דוח מטלת מטלב DSP1

סמסטר ב' תשפ"ב

דו"ח מסכם

תאריך הביצוע: ______ תאריך הגשת דו"ח: ________

שמות מגישי הדו"ח ומס' סטודנט:

יהודה גולדמן 206665341

נדב יוסף זדה 208825257....................................

....

שם המדריך הבודק: ______________________

ציון: ______

הערות: ____

עבורנו יצא d1=0.206665341

d2=0.208825257

יצא לנו : d=0.415490598

ולכן ממשנו FFT ע״י דצימציה בזמן.

מבוא מימוש FFT :

בחלק זה כתבנו את הפונקציות הבאות :

עבור התמרה - מעבר ממישור הזמן למישור התדר :

```
function X = dit(x)
%need to make sure that N is power of 2
N = numel(x);
if nextpow2(N)~=log2(N)
    x1=zeros(1,(2^nextpow2(N)));
    x1(1:N)=x(1:N);
    x=x1;
end
even = x(1:2:end);
odd = x(2:2:end);
if N>=4
    % The recorse part
    Even = dit(even);
    Odd = dit(odd);
    X = zeros(N,1);
    Wn = exp(-1i*(2*pi/N)*(0:(N/2)-1)');
    temp = Wn.*Odd;
    X = [(Even+temp);(Even-temp)];
end
if N==2
   X = [1 1;1 -1]*x;
end
end
```

הקוד שלנו בודק את האורך של האות, ואם הוא אינו באורך של חזקה של 2 הוא מרפד אותו באפסים עד לאורך של החזקה הכי קרובה של 2 מלמעלה. ההתמרה נעשית ע״י חלוקה של האות למקדמים זוגיים ואי-זוגיים בזמן, והתמרה בצורה רקורסיבית על כל חלק.

עבור התמרה הפוכה - מעבר ממישור התדר למישור הזמן :

```
function x = Idit(Xk1)
N = numel(Xk1);
Xk = conj(Xk1);
middle_res = dit(Xk);
x=(1/N)*conj(middle_res);
end
```

ההתמרה ההפוכה נעשית בדיוק כפי שראינו בהרצאה – פעולת הצמדה על ההתמרה, ביצוע התמרה ישירה על האות המוצמד, ולאחר מכן שוב פעולת הצמדה וחלוקה ב-N (אורך הווקטור).

דוגמת הרצה :

```
>> t=(1:4)';
T1 = dit(t)
t1 = Idit(T1)

T1 =

  10.0000 + 0.0000i
  -2.0000 + 2.0000i
  -2.0000 + 0.0000i
  -2.0000 - 2.0000i


t1 =

   1.0000 + 0.0000i
   2.0000 + 0.0000i
   3.0000 + 0.0000i
   4.0000 - 0.0000i
```

ההתמרה היא בעלת ערכים מתאימים ונכונים לפי חישוב שעשינו בתרגול 5 (ששם ראינו התמרה לאות באורך של 4 איברים).

חלק ראשון – ניתוח ועיבוד תמונה ע״י שימוש בDTFT דו מימדי :

א.

מטלת מטלב עיבוד ספרתי של אותות 1

2 א.

DTFT (חד מימדי) $\Rightarrow X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$

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

$$= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} x[n,m] e^{-j\omega_1 n} \cdot e^{-j\omega_2 m}$$

עבור הסכימה הימנית (m) נראה שהאינדקס n קבוע

$$= \sum_{n=-\infty}^{\infty} e^{-j\omega_1 n} \underbrace{\sum_{m=-\infty}^{\infty} x[n,m] e^{-j\omega_2 m}}_{\tilde{X}[n,m]}$$

$$= \sum_{n=-\infty}^{\infty} \tilde{X}[n,m] e^{-j\omega_1 n}$$

כאשר צורה זו היא של DTFT חד מימדי, והאינדקס m הוא "קבוע" כי כבר דיברנו על כל עמודה fft לכן הוא לא "משתנה" בסכימה הנוכחית על אינדקס n (השורות).

התחלנו מההגדרה של DTFT דו מימדי, פתחנו את האקספוננטים, ואז הסתמכנו על העובדה שכאשר עוברים על העמודות (אינדקס m) השורה שעוברים עליה היא אותה שורה, כלומר האינדקס n הוא קבוע כאשר עוברים על העמודות. לכן ניתן להפעיל DTFT חד מימדי על כל עמודה, ואז על התוצאה שמקבלים (מסומן אצלנו ב$\tilde{x}$) מפעילים שוב DTFT חד מימדי, הפעם בריצה על השורות.

ב.

