

## תרגיל DSP - MATLAB

הפרמטרים שהתקבלו:

$$d_1 = 0.211715529, \quad d_2 = 0.212466782, \quad d = \text{mod}((d_1 + d_2), 0.5) = 0$$

### 1. מבוא- מימוש FFT

עבור אות סופי בזמן בדיד  $x[n]$ ,  $n = 1:N-1$ , התמרת DFT הינה:

$$X[k] = \text{DFT}\{x[n]\} = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn}, \quad k = 0:N-1$$

FFT הינו אלגוריתם למימוש DFT באופן מהיר יותר.

ניתן לממש FFT ע"י דצימציה בזמן/ בתדר. אנו קיבלנו כי  $d \bmod 0.1 = 0.0242 < 0.05$ , ולכן מימשנו ע"י דצימציה בזמן.

FFT באמצעות דצימציה בזמן:

$$X[k] = G[\langle k \rangle_L] + W_N^k \cdot H[\langle k \rangle_L], \quad k = 0:N-1$$

$$G[k] = \text{FFT}\{g[n] \triangleq x[2n]\}, \quad H[k] = \text{FFT}\{h[n] \triangleq x[2n+1]\}, \quad W_N^k = e^{-j \frac{2\pi}{N} k}, \quad L = \frac{N}{2}$$

הערה: אלגוריתם זה מניח כי  $x[n]$  הינו אות באורך  $N$ , כאשר  $N$  הינו חזקה שלמה של 2. עבור אות שאינו באורך כזה, יש לרפד את  $x[n]$  באפסים בסוף, עד לאורך כולל של  $N$  שמקיים את ההנחה. גם את הריפוד באפסים במקרה הצורך מימשנו בפונקציה.

אנו מימשנו את האלגוריתם באופן רקורסיבי. ניתן להבחין כי האותות  $g, h, G, H$  הינם באורך של  $\frac{N}{2}$ . בעצם עבור  $\frac{N}{2}$  האינדקסים הראשונים של ההתמרה,

$$X[k]_{part\_1} = G[k] + W_N^k \cdot H[k], \quad k = 0:\frac{N}{2}-1$$

ועבור האינדקסים האחרונים,

$$X[k]_{part\_2} = G\left[k - \frac{N}{2}\right] + W_N^k \cdot H\left[k - \frac{N}{2}\right], \quad k = \frac{N}{2}:N-1$$

לכן בתוך הפונקציה חישבנו (באופן רקורסיבי-ע"י קריאה לפונקציה עצמה) את  $G[k]$ ,  $H[k]$ , והתוצאה ( $X[k]$ ) היא בעצם שרשור פעמיים של  $X[k]_{part\_1}$ , כאשר השינוי היחיד בין האיבר הימני לשמאלי בשרשור הוא  $W_N^k$ .

IFFT:

כפי שלמדנו בקורס,

$$\text{IDFT}\{X[k]\} = \frac{1}{N} (\text{DFT}\{X^*[k]\})^*$$

להלן מימוש FFT ע"י דצימציה בזמן:

```
function [X] = my_fft(xshift)
Xshift=my_fftshift(xshift);
X=fftshift(Xshift);
end
```

```
function [X] = my_fftshift(x)
N=length(x);

if N==1
    X=x;
    return
end

if floor(log2(N))~=log2(N)
    c=ceil(log2(N));
    x=[x zeros(1, 2^c-N)];
    N=2^c;
end

even=1:2:N;
odd=2:2:N;

X_even=my_fftshift(x(even));
X_odd=my_fftshift(x(odd));

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

השתמשנו בפונקציה המובנית `fftshift`, משום שתוצאת אלגוריתם FFT היא עבור תחום התדרים  $[0, 2\pi]$ , ואילו אנו מעוניינים בתחום  $[-\pi, \pi]$ . פונקציה זו בעצם הופכת בין 2 חצאי ההתמרה, לקבלת התוצאה הרצויה.

להלן מימוש IFFT באמצעות FFT:

```
function [x]=my_ifft(X)
N=length(X);
x=(conj(fftshift(my_fft(conj(X))))) / N;
end
```

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

א. התמרת DTFT בשני מימדים בדידים (n,m) מוגדרת לפי:

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

התמרת DTFT חד מימדית מוגדרת לפי:

$$X(e^{jw}) = \sum_{n=-\infty}^{\infty} x[n] e^{-jwn}$$

נוכיח כי ניתן לחשב את DTFT הדו מימדית באמצעות הפעלה של DTFT חד מימדי על כל עמודה (m קבוע), ולאחר מכן הפעלת DTFT חד מימדי על כל שורה:

$$DTFT \left\{ x \left[ n, \underset{\text{const}}{m} \right] \right\} \triangleq X(e^{jw_1}) = \sum_{n=-\infty}^{\infty} x[n, m] e^{-jw_1 n}$$

$$DTFT \{ X(e^{jw_1}) \} \triangleq X(e^{jw_1}, e^{jw_2}) = \sum_{m=-\infty}^{\infty} \underbrace{\sum_{n=-\infty}^{\infty} x[n, m] e^{-jw_1 n}}_{X(e^{jw_1})} e^{-jw_2 m}$$

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

ב. נוכיח כי התמרת DFT הדו מימדית מתקבלת מדגימות של התמרת DTFT הדו מימדית:

$$X[k_1, k_2] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x[n, m] e^{-j(\frac{2\pi n}{N} k_1 + \frac{2\pi m}{M} k_2)} = X(e^{j\omega_1}, e^{j\omega_2}) \cdot \sum_{l=0}^{N-1} \sum_{g=0}^{M-1} \delta[w_1 - \frac{2\pi}{N} l, w_2 - \frac{2\pi}{M} g]$$

ניתן לראות כי תדרי הדגימה הם  $\{l = 0: N-1, g = 0: M-1\}$   $w_1 = \frac{2\pi}{N} l, w_2 = \frac{2\pi}{M} g$ .

ג. נוכיח כי עבור  $x[n, m] = y[n] \cdot z[m]$  מתקבל  $X(e^{jw_1}, e^{jw_2}) = Y(e^{jw_1})Z(e^{jw_2})$ :

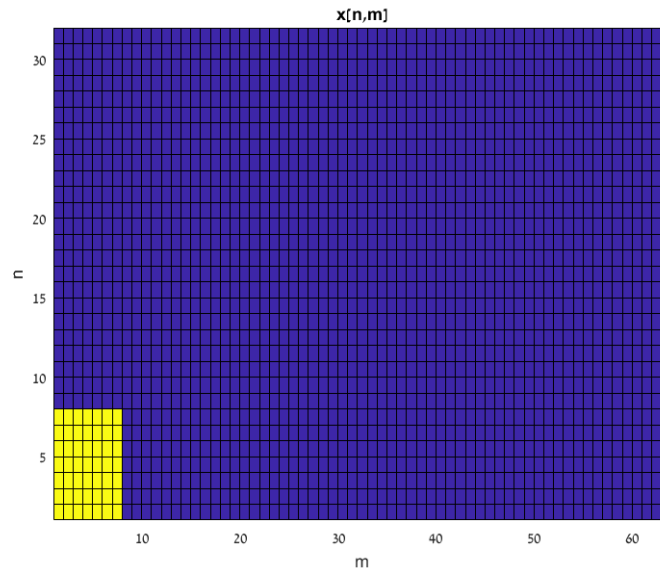
$$\begin{aligned} X(e^{jw_1}, e^{jw_2}) &= DTFT \{ x[n, m] \} \stackrel{\text{סעיף א'}}{\triangleq} \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} x[n, m] e^{-j(\omega_1 n + \omega_2 m)} \\ &= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} y[n] \cdot z[m] e^{-j(\omega_1 n + \omega_2 m)} = \sum_{n=-\infty}^{\infty} y[n] e^{-jw_1 n} \sum_{m=-\infty}^{\infty} z[m] e^{-jw_2 m} \\ &= Y(e^{jw_1}) Z(e^{jw_2}) \end{aligned}$$

ד. נחשב באופן אנליטי את התמרת DFT של האות:

$$x[n, m] = \begin{cases} 1 & 0 \leq n < B_1, 0 \leq m < B_2 \\ 0 & 0 \leq n < N, B_2 \leq m < M \end{cases}$$

כאשר  $N = 32, M = 64, B_1 = B_2 = 7$ .

נציג תחילה את האות לשם המחשה:



זהו בעצם חלון דו מימדי בזמן. מהתמרת חלון חד מימדי בדיד מתקבל גרעין דריכלה, ולכן נצפה לקבל גרעין דריכלה דו מימדי בתדר.

ראשית נחשב את ההתמרה באופן אנליטי:

תחילה נתמיר את העמודות:

$$DFT \left\{ x \left[ \underbrace{n}_{const}, m \right] \right\} = X_{col}[k, l] = \begin{cases} \frac{\sin\left(\frac{7\pi k}{32}\right)}{\sin\left(\frac{\pi k}{32}\right)} & l = 1:7 \\ 0 & else \end{cases}$$

כלומר כעת כל שורה היא חלון באורך 7, כאשר גובה החלון משתנה משורה לשורה עפ"י גבהי גרעין הדריכלה בנקודות אלו, ובתוספת אפסים בצד ימין.

כעת נתמיר את השורות:

$$\Rightarrow DFT\{x[n, m]\} = DFT\{X_{col}[k, l]\} = X[k, l] = \frac{\sin\left(\frac{7\pi k}{32}\right)}{\sin\left(\frac{\pi k}{32}\right)} \cdot \frac{\sin\left(\frac{7\pi l}{64}\right)}{\sin\left(\frac{\pi l}{64}\right)}$$

ניתן לראות כי קיבלנו בכל מימד גרעיני דריכלה. נשים לב כי עבור  $m$  קבוע, נקבל אונות (ראשית ומשניות) רחבות יותר אך נמוכות יותר ביחס למקרה של  $l$  קבוע, בו נקבל אונות צרות וגבוהות יותר.

הערה: מכיוון שאנו מעוניינים במגניטודה, התייחסנו מראש לאות כאל חלון סימטרי ולא מוזז. התמרת חלון מוזז כוללת כפל באקספוננט מרוכב, ועקב הסיבה שציינו, תוצאת החישוב האנליטי שלנו לא כוללת אקספוננט מרוכב.

חישוב ההתמרה באופן נומרי:

על מנת לחשב נומרית את ההתמרה, מימשנו את פונקציות FFT2, IFFT2.

להלן מימוש FFT2:

```
function [matrixFFT] = my_fft2 (matrix)
N=size(matrix,1);
M=size(matrix,2);

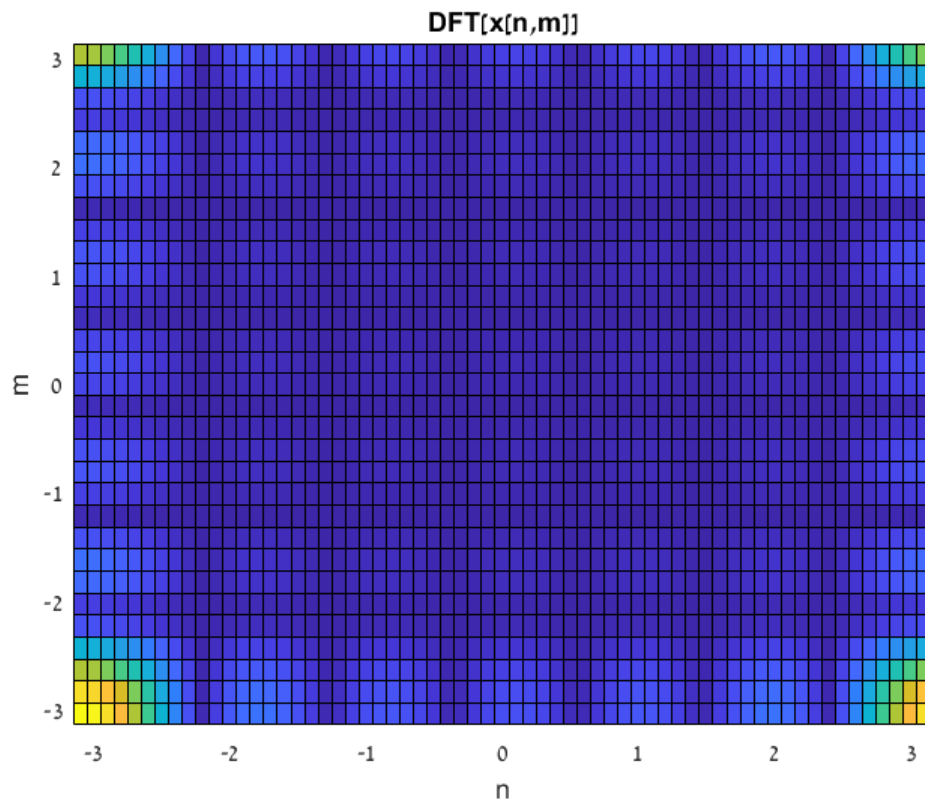
if (floor(log2(N))~=log2(N)) || (floor(log2(M))~=log2(M))
    cN=ceil(log2(N));
    cM=ceil(log2(M));
    matrix_padded=zeros(2^cN, 2^cM);
    for k=1:N
        for l=1:M
            matrix_padded(k,l)=matrix(k,l);
        end
    end
    N=2^cN;
    M=2^cM;
else
    matrix_padded=matrix;
end

