2023

# DSP 1


**אביתר כהן – 205913858**

**אייל ארד - 212552442**

## 1. מבוא – מימוש FFT

בחלק זה התבקשו לממש פונקציית FFT ופונקציית IFFT ע"י שימוש בפונקציה הראשונה.

המימוש בוצע ע"י דצימציה בזמן ( d=0.41.. )

נממש FFT ו IFFT כפי שלמדנו בהרצאה, בעזרת ה "פרפרים":

```
% -----------------------------------------------------
%Q1

function X = my_fft(x)
    N = numel(x);

    if N == 1
        X = x;
    else
        x_even = my_fft(x(1:2:end));
        x_odd = my_fft(x(2:2:end));

        factor = exp(-2i*pi*(0:N/2-1)/N);
        X = [x_even + factor.*x_odd, x_even - factor.*x_odd];
    end
end

function x = my_ifft(X)
    N = numel(X);

    if N == 1
        x = X;
    else
        X_conj = conj(X);
        x_conj = my_fft(X_conj);
        x = conj(x_conj)/N;
    end
end

%-------------------------------------------------------------------
```

חלוקה לדגימות זוגיות ואי זוגיות

w

איחוד ענפים

צמוד

הסבר FFT : הפונקציה my_fft() הינה פונקציה רקורסיבית לחישוב ה FFT, אשר מקבלת אות \סיגנל בכניסה, ובכל איטרציה מחלקת את האות לדגימות זוגיות ואי זוגיות עד אשר הגענו לדגימה בודדת שזה הוא תנאי העצירה.

לאחר מכן מתבצע שלב האיחוד בין הענפים, כאשר הפונקציה מבצעת הכפלה בין הענף האי זוגי ל W - שבקוד מצוי בשם "factor" ובמבצעת חיבור או חיסור בין התוצאה לענף הזוגי.

הסבר IFFT : הפונקציה my_ifft() הינה פונקציה רקורסיבית שמשתמשת ב FFT שיצרנו.

בכל איטרציה, ניקח את הסיגנל, נבצע לו צימוד, נשלח ל FFT , מבצע צימוד ונחלק במספר הדגימות, וזה לפי הנוסחה הבאה:

$$IFFT(X[K]) = \frac{1}{N}\left(FFT(X^*[K])\right)^*$$

נריץ את סיגנל בדיקה על הפונקציות שבנינו

```
% Example usage
x = [1, 4, 2, 9];
X = my_fft(x);
reconstructed_x = my_ifft(X);

disp('Original signal:');
disp(x);
disp('FFT:');
disp(X);
disp('Reconstructed signal:');
disp(reconstructed_x);
```

Command Window

```
>> part1
Original signal:
     1     4     2     9

FFT:
  16.0000 + 0.0000i  -1.0000 + 5.0000i -10.0000 + 0.0000i  -1.0000 - 5.0000i

Reconstructed signal:
   1.0000 + 0.0000i   4.0000 + 0.0000i   2.0000 + 0.0000i   9.0000 - 0.0000i
```

ונריץ את סיגנל הבדיקה עם על פונקציית ה FFT המובנת כדי לוודא שיש שוויון בתוצאות

```
x=[1 4 2 9]
fft(x);
disp(fft(x))
```

Command Window

```
>> part2_4

x =

     1     4     2     9

  16.0000 + 0.0000i  -1.0000 + 5.0000i -10.0000 + 0.0000i  -1.0000 - 5.0000i
```

נוכל לראות כי מתקיים שוויון ואכן ניתן לומר כי הפונקציות שבנינו ממשות באופן תקין את אלגוריתמי ה FFT ו IFFT

## 2. חלק ראשון – ניתוח ועיבוד תמונה ע"י שימוש ב DTFT דו מימדי

**א)** נוכיח כי ניתן לחשב DTFT ע"י הפעלת DTFT חד מימדי על כל עמודה, ושוב פעם DTFT על כל שורה.

$$X(e^{j\omega_1}, e^{j\omega_2}) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} x[n,m]\, e^{-j(\omega_1 n+\omega_2 m)}$$

$$= \sum_{n=-\infty}^{\infty}\left(\sum_{m=-\infty}^{\infty} x[n,m]\, e^{-j\omega_2 m}\right) e^{-j\omega_1 m}$$

$$= \sum_{n=-\infty}^{\infty} X(e^{j\omega_2})\, e^{-j\omega_1 n} = \boxed{DTFT\{X(e^{j\omega_2})\}\big|_{e^{j\omega_1}}}$$

