

# **מעבדה בהנדסה ביו-רפואית 3 (עיבוד אותות פיזיולוגיים)**

---

**מעבדה מס' 4 – עיבוד אותות דיבור**

**שמות המציגים :**

**מור יוסף 318476850**

**מייכאל פולונייק 203833041**

**שם המדריך :**

**שני ערומי**

**תאריך הגשה :**

**11.01.2021**

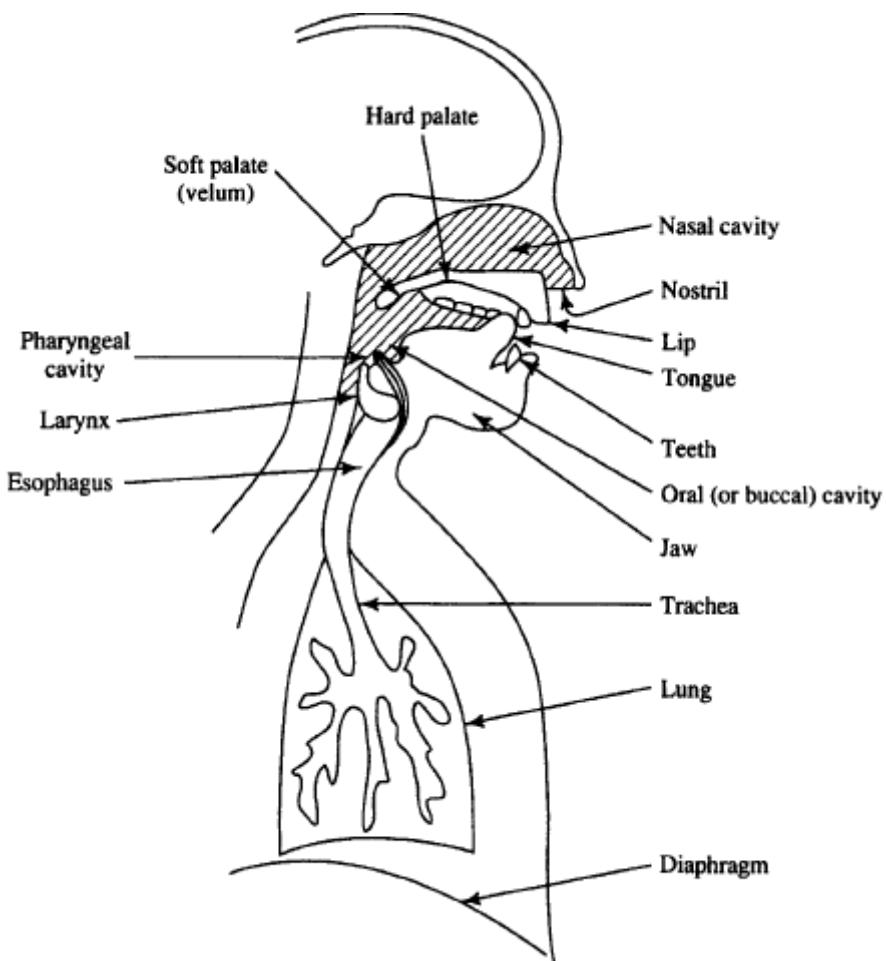
## **תוכן עניינים:**

4	1.	רקע תיאורטי .....
7	2.	שימוש מקדים לאות הדיבור .....
7		תשובה לשאלה 1.1 : .....
9		תשובה לשאלה 1.2 : .....
10	3.	סגננטציה אוטומטית .....
10		תשובה לשאלה 2.1 : .....
11		תשובה לשאלה 2.2 : .....
11		תשובה לשאלה 2.3 : .....
12		תשובה לשאלה 2.4 : .....
13	4.	שערץ ספקטRALי ומציאות פורמנטוות .....
13		תשובה לשאלה 3.1 : .....
14		תשובה לשאלה 3.2 : .....
17		תשובה לשאלה .....
17		תשובה לשאלה 3.4 : .....
18		תשובה לשאלה 3.5 : .....
19		תשובה לשאלה 3.6 : .....
21		תשובה לשאלה 3.7 : .....
22	5.	מצוי מאפיינים .....
22		תשובה לשאלה 4.1 : .....
23		תשובה לשאלה 4.2 : .....
23		תשובה לשאלה 4.3 : .....
23		תשובה לשאלה 4.4 : .....
24		תשובה לשאלה 4.5 : .....
24		תשובה לשאלה 4.6 : .....
26	6.	בנייה המודל – יצרת בסיס הנתונים .....
26	7.	ניתוח STFT (ספקטרוגרפיה) .....
26		תשובה לשאלה 6.1 : .....
27		תשובה לשאלה 6.2 : .....
29	8.	סיווג דובר באמצעות המערכת .....
29		תשובה לשאלה 7.5 : .....
31		תשובה לשאלה 7.6 : .....

31 .....	תשובה לשאלת 7.8 :
34 .....	תשובה לשאלת 7.9 :
42 .....	9. מסקנות כלליות .....
44 .....	10. מקורות .....
45 .....	11. נספחים .....

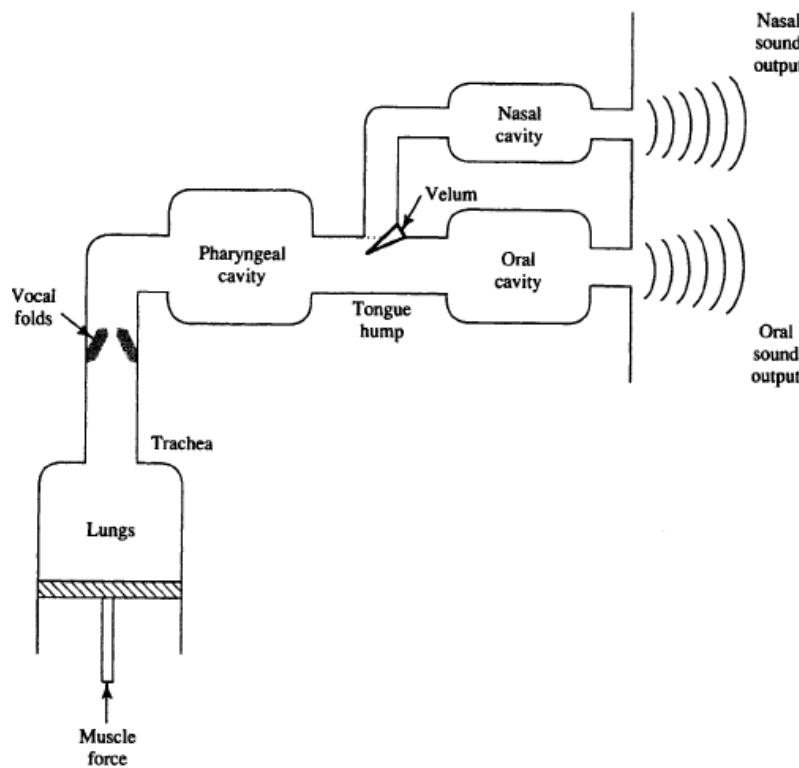
## 1. רקע תיאורטי

אותות דיבור אלו גלי קול אקוסטיים שמקורם בתנועה וולונטרית של מבנים בגוף האדם היוצרים את אות הדיבור.



איור 1- איברים המשתתפים ביצירת אות דיבור [1]

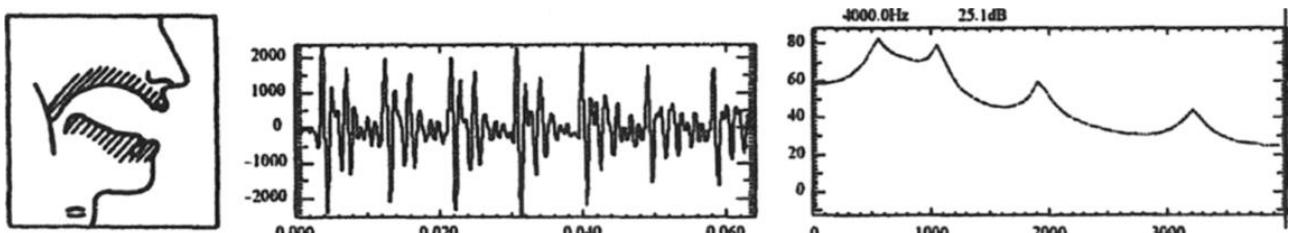
באיור 1 ניתן לראות מבט סגיטלי (פרופיל) את האיברים המשתתפים ביצירת אות דיבור. המערכת מורכבת בעיקר מריאות, קנה הנשימה,喉 (תיבת הקול)- איבר האחראי על יצירת קול, oral cavity, pharyngeal cavity (לוע), (חלל הפה), nasal cavity (חלל האף). באופן כללי, אנו מחלקים את מערכת הדיבור לשולשה חלקים בגוף האדם שייצרים את אות הדיבור. ה-Vocal tract (חלל הלוע והפה), זה השלישי העליון של מערכת הדיבור. מוצאת אות תיבת הקול מגיעה אל ה-Vocal tract אשר מסתiens בקצת השפתיים. השלישי הוא喉 (thyroid) והשלישי הוא חלל הבطن, המוכנה subglottal respiratory system, לומר מה שנמצא מתחת לתיבת הקול. נוח להתבונן ביצירת אות דיבור בתווך סינון אקוסטי של אותן שמע. גלי קול בעליים באמצעות תנומת הסרעפת דרך הריאות אל קנה הנשימה ומגיע לתיבת הקול. בתיבת הקול יש לנו את מתרי הקול השולטים על אופי הדיבור. כאשר באות א-קולי, מתרי הקול פתוחים וישנו מעבר של גלי קול ללא הפרעה ובאות קולי מתרי הקול נסגרים בקצב מסוים pitch- גברים – 20 נשים [Hz] – 120 [Hz]. [1]



איור 2 - סימולציה של יצירת אות קול[1]

באיור 2 ניתן לראות את הסימולציה של יצירת המבנה הרזונטי (פיקים בספקטרום הנקבעים על ידי ה- Vocal tract) של אות דיבור. בסימולציה זו אנו מניחים כי מדובר בגלוי קול אקוסטיים המתקדמים בתוך החללים השונים ("צינורוטי") ללא כל הפסדים. אנו רואים כי אות השמע יכול לעבור דרך שני חללים, אחד הוא "רגיל" (דרך חלל הפה). השני מתקיים כאשר העובל מונחך, נכנסים גלי קול אל חלל האף וזה נשמע כדיבור מן האף. מה שמאפשר את ה Acoustic coupling בין האות מן האף והפה.

היחידה הבסיסית המתארת כיצד דיבור מקבל משמעות לינגייסטית נקראת פונמה. כאשר באמצעות מס פונומות בסדר תקין, נוכל להרכיב "מילה". כל פונמה מורכבת ממספר פורמנטוֹת (המייצגות את תדרי הפיקים בתגובה לתדר- רזוננס אקוסטי) נציג דוגמה מן הספרות לפונמה ואו



איור 3 - פונמה \a [1]

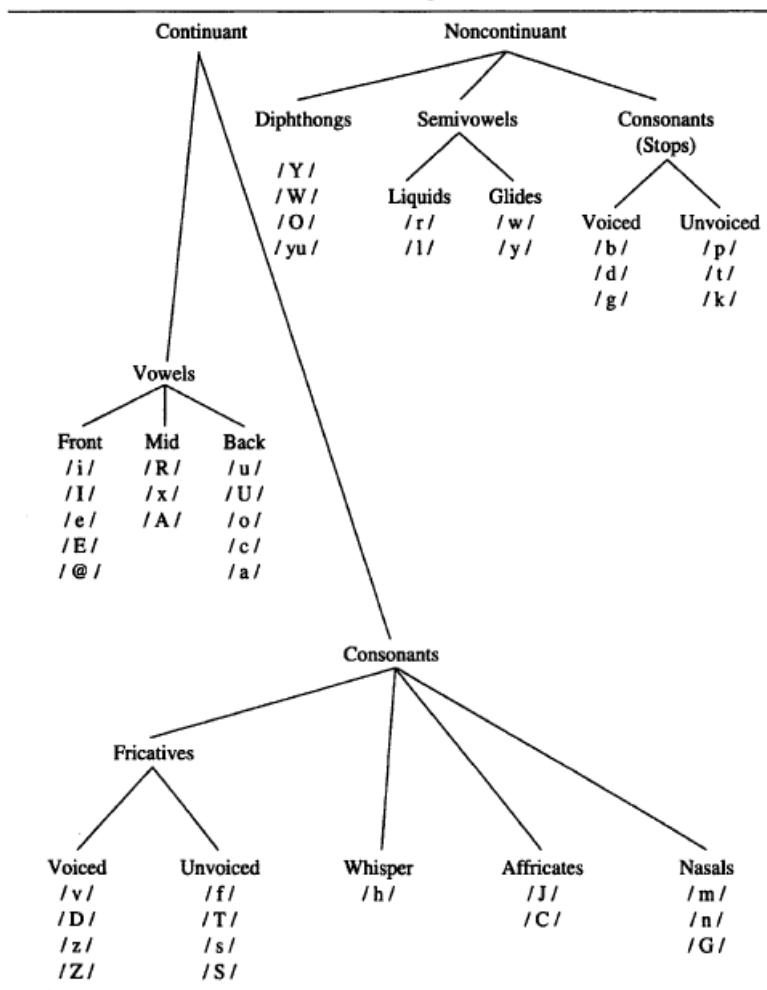
באיור 3 ניתן לראות הייצוג של ה Vocal tract שמאלי, את הייצוג הזמני של אות הדיבור (במרכז) ואת הייצוג התדרי שלה (מימין). ניתן לראות כי ישנו כ 3 פיקים מרכזיים עד [Hz] 3000 ועוד אחד בתדר גבואה יותר.

ככל, אנו יודעים כי אותות דיבור אלו אותות המשתנים בתוכנותיהם בזמן (לא סטציוני), אך אם נחלקו למקטעי זמן של ההברות השונות הוא בקירוב יהיה סטציוני, כך שנitin ישיה לחקור את הסטטיסטיות של האות במקטעי זמן קצרים [1].

בכדי לייצור אותן דיבורים, יש צורך ביצירוף סופי של הבחרות מסוימות, כאשר כל אחת מסמנת אותן מסוימות. היחידה הבסיסית אליה ניתן לתאר משמעותו לשונית היא פונמה. באנגלית, יש בערך 42 פונמות מהבחרה אחת, או שניים יחד. הפונמות מחולקות להמוני תתי קטגוריות, כאשר מבנה חלל הפה והאף מחשך תפקיד חשוב, כמו כן אם יש לנו תנוצה כלשהי של חלל הפה תוך כדי ייצור הפונמה. באופן כללי ניתן לומר כי כל פונמה מהויה סוג מסוים של תדרים אקוסטיים(פורמננטות) המייצגים הבחירה מסוימת.

פונמה נחשבת סטציונרית או רציפה, אם הפונמה נוצרת בעקבות קונפיגורציה יציבה של ה- vocal tract. לעומת זאת, פונמה לא רציפה היא פונמה כזו שבמהלכה היה שינוי של חלל הפה והאף ובשל תנועתם יהיה יותר קשה לזהות אותן.

**TABLE 2.3. Phonemes Used in American English.**



איור 4 - פונמות באנגלית [1]

באילור 4 ניתן לראות הפונמות השונות הממצוות באנגלית. לפיה ישנה חלוקה כללית בין פונמות רציפות או לא. בכלל, ניתן לראות כי הפונמות הרציפות מחולקות לשתי תתי קטגוריות, רציפות ולא רציפות. כאשר בתוכן ישנן גם פונמות קוליות וגם לא קוליות. החלוקה הראשית מדברת על התנועה העיקרית של חלל הפה והולע. Vowels אלו פונמות קוליות בהן האוויר זורם ללא כל הפרעה מהריאות אל חלל הפה והחוצה. consonant עלי פונמות המוגדרות כעיצורים, ישנה חסימה כלשהי של האוויר לאורך vocal tract כך שנקלט שחלקו קוליות וחלקו לא כמו כן ישנו פונמות אשר נאמרות עם חסימה של הפה ויציאת אוויר דרך הנחירויות(Nasal)-/m/ למשל.

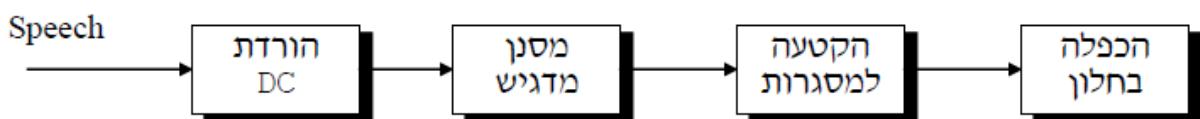
כאשר מתרי הקול מרוחקים, יש מעבר של אויר וכתוצאה מכך הfonema נחשבת א-קולי. לעומת זאת, בfonema קולית, ישנה תנואה של מתרי הקול בקצב אחד pitch. מה שיכל לראות בזמן כאוט מוחורי.

אותות דיבור בדרך כלל מכילים תכולה ספקטרלית [Hz][0, 8000]. כאשר האוזן האנושית מסוגלת לשמעו [Hz] 20 – 20,000. לכן, דגימתו אותו דיבור בדרך כלל נעשית בתנים תדרים [Hz] 48,000 – 8,000. כאשר תדר הדגימה ייבחר על ידי האפליקציה הרצוייה. רוב האינפורמציה נמצאת עד [Hz] 3500 כך ש ניתן לדוגם בקצב המינימלי ועדין להבחן ב 4 פורמננטות ראשונות המהוות את רוב הקידוד של הfonema הבסיסית. דגימה בקצב נמוך יכולה לגרור Aliasning, כך שתדרים גבוהים יקפו לנמוכים ולא יוכל להבין את הfonema אם נסעה לשמעו אותה או אפילו להתחזות לפונמה אחרת [2].

## 2. עיבוד מקדים לאורות הדיבור

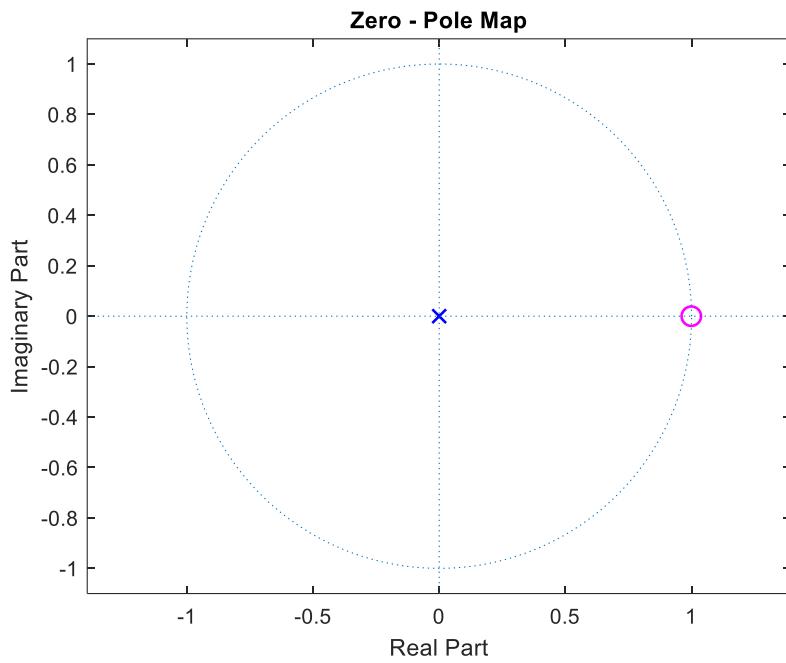
### תשובות לשאלת 1.1:

כאן נסביר על העיבוד המקדים שיש לעשות על האות הספרתי.



איור 5 - סכימה של Pre processing

- **הורדת DC :** מתבצעת באמצעות החסרת הממוצע של האות מכל אחד מן האיברים באותו.
- **מסנן מדגיש :** אותו אנו מבצעים על מנת להעלות את ה SNR של האות, המסנן הינו HP שתפקידו לבצע הגברה של תדרים הגבוהים, שנפגעים יותר כאשר ישנו מעבר של גלי קול בין החללים השונים במערכת הדיבור. מסנן כזה נראה מן הצורה של  $H(z) = 1 - \alpha z^{-1}$ . ככלمر אנו מקבלים פילטר בעל קווטב בראשית ואפס הקרוב ל 1, ככלמר הנחתת DC.

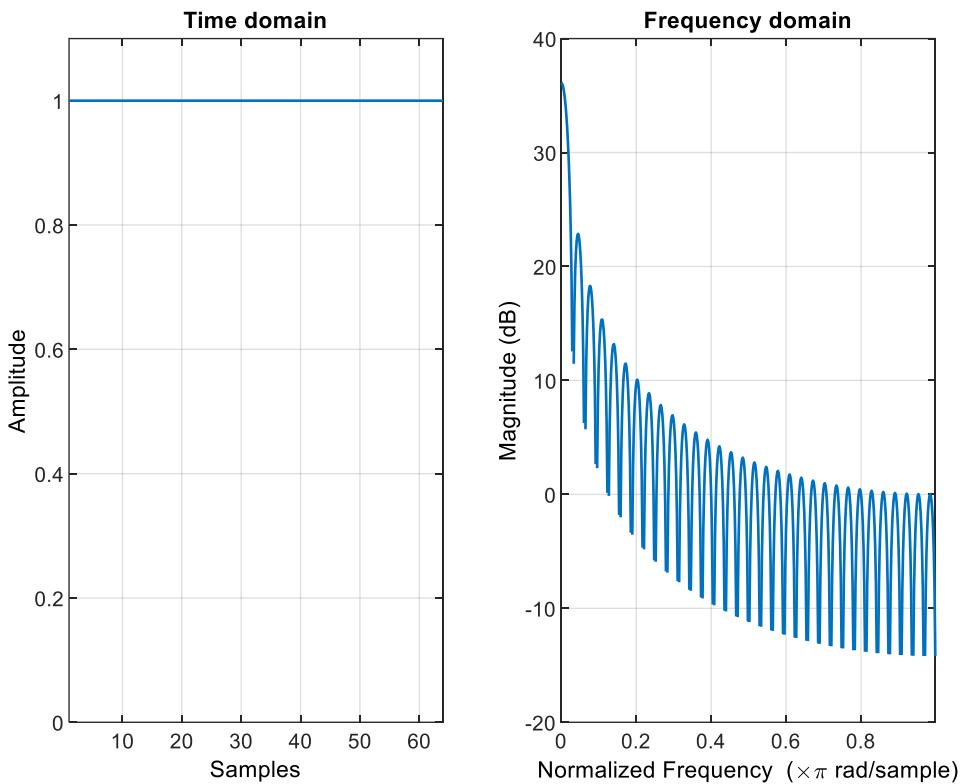


איור 6 – מפת הקטבים והאפסים של מסנן מדגש

באיור 6 ניתן לראות את הקוטב בראשית ואת האפס הנמצא ב DC.

