PROJECT 3

Andrea Rachele Aparo, Virginia Furlani, Francesco Pio Monaco, Anna Pegreffi, Lorenzo Sterzi

INTRODUCTION

Brain computer interfaces (BCIs) allow to decode a user's intentions, starting from neural activity, and execute them using external devices. Indeed, brain activity produces electrical signals that can be detected through the cortical surface, or with electrodes placed in the brain itself. BCIs transform these simple reflections of brain activity, communicating the user's intention to the outside world. Because they allow the use of control and communication channels that do not depend on the brain's normal response channels, such as nerves and muscles, they are often used with people with severe neuromuscular disorders. The BCI's techniques can be invasive or noninvasive. The former requires the use of electrodes directly implanted in the brain, which allow the detection of neuronal action potentials or local field potentials. Invasive BCI has been studied primarily in nonhuman primates and presents significant technical difficulties as well as obvious clinical risks. On the other hand, noninvasive BCI allows the use of electroencephalographic (EEG) activity recorded through the scalp. It has been demonstrated [3] that the use of noninvasive techniques yields results comparable in time, precision and accuracy to those of invasive BCI, while limiting the risks and difficulties of use.

Noninvasive EEG-based BCI is already commonly used for simple control and communication tasks. In particular, the combination of this technique with the recording of EEG signals during imaginary motor commands (Motor Imagery based BCI), has made it possible [2] to study the restoration of damaged or lost brain functions, eventually leading to direct control of external devices. This result is possible due to the fact that mental imagination of movements causes changes in the neural activity of primary sensorimotor areas quite similar to those that can be observed during the execution of real movement.

Imagery of different types of movements (e.g., right hand, left hand, or feet), can change the functional connectivity within the cortex and results in an amplitude decrease (event-related desynchronization, ERD) or an increase in amplitude (event-related synchronization, ERS) of the Rolandic μ rhythm (7-13 Hz) and the central β rhythm (13-30 Hz). Both rhythms originate in the sensorimotor cortex and represent the two oscillation types of greatest interest in BCI.

It has been observed [2] that ERD during mental simulation of movement remains mostly limited to the contralateral hemi-sphere, so suppression of μ and central β rhythms is more pronounced when subjects imagine movements than when they actually perform them. This observation justified the use of motor imagery as a control strategy to achieve electrocortical responses.

Several studies [2][4] have presented protocols involving multiple training sessions, consisting of both calibration phases and online experiments. The need for a calibration phase enables proper use of the BCI; in fact, it is essential that participants learn to voluntarily modulate EEG oscillatory rhythms by performing certain tasks, and that the

system learns which patterns are characteristic of the subjects, thus creating specific classifiers.
This mutual learning methodology between BCI and subject could strongly promote the acquisition of skills for users to use the interface and lead to a system that can succeed in real-world scenarios [5].
Subjects therefore play an equally important role, and most of the literature is based on the analysis of healthy subjects; however, it has been pointed out [4] that the study of BCI should also be extended to subjects with disabilities. Indeed, such studies have shown that in this case patients were able to obtain good performance, sometimes even better [3] in comparison with healthy subjects, ensuring success in applications.
The project we present has several aspects in common with those previously introduced; in fact, starting from data collected on healthy subjects, the whole BCI loop is simulated. Specifically, subjects are asked to perform 2 imaginary motor tasks, both during the calibration phases and during online runs. The data obtained were then analyzed with the aim of creating a specific classifier on the extracted features for each subject. As reported in the previously cited studies [2] the speed of information transfer is highly dependent on classification accuracy, and therefore, high classification accuracy is critical for good performance.
Therefore, this work echoes what was previously introduced on the mutual learning between BCI and subject, and provides an evaluation of the effectiveness of the created classifier.

METHODS

Subjects and data. Eight healthy subjects were involved in the work presented here. Data were recorded over 3 days, using a 16-channel EEG amplifier (g.USBamp, g.Tec) at a frequency of 512 Hz. *Figure1* shows the placement and order of the electrodes, which were placed according to the international 10-20 layout. Each subject did at least two days of recording; the first was dedicated to performing 3 offline runs, for the calibration phase, and 2 online runs, while on the second day they did 2 online runs each.

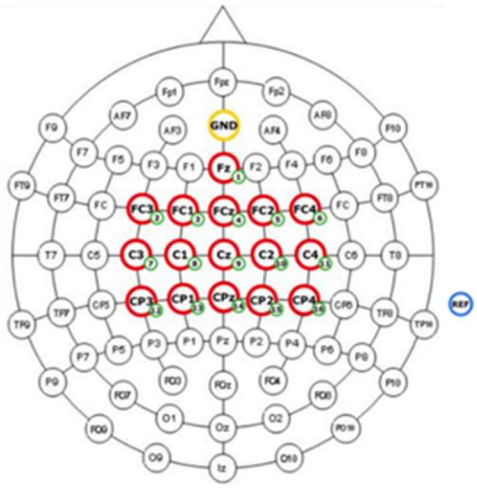


Figure 1 - 10-20 international layout used to place the electrodes

Protocol. During both online and offline phases, subjects were asked to perform two imaginary motor tasks, i.e., move both hands or both feet, or rest, following what was indicated by a visual paradigm (*Fig.2*). During calibration, the feedback associated with the visual paradigm was moved automatically in the correct direction, while during online runs this was moved according to the output of the classifier.