**ב)** נוכיח כי הגדרת DFT בשני מימדים מקבלת מדגומות של DTFT בשני המימדים, ומבנו את הקבועים בדרך ובשלב הגדרה

$$X[k_1,k_2] = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x[n,m]\, e^{-j\left(\frac{2\pi n}{N}k_1 + \frac{2\pi m}{M}k_2\right)}$$

$$\boxed{= X(e^{j\omega_1}, e^{j\omega_2})\Big|_{\omega_1=\frac{2\pi n}{N}k_1,\ \omega_2=\frac{2\pi m}{M}k_2}}$$

**ג)**

$$X(e^{j\omega_1}, e^{j\omega_2}) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} y[n]\, z[m]\, e^{-j(\omega_1 n+\omega_2 m)}$$

$$= \sum_{n=-\infty}^{\infty} y[n]\, e^{-j\omega_1 n} \sum_{m=-\infty}^{\infty} z[m]\, e^{-j\omega_2 m} = \boxed{Y(e^{j\omega_1})\, Z(e^{j\omega_2})}$$

הדגימה (ב) פתרון 2

$$X[k_1,k_2] = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x[n,m]\, e^{-j\left(n\frac{2\pi}{N}k_1 + m\frac{2\pi}{M}k_2\right)}$$

$$= \sum_{n=0}^{B_1-1}\sum_{m=0}^{B_2-2} 1\cdot e^{-j\left(n\frac{2\pi}{N}k_1 + \frac{2\pi k_2 m}{M}\right)}$$

$$= \sum_{n=0}^{B_1-1} e^{-j\frac{2\pi n k_1}{N}} \sum_{m=0}^{B_2-1} e^{-j\frac{2\pi k_2 m}{M}}$$

$$= \sum_{n=0}^{B_1-1} e^{-j\frac{2\pi k_1 n}{N}} \cdot \frac{1-e^{-j\left(\frac{2\pi B_2 k_2}{M}\right)}}{1-e^{-j\left(\frac{2\pi k_2}{M}\right)}}$$

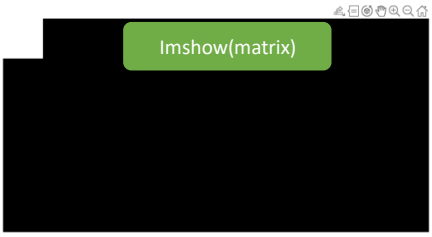
$$= e^{-j\left[\left(\frac{2\pi k_1}{N}\right)(B_1-1) + \left(\frac{2\pi k_2}{M}\right)(B_2-1)\right]} \cdot \frac{\sin\left(\frac{2\pi k_1}{N}\cdot\frac{B_1}{2}\right)}{\sin\left(\frac{\frac{2\pi}{N}k_1}{2}\right)} \cdot \frac{\sin\left(\frac{2\pi k_2}{M}\cdot\frac{B_2}{2}\right)}{\sin\left(\frac{\frac{2\pi}{M}k_2}{2}\right)}$$

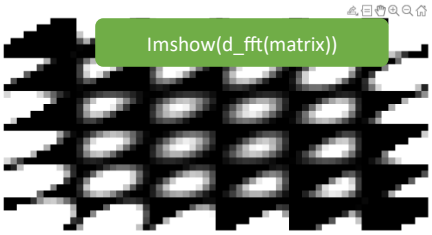
(נכון)

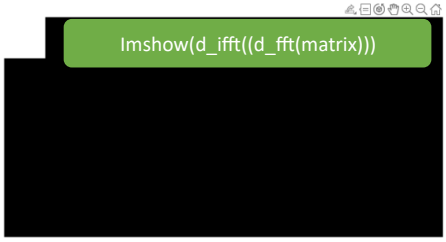
(נכון)

ד) חישוב נומרי:

**פלט**




**קוד**

```
% Q2-d)

n = 32;   % Number of rows
m = 64;   % Number of columns

% Create a zero matrix
matrix = zeros(n, m);

% Set elements to 1 within the specified range
matrix(1:6, 1:6) = 1;

%-------------------------------------------------------
imshow(matrix)
imshow(d_fft(matrix))
imshow(d_ifft(d_fft(matrix)))
%-------------------------------------------------------

function X = d_fft(x)
n = 32;   % Number of rows
m = 64;   % Number of columns

    % Perform FFT on COLUMNS
    fft_columns = zeros(n,m);

    for col = 1:m
        fft_columns(:, col) = my_fft(x(:, col));
    end

    % Perform FFT on rows
    X = zeros(n, m);
    for row = 1:n
        X(row, :) = my_fft(fft_columns(row, :));
    end

end

function x = d_ifft(X)
n = 32;   % Number of rows
m = 64;   % Number of columns

    % Perform IFFT on COLUMNS
    ifft_columns = zeros(n, m);
    for col = 1:m
        ifft_columns(:, col) = my_ifft(X(:,col));
    end

    % Perform IFFT on rows
    x = zeros(n, m);
    for row = 1:n
        x(row, :) = my_ifft(ifft_columns(row, :));
    end
end
%-------------------------------------------------------
function X = my_fft(x) ...

function x = my_ifft(X) ...

%-------------------------------------------------------
```

**הסבר לקוד:** על מנת לחשב את הDFT הדו ממדי של התמונה יצרנו את הפונקציה "d_fft()" שמקבלת מטריצה ומחשבת את ה DFT של העמודות ולאחר מכן את ה DFT על השורות של מה שיצא. נעזרנו בפונקציה שבנינו בשאלה 1.

הפונקציה "d_ifft()" מבצעת את הפעולה ההפוכה.

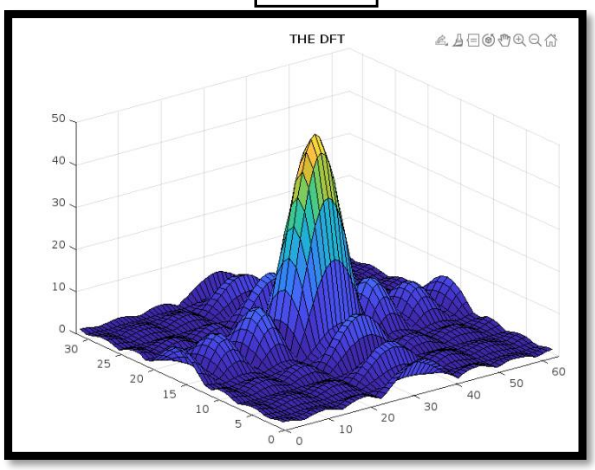
ניתן לראות כי אכן מתקבלת המטריצה המקורית לאחר שאנו מבצעים DFT על המטריצה ואז IDFT.

כעת נצייר את ה DFT הדו ממדי של התמונה.

**קוד**

```
%Q2 THE 3D

a=ones(7,7);
b=zeros(7,57);
c=zeros(25,64);
x=[a b;c];
x=fft(x);
ft_x=fft(x.');
ft_x=ft_x.';
ft_x=abs(ft_x);
ft_x=fftshift(ft_x);
surf(real(ft_x));
title('THE DFT')
```

**פלט**



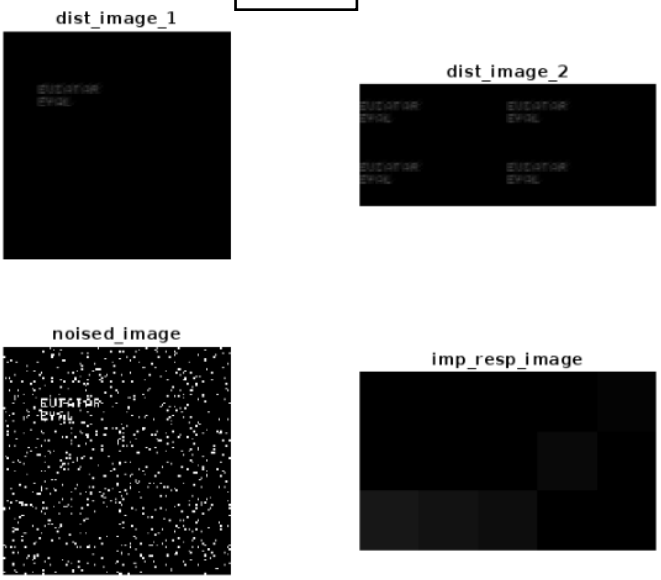

ה) נפעיל את img_gen.p ונחשב את $h_0 = h[n,0]$ התגובה להלם של הערוץ:

**קוד**

```
%PART 2.5
[dist_image_1,dist_image_2,noised_image,imp_resp_image]=img_gen('EVIATAR','EYAL')


% Plot dist_image_1
subplot(2, 2, 1);
imshow(dist_image_1);
title('dist\_image\_1');

% Plot dist_image_2
subplot(2, 2, 2);
imshow(dist_image_2);
title('dist\_image\_2');

% Plot noised_image
subplot(2, 2, 3);
imshow(noised_image);
title('noised\_image');

% Plot imp_resp_image
subplot(2, 2, 4);
imshow(imp_resp_image);
title('imp\_resp\_image');

% compute THE DTFT of H0
n=2;
h0=[ 0;0;0.0909];
w=[0 2*pi/6 4*pi/6 8*pi/6];
H0=exp(-1i*w'*n)*h0';
disp(H0)
```

חישוב התגובה להלם בתדרים השונים

**פלט**



התגובה להלם הינה : $h_0$

```
imp_resp_image =

         0         0         0         0    0.0182
         0         0         0    0.0364         0
    0.0909    0.0727    0.0545         0         0
```

התגובה להלם בתדרים השונים: $H[e^{jw}]$

$$\sum_{m=-\infty}^{\infty} h_0(n) \cdot e^{-j[0,\frac{2\pi}{6},2\frac{2\pi}{6},4\frac{2\pi}{6}]m}$$

**קוד**

```
% compute THE DTFT of H0
n=2;
h0=[ 0;0;0.0909];
w=[0 2*pi/6 4*pi/6 8*pi/6];
H0=exp(-1i*w'*n)*h0';
disp(H0)
```

**פלט**

```
0.0000 + 0.0000i   0.0000 + 0.0000i   0.0909 + 0.0000i
0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0454 - 0.0787i
0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0455 + 0.0787i
0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0454 - 0.0787i
```

<u>ו) נחשב את הקונבולוציה הציקלית במחזור 32 של $h_0[n]$ עם האות:</u>

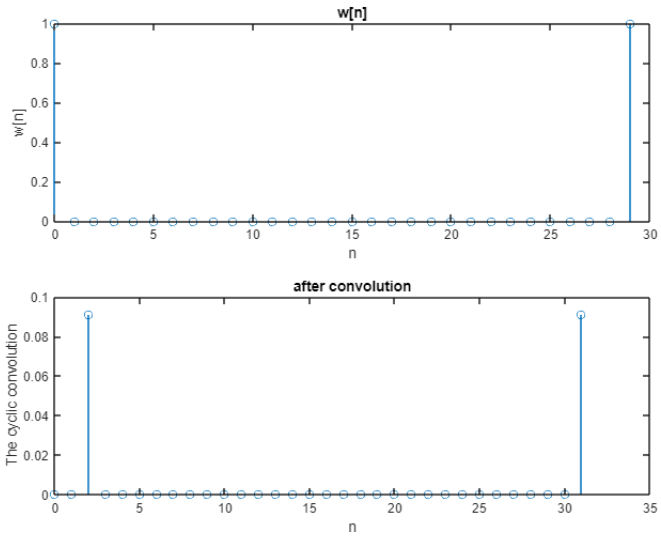
$$w[n] = \delta[n] + \delta[n-29]$$

$$\sum_{m=0}^{31} h_0(n) \cdot w[< n-m >_{32}]$$

**קוד**

```
%Q2.6
subplot(2,1,1)
w = [1 zeros(1, 28) 1]; %  creat w[n] from 2 delta
stem(0:length(w)-1, w);
title('w[n]');
xlabel('n');
ylabel('w[n]');

% cyclic convolution with h0
cyclic_conv = cconv(h0, w, 32);

% result of convolution
subplot(2,1,2);
stem(0:length(cyclic_conv)-1, cyclic_conv);
title('after convolution');
xlabel('n');
ylabel('The cyclic convolution');
```

**פלט**




תוצאת הקונבולוציה היא 2 דלתאות כיוון ש h היא דלתא בודדת ו W מורכבת מ 2 דלתא

ז) נחשב ונצייר את התמונות המקוריות $x_1[m,n], x_2[m,n]$.