- הקטעה למסגרות: מן הנאמר בקורס התיאורטי, אנו יודעים כי בחולנות זמן קיצרים אותן הדיבורים בקשר סטציונרי. לכן, אנו נחלק את האות למקטעים של 30 מילישניות עם חפיפה של 50 אחוזים בין כל אחד מן החולנות. בחרה זו, נוכל לשערך את השינויים הזמנניים ומתוכם למדוד את הסטטיסטיקה של אותן.
  - הכפלת החלון: לאחר ואנו רוצחים להתבונן במסגרות, אנו מכפילים את המקטע הזמני בחלון כלשהו בזמן,
- משמע קונולוציה בתדר של המקטע המוחושב עם המקטע הייצוג הספקטרלי של החלון

חישוב FFT משתמש בהנחה שהאות מחזורי וחוזר על עצמו בכל התחום הזמני. עבור ניתוח מעשי של אותן דיגיטליים, חלק קצר מן אותן נדרש להיות משוערך. אם ישנו הבדל בין FFT של אותן הקצר לבין FFT של אותן המוקורי, ישנה הנחה כי FFT של אותן הקצר חוזר על עצמו בשכפולים אינסופיים, אך זה לא המצב, ויהיה לנו ספקטרום מעט שונה בקטע החלון הבא. על כן, תהיה לנו השפעת קטנה בין מקטעים חוזרים ולכן תהיה לנו זליגה. בעקבות זליגה אנו עלולים לקבל אנרגיה בתדרים בהם הספקטרום המקורי עלול להתאפס. אנו נציג את התצוגה הזמנית וההתדרית של חלון ריבועי.



איור 7- חילון ריבועי בגודל 64

מאייר 7 רואים את התמונה הזמן והתדרית של החלונות. חילון זה לא יגרום לעיוות באוט הזמני אך נקבל במעט אפקט מריחה בספקטרום המשוערך. זה יגביל את היכול שלנו להבחין בתדרים בהפרש נמוכים. בעוד שבבואה שלנו אנו מעוניינים לשימוש לבתדרים שייחסית מרווחים אחד מן השני.

### תשובה לשאלת 1.2:

כתבנו את הפונקציה הבאה :

```
function [ProcessedSig,FramedSig]=PreProcess(Signal,Fs,alpha,WindowLength,Overlap)
%
%
% Signal - the raw speech signal; Fs- Sampling frequency
% alpha - pre-emphasis filter parameter;
% WindowLength - window length [seconds];
% percentage of overlap between adjacent frames [0-100]
% FUNCTION OUTPUT:
% ProcessedSig - the preprocessed speech signal;
% a matrix of the framed signal (each row is a frame)

N = WindowLength*Fs; % [sec]*[sample/sec]=[sample]
% Remove DC noise:
Signal = Signal - mean(Signal);
% Pre-Emphasis filter
ProcessedSig = filter([1 -alpha],1,Signal);
FramedSig = enframe(ProcessedSig ,rectwin(N), ((Overlap)*N)/100 );
end
```

כאשר השתמשנו בפונקציה enframe שהייתה ברשותנו מעיבוד אותות פיזיולוגיים. הייתה לנו בעיה עם כך שהארגון של הפונקציה קיבל את החיפוי באינדקסים. לכן, הומרה הייתה לבטל את האחיזים ולהכפיל באורך החלון המקורי.

### 3 סגמנטציה אוטומטית

#### תשובות לשאלת 2.1:

משערך צפיפות ספקטרלית מסוג Periodogram :

$$(1) \quad \hat{S}_x(e^{j\omega}, n) = \sum_{k=-N+1}^{N-1} \hat{r}_x(k, n) e^{-jk\omega k}$$

הוא התמרת פורייה על  $[1 - n, n + N]$  דגימות המשערכוות את פונקציית האוטוקורלציה :

$$(2) \quad \hat{r}_x(k, n) = \begin{cases} \sum_{l=0}^{N-1-k} x(l+n+k) \cdot x(l+n); & k = 0, \dots, N-1 \\ 0 & ; k = N, N+1, \dots \end{cases}$$

מדד Spectral Error Measure הינו ממד אשר מאפשר לראות את השינויים הספקטרליים בין הסגמנטים השונים באות. הממד האינטואיטיבי הוא לראות את השגיאה הספקטרלית הריבועית בין חלון המדידה לחלון הרפרנס.

$$(3) \quad \Delta(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} [S_x(e^{j\omega}, n) - S_x(e^{j\omega}, 0)]^2$$

כאשר  $S_x(e^{j\omega}, 0)$  זו הפריזודוגרמה של חלון הרפרנס ו  $S_x(e^{j\omega}, n)$  זו הפריזודוגרמה של חלון המדידה. בעיבוד אותן דיבור, נרצה לחלק את אותן לפי התכונות הסטטיסטיות שלו כך שכל חלק באות יבדיל הפונקציות השונות. לשם כך אנו נרצה לבצע סגמנטציה עם מطلב. לפניו ביצוע הסגמנטציה אנו נרצה לעבד בצורה ראשונית את אותן שבעצם אנו לבצע סגמנטציה לאות המורכב מסדרה של מקטעים בעלי אורך קבוע, כאשר כל מקטע בעל אותן תכונות סטטיסטיות שלא משותנות בתוך הסגמנט עצמו. ככלمر זיהוי השינויים בין הסגמנטים יתבסס על סטטיסיקה מסדר שני. לצורך ביצוע של סגמנטציה נדרש להיקבע שתי חלונות, חלון הרפרנס וחלון המדידה. את התכונות הסטטיסטיות של הסגמנט המדידה אנו נגלה באמצעות השוואה לחלון הרפרנס. לצורך כימיות ההשוואה בנים נרצה להגדיר ממד שניי  $(n)\Delta$ . השוני בין הסגמנטים יקבע עבור מדידה  $n$  כך ש  $\eta > (n)\Delta$ . ברגע שבו עברו את הסף, תהליך הסגמנטציה יעבור לשלב הבא בו מגדרים את חלון הרפרנס וחלון המדידה הבאים. כדי להימנע מסגמנטציה קצרה מזמן הרצוי, נבדק שהרף עבר פרק זמן מסוים, של 30 מילישניות עם חפיפה של כמעט 50 אחוזים.

כפי שראינו בקורס עיבוד אותות פיזיולוגיים, ממד זה לא סימטרי מבחינה ירידית או עלייה בהספק. לצורך התמודדות עם בעיה זו הוגדר ממד שגיאת ספקטרלית נוספת המנורמל את  $(n)\Delta$ .

$$(4) \quad \Delta_1(n) = \frac{\Delta(n)}{\frac{1}{2\pi} \int_{-\pi}^{\pi} S_x(e^{j\omega}, n) d\omega \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} S_x(e^{j\omega}, 0) d\omega}$$

כך שעבור הקטנה או הגדלה של הספקטרום פי קבוע כלשהו מודד השגיאה משתנה בדרך זהה, لكن המודד סימטרי.

## תשובה לשאלת 2.2:

```

function Idx = FindWordIdx(FramedSig,Fs,WindowLength,Overlap)
% FindWordIdx finds the start and end of speech using energy
% calculation. based of certain threshold
% FramedSig - the framed speech signal after preprocessing.
% Fs - sampling frequency
% WindowLength - length of test and reference windows [seconds]
% Overlap - percentage of overlap between adjacent frames [0-100]
% OUTPUT:
% Idx - 2 integer vector: start and end indices of detected word.

Signal_Energy=calcNRG(FramedSig);%calculate of each frame
thresh=max(Signal_Energy)*0.001;
Frames_with_Speech=find(Signal_Energy>thresh);
WindowLength_samples=WindowLength*Fs; % [sec]*[sample/sec]=[sample]
overlap_in_samples=((Overlap)*WindowLength_samples)/100; % overlap in samples

Idx=zeros(1,2);
% find relevant index according to the overlapping.
idx_start=(Frames_with_Speech(1))*overlap_in_samples;
Idx(1)=idx_start;
idx_end=(Frames_with_Speech(end))*overlap_in_samples;
Idx(2)=idx_end+WindowLength_samples-1;

end

```

## תשובה לשאלת 2.3:

```

function [segment_ind,delta]=segmentation(signal,winlen,eta,dt,Fs,Idx)
% this function takes speech signal and convert it to segments based on
% spectral error measure
% signal - the speech signal after preprocessing
% winlen - length of test and reference windows [seconds]
% eta - threshold for Δ1 spectral error measure
% dt - minimum time above threshold 'eta' [seconds]
% Fs - sampling frequency
% Idx - start & end indices of the word [samples]
% OUTPUT
% Seg_ind - index for the beginning of each segment
% delta - spectral error measure Δ1

% stay only with the indices of the word
delta=zeros(length(signal),1);
% Num in samples window reference, test & dt [S]*[sample/S]=[sample]:
N = winlen*Fs;
dt_in_samples = dt*Fs;
% start the problem from Idx(1)
index_ref=Idx(1);
index_test=Idx(1);
segment_ind=Idx(1);

% main while loop.. end when the next reference window + dt in samples will
% cross the length of Idx(2)
while index_ref+N-1+dt_in_samples <= Idx(2)
    flag=1;
    % create reference and test signals

```

```

Ref=signal(index_ref:index_ref+N-1);
% another loop for the test window. stop when the end of test will cross
% Idx(2) or when delta>eta for at least dt time
while index_test+N-1<=Idx(2) && flag==1
    Test=signal(index_test:index_test+N-1);
    % assuming we have a wide-sense stationary random process.
    % calculate the periodogram of each window
    pxx_ref_win=periodogram(Ref,[],length(Ref));
    pxx_test_win=periodogram(Test,[],length(Ref));
    % create omega vector to integrate over it. the signal is real so
    % we will integrate over [-pi,pi]
    w=linspace(-pi,pi,length(pxx_ref_win));
    % calculate spectral error measure over test and ref window
    delta_numerator=(2*pi)*trapz(w,((pxx_test_win-pxx_ref_win).^2));
    Delta=(delta_numerator)*(1/( trapz(w,((pxx_test_win))) * ...
        trapz(w,((pxx_ref_win)))) );
    % update the spectral error measure Δ1 in delta vector
    delta(index_test+dt_in_samples)=Delta;
    % verify that we reached threshold and that the minimum length of
    % foneme is 40 [ms]
    if Delta>eta && index_test-index_ref >((40*10^-3)*Fs)
        % verify that the threshold is reached for dt time
        How_many_delta_greater_then_eta = delta(index_test-dt_in_samples+1:index_test)>eta;
        check_dt=sum(How_many_delta_greater_then_eta);
        % if we did, update flag for us to go for the next segment
        if check_dt>=dt_in_samples
            flag=0; % go to next segment && index_test-index_ref >((100*10^-3)*Fs)
        end
    end
    % move the test window index by one
    index_test=index_test+1;
end
% update the segment indices, add one index on the right side of segment_ind
if check_dt>=dt_in_samples
    segment_ind=[segment_ind (index_test-(dt_in_samples))];
end
% update the test and reference indices before advancing to the
% next segment:
% for the index_test: calculate where we started to cross the threshold
% with the test window. and then subtract the length of dt in [samples]
index_ref=index_test-(dt_in_samples);
% now we can update
index_test=index_ref;
end

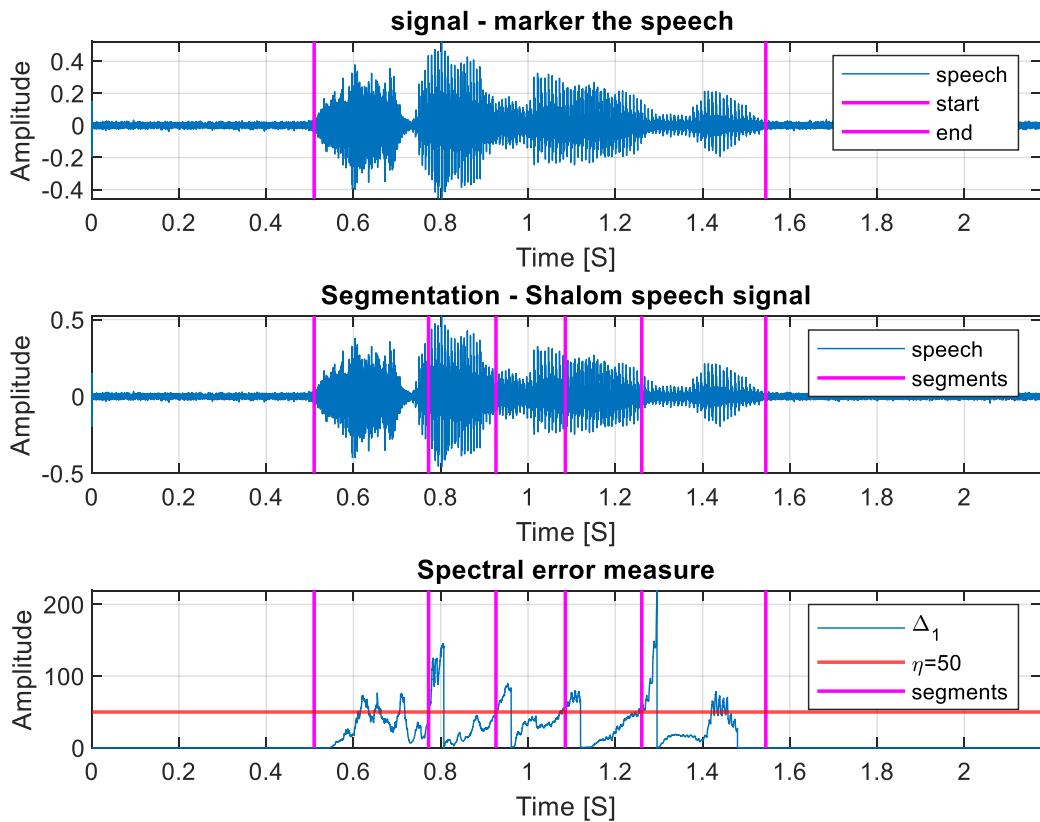
% give the final delta as the current last one in order delta vec will be
% at the length of the signal
delta(index_test:end)=0;
% the last segment is the Idx(2)
segment_ind=[segment_ind Idx(2)];
end

```

#### תשובה לשאלת 2.4:

את רוב ההסבר להבדל בין פונמה קולית ולא קולית הבנו ברקע התיוורטי. פונמה קולית מתקבלת מרעד של מתי הקול בתדר יסודי. פונמה קולית תהיה נראית כמעט מוחזורת. כמו כן היא מתאפיינת באנרגיה גבוהה. וקצב חיצית אפס נמוך.

פונה א-קולית מתקבלת מפתיחה של מתרי הקול ומעבר אויר אל עבר חלל הפה, של תנועות מסוימות גורמות לפונמה להישמע כפי שהיא. עם אנרגיה נמוכה וקצב חיצית אפס יחסית גבוהה[1].



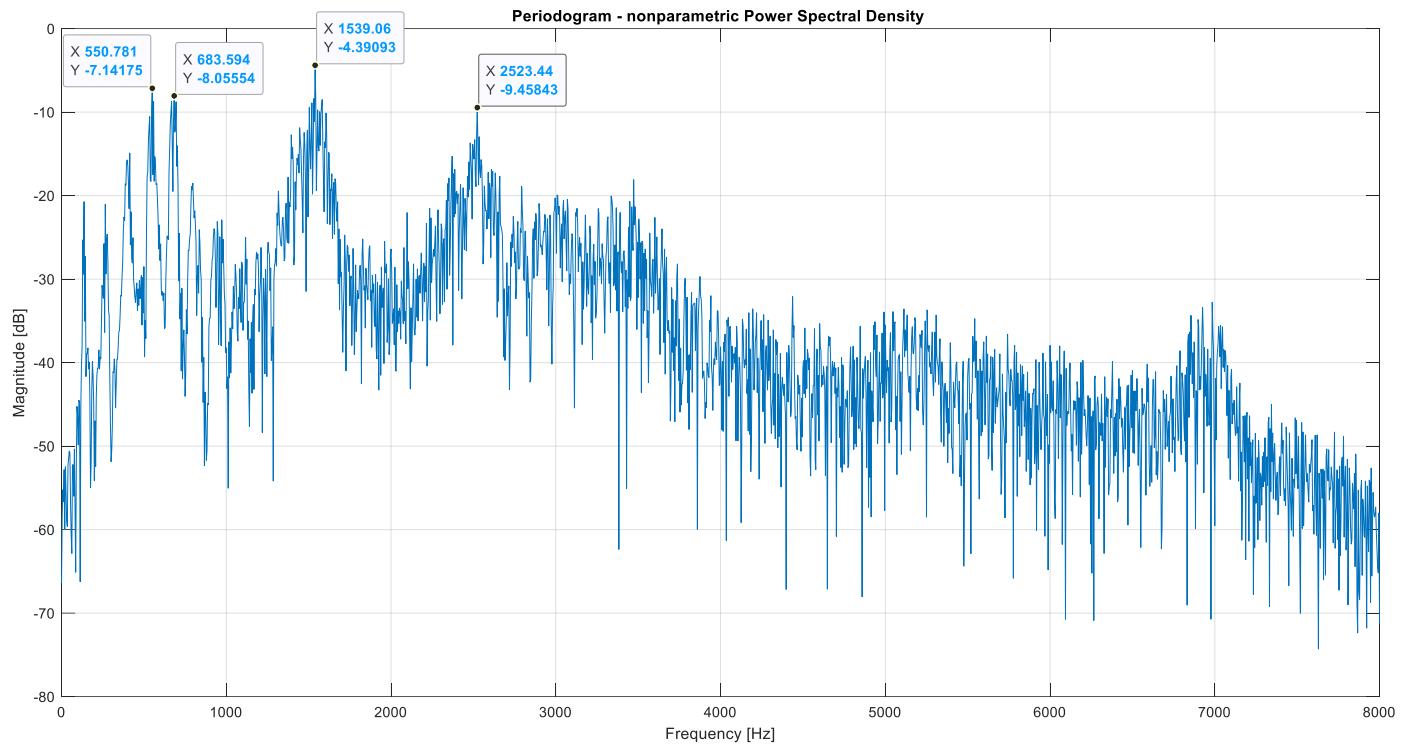
איור 8 – סגמנטציה אוטומטית על מقطع הדיבור

באיור 8 רואים את הסגמנטציה לפי האלגוריתם שתואר בספרו של סומחו ולגונה [3]. רואים בצורה יחסית טובה שהחוצאות הינן די טובות, הצלחנו להבדיל בין הfonומות השונות. כמו כן, רואים כי ישנו מקומות בהם עברנו את הסף, אך הזמן המוגדר של  $\Delta t$  מנע מזיהוי שווה של הסגמנט. כמו כן, עלול להראות שהסימונים שלנו מוטים מעט ימינה, אך צריך לזכור כי בעת המעבר בין הfonומות יש לנו מקטעים קצריים בזמן בהם תהיה חפיפה בין שתי הfonומות.

#### 4. שעורך ספקטרלי ומיציאת פורמננות

**תשובה לשאלת 3.1:**

נראה את תוצאת הperiódogramma



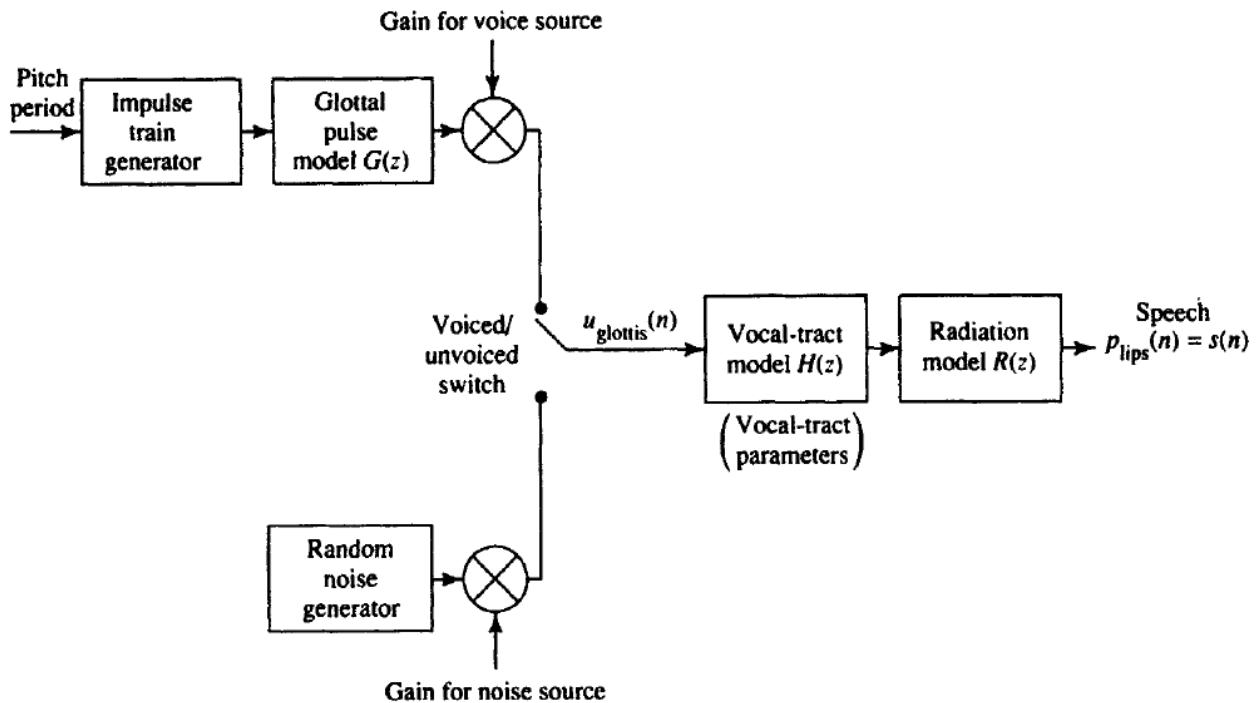
איור 9 - שיעורך לא פרמטרי לספקטראום

maiour 9 ניתן לראות כי קיבלנו שיעורך די טוב הדומה לזה שראינו באיוור 3. נשים לב כי קיבלנו 3 פורמנטות מרכזיות, כאשר הראשונה מפוצלת בין 2 פיקים כך שנΚבל

$$Formant_1 = \frac{550.781 + 683.594}{2} = 617.185 ; Formant_2 = 1539.06 ; Formant_3 = 2523.44 [Hz]$$

### תשובה לשאלת 3.2:

LPC-Linear Predicting Coding זו שיטה באמצעותה ניתן ליצור מודל בזמן בדיד. מודל זה נקרא-terminal analog model כי הן האותות והן המרכיבות הין אנלוגיות באופן שטחי. כלומר, מודל זה מנסה לקרב את אות הדיבור על סמך אות המוצא. ובעזרתו ניתן לקבל הערכה די טובה של הfonema שנאמרה. נציגו כי למאות שיטות פונמות שונות, אך צורת ה vocal tract תישאר זהה. כאן אנו לא דנים בבעה זו.



