



מטלה 4

מבוא:

במטלה אנו נשתמש בכלים שונים המתבססים על התמרת פורייה על מנת לנתח את הגלים המתקבלים בביצוע EEG בניסוי בו הנבדקים מתבקשים להיות בעיניים סגורות/פקוחות. באופן ספציפי נתמקד בתדרי אלפא אשר לניתוחם נגיע באמצעות האנליזה המתמטית שהציע פורייה. במשימה זו נממש את התמרה זו בתוכנת matlab בשלושה אופנים שונים עליהם נפרט בהמשך. לבסוף, נאתר את ה Individual Alpha Frequency (IAF) עבור כל נבדק בהתאם לכל שיטת ניתוח.

שיטה:

השיטה הבסיסית שבה השתמשנו במטלה זו היא התמרת פורייה, כלי מתמטי המאפשר להציג פונקציות רבות כסכום של פונקציות סינוס וקוסינוס בעלי תדירויות ומשרעות שונות. התמרת פורייה מאפשרת פירוק של פונקציה לתדרים הבודדים המרכיבים אותו, וציון היחס בין העוצמות של כל אחד מהתדרים.

טעינת הקבצים:

ראשית נסקור את הנתונים שקיבלנו: קובץ zip שקיבלנו מכיל קבצים בעלי סיומת .edf, בשמות הקבצים מצוין מספר הנבדק ותנאי הניסוי (EO/EC). לאחר טעינה של הקבצים באמצעות הפונקציה edfread (על אופן הטעינה והסיווג נפרט בהמשך), מתקבלת עבור כל נבדק עבור כל תנאי מטריות קלטים. המטריצה בגודל של מספר האלקטרודות (שורות) * אורך הניסוי (בשניות * קצב הדגימה). בנוסף ישנו גם קובץ פרטים המכיל פרטים נוספים אודות המדידה אך קובץ זה אינו רלוונטי לניתוח המבוקש ולכן אנו מתעלמים ממנו.

הקוד מתחיל בשלב טעינת הקבצים. נק' ההנחה ההתחלתית שלנו היא כי קובץ הקוד הראשי נמצא בתיקיה אחת יחד עם קובץ הZIP של המידע ופונקציית edfread. כלל טעינת הקבצים והמרתם למשתנה עליו נבצע את הניתוחים בוצעו באמצעות פונקציה שכתבנו בשם 'data2cell'.

```
function [datacell, number_subjects] = data2cell(zip_file_name,channel,conditions_num)
%This function takes a zipfile with edf files and converts it to a data cell of
%the selected channel. also saves the original subject number, total number of
%subjects and number of conditions.
unzip(zip_file_name,'..\DATA_DIR\'); %unzipping the data.

%first filter: files will contain only files in 'DATA_DIR' folder and subfolders
%that end with '.edf'.
files = dir('..\DATA_DIR\**/*.edf');

%This is the allowed pattern. The first digits that shown will be stored as
```



```
%subject number, next, the letter after 'E' will be stored as eye condition.  
look_str = '(?<subject_number>\d+).+E(?<eyes>[CO])';
```

%Generally, for each file in files (that contain only the edf files), this loop
%will check and extract the details from the file name. Using edfread will read
%the file and store the relevant information.

%The genral idea of the loop - we want to be able to get any subject number and
%to sort them by index (for easier analyze next) while keeping their subject ID.
%for doing that, after making sure that the file name is according to the
%requasted pattern, we use 'subject index' that increases by one only if the
%current subject id is not the same as the last one id.

%preparing memory - 1st row is for subject id and 2/3 are for each condition.
%the length is number of files divided by number of conditions.

```
nrows = conditions_num+1;  
ncol = length(files)/conditions_num;  
datacell = cell(nrows,ncol);
```

```
subject_index = 0;  
last_subject = '0';  
for i = 1:length(files)  
    current_str = string(files(i).name);
```

```
    %checks if file name(current_str) is fit for the allowed pattern (look_str).  
    %if it's not, error msg is shown. if it is, it stores it in temp variable.  
    current_data = regexp(current_str,look_str,'names');  
    if isempty(current_data) == 1  
        error('unvalid file name: %s' ,current_str)  
    end
```

%fullfile helps us to apply special function ('edfread') on files that are
%not in the current folder. Using this function we provide the full path for
%the files. In temp_data we store the relvant data matrix (not by channel
%yet).

```
current_file = fullfile(files(i).folder,files(i).name);  
[~, temp_data] = edfread(current_file);
```

%if the subject id is diffrent from the last id ('last subject') the
%'subject_index' increase by 1 and insert the id subject to the first row.

```
if strcmp(last_subject,current_data.subject_number) == 0  
    subject_index = subject_index+1;  
  
    datacell{1,subject_index} = char(current_data.subject_number);  
end
```

%Eyes Close/Open data from channel will be stored in row 2/3.

```
if current_data.eyes == "C"  
    datacell{2,subject_index} = temp_data(channel,:);  
  
elseif current_data.eyes == "O"  
    datacell{3,subject_index} = temp_data(channel,:);
```

```
end
```



```
%saves last subject's id for comparing in the next file.
last_subject = current_data.subject_number;

end

%setting number of subjects & conditions for later.
number_subjects = length(datacell1);

end
```