Figure 2 - Visual paradigm. The color of the cue indicated which motor imagery task to perform.

DATA ANALYSIS

Initially the data were processed and filtered, and the main properties were evaluated by considering the ERD/ERSs of the signals during the tasks.

Point 1.

EEG data from each subject were imported and organized into structures. For this purpose, the *conc_sload* function was used, which loads the EEG data contained in the input file, sorting them, and giving as output 2 arrays, denoted in the code as *si* and *hi*. These arrays are structured to contain within them, respectively, the EEG data from the different trials, as far as *si* is concerned, while *hi* contains the main information regarding the different EEG recordings. The data extracted from the function are then saved in two structures, *S* and *H*, for all patients. Please note that the same concatenated data were saved in the *S_conc* and *H_conc* structures.

```
[si,hi,si_conc,hi_conc] = conc_sload(eeg_sbj_i,1);

%Saving within different structures of the data extracted by conc_sload
[S.(fields_S{d})] = si;
[H.(fields_H{d})] = hi;
[S_conc.(fields_S_conc{d})] = si_conc;
[H_conc.(fields_H_conc{d})] = hi_conc;
```

It's underlined that at this stage of data organization the subjects were identified as:
ai6 = s1 , ai7 = s2 , ai8 = s3 , aj1 = s4 , aj3 = s5 , aj4 = s6 , aj7 = s7 , aj9 = s8 .
Next, ERD and ERS were calculated using the function *Computing_ERD*.
This function takes as input the concatenated data matrices computed previously si_conc and hi_conc, and the labels used to identify subjects. In output it returns graphs of ERD/ERS over time, and their topoplots.
In order to highlight some components, the signal was processed with different filters: at first, a Laplacian filter is applied to improve spatial localization, and consequently the signal is filtered in the mu and beta bands using a 5^{th} order Butterworth filter. Next, the moving average was applied (1-second window), and finally a logarithmic transformation. The choice of using the Laplacian filter was justified considering what has been reported in the literature [6], where the usefulness of data-independent spatial filtering is highlighted, especially when using a system calibration approach on specific subject characteristics.

Since the calculation of ERD/ERS must be performed trial by trial, the trials were divided. To do this, events (contained within the previously derived *h* structure) were initially extracted to create the matrix *trialdata_mu* and *trialdata_beta*. The activity period was considered as the period marked as continuous feedback and the reference period as the period marked as fixation cross. The two matrices were thus calculated, where each trial starts from the event related to the fixation cross to the end of the event related to the continuous feedback.

```
%%Creation of the matrix trialdata_mu and trialdata_beta [samples x channels x trials]
start_pos=h.EVENT.POS(h.EVENT.TYP == 786);
ck_pos=h.EVENT.POS(h.EVENT.TYP == 781);
ck_dur=h.EVENT.DUR(h.EVENT.TYP == 781);

end_pos=ck_pos+ck_dur -1;
durata=end_pos-start_pos+1;

length_min=min(durata);
channels=size(s,2);
ntrials=length(start_pos);

trialdata_mu=zeros(length_min,channels,ntrials);
trialdata_beta=zeros(length_min,channels,ntrials);

for t=1:ntrials
    trialdata_mu(:,:,t) = log_mu(start_pos(t):(start_pos(t)+length_min-1),:);
    trialdata_beta(:,:,t) = log_beta(start_pos(t):(start_pos(t)+length_min-1),:);

end
```

fixdata_mu and *fixdata_beta* were created in the same way and they were only related to the fixation period.

From these, the baseline period and ERD were calculated for both frequency bands. To calculate the latter, the percentage was used to optimize the results.

```
%Computing the ERD/ERS for mu and beta band
Baseline_mu=repmat(mean(fixdata_mu), [size(trialdata_mu,1) 1 1]);
ERD_mu=100*(trialdata_mu-Baseline_mu)./Baseline_mu;
Baseline_beta=repmat(mean(fixdata_beta), [size(trialdata_beta,1) 1 1]);
ERD_beta=100*(trialdata_beta-Baseline_beta)./Baseline_beta;
```

To display the data we decided to perform a temporal visualization and the *topoplots*; after selecting the most meaningful channels, the signals were averaged in the fixation period and in the continuous feedback period. These steps were performed for both MI tasks and for both frequency bands.