אייר 10 - מודל terminal-analog ליצירת אות דיבור

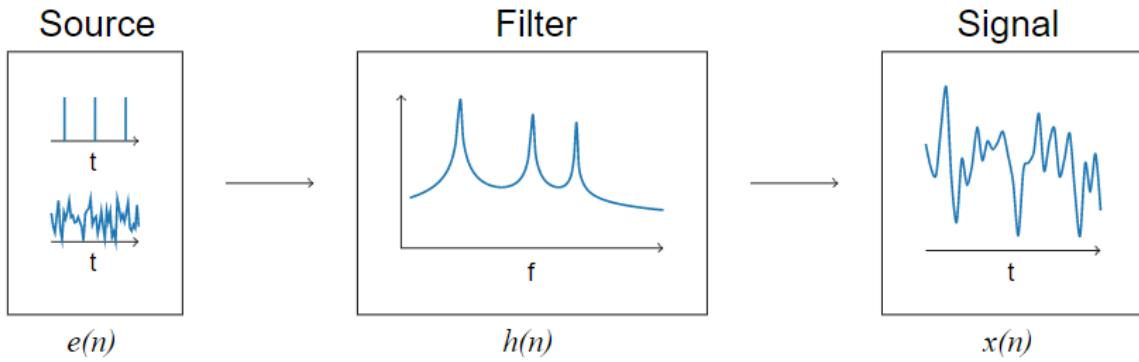
באייר 10 ניתן לראות את המודל הבסיסי הכללי לצירת אות הדיבור. אנו רואים כי הנו התמסורתה המייצגת את ה- $R(z)$  והן התמסורת של  $H(z)$  (vocal tract model) המיצגת את מעבר התדרים הגבוהים במעבר בין תוך לתוך) מאותחולות על ידי פונקציית אקסיטציה בדידה של  $u_{glottis}(n) = u(n)$  המתארת את המעברים בין הפונמות הקוליות ו-א-קוליות.

בזמן פונמה א-קולית פונקציית אקסיטציה random noise generator מקבל קלט מ- random noise generator שמאופיין ברעש קבועם בספקטרום.

בזמן פונמה קולית, אנו נקבל כי ישנו שיעורן של התדר היסודי בו נעים מוטרי הקול (pitch) היוצא את רכיבת החלמים המדמה לנו את המבנה של תנועת מוטרי הקול. כאשר הרווח בין שני החלמים יהיה התדר הבסיסי. תדרי התהודה שאוטם יוצר ה- $H(z)$  הינם פיקרים הנקראים פורמנטו.

נכונה לפחות מעט מה שקרה,

ניתן להתבונן גם בדרך מעט שונה. בה אנו מייחסים את שרשרת הפילטרים  $H(z)$  ו-  $R(z)$ , כמערכת כל קוטב אחת שנקראה לה כעת  $H(z)$ .



אייר 11 - מודל LPC [4]

באייר 11 ניתן לראות כי המשווה שמתארת את הבעה היא  $x(n) = h(n) * e(n)$ . מודל ה LPC הוא מודל מסדר P כך שכל זוג קטבים מרוכבים צמודים מנסים לשערך את תדרי הרזוננס(הפורמנוט) של הפונמה. אמנים הוא לא מדויק פיזיולוגית אך הוא יכול לתת תיאור די טוב של הפונמה. כל קווט מתאים להשיה כך שכל דגימה באות המוצא הינו השילוב של הכניסה הנוכחי  $e(n)$  ו- P הדגימות הקודמות.

(5)	$X(z) = H(z)E(z) = \frac{1}{1 - \sum_{k=1}^P a_k z^{-k}} E(z)$
-----	--

מכאן, נוכל לרשום בצורה נוספת, כתלות בדgesmotot הקודמות

$$X(z) = \sum_{k=1}^P a_k z^{-k} X(z) + E(z) \rightarrow x(n) = \sum_{k=1}^P a_k x(n-k) + e(n)$$

במודל המעניili לא ידוע לנו אופי הכניסה, לכן

$$e(n) = x(n) - \hat{x}(n) = x(n) - \sum_{k=1}^P a_k x(n-k)$$

קיבלנו מודל (AR). ראיינו כי המקדים שמביאים למינימום את שגיאת החיזוי מתקבלים על ידי סט משוואות לינאריות Yule Walker [5]

(6)	$\vec{a} = -R_{[PXP]}^{-1} \cdot \vec{r}$
-----	---

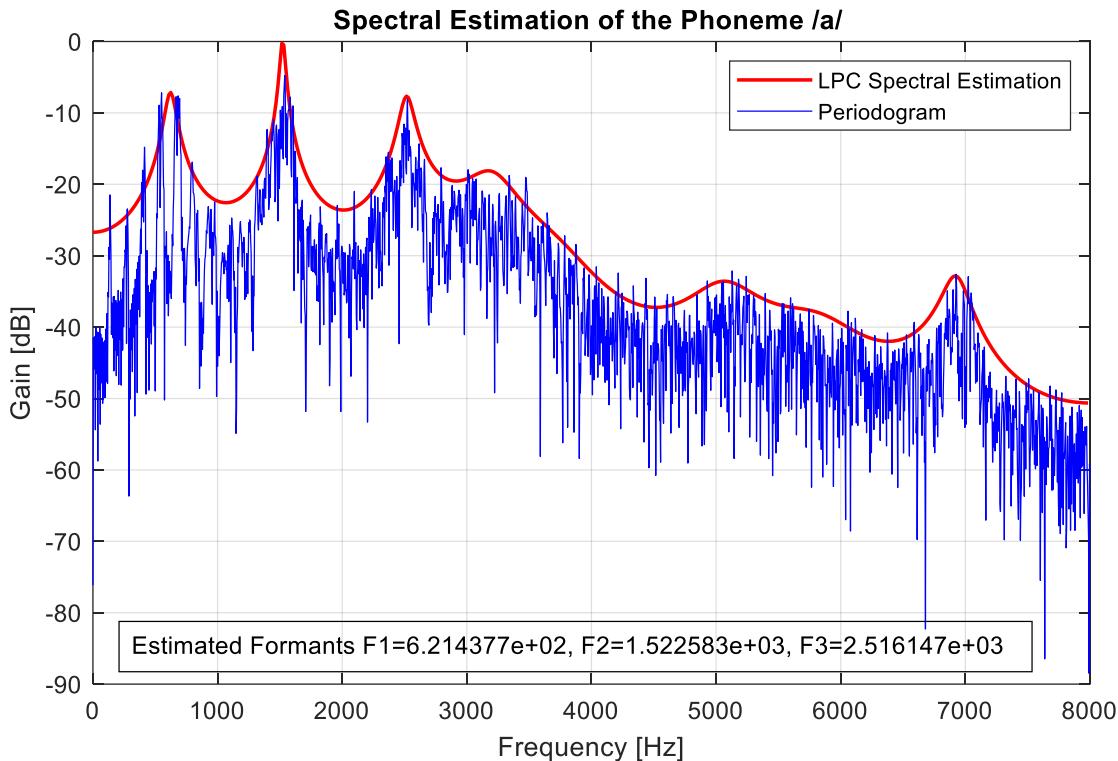
כאשר R זו מטריצת הקורלציה ו-  $\vec{r}$  וקטור אוטוקורלציה

משנקבו המקדים, ותחת ההנחה כי ההגבר הוא סטיית התקן של הרעש הלבן, בספקטרום, האות ישוערך לפי

$$S_x(e^{j\omega}) = \frac{\sigma_u^2}{|1 + \sum_{k=1}^P a_k e^{-j\omega}|}$$

כאשר  $\frac{F_s}{1000} + (2\sim 4) = P$  כאשר (2~4) נועד למדל את אפקט  $\sigma^2$  הינה שונות הרעש הלבן. נהוג להשתמש ב (2~4) כהארן [1]. בהינתן המקדמים, במעטפת הספקטרלית אנו נצפה לפיקים בפורמנטוות.

### תשובה לשאלת 3.3:



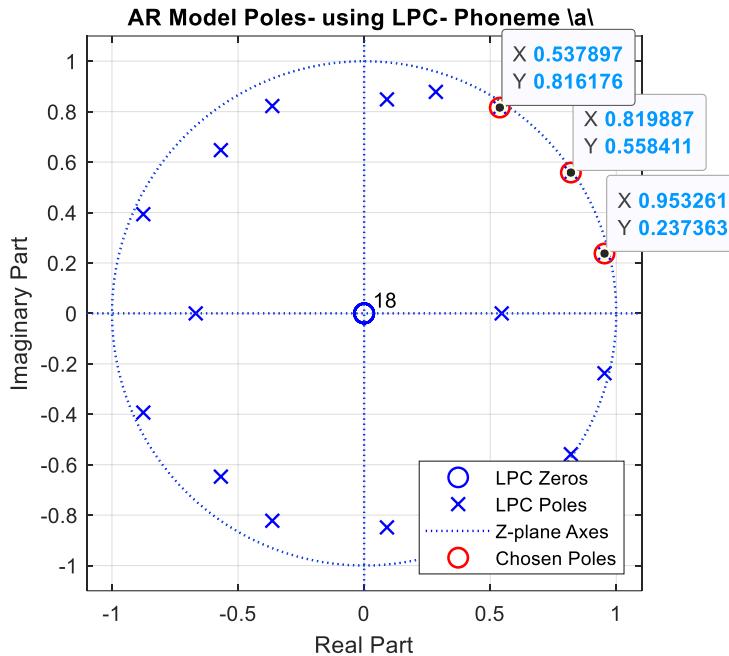
איור 12- שיעורך פרמטרי ושורך שאינו פרמטרי לספקטרום

באיור 12 ניתן לראות את התוספת של שיעורך הספקטרום הפרמטרי (אדום) אל מול השיעורך הלא פרמטרי (פריזודוגרמיה- כחול), כפי שראינו באיור 9. מצד אחד ניתן לראות כי השיעורך הלא פרמטרי נותן לנו שיעורך מפורט מאוד של הספקטרום. ניתן לראות עבור 2 הפורמנטוות הראשונות כי המרווחים בין כל שני פיקים סטטיסטיים קבוע ובעזרתו ניתן לחשב את התnder היסודי. כמו כן, עבור השיעורך הפרמטרי אנו לא יכולים לראות זאת. ניתן לראות כי השיעורך הפרמטרי מפרט לנו היבין מתרחשים הפיקים המרכזיים בספקטרום, בהתאם לכמות המקדמים שנבחרו ב-LPC. נציין כי תוצאה זו נראה יחסית טובה וכי דומה למה שראינו מן הספרות באיור 3. נציין כי אם היינו משתמשים ביוטר מקדמים אז השיעורך הפרמטרי היה הופך לדומה יותר לשיעורך הלא- פרמטרי.

### תשובה לשאלת 3.4:

כפי שראינו ברקע התיאורטי, פורמנטוות אלו תדרי התהודה של המSEN ( $H(z)$  האקוסטיים שמייצר ה tract vocal). מהתובנות באיור 12 ניתן לראות את שלושת תדרי התהודה הראשוניים:

$$\hat{F}_1 = 621.437 ; \hat{F}_2 = 1522.538 ; \hat{F}_3 = 2516.147[\text{Hz}]$$



איור 13- מפת קטבים ואפסים עבור פונמה /א/

באיור 13 ניתן לראות את  $P = 18$  הקטבים המתארים לנו את השיעורן הפרטורי של הספקטרום. לאחר והמקדים מהמודל ממשיים, אנו מסיקים כי הקטבים אמורים להופיע בזוגות קטבים צמודים. מיציבות המשן אנו מצפים כי כל הקטבים יהיו מרכזים בתוך מעגל היחידה.

בדי למצא את תדרי הפורמנטו המתאימים אנו נבצע חיפוש וסידור של לפי הקטבים הנמצאים בחצי המישור העליון של הציר ממשי. שלוש הפורמנטו הראשונות יתאימו לקטבים בעלי האמפליטודה המקסימלית. לאחר מכן, נבצע את החמרה לזוויות המתאימה לפי נוסחה

(7)	$\hat{F}_i = \frac{Fs}{2\pi} \cdot \tan^{-1} \left[ \frac{Im(P_i)}{Re(P_i)} \right]$
-----	--

טבלה 1- השוואה בין שיעורי שלושת הפורמנטו הראשונות

$F_3[\text{Hz}]$	$F_2[\text{Hz}]$	$F_1[\text{Hz}]$	
2523.44	1539.06	617.185	<b>Nonparametric PSDE</b>
2516.147	1522.538	621.437	<b>התובנות באיור 12 - parametric PSDE</b>
2516.147	1522.582	621.438	<b>התובנות באיור 13 – parametric PSDE</b>
2440	1090	730	<b>שלוש הפורמנטו הראשונות מהספרות</b>

מהתבוננות בטבלה 1 ניתן לראות כי שלושת השיטות קיבלו בקירוב את אותם התזירים מן השערוכים השונים. נציין כי תוצאות שתי הפורמנטוט הריאנסיות יחסית ורוחוקות מן התיאוריה. הסבר לכך יכול להגיע מכך שהטונציה של ההברה /a/ בעברית שונה מכך שהיא נשמעת באנגלית.

### תשובה לשאלת 3.6

```

function [h1,h2]=estimatePhonemeFormants(PhonemeSig,Fs,phonemeName)
% estimatePhonemeFormants takes the PhonemeSig and uses the Power spectral
% density by calculating the P coefficients using AR LPC model estimation of
% A general discrete-time model for speech production.
% Input:
% PhonemeSig - one phoneme (after pre-processing)
% Fs - sampling frequency
% phonemeName - a string with the phoneme name
% OUTPUT:
% h1 - handle for spectral estimation graph with Formants values
% h2 - handle for zero-pole map with chosen poles
rng default

[Peri,omeg]=periodogram(PhonemeSig,[],'onesided');

P = Fs/1000 +2;
% g-variance of the prediction error
[aa,gg]=lpc(PhonemeSig,P);
std_g = sqrt(gg);
M = length(PhonemeSig);
%use the Random Noise Generator on samples
add_noise_to_filt=filter(1,aa,std_g*randn(M,1));
% Autoregressive all-pole model parameters – Yule-Walker method
[aa1,gg1]=aryule(add_noise_to_filt,P);
std_gg1 = sqrt(gg1);
% add the noise to the parametric periodogram
[hh,omegg1]=freqz(std_gg1,aa1);

h1 = figure;
plot((Fs/(2*pi))*omegg1,20*log10(abs(hh)), 'LineWidth',1.4, 'Color', 'r');
xlabel 'Frequency (Hz)' ; ylabel 'Magnitude [dB]' ; hold on; grid on
plot((Fs/(2*pi))*omeg,10*log10(Peri), 'Color', 'b')
xlabel 'Frequency [Hz]' ; ylabel 'Gain [dB]'
hold on

% find formants:
rts = roots(aa1);
%Because the LPC coefficients are real-valued, the roots occur in complex
% conjugate pairs. Retain only the roots with one sign for the imaginary
% part and determine the angles corresponding to the roots.
rts = rts(imag(rts)>=0);
[~,Poles_idx]=maxk(abs(rts),3);
angz = atan2(imag(rts),real(rts));
Formantss = angz(Poles_idx).*(Fs/(2*pi));
Formantss = sort(Formantss);
hold on
title(sprintf('Spectral Estimation of the Phenome %s',phonemeName))
legend('Periodogram','LPC Spectral Estimation')
annotation(h1,'textbox',[ 0.15 0.09 0.09 0.1], 'String',...
{sprintf('Estimated Formants F1=%d, F2=%d, F3=%d',Formantss(1),Formantss(2),...
Formantss(3))}, 'FitBoxToText', 'on');grid on

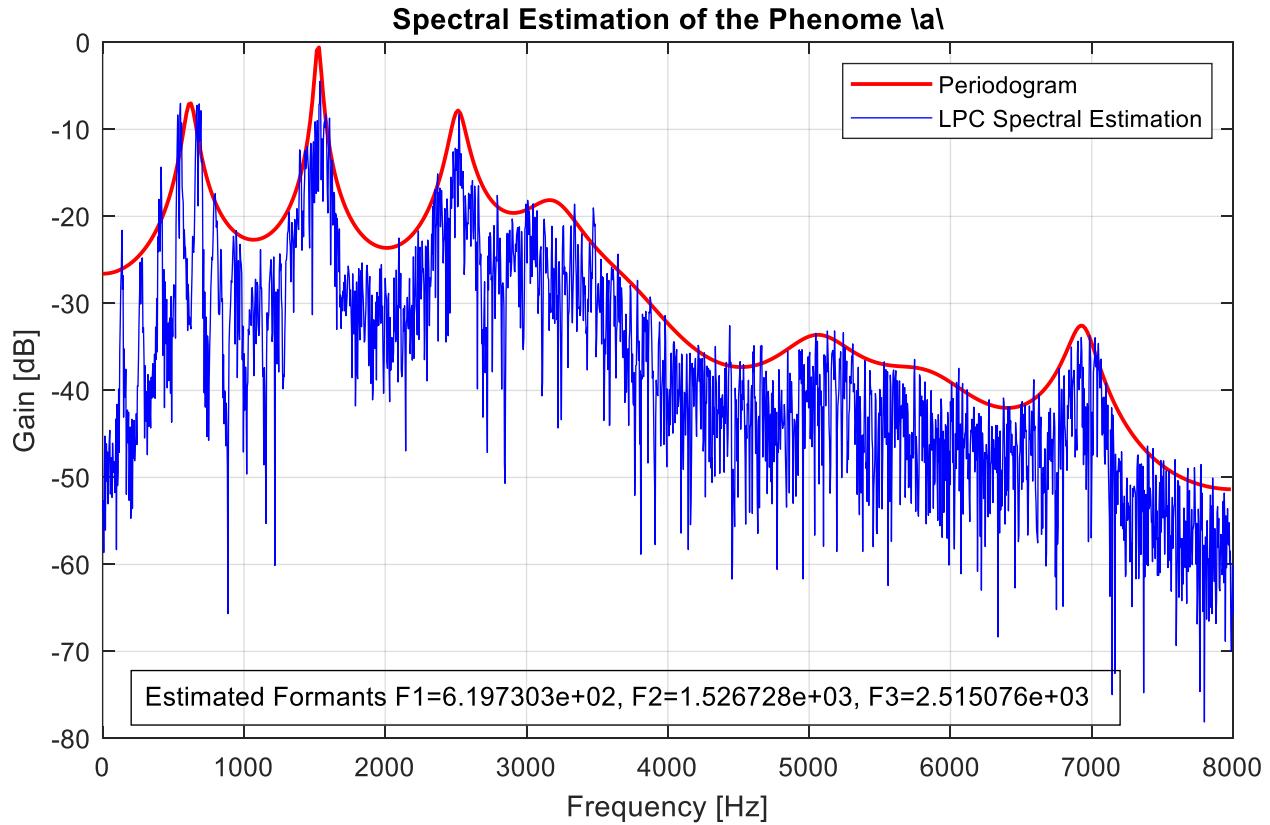
h2 = figure;
[z,p,k] = zplane(1,aa1); hold on; grid on
findzeros = findobj(z, 'Type', 'line'); findpoles = findobj(p, 'Type', 'line');
findk = findobj(k, 'Type', 'line');
set(findzeros, 'Color', 'b', 'linewidth', 1.1, 'markersize', 9);
set(findpoles, 'Color', 'b', 'linewidth', 1.1, 'markersize', 9);
set(findk, 'Color', 'b', 'linewidth', 1.1, 'markersize', 9);
hold on

```

```

[z1,~,~]=zplane(rts(Poles_idx(1:3)));
findzeros1 = findobj(z1,'Type','line');
set(findzeros1,'Color','r','linewidth',1.2,'markersize',9);
xlim([-1.1 1.1]);ylim([-1.1 1.1]);grid on
title(sprintf('AR Model Poles- using LPC- Phoneme %s',phonemeName))
legend('LPC Zeros','LPC Poles','Z-plane Axes','Chosen Poles','Location','southeast')
end

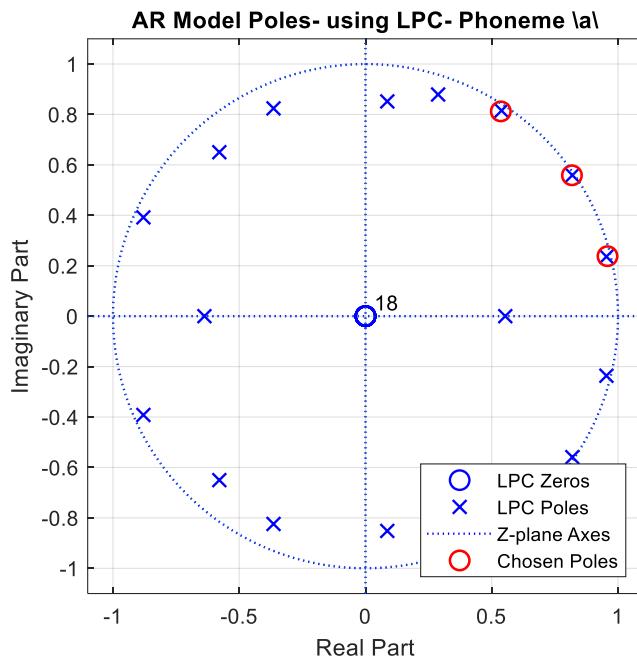
```



איור 14- מוצא הפוןקציה estimatePhonemeFormants. שיעורן פרמטרי ולא פרמטרי של אות ושולוש הפורמנטאות הראשונות לפונמה /a/

まいور 14 にて見られるのは、fonktsiya の出力です。estimatePhonemeFormants を使用して、音素 /a/ の最初の三つの形式音を計算します。この結果は、vocal tract の物理的構造に適応する三次元的な形で示されています。この図では、3000[Hz] 附近に主なピクルがあり、その後 7000[Hz] 付近に小さなピクルがあります。これらの値は、マイナス 60dB からマイナス 20dB の範囲内にあります。また、この図には、LPC のスペクトル推定と周波数領域での信号の振幅を示す線形グラフが併せて表示されています。

マイアード 14 の下部には、計算された形式音の値が示されています。



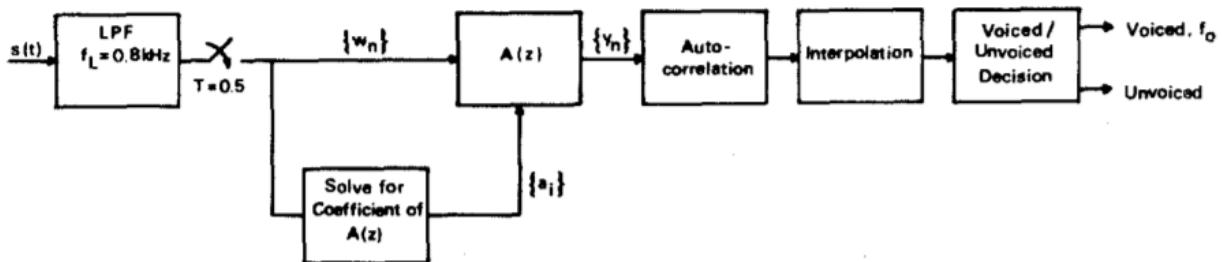
איור 15- מפת קטבים וඅපසים מן הפונקציה estimatePhonemeFormants לפונמה /a/

ניתן לראות מאIOR 15 כי הקטבים שנבחנו נראים כמוו הקטבים הקרובים ביותר אל מעגל היחידה, שכן לכל קווטב יש את צמוד לו המתאים לנו את ספקטרום התדרים השלילי ולכן אנו לא התייחסנו אליהם.

### תשובה לשאלת 3.7:

Pitch הינו התדר הבסיסי בו נעים מתרי הקול, כאשר אנו נראה אותו בהפרש הזמנים בין שני פיקים סמוכים בספקטrogramma. ראיינו אותו בצורה די ברורה באIOR 9, עד לתדרים של כ  $[kHz] 1$ . כמובן שזה רלוונטי לפונמות בוחן מתרי הקול פעילים, ככלומר לפונמות קוליות.

