

# עיבוד ספרתי של אותות סמסטר א תשע"ט

בית הספר למדעי המחשב  
אוניברסיטת ת"א

## כותבים, עורכים ועוזרים:

ספיר בגימוב – כותבת ועורכת

בר שופן – אחראי נוסחאות ועוזר המחשות

אביב יניב וריה (מהסיכום בדרייב) – עוזרים בהמחשה (חלק מהתצלומים)

כל הזכויות שמורות

## הערות:

- שיעור 3-4 לא הספקתי לכתוב הכל (/ כ"כ הרבה)
  - כל מה שנאמר בשיעור חשוב, לקראת הלימוד למבחן (גם מה שלא נראה כ"כ), לכן מומלץ לכתוב הכל. גם נק' היסטוריות
  - הסברים כלליים בנוגע ל"איך לעשות דברים / חישובים/ להסתכל על דברים" חשובים ביותר, גם למבחן!
  - לא כתוב לי כמעט בכלל על FFT, באחריותכם להוסיף.
  - העלתי בגירסת וורד, כדי שמי שירצה, יוכל להוסיף על מה שכתוב, כאן. ולתקן ולהבהיר, כמובן, אם יש צורך.
- אבל אם אתם עושים זאת – תעדכנו את המרצה, תשלחו לו קובץ מעודכן, תעדכנו בדרייב את הקובץ. **תשתפנו!**
- כי מה אכפת לכם? את רוב עבודת הכתיבה, התיקון העריכה, וכו', לא אתם עשיתם. מי שעושה זאת - תרגישו חופשי להוסיף את שמכם לכותרת הקודמת (דה ☺)

## טכני:

חובת נוכחות- אם לא מגיעים לשיעורים, לא ניתן להגיע למבחן  
חיסור של 1-2 מקובל, יהיה רשימת נוכחות מהשבועות הקרובים  
הספר: DIGITAL SIGNAL PROCESSING A COMPUTER SCIENCE PERSPECTIVE

האתר: [www.dspscsp.com/tau](http://www.dspscsp.com/tau)

גישה: במייל, עדיף באנגלית, בשורת הנושא \_\_\_\_\_ DSP

[AUTHOR@DSPCSP.COM](mailto:AUTHOR@DSPCSP.COM)

התרגילים לבית ברמת מבחן, אין חובת הגשה ואין בודק

## DIGITAL SIGNAL PROCESSING

digital\* signal processing - בזה נעסוק- עיבוד ספרתי של אותות = עיבוד אותות באופן ספרתי, נניח שיש לנו סיגנלים ספרתיים ונראה מה אפשר לעשות איתם, אמנם אנו לא ניקח סיגנלים מהעולם ונהפוך אותם לאוסף של מספרים (זו עבודה של מהנדסים).

digital\* signal\* processing – עיבוד אותות ספרתיים.

## דוגמאות לעיבוד אותות ספרתי:

### RC505- לופר

בו משתמשת נטע ברזילי, אמנם יש לו בערך 500 פיצ'רים

כל הלופרים יכולים לסדר את הזמנים- שכל הקטעים יהיו באותו אורך, וגם לעשות הד או לעשות רברס, לשנות את התדר (PITCH) ועוד, לאחר הקלטת הקול. זה הקשר ל DSP.

### אורגנית

בד"כ הקולות של האורגנית הם הקלטות- יש הקלטה של כל מיני כלים.

### אלגוריתמים

ניתן לכתוב אלגוריתם שמפיק צליל של כלי נגינה. זה מודל לכלי נגינה, לא קול, אלא אלגוריתם שיש לו לופים והכפלות, ואין בו שום הקלטות.

לנגן קטע מהסוף להתחלה- מאוד קל לעשות בצורה דיגיטלית (יש לנו דגימות). קשה מאוד אנאלוגית- אפשר לשים על סרט מגנטי ולהפוך את הסרט.

לקחת קטע שעשו עליו כבר reverse (הפכו אותו) ולעשות עוד משהו שמדמה הד, כאילו נמצאים בתוך חדר סגור יחסית קטן. זה נותן את מה שנקרא reverb.

להפוך את reverb הזה חזרה לכיוון הרגיל. נקבל אפקט מאוד מוזר- משהו שלא שומעים אף פעם בטבע- הד הפוך. זה מה שאנו קוראים לו- לא סיבתי- כי שומעים את ההד לפני הקול.

### טלפון קווי

עובד בצורה תקינה ב%99.999 ("5 תשיעיות") מהזמן

הקווים של הטלפון מעבירים גם את החשמל וגם את המידע היוצא והנכנס. כי החשמל זה 3-9 יות, ולא 5 כמו טלפון. ז"א כי רצינו שיהיה טלפון כשאינן חשמל (כשהוא נופל).

### צליל חיוג-

מחוללת המרכזייה שלי, בהרמת השפופרת, כדי לסמן לנו שניתן להתחיל לחייג ולא צריך לחכות למרכזנית. בתחילת החיוג הוא נפסק. הוא לא נעים, כי רוצים שנתחיל לחייג כמה שיותר מהר, כדי שנתפוס לכמה שפחות אנשים את הקו וכדי שהם יוכלו להתחיל להרוויח כסף. לכן, אם לוקח לנו יותר מדי זמן לחייג, הצליל נהיה צורם יותר.

מורכב מ2 סינוסים - צלילים בסיסיים, וניתן לשמוע באוזן את הפרש התדרים. כאשר שמים 2 סינוסים קרובים אחד לשני זה גורם לצליל להיות צורם.

### צליל הקשה-

בהקשה (/ חיוג) הספרה המבוקשת נשמע צליל. מזהים כי בין הספרות יש צלילים שונים ומרחקים משתנים, כך מזהים שאדם עושה זאת, כי חיוג של מכשיר הוא מהיר.

### צליל תפוס-

מחוללת המרכזייה שלי, בזיכרון אינפורמציה ספרתית- דיגיטלית, שעוברת מהמרכזייה של הטלפון אליו חייגתי.

יכול להישמע גם כאשר מקישים את המספר לאט מדי, או כשלא מקישים בכלל יותר מדי זמן.

גם הוא לא נעים כדי שנתק את הקו כמה שיותר מהר, כי היום אנחנו משלמים רק על זמן שיחה, ולא על זמן אוויר, ככה שכל זמן שהשפופרת מורמת- החברה "מפסידה" עלינו כסף/ שיחות/ חשמל.

### צליל הצלחת חיוג- RING- BACK-

מחוללת המרכזייה שלי, כשהחיוג מצליח, כשהטלפון אליו חייגתי מצלצל בצד השני. אמנם לא אומר שזה בדיוק מצלצל הרגע בדיוק כשאנו שומעים את הצליל, אלא זה אומר שזה מצלצל ואני אמור לחכות שמישהו ירים את הטלפון.

איך המרכזייה שלי יודעת אם הקו תפוס? יש רשת דיגיטלית שמעבירה אינפורמציה סיפרתית. כל הצלילים האלה הם אלגוריתמים שרצים במרכזייה שלי.

## פקס

פקס הוא סוג של מודם, והמודם הזה הוא מהיר, לפני הADSL, שעדיין אפשר לשמוע אותו. הADSL הוא כ"כ מהיר שאי אפשר כלל לשמוע אותו.

### צליל חיוג והקשות-

תחילת החיוג והקשות מהירות עם זמן קבוע ביניהן, מחוללות ע"י המרכזייה שלי.

### צפצוף- calling tone

לא הרימו בצד השני, אז הפקס שלי עושה את הצליל הזה.

סינוס טהור (1000 Hz), נשמע כחליל. אם הפקס מתקשר למספר טלפון שאינו פקס, ינתק, או אם אנו מרימים טלפון ושומעים את הצפצוף הזה- אנו יודעים שבצד השני יש פקס וננתק.

### צליל גבוה ממושך-

כאשר הפקס בצד השני מרים את הטלפון ושומעים צליל כזה, זה בעצם הפקס העונה שאומר שהוא פקס.

לאחר מכן שומעים את הצליל המוכר של הפקס, המידע שהוא מעביר בעצם, באותות. בצליל זה יש הרבה אינפורמציה בין ביטים ליח' זמן.

בעצם יש כאן תוכנה- אלגוריתם. המרכזייה בפנים, יש לה אלגוריתם שאומר- אם מישהו מרים טלפון- תתחיל להשמיע את צליל החיוג וכו'. אז במערכת של הטלפונייה יש אלגוריתמים.

### אינפורמציה באותות ורעש

אינפורמציה- זו הפתעה- אומר לי משהו שלא ידעתי קודם.

כל עוד ששומעים צליל שנשמע מלודי, יש שם אינפורמציה אבל לא הרבה. ככל שהוא רועש יותר ככה יש יותר אינפורמציה. ברעש לבן יש הכי הרבה אינפורמציה.

אינפורמציה מתנדפת עם הזמן, אבל היא עושה סדר, כאשר אנטרופיה עושה בלאגן אנטרופיה = אינפורמציה \* (-1).

### למה אנשי מדמח צריכים ללמוד DSP?

פעם זה היה רק למהנדסים, שעשו עיבוד אותות אנאלוגי.

לדוג': מהנדס חשמל בונה מעגל שלוקח אותות ומעבד אותם ומוציא אותות אחרים. המון שנים עשו עיבוד אותות בצורה אנאלוגית.

### יתרונות עיבוד אותות בצורה ספרתית:

#### 1. הגמישות.

- a. אם מהנדס בונה מעגל שעושה איזושהי עבודה מסוימת והוא רוצה לעשות משהו אחר, אפילו מאוד דומה- הוא צריך לבנות מעגל חדש.
- b. בניית מעגל, תכנון המעגל, ייצור המעגל, הרכבה- אלה דברים שלוקחים המון זמן ומשאבים.
- c. מנגד, אם יש לי תוכנה ואני רוצה לעשות משהו דומה אך שונה- זה די טבעי ומהיר. אז אני יכול לעשות דברים הרבה יותר גמישים עם מחשב מאשר מה שאפשר לעשות עם מעגלים.
- c. אמנם אפשר לשים כמה כפתורים- אם חשבתי על זה מראש, אמנם לא יותר מזה.

#### 2. הדייק

- a. כאשר רוצים להגביר סיגנל, הדייק של מעגל אלקטרוני הוא לרוב הרבה יותר נמוך מאשר של תוכנה.

העבודה שנוכל לעשות במחשב היא באמת עבודה יותר טובה, יותר מדויקת, יותר יציבה ויותר מתוחכמת.

### מגברים

היום משתמשים בטרנזיסטור, שיכול לקחת קלט קטן ולהפוך אותו לפלט גדול. אף מעגל אלקטרוני לא יכול באמת לעשות את הפעולה הזו, אלא רק להתקרב לזה.

הטרנזיסטור יכול לעבוד רק במתחים מסוימים (מינימלי ומקסימלי). אם מעבר למתחים שלו, הוא לא יכול להגיב. כאשר הוא מקבל מתח מעבר למקסימלי/ המינימלי שלו- הוא נחסם.

הוא אפילו לא כ"כ לינארי- הוא מקרב בלבד. ככל שאני מתכנן את המעגל שלי יותר טוב אני יכול לעשות קירוב יותר/ פחות טוב.

פעם השתמשו במגבר שפופרות, הקירוב שלו נוראי לסיגנל המקורי והרבה יותר גרוע מהטרנזיסטורים.

משמע, ניתן לקחת אות בעולם האמיתי, להפוך אותו למספרים ולעשות עליו עיבודים עם מחשב, שיהיו הרבה יותר מדויקים מאשר המעגלים האלקטרוניים וקירוביהם, ושליחתם בחזרה לעולם. כי הכפלה של מספר זו פעולה מאוד פשוטה במחשב.

### אותות

סוג האות	אות אנלוגי-	אות ספרתי/ דיגיטלי-
הגדרה	s(t), t is time	Sn סדרה, לא פונ'
תחום	ממשי	ממשי
תחום המשתנה	-inf. <= t <= inf.	n = -inf.... Inf
אנרגיה	$E_s = \int_{t=-\infty}^{\infty}  s(t) ^2 dt$	$E_s = \sum_{n=-\infty}^{\infty}  s_n ^2$
תנאים נוספים	הפונ' חייבת להיות בעלת אנרגיה סופית- FINITE ENERGY ובעלת רוחב פס/ סרט סופי- BOUND WIDTH	אנרגיה סופית רוחב פס סופי

יכול להיות שהיו דוגמאות שלא הספקתי לכתוב?

## הסבר התנאים:

אנרגיה היא משהו שאני מוכן לשלם עבורה, כי אי אפשר לייצר אותה לבד, בגלל חוק שימור האנרגיה ולמרותו, לכן ברור מאליו שעל אינפורמציה, שמתנדפת עם הזמן, שאהיה מוכנה לשלם עליה.

אמנם בגלל זה אנו דורשים אנרגיה סופית ורוחב פס סופי, כי אחרת, זה יעלה לנו אינסוף, ולאף 1 אין כ"כ הרבה כסף לשלם על זה.

ויותר מכך, אנו רוצים משהו רלוונטי לעולם שלנו, לכן גם הפונק' ממשית

\* אמנם אנו לא באמת מאמינים שתמיד זה 0, או K, אלא רק בנק' הזמן שבה אנו מסתכלים

## סוגי סיגנלים ודוגמאות:

פונקציה- מהנדסים	סדרה- מדמח	
$s(t)=0$ לכל t	$S_n=0$ לכל n	<b>סיגנל ה-0*</b>
$S(t)=k$ לכל t	$S_n=k$ לכל n	<b>סיגנל ה-DC*</b>
(של דיראק, $\delta_{n,k} = \begin{cases} 1, n = k \\ 0, n \neq k \end{cases}$ )	$S_n = 0$ , for each $n \neq 0$ $s(n)=1$ , for $n=0$	<b>סיגנל וי (ההלם)</b>
באופן דומה.	$S_{n,m} = 0$ , for each $n \neq m$ $S_{n,m} = 1$ , for $n=m$	<b>SUI[n,m]</b>
$s(t)$ . exists $T > 0$ . To any t, $s(t+T) = s(t)$	$S_n$ . Exists $N > 0$ . To any n, $S_{n+N} = S_n$ הנ הקטן ביותר כך ש... ז"א, גודל מינימלי של מחזור.	<b>מחזורי</b>

<b>דטרמיניסטי</b>	<b>סטוכסטי</b>	
אם הוא ניתן לניבוי בכל זמן בעתיד, כאשר נותנים את הT	שאינו דטרמיניסטי. לא אומר שלא יודעים לקרב אליו	<b>הגדרה</b>
סינוס	רעש לבן- הכי סטוכסטי, לא ניתן להגיד עליו כלום	<b>דוג'</b>
יכול להיות מחזורי/ לא	לעולם לא יכול להיות מחזורי	<b>מחזוריות</b>

## תכונות של סיגנלים:

לכל סיגנל יש את היפוכו:  $x=-s$

יש את סיגנל ה-0

ניתן לחבר בין סיגנלים:  $x+y = x_n+y_n = x(t)+y(t)$

משמע, **אוסף הסיגנלים מהווים מרחב וקטורי** (קיום 0, קיום הופכי, חיבור אפשרי).

זה חשוב משום שלכל מרחב וקטורי יש בסיס, לפחות 1. נלמד 2 בסיסים מאוד חשובים לאוסף כל הסיגנלים.

## בסיס למרחב וקטורי:

אוסף של איברים שפורשים את המרחב, הוא ב"ת לינארית/ יחיד

מימד- מס' האלמנטים בבסיס.

בסיס פורש: SUI

$$s = \sum_n S_n SUI[m]$$

ז"א ש  $S_n$  אלה המקדמים של הוקטור n-יה אינסופית לייצוג

סיגנל הוא יותר מסתם סדרה, הוא אובייקט,  $S_n$  הוא ייצוגו עפ"נ מסלול הזמן.

**כלל:** ככל שהמחזור גדל, התדר קטן, ולהפך

**טבלת האופרטורים על סיגנלים**

שם ומאפיינים	אופרטור
הופכי	-
חיבור	+
אפס	0
מקדם בזמן ניתן לחשב רק עבור סיגנל דטרמיניסטי	$\hat{z}X_n = X_n + 1$
משהה- מה הוא היה בזמן הקודם. אין דומה לו בעולם האנלוגי. לא צריך לחשב, אלא רק לשמור בזיכרון.	$\hat{z}^{-1}X_n = X_n - 1$
ההפרש הסופי הראשון יש לו תכונות כמו לנגזרות (אמנם כלל השרשרת לא בדיוק)	$(\Delta x)_n = X_n - X_n - 1$

**ניתן לכתוב:**

$$SUI^{[m]} = \hat{z}^{-m}UI$$

**סינוסים מיוחדים:**

$S(t) = A \sin(\omega t)$ , לכל  $t$ , זה מחזורי

$S_n = a \sin(kn)$ , זה מחזורי רק עבור  $k$ -ים שהם מספר רציונלי \*  $\square$

**סיגנלים מחזוריים שאינם סינוסים:**

ספרתי	דיגיטלי	
		מרובע- צריך לעגל פינות כדי שיהיה באמת סיגנל, אחרת רוחב פס אינסופי
		משולש
		שיני מסור

**טור פורייה:**

$$\omega_0 = \frac{2\pi}{T}$$

$$s(t) = \sum_{k=-\infty}^{\infty} (S_k e^{ik\omega_0 t})$$

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right)$$

סיגנל מחזורי, יהיה לו מס' ב"מ של "מקלות"

## טרנספורם פוריה:

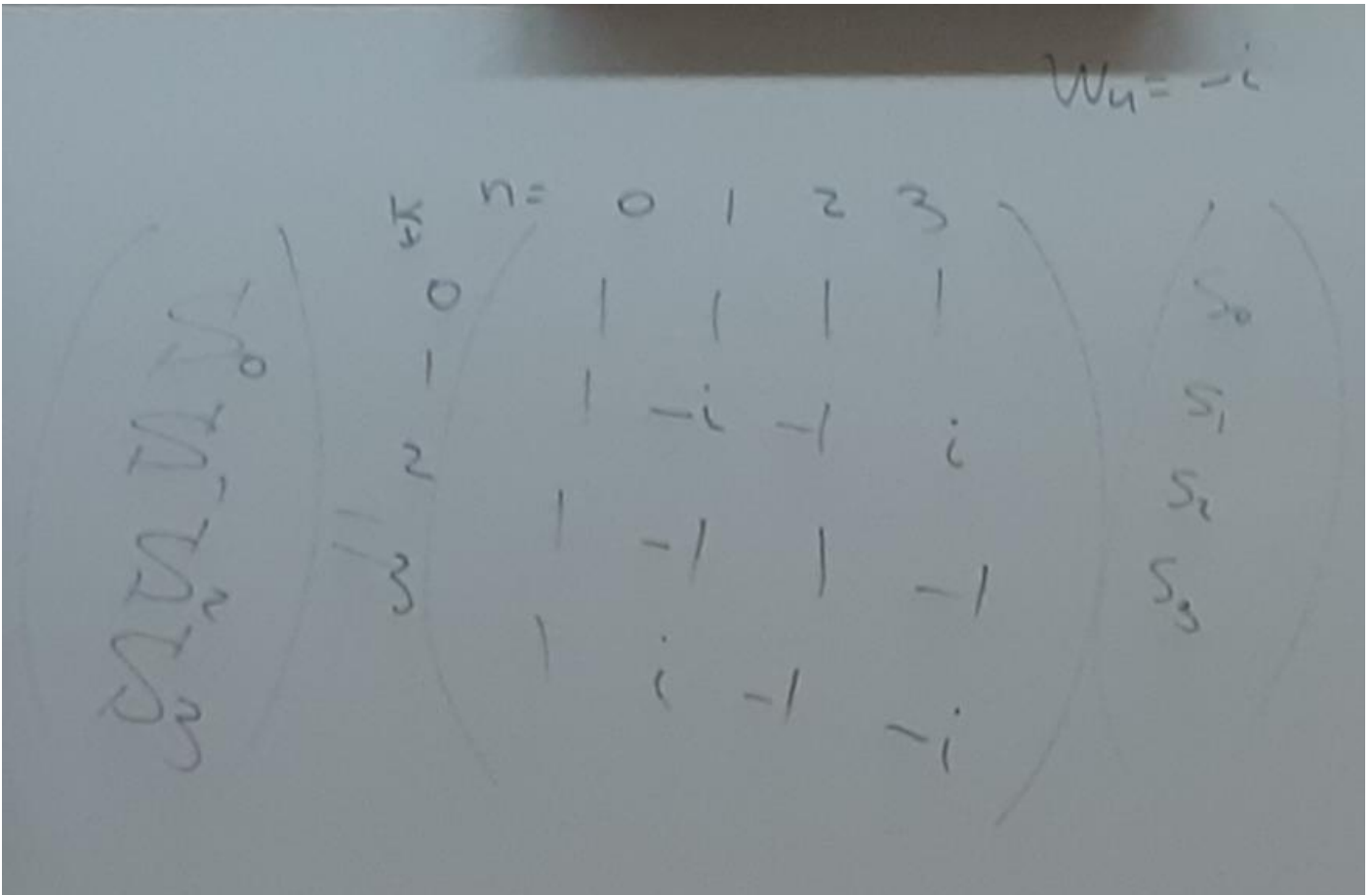
לוקח פונ'  $s(t)$  ומחזיר פונ'  $S(w)$

יש גם את טרנספורם פוריה דיסקרטי שלוקח  $S_n$  ומחזיר  $S_k$  (?) ← עובר בין הסיגנל מציר הזמן לספקטרום

$$\omega_N^N = \left( e^{-i \cdot \left( \frac{2\pi}{N} \right)} \right)^N = 1$$

$$\omega_N = e^{-i \cdot \left( \frac{2\pi}{N} \right)}$$

$$S_k = \sum_{n=0}^{N-1} S_n \omega_N^{nk}$$



ממחיש איך עוברים בין הסדרה על ציר הזמן לבין הסדרה על הספקטרום.

