

BAR-ILAN UNIVERSITY (RA)

School of Engineering

Ramat-Gan 52900, Israel



Tel: 03-5317721

engbi@mail.biu.ac.

אוניברסיטת בר-אילן
(ע"ר)

בית הספר להנדסה

רמת-גן 52900

עיבוד אותות ספרתי

תרגיל מטל"ב

סמסטר ב' תשע"א

דוקטור איציק ברגל

קובי כהן

אבי זנקו

אפרת טייטלבוים - 302609854

נעמה אברהמי - 200719359

D1 = 0.200719359 נעמה

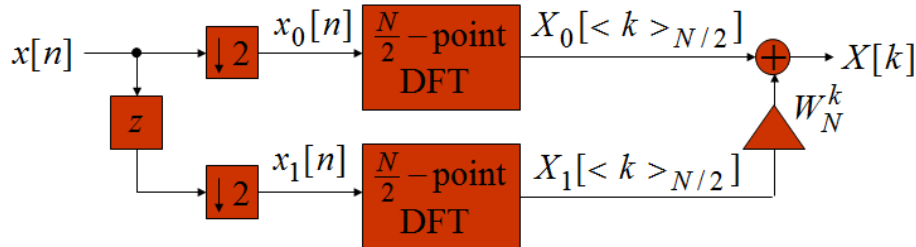
D2 = 0.302609854 אפרת

D = 0.503329213

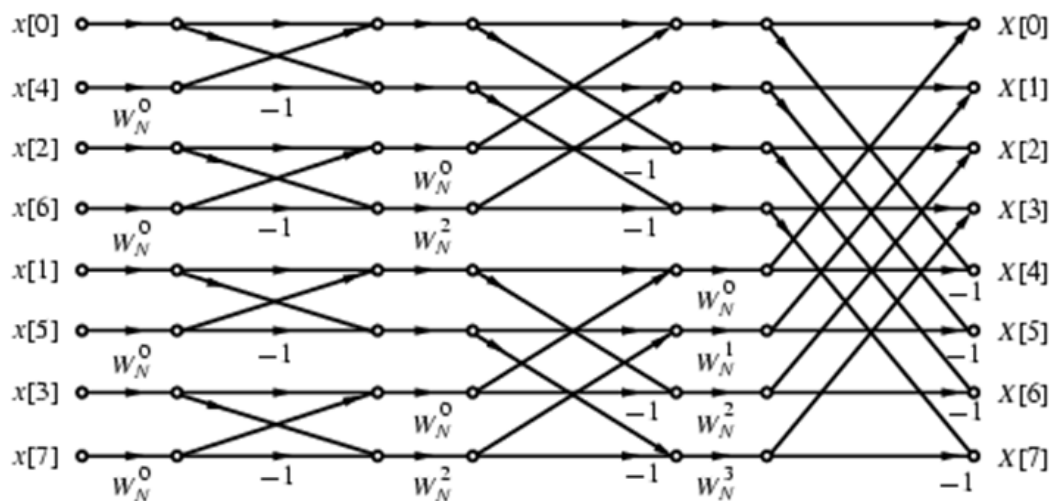
סעיף א -

דצימציה בזמן $D \bmod 0.1 = 0.003329213 < 0.05 \rightarrow$

שיגרה המממשת FFT ע"י דצימציה בזמן, בכדי לממש זאת נשתמש ב:



ובאלגוריתם של שיטת הפרפרים:



הרעיון של האלגוריתם הוא שבשביל לממש DFT של סדרה באורך N, ניתן להשתמש בפעמיים DFT של סדרה באורך N/2.

הפונקציה שכתבנו:

```
function [ y ] = EN_fft( x )    % function of FFT by decimation on time
len = length(x);              % length of vector
l = log2(len);
x_reverse = bitrevorder(x);    % reverse the bit order to use "butterfly" algorithm
y=zeros(1,len);               % add zeros
temp=zeros(1,len);            % temporary array
for j=1:len
    y(j)=x_reverse(j);        % save reverse of original vector
end

n=2;
Wn=exp(-2*pi*1i/len);

for k=1:1:l;
    for m=1:1:len              % multiply by 'Wn'.
        if ( (mod(m,n) <= (n/2)) && (mod(m,n)~=0) )
            temp(m)=y(m);      % for save the original vector
        end
    end
end
```

```

else
    y(m)=y(m) * (Wn^( (mod(m-1,2^(k-1)) ) * (len/2^k) ) );
end
end
for m=1:1:len % num of points calcula ted in spec. stage
    if ( (mod(m,n) <= (n/2)) && (mod(m,n)~=0) )
        y(m)=y(m)+y(m+2^(k-1)); % calculating point according to condition
    else
        y(m)=(-1)*y(m)+temp(m-2^(k-1)); % before adding other value - multiply by '-1'
    end
end
end
n=n*2; % update for next stage
end

```

הערה: הנחנו ש $N=2^V$

$$x[n] = \frac{1}{N} \left\{ DFT \left[X(K^-) \right] \right\} \quad \text{הפונקציה של inverse לפי מה שלמדנו אכן משתמשת בהתמרה:}$$

```

function [ y ] = EN_ifft( x ) % function of IFFT by decimation on time
len = length(x); % length of vector
l = log2(len);
x_reverse = bitrevorder(x); % reverse the bit order to use "butterfly" algorithm
y=zeros(1,len); % add zeros
temp=zeros(1,len); % temporary array
n=2;
Wn=exp(2*pi*1i/len);
for j=1:len
    y(j)=x_reverse(j); % save reverse of original vector
end
for k=1:l;
    for m=1:1:len % multiply by 'Wn'.
        if ( (mod(m,n) <= (n/2)) && (mod(m,n)~=0) )
            temp(m)=y(m); % for save the original vector
        else
            y(m)=y(m) * (Wn^( (mod(m-1,2^(k-1)) ) * (len/2^k) ) );
        end
    end
end
for m=1:1:len % num of points clculated in spec. stage
    if ( (mod(m,n) <= (n/2)) && (mod(m,n)~=0) )
        y(m)=y(m)+y(m+2^(k-1)); % calculating point according to condition
    else
        y(m)=(-1)*y(m)+temp(m-2^(k-1)); % before adding other value,multiply by '-1'
    end
end
end
n=n*2; % update for next stage
end
y=y/len;

```

האלגוריתם שמימשנו יעיל כי ראינו בהרצאה שהמימוש הרגיל של DFT ע"י הכפלת מטריצה בגודל $N \times N$, יוצר אלגוריתם בסיבוכיות של N^2 , לעומת המימוש באלגוריתם הפרפרים, שהסיבוכיות בו היא $N \log(N)$. כי: כל פעם אנחנו מחלקים את ההתמרה ל 2 התמרות בגודל של חצי

מההתמרה המקורית, ובכל שלב אנחנו מחלקים ל-2. לכן זה יוצא מספר פעולות יותר קטן (בסדר גודל).
ואותו כנ"ל לגבי ההתמרה ההפוכה (היא הרי משתמשת בהתמרה ישירה).

