



מטלה 6

מבוא:

במטלה זו יצרנו מסווג המפריד בין מחשבה על הרמת יד ימין ויד שמאל וזאת באמצעות הקלטות EEG (Electroencephalogram) מאלקטרודות C3 ו-C4. עשינו זאת באמצעות תוכנת Matlab תוך בחינה מעמיקה של ההבדלים באותות בין התיוגים השונים ובין האלקטרודות השונות. בהמשך, באמצעות שיטות סטטיסטיות, בחנו ואימנו את המסווג תחת תנאים שונים ולבסוף בחנו את רמת ההצלחה שלו בדוגמאות לא מוכרות והצענו כיווני מחקר נוספים.

שיטה:

על מנת לבצע את המטלה השתמשנו בקובץ הנתונים motor_imagery_train_data.mat המכיל מבנה struct בשם P_C_S. מבנה זה מכיל בין היתר נתונים שהוקלטו על ידי 2 אלקטרודות EEG, אלקטרודות C3 ו-C4. מדובר בהקלטה באורך 6 שניות (trail_time), עם תדר דגימה (fs) של 128Hz. ההקלטה התבצעה 128 פעמים (n_trails), כאשר הנבדק דמיין בחציים כי הוא מזיז את ידו הימנית ובחציים שהוא מזיז את ידו השמאלית. מתוך מבנה זה, שני המבנים שהיוו חלק מרכזי בהמשך הניתוח הינם: data, המכיל את הקלטות ה-EEG. מדובר במבנה תלת מימדי, כאשר מימדיו הינם: 128 (n_trails), על 768 (t_samples, מספר הדגימות), על 3 (מספר הערוצים שהוקלטו). בנוסף, P_C_S.attribute מכיל מטריצה בגודל 4 על 128, הכוללת את התיוגים המתאימים לכל trail. בשורה השלישית מסומנים ב-1 מקצים בהם הנבדק דמיין את ידו השמאלית וזוהי, ובשורה ה-4 מסומנים מקצים בהם הנבדק דמיין את ידו הימנית וזוהי.

סידור והנגשת המידע:

בשלב הראשון, על מנת לאפשר גישה נוחה לנתונים בהמשך באופן שיאפשר קוד מסודר ויעיל ככל האפשר, יצרנו את הפונקציה data_proc. פונקציה זו מקבלת כקלט את מבנה הנתונים P_C_S.

```
function [electrode_cell, labels_idx, fs, t_samples, n_trails, names] =  
data_proc(data_struct)  
%this function takes the raw data from the EEG and converts it to cells that  
%will be more easily used in the analyse.  
  
%taking only the first two relveant electrodes (C3 & C4)  
C3 = data_struct.data(:, :, 1);  
C4 = data_struct.data(:, :, 2);  
  
%extracting the labels for all trails  
labels = data_struct.attribute;  
  
left = find(labels (3, :) == 1);  
right = find(labels (4, :) == 1);  
  
%sorting the data together in cells.  
electrode_cell = {C3 C4};  
labels_idx = {left right};  
names = {'C3' 'C4'; 'Left', 'Right'};  
  
%extra information  
fs = data_struct.samplingfrequency;  
t_samples = length(C3);  
n_trails = size(data_struct.data, 1);  
end
```



ראשית, הפונקציה מפרידה בין נתונים שנאספו על ידי אלקטרודה C3 ונתונים שנאספו על ידי אלקטרודה C4. לאחר שבוצעה ההפרדה, נתוני כל אלקטרודה נשמרים במשתנה `electrode_cell`. משתנה זה הינו מבנה מסוג `cell array`, על מנת שתתאפשר גישה פשוטה לנתונים בהמשך הקוד. באופן דומה, הפונקציה מוציאה את תיוג החזרות (ימין או שמאל) ושומרת אותם תחת המשתנה `labels`. בשלב הבא, באמצעות הפונקציה `find`, הפונקציה מוציאה מתוך המשתנה `labels` המיקומים של החזרות בהן הנבדק דמיין כי הוא מזיז את ידו הימנית/שמאלית, ושומרת אותם תחת המשתנים `left` ו-`right` בהתאמה. את הנתונים הנ"ל הפונקציה שומרת במשתנה `labels_idx`. משתנה זה גם הוא הינו `cell array`. לאחר הגדרת מבנה המידע ומבנה התיוגים, מוגדרים השמות של האלקטרודות והתיוגים תחת 'names' וזאת על מנת לאפשר שימוש נוח בבניית הגרפים והצירים בהמשך הקוד. בסיום, הפונקציה מגדירה את תדר הדגימה (`fs`), את מספר דגימות הזמן בכל חזרה (`t_samples`) ואת מספר החזרות שנעשו (`n_trails`) על ידי שליפתם ממבנה הנתונים המקורי.

בשלב הבא, יצרנו את הפונקציה `time_calc`. פונקציה זו מקבלת כקלט את מספר דגימות הזמן שנדגמות בכל חזרה (המשתנה `t_samples`) ואת תדר הדגימה (`fs`). הפונקציה פולטת את הזמן הכולל של כל חזרה בשניות (`trail_time`), ואת וקטור הזמנים `t_vec`, אשר מציג את הזמן המתאים בשניות עבור כל דגימת זמן. משתנים אלו ישמשו אותנו בהמשך.

```
function [t_vec, trail_time] = time_calc(t_samples,fs)
%this function calculates different time parameters using the time samples vector
%and the frequency sample.

trail_time = t_samples/fs;
t_vec = [0:(trail_time/(t_samples-1)):trail_time];

end
```

visualize the EEG signal

על מנת לאפשר הצגה של קלט ה-EEG יצרנו את הפונקציות `rand_trails` ו-`plot_EEG`. הפונקציות הנ"ל מנפיקות שני חלונות עם גרפים. כל חלון מכיל הצגה ויזואלית של 20 חזרות אקראיות בתנאי מסוים (תנאי ימין/תנאי שמאל). כלומר, כל חלון מכיל 20 מערכות צירים, וכל מערכת צירים מציגה את קלט ה-EEG כפי שנקלט בשתי האלקטרודות (C3, C4) באחת מהחזרות.

```
function [memory] = rand_trails(electrode_cell,labels_idx,n_rand)
%this function takes the data from all electrodes, the index of each label
%and the number of wanted random trails and produce one cell with all the data.
%each row stands for a different electrode and the columns for each label.

%preparing memory
memory = cell(length(electrode_cell),length(labels_idx));

%the loop takes the data from both electrodes from 20 random trails from each
%label and sort it in memory.
for label = 1:length(labels_idx)
    n_label_trails = length(labels_idx{label}); %number of trails per side
    idx_rand = randperm(n_label_trails,n_rand); %20 random trails
    for elec = 1:length(electrode_cell)

        %this method repeats in the code- we want to extract the trails only from one
        %label from one electrode - so we use the index of the labels and take with
        them

        %only the relevant trails from the selected electrode.
        trails_by_label = electrode_cell{elec}(labels_idx{label},:);
        memory{elec,label} = trails_by_label(idx_rand,:);
    end
end
```



end

הפונקציה rand_trails מקבלת כקלט את המשתנים electrode_cell ו-labels_idx שנוצרו בפונקציה data_proc. כמו כן, הפונקציה מקבלת את המשתנה n_rand, שקובע את מספר החזרות האקראיות הנדרשות ובהתאם לכך את כמות מערכות הצירים שיוצגו בכל חלון (מוגדר כ-20). תחילה, הפונקציה יוצרת את המשתנה memory. משתנה זה הינו cell array, ובו ישמרו הנתונים שימשו ליצירת הגרפים בהמשך. לאחר מכן, תתבצע הגרלה של n_rand מספרים, מתוך הטווח שבין 1 ל n_label_trails (מוגדר כ-64, מספר הtrails בכל תנאי). idx_rand מכיל את האינדקסים הנבחרים מכל צד. בשלב זה, נעשה שימוש במשתנה electrode_cell. הפונקציה ניגשת ל-cell המתאים (כאמור, cell מכיל נתונים מ-128 trails ומאלקטרודה 1), ומתוכו מוציאה נתונים של trails שהתקיימו בתנאי המתאים (ימין/שמאל). מתוך הtrails הללו, הפונקציה בוחרת את נתונים של 20 ה-trails שממוקמים באינדקסים התואמים למשתנה idx_rand, ושומרת אותם במקום המתאים במשתנה memory. תהליך זה יחזור על עצמו עבור כל צירוף של תנאי (ימין/שמאל), ושל אלקטרודה (C3/C4). הפונקציה פולטת את המשתנה memory. משתנה זה הינו cell array בעל 4 תאים. האופן בו המידע שמור במשתנה memory חוזר על עצמו במקומות הנדרשים גם בהמשך העבודה ולכן נסביר כאן פעם אחת – בכל מקום בו ישנו צורך לסיווג לפי אלקטרודה ותווית (צד), השורות מייצגות את מספר האלקטרודה (שורה 1 – C3, שורה 2 – C4) והעמודות מייצגות את הסיווג לפי תווית (עמודה 1 – שמאל, עמודה 2 – ימין) כך המידע מתחלק בעצם ל-4 תאים המורכבים כל אחד מ-64 חזרות.

לשם הפקת הגרפים הדרושים, יצרנו את הפונקציה plot_EEG. פונקציה זו מקבלת כקלט את המשתנה rand_cell (שמו של המשתנה memory בקוד הראשי), ובנוסף את המשתנים t_vec, ו-names שדרושים לשם יצירת מערכות הצירים. באמצעות לולאת for פשוטה הפונקציה מפיקה את החלונות והגרפים הדרושים.

```
function [] = plot_EEG(rand_cell,t_vec,names)
%this function takes the cell that contains the 20 random trails from each side
%for each electrode and using the time vector and the names cell produce 2
%plots - one for each label. each plot contains the data from 20 random trails
%from both electrodes.

colors = {'r','b'};
titles = names(2,:);
for label = 1:size(rand_cell,2)
    figure
    sgtitle(titles{label})

    for elec = 1:size(rand_cell,1)
        current_data = rand_cell{elec,label};
        for trail = 1:size(current_data,1)
            subplot(4,5,trail)
            plot(t_vec,current_data(trail,:),colors{elec})
            xlabel('time [sec]')
            ylabel('voltage [microvolt]')
            ylim ([-20 20])
            hold on
        end
        legend({'C3','C4'})
    end
end
```

end



end

נשים לב כי טרם הלולאה, הפונקציה מגדירה את הצבעים לגרפים השונים ואת שמות החלונות בהתאם למפורט במשתנה names שהוגדר קודם לכן.

power spectrum

בשלב הבא, יצרנו את הפונקציה plot_power_label, המאפשרת את הצגת ה-power spectrum (הצגת עוצמת התדרים השונים המרכיבים את האות המקורי) של נתוני ה-EEG, כפי שהתקבלו בכל תנאי ובכל אלקטרודה. פונקציה זו מקבלת כקלט את המשתנים: electrode_cell, labels_idx, f, fs, names. בנוסף למשתנים הנ"ל הפונקציה מקבלת 3 משתנים חדשים שהוגדרו לטובתה: p_window, p_overlap, אשר הוגדרו לטובת השימוש בפונקציית ה-pwelch בהמשך. המשתנה img_t_frame הוגדר והוזן לפונקציה על מנת לאפשר את בחינת ה-power spectrums בזמנים הרלוונטיים בניסוי. היות ובכל trail הנבדק החל את הדמיון המוטורי לאחר 2 וחצי שניות, משתנה זה מאפשר את שליפת האינדקסים המתאימים למסגרת זמנים זו בלבד מכלל נתוני ה-EEG.

```
function [] =
plot_power_label(electrode_cell, labels_idx, f, fs, t_window, t_overlap, img_t_frame, names)
%this function takes the data from the EEG, the labels and all the relavent
%setting for pwelch and returns plots for each electrode with the power
%accordint to the label.

colors = {'r','b'};
titles = names(1,:);

%for each electrode the loop will make a new figure with power spectrum
%according to the label.
for elect = 1:size(electrode_cell,2)
    figure
    sgtitle(['Power spectrum from ',titles{elect},' electrode'])
    for label = 1:size(electrode_cell,2)
        current_data = electrode_cell{elect}(labels_idx{label},img_t_frame);
        [y,x] = pwelch(current_data',t_window,t_overlap,f,fs);
        y = 10*log(mean(y,2));
        plot(x,y,colors{label})
        hold on
    end
    xlabel ('frequency [Hz]')
    ylabel ('power')
    legend(names(2,:))
end
```

end

באמצעות electrode_cell ו labels_idx מתאפשרת הפרדה בין החזרות שנערכו בתנאים שונים (שמאל/ימין) עבור כל אלקטרודה (C3/C4). באופן זה, יצרנו לולאת for פשוטה שבה בכל loop מוגדר המשתנה current_data שכולל את מידע ה-trails שנאסף באלקטרודה ובתנאי מסויימים. כמו כן, באמצעות המשתנה img_t_frame, המשתנה current_data מכיל את המידע שנאסף בטווח הזמנים הרלוונטי. על משתנה זה תופעל פונקציית ה-pwelch. פונקציית pwelch מקבלת 5 קלטים: א. הקלט עליו היא תבצע את הניתוח הדרוש. במקרה זה מדובר במשתנה current_data. ב. גודל החלון - פונקציית pwelch מחלקת את וקטור הקלט לחלונות על מנת להפחית את השפעתו של רעש. ג. גודל החפיפה בין חלונות. ד. וקטור תדרים. ה. תדר הדגימה (במקרה שלנו $fs = 128\text{Hz}$). פונקציה זו מחזירה את ה-power spectrum המוערך של הסיגנל שמזן אליה. לשם הפעלתה באופן מיטבי הגדרנו את גודל החלון כ- $fs*1.5$, ואת גודל החפיפה כחצי מגודל החלון. עבור הצגה ויזואלית נוחה



ואיכותית מיצענו את הפלט מעבר לחזרות ועל התוצאה הפעלנו \log והכפלנו ב-10. הפעולות המתוארות חוזרות על עצמן באמצעות לולאות ה-for המתאימות, לטובת יצירת שני חלונות, כאשר כל אחד מהם מכיל את נתוני אלקטרודה מסוימת בשני התנאים (ימין/שמאל). נציין גם כי הגדרת טווח הזמן ('img_t_frame') רלוונטית והכרחית במקרה זה מאחר ובגרף זה אין התייחסות למימד הזמן (רק לעוצמה והתדר) – במצב כזה דגימות שאינן מהשלב בו הנבדקים מדמיינים עלול לפגום ביכולת להסיק מסקנות מהגרפים.

