

דוח מכין - מעבדת עיבוד אות דיבור - מעבדה 2

ג'וש לזרוס - 200138030 צפריר דרדיק - 30556698

19 בדצמבר 2018

1. כאשר נכניס אות שהוא אינו אות דיבור לתוך המערכת המערכת שלנו, ישנם הנחות מסויימות שהנחנו לגבי אות דיבור שלא יהיו נכונות עבור אותות אחרים, למשל:

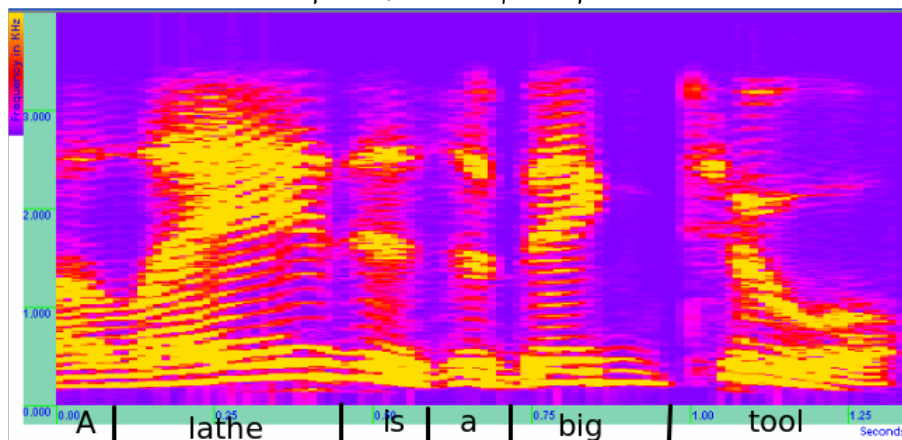
- הנחנו שאות הדיבור שלנו נמצא בתחום מסויים, כלומר ה *pitch* הוא בטווח של 50, 400 ולכן עבור אותו שיש להם הרמוניות שונות מהטווח הזה אנחנו לא נוכל לקודד אותם.
- הנחנו שאות הדיבור נמצא 2 מצבים, או קולי או א-קולי ולכן עבור למשל מוסיקה קלאסית, ייתכנו צלילים שהם גם קוליים וגם א-קוליים שלא נוכל לפענח.
- הייתה לנו גם הנחה בתחילת המעבדה הקודמת כי ה *pitch* לא חורג מתחום הדיבור. ייתכן ועבור צלילים מסויימים לא נוכל להניח הנחה זו.

2. מכיוון שבמסגרת מסויימת ייתכן שיש לנו שילוב של אות קולי וא-קולי, כאשר נשתמש בשיטת MELP אנחנו בעצם חופפים בין מסגרות, מה שיגרום למצב שההחלטה שלנו לגבי האם מסגרת היא קולית או לא תהיה הרבה יותר נכונה מכיוון שאנחנו נוכל בצורה יותר ממוקדת לזהות האם המסגרת היא קולית או לא.

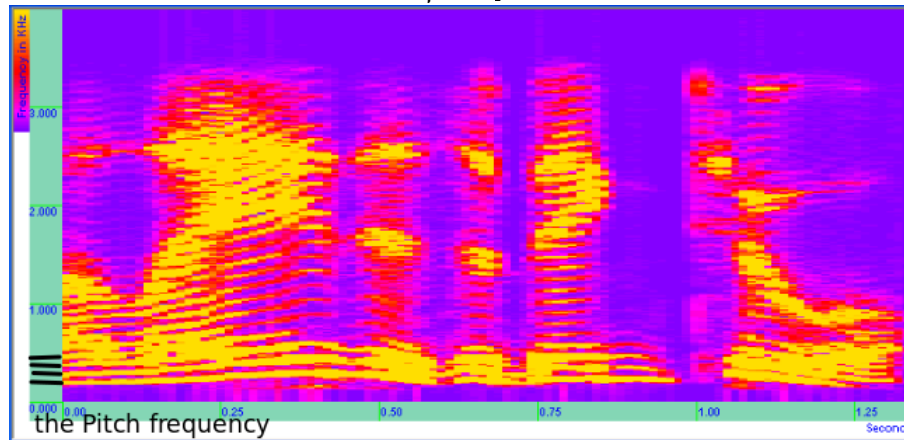
3. (א) החתך אנכי ניתן לראות עבור כל זמן את תכולת התדר. בחתך אופקי ניתן לראות עבור כל תדר כיצד הוא מתנהג בזמן, כלומר באילו זמנים הוא יותר נמוך ובאילו הוא יותר גבוה.

