דו"ח תרגיל MATLAB מס' 1

עיבוד דיגיטלי של אותות 1 – 83320-01

סמסטר אביב תשפ"ג

מגישים :

איתי אהרון גולדברג

מתן פויכטונגר

הדו"ח כולל פתרונות עבור ארבעת השאלות, גרפים מתאימים וכן את קטעי הקוד שכתבנו.

מבוא – מימוש FFT ו-IFFT

כתלות בערך הפרמטר d נתבקשנו לממש FFT ע"י דצימציה בזמן וכן לממש IFFT ע"י שימוש בפונקציה הראשונה. תחילה נפתח ביטוי מתמטי מתאים מתוך הגדרת ההתמרה:

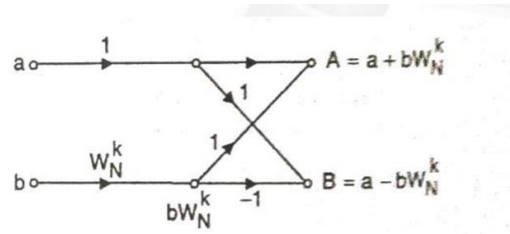
$$X[k]=\sum_{n=0}^{N-1}x[k]W_N^{kn}=$$

$$=\sum_{l=0}^{\frac{N}{2}-1}x[2l]W_N^{2lk}+\sum_{l=0}^{\frac{N}{2}-1}x[2l+1]W_N^{(2l+1)k}=$$

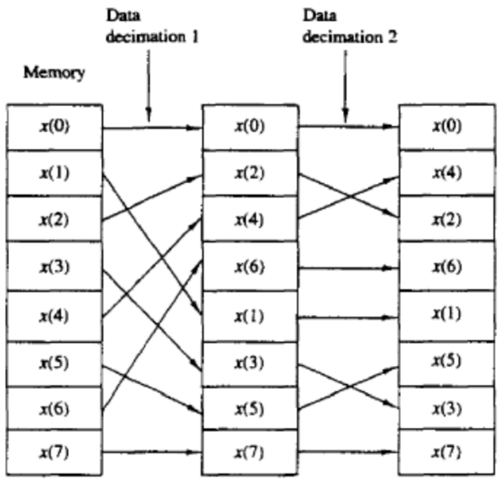
$$=\sum_{l=0}^{\frac{N}{2}-1}x[2l]e^{-j\frac{2\pi 2lk}{N}}+\sum_{l=0}^{\frac{N}{2}-1}x[2l+1]\,e^{-j\frac{2\pi(2l+1)k}{N}}$$

קיבלנו שתי סכימות, אחת עבור הערכים הזוגיים והשנייה עבור האי-זוגיים. כאשר נמשיך לפצל את הסכימות שלנו נקבל חזקה של 2 של ערכי האות המקורי מוכפלים כל אחד במקדם יחיד המשתנה בין כל דגימה ודגימה המבטא את הסיבוב של האות במעגל היחידה.

מצאנו בספרים שרטוט, התואם את הנלמד בכיתה, המתאר את הפעולה על שני ערכים בודדים:



עבור אות שאורכו מעל 2 נקבל מספר רמות המבצעות את אותו החישוב כמתואר בתמונה הבאה:



להלן הקוד המבצע FFT כמבוקש:

```
function X = fft_dft(x)
    N = length(x);

    % Base case of the recursion
    if N == 1
        X = x;
    else
        % Split the input sequence and make a recursive calls
```

```
        x_even = x(1:2:end);
        x_odd = x(2:2:end);

        X_even = fft_dft(x_even);
        X_odd = fft_dft(x_odd);

        % Combine the results of the recursion
        W = exp(-1i * 2 * pi / N * (0:N/2-1));
        X = [X_even + W .* X_odd, X_even - W .* X_odd];
    end
end
```

בכל קריאה של הפונקציה, האות המתקבל מתחלק לשני חלקים – דגימות זוגיות ואי זוגיות ביחס לאות שהתקבל בקריאה זו. כל חצי אות עובר התמרה ע"י קריאה רקורסיבית לפונקציה עצמה. שני האותות המותמרים מצטרפים להתמרה אחת ע"י חיבור ההתמרה הזוגית עם ההתמרה האי זוגית כאשר האחרונה מוכפלת במקדם W. בהתאמה אנו משרשים את החצי השני של ההתמרה כמו שראינו בשרטוט עבור שני ערכים יחידים קודם לכן. הוספנו תנאי עצירה עבור הדגימה הבודדת שתחזיר את הערך עצמו מאחר שהיחידה הבסיסית של ההתמרה היא בין שתי דגימות.