Spectrogram

בשלב הבא יצרנו את הפונקציה `plot_spectu` שמטרתה ליצור ולפלוט ספקטוגרמה ממוצעת של נתוני החזרות בכל אחת מהאלקטרודות ובכל אחד מהתנאים. הספקטוגרמות יאפשרו לבחון את `power spectrum` בכל אחת מהאלקטרודות ובכל אחד מהתנאים כתלות בזמן. הפונקציה מקבלת כקלט את המשתנים: `electrode_cell, labels_idx, f, fs, t_vec, names`. כמו כן, הוגדרו והוזנו לפונקציה `s_window` ו-`s_overlap`, המייצגים את גודל החלון וגודל החפיפה בין חלונות, ודרושים לשימוש עתידי בפונקציה `spectrogram`. בפונקציה `plot_spectu` לא הוזן חלון זמן מסוים עליו יבוצע הניתוח, שכן הניתוח יבוצע על כל זמן הקלטת ה-EEG, זאת מפני שבשונה מהמתואר בפונקציה הקודמת, הספקטוגרמות מכילות התייחסות למימד הזמן ולכן ניתן להסיק מסקנות מאופן הגרף בהתייחסות שונה לכל שלב בניסוי והתוצאות בשלב הדמיון אינן מושפעות מהשלב שלפניו.

```
function [] =
plot_spectu(electrode_cell, labels_idx, f, fs, s_window, s_overlap, t_vec, names)
```

```
%this function takes the data from the EEG and the labels and with the relevant
%setting calculates the power for each electrode and label using spectrogram.
%later, different calculations and plots are made in order to compare the
%conditions more easily.
```

calculations

```
memory = cell(length(electrode_cell), length(labels_idx));

%for each electrode according to the label, data will be transform into power
%using the function spectrogram and other transformations.
for elec = 1:length(electrode_cell)
    for label = 1:length(labels_idx)
        current_data = electrode_cell{elec}(labels_idx{label},:);

        %cum_spec- cumulative spectrogram. we use this variable in order to sum all the
        %spectrograms and then in the end to calculate average spectrogram.
        cum_spec = 0;
        for trail = 1:size(current_data,1)
            spec_trail =
spectrogram(current_data(trail,:), s_window, s_overlap, f, fs, 'yaxis');
            spec_trail = 10*log(abs(spec_trail).^2);
            cum_spec = cum_spec+spec_trail;
        end

        mean_spec = cum_spec*(1/size(current_data,1));
        memory{elec, label} = mean_spec;
    end
end
```

ראשית, יצרנו את המשתנה `memory` בו ישמרו הנתונים הדרושים לספקטוגרמות. מדובר ב-cell array בגודל מספר האלקטרודות על מספר התנאים. באופן זה תתאפשר שמירה של הנתונים



השייכים לכל אלקטרודה ותנאי מסויימים בתא נפרד. באמצעות לולאות for פשוטות, בכל loop יוגדר המשתנה current_data כנתוני ה-EEG של כל החזרות שהוקלטו באלקטרודה ובתנאי הרלוונטי. לאחר מכן, בלולאה נוספת שרצה על כל אחד מהחזרות במשתנה current_data, תופעל הפונקציה spectrogram על נתוני החזרה הרלוונטית. פונקציה זו מחזירה את ה-short-time Fourier transform של האות שהיא מקבלת. למעשה, הפונקציה מחלקת את הסיגנל הארוך שמוזן אליה לסיגנלים קצרים יותר באורך s_window (על מנת לאפשר תצוגה מיטבית הוגדר כ-0.5*fs) עם חפיפה של s_overlap (הוגדר כחצי מגודל החלון). לאחר מכן הפונקציה מחשבת את Fourier transform לכל מקטע-סיגנל שכזה. פלט הפונקציה ישמר תחת המשתנה spec_trail. על משתנה זה תופעל העלאה בריבוע והפונקציה abs על מנת לקבל ערכי power. כמו כן תופעל הפונקציה log והתוצאה תוכפל ב-10 על מנת לאפשר הצגה מיטבית. כך ניתן לבחון את השינוי ב-power spectrums של האות כתלות בזמן. באופן זה מתבצע חישוב עבור אות של כל חזרה. היות ואנו מעוניינים למצוא את הספקטוגרמה הממוצעת, המשתנים spec_trail המייצגים את הספקטוגרמה של כל חזרה ייסכמו. לאחר שהלולאה תרוץ על כל הtrails באלקטרודה ובתנאי מסויימים, סכימת כל הספקטוגרמות, שנשמרת תחת המשתנה cum_spec, תוכפל באחד חלקי מספר החזרות. באופן זה תתקבל מטריצה המייצגת את הספקטוגרמה הממוצעת לאלקטרודה ולתנאי מסויימת, והיא תשמר בתא המתאים במשתנה memory.

על מנת לאפשר השוואה ויזואלית נוחה בין הספקטוגרמות של תנאים שונים בכל אלקטרודה, ובין הספקטוגרמות של האלקטרודות השונות בכל תנאי, החלטנו ליצור את ספקטוגרמות ההפרשים.

delta calculations

```
%here we make some extra calculations in order to produce more informative  
%spectrograms. we calculate the differences between labels within each electrode  
%and the differences between electrodes within each label.
```

```
delta_labels = cell(1,length(labels_idx));  
for label = 1:length(labels_idx)  
    delta_labels{label} = memory{2,label}-memory{1,label};  
end  
  
delta_elec = cell(1,length(electrode_cell));  
for elec = 1:length(electrode_cell)  
    delta_elec{elec} = memory{elec,1}-memory{elec,2};  
end
```

ראשית, בוצעו החישובים עבור ספקטוגרמות ההפרשים בין הספקטוגרמות של אלקטרודות שונות באותו התנאי. לשם כך יצרנו את המשתנה delta_labels, בו ישמרו הנתונים שיחושבו. משתנה זה הינו cell array בגודל 1 על מספר התנאים (במקרה זה 2- ימין ושמאל). על ידי ביצוע פעולת חיסור בין התאים המתאימים במשתנה memory, מחושבת תחילה ספקטוגרמת ההפרש בין הספקטוגרמות של שתי האלקטרודות בתנאי שמאל, ונשמרת בתא המתאים במשתנה delta_labels. פעולה זו חוזרת על עצמה עבור תנאי יד ימין.

לאחר מכן, באופן דומה יצרנו את המשתנה delta_elec, שהוא cell array בגודל 1 על מספר האלקטרודות (2). משתנה זה יאפשר שמירה של הנתונים הדרושים להפקת ספקטוגרמת ההפרשים שבין הספקטוגרמות של תנאי ימין ותנאי שמאל בכל אחת מהאלקטרודות.

לאחר ביצוע החישובים הדרושים, המשך הפונקציה מאפשרת את הפקתם של שלושת החלונות הדרושים. הראשון, מציג 4 ספקטוגרמות- ספקטוגרמה עבור כל אלקטרודה בכל אחד מהמתנאים. שני החלונות הנוספים מציגים את ספקטוגרמות ההפרשים שתוארו, על בסיס הנתונים השמורים במשתנים delta_labels ו-delta_elec.



Bandpower

הספקטוגרמות שהופקו איפשרו מציאת מקטעי זמן וטווחי תדר העשויים, בשל השוני בתבנית הפעילות בהן בין התנאים, לאפשר סיווג עתידי לתנאים (ימין/שמאל) על פי הקלטת EEG במהלך החזרות. לטובת יישומו של רעיון זה ייצרנו את הפונקציות band_power_calc ו-band_power_plot. בחנו את הספקטוגרמות ומצאנו שני מקטעי זמן ותדר בהם נמצא הפרש גדול בין הפעילות שנמדדה באלקטרודה בתנאי ימין ובתנאי שמאל (ספקטוגרמת ההפרשים אפשרה לנו לבחון זאת באופן פשוט). את טווחי התדר הרלוונטים הגדרנו תחת המשתנה freq_range, ואת מקטעי הזמן הרלוונטים תחת המשתנה time_frame. שתי הפונקציות שיתאורו כעת נמצאות בתוך לולאה שתרוץ כמספר מקטעי הזמן והתדר שהוגדרו במשתנים freq_range ו-time_frame.

האינדקס המתאים במשתנים freq_range ו-time_frame הם הקלט של הפונקציה band_power_calc, יחד עם המשתנים labels_idx, electrode_cell ו-fs.

```
function [band_power_cell] =  
band_power_calc(electrode_cell, labels_idx, freq_range, time_frame, fs)
```

```
%this function takes the data from the electrodes and calculate the band power  
%according to specific range (time and frequency). the output is a cell with all  
%the calculations according to electrode and label.
```

```
time_idx = time_frame*fs;
```

calculations

```
memory = cell(length(electrode_cell),length(labels_idx));  
for elec = 1:length(electrode_cell)  
    for label = 1:length(labels_idx)  
        current_data =  
        electrode_cell{elec}{labels_idx{label}},(time_idx(1):time_idx(2));  
        memory{elec,label} = bandpower(current_data',fs,freq_range);  
    end  
end  
  
band_power_cell = memory;  
  
end
```

ראשית, הקלט time_frame מוכפל בתדר הדגימה fs על מנת ליצור את המשתנה time_idx, שיאפשר שימוש נוח בטווחי הזמנים בהמשך. לאחר מכן, מוגדר המשתנה memory שישמור את הנתונים הרלוונטים. משתנה זה הינו cell array בגודל מספר האלקטרודות על מספר התנאים. בשלב זה יוגדר המשתנה current_data, על פי ההגדרות הרלוונטיות לאותו loop בלולאה. בכל פעם המשתנה יכיל את הקלטות החזרות באלקטרודה ובתנאי מסויימים בין טווחי הזמן שהוגדרו במשתנה time_idx. לאחר מכן תופעל על המשתנה הפונקציה bandpower. פונקציה זו מקבלת כקלט את הסיגנלים הרלוונטים (current_data), את תדר הדגימה fs, ואת טווח התדרים הרלוונטי freq_range. הפונקציה מחשבת את מדד energy בכל אחת מהחזרות שהוזנו (שכבר מתייחסים לחלון הזמן הרלוונטי בלבד) בטווח התדרים שהוזן. בשל העובדה כי בחרנו מקטעי זמן ותדר שע"פ הספקטוגרמה יש בהם הבדלים בפעילות שהוקלטה בין תנאי ימין ושמאל, ייתכן כי מדד bandpower יכול להיות מאפיין שייצג בצורה מייטבית את הבדל זה. הפונקציה פולטת וקטור באורך מספר החזרות בתנאי מסויים (64), ומאחסנת אותו בתא המתאים במשתנה memory. הלולאה חוזרת על עצמה עבור כל צירוף בין אלקטרודה לתנאי. הפונקציה פולטת את המשתנה band_power_cell שזהה למשתנה memory.



בשלב זה נרצה להציג את המידע שחושב בצורה ויזואלית נוחה שתאפשר לזהות בקלות האם מאפיין ה-bandpower בטווחי הזמן והתדר אכן שונה (וכותצאה מכך יכול לאפשר סיווג עתידי של הקלטות EEG) בין הקלטות החזרות בתנאי שמאל ובתנאי ימין. במידה והוא שונה, נרצה לבחון האם השוני קיים בשתי האלקטרודות או רק באחת מהן. לשם כך יצרנו את הפונקציה `plot_band_power`.

```
function [] = plot_band_power(band_power_cell,names,hist_range,Binwidth)
%this function plots the bandpowers from bandpower cell in histograms according
%to histogram settings.

figure
for elec = 1:size(band_power_cell,1)
    nexttile
    for label = 1:size(band_power_cell,2)
        title (names(1,elec));
        a = histogram(band_power_cell{elec,label},hist_range);
        a.Binwidth = Binwidth;
        hold on
    end
    legend(names(2,:))
    xlabel('power')
    ylabel('number of trails')
end
```

פונקציה זו מקבלת את הנתונים שחושבו ונשמרו במשתנה `band_power_cell`, וכמו כן את המשתנים `names`, `hist_range` ו-`Binwidth` הדרושים להצגת הגרפים בצורה מיטבית. באמצעות הפונקציה `histogram` יופק חלון המכיל שתי מערכות צירים: היסטוגרמות המציגות את התפלגות מדד ה-bandpower שחושב עבור כל החזרות בטווח הזמן והתדר הרלוונטים, בכל אחת מהאלקטרודות. תתאפשר השוואה נוחה בין ההתפלגות של המדד בתנאי ימין לתנאי שמאל בכל אחת מהאלקטרודות.

כאמור, השימוש בשתי הפונקציות המתוארות יחזור על עצמו עבור כל טווח זמן ותדר רלוונטי שמוגדר על ידי המשתמש.

features extract

לאחר הניתוחים המקדימים שבוצעו, יצרנו פונקציה בשם `classification` המחלצת מנתוני ה-`trails` מספר מאפיינים (features) שנמצאו כיעילים למטרת סיווג של הקלטים על פי התנאי שבו הם הוקלטו (דמיון של הזזת יד ימין או שמאל), ולניבוי בהמשך של התנאי שבו הוקלטו קלטים לא מתוייגים.

```
function [features] = classification(data,fs,f)
%this function takes the data (train/test) and using f and fs calculates the
%features for the data in 'features'. each additional feature saved as
%'new_feature' and then added as a new row at the end of the matrix 'features'.
%the index of data sets the specific electrode. we did not use here a loop or
%complex indexing in order to be able to check features easily and seperately for
%each electrode.

n_trails = size(data{1},1);           %number of trails

features = [];                         %creating/reseting memory

%(1-2)band power 1
freq_range = [15 18];
time_frame = [4.5 6];
```




```

new_feature =
bandpower((data{1}(:,(time_frame(1)*fs):(time_frame(2)*fs))),fs,freq_range);
features = [features ; new_feature];

new_feature =
bandpower((data{2}(:,(time_frame(1)*fs):(time_frame(2)*fs))),fs,freq_range);
features = [features ; new_feature];

%(3-4)band power 2
freq_range = [30 35];
time_frame = [5 6];

new_feature =
bandpower((data{1}(:,(time_frame(1)*fs):(time_frame(2)*fs))),fs,freq_range);
features = [features ; new_feature];

new_feature =
bandpower((data{2}(:,(time_frame(1)*fs):(time_frame(2)*fs))),fs,freq_range);
features = [features ; new_feature];

```

הפונקציה מקבלת כקלט את המשתנים `electrode_cell`, `fs`, `f` (לשם נוחות בפונקציה משתנה זה נקרא `data`). ראשית, היא מגדירה את המשתנה `n_trails` כמספר החזרות שבוצעו בסך הכל (128) על פי המימד הרלוונטי במשתנה `data`. שנית היא מגדירה את המשתנה `features` כמטריצה ריקה. כל מאפיין חדש נשמר תחת `new_feature` ובהמשך מתווסף לתחתית המטריצה `features` כך שבאופן זה נוח להתנסות בחילוף מאפיינים שונים בתוך הפונקציה וזאת תוך שמירה על מבנה שיאפשר ניתוח קל ונוח בהמשך. בנוסף, על מנת לאפשר גמישות מקסימלית ופשטות, בחרנו לקרוא לכל מידע מכל אלקטרודה באמצעות אינדקסים פשוטים (1 - C3 - C4) וזאת על מנת לאפשר בדיקת מאפיינים גם רק עבור אלקטרודה אחת בצורה פשוטה יחסית.