ציג דיאגרמת דלוקים לאלגוריתם SIFT



איור 16- דיאגרמת בלוקים לאלגוריתם SIFT [6]

באIOR 16 ניתן לראות את השלבים העיקריים באלגוריתם. תחילת השמע מסונן על ידי מסנן LP עם תדר קיטועון של  $[kHz] 0.8$ . לאחר מכן, מחושבים 5 המקדמים הראשונים של האוטוקורלציה  $\{ \rho_j \}_{j=0}^4$  אותן המתאימים לחלון זמן של כ 30 מילישניות. אנו מקבלים סט משוואות לינאריות המנסה לモזר את השגיאה הריבועית.

(8)

$$p_j = \sum_{i=1}^4 a_i p_{i-j} ; j = 1, \dots, 4$$

כאשר את מקדמי האוטוקורלציה  $\{a_i\}$  ממשוואת 8 אנו מקבלים מהכפלת באות בעצמו מוז.

(9)

$$p_j = \sum_{n=0}^{N-1-j} w_n \cdot w_{n+j} ; j = 0, \dots, 4$$

על מנת לקבל את ערכי  $\{a_i\}$  ממשוואת 8 משתמשים באלגוריתם Levinson Durbin(LD) [1]. המנצלת את העובדה שמטריצת האוטוקורלציה היא סימטרית ובעל מבנה טופלי, (כל האלכסונים הראשיים שוויים).

בכתיב מטריצי:

$$R = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \\ p_1 & p_0 & p_1 & p_2 \\ p_2 & p_1 & p_0 & p_1 \\ p_3 & p_2 & p_1 & p_0 \end{bmatrix} \cdot \left( p = - \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \right) = \vec{a}$$

כעת, אנו יודעים מה הם ערכי  $a_i$ . נוכל כתע להציג את הפילטר ההיפוך

(10)

$$A(z) = 1 + \sum_{i=1}^4 a_i z^{-1}$$

לאחר שיעורך פרמטרי של הספקטראום אנו יכולים לבצע אוטוקורלציה על האות  $\{y_n\}$ . לבסוף, מבוצעת החמרה של התנאים למציאת התדר היסודי (שלב- האינטרפולציה באירור 16), שם מאפסים מכל הדגימות עבורות האוטוקורלציה שלילית. עברו פונמות קוליות, בהן רלוונטי לחשב *Pitch*, ערכו המספרי ינוע בתחום  $[70 - 300] \text{ Hz}$ . כמו כן, ידוע שהרוב הגברים יש טו נמוך יותר כך שם *Pitch* יהיה נמוך יותר ביחס לבנות(שלרוב, הטונצייה גבוהה יותר) כך *sheh* גבוהה יותר.

## 5. מיצוי מאפיינים

### תשובה לשאלת 4.1:

נוסחה לחישוב האנרגיה של האות

(11)

$$E(n) = \frac{\sum_{m=-\infty}^{\infty} (x(m) \cdot w(m-n))^2}{\sum_{m=-\infty}^{\infty} w^2(m-n)}$$

כאשר לפונמות הקוליות תהיה אנרגיה גבוהה ולפונמות א- קוליות האנרגיה תהיה נמוכה[1].

#### תשובה לשאלת 4.2:

```

function EnergySignal=calcNRG(framedSignal)
% calcNRG calculate the the energy of each frame
% framedSignal - a matrix of the framed signal, after preprocessing
% OUTPUT:
% EnergySignal - a column vector of the energy values of the signal

% take the length of each frame
[~,N]=size(framedSignal);
window=rectwin(N);
% sum( _ ,2) in order to sum each frame
EnergySignal=(sum((framedSignal).^2,2))/sum(window.^2);
end

```

#### תשובה לשאלת 4.3:

חישוב ZCR יבוצע על מקטע על כל מסגרת בנפרד. נסחטו מתוארת באופן הבא

$$(12) \quad Z(m) = \frac{1}{N} \sum_{n=m-N+1}^m \frac{|sgn(s(n)) - sgn(s(n-1))|}{2} \cdot w(m-n)$$

כאשר  $sgn(s(n)) = \begin{cases} 1 & s(n) \geq 0 \\ -1 & s(n) \leq 0 \end{cases}$

שימוש במדד זה יכול לעזור לנו בזיהוי בין פונומות קוליות ופונומות א-קוליות. אנו יודעים כי בפונמה קולית, יש פחות חציית אפסים. לעומת זאת פונמה א-קולית המורכבת מהרבה חציית אפסים [1].

#### תשובה לשאלת 4.4:

```

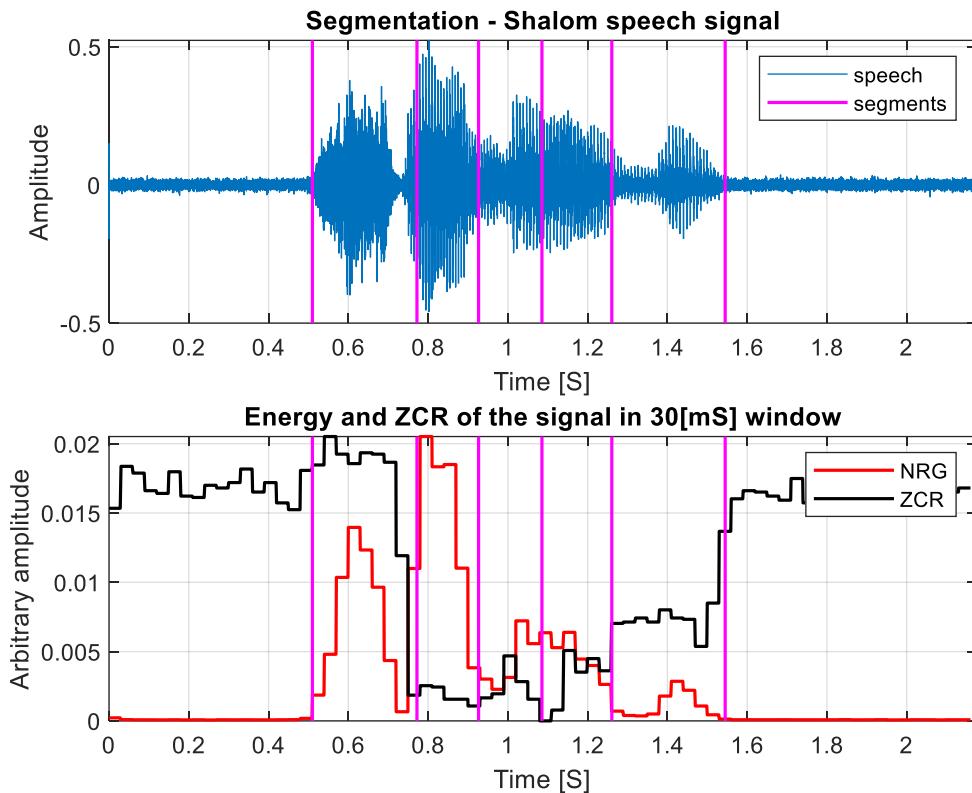
function ZeroCrossingSignal = calcZCR(framedSignal)
% calcNRG calculate the the energy of each frame
% framedSignal - a matrix of the framed signal, after preprocessing
% OUTPUT:
% ZeroCrossingSignal - a column vector of the zero-crossing values of the signal
[M,N]=size(framedSignal);
framedSignal1 = framedSignal;
% create delay
framedSignal2 = cat(2,zeros(M,1),framedSignal(:,1:end-1));
ZeroCrossingSignal = (1/(2*(M*N-1)))*sum( abs( sgn( framedSignal1 ) - ...
sgn( framedSignal2 ) ),2 );
end

function y = sgn(x)
%SGN Modified signum function.
% For each element of X, SIGN(X) returns 1 if the element
% is greater than or equal to zero, and -1 if it is
% less than zero.

y = (x>=0) + (-1)*(x<0);
end

```

#### תשובה לשאלת 4.5:



איור 17- סגמנטציה אוטומטית (מעלה) מדדי אנרגיה ו- ZCR (מטה)

まいور 17 ניתן לראות כי בחלק בו אין כל דיבור, האנרגיה אפסית לעומת זרימת גבואה. ככלומר מבחינת המקטע השקט הצלחנו להציגו לממדים הנראים הגיוניים. מהתבוננות בפונמות הא-קוליות (/sh/-/m/), רואים כי ZCR הינו גבואה מאוד בפונמה /sh/ ומעט נמוך יותר (אך עדין גבואה) בפונמה /m/. האנרגיה של פונמה /sh/ אמנים יחסית גבואה, אך היא עדין תהיה נמוכה יותר מערך ה-ZCR של הפונמה. כמו כן, רואים כי האנרגיה של פונמה /m/ הינה נמוכה מאוד, מה שתואם את התיאוריה. באופן כללי אנו רואים כי ערכי ZCR גבוהים יותר מאשר ערכי האנרגיה של הפונמות הא-קוליות. מה שיעזר לנו לאבחן בין פונמה קולית וא-קולית.

בהתבוננות בפונמות הקוליות (/a/, /L/, /o/) ניתן לראות שלפונמה /a/ יש אנרגיה גבוהה ו-ZCR נמוך. לפונמה /L/ יש אנרגיה המשתנה ולא ניתן לומר באופן כללי אם גבואה או נמוך.

#### תשובה לשאלת 4.6:

```
function [FeatsVector,Feat_title]=FeatExt(Phoneme,Fs,framedPhoneme)
%FeatExt the phoneme and give calculates all the features that is required
% Phoneme - one phoneme (after pre-processing)
% Fs - sampling frequency
% framedPhoneme - the phoneme after framing
% OUTPUT:
% FeatsVector - 1X24 vector of features of the analyzed phoneme
% Feat_title - 1X24 cell array of the names of the calculated features
P = Fs/1000 +2;
FeatsVector = [];
Feat_title = [];

% 1 - mean energy
Mean_NRG = mean(calcNRG(framedPhoneme));
```

```

FeatsVector = [FeatsVector; Mean_NRG];
Feat_title = [Feat_title; {'mean energy'}];
% 2 - mean ZCR
Zero_mean_Crossing_Signal = mean(calcZCR(framedPhoneme));
FeatsVector = [FeatsVector; Zero_mean_Crossing_Signal];

Feat_title = [Feat_title; {'mean ZCR'}];
% 3- Pitch
[Pitch,~] = sift(framedPhoneme,Fs);

FeatsVector = [FeatsVector; Pitch];
Feat_title = [Feat_title; {'pitch'}];
% 4 - LPC coefficients
Lags = 2;
[Formamt,outAR] = formants(Phoneme,P,[],Fs,2*Fs/Lags);
FeatsVector = [FeatsVector; outAR(2:end)'];
for i=1:18
    Feat_title = [Feat_title; {sprintf('LPC coefficient #%d',i)}];
end
% 5 - first 3 formants
FeatsVector = [FeatsVector; Formamt(1); Formamt(2); Formamt(3)];
for i=1:3
    Feat_title = [Feat_title; {sprintf('Formant #%d',i)}];
end
end

```

לאחר מימוש הפונקציה קיבלנו את סט הפיצרים. כאשר הפעילו את הפונקציה על כל אחת מן הפונומות שנמצאו. כל וקטור המתאים לפונמה הוא בגודל 24 ומכיל בתוכו את האנרגיה הממוצעת, את ZCR הממוצע, Pitch הממוצע, 18 מקדמי LPC (כלומר המקדים ה 2 עד ה 19) ולבסוף את 2 הפורמנטוות המשוערכות, בעלות האנרגיה הגבוהה ביותר.

טבלה 2 - הוצאת מאפיינים מאות השמע

Phoneme	\Sh\	\A\	\L\	\O\	\M\
Mean ENG	0.0072	0.0152	0.0043	0.0051	0.001
Mean ZCR	0.0708	0.0433	0.0458	0.0476	0.0369
Pitch	219.178	150.9434	79.602	99.3289	347.8261
LPC	In Model.mat				
Formants	570.071 1584.198 2721.34	622.077 1525.19 2517.314	475.059 1410.176 2496.312	527.065 1051.131 2471.308	572.071 1436.179 3267.408

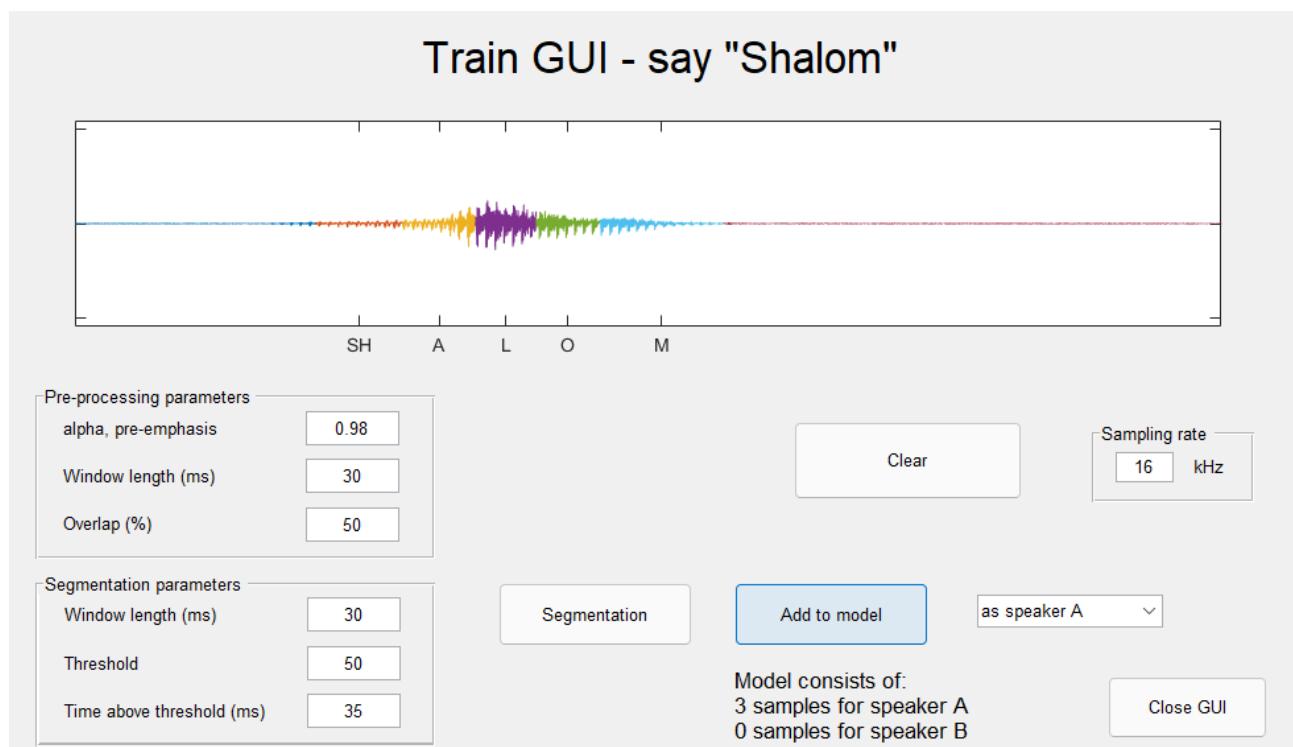
טבלה 2 ניתן לראות את רוב המאפיינים שהוציאו מכל אחד מן הפונומות. ניתן לראות כי בפונומות הא-קוליות אנו מקבלים גובהה רק עבור /sh/. וכן בכל הפונמות הקוליות רואים כי ה ZCR יחסית גבוהה, בשונה ממה

שראינו בספרות. כמו כן רואים כי האנרגיה של הפונמה הקולית הרכונה היא הגבוהה ביותר. כמו כן, האנרגיה של שתי הפונמות הקוליות האחוריות יחסית נמוכה (לעומת /sh/, למשל). אנו רואים בנוסף כי התדר היסודי של שלושת הפונמות הקוליות (בחון רלוונטי בכלל לחשב pitch) הינו יחסית גבוה בפונמה /a/ ומעט נמוך יותר בשתי הפונמות הקוליות האחוריות.

## 6. בניית המודל – יצירה בסיס הנתונים

בחלק זה, אנו יוצרים את בסיס הנתונים. העברנו את כל הפונקציות הרשומות מעלה לתיקייה של Train\_Gui. נציגו כי שינינו את הסך האנרגיה ב FindWordIndex לערך המוצע בפרוטוקול על מנת לzechות את הפונמות באופן תקין. בחרנו לעבוד בחלון זמן של 30 מילישניות עם סף של  $50 = \eta$  וזמן מעבר אחרי הסף הוא 35 מילישניות.

בחרנו שמייכאל יהיה דובר A ומור תהיה דובר B.



איור 18- תיעוד תהליך הוספת הקלטות למודל- דובר B

באיור 18 ניתן לראות את תהליך ההקלטה של אימון המודל, לאחר חיצתה על כפטור סגמנטציה. ניתן לראות באיור כי המודל מצליח להזות את חמישת הפונמות, כפי שרצינו. לאחר שאנו מודדים כי אכן התקבלו לנו 5 פונמות מן הסגמנטציה, אנו מוסיפים את הקלטה לדובר. אופן ההוספה למודל הוא יצרה של וקטור מאפיינים בשורה נוספת בתוך המערך Model.

## 7. ניתוח STFT (ספקטrogramma)

### תשובה לשאלת 6.1:

משתמשים בשיטה זו על מנת לנצל את השינויים הזמניים המתארחים באוט בצורת אנליזת זמן-תדר. באופן כללי, אותן דיבור לא סטציונירי וכן הסטטיסטיות שלו משתנות כפונקציה של הזמן. אם נעבד בחולנות זמן המאפיינים זמינים של פונמה (כ 30 מילישניות) נוכל לראות את התנהגות זמן תדר של אותן, תוך הנחת

הסטציונריות (התכונות הספקטרליות לא ישתנו המון לאורך המקטע הנוכחי). ישנה גם אופציה לחפיפה בין החלונות השונים (התקנות הספקטרליות לא ישתנו המון לאורך המקטע הנוכחי). ישנה גם אופציה לחפיפה בין החלונות השונים

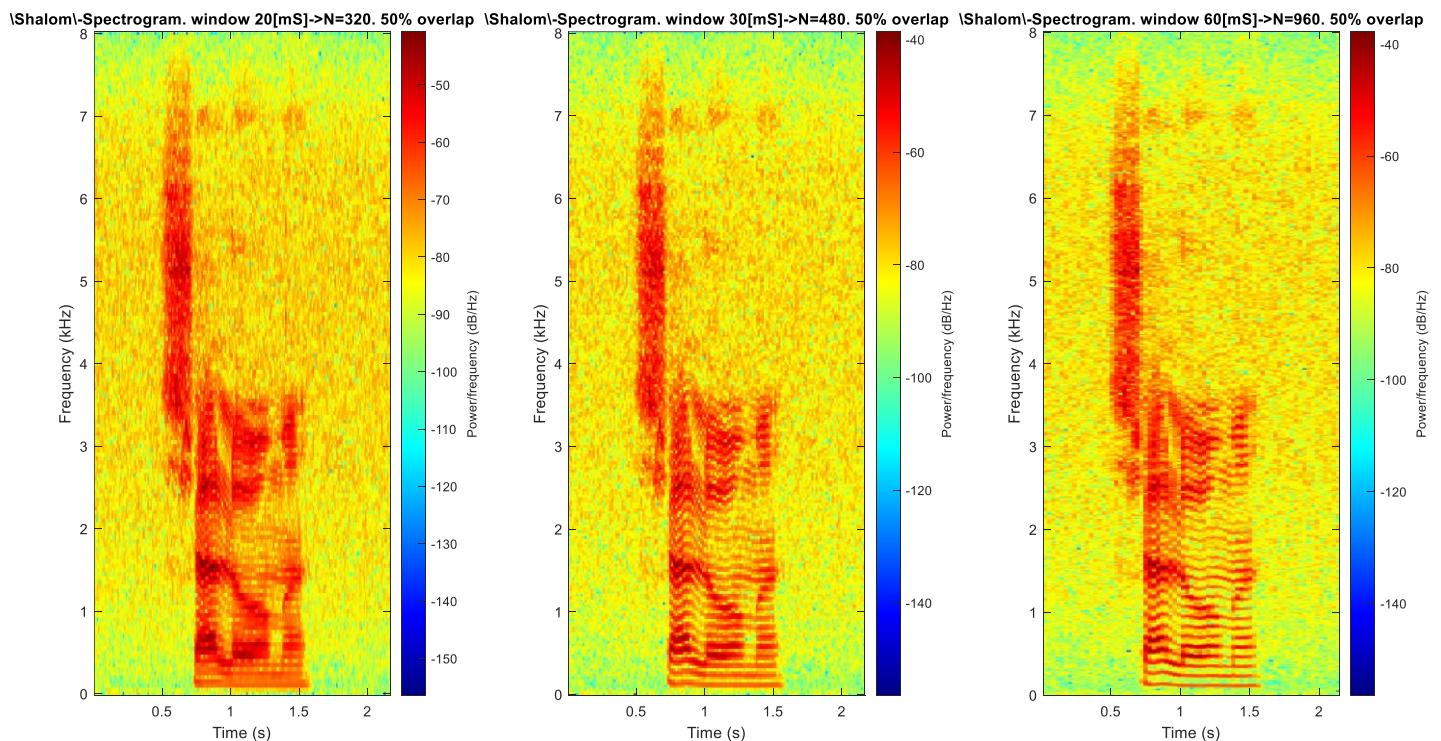
(13)	$X[m, k] = \sum_{n=-\infty}^{\infty} x(n)w(n-m) \cdot e^{\frac{j\omega}{N}kn}$
------	--

כאשר  $(n)x$  הינו האות המקורי ו  $(m-n)w$  הינו החלון הרץ.