לוקחים את  $WN$ , ומעלים אותו בחזקה של  $n \cdot k$  לפי הטור והשורה, כאשר  $N = n$  בסיס

**משפט הדגימה הבסיסי:** אינטואיטיבית, אם אני דוגמת מספיק טוב (3 נק' של כל מינ' / מקס' מקומיים

אם אני לוקחת סיגנל אנלוגי ויש לו ספקטרום מסוים, משום שהסיגנל אמיתי, יש לו תדר גבוה ביותר.

אם דוגמים אות אנלוגי  $s(t)$  בקצב יותר מפי 2 מהתדר הגבוה ביותר בספקטרום  $S(t)$ , אזי אין איבוד מידע.

בעזרת הנוסחא או בדרכים יעילות יותר,  $s(t) = \sum_n S_n \text{sinc}(at)$ , אחרי שאני מסיימת את העיבודים שלי, אני רוצה לקחת את הסיגנל הדיגיטלי ולהחזיר לאנלוגי,

$$\text{Sinc}(t) = \frac{\sin(t)}{t}$$

**רוחב סרט-** התדר הגבוה ביותר על ציר הספקטרום- התדר הנמוך ביותר על ציר הספקטרום. אמנם הם חייבים להיות סופיים, כי אחרת, רוחב הסרט הוא אינסופי.

לא כתבתי. מת לי המחשב, הקלטתי, לשמוע ולכתוב או ללמוד מהספר. בר השלים קצת, אבל היו דברים שלא הבין, לכן הרווחים.

**משפט הדגימה של נייקוויסט -** אם דוגמים אות אנלוגי  $s(t)$  בקצב של יותר מפי 2 מהקצב הגבוהה בספקטרום, אזי אין איבוד מידע.

כאשר דגמנו פחות מקצב של פי 2 מהקצב הגבוהה ביותר, אז יש לפחות שני גלי סינוס מתאימים.

### משפט הילברט:

אם הסיגנל מקיים את שני התנאים הבאים:

1. מוגבל בסרט ( כלומר בעל תדר מקסימלי)

2. נטול DC - אין רכיב DC - כלומר ממוצע סיגנל 0.

אז ניתן לכתוב אותו באופן הבא:  $X(t) = A(t) \cdot \cos\phi(t)$

כאשר  $A(t)$  הוא האמפליטודה תלויה הזמן ו  $\cos\phi(t)$  הוא הפאזה תלויה הזמן  $\omega$ .

**אפנון =** שינוי תדר סיגנל. רצוי שהסיגנל יישא אינפורמציה

Amplitude modulation – כפתור AM ברדיו

Frequency Modulation כפתור FM ברדיו

בכיוון ההפוך  $\leq$  טרנספורם הילברט לוקח סיגנל אלוגי ואופרטור, ונותן סיגנל אנלוגי אחר.

$$y(t) = HX(t) = A(t)\sin\phi(t)$$

תרגום אמפליטודה:  $A(t) = \sqrt{x^2(t) + y^2(t)}$

תדר:  $\phi(t) = \tan^{-1} \frac{y(t)}{x(t)}$

אם  $\phi(t) = \omega(t)$  אז המעבר דרך גזירה, אבל עבור: (לא ברור הכתב כאן)

### עקרון האי הוודאות

כמה מדוייק אפשר למדוד תדר של סינוס (למשל) בהגבלת זמן של שנייה (למשל)

$$\Delta\omega\Delta t \geq 2\pi$$

ככל שרואים יותר זמן  $\leq$  אי דיוק הולך וקטן



## טרנספורם Z :

הקדמה – פונקציות יוצרות. בבדידה ראינו שבעזרת פונקציות יוצרות ניתן למצוא פתרון של סדרות. טרנספורם Z דומה במובן הזה שהוא לוקח עושה מעבר מפונקציה (רציף) לסדרה (בדיד), וההופכי שלו עושה את המעבר ההפוך.

צריך לקחת סדרה ולהפוך פונקציה מעל מרוכבים: לוקחת קבוצת דיגיטליים והופך לאנלוגי.

מערכת לעיבוד אותות

אינפוט – יכול להיות 0

אוטפוט – לפחות אחד

**מערכת סיבתית, קוזלית-** אם  $y(t)$  הוא פונקציה של  $x(t)$  וב- $t$  קודמים אבל לא בזמנים עתידיים,  $t+n$ , עבור  $n$  חיובי.

**מערכת לינארית – (לא  $y=ax+b$  אלא: )**

אם:  $x^{[1]} \rightarrow y^{[1]}$  ;  $x^{[2]} \rightarrow y^{[2]}$  אז:

$x^{[1]} + x^{[2]} \rightarrow y^{[1]} + y^{[2]}$  וגם:

$ax \rightarrow ay$

ובשורה אחת:  $ax^{[1]} + bx^{[2]} \rightarrow ay^{[1]} + by^{[2]}$

מערכת אינווריאנטית כלפי הזמן:

אותו אינפוט בזמן אחר ייתן אותו פלט. קלט זהה בזמנים שונים ייתן את אותו הפלט.

$$x \rightarrow y \Rightarrow \hat{z}^m x \rightarrow \hat{z}^m y$$

כלומר אין במערכת שעון.

**מערכת לינארית ואינווריאנטית כלפי הזמן = מסנן!**

### תרגיל 6.1.1

האם הדברים הבאים הם מערכות לעיבוד אותות?

1. האינפוט  $X$ , אאוטפוט  $Y=X$ 
  - כן, כי אף אחד לא אמר שהיא חייבת להוציא משהו.
2. האינפוט  $X$ , האאוטפוט  $Y=K$ , כאשר  $K$  ב"ת בא
  - כן. הסיגנל הוא DC. אנחנו יודעים שהוא לא סיגנל בכלל כי האנרגיה שלו אינסופית. אבל לכן נסתכל על חלון שלו ובכל שאר הזמן נניח שבעבר ובעתיד הוא היה 0. לכן זו מערכת לעיבוד אותות.
3. אאוטפוט  $Y = +, - \sqrt{X}$ 
  - לא, כי זה לא יכול להיות גם שלילי וגם חיובי, וגם כי זה מרוכב אם נותנים מס' שלילי
4. אין אינפוט, אאוטפוט הוא  $y = \sin(1/t)$ 
  - לא, כי רוחב הסרט של זה הוא אינסופי
5. טלויזיה- קולטת מידע בגלי רדיו ומוציאה וידאו וקול.
  - כן, קול- ידוע. וידאו הוא גם אות, אפילו רצף של אותות, כאשר יש אות של ירידת שורה, אז כל אות הוא נק' צבע על המסך, פרט לאלה.
6. אינפוט- אות דיגיטלי, אאוטפוט- אות אנלוגי
  - כן, אף 1 לא הגביל את האות להיות מסוג כזה או כזה

### משפט המסנן בציר התדר

תגובה לתדר

$$Y(\omega) = H(\omega)X(\omega)$$

אם לא יהיה באינפוט תדר מסויים, אז הוא בוודאות לא יופיע בפלט. כי אם  $X(\omega)$  בתדר מסויים אז  $Y(\omega)$  הוא 0 בתדר מסויים.

פונ' התמסורת

$$Y(z) = H(z)X(z)$$

מופיעים תדרים בפלט שלא היו בקלט, משמע, לא אינווריאנטי לזמן.

טכניקות מיצוע

שינוי האות	מיצוע
קבוע	מיצוע
משתנה מאוד לאט	נדגום על קטע זמן גדול ונמצע
משתנה מהיר	MA- MOVING AVERAGE את דוגמיות הזמן שלנו והמיצוע נזיז ב1
משתנה מאוד מהר	

קונבולוציה

$$y_n = \sum_{k=0}^{N-1} a_k x_{n-k}$$

גם לינארי וגם אינווריאנטי כלפי הזמן, לכן הוא מסנן

למדנו את זה גם בכיתה ב' וגם במכפלת פולינומים, משום שבשניהם התוצאה ( כל ספרה בה / כל מקדם  $X$  בחזקה כלשהי), הוא סכום של זוגות מספרים, שהאינדקס של הראשון עולה לאורך הסכום, והאינדקס של השני יורד (או להפך).

לא כל מסנן הוא קונבולוציה, כי יש, למשל, מסננים רקורסיביים, והם מהגדרה, לא קונבולוציה.

דוגמא לולאה סופית (*floop*) - קונבולוציה

*bloop – bounded loop*

*general loop- gloop* – דוגמא לזה היא רקורסיה

## מסננים רקורסיביים

לא הספקתי לכתוב

סיבתי- אין בנוסחא 1 שמתקדם בזמן.

התגובה להלם- מהנדסים קוראים כך לסיגנל ה/ $U$ , שהוא אינו 0 רק ב $N=0$ , כי זה כמו שהולמים בפטיש על המערכת ומחכים לראות מה יוצא, וחוצמזה לא עושים כלום.

משמע, התגובה להלם בקונבולוציה זה המקדמים בכיוון ההפוך, לכן אנו רוצים שאינדקס 1 יעלה והשני ירד (בנוסחה).

יש מסננים שתגובת ההלם שלהם תהיה לא 0, רק למס' סופי של זמנים, ויש אחרים, שתגובת ההלם שלהם תהיה לא 0, לאינסוף זמנים

## סוגי מסננים :

HP high pass – מעביר רק את התדרים הגבוהים

LP Low pass – מעביר רק את התדרים הנמוכים

BP – band pass - מוריד נמוכים וגבוהים ומעביר את האמצע

BS – band stop הפוך לBP

BS חד שמעביר כמעט הכל חוץ מסביבה קטנה של תדר מסויים נקרא *NOTCH*

**מסנן** - מערכת לעיבוד אותות עם אינפוט 1 שהוא סיגנל ספרתי, ופלט שגם הוא סיגנל ספרתי.

סוג מסנן	נוסחא	הסבר
מסנן FIR, MA	$y_n = \sum_{l=0}^{L-1} Q_l X_{n-l}$	בזמן n, (מסנן סיבתי). m מתחיל ב-1, כי כאשר $\gamma$ הוא 0, אנו מסתפקים באיבר הראשון. מסנן מסוג זה נראה קונב', סכום של מכפלות, כאשר אינדקס של כפולה 1 עולה, ושל השני יורד
מסנן IIR, AR	$y_n = x_n + \sum_{m=1}^{M-1} b_m y_{n-m}$	שם ענתיקה לרקורסיה, גם היא קונב'
מסנן IIR, ARMA	$y_n = \sum_{l=0}^{L-1} a_l X_{n-l} + \sum_{m=1}^{M-1} b_m y_{n-m}$	המסנן הכי כללי, שמכיל בתוכו את 2 המסננים העליונים, כי אם כל ה-B-ים =0, חזרנו ל-MA, ואם כל ה-A-ים =0, חזרנו ל-AR, הוא בעצם 2 קונב'

**FIR** - להכניס למסנן אימפולס שהוא 1 בזמן 0, ומקבלים בו פלט שהוא סיגנל. אם לסיגנל יש מס' סופי של זמנים שהוא לא 0, ובכל היתר הוא 0

התגובה להלם שלו אינה 0 בזמן סופי

**IIR** - אם הוא ממשיך עד אינסוף להיות לא 0, כאשר הכנסנו אליו את סיגנל ההלם.

### הבעיה הקשה לזיהוי מערכת:

לא ניתן לבחור איזה X רוצים, אלא רואים איזה נכנס, ואיזה סיגנל יוצא, וככה מחליטים מה עושה הקופסא.

יש מקרים בהם לא ניתן לפתור את הבעיה.

דוג' כללית	1	2	3
נתונים	מסנן FIR $\leftarrow$ MA, בעל 3 מקדמים (הוא L), הכל 0, ופתאום המסנן מתחיל לעבוד. בנוסף, $X_0 \neq 0$	מסנן MA, 3 מקדמים בלבד. האינפוט מהתחלה אינו 0, אלא מיד רואים ערכים. X יכול להיות 0. אי אפשר לדעת.	מסנן AR, 3 מקדמים.
מהו Y לכל X?	$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2}; x_0 \neq 0$	$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2}$	$y_n = x_n + b_1 y_{n-1} + b_2 y_{n-2} + b_3 y_{n-3}$
הסבר	אמנם אנחנו צריכים 3 משוואות כדי לפתור את הבעיה, אך אנחנו יכולים להשתמש בזמן 0 ובנוסחה כדי ליצור עוד משוואות כל סיגנל X שקיבלנו בעבר/ בהווה, אנחנו יודעים. כל מקדם שחישבנו בעבר, אנחנו גם יודעים, לכן יש לנו פתרון לבעיה	כעת כבר לא ניתן להשתמש בזמן 0, משום שאנו רואים מיד ערכים. לכן במקום זה, נשתמש ב-3 משוואות אחרות ב-3 נעלמים, כאשר עדיין מתבססים על הנוסחה המקורית	
התהליך המתמטי	$y_0 = a_0 x_0$ $a_0 = \frac{y_0}{x_0} \leftarrow$ $y_1 = a_0 x_1 + a_1 x_0$ $a_1 = \frac{y_1 - a_0 x_1}{x_0} \leftarrow$ $y_2 = a_0 x_2 + a_1 x_1 + a_2 x_0$ $a_2 = \frac{y_2 - a_2 x_2 - a_1 x_1}{x_0} \leftarrow$	$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2}$ $y_{n+1} = a_0 x_{n+1} + a_1 x_n + a_2 x_{n-1}$ $y_{n+2} = a_0 x_{n+2} + a_1 x_{n+1} + a_2 x_n$	$y_n = x_n + b_1 y_{n-1} + b_2 y_{n-2} + b_3 y_{n-3}$ $y_{n+1} = x_{n+1} + b_1 y_n + b_2 y_{n-1} + b_3 y_{n-2}$ $y_{n+2} = x_{n+2} + b_1 y_{n+1} + b_2 y_n + b_3 y_{n-1}$

<p>מסתכלים במשך 5 זמנים, ב2 הראשונים רק על הפלטים, ב3 האחרונים, גם על הפלט וגם על הקלט, כדי שנוכל להכניס את המידע הנכון לכל איבר. חייבים למלא את הזכרון של המערכת 2L-1 פעמים, כי גם למערכת יש זכרון, אמנם רק L-1, אבל כדי לדעת את כל המידע, צריך לדעת מה היה קודם אמנם במקרה 1, מספיק להסתכל רק על 3 זמנים, כי 2 הראשונים מלאים 0-ים.</p>	<p>הסבר מילולי</p>	
$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_n \\ x_{n+1} \\ x_{n+2} \end{pmatrix} + \begin{pmatrix} y_{n-1} & y_{n-2} & y_{n-3} \\ y_n & y_{n-1} & y_{n-2} \\ y_{n+1} & y_n & y_{n-1} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$	$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_n & x_{n-1} & x_{n-2} \\ x_{n+1} & x_n & x_{n-1} \\ x_{n+2} & x_{n+1} & x_n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$	$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_0 & 0 & 0 \\ x_1 & x_0 & 0 \\ x_2 & x_1 & x_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$ <p>טבלת מקדמים</p>
<p>סוכמים את הוקטור של ה-x-ים עם המכפלה בין המטריצה והוקטור, כאשר מכניסים את ה-y-ים למטריצה, וה-B-ים (המקדמים) בוקטור. כל Y שמוכפל ע"י ה-B המתאים, נכנס לפי סדר המשוואות גם היא בצורת טפליץ</p>	<p>המטריצה כעת אינה משולשת, לכן לא נוכל לעשות את הטריק של הפתרון הפשוט, אמנם בעלת צורת טפליץ! <math display="block">a = \overline{x^{-1}y} \leftarrow \bar{y} = \overline{\bar{x}a}</math> 2 קווים למעלה יהיו מטריצה, קו 1 למעלה יהיה וקטור, המרצה מסמן אותם למטה. כאשר מטריצה היא בעלת צורה זו, ניתן להפוך אותה בסיבוכיות של N^2 (אלגוריתם לוינסון ודורבין) אמנם להפוך מטריצה לפעמים יכול להיות מסוכן, כאשר המטריצה סינגולרית ()</p>	<p>אנו מחפשים את המקדמים, לכן המקדמים ימצאו בוקטור (הימני), ולא להפך כפי שאנחנו רגילים (כי הרי ה-A-ים הם מקדמים) (משולשת תחתונה) ייתן בדיוק את הנוסחאות שאנו צריכים. זוהי גם מטריצת טפליץ (TOEPLITZ) מתמטיקאי גרמני יהודי שהמציא את האלגברה הליניארית, שעלה ארצה ומת באופן טבעי ולא בשואה, אחרי שהרצה בעברית.)</p> <p>הסבר טבלה</p>
<p>משום שיש רעש, (X ו Y קצת מורעשים), ה-A-ים לא יוצאים לי הכי מדויקים ומתאימים לקופסא. לא ניתן לעשות ממוצע נע, משום שזה מוטה סטטיסטית וגם משום שהסיבוכיות נוראית.</p>		<p>השפעת רעש ותיקונו</p>

### תכונת טפליץ- לאורך כל אלכסון יש אותם ערכים

מסנן ARMA קשה לפתור, משום שהמטריצה ענקית וגם לא טפליץ. לכן גם בספרות בקושי רואים אותו.

