

# תרגיל בית מספר 1

מגישים:

\* לינוי אלימלך 319122610

\* ניתאי כהן 311502389

## שאלה מספר 1:

נתונה פונקציה האוטוקורלציה:

$$r_k = \sum_n^{N-k-1} X_n * X_{n+k} \quad k = 0, \dots, N-1$$

1. נוכיח כי  $r(k)$  סימטרית:

1.  $r_k = \sum_n^{N-k-1} X_n X_{n+k}$

אנחנו מניחים שזאת פונקציית האוטוקורלציה של אות  
באורך  $N$  פזמר  $X_n = 0$  עבור  $n < 0$  או  $n \geq N$ .  
במקרה כזה א הוא פירמטל הפזמר של פונקציית  
האוטוקורלציה.

$\rightarrow r(-k) = \sum_n^{N+k-1} X_n X_{n-k}$

נבצע הקרפ:  $t = n - k$ , ונניח גם כי אחרת  
הסימטריות לא מסתברת שבסבס מתפל  $N$   
היכחור פזמר ניתן עשות אם נניח שפזמר מתפלס  
מתפלס  $[X_n, X_{N-1}]$ .

$\rightarrow r(-k) = \sum_{t=-k}^{N-1} X_t X_{t+k} = \sum_{t=0}^{N-1} X_t X_{t+k} + \sum_{t=-k}^{-1} X_t X_{t+k}$

$= \sum_{t=0}^{N-k-1} X_t X_{t+k} + \sum_{t=N-k}^{N-1} X_t X_{t+k}$

$= \sum_{t=0}^{N-k-1} X_t X_{t+k} = r(k)$

2. כאשר  $k=0$  מתקבל הביטוי הבא:

$$r_0 = \sum_n^{N-0-1} x_n \cdot x_{n-0} = \sum_n^{N-1} (x_n)^2$$

ראשית מהגדרה ניתן לראות כי זוהי אנרגיית האות. זהו גם מקסימום גלובלי. ניתן להבין זאת משום שזוהי בין שני וקטורים זהים. כל אלמנט (וזו לא משנה אם הוא חיובי או שלילי) יוכל באותו האלמנט. כלומר נסכום רק ערכים חיוביים, וגם נראה הערכים הגדולים ביותר (מבחינה אבסולוטית) מוכפלים גם כן בערכים הגדולים ביותר (גם כן מבחינה אבסולוטית). לכן לא ייתכן סידור אחר של איברים שייתן לנו תוצאה גדולה יותר. דרך קלה יותר לראות זאת היא מכך שאנו יודעים שפונקציית האוטוקורלציה היא מדד לכמה האותות דומים. עבור  $k=0$  זהו למעשה הדמיון של האות עם עצמו. עבור כל ערך  $k$  אחר, הדמיון יהיו עם אותו מוסט, מה שכמובן פחות דומה.

3. הקורליציה הכי גבוהה מתקבלת עבור האות עצמו - האות הוא מחזורי אז יחזור על עצמו בכפולה של  $p$ .

$$r_{(k+lp)} = \sum_{n=0}^{N-k-lp-1} x_n \cdot x_{n+k+lp} \stackrel{(*)}{=} \sum_{n=0}^{N-k-lp-1} x_n \cdot x_{n+k} = r_k$$

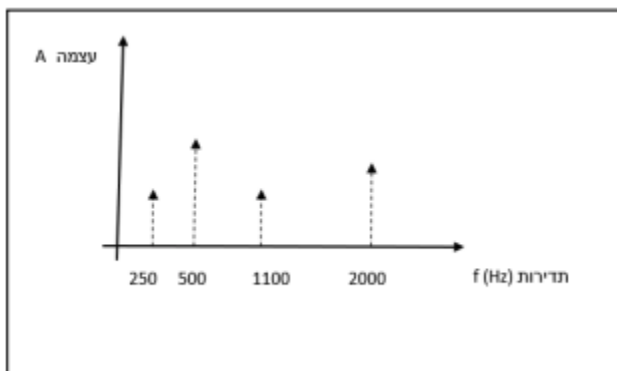
↓

$$x_{n-k-lp} = x_{n+k}$$

$$\Rightarrow r(lp) = r(0) \rightarrow l = \pm 1p, \pm 2p, \pm 3p, \dots$$

4. ערכי המקסימום של אנרגיית האות  $R_k$  יתקבלו סביב  $k=lp$  לכל מספר ממשי וזה בעצם מתורגם למספר הזזות בין כל מחזור שיתורגם לחישוב הpitch.