עבור IFFT מתקיים:

$$X[n] \xrightarrow{FFT} Nx[< -k >_N]$$

וע"פ תכונת הדואליות נקבל:

$$x^*[n] \xrightarrow{FFT} X^*[< -k >_N]$$

ולכן סה"כ מתקיים:

$$X^*[n] \xrightarrow{FFT} Nx^*[k]$$

ולכן המימוש בקוד יהיה:

$$x[n] = \frac{1}{N} FFT\{X^*[k]\}$$

להלן קטע הקוד:

```
function x = ifft_dft(X)


    X_conj = conj(X);
    x_conj = Fft_dft(X_conj);

    % Take the conjugate and scale the result by 1/N
    x = conj(x_conj) / length(X);
end
```

להלן קטע קוד המדגים את פעולת הפונקציות:

Part_A.mlx

Generate a test signal

```
Fs = 1024;                % Sampling frequency
t = -0.5:1/Fs:0.5-1/Fs;  % Time vector
f = pi;                   % Frequency of the signal
x = sin(2*pi*f*t);        % Test signal (sine wave)

N = length(x);
f_axis = (-N/2:N/2-1) * (Fs/N);
```

Compute the FFT

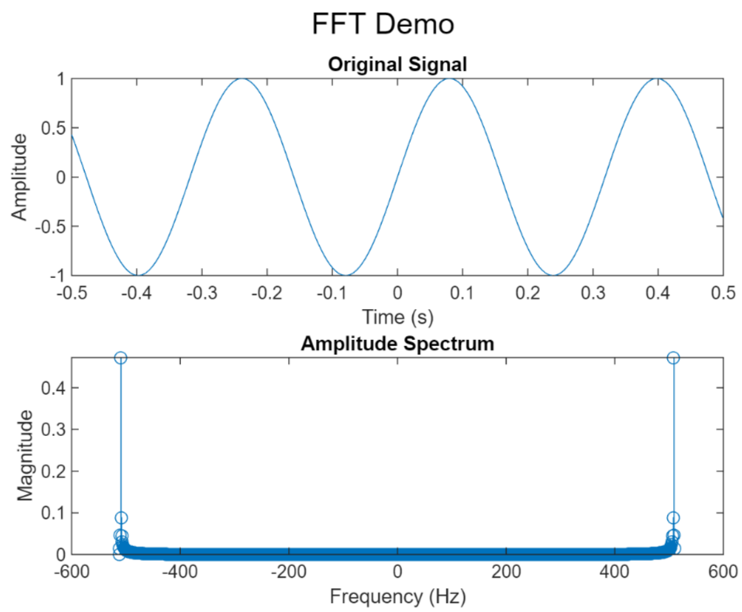
```
X = fft_dft(x);
```

Display the original signal & the single-sided amplitude spectrum

```
subplot(2, 1, 1);
plot(t, x);
title('Original Signal');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2, 1, 2);
stem(f_axis, abs(X/N));
title('Amplitude Spectrum');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

sgtitle('FFT Demo');
```



Perform the inverse FFT

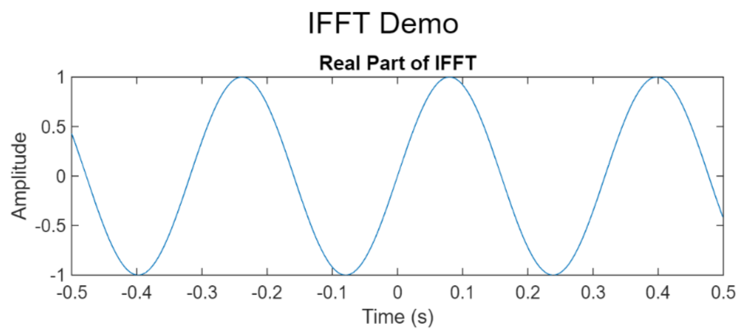
```
x_inv = ifft_dft(X);
```

Plot the real part of the IFFT

```
t_inv = -0.5:1/Fs:0.5-1/Fs;

figure;
subplot(2, 1, 1);
plot(t_inv, real(x_inv));
title('Real Part of IFFT');
xlabel('Time (s)');
ylabel('Amplitude');

sgtitle('IFFT Demo');
```



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

א.

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

$$\sum_n 1*e^{-j\omega n}\sum_m x[n,m]*e^{-j\omega m}=DTFT(DTFT(x[n,m]on\ (m))$$

ב.

$$DFT(x[n,m])=\sum x[n,m]*e^{-j\left(\frac{2\pi kn}{N}+\frac{2\pi lm}{M}\right)}$$

$\omega_1=\frac{2\pi k}{N},\omega_2=\frac{2\pi l}{M}$ כאשר נציב $DTFT(x[n,m])=\sum x[n,m]e^{-j(\omega_1 n+\omega_2 m)}$ <=
ממש"ל. $DFT(x[n,m])=DTFT(x[n,m])$

ג. צ"ל: $X\left(e^{j\omega_1},e^{j\omega_2}\right)=Y\left(e^{j\omega_1}\right)Z\left(e^{j\omega_2}\right)$

ע"פ ההגדרה מתקיים

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

נניח שקיימים $y[n],z[m]$ כך שמתקיים $x[n,m]=y[n]\cdot z[m]$. כלומר, שתיהן מטריצות שוות גדול, באחת של העמודות שוות ובשנייה כל השורות שוות ומכפלתן היא המטריצה המבוקשת. נפתח את הביטוי תוך שימוש בתכונת הלינאריות של הסכום.