### נוס' וינר- הופף- כל הנוסחאות שמופיעות בעמודה השמאלית של הטבלה + 2 האחרונות כאן (כל היתר, זה העתק)

$$y_n = x_n + b_1 y_{n-1} + b_2 y_{n-2} + b_3 y_{n-3}$$

$$y_{n+1} = x_{n+1} + b_1 y_n + b_2 y_{n-1} + b_3 y_{n-2}$$

$$y_{n+2} = x_{n+2} + b_1 y_{n+1} + b_2 y_n + b_3 y_{n-1}$$

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_n \\ x_{n+1} \\ x_{n+2} \end{pmatrix} + \begin{pmatrix} y_{n-1} & y_{n-2} & y_{n-3} \\ y_n & y_{n-1} & y_{n-2} \\ y_{n+1} & y_n & y_{n-1} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$\bar{y} = \bar{x} + \bar{y} \cdot \bar{b} \rightarrow \bar{b} = \overline{\bar{y}^{-1}} \cdot (\bar{y} - \bar{x})$$

### פתירת הבעיה הקשה בציר התדר- נוס' יול- ווקר

$$Y(w) = H(w)X(w)$$

$$Y_k = H_k X_k$$

$$H(w) = \frac{Y(w)}{X(w)}$$

הבעיה עם המשוואה היא, מה קורה כשהמכנה הוא 0?

$$y_n = x_n + \frac{1}{2} y_{n-1} \text{ עבור}$$

3	2	1	
תדר נייקוויסט	תדר DC (XN=1)	$h_0 = 1$ $h_1 = \frac{1}{2}$ $h_2 = \frac{1}{4}$	התדר
$H = 1 + \frac{1}{2}(-H) \rightarrow H = \frac{2}{3}$ $y_n = x_n + \frac{1}{2}(x_{n-1} + \frac{1}{2}y_{n-2})$ ניתן להכניס את $y_{n-2}$ , בחזרה לתוך המשוואה $y_n = \sum_{k=-\infty}^n h_k x_{n-k}$	$H = 1 + \frac{1}{2}H$ $H = 2$ ←		?=H
	התדרים שיהיו Y יהיו זהים, משום שזה מסנן	IIR, גם אם הוא דועך, הוא לא זהותית 0.	הסבר

מצאנו דרך חדשה לכתוב את נוס' ARMA:

$$y_n = \sum_{k=-\infty}^n h_k x_{n-k}$$

אמנם הדרך הזו לא באמת שימושית, כי צריך לקחת את כל הערכים מ  $-\infty$ , וכמובן שזה בלתי אפשרי הדרך המקורית, עבור Z טרנספורם:

$$Y(z) = \sum_{n=-\infty}^{\infty} y_n z^{-n} = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h_k x_{n-k} z^{-n} = \dots = \left( \sum_k h_k z^{-k} \right) \left( \sum_k h_k z^{-n} \right)$$

הראשון בסוגריים הוא הZ טרנספורם של h, לכן נקרא לו H(z)

השני הוא כנ"ל של x, לכן נקרא לו X(z)

האימפולס רספונס h והFREQUENCY RESPONSE H: הם טרנספורמי פורייה 1 של השני:

$$Y(w) = H(w)X(w) \quad y_n = \sum_{k=-\infty}^n h_k x_{n-k}$$

לא רק שהוכחנו את מה שטענו עד כה, אלא גם מצאנו מה הם h, H

$$y_n = \sum_{l=0}^{L-1} a_l x_{n-l} + \sum_{m=1}^{M-1} b_m y_{n-m} \text{ -ARMA}$$

$$\sum_{l=0} \alpha_l x_{n-l} = \sum_{m=0} \beta_m y_{n-m}$$

פשוט לקח את המשוואה, העביר אגפים וכתב הפוך, זה נקרא המשוואה הסימטרית (?)

אמנם שימושי רק כאשר יש לי את כל ה-x-ים ואת כל ה-y-ים ← לא מחולל.

אז מה זה נותן?

עוד 4-5 שורות של נוסחאות שצילם, כי כתב מהר מדי והיה מורכב.

כי חילק ב, אבל ידוע מה זה  $Y/X$ , משמע  $X(z)$  ו  $B(z)$  הם פולינומים של  $Z \leftarrow H(z)$  היא פונ' רציונלית. כרגע עוזבים את עניין ה0.

השורשים של  $A(z)$  מאפסים את  $H$ , נקראים ה-0ים של  $H(z)$ , בעוד שהשורשים של  $B(z)$  מפוצצים את  $H$  (במרוכבים יש לכל פולינום את אותו מס' שורשים כמו מעלת הפולינום).

פונ' רציונלית (מנת 2 פולינומים) נקבעת חד ערכית- עד כדי הכפלה בקבוע, ע"י השורשים המאפסים והקטבים שלה,  $\leftarrow$  השורשים והקטבים של פונ' התמסורת קובעים את הפונ' חד ערכית עד כדי GAIN 1 כללי.

אם למסנן יש אפסים אך אין קטבים, זה מסנן MA

אם יש לו קטבים אך לא אפסים, זה מסנן AR

מסנני FIR, MA, הם תמיד ALL-ZEROS בפונ' התמסורת (אין קטבים).

מסנני IIR, AR, הם תמיד ALL-POLES בפונ' התמסורת (אין אפסים).

אמנם ARMA יש גם וגם.

<http://dspcsp.com/tau/filtconv.pdf> איך לעבור מכל צורה לכל צורה:

to $\rightarrow$	a, b coefficients	$\alpha, \beta$ coefficients	impulse response	frequency response	transfer function	gain and pole-zero diagram
from $\downarrow$						
a, b coefficients	identity	subtraction of y terms	MA: $h=a$ AR + ARMA: recursion	substitute $x=e^{ikn}$	write using $z^{-1}$ and extract	through transfer function
$\alpha, \beta$ coefficients	addition of y terms	identity	same as a,b	same as a,b	same as a,b	same as a,b
impulse response	MA: $a=h$ ARMA: recursion	through a,b	identity	DFT	$zT$	through transfer function
frequency response	through IR or transfer function	same as a,b	iDFT	identity	analytic continuation	through transfer function
transfer function	through $\alpha, \beta$	$B(z) Y(z) = A(z) X(z)$	$izT$	substitute $z = e^{i\omega}$	identity	find roots
gain and pole-zero diagram	through transfer function	through transfer function	through transfer function	substitution	multiply terms to get polynomial	identity

מעבר לפונ' התמסורת:

מייד אחרי המשפט, להוסיף לכל ספרה את הנוסחא המתאימה.

1. לא רוצים לראות ח-ים, לכן במקום זה עוברים ל-Z-ים
2. מעבירים אגפים כך ש  $X$  יהיה בצד 1, ו  $Y$  בצד שני.
3. מעבירים את  $X$  ואת  $Y$  להיות טרנספורמי  $Z$  שלהם
4. מחליפים את  $Y(z)/X(z)$  להיות  $H(z)$ , ומפשטים את המנה בצד ימין.
5. במקום בתוך המעגל שמקבלים 0, מסמנים את הנק' בנק'.
6. במקום בתוך המעגל שהפונ' מתפוצצת, מסמנים בא

7. תזכורת: זה (1,0) DC, זה (-1, 0) זה נייקוויסט (כי כל פעם מעלים בחזקה כלשהי את ערך ה-x, לפי הזמן שעבר)

דוג':

$$y_n = x_n + \frac{1}{2}y_{n-1}$$

$$y_n = x_n + \frac{1}{2}x_{n-1} + \frac{1}{2}y_{n-1}$$

$$y = x + \frac{1}{2} \cdot \hat{z}^{-1} \cdot x + \frac{1}{2}\hat{z}^{-1}y$$

$$y - \frac{1}{2}\hat{z}^{-1}y = x + \frac{1}{2}\hat{z}^{-1}x$$

$$\left(1 - \frac{1}{2}\hat{z}^{-1}\right)y = \left(1 + \frac{1}{2}\hat{z}^{-1}\right)x$$

$$\left(1 - \frac{1}{2}\hat{z}^{-1}\right)y(z) = \left(1 + \frac{1}{2}\hat{z}^{-1}\right)x(z)$$

$$H(z) = \frac{z\left(1 + \frac{1}{2}z^{-1}\right)}{z\left(1 - \frac{1}{2}z^{-1}\right)} = \frac{z + \frac{1}{2}}{z - \frac{1}{2}}$$

$$y = x + \frac{1}{2}\hat{z}^{-1}y$$

---

### אלגוריתמים

---

תכנות = אלגוריתמים + מבנ"ת

תרשימי זרימה של אלגוריתמים, מציגים רק את האלגוריתם, ושום דבר על מבנ"ת. פרט לכך שקשה מאוד, כאשר מעדכנים את הקוד, לעדכן גם את תרשימי הזרימה.

רק ב-2 מקצועות בתוכנה משתמשים בתרשימי זרימה: מיקרו-קונטרולרים, DSP

אמנם הגרפים ב-DSP מציגים גם את האלג' וגם את המבנ"ת, והם באים עם טכנולוגיה (טרנספורמציות) שנותנים אפשרות לשפר את הקוד שלנו- לייעל אותו כך שירוך יותר מהר/ שיצרוך פחות זכרון.

לכן נשתמש בתורת הגרפים ב-DSP, ונקבל המון תובנות באמצעות השימוש.

הגרפים שלנו:

1. יש כיוון לקשתות
2. נק' היא נק' והיא מייצגת אות a. האות שליד הנק' זה שם האות הדיגיטלי.
3. קשת היא קשת.



**טופולוגיה**

בעיית הגשרים של קניגסברג ופתרונה המציאה גם את תורת הטופולוגיה.

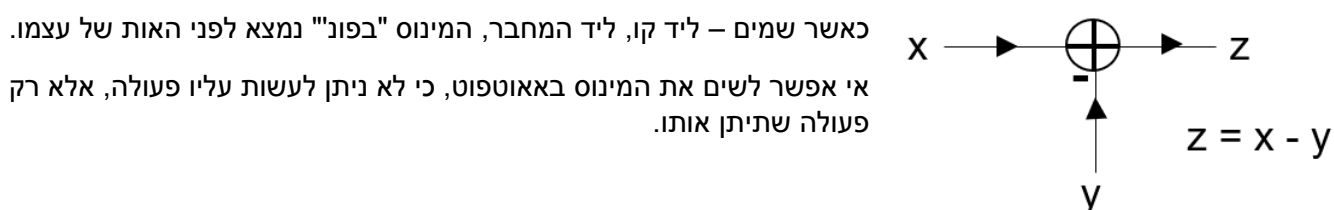
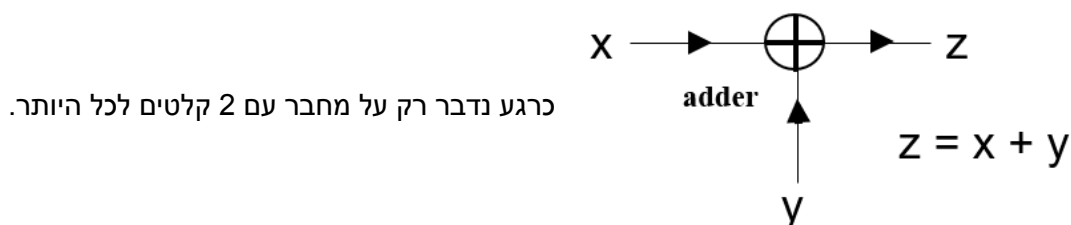
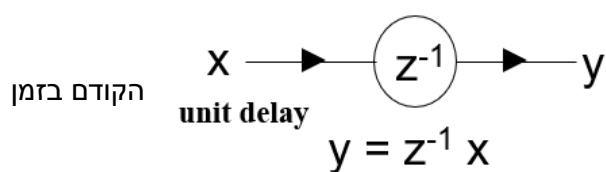
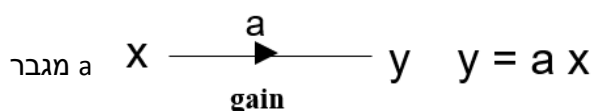
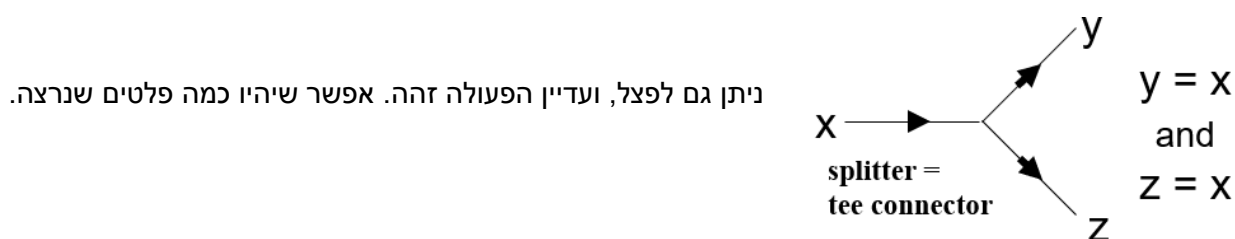
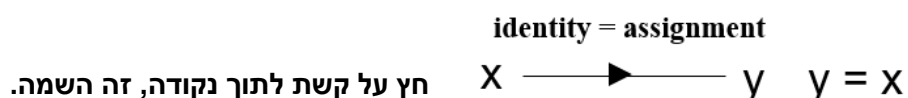
**גיאומטריה במישור:** מה שנובע מ-4 אקסיומות של אוקלידס. חפיפת משולשים- סיבוב המשולש והזזתו כך ש-1 יושב על השני.

**גיאומטריה פרויקטיבית:** ניתן להגדיל ולהקטין לכל המרחב  $\leftarrow$  מזה יצא דימיון משולשים

**גיאומטריה אפינית:** ניתן להגדיל ולהקטין לכל המרחב וגם רק לאורך ציר 1.

**טופולוגיה:** אין משמעות למרחק, אוריינטציה, כיוונים, אלא רק מי מחובר למי. ז"א, אם יש לי צורה מגומי שניתן להפוך אותה לצורה אחרת, בלי לחתוך או להדביק, אז כל הצורות שניתן לשנות אותן אליהן, הן טופולוגיות = אומו-הומורפיות (הרבה הומומורפיזם).

משמעות של גרף טופולוגית בלבד: צורות ואורך הקשתות והנקודות לא משנות, וגם לא הסדר שלהן.

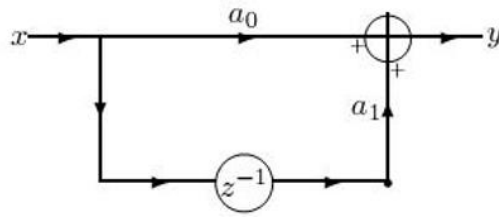


**טריק:**

להציב  $x_n$  במקום X, בכל נק', כדי להבין מה עושה המערכת.

זה קונב'- הכי פשוטה.

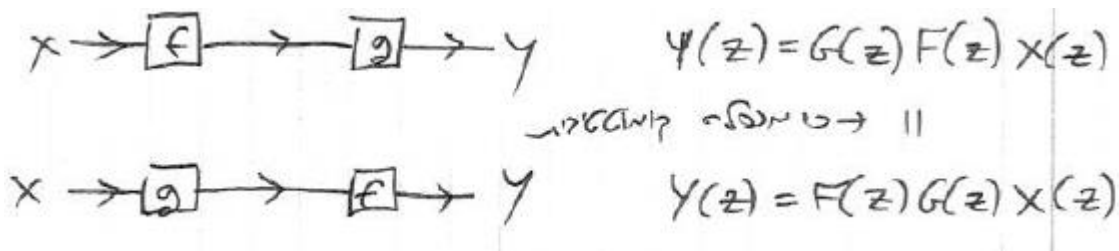
אמנם החלפה במיקום של ההכפלה בקבוע לבין ההשגיה, למרות שלא שומרת על הטופולוגיה, לא משנה את הפלט, כי אלה מסננים, וכל 2 מסננים מתחלפים.



$$y_n = a_0 x_n + a_1 x_{n-1}$$

**דוג':**

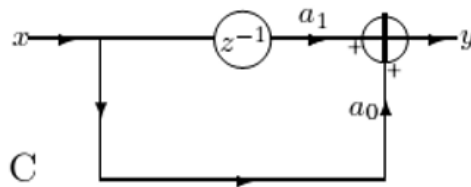
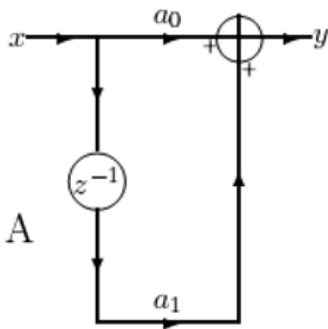
לא לינארית: פעולה ראשונה, הגברה בא, פעולה שנייה- העלאה בריבוע, אם מחליפים ביניהם, משנים את תוצאת הפלט, במקום  $(ax)^2$  ל  $a^*(x^2)$



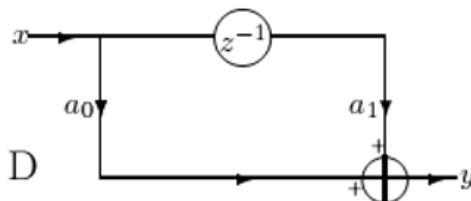
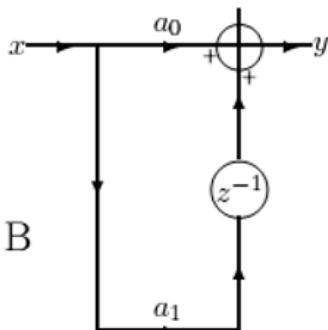
**לינארית:**

נוס' 1-3, עבור כל הצורות שמסתכמות ← האותות בציר הזמן, לאלה.

**הכלל בDSP:** מה שקשה בציר הזמן, עושים בציר התדר או בטרנספורם Z.



$$y_n = a_0 x_n + a_1 x_{n-1}$$



מעבר בין A לC, הוא טופולוגי

בין A לB, החלפה של מסננים.

בקרוב, כיוון החצים ישנה, כנ"ל המיקומים של האותות והמסננים.

4 הסוגים האלו חשובים, כדי שנוכל לדעת איך לבנות (עם איזה חלקים) את האותות שלנו, כי כמו לגו, לא כל חלק מתחבר עם כל 1 אחר, בצורה מושלמת.

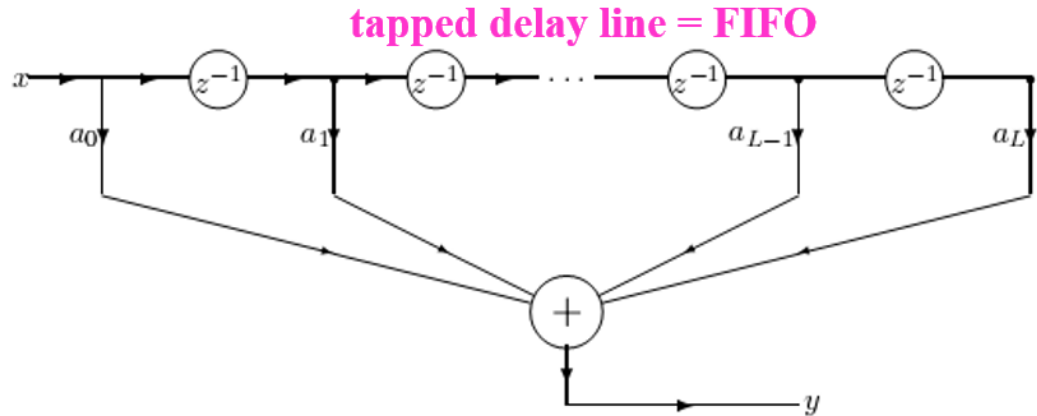
$$y_n = \sum_{l=0}^L a_l x_{n-l}$$

we would like to build

-TAPPED DELAY LINE  
TAP זה לצתת. לתפוס  
סיגנל ומוציאים אותו.  
דגימה. נקרא כך ע"י  
המהנדסים, קוראים לו  
גם SHIFT REGISTER.