באופן כללי, מכניסים לפונקציה את שם קובץ הZIP, את ערוץ הניתוח המבוקש ('channel') ואת מספר התנאים בניסוי כמספר (במקרה שלנו 2). כפלט, מחזירה הפונקציה Cell Array המכיל את כלל המידע המבוקש מקובץ הZIP : בשורה הראשונה מופיעים מספרי הנבדקים כפי שמופיע בקובץ ועבור כל נבדק כלל תנאי הניסוי שלו (בשורות 2 ו-3). נציין גם כי הפונקציה כרגע יכולה להכיל שני תנאי ניסוי ופועלת רק על קבצים בעלי שם התואם את התבנית שתוארה בהוראות המטלה (בהתאם להוראות בעבודה). במידה ושם הקובץ אינו תואם את התבנית המוגדרת, יוצאת הודעת שגיאה מסודרת. נציין גם כי בנינו פונקציה זו כך שתוכל לקלוט כל מספר נבדק ללא תלות בגודלו (הכוונה : לא רק נבדקים [1 2 3] אלא גם [2 4 76] ולנתחם בהתאם).

בתוך הפונקציה, השתמשנו בפונקציית unzip על מנת לחלץ את הקבצים לתיקיה המבוקשת ('DATA_DIR') הנוצרת באמצעות פונקציה זו ברמה אחת מעל התיקיה הנוכחית של הקוד. לאחר שהקבצים מחולצים לשם, המשתנה files מכיל רשימה הנוצרת באמצעות הפונקציה dir שמכילה את כלל הקבצים המסתיימים בסיומת edf (סינון ראשוני). בהמשך, אנו מגדירים את הזיכרון לטובת שמירת המידע כפונקציה של מספר הקבצים המסתיימים ב edf ומספר התנאים בניסוי.

במסגרת הלולאה המרכזית בפונקציה זו, עבור כל אחד מהקבצים, הלולאה בודקת האם שם הקובץ עומד בתבנית הרצויה ('look_str') ומוציאה למבנה זמני את הפרטים הרלוונטיים. במידה והשם אינו תואם מונפקת הודעת שגיאה בצירוף שם הקובץ הבעייתי.

תחת הנחת העבודה כי הקבצים files מסודרים בסדר א"ב ואורדינלי עולה, אנו מכניסים את המידע הרלוונטי מהקבצים לתאים המתאימים. על מנת לאפשר טעינת קבצים בעלי מספרי נבדקים שונים, יצרנו אינדקס רץ בתוך הלולאה אשר מתקדם רק במידה ומדובר במספר נבדק חדש.

נציין גם כי מעבר ל cell array מסודר של הנתונים, מתקבל כפלט נוסף בפונקציה זו מספר הנבדקים בניסוי, מספר זה ישמש אותנו בעתיד להמשך הניתוחים.



הלולאה המרכזית:

לאחר טעינת הקבצים וביצוע ההגדרות הדרושות, יצרנו לולאת for מרכזית שתרוץ מ-1 עד מספר הנבדקים (number_subjects). בתוכה יצרנו לולאת for נוספת שתרוץ מ-1 ועד מספר התנאים (conditions_num). למעשה שתי הלולאות הנ"ל יוצרות את המבנה הבסיסי של הקוד שלנו. לאחר מכן יוגדר וקטור הנתונים שעליו יתבצעו הניתוחים (data_vec) על פי שליפה מהמיקום הרלוונטי במטריצת הנתונים data שנוצרה בשלב טעינת הקבצים. בשלב זה תחל סדרת הניתוחים.

```
%The main loop of the work.
%we run the entire analyze for each subject in each condition.
%we mainly use temp variables, but in order to calculate the IAF later, we save
%the relevant information for the relevant time (more details inside the loop).
%The functions FFT & DFT are made by us, further explanation inside the functions.

for subject_index = 1:number_subjects

%   each subject gets a new window with title. we wanted a full screen figure
%so it be large enough to check all 6 plots carefully.
figure('Units','normalized','Position',[0 0 1 1]);
hold on
sgtitle(['Subject Number - ',data{1,subject_index}])

for eyes_condition = 1:conditions_num

%   sets the relevant vector from data and sets the color.
data_vec = data{eyes_condition+1,subject_index};
current_color = EC_EO_colors{eyes_condition};
```

הניתוח הראשון שיתבצע הוא- (Fast Fourier Transform) fft. מדובר באלגוריתם יעיל לביצוע (Discrete Fourier Transform) DFT. על מנת להפעיל ניתוח זה יצרנו את הפונקציה FFT.

פונקציית FFT:

```
function [freq_x,ps_y] = FFT(data,fs)
%FFT takes data as vector (from one specific channel) and calculate the power
%spectra using fft function.
%the output is x & y vectors for plotting the results.

dt = 1/fs; % time step [sec]

%time settings
dur = length(data)/fs; % duration of signal [sec]
t = 0:dt:dur; % times vector [sec]
N = length(t); % number of time components
freq_x = (0:round((N-1)/2))/(N*dt); % the first half of freq.
N1 = length(freq_x); % number of freq.
```



```
%calculating the power spectra.
ft= fft(data);
ft= ft/length(ft);
ps_y = abs(ft).^2;

ps_y = ps_y(1:N1);
end

%fft the input.
%normalizing the fft by it length.
%calculat the power spectra.

%taking only the first half.
```