באמצעות ספקטוגרמת ההפרשים והצגת היסטוגרמות `bandpower` מצאנו שני טווחי תדר וזמן שחישוב ה-`bandpower`, כלומר מדד ה-`energy` בהם, עשוי לאפשר סיווג וניבוי מוצלחים של התנאי שבו התרחש סיגנל EEG שהוקלט. לטובת חישוב ה-`bandpower` של כל `trail` בטווח התדרים (המשתנה `freq_range`) ובטווח הזמן (המשתנה `time_frame`) שהוגדרו, נעשה שימוש בפונקציה `bandpower`. ראשית, הפונקציה הופעלה על טווח הזמן הרלוונטי באלקטרודות C3 וC4. כמו כן הוזן טווח התדרים הרלוונטי. הפונקציה פולטת וקטור באורך מספר ה-`trails` המכיל את ה-`energy` בכל `trail` בטווחים שהוגדרו.

```

%(5-10)
p_window = 1.5*fs;          %t = time, length of time windows [sec]
p_overlap = p_window/2;     %overlap between windows [sec]/[as part of window]

%frequency bands
f_resolution = 0.1;         %frequency vector resolution [Hz].
%this 2 vectors can be modified. each index in the name vectors is connected
%later with the same range index. more info inside 'freq_band'.
Bands_Name = ["delta" , "theta" , "low_alpha" , "high_alpha" , "beta" , "gamma"];
Bands_range = [ 1, 4.5; 4.5,8; 8, 11.5; 11.5, 15; 15,30 ; 30, 40];

[freq_map] = freq_band(Bands_Name,Bands_range,f_resolution);
[y,~] = pwelch(data{1}',p_window,p_overlap,f,fs);

new_feature = relative_power(y,freq_map,n_trails);
features = [features ; new_feature];

[y,~] = pwelch(data{2}',p_window,p_overlap,f,fs);
new_feature = relative_power(y,freq_map,n_trails);

```



```
features = [features ; new_feature];
```

```
end
```

בשלב זה בפונקציה מחושבים features נוספים הנמצאו כשימושיים לתהליך הסיווג- relative power, דהיינו החלק היחסי בסיגנל של ה-power בטווח תדרים מסויים מסך ה-power בסיגנל. לפני חישוב המדד עצמו, ולטובת נוחיות החישוב יצרנו את הפונקציה freq_band (אותה יצרנו בתרגיל הקודם) עליה לא נפרט שוב (פונקציה טכנית מאוד בעיקרה) אשר מייצרת את המשתנה freq_map בהתאם לשמות התדרים ולטווחים המוזנים (מייצרת מהם cell array).

בשלב הבא, הופעלה הפונקציה pwelch על נתוני trails כפי שהוקלטו על ידי אלקטרודה C3 על ידי שליפתם מהמשתנה data. פונקציה זו מחזירה את ה-power spectrum המוערך של הסיגנל שמזן אליה. גודל החלון הוגדר כ- $fs*1.5$, וגודל החפיפה בין החלונות הוגדרה כחצי מגודל החלון. המטריצה שהיתקבלה נשמרה תחת המשתנה y.

לטובת חישוב מדד ה-relative power השתמשנו בפונקציה relative_power (אותה יצרנו בתרגיל הקודם).

```
function [relative_power_per_window] = relative_power(power,freq_map,n_windows)
%This function takes the power matrix and converts the power to relative power
%according to bands in the settings.

n_freq_bands = length(freq_map);

total_power = sum(power);

%creating memory
relative_mat = zeros(n_freq_bands,n_windows);

%for each band, the loop sums only the relevent band width and then multiplying
%each value in 1/the total power that relvent for the band.
for band = 1:n_freq_bands

    %we take only the relvent spectrum using the index vectors in 'freq_map' and
    %sum it.
    relative_mat(band,:) = sum(power(freq_map{3,band},:));

    %then we split by the total power fot the selected band width.
    relative_mat(band,:) = relative_mat(band,:).*(1./total_power);
end

relative_power_per_window = relative_mat;

end
```

הפונקציה מקבלת כקלט את המשתנה freq_map, את מספר החלונות שהוגדר מראש, ואת המטריצה y שמכילה את ה-power spectrum של כל אחת מהחזרות. ראשית, ביצענו חישוב של ה-total power של כל אחת מהחזרות על ידי הפעלת הפונקציה sum על המטריצה y. לאחר מכן יצרנו מטריצת אפסים בשם relative_mat, בה נשמור את הנתונים שיוחשבו בהמשך. במטריצה זו כל שורה תייצג קבוצת תדרים, וכל עמודה תייצג trail. גודלה של המטריצה נקבע בהתאם. על מנת לחשב את ה-relative power של כל אחת מקבוצות התדרים יצרנו לולאת for. בתוך הלולאה, תחילה נתבצע סכימה של מדד ה-power של קבוצת התדרים הרלוונטית על ידי שימוש בנתוני המשתנה freq_mat שהוצג קודם, ושמירה בשורה ובעמודה המתאימה במשתנה relative_mat. לאחר מכן, על מנת לחשב את העוצמה היחסית של קבוצת התדרים, נתבצע חלוקה של מדד ה-power שנסכם קודם במדד ה-total power. תהליך זה יחזור על עצמו עבור כל אחת מקבוצות התדרים שהוגדרו.



לאחר מכן, המשתנה `relative_mat` יישמר בשורות המתאימות במשתנה `features`. תהליך זהה יחזור על עצמו עבור `trails` כפי שהוקלטו באלקטרודה C4.

הפונקציה `classification` פולטת את המשתנה `features` המכיל את כל המאפיינים שחושבו.

צמצום מימדים וויזואליזציה

בשלב זה רצינו להציג באופן ויזואלי את איכות ההפרדה שהמאפיינים שחילצנו מייצרים בין חזרות שנערכו בתנאי יד שמאל לבין חזרות שנערכו בתנאי יד ימין. בחרנו לצמצם את המימדים באמצעות אלגוריתם PCA ולשם כך יצרנו את הפונקציה `plot_pca`.

```
function [] = plot_pca(features, labels_idx, names)
%this function calculate PCA for features matrix and plots 2 and 3 dim scatter
%plots for them.

features = zscore(features,0,2);

pca1 = pca(features);
colors = {'r','b'};
figure
for i = 1:length(labels_idx)
    scatter(pca1(labels_idx{i},1),pca1(labels_idx{i},2),colors{i})
    hold on
end
legend(names(2,:))
xlabel ('PCA 1')
ylabel ('PCA 2')

figure
for i = 1:length(labels_idx)

scatter3(pca1(labels_idx{i},1),pca1(labels_idx{i},2),pca1(labels_idx{i},3),colors{i})
    hold on
end
legend(names(2,:))
xlabel ('PCA 1')
ylabel ('PCA 2')
zlabel ('PCA 3')

end
```

על מנת שהממדים שחושבו לא יושפעו מיחידות המידה השונות של כל אחד מהממדים, ביצענו חישוב של ציוני תקן, כך שממוצע כל אחד מהממדים יהיה 0 וסטיית התקן שלו תהיה 1. על מנת לבצע את החישוב השתמשנו בפונקציה `zscore`. לפונקציה הכנסנו את המשתנה `features` עליו היא פועלת. כמו כן הפונקציה מבצעת חישוב שמתאים למדגם ולא לאוכלוסיה. הפלט הינו המשתנה `feature1`.

משתנה זה, יחד עם המשתנים `labels_idx` ו-`names`, הינם הקלט של הפונקציה `plot_pca`. בפונקציה זו, מופעלת הפונקציה `pca` על המשתנה `feature1`. אלגוריתם זה (PCA Analysis Components) מאפשר לבצע דחיסה של המידע שברשותנו על ידי הפחתה של מימדיו, באופן שיותר בדינו מידע רב ככל האפשר. על ידי ביצוע האלגוריתם נוריד את מימד המידע שאספנו מ-16 מימדים (16 הוא מספר הממדים שחושבו ומרכיבים את המטריצה `features1`) ל-3 מימדים ול-2 מימדים, כך שתתאפשר הצגה ויזואלית מיטבית של המידע, ותתאפשר הסקת מסקנות. הפונקציה `pca` מקבלת כקלט את המשתנה `features1` ופולטת את המשתנה `pca1`. משתנה זה הינו מטריצה בגודל 128 על 15, המכיל את 15 הוקטורים העצמיים בעלי השונות הגדולה ביותר (בסדר יורד) של מטריצת ה-covariance של `feature1`. לטובת הצגה תלת מימדית יעשה שימוש בשלושת הוקטורים העצמיים הראשונים (בעלי השונות הגדולה ביותר). לטובת הצגה דו מימדית יעשה שימוש בשני הוקטורים



העצמיים הראשונים בלבד. באמצעות הוקטור colors, הנקודות על גבי הגרפים הנפלטות המייצגות חזרות בתנאי יד שמאל ייפלטו בצבע אדום, ונקודות המייצגות חזרות בתנאי יד ימין ייפלטו בצבע כחול.

סיווג ואימון

בשלב זה יצרנו את הפונקציה classfun. מטרתה של פונקציה זו הינה לאמן classifier שיפריד באופן לינארי בין הקלטות החזרות השונות על פי התנאי בו הם התקיימו (ימין/שמאל), ולאחר מכן לבחון את יכולתו ואת מידת הדיוק בה הוא מסווג. תהליך זה התבצע בשיטת k-fold cross-validation, שיטה סטטיסטית המאפשרת להעריך את יכולת הניבוי של ה-classifier. בשיטה זו מחלקים את קובץ האימון (128 החזרות) המקורי ל-k מקטעים. בכל פעם יבחרו k-1 מקטעים מתווייגים על מנת לאמן את ה-classifier, והמקטע הנותר שתיוגיו לא יוזנו ל-classifier ישמש כמקטע בוחן הבוחן את יכולת ההכללה, הניבוי והדיוק שלו. תהליך זה יחזור על עצמו k פעמים.

```
function [mean_success,sd_success,labels] = classfun(features,labels_idx,k)
% this function train a classifier on part of the data and then test its
%performance on the rest of the data. the function Use K-fold cross-validation
%to measure the classifier performance

n_trails = size(features,2); %sets number of trails.
labels = zeros(length(features),1);
labels(labels_idx{1}) = 1; %create labels vector that fits to trails order. left=1,
right=0

class_data = [features' labels]; % connect each trails features to the appropriate
label.

random_class_data = class_data(randperm(size(class_data, 1)), :); %randomize trails
order.
rand_labels = random_class_data(:,end);

%this variable will keep the accuracy measure of the classicication
%in each one of the k repetition. first row- accuracy in train.
%second row- accuracy in validation.
group_size = n_trails/k;
success_per_fold= zeros(2,k);

for i = 1:k
%sets the index of the current validation trails.
test_idx = [(((i-1)*group_size+1):(group_size)*i)];
% save the features of the validation trails.
test_group = random_class_data(test_idx,(1:end-1));

%sets the index of the current train trails.
train_idx = setxor(1:length(class_data),test_idx);
% save the features of the train trails.
train_group = random_class_data(train_idx,(1:end-1));
%use of classify function to predict the condition of the validation
%trails(left\right).
[class,err] = classify(test_group,train_group, rand_labels(train_idx));

%1-err represents the accuracy of the predictions in the train group.
success_per_fold(1,i) = 1-err;
%calculate the number of correct predictions in the validation group.
test_success = sum(class == rand_labels(test_idx));
%represents the accuracy of the predictions in the train group.
test_success = test_success/length(test_group);
```



```

success_per_fold(2,i) = test_success;
end

%calculate the mean & SD prediction accuracy of the k repetition (in train &
validation).
mean_success = mean(success_per_fold,2);
sd_success= std(success_per_fold,1,2);

end

```

הפונקציה מקבלת כקלט את המשתנים features, labels_idx ואת ה-k המבוקש. ראשית, הפונקציה יוצרת את המשתנה labels המכיל את התיוגים לתנאי ה-trails. מדובר בוקטור באורך מספר ה-trails שבוצעו (128). חזרות בהן הנבדק דמיין כי הוא מזיז את ידו השמאלית מסומנים ב-1, ו-חזרות בהן הוא דמיין כי הוא מזיז את ידו הימנית מסומנים ב-0. בשלב הבא משתנה זה ישורשר למטריצת המדדים המשוכלפת features, כך שלכל שורת מדדים של trail ישורש התיוג של החזרה המתאימה. משתנה זה נקרא class_data. לאחר מכן תתבצע רנדומיזציה של סדר השורות במשתנה. בשלב הבא יוגדר המשתנה success_per_fold, בו ישמרו אחוזי ההצלחה של הסיווג במקצי ה-training ובמקצי ה-validation, בכל אחת מ-k החזרות.