סעיף ב:

הקלטנו אות דיבור. האמת – הקלטנו את "את הכי טובה בעולם" מתוך יסמין/הפיל הכחול. תודה ליוטיוב.
רשמנו במטלב:

```
Fs = 16000; % Fs = 16000 Samples/sec
N = 2^16; % the num of Samples
x = wavrecord(N,Fs); % record
```

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

וחישבנו את הספק לפי:

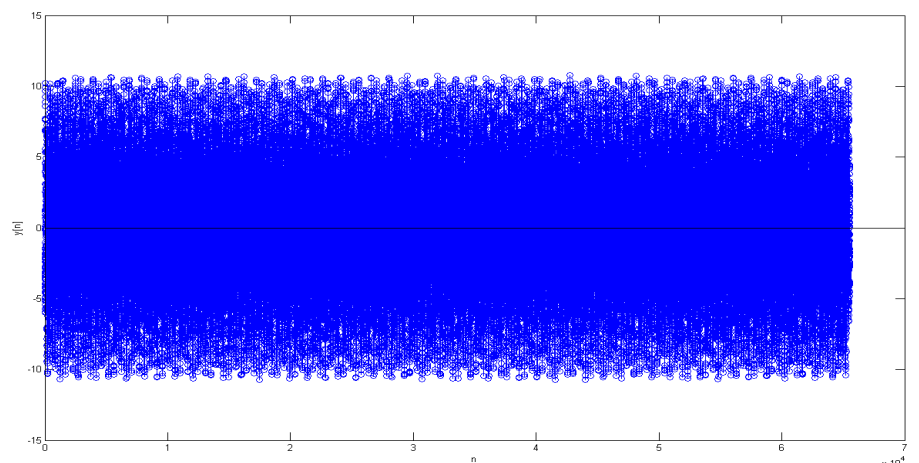
```
N = 2^16;
Px = 0; % power by: Px = 1/N*sig(x[n])^2
for j=1:N
    Px = Px+(x(j))^2;
end
Px = Px*(1/N); % = 0.0081
```

כדי להוסיף את הרעש, כתבנו:

```
d1=0.200719359;
d=0.503329213;
a1=50*sqrt(Px); % = 4.5084
a2=50*sqrt(Px); % = 4.5084
a3=50*sqrt(Px); % = 4.5084
w1=1.6+0.1*d1; % = 1.6201
w2=1.6+0.1*d; % = 1.6503
w3=3;

z=zeros(1,N); % the noise
for j=1:N
    z(j) = a1*cos(w1*j)+a2*cos(w2*j)+a3*cos(w3*j);
end
y=zeros(1,N);
for j=1:N % noise + signal
    y(j) = x(j)+z(j);
end
n=[0:1:(2^16)-1];
stem(n,y) % plot
xlabel('n')
ylabel('y[n]')
```

זהו הגרף שקיבלנו של הסיגנל ביחד עם הרעש.
2¹⁶ דגימות.
תכל"ס, כשהשמענו את זה עם wavplay, לא שומעים בכלל את השיר וזה נשמע כמו רעש אחד גדול.

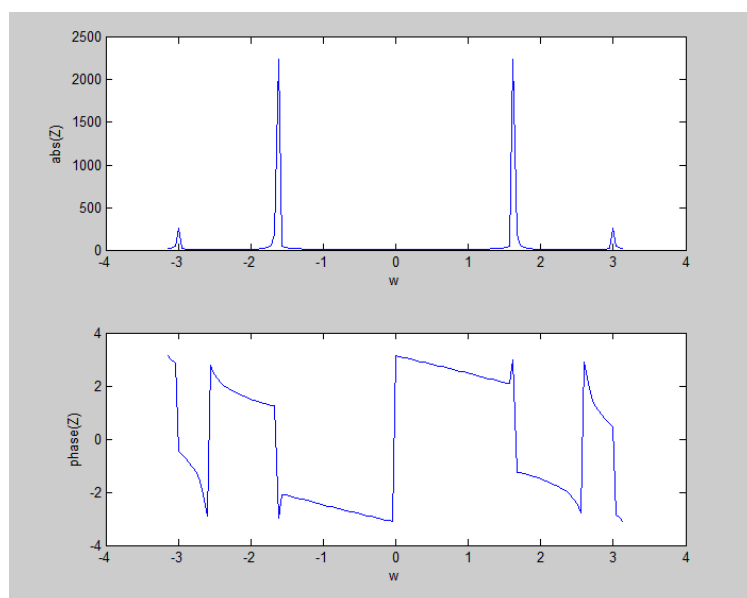


סעיף ג:

בסעיף זה התבקשנו לחשב את ההתמרת פורייה של האות, חישבנו על פי הגדרה של התמרת פורייה:
 אנחנו לא נרוץ על הסכום מינוס אינסוף עד אינסוף, אלא על 2^{16} דגימות, ומתוכם ניקח אחת מכל $N/128$ דגימות, ונקבל 129 דגימות בין 0 ל- 2π שמייצגות לנו את האות של הרעש (הזזנו להיות בין מינוס π ל- π - ניתן לעשות זאת בגלל המחזוריות).
 הפונקציה שכתבנו:

```
w = linspace(-pi,pi,129); % frequency range
for k=1:1:129
    temp=0;
    for n=1:1:2^16
        temp=temp+z(n)*exp(-1i*w(k)*n); % calculate the transform
    end
    Z(k)=temp;
end
subplot(2,1,1) % plot amplitude
plot(w,abs(Z))
xlabel('w')
ylabel('abs(Z)')
subplot(2,1,2) % plot phase
plot(w,angle(Z))
xlabel('w')
ylabel('phase(Z)')
```

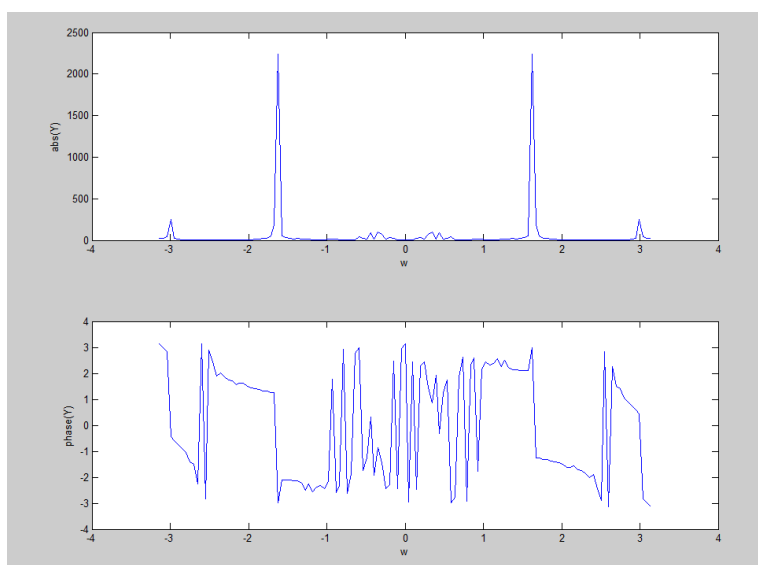
→ הגרף שקיבלנו, ניתן לראות את הדלתות הקטנות שיש ב $|w|=3$,
 ועוד דלתא גדולה יותר מסביב לתדרים w_1, w_2 .



סעיף ד:

הרצנו בדיוק את אותה פונקציה כמו בסעיף הקודם, רק הפעם כל האות של הסיגנל + הרעש, וניתן לראות בגרף שקיבלנו, את החוזק של הרעש לעומת האות שלנו, וזה הסבר לזה שהוא כל כך דומיננטי, שכאשר משימעים את האות המשולב שומעים רק רעש.

(מסביב לתדר אפס ניתן לראות את האות של השיר).



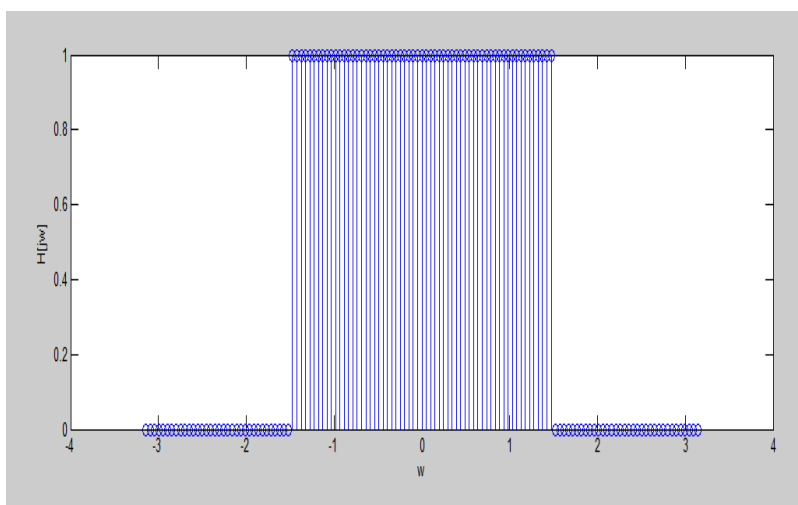
סעיף ה:

מסנן LPF אידיאלי עם תדר קטעון $\omega_c = 1.5$ [rad/sec] נראה כך:

במטלב:

```
function [ H ] = LPF()

Wc = 1.5;
w = linspace(-pi,pi,129);
H = zeros(1,129);
for j=1:1:129
    if(abs(w(j))<Wc)
        H(j)=1;
    end
end
stem(w,H);
xlabel('w');
ylabel('H[jw]');
end
```



תגובת התדר כידוע מחזורית 2π , עפ"י טבלאות ידועה גם התגובה להלם של מסנן זה:

$$h_i[n] = \frac{\omega_c}{\pi} \text{sinc}\left(\frac{\omega_c n}{\pi}\right)$$

סעיף ו:

נתבקשנו לתכנן מסנן מעשי יותר ע"י התגובה להלם,

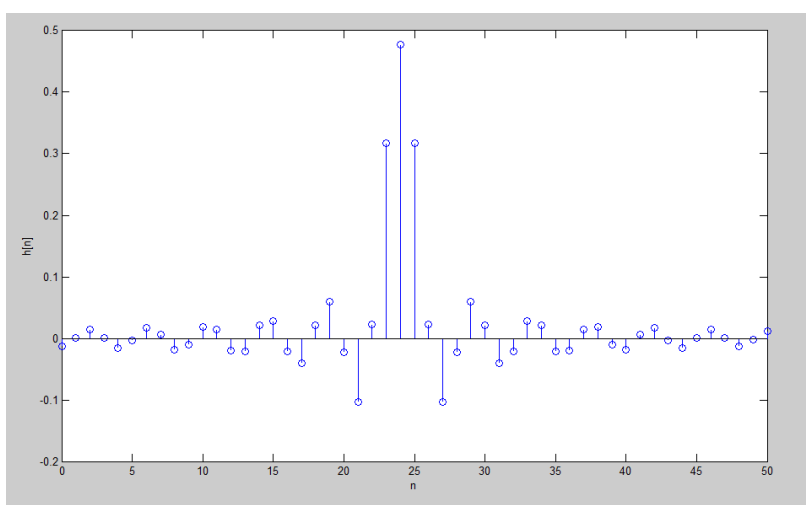
המסנן הבא הוא גם סופי () וגם סיבתי – ולכן גם מעשי הרבה יותר:

$$h_p[n] = \begin{cases} h_i[n-25] & 0 \leq n \leq 50 \\ 0 & \text{o.w.} \end{cases}$$

ובמטלב:

```
function hp()
for n=1:1:51
    if( n~=26)
        hp(n) = sin(1.5*(n-26))/(pi*(n-26));
    end
end
hp(26) = 1.5/pi;

n=[0:1:50];
stem(n, hp);
xlabel('n');
ylabel('h[n]');
end
```



$$h_p[n] = \begin{cases} h_i[n-25] & 0 \leq n \leq 50 \\ 0 & \text{o.w.} \end{cases} = [h_i[n] * \delta(n-25)] \cdot \text{Window}(0,50) \\ = [h_i[n] \cdot \text{Window}(-25,25)] * \delta(n-25)$$

$$H_p(e^{j\omega}) = \left[H_i(e^{j\omega}) * \frac{\sin\left(\omega\left(25 + \frac{1}{2}\right)\right)}{\sin\frac{\omega}{2}} \right] \cdot e^{-j\omega \cdot 25} \cdot \frac{1}{2\pi}$$

$$H_p(e^{j\omega}) = \begin{cases} \frac{\sin\left(\omega\left(25 + \frac{1}{2}\right)\right)}{\sin\frac{\omega}{2}} \cdot e^{-j\omega \cdot 25} \cdot \frac{1}{2\pi} & |\omega| < 1.5 \\ 0 & \text{o.w.} \end{cases}$$

$h_p[n]$ זאת ההגדרה של האות בזמן.

הוא מורכב מ $h[n-25]$ שמכפילים

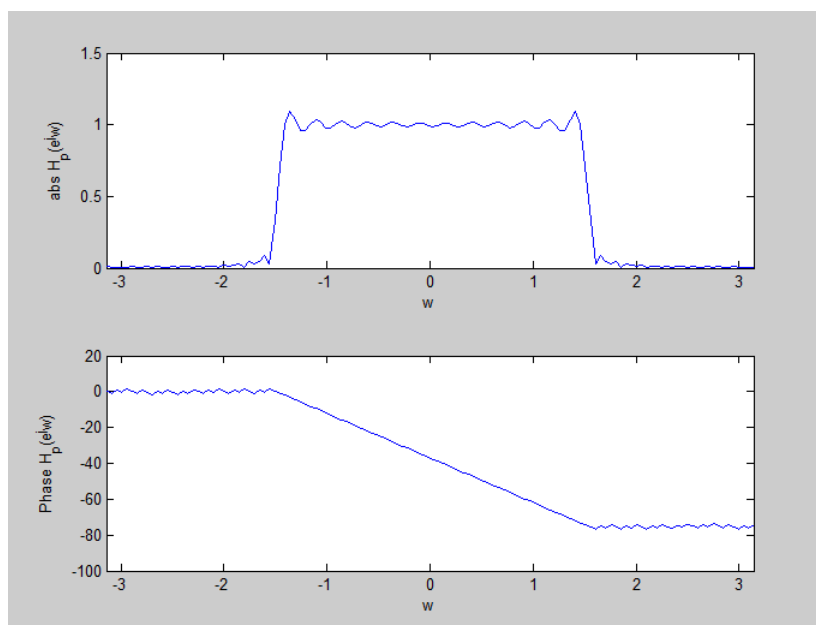
אותו בחלון מ-50 עד 50, כפול $h[n]$ (מדרגה) כלומר, כל העסק יהיה עכשיו מ0 עד חמישים.

וכשנעשה לזה התמרה, זה התמרה של התגובה להלם מוזזת (הכפלה באקספוננט) עם קונבולוציה בהתמרה של החלון.

```
w = linspace(-pi,pi,128);
for n=1:1:51
    if( n~=26)
        hp(n) = sin(1.5*(n-26))/(pi*(n-26));
    end
end
hp(26) = 1.5/pi; % for not divide zero
Hp=our_dtft(hp); % frequency response of hp

subplot(2, 1, 1) % plot
plot(w, abs(Hp)); % Magnitude
xlabel('w');
ylabel('abs Hp(e^jw)');
subplot(2, 1, 2)
plot(w, phase(Hp)); % Phase
xlabel('w');
ylabel('Phase Hp(e^jw)');
```

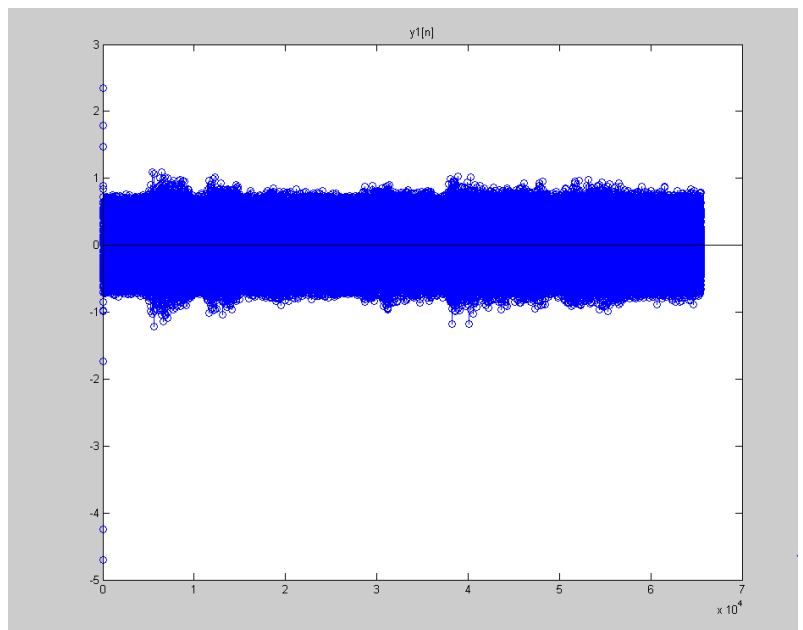
תגובת התדר היא כמובן מחזורית 2π , ניתן לראות
בבירור שהמסנן הוא לא אידיאלי.