פונקציה זו מקבלת כקלט את וקטור הנתונים הרלוונטי אותו יש לנתח (data_vec), ואת תדר הדגימה (fs). תחילה, הפונקציה מחשבת באמצעות תדר הדגימה מספר משתנים הכרחיים להמשך ניתוח והצגת התוצאות. המשתנה freq_x יכול את וקטור התדרים שיהווה את ציר ה-x בהצגת ה-power spectrum. ניתן לראות שוקטור זה באורך (N/2) כמספר שלם, כלומר באורך כחצי מוקטור הזמן t, וזאת על מנת לקצץ ממנו את החצי הלא רלוונטי. לאחר ביצוע ההגדרות הנ"ל, תופעל הפונקציה fft על הוקטור data. פונקציה זו מבצעת גרסה מהירה של טרנספורמציה פורייה. לאחר הפעלת הפונקציה, יתבצע נירמול של הוקטור שהתקבל (ft) על ידי חלוקה שלו באורכו. לאחר מכן נפעיל על הוקטור המנורמל ערך מוחלט וכמו כן נעלה אותו בריבוע. מהוקטור הסופי (ps_y), נשמור רק חצי על מנת לא להציג מידע מיותר (שכן מדובר בוקטור סימטרי). הפונקציה תפלוט את הוקטורים freq_x ו-ps_y. הוקטורים הללו יאפשרו את הצגת ה-power spectrum.

חישוב IAF וחיסור power spectra :

```
%IAF
% EC is the first condition. the vector y is stored until the next
% condition.
if eyes_condition == 1
    fft_EC_y = y;

    %when getting to the next condition (EO), the IAF is calculated with the
    %previous y conditions that was stored (fft_EC_y) - y.
elseif eyes_condition == 2

    dif_spec = fft_EC_y-y;
    IAF_max = find(dif_spec == max(dif_spec));

    fft_IAF_subplot = subplot(2,3,4);
    plot(x,dif_spec,EC_EO_graph)
    xline(x(IAF_max), IAF_line,'Linewidth', 1.5);
    ylabel('Power');
    xlabel('Frequency [Hz]');

    title(['EC-EO : IAF = ', num2str(x(IAF_max)), ' Hz']);
end
```



על מנת לבצע את הניתוחים יצרנו תנאי if המסווג לפי סוג תנאי הניסוי. במידה והמשתנה eyes_condition (כאמור משתנה זה הוגדר בלולאה שבתחילת הקוד ורץ מ-1 ועד מספר התנאים) שווה ל-1, ייווצר הוקטור fft_EC_y שישמור את הנתונים שהיתקבלו בשלב הfft. מידע זה יהיה חיוני לביצוע הניתוחים הדרושים בהמשך (בשלב זה לא ניתן לבצע אותם עדין שכן לא היתקבלו נתוני הfft בשני התנאים). שמירת המידע תחת המשתנה fft_EC_y תמנע את דריסתו עד לשלב בו יתבצעו הניתוחים.

במידה והמשתנה eyes_condition שווה ל-2, ומכיוון שיש בשלב זה את כל הנתונים הדרושים לביצוע הניתוח (נתונים שהיתקבלו בניתוח fft לאותו אדם בשני התנאים), יתבצעו הדברים הבאים: חיסור של המשתנה y (נתוני הfft בתנאי העיניים הפקוחות) מהמשתנה fft_EC_y (כאמור משתנה זה נשמר מקודם ומכיל את נתוני הfft בתנאי העיניים הסגורות). המשתנה שהתקבל יקרא dif_spec. לאחר מכן ישמר האינדקס של נקודת המקסימום בוקטור ה-dif_spec בשם IAF_point. הנתונים הנ"ל יאפשרו את הפקתו של גרף ההפרשים בין ה-power spectrums של התנאים ובנוסף את מדד ה-IAF.

ניתוח זה יתבצע בהמשך באופן זהה לאחר ניתוחי pwelch וה-dft. השוני היחיד הוא שם המשתנה שבו נשמר וקטור הנתונים כאשר 1 == eyes_condition. זהו שוני שולי ולכן לא נסביר את הניתוח הזה שוב בהמשך.

Pwelch ויצירת גרפים:

```
pwelch_subplot = subplot(2,3,2);  
hold on  
[y,x] = pwelch(data_vec,window_size,overlap,f,fs);  
plot(x,y,current_color)  
title('Pwelch')
```

פונקציית pwelch מקבלת 5 קלטות: א. וקטור הקלט עליו היא תבצע את הניתוח הדרוש. במקרה זה מדובר בוקטור data_vec. ב. גודל החלון - פונקציית pwelch מחלקת את וקטור הקלט לחלונות (שיטה זו מאפשרת הפחתה של הרעש הרב שמתקבל בניתוח שאינו כולל את שיטת החלונות). את גודל החלון (window_size) קבענו כ-5*fs (זאת בכדי להיות מסוגלים להסתכל על גודל החלון כפונקציה של זמן). ג. גודל החפיפה בין חלונות (overlap) – הגדרנו כפונקציה של גודל החלון ודרשנו חפיפה באורך מחצית כגודל החלון. ד. וקטור תדרים - אנו מעוניינים בטווח תדרים אלפא, ולכן וקטור התדרים הוא 0.1:14:6. f. ה. תדר הדגימה - במקרה שלנו fs=256. הפונקציה פולטת את הוקטורים y ו-x, שמאפשרים את הצגת ה-power spectrum.