לשם ביצוע החישובים יצרנו לולאה פשוטה שמבצעת k חזרות. את ה-k הגדרנו בקוד כ-8. הגענו למספר זה תוך התחשבות במספר שיקולים. מחד, k גדול עלול להוביל לזמן ריצה ארוך ויקר. מאידך, k קטן עלול להוביל לשונות גדולה בתוצאות שיתקבלו. לכן, ה-k שהגדרנו מאפשר הגעה ליכולת סיווג מקסימלית, תוך התחשבות בשונות התוצאות (שמירה על שונות קטנה ככל האפשר) ובזמן ריצת הקוד (שמירה על זמן ריצה קצר ככל האפשר). בכל חזרה יוגדרו האינדקסים של קבוצת האימון ושל קבוצת ה-validation. באמצעות האינדקסים יוגדרו המשתנים test_group (שיכיל מאפיינים של $1/k$ מהחזרות שבוצעו), train_group (שיכיל את יתר מאפייני החזרות). החזרות המרכיבות את שתי הקבוצות יהיו תמיד חזרות שונות, ללא חפיפה בין הקבוצות. זאת מכיוון שבניבוי התיוגים של קבוצת ה-test, אנו נרצה לבחון את יכולת ההכללה של הסיווג, ולהימנע מהצלחה מזויפת הנובעת מ-overfitting. בשלב זה נשתמש בפונקציה classify. פונקציה זו מקבלת כקלט את המשתנה test_group, את המשתנה train_group ואת התיוגים לקבוצת האימון בלבד. במצב ברירת המחדל של הפונקציה היא מבצעת סיווג בשיטת LDA (Linear Discriminant Analysis), דהיינו סיווג אופטימלי באמצעות הפרדה לינארית בלבד. הפונקציה משתמשת בקבוצת האימון ובתיוגיה על מנת ליצור תיוגים מנובאים לקבוצת ה-validation. הפונקציה פולטת שני משתנים: המשתנה err הינו מדד המתייחס לשגיאה המוערכת בסיווג שמבוססת על קבוצת האימון. $(1-err)$, כלומר ההצלחה המוערכת בסיווג המבוססת על קבוצת האימון, תשמר בשורה הראשונה במשתנה (success_per_fold). המשתנה class הינו התיוגים המנובאים לקבוצת ה-validation. על מנת לחשב את אחוזי ההצלחה של הניבוי, הפונקציה בודקת בכמה מתוך ה-validation trails הניבוי היה זהה לתיוג האמיתי של ה-trails, ומחלקת מספר זה במספר ה-trails בקבוצת ה-validation. מדד זה ישמר בשורה השניה במשתנה success_per_fold.

לאחר ביצוע התהליך המתואר k פעמים, יתבצע מיצוע של המדדים המחושבים מעבר ל-k החזרות (mean_success). כמו כן תחושב שגיאת התקן של התוצאות שהיתקבלו (sd_success). פלט הפונקציה הינו שני המשתנים הנ"ל. בנוסף אליהם הפונקציה פולטת את המשתנים labels ו-class.

סיווג קובץ המבחן

test

```

load motor_imagery_test_data.mat
test_trails_cells = {data(:,:,1) data(:,:,2)};
test_features = classification(test_trails_cells,fs,f);

[class, err] = classify(test_features',features',labels);

```



לאחר שה-features המחולצים איפשרו השגת סיווג מדויק באופן מספק, תוך שמירה על שונות נמוכה בין התוצאות וזמן ריצה קצר ככל האפשר, בחנו את יכולת הסיווג על קובץ ה-test. שם הקובץ הינו motor_imagery_test_data.mat, והוא מכיל את המשתנה data. משתנה זה מכיל נתונים מ-32 trails נוספים, כפי שנמדדו ב-3 אלקטרודות. מתוך הנתונים חילצנו את הנתונים כפי שנמדדו על ידי אלקטרודה C3 ו-C4 בלבד. הנתונים שנמדדו על ידי כל אחת מהאלקטרודות נשמרו ב-cell array בשם test_trails_cells. משתנה זה הוזן לתוך הפונקציה classification שתוארה קודם. באופן זה חולצו מהנתונים אותם המאפיינים שהפונקציה חילצה בשלב מוקדם יותר של הקוד (relative power, power band). הפונקציה פולטת את המשתנה test_features, המכיל את מטריצת המאפיינים של המשתנה test_trails_cells. בשלב זה הופעלה שוב הפונקציה classify. בשימוש הזה בפונקציה, הוזנו המשתנה features במלואו כסט האימון, והמשתנה labels המכיל את התיוגים לסט האימון. סט המבחן המוזן הוא test_features. הפונקציה פולטת את המשתנה err (מדד המתייחס לשגיאה המוערכת בסיווג שמבוססת על קבוצת האימון), ואת המשתנה class (התיוגים המנובאים לקבוצת המבחן).

תוצאות

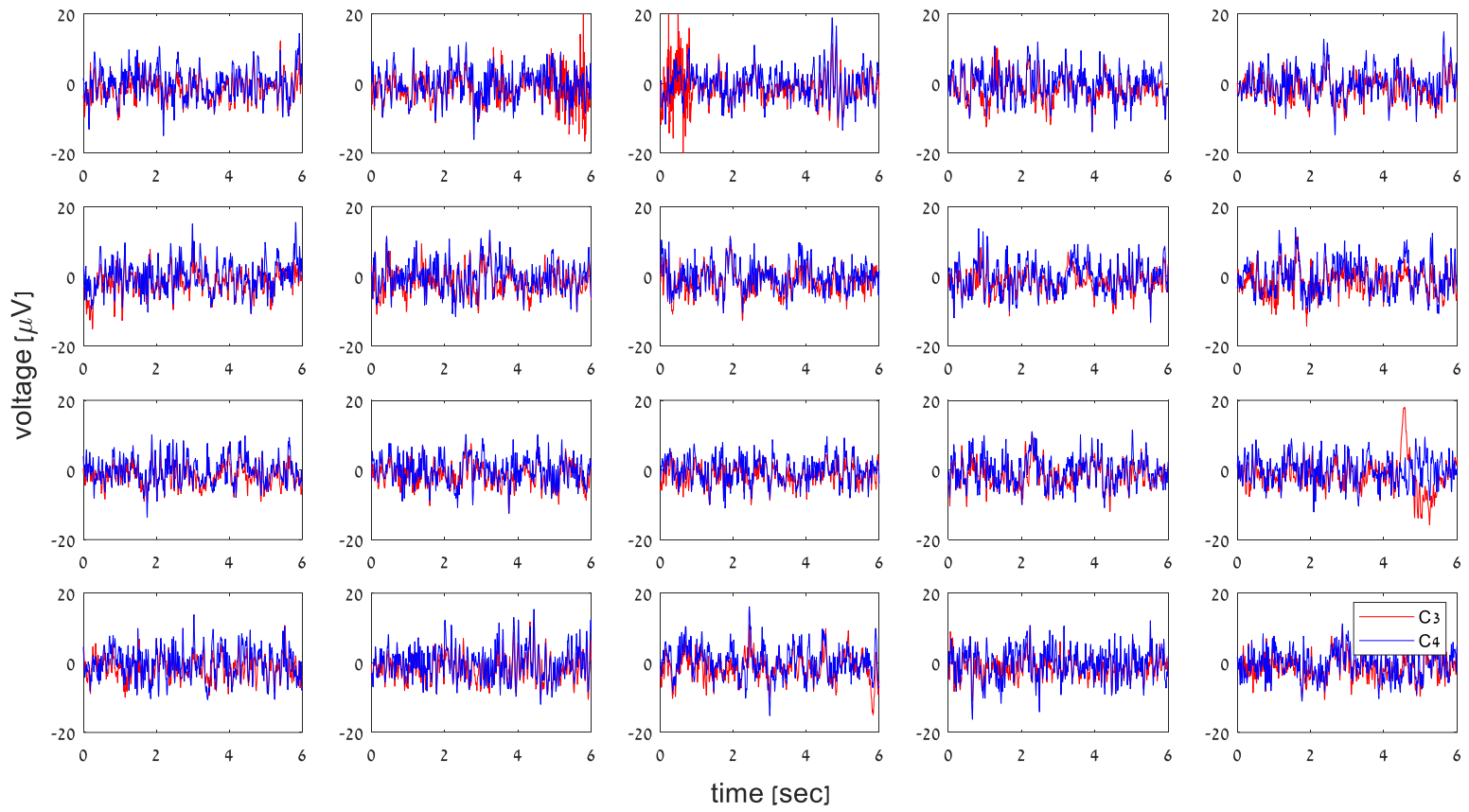
EEG signals visualization

תחילה יוצגו מספר ניתוחים מקדימים, המאפשרים להשיג הבנה בסיסית של מאפייני החזרות שהוקלטו באמצעות הצגות ויזואליות פשוטות. על מנת לבחון באופן כללי את הנתונים, הופקו שני חלונות, שבכל אחד מהם 20 מערכות צירים. בחלון הראשון, 20 מערכות הצירים מציגות את הסיגנל שנמדד על ידי ה-EEG בחזרות שנערכו בתנאי יד ימין, ובחלון השני 20 מערכות הצירים מציגות חזרות שנערכו בתנאי יד שמאל. בכל מערכת צירים, ציר ה-x מייצג זמן, וציר ה-y מייצג מתח. על כל מערכת צירים מופיעים שני גרפים. הצבע האדום מייצג את האלקטרודה C3, והצבע הכחול את האלקטרודה C4.