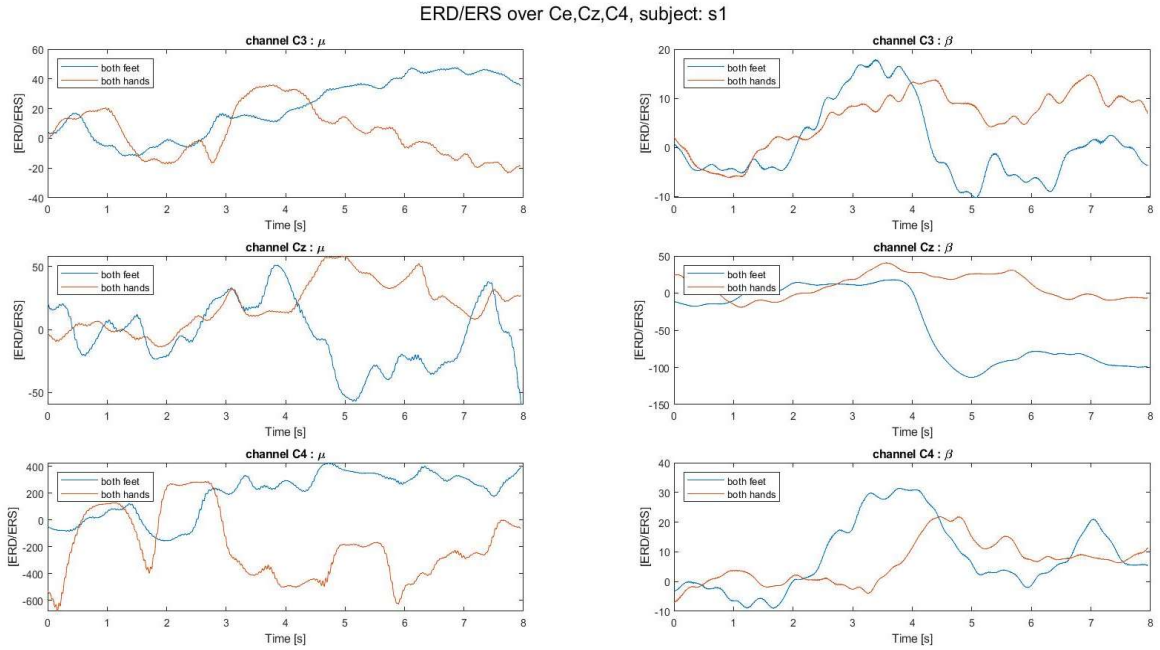


Figure 4 - Temporal visualization of ERD/ERS for subject 1, considering C3, Cz and C4 channels and the two MI movements 'both hands' and 'both feet'.

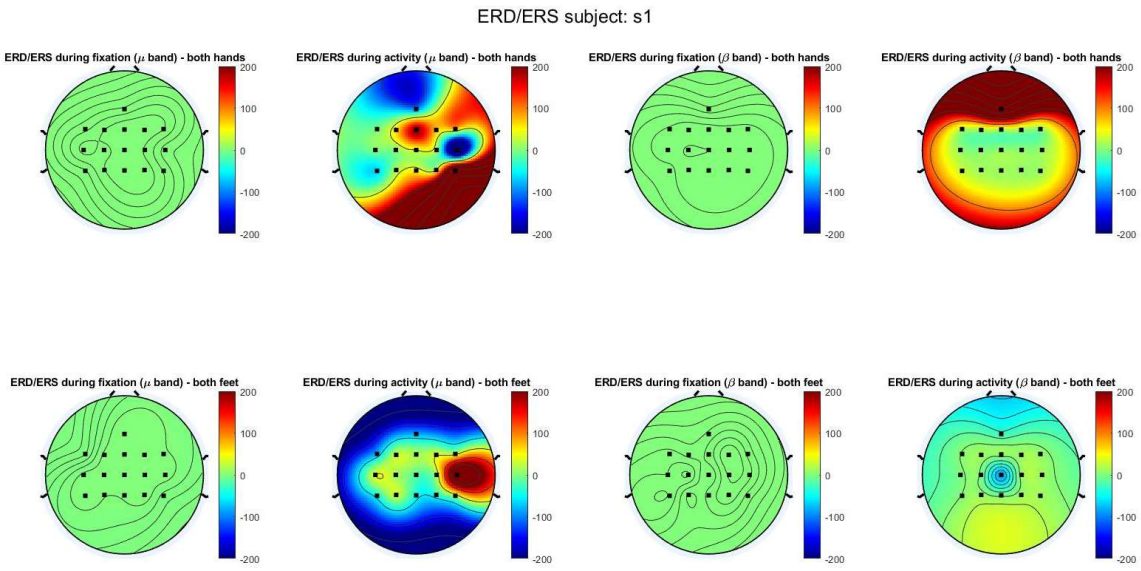


Figure 3 - Data visualization using topoplot for subject 1.

Through such visualization of EEG data, it was possible to provide an assessment of signal characteristics for each subject; in particular, it is observed that during the "both feet" MI task there are significant ERDs for the beta rhythm in central scalp areas, and similarly for the "both hands" MI task. β and μ ERDs/ERSs are also observed on the bilateral motor areas for the two MI tasks.

PSD is estimated to extract the powers of EEG signals at different frequencies. These steps are performed within the *lap_psd* function; the latter requires as input the previously derived structures, the matrix containing the Laplacian filter mask and the minimum and maximum frequencies. The output provides the *PSD* matrix, characterized by the 3 dimensions

[windows x frequencies x channels], the *f_ sel* vector containing the selected frequencies, and the *h_PSD* matrix calculated during PSD estimation. The extracted data are then saved within the *h_psd* and *PSD* structures.

Point 2a.

The focus of the second part of the work is on analyses on BMI decoding on each subject. Initially, the calibration was performed considering only the offline data, representing the "trainset", for each subject.

First, PSD is estimated to extract the powers of EEG signals at different frequencies. These steps are performed within the *lap_psd* function; the latter requires as inputs the previously derived structures *s* and *h*, the matrix containing the Laplacian filter mask and the minimum and maximum frequencies. The output provides the *PSD* matrix, characterized by the 3 dimensions [windows x frequencies x channels], the *f_ sel* vector containing the selected frequencies, and the *h_new* matrix calculated during PSD estimation. The extracted data are then saved within the *H_psd* and *PSD* structures.