בחלק קוד זה ניתן גם להדגים את אופן יצירת הגרפים בעבודה (אשר חוזרת על עצמה עבור כלל השיטות ולכן נפרט רק כאן). כל נבדק מופיע בחלונות חדשה, כל חלונות מכילה 6 מערכות צירים (2 שורות של שלוש מערכות בכל שורה). השורה הראשונה מכילה את מערכות הצירים עבור כלל שיטות הניתוח בהתאם לסדר הופעתן בקוד (fft, pwelch & DFT), בכל מערכת צירים שני גרפים (אחד עבור כל תנאי ניסוי).



בשורה השנייה ניתן להבחין בחיסור התנאים על פני הגרף עבור כל שיטה (כל חיסור מופיע מתחת לשיטה הרלוונטית) ובכל גרף מסומנת ומוצגת גם נק' המקסימום של חיסור זה (IAF).

```
% linking all x axes in order to scale them into the alpha band.
```

```
linkaxes([fft_subplot fft_IAF_subplot pwelch_subplot IAF_pwelch_subplot...  
DFT_subplot IAF_DFT_subplot], 'x');  
fft_subplot.XLim = alpha_band;
```

בסיום יצירת החלונות, אנו מנחים את הקוד להתמקד בטווח של תדרי אלפא עבור כלל הניתוחים.

:DFT

```
%w matrix  
%first we prepare the constant 'w' expression. then, using meshgrid, we prepare 2  
square matrix  
%in the size of 'window_size': each one starts from 0 and running by one until  
%getting to 'window_size-1'. they do it in different direction (by column/row).  
%next, in w, we first dot multiply I & J in order to get the right multiplication for  
each  
%location in w. last, using power rules, we make sure that the power of every  
%element will fit the formula.  
w = exp(-2*pi*1i/window_size);  
[I, J] = meshgrid (0:(window_size-1), 0:(window_size-1));  
w = (w.^(I).*(J));  
  
%using buffer function, we split the long data vector into windows at the size  
%of 'window_size' and with overlapping at the size of 'overlap'. 'nodelay' is  
%used for zero padding at the end.  
%each row in the matrix represents a window.  
input_mat = buffer(data, window_size, overlap, 'nodelay');  
  
%The DFT - according to the formula, we do matrix multiply and calculate the  
%DFT for each window. the output is a matrix of complex numbers in the same size  
%of the original input_mat.  
output_mat = w * input_mat;
```

הפונקציה מקבלת כקלט 4 פריטים: וקטור הקלט עליו היא תבצע את הטרנספורמציה (data_vec), גודל החלון (window_size), גודל החפיפה בין חלונות (overlap) ותדר הדגימה (fs). מכיוון שוקטור הקלט ארוך, חישוב הטרנספורמציה לכולו בפעם אחת מוציא הודעת שגיאה, ולכן ביצענו את הטרנספורמציה על ידי שימוש בחלונות.

תחילה, הפונקציה יוצרת את מטריצת הנוסחה. w הוא האיבר הקבוע בכל אחד מהאינדקסים במטריצה. באמצעות הפונקציה meshgrid יצרנו שתי מטריצות: מטריצה I היא מטריצה ריבועית בגודל window_size שאיבריה מייצגים את אינדקסי השורות (מס עד גודל החלון) באופן תואם למיקומם. מטריצה J מכילה איברים בעלי אינדקסים תואמים לעמודות (באופן דומה). העלאה



בחזקה של האיבר הקבוע w בתוצאת ההכפלה של המטריצות $(J) \cdot (I)$. מאפשר מימוש את מטריצת הנוסחה W .

על מנת לממש את נוסחאת ה-DFT על ידי שימוש בחלונות היה צורך לארגן את וקטור הקלט באופן המתאים. לשם כך הישתמשנו בפונקציה `buffer`. פונקציה זו מקבלת כקלט את וקטור הקלט הרלוונטי, גודל החלון הרצוי וגודל החפיפה בין החלונות. הפונקציה פולטת מטריצה שבה כל עמודה היא חלון על פי הגדרות הקלט (`input_mat`). בנוסף הפונקציה מרפדת את העמודה האחרונה באפסים (שימוש בפקודה `'nodelay'`). בשלב זה ניתן לממש את נוסחאת ה-DFT. לאחר פעולת כפל המטריצות קיבלנו את המטריצה `output_mat`. למטריצה זו בוצעו טרנספורמציות הדרושות, וכמו כן מיצוע מעבר לחלונות (`dft_y`). לאחר מכן נשמור רק חצי מהוקטור שהיתקבל על מנת לוותר על מידע מיותר. הוקטור `freq_x` יכול את וקטור התדרים שיהווה את ציר ה- x בהצגת ה-power spectrum. הוא נוצר באופן תואם ליחידת הזמן של הדגימה ולאורכו של חלון. פלט הפונקציה מורכב משני איברים: `dft_y` ו-`freq_x`. איברים אלו מאפשרים תצוגה של התוצאות בגרף.