מציגים אותו באמצעות ספקטrogramma. עבור חלונות זמן גדולים, הרזולוציה בתדר גבוהה אך יהיה קשה יותר לזהות שינויים זמניים. עבור חלונות זמן קטנים, הרזולוציה בזמן תגדר, אך בתדר תקען. כמו כן, שימוש בחפיפה בחלונות הזמן מאפשר לראות טוב יותר את השינויים הזמניים ללא כל שינוי ברזולוציית התדר. הגדלת מס הדגימות לכל DFT יגדיל את הרזולוציה בתדר. [7]

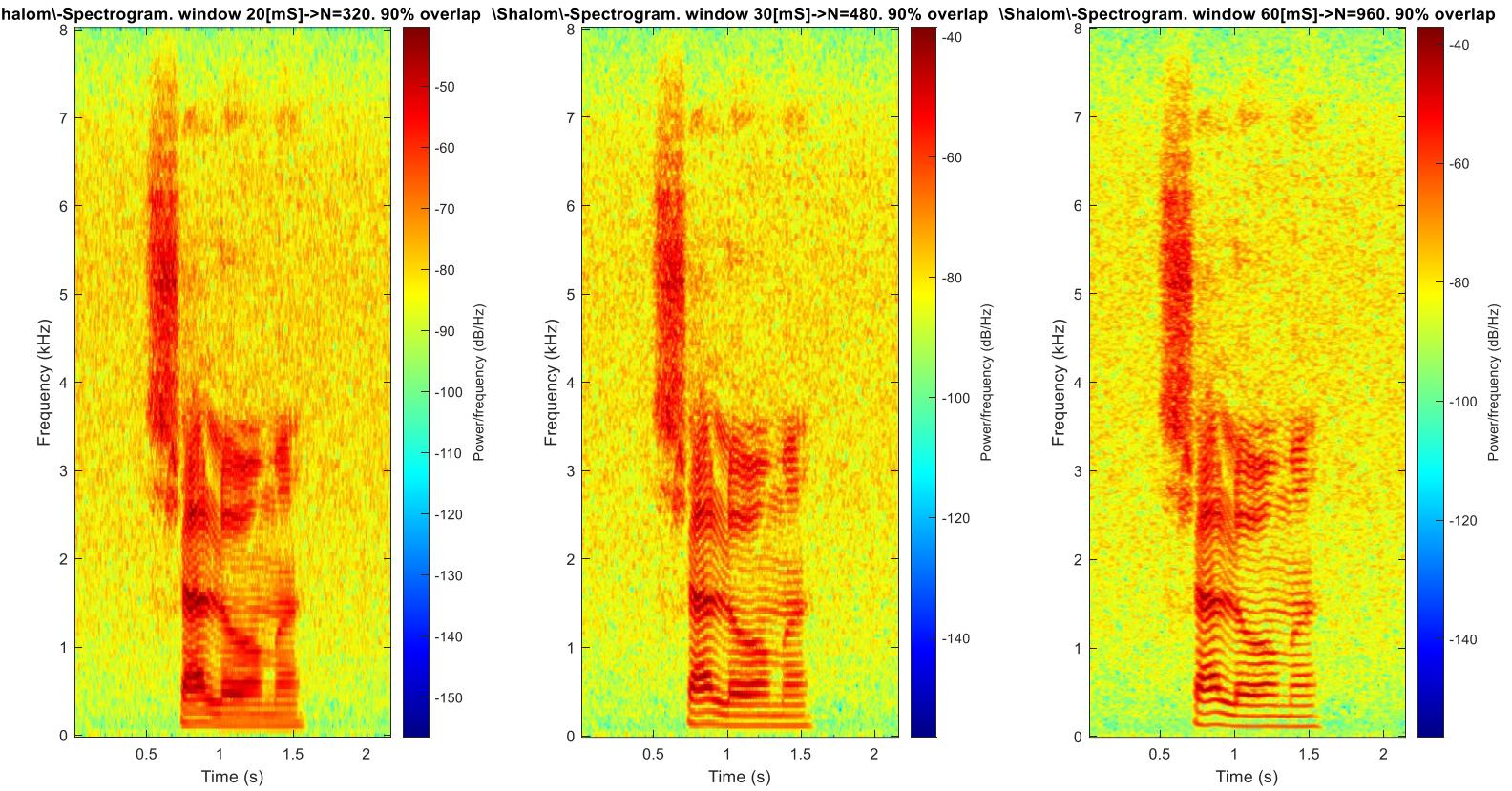
## תשובה לשאלת 6.2

נציג את הספקטrogramma של המילה "שלום"



איור 19- ספקטrogramma בחלונות זמן שונים. 50% חפיפה

maiour 19 ניתן לראות 3 ספקטrogrammas של המילה "שלום". ניתן לראות את ההפרדה שיש לנו בין הfonema הראשון /sh/ לבין שאר הfonומות, אנו רואים זאת סביר החצי שנייה בכל אחד מן הספקטrogrammas. כמו כן, ניתן לראות בחלונות זמן של 30 מילישניות באופן יותר ברור את הפורמנטות של ההבהרות הקוליות.



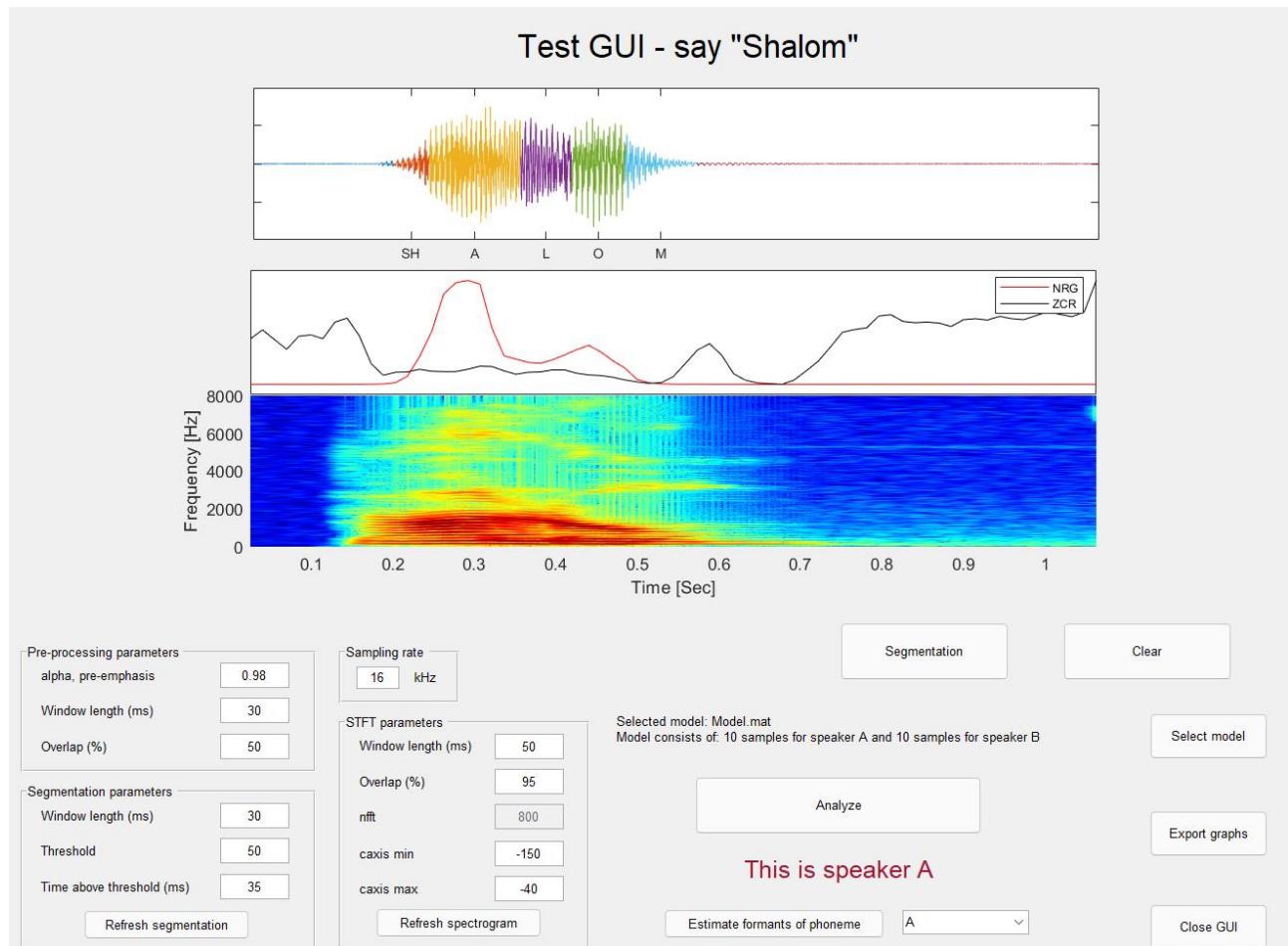
איור 20 – ספקטrogramma בחלונות זמן שונים. 90% חיפוי

באיור 20 ניתן לראות את הספקטרוגרמות עבור חיפוי זמן של 90%. עבור כל אחד מחלונות הזמן אלו רואים שלא איבדנו מן הרזולוציה התדרית. אך מבחינות זמן אלו רואים בצורה מעט יותר ברורה היכן משתנות התכונות התדריות של המילה, ככלומר באיזה רגע מתחלפת הפונמה. אלו מסיקים לכך שכדי לשמר את הרזולוציה התדרית אלו נבצע FFT באורך החלון שבו בחרנו לעבוד (במקרה שלנו עושים רושם של חלון זמן של 50 מילישניות יכול לתת את התוצאות הטובות ביותר), אך גודיל מאוד את החיפוי הזמן של החלונות.

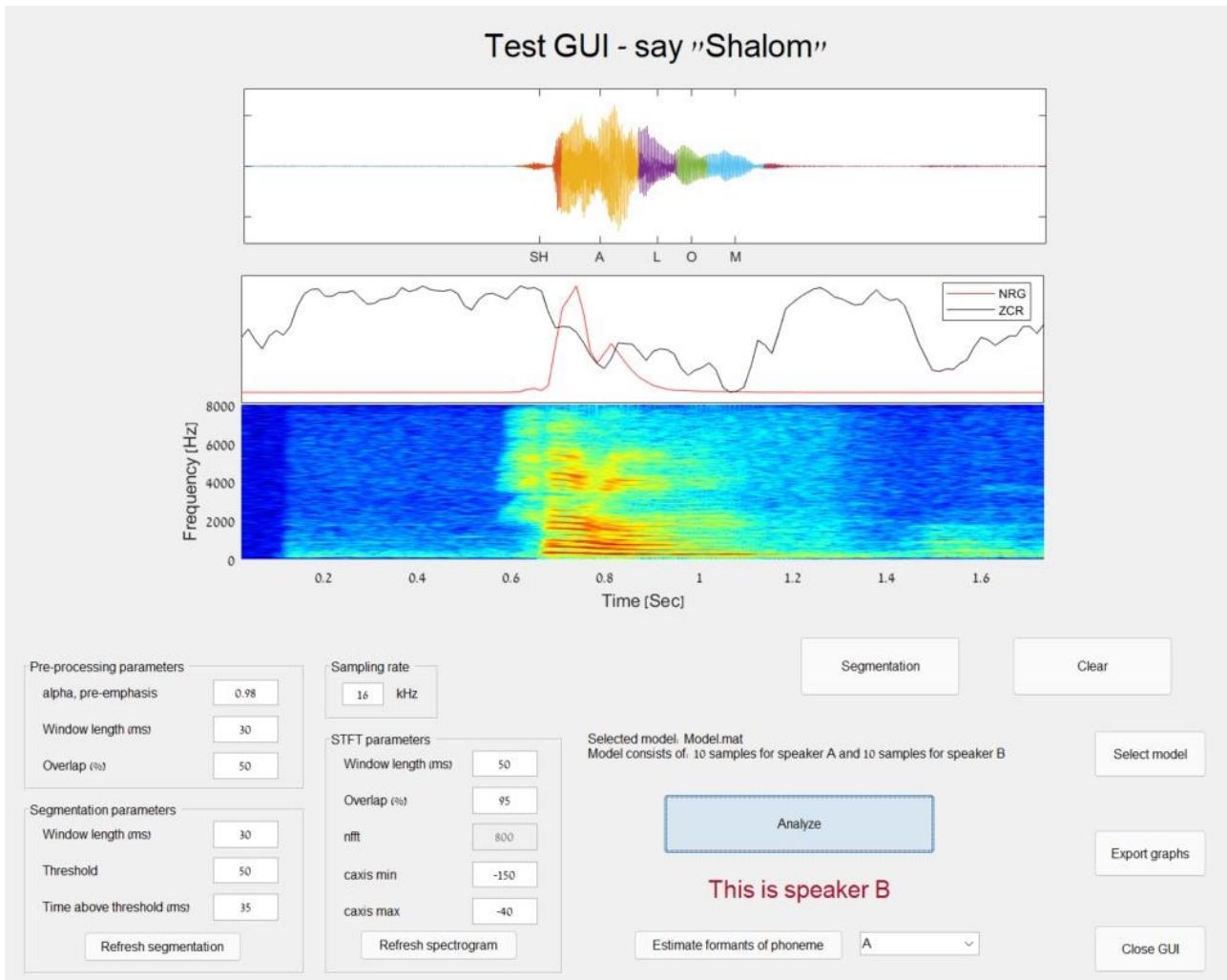
## תשובה לשאלת 7.5:

לצורך זיהוי הדוברים אנו נוערים ב Test GUI. אנו פותחים אותו ובוחרים בפרמטרים אותם רצינו לעבוד בשלב ההקלוטות. כמובן, לצורך מיפוי פיצירים בחילוניות זמן של 5 מילישניות עם חפיפה של 50%. ולצורך ביצוע הסגמנטציה בחילונו לעובך עם חלון זמן באורך של 30 מילישניות עם ספ' של 50. כמו כן, הזמן המינימאלי אותו השגיאה הייתה צריכה לעבור היא 25 מילישניות. נזכיר כי מיכאל הוא דובר A ומור היא דובר B.

נציג כעת את הממצאים פעמי אחד עבור כל אחד מהדוברים, ובטבלה מטה נרכז את הזיהויים של המערכת שבנוינו.



איור 21 – סיווג דוברים – דובר A (מיכאל) Test 1



איור 22 – סיווג דוברים – דובר B (מור) Test 5

באיור 22 ניתן לראות כי המערכת אכן מצליחה לזהות שזהו דובר B. כמו כן, מוצג לנו גраф במרכז המציג את ZCR ואת האנרגיה של החלונות לפי 5 מילישניות.

ນີ້ຈະແກ່ຕົວຕັລະ ສອງຂອງມີຄວາມ

ຕັລະ 3 – ຊື້ວິດີ ດົບຣິມ ໃຫ້ ຍິມ ມູດລ

Record Number	classification Speaker A	classification Speaker B
1	A	B
2	A	B
3	A	B
4	A	B
5	A	B

מطلبה 3 ניתן לראות כי המערכת הצליחה לזהות באופן מובהק מתי מדובר כל דבר. זאת כמובןouri שהצליחנו לסוג 5 פונומות עבור כל אחת מן הקלטות.

#### תשובה לשאלת 7.6:

המודל אותו אנו טענים בינוי ממטריצה בגודל 10 שורות כאשר כל שורה מאופיינית בסופר-וקטור המורכב משרשור של וקטור המאפיינים בגודל 24. לעומת זאת, כל שורה מתאימה ל 5 פונומות כך שאורך שורה אחת היא 120. הפונקציה classifySpeaker מבצעת תחילה ממוצע וסטיית תקן על כל אחת מן העמודות בנפרד. לאחר מכן, מייצרים עבור כל נקודה של הווקטור אותו אנו הקלינו ברגע זה באמצעות TEST GUI, את השיעורץ שלו בהנחה והוא מגיעה מהתפלגות נורמלית עם ממוצע וסטיית תקן כפי מתאים למיקום ווקטור מן המודל.

$$(14) \quad f_x = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \cdot e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma}\right)^2}$$

במשואה 14 ניתן לראות את ההשלה שבוצע עבור כל אחד מן הכנסיות המתאימות לסופר-וקטור המשרשת את וקטור המאפיינים לחמשת הפונומות. כל אחד מן התאים מוחזר כשהוא מתאים לאיבר מפונקציית צפיפות של משתנה המתפלג נורמלי עם הממוצע וסטיית התקן המתאימה.

על כל אחד מן הווקטוריים מבוצע LOG בסיס 6, זאת על מנת לחשב את ערכי הסף של x. לאחר ביצוע הלוגריתם מבוצעת סכמה כדי לראות האם אנו בסף המתאים לדבר A או B.

#### תשובה לשאלת 7.8:

```

function exportGraphs(folder_name,Signal,Fs,phon,seg_ind,STFTwinLength, ...
    STFToverlap,STFTnfft,STFTcmin,STFTcmax,NRG,ZCR,Flag)
% • folder_name - full address of the folder to save the files in
% • Signal - the recorded signal
% • Fs - sampling frequency
% • phon - an array of chars, representing the phonemes
% • seg_ind - result of the segmentation process
% • STFTwinLength - window length for the STFT (in samples)
% • STFToverlap - overlap length for the STFT (in samples)
% • STFTnfft - nfft length (in samples)
% • STFTcmin - minimum value for color axis scaling
% • STFTcmax - maximum value for color axis scaling
% • NRG - energy signal
% • ZCR - ZCR signal
% • Flag - indicates which graphs to generate.

alpha=0.999;
WindowLength=30*10^-3; % 30 [mS] window
Overlap=50; % 50% overlap
[ProcessedSig,FramedSig]=PreProcess(Signal,Fs,alpha,WindowLength,Overlap);

Idx=FindWordIdx(FramedSig,Fs,WindowLength,Overlap);
t=[0:length(Signal)-1]*1/Fs; % time vector

N=(30*10^-3)*Fs; % number of samples in each frame
Overlap_sam= ((Overlap)*N)/100;

% \sh\

```

```

Phoneme_sh=ProcessedSig(seg_ind(1):seg_ind(2));
framed_Phoneme_sh=enframe(Phoneme_sh,hamming(N),Overlap_sam);
[FeatsVector_sh,~]=FeatExt(Phoneme_sh,Fs,framed_Phoneme_sh);

% \a\
Phoneme_a=ProcessedSig(seg_ind(2):seg_ind(3));
framed_Phoneme_a=enframe(Phoneme_a,hamming(N),Overlap_sam);
[FeatsVector_a,~]=FeatExt(Phoneme_a,Fs,framed_Phoneme_a);

% \l\
Phoneme_L=ProcessedSig(seg_ind(3):seg_ind(4));
framed_Phoneme_L=enframe(Phoneme_L,hamming(N),Overlap_sam);
[FeatsVector_L,~]=FeatExt(Phoneme_L,Fs,framed_Phoneme_L);

% \o\
Phoneme_o=ProcessedSig(seg_ind(4):seg_ind(5));
framed_Phoneme_o=enframe(Phoneme_o,hamming(N),Overlap_sam);
[FeatsVector_L,~]=FeatExt(Phoneme_L,Fs,framed_Phoneme_L);

% \m\ - 5
Phoneme_m=ProcessedSig(seg_ind(5):seg_ind(end));
framed_Phoneme_m=enframe(Phoneme_m,hamming(N),Overlap_sam);
[FeatsVector_L,~]=FeatExt(Phoneme_L,Fs,framed_Phoneme_L);

h = figure;
XT=[];
for i=1:length(seg_ind)-1
    plot(t(seg_ind(i):seg_ind(i+1)),Signal(seg_ind(i):seg_ind(i+1)));hold on
    XT=[XT t(round((seg_ind(i)+seg_ind(i+1))/2))];
end
hold on
for i=2:length(seg_ind)-1
    line([t(seg_ind(i)) t(seg_ind(i))],[-1 1], 'color', 'm', 'linewidth', 1.4);hold on
end
hold on
ylim(1.2*max(abs(Signal))*[-1 1]);
xticks(XT(2:end-1));
xticklabels({'SH','A','L','O','M'});
xlabel 'Time [S]'; xlim([0 t(end)])
title 'Segmented pronunciation of the word 'Shalom' ' ; ylabel 'Amplitude'
% legend('speech','start_{speech} ','end_{speech}');
grid on
saveas(h,fullfile(folder_name,'Processed_Signal'), 'fig');
saveas(h,fullfile(folder_name,'Processed_Signal'), 'jpg');

h1 = figure;
[~,F,T,P] = spectrogram(Signal,STFTwinLength,STFToverlap,STFTnfft,Fs, 'yaxis');
surf(T,F,10*log10(P), 'edgecolor', 'none');
view([0,90]);
title '"shalom" Spectrogram'; xlabel 'Time [S]'; ylabel 'Frequency [Hz]';
caxis([STFTcmin STFTcmax]); colormap jet
xlim([t(1),t(end)]);
ylim([0,Fs/2]);
for i=2:length(seg_ind)-1
    line([t(seg_ind(i)) t(seg_ind(i))],ylim,[-1 1], 'color', 'k', 'linewidth', 1.4)
end

saveas(h1,fullfile(folder_name,'Spectrogram'), 'fig');
saveas(h1,fullfile(folder_name,'Spectrogram'), 'jpg');

if Flag~=0
    h2 =figure;