FIFO ← ז"א תור (לא  
מחסנית, כי היא LIFO)

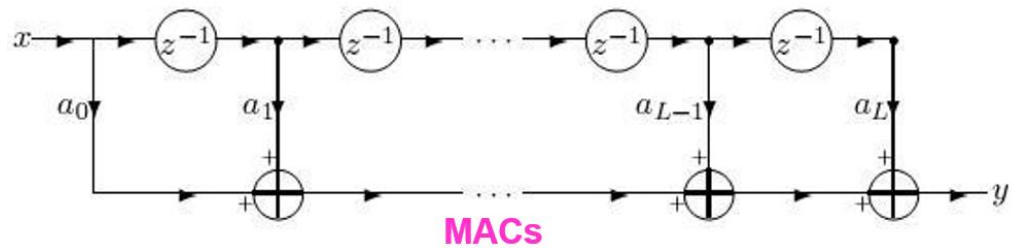
כל מקום שבו יש נק',  
צריך זיכרון, כדי לזכור  
כל ערך של X לאורך  
הזמן, כי אחרת, לא  
ניתן לעשות עליו  
פעולות, ובפרט, לחבר  
אותו בסוף.



tapped delay line = FIFO

but we only have 2-input adders !

אמנם המחבר לא יכול לחבר יותר מ2 אותות, לכן זה בעצם "שרשרת" של הרבה מחברים:



MAC= Multiply and  
Accumulate ← קודם כופלים  
ואח"כ מחברים/ צוברים.

### AR- טריק לקריאה:

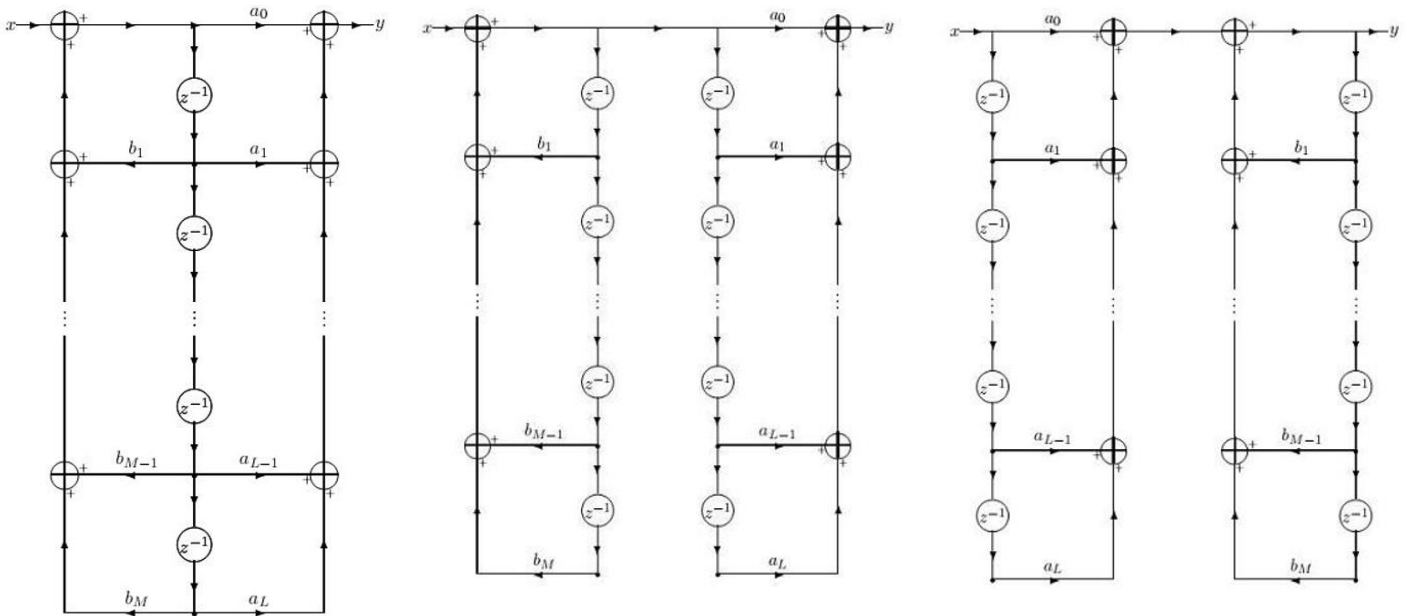
קוראים מהסוף להתחלה, כאשר אנחנו קוראים לנק' הראשונה לפני היציאה,  $y_n$ , ואז אנחנו הולכים לפי החיצים כדי לעשות את הפעולות המתאימות להן, ותמיד מקבלים מהכניסה של X, את  $x_n$

### ARMA

ניתן להחליף בין AR לMA, כי 2 הבלוקים הגדולים האלה הם מסננים, וניתן להחליף ביניהם, ידוע.

כעת, ניתן לראות שבכל נק' זכרון לאורך הטורים, זה בעצם אותו אות, באותו זמן, לכן "נאחד" ביניהם/ נדביק אותם, וכך נחסוך חצי מנק' הזכרון.

למטה, רואים את התהליך, מימין לשמאל (בכל 1 יש 2 טורים).



**משפט הטנספורמציה (פבניל): אם**

1. נחליף את הקלט והפלט
  2. - " - הכיוון של כל החיצים
  3. כל פיצול, מחליפים במחבר, וכל מחבר מחליפים בפיצול (כדי שיסדר מהבחינה של כמה נכנסים וכמה יוצאים)
- כעת, אם עושים את כל זה בו זמנית, מקבלים גרף שזהה במובנו לגרף המקורי

**סיכום- 4 טרנספורמציות על גרפים:**

1. טופולוגיה
2. החלפת הסדר בין מסננים
3. מציאת נק' זיכרון מיותרות, והסרתן.

**שיטות באלגוריתמים:**

לקיחת סדרה וחלוקתה לתתי סדרות עם אותה בעיה ← דסימציה (לקיחה של 1 מ 10, הרחיבו את המושג ללקיחה של 1 מ N) אלגו' IN PLACE - שדורש לכל היותר זיכרון קבוע (לא תלוי בN), בנוסף לזיכרון שנדרש כדי לשמור את הקלטים.

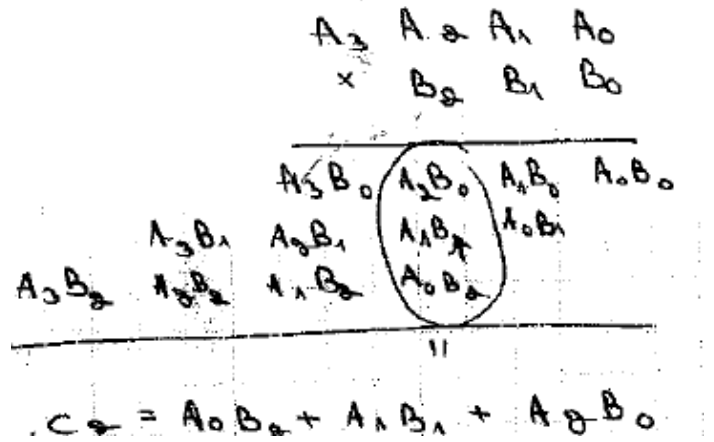
**דסימציה:** לפי LSB (חלוקת האינדקסים לקבוצות לפי זוגיות או אי- זוגיות)

**חלוקה:** לפי MSB (חלוקה באמצע)

בעיית החיבור (?):

להשלים, תכלס היה שיעור קודם ולא כתבתי

בעיית המכפלה



משמע, מכפלה זה קונב' של הכופל והמוכפל, וסיבוכיות הזמן היא  $O(N^2)$

האלגוריתם של תום וקוק (Toom Cook):

נרצה לראות אם ניתן להקטין את הסיבוכיות מ  $O(N^2)$

$$A = A(L)2^{\frac{N}{2}} + A(R)$$

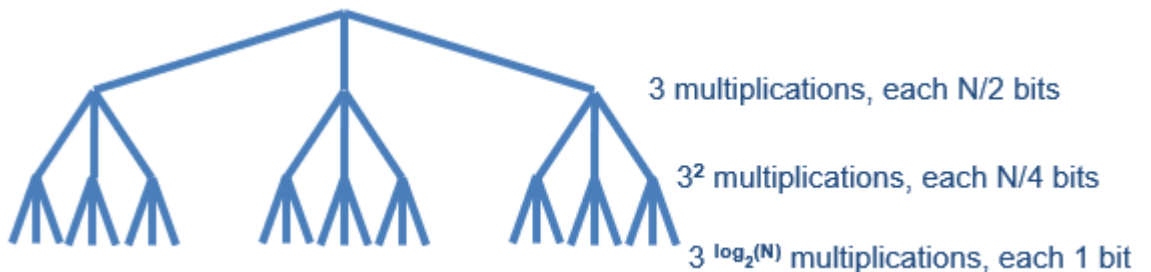
$$B = B(L)2^{\frac{N}{2}} + B(R)$$

$$AB = \left( A(L)2^{\frac{N}{2}} + A(R) \right) \left( B(L)2^{\frac{N}{2}} + B(R) \right) = A(L)B(L)2^N + (A(L)B(R) + A(R)B(L))2^{\frac{N}{2}} + A(R)B(R) =$$

נוסיף ונחסיר 2 איברים זהים, עמ"נ לקבל משהו שהסיבוכיות שלו קטנה מ  $n^2$

$$A(L)B(L) \left( 2^N + 2^{\frac{N}{2}} \right) + (A(L) - A(R)) * (B(R) - B(L)) * 2^{\frac{N}{2}} + A(R)B(R) \left( 2^{\frac{N}{2}} + 1 \right)$$

ככל שנמשיך לחלק לתתי קבוצות כל בעיה, ונמשיך בצורה הזו, נוכל להקטין את הסיבוכיות ל:  $O(N^{\log_2(3)})$



זמן אמת קשיח

$$S_k = \sum_{n=0}^{N-1} S_n W_N^{nk} : \text{DISCRETE FOURIER TRANSFORM-DFT}$$

אם יש לי זמן בין 2 דגימות של X, אני חייב בפחות מהזמן הזה, לחשב את Y לזמן T, כי אחרת, כמות הדגימות שאשים בצד תמשיך לגדול אקספוננציאלית, ולא יהיה לנו מספיק זיכרון להכיל את הכל, עד שנאבד מידע, לבסוף.

אנו רוצים זמן אמת קשיח.

## DFT-2 אופציות:

- עד הדגימה ה-N-ית, לא עושים כלום חוץ מלשמור את המידע. כאשר מקבלים את המידע ה-N-י, עושים את כל החישוב שצריך כדי להוציא בזמן את הפלט
- DOUBLE BUFFERING - יש 2 באפרים בגודל N, כאשר N זה גודל הקלט שמעבדים כל פעם. בהתחלה, ממלאים את הראשון. עד שהבאפר השני נגמר, רוצים לסיים את כל החישובים שיש לי על הבאפר הראשון, ואז מוציאים את הפלט עליו, כאשר מסיימים "לקלוט את הדגימות" הבאות, שנכנסות לבאפר השני.
  - בממוצע, על כל דגימה, צריך לגמור בזמן בין 2 דגימות.
  - אמנם על כל הבאפר צריך לסיים את החישוב בN מילי שניות.

## זמן אמת רך

לא צריך WORST CASE, מספיק שבממוצע מסיימים את החישוב של Y לזמן T, בפחות הזמן בין 2 הדגימות.

PARTITION IN FREQUENCY= PIT= DECIMATION IN TIME=DIT

PIT= PARTITION IN TIME= DIF= DECIMATION IN FREQUENCY

לא הספקתי לכתוב את ההסבר

## FFT

$$S_k = \sum_{n=0}^{N-1} S_n W_N^{nk} \quad \text{:DFT}$$

$$W_N = e^{-i \frac{2\pi}{N}} \quad \text{כי}$$

$$\begin{aligned} W_N^N &= 1 \\ W_N^{\frac{N}{2}} &= -1 \\ W_N^2 &= W_{\frac{N}{2}} \end{aligned}$$

נסתכל בנפרד על כל אלה ש n זוגי ועל כל אלה ש n א"ז (אפשר להחליף את X בS באותה מידה, זהה).

$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk} = \sum_{n=0}^{\frac{N}{2}-1} (x_{2n} W_N^{2nk} + x_{2n+1} W_N^{(2n+1)k})$$

$$\begin{aligned} \text{E- even} & \\ \text{O- odd} & \end{aligned} \quad = \sum_{n=0}^{\frac{N}{2}-1} x_n^E W_{\frac{N}{2}}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_n^O W_{\frac{N}{2}}^{nk}$$

ז"א שהאיבר הראשון זה DFT לאיברים הזוגיים, והשני זה DFT לא"ז.

$$X_k \quad \text{זה בעצם האיבר} \quad X_{k+\frac{N}{2}} \quad \text{בפרטישן הימני.}$$

לו הייתה סידרה של הא"ז בלבד עם חצי מN איברים, DFT שלה היה...

כל 1 כזה אנחנו יודעים לחשב בהרבה פחות זמן

וכל ההבדל בין הפרטישן השמאלי לימני זה המינוס.

**הסדר של האינפוטים:** עושים שיפט לפט ציקלי (אם משהו גולש, מכניסים אותו).

אמנם אם היה 0 או 1 בLSB, לא מזיזים אותו.

בכל שורה  $j$  שיורדים ממנה, מקבעים את  $j$  הביטים הLSB, ועושים שיפט לפט ציקלי רק על אלה הלא מקובעים.

יש אותו מספר חיבורים כמו המכפלות (אסימפטוטית), אמנם לכל פרפר יש בדיוק 2 חיבורים.

מעבדי אותות- DSP= DIGITAL SIGNAL PROCESSORS

היינו רוצים לעשות  
1 מהMACים  
האלה, עם כל מה  
שמסביב, בתיק 1  
של שעון.

אנו רוצים להוסיף  
לCPU הבסיסי, כדי  
שיהיה יותר ויותר  
יעיל, עד שנוכל  
לעשות לכל תיק  
של שעון- MAC

לכל CPU יש גביש  
שמריץ אותו, שכל  
שהוא מהיר יותר,  
ככה החישוב מהיר  
יותר (ה XTAL)

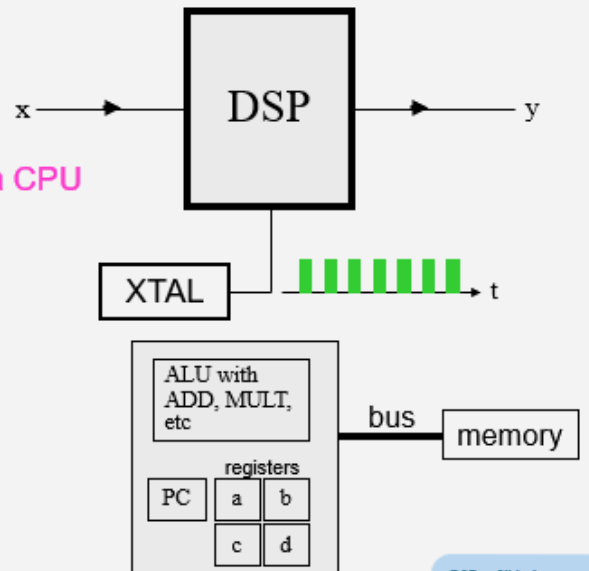
We have seen that the Multiply and Accumulate (MAC) operation is very prevalent in DSP computation

- computation of energy
- MA filters
- AR filters
- correlation of two signals
- FFT

A Digital Signal Processor (DSP) is a CPU that can compute each MAC tap in 1 clock cycle

Thus the entire L coefficient MAC takes (about) L clock cycles

For in real-time the time between input of 2 x values must be more than L clock cycles



ALU הוא הדבר שיועד לעשות חיבורים, מכפלות, לוגיקה, GOTO

PROGRAM COUNTER-PC - מצביע איפה אנחנו בקוד, ואיפה הפעולה הבאה. הוא מתקדם סריאלית עד שכתוב משהו כמו GOTO, שאז הוא נדרס.

הרגיסטרים- במקום לעבוד על כל הזיכרון החיצוני, נכניס את הזיכרון הנחוץ לרגיסטר. כך נחסוך כתובות (תחילת הביטים במחרזת האופקוד) ונגביר את המהירות.

BUS- הרבה חוטים שמקשרים בין CPU לזיכרון מסוג LAM - LOCATION ADDRESSABLE MEMORY ומעבירים את המידע המבוקש.



### the basic MAC loop is

```

loop over all times n
  initialize  $y_n \leftarrow 0$ 
  loop over  $i$  from 1 to number of coefficients
     $y_n \leftarrow y_n + a_i * x_i$  (j related to i)
  output  $y_n$ 

```

### in order to implement in low-level programming

- for real-time we need to update the static buffer
  - from now on, we'll assume that x values in pre-prepared vector
- for efficiency we don't use array indexing, rather pointers
- we must explicitly increment the pointers
- we must place values into registers in order to do arithmetic

```

loop over all times n
  clear y register
  set number of iterations to n
  loop
    update a pointer
    update x pointer
    multiply  $z \leftarrow a * x$  (indirect addressing)
    increment  $y \leftarrow y + z$  (register operations)
  output y

```

DSP

### We still can't count cycles

- need to take fetch and decode into account
- need to take loading and storing of registers into account
- we need to know number of cycles for each arithmetic operation
  - let's assume each takes 1 cycle (multiplication typically takes more)
- assume **zero-overhead** loop (clears y register, sets loop counter, etc.)

Then the operations inside the outer loop look something like this:

1. Update pointer to  $a_i$
2. Update pointer to  $x_j$
3. Load contents of  $a_i$  into register a
4. Load contents of  $x_j$  into register x
5. Fetch operation (MULT)
6. Decode operation (MULT)
7. MULT  $a*x$  with result in register z
8. Fetch operation (INC)
9. Decode operation (INC)
10. INC register y by contents of register z

So it takes at least 10 cycles to perform each MAC using a **regular CPU**

אנחנו אפילו מורידים את העניין של הבדיקה אם  $n=0$ , כאשר מפחיתים אותו כל הזמן ב1. אמנם אצלנו אין צורך לעשות את זה.

הכנסה מהזיכרון לרגיסטרים  
בCPU

- למשוך את הפק' שצריך  
ולתרגם אותה לפעולה אמיתית.

אנו רוצים למזער את כל ה10  
האלה לפק' 1.