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

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

מש"ל

ד. נחשב את $x[n,m]$ אנליטית ע"י בחירת הפונקציות כך:

$$y[n]=\begin{cases}1 & 0\le n<7\\ 0 & 7\le n<32\end{cases},z[m]=\begin{cases}1 & 0\le n<7\\ 0 & 7\le n<64\end{cases}$$

נקבל את ההתמרות:

$$Y[k]=\sum_{0}^{32}y[n]W_{32}^{kn}=\sum_{0}^{6}y[n]W_{32}^{kn}=\sum_{0}^{6}W_{32}^{kn}$$

$$Z[k]=\sum_{0}^{64}z[m]W_{64}^{km}=\sum_{0}^{6}z[n]W_{64}^{km}=\sum_{0}^{6}W_{64}^{km}$$

נציב בנוסחה ונקבל:

$$X[k_1,k_2]=\sum_0^6 W_{32}^{k_1 n}\sum_0^6 W_{64}^{k_2 m}=\frac{1-e^{-7j\omega_1}}{1-e^{-j\omega_1}}\frac{1-e^{-7j\omega_2}}{1-e^{-j\omega_2}}$$

בכדי שניתן יהיה לעמוד את תוצאת ההתמרה נציג את הערך המוחלט המתקבל. להלן קטע הקוד עבור החישוב הנומרי של האות הדו-מימדי וכן מצורפות תמונות של המטריצה והערך המוחלט של התמרתה.

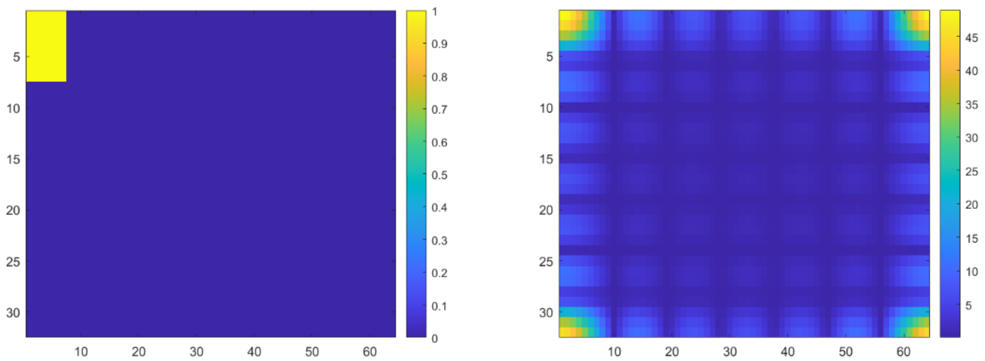
```
N = 32;
M = 64;
b1 = 7;
b2 = 7;

matrix = zeros(N, M);
matrix(1:b1, 1:b2) = 1;
figure
imagesc(matrix);
colorbar()

y = zeros(N,1);
y(1:b1) = 1;
Y = fft(y);

z = zeros(1,M);
z(1:b2) = 1;
Z = fft(z);

X = Y*Z;
figure
imagesc(abs(X));
colorbar()
```



להלן המשך המעקב אחרי הוראו המטלה:

```
[y_1,y_2,y_3,h]=img_gen('MATAN','ITAY');
h_0 = h(:,1);
```

ה.

החישוב הנומרי המבוקש. מאחר שנתבקשנו לייצא ערכים עבור תדרים מסוימים המתחלקים ב-6, ריפדנו את האות כך שנקבל התמרה של 6 נקודות:

```
h_0_zp = [h_0; zeros(size(h_0,1),1)];
H_0 = fft(h_0_zp);
% omega = 0
H_0(1)
```

```
ans = 0.0909
```

```
% omega = 2pi/6
H_0(2)
```

```
ans = -0.0455 - 0.0787i
```

```
% omega = 2*2pi/6
H_0(3)
```

```
ans = -0.0455 + 0.0787i
```

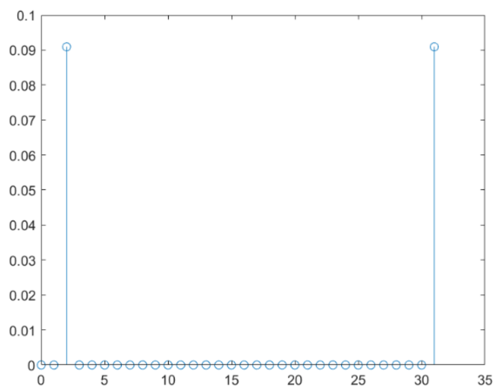
```
% omega = 4*2pi/6
H_0(5)
```

```
ans = -0.0455 - 0.0787i
```

ו.