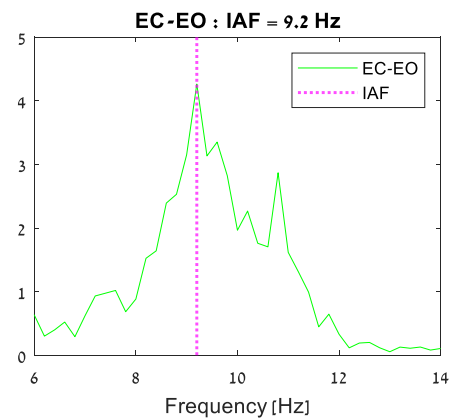
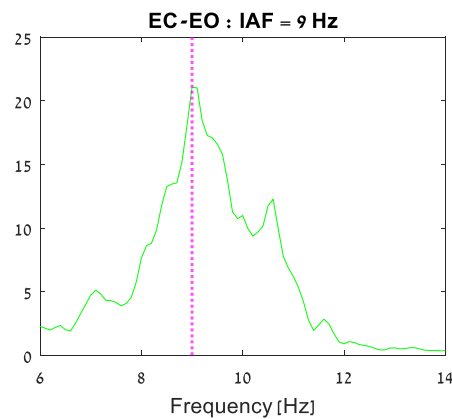
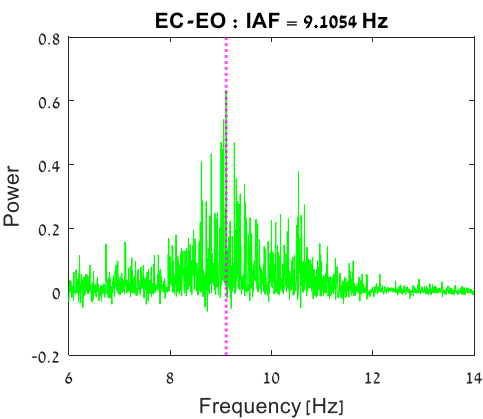
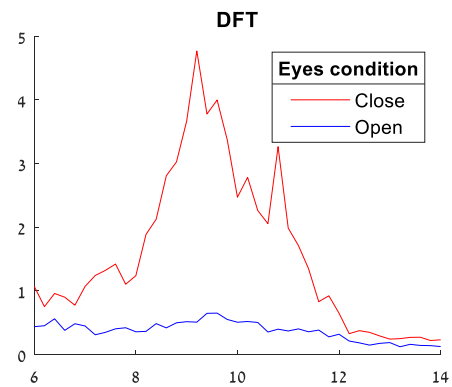
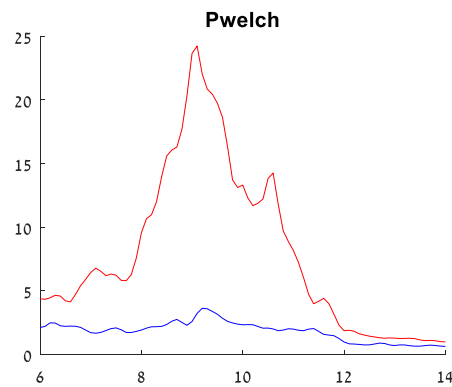
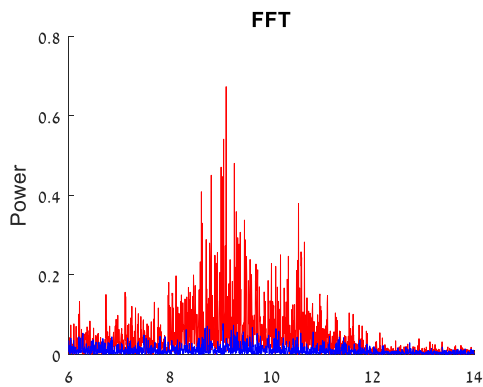
הפעולות הנ"ל חוזרות על עצמן על ידי שימוש בלולאות שהוזכרו בתחילת הקוד. הפעולות יתבצעו עבור כל אחד מהתנאים של נבדק אחד. לאחר מכן הקוד יעבור לנבדק הבא וחוזר חלילה.



תוצאות:

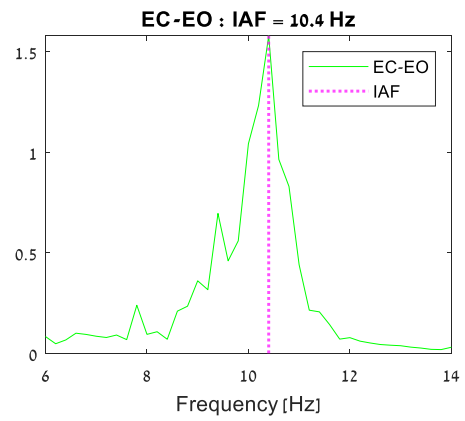
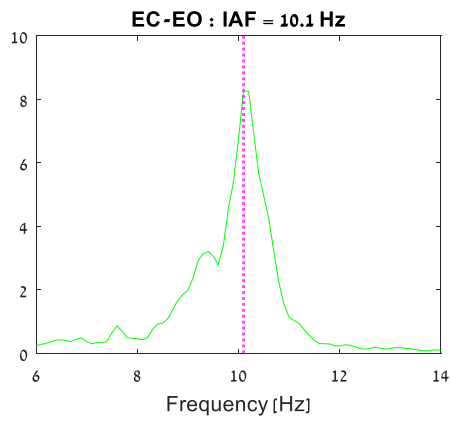
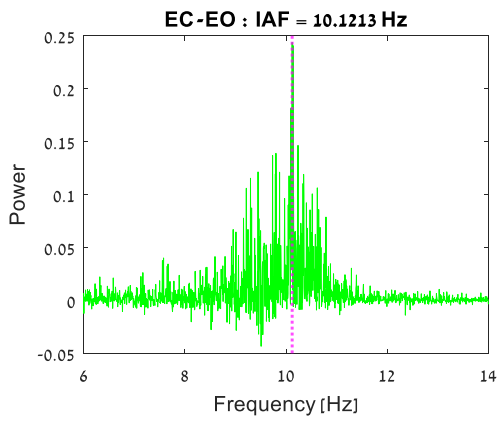
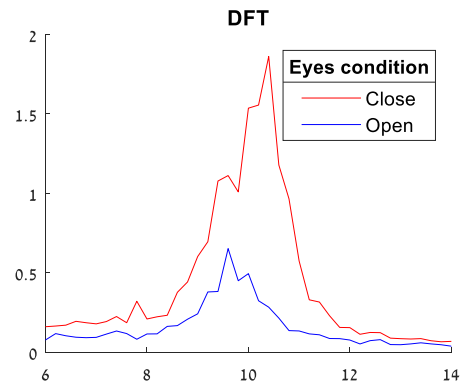
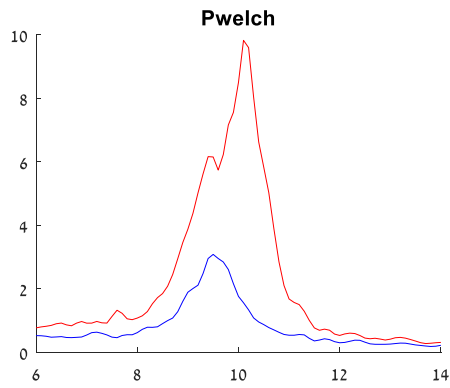
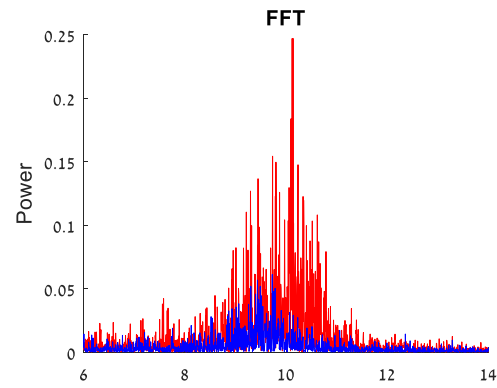
לכל אחד מהנבדקים הונפקו בסך הכל 6 מערכות צירים שונות. על כל אחת ממערכות הצירים מופיעים שני גרפים. בכל אחת ממערכות הצירים ציר ה-x מייצג את טווח התדרים. החלטנו להציג את טווח התדרים 6-14 Hz מכיוון שבמטלה זו עיקר ענייננו הוא בגלי האלפא של הנבדק. בכל אחד מהגרפים ציר ה-y מייצג את הממד power.

Subject Number - 1

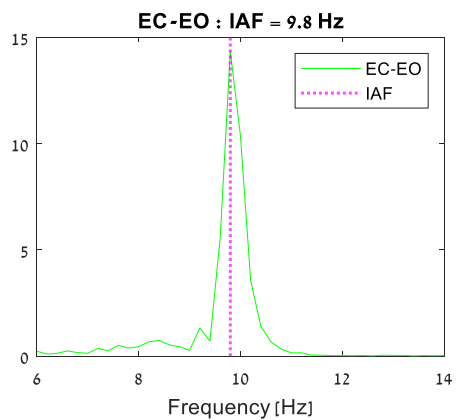
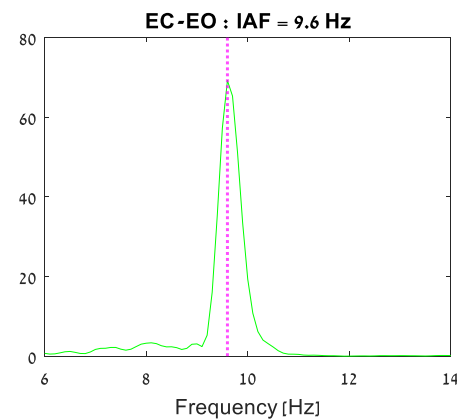
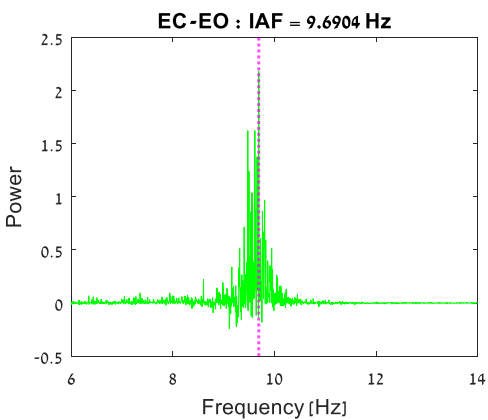
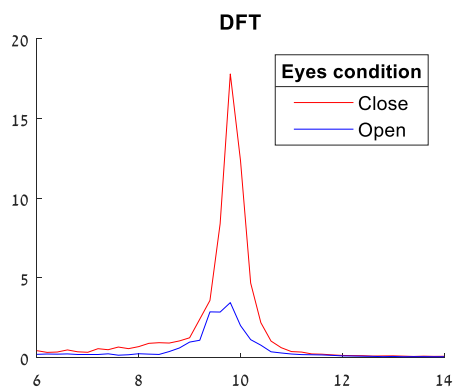
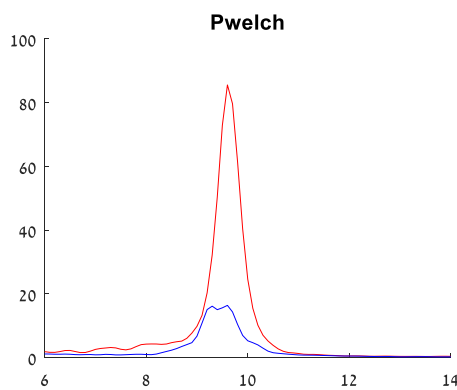
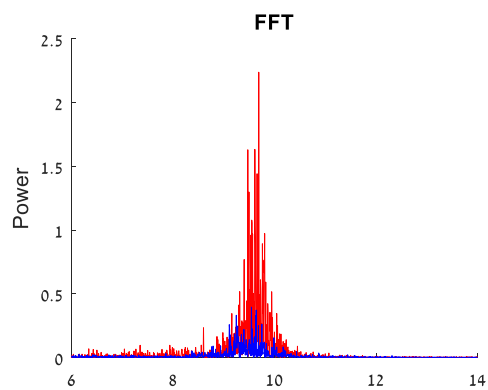




