_Corla)
```

```
power_Input =

    4.6901e+03

power_Output =

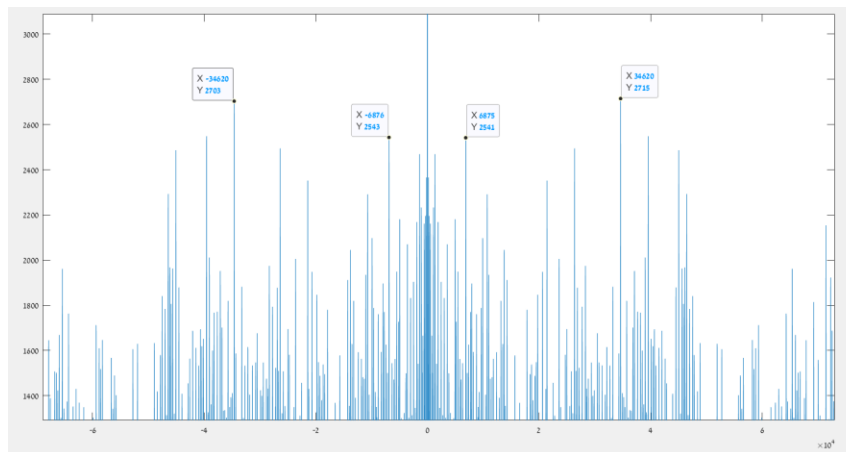
    4.5444e+03
```

\*ניתן לראות כי כמעט לא איבדנו מידע ההספק במוצא שווה כמעט להספק בכניסה.

## סיכום ומסקנות:

### חלק ראשון:

- בחלק הראשון קיבלנו סיגנל מלא באקו, תחילה העברנו את האות למשור התדר על מנת לבדוק האם ניתן לפתור את הבעיה באמצעות פילטרים דיגיטליים כפי שנלמד בקורס. ניתן לראות באיור מספר 3 כי במישור התדר אין כמעט שינוי בין האות בכניסה לאות במוצא כלומר קשה מאוד להחליט איפה האקו ואיך הוא השפיע על האות המקורי ולכן פנינו למישור הזמן על מנת לנסות לקבל מידע.
- את השפעת האקו במישור הזמן ראינו באמצעות פונקציה אוטו-קורולציה כפי שניתן לראות באיור הבא:



באיור ניתן לראות את פונקציית האוטו קורולציה של אות הכניסה, באמצעות הצגה זאת הצלחנו להבחין בתופעת האקו ש"ליכלכה" את הסיגנל(באיור הערכים הכפולים בחלק החיובי ובחלק השלילי).

- בגלל חוסר היכולת להתמודד עם בעיית האקו במישור התדר פתרנו את הבעיה בעזרת Leas Mean Square Algorithm אשר מקבל את אות הכניסה מלא באקו ואת הסיגנל הרצוי הנקי ומחשב את השגיאה ביניהם-וכך הצלחנו להתגבר על בעיית האקו כפי שניתן לשמוע בקבצי האודיו המצורפים.

## חלק שני

- בחלק זה של העבודה קיבלנו סיגנל עם בעיית אקו כמו בחלק הקודם אך עם הזזה בתדר-הסחת תדר.  
תחילה ניסינו לפתור את הבעיה עם הפונקציה שכתבנו בחלק הראשון של המטלה אשר מתבססת על אלגוריתם LMS, התוצאות היו דיי טובות אך באמצעות אלגוריתם RLS (אשר איתו התבקשנו לפתור את החלק השני) אשר בדומה ל-LMS גם הוא עובד על עיקרון חישוב השגיאה בין האות הרצוי לאות המצוי ומציאת המקדמים האופטימליים של המסנן הצלחנו להגיע לתוצאות טובות יותר.
- לאלגוריתם זה התווסף פרמטר חשוב אשר הופך אותו שונה מאלגוריתם ה-LMS הנקרה Forgetting Factor המסומן באמצעות האות היוונית  $\lambda$  כאשר גודלו של פרמטר זה הוא  $0 < \lambda < 1$ . מטרתו של פרמטר זה הוא להקטין את השגיאה בצורה משמעותית יותר.  
פונקציית השגיאה הנקראת פונקציית המחיר החדשה שתתקבל:

$$\varepsilon^2 = \sum_{i=0}^n \lambda^{n-i} \cdot (d[i] - y[i])^2$$

באמצעות אלגוריתם זה הצלחנו להגיע לצמצום בצורה משמעותית את השגיאה כפי שניתן לראות באיור מספר 4, אות המוצא כמעט זהה לאות הרצוי כלומר השגיאה אפסית.

## ביבליוגרפיה:

- ספרים הקשורים לעיבוד אותות ואותות אקראיים:  
1. Solution\_scott\_l\_miller\_donald\_g\_childer  
2. Probability-and-Random-Processes-With-Applications-to-Signal-Processing-and-Communications-2nd-Edition-by-Scott-L.Miller-Donald-Child.  
• לכתובת אלגוריתם ה RLS נעזרנו באלגוריתמים מתוך ל MathWork.  
קישור: <https://www.mathworks.com/matlabcentral/fileexchange/36858-recursive-least-squares-filter>

## קוד המטלב לכל המטלה:

### חלק ראשון:

```
clc
[x,Fs]=audioread('resona.m4a');
[d,fs]=audioread('imTitrzi2.m4a');
p=audioplayer(x,Fs);
a=audioplayer(d,fs);
x=x(:,2); d=d(:,1);
%%
k=xcorr(x,x);%Autocorrelation function
L=length(x);
plot(-(L-1):(L-1),k)
power_Input=max(k)%max value of Autocorrelation
function(this is the power of the input signal)

Output_Corla=xcorr(y,y);%Autocorrelation function
power_Output=max(Output_Corla)

Error_Coral=xcorr(e,e);
power_Error=max(Error_Coral)

SNR=10*log10(power_Output/power_Error) % יחס אות לרעש
הוא יחס ההפסקים של אות המוצא ואת הכניסה.
% נרצה תמיד כי יחס זה יהיה גדול מ1 כלומר הספק המוצא
גדול מהספק הרעש.

%%
M=2100;%Filter order
mu=0.01;%Step parameter
lms = dsp.LMSFilter('Length',M,'StepSize',mu);%The
LMS algorithm function
[y,e,w] =step(lms,x,d);
plot([d,y,e])%plot Disired,Error,Outpout in the same
gaph.
title('System Identification of an FIR filter')
legend('Desired','Output','Error')
xlabel('Time index')
ylabel('Signal value')
```

```

%filename = 'OutputNoEco2.m4a';
%audiowrite(filename,y,Fs);
%%
subplot (3,1,1);
k=xcorr(x,x);
L=length(x);
plot(-(L-1):(L-1),k)
title 'Autocorrelation-Signal With Eco'
xlabel 'Time[sec]'
ylabel 'Amplitude'
hold on

subplot (3,1,2);
hy=xcorr(y,y);
L=length(y);
plot(-(L-1):(L-1),hy)
title 'Autocorrelation-Signal Output'
xlabel 'Time[sec]'
ylabel 'Amplitude'
hold on

subplot (3,1,3);
hd=xcorr(d,d);
L=length(d);
plot(-(L-1):(L-1),hd)
title 'Autocorrelation-Signal Output Desired'
xlabel 'Time[sec]'
ylabel 'Amplitude'
hold on

%%
subplot(3,1,1);

xf=fft(x);
N=length(x);
freq_vector=[0: 1/N :1-1/N].*(fs/1000);
plot(freq_vector,abs(xf))
title 'FFT Input sinal'
xlabel ('Frequency') ;
ylabel ('Amplitude');
hold on

subplot(3,1,2);
df=fft(d);
Nd=length(d);
freq_vector2=[0:1/Nd:1-1/Nd].*fs/1000;
plot(freq_vector,abs(df))

```



```

title 'FFT Disired signal'
xlabel ('Frequency') ;
ylabel ('Amplitude');
hold on

subplot(3,1,3);
yf=fft(y);
Ny=length(y);
freq_vector3=[0:1/Ny:1-1/Ny].*fs/1000;
plot(freq_vector,abs(yf))
title 'FFT Output signal'
xlabel ('Frequency') ;
ylabel ('Amplitude');
hold on
%%
קריאה לפונקציה שכתבנו ושמירת קובץ השמע
[y,e,h]=LmsAlgorithm(x,d);
B=y';
A=e';
sound(B,fs)
clear sound
filename = 'OutputNoEco1.m4a';
audiowrite(filename,A,Fs);

אשר מקבלת את lms כתבנו פוקנציה המתבססת על אלגוריתם
האות הרצוי והאות המצוי
ומחזירה את מקדמי המנסן השגיאה והסיגנל המתקבל במוצא
המסנן.