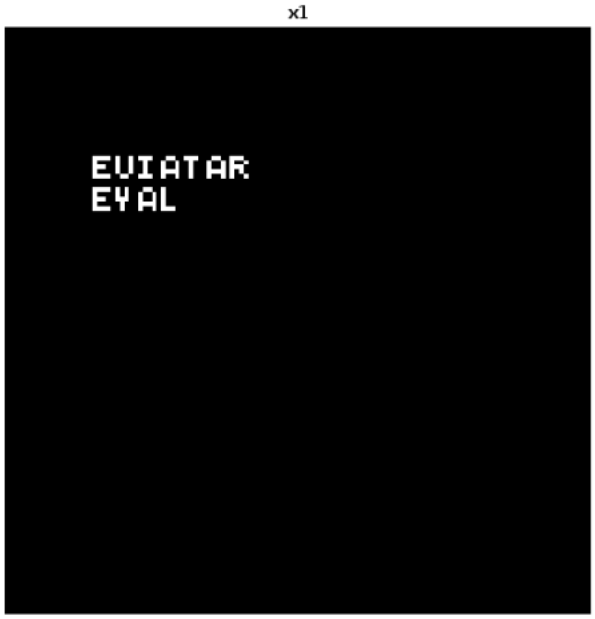
$$Y[K] = X[K] * H[K]$$

$$x[n] = IDFT(\frac{Y[K]}{H[K]})$$

יצרנו פונקציה לחישוב FFT דו מימדי בשם fft_2d כיוון שהאות הוא דו מימדי.

פונקציה זו מבצעת FFT על כל עמודה במטריצה ואז עושה FFT שנית על כל שורה במטריצה

**פלט**


**קוד**

```
%x1:
padded_h1 = [imp_resp_image zeros(3,123);zeros(125,128)];
Y1 = fft_2d(dist_image_1);
H1 = fft_2d(padded_h1);
X1 = Y1./H1;
x1 = abs(Ifft_2d(X1));
figure
imshow(x1);
hold on;
title('x1');

%x2:
h2_padded = [imp_resp_image zeros(3,165);zeros(67,170)];
Y2 = fft_2d(dist_image_2);
H2 = fft_2d(h2_padded);
X2 = Y2./H2;
x2 = abs(Ifft_2d(X2));

figure
imshow(fftshift(x2));
hold on;
title('x2');


function [Xk] = fft_2d(xn)
xn=fft(xn);
Xk=fft(xn.');
Xk=Xk.';
end

function [xn] = Ifft_2d(Xk)
xn=ifft(Xk);
xn=ifft(xn.');
xn=xn.';
end
```

ח) ההבדל בתוצאות השחזור:

נשים לב שגודלן של התמונות זהה לקודמות ושהצבע הלבן בתמונות המשוחזרות בולט יותר וחד יותר, וזה עקב סינון הרעש.

א) הוספת אפסים = דגימה בתדירות גבוהה יותר ומכך שחזור X1 איכותי וברור יותר מהתמונה המקורית. נשים לב שהתמונה X1 רופדה ב 20 אפסים מכל כיוון, ולכן כך גם התמונה המשוחזרת , ניתן לראות לפי המיקומים על התמונה. שכן הוספת אפסים אינה משנה את ה DFT של הסדרה .

ב) שכפול בזמן = דגימה בתדר. נשים לב בהתאם לציפיות התקבלו 4 תמונות משוכפלות בצירים. נוסף לנו 0 בין כל 2 דגימות כיוון שישנה הרחבה של האות בתדר = כיווץ בזמן.

## 3. חלק שני – ייצור וניתוח אותות דיבור

א) בחלק זה הקלטנו אות דיבור בקצב דגימה של $Fs = 16000_{Hz}$ ולאחר מכן חתכנו אותו לכדי $2^{16}$ דגימות כנדרש – אות זה יסומן כ $x\_n$

Recording a speech signal

```
%Q3.a.1
my_record = audiorecorder(16000,8,1,-1); %(Fs,BitNum,Channel,Id)
disp("Record begin")
recordblocking(my_record,5);  % 4 seconds record

disp("Record end")
play(my_record);
```

Taking 2^16 samples

```
sampled_record=getaudiodata(my_record); % extract the audio data from the signal (vector)
filename = 'my_record.wav';
audiowrite(filename,sampled_record,16000);
[sampled_record]=audioread('my_record.wav');
sampled_record= sampled_record.'; %   converting from a column vector to a row vector.
sampled_record=sampled_record(1:2^16); % save only 65536 samples
```