Subject Number - 2



Subject Number - 3





בשורה העליונה עבור כל אחד מהנבדקים ניתן לראות את תוצאותיהם של שלושת הניתוחים שבוצעו. הגרף הכחול מייצג את תנאי העיניים הפקוחות וגרף אדום מייצג את תנאי העיניים הסגורות. הצגה זו מאפשרת התבוננות נוחה והשוואה פשוטה בין השפעת התנאים על ה power spectrum של הנבדק. השורה העליונה מכילה : משמאל ניתוח fft. מדובר באלגוריתם יעיל לביצוע ניתוח DFT. במרכז השורה הראשונה ניתן לראות את ניתוח ה-pwelch. בשורה העליונה מימין ניתן לראות את תוצאות הניתוח באמצעות נוסחאת ה-DFT.

במטלה זו מטרטנו היא לחלץ את מדד IAF של כל נבדק, כלומר את תדר גלי האלפא המרכזי בפעילותו הנוירולית. מדד זה משתנה בין אנשים ונמצא כבעל קשר ליכולות קוגניטיביות שונות. לשם כך ביצענו ניתוח בו חישבנו את ה difference spectrum של כל אחד מהנבדקים בכל אחד מהניתוחים : לכל אחד מהניתוחים בשורה העליונה ביצענו חיסור של התוצאות בתנאי העיניים הפקוחות מהתוצאות בעיניים הסגורות. שלושת מערכות הצירים בשורה התחתונה מציגות את התוצאות. בירוק ניתן לראות את גרף ה difference spectrum שהתקבל. בצבע ורוד ניתן לראות קו אנכי לציר ה-X שמייצג את מדד ה-IAF, כלומר את ה- peak frequency של הנבדק. מעל כל אחת ממערכות הצירים בשורה התחתונה מצויין ה- IAF שהתקבל בניתוח. תצוגה זו מאפשרת השוואה בין המדדים שהתקבלו בכל אחת משיטות הניתוח של אותו נבדק.

בנוסף, החלטנו לבדוק את ממוצע ה-IAF מעבר לנבדקים ומעבר לשיטות על מנת לנסות ולהבחין בהבדלים אינדיבידואלים בין נבדקים ובהטיות של שיטות החישוב השונות.

טבלת מדדי ה IAF של כל נבדק בכל שיטה (כל המדדים ביחידות Hz).

נבדק/שיטה	FFT	Pwelch	נוסחת DFT	ממוצע נבדקים
1	9.1	9	9.2	9.1
2	10.12	10.1	10.4	10.2
3	9.69	9.6	9.8	9.7
ממוצע שיטות	9.63	9.56	9.8	

(טבלה 1)



דיון ומסקנות:

המידע שהתקבל במסגרת ניסוי EEG מכיל את כלל ספקטרום התדרים אשר יכול מכשיר זה לקלוט. פלט של תדר הגל שהתקבל באלקטרודה הנתונה אינו מאפשר הסקת מסקנות אמיתית לגבי אופי הפעילות המוחית מאחר ומכיל שילוב של כלל התדרים הנקלטים יחדו על גרף אחד. ספציפית במטלה זו, התעניינו בטווח תדרי האלפא והקשר ביניהם לבין עצימת עיניים. על מנת להפריד את טווח תדרי האלפא ולבודד את עוצמת התדר עבור הגלים בטווח זה, מתבקש להשתמש באנליזת פורייה. באמצעות כלי מתמטי זה, ניתן לפרק את הגל המקורי לסכום התדרים המרכיבים אותו. לאחר "הפירוק", קל יותר להתמקד בטווח בו אנו מתעניינים ולחקור אותו לעומקו על אף שלא נמדד בצורה אינדיבידואלית בניסוי.