## שאלה מספר 2:



מערכת נדרש לדגום אות אנלוגי בעל המאפיינים הבאים:

- בתדר: מורכב מארבעת התדרים הבאים לפי השרטוט.
- בזמן: אמפליטודת האות היא  $\pm 1 \text{ Volt}$ .

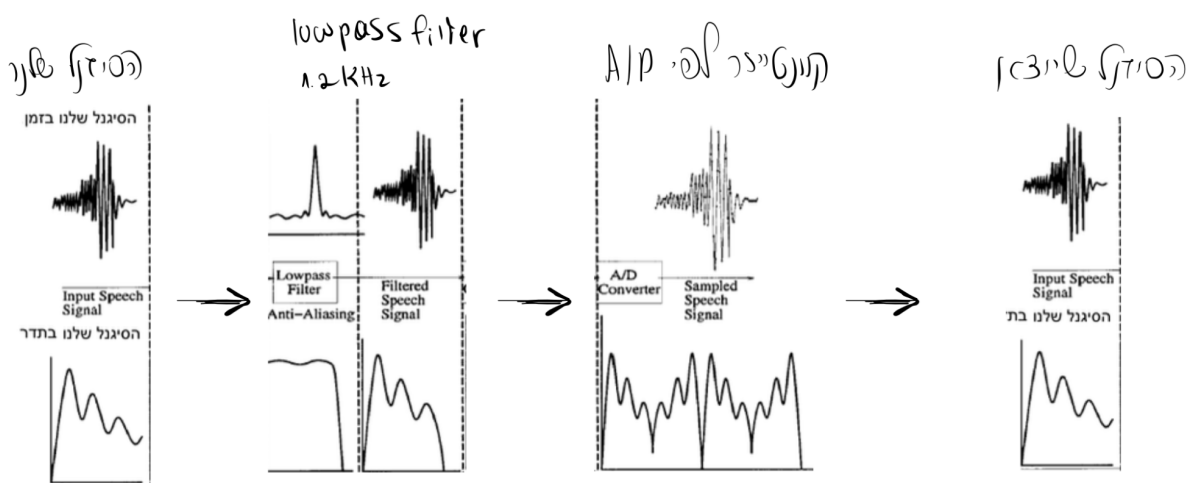
א. התדר המקסימלי שנרצה להאזין לו הוא 1.2 KHZ לכן התדר דגימה נדרש הוא 2.4 KHZ - פי 2. נשתמש במסנן lowpass filter שמחפש תדר גבוהה ושינוי בזמן. נחשב את רזולוציית הסיביות:

$$\Delta \approx 2 \cdot 10^{-3} = 2 \text{ ms}$$

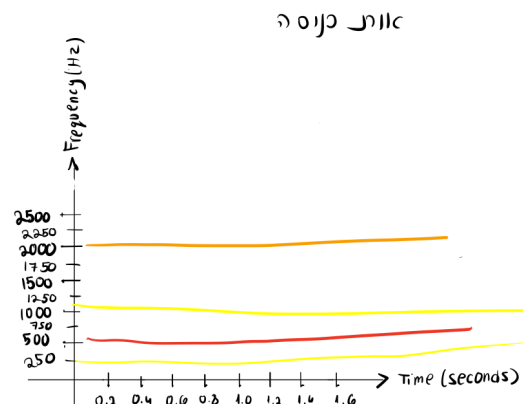
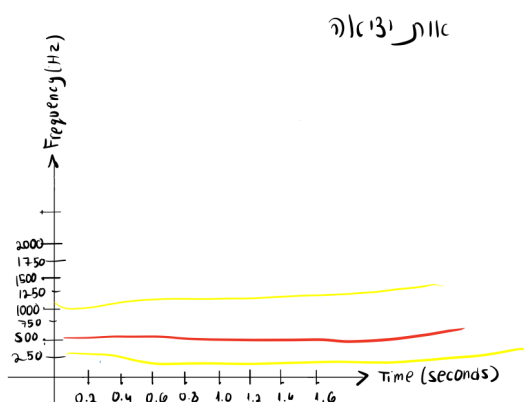
$$\frac{1 - (-1)}{\Delta} = \frac{2}{2 \cdot 10^{-3}} = 1000$$