סה"כ נקבל $2^{16} = 65536$ דגימות

| sampled_record | 1×65536 d... | 1×65536 |
|---|---|---|

נחשב את ההספק הממוצע של האות ע"י הנוסחה :

Calculate the average power

```
%Q3.a.2
N=2^16;
P=(1/N)*(sampled_record)*(sampled_record.');
```

$$P_X = \frac{1}{N}\sum_{n=0}^{N-1}(x[n])^2$$

קיבלנו כי ההספק של האות הוא:

| P | 0.0104 |
|---|---|

ב) נגדיר אות רעש $z\_n$ באופן הבא:

$$z[n] = 50\sqrt{P_x}[cos(\omega_1 n) + cos(\omega_2 n) + cos(\omega_3 n)], \quad n = 0, \ldots, N-1$$

$$a_1 = 50\sqrt{P_X}, \omega_1 = 1.6 + 0.1 \cdot d_1$$
$$a_2 = 50\sqrt{P_X}, \omega_2 = 1.6 + 0.1 \cdot d$$
$$a_3 = 50\sqrt{P_X}, \omega_3 = 3$$

כאשר

$$d_1 = 0.212552442, \qquad d_2 = 0.205913858$$

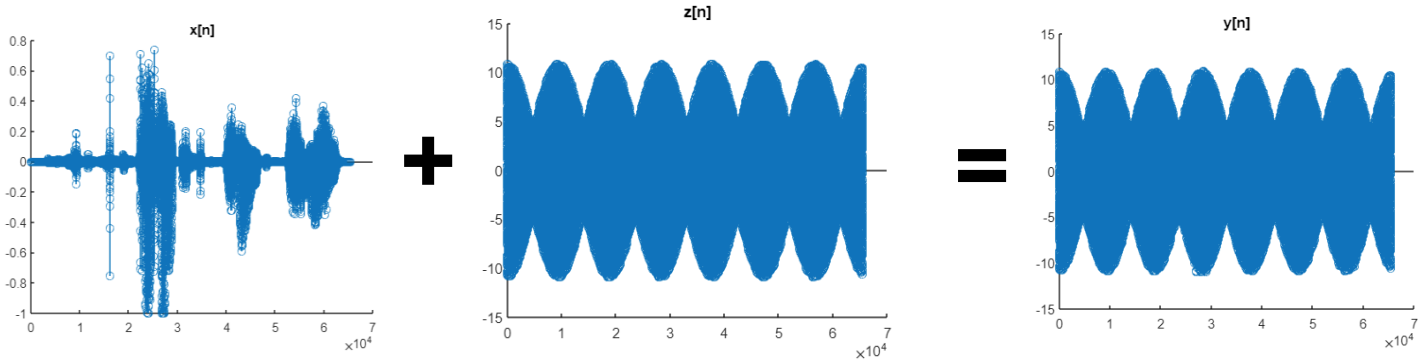
נוסיף את ההפרעה לאות שדגמנו קודם, לתוצאה נקרא y_n כך שמתקיים :

Adding noise to samples

```
%Q3.b -
d2=0.205913858;
d1=0.212552442;
n=(0:1:N-1);
w1=1.6+0.1*d1;
w2=1.6+0.1*d2;
w3=3;
%----------------- BY DEFENITION----------------
x_n=sampled_record;
z_n=50*sqrt(P)*(cos(w1*n)+cos(w2*n)+cos(w3*n));
y_n=z_n+x_n;
soundsc(y_n,16000);
```

לאחר הפעלת האות y_n כמעט ולא שמענו את ההקלטה שקודם לכן הקלטנו, וזאת בעקבות רעש שהתווסף לאות המקורי

ג) נציג את האותות שקיבלנו:

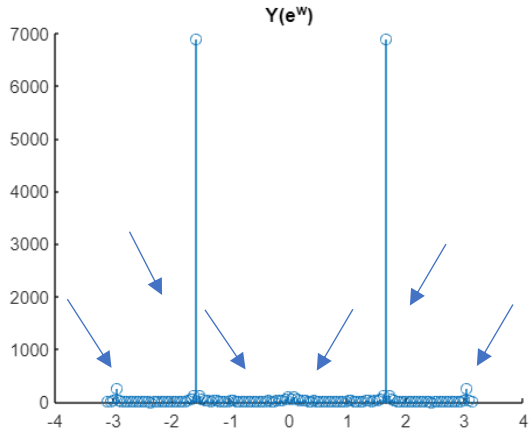



=

ד) נחשב את התמרת פורייה $Y(e^{jw})$ בתדרים הבאים

$$\omega = \ldots, -3\cdot\frac{2\pi}{128}, -2\cdot\frac{2\pi}{128}, -1\cdot\frac{2\pi}{128}, 0, 1\cdot\frac{2\pi}{128}, 2\cdot\frac{2\pi}{128}, 3\cdot\frac{2\pi}{128}, \ldots$$

**פלט**




**קוד**

Ploting Y(exp(jw)):

```
%Q3.d: Y_e_omega
k=(-63:64);
omega=2*pi*k/128;
Y_e_omega=fft(y_n);
Y_e_omega=fftshift(Y_e_omega);

figure;
hold on
title('Y(e^j^w)');
stem(omega,abs(Y_e_omega(1:512:N)));
```

נשים לב שקיימות כאן 6 דלתאות כיוון שאות הרעש מורכב מ 3 קוסינוסים.

הדלתאות האמצעיות הן בעוצמה גבוהה הרבה יותר מהדלתאות בצדדים. (7000) לעומת 300.

ה) נבצע דצימציה ביחס של 2 של האותות y_n, z_n

נרשום ביטוי מתמטי לאות $z_2[n]$ ולהתמרת פורייה שלו $Z_2(e^{jw})$

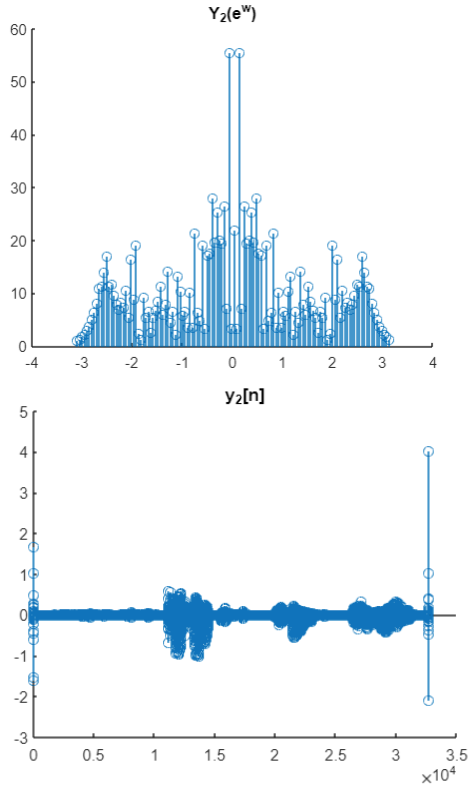
$$z_2[n] = z[2n]\ for\ 1 \le n \le \frac{N}{2}$$

עבור ההתמרה נקבל:

$$Z_2(e^{jw}) = \frac{1}{2}\sum_{k=0}^{1} Z(e^{\frac{j(w-2\pi k)}{2}}) = \frac{1}{2}(Z\left(e^{\frac{j(w-2\pi)}{2}}\right) + Z\left(e^{\frac{jw}{2}}\right))$$

ו) נצייר את $y_2[n]$ ואת ההתמרת פורייה שלו:

**הפלט**




**קוד**

```
y2_n=decimate(y_n,2);
Y2_e_omega=fft(y2_n);
Y2_e_omega=fftshift(Y2_e_omega);
figure
hold on
title('y_2[n]');
stem(n(1:N/2),y2_n);

figure
hold on
title('Y_2(e^j^w)');
stem(omega,abs(Y2_e_omega(1:256:N/2)));
soundsc(y2_n,8000);
```

ביצענו דצימציה ביחס 2 ע"י הפקודה decimate הרעש נעלם סונן והאות חזר להיות ברור.

זאת מכיוון שפקודת decimate מבצעת גם LPF בנוסף לדגימה בחדשה שהיא מבצעת על האות שהיא מקבלת על מנת למנוע aliasing.

נשים לב שכדי שנוכל לשמוע את האות באופן תקין נצטרך לשנות את פרמטר הדגימה בפקודה soundc להיות קטן פי 2 ממה שהיה קודם (8000 במקום 16000), וזאת בעקבות הדצימציה פי 2 .