matrixFFT_cols=zeros(N,M);
matrixFFT=zeros(N,M);
for m=1:M
    matrixFFT_cols(:,m)=fft(matrix_padded(:,m)');
end
for n=1:N
    matrixFFT(n,:)=fft(matrixFFT_cols(n,:));
end
end
```

להלן מימוש IFFT2:

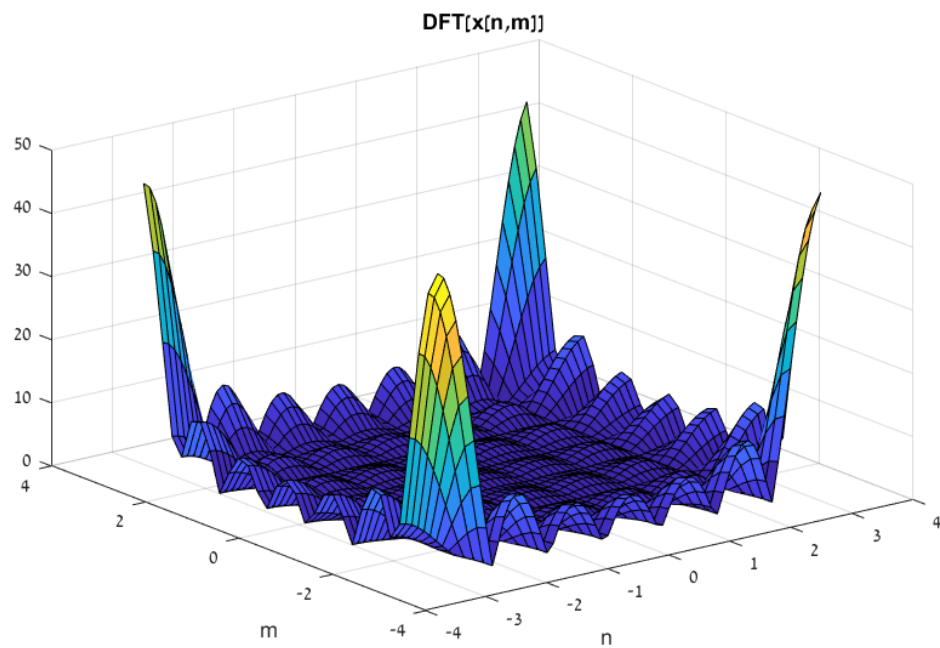
```
function [x]=my_ifft2(X)
N=size(X, 1);
M=size(X, 2);
x=(conj(fft2(conj(X))))/(N*M);
end
```

להלן ההתמרה שהתקבלה:



ניתן לראות כי אכן קיבלנו גרעין דריכלה בכל ציר (בגובה מתאים לכל שורה/ עמודה), כאשר היחסים בין הגרעינים הם כפי שחישבנו.

נראה את התוצאה בדרך נוספת:



ה. להלן 4 המטריצות שהתקבלו מהפונקציה `img_gen`:

$y_1[n, m]$  - תמונה שהתקבלה ממעבר של תמונה מקורית דרך ערוץ, שמדמה תנועה אקראית של המצלמה תוך כדי הצילום:

$y_1[n, m]$



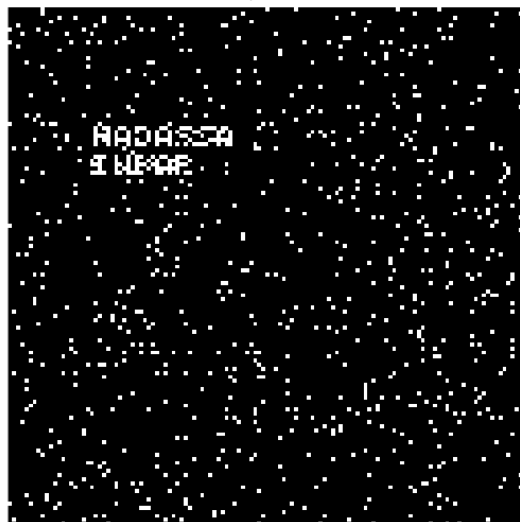
$y_2[n, m]$  - מעבר של תמונה באותו ערוץ:

$y_2[n, m]$

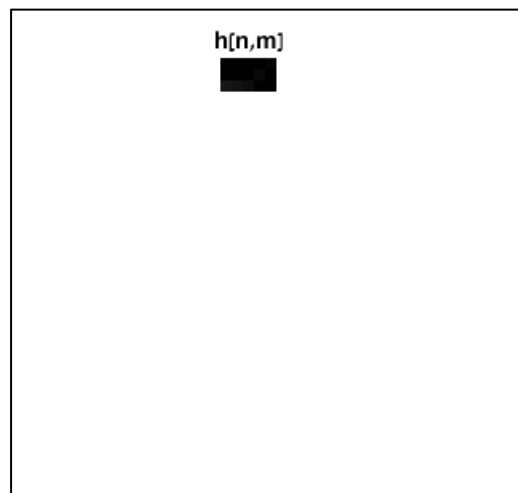


$y_3[n, m]$  - תמונה שהתקבלה מתוספת של רעש בתדרים גבוהים לתמונה מקורית:

$y_3[n, m]$



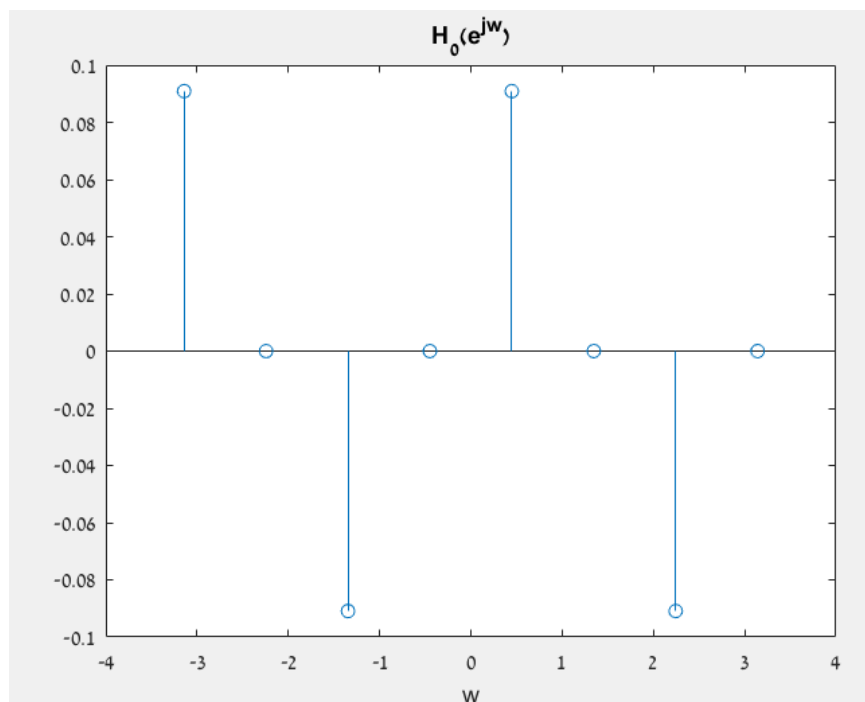
$-h[n, m]$  - תמונה המכילה את התגובה להלם של הערוץ:



מעתה השתמשנו בעמודה הראשונה מתוך התגובה להלם של הערוץ:

$$h_0[n] = h[n, 0], \quad N = 3$$

להלן  $H_0(e^{jw}) = DTFT\{h_0[n]\}$  בתדירים  $w = 0, \frac{2\pi}{6}, 2 \cdot \frac{2\pi}{6}, 4 \cdot \frac{2\pi}{6}$