Starting from the *PSD* matrix and the structure of the concatenated events (TYP, POS, DUR), trial extraction was carried out, first obtaining the information concerning the "Cue" and "continuous feedback" periods, and then extracting only the windows referring to them for each class, with the use of labels.

```
s_lap = s*lap;

wlength = 0.5;              % [s]
pshift = 0.25;              % [s]
wshift = 0.0625;            % [s]
samplerate = h.SampleRate;
mlength = 1;                % [s]

[PSD, f] = proc_spectrogram(s_lap, wlength, wshift, pshift, samplerate, mlength);
```

```
for j = 1:hi.OfflineSessions

    sj = si.(fields_sj{j});
    sj = sj(:,1:16);
    hj = hi.(fields_hj{j});

    [PSDj,f_sel,h_PSDj] = lap_psd(sj,hj,lap,f_min,f_max);

    [PSDi.(PSDi_fields{j})] = PSDj;
    [H_psdi.(H_psdi_fields{j})] = h_PSDj;
end

H_psdi.OfflineSessions = hi.OfflineSessions;
H_psdi.OnlineSessions = hi.OnlineSessions;
H_psdi.Sessions = hi.Sessions;

[H_psd.(fields_H_psd{i})] = H_psdi;
[PSD.(fields_PSD{i})] = PSDi;
```

The data obtained were so reshaped to obtain the 2D 'feature matrix' [windows x features] and then the Fisher score was calculated for each feature using the following formula:

$$FS(k) = \frac{abs(\mu_{C_1}(k) - \mu_{C_2}(k))}{\sqrt{\sigma^2_{C_1}(k) + \sigma^2_{C_2}(k)}}$$

μ_{C_1}, μ_{C_2} Mean for classes C_1, C_2

$\sigma^2_{C_1}, \sigma^2_{C_2}$ STD for classes C_1, C_2

It gives the information regarding the distance between the distributions of the two classes, based on the calculation of their means and variances.
Within the fisher score matrix, the rows represent the so-called observations, while the columns contain all the features we take into consideration.
The aim is to select the features that are most informative and discriminant to determine with certainty the task activated.
To do this, the graphs obtained were visually analyzed by comparing those for each run of each individual subject, and the most significant features were manually selected, i.e. those repeatedly active in all sessions. In practice, these are the features that tend more towards the color yellow in the graph (blue least active, yellow most active).

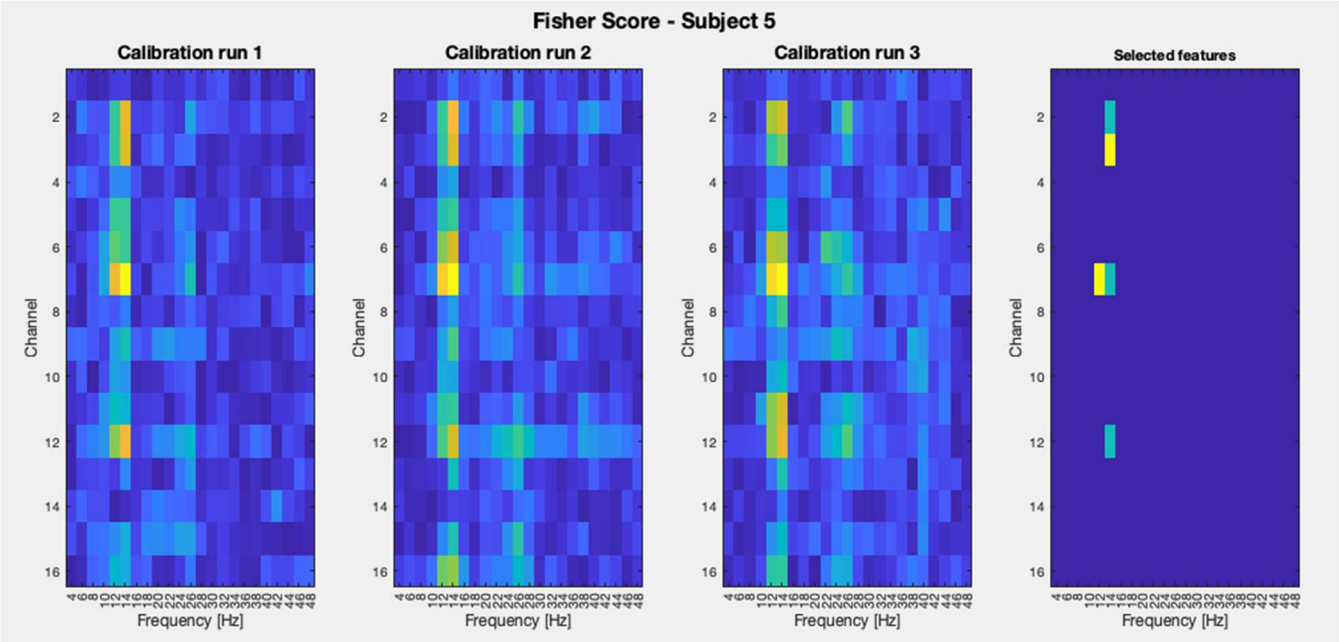


Figure 5 - Fisher Score of subject 5. From the right, there are the Fisher Score results of the 3 offline runs of subject 5. The 4th image highlights the selected features: in green are the features used to do the test with 3 features and in yellow are the 2 additions for the one with 5.