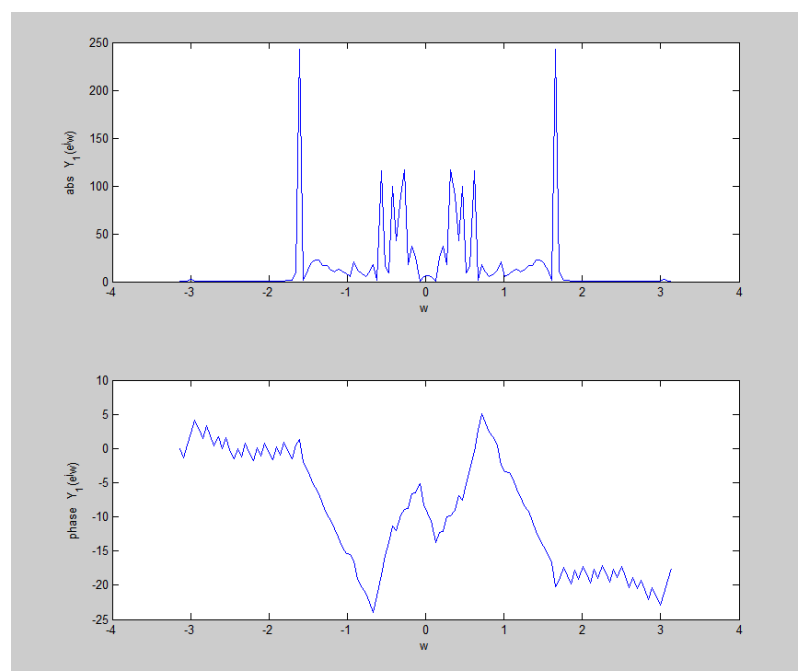
סעיף ח:

כתבנו פונקציה שתחשב את $y_1[n]$ ע"י מעבר של $y[n]$ בתוך המסנן $h_p[n]$, כמו שרואים גם בגרפים, אכן הרעש חלש יותר. כשהקשבנו ל y_1 הצלחנו לשמוע את האות המקורי, אבל עדיין נשמע רעש (כצפוף חזק ברקע).

```
Hp=fft(hp, N); % HP dtft(hp)
Y=fft(y); % Y dtft(hp)
Y1_long=Hp.*Y; % long of N points
y1=ifft(Y1_long); % y1 the signal after pass in filter
Y1=our_dtft(y1); % Y1 dtft(y1)
%plot y1
figure(2)
stem(y1);
title('y1[n]')
wavplay(y1,16000); %play
subplot(2, 1, 1) % plot
plot(w, abs(Y1))
xlabel('w');
ylabel('abs Y_1(e^jw)');
subplot(2, 1,2)
plot(w, phase(Y1))
xlabel('w');
ylabel('phase Y_1(e^jw)');
```



כפי שניתן לראות בתגובת התדר שקיבלנו, המסנן החליש לחלוטין את הרעש בתדר $w_3 = 3$, אך בשאר התדרים הרעשים הוחלשו ביחס ל y אך עדיין ניתן לראותם וכמובן – לשמוע.

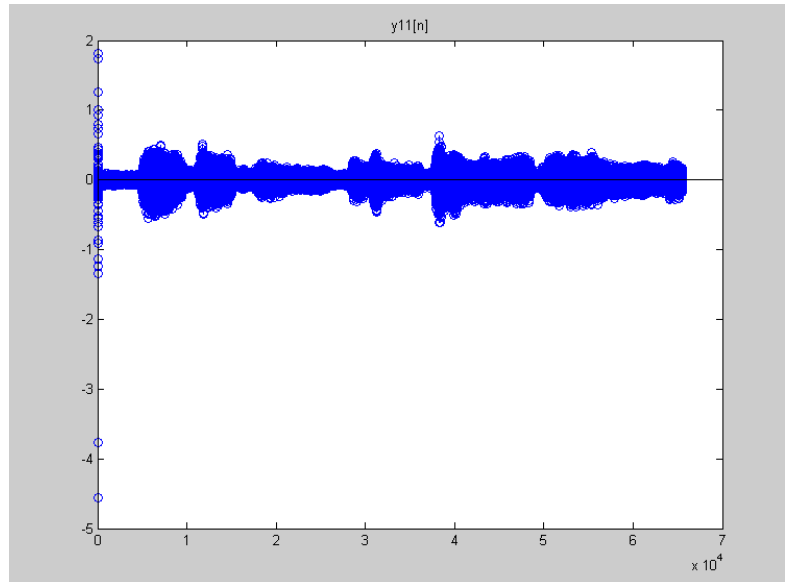


סעיף ט:

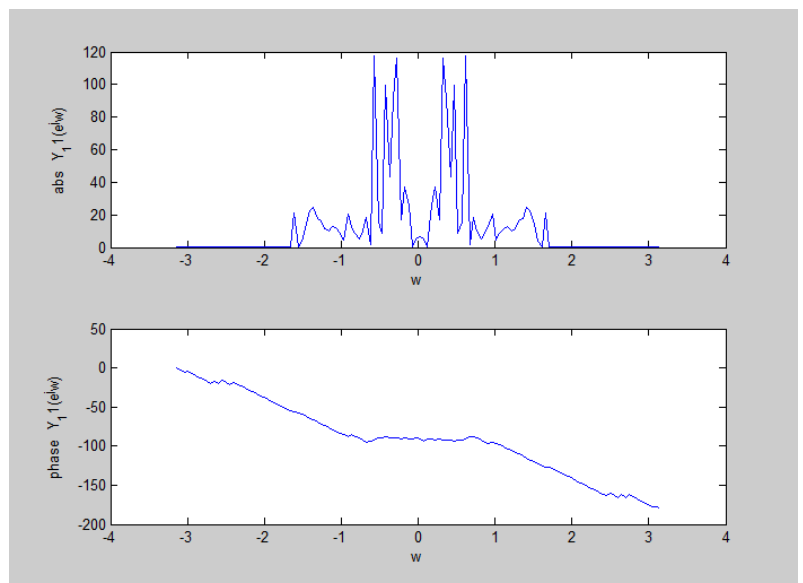
העברנו שוב את האות y_1 במסנן $h_p[n]$ וקיבלנו את האות $y_{11}[n]$. זה בהחלט שיפר את הביצוע של מה שחיפשנו. האות שהתקבל דומה לאות המקורי בצורה ניכרת, אף שהקשבנו לו שמענו כי הרעש הוחלש כמעט לגמרי, וניתן לשמוע את האות המקורי כמעט ללא הפרעה. אבל הרעש לא נעלם לגמרי. אולי זה בגלל שהמסנן הוא לא אידיאלי, התגובה להלם שלו סופית, ולכן לא קיבלנו הנחתה מושלמת של הרעש:

```
Y11_long=Y1_long.*Hp;           % Freq Multiplication length N
y11=ifft(Y11_long);             % y11 the signal after pass in filter in sec time
Y11=our_dtft(y11);              % Y11 (y11)

stem(y11);                      % plot
wavplay(y11,16000);             % play y11
title('y11[n]')
subplot(2, 1, 1)                %plot Y11
plot(w, abs(Y11))
xlabel('w');
ylabel('abs Y_11(e^jw)');
subplot(2, 1, 2)
plot(w, phase(Y11))
xlabel('w');
ylabel('phase Y_11(e^jw)');
```



בהתחלה של האות עדיין שמענו רעשים, וניתן לראות בגרף שקיבלנו עבור $y_{11}[n]$ את הרעשים (סמוך לקצה השמאלי של הגרף) שעדיין לא הופרדו מהדגימות בצורה אידיאלית.