חישבנו את הקונבלוציה הציקלית ע"י מכפלת התמרות ה-DFT של האותות:

```
w = zeros(32,1);
w([1,30]) = 1;
W = fft(w);
h_0_zp32 = [h_0; zeros(29,1)];
H_0_zp32 = fft(h_0_zp32);
Mult = W.*H_0_zp32;
Conv = ifft(Mult);
figure
stem(0:31,Conv);
```



ז.

עבור חלק זה יצרנו פונקציה המממשת התמרה דו-מימדית באמצעות המשפט אותו הוכחנו בסעיף א':

```
function X = fft_2D(x)
[N, M] = size(x);
X_temp = zeros(N, M);
X = zeros(N, M);
for i=1:N
    X_temp(i,:) = fft(x(i,:));
end

for i=1:M
    X(:,i) = fft(X_temp(:,i));
end
end
```

על בסיס אותו עיקרון ממשנו גם פונקציה עבור התמרה הופכית:

```
function x = ifft_2D(X)
[N, M] = size(X);
x_temp = zeros(N, M);
x = zeros(N, M);

for i=1:N
    x_temp(i,:) = ifft(X(i,:));
end
for i=1:M
    x(:,i) = ifft(x_temp(:,i));
end
end
```

התמרנו את התמונה והתגובה להלם מהערוץ שקיבלנו מהפונקציה בסעיף ד' וחילקנו בכדי לקבל את ההתמרה של התמונה המקורית. לאחר מכן התמרנו אותה התמרה הפוכה והצגנו אותה. להלן שיחזור התמונות המבוקשות:

```
Y_1 = fft_2D(y_1);
padSize = [125, 123];
h_zp = padarray(h, padSize, 0, 'post');
H = fft_2D(h_zp);
X_1 = Y_1./H;
x_1 = ifft_2D(X_1);
figure
imshow(real(x_1));
```

```
Y_2 = fft_2D(y_2);
padSize = [67, 165];
h_zp = padarray(h, padSize, 0, 'post');
H = fft_2D(h_zp);
X_2 = Y_2./H;
```

```
x_2 = ifft_2D(X_2);
figure
imshow(real(x_2));
```

ח.

הנחנו שהתמונות המעוותות נוצרו מקונבלוציה ציקלית בין התמונות המקוריות ותגובת ההלם של הערוץ. כפי שנאמר לנו, הנחה זו שגויה, מה שבאמת קורה הוא קונבלוציה ליניארית של התמונות המקוריות ותגובת ההלם. שתי התמונות שקיבלנו מציגות שתי דרכים לטפל בבעיה זו של קונבלוציה ליניארית:

בשיטה הראשונה, אנו מרפדים את התמונה בלפחות 20 אפסים מכל כיוון, ובהרצאה ובתרגול ראינו שניתן לחשב קונבלוציה ציקלית ע"י קונבלוציה לינארית כאשר מרפדים באפסים. עבור אות באורך N ואות נוסף באורך M נקבל שאם נרפד כל אחד מהאותות כך שיהיה באורך של $N+M-1$ ערכה של הקונבלוציה הציקלית יהי שווה לערכה של הקונבלוציה הלינארית. לכן נרפד את האות באופן כזה שהתנאי יתקיים.

בתמונה השנייה, התמונה המקורית שוכפלה כלומר קיבלנו אות מחזורי וסופי. המשמעות היא שאם שהאות מעתה יתנהג כך:

$$\widetilde{x_2}[n,m] = x_2[<n>_N, <m>_M]$$

סה"כ קיבלנו שמתקיים שהקובלציה הלינארית עם $\widetilde{x_2}$ שקולה לקונבלוציה הציקלית עם x_2.

