אותות ומערכות-תרגיל matlab

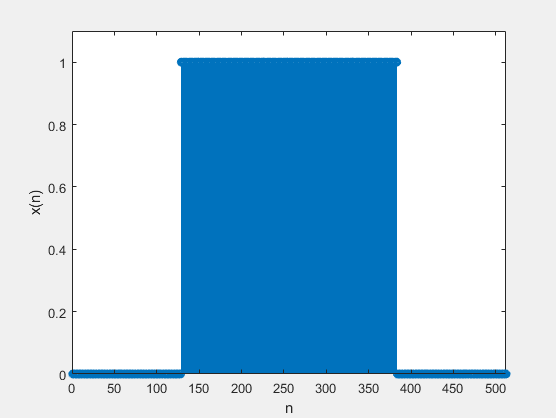
תשפ"א

מגיש: אליהו אטין

ת.ז: 205868771

חלק ראשון:

הכנסנו לפונקציה FourierCoeffGen() חלון בזמן בדיד בצורה הבאה:



כלומר מחזור יחיד של אות באורך 500 עם ערכים של 1 במקומות 125-375 ו-0 בשאר המקומות.

חישוב אנליטי של מקדמי פורייה:

עבור k כללי:

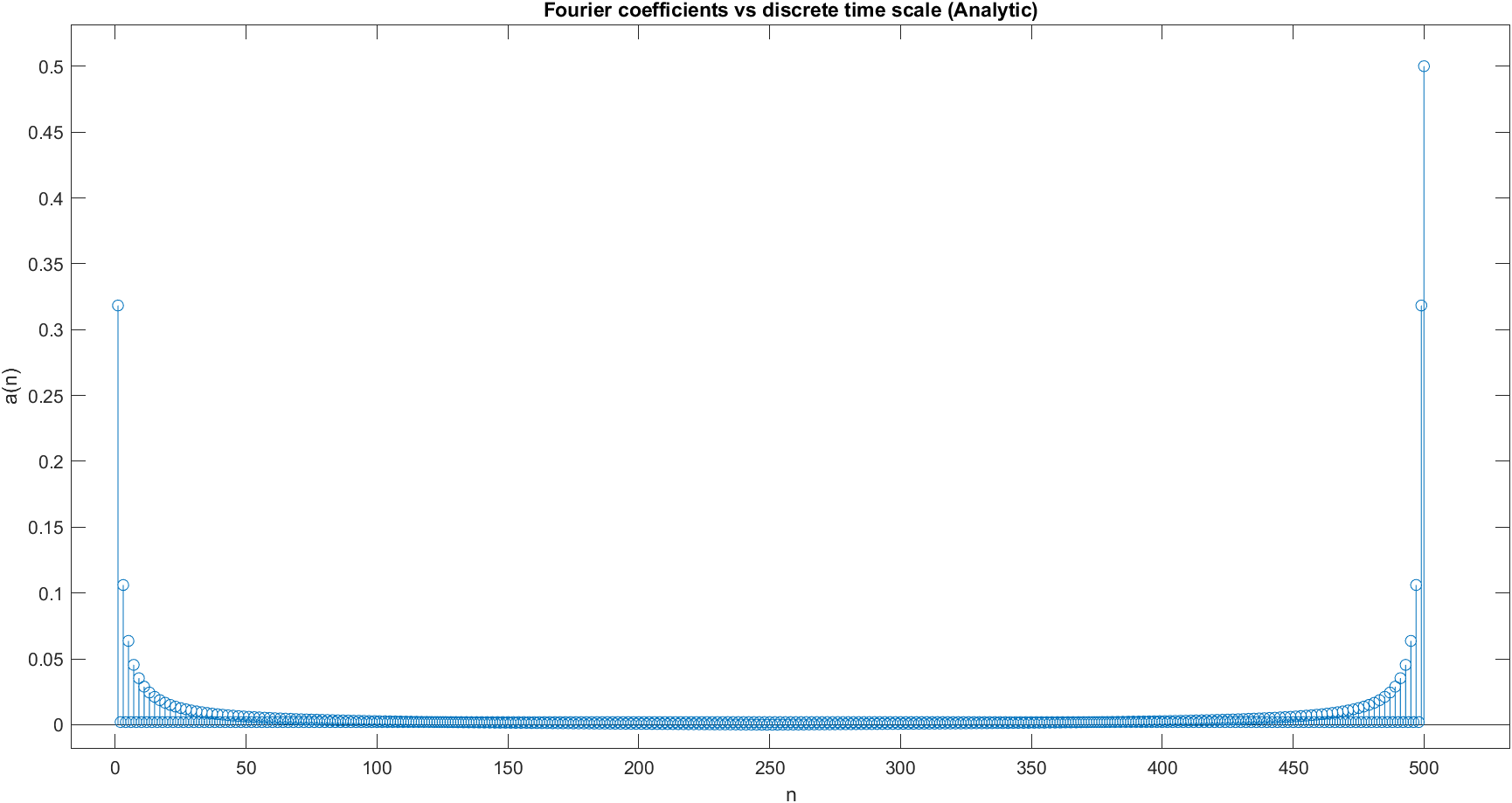
כאשר אורך המחזור N=500 .