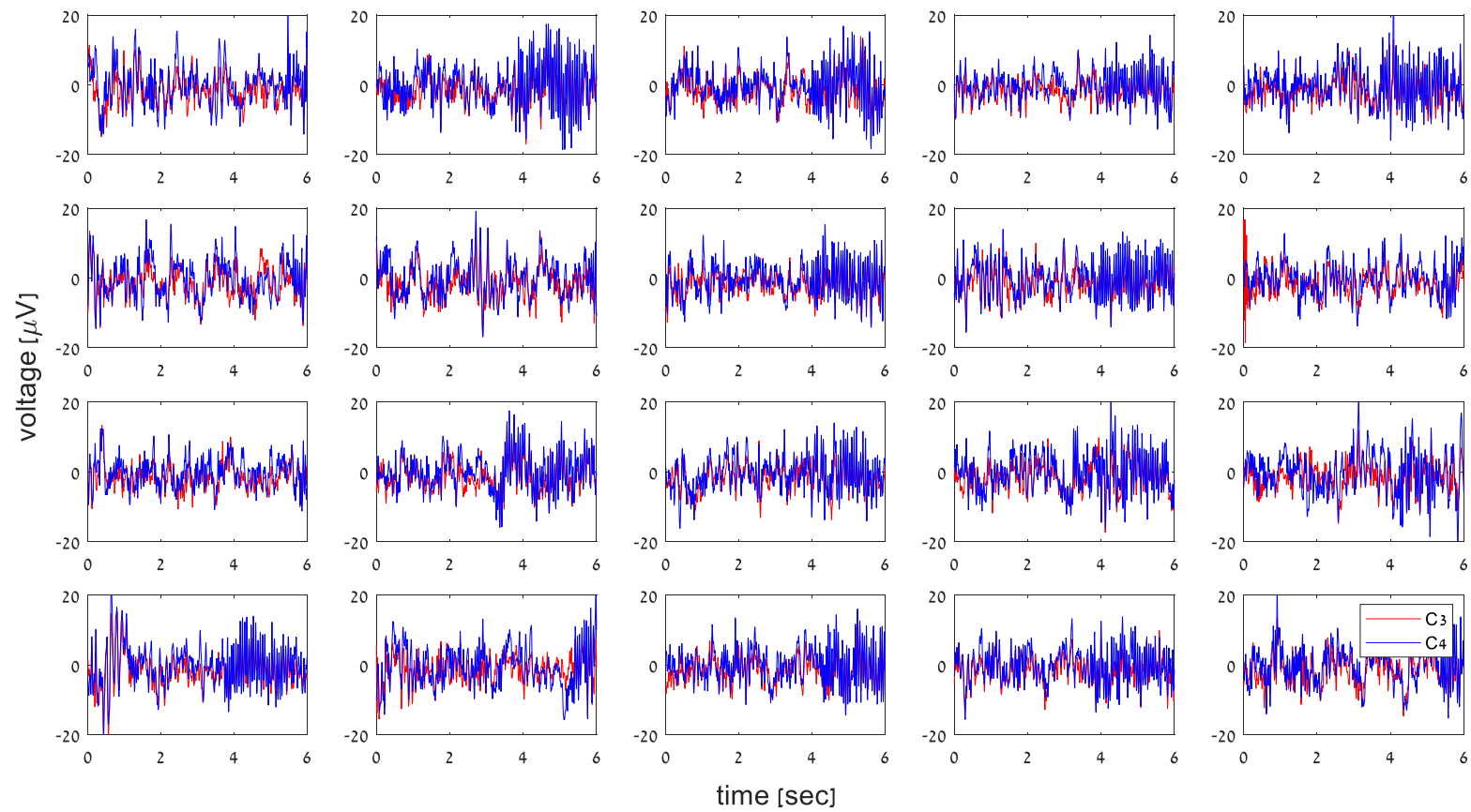
EEG signals for Left hand imagery

איור 1.1



EEG signals for Right hand imagery

איור 1.2





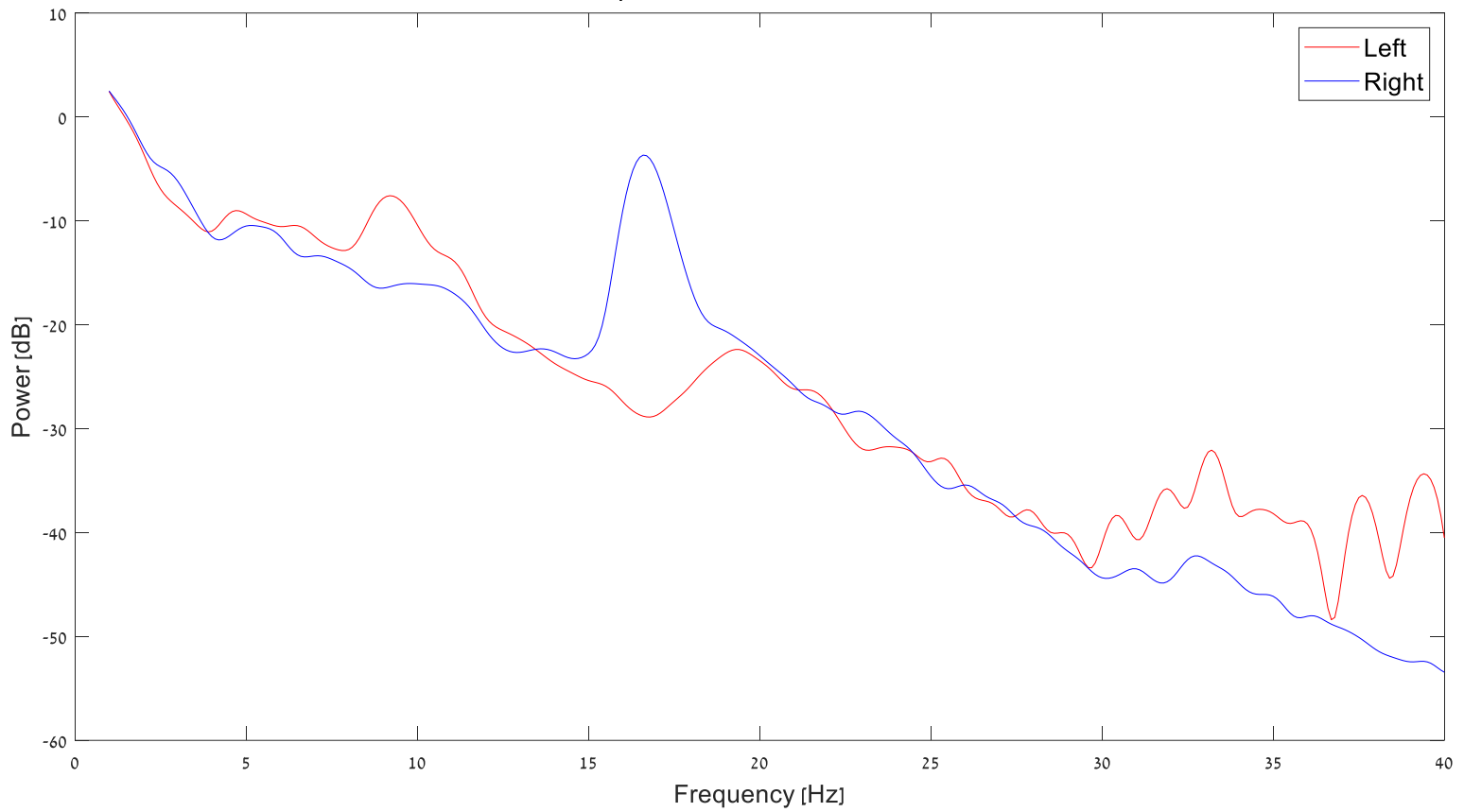
Power spectrum

ניתוח והצגה ויזואלית נוספת שעשויה להיות אינפורמטיבית ולאפשר הבנה של אופי הפעילות הנמדדת באלקטרודות ובתנאים שונים, הינה הצגת ה- power spectrum של הסיגנלים. לשם כך הופקו שתי מערכות צירים. כל מערכת צירים מייצגת את ההקלטות באלקטרודה אחת בלבד (C3\C4). בכל מערכת צירים שני גרפים. הגרף האדום הינו ה- power spectrum הממוצע של החזרות שהוקלטו בתנאי יד שמאל, באלקטרודה הרלוונטית. הגרף הכחול הינו ה- power spectrum הממוצע של trails שהוקלטו בתנאי יד ימין, באלקטרודה הרלוונטית. יש לציין, כי עבור כל הסיגנלים, ה- power spectrum חושב על פי מקטע הזמן שבו הנבדק ביצע את הדמיון המוטורי בלבד. בכל מערכת צירים, ציר ה-x מייצג frequency, וציר ה-y מייצג power.



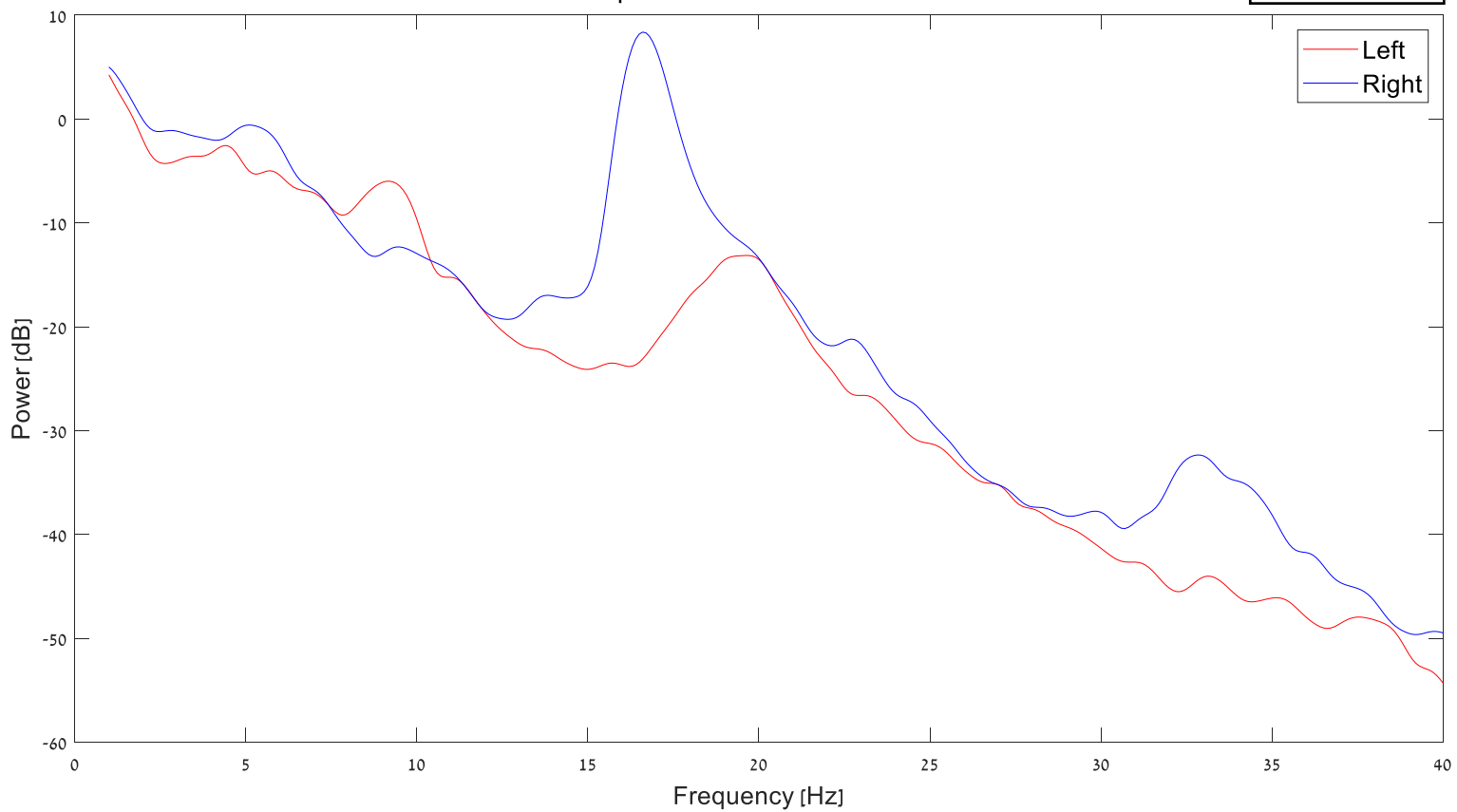
איור 2.1

Power spectrum from C3 electrode



איור 2.2

Power spectrum from C4 electrode



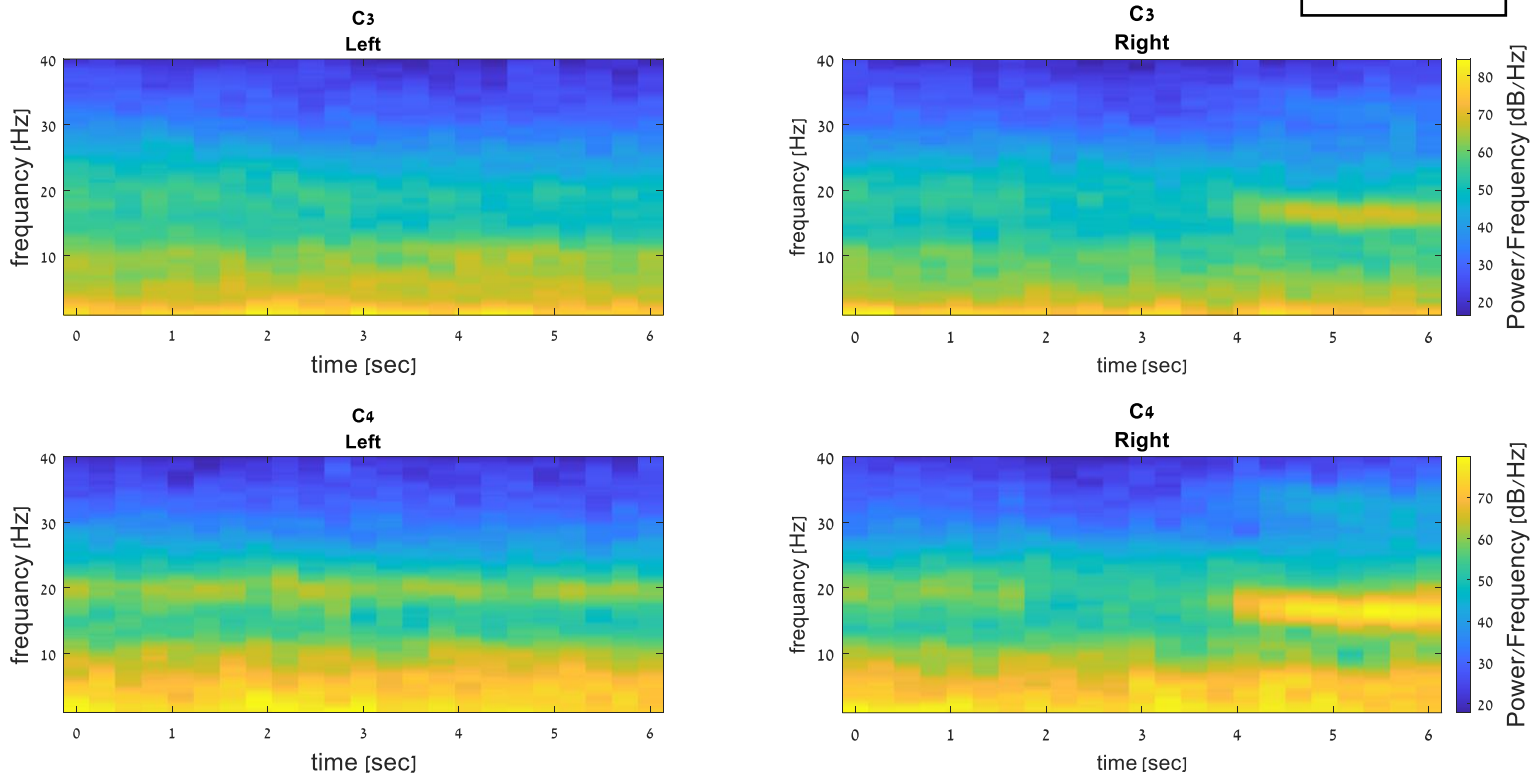


Spectugrams

הצגת ה-power spectrums באופן זה אינה מאפשרת הסקה של השפעת השינוי בזמן על עוצמת התדרים המרכיבים את הסיגנל. לשם כך, בשלב זה, ניתן לבחון את ארבעת הספקטרוגרמות המופקות באמצעות הקוד. כל ספקטוגרמה מכילה מידע מאלקטרודה אחת (C3\C4), בתנאי אחד (יד ימין/יד שמאל). ציר ה-x של כל ספקטוגרמה מייצג את הזמן, וציר ה-y מייצג תדרים. השינוי בצבעים מייצג את מדד ה-power/frequency.

Spectugrams by electrode & label

איור 3.1

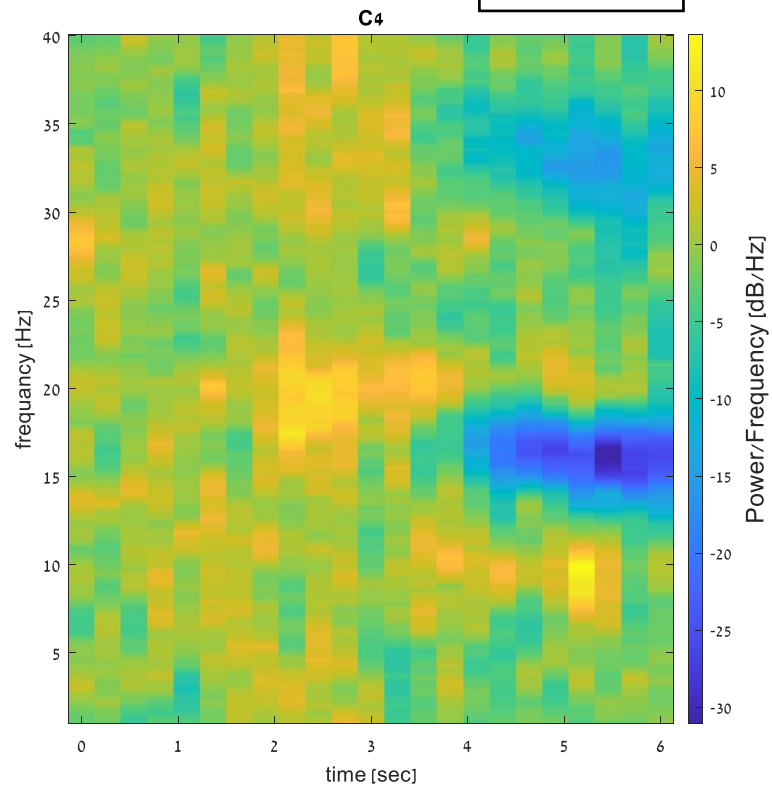
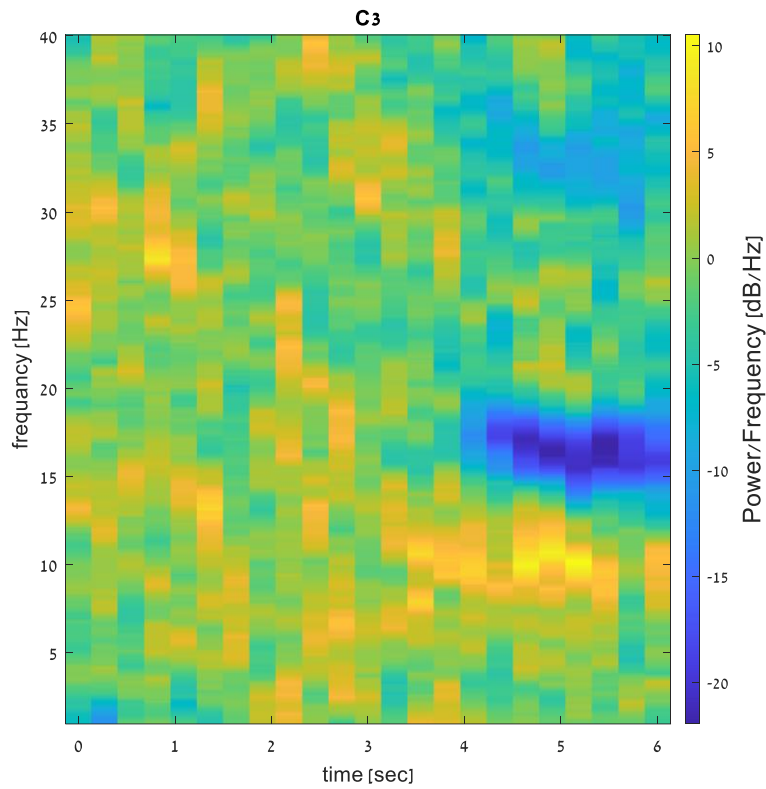