```

```

plot(NRG,'color','r','LineWidth',1.2);hold on
plot(ZCR,'color','k','LineWidth',1.2);hold on
for i=2:length(seg_ind)-1
    line([(seg_ind(i))/(Overlap_sam)
(seg_ind(i))/(Overlap_sam)],ylim,'color','m','linewidth',1.4)
end
grid on
title 'Energy and ZCR of the signal' ;% xlabel 'Time [S]' ;
ylabel 'Arbitrary amplitude'
legend('NRG','ZCR')
saveas(h2,fullfile(folder_name,'Energy_and_ZCR'),'fig');
saveas(h2,fullfile(folder_name,'Energy_and_ZCR'),'jpg');

[h11,h21]=estimatePhonemeFormants(Phoneme_sh,Fs,phon(1,:));
saveas(h11,fullfile(folder_name,'estimate_Phoneme_sh_Formants'),'fig');
saveas(h11,fullfile(folder_name,'estimate_Phoneme_sh_Formants'),'jpg');
saveas(h21,fullfile(folder_name,'AR_Model_Poles_Phoneme_sh'),'fig');
saveas(h21,fullfile(folder_name,'AR_Model_Poles_Phoneme_sh'),'jpg');

[h12,h22]=estimatePhonemeFormants(Phoneme_a,Fs,phon(2,:));
saveas(h12,fullfile(folder_name,'estimate_Phoneme_a_Formants'),'fig');
saveas(h12,fullfile(folder_name,'estimate_Phoneme_a_Formants'),'jpg');
saveas(h22,fullfile(folder_name,'AR_Model_Poles_Phoneme_a'),'fig');
saveas(h22,fullfile(folder_name,'AR_Model_Poles_Phoneme_a'),'jpg');

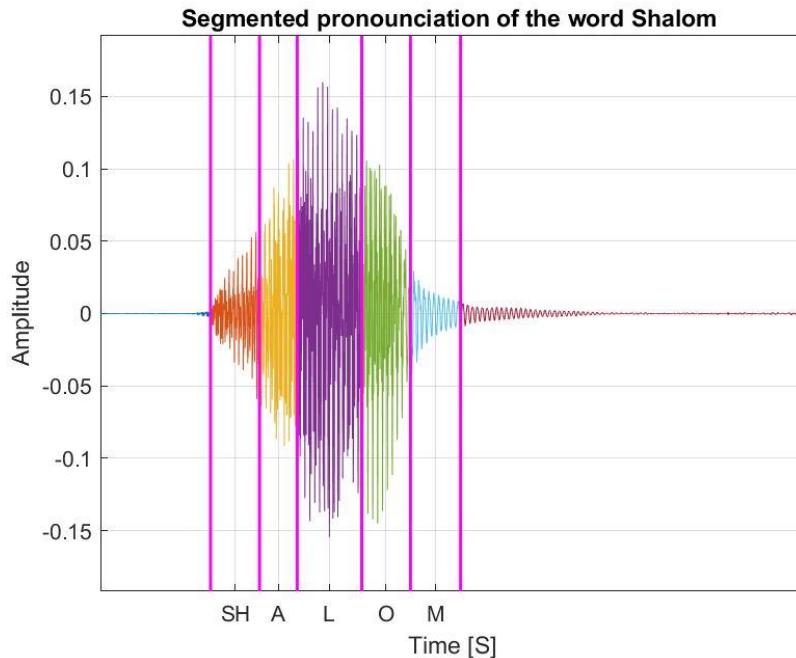
[h13,h23]=estimatePhonemeFormants(Phoneme_L,Fs,phon(3,:));
saveas(h13,fullfile(folder_name,'estimate_Phoneme_L_Formants'),'fig');
saveas(h13,fullfile(folder_name,'estimate_Phoneme_L_Formants'),'jpg');
saveas(h23,fullfile(folder_name,'AR_Model_Poles_Phoneme_L'),'fig');
saveas(h23,fullfile(folder_name,'AR_Model_Poles_Phoneme_L'),'jpg');

[h14,h24]=estimatePhonemeFormants(Phoneme_o,Fs,phon(4,:));
saveas(h14,fullfile(folder_name,'estimate_Phoneme_o_Formants'),'fig');
saveas(h14,fullfile(folder_name,'estimate_Phoneme_o_Formants'),'jpg');
saveas(h24,fullfile(folder_name,'AR_Model_Poles_Phoneme_o'),'fig');
saveas(h24,fullfile(folder_name,'AR_Model_Poles_Phoneme_o'),'jpg');

[h15,h25]=estimatePhonemeFormants(Phoneme_m,Fs,phon(5,:));
saveas(h15,fullfile(folder_name,'estimate_Phoneme_m_Formants'),'fig');
saveas(h15,fullfile(folder_name,'estimate_Phoneme_m_Formants'),'jpg');
saveas(h25,fullfile(folder_name,'AR_Model_Poles_Phoneme_m'),'fig');
saveas(h25,fullfile(folder_name,'AR_Model_Poles_Phoneme_m'),'jpg');

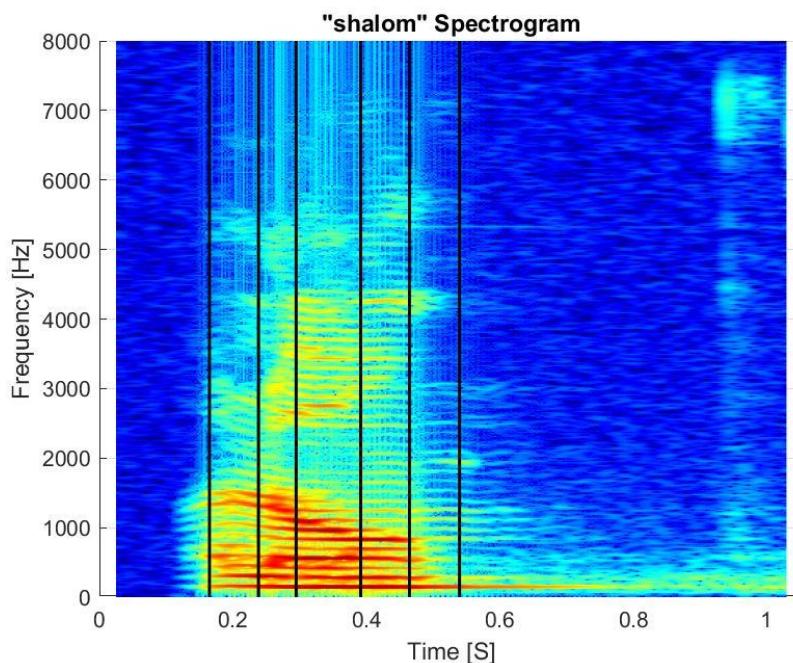
end

```



איור 23 - סגמנטציה אוטומטית עבור דובר A - מיכאל

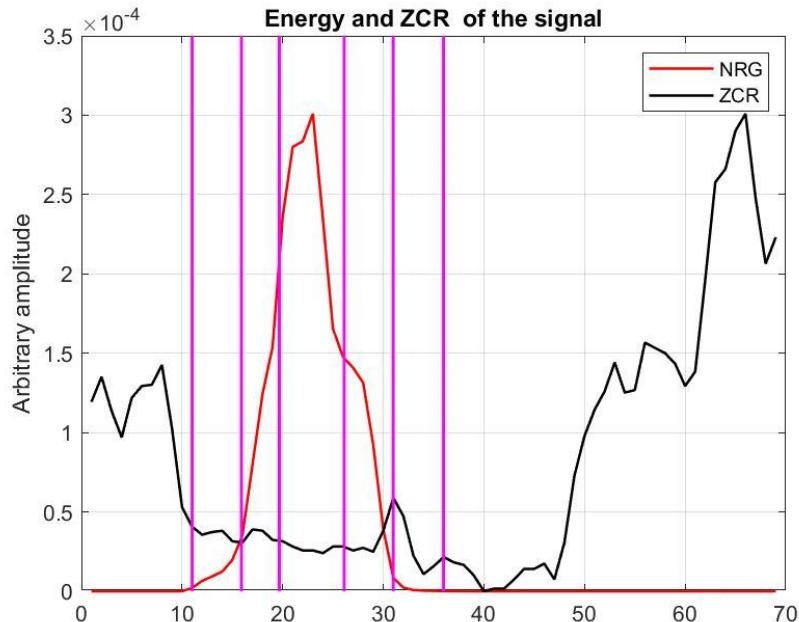
ניתן לראות, מיור 23 את האות המוקלט שלנו לאחר העבודה המקדים. כמו כן, מסומנים עליו הפונומות השונות. ניתן לראות באופן כללי כי הצלחנו לזהות 5 פונומות. כמו כן, אורך בזמן די דומה ולכון המרוחחים בין הסגמנטים כמעט קבועים.



איור 24 - ספקטרוגרפיה עבור דובר A

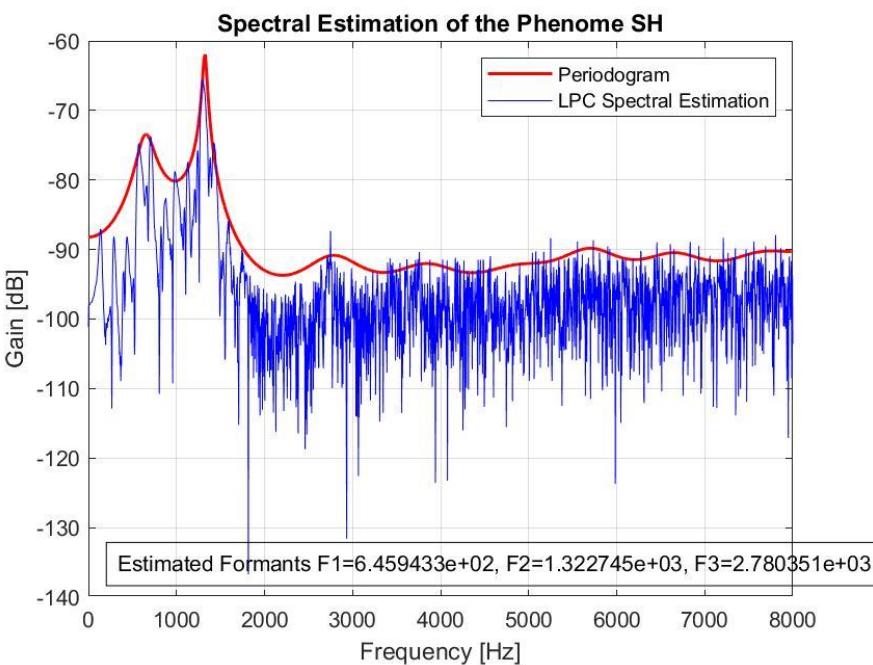
באיור 24 רואים את הספקטרוגרפיה של המילה שלום. ניתן לראות כי גבולות הסגמנט שהוספנו מסעיף הבonus נראים היטב וכך ניתן לראות היכן סיוגנו כל פונמה. נשים לב שמעט קשה לזהות את הפורמנטות בתדרים הגבוהים

עבור כל אחת מן הfonemoת. כמו כן, אנו יודעים כי הפונמה /SH/ צריכה להיות עם פורמנטוות גבוהות מאחר ומדובר בfonema א-קוליית.

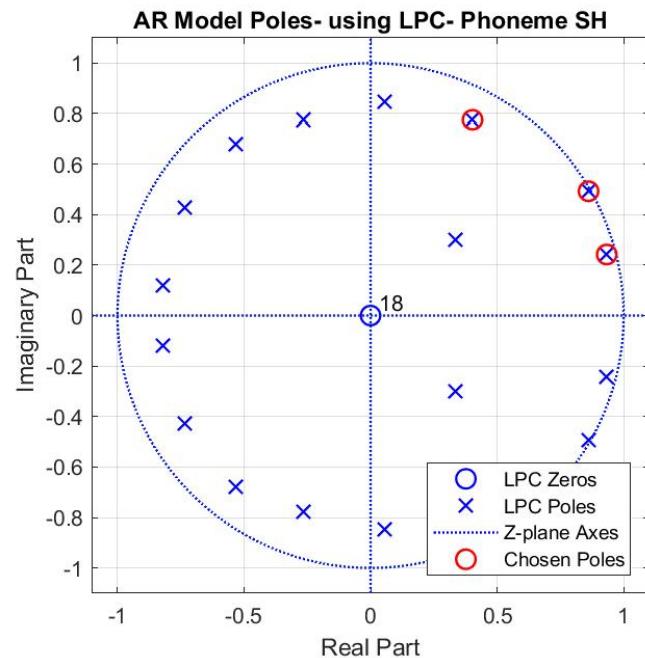


אייר 25- אנרגיה ו עבור דובר 1

באייר 25 ניתן לראות את האנרגיה ואת ZCR עבור על אחד מן הפרויימים. נזכיר כי בחרנו לעבוד בחלונות זמן של 30 מילישניות כך רואים כי מבחינת אנרגיה וZCR אנו מושרים עם התיאוריה. כך שבfonema קולית יש לנו אנרגיה גבוהה (3- הfonemoת האמציאות) ו- ZCR נמוך. כמו כן, ניתן לראות כי עבור הfonemoת הא-קוליות אנו מקבלים כי ה ZCR גבוהה (דומה לשקט) וכן האנרגיה נמוכה. מתוארכות אלו ניתן לראות כי בקרוב הסגמנטציה הצלילה.

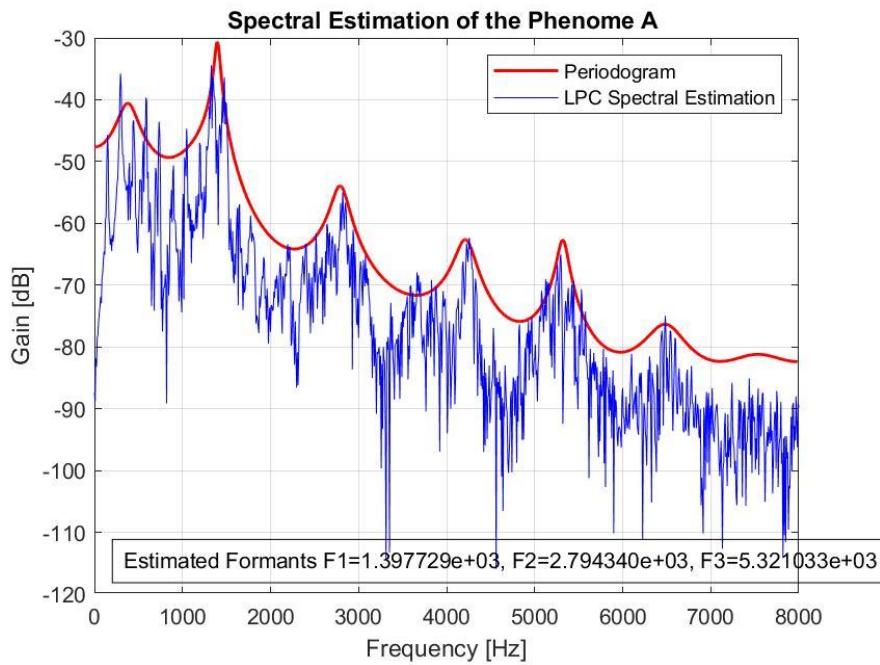


אייר 27- פריזודוגרמה /SH/-

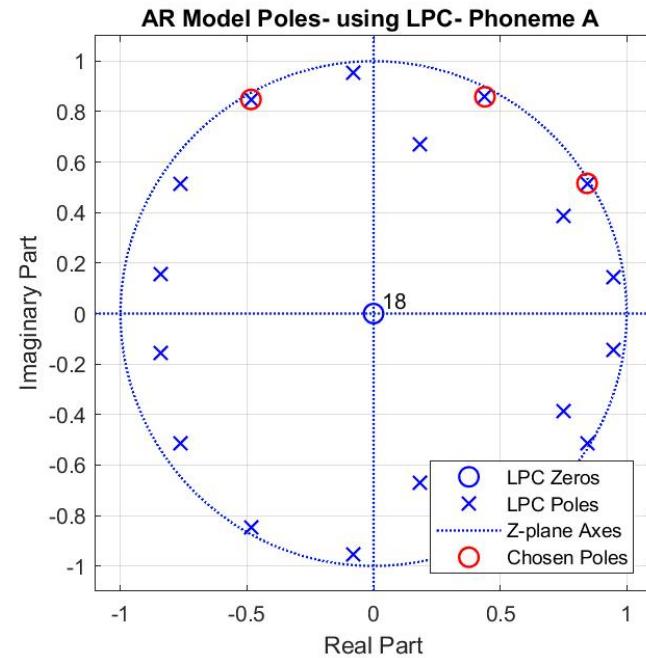


אייר 26- קטבים ואפסים /SH/-

באיור 26 רואים את מפת הקטבים ואפסים של הfonema הא- קוילת הראשונה /SH/. רואים כי זהיהוי לא מדויק מאחר והפורמננות העיקריות מתאימות בכלל לפונמה /ə/. ייתכן כי הfonema /SH/ נאמרה בזמן דיבר קצר, מה שגורם לנו לא ליזוחות אותה.

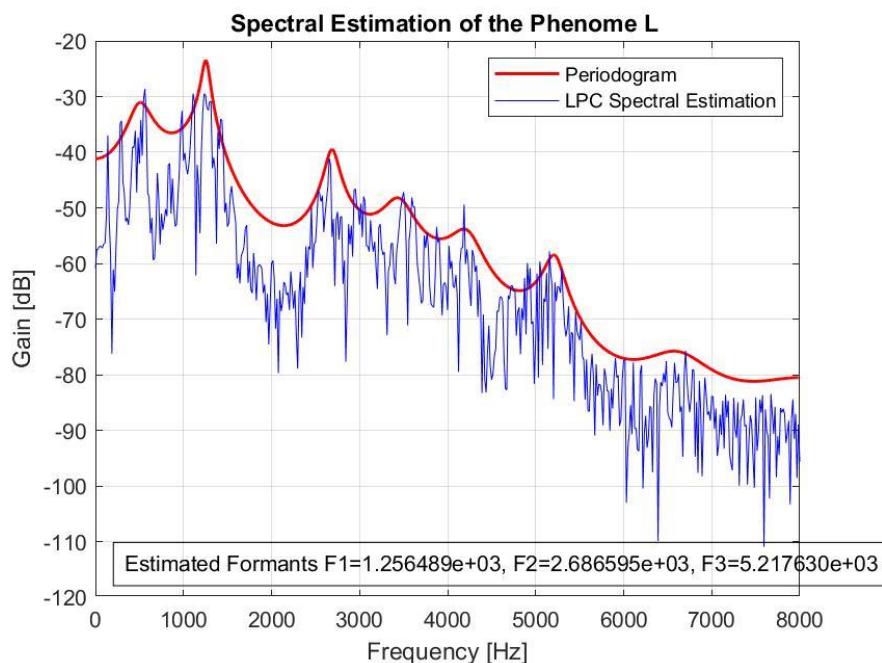


איור 28- פריזוגרמָה /ə/-

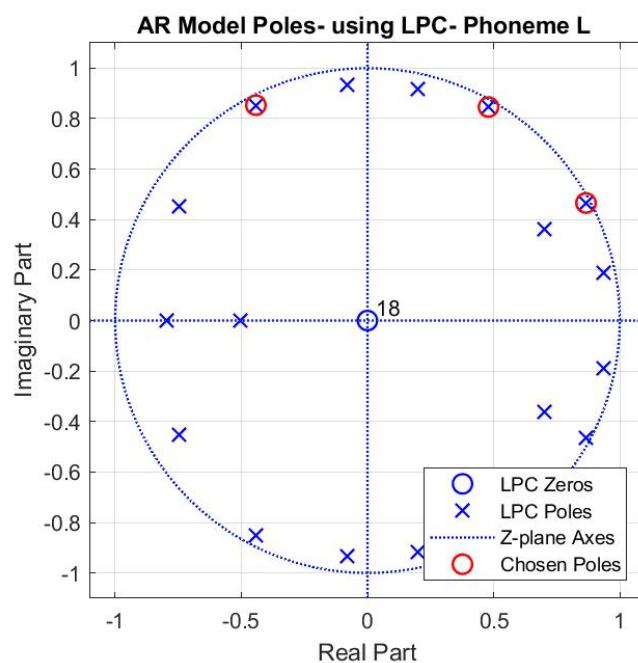


איור 29- קטבים ואפסים /ə/-

באיור 28 רואים את הקטבים ומfasים המתאימים לפונמה /ə/. ניתן לראותו, כי בדומה לזהיהו הfonema הקודומה (בהתבוננות באיור 27) אנו מקבלים פורמננותות בחלקים יחסית ראשונים. ככלומר אנו זיהינו את הfonםות בצורה דיבר דומה.

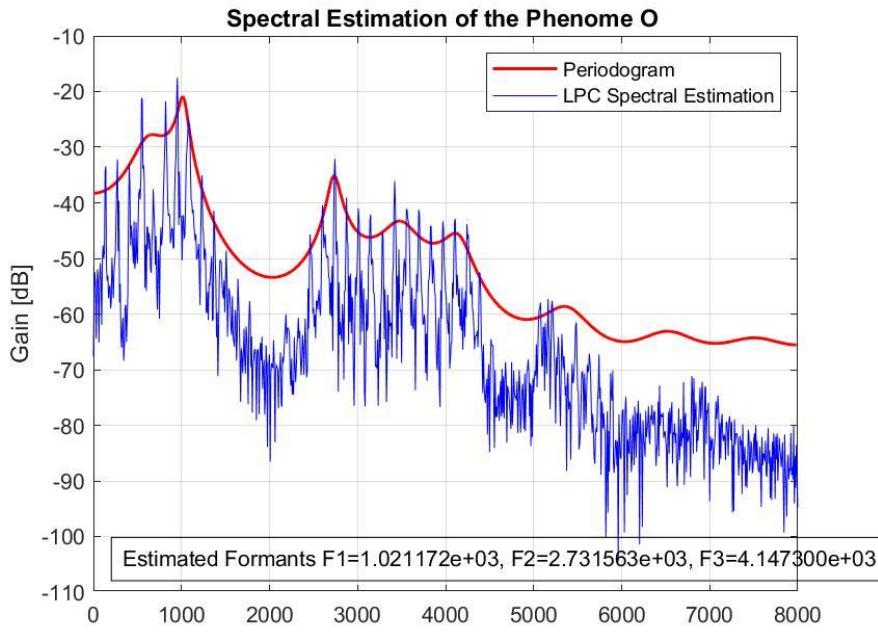


איור 30- פריזוגרמָה /ə-/L/

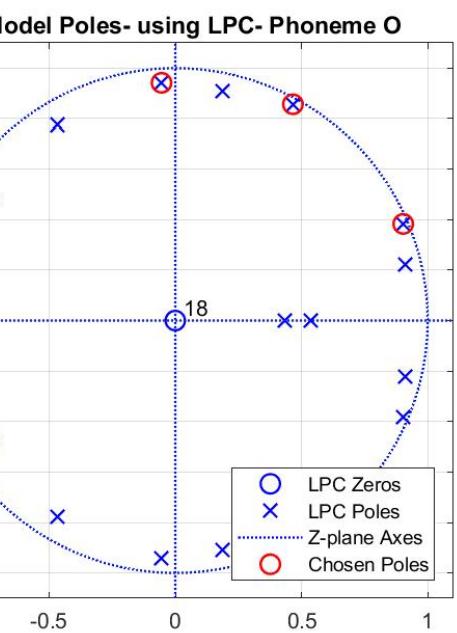


איור 31- קטבים ואפסים /ə-/L/

באיור 31 ניתן לראות את הקטבים המתאימים לפורמננטות העיקריות בפונמה /ɔ/. רואים מאייר 30 כי יש לנו 3 פורמננטות עיקריות המרכיבות את המודל של ה-vocal tract.

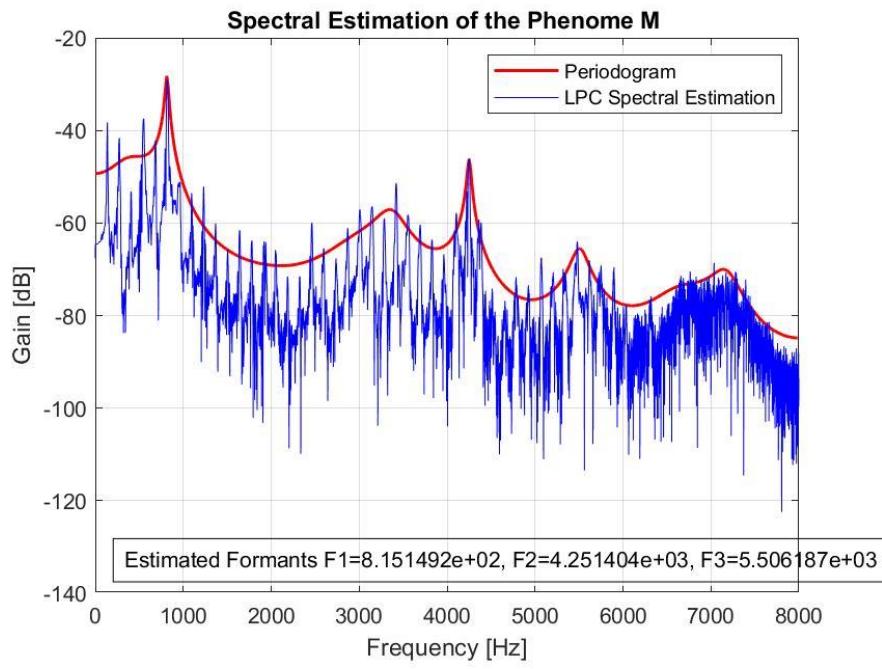


איור 32- פריזוגרמה /ɔ/-/ə

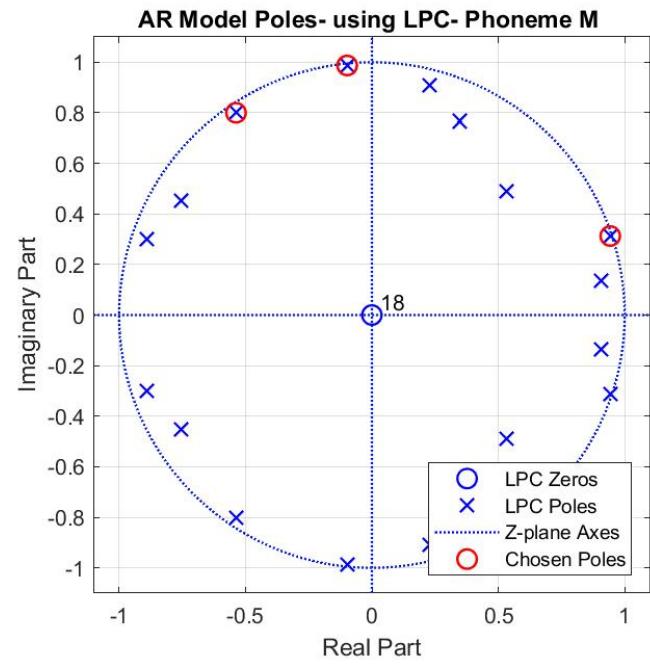


איור 32- קטבים ואפסים /ɔ/-/ə

באיורים אלו רואים את השיעורן של הפונמה /ɔ/. רואים כי זו פונמה קולית המורכבת בעיקר מפיקים הנמצאים בתדרים נמוכים.