נשים לב כי אורך  $h_0[n]$  הוא 3, ולכן גם אורך  $H_0(e^{jw})$  הוא 3. על מנת לקבל את ההתמרה בתדירים הדרושים (כפולות מסוימות של  $\frac{2\pi}{6}$ ), יש להביא לכך שאורך  $H_0(e^{jw})$  יהיה 6. לכן, ריפדנו את  $h_0[n]$  בשלושה אפסים בסוף.

קיבלנו:

$$H_0(e^{j0}) = 0.0909, \quad H_0\left(e^{j \cdot \frac{2\pi}{6}}\right) = -0.0909, \quad H_0\left(e^{j \cdot 2 \cdot \frac{2\pi}{6}}\right) = 0.0909, \quad H_0\left(e^{j \cdot 4 \cdot \frac{2\pi}{6}}\right) = 0.0909$$



ו. כעת חישבנו את הקונבולוציה הציקלית במחזור 32 של  $h_0[n]$  עם  $w[n]$ , כאשר:

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

אנו מימשנו את הקונבולוציה הציקלית באמצעות הכפלת  $w[n]$  (כווקטור עמודה) במטריצה הסירקולנטית של  $h_0[n]$ , כפי שלמדנו:

$$\begin{bmatrix} (a \circledast_N v)[0] \\ (a \circledast_N v)[1] \\ \vdots \\ (a \circledast_N v)[N-1] \end{bmatrix} = \begin{bmatrix} a[0] & a[N-1] & \dots & a[1] \\ a[1] & a[0] & a[N-1] & \dots & a[2] \\ \vdots & a[1] & a[0] & & \vdots \\ & & & \ddots & \\ a[N-1] & a[N-2] & \dots & & a[0] \end{bmatrix} \begin{bmatrix} v[0] \\ v[1] \\ \vdots \\ v[N-1] \end{bmatrix}$$

להלן מימוש הקונבולוציה הציקלית:

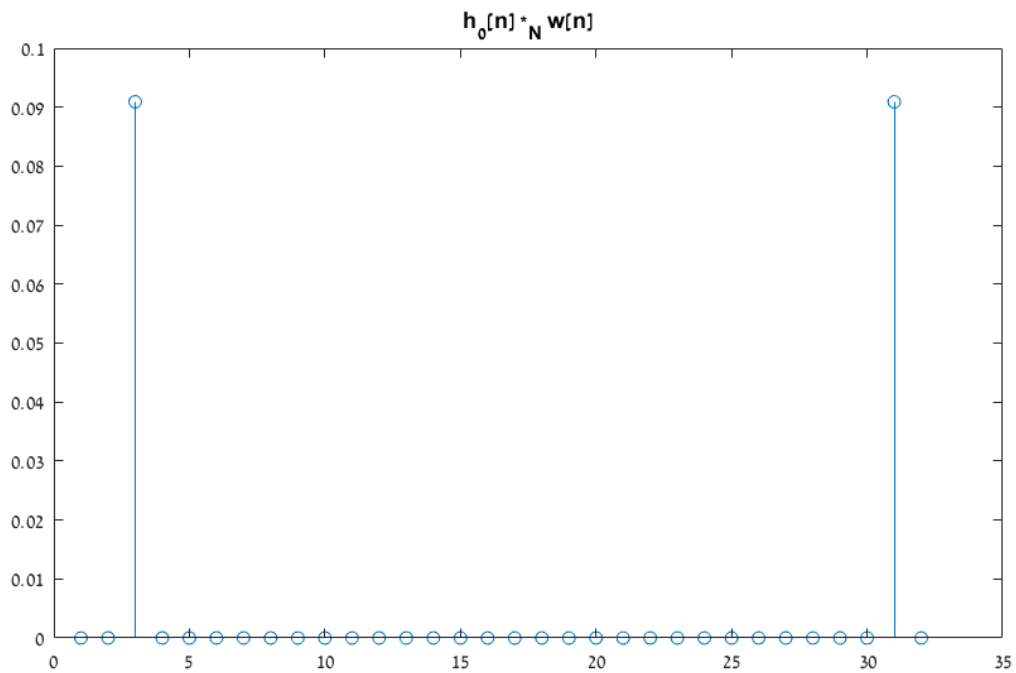
```
function [z]= my_convolution (x,y)
N=length(x);
M=length(y);
if N>M
    Nmin=M;
    Nmax=N;
    y=[y zeros(1,N-M-1)];
    min=x;
    max=y;
end
if M>N
    Nmin=N;
    Nmax=M;
    x=[x zeros(1,M-N-1)];
    min=y;
    max=x;
end