על מנת לאפשר בחינה ויזואלית נוחה, ניתן גם לבחון את ספקטוגרמות ההפרשים. ראשית, הופקו 2 ספקטוגרמות המכילות את ההפרש בין ספקטוגרמת תנאי 1 (לדוגמא, ימין) לבין ספקטוגרמת תנאי 2 (לדוגמא, שמאל), כפי שנמדדו בכל אחת מהאלקטרודות. הצגה זו מאפשרת בחינה נוחה של השוני בסיגנלים שנקלטו באותה אלקטרודה בתנאים שונים. בנוסף, הופקו 2 ספקטוגרמות נוספות המציגות את ההפרש בפעילות שנקלטה בשתי האלקטרודות השונות באותו התנאי. בכל הספקטוגרמות המתוארות הציר ה-x מייצג זמן, ציר ה-y מייצג תדר, והשינוי בצבעים מייצג את מדד ה-power/frequency.



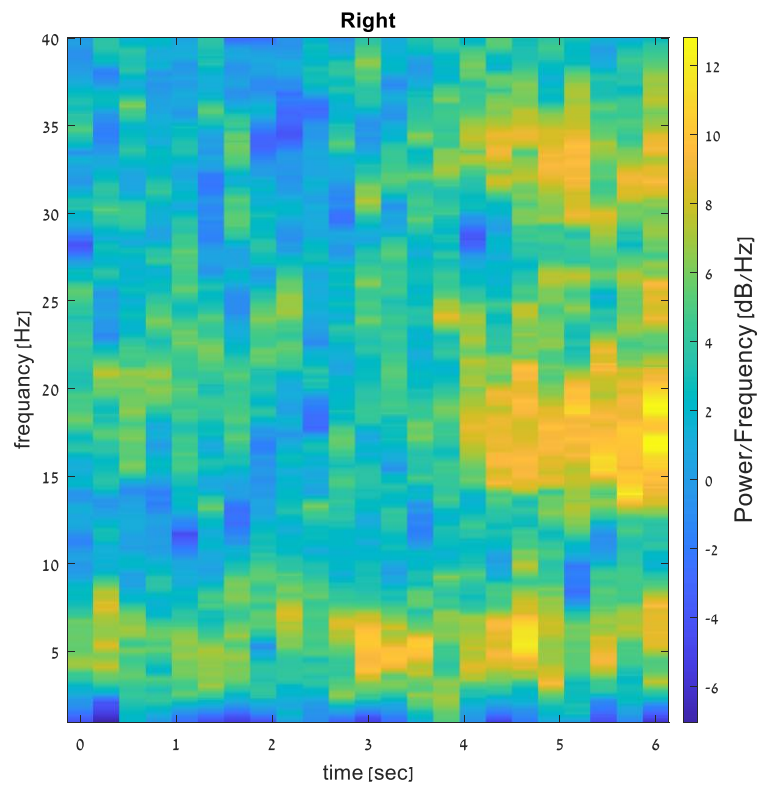
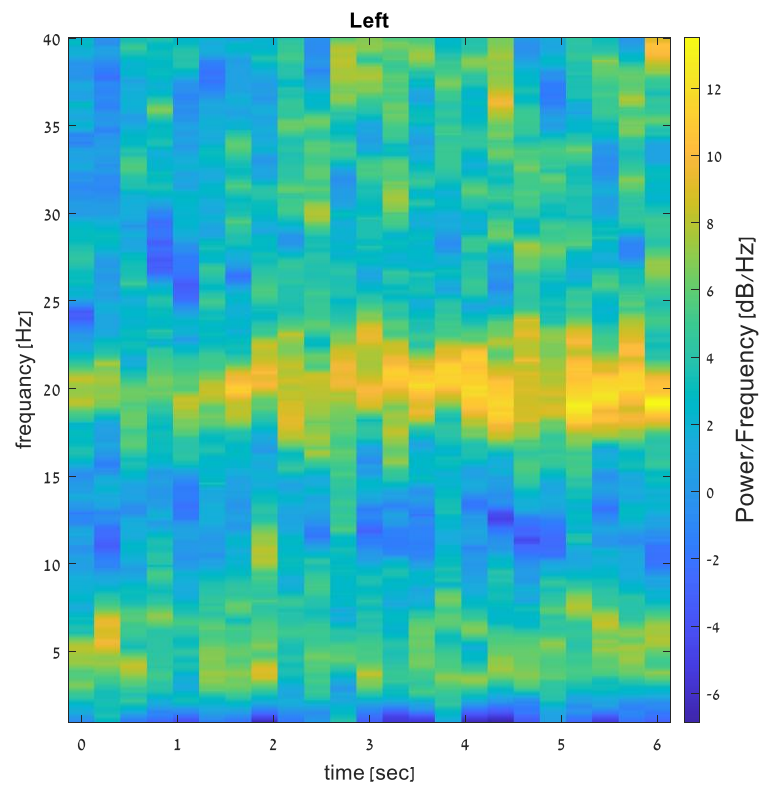
Differences Spectugrams by electrode

איור 3.2



Differences Spectugrams by label

איור 3.3



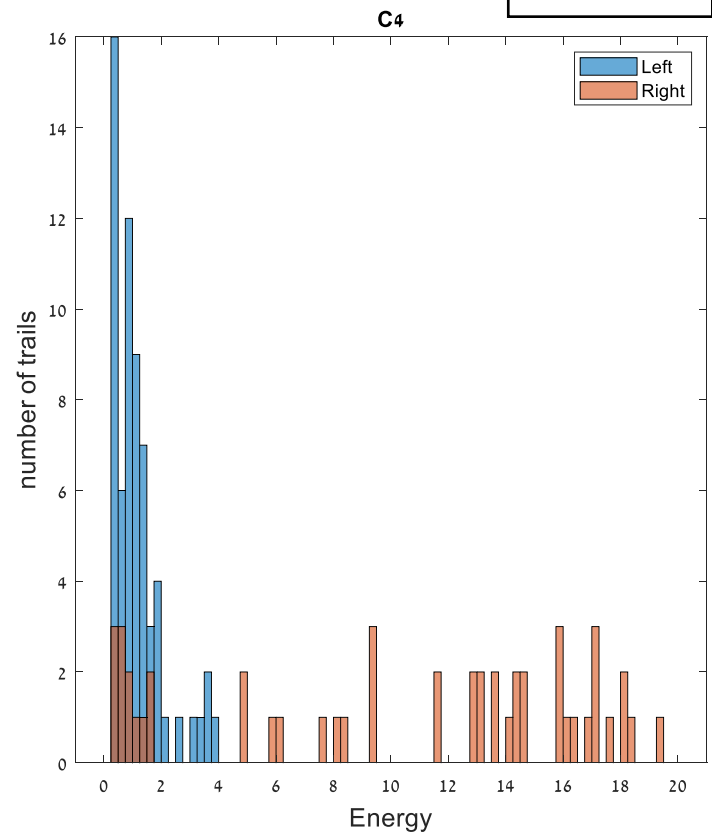
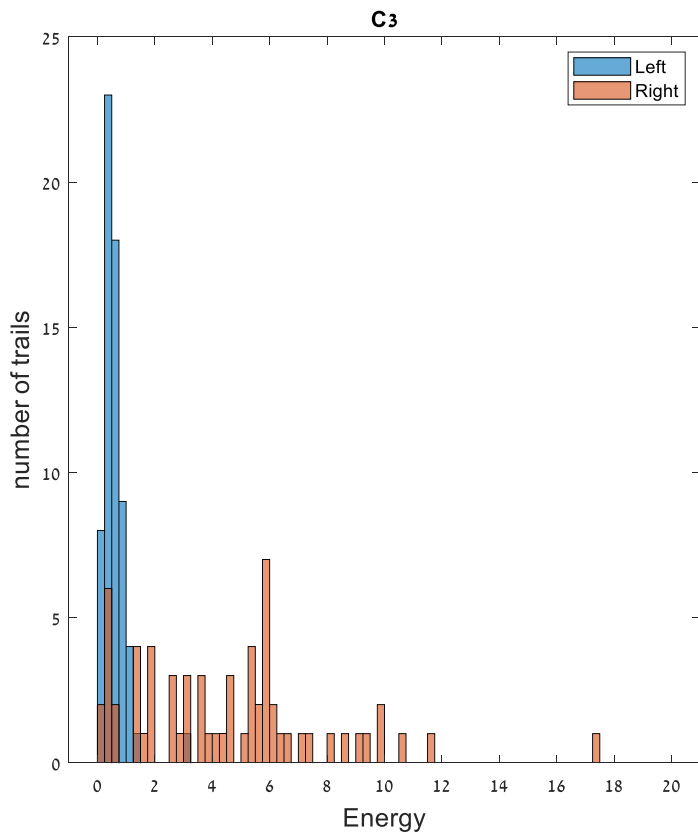


Band power

על בסיס בחינת הספקטרוגרמות השונות ניתן לבחון איזורי זמן וטווחי תדרים בהם קיים שוני במדד ה-power בין סיגנלים שהוקלטו בתנאים שונים (ימין/שמאל). על בסיס טווחי הזמן והתדרים הנ"ל חושב מדד ה-bandpower. מדד זה מהווה את הenergy בטווח זמן ותדרים מסויים. על מנת לבחון את התפלגות מדד זה בקרב 128 החזרות הופקה היסטוגרמה עבור כל band שעבורו חושב ה-bandpower. בכל היסטוגרמה, ציר ה-x מייצג את מדד ה-energy, וציר ה-y מייצג את מספר ה-trails. כמו כן, trails שהתבצעו בתנאי דמיון מוטורי של יד שמאל מסומן בכחול, ו-trails שהתבצעו בתנאי דמיון מוטורי של יד ימין, מסומן באדום. באופן זה ניתן לראות כי ההתפלגויות של מדד זה בין התנאים השונים אכן שונות. מידע זה מאפשר הסקה לגבי טיבם האפשרי של מדדים אלה בתהליך ה-classification שיקרה בהמשך.

band power between 15 - 18 Hz in time between 4.5-6 seconds

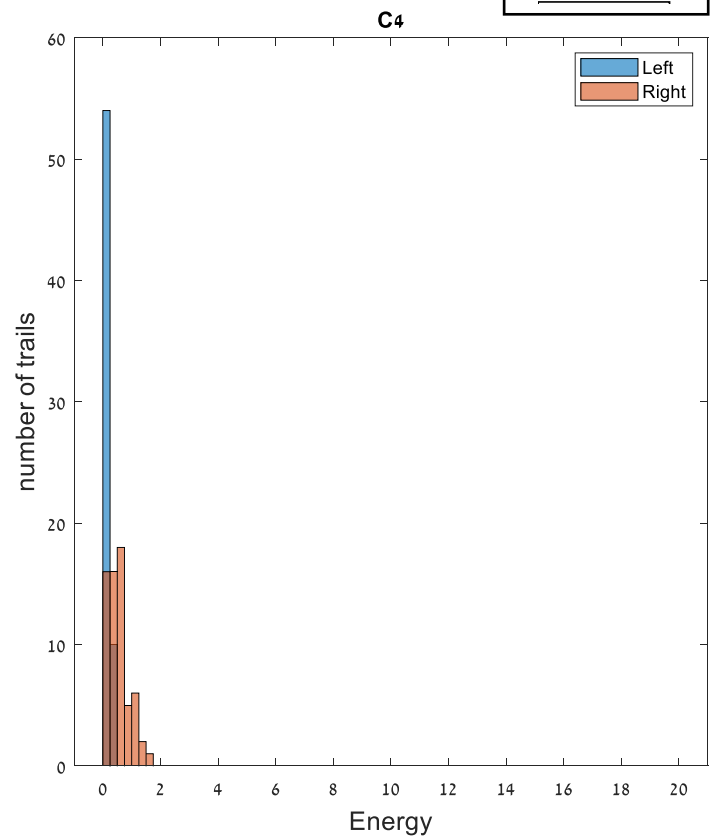
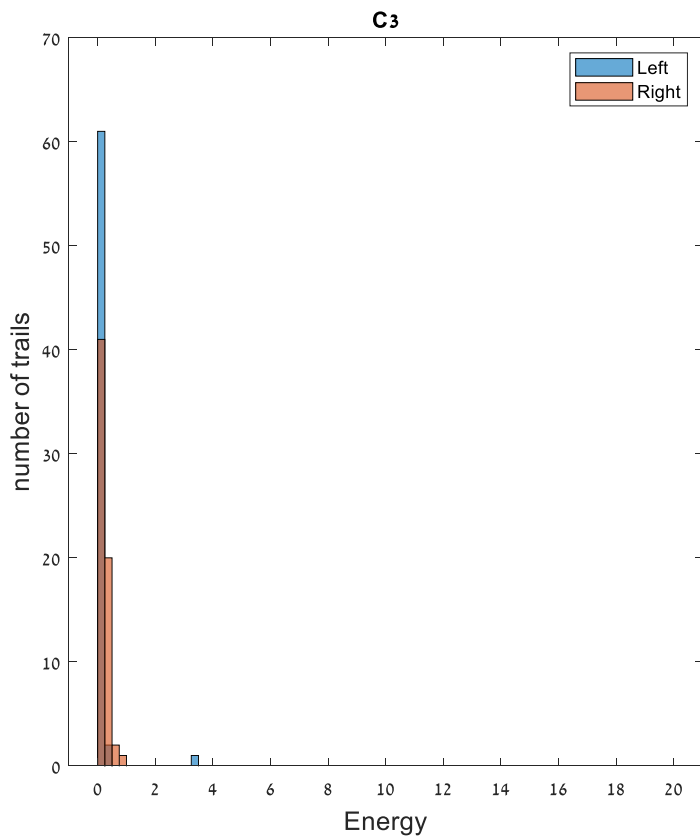
איור 4.1