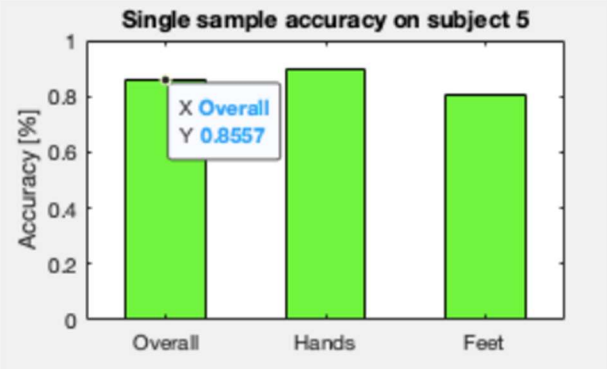
Then, we came to the stage of creating the classifier.
To do this, samples were extracted using a for loop and the function *fitcdiscr* with the specification *quadratic*. A test using *diagquadratic* was also carried out, but the results were worse than the previous ones in terms of accuracy and this alternative was therefore discarded.

```
% We initialize two vectors that will keep our samples and our labels
holder = [];
labels = Cki;

% We take the informations about the PSD
holder_big = Continuos_feedback_PSDi;
holder_size = size(holder_big,1);
di_max = size(holder_big,4);

% We repeat the labels for each window in the PSD
labels = repelem(labels,holder_size);

% We unfold the matrix trial-by-trial
for fi = 1:di_max
    sqzd = squeeze(holder_big(:,:,:,fi));
    % and we take a sample for each element in that trial

    for di=1:holder_size
        holder(end+1,:) = [sqzd(di,fr(i,1),ch(i,1)), sqzd(di,fr(i,2),ch(i,2)), sqzd(di,fr(i,3),ch(i,3))];
    end
end
```

```
Classifier = fitcdiscr(holder, labels,'DiscrimType','quadratic');

%obtain results
[Gk, pp] = predict(Classifier,holder);
```

This operation was done using 3 manually chosen features per subject and then 5 features (piece of code shown above), so that we could see which number of features was the best to obtain results with the highest accuracy.
Indeed, the classification accuracy of the two classes and the overall classification accuracy of the classifier was calculated for each individual subject.
The choice of the number of features used in the classifiers is based on the results on training data; comparing the results between 3 (Figure 6) and 5 features (Figure 7) it's possible to see a light improvement in the training accuracy for the latter, additionally from the confusion matrices we can notice a particular improvement in correctly classifying the "both feet" class.
After doing the two tests in feature space, we repeated them also in logarithmic space.
The best combination turned out to be the one in logarithmic space using five features.

The number of features used is thus 5 after verifying that using 7 features did not lead to better results and to avoid overfitting.
Finally, the classifiers of each individual subject were saved in such a way that they could be used in the evaluation of the online sessions.



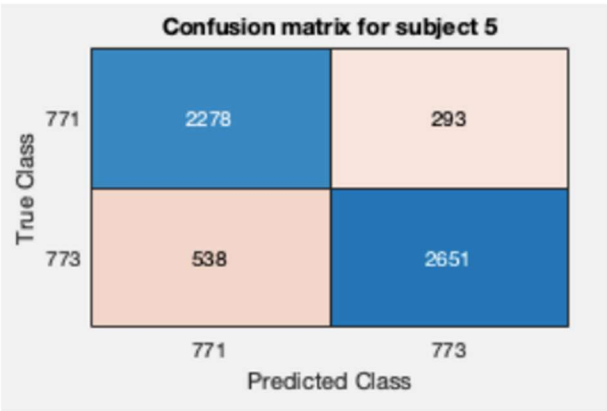
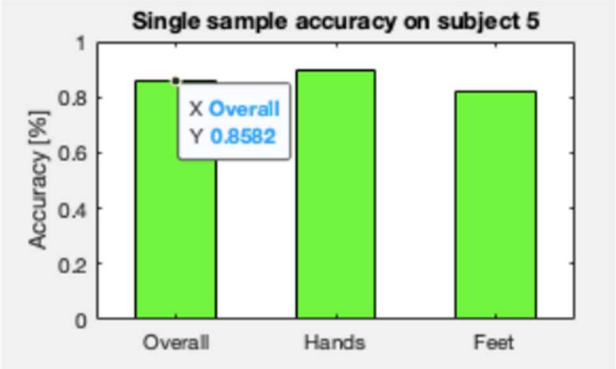


Figure 6 – Single sample accuracy and confusion matrix for subject 5, using 3 features in features space.