(ב) אם נגדיל את החלון הזמני יותר מידי אנחנו נאבד את רזולוציית הזמן, כלומר אנחנו תמיד נעשה *DFT* על חלון זמן רחב ובחלון רחב כזה אנחנו יכולים גם להיתקל ביותר מעברים בין אות קולי לא-קולי וגם הרזולוציה תקטן. אם נקטין את החלון, אמנם נקבל רזולוציה טובה יותר אך אם זאת התדרים "יימרחו" יותר ויהיה לנו יותר קשה להבדיל בין תדרים שהם קרובים.

(ג) איור 1: גרף הספקטרוגרמה עם סימון המילים



(ד) איור 2: תדר ה-pitch מסומן בציר האנכי בשחור



4. קוד המטלב:

איור 3: קוד המטלב של הפונקציה residual_energy

```
function [Eu] = residual_energy(frame,A_vec)
%UNTITLED Recieves a frame and a vector of a filter and returns
%energy of the linear approx.
% Detailed explanation goes here
filtered=filter(A_vec,1,frame);
Eu=sum(abs(filtered).^2)/(length(filtered));
end
```

(א) ה-residual_energy אחראית ל-gain מהסכמה של מקודד LPC. למעשה זהו ההגבר שעלינו להפעיל על המסגרת הקולית או האקולית על מנת לקבל עוצמת קול כפי שהייתה בכניסה. אנחנו קודם מפלטרם את האות בכדי להישאר עם תדרים נמוכים ולאחר מכן אנחנו מחשבים את ה-gain.

(ב) נשתמש בפונקציה זו במקודד. כלומר נעביר את האות המקורי דרך הפונקציה ונקבל בסוף את ה-gain.

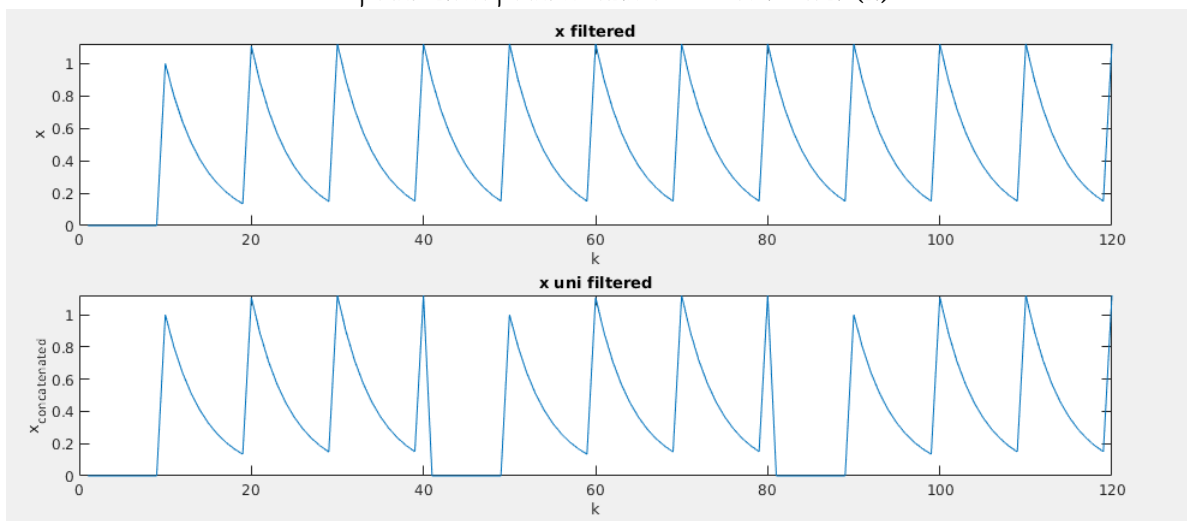
(ג) במודל החיזוי הליניארי הייתה הנחה שאותות הדיבור שלנו הם מצורה של all-pole כלומר שהוא כולו קטבים:

$$H(z) \frac{X(z)}{W(z)} = \frac{1}{A(z)}$$

כלומר אם נתון לנו כבר מקדמי הסינון אנחנו צריכים את התמסורת ההופכית, כלומר FIR.

5. סינון המקטעים:

(א) איור 4: גרפים של האות המחולק והלא מחולק



ניתן להבחין כי 2 הגרפים אינם זהים. זאת מכיוון שכאשר חילקנו את האות, ישנם מקטים שיש בהם אפסים רבים יותר וכאשר איחדנו איבדנו את המידע הזה ועל כן הגרפים אינם דומים.

איור 5: קוד המטלב הרלוונטי לחלק א'

```
N = 120;
n = 1:N;
L = 40;

x = dirac(mod(n,10));
x(x > 0) = 1;
IRR = [1 -0.8];

x_1 = x(1:L);
x_2 = x(L+1 : 2*L);
x_3 = x(2*L+1 : N);

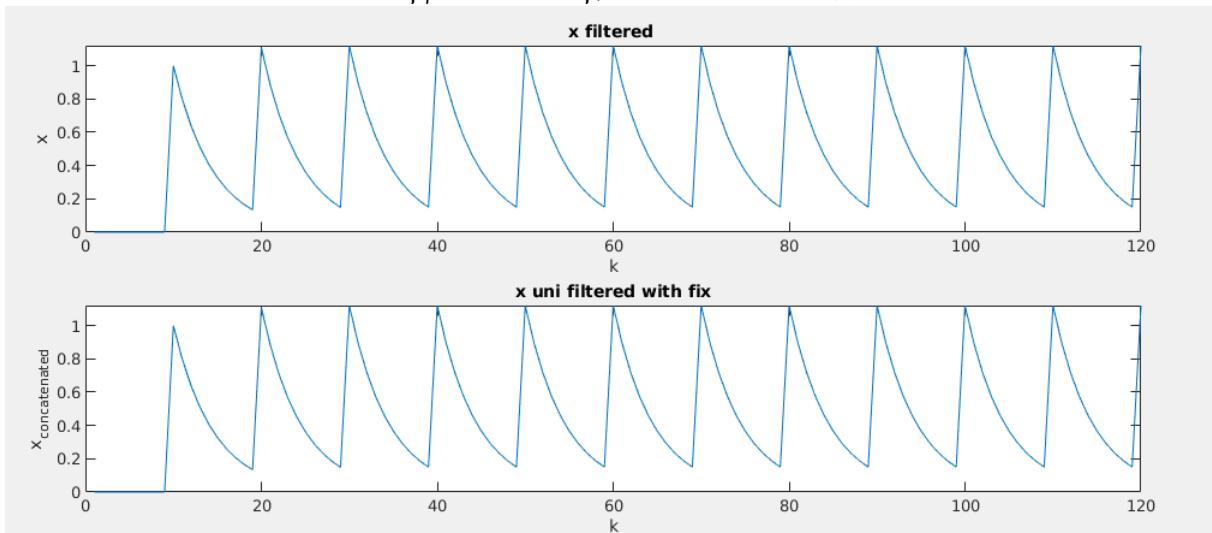
x_filtered = filter(1,IRR,x);
x_1_filtered = filter(1,IRR,x_1);
x_2_filtered = filter(1,IRR,x_2);
x_3_filtered = filter(1,IRR,x_3);

x_uni_filtered = [x_1_filtered x_2_filtered x_3_filtered];

figure(1);
subplot(2,1,1);
plot(x_filtered);
title("x_filtered")
xlabel("k")
ylabel("x")

subplot(2,1,2);
plot(x_uni_filtered);
title("x_uni_filtered")
xlabel("k")
```