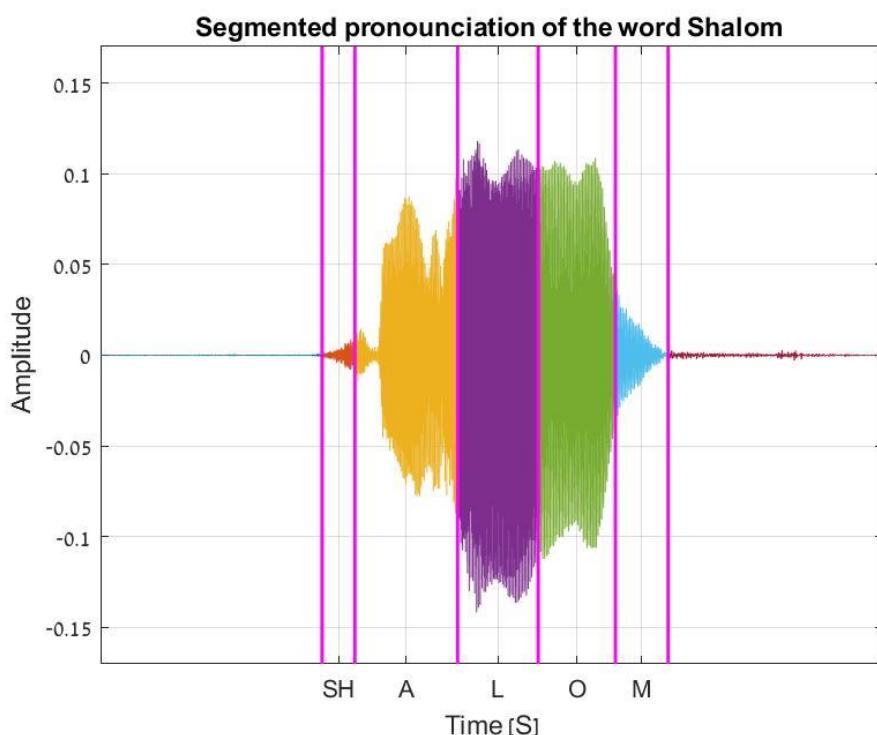
איור 33- פריזוגrama /M/-/ə



איור 34- קטבים ואפסים /M/-/ə

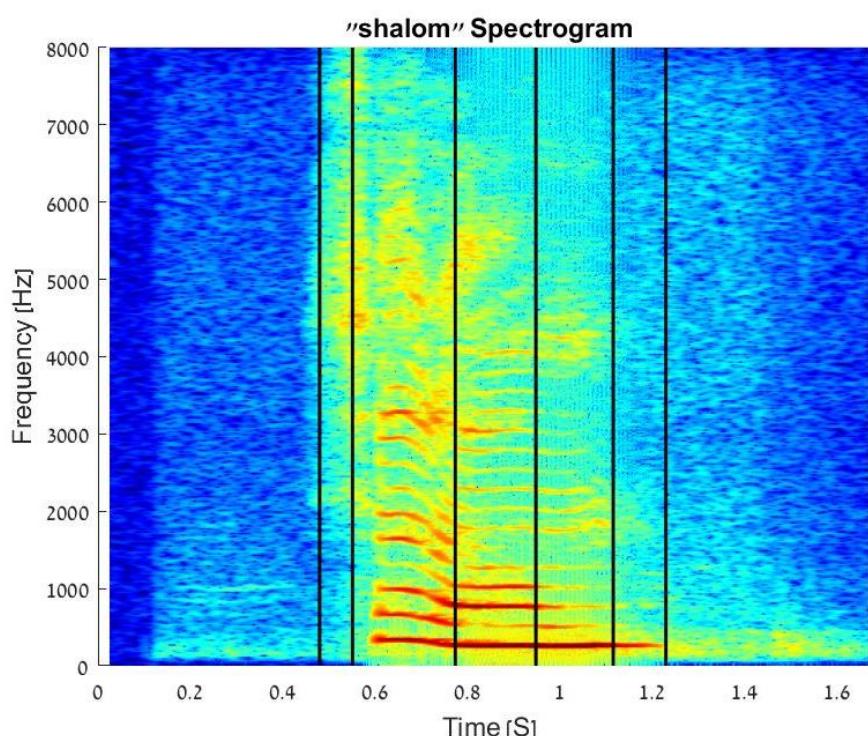
כאן אנו מציגים את הפונמה היחידה שנחשבת Nasal. מתאפיינת בפורמננטות בתדר גבוהה למעט שיא אחד דומיננטי בתדר קרוב ל-1000 הרץ.

כעת נציג את התוצאות של הטעסן הראשוני של זיהוי של מ/or

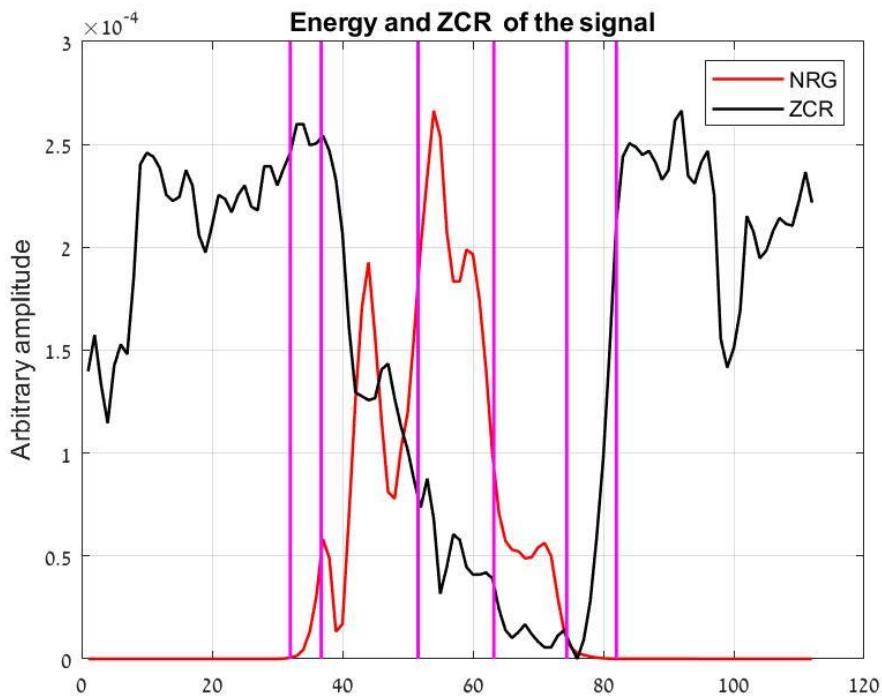


איור 35-SEGMENTATION AUTOMATIQUE EN VOIX TEST 1 - B מ/or

ניתן לראות מיור 35 כי הסימון של הפונמה הראשונה מעט קצר. נצפה לראות את הפורמנוטות של /SH/ בפריזוגרפיה של הפונה /a/.

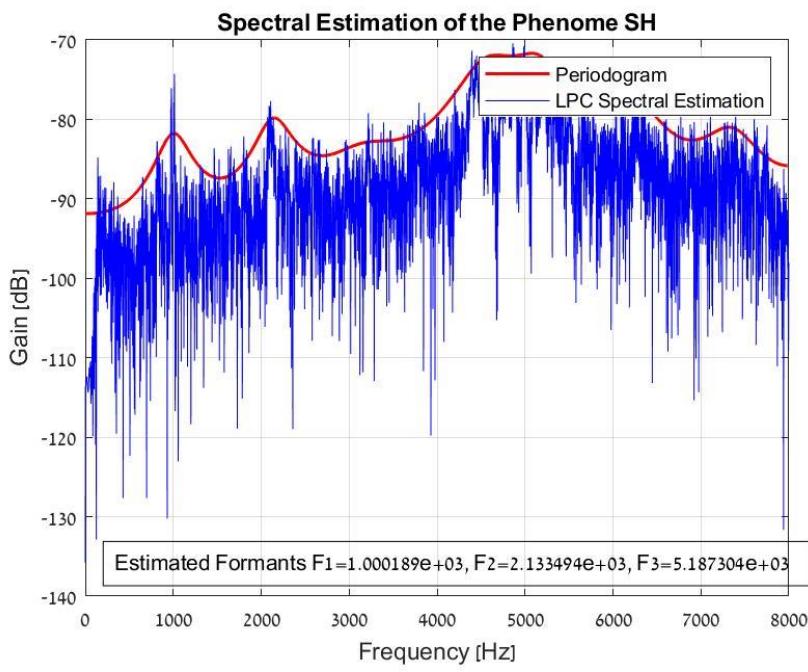


איור 36-СПЕКТРОГРАММА ВОВОДА ТЕСТ 1 - B

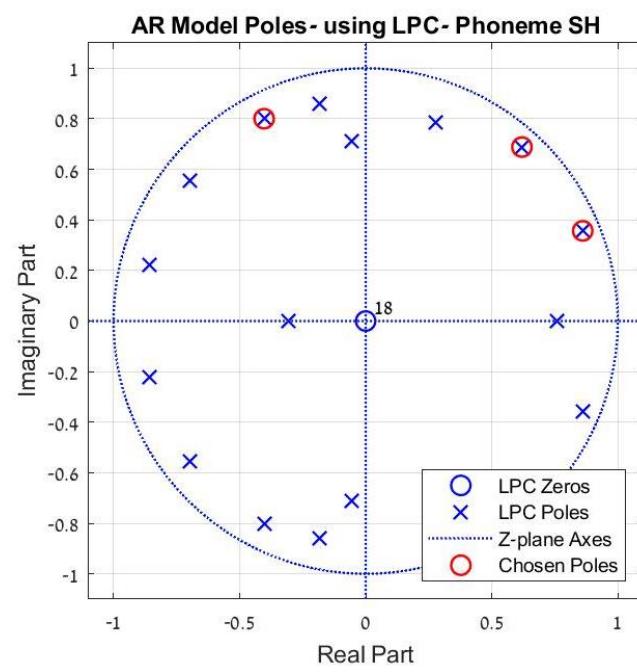


אייר 37- אנרגיה ו עבור דובר -B Test 1

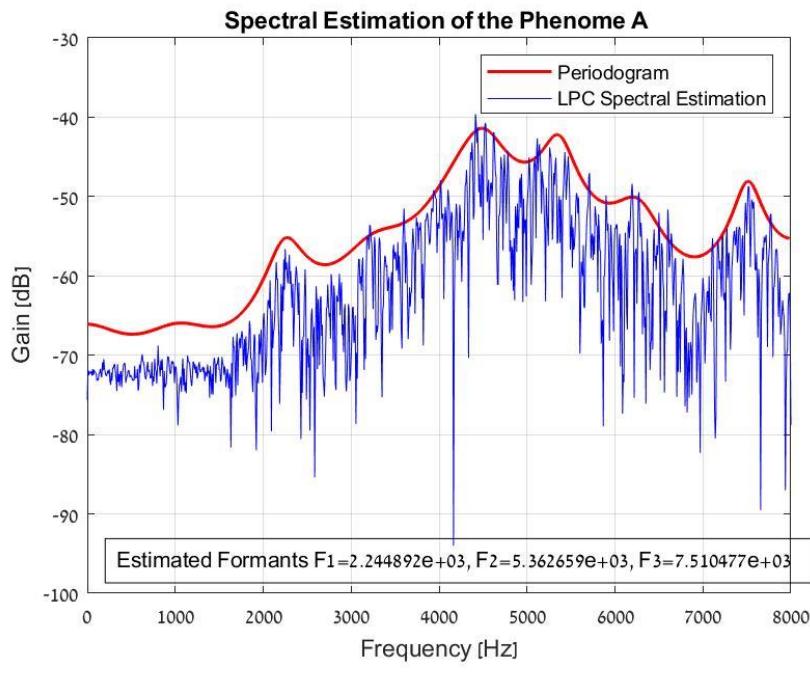
באייר 37 ניתן לראות את האנרגיה ו עבור אותן הדיבור של דובר B (מור). ניתן לראות כי גם במקרה שלה, הפונמות הא-קוליות בעלות ZCR גובהה ואנרגיה נמוכה. בנוסף, רואים כי הfonmotot הקוליות בעלות ZCR נזוק ואנרגיה גבוהה.



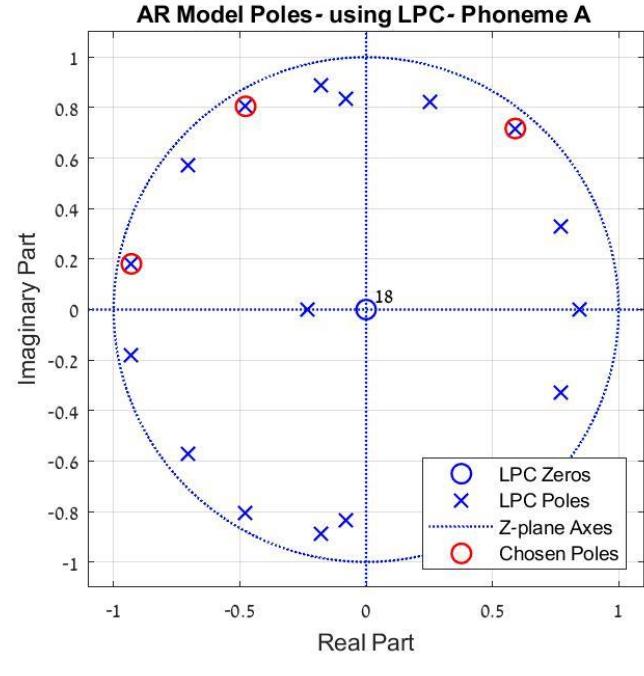
אייר 38- פריזודוגרפיה /SH/B-



אייר 39- קטבים ואפסים /SH-/B-

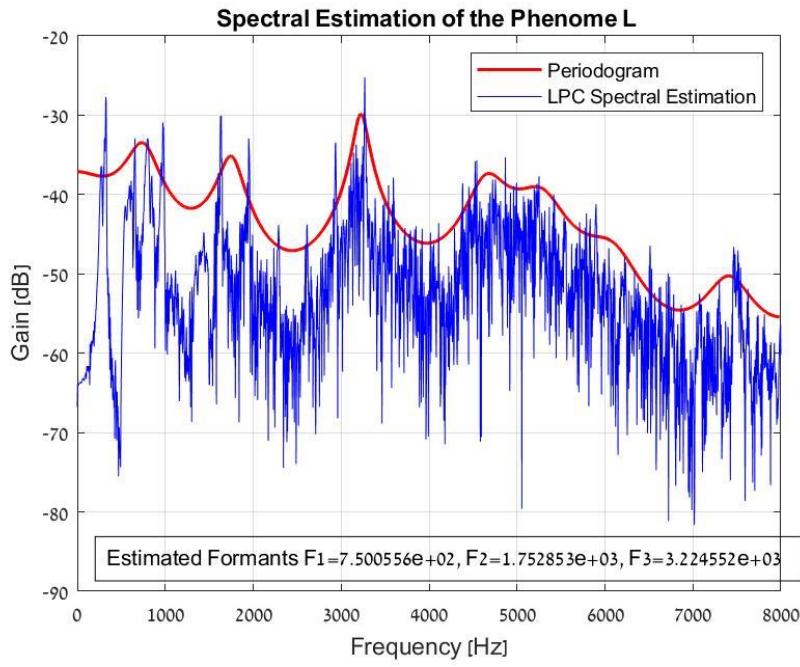


איור 40- פריזודוגרפיה /A-/

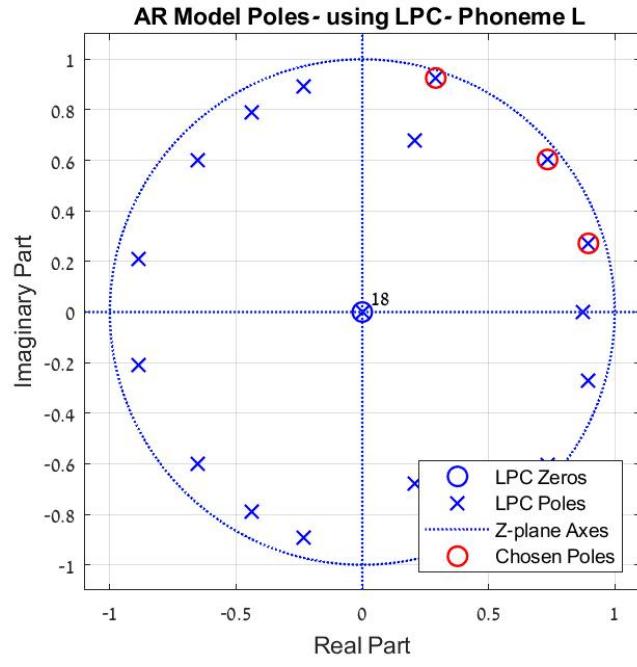


איור 41- קטבים ואפסים פונמה /A-/

כאן ניתן לראות את הפונמה /A/. נראה בעיקר כמו /sh/.

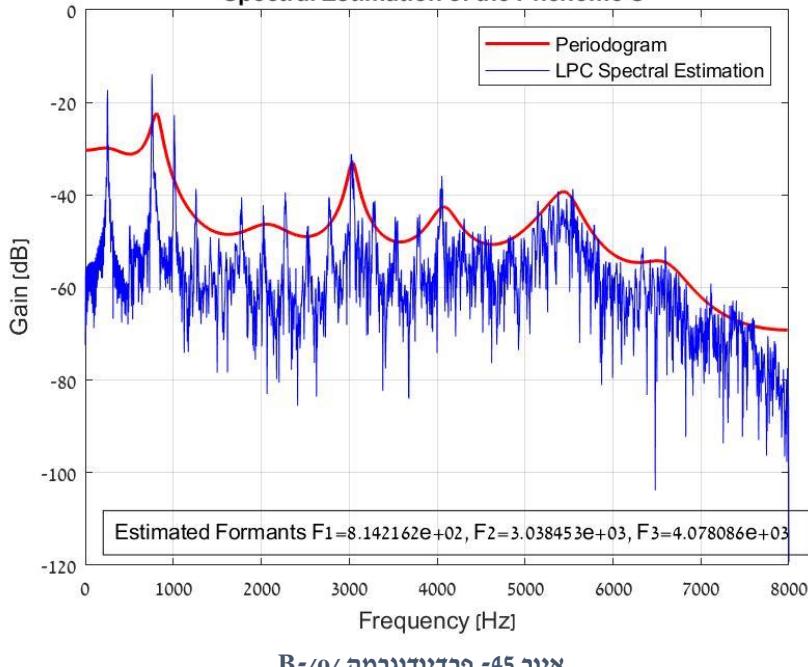


איור 43- פריזודוגרפיה /L-/



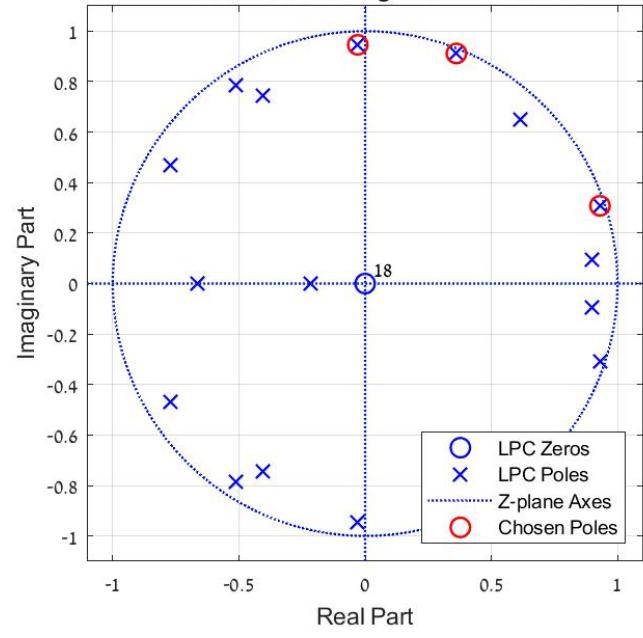
איור 42- קטבים ואפסים פונמה /L-/

**Spectral Estimation of the Phoneme O**



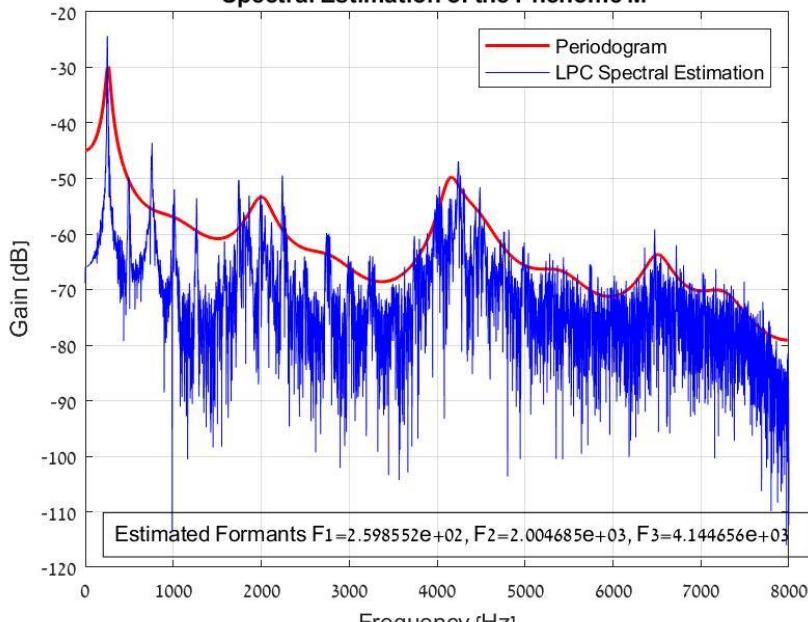
איור 45- פריזיאודוגרפיה /ו/

**AR Model Poles- using LPC- Phoneme O**



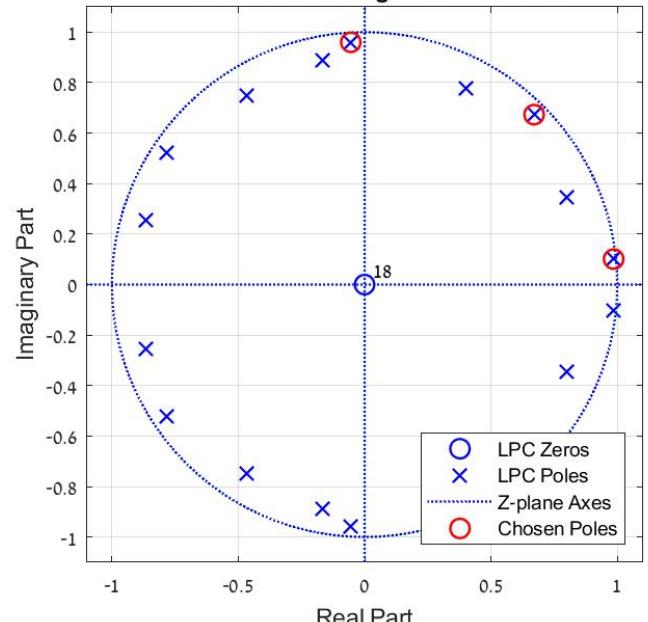
איור 44- קטבים וඅපසිම පොන්මා /ו/

**Spectral Estimation of the Phoneme M**



איור 46- פריזיאודוגרפיה /M/

**AR Model Poles- using LPC- Phoneme M**



איור 47- קטבים וඅපසිම පොන්මා /M/

## 9. מסקנות כלליות

בניסוי זה אנו עוסקים בעיבוד אוטומטי של דיבור. אנו ראיינו שאות דיבור משתנה מהר בזמן ועל כן עדיף לפצל את הבעיה בעיות קטנות יותר, כך שאנו מניחים סטציונריות של האות במרקורי זמן של 30 מילישניות. נזכר כי הfonema הקצרה ביותר בדרך כלל אורך 40 מילישניות כך שנוכל להבין באיזו פונמה מדובר. ראיינו כי ישנו צורך חשוב מאוד בעיבוד מקדים, בו אנו מורידים את DC של האות, מעבירים עם מסנן HP שתפקידו הוא להגדיל SNR ולפצות על דעיכת תדרים גבויים דרך חללי הפה. כמו כן, כדי למנוע השפעת קצה בין מקטעים חזרים, אנו מכפילים בחולון כלשהו בזמן (בחרנו לעובד עם חלון מלכני) כך שהקונולוציה בתדר תאפשר לנו למנוע תופעת זליגה. כמו כן, ישנו צורך בשימוש בחיפוי זמני בין החלונות כך שנוכל להתרכו בתדרים שימושיים אותןנו ולהנחת עוד יותר את התדרים שלא משמשים אותנו.