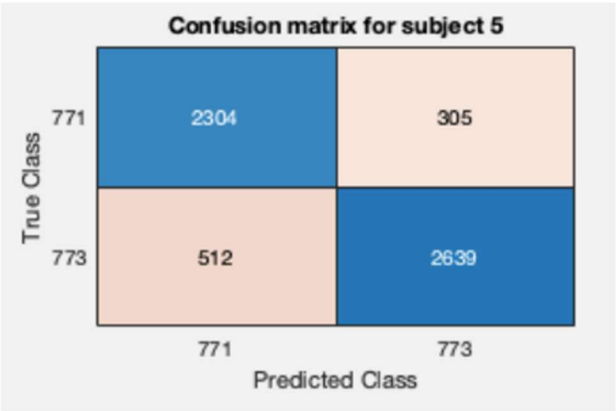
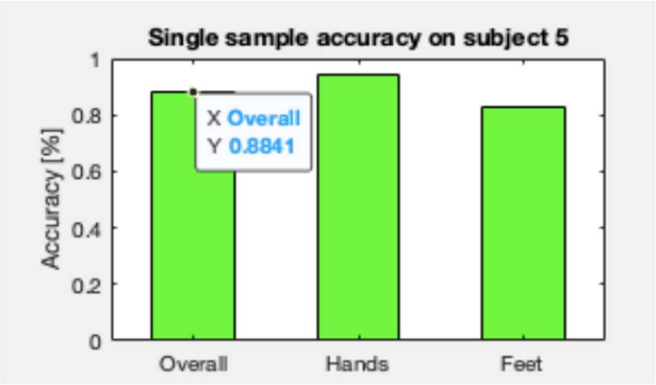


Figure 7 – Single sample accuracy and confusion matrix for subject 5, using 5 features in features space.

Point 2b.

Next, the evaluation of the online runs, also called “test sets”, was performed. The first part of the work corresponds to that executed for the offline “trainset”. In fact, the extraction of the “Cue” and “continuous feedback” runs was performed from the 3D matrix of the PSD and the event structures POS, TYP and DUR.
As far as the classifiers were concerned, those created previously through the use of the extraction of the 5 features for each subject chosen in the calibration phase with the trainset were used.



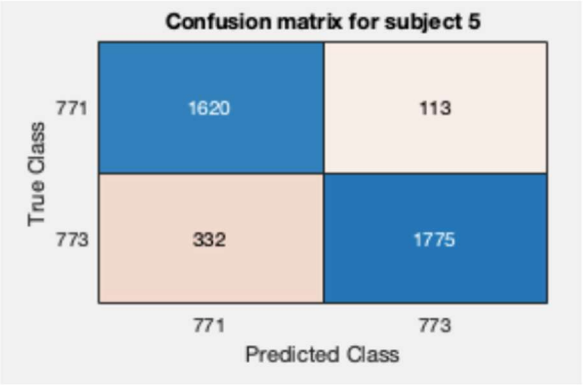


Figure 8 – Single sample accuracy and confusion matrix for online runs of subject 5, obtained using the classifier.

The posterior probability was then calculated through the use of an evidence accumulation framework based on exponential smoothing, implemented according to the following formula:

$$D(t) = D(t-1) \cdot \alpha + pp(t) \cdot (1 - a)$$

where pp(t) is the posterior probability at the window t, D(t-1) is the output of the control framework at the time t-1 and alpha is the integration parameter.

At the beginning of each trial, the value of the accumulator output, D(t), was reset to the value 0.5.
For the determination of the class of each trial, two thresholds were set: one at 0.8 and one at 0.2.
Exceeding the threshold at 0.8 determines that in that trial the subject has developed motor imagery concerning the movement of “both feet”, while reaching the threshold at 0.2 (representing a decrease in the function starting from 0.5) decrees the assignment to the class “both hands”.