סעיף י:

ביטוי מתמטי שקול למסנן מסעיף ח': המסנן השקול הוא כמו לעבור פעמיים קונבולוציה עם $h_p[n]$ ולכן בתדר זה יהיה מכפלה:

$$Y_{11}(e^{j\omega}) = Y(e^{j\omega}) \cdot [H_p(e^{j\omega})]^2$$

$$H_{eq}(e^{j\omega}) = [H_p(e^{j\omega})]^2 = \begin{cases} \left[\frac{\sin\left(\omega\left(25 + \frac{1}{2}\right)\right)}{\sin\frac{\omega}{2}} \cdot e^{-j\omega \cdot 25} \cdot \frac{1}{2\pi} \right]^2 & |\omega| < 1.5 \\ 0 & o.w. \end{cases}$$

סעיף י"א:

ע"פי הגדרה, בכדי לקבל אות מורד בקצב של 2, בזמן - נציב 2n במקום n:

$$Z_2[n] = Z[2*n] = a_1 \cos(w_1 * 2*n) + a_2 \cos(w_2 * 2*n) + a_3 \cos(w_3 * n * 2)$$

```
y2=y(1:2:N); %define decimated signals: M=2
z2=z(1:2:N);
```

$$Z_2(e^{j\omega}) = \frac{1}{M} \sum_{m=0}^{M-1} Z(e^{j(\frac{\omega-2\pi m}{M})}) = \frac{1}{2} \left(Z(e^{j\frac{\omega}{2}}) + Z(e^{j\frac{\omega-2\pi}{2}}) \right)$$

בתדר, בגלל שאנחנו מורידות את הקצב ב2 נקבל עוד שכפול כל π :

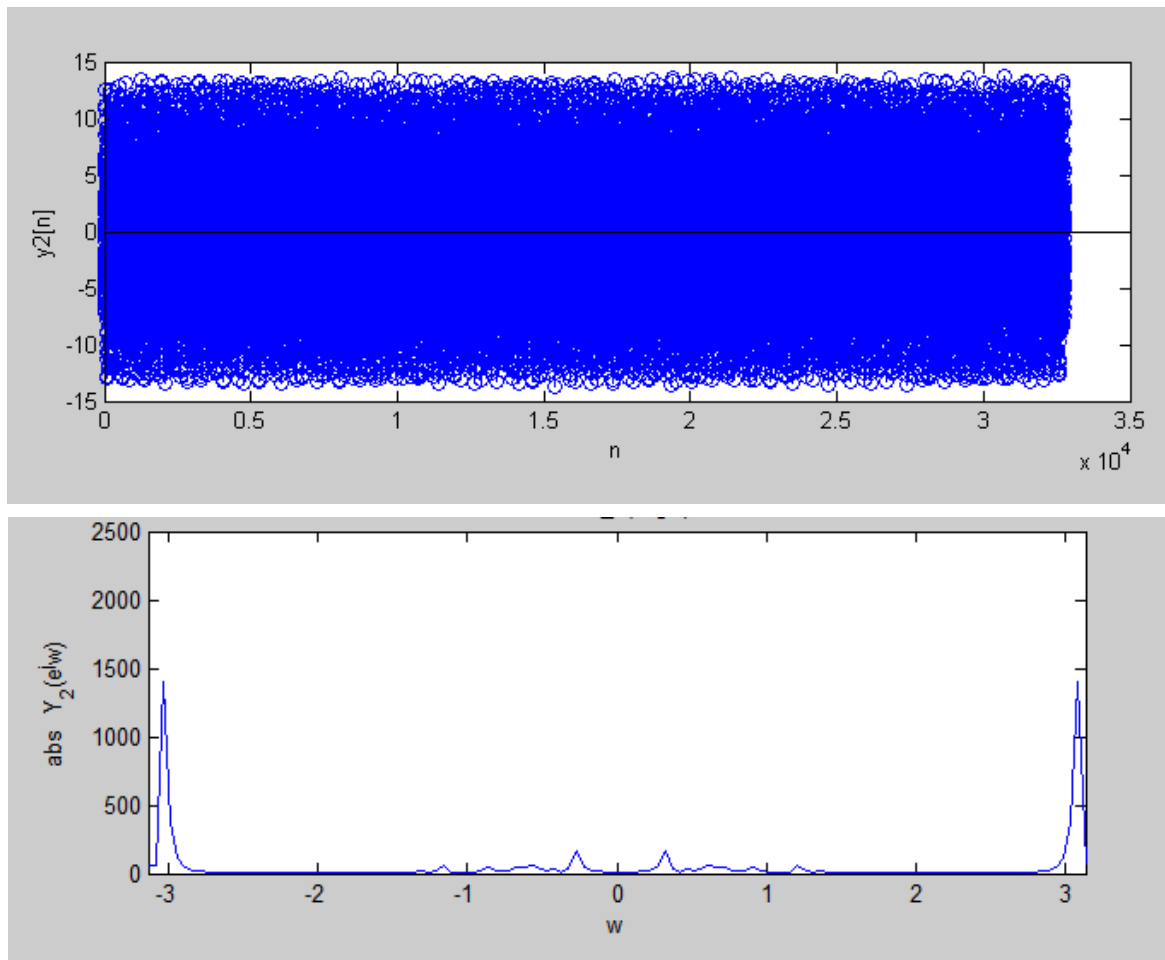
סעיף י"ב:

ציירנו את האות $y_2[n]$ ואת ההתמרת פורייה שלו $Y(e^{jw})$:

```
y2=zeros(1,N/2); % signal + noise
for j=1:1:N/2
    y2(j) = x(2*j)+z(2*j);
end
n=0:1:((2^16)-1)/2; % plot
stem(n,y2);
xlabel('n');
ylabel('y2[n]');
wavplay(y2, Fs/2);
```

גם לאחר הורדת קצב, עדיין לא ניתן להבדיל בין האות המקורי לרעש.

גם כאשר ניסינו לשמוע את האות לא הצלחנו להבדיל, שמענו רק צפצוף חזק.



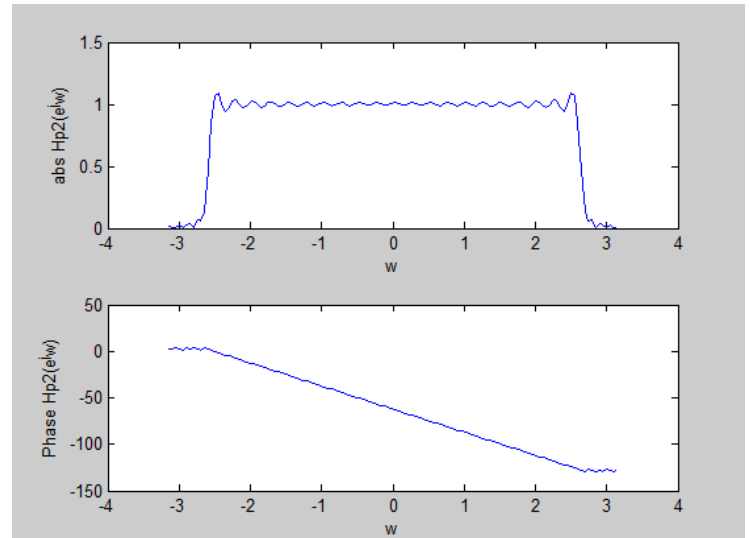
ניתן לראות שהאמפליטודה של האות של השיר מאוד חלש יחסית לרעש, והרעש חזק. ולכן עדיין מה ששמענו היה רעש.