$$1000 \rightarrow 1024 \rightarrow 10 \text{ bits}$$

דיאגרמה של מערכת דגימה מלאה:



ב. ציר זמן אל מול ציר תדר: מייצג את עוצמת המתדרים כתלות בזמן



הסבר:

- \* לשים לב שאדום הוא הצבע החזק יותר, וסמנו שתי תדירויות באותו הצבע כי הן מאוד קרובות בעוצמה
- \* באות היציאה התדר הגבוה סונן על ידי הפילטר (1.2 KHz)

ג. בסעיף א ראינו כי ישנם 10 סיביות ותדר הדגימה הנדרש הוא 2.4 ועל כן החישוב הבא:

$$f = 2.4 \text{ KHz} = 2400 \text{ Hz}$$

$$f = \text{קצב הסימנים} \times \text{מספר סיביות}$$

$$\Rightarrow 10 \cdot 2400 = 24,000 \text{ bps}$$

### שאלה מספר 3:

1. שיטת קידוד DPCM - בשיטה הזו נעביר ממש את גודל השגיאה למספר הרמות שנרצה. הוא מקודד לא רק את כיוון השינוי אלא גם את גודלו וכיוונו.  
שיטת קידוד ADPCM - משתמשים בקוואנטיזר אדאפטיבי זה אומר שהקוואנטיזר יהיה ניתן להתאמה ונכניס לתוכו קונטרולר כך שבכל פעם שרמת הייצוג של הסיביות משתנה בהתאם לגודל השגיאה. בעצם במילים שלנו זה אומר שהוא מקודד את גודל שגיאת החיזוי ולא רק את הכיוון שלה ומסוגל לשנות את צעד הכימוי.  
אז היתרונות של ADPCM לעומת DPCM - הגישה האדפטיבית של ADPCM יתן לנו מעקב טוב יותר מ DPCM לפתרון בעיית העכבה, נוכל גם להקצות דיוק משתנה לאורך תהליך הקידוד (כמות ביטים שונה). ההבדלים המהותיים יופיעו כאשר השינוי באות הוא קיצוני ומהיר ולא בשינויים קטנים.  
החסרונות של ADPCM לעומת DPCM - החישוב של ADPCM דורש סנכרון של הקוונטיזר והיא מורכבת יותר כי מנסה לחזות את המשתנים... חישוב מסובך וארוך
2. אין יכולת חיזוי לגבי רעש לבן כי הוא אקראי ולא ניתן לחיזוי- לכן קידודים מסוג DM PCM ו DPCM יכשלו כי אין מידע מוקדם על ההתנהגות.
3. (ניתן להניח שהדגימות הקודמות והבאות כולן בעלות הערך: 100)  
 $X(n) = [98, 98, 100, 102, 105, 103, 101, 99]$

$X_n$	98	98	100	102	105	103	101	99
מיצוץ 2 דגימות	100	99	98	99	101	103.5	104	102
E	-2	-1	2	3	4	0.5	3	3

$$MSE = (4+1+4+9+16+0.25+9+9)/8 = 6.53125$$

השגיאה תהיה 0 כשלא קיים שינוי באיברי הסדרה. שגיאת החיזוי תקטן ככל שדגימת האות תהיה קרובה לדגימות הקודמות שלה.

Xn	98	98	100	102	105	103	101	99
מיצוע דגימה קודמת ובאה	99	99	100	102.5	102.5	103	101	100.5
E	-1	-1	0	-0.5	2.5	0	0	1.5

$$MSE = (1+1+0+0.25+6.25+0+0+2.25)/8 = 1.34375$$

השגיאה תהיה 0 כאשר השיפוע של האות אחיד. שגיאת החיזוי תקטן ככל שיש פחות שינויים בשיפוע.