חלק שני – ייצור וניתוח אותות דיבור

הקלטנו באמצעות פונקציית אודיורקורדר ושמרנו את הקובץ כ'איתי' (הקובץ עצמו הוא אני (מתן) מזייף את השיר 'איתי' של קרן פלס, כלל לא מומלץ). לאחר מכן שמרנו את ווקטור המידע של האות באמצעות פונקציה לשמירת המידע מקובץ השמע ושמרנו את הקובץ.

```
itay=audiorecorder (16000,8,1,-1);
disp('Start speaking.')
```

```
Start speaking.
```

```
recordblocking(itay,5);
disp('End of Recording.');
```

```
End of Recording.
```

```
data_itay= getaudiodata(itay);
```

```
save itay
play (itay);
```

חתכנו את האות לגודל הרצוי וחשבנו את ההספק לפי ממוצע האות בריבוע -

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

```
[data_itay,f] = audioread("itay.wav");
sound(data_itay,f);
```

נחתוך את ההקלטה לאורך של 16^2

```
itay_ef=data_itay(1:2^16); %x(n)
```

חישוב ההספק

```
p_itay=mean(itay_ef.^2)
```

```
p_itay = 3.4493e-04
```

חשבנו את אות ההפרעה ע"פ ההדרכה.

אות ההפרעה

```
n=0:1:2^16-1;%linspace(0,2^16,2^16);%
w1=1.6+0.1*0.316332808;
w2=1.6+0.1*0.025450264;
w3=3;
z_itay=50*sqrt(p_itay)*(cos(n*w1)+cos(n*w2)+cos(n*w3));
```

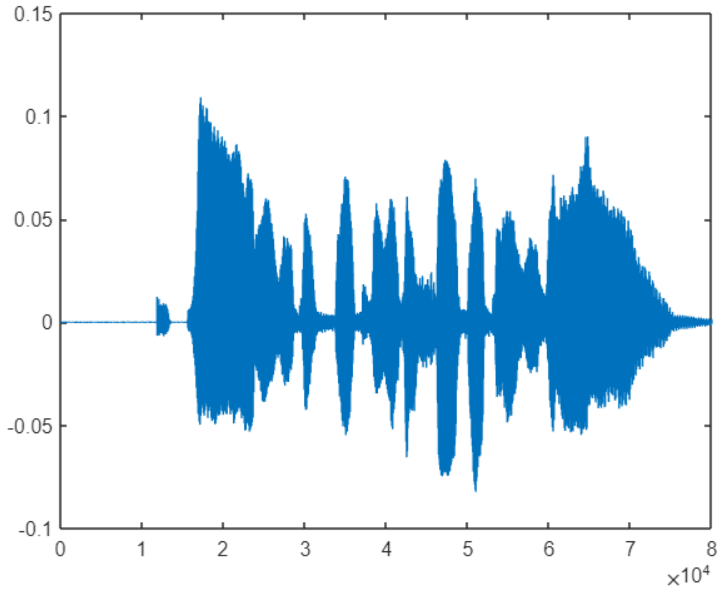
```
p_z_itay=mean(z_itay.^2)
```

```
p_z_itay = 1.2938
```

האות המופרע

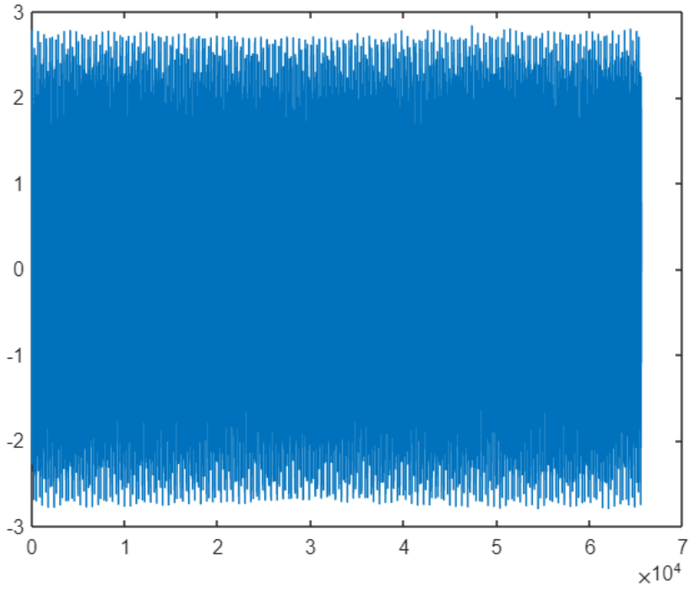
מקודם כפי הנראה בקוד השמענו את ההקלטה וקיבלנו צליל לא יפה כ"כ, אבל מדויק ביחס לקול שלי.