סעיף י"ג:

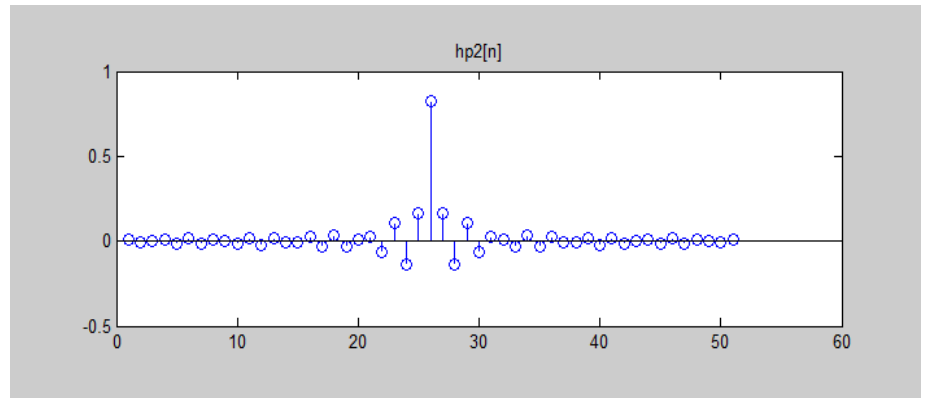
$$H_i(e^{j\omega}) = \begin{cases} 1 & |\omega| < 1.5 \\ 0 & o.w. \end{cases}$$

תכננו LPF מעשי באורך סופי - 51, הביטוי המתמטי (כמו שראינו בסעיף ה') האידיאלי הוא:
נבחר בתדר קטעון $\omega_c = 2.6$ [rad/sec], כי מעל לתדר הזה האות של השיר כמעט ולא נמצא
ולעומת זה לרעש יש שם נוכחות גבוהה. לפי הדרישות נעביר את האות מורד הקצב פעמיים במסנן המעשי, ונראה מה יהיו התוצאות:

```
for n=1:1:51
    if ( n~=26) % Wc=2.6
        hp2(n) = sin(2.6*(n-26))/(pi*(n-26));
    end
end
hp2(26) = 2.6/pi;
Hp2=our_dtft(hp2); % frequency response of hp2
```



→ הגרפים של המסנן שקיבלנו.
נעביר במסנן זה את האות $y_2[n]$ פעם אחת ופעם
שניה



$$Y_{2,lpf}(e^{j\omega}) = Y_2(e^{j\omega}) \cdot [H_p(e^{j\omega})]$$

המוצא עבור מעבר ראשון במסנן:

$$Y_{2,lpf}(e^{j\omega}) = Y_2(e^{j\omega}) \cdot [H_p(e^{j\omega})]^2$$

והמוצא עבור מעבר שני במסנן:

הרצנו את קטע הקוד הבא, וקיבלנו גרפים בהתאם:

```
Y21_long=fft(hp2, N/2+50).*fft(y2,N/2+50);
y21=ifft(Y21_long);
Y21=our_dtft(y21);
stem(y21);
title('y21[n]');
wavplay(y21,8000);
subplot(2, 1, 1)
plot(w, abs(Y21))
xlabel('w');
ylabel('abs Y_21(e^jw)');
subplot(2, 1, 2)
plot(w, phase(Y21))
xlabel('w');
```

```
% Y21_long=Y*HP2
% y21 signal after first pass in hp2
% Y21 dtft(y21)
% plot y21

% play y21

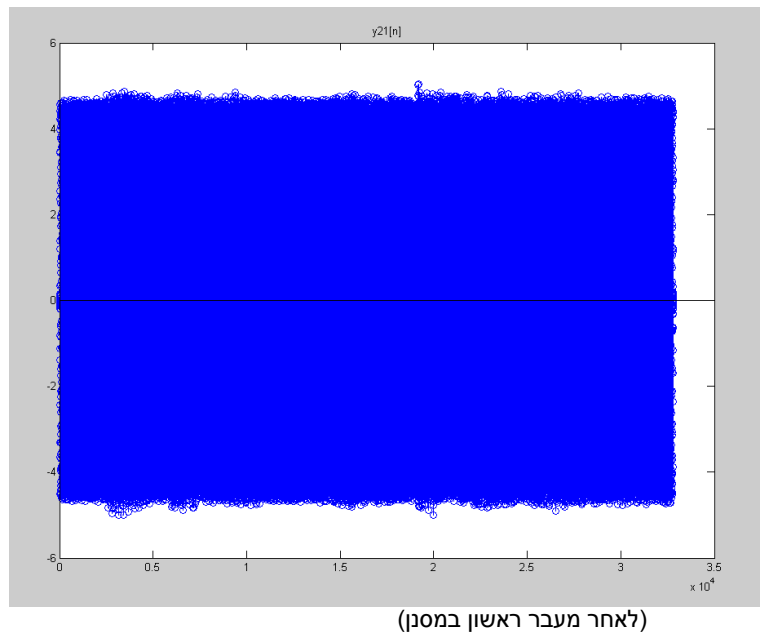
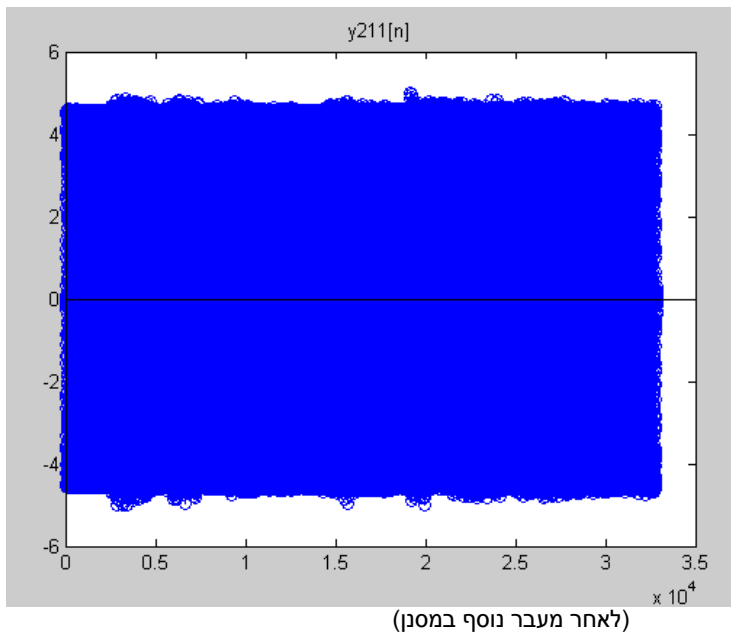
% plot Y21
```

```

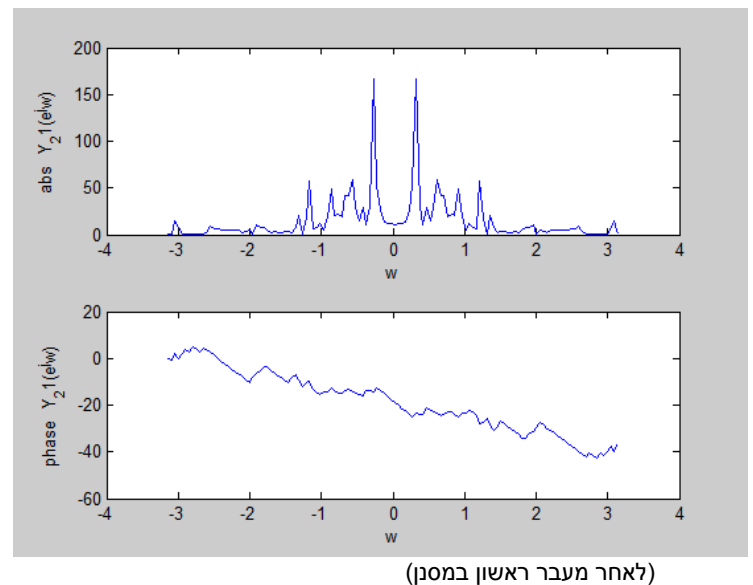
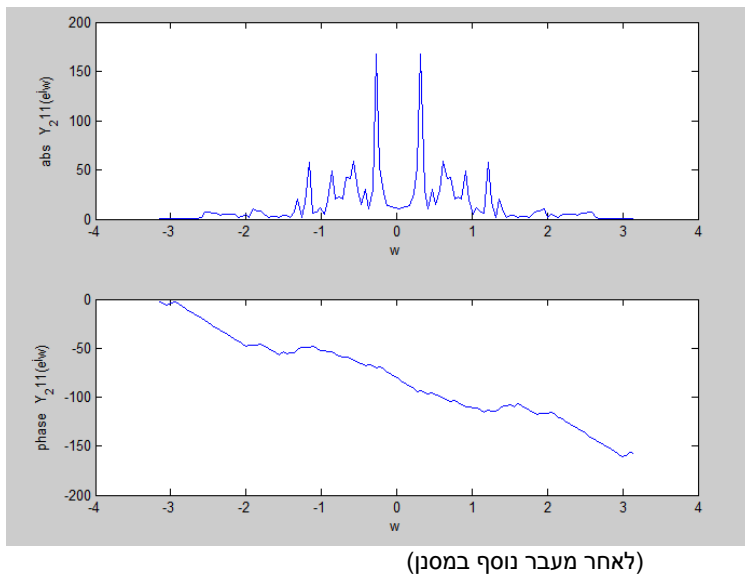
ylabel('phase Y_21(e^jw)');
Y211_long=fft(hp2,N/2+100).*fft(y21,N/2+100); % Freq Multiplication length N/2+100
y211=ifft(Y211_long); % y211 the wanted signal
Y211=our_dtft(y211); % Y211 dtft(y211)
stem(y211); % plot y211
wavplay(y211,8000); % play y211
title('y211[n]')
subplot(2, 1, 1) % plot Y211
plot(w, abs(Y211))
xlabel('w');
ylabel('abs Y_211(e^jw)');
subplot(2, 1, 2)
plot(w, phase(Y211))
xlabel('w');
ylabel('phase Y_211(e^jw)');