function [y,e,h]=LmsAlgorithm(x,d)%The name of the
function
N=250;%Filter order
h=zeros(1,N);%Initial zeros vector for filter
coefficients
mu=0.0117;
for k=1:length(d)
    y(k)=h(k)'.*x(k);
    e(k) = d(k)-y(k);%חישוב השגיאה
    h(k+1)= h(k)+ mu*(e(k).*x(k));%עידכון מקדמי המסנן
end

return

```

## חלק שני:

```
clc
x=audioread('sampling.m4a');
[d,fs]=audioread('imTitrzi2.m4a');
x=x(:,1);d=d(:,1);
[y,w,delta] = RLSFilterI232(x,d);
y=y';
plot([d,y])
title('System Identification of an FIR filter')
legend('Desired','Output')
xlabel('Time index')
ylabel('Signal value')

%%sound(x,fs)
%clear sound

%%filename = 'OutputNoEcoNoDiv.m4a';
%%audiowrite(filename,e,fs);
function [y,w,delta] = RLSFilterI232(x,d)
p      = 30;                % filter order
lambda = 0.999;             % forgetting factor
laminv  = 1/lambda;
delta   = 1.0;              % initialization
parameter
% Filter Initialization
w       = zeros(p,1);       % filter coefficients
P       = delta*eye(p);     % inverse correlation
matrix
for m = p:length(d)
    j = x(m:-1:m-p+1); % Acquire chunk of data
    e(m) = d(m)-w'*j;% Error signal equation
    Pi = P*j;% Parameters for efficiency
    k = (Pi)/(lambda+j'*Pi); % Klman Filter gain
vector update
    P = (P - k*j'*P)*laminv;% Inverse correlation
matrix update
    w = w + k*e(m);% Filter coefficients adaption
end
y=e;
return
```

```

%%
% בחלק זה בחרנו להראות את תגובת התדר של המסנן שנוצר
% באמצעות האלגוריתם.
h_overall = [ 1 w']
[H,j] = freqz(h_overall,1);% N+1 יצירת ווקטור של
% הוא סדר המסנן N איברים כאשר
zplane(h_overall,1)
V=10*log10((abs(H)));% db הצגה של תגובת התדר ב

subplot (2,1,1);
plot(j,V)
title 'Normlize frequency response'
xlabel ('Frequency[Hz]')
ylabel ('Gain[db]')
grid on
hold on

subplot (2,1,2);
zplane(h_overall,1)
title 'Zplane'
grid on

fvtool(h_overall)

%%

subplot (3,1,1);
k=xcorr(x,x);
L=length(x);
plot(-(L-1):(L-1),k)
title 'Autocorrelation-Signal With Eco and Shift
Phase'
xlabel 'Time[sec]'
ylabel 'Amplitude'
hold on

subplot (3,1,2);
he=xcorr(e,e);
L=length(e);
plot(-(L-1):(L-1),he)
title 'Autocorrelation-Signal Output'
xlabel 'Time[sec]'
ylabel 'Amplitude'
hold on

```

```

subplot (3,1,3);
hd=xcorr(d,d);
L=length(d);
plot(-(L-1):(L-1),hd)
title 'Autocorrelation-Signal Output Desired'
xlabel 'Time[sec]'
ylabel 'Amplitude'
hold on

%%
subplot(3,1,1);

xf=abs(fft(x));
N=length(x);
freq_vector=[0: 1/N :1-1/N].*(fs/1000);
plot(freq_vector,xf)
title 'FFT Input sinal'
xlabel ('Frequency') ;
ylabel ('Amplitude');
hold on

subplot(3,1,2);
df=fft(d);
Nd=length(d);
freq_vector2=[0:1/Nd:1-1/Nd].*fs/1000;
plot(freq_vector,abs(df))
title 'FFT Disired signal'
xlabel ('Frequency') ;
ylabel ('Amplitude');
hold on

subplot(3,1,3);
yf=fft(e);
Ny=length(e);
freq_vector3=[0:1/Ny:1-1/Ny].*fs/100;
plot(freq_vector,abs(yf))
title 'FFT Output signal'
xlabel ('Frequency') ;
ylabel ('Amplitude');
hold on

%%
Input_Coral=xcorr(x,x);%Autocorrelation function
L=length(x);
%plot(-(L-1):(L-1),k)
power_Input=max(Input_Coral)%max value of
Autocorrelation function(this is the power of the
input signal)

```

```
Output_Corla=xcorr(y,y);%Autocorrelation function  
power_Output=max(Output_Corla)
```