ב. ה DFT מוגדר בצורה הבאה:

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

כאשר את הדגימות שלא היו בשלב הראשון הם אפסים, ולכן לא השתנו בפועל בדגימות

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

$$= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} x[n,m] e^{-j(\omega_1 n + \omega_2 m)} \Bigg|_{\omega_1 = \frac{2\pi k_1}{N},\ \omega_2 = \frac{2\pi k_2}{M}}$$

$$= X(e^{j\omega_1}, e^{j\omega_2})$$

התחלנו עם ההגדרה של DFT דו מימדי, אנחנו יכולים לשנות את הסכימות הנתונות לסכימות אינסופיות, כאשר כל שאר האיברים שלא היו בסכימה המקורית הם אפסים. לאחר מכן אנחנו מציבים w1, w2 במקום $\frac{2\pi k1}{N}, \frac{2\pi k2}{M}$, ואז קיבלנו את ההגדרה של DTFT דו מימדי. כלומר (בדרך ההפוכה) דגמנו את התמרת ה DTFT בתדרים : $\frac{2\pi k1}{N}, \frac{2\pi k2}{M}$.

ג.

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

$$\overset{\text{הגדרת } x[n,m]}{=} \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} y(n)\cdot z(m)\, e^{-j\omega_1 n}\, e^{-j\omega_2 m} =$$

$$= \underbrace{\sum_{n=-\infty}^{\infty} y(n)\, e^{-j\omega_1 n}}_{Y(e^{j\omega_1})} \cdot \underbrace{\sum_{m=-\infty}^{\infty} z(m)\, e^{-j\omega_2 m}}_{Z(e^{j\omega_2})}$$

התחלנו עם ההגדרה של DTFT דו מימדי, ואז הצבנו את האות שנתון בשאלה – מכפלה של אות שתלוי בn באות שתלוי בm. לאחר מכן פיצלנו את האות והאקספוננטים לסכימות הרלוונטיות אליהם (כל אחד לפני האינדקס זמן שהוא תלוי בו), ואז פשוט מקבלים הכפלה של 2 סכימות שכל אחת מהם היא הגדרה של DTFT חד מימדי.

ד.

$X[k_1,k_2]$

$$y[n] = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \xrightarrow{DFT} \sum_{n=0}^{31} y[n]\, e^{-jk\frac{2\pi}{32}n} = \sum_{n=0}^{6} 1\cdot e^{-jk\frac{\pi}{16}n} = \sum_{n=0}^{6} \left(e^{-jk\frac{\pi}{16}}\right)^n$$

סכום סדרה הנדסית סופית

$$\Rightarrow \frac{a_1(q^n-1)}{q-1} = \frac{\left(e^{-jk\pi/16}\right)^7-1}{e^{-jk\pi/16}-1} \cdot \frac{\left(e^{-jk\pi/32}\right)^7-1}{e^{-jk\pi/32}-1} = res$$

נקבל עבור z[n] ←

כאשר היה התוצאה, לכן התוצאה

הסופית של ה DFT הדו-מימדי

תהיה הכפלה של שניהם, בהתאם לסעיף ג.

<u>חישוב אנליטי</u>: בחישוב זה הסתמכנו על העובדה שהוכחנו בסעיף ג. המטריצה הנתונה ניתנת לחישוב ע״י הכפלה של 2 וקטורים, כאשר הווקטור הראשון הוא וקטור עמודה באורך 32 מהצורה: $(111111100\ \ldots\ldots 0000)^T$, וכנ״ל הווקטור השני הוא וקטור שורה באורך 64 מהצורה: $(111111100\ \ldots\ldots 0000)$. על כל אחד מהווקטורים עשינו התמרת DTFT חד מימדי, והתשובה הסופית היא מכפלה של 2 ההתמרות.

<u>חישוב ע״י matlab</u>:

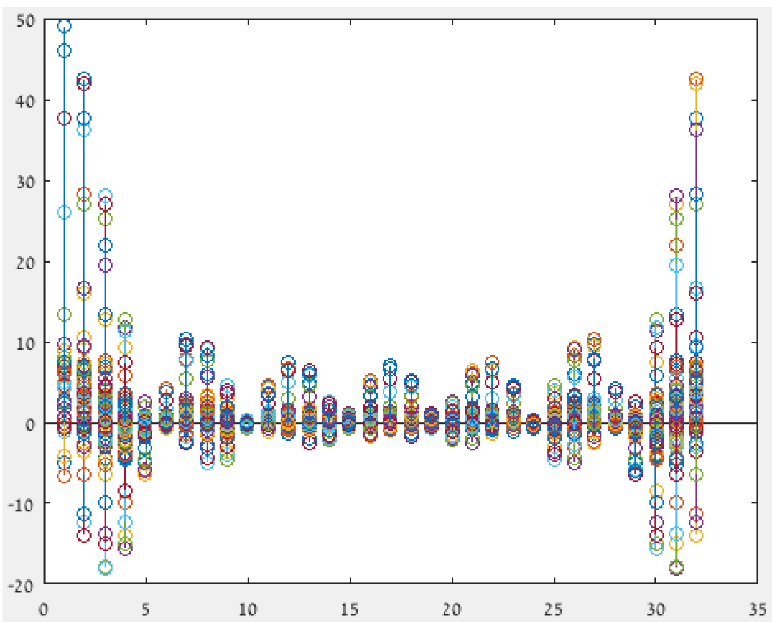
כתבנו את הקוד הבא:

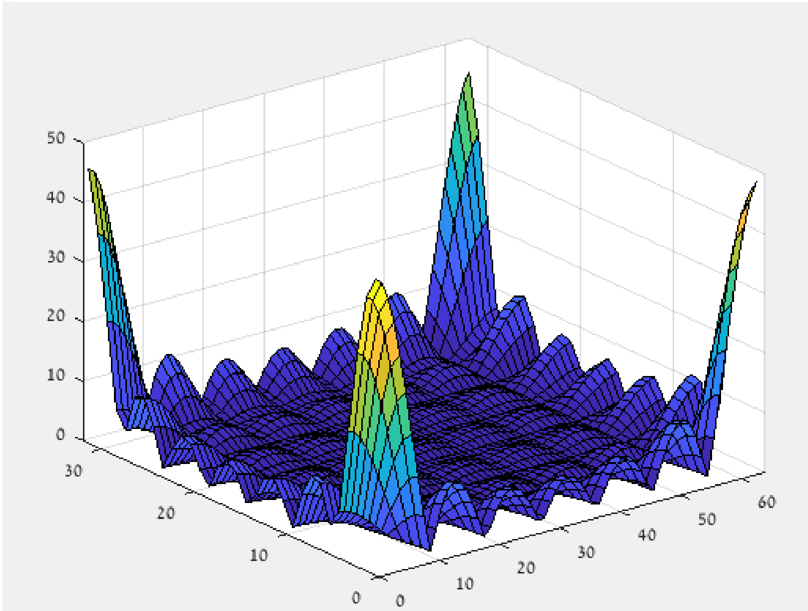
```
image=zeros(32,64);
image(1:7,1:7)=1;
X = Two_d_dft(image); %סעיף ד
stem(X)
figure
surf(abs(X))

function X = Two_d_dft(x)
N = size(x,1);
M = size(x,2);
X = zeros(N,M);
for i=0:M-1
    X(:,i+1) = fft(x(:,i+1));
end
for j=0:N-1
    X(j+1,:) = fft(X(j+1,:));
end
end
```

וקיבלנו את המטריצה והגרפים הבאים:

32x64 complex double

	1	2	3	4	5	6	7	8	9
1	49.0000 + 0....	45.9913 - 13...	37.6705 - 25...	25.9589 - 31...	13.4672 - 32...	2.7932 - 28....	-4.1490 - 20...	-6.5778 - 12...	-4.9497 - 4....
2	37.6705 - 25...	28.1908 - 34...	16.0307 - 38...	3.7084 - 37....	-6.3479 - 31...	-12.4207 - 2...	-13.9042 - 1...	-11.3784 - 6...	-6.3479 - 1....
3	13.4672 - 32...	3.3832 - 34....	-6.3479 - 31...	-13.8533 - 2...	-17.8715 - 1...	-18.0497 - 9...	-14.9802 - 2...	-9.9733 + 0....	-4.6447 + 1....
4	-4.1490 - 20...	-9.8330 - 18...	-13.9042 - 1...	-15.6626 - 8...	-14.9802 - 2...	-12.3087 + ...	-8.5276 + 3....	-4.6815 + 3....	-1.6879 + 2....
5	-4.9497 - 4....	-6.0551 - 3....	-6.3479 - 1....	-5.8175 + 0....	-4.6447 + 1....	-3.1469 + 2....	-1.6879 + 2....	-0.5787 + 1....	0.0000 + 1.0...
6	4.2278 + 0.8...	4.2076 - 0.4...	3.6822 - 1.5...	2.7826 - 2.2...	1.7199 - 2.5...	0.7277 - 2.3...	0.0000 - 1.8...	-0.3563 - 1....	-0.3421 - 0....
7	9.6788 - 4.0...	7.9430 - 6.5...	5.3815 - 8.0...	2.5396 - 8.3...	0.0000 - 7.5...	-1.7686 - 5....	-2.5261 - 3....	-2.3062 - 1....	-1.3827 - 0....
8	6.1007 - 9.1...	3.1265 - 10....	-0.0000 - 10...	-2.6619 - 8....	-4.3815 - 6....	-4.9367 - 4....	-4.4032 - 1....	-3.1120 - 0....	-1.5386 + 0....
9	0.0000 - 7.0...	-1.9930 - 6....	-3.5958 - 5....	-4.5187 - 3....	-4.6447 - 1....	-4.0514 - 0....	-2.9797 + 0....	-1.7580 + 0....	-0.7071 + 0....
10	-0.4931 - 0....	-0.6730 - 0....	-0.7582 - 0....	-0.7376 - 0....	-0.6252 + 0....	-0.4552 + 0....	-0.2724 + 0....	-0.1192 + 0....	-0.0247 + 0....
11	4.3212 + 1.7...	4.5655 + 0.4...	4.2415 - 0.8...	3.4447 - 1.8...	2.3753 - 2.3...	1.2823 - 2.3...	0.3960 - 1.9...	-0.1306 - 1....	-0.2557 - 0....
12	7.4495 - 1.4...	6.5702 - 3.5...	4.9659 - 4.9...	2.9900 - 5.5...	1.0642 - 5.3...	-0.4330 - 4....	-1.2615 - 3....	-1.3722 - 1....	-0.9022 - 0....