Note that the result needs a special register

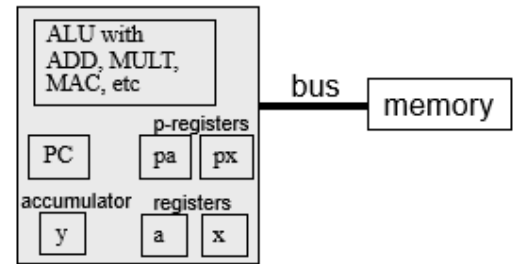
Example: if registers are 16 bit

product needs 32 bits

And when summing many need  $\geq 40$  bits

The code now looks like this:

1. Update pointer to  $a_i$
2. Update pointer to  $x_j$
3. Load contents of  $a_i$  into register a
4. Load contents of  $x_j$  into register x
5. Fetch operation (MAC)
6. Decode operation (MAC)
7. MAC  $a*x$  with incremented to accumulator y



However  $7 > 1$ , so this is still NOT a DSP !

### REGISTER ARITHMETIC

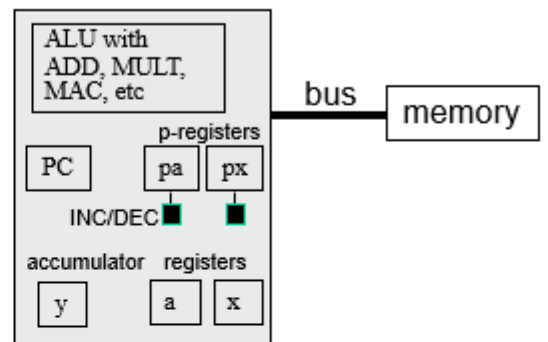
יש רק ALU 1 ובכל רגע הוא יכול לעשות רק פעולת חישוב 1, לכן נוסף רגיסטרים מיוחדים, שידעיים להוסיף או להחסיר קבועים, בלי להעסיק את הALU. נוסף 2 כאלה.

### The two operations

- Update pointer to  $a_i$
  - Update pointer to  $x_j$
- could be performed in parallel  
but both performed by the ALU

So we add pointer arithmetic units  
one for each register

Special sign || used in assembler  
to mean operations in parallel



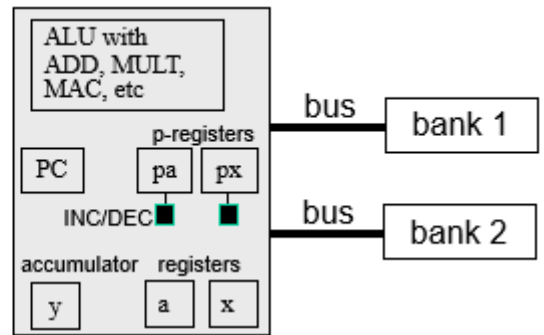
1. Update pointer to  $a_i$  || Update pointer to  $x_j$
2. Load contents of  $a_i$  into register a
3. Load contents of  $x_j$  into register x
4. Fetch operation (MAC)
5. Decode operation (MAC)
6. MAC  $a*x$  with incremented to accumulator y

However  $6 > 1$ , so this is still NOT a DSP !

|| בDSP  
מראה על  
מקבול של  
2 פקודות.

בזיכרון אפשר לעשות בו זמנית פק' 1 בלבד. אם אנו רוצים לעשות בו זמנית 2 פעולות לזיכרון, צריך 2 זיכרונות. BUSים הם אלה שמעבירים את המידע, ו1 לא יכול להעביר מידע ל או מ2, לכן יש 2 ג"כ.

So we add another bus  
and we need to define memory banks  
so that no contention !



מתכנת  
DSP  
הוא  
האחראי  
להכניס  
את  
המידע  
הנכון  
לבנק  
הנכון.

There is dual-port memory  
but it has an arbitrator  
which adds delay

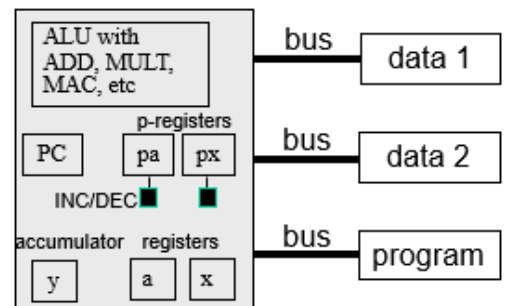
1. Update pointer to  $a_i$  || Update pointer to  $x_j$
2. Load  $a_i$  into a || Load  $x_j$  into x
3. Fetch operation (MAC)
4. Decode operation (MAC)
5. MAC  $a*x$  with incremented to accumulator y

However  $5 > 1$ , so this is still NOT a DSP !

אם הוא לא במקום הנכון, יש 2 סוגים של מעבדים:

- 1 כדי למשוך כל משתנה, צריך 2 קלוקים (כי הם באותו בנק ויש רק באס 1).
- 2 מחזיר זבל

שיכול בעזרת הזיכרון



Van Neumann architecture

- one memory for data and program
- can change program during run-time

Harvard architecture (predates VN)

- one memory for program
- one memory (or more) for data
- needn't count fetch since in parallel
- we can remove decode as well (see later)

משום שזה  
במקום 1  
זה MARK2,  
המחשב הראשון  
מעבד DSP הוא  
היחיד שמתמש  
בסוג הרווארד  
היום. מאפשר לנו  
לותר על  
FETCH  
והDECODE  
ופשוט להכפיל.

1. Update pointer to  $a_i$  || Update pointer to  $x_j$
2. Load  $a_i$  into a || Load  $x_j$  into x
3. MAC  $a*x$  with incremented to accumulator y

However  $3 > 1$ , so this is still NOT a DSP !

לפני שמעדכנים את הפוינטר, אין מה לעשות אתו, ולפני שעושים למעודכן LOAD, לא ניתן לחשב איתו. אז כביכול לא ניתן לשפר עוד, אבל ניתן להשתמש במיקבול מסוג PIPELINE

Then, on average, it takes 1 tick per tap  
actually, if pipeline depth is D, N taps take N+D-1 ticks

For large  $N \gg D$  or when we fill the pipeline  
the number of ticks per tap is 1 (this is a DSP)

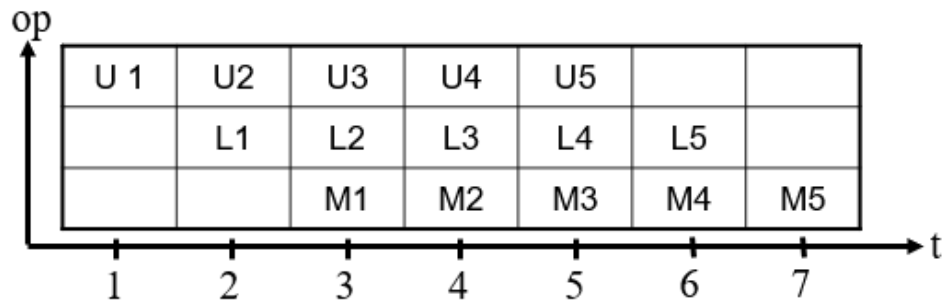
U זה היח' של ה update arithmetics

L זה ה LOAD

M זה ה MAC

אם עומק ה PIPELINE הוא D, הזמן שצריך הוא  $2+D$

העבודה של מתכנת ה DSP היא למלא את ה PIPELINE



### בעיה

Most DSPs are **fixed point**, i.e. handle integer (2s complement) numbers only

- floating point is more expensive and slower
- floating point numbers can underflow
- fixed point numbers can overflow

Accumulators have **guard bits** to protect against overflow

When regular fixed point CPUs overflow

- numbers greater than MAXINT become negative
- numbers smaller than -MAXINT become positive

Most fixed point DSPs have a **saturation arithmetic** mode

- numbers larger than MAXINT become MAXINT
- numbers smaller than -MAXINT become -MAXINT

this is still an error, but a smaller error

There is a tradeoff between safety from overflow and SNR

אלה שעושים MAC לתוכם, יהיו גדולים יותר מלכתחילה. והאיש שעושה את מעבד ה DSP דואג שלא יחברו יותר מהמקסימום

גורמים לכך שזה לא יקרה כמעט

### המטרה של ה DSP

מקבל סיגנלים דיגיטליים מהסיגנל החיצוני, מעבד אותם, ומוציא לעולם החיצוני.

עושה את זה בעזרת פורטים- חלונות.

### INTERUPT

אומר CPU שיש משהו שמחכה- וכך במקום כל כמה זמן, שהמעבד יעצור את עצמו מהחישובים וילך לדגום את החלון, לבדוק אם יש שם מידע, ה INTERRUPT אומר לו מתי לעשות את זה.

אפשר בפק' 1 להעתיק את כל הרגיסטרים לצל- לכל רגיסטר יש צל, לשרת את האינטרפט, כשהשירות הוא תמיד קצר במס' פק'- לא יותר מ4 (רק ב DSP), ואז לחזור. גם מעתיקים בבת 1 את כל מה ששמרנו בצל.

## בין DSP אינן CACHE

CACHE- זיכרון על CPU שאפשר לשים שם נתונים בהם נשתמש בסבירות גבוהה, כך שנחסך מאיתנו זיכרון להגיע לזיכרון החיצוני, ולחזור- ז"א לפחות 2 קלוקים.

מה שקורה בכל ה-CPUים זה שה-CACHE עובד רק ב% מסוים של הפעמים, אבל אפילו אם הוא גבוה, אנחנו רוצים לעבוד ב hard real time ב-DSP, ז"א שאסור לנו להגיע בשום מצב, גם אם זה לעיתים רחוקות באופן סטטיסטי, לחכות עד שמוציאים את המידע.

## בעיית הקורלציה

### קרוס קורלציה

2 סיגנלים  $x, y$ . אנו רוצים לדעת כמה הם דומים 1 לשני, למשל בשביל לשייך סיגנלים שמקבלים לסיגנלים הכי קרובים שמכירים, כדי להחזיר את הצליל המתאים או לדעת איך לנתח את מה ששמע.

לכאורה הבעיה פשוטה. נקרא ל:  $\epsilon = x - y$ , סיגנל השגיאה.

אם הוא מאוד קטן  $\leftarrow$  האנרגיה שלו קטנה.

$$E_\epsilon = \sum_n \epsilon_n^2 = \sum_n (x_n - y_n)^2 = \sum_n x_n^2 - \sum_n 2x_n y_n + \sum_n y_n^2 = E_x - 2C_{xy} + E_y$$

יש קשר הפוך בין האנרגיה של סיגנל השגיאה לבין  $2C_{xy}$ , ככל ש 1 גדל, ככה השני קטן (או לפחות, נרצה שיקטן).

- גם אם  $x=y$ , עדיין סיגנל השגיאה  $\neq 0$ , כי אנחנו תמיד מקבלים אותם עם קצת רעש.
- נניח ש  $x=y$  אבל כל סיגנל שאנחנו מקבלים בכל רגע, מוזז בזמן יחסית לשני, הם עדיין זהים, אבל הנוס' מעלה לא מראה זאת, לכן נשנה את  $C_{xy}$ , ל:  $C_{xy}^{(m)} = \sum_n x_n y_{n-m}$
- תשומת 3: ניתן לראות את הדימיון בעזרת  $C_{xy}$ , גם אם 1 הסיגנלים מוגבר יחסית לשני.

### אוטו קורלציה

חיפוש מקסימה של האוטו-קורלציה כפונ' של  $m$  ייתן את המחזוריות של סיגנל מחזורי  $\leftarrow$  כמה הסיגנל דומה לעצמו בזמנים שונים.

ב  $m=0$ , האוטו קורלציה הוא המקסימלי האפשרי, כי שווה לאנרגיה של  $x$

$$C_{xx}^0 = E_x = \sum_n x_n^2$$

$$c_{xx}^m = \frac{C_{xx}^m}{E_x}$$
 לכן מגדירים:

אז ניתן לראות שבזמן 0 הוא הכי דומה לעצמו  $\leftarrow 1$

ובאופן כללי הוא תמיד בין -1 ל +1

### גילוי סיגנל מוכר ע"י מסנן וינר

מכ"מ משדר סיגנל מוכר, הוא הולך במרחב, פוגע במשהו, מוחזר אלי, ואני קולט אותו. אבל אני קולט אותו יותר חלש ממה ששודר בהתחלה, וגם עם הרבה רעש, כאשר מאוד חשוב לנו מתי הוא חוזר, כדי לדעת לחשב בעזרת זה את המרחק בינינו לבין המטרה.

עושים את זה ע"י שליחת פולס, עם חתימה מסוימת, ואז לכל סיגנל שנכנס, בודקים את הקורלציה בינו לבין הסיגנל ששודר. מסנן וינר- לוקחים את הסיגנל ששידרתי, הופכים אותו (כדי להפוך קורלציה לקובולוציה), מכניסים למסנן, ואז מכניסים גם את הקלט שמתקבל. אם הסיגנל המתקבל שקט יחסית עד קבלת פיק, ואז שוב שקט, זה אומר שזה הסיגנל ששידרתי.

## מסנן קלמן

במקום לחפש סיגנל ידוע, מחפשים סיגנל שיוצא ממערכת שיוצרת סיגנלים, באותה צורה של חיפוש הפיק, בהכנסתו למערכת, כאשר אנו יודעים מה היא עושה.

## פרדיקציה- ניבוי סיגנלים

**אינטרפולציה- ניבוי סיגנל בין 2 נק' שראינו.** כמו משפט הדגימה.

**החלקה – סיגנל מורעש לצורך הורדת הראש**

איך ניתן לנבא סיגנלים לא מחזוריים? יול כבר עשה את זה, ע"י מסנן AR.

נניח שניתן לנבא את sn ע"י נק' 1 בעבר.

$$\widetilde{S}_n = bS_{n-1}$$

$$\epsilon = S_n - \widetilde{S}_n$$

$$\begin{aligned} E_\epsilon &= \sum_n \epsilon_n^2 = \sum_n (S_n - \widetilde{S}_n)^2 = \sum_n (S_n - bS_{n-1})^2 = \\ &= \sum_n S_n^2 - 2b \sum_n S_n S_{n-1} + b^2 \sum_n S_{n-1}^2 = (1 + b^2)E_S - 2bC_{SS}^{(1)} = 0 \end{aligned}$$

$$b = \frac{C_{SS}^{(1)}}{\epsilon_S} = C_{SS}^{(1)}$$

אין דרישה לsn להיות מחזורי או אפילו דטרמיניסטי, אבל אם הם יחסית דומים לעצמם, גם ככל שנכנסים לזום יותר גבוה או יותר נמוך, קוראים להם self-similar או פרקטלי.

דברים פרקטלים הם קשים לניבוי, דברים חלקים קלים לניבוי.

כדי למצוא קורלציה בין סיגנלים, והגבהים שלהם שונים, ניתן לנרמל אותם קודם.

דוג': שימוש בקווי זוויות פיבונאצ'י ( ע"י שימוש ביחס הזהב ולא הסדרה), עמ"נ לחזות את המניות – מתי כדאי לקנות ולמכור למרות שאין באמת שום סיבה שזה יעבוד, חוץ מהעובדה שאנשים מאמינים בזה.

לסיגנל יש תופעה שהוא יחסית קרוב לדגימה הקודמת, אבל אם גוזרים אותו ← לוקחים את ההפרש הסופי הראשון, מקבלים סיגנל רעש, אבל הספקטרום שיוורד לינארית, אבל מתמתן מאוד מהר והופך – לא בדיוק לקו ישר, אבל לא קופץ או נופל בהרבה מאיזה קו ישר.

ראינו שאקומולטור (אינטגרל) של רעש לבן (ערכים רנדומליים שקבענו) נראה מאוד דומה לסיגנלים של מניות.

לנג'וין- לפני שעושים אינטגרל, מוסיפים מקדם לסיגנל (ולא לרעש הלבן), שלוקח רק 0.99999 (או פחות) ממנו. נשים לב שככל שהמקדם קטן יותר, ככה הסיגנל המתקדם מורעש יותר.

טור טיילור- פולינומים, לא טובים לאקסטרפולציות, אלא רק לאינטרפולציות, כי מחוץ לתחום שבו הוא טוב, הוא יכול לשאוף ל+ או – אינסוף.

המסננים של FIR וסביצקי גולאי צריכים, כל 1, כדי לדעת מה יהיה בכל נק', numCoefficients/2 נק' קדימה ו numCoefficients/2 אחורה, מה שגם גורם לעיתים LAG בין הסיגנל לבין המסנן, לכן לא ניתן לעשות פרדיקציה עם מסננים פולינומיאליים.

## אדפטציה

מסנן אדפטיבי אינו מסנן, כי הוא לא אינווריאנטי כלפי הזמן.

בעזרת אדפטציה ניתן למחוק את הרעש, ע"י קליטתו בנפרד, והחסרתו. אמנם לא בדיוק יודעים מהו, לכן עושים הערכה לא אז עושים פרדיקציה לסיגנל ע"י האנרגיה.

בעזרת זה שבכל נק' שאני נמצא, מוצאים את הנגזרת, והולכים בכיוון הפוך לכיוון הנגזרת, אבל אם זה בהרבה מימדים, מחשבים את הגרדיאנט, והולכים נגד הכיוון שלו.

$$y_n = X_n + hq_{n-k}$$

$$\tilde{x}_n = y_n - cq_{n-k}$$

$$\begin{aligned} E_{\tilde{x}} &= \sum \tilde{x}_n^2 = \sum (y_n - cq_{n-k})^2 = \sum (x_n + hq_{n-k} - cq_{n-k})^2 = \sum (x_n - (h - c)q_{n-k})^2 \\ &= \sum x_n^2 + (h - c)^2 \sum q_{n-k}^2 \end{aligned}$$



משום שזו פרבולה (או פרבליוד, תלוי במס' המשתנים), בוודאות נגיע למינימום.

מבטל הדים עושה קרוס-קורלציה של הסיגנל ומבטל הדים ע"י גרדיאנט דסנט, כמו כאן.

## רשת עצבים – NEURAL NETWORKS

בהינתן סיגנל, אנו רוצים לגלות האם נאמרה המילה "שלום" / לא?

נניח שיש לנו דגימות ששומרים אותן בזיכרונות ואנו שמים עצב חדש, שהוא 1, אם כל הדגימות היו של המילה "שלום", ו-0 אם 1 מהן (לפחות) לא הייתה.

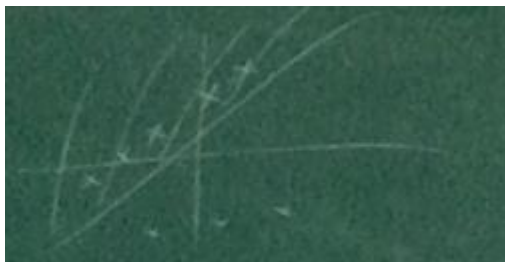
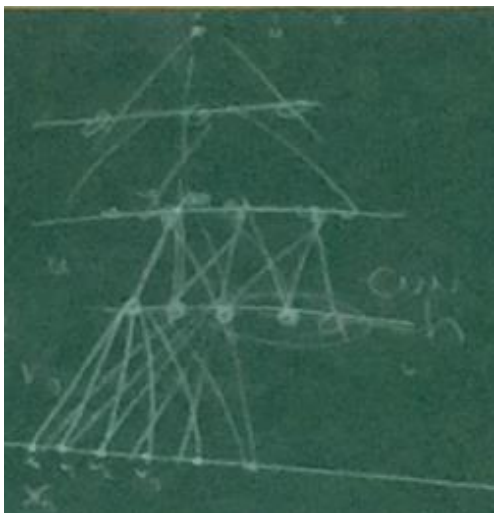
אם רוצים לגלות הרבה מילים, יש לנו הרבה עצבים  $\gamma$  כאלה.

אם היינו מנסים לפתור לפי מטריצות, נמצא שלא כל המשתנים רלוונטים, ואז גם לא יודעים מה לשים, ולכן המטריצה לא טמפליצית, אז קשה להפוך אותה ולהכפיל אותה. וגם כל העניין של העומק מתבטל, כי לבסוף זה הופך למטריצה 1.