```

בציר הזמן לא ניתן לראות הבדל כלשהו בגרפים לאחר מעבר נוסף במסנן:



לעומת זאת, בציר התדר ישנו הבדל, הרעש בתדר הגבוה נקטם:



למרות ההבדל הזה, איכות השמיעה יחסית לרעש היא כמעט אותו דבר בשני המקרים. אפשר להבחין בשיר, אבל באיכות לא טובה.
זה נובע מכך שכעת גם הרעש יורד בקצב ביחס של 2, ולכן מסנן LPF לא מסנן לנו את תדרי הרעש שנמצאים סמוך לאפס.

סעיף י"ד:

ננסה להפריד טוב יותר את הדיבור מהרעש, נתכנן מסנן BPF מעשי באורך 251.

בחרנו במסנן מסוג BPF משום שיש לנו 'דלתות' גם בתדר קרוב ל3, וגם בתדרים נמוכים, ולכן נבחר את ה-BP להיות בין התדרים 2.6 לבין 0.4 כדי לממש אותו, נקח פונקצית חלון עם אותו רוחב: $1.1 = 2.6 - 0.4$ סביב 0,

$$H_{i-LPF}(e^{j\omega}) = \begin{cases} 1 & |\omega| < 1.1 \\ 0 & \text{o.w} \end{cases}$$

$$h_{i_lpf}[n] = \sin(1.1n)/(\pi \cdot n)$$

כעת נכפיל את המסנן האידיאלי שלנו בחלון (תגובה סופית להלם):

$$h_{p-LPF}[n] = \begin{cases} h_{i-LPF}[n-125] & 0 \leq n \leq 250 \\ 0 & \text{o.w.} \end{cases} = [h_{i-LPF}[n] \cdot \text{Window}(0,250)] * \delta(n-125)$$

ונזיז אותה לתדר הרצוי ($w=1.1+0.4=1.5$).

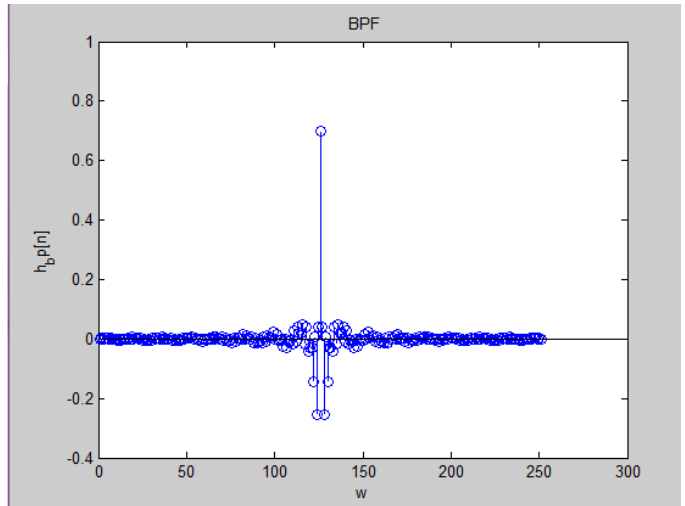
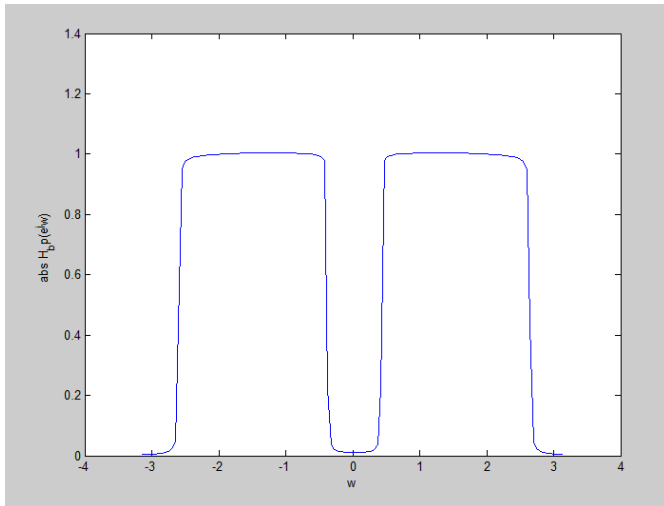
נעשה זאת ע"י הכפלה ב $\cos$ שהרי בתדר זה קונבולוציה עם דלתות. (הקוסינוס יזיז את ה-LPF לתדר ביניים וככה נקבל BPF):

$$h_{p_bpf}[n] = h_{p_lpf}[n] \cdot \cos(1.5n) = [\text{conv}[h_{i_lpf}[n] \cdot \text{window}(0,250)] * \delta(n-125)] \cdot \cos(1.5n)$$

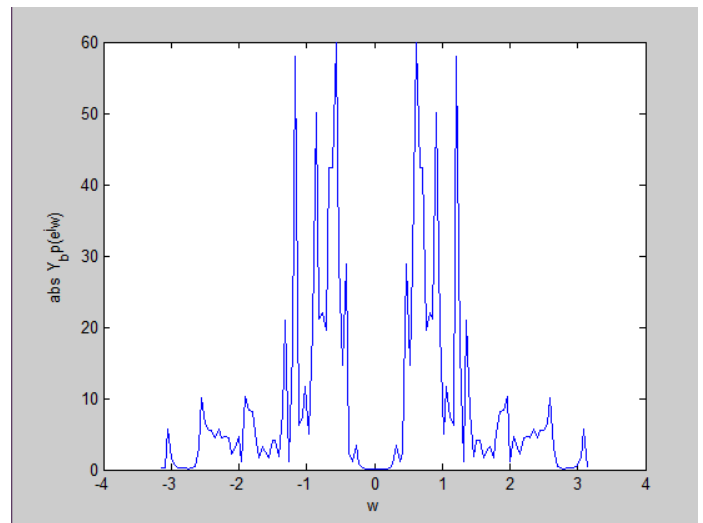
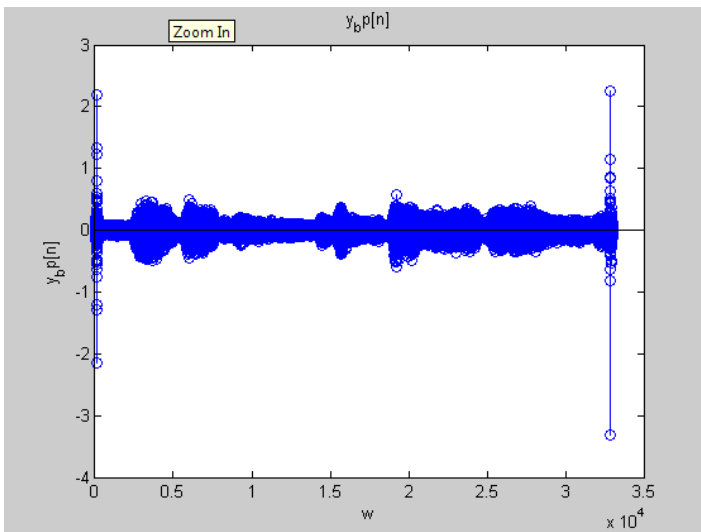
```
h_bp=zeros(1,251); % Wh the high frequency = 2.6, Wl the low frequency = 0.4
for n=1:251 % BPF- according to our calculation
    h_bp(n)=2.2/(pi)*sinc((1.1/pi)*(n-126))*cos(1.5*(n-126));
end % (Wh+Wl)/2 = 1.5 , (Wh-Wl)/2 = 1.1

Y_bp=fft(h_bp, N/2+250).*fft(y2,N/2+250); % multiply in freq → Y_bp=H_bp*Y2
y_bp=ifft(Y_bp); % y_bp the signal in time .
stem(h_bp) % plot h_bp
title(' BPF')
xlabel('w');
ylabel('h_bp[n]');
H_bp=our_dtft(h_bp); % plot H_bf(e^jw) ,length 128
plot(w, abs(H_bp))
xlabel('w');
ylabel('abs H_bp(e^jw) ');
stem(y_bp) % plot y_bp the signal after pass in BP filter
title('y_bp[n]')
xlabel('w');
ylabel('y_bp[n]');
wavplay(y_bp,8000); % play y_bp
Y_bp=our_dtft(y_bp); % plot Y(e^jw), length 128
plot(w, abs(Y_bp))
xlabel('w');
ylabel('abs Y_bp(e^jw)');
```