ה. בסעיף זה חישבנו את התמרת ה DTFT של h0 בתדרים : $0, \frac{2\pi}{6}, 2 * \frac{2\pi}{6}, 4 * \frac{2\pi}{6}$. הסתמכנו על העובדה שהתמרת DFT היא דגימה של התמרת DTFT בתדרים : $\frac{2\pi k}{N}$.

כתבנו את הקוד הבא :

```
[dist_image_1,dist_image_2,noised_image,imp_resp_image]=img_gen('YEHUDA','NADAV');
h00=imp_resp_image(:,1);
X0124 = [];
X0124=dtft_h0(h00);
h0=cat(2,(h00)',zeros(1,29));

function T = dtft_h0(h00)
    X = fft(h00);
    T=zeros(1,4);
    T(1) = X(1); %T(1) is X(0)
    T(2) = X(2); %T(2) is X(1)
    T(3) = X(3); %T(3) is X(2)
    T(4) = X(2); %T(4) is X(1)
end
```

במקום להרחיב את h0 לווקטור של 6 ערכים ע"י ריפוד ב3 דגימות של אפסים, עשינו התמרה רגילה לווקטור המקורי, ובתור התדר הרביעי שצריך לחשב לקחנו את התוצאה של התדר השני, מפני ש :

$$X[4] = \sum_{n=0}^{N-1} x[n] * e^{-\frac{i(1+3)2\pi n}{3}} = \sum_{n=0}^{N-1} x[n] * e^{-\frac{i*1*2\pi n}{3}} * \underbrace{e^{-\frac{i*3*2\pi n}{3}}}_{\boxed{=1}} = X[2]$$

תוצאה מספרית :

1x4 complex double

	1	2	3	4
1	0.0909 + 0.0000i	-0.0455 + 0.0787i	-0.0455 - 0.0787i	-0.0455 + 0.0787i

ו. כתבנו את הקוד הבא:

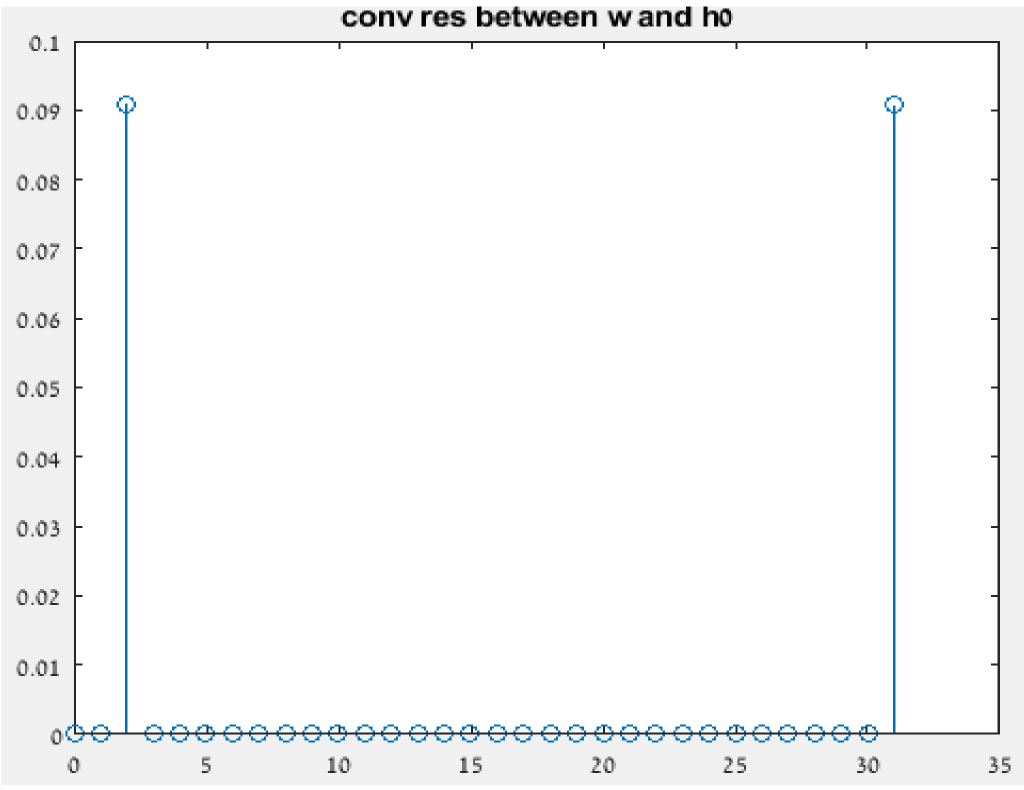
```
x = 0:1:31;
w = dirac(x)+dirac(x-29);
idx = w == Inf; % find Inf
w(idx) = 1;     % set Inf to finite value
W=fft(w);
H0=fft(h0);
conv_res = W.*H0;
conv_res = ifft(conv_res);
stem(x,conv_res)
title("conv res between w and h0")
```

חישוב הקונבלוציה נעשה ע"י התמרת DFT לכל אות בנפרד, ואז הכפלה בתדר של ההתמרות, ולבסוף התמרה הפוכה של התוצאה מפני שידוע שקונבלוציה ציקלית בזמן שקולה להכפלת התמרות ה DFT בתדר.

קיבלנו את התוצאה הבאה:

1x32 double

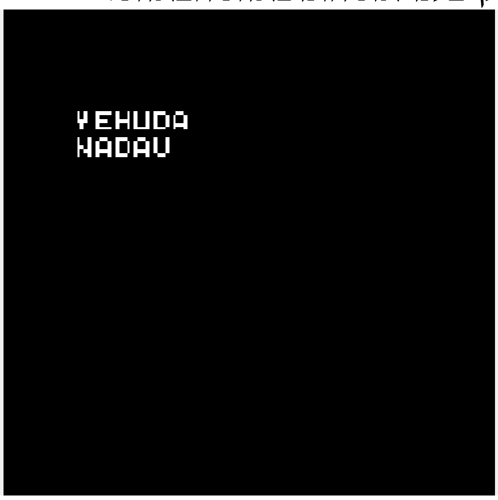
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	8.6736e-19	2.4927e-18	0.0909	-8.6736e-18	1.8400e-18	-1.6280e-18	0	6.9389e-18	0	4.9582e-19	0	-1.7347e-18	-1.8400e-18	-2.4365e-18	0



ז. כתבנו את הקוד הבא:

```
h1=zeros(128,128);
h2=zeros(70,170);
h1([1:3],[1:5])=imp_resp_image;
h2([1:3],[1:5])=imp_resp_image;
H1=Two_d_dft(h1);
H2=Two_d_dft(h2);
Y1 = Two_d_dft(dist_image_1);
Y2 = Two_d_dft(dist_image_2);
X1 = Y1./H1;
X2 = Y2./H2;
x1_time=Two_d_Idft(X1);
x2_time=Two_d_Idft(X2);
imshow(x1_time)
imshow(x2_time)


function x = Two_d_Idft(X)
N = size(X,1);
M = size(X,2);
x = zeros(N,M);
for j=0:N-1
    x(j+1,:) = ifft(X(j+1,:));
end
for i=0:M-1
    x(:,i+1) = ifft(x(:,i+1));
end
end
```

קיבלנו את התוצאות הבאות:



החישוב התבצע על סמך ההסבר שהתמונות התקבלו מהפעלה של קונבלוציה ציקלית בין התמונה המקורית לתגובה להלם של הערוץ. במישור הזמן נעשתה קונבלוציה ציקלית ולכן זה שקול לפעולת כפל בין ההתמרות. כלומר ההתמרה של התמונה המקורית כפול ההתמרה של התגובה להלם של הערוץ שווה להתמרה של התמונה שהתקבלה ($Y=X*H$). לכן אם נחלק את Y בH (שיש לנו) נקבל את ההתמרה של התמונה המקורית – X, ועליה נוכל לחשב התמרה הפוכה ולקבל את התמונה המקורית בזמן.