z=zeros(1,N+M-1);

for m=1:Nmax
    for l=1:Nmin
        z(m+l-1)=z(m+l-1)+max(1,m)*min(1,l);
    end
end

end
```

להלן התוצאה שהתקבלה:



נוודא בעזרת חישוב אנליטי.

נשתמש בקשר בין קונבולוציה ציקלית ללינארית:

$$\begin{aligned}
 z[n] &\triangleq h_0[n] * w[n] = 0.0909(\delta[n-2] + \delta[n-31]) \\
 \Rightarrow h_0[n] \circledast_{32} w[n] &= z[n] + z[n+32] = 0.0909(\delta[n-2] + \delta[n-31]) + \underbrace{z[n+32]}_{=0 \text{ } (\forall n: 0 \leq n \leq 32)} = \\
 &0.0909(\delta[n-2] + \delta[n-31])
 \end{aligned}$$

ז. כעת רצינו לשחזר את התמונות המקוריות תוך הנחה (שגויה) ש-3 התמונות שעברו דרך הערוץ, התקבלו מתוך קונבולוציה ציקלית בין התמונות לתגובה להלם של הערוץ. כלומר, נניח כי:

$$y_1[n, m] = x_1[n, m] \circledast_{128, 128} h[n, m]$$

$$y_2[n, m] = x_2[n, m] \circledast_{128, 256} h[n, m]$$

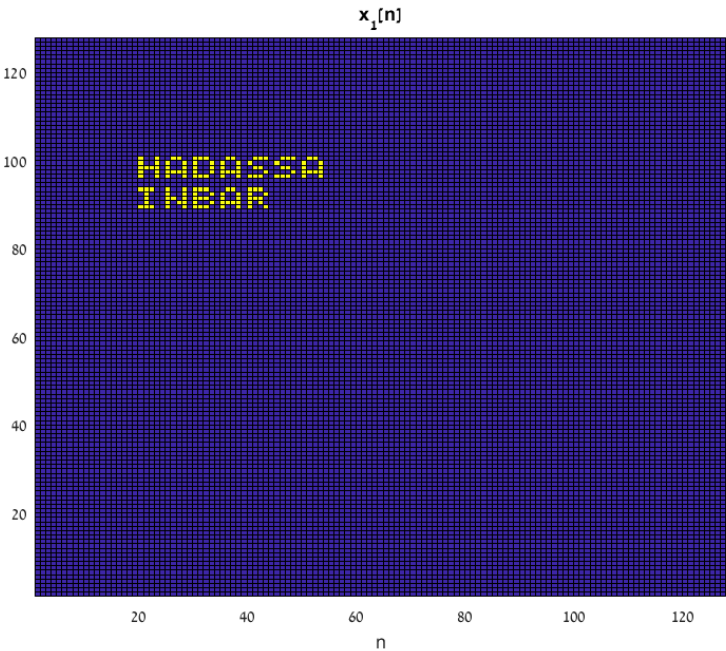
כאשר בפועל, התמונות התקבלו מתוך קונבולוציה לינארית בין התמונות לתגובה להלם, כלומר:

$$y_1[n, m] = x_1[n, m] * h[n, m]$$

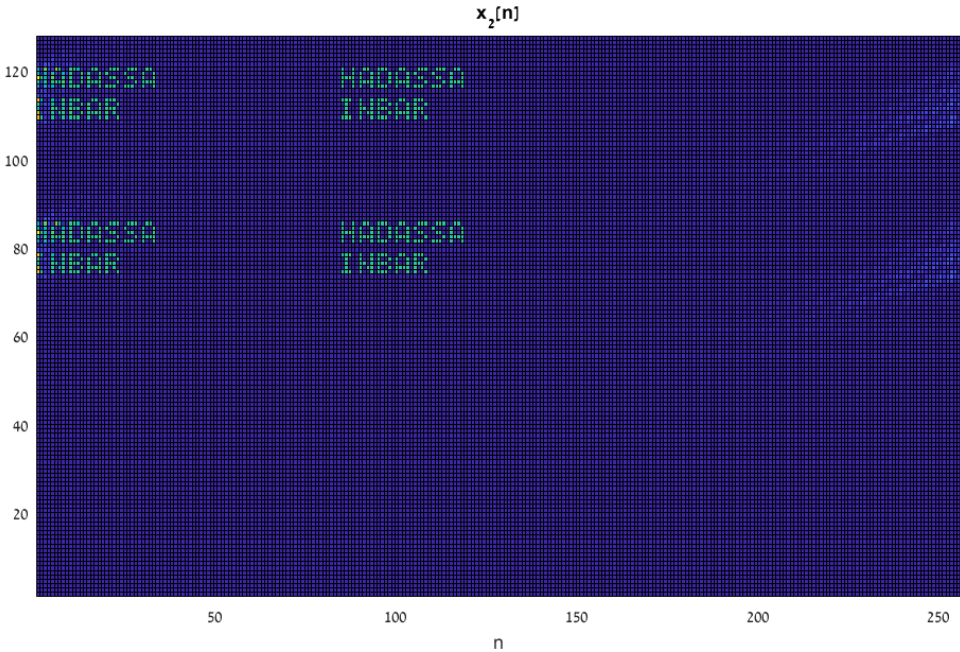
$$y_2[n, m] = x_2[n, m] * h[n, m]$$

להלן השחזורים שהתקבלו:

$x_1[n]$ :



$x_2[n]$ :