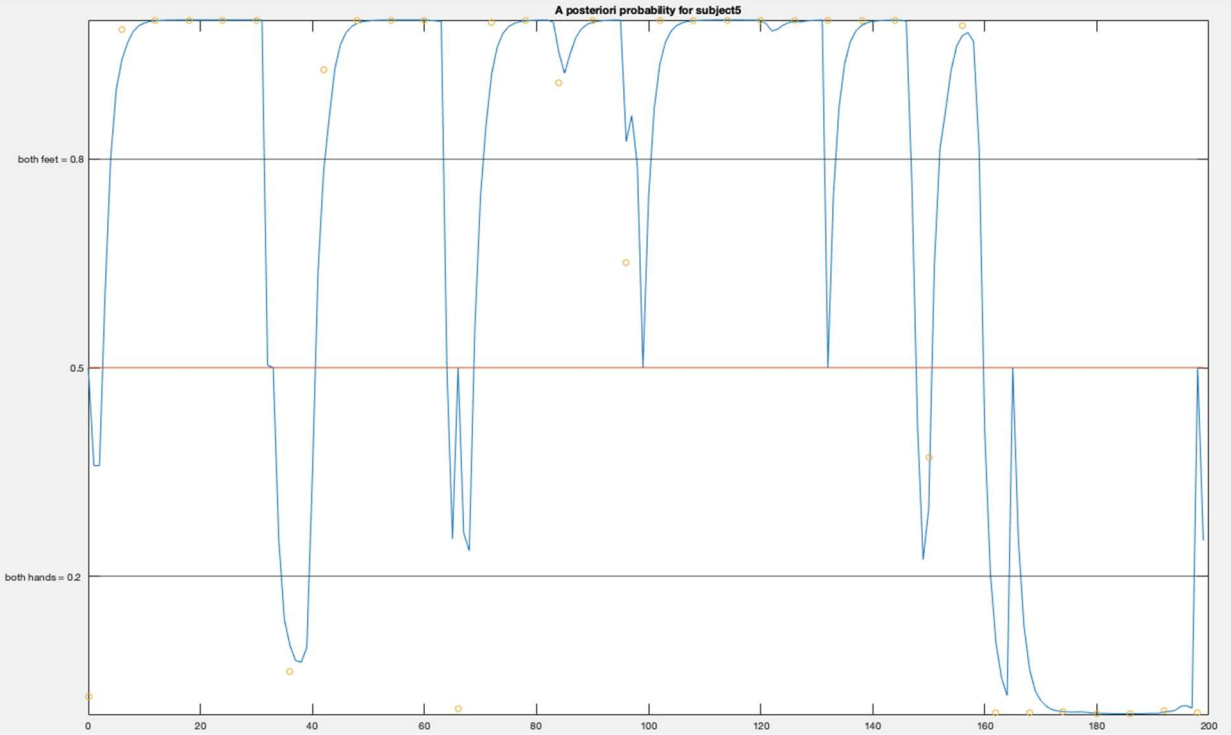


Figure 9 – Evidence accumulation graph for subject 5

In the “evidence accumulation” graph, the circles represent the 'raw probabilities' of the classifier, where 1 represents the class of both feet, and 0 that of both hands; the function represented is the posterior probability (i.e. a time-integrated probability).

Point 2c.
Finally, the average time required to issue a command was calculated.
To do this, the number of windows for each trial of each subject was calculated and, after finding the time of the individual trial (multiplying this number by the duration in seconds of each individual window *wlength*, information come from *lap_psd*), the total overlap period of the windows (taking the information of *wshift* from l*ap_psd*) was subtracted.
To obtain the average time required to issue a command for each participant, the values of the times obtained for each individual trail were averaged.

```
% We remove the time due to the overlap
time = time_w*wlength - wshift*(time_w-1);
time_m = mean(time);
times_m(end+1) = time_m;
```

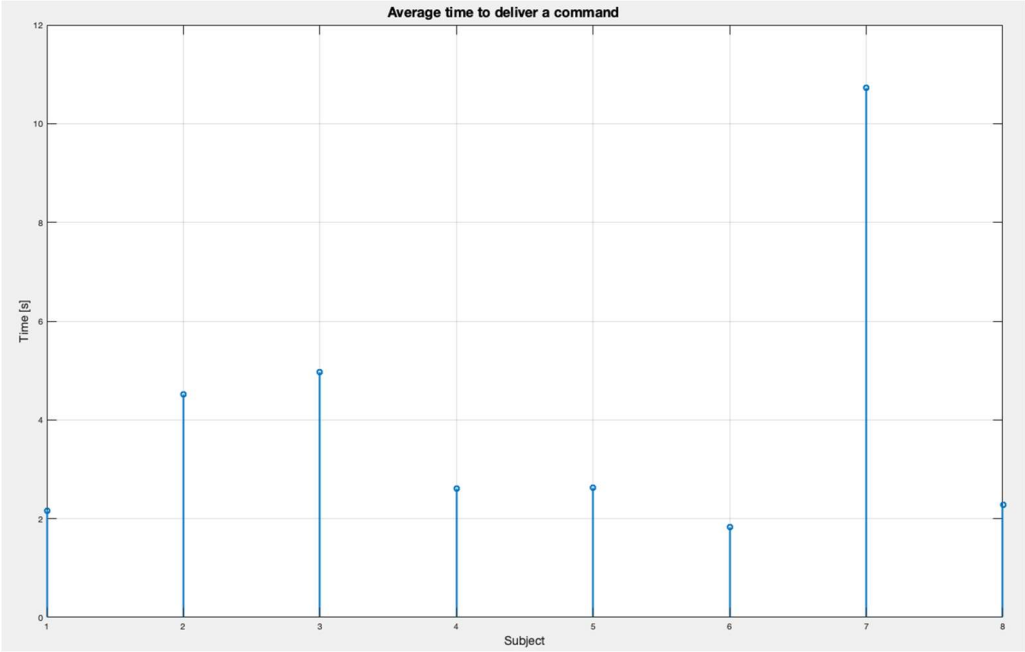


Fig. 10 - Average time to deliver a command for each subject.

RESULTS AND DISCUSSION

Considering the previously reported EEG data visualizations (Figures 3 and Figure 4) we were able to verify the consistency between the results obtained with the temporal visualization and that obtained with topoplots.
Moreover, as it should be, consistent uniformity of topoplots was obtained for the fixation phase, and the presence of more or less localized ERD/ERS was observed for mu and beta bands in relation to the task performed. For example, a strongly localized ERD in the central scalp area (Cz electrode) in mu band is evidenced for subject 1.
However, such evidences were not found completely equally for all subjects, preventing the identification of a specific unambiguous pattern.
Considering the classifier creation phase, as previously highlighted, the combination that gave us the best results was to consider 5 features, working in logarithmic space.

Looking at the accuracy graphs, it's possible to notice that for all subjects the results on the class "both feet" are somehow underwhelming; we also report the scarce results by subject 7 and the almost perfect results by subject 5. We focus on these subjects and analyze their data to better understand the behavior of the classifiers.