במטלה הנוכחית השתמשנו בשלושה מימושים שונים של DFT, על מנת לבחון את ה-power spectrum של פעילות המוחית בטווח גלי האלפא של שלושה נבדקים כפי שנמדדה באלקטרודה מספר 19 (electrode Pz), כתלות במצב העיניים שלהם.

תחילה, ניתן לבחון את מאפייני הגרף שיוצר על ידי כל אחת משיטות הניתוח. כפי שניתן לראות בשלב התוצאות, ניתוח הfft מייצר גרף שמציג מידע רב, וחלק גדול ממנו מכיל רעש שאינו נחוץ על מנת להבין את מגמתם של פעילות גלי האלפא (ומדדים כמו peak frequency אליו נתייחס בהמשך). לעומתו, הניתוח שמתבסס על פונקציית Pwelch מייצר גרפים שאינם מכילים רעש ומציגים את מגמתו הכללית של ה-power spectrum. הצגה זו מתאפשרת באמצעות חלוקה של הקלט לחלונות. תחת ההנחה כי הקלטים שנמצאים בכל אחד מהחלונות מייצגים פעילות מוחית דומה, מתבצעת טרנספורמציה לכל חלון בנפרד ולאחר מכן מיצוע של החלונות כך שמתקבל וקטור פלט שאינו מושפע מרעש רב. גודל החלון משפיע על הגרף שיתקבל: ככל שהחלון גדול יותר כך הגרף יכיל יותר מידע אך גדל הסיכון שרעש ישפיע על התוצאות שיתקבלו. לעומת זאת חלון קטן יגרור לגרף לא להיות מושפע מרעש אך עלול להוביל לאיבוד מידע חשוב. הניתוח השלישי אותו ביצענו הוא באמצעות נוסחת הDFT. מכיוון שבניתוח זה חילקנו את הקלט לחלונות בגודל זהה ובעלי חפיפה זהה לניתוח Pwelch, ניתן לראות כי הגרפים שהתקבלו בשני הניתוחים דומים למדי.

לשימוש בשיטות השונות יתרונות וחסרונות שונים. בעוד חישוב באמצעות פונקציית הfft מעביר את משקף את מרבית המידע המתקבל מהאלקטרודה, הוא בסבירות גבוהה מכיל רעשים רבים ואינו רציף. שילוב זה של הרעש הגבוה והיעדר הרציפות עלול להקשות על הבנת מגמות והסקת מסקנות (נציין גם כי היעדר הרציפות נובע ככה"נ מקצב הדגימה אשר מעצם הגדרתו ברור כי אינו דוגם באופן רציף). לעומת פונקציית הfft, פונקציית הpwelch והDFT מבצעות תהליך דומה יחסית המבוסס על ניתוח חלונות מהקלט ומיצועו לאורך. על מנת להתגבר על היעדר הרציפות ובכדי לייצר תוצאה כמה שפחות מוטית, משתמשות פונקציות אלה בחפיפה בין החלונות (אנו הגדרנו את גודל החפיפה כמקובל - בגודל מחצית החלון).



שנית, ניתן לראות כי פעילות גלי האלפא של כל הנבדקים בכל הניתוחים גבוהה באופן משמעותי כאשר העיניים שלהם עצומות מאשר כשעיניהם פתוחות. ממצא זה צפוי מפני שידוע כי גלי אלפא באים לידי ביטוי במצב מנוחה ערני עם עיניים עצומות ופוחתים כאשר העיניים פקוחות.

כידוע, גלי אלפא מתייחסים לפעילות חשמלית מוחית בטווח התדרים 8-12 Hz. במטלה זו התבקשנו לחלץ את מדד ה-IAF, שהוא התדר הדומיננטי ביותר בפעילות גלי האלפא של הנבדק. נמצא כי מדד זה משתנה בין אדם לאדם ונמצא במתאם עם יכולות קוגניטיביות שונות. ממצאי הניתוח שלנו מאששים את הטענה כי מדד ה-IAF הוא אינדיקטור. ניתן לראות (טבלה 1) כי יש שוני משמעותי במדד ה-IAF בין שלושת הנבדקים. ניתן להתבונן בתוצאות של כל אחד מהניתוחים ולראות כי מדד ה-IAF של נבדק מספר 2 הוא הגבוה ביותר ושל נבדק מספר אחד הוא האיטי ביותר. על מנת לקבל מדד מהימן יותר חישבנו את ממוצע ה-IAF שמתבסס על 3 הניתוחים השונים לכל נבדק, וגם בהם אותה מגמה נשמרה.

נציין גם כי בהסתכלות (לא באופן סטטיסטי) על ממוצע ה-IAF מעבר לנבדקים (טבלה 1), ניתן להבחין כי לא קיימים הבדלים מהותיים בין הממוצעים עבור שיטות הניתוח השונות ונתון זה מעיד ככה"נ על מהימנות פנימית גבוהה עבור שיטות ניתוח אלו.

ניסויי המשך מעניינים שניתן לבצע הם חישוב מדדים של יכולות קוגניטיביות שונות של אותם נבדקים, לדוגמא זיכרון עובד ואינטליגנציה. לאחר מכן יהיה ניתן לבחון את המתאם של מדדי ה-IAF של הנבדקים עם תוצאותיהם במבחנים השונים, ולראות האם נקבל אישוש לממצאים קודמים בתחום שמעידים על מתאם חיובי בין המשתנים הנ"ל.