נושא:

Simulation for noise cancellation using LMS adaptive filter

מנחים:   
גולן עין צבי ופרופסור יוסף בן עזרא

מגישים:   
רוסלן אוסמנוב - 327480026

טאל חיים - 312464639

# In this model we will show you the process of canceling noise on a noisy

signal using an adaptive filter applied by LMS algorithem as an implementation to an article called: 'Simulation for noise cancellation using LMS adaptive filter.'

The process included:

1. Import and export original noise-free and pure noise signal as a wav file.   
2. Implement noise components on the signal file randomly.   
3. Define original functions. functions of:  
 Creating a struct for a variable  
 LMS algorithm execution  
 FFT algorithm execution   
4. Create an adaptive filter model and applying it on the signals   
5. Graphs figures of all relevant signals including:  
 Algorithm efficiency feedback  
 All relevant signals  
 All the graphs showed in time domain and frequency domain  
 Stem figure of the coeffs and weights of the model

clc  
clear all  
close all  
% Load the signal and the corrupting factor of the signal  
[y,~] = audioread('Signal.wav');  
[noise,Fs] = audioread('Noise.wav');  
signal =y;  
% Set the noise as a random configuration  
index = randi(numel(noise) - numel(y) + 1,1,1);  
noiseSegment = noise(index:index + numel(y) - 1);  
% Calculate the power components of the siganls  
speechPower = sum(signal.^2);  
noisePower = sum(noiseSegment.^2);  
% Set the factor 'y1' of the system - signal + random noise  
d = y + sqrt(speechPower/noisePower)\*noiseSegment;  
% Set the step size for algorithm updating.  
mu = 0.1;  
% Filter length (num of taps)  
M = 16;  
% Set the start point of the weights of the adaptiv filter  
coeffs = zeros(1,M);  
% Create struct in a new variable (external funciton)  
S = LMSinit(zeros(M,1),mu);  
% Perform LMS-algo. + export set of weights and coeffs(external funciton)  
[~,e,S] = LMSadapt(noise,d,S);  
e = e';  
w = S.coeffs;

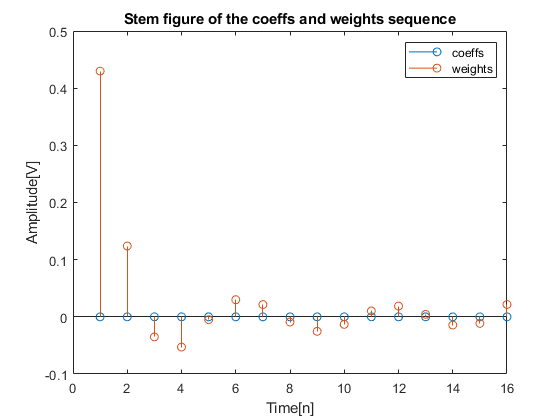
% Figure of three 'time X amp' plots  
figure(1)  
subplot(3,1,1)  
plot(abs(e-signal));% Filt.effectiveness (wieghted signal-origin signal)  
title('Indication of the effectiveness of the LMS');  
xlabel('Time[s]');  
ylabel('Amplitude[V]');  
  
subplot(3,1,2)  
plot(signal);  
title('Original signal in time domain');  
xlabel('Time[s]');  
ylabel('Amplitude[V]');  
  
subplot(3,1,3)  
plot(noise);  
title('The corrupting factor of the signal in time domain');  
xlabel('Time[s]');  
ylabel('Amplitude[V]');  
  
% Figure of signal and filter result combined  
figure(2)  
plot([e,signal]);% Filt.result and original siganl for comparison  
legend('Result of noise cancellation','Actual signal');  
title('Indication of the effectiveness of the LMS');  
xlabel('Time[s]');  
ylabel('Amplitude[V]');  
  