#### שאלה מספר 4:

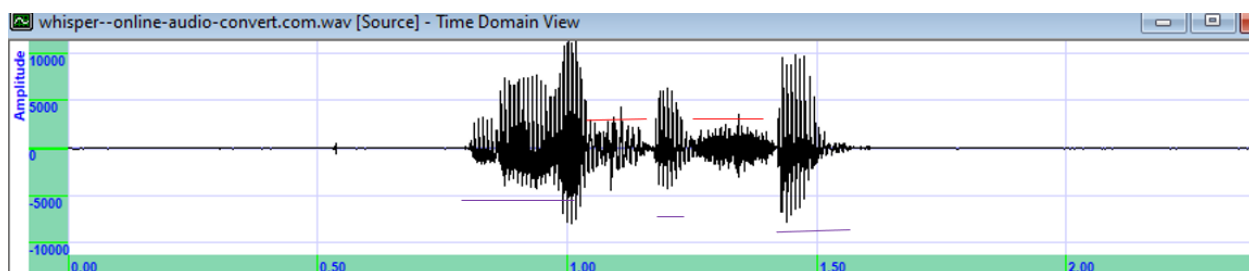
א. קצב הסיביות במוצא המקודד לא ישתנה בגדול, זהו פרמטר של ההתקן. אם היינו מייצגים יותר מקדמים, אז כמות הביטים בה נשתמש כדי לקודד כל מסגרת יגדל (בערך ב-8 ביטים). מאחר ואנחנו לא יכולים להוציא יותר מ-2400 ביטים בשניה, אבל כעת יש לנו 44 מסגרות שכל מסגרת מקודדת ב-62 סיביות, מה שייתן לנו 2728 ביט, לא ניתן להוציא זאת מהמקודד. לא נכנסנו בדיוק למה יעשה המקודד במקרה כזה, אבל הייתי מניח שהוא יעביר 38 מסגרות (כי 39 הוא לא יכול להעביר) מה שאומר שבכל שניה הוא 2356 ביט בשניה.

ב. ככל הנראה יהיה שינוי קל באיכות הקול במוצא המקלט, וסאונד מעט ייפגע. אומנם נרוויח קצת יותר על כך שהקודדנו יותר מקדמים, ואולי הסאונד במסגרות שאנחנו כן מעבירים יהיה מעט טוב יותר, אבל לא ניתן יהיה להעביר את כל המסגרות. כלומר אם נעביר רק 38 מסגרות, אבדנו למעשה 6 מסגרות שהן כ-0.16 שניות. לא נכנסנו לכך בשיעור, אך אני חושב שהמקלט אמור לדעת שהוא לא קיבל את כל המסגרות, וככל הנראה ישכפל את המסגרת האחרונה. מאחר שאיבדנו חלק מהשמע, ככל הנראה איכות השמע מעט מעט תפגע. אומנם יש כאן טרייד-אוף שקצת קשה להעריך אם הוא תורם או פוגע, אך מאחר וההתקן הרשמי הוא עם 10 מקדמים, ותוך מחשבה שהוספת שני מקדמים תתרום פחות מהפגיעה באיבוד חלק מהשמע.

ג. הקטנת מסגרת השמע ל-160 מציגה גם כן טרייד-אוף. הייתרון בהקטנת המסגרת הוא שניתן יהיה לתאר אותה בדיוק רב יותר. לעומת זאת ישנם שני חסרונות. חסרון אחד הוא שייתכן וככל שנקטין את המסגרת לא נצליח בכלל לתפוס מה הוא הפיצ'. שינוי קטן כזה בגודל המסגרת, ככל הנראה לא יגרום לנו לאבד את הפיצ' במסגרות רבות, אך מצד שני כנראה גם לא יתרום רבות לדיוק בה נקודד את המסגרת. חסרון נוסף, הוא שאם אנחנו מעבדים את הסאונד במסגרות של 160 דגימות, אנחנו יהיו לנו  $160/800 = 0.2$  מסגרות ואנחנו נצטרך 2700 ביטים לשניה במוצא המקודד, מה שאנחנו לא יכולים לתת כי 2400 ביט בשניה הוא פרמטר של המקודד. גם כאן נאבד למעשה מסגרות. ב-54 ביט למסגרת ניתן לשדר רק 44 מסגרות מה שאומר שאנחנו מאבדים 6 מסגרות שהן כ-0.12 שניות. בדומה לסעיף הקודם אני מאמין שלא תהיה פגיעה מהותית באיכות השמע, אולם אני חושב שהחסרונות יעלו על הייתרונות (וזו גם הסיבה שהמספר בו אנו משתמים הוא 180 דגימות במסגרת).