band power between 30 - 35 Hz in time between 5-6 seconds

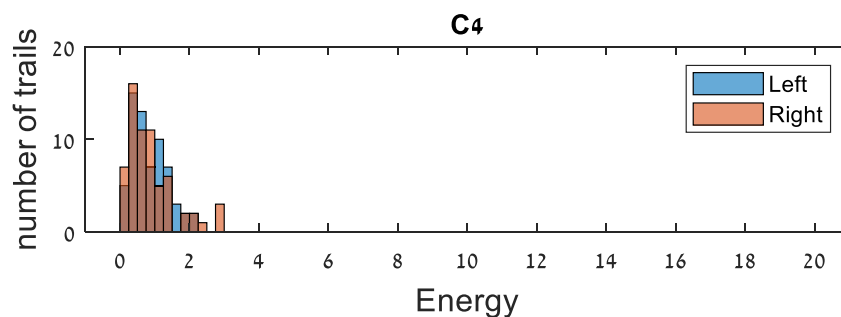
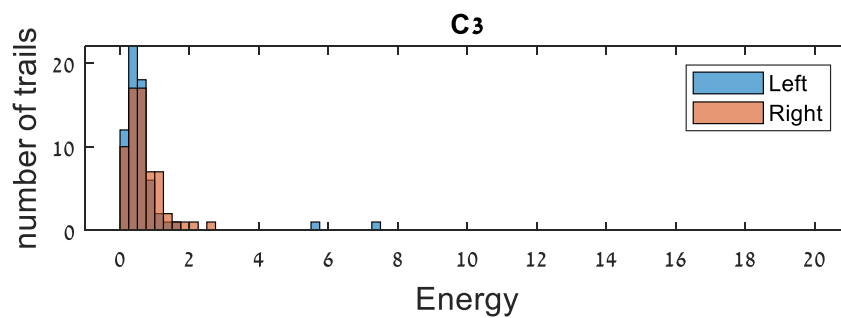
איור 4.2



לשם המחשה, ביקשנו מהפונקציה גם להנפיק היסטוגרמה עבור תחום שאינו נראה חריג בספקטוגרמה ואכן ההתפלגויות נראות דומות (ולכן פחות רלוונטיות למשימה).

band power between 22 - 27 Hz in time between 2-3 seconds

איור 4.3

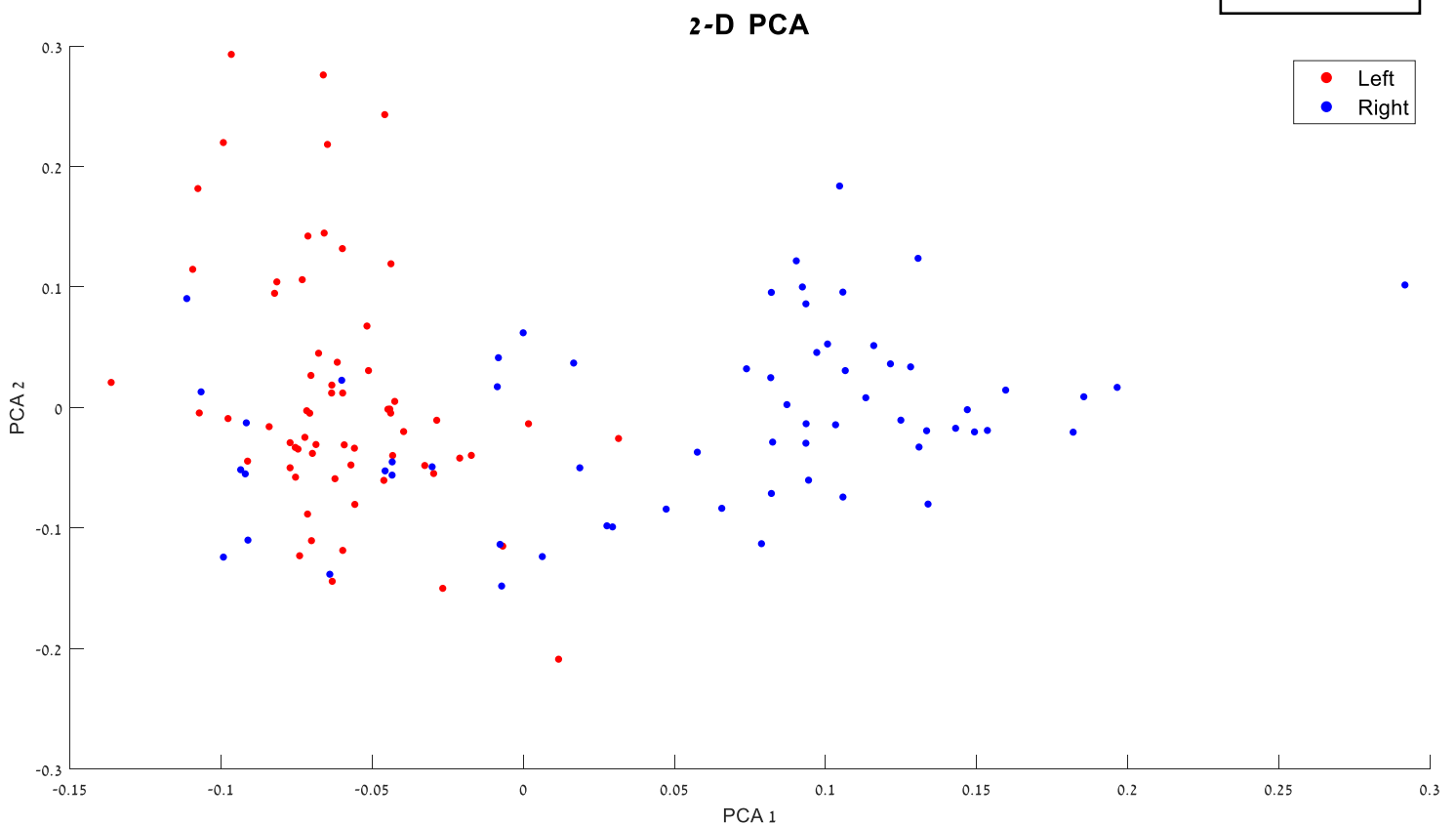




PCA

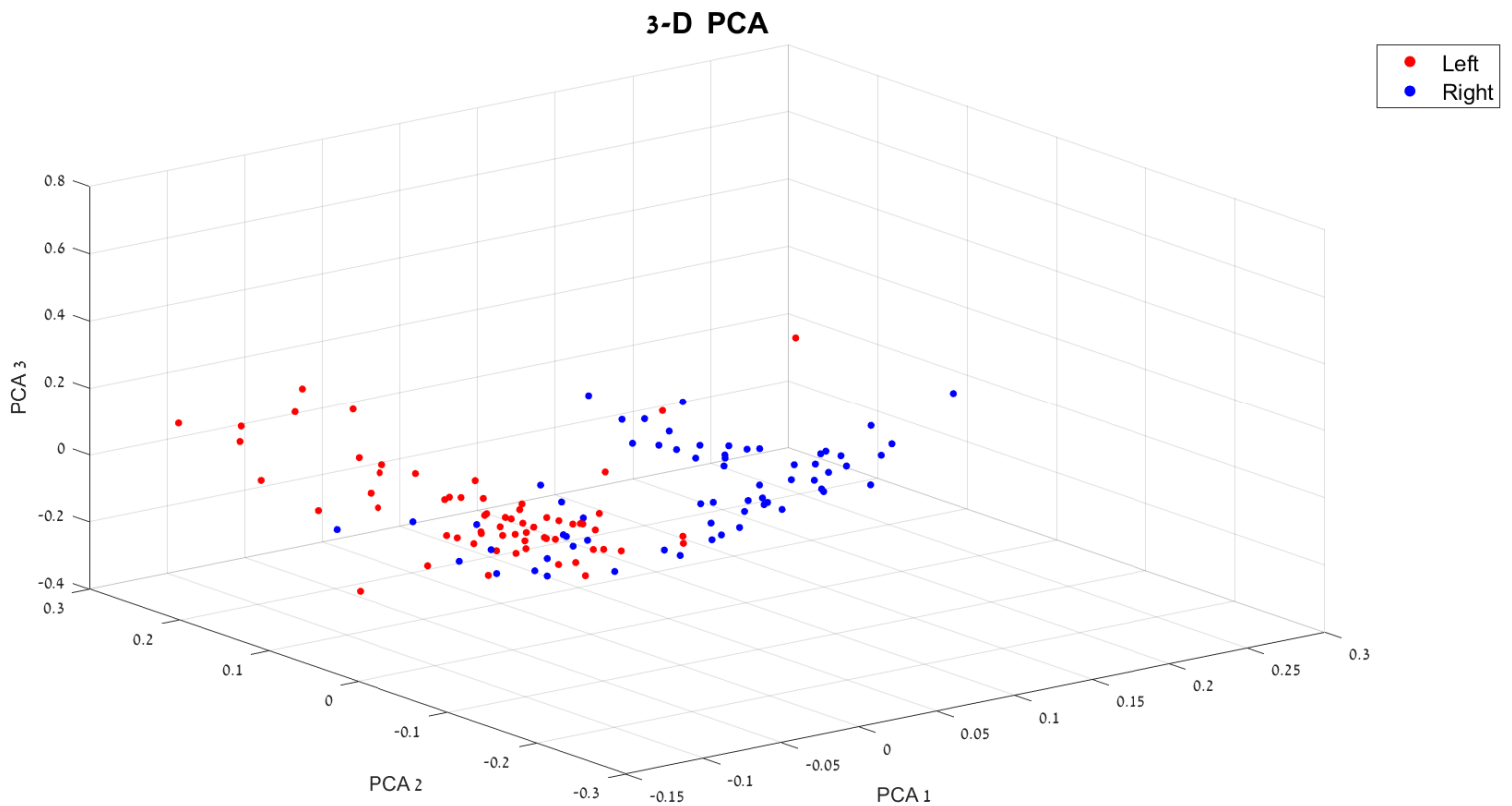
בשלב זה בוצע חילוץ ה-features הרלוונטים (band-power ו- relative power של הטווחים המתאימים) מכלל סיגלי ה-EEG. היות ומספר המימדים של מטריצת ה-features אינו מאפשר הצגה ויזואלית, הישתמשו באלגוריתם ה-PCA על מנת לדחוס את המידע לשני ולשלושה מימדים, תוך כדי שימור שונות רבה ככל האפשר. באופן זה ניתן לבחון באופן ויזואלי האם ניתן לבצע הפרדה לינארית איכותית בין התנאים השונים, באמצעות המידע שנאסף ב-features לאחר דחיסתו. כמו כן, הנקודות על פני הגרפים המייצגות הקלטות שנעשו בתנאי שמאל מסומנות באדום, ונקודות המייצגות הקלטות שנעשו בתנאי ימין מסומנות בכחול. ציר ה-X של כל גרף מציג את ה- principal component את השונות הרבה ביותר, ציר ה-Y את הבא אחריו. בגרף התלת מימדי, ציר ה-Z מציג את ה- principal component שמסביר את השונות השלישית הרבה ביותר.

איור 5.1





איור 5.2



כפי שתואר בפרק השיטה, על מנת לאמן את ה-classifier ולבחון את יכולת הסיווג שלו, נעשה שימוש בשיטת k fold cross validation. ניתן לבחון את הדיוק שהתקבל בכל אחת מ-k החזרות, הן בקבוצת הניסוי והן בקבוצת ה-validation. כמו כן ניתן לבחון את ממוצע הדיוק בניבוי מעבר לחזרות, ואת סטיית התקן בין החזרות. כל המוצג בטבלה נמדד ביחידות של אחוזים.

(טבלה 1)

group \ K	1	2	3	4	5	6	7	8	mean	sd
train	92.8	91.9	92.9	90.2	93.8	93.8	89.8	89.1	91.8	1.7
validation	93.7	81.2	81.2	87.5	81.2	81.2	93.7	100	87.5	6.9

(כלל המספרים מייצגים אחוזים)

לאחר הגעה לרמת דיוק מספקת ביכולות הסיווג (ממוצע וסטיית תקן), הפעלנו את ה-classifier על קובץ נתוני ה-test. הניבוי שהקוד הפיק סיפק 87.5% דיוק ביחס לתיגים הנכונים כפי שהן מופיעים ב-moodle.



דין ומסקנות

באופן כללי, מטרת המטלה הייתה לייצר מסווג לינארי המשתמש בהקלטות מאלקטרודות C3 ו-C4 ומאפשר להבחין בין חזרות בהן הנבדק מדמיין את הרמת יד לבין חזרות בהן הנבדק מדמיין את הרמת יד שמאל. לטובת המטלה, יצרנו קוד המאפשר בחינה ויזואלית ראשונית של מאפייני ההקלטות המתויגות שבעזרתם ניתן לשער על מאפיינים רלוונטים לטובת הסיווג הדרוש ולחלצם גם מדוגמאות לא מתויגות.

ניתוחים ראשוניים

ניתוח EEG

ראשית, הפקנו מערכות צירים של מספר הקלטות EEG שהתבצעו תחת כל אחד מהתנאים (איור 1). הצגה ויזואלית זו נועדה לאפשר מבט ראשוני על הנתונים בצורתם הגולמית.

באופן תיאורטי, היות ומיקום אלקטרודה C3 מקליטה אותות מהאזור המוטורי בהמיספירה שמאל (האחראי על צד ימין בגוף) ואלקטרודה C4 מקליטה אותות מהאזור המוטורי בהמיספירה ימין (האחראי על צד שמאל בגוף), הנחנו כי בבדיקה של ההקלטות המתויגות, נראה מתח או פעילות גבוהים יותר באלקטרודה המתאימה לצד המתויג (ימין - C3 ושמאל - C4).