באמצעות הפקודה `>> plot(data_itay)` נציג את מידע האות ללא הרעש המצורף בהמשך



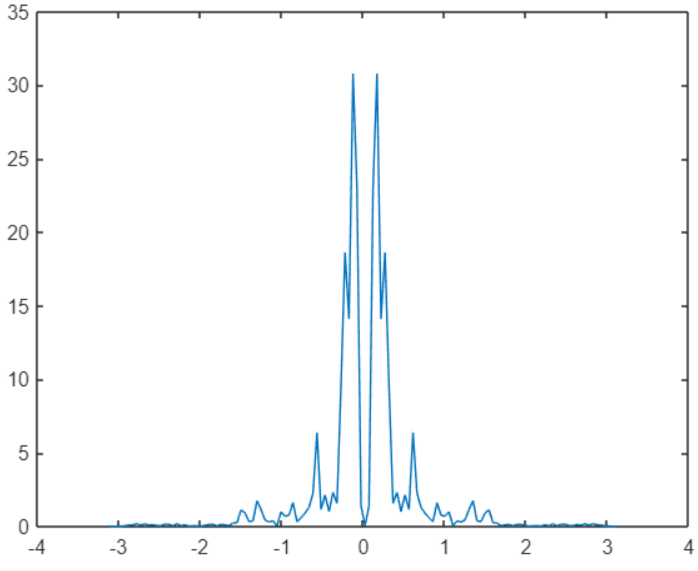
נחבר את הרעש כך שנקבל אות עם הרבה הפרעה, אפשר לראות שהרעש מאוד משמעותי ולא מאפשר כלל לשמוע את ההקלטה.

```
y_itay=itay_ef+z_itay';
plot(y_itay)
```

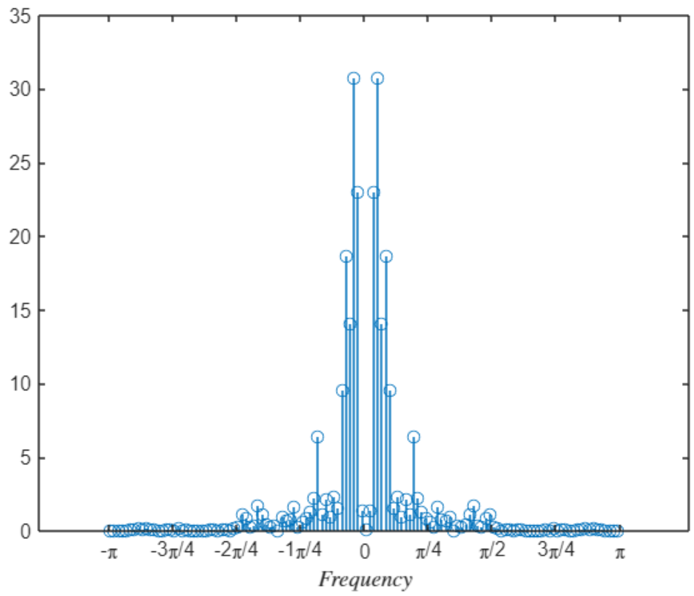


חשבנו את התמרת האות עבור התדרים הנדרשים והצגנו את הגרף שלה (הסבר לגרף ניתן לראות בתחתית התמונה). רק אוסיף מאחר ולא התייחסתי לכך שהאות המרכזי הנמוך ככל הנראה נובע מתדר הדיבור של ההקלטה, ניתן לראות שמאחר וניסיתי לשיר יצאו תדרים שונים גם בדיבור עצמו(עליה וירידה של הטון).

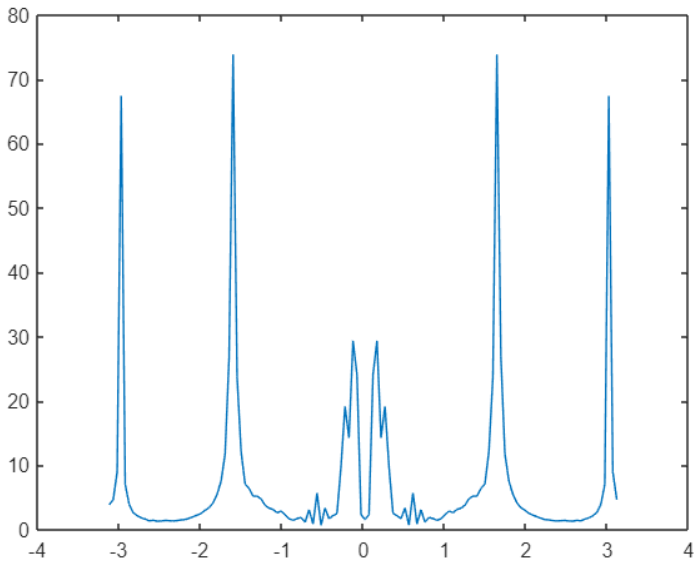
```
ITAY_ef = fft(itay_ef);
M = 128;
Y_plot = ITAY_ef(1:size(ITAY_ef,1)/M:end);
Y_plot = circshift(Y_plot,M/2);
W = 2*pi/M * (-(M-1)/2:(M-1)/2);
figure
plot(W,abs(Y_plot));
```