ח. כעת נרצה לעמוד על ההבדל בין אופני השחזור. נתון כי  $x_1$  נוצר מתמונה בסיסית שרופדה בלפחות 20 אפסים מכל כיוון. ניתן לראות כי קיבלנו עבור השחזור של  $x_1$  תמונה יחסית ברורה. ניתן להסיק כי כנראה אורך המחזור שבו התבצעה הקונבולוציה הציקלית היה מספיק ארוך (בזכות הריפוד), כך שלא היו דריכות של מחזור אחד בשני.

לפי הנלמד בקורס וכפי שכבר ראינו:

$$y[n] = x_1[n] * h[n] \quad , \quad \underset{\text{עבור } y[n]}{n} = 1:N + M - 1$$

$$\Rightarrow y_c[n] = x_1[n] \circledast_L h[n] = y[n] + y[n + L] \quad , \quad \underset{\text{עבור } y_c[n]}{n} = 1:L$$

כלומר התקבל כי  $L > N + M - 1$ , ולכן מקבלים:

$$y_c[n] = x_1[n] \circledast_L h[n] = y[n] + \underbrace{y[n + L]}_{=0} = y[n]$$

לגבי  $x_2$ : נתון כי זוהי תמונה ששוכפלה 4 פעמים. ניתן לראות בשחזור שאכן קיבלנו 4 שכפולים של השמות, אולם בחלק מהשכפולים ישנו טשטוש קל. ניתן להסביר זאת לאור העובדה שבאות המשוכפל אין הרבה אפסים בסוף (ביחס ל- $x_1$  למשל), כך שהקונבולוציה הציקלית עבורו כן כוללת דריכות, ומקבלים עיוות, כך שהקונבולוציה הציקלית לא מתלכדת לגמרי עם הקונבולוציה הלינארית.

ט. על מנת לשחזר את  $x_2$  בצורה טובה, ניתן לרפד את  $y_2$  בעוד אפסים, עד שנגיע למידה מספקת. כך, אורך המחזור של הקונבולוציה הציקלית יגדל, ולא נקבל דריכות. ככל שנרפד ביותר אפסים משני הכיוונים השחזור יהיה טוב יותר, עד שעבור ריפוד מסוים כבר יהיה שקול לשחזור באמצעות קונבולוציה לינארית.