% Figure of the coeffs and weights in one stem  
figure(3)  
stem(coeffs)  
hold on  
stem(w)  
legend('coeffs','weights');  
title('Stem figure of the coeffs and weights sequence');  
xlabel('Time[n]');  
ylabel('Amplitude[V]');  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
% Figure of the FFT of the signals 'amp X freq'  
figure(4)  
limit1 = [-4e3,4e3];% Relevant spectrum of regular speech frequency  
subplot(4,1,1)  
[FFT\_amp,FFT\_freq] = FFT(Fs,signal,0);  
plot(FFT\_freq,FFT\_amp)  
xlim(limit1)  
title('origin signal in frequency domain');  
xlabel('Frequency[f]');  
ylabel('Amplitude[V]');  
  
subplot(4,1,2)  
[FFT\_amp,FFT\_freq] = FFT(Fs,noise,0);  
plot(FFT\_freq,FFT\_amp)  
xlim(limit1)  
title('original noise in freq domain');  
xlabel('Frequency[f]');  
ylabel('Amplitude[V]');  
  
subplot(4,1,3)  
[FFT\_amp,FFT\_freq] = FFT(Fs,d,0);  
plot(FFT\_freq,FFT\_amp)  
xlim(limit1)  
title('noisy signal in frequency domain');  
xlabel('Frequency[f]');  
ylabel('Amplitude[V]');  
  
subplot(4,1,4)  
[FFT\_amp,FFT\_freq] = FFT(Fs,e,0);  
plot(FFT\_freq,FFT\_amp)  
xlim(limit1)  
title('filter results in frequency domain');  
xlabel('Frequency[f]');  
ylabel('Amplitude[V]');

sound(e,44100)

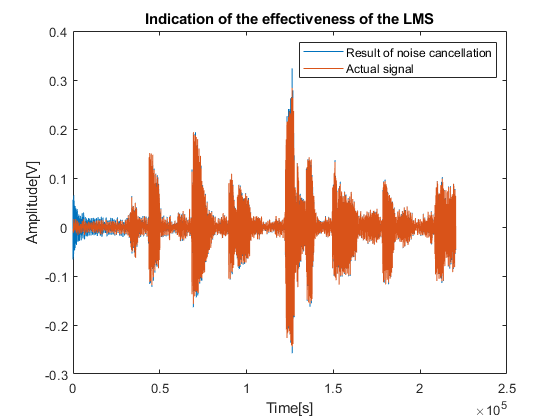
sound(d,44100)

sound(signal,44100)

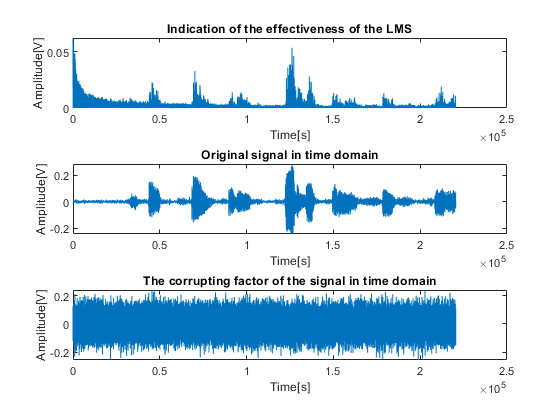
[*Published with MATLAB® R2022a*](https://www.mathworks.com/products/matlab)



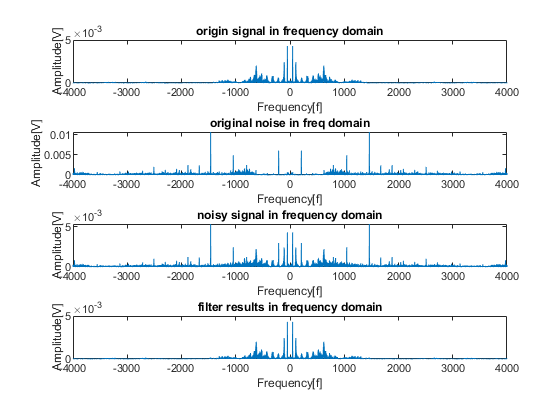
גרף 1:  
הסבר....



גרף 2 :  
הסבר.....



גרף 3:  
הסבר.....



גרף 4:  
הסבר.......