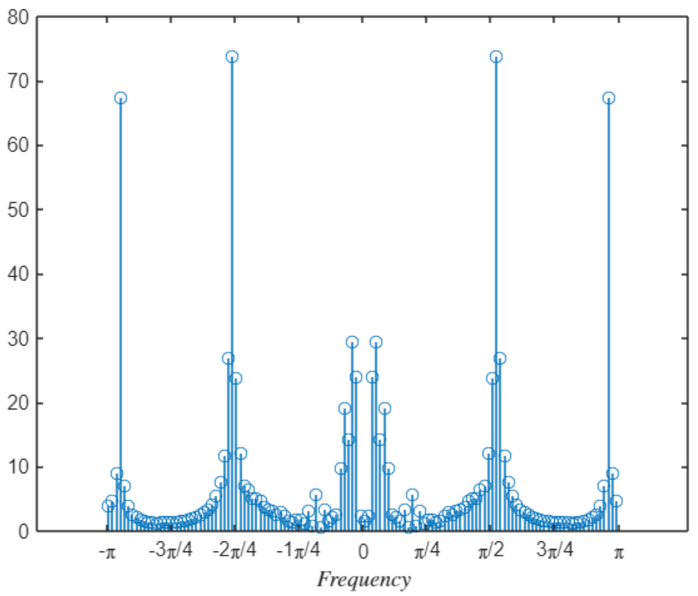
```
figure
stem(W,abs(Y_plot));
xlabel('$$ Frequency $$','Interpreter','latex');
set(gca,'XTick',-pi:pi/4:pi,'XTickLabel',{'-\pi','-3\pi/4','-2\pi/4','-1\pi/4','0','\pi/4','\pi/2','3\pi/4','\pi'})
```




```
Y_ITAY=fft(y_itay); % dsp_fft(y_itay);
M = 128;
X_plot = Y_ITAY(1:size(Y_ITAY,1)/M:end);
X_plot = circshift(X_plot,M/2);
W = 2*pi/M * [-(M-1)/2:(M-1)/2];
figure
plot(W,abs(X_plot))
```




```
figure
stem(W,abs(X_plot))
xlabel('$$ Frequency $$','Interpreter','latex');
set(gca,'XTick',-pi:pi/4:pi,'XTickLabel',{'-\pi','-3\pi/4','-
2\pi/4','-1\pi/4','0','\pi/4','\pi/2','3\pi/4','\pi'})
```




```
%sound(y_itay,16000);
```

אפשר לראות בגרף לעיל שאנומקבלים זוגות של דלתאות מקבילות, הסיבה לכך היא שבהתאם 3למספרי הת.ז שלנו אנו מקבלים ערכי אומגה קרובים מאוד כך שהם מתאחדים בהמרה לזוג דלתאות יחיד ועבור התדר השווה 3 נקבל עוד זוג דלתאות וכן זוג דלתאות סביב האפסה התואמות את אות הדיבור המקורי.

בצענו דצימציה בזמן לשני האותות וחשבנו באופן אנליטי את התמרת האות Z2 ואת ואת האות Z 2 בזמן.

כעת נבצע דצימציה של פי 2 עבור

Y ו-Z

```
y_itay2= y_itay(1:2:2^16);
z_itay2= z_itay(1:2:2^16);
```

z_itay=50*sqrt(p_itay)*(cos(n*w1)+cos(n*w2)+cos(n*w3)) -> 2~->

N=n*2

z_2(n) = 50*sqrt(p_itay/2)*(cos(N*w1)+cos(N*w2)+cos(N*w3))

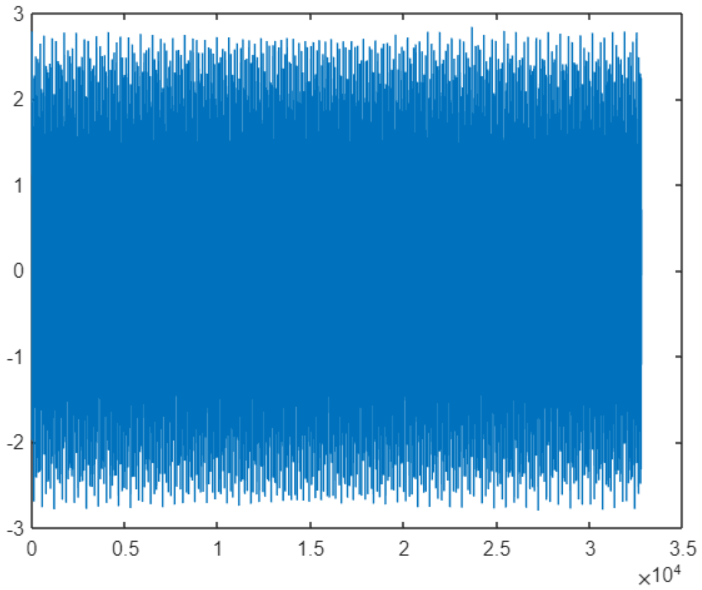
Z_2(e^-jw)=50sqrt(p_itay/2)

אפשר לראות שבהתאם לתאוריה לאחר דצימציה קיבלנו בזמן את אותו אות אולם צפוף יותר ובעל מחזורים קטנים יותר.