הערה: הריפוד באפסים יגדיל את מימדי השחזור. אם נרצה שהמימדים יישארו זהים, נוכל לחתוך את תוצאת השחזור (שבוצעה באמצעות ריפוד) למימדים המקוריים. לחילופין, ניתן לקחת תת-מטריצה של  $y_2$  המכילה רק שכפול אחד, לרפד אותה במספיק אפסים (לא יידרשו הרבה), לשחזר, ולבסוף לשכפל שוב למטריצה הגדולה.

### 3. חלק שני – ייצור וניתוח אותות דיבור

הקלטנו אות דיבור  $x[n]$  בקצב דגימה ל  $f_s = 160,000 [Hz]$ , וחתכנו אותו לאורך של  $N = 2^{16}$ .  
א. ההספק הממוצע של האות:

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} (x[n])^2 = 0.0163 [W]$$

ב. הגדרנו הפרעה:

$$z[n] = 50\sqrt{P_x}(\cos(w_1 n) + \cos(w_2 n) + \cos(w_3 n)) , \quad n = 0:N-1$$

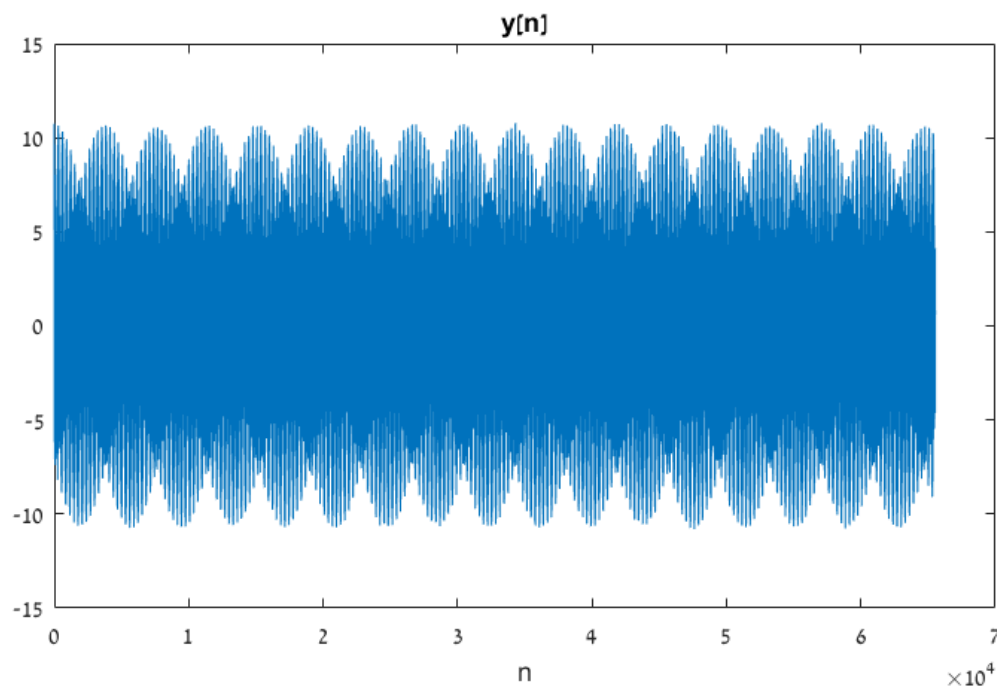
כאשר:

$$w_1 = 1.6 + 0.1 \cdot d_1 , w_2 = 1.6 + 0.1 \cdot d , w_3 = 3$$

והגדרנו את אות הכניסה  $y[n] = x[n] + z[n]$ .

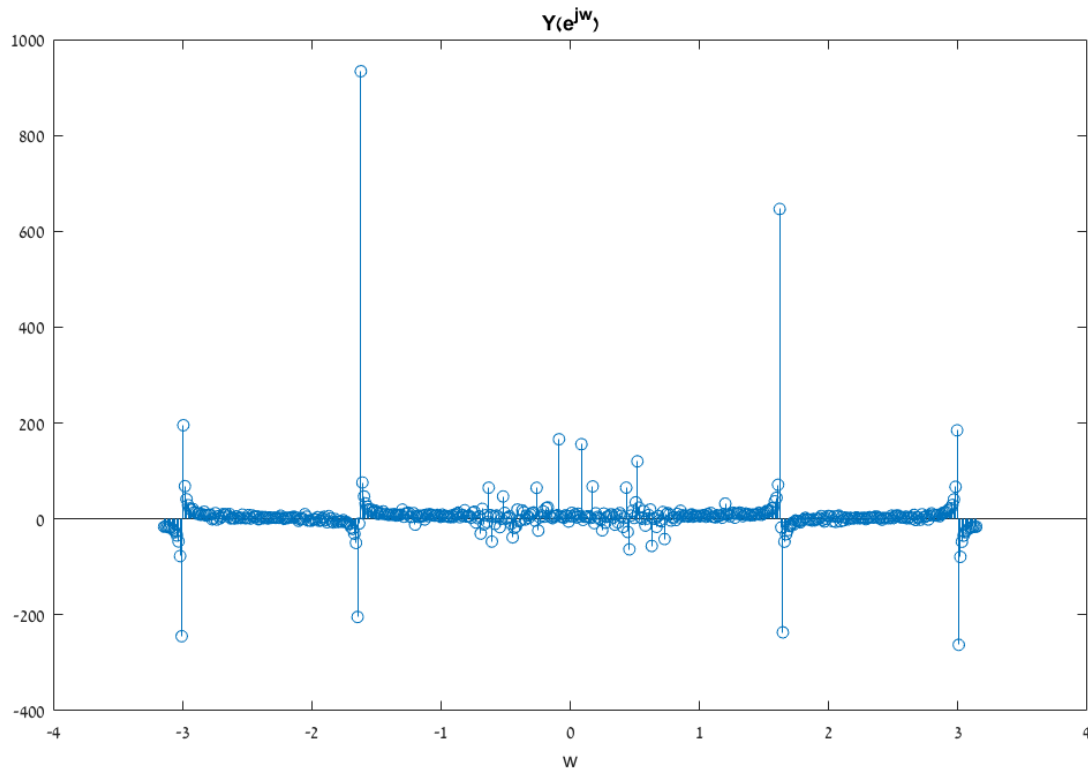
השמענו את האות  $y[n]$ . הצליל הדומיננטי ששומעים הוא הרעש (צפצוף חזק), כאשר האות  $x[n]$  מורגש בצורה חלשה מאוד. תוצאה זו תואמת את היחסים בין ההספקים שקיבלנו עבור שני האותות:  $P_x = 0.0163$ ,  $P_z = 61.247$ .

ג. להלן האות  $y[n]$ :



אכן, רואים בעיקר את ההפרעה (הרעש) הסינוסואידלי.

ד. להלן  $Y(e^{jw}) = DTFT\{y[n]\}$  בתדירים  $w = \left[ -\pi : \underbrace{\frac{2\pi}{128}}_{\text{גודל הקפיצה}} : \pi \right]$



ניתן לראות מהגרף כי תדרי ההפרעה הינם בעלי הספק גבוה מאוד ביחס לתדרי  $x[n]$ .