ח. בנוגע לתמונה x1 בעצם הקונבלוציה הציקלית שנעשתה בין התמונה הזאת לתגובה של הערוץ מימשה קונבלוציה ליניארית, משום שכמו שלמדנו במקרה החד מימדי על מנת לממש קונבלוציה ליניארית בעזרת קונבלוציה ציקלית צריך לרפד את 2 האותות באפסים עד לאורך של סכום האורכים שלהם פחות אחד, וזאת על מנת להרחיק את השכפולים כך שלא ידרכו אחד על השני. וכך בעצם במקרה זה מומשה קונבלוציה ליניארית שזאת הדרך שבה יוצרה התמונה שקיבלנו (y).

בנוגע לתמונה x2 לא ניתן להגיד שמומשה קונבלוציה ליניארית באמצעות הקונבלוציה הציקלית מפני שלא היה ריפוד באפסים לאורך מוסכם, ולכן בשוליים של התמונה מקבלים הפרעות ("לכלוכים").

ט. על מנת לשחזר את x2, שזאת בעצם תמונה שיש בה 4 שכפולים של x1, בהינתן ששחזרנו כבר את x1 (באותה דרך שהראנו בסעיף ז), ניתן לשכפל אותו 4 פעמים (2 לאורך ו2 לרוחב) ולקבל את x2.

חלק שני – ייצור וניתוח אותות דיבור

תחילה הקלטנו הקלטה שלנו באורך 5 שניות בצורה הבאה :

```
% myVoice = audiorecorder(16000,8,1);
%
% % Define callbacks to show when
% % recording starts and completes.
% myVoice.StartFcn = 'disp(''Start speaking.'')';
% myVoice.StopFcn = 'disp(''End of recording.'')';
%
% record(myVoice, 5);

% audiowrite('myVoice.wav',a,16000)
```

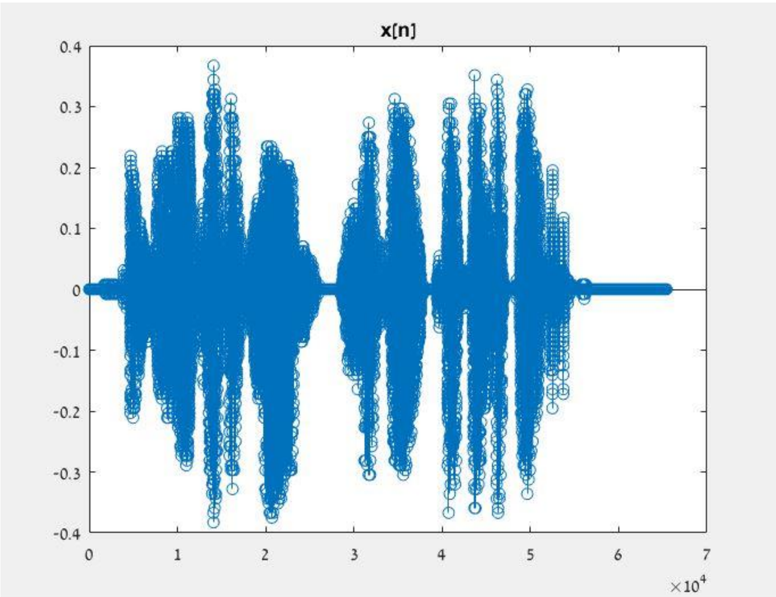
כתוצאה מכך קיבלנו הקלטה. לאחר מכן השתמשנו בפקודה audiowrite כדי לשמור את ההקלטה כקובץ ולא רק כמשתנה על מנת שנוכל להשתמש בה בהמשך.

המרנו את ההקלטה לdata array כדי שנוכל לחתוך את המערך לאורך של 2^{16}.

<u>סעיף א</u> :

חישוב ההספק הממוצע של האות.

```
%סעיף א
N = 2^16;
xn=a(:,1:N);
stem(xn)
title("x[n]")
avg_power = (1/N)*(sum(xn.*xn));
```



