דצמבר 2012

מבוא לחישוב עצבי – תרגיל אמצע

בתרגיל זה נממש אלגוריתם לאיתור מקורות קול המבוסס על עקרונות הנדסיים וביולוגיים. התכנות יבוצע באמצעות תוכנת Matlab.

ראשית אנא קיראו את המאמר המצורף לקבצי התרגיל :

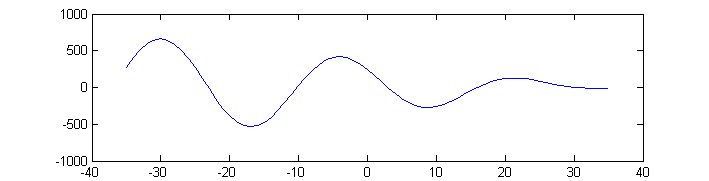
Acoustic binaural correspondence used for localization of natural acoustic signals" by [N. Arad, E. Schwartz, Z. Wollberg, Y. Yeshurun](http://www.cs.tau.ac.il/~hezy/papers/j15.pdf), **J. Neural Networks** 7:3 (1994), 441-447.

שלב א':

ראשית נממש וריאציה פשוטה של אלגוריתם המזהה כיוונים של מקורות קול בהתבסס על ההפרש בזמן הגעת הקולות לאוזניים. המקרה של מקור קול אחד (גם אם נייד) פשוט למדי וניתן לפתור אותו ע"י חלוקת האות למקטעים ומציאת ההזזה שתיתן מתאם (קרי שימוש בקרוס קורלציה) מקסימאלית בין שני ערוצי הקול באותו מקטע.

המקרה של ריבוי מקורות הינו בעייתי יותר. פתרון אפשרי לבעיה זו, המוצג במאמר, מציע חלוקה של האות לרכיבי תדר, מציאת פונקציית הקרוס-קורלציה (המנורמלת לטווח [-1,1] ) בכל רכיב תדר בנפרד ומיצוע פונקציות אלו בכל מקטע זמן.

**Cross correlation**

דוגמא לפונקצית קרוס קורלציה (cross-correlation) כזו:

הזזה אופטימאלית

אפס הזזה

נוסחת הקרוס-קורלציה (לא מנורמלת):



ניתן להשתמש בפקודה המובנית במטלב: xcorr

בשביל לקבל תוצאה מנורמלת ניתן להשתמש בדגל 'coeff' בפונקציה. אנא הסתכלו בעזרה על הפונקציה: doc xcorr.

**חלוקה לרכיבי תדר**

ניתן לבצע חלוקה לרכיבי תדר ע"י קונבולוציה עם קוסינוס בתדר אותו רוצים לבודד, לדוגמה:

samples=1:256;

filter=cos(2\*pi\*(2)\*samples./length(samples));

קוסינוס זה חוזר על עצמו כל 128 דגימות (פעמיים בתוך המקטע של ה-256 דגימות) ולכן במקרה של אות הדגום ב-44100, יחזור על עצמו בערך 345 פעמים בשניה. במילים אחרות, זהו פילטר שמשאיר סינוסים שהמחזור שלהם קרוב ל 345 Hz. בכדי לשנות את תדר המסנן נשנה את המכפלה "(2)\*" שבתוך הקוסינוס בהתאם לתדר בו אנו מעוניינים.

הערה: ככל שאורך הפילטר גדול יותר (יותר samples - אני ממליץ על איזור ה- 1024) אז הוא יותר חד (קרי תגובת התדר שלו יותר חדה והוא מעביר רק תדרים מאוד קרובים אליו.

תגובת התדר של מסנן עם תדר של 400 Hz כזה תהיה:



400 Hz

כלומר המסנן ישאיר בעיקר רכיבי תדר בסביבות 400 הרץ.

הערה: סיגנל הדגום בקצב X (במקרה שלנו 44.1 KHz) יהיה בעל תדר nyquist של X/2 = 22.05 KHz. תדירות מנורמלת הינה תדירות ביחס ל nyquist.

ולכן 400 Hz נופל ב : 400/22050 🡨 0.018

\* טיפ לשיפור הביצועים:

רוב הפילטרים ובמיוחד בפילטר הנוכחי משפיע על הפאזות של התדרים השונים בסיגנל בצורה משתנה. כך, לאחר סינון, לא רק עוצמות התדרים משתנות אלא גם הפאזות שלהם. ניתן להתגבר על עיוות זה ע"י סינון בכיוון אחד, היפוך התוצר, סינון בשנית והיפוך סופי.