## שאלה מספר 5:

- א. הקובץ מיוצג ב-16 סיביות לדגימה.
- ב. 12 סיביות.
- ג. ככל שאיכות האות גבוהה יותר כך SNR גבוהה יותר. כאשר אין רעש הערך הוא אינסופי (כי הרעש הוא 0), וככל שהרעש גדל כך SNR קטן ואף יכול להיות שלילי.
- ד. ניתן להסיק ש-adpcm יעיל יותר מ-pcm. ניתן לראות זאת גם מהתבוננות מגרף השגיאה, גם מהאזנה לו, וגם מההאזנה לסאונד שעבר כימוי בו ניתן לשמוע הרבה פחות רעש כאשר משתמשים ב-adpcm. ניתן כמובן לראות זאת גם מהתבוננות ב-snr. עבור adpcm קבלנו כ-24db, ועבור pcm קיבלנו כ-15.8db. באופן כללי אם נשתמש ב-adpcm טוב, נקבל תוצאות טובות יותר משום שניתן גם להחליט לא לעשות שום דבר אדפטיבי ואז נקבל Pcm למעשה.
- ה. האזנו.
- ו. עבור הקטע הקולי הראשון ב-speech0 אנו רואים 3 פורמנטים, כאשר הפורמנט הרביעי הוא מחוץ לטווח שאנחנו רואים בתמונה. הפורמנט הראשון הוא ב-587Hz. ישנם קטעים קוליים נוספים בהם רואים גם 4 פורמנטים בתוך התווך שנראה בתמונה.
- עבור הקטע הקולי הראשון ב-speech8 אנו רואים 4 פורמנטים כאשר הפורמנט הראשון הוא ב-570Hz. גם ביתר הקטעים הקוליים נראה גם כן שיש 3-4 פורמנטים.
- באיזורים שאינם קוליים נראה הפורמנטים לא נראים היטב.
- באופן כללי, על סמך הקטעים הללו בלבד קשה להבדיל בין גבר לאישה. נראה שהקול הנשי הוא בעל פורמנטים מעט גבוהים יותר וכן בעל פיצ' גבוה יותר, אך נראה נראה שהוא מדד שקשה להסתמך עליו באופן מובק. אנחנו רואים הרבה מקומות בהם הפורמנט של האישה נמוך, והפורמנט של הגבר גבוה. אצל הגבר רואים פיקים מעט חדים יותר בפיצ'. מהתבוננות בשני האותות שקבלנו דווקא נראה שמדד הפיצ' מבדיל קצת טוב יותר בין גבר לאישה, כאשר לגברים פיצ' נמוך יותר. אני חושב שקשה לזהות אם הקול הוא גברי או נשי רק על סמך הפרמטרים הללו, משום שאנחנו יודעים שאורך ומבנה המסלול שעובר האוויר משתנה מאוד בין בני אדם.
- ז.
- המילה שהוקלטה היא "לחישה" כאשר באדום סומנו הקטעים הא-קוליים שהם תחילת ה"ח" וה-"ש", ובסגול סומנו הקטעים הקוליים (שהם ה"לה", "אי", "אה"). בקטע הקולי הראשון ניתן לראות 4 פורמנטים ב-454Hz, 1659Hz, 2567Hz, 3491Hz. הפיצ' התקבל להיות 116.5Hz. העיבוד של הפורמנטים והפיצ' נעשה על החלק של ה"אה" שבא אחרי ה"ל".