כמו כן:

עבור k זוגיים נקבל 0:

תדר הבסיס של האות w0:

והפיתוח לטור פורייה:

גרף התוצאות שהתקבל בmatlab למקדמי הפורייה לפי הנוסחא האנליטית:

הקוד שהשתמשתי בו:

for k=1:500

ak\_analyt(k)= (sin(k\*pi\*251/500))/(500\*sin(k\*pi/500));

end

ak\_analyt(end)=0.5;

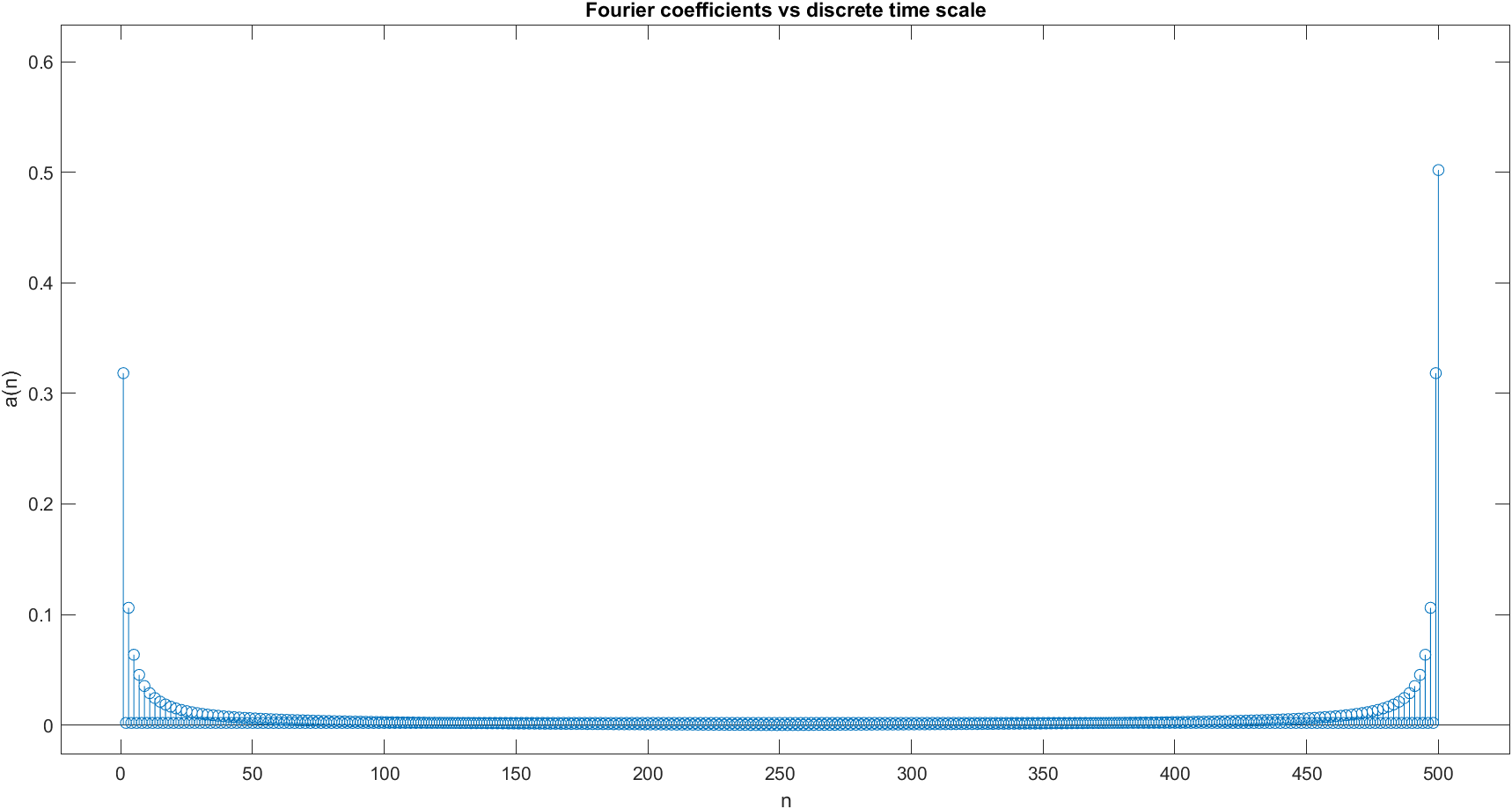
ak\_analyt=abs(ak\_analyt);