(ב) איור 6: האותות בסעיף ב' אחרי התיקון



כעת נבחין כי התוצאות זהות, פונקציית ה-*filter* עשתה עבורנו את עבודת החיבור בין המקטעים. כלומר לא ניתן לזהות יותר שהיה איזה הפרדה ואז חיבור מחדש. למסנן ה-*IIR* יש תלות ביציאות הקודמות ובכניסות הקודמות ולכן יש להעביר בכל מקטע את הכניסות והיציאות של המקטע הקודם. עבור מסנן ה-*FIR* אין תלות ביציאות אלא רק בכניסות ולכן יש להעביר את כניסות העבר של המקטע הקודם.

איור 7: קוד מטלב לחלק ב'

```
%% PART B

[x_filtered_b , z1] = filter(1,IIR,x, 0);
[x_1_filtered_b, z2] = filter(1,IIR,x_1);
[x_2_filtered_b, z3] = filter(1,IIR,x_2,z2);
[x_3_filtered_b, z4] = filter(1,IIR,x_3,z3);

x_uni_filtered_b = [x_1_filtered_b x_2_filtered_b x_3_filtered_b];

figure(2);
subplot(2,1,1);
plot(x_filtered_b);
title("x filtered ")
xlabel("k")
ylabel("x")

subplot(2,1,2);
plot(x_uni_filtered_b);
title("x uni filtered with fix")
xlabel("k")
ylabel("x_{concatenated}")
```

(ג) לפי תיעוד פונקציית הפילטר עבור $P(z) = 1$ ומסנן מכנה באורך L , עלינו לשרשר $L - 1$ יציאות עבר. עבור מסנן מונה באורך $L - 1$ עדיין נצטרך לשרשר $L - 1$ ערכי עבר מכיוון שמסנן המכנה עדיין יהיה תלוי ב- $L - 1$ איברים.

(ד) נעשה שימוש ב-IIR במהלך הפיענוח מכיוון שהמסנן הזה יעזור לנו להחליק את המעטפת הספקטרלית של האות שפענחנו.

6. (א) הטעות של הסטודנט היא שהוא לא התחשב בעובדה שמרווח הדגימה לא מתחלק ב- N ולכן נוספו אפסים אחרי הדגימה האחרונה של הדלתא מה שיצר את הפער בין האות שהוא קיבל לאות הרצוי.

(ב) נמצא את הדלתא האחרונה, ונבדוק כמה אפסים יש בינה ובין סוף המקטע הראשון. לאחר מכן נוסיף את שאשר האפסים שמשלמים ל- $P - 1$ לתחילת המקטע הבא וכך נוכל להיות בטוחים שיש לנו תמיד $P - 1$ אפסים בין כל 2 דלתאות.

(ג) איור 8: הפונקציה במטלב

```
function [x] = P_seg(segment_length,P,segment_num)
%creates a periodic impuls train
% inputs
% x = signal
% N = # of samples
% P = # of prtitions
%outputs
% y = unifide segmented signals vector(continues)

%create delta of lenght segment_length|
x=zeros(1,segment_length);
x(1:P:segment_length)=1;

%here we concatenate the segments
for i=2:segment_num
    delta_placement = P+1-mod(segment_length*(i-1),P);
    if delta_placement>P
        delta_placement=1;
    end
    x((i-1)*segment_length+delta_placement:P:i*segment_length)=1;
end

figure(2); stem(x);

end
```

7. הפתרון הראשון כנראה פחות טוב מכיוון שכל מטרתינו היא לבדוק כמה האות המקורי דומה לאות המשוחזר וכאשר אנחנו מתעלמים ממקטעים שהם דומים (כלומר שהשחזור שלהם התבצע בצורה טובה) אנחנו למעשה מתעלמים מההצלחה הזו. לכן הפתרון השני הוא הטוב יותר עבורנו.