אבל זה אם היינו מתעלמים מהעניין של חוסר הלינאריות, עם חוסר לינאריות זה יותר מסובך לעבוד לפי מטריצות.

$$y_i = \sum_n w_{i,n} x_n$$

עושים את ההפרש בין הרצוי למה שיוצא, בריבוע, מוצאים את השגיאה, עושים נגזרת לפונ' ומוצאים את הגרדיאנט, ועושים את זה עוד ועוד כדי לתקן את כל מה שהיה בדרך

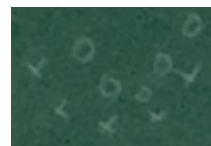


$$\sum_n W_{1n} X_n^{[1]} > \frac{1}{2}$$

$$\sum_n W_{2n} X_n^{[2]} < \frac{1}{2}$$

כל א"ש כזה מחלק את המרחב לקו  $\leftarrow$  היפר מישור. צד 1 הוא משהו 1, וצד שני הוא משהו אחר.

אבל אם השכבות די קרובות, לא ניתן לחלק:



**הפיתרון:** לעשות עוד שכבה בין  $x$  ל  $\gamma$ - שכבה מוסתרת,  $u$ , שהיא לא לינארית. בשנים האחרונות עושים יותר מאשר שכבה 1 מוסתרת, כאשר לאו דווקא כולם מחוברים לכולם.



## יישום של עיבוד דיבור

ניתן להסתכל עליו בתור אות אנלוגי, וזה אפילו גל.  
ניח שנסתכל בדיוק על הגל ליד הפה, ואז מקבלים סיגנל, כי לא מסתכלים עליו במרחב.

## **SPEECH RECOGNITION**

הבעיה מחולקת ל2 תתי תחומים:

**SPEAKER DEPENDENT** – 97% דיוק ברמת המילים

**SPEAKER INDEPENDENT** – 80% דיוק ברמת המילים

יש מערכת שעובדת טוב על הקול שלי, אבל לא על הקול של מישהו אחר.  
מעדיפים מערכת כזו או אחרת כתלות בביצועים.

אם לא ב99.4% - המאמץ למצוא את השגיאות ולתקן אותן, עולה יותר מאשר לקחת מישהו שיכתוב בעצמו.  
היום מדברים בצורה שמותאמת למערכת. ז"א אפשר לשפר כך שזה לא יהיה רק תלוי דובר, אלא גם דובר משתף פעולה.

## **SPEAKER RECOGNITION**

**זיהוי דובר פורנזי – אימות דובר** - אמור להיות דובר מסוים, יודעים מי זה, צריך להוכיח שזה הוא. מאוד מקובל בבימ"ש בתור בעיות, משום שבמחשוב ניתן לפתור את הבעיה הזו בצורה יותר טובה מאשר ע"י אנשים, ויכול להגיע לרמות דיוק, אפילו אם הדיבור מעוות, הרבה יותר גבוהות.

**זיהוי דובר מבין דוברים** - איזה מבין הדוברים שאני מכיר, מדבר, למשל כדי לכוון את השיחה למישהו מסוים

**זיהוי דובר למטרת בקרת גישה** – כמו בסרטים, שנותנים משפט והדובר צריך לחזור עליו, ורק אז הדלת נפתחת. רוצים משהו שאי אפשר להקליט, לכן מאתגרים את הדובר עם משפט אחר כל פעם. רמת הדיוק היא נמוכה מאשר זיהוי דובר פורנזי, אבל עדיין גבוהה יותר מאשר מה שבנ"א יכולים לעשות

**שאלה לזיהוי** - היום זה לא שווה הרבה, כי הרבה פעמים ניתן למצוא את המידע באינטרנט, אבל אם המערכת יכולה לזהות את הדובר, אפשר להמשיך, אפילו בלי שאלת זיהוי.

**ביצוע פקודות שבנ"א אומר למכשיר, תיעוד מדויק של פעולות** - שונים מהבנה כללית של מה שנאמר, משום שיש לזה דקדוק מאוד מיוחד, ומדויק, ולא ניתן להגיד את זה אלא במערכת הדקדוק הזו.

## **זיהוי מצב רוח, זיהוי שקר**

באותן מגבלות כמו בפוליגרף הרגיל - לא מזהה שקר אלא לחץ נפשי של הבנ"א.

לא ניתן לעשות את זה בטלפון, כי המאפיינים של הקול, באמצעותם ניתן לזהות שקר, לא עוברים בטלפון.  
אפשר למצוא מצבים נפשיים רק במידה מסוימת.

## **זיהוי שפה**

למשל כשמתקשרים לאיזה גוף, במקום להתחיל ללחוץ על מקשים, שיגידו משהו, ואוטומטית יפנו אותנו לדוברים שלה.

בעיה יותר קלה מתעתוק = זיהוי דיבור.

### יצירת דיבור - סינתזה

הקראת ספרים, המדריכה בויז.

האם ניתן לעשות דיבור, שבנ"א יאמין שהדיבור הוא של בנ"א אמיתי?

כשעושים סרטים באנימציה, את הוידאו עושים באנימציה ובכוונה מקלקלים, כדי שאנשים ידעו שזו אנימציה. אבל עדיין לוקחים שחקנים כדי לדבב.

### TEXT TO SPEECH ↔ SPEECH TO TEXT

בו זמנית, כמו סירי.

### VOICE MANIPULATION

מישהו 1 מדבר, נדביק את תכונות הקול של מישהו אחר, כדי שישמע כמו מישהו אחר.

שינוי מהירות דיבור:

1. האטה אחידה של סיגנל הדיבור, בלי להוריד את הPITCH, למשל.
2. לשנות באופן לא אחיד את מהירות הדיבור.
3. ליפ-סינקינג – כשמסריטים, לא רוצים להדביק מיקרופונים, הם מדברים אח"כ, וצריך לסנכרן ביניהם, וזו בעיה קשה מאוד.

### הטלפון המתרגם

כל צד מדבר שפה אחרת, בו"ז, כל צד מקבל את הדיבור המתורגם לשפה שלו.

### דחיסת דיבור – בשיעור הבא

### BSP – עיבוד אותות בצורה ביולוגית

1. המחוללת
2. המעבירה את המידע למוח

2 המערכות האלה לא מתאימות או מותאמות 1 לשנייה. הן עובדות על 2 עקרונות שונים לגמרי.

עקב כך, המערכת של העברת אינפורמציה ע"י דיבור, היא מאוד לא יעילה (ניתן להעביר מספר מאוד קטן של ביטים לשנייה, הרבה יותר קטן ממהירות היצירה והעברת המידע בטלפון או בפקס).

### המערכת הביולוגית המחוללת דיבור

2 סוגים של אקוסטיקה – אות, בהם משתמשים:

קולי-VOICED

בלתי קולי – UNVOICED

אם מרגישים ויברציה במיתרי הקול ביצירת קול, זה נקרא קולי, ואם לא, זה בלתי קולי.

פונמה

היחידה הקטנה ביותר של דיבור קולי. לכל שפה יש אוסף של פונמות, ואין לכל שפה בדיוק אותה פונמה כמו שפה אחרת.

**ההגדרה המדויקת:** היחידה, שאם מחליפים אותה ביח' אחרת, עלולה לשנות את משמעות המילה.

דוג': עבור TH ,theater הוא בלתי קולי. אמנם עבור TH ,then הוא קולי.

בעיית יצירת הפונמה היא לא במערכת המחוללת את הדיבור, אלא בזו השומעת את הדיבור. משום שהם לא רגילים לשמוע את הפונמה, הם לא שומעים את ההבדל בין הפונמות השונות בשפות השונות, כי אין להם את הפונמות האלה בשפה בכלל, לכן מעולם לא היו צריכים להבדיל ביניהם, לכן באמת לא מבדילים.

יש עשרות אחוזים של סינים לעומת 1-2% אנשים בעולם המערבי, שיש להם PITCH מושלם, משום שבסינית למשל, אותה פונמה, כשהיא עולה למעלה (PITCH) או למטה, יש לה משמעות שונה. תינוק נולד עם PITCH מושלם, אבל

## פרוסודיה - PROSODY

שינוי אינטונציה בצורה איטית במהלך המשפט- מעביר אינפורמציה, לא ברמה של פונמות, אלא עושה מודולציה ברמת המשפט, ומעביר את המשמעות של המשפט מרמה של ציטוט עובדה לשאלה.

## איך עובד סיגנל הדיבור?

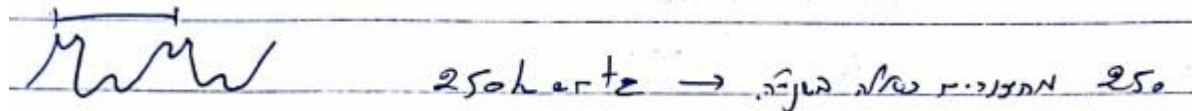
לא מורכב מסינוסים, אלא אחרת לגמרי, בכל רגע נתון יש המון תדרים בסיגנל.

התדר שאנו שומעים הוא הנמוך ביותר- ה PITCH

משתמשים באותה מילה עבור הזמן: התדר/1 – זמן המחזור של ה PITCH

כאשר מדובר ב

ברוב המקרים ההבדל ב PITCH בין אישה לגבר בקושי קיים, ההבדל הוא בין התהודה, כי לנשים יש פה יותר קטן, לכן ההגברה זה בתדירות יותר גבוהות



## PITCH

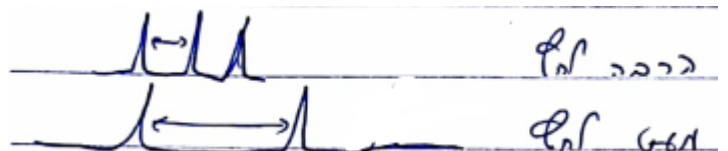
ממלאים את הריאות באוויר, ואז דוחפים אותו כלפי מעלה בלחץ מסוים, לקנה (בינתיים נזניח את האף, שהוא חשוב בשפות כמו צרפתית).

מיתרי הקול – שמם האמתי – קפלי קול – 2 קפלים כמו שרירים, שניתן לפתוח או לסגור אותם, כדי להעביר אוויר.

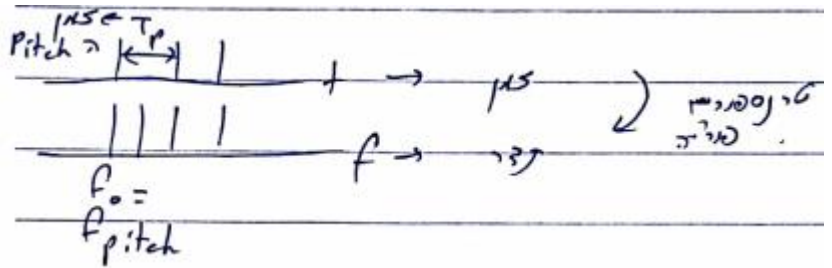
כשמדברים, לא פותחים או סוגרים את מיתרי הקול לגמרי, אלא שמים עליהם לחץ מסוים, עם אוויר לחוץ שנדחף מהריאות. אם הלחץ מספיק חזק, המיתרים נפתחים ומעבירים את האוויר החוצה.

ככל שהלחץ על קפלי הקול גדל, ככה צריך לחץ אוויר גבוה יותר כדי שהם יפתחו, ואם הוא לא גבוה מספיק, לוקח זמן, עד שהוא מצטבר, כדי שהם יפתחו.

הסיגנל הוא מחזורי, לכן הפולס יהיה קווי, גם הוא.



– לפי עקרון אי הוודאות, ככל ש  $t/f$  מצטמצם, אז השני  $f/t$  מתרחב, ולהפך.

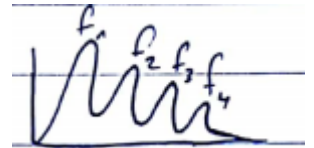


משם האוויר עובר דרך היוביולה (הענבל בפה). יש שפות שמתמשות בה, ויש שפות שמתמשות בה ע"י הנחה זמנית ע"ב הלשון. ויש שפות שלא משתמשות בה כלל.

ואז האוויר עובר דרך הפה, ותוך הזזת הלשון, ניתן לגרום לכל מיני תהודות בחלל הפה.

יש 4 איזורים באיזור השפתיים, וניתן להגדיל או להקטין כל איזור, ובכך לשנות את התהודה.

ב "נ" סוגרים את הפה עם הלשון, וב "מ" סוגרים את הפה עם השפתיים, לכן כל האוויר יוצא דרך האף.



לקחנו אוסף של פולסים והכנסנו אותו לתוך צינור, כאשר העומק של הצינור הוא אחראי על התהודה.



בשפות שיש גם אף וגם פה, יש אפסים בספקטרום, (כי יכולים

לצמצם האחד את השני  $\leftarrow$  ARMA), לעומת שפות שיש בהן רק פה, שיש בהן רק קטבים, יש רק 1-ים  $\leftarrow$  ציליים) בספקטרום  $\leftarrow$  AR (רק קטבים)

### פורמנט

לוקחים את הסיגנל התגובה להלם שנכנס, ומסתכלים על 4 תדרים שמגביר  $\leftarrow$  נקרא פורמנטים – פונמות שמחזיקים את הפה באותה צורה.

בדיבור קולי יש PITCH כי הוא מחזורי

ובדיבור לא קולי אין PITCH.

**לחישות –** ניתן להוריד את התדר הקולי מפונמות.

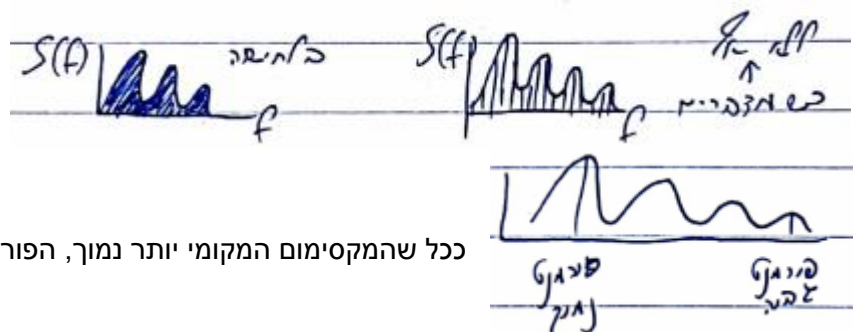
חלל הפה מגביר תדרים מסוימים.

כל פורמנט הוא זה שקובע מה אמרתי. ההבדל בין כל צליל הוא רק לפי מיקום הפורמנטים (כל עוד זה אותו PITCH)

יש עיצורים שנקראים PLOSIVES- שאוגרים את האוויר ואז משחררים בבת אחת. לכן צריך גם דינמיות, ולא רק פורמנטים.

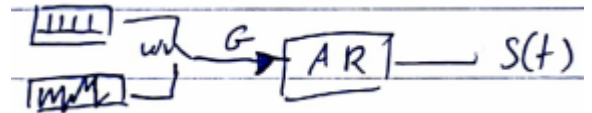
ממיתרי הקול לא יוצרים פולסים, אלא משהו שיש לו רוחב סופי.

יש אנשים שיש להם דיפורמציה של מיתרי הקול – למשל אנשים שמעשנים, לכן כל קו PITCH מחולק ל2 קווים, כי מיתרי הקול לא נסגרים כמו שצריך, אז כל הזמן עובר אוויר ביניהם, לכן נשמעים צרודים.



ככל שהמקסימום המקומי יותר נמוך, הפורמנט יותר גבוה (מבחינת הרץ)

מודל LPC לחולל דיבור – המציא ביסנו אתן(?) (הודי)



מתחילים ב-1 של 2 סיגנלי אקסיטציה או של פולסים בתדר הפיטץ', או רעש לבן. יש מתג שבוחר איזה מהם ניקח = u/v un/voiced

אח"כ יש G - Gate/Gain – האם מדברים חלש או חזק.

אז נכנסים למסנן בשם LPC synthesis filter, ומה שיוצא הוא סיגנל דיבור.

$$S_n = \sum_{m=1}^M b_m S_{n-m}$$

כיצד המשוואה של  $S_n$  נגזרת מהמשוואה של  $S_{n-1}$  ומה הקוד איז.

חייבים לדגום, לפי משפט הדגימה, ביותר מ-2 מהתדר.

הקצב שצריך כדי לדגום את הסיגנל / לאגור אותו על דיסק:

16 ביטים \* 2 \* 4000 (התדר) = 128000 ביטים לשנייה, כדי להעביר 16 ביטים לשנייה

בפחות מ-14 ביטים יש יותר מדי רעש, כי משווים לפי הרמה ← גורמים להזזה בסיגנל אחר.

סה"כ זה הרבה יותר מדי מידע בשביל להעביר די מעט, לכן מכווצים את המידע ל-8 ביט, תכף נראה איך.

מצב ה-PITCH – צריך לשמור רק 0 ביטים בשביל זה. כי ניתן לקחת את הסימן מ-U, וכי רק ב-0 אין PITCH

סה"כ 2000 ביט לשנייה זה בערך מספיק כדי לקבל משהו שנשמע כמו סטיבן הוקינג.

1/20 של שנייה – רואים סיגנל כלשהו.

איך יודעים אם סיגנל שרואים הוא קולי V או בלתי קולי U?

מסתכלים על הספקטרום – אם הוא קווי ← מחזורי, הוא למעלה, אחרת, למטה.

ניתן לעשות זאת גם ע"י אוטו - קורלציה. אם יש לו פיקים, הוא מחזורי, אם לא, הוא לא.

הLAG של ה-PITCH הראשון אומר מה תדר ה-PITCH

איך יודעים מה ה-GAIN?

אם שניהם מנורמלים ל-1, מוצאים אנרגיה, ואז עושים את סכום הריבועים, וזה ה-GAIN

מסנן AR

יש מסנן AR ואנו יודעים מה סיגנל הכניסה ומה סיגנל היציאה, זוהי הבעיה הקשה של זיהוי המערכת, בסוג AR, לכן הפתרון הוא ע"י משוואות YULE-WALKER

הצלחנו להוריד מ-128000 ביטים לשנייה, ל-2000

### המערכת הביולוגית שקולטת דיבור

צורת האוזן – המטרה היא לתפוס כמה שאפשר מהאנרגיה, ולרכז את זה למקום של התעלה (/ האנטנה בצלחות לוויין)

2 אוזניים – כדי שנדע מאיזה כיוון הגיע הרעש, כדי לדעת לאן לברוח.

הסיגנלים מ-2 האוזניים, מגיעים לאיזור במוח בו משווים את הסיגנלים מ-2 האוזניים, וב-1 מ-2 אלגוריתמים, מצליחים לזהות מאיזה כיוון בא הצליל:

1. אם הצליל הוא קליק או משהו חד, משווים את זמן ההגעה בין אוזן 1 לשנייה.
2. אם לא, שומעים ומשווים בין הפרש הפאזה של הסינוסים.

יש יותר הגברה קדימה מאשר אחורה, כי צורת האוזן מנחיתה, וככה אנחנו יודעים אם הצליל מגיע ממאחורה או מקדימה.

תנוך – לצורך שבירת סימטריה בין למעלה ולמטה, לדעת מאיפה מגיעה הסכנה.

עצם הפטיש – צד 1 מונח על עור התוף, שלמטה וצד שני על החלון של הקוכניה (השבול) שלמעלה.

עצם השבול / הקוכליה – מסובב ככה לצורך מזעור, כי אין מקום. כמו אמבטיה, צרה בצד 1 ורחבה בצד 2.