כעת נצייר את

y_2ואת Y_2

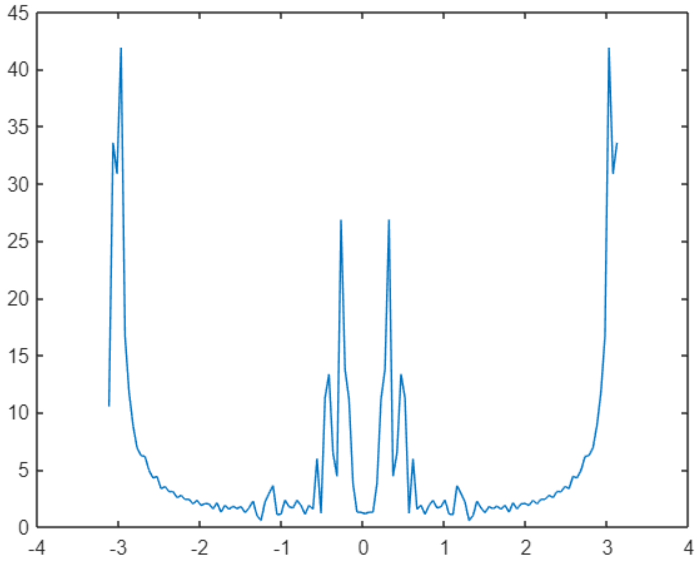
```
plot(y_itay2)
```



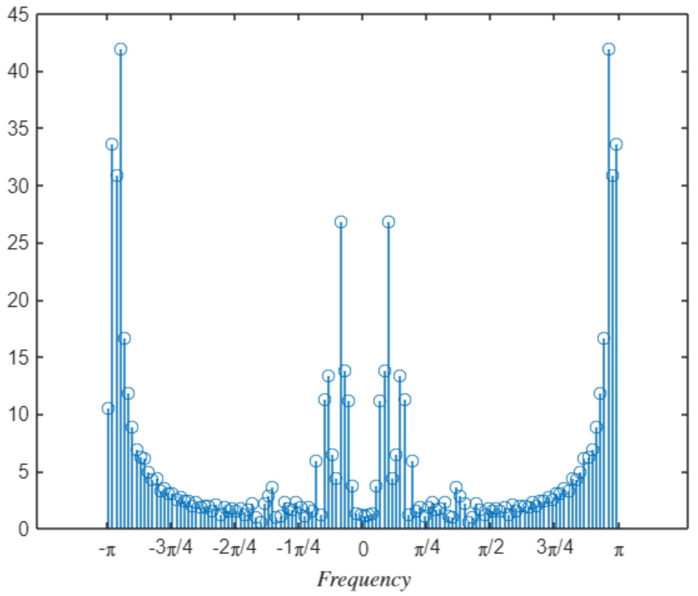
חישוב התמרת הפוריה והצגתם עבור התדרים הרלוונטים.

```
Y_ITAY2=fft(y_itay2);
M = 128;
Y_plot = Y_ITAY2(1:size(Y_ITAY2,1)/M:end);
Y_plot = circshift(Y_plot,M/2);
W = 2*pi/M * (-(M-1)/2:(M-1)/2);
```

```
figure
plot(W,abs(Y_plot));
```




```
figure
stem(W,abs(Y_plot));
xlabel('$$ Frequency $$','Interpreter','latex');
set(gca,'XTick',-pi:pi/4:pi,'XTickLabel',{'-\pi','-3\pi/4','-
2\pi/4','-1\pi/4','0','\pi/4','\pi/2','3\pi/4','\pi'})
```



גם בגרף זה אפשר לראות בצורה יפה את מימוש התאוריה בכך שגובה הגרף קטן ונעשה צפוף יותר (התדרים קטנים יותר בהתאמה לכך שעבור האות בזמן אנו מקבלים מחזורים קטנים ויותר מחזורים עבור אותו משך זמן).

לסיום נשמע את האות , נשים לב שעל מנת לשמוע אותו כראוי אנחנו צריכים להגדיל את תדירות השמע פי 2 (הקלטנו עבור 16000 הרץ ולכן עכשיו נשמע עבור תדירות 32000 הרץ) ונראה שקיבלנו את אותו אות עם הפרעה אבל בתדר נמוך יותר ולכן הוא פחות יהרוס לנו את האוזניים.
בנוסף נשים לב שבעקבות הדצימציה צליל הדיבור עצמו גם מתעוות וקשה יותר להבין מה תוכן ההקלטה.