פונקציות בהן אתם יכולים להיעזר כדי לבצע את הסינון:

בתיקיית starter\_code נמצאות הפונקציות:

1. makeBinauralFilters.m מייצרת מערך תאים שכל תא בו מכיל פילטר מסוים, וזאת לפי הפרמטרים שקיבלה הפונקציה. כדי לגשת לפילטר הראשון שהחזירה הפונקציה (בהנחה ששם המשתנה הוא filters) נעשה כך: filters{1}.
2. freqResponse.m מסננת מקטע אות באמצעות פילטר נתון (לדוגמה filters{3}).

**על מה עובדים?**

1. עם התרגיל תמצאו 7 קבצי קול בעלי שני מקורות ושני קבצי מקור יחיד (עימם כדאי להתחיל בפיתוח) עליהם אתם יכולים לבדוק את האלגוריתם שכתבתם. קבצי שני המקורות יוצרו מקטעים חד-ערוציים (mono) מוזזים באופן סינטטי. שימו לב שההסטה המקסימאלית לא אמורה להיות גבוהה מ 34 דגימות במצב טבעי – כפי שצוין במאמר (17 דגימות ב- 22 KHZ- קרי 34 דגימות ב 44 KHZ). מכיוון שניתן להסיט 34 דגימות קדימה ואחורה ישנן סך הכול 69 הסטות אפשריות (כולל נקודת האמצע – הסטה של 0)

**פונקציות**

**צריך לכתוב פונקציה עם header כזה:**

* function H=binaural(filename,FrameSize,OverlapSize, MaxShift)

כאשר: filename – שם קובץ ה-wav הסטריאופוני.

הערה, קבצי קול קוראים באמצעות הפקודה:

[data, fs] = wavread(filename);

כאשר fs הוא קצב הדגימה – אמור להיות 44100 Hz

* FrameSize – גודל המקטע (במילי-שניות) לצורך חלוקת האות למקטעים.
* OverlapSize- גודל החפיפה (במילי-שניות) בין זוג מקטעי אות עוקבים.
* MaxShift – טווח ההסטות הנבדק (בדגימות) – מומלץ להשתמש ב-34.
* H – מטריצה בה כל שורה היא הזזה אפשרית וכל עמודה מייצגת מקטע.

הצגה של H כתמונה אמורה לאפשר לצופה לזהות הזזות הקשורות למקורות קול.

**טיפ -** בכדי להמיר הסטה בדגימות להסטה במעלות עליכם להשתמש בנוסחה הבאה:

Angular\_Shift=arcsin(Sample\_Shift/Maximum\_Sample\_Shift)\*180/pi

כאשר השיפט המקסימלי הוא 34 דגימות, כפוי שצויין לעיל.

להלן שתי תמונות המתארות את H שהתקבל בחישוב ההסטה עבור: 1) **אחד** משבעת הקבצים 2) קובץ מקור יחיד בשם Binaural\_Flask2.wav . בציר ה -Y שלהן יש הסטה במעלות.

1) : במקרה זה ישנם שני דוברים – הראשון 17 מעלות מימין והשני 62 מעלות משמאל



2) כאן ניתן לראות מקור יחיד הנע במרחב מימין לשמאל



בכדי לציין בציר ה-Y ערכים שהם לא בעלי מרווחים קבועים ביחס לנתונים במטריצה (קרי הסטה במעלות מול הסטה בדגימות) צריך לבחור נקודות על ציר Y שבהן נרצה לשים ערך כלשהו. במקרה של התמונות לדוגמה שמתי את תוצר החישוב של ההסטה בזווית עבור הסטות מסוימות בדגימות (כל הסטה של 10 דגימות).

הקוד:

imagesc(1:size(H,2),-34:34,H)

locations=[-30,-20,-10,0,10,20,30]; %chosen sample shifts

yL=asin((1/34)\*locations)/pi\*180; %calculate angle at chosen shifts

yLC=num2cell(round(yL)); %round and transform into cell array

set(gca,'YTick', locations) %specify location of ticks

set(gca,'YTickLabel',yLC) %apply values to ticks

colormap(hot)

שלב ב':

כעת נוסיף למודל מוטיבציה ביולוגית. ב-Inferior Colliculus (IC - <http://en.wikipedia.org/wiki/Inferior_colliculus>) אשר בגזע המח ישנם נוירונים אשר מגבירים את הקצב ירי שלהם כאשר הסיגנל המגיע משתי האוזניים נמצא בהסטה האופטימלית מבחינתם (Best Delay – BD). להלן תרשים סכמטי של המתרחש בגזע המוח:



לכן, המודל שלנו יהיה כדלהלן:

עבור כל הסטה אפשרית (ישנן 69 , כפי שצוין לעיל), יש לנו נוירון Leaky Integrate-and-Fire. הקלט בכל רגע נתון של הנוירון הוא תוצר הקרוס קורלציה (המנורמלת וממוצעת מעבר לתחומי תדר) המחושבת על מקטע הזמן הנוכחי, בהסטה התואמת לו, כפול איזשהו קבוע (מכיוון שהקלט נע רק בין 1- ל 1). אם נריץ את הסימולציה על כל נקודת זמן אז נוכל לחשב את הקצב הירי הממוצע של כל אחד מן הנוירונים הללו במקטעי הזמן שתוארו לעיל ולייצר מטריצת תוצאות דומה ל H שייצרנו קודם, אך טובה יותר וגם יותר תואמת למציאות הביולוגית.

פונקציה בה אתם יכולים להיעזר כדי לממש הרחבה זו:

בתיקיית starter\_code נמצאת הפונקציה:

1. LIF\_one\_step\_multiple\_neurons.m מריצה צעד סימולציה אחד (בתדר הדגימה הנוכחי, כל צעד קטן ממילי שניה) על וקטור עמודה של נוירונים בעלי מתח מסויים (וקטור עמודה באותו גודל) ומחזירה אילו נוירונים ירו בצעד זה (וקטור עמודה באותו גודל, המכיל 0 ו- 1, בלבד) ואת המתח העדכני שלהם.

**כעת יש להרחיב את הפונקציה המקורית כדי להחזיר פלט נוסף:**

function [H, S]= binaural(filename,FrameSize,OverlapSize, MaxShift)

S, מטריצה בעלת גודל זהה ל-H , בה השורות הן הסטות אפשריות והעמודות הן חלונות זמן והערכים הם קצב הירי הממוצע בחלון הזמן בהסטה המתאימה.

להלן 4 תת-תמונות המייצגות שינוי בשני פרמטרים במודל LIF, עבור אותו קובץ בעל שני מקורות שהוצג לעיל (משמאל לימין ומלמעלה למטה):

1. מטריצתS ללא רעש בקלט (params.noise\_sd=0) וללא lateral inhibition (params.lateral\_inhibition=0).

Lateral inhibition (<http://en.wikipedia.org/wiki/Lateral_inhibition>) היא תופעה נפוצה בה נוירונים שכנים (פיזית או מבחינה תפקודית) מדכאים את פעילות שכניהם כאשר הם יורים. ניתן לראות כיצד זה ממומש בפשטות בפונקציה שצוינה לעיל.

1. מטריצתS עם רעש בקלט (params.noise\_sd=3)
2. מטריצת S עם lateral inhibition (params.lateral\_inhibition=0)
3. מטריצת S עם שניהם

**הערה**: כאן ה colormap הוא jet



**הערה: אפשר ואף מומלץ לשחק עם הפרמטרים (גודל חלון, חפיפה, והפרמטרים השונים של מודל ה- LIF) ולראות אילו תוצאות זה נותן ומה מאפשר הפרדה יותר טובה של המקורות.**

**להגשה: :**

1. קבצי הקוד.
2. קבצי הקול איתם עבדתם. (כולל אלה שניתנו עבור התרגיל)
3. מסמך המקבץ הסבר נוח וממצה על השיטה שהפעלתם (דברים שהוספתם ופרמטרים שבחרתם) ותרשים זרימה פשוט המתאר את פעולת האלגוריתם. כמו כן המסמך צריך להכיל:

* עבור חלק א': תמונה של H (כפי שהוצג לעיל) עבור הקובץ Binaural\_Flask2.wav . בתמונה זו ובתמונות הבאות נא לוודא כי: בציר ה -Y המספרים הם הסטה במעלות כפי שהוסבר לעיל.
* עבור חלק ב': תמונה של S (כפי שהוצג לעיל, לפי הפרמטרים שבחרתם) עבור הקובץ Binaural\_Flask2.wav
* תמונה ייצוגית של S עבור כל אחד משבעת קבצי שני המקורות.
* טבלת מיקומי המקורות עבור שבעת קבצי שני המקורות – כפי שנובע מהתמונות המיוצרות מ S. דוגמה (ערכים פיקטיביים) לטבלה זו:

|  |  |  |
| --- | --- | --- |
| Right Speaker  (angle) | Left speaker (angle) | File name |
| 40 | -20 | Set1 |
| .. | .. | .. |
| .. | .. | .. |
| 10 | 80- | Set7 |

**בהצלחה ☺**