האוזן מסוגלת לשמוע טווח דינמי של 7 סדרי גודל:

הצליל הנמוך ביותר עד הצליל שכבר יעשה נזק לאוזן, קשה להשוות את שניהם בבת 1.

ראשית צריך לשמוע את הצליל הנמוך – יש לנו את ההגבר, התעלה ועור התוף.

הגלים שמגיעים מבחוץ, אלה שינויי לחץ אוויר, ומתופפים עליו ← מזיזים אותו. אמנם יכול להיות שהתנועה מאוד קטנה, אז צריך להגביר, הביולוגיה פתרה את בעיית ההגברה ע"י עצם הפטיש, שכאשר שומעים צליל נמוך, הוא זז מעט, ומזיז מעט את הפטיש מלמטה, אבל הוא זז הרבה למעלה על הקוכליה, ולכן מגביר.

גלים שונים יגרמו לגלים ברמות עומק שונות של הקוכליה, שכל עומק נוגע בנוירונים שונים, שמוביל לסיבים שונים, שמובילים לאיזור במוח בו קולטים את הצלילים ומנתחים אותם.



## מבחן

מבחן אמריקאי, כי אין בודק השנה. הבטיחו ששנה הבאה יהיה בודק.

40-42 שאלות. כל 1 2.5 נק', בדיוק תש' 1 נכונה, 2-3 תשובות שנראות נכונות. 4 אפשרויות.

מבחן לדוג' באתר

## פסיכופיזיקה

המדע שמנסה לקשור בין פסיכולוגיה לפיזיקה – מה הקשר בין סיגנל והנתונים שלו, לבין איך שהוא משפיע על האדם.  
**הניסויים של וובר – אדם גרמני:**

1. אוסף אנשים, בעיניים קשורות, בידיים פתוחות לצדדים, ומטבעות על כל יד.
  - a. 1 בכל יד – כולם הרגישו שזה זהה
  - b. 2 ו 3 – כולם הרגישו את ההבדל בין יד 1 לשנייה
  - c. 20 בכל יד – הרגישו שזה זהה
  - d. 20 ו 21 – לא הצליחו להבחין בהבדל.
  - e. 20 ו 22 – מצליחים להבחין בהבדל
  - f. 40 ו 43 – לא הצליחו להבחין.
  - g. 40 ו 44 – הצליחו להבחין.

המסקנה היא, שלא תמיד אפשר להבחין בתוספת של מטבע.

**מה שאנשים מרגישים זה לא תוספת של מטבע, אלא תוספת של % מסוים על הקיים.**

2. כוס מים עם מלח
  - a. קצת –
  - b. כפית – מרגישים בהבדל
  - c. %X יותר מלח ממה שהיה
3. 2 קווים – איזה יותר ארוך?
  - a. קטן וגדול מעט יותר – שמים לב
  - b. גדול וגדול מעט יותר – לא שמים לב

כל זה כדי למצוא מה הקשר בין היח' הפנימיות לפיזיקליות – החוק הפסיכופיזי.

## **החוק הפסיכופיזי**

פכנר המשיך את העבודה של וובר

ישב לכיוון השמש כל יום, עד שהתעוור, כדי לסמן מתי הוא שם לב להבדל בין האור לאור חזק יותר, מלפנות בוקר, עד אור יום.  
JND – JUST NOTICEABLE DIFFERENCE – ההפרש הקטן ביותר שניתן להבחין בו.  
לאחר הרבה שנים, ראייתו חזרה, והוא הצליח לפתור את הבעיה.

היחידות הפנימיות הן לוגריתם של היחידות החיצוניות  $\rightarrow \alpha = 1 + x\%$

**נוס' 1 ו 2**

JND גדל ביחידה, ככל שמכפילים את ה I

## האוזן

אנחנו שומעים בתחום דינמי של 7 סדרי גודל, כאשר השינוי הפיזיקלי ביניהם היא  $10^7$ , אנחנו יכולים להתמודד עם זה, משום שגם השמיעה שלנו היא לוגריתמית, גם בציר התדר (לכן אנו משתמשים באוקטבות – שמחלקים ל 12, ולא בהרצים) וגם בציר העוצמה (האנרגיה), לכן בו אנחנו מודדים ב dB, דציבל (כי בל זה גבוה מדי)

**נוס' 3**

P1, 2 אלה הספקים – האנרגיה / הזמן (מימין לשמאל).

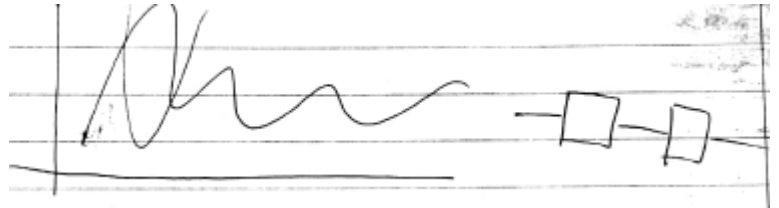
## עיניים

לא רק שאנו רגישים רק לוגריתמית לעולם החיצוני, אנו רגישים רק לשינויים. אנו אפילו לא רואים דברים שלא זזים. לדינזאורים לא היה שכלול בעין, לכן לא ראו דברים שלא זזים. ליונקים אין בעיה כזו, כי העין שלהם התפתחה. יונקים כל פעם קצת מזיזים את העין, לכן זה מכפר על כך שאנחנו לא רואים דברים שלא זזים. לכן בניתוחי עיניים, משאירים את האנשים ערים, כדי לקבל פידבק, אבל מרדימים את 3 השרירים של העיניים, כדי שהן לא יזזו. ואז המנתח רואה רק תנועה.

## קול

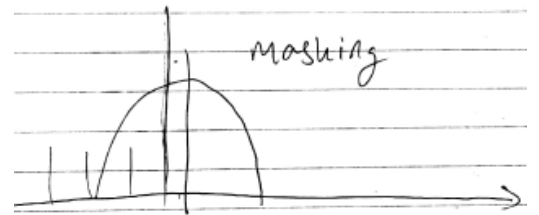
אם לוקחים פורמנטים של תנועה מסוימת, והופכים אותה, ומחברים אותם, מקבלים קו חלק.

אנחנו לא שומעים את הפורמנטים, שמכילים את האינפורמציה, אבל אנחנו שומעים את השינויים בפורמנטים, לכן עדיין ניתן להבין מה שנאמר, אבל זה יותר קשה.



## מיסוך – MASKING

אם אנחנו שומעים תדר מסוים, ומקרבים אותו יותר ויותר, יש רק תחום מסוים שבו אנחנו שומעים אותו, ואז אם משמיעים לנו צליל הרבה יותר גבוה/ חזק, אנחנו לא נוכל לשמוע יותר את הקודם.

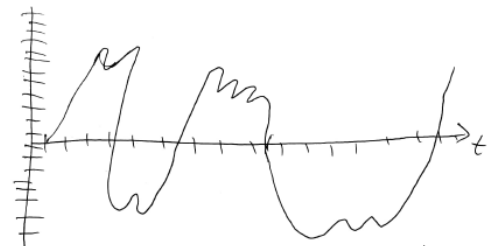


## דחיסת דיבור באיכות טלפון

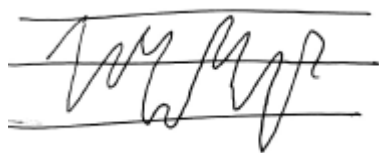
כבר דיברנו על מודל LPC, וגם מצאנו שאפשר להיעזר בו כדוחס לבערך 2000 ביטים לשנייה. אבל LPC אינו נשמע טבעי (כמו Steven Hawking), ואנחנו רוצים דוחס באיכות הרבה יותר טובה. נשתמש בחוק פכנר, לגבי העוצמה של סיגנל שאנחנו שומעים. סיגנל הדיבור –

256 מקטעים, כל פעם שמגיע הזמן, בוחרים בקרוב ביותר, ואז מוציאים את מספר הרמה.

אבל משום שבחרים בקרוב ביותר, ולא מדויק, כי יש מספר סופי של רמות, כל פעם יש שגיאה קטנה של קוונטיזציה - לפעמים שלילית ולפעמים חיובית, שלכל היותר ב+/- חצי רמה, כי אם זה היה מעבר לחצי, היינו משייכים את הדגימה לרמה הגבוהה / הנמוכה יותר.



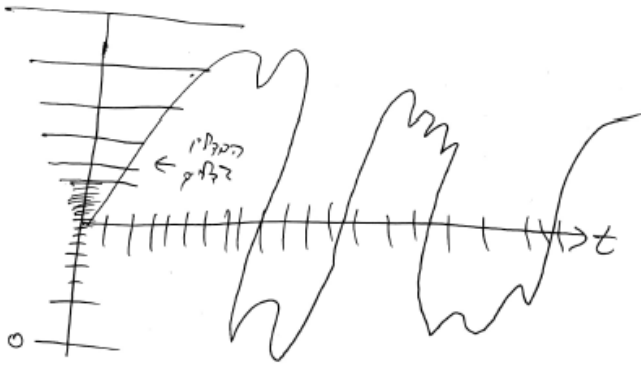
אז יש לנו סיגנל השווה לדיבור בתוספת רעש לבן.



חילקנו את הציר בצורה לינארית,



כעת נחלק בצורה לוגריתמית. בהתחלה אמנם יהיה מאוד צפוף, אבל אח"כ מתרווח יותר ויותר.



אם עושים את זה נכון, 8 ביט זה מספיק טוב, כאשר צריך להיות רגישים לשליליות והמעבר בין ה0 (כי לוגריתם עושים על ערך מוחלט, כי לא מוגדר לשלילי).

## שלבים לדחיסה:

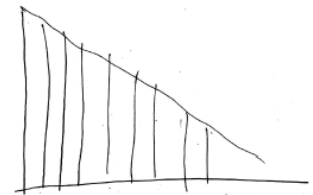
### 1. 64Kbps – PULSE CODE MODULATION – PCM

אם לוקחים 4000 כאלה בשנייה, ולכל 1 מספיק 8 ביט, הורדנו את קצב השידור ב 64 קילו ביט לשנייה, הרווחנו פי 2, בלי שהאוזן תשמע שום הבדל. 64 זה קצב השידור של עולם הטלפוניה הישן, כי הופכים כל זוג קווים לזה. עובד עם פקס, כי התאימו את הפקס לזה.

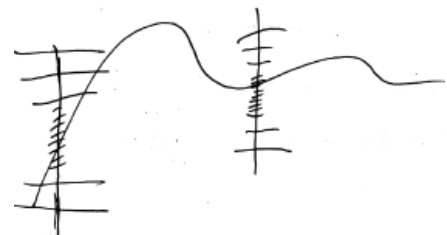
### 2. 32Kbps – ADPCM – Adaptive Delta (Difference) PCM

עד היום, כל השיחות הבינ"ל הן במתודה הזו. סיגנל דיבור הוא לא הסיגנל הכללי ביותר שתופס את ה4000 הרץ של קו הטלפון. הסיגנל הכללי ביותר נשמע כמו "פששש" של פקס. אמנם הסיגנל ממיתרי הקול, יורד עבור כל אוקטבה ב 12dB

יש יותר אנרגיות בתדרים הנמוכים, ולכן זה זז יותר לאט בציר הספקטרום והתדר. בסיגנל דיבור רגיל, ברוב הזמנים, 2 דגימות קרובות בזמן, הן גם קרובות בערך. ז"א שניתן לייצג את השינוי בין 2 דגימות, ולא כל דגימה באופן אבסולוטי – את הפרש הסופי הראשון



מה אדפטיבי? צריך לפתוח מסביב לדגימה מסוימת 16 רמות, ולקודד את זה באופן דיפרנציאלי. אמנם לא נפתח אותו גודל של דגימה לכל דגימה. לדגימה מחוץ לסקאלה, נפתח את הסקאלה, ובתוך הסקאלה, נקטין אותה.



אמנם כאשר יש לנו דגימה שהיא בין הרמות, זו בעיה שיוצרת לנו רעש. בתוך המנגנון המקודד, שמים גם את המנגנון שהמקלט משתמש בו כדי לשחזר את הדגימה. לכן כשבא המשדר לקודד בדגימה הבאה, הוא לא פותח את הסקאלה מסביב לדגימה הבאה האמיתית, אלא מסביב למה שהמקלט חושב שזו הדגימה האמיתית, כדי שלא תהיה צבירת שגיאות.



אנו יודעים מה המקלט חושב, כי יש סטנדרט לצורה בה עובד. ז"א ניצל את המערכת שמשמיעה. יש מזהה לטונים בפקסים. כי לא עובד בזה.

### 3. 16 Kbps – AVQSB – Adaptive Vector Quantize Sub Band Coding

חילקנו את תחום התדר בצורה לוגריתמית ל32 אזורים, אבל בדקו 16. בתדרים הנמוכים יש קו מאוד קטן, כשהאזורים גדלים, כך גם הקווים. קוונטיזציה סקלרית – לקחת מספר ולחלק אותו בצורה כלשהי – לינארית/ לוגריתמית/ כל צורה שאפשר ללמוד אותה.

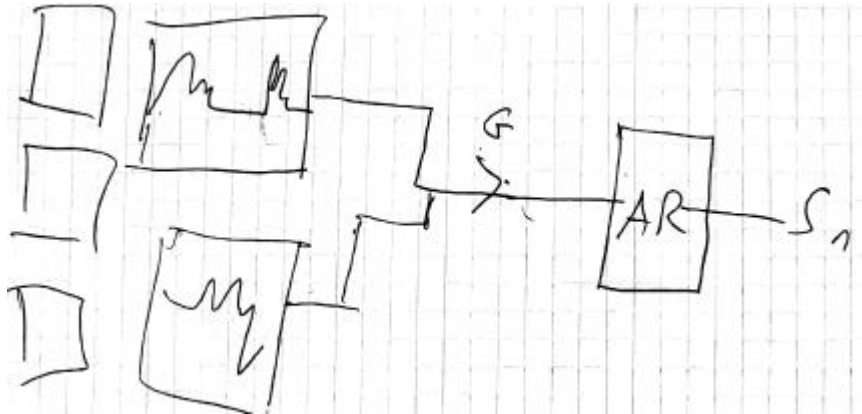
מחלקים את ציר העוצמות של הסיגנל בצורה לוגריתמית. רואים את עוצמות הסיגנל שמשמיעים, ועושים היסטוגרמה, כדי לראות את אופן השימוש בעוצמות, והיינו רואים שמשמשים יותר בנמוכות.

לכן כדי למזער את השגיאה, כדאי לשים יותר רמות באיזור שבו הוא יותר נמוך. אם עושים את זה נכון, ניתן להגיע לאיכות מאוד דומה ל32, אבל רק לדיבור. שוב ניצל את מערכת השמיעה.



**4. 8Kbps, 6 Kbps – CELP – CODE EXCITED**

במנגנון LPC יש אוסף של פולסים או רעש לבן. בוחרים ב1 או בשני, שמים Gain, מכניסים לתוך AR פילטר, ומוציאים. אמנם זה נשמע מאוד סינתטי, כי הסיגנל שיוצא מכפלי הקול, הוא לא של פולסים חזקים, אלא יש לו מבנה. אם מראש היינו יכולים לצייר את הפולסים במבנה כזה, היה יוצא קול פחות סינתטי.



**הרעיון:** אם היינו יודעים מה יהיה הסיגנל למעלה, ברור שהוא היה מחזורי.

בונים דבר שנקרא CORD BOOK - סיגנלי אקסיטיביה, אוסף של צורות פולסים (אמנם לא בדיוק ככה, אבל אפשר לחשוב). Analytical by Sympathis - ABS צריך להגיע למשהו שדומה לסיגנל שאני רוצה לקודד, כמה שאפשר. אנו יודעים למצוא את כל המקדמים, מוצאים את צורת הפולס, ע"י שמנסים את כולם, ומי שמוציא את הטוב ביותר, מנצח. לא זה שנראה הטוב ביותר, אלא נשמע הכי קרוב.

נשמע הכי קרוב - לוקח בחשבון את הלוגריתם בצורת העוצמה, הלוגריתם בתדר, המיסוך בתדרים חזקים, השינוי בפורמנטים. כל אלה ביחד בונים לי מדד שנקרא Itakura - Saito, אז ניתן לעשות "מרחק" כזה, שהוא יוציא לי את זה איך לבנות CODE BOOK שלא חייב לבדוק את כולם, אלא שניתן להקטין את הזמן ע"י כל שמנצלים מבנה אלגברי של הקוד.

כל המטרה של המבנים האלגבריים, היא שלא נצטרך לחזור על אלפי סיגנלים שונים, כדי למצוא את הסיגנל שרוצים להוציא.

**5. 2.7 Kbps**

בכלל לא מתאים לשום דבר, פרט לדיבור, אפילו לא למוזיקת רקע. לכן זה עף. בהתחלה חשבו שזה יהיה טוב ללוויינים, אבל לא מתאים לזה.

**הבנת דיבור**

How do you wreck a nice beach?  
How do you recognize speech?

נשמע מאוד דומה, אבל למה הוטענו?

The man cried fire  
The man cried fire the gun  
The man cried fire the gun maker

במשפט הראשון, העלנו את % הסבירות של FIRE אש, והורדנו את ההסתברות של המשמעויות האחרות. וככה בכל משפט, ההסתברויות של כל משמעות השתנו.

כל הזמן מה שאנחנו עושים, כשאנו שומעים דיבור, אלה הערכות, בקשר לאיזה פונמות שמענו, וכאשר מתקדמים, אנחנו הולכים אחורה - BACKTRACKING ועושים הערכה, ושוב ושוב, בלי שאנחנו שמים לב.

מחשב הרבה פחות מוכשר במחיקת "הארכות" של מילים / שינוי של טונים, כשזה לא משנה משמעות. אז זה לא פשוט כמו לשמוע פונמות ולכתוב אותן 1 אחרי השנייה.

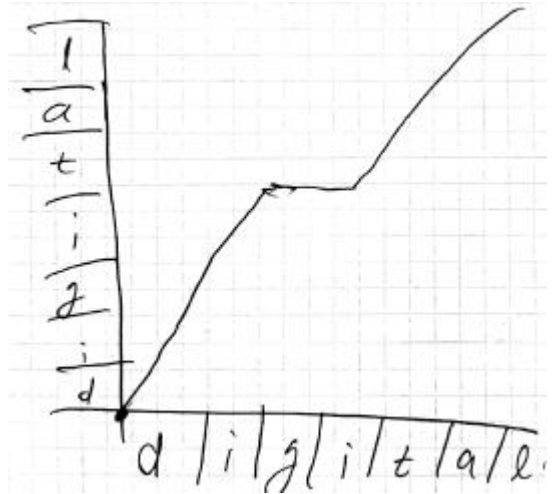
בתחילת הדיבור עשו ISOLATED WORD RECOGNITION, ואז זה עניין של השוואות מילה למילה. אמנם אנחנו לא עושים את זה בחיים האמיתיים.



- בסונגרמה, כשיש הפסקה, זה כנראה 'ב' או 'ה', ואין הפסקה (/ רווח בשיעור) בין מילה למילה.

### המרחק של לבנשטיין

סופר כל החסרה או הוספה בתור שגיאה 1 – גם יכול להיות תלוי באות וכל החלפה בין אותיות, או בתור שגיאה או כתלות במרחק במקלדת. מוצאים את המספר הקטן ביותר של שינויים של המילה, ביחס למילון.