קיבלנו :

```
avg_power =

    0.0035
```

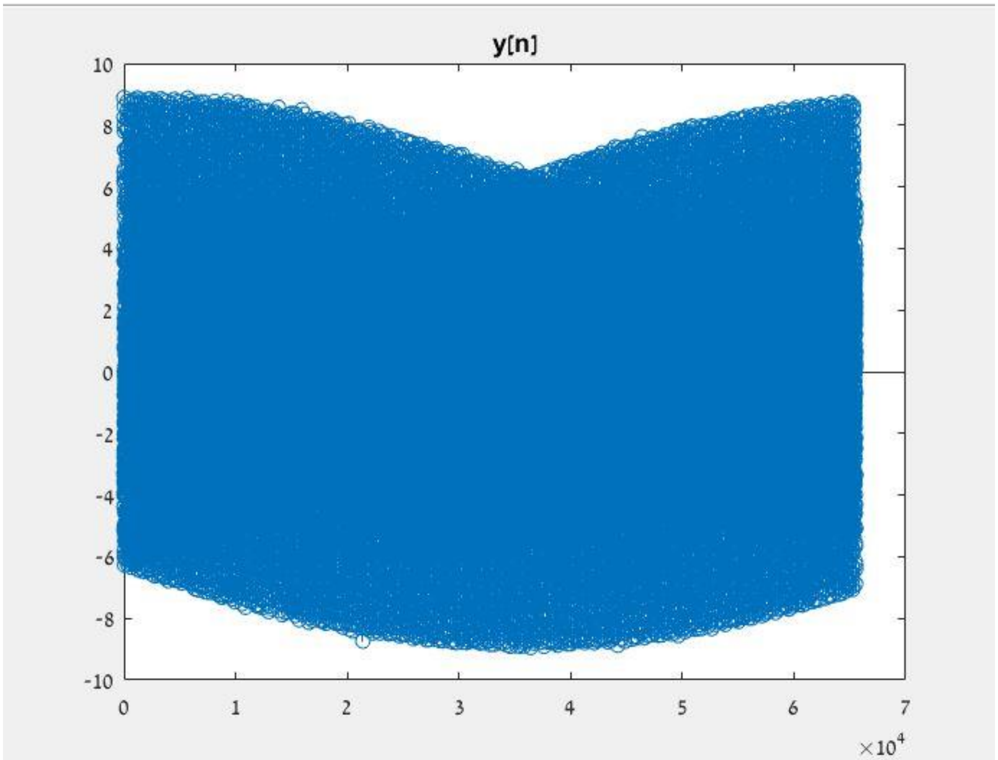
כעת הגדרנו את אות ההפרעה zn ואת אות הכניסה שהוגדר להיות החיבור של xn עם zn (רעש/הפרעה אדיטיבית).

<u>סעיף ב</u> :

השמענו את yn ושמענו את מה שהקלטנו באיכות נמוכה יותר וחלש יותר, יחד עם ציפצוף גבוה וחזק.

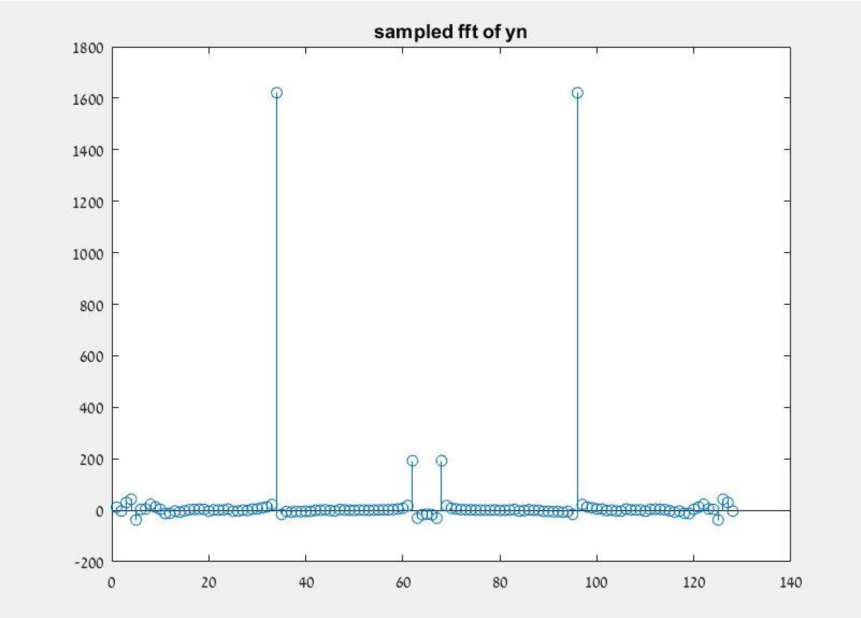
```
%סעיף ב
d1=0.206665341;
d=0.415490598;
w1=1.6+0.1*d1;
w2=1.6+0.1*d;
w3=3;
n=0:N-1;
a1=50*(sqrt(avg_power));
zn=a1.*(cos(w1.*n)+cos(w2.*n)+cos(w3.*n));
yn=xn+zn;
soundsc(yn,16000)
```

<u>סעיף ג</u> :



סעיף ד :