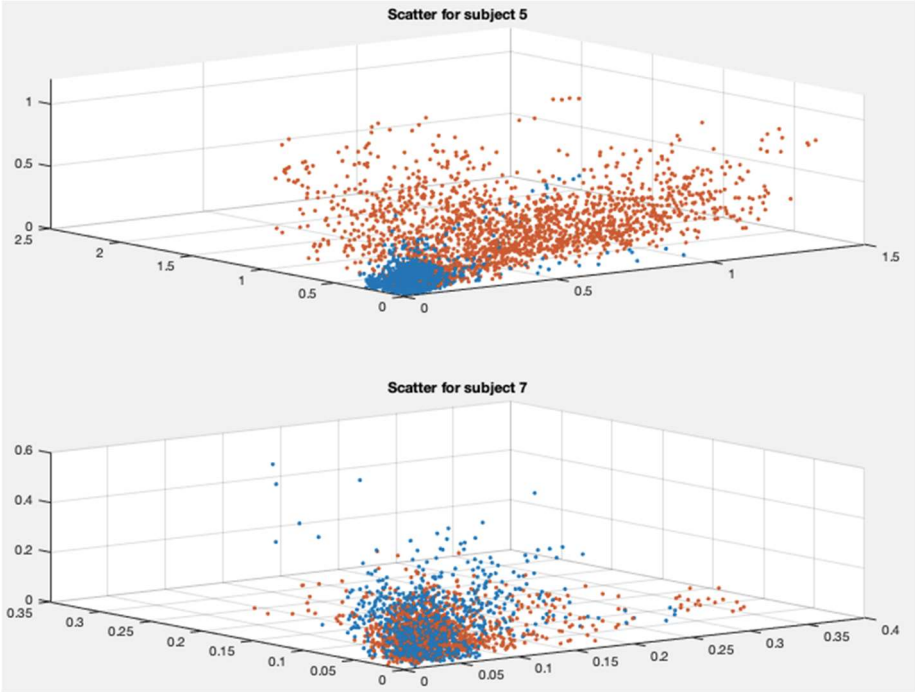


Figure 11 – Scatterplot for subject 5 and subject 7

The decision to perform classification tests in logarithmic space stems from the fact that this should lead to a better separation of classes.
In fact, we could see that in the offline data, overall accuracy increased by 2% for each subject than that obtained by performing the analysis in features space; considering the results of the online data, however, we noticed an increase in overall accuracy in subjects 3, 4, 6, 8, while a decrease in the others, as we can see in the figure below.
Moreover, in the case of logarithmic space, the average time increased by 9 seconds for each subject.
These are the reasons why, in the end, we decided to use the classifiers obtained by working in features space.

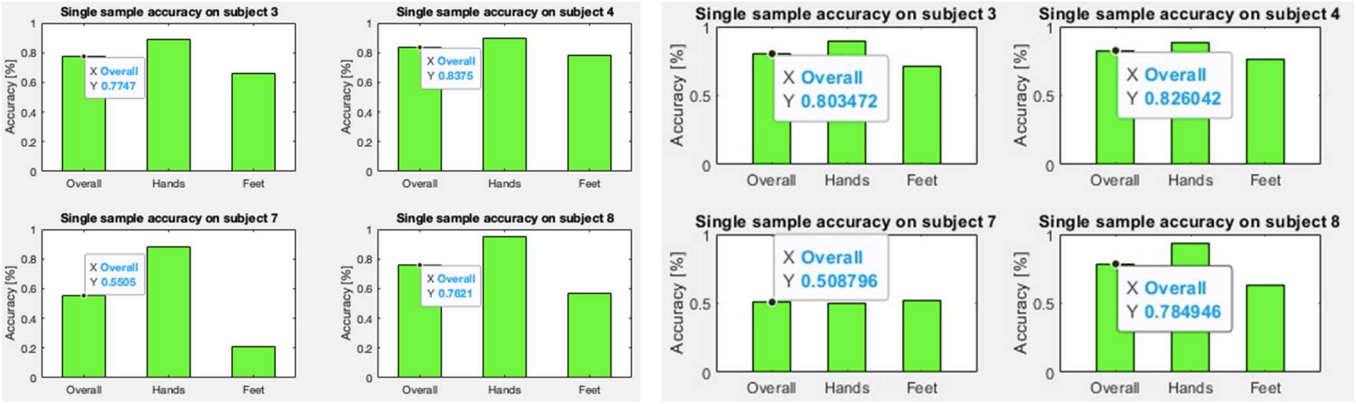


Figure 12 – Comparison of the results obtained with the classification of online data in feature space (left) and that in logarithmic space (right), for subject 3,4,7,8.

References

[1] Tonin L et al. The role of the control framework for continuous tele-operation of a BMI driven mobile robot. IEEE Transactions on Robotics, 36(1):78-91, 2020. doi: 10.1109/TRO.2019.2943072

[2] Pfurtscheller G et al. Motor imagery and direct brain-computer communication. Proceedings of the IEEE, 89(7):1123-34, 2001. doi: 10.1109/5.939829

[3] Wolpaw JR et al. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. Proc Natl Acad Sci USA, 101(51):17849-54, 2004. doi: 10.1073/pnas.0403504101

[4] Leeb R et al. Transferring brain–computer interfaces beyond the laboratory: Successful application control for motor-disabled users. Artificial Intelligence in Medicine, 59(2):121-32, 2013. doi: 10.1016/j.artmed.2013.08.004

[5] Perdikis S et al. The Cybathlon BCI race: Successful longitudinal mutual learning with two tetraplegic users. PLOS Biology 16(5):e2003787, 2018. doi: 10.1371/journal.pbio.2003787

[6] B. Blankertz et al. Optimizing Spatial filters for Robust EEG Single-Trial Analysis in IEEE Signal Processing Magazine, vol. 25, no. 1, pp. 41-56, 2008, doi: 10.1109/MSP.2008.4408441.