**סיכום סתמי על המאמר לצורך הכנת המצגת:**

רעשים (אותות בלתי רצויים) קיימים ויהיו קיימים בכל מקום ובכל מערכת.  
ספציפית במערכות שידור מידע, ישנן 2 גישות להורדת רעשים.  
האחת היא באופן פיזי מכאני חומרי.  
והשנייה באופן דיגיטלי תוכנתי מערכתי.  
ל2 הגישות יתרונות וחסרונות אך לגישה הדיגיטלית יתרונות עדיפים בשל היעילות החסכון במשאבים והקומפקטיות שניתן להשיג וליישם אותה במערכות.  
עם השנים הצלחנו להגיע לרמות סינון מאוד גבוהות של רעש והמון פיתוחים נעשו בתחום.  
אחת השיטות הדיגיטליות שפותחו לשם כך היא סינון אדפטיבי.

במאמר זה אנו נתבונן באלגוריתם סינון, באמצעות שיטת סינון אדפטיבי, בשם LMS.  
אלגוריתם LMS מאופיין כמסנן אדפטיבי שמסנן רעש יעיל ואפקטיבי.  
בתהליך המחקר לקחו 2 אותות (אות דיבור ואות רעש) עם דגימה של 44100 Hz ועשו בינהם מניפולציה.  
הרעש, הפוגם באות המידע המקורי, מוכנס ככניסה למסנן האדפטיבי(y2).  
אות הדיבור ללא הרעש עובר מיקס עם רעש ונהפך לאות מורעש (y1) שנכנס לסוכם עם תוצאת הLMS (y2’).  
y1 וגם y2’ עוברים פעולת הפרש והתוצאה (e) נכנסת כאינדיקציה חדשה לשינוי המשקלים של הפילטר.  
פרמטר שינוי המסנן (μ) נקבע מראש לפי ניסוי ותהייה על אופי התוצאות.   
הפילטר מוגדר עם כמות טאפים (סדר מסנן, אצלינו 'M') קבועה מראש (גם כאן לפי ניסוי ותהייה).  
אות היציאה הסופי של הLMS (הסיגנל המפולטר, אצלינו 'e') והאות המקורי (אצלינו 'signal') עוברים הפרש בינהם והתוצאה של ההפרש נותנת אינדיקציה על יעילות הLMS בביטול הרעש בכךשמצפים לקבל הפרש אפסי ומבינים אם יש הבדל היכן ההבדל (באילו קומפוננטות של האות).  
לאחר מכן מבצעים FFT ל2 האותות כדי לזהות את תחום התדרים המבוטל.

μ הסבר:  
הפרמטר μ מסמל את קצב שינוי המסנן.  
זה מאפשר לשלוט בכמות ההתאמה (קצב הלמידה) המבוצעת למקדמי המסנן בכל איטרציה.   
ערך קטן של μ יגרום להתכנסות איטית אך גם יפחית את הסיכוי לחרוג מהפתרון האופטימלי  
ובדיוק הפוך לערך μ גדול (התכנסות מהירה והגדלת סיכוי חרגיה)  
הפרמטר נע בין ערכים 0 עד 1.

M הסבר:  
הפרמטר M מסמל את סדר המסנן.  
סדר המסנן (זיכרון המסנן \ מספר מקדמי המסנן) מאפשר לשלוט במספר כניסות העבר ששימשו לניבוי הפלט הנוכחי של המסנן.  
למסנן מסדר גבוה יותר יהיו מקדמים רבים יותר, יותר זיכרון, מה שאומר שהוא יכול ללכוד דפוסים מורכבים יותר באות הקלט.  
הטרייד אוף הוא שישנה דרישה ליותר נתונים כדי להעריך את המקדמים, מה שיכול להגדיל את העלות החישובית וכן אף להורדת ביצועים מהתאמת יתר (אם הנתונים מוגבלים).  
למסנן מסדר נמוך השלכות הפוכות.  
אלגוריתם LMS פחות רגיש לסדר הסינון ודורש יחסית פחות נתונים כדי להעריך את המקדמים (סביב 10 עד 20).