בסעיף זה נתבקשנו לשרטט את ה DTFT של yn בתדרים הרלוונטים.
זה למעשה לשרטט את ה DFT של yn כי אנו יודעים מההרצאה שהDFT זה דגימה של ה DTFT .
האות yn הוא באורך של 65536 דגימות. אנו יודעים כי אורך התמרת הDFT זהה לאורך של האות בזמן. אז על מנת לקבל 128 דגימות של הDTFT כפי שנתבקשנו (הN הוא 128), נדגום מהתמרת הDFT כל 512 דגימות.




```
%סעיף ד
Y=fft(yn);
Y_sample = Y(1:512:N);
stem(Y_sample)
title("sampled fft of yn");
```

סעיף ה :

חלק 2 סעיף ה

$$z[n] = 50\sqrt{P_x}\left[\cos(w_1 n) + \cos(w_2 n) + \cos(w_3 n)\right]$$

$$\boxed{z_2[n] = 50\sqrt{P_x}\left[\cos(2w_1 n) + \cos(2w_2 n) + \cos(2w_3 n)\right]}$$

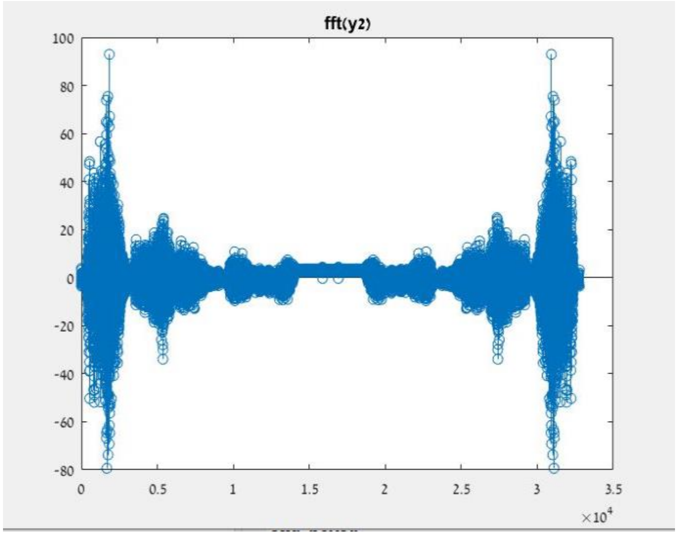
$$Z_2(e^{jw}) = \frac{1}{2}\sum_{m=0}^{1} Z\left(e^{j(w-2\pi m)/2}\right)$$

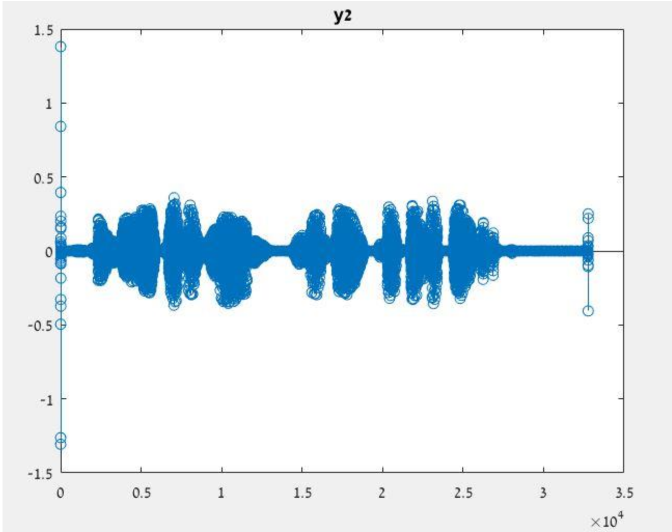
$$= \frac{1}{2}\left[Z(e^{jw/2}) + Z\left(e^{j\frac{(w-2\pi)}{2}}\right)\right]$$

כאשר:

$$Z(e^{jw}) = \pi\left[\delta(w+w_1) + \delta(w-w_1) + \delta(w+w_2) + \delta(w-w_2) + \delta(w+w_3) + \delta(w-w_3)\right]$$

<u>סעיף ו</u> :






```
y2=decimate(yn,2);
stem(y2)
title("y2")
figure
stem(fft(y2))
title("fft(y2)")
```

האזנו לy2 וההקלטה הייתה באיכות מעט נמוכה יותר מההקלטה המקורית, אבל הייתה הרבה יותר נקייה מאשר האות פלוס ההפרעה. השמענו את האות בקצב של 8,000hz (לעומת 16,000hz בהקלטה המקורית) וזאת משם שקיבלנו פחות דגימות לאחר הדצימציה (בפקטור 2), ולכן צריך להשמיע את הדיבור בקצב נמוך פי חצי.

למעשה הדצימציה היוותה סינון של האות.