stem(ak\_analyt);

xlabel('n')

ylabel('a(n)');

title('Fourier coefficients vs discrete time scale (Analytic)');

גרף התוצאות שהתקבל בmatlab למקדמי הפורייה ע"י שימוש בפונקציית FourierCoeffGen:

הקוד שהשתמשתי בו:

signal=zeros(1,500);

signal(125:375)=1;

N=length(signal);

ak=FourierCoeffGen(signal);

ak=abs(ak);

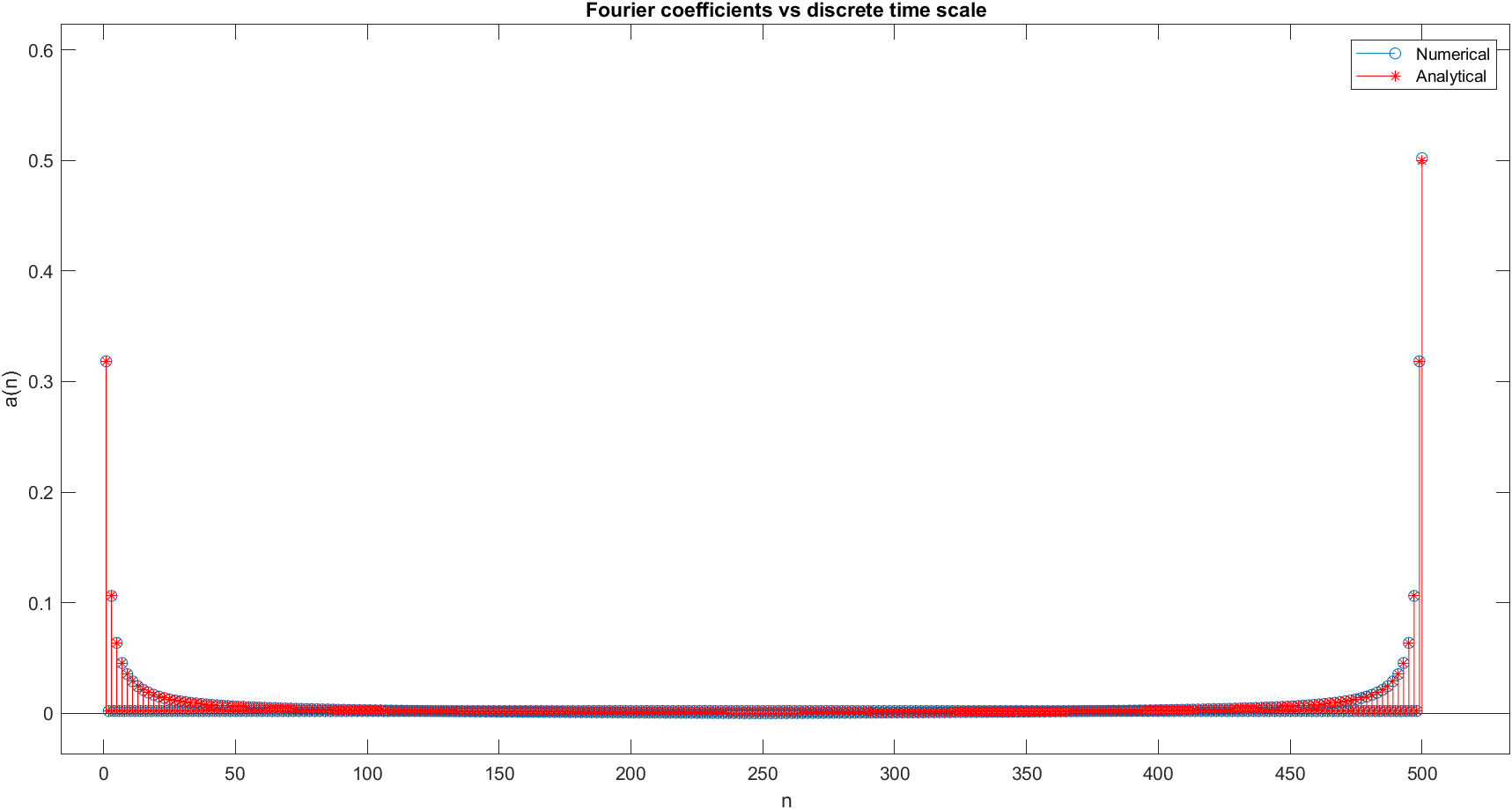
stem(ak);

xlabel('n')

ylabel('a(n)');

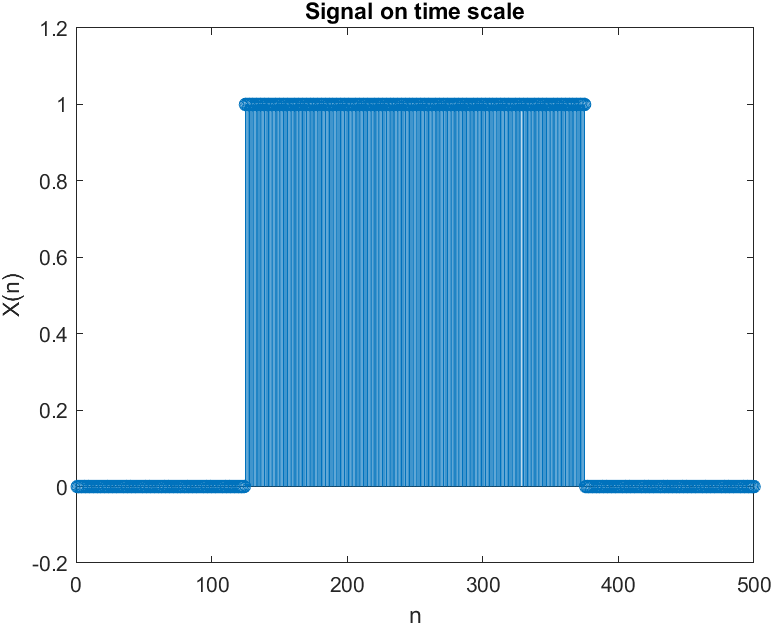
title('Fourier coefficients vs discrete time scale');

\*חשוב לציין שלקחתי את הערך המוחלט של המקדמים המרוכבים.