ה. האותות  $y_2[n], z_2[n]$  מתקבלים מדצימציה ביחס של 2 של האותות  $y[n], z[n]$ .

נחשב את  $Z_2(e^{jw}), z_2[n]$

$$z_2[n] = z[2n] = 50\sqrt{P_x}[\cos(2w_1n) + \cos(2w_2n) + \cos(2w_3n)]$$

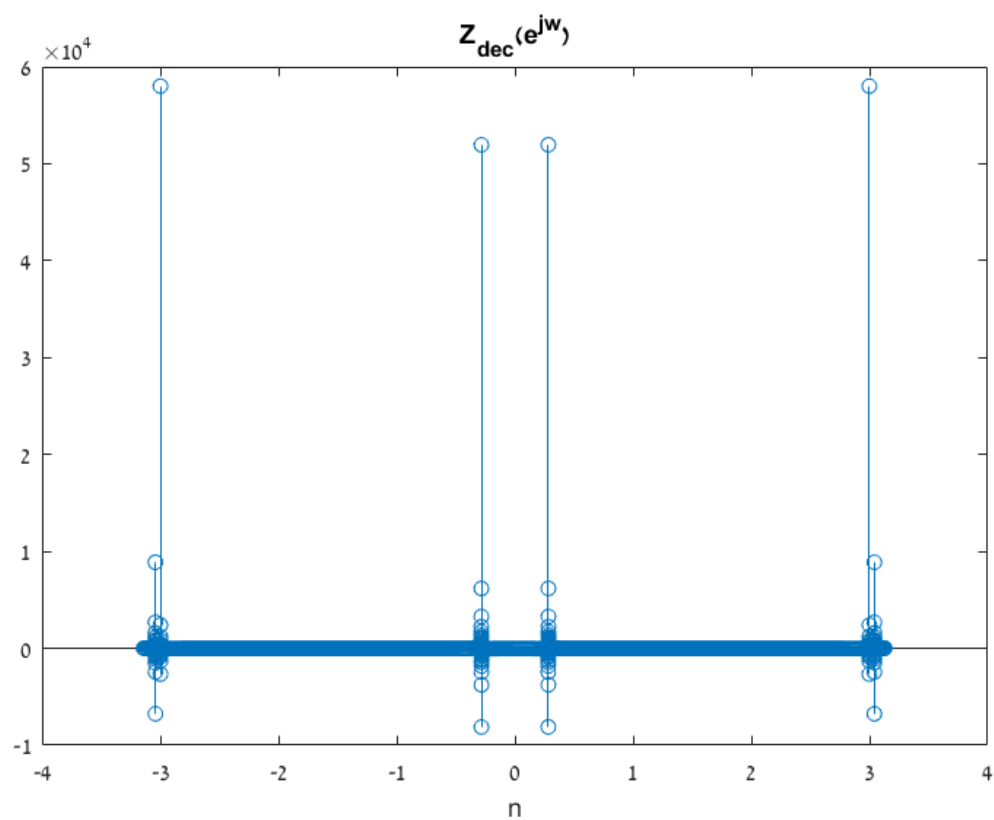
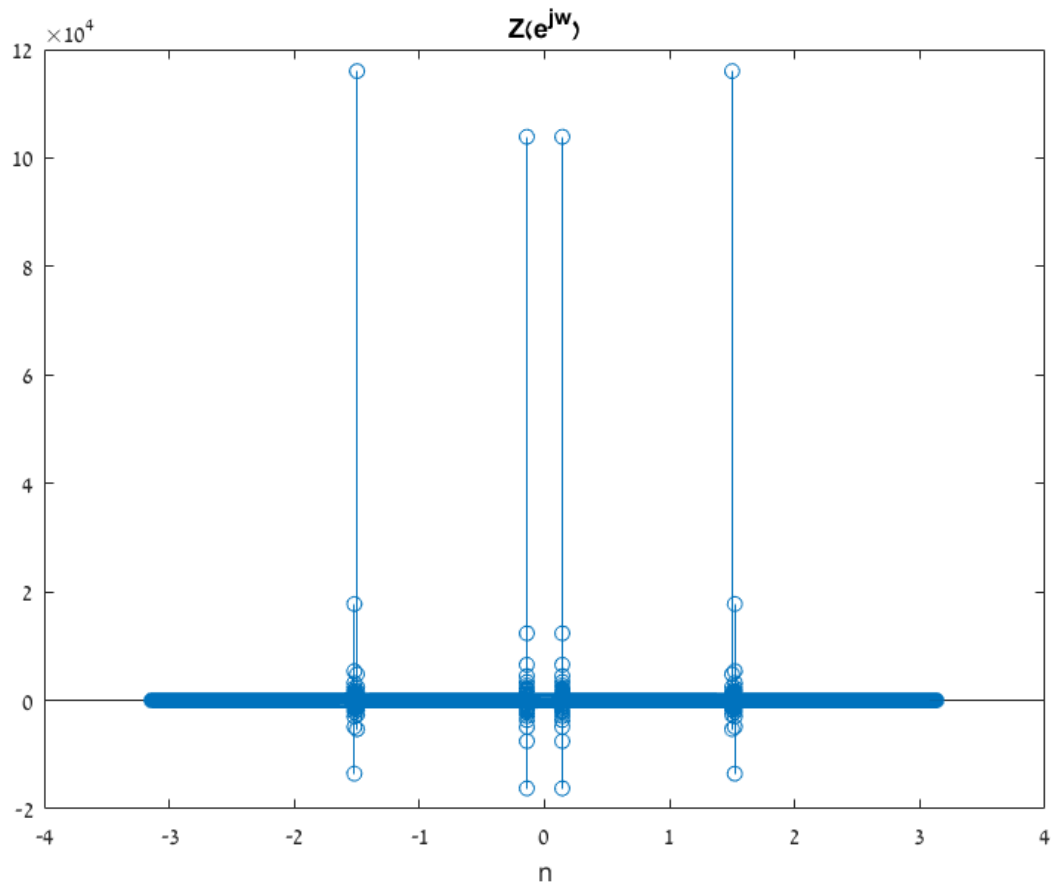
$$\Rightarrow Z_2(e^{jw}) = \frac{1}{2}Z(e^{j\frac{w}{2}})$$

$$Z(e^{jw}) = DTFT\{50\sqrt{P_x}(\cos(w_1n) + \cos(w_2n) + \cos(w_3n))\}$$

$$= 50\pi\sqrt{P_x}(\delta[w - w_1] + \delta[w + w_1] + \delta[w - w_2] + \delta[w + w_2] + \delta[w - w_3] + \delta[w + w_3])$$

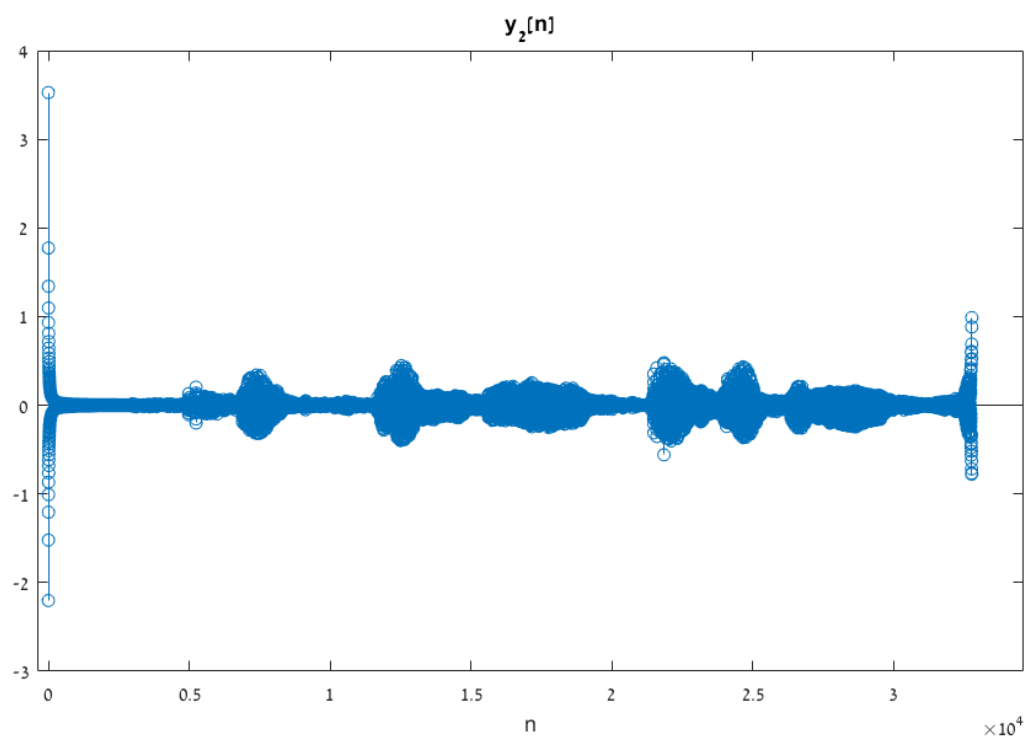
$$\Rightarrow Z_2(e^{jw}) = 25\pi\sqrt{P_x}(\delta[w - 2w_1] + \delta[w + 2w_1] + \delta[w - 2w_2] + \delta[w + 2w_2] + \delta[w - 2w_3] + \delta[w + 2w_3])$$

ניתן לראות את היחס בין  $Z(e^{jw})$  לבין  $Z_2(e^{jw})$



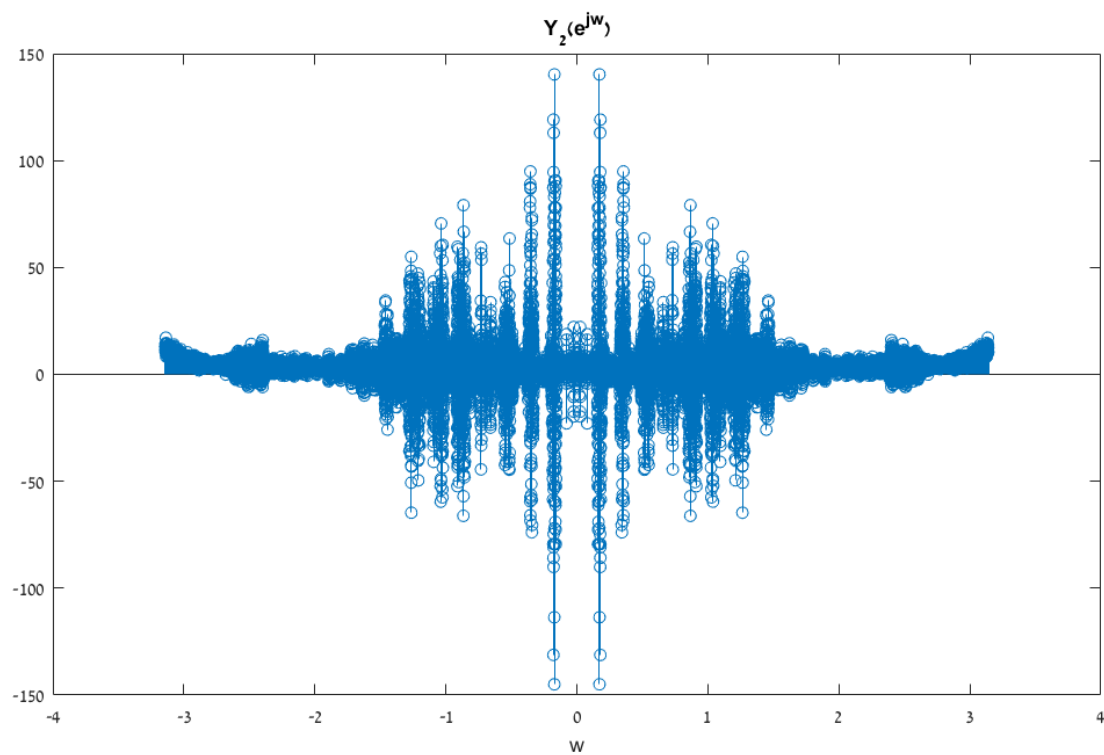
כמצופה, מבחינת התדרים והגבהים.

ו. להלן האות  $y_2[n]$ :



כעת האות דומה יותר לאות דיבור סטנדרטי.

להלן  $Y_2(e^{j\omega}) = DTFT\{y_2[n]\}$ :





כעת לא מבחינים כלל ברעש כלשהו ושומעים טוב את הדיבור, אולם ישנו עיוות כלשהו בצליל. ניתן להסביר זאת ע"י כך שהקול שלנו מורכב מתדרים שונים, וכשסיננו את התדרים הגבוהים מ- $\frac{\pi}{2}$ , כנראה מחקנו חלק מהטונים הטהורים שהרכיבו את הקול בהקלטה, ולכן התקבל עיוות.