בנוסף, ביצענו סגמנטציה אוטומטית לאות הדיבור שלנו באמצעות חלון זמן באורך 30 מילישניות. הסף שנבחר עבורנו היה  $50 = \eta$ . נציין, כי בחירה בערכי סף גדולים יותר גרמה לזיהוי רחב יותר של הfonemoות, ובבחירה סף נמוך אנו לא זיהינו פונמוות, או שלפחות כל הfonemoות הקוליות זוחו יחד. כך שבסופו של דבר, התיישרנו אל מול ערכאים שהניבו לנו את תוכנת הסגמנטציה שטובה ביותר לשאלה 2.4 מתאים לנו גם, ועל כן השתמשנו בערכים וראיינו בחלק המוביל TraiTest כי הסף שבחרנו לעובד אותו בתשובה לשאלה 2.4 מושך לנו מטה. בחירה בזמןים קצרים יכולת לפגוע בזיהוי הfonemoות, כפי שראינו באIOR 8 שם נחזה מספר פעמים הרף עבור הfonemoות השונות, אך רק אלו שעברו לאורך מספיק זמן גמרו לנו להבין כי מדובר בגבול סגמנט.

חלק חשוב מן הניסוי היה לשערץ את הfonemoות של כל אחת מן הפורמנטוות על ידי LPC. כלל האצבעו אותו למדנו בעיבוד אוטות פיזיולוגיים לפיו בוחרים את P לפי חלוקה של תדר הדגימה ב 1000 והוספה של 2 מקדים למידול תופעת הקירינה(מעבר תדרים מטווק לתווך), מאפשר לנו לשערץ בזורה דyi טובہ את הספקטרום של הfonema. על ידי התבוננות בשיאים של התגובה לתדר של הפילטר המmdl את-h Vocal Tract, אנו יכולים לזהות את תדרי התהודה(פורמנטוות) של המערכת. ישים יתרונות וחסרונות לשערוך שימוש בשערוך פרמטרי ולא פרמטרי של הספקטרום. באמצעות שימוש בשערוך לא פרמטרי אנו יכולים לקבל את השערוך של הספקטרום באופן מפורט מאוד. איתנו נוכל למצוא למשל את Pitch. בעת התבוננות באIOR 9 ניתן לראות כי בתדרים עד לסדר גודל של  $[kHz] 1$  רואים את הרוחים בין השיאים המשניים. ככלمر ברוחים בהםים יכולים לראות שיעורן לא רע דרך "העין". כמו כן, בטבלה 2 ראיינו את הערכאים האמייטיים של Pitch. ראיינו בווקטור המאפיינים שהוצאנו עם הפונקציה מתשובה לשאלה 4.6 כי אצל מורה היה גובהו יותר מאשר מיכאל, מה שעזר לנו מאוד בזיהויים. ראיינו כי הדריך התובה ביוטר לישם את מציאת Pitch היא על ידי אלגוריתם Levinson Durbin(LD) [1]. המנצלת את העובדה שמטריצת האוטוקורלציה היא סימטרית ובעל מבנה טופלייך, (כל האלכסונים הראשיים שוויים). בזורה זו נוכל למצוא את מסנן ההיפוך LPC מסדר 4. נציין כי שימוש בחולנות זמן הקטנים מ 10 כפי שמצוין [1], מקבלים שעורך לא נכון של הפרמטרים הרצויים, כמו התדר היסודי, קצב חציית האפסים והאנרגייה בכל פריים. כמו כן, שינוי הטונציה יכול להשפיע על הממצאים.

ראיינו כי שלושת הקטבים הקרובים ביותר בערכם המוחלט אל מעגל היחידה מתאימים לנו את הפורמנטוות העיקריות של התגובה לתדר של מסנן ה-Vocal Tract. שיטת LPC משתקת כאן תפקיד חשוב בקידוד פונמוות ובעזרת השערוך הפרמטרי של הספקטרום אנו מצליחים לראות את המעטפת של הספקטרום בזורה גלויה לעין. ראיינו מן התוצאות של GUI Test כי הפורמנטוות בחלק מן המקרים תאמו למאה שראינו בתיאוריה. ובחלק השני הן דוקא התאימו לפונמוות אחרות. ראיינו זאת בעיקר בשתי הפונמוות הראשונות. במקרים בהם ההבחנה של הfonema /sh/, הייתה ארוכה, ראיינו את המעטפת הספקטורלית של אותה פונמה מופיעה דוקא בשערוך של הfonema /s/, מה שמעיד על ביצועים פחות טובים בסגמנטציה. כמו כן, בהרבה מן הזיהויים שלנו, שmeno לב כי המעטפת

הספקטראלית של /א/, נראהית מעט שונה. לאחר הרבה הקלותות שmeno לב כי דוקא /א/ נמכה (ב- [1] מסומנת כ/@@/) יותר מתאימה לנו. אמם לא זיהינו באיר 29 את הפורמנטה הראשונה אך בכל זאת ניתן לראות מעטפת ספקטראלית מתאימה. כמו כן, רואים באיר 28 כי אכן קיים קווט הנמצא יחסית קרוב למגל היחידה, אך נראה עקב זיהוי שווה של הגבולות הסגמנט קיבלו פורמנטה מן הפונמה /bl/. ככלומר נקודת התויפה שלנו הייתה בזיהוי הפונמות, שפוגע בזיהוי הפונמות. תופעה זו חוזרת על עצמה בזיהויים השונים, אך אם היו מסומנים את 4 הפורמנטוות הגבוהות ביותר

ב尤תך

נציין כי המודל שלנו הצליח בצהורה טובה לזהות בין הדוברים. ראיינו כי אלגוריתם הקלסיפיקציה שהייתה נתנו עבור על עיקרונו בו אנו מחשבים את כל אחד מן הפיצ'רים במהלך המבחן כאילו הוא מגיע מהתפלגות נורמלית עם ממוצע ושונות המתאימים להתפלגות נורמלית עם ערכיהם המתאימים כפי שהוא מאננים מן המודל. בצהורה זו אנו סכמים על לוגריתם בסיס 4 של כל המאפיינים. בעצם אנו משווים בין הסכומים כדי לראות למי הדבר הנבחן דומה יותר, לדובר A או B. נציין כי אנו לא בטוחים מה יהיו תוצאות המודל במידה ונשנה ממש את הטונציה בה אנו אומרים את המילים. שכן, הטונציה יכול להשפיע על הפיטץ'.

## 10. מקורות

- [1] J. G. Deller, J. R., Hansen, J. H. L., & Proakis, *Discrete-time processing of speech signals*. IEEE Press. .
- [2] P. H. Milenkovic, M. Wagner, R. D. Kent, B. H. Story, and H. K. Vorperian, “Effects of sampling rate and type of anti-aliasing filter on linear-predictive estimates of formant frequencies in men, women, and children,” *J. Acoust. Soc. Am.*, vol. 147, no. 3, pp. EL221–EL227, Mar. 2020, doi: 10.1121/10.0000824.
- [3] P. (2005). Sörnmo, & Laguna, “Electrical signal processing,” *Int. J. Electron.*, vol. 73, no. 5, pp. 1085–1086, 1992, doi: 10.1080/00207219208925773.
- [4] H. Kim, “Linear Predictive Coding is All-Pole Resonance Modeling,” *Cent. Comput. Res. Music Acoust. Stanford Univ.*, no. Figure 1, pp. 1–7, Accessed: Jan. 02, 2022. [Online]. Available: <https://ccrma.stanford.edu/~hskim08/lpc/>.
- [5] M. H. J. Gruber and M. H. Hayes, “Statistical Digital Signal Processing and Modeling,” *Technometrics*, vol. 39, no. 3, p. 335, 1997, doi: 10.2307/1271141.
- [6] J. D. Markel, “The SIFT Algorithm for Fundamental Frequency Estimation,” *IEEE Trans. Audio Electroacoust.*, vol. 20, no. 5, pp. 367–377, 1972, doi: 10.1109/TAU.1972.1162410.
- [7] M. M. Goodwin, “The STFT, Sinusoidal Models, and Speech Modification,” in *Springer Handbooks*, Springer, Berlin, Heidelberg, 2008, pp. 229–258.

```

%% Exp4 - Group 8
close all;clear all

% 1.2
[speech,Fs] = audioread('shalom_example.wav');

alpha =0.999; % Degree of pre-emphasis
figure
freqz([1 -alpha],1,[],Fs); % Calculate and display frequency response
title 'Pre-Emphasis filter';
figure
[z,p,~] = zplane([1 -alpha],1);
findzeros = findobj(z,'Type','line'); findpoles = findobj(p,'Type','line');
title('Zero - Pole Map');
set(findzeros,'Color','m','linewidth',1.2,'markersize',9);
set(findpoles,'Color','b','linewidth',1.2,'markersize',9);

% show Raw speech in time:
t = [0:length(speech)-1]*1/Fs; % time vector
figure,subplot(2,1,1), plot(t,speech); xlim([0 t(end)]);
xlabel 'Time[S]'; ylabel 'Amp'; title 'speech';
% show Raw speech in Frequency:
Y = fft(speech); L=length(speech);
P2 = abs(Y/L); P1 = P2(1:L/2+1); P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(L/2))/L;
subplot(2,1,2), plot(f,P1) ; xlim([0 8000]);
xlabel 'Frequency[Hz]'; ylabel '|Amplitude|'; title 'speech in Frequency';

WindowLength=30*10^-3; % 1 seconds window
Overlap=50; % 50% overlap
[ProcessedSig,~]=PreProcess(speech,Fs,alpha,WindowLength,Overlap);

% show ProcessedSig speech in time:
figure,subplot(2,1,1), plot(t,ProcessedSig)
xlabel 'Time[S]'; ylabel 'Amp'; title 'speech after PreProcess'; xlim([0 t(end)]);
% show ProcessedSig speech in Frequency:
Y = fft(ProcessedSig); L=length(ProcessedSig);
P2 = abs(Y/L); P1 = P2(1:L/2+1); P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(L/2))/L;
subplot(2,1,2), plot(f,P1) ; xlim([0 8000]);
xlabel 'Frequency[Hz]'; ylabel '|Amplitude|'; title 'speech in Frequency';

% Window
N = 64;
rect = rectwin(N);
wvtool(rect)

%% segmantation:

close all; clear all; clc
[speech,Fs] = audioread('shalom_example.wav');
alpha=0.999;
WindowLength=30*10^-3; % 30 [mS] window
Overlap=50; % 50% overlap
[ProcessedSig,FramedSig]=PreProcess(speech,Fs,alpha,WindowLength,Overlap);

% 2.2

```

```

Idx=FindWordIdx(FramedSig,Fs,WindowLength,Overlap);

t=[0:length(speech)-1]*1/Fs; % time vector
figure, subplot(3,1,1)
plot(t,ProcessedSig); xlim([0 t(end)]);
ylim([min(ProcessedSig) max(ProcessedSig)]);
hold on
line([t(Idx(1)) t(Idx(1))], ylim, 'color', 'm', 'linewidth', 1.2)
line([t(Idx(2)) t(Idx(2))], ylim, 'color', 'm', 'linewidth', 1.2)
title 'signal - marker the speech ' ; xlabel 'Time [S]' ; ylabel 'Amplitude'
legend('speech','start ','end'); grid on

% 2.4
%playerObj = audioplayer(speech,Fs);
%play(playerObj);
%
% playerObj = audioplayer(ProcessedSig,Fs); play(playerObj);
dt = 35*10^-3; % minimum time above threshold 'eta' [mS]
eta = 50;
winlen = 30*10^-3; % 30 [mS]

[seg_ind,delta]=segmentation(ProcessedSig,winlen,eta,dt,Fs,Idx);

% ProcessedSig = ProcessedSig(Idx(1):Idx(2));
t=[0:length(ProcessedSig)-1]*1/Fs;
delta=delta(1:length(ProcessedSig));
% show results:
subplot(3,1,2), plot(t,ProcessedSig); hold on; grid on
% show segmentation lines:
for i=1:length(seg_ind)
    line([t(seg_ind(i)) t(seg_ind(i))], ylim, 'color', 'm', 'linewidth', 1.2)
end
xlim([0 t(end)])
title 'Segmentation - Shalom speech signal' ; xlabel 'Time [S]' ; ylabel 'Amplitude'
legend('speech','segments'); axis tight;
hold on

subplot(3,1,3), plot(t,delta); hold on
yline(eta, 'r', 'linewidth', 1.2)
for i=1:length(seg_ind)

    line([t(seg_ind(i)) t(seg_ind(i))], ylim, 'color', 'm', 'linewidth', 1.2)
end
xlim([0 t(end)]); grid on
title 'Spectral error measure' ; xlabel 'Time [S]' ; ylabel 'Amplitude'
legend('\Delta_1',['\eta=' num2str(eta)], 'segments');

%% 3

PhonemeSig = ProcessedSig(seg_ind(2):seg_ind(3));

% 3.1 Periodogram
figure
[Peri,omeg]=periodogram(PhonemeSig,[],'onesided');

plot((Fs/(2*pi))*omeg,10*log10(abs(Peri))); grid on
title 'Periodogram - nonparametric Power Spectral Density' ;
xlabel 'Frequency [Hz]' ; ylabel 'Magnitude [dB]'

rng default
% 3.3
% LPC analysis
P = Fs/1000 +2;
% g-variance of the prediction error
[a,g]=lpc(PhonemeSig,P);
std_g = sqrt(g);

```

```

M = length(PhonemeSig);
%use the Random Noise Generator on samples
add_noise_to_filt=filter(1,a,std_g*randn(M,1));
% Autoregressive all-pole model parameters – Yule-Walker method
[a1,g1]=aryule(add_noise_to_filt,P);
std_g1 = sqrt(g1);
%add noise var to frequency response
[h,omeg1]=freqz(std_g1,a1);

% find formants:
rts = roots(a1);
%Because the LPC coefficients are real-valued, the roots occur in complex
% conjugate pairs. Retain only the roots with one sign for the imaginary
% part and determine the angles corresponding to the roots.
rts = rts(imag(rts)>=0);
[~,Poles_idx]=maxk(abs(rts),3);
angz = atan2(imag(rts),real(rts));
Formants = angz(Poles_idx).*(Fs/(2*pi));
Formants = sort(Formants);
h1 = figure;
plot((Fs/(2*pi))*omeg1,20*log10(abs(h)), 'LineWidth',1.4, 'Color', 'r');
 xlabel 'Frequency (Hz)' ; ylabel 'Gain [dB]' ; hold on
 plot((Fs/(2*pi))*omeg,10*log10(Peri), 'Color', 'b') ; grid on

hold on
title 'Spectral Estimation of the Phoneme /a/ ' ;
xlabel 'Frequency [Hz]' ; ylabel 'Gain [dB]'
legend('LPC Spectral Estimation', 'Periodogram')
annotation(h1,'textbox',[ 0.15 0.09 0.09 0.1], 'String',...
{sprintf('Estimated Formants F1=%d, F2=%d, F3=%d', Formants(1),Formants(2),...
Formants(3))), 'FitBoxToText', 'on');grid on

h2 = figure;
[z,p,k] = zplane(1,a1); hold on; grid on
findzeros = findobj(z, 'Type', 'line'); findpoles = findobj(p, 'Type', 'line');
findk = findobj(k, 'Type', 'line');
set(findzeros, 'Color', 'b', 'linewidth', 1.1, 'markersize', 9);
set(findpoles, 'Color', 'b', 'linewidth', 1.1, 'markersize', 9);
set(findk, 'Color', 'b', 'linewidth', 1.1, 'markersize', 9);
hold on
[z1,~,~]=zplane(rts(Poles_idx(1:3)));
findzeros1 = findobj(z1, 'Type', 'line');
set(findzeros1, 'Color', 'r', 'linewidth', 1.2, 'markersize', 9);
xlim([-1.1 1.1]); ylim([-1.1 1.1]);grid on
title('AR Model Poles- using LPC- Phoneme \a\ ')
legend('LPC Zeros', 'LPC Poles', 'Z-plane Axes', 'Chosen Poles', 'Location', 'southeast')

% [h1,h2]=estimatePhonemeFormants(PhonemeSig,Fs, '\a\');

%% 4.5
N=(30*10^-3)*Fs; % number of samples in each frame
% win=rectwin(N);
Overlap=100;
Overlap_sam= ((Overlap)*N)/100;
FramedSig=enframe(ProcessedSig,Overlap_sam);
% Energy and ZCR
NRG=calcNRG(FramedSig);
[M,~]=size(FramedSig);
NRG=reshape((repmat(NRG,1,N))',[N*M,1]);
ZCR=calcZCR(FramedSig);
ZCR=reshape((repmat(ZCR,1,N))',[N*M,1]);
ZCR = ZCR-min(ZCR);ZCR = ZCR/max(ZCR)*max(NRG);

```

```

figure
subplot(2,1,1), plot(t(1:length(ProcessedSig)),ProcessedSig); hold on;grid on
% show segmentation lines:
for i=1:length(seg_ind)
    line([t(seg_ind(i)) t(seg_ind(i))],ylim,'color','m','linewidth',1.2)
end
xlim([0 t(end)])
title 'Segmentation - Shalom speech signal' ; xlabel 'Time [S]' ; ylabel 'Amplitude'
legend('speech','segments'); axis tight;
hold on

subplot(2,1,2)
plot(t(1:length(NRG)),NRG,'color','r','LineWidth',1.2);hold on
plot(t(1:length(ZCR)),ZCR,'color','k','LineWidth',1.2);
xlim([0 t(end)]);grid on;hold on
for i=1:length(seg_ind)
    line([t(seg_ind(i)) t(seg_ind(i))],ylim,'color','m','linewidth',1.2)
end
title ([ 'Energy and ZCR of the signal in ' num2str((N/Fs)*10^3) '[mS] window']) ; xlabel 'Time [S]' ;
ylabel 'Arbitrary amplitude'
legend('NRG','ZCR')

%% 4.6 FeatExt :
% extract features from each frame:
% \sh\
% take the first phoneme from segmentation
Phoneme_sh=ProcessedSig(seg_ind(1):seg_ind(2));
framed_Phoneme=enframe(Phoneme_sh,rectwin(N),Overlap.sam);
[FeatsVector_sh,~]=FeatExt(Phoneme_sh,Fs,framed_Phoneme);
[h11,h21]=estimatePhonemeFormants(Phoneme_sh,Fs,'\'Sh\'');
% \a\
Phoneme_a=ProcessedSig(seg_ind(2):seg_ind(3));
framed_Phoneme=enframe(Phoneme_a,rectwin(N),Overlap.sam);
[FeatsVector_a,~]=FeatExt(Phoneme_a,Fs,framed_Phoneme);

% \l\
Phoneme_L=ProcessedSig(seg_ind(3):seg_ind(4));
framed_Phoneme=enframe(Phoneme_L,rectwin(N),Overlap.sam);
[FeatsVector_L,~]=FeatExt(Phoneme_L,Fs,framed_Phoneme);
% [h12,h22]=estimatePhonemeFormants(Phoneme_L,Fs,'\'L\'');
% \o\
Phoneme_o=ProcessedSig(seg_ind(4):seg_ind(5));
framed_Phoneme=enframe(Phoneme_o,rectwin(N),Overlap.sam);
[FeatsVector_o,~]=FeatExt(Phoneme_o,Fs,framed_Phoneme);
% [h13,h23]=estimatePhonemeFormants(Phoneme_sh,Fs,'\'o\'');
% \m\ - 5
Phoneme_m=ProcessedSig(seg_ind(5):seg_ind(end));
framed_Phoneme=enframe(Phoneme_m,rectwin(N),Overlap.sam);
[FeatsVector_m,~]=FeatExt(Phoneme_m,Fs,framed_Phoneme);
% [h14,h24]=estimatePhonemeFormants(Phoneme_m,Fs,'\'M\'');

%% 6
% 6.2
% do stft with 50 % overlap
N=[ (20*10^-3) (30*10^-3) (60*10^-3)]*Fs;
figure
for i=1:length(N)
    window=hamming(N(i));
    hold on
    subplot(1,3,i)
    noverlap = ((50)*N(i))/100;
    nfft = N(i);
    spectrogram(ProcessedSig,window,nooverlap,nfft,Fs,'yaxis')
    title(['\'Shalom\'-Spectrogram. window ' num2str((N(i)/Fs)*10^3) '[mS]->N=' num2str(N(i)) ' .
50% overlap'])
    colormap jet

```

```

end

% do stft with 90 % overlap
N=[ (20*10^-3) (30*10^-3) (60*10^-3)]*Fs;
figure
for i=1:length(N)
    window=hamming(N(i));
    hold on
    subplot(1,3,i)
    noverlap = ((90)*N(i))/100;
    nfft = N(i);
    spectrogram(ProcessedSig,window,noverlap,nfft,Fs,'yaxis')
    title(['\Shalom\ -Spectrogram. window ' num2str((N(i)/Fs)*10^3) '[mS]->N=' num2str(N(i)) ' .
90% overlap'])
    colormap jet
end

```