התוצאות שהתקבלו היו בניגוד להשערותנו. ניתן לראות כי באיור 1.1 (יד שמאל) כי הפעילות המוקלטת בשתי האלקטרודות זהה יחסית ובאיור 1.2 ניתן לראות כי הפעילות המוקלטת באלקטרודה C4 אף חזקה מ-C3 (הכיוון ששיערנו היה הפוך).

על אף כי השערותנו הראשונית לא אוששה, ניתן להבחין באופן כללי בדפוס חוזר מסוים. בטווח הזמנים בין 4-6 שניות (זמן הדמיון בקירוב) בהקלטות המתויגות כיד ימין (איור 1.2) ניתן לראות כי הפעילות המוקלטת באלקטרודה C4 חזקה יותר. בהמשך הדיון נראה כי טווח זמנים זה אכן מהווה מאפיין משמעותי לסיווג ההקלטות.

ניתוח Power spectrum

מתוך הבנה כי יתכן והתוצאות המוצגות באיור 1 נבעו מרעש רב ועל מנת לספק יכולת הכללה גדולה יותר מהתבוננות בכל הקליטה בנפרד, בחרנו לבחון ה power spectrum הממוצע של הקלטות שבוצעו על ידי אלקטרודות שונות ותיוגים שונים. התוצאות שהתקבלו בשלב זה מתוארות באיור 2. באופן כללי ניתן לראות דמיון גבוה יחסית בין התיוגים והפעילות באלקטרודות ובכל זאת, באופן ספציפי יותר, ניתן לסמן שני טווחי תדרים המעוררים עניין בגרפים אלו: 1. 15Hz-20Hz ו-2. 30Hz-40Hz. טווחי התדרים בהם ניכרת שונות בין ה- Power spectrums עשויים להעיד על טווח תדרים שיאפשר הבחנה בין התיוגים השונים.

נציין גם כי ממצא זה מתיישב עם הממצאים הקודמים שהצגנו (כי הפעילות המוקלטת באלקטרודה C4 בתנאי ימין חזקה יותר). ניתן להבחין כי בעוד עבור צד שמאל, קיים דמיון גבוה יחסית ב power spectrum של C3 ו-C4 (איורים 2.1 ו-2.2 גרף אדום) בבדיקה של צד ימין ניכר הבדל משמעותי ב power spectrum ועוצמתו ב-C4 גבוהה מעוצמתו ב-C3 (פיק גבוה יותר בטווח בין 15Hz-20Hz).

ניתוח הספקטרוגרמות וההיסטוגרמות

לאחר הניתוחים הקודמים, היה חסר ניתוח המתייחס לשינוי במדד ה- power של התדרים השונים בסיגנלים בהתאם לשינוי במימד הזמן. על מנת לבחון את המאפיין הנ"ל ניתן להתבונן באיור 3 המציג את הספקטרוגרמות הממוצעות של ההקלטות שבוצעו באלקטרודות ובתנאים שונים. לשם בחינה פשוטה ומדויקת יותר של טווחי הזמן והתדרים העשויים לעורר עניין, ניתן לבחון את ספקטרוגרמות ההפרשים (איורים 3.2 ו-3.3). בספקטרוגרמות הנ"ל ניתן לראות מספר טווחי זמן ותדר, בהם יש שוני ברור בין התנאים השונים באותה אלקטרודה. הבולט ביותר הוא טווח התדרים 15Hz-20Hz בזמנים שבין 4.5-6 שניות, הן באלקטרודה C3 והן באלקטרודה C4 (איור 3.2). דרך נוספת לבחון את השוני הקיים בטווחים הללו, הוא באמצעות חישוב מדדי ה-energy בטווחים בכל אחת מהסיגנלים שהוקלטו



והצגתם בהיסטוגרמה (איור 4.1). ניתן להבחין בהצגה זו כי התפלגויות מדד ה-energy בתנאים השונים אכן שונות למדי.

איזור מעניין נוסף בבחינה של איור 3.2 (אם כי פחות בולט), הינו טווח התדרים 30Hz-35Hz בזמנים שבין 5 ל-6 שניות. גם במקרה זה בחנו את את השוני הקיים בין התנאים באמצעות חישוב מדד ה-energy בטווח הנ"ל בכל אחד ממדידות ה-EEG (איור 4.2). ניתן להבחין בהצגה זו כי התפלגויות מדד ה-energy בתנאים השונים שונות, אך לא כמו השוני שנמצא בטווחים שהוצגו קודם. במקרה זה, ניתן לראות כי ההתפלגויות שונות במיוחד בהקלטות שבוצעו ב-C4.

כמו כן, ניתן לזהות הבדל במדד ה-power בין התנאים השונים גם בטווח התדרים שבין 15-20 בזמנים שבין 2 ל-3 שניות באלקטרודה C4, ובטווח התדרים 8-12 בזמנים שבין 4.5 ל-5.5 שניות באלקטרודה C3. מאפיינים אלו היו מובהקים פחות ולכן בחרנו שלא להתמקד בהם בנינוח. בבחינה מאוחרת יותר כאשר בחנו את השפעתם על יכולת הסיווג התגלה כי אכן מדובר במאפיינים בעייתיים הגורמים בין היתר לפגיעה באחוזי ההצלחה (פירוט נוסף בהמשך).

PCA

על מנת לבחון את טיב היכולת להפרדה לינארית של המאפיינים, בחרנו לצמצם את מימדי המאפיינים באמצעות אלגוריתם PCA (principal component analysis). אלגוריתם זה מאפשר לבצע דחיסה של המידע למימדים נמוכים יותר תוך איבוד מינימלי של מידע. דחיסה למימדים נמוכים יותר מאפשר ויזואליזציה פשוטה יותר ובחינת היתכנות של טיב היכולת להפרדה לינארית.

באיור 5.1 ניתן לצפות בשני המימדים הדחוסים בעלי השונות המרבית הגדולה ביותר ובאיור 5.2 ניתן לבחון את יכולת ההפרדה בתלת-מימד עבור שלושת המימדים בעלי השונות המרבית הגדולה ביותר. ניתן לשים לב כי באופן כללי ניתן להפריד באופן לינארי בין שתי הקבוצות.

בחירת מאפיינים

באופן כללי, היו מספר קווים מנחים עיקריים באופן בחירת המאפיינים ובמספרם. מהצד האחד, רצינו לבחור מספר גבוה מספיק של מאפיינים אשר בעזרתם נוכל לסווג באופן המדויק ביותר את הדוגמאות. מהצד השני, מספר רב מדי של מאפיינים עלול היה להעמיס על המערכת (זיכרון וזמן ריצה) ואף לפגום ביכולת הניבוי הסופית של המערכת בעקבות יצירת over-fitting על קבוצת האימון. תחת ההתחשבות בקווים אלו, בחרנו להכניס את המאפיינים אשר בהם זיהינו במהלך הניתוח הראשוני את רמות השונות הגבוהות ביותר בין הצדדים בתקווה כי מאפיינים בודדים אלו יספיקו על מנת לייצר למערכת יכולת סיווג מספקת.

בחירתנו הסופית של המאפיינים שחולצו מתוך הדוגמאות היתה כדלקמן:

1. Powerband בשני תחומים עבור כל אחת משתי האלקטרודות (סה"כ 4 מאפיינים):
 - a. טווח תדרים: 15Hz-18Hz, זמנים: 4.5-6 שניות (2 מאפיינים – 1 עבור כל אלק').
 - b. טווח תדרים: 30Hz-35Hz, זמנים: 5-6 שניות (2 מאפיינים – 1 עבור כל אלק').
2. Relative power עבור כלל סוגי הגלים (סה"כ 12 מאפיינים, 6 עבור כל אלק').

נציין כי בעוד הבחירה בpower band שבחרנו טריוויאלית יחסית בהתייחס לתוצאות שהצגנו לאורך העבודה, הבחירה בחילוך מאפיינים מסוג relative power אינה מתבקשת. לאחר בחינה של מספר רב של מאפיינים שונים, אשר רובם פגעו ביכולת הניבוי, בחרנו בסוג מאפיינים לאחר שראינו כי אחוזי הדיוק משתפרים במעט ומאפיין זה כללי יחסית ו"משלים" את התמונה שאיננה נכללת בתחומי ה-power band.

באמצעות מאפיינים אלו הגענו לרמות דיוק המפורטות בטבלה 1 ובאופן כללי לרמת דיוק בקובץ המבחן של 87.5%.

הגענו לבחירה סופית זו לאחר ניסיונות רבים ומגוונים. נדגים את מקרי הקצה באמצעות יכולות המערכת עבור מאפיין אחד ומאפיינים רבים מדי (נציין גם כי כלל החישובים מבוססים על $k=8$ בשיטת k fold cross validation עליה יפורט בהמשך).



מאפיין בודד

לצורך בדיקה, הרצנו מאפיין בודד מתוך קבוצת המאפיינים הסופית ובדקנו את יכולת הניבוי המשוערת של המערכת עבור מאפיין אחד. עשינו בדיקה זאת עבור מאפיין a.1 רק עבור אלקטרודה C3. במצב המתואר לעיל, אחוז הדיוק המשוער הממוצע (=יכולת ההצלחה על קבוצת ההצלה) הינו $82.81 \pm 8.12\%$.

על אף כי מדובר באחוז דיוק גבוה מאוד יחסית למאפיין אחד, נציין כי מאפיין זה אינו נבחר באופן מקרי ומייצג שונות גבוהה שמצאנו בחלקים הראשונים של העבודה. לכן, על אף אחוז ההצלחה הגבוה, איננו היינו מופתעים.

ריבוי מאפיינים

לצורך בדיקה, הרצנו מספר רב של מאפיינים (30) מקובלים לניתוחי EEG. בניתוח של אחוזי הדיוק התקבל $93.97 \pm 1.12\%$ על קבוצת האימון, אך במדד הרלוונטי יותר, המתייחס לאחוז הדיוק הממוצע עבור קבוצת הבדיקה התקבל $58.33 \pm 6.59\%$ בלבד. תוצאות אלה מעידות בבירור על מצב של over-fitting על סט האימון – מצב ממנו ניסינו להימנע.

בחירת K

בחירת K לשימוש באלגוריתם k fold cross validation מהווה חלק חשוב וקריטי בעבודה. בחירת K מדויקת יאפשר ניבוי מהימן של יכולת סיווג המערכת עבור דוגמאות שטרם פגשה. מהצד האחד, בהתאם לספרות המקצועית, K קטן מדי גורם להערכה לא מהימנה של יכולות סיווג המערכת. מנגד, K גדול מדי מייצר עומס רב על המערכת ולפי הספרות אינו מומלץ לשימוש עקב תופעות סטטיסטיות שונות.

בטבלה הבאה ניתן להבחין בתוצאות הניבוי של המערכת עבור ערכי K שונים.

(טבלה 2)

k	2	4	8	16	32	64	128
mean train	94.5	91.2	91.8	91.8	91.8	91.7	91.6
sd train	0.8	2.4	1.7	0.08	0.06	0.5	0.04
mean valid	88.3	89.1	87.5	42.9	21.7	10.7	5.4
sd valid	0.8	3.4	6.9	7.3	4.7	2.8	2.1

(כלל המספרים פרט לא מייצגים אחוזים)

נשים לב כי עיקר השוני בטבלה עבור ערכי k שונים מתבטא ב 'mean valid' וב 'sd valid'. נציין גם כי אלו המדדים הרלוונטיים המנבאים את יכולת ההצלחה של המערכת על סט דוגמאות לא מוכר. קל לראות, כי מעל $k=8$ אחוז הדיוק הופך לא מהימן (מתרחק ממוצע האימון וההצלחה בפועל).

לגבי $k=2$, ניכר כי חלוקה לשתי קבוצות בלבד אמנם מציגה נתונים מבטיחים (שונות קטנה וממוצעים קרובים יחסית) אך במצב כזה ישנן רק 2 בדיקות ולכן המיצוע לא משקף באופן מהימן את הביצועים בסט דוגמאות לא מוכר.

בשלב זה נותרה השאלה האם לבחור $k=4$ או $k=8$. בהתבסס על הספרות המקצועית בתחום, קיימת המלצה די גורפת שבהיעדר ידע מדויק לגבי סט הנתונים, מומלץ לבחור $k=10$. בהסתמך על המלצה זו, על אף כי השונות ב $k=4$ קטנה יותר בחרנו להשתמש ב $k=8$.

בחינת ההצלחה והצעות להמשך

בהרצת המסווג על קובץ המבחן התקבלו 87.5% דיוק. מסקירה של עבודות נוספות בתחום, ניכר כי אכן רף 90% הינו מקובל יחסית למידע המתקבל מהאלקטרודות הנ"ל בלבד. בדיעבד, הסתבר לנו גם כי בחירת ה-5 הייתה מוצלחת למדי מאחר ונבאה באופן מדויק לחלוטין את ההצלחה בסט הדוגמאות החדש.



במחקרים הבאים בתחום, אנו חושבים שיש מקום לבחון הקלטות מאלקטרודות נוספות פרט לאלו שהוזכרו בעבודה זו. אלקטרודות מאזורים שאינם בהכרח קשורים לתנועה המוטורית עשויים לסייע בסיווג מדויק יותר באמצעות מאפיינים שטרם נחשפנו אליהם או אפילו רק באמצעות ניקוי שונות לא רלוונטיות מההקלטות באזור המוטורי.

בנוסף, מאחר והשימוש הקליני בסוג מסווג כזה מיועד בעיקרו לקטועי גפיים, יתכן ומדידה של המתח החשמלי בקצות העצבים בגף הקטועה יכולה גם היא לשרת את מטרת העל - הפעלה מוטורית באמצעות מחשבה בלבד ובכך לשפר את אחוזי הדיוק של המערכת.

לסיכום, במטלה זו יצרנו מסווג המפריד בין מחשבה על הרמת יד ימין לשמאל וזאת באמצעות הקלטות EEG מאלקטרודות C3 ו-C4. עשינו זאת תוך בחינה מעמיקה של ההבדלים באותות בין התיוגים השונים ובין האלקטרודות השונות. בהמשך, באמצעות שיטות סטטיסטיות, בחנו ואימנו את המסווג תחת תנאים שונים ולבסוף בחנו את רמת ההצלחה שלו בדוגמאות לא מוכרות והגענו לאחוזי דיוק גבוהים יחסית ומקובלים בתחום.