– מהמשפחה של DYNAMIC PROGRAMMING, כמו דינקסטר – LTW אם אני אומר אותה מילה, רק יותר איטי. Dynamic time workpeak – DTW – חלק מהמילה דחוסה, חלק מרווחת. הייתה נפוצה לכמה זמן, אבל העלות שלה בזמן הריצה היא גבוהה. צריך דוגמאות של כל מילה מראש, ואיך היא נאמרת, וכל הצורות בה היא נאמרת, לכן מאוד כבד ולא יעיל.

### HIDDEN MARKOV MODEL – HMM

מודל מרקובי – יש בו מצבים, שבכל רגע המערכת במצב מסויים, ויש לו הסתברות לעבור מהמצב שהוא נמצא למצב הבא, והסתברות להישאר באותו מצב.

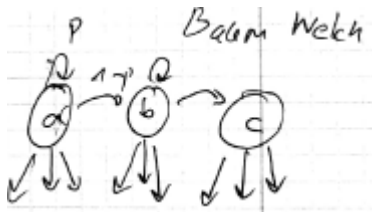
– לדוג' למחרוזת BCA, ההסתברות היא 0



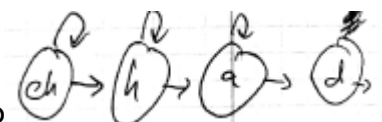
אבל יש גם מודל מרקובי שמאפשר דילוגים, ואז זה הופך להיות יותר מסובך. יש גם כאלה שמאפשרים לחזור אחורה (לא LEFT TO RIGHT), אבל לא נדבר על אלה.

בונים מודל מרקובי לכל מילה, ויש אלגוריתמים למצוא את ההסתברויות לכל פונמה. אבל אנחנו מראש לא יודעים למצוא פונמות, כי כל 1 יכול ליצור "אותה" פונמה בצורה אחרת.

מודל מרקובי חבוי – אי אפשר לראות את המצבים, אבל בכל מצב יש הסתברות לפלוט צליל מסויים. אז יש הסתברויות מעבר ויש הסתברויות קליטה.



השיטה עצמה: יש אלגוריתם Baum – Welch, שבהינתן אוסף מחרוזות של מילים, שידוע מה הן, ניתן לאמן – למצוא את הסתברויות המעבר/ הדילוג, ואת הסתברויות הפליטה. משקיע מראש את זמן הלימוד.



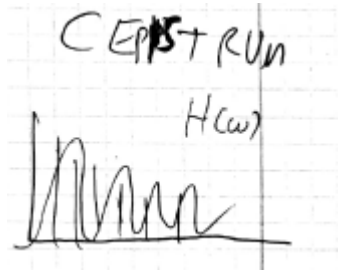
כאן דרגות החופש הן יחסית מעטות. וכל מצב, ההסתברות להישאר בו היא

אקספוננציאלית בזמן.

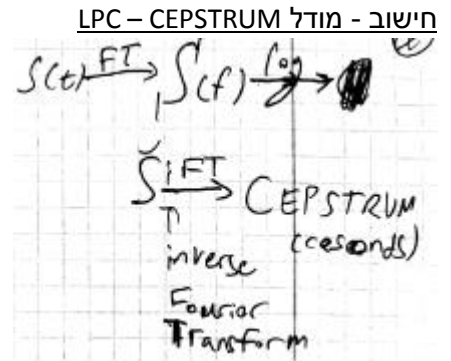
בסוף שנות ה-90 נתקעו, עם 70-85% דיוק לא תלוי דובר, 93% דיוק תלוי דובר, בגלל הטכנולוגיה. היום עם DEEP LEARNING, זה הרבה יותר גבוה.

סירי וקורטנה הן מעורבות HMM, deep learning אמנם הרעיון הוא להשקיע כמה שיותר זמן מראש כדי שיהיה כמה שיותר מדוייק.

המודל שלנו שמחולל דיבור, לוקח סיגנל ומכפיל אותו ב  $H(w)$  התוצאה נובעת ממכפלה של 2 חלקים, מה שיוצא ממיתרי הקול ומחלל הפה (והאף) היח' לסיגנל אנלוגי של מה שיש בסוף, זה כאילו חזרנו לציר הזמן  $\leftarrow$  שניות, אבל כדי להבדיל כותבים cesonds באופן דומה: מפליטודה (אמפליטודה), קיפרנסי ( פרקוונסי) וסאפה (פאזה). ה LIFTERING יודע להוציא את הפורמנטים, ולמצוא את הקיפרסני של ה PITCH והקיפרנסי של הפורמנט.



אם מסתכלים על הקפסטרום במרחב שלו, יש לו תכונה שהמרחק האוקלידי במרחב הזה, הוא כמעט זהה למרחק איטקורה- סאיטו (כמה 2 דברים נשמעים אותו דבר).  $\leftarrow$  נק' קרובות נשמעות דומות, ורחוקות, נשמעות שונות. הכי טוב כי זה טרנספורמציה של הדגימות המקוריות למרחב שיש בו משמעות למרחק ולקרבה. יש עוד דרכים לחישוב, לא נדבר עליהן. גם בזיהוי דובר עושים את זה.



## תקשורת נתונים DSP

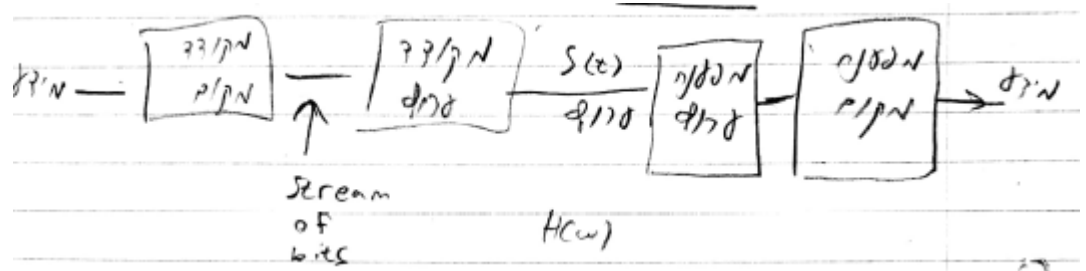
### משפטי שאנון

רק השלישי קשור ל-DSP – אותו נוכיח. 1 אחר קשורים לתורת התקשורת ו1 לתורת האינפורמציה.  
לפני שאנון לא ידעו אם עדיפה תקשורת אנלוגית או דיגיטלית (ספרתית). אחריו, היה ברור שעדיף דיגיטלי.  
משום שאם נתון לנו מידע אנלוגי, ונאמר לנו שהוא מורעש, לא נוכל לדעת מה הסיגנל המקורי. אמנם באנלוגי, אם אומרים לנו ספציפית שאיזה ביט מורעש בוודאות, או בוודאות לא מורעש, נוכל לדעת בוודאות מה הסיגנל ללא הרעש.  
CD מקליטים דיגיטלית, לעומת קסטות שמקליטים אנלוגית – ולא היה אכפת לאף אחד שמשכפלים את זה, כי כל שכפול הגביר את הרעש, בשונה מהקלטה/ צריבה דיגיטלית.

### משפט ההפרדה – תורת התקשורת

**הביט (או הנאט) של שאנון** – חפיפה של אינפורמציה (בנאטים משתמשים פיזיקאים).

אם רוצים להעביר אינפורמציה (הגדרת אינפורמציה זה במשפט קידוד המקור) ממקום למקום, הדרך הטובה ביותר לעשות זאת היא ע"י הפרדת המערכת כדלהלן:



**מקודד המקור** – לוקח מידע מכל סוג (תמונה, שיר, רגש, ...) שהוא והופך אותו לביטים ע"פ שאנון. מוריד למינימום הביטים האפשריים.

**מקודד ערוץ** – לוקח סטרים של ביטים, שפולט מקודד המקור, ובונה סיגנל אנלוגי שניתן לשלוח בערוץ.

**ערוץ** – יכול להיות האוויר כשמעבירים בגלי רדיו, או גל, סיב אופטי וכו'. חייב להיות סיגנל אנלוגי.

**מפענח הערוץ** – הפונ' ב-1 של מקודד הערוץ.

**מפענח המקור** – הפונ' ב-1 של מקודד המקור.

ואם עושים הכל לפי הכללים, מקבלים אותו מידע שנכנס בקלט.

### כל ערוץ פיזיקלי עושה 2 דברים לכל סיגנל שעובר דרכו:

1. מוסיף לו רעש  
a. כי הוא מושפע מ:  
i. קרינה קוסמית  
ii. טמפרטורה – כשהיא שונה מ0, האלקטרונים זזים, וכל תזוזה גורמת לשינוי של המידע, והיא נקלטת ולכן מתווסף רעש.
2. מעוות – מכפיל את הסיגנל בתגובה לתדר שאינו שטוח לכל התדרים – כל ערוץ מעביר איזה טווח של מתחים, ומה שלא יכול לעבור דרכו, הוא מנחית.  
a. גם לסיב אופטי (אור – לפוטונים אין משקל) וגם לערוץ של ריק יש טווחים.

משפט קידוד המקור - source encoding - תורת האינפורמציה - מזה היא נולדה.

### הדרך לכמת אינפורמציה:

המספר הקטן ביותר של שאלות שניתן לשאול כדי למצוא על מה חושב האחר.

**מספר הביטים של אינפורמציה** – המספר הקטן ביותר של שאלות שניתן לשאול כדי לקבל את התשובה.

לדוג'

אם ידוע שנבחר מספר בין 0 ל-15:

- נשאל אם המספר גדול מ-7: 1 אם כן, נניח שכן. ← נשאר כל המספרים מ-8 עד 15
- נשאל אם המספר גדול מ-11: 1 אם כן, נניח שלא ← נשאר כל המספרים מ-8 עד 11
- נשאל אם המספר גדול מ-9: 1 אם כן, נניח שכן ← נשאר כל המספרים מ-10 עד 11
- נשאל אם המספר גדול מ-10: 1 אם כן, נניח שלא. ← נשאר רק 10.

קיבלנו 1010 = 10 בבינארית ← המצב היחיד בו הביט של תוקי = הביט של שאנון.

יש גם נוסחא שאומרת, בהינתן אינפורמציה, כמה ביטים יש. אמנם הוא היה מתמטיקאי, ולכן הוא לא אמר איך לבנות אותם, אלא מה יהיו הביצועים האופטימלים שלו.

למפל – זיו (מהטכניון) היו הראשונים שהוציא מקודד מקור גנרי שהתקרב לביצועים של שאנון.

### משפט קיבולת הערוץ – channel capacity

**C (bit/sec)** – אם לא נעבור את קצב הקיבולת של הערוץ, נוכל – גם אם יש רעש וגם עם רוחב סרט סופי, לקבל בצד השני את אותו המידע ששלחתי.

**N** – הרעש שמתווסף בזמן N שלוקח לשלוח את המידע

**Signal Noise Ration - SNR = S/N** → מספר הביטים לסימבול =  $\log_2(1 + SNR)$

$$C(\text{bit/sec}) = BW * \log_2(1 + SNR)$$

### הוכחה:

1. נניח שרוחב הסרט אינסופי, נראה שניתן להמיר את כל המידע המתקבל למקורי (שנשלח), גם אם הרעש מאוד גדול (הרעש חסום בין -N/2 ל N/2)

- a. אם אנחנו שולחים סיגנל  $S = 5$ , שמים אצלנו 1, בצד השני מקבל קצת מעל או קצת מתחת ל-5, בכ"א תחום עד  $S + / - N/2$ , לכן, גם אם הצד השני מקבל אותו מורעש, הוא עדיין ידע אם לסמן 0 או 1, בדיוק לפי הסימן שסימנו אנחנו, לפני הרעש
- b. מסקנה: לכל כמות של רעש (חסום), אמנם שאנון הוכיח גם ללא חסום), ניתן "לסנן" אותו ולקבל את כל המידע המקורי.

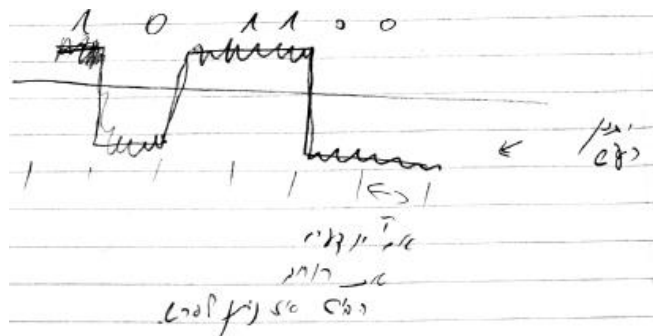
2. נניח שאין רעש, נראה שניתן לשלוח אינסוף מידע על הסרט:

- a. נקודד את כל המידע בכל הספרים לפי האסקי, נוסף "0" לתחילת הספרים שקיבלנו, כעת המספר חסום, ולכן "רוחב הסרט חסום", משום שקטן מ-1, ואין רעש, לכן בדיוק המספר הזה מגיע לצד השני.
- b. בצד השני יודעים פשוט להסיר את ה "0" כדי לקבל את כל הספרים, ומצאנו דרך להעביר מידע ברוחב סרט לא מוגבל בסרט מוגבל, משום שאין בו רעש.

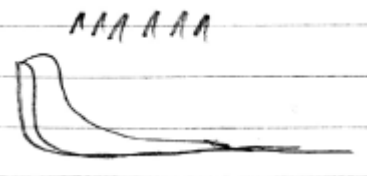
שאנון לא היה קונסטרוקטיבי ב-2 המשפטים האחרונים – לא אמר איך לבנות מקודד מקור / ערוץ.

No Return to Zero – NRZ

כמו במשפט הראשון, משדרים או 0 או 1. אז אם מישהו יודע מה זמן הביט, הוא יכול למצוא ע"י השוואה לסף. כמובן שיש רעש, אבל כל עוד הוא לא גבוה מדי, עדיין ניתן למצוא.



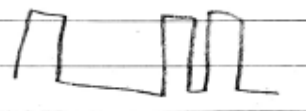
כשרוחב הסרט גדל, גרף הסיגנל נמתח לרוחב, ויותר ויותר קשה לראות את ה-1-ים. אם יש הרבה 1-ים, הוא דועך מאוד מהר



המספר המקסימלי של שינויים שיכולים להיות בשנייה זה גודל הביט/1 ככל שהחוט יותר ארוך, רוחב הסרט שלו הולך וקטן (נהיה יותר LOW PASS) – לכן NRZ זה טוב, כשהחוט קצר יותר.

### RZ – Return to Zero

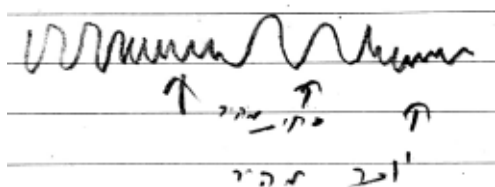
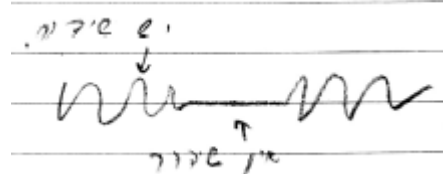
מחלק כל ביט ל-2, כדי שכל "1" יהיה ניתן לראות בעלייה משלו, במקום מ"רצף" של "למעלה", שדועך מאוד מהר.



צריך רוחב סרט כפול מ NRZ, כי מחלק כל ביט ל-2.

### FSK VS OOK

OOK – On Off Key יש יותר "מרחק" ← אפסים, בין כל סינוס (או 0) שונה



FSK – כמו הצליל של הפקס, יש בו יותר מידע, כי אין 0-ים בין הסינוסים, אלא ישר עובר לסינוס שונה.

FSK יותר יעיל כי, כשמשמיעים אותו יחד עם סיגנל אחר, הוא לא מתבטל כמו OOK ← עדיין ניתן לשמוע אותו, וניתן גם להחליט לבטל אותו, ע"י בדיקה איפה יש אנרגיה:

- אם יש אנרגיה למעלה ולמטה, זה רעש. לא אכפת לנו
- אם יש אנרגיה רק למעלה, זה מה שרוצים
- אם יש אנרגיה רק למטה, מתעלמים
- אם אין כלום בשניהם – 0 זה 0

### בעיה ב FSK

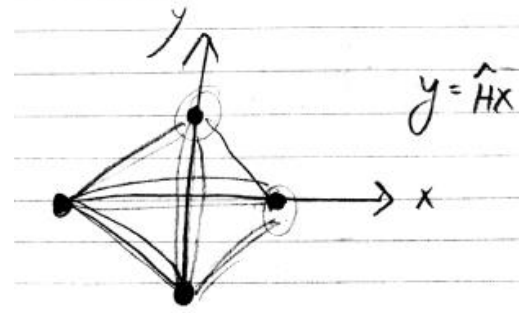
ככל שרוחב הסרט גדל – הסינוסים מתמרחים, עד שיכולים לחפוף אחד עם השני, ואם הם מורעשים, לא ניתן להפריד ביניהם – למרות האורתוגונליות של הסינוסים, בגלל הרעש.



ע"ס מחזור 1 של הסיגנל שמאפננים – בתדר או בפאזה, ניתן להפריד בין 2 סינוסים / קוסינוסים, ולקודד על זה 4 ביטים – 2 על כל ציר, בזכות החיוביות והשליליות, ו 1 עם ההתמרה של H.

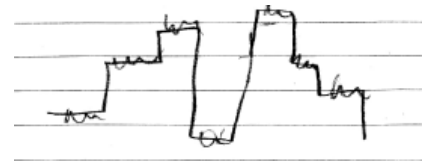
ניתן להגיע ל 4800 ביט לשנייה, אבל ככל שמתקרבים, העיגולים נהיים יותר גדולים, והם גם מתחילים לחפוף.

הצורות שנוצרות עקב ההתמרה נקראות קונסטלציות.



QAM– Quan Amplitude Modulation

ניתן לקודד על זה 8 ביטים, ולהגיע ל 9600 ביט לשנייה.



למה הגיעו רק ל 9600?

לא עבדו לפי שאנון, אלא חילקו את מקודד הערוץ ל 2 :

FEC – Forward error detecting Code – גורם לניפוח של האינפורמציה, ע"י כך שמנסים לתקן את השגיאות שאולי נפלו, מראש (או בדיעבד, תלוי במיקום), אמנם הוא יכול, לכל שורה, לתקן רק ביט 1, ולמצוא לכל היותר 2.

MOD – Modulator

אונגרבק

היה מהנדס שהיה גם מתמטיקאי, הצליח לאחד את 2 אלה חזרה למקודד ערוץ.

אז במקום לשים את הקוד לתיקון שגיאות לפני/ אחרי המודולטור, שמו בפנים, כדי לא להגדיל את מספר הביטים שצריך לצורך המציאה והתיקון.

משמע, עושים את התיקונים על הסיגנל, לפני שעוברים את הסף מקבלים את ההחלטה.

פילטרים בטלפון

בטלפון במקור מעבירים רק 4 קילו הרץ, ואז מעבירים אותו ב LOW PASS FILTER, ורק כשזה יוצא, ומעובד למידע ספרתי ונשלח ל PSP (פעם זה היה המרכזיה)

אמנם בציר הספקטרום, ניתן להעביר הרבה יותר מידע, לכן עושים זאת בחלוקה ל ADSL, HN (Home Network), VDSL, שכל 1 מהם הוא בטווח אחר של הרצים ← אין שום חפיפה ביניהם, לכן ניתן לשלוח מידע בכלום, ומושכים את המידע של כל 1 מהם בנפרד, בעזרת גבולות באינטגרלים על קיבולת הערוץ.