התגובה להלם ותגובת התדר למסנן שבחרנו:



נעביר את האות $y_2[n]$ במסנן הנבחר, מסנן זה עדיין לא אידיאלי, אך טוב יותר מקודמיו (בעיקר בגלל מספר הדגימות שלו). מוצא המסנן:

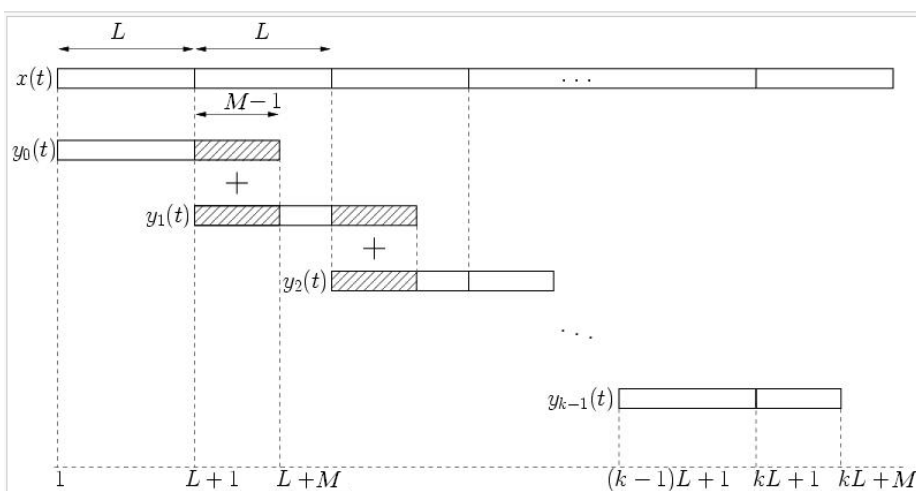


ניתן לראות לפי המוצא בזמן שהרעש כמעט ולא נמצא על הסקלה. והמידע נמצא בשלמותו. שיפור גדול באיכות מהסעיף הקודם! איכות השמיעה טובה, כל מי שהיה סביבנו התפעל ©

סעיף ט"ו:

$$D = 0.503329213 \text{ Mod } 0.05 = 0.003329213 \rightarrow \text{Overlap \& Add}$$

בשיטה זאת נחלק את האות למקטעים, כמו שלמדנו בכיתה. נרפד באפסים, ובצע DFT לכל מקטע, נעשה קונבולוציה עם התגובה להלם לכל מטע, לאחר מכן נסכום את האזורים החופפים. חילקנו את האות 64 למקטעים שבכל אחד 1024 דגימות. $(2^6 * 2^{10} = 2^{16})$. $L =$ אורך המקטע. השתמשנו במסנן BPF מהסעיף הקודם, לאחר שסדרנו אותו לאורך המתאים לפי האלגוריתם של Overlap & Add.



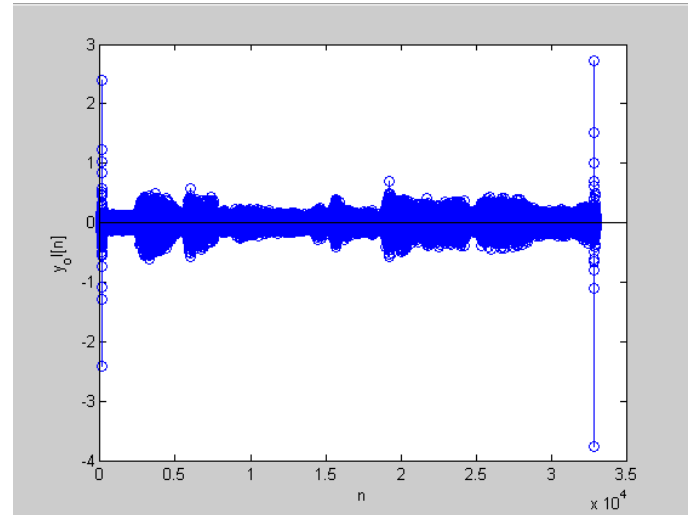
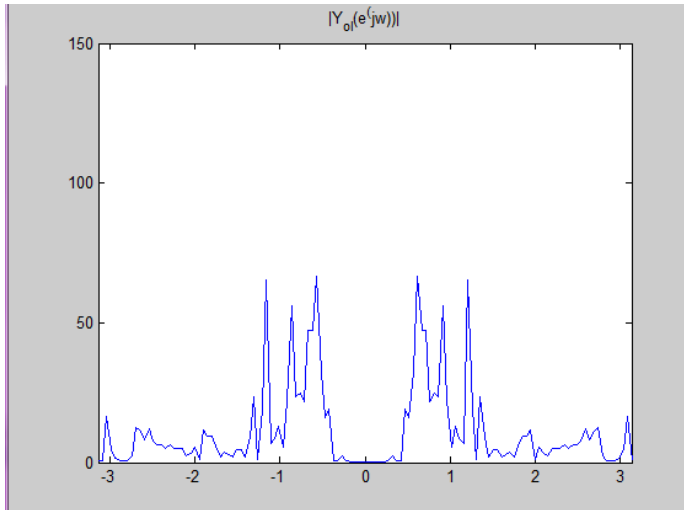
המחשה של האלגוריתם:

```

L=N/64;
y_ol=zeros(1,N/2+250);           % y_ol = after Overlap&Add
H_ol=fft(h_bp, 1274);             % dft of Adjustment h_bp to 1274= 1024+250
for k=0:31                         % overlap & add
    y_ol(k*L+1:L*(k+1)+250)= y_ol(k*L+1:L*(k+1)+250)+ifft(fft(y2(k*L+1:(k+1)*L), 1274).*H_ol);
end
wavplay(y_ol,8000);               % play
Y_ol=our_dtft(y_ol);              % in freq.
plot(w, abs(Y_ol))                % plot overlapped signal
title('|Y_o_1(e^{jw})|')

```

האות שקיבלנו לאחר מימוש האלגוריתם Overlap&Add:



מבחינת איכות השמעה, קיבלנו אותה איכות כמו בסעיף י"ד, ההקלטה נשמעת בבירור ללא הרעש.

אבל לשיטה בסעיף זה יש עדיפות על פני הסעיף הקודם כי היא טובה יותר מבחינת יעילות.

אין צורך לשמור מסנן גדול – אפשר להגיע לתוצאה זהה עם מסנן קטן יותר,

בשיטה זו אין השהיות – אנחנו לא צריכים לחכות עד שכל האות ייכנס כדי להתחיל לעבד אותו.

וגם, אם בתחילת התהליך איננו יודעים עדיין את אורך האות, ניתן להתחיל בסינון – כיוון שאנחנו עובדים על מקטעים (מקסימום נוסף בסוף אפסים וזה לא יפגע לנו במוצא).