ח. Vega.wav נשמע ממש טוב גם לאחר מעבר דרך הטלפון, וניתן לראות בעין גם שהאות נראה דומה לאות המקורי. לעומת זאת Depech Mode.wav נשמע פחות טוב, ובעיקר פחות עשיר מהאות המקורי. החלק הקולי נשמע דיי טוב, אך החלק המוזיקלי לאורך כל הקטע פחות עשיר ולא מכיל את כל הכלים והצלילים מהאות המקורי. ההבדלים בין האותות המקוריים לאותות החדשים נובעים מהעובדה שהאותות מועברים דרך מודל המיועד לקידוד דיבור בני אדם, ולא למוזיקה ושירה. מאחר ושירה כפי שקיימת בשירים אלו דומה מאוד לדיבור, אז היא נשמעת טוב, אך המוזיקה לא נשמעת טוב. כפי שציינתי ההבדל העיקרי בDepech mode.wav הוא בעושר של המוזיקה, לא ניתן לשמוע כמעט תופים ובאס באות החדש, וזאת משום שהמודל הטלפוני לא יודע להתמודד עם תדרים נמוכים ומיועד לתדרים החל מכ-300Hz ועד לכ-3.4KHz.

## שאלה מספר 6:

- א. אורך התוצאה יהיה  $1 - \max(n, m) * 2$  כאשר למעשה מה שקורה זה שהפונקציה ראשית מרפדת את הוקטור הקצר יותר באפסים, כך שכעת יש לה שני וקטורים באותו האורך. כעת הפונקציה תבצע הזזה (leg) של אחד הוקטורים בצעד אחד כל פעם. בצעד הראשון תהיה חפיפה רק של האלמנט הראשון והאלמנט האחרון של שני הוקטורים, ובצעד האחרון תהיה חפיפה הפוכה בדיוק בין האלמנט האחרון לאלמנט הראשון. בכל צעד תחושב הקרוס קורלציה וסה"כ נקבל  $\max(n, m) * 2 - 1$  תוצאות. נשים לב כי זוהי התואה הדיפולטית, אך ניתן לשחק עם הפרמטרים ולקבל גדלי מוצא שונים.
- ב. הערך המירבי יתקבל עבור ההזזה בה האותו הם הדומים ביותר (ערך הקרוס קורלציה הוא הגבוה ביותר). במקרה שלנו זה התקבל עבור 411086 הזזות, ועל ידי חלוקה ב-sampling rate ניתן להבין באיזה זמן בדיוק.

- א. כל העניין בפונקציה הוא להכניס ערכים שונים לתוך אותו התווך. אנחנו עושים זאת על ידי פונקציית floor. הפונקציה יודעת מראש שהיא מקבלת ערכים בין 1 ל-1.1. וכתלות ב-N ולאחר החלוקה ב נקבל את הערכים המבוקשים. אלו הם כולן ערכים גדולים מ-0 וקטנים מ-2 כך שהחיסור

בשורה האחרונה ייתן לנו ערכים בין 1-1 אך נוכל לקבל רק מספר ערכים קטן שניתן לייצג ב-  
ביטים.

ב.

```
[18] import numpy as np
      def QuantizeWave(X, n):
          X_q = np.floor((X+1)*(2**(n-1)))
          X_q = X_q / (2**(n-1))
          X_q = X_q - (2**n - 1)/(2**n)
          return X_q
```

```
[22] def snr(in_vec,out_vec):
      noise = in_vec-out_vec
      return np.var(in_vec) / np.var(noise)
```

```
import numpy as np
import matplotlib.pyplot as plt
import librosa
x, sr = librosa.load("vega.wav")
Snr_table = np.zeros(15);
waveData = x
for n in range(1,16):
    waveData_q = QuantizeWave(waveData,n)
    Snr_table[n-1] = snr(waveData, waveData_q)

#plot snr
fig = plt.figure(figsize = (10, 5))
bit_depth = np.arange(1,16)
# creating the bar plot
plt.bar(bit_depth, 10*np.log(Snr_table), color = 'maroon', width = 0.1)

plt.xlabel("Bit depth")
plt.ylabel("SNR")
plt.title("SNR as function of quantized bit depth")
plt.show()
```