שני התוצאות שקיבלתי בגרף השוואה אחד:

סה"כ ניתן לראות שקיבלנו התאמה מלאה בין התוצאות האנליטיות לתוצאות במטלב.

כעת השתמשתי בפונקציית FourierSeries כדי לבנות מחדש את האות בזמן ע"י מקדמי הפורייה האות שקיבלתי הוא:

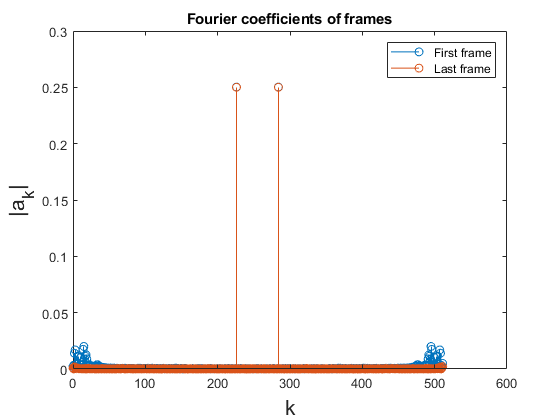


מכיוון שהכנסנו מחזור אחד של האות המקורי השחזור על ידי מקדמי הפורייה נתן לנו את שחזור מלא של מחזור האות.

חלק ב

לאחר האזנה לאות שקיבלנו שמעתי את המשפט: "well it's about time you got here"

עם רעש ברקע.

לשם מציאת ההרמוניות שבהם יושב הרעש הכנסנו את הסיגנל לפונקצייה FourierCoeffGen כמו שניתן לנו בקוד וקיבלתי את גרף המקדמים:

שלב ראשון-זיהוי התדרים בהם נמצא הרעש:

מגרף זה ניתן לראות שהרעש נמצא בערכים הגבוהים ביותר כלומר k0=228 ולכן עלינו לאפס את מקדמי הפורייה במקומות 227=k0 ו512-k0=285 -.

שלב שני-בניית המסנן:

לשם בניית המסנן יצרתי וקטור של אחדות H בגודל 512 כגודל אורך הסיגנל ואיפסתי אותו במקומות הנדרשים (k0 ו- 512-k0)

לשם פעולת הסינון טיפלתי כל פעם בפריים אחד-מסגרת של 512 דגימות מהאות וככה עשיתי לכל הפריימים ולבסוף שרשרתי את הפריימים לאחר פעולת הסינון אחד לשני ויצרתי את האות הנקי ללא הרעש.

הקוד לחלק זה:

ID = 205868771;

[inputSignal,fs,SNR\_in] = inputSignalBuilder(ID);

Nframe = 512;

k0 = 227;

%% Generate an appropriate filter

%2.modify H as needed, use k0

H = ones(1,512);

H(k0)=0;

H(Nframe-k0)=0;

%% Filter frame by frame

z = zeros(size(inputSignal));

for n=1:floor(length(inputSignal)/Nframe) % Frame based operation,

y\_frame = inputSignal((n-1)\*Nframe+1:n\*Nframe);

ak\_frame=FourierCoeffGen(y\_frame);

new\_ak=ak\_frame.\*H;

z\_frame=FourierSeries(new\_ak);

z((n-1)\*Nframe+1:n\*Nframe) =z\_frame(1:512) ;

end

שלב שלישי-בדיקת התוצאות:

בחלק זה בדקנו מהו ה-SNR היחס בין הסיגנל לרעש על ידי הקוד הנתון:

%% Check results

z = real(z);

soundsc(z,fs)

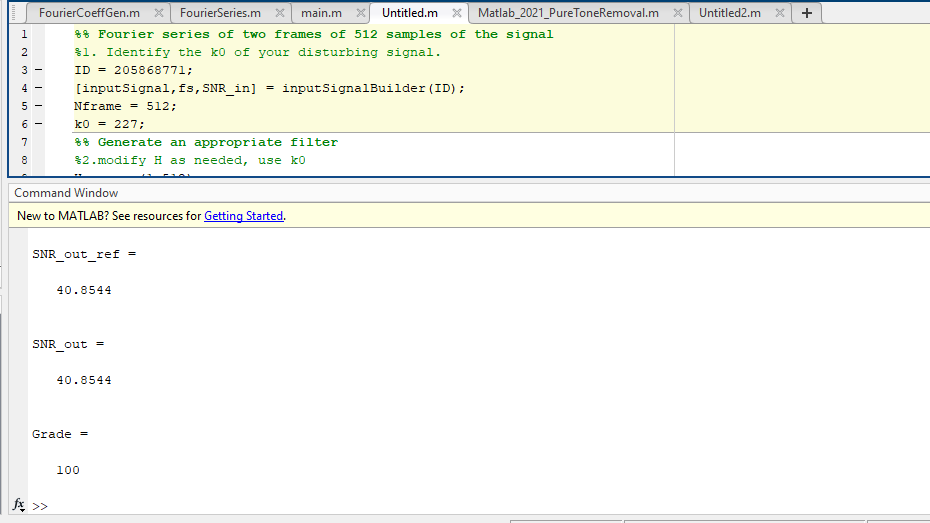
audiowrite(['Output' num2str(ID) '.wav'],z,fs)

[x, fs]= audioread('about\_time.wav');

SNR\_out = 10\*log10(mean(x.^2)/mean((z-x).^2))

%% Performace evaluation:

[Grade, SNR\_out\_ref]= GradeMyOutput(ID,z);

לאחר הרצת הקוד נשמע שוב סיגנל הקול "well it's about time you got here" הפעם בלי הרעש, וקיבלנו SNR\_out=40.8544 כדרוש.

הקודים למטלה:

פונקציית FourierCoeffGen:

function [ak] = FourierCoeffGen(signal)

N=length(signal);

w0=2\*pi/N;

ak = zeros(1,N);

syms n;

for k = 1:N

ak(k)=0;

for n= 1:N

val=signal(n);

ak(k)= ak(k)+val\*exp(-1i\*k\*w0\*n); % ak is fourier coefficient

end

ak(k)=ak(k)/N;

end

end

פונקציית FourierSeries:

function [fs]=FourierSeries(ak)

N=length(ak);

w0=2\*pi/N;

fs = zeros(1,N);

syms k;

for n = 1:N

fs(n)=0;

for k= 1:N

val=ak(k);

fs(n)= fs(n)+val\*exp(1i\*k\*w0\*n); % fs is fourier series of the signal

end

end

end

קוד לחלק ב' של המטלה- סינון הרעש:

%% Fourier series of two frames of 512 samples of the signal

%1. Identify the k0 of your disturbing signal.

ID = 205868771;

[inputSignal,fs,SNR\_in] = inputSignalBuilder(ID);

Nframe = 512;

k0 = 227;

%% Generate an appropriate filter

%2.modify H as needed, use k0

H = ones(1,512);

H(k0)=0;

H(Nframe-k0)=0;

%% Filter frame by frame

z = zeros(size(inputSignal));

for n=1:floor(length(inputSignal)/Nframe) % Frame based operation, Add as many lines as needed

y\_frame = inputSignal((n-1)\*Nframe+1:n\*Nframe);%...

ak\_frame=FourierCoeffGen(y\_frame);

new\_ak=ak\_frame.\*H;

%stem(new\_ak);

z\_frame=FourierSeries(new\_ak);

z((n-1)\*Nframe+1:n\*Nframe) =z\_frame(1:512) ; %...

end

%% Check results

z = real(z);

soundsc(z,fs)

audiowrite(['Output' num2str(ID) '.wav'],z,fs)

[x, fs]= audioread('about\_time.wav');

SNR\_out = 10\*log10(mean(x.^2)/mean((z-x).^2))

%% Performace evaluation:

[Grade, SNR\_out\_ref]= GradeMyOutput(ID,z);