Automatic Label Creation of MRCP for Online Control of a Robotic Soft Glove

Christoffer Hansen
The Technical Faculty of IT and Design
Aalborg University, Denmark
ch17@student.aau.dk

Fredrik De Frène
The Technical Faculty of IT and Design
Aalborg University, Denmark
fdefre16@student.aau.dk

Simon Park Kærgaard
The Technical Faculty of IT and Design
Aalborg University, Denmark
skarga17@student.aau.dk

***Abstract*—Recent advances in brain-computer interface (BCI) systems have assisted in the neurorehabilitation of physically disabled patients to regain motor skills in semi-paralyzed body parts. Movement-related cortical potentials (MRCPs) are appealing in the BCI field as they occur naturally before movement in electroencephalography (EEG) data and capture movement intention. However, collecting and labeling the MRCPs is cumbersome and requires domain knowledge. We propose an automatic labeling technique that improves the accuracy of a selection of classifiers by up to 15% in offline experiments. To achieve the performance increase, we identify movement onsets found in electromyography (EMG) data, which is collected simultaneously with the EEG data, and use these as a guide to find the peak negativity that is a key characteristic of MRCP. We also simulate an online environment to test the feasibility of our proposed labeling technique, and our best model detects 11 out of 15 MRCPs with an 8.3% false positive rate and a mean delay of 0.26 s.**

***Index Terms*—Brain-Computer Interface (BCI), Movement Related Cortical Potential (MRCP), Electroencephalogram (EEG), Electromyography (EMG), Motor Imagery (MI), Machine Learning**

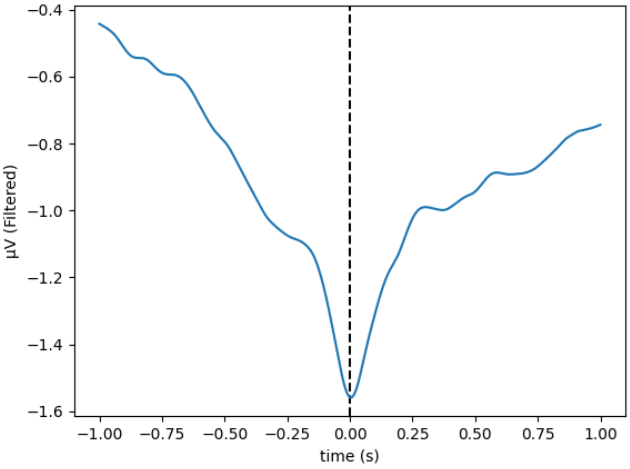


Fig. 1: The average of 30 MRCP samples from a single channel shows the clear morphology of MRCP: a slow amplitude decrease prior to the negative peak, centered at 0 s and shown with a dashed line, followed by an amplitude increase.

I. Introduction

Brain-computer interface (BCI) systems translate brain signals into commands that capture the movement intentions of humans, thereby enabling partial or full control of external devices. For example, the movement intention of grasping a hand as captured in the brain signals, can be translated into commands for controlling a robotic soft glove that assists in executing the actual grasping. Thus, BCI systems are appropriate platforms for the development of new methods in neurorehabilitation for patients with a lack of motor function, e.g., patients that are paralyzed, amputated, or suffer from amyotrophic lateral sclerosis (ALS) [1]. In particular, scientists at the Health Science and Technology Department at Aalborg University are studying BCI systems for ALS patients. We aim to contribute to this research by improving the detection of movement intentions, thereby improving the effectiveness of the BCI system.

BCI systems may rely on invasive or non-invasive neuroimaging methods. Abiri et al. [1] report that electroencephalography (EEG) is the most popular method among human participants due to its non-invasive nature and because it features *"direct measures of neural activity, inexpensiveness, and portability for clinical use."* Non-invasive EEG measurements are obtained from electrodes attached to a specialized cap placed on a subject's scalp. The voltage fluctuations within the EEG signals can be used to detect motor execution (ME) or motor imagery (MI), i.e., the subject's intention to perform a motor task [2]–[4]. Thus, EEG has been used to create pathways from the brain to brain-controlled assistive or rehabilitative devices to aid people suffering from neurological deficits.

A recent study by Savić et al. [5] tests the feasibility of online detection of reaching and grasping using movement-related cortical potentials (MRCP) for the application in patients suffering from ALS. MRCP is a naturally occurring pattern in EEG signals that indicate movement intention. Other patterns in EEG signals might require appropriate training for the subject to evoke them clearly, whereas MRCP does not require specific training and captures both ME and MI, making it possible to test on healthy and non-healthy subjects [6], [7]. The signal morphology of an MRCP is characterized by the occurrence of a slow amplitude decrease between 2 and 0.5 s prior to the movement onset, known as the bereitschaftspotential (BP), a negative peak around 0.4 s prior to the movement onset, and finally, an amplitude increase

(rebound phase) lasting around 1 s after the negative peak [8]. An example can be seen in Figure 1.

Other studies focus on event-related desynchronization (ERD), and event-related synchronization (ERS) based BCIs [9]–[11]. ERD is characterized by a localized amplitude decrease within given frequency bands of the EEG signal, while ERS is characterized by an amplitude increase [12], [13]. In comparing ERD/ERS and MRCP, Seeland et al. [14] found notable movement detection classification differences between the two. MRCP-based classification is superior to ERD/ERS-based when the task is to detect movement close to the actual movement onset, while ERD/ERS seems to perform better further away from the movement onset. In a real time system, detection latency is an important factor, which is why we focus on MRCP instead of ERD/ERS.

This report surveys the process used at Aalborg University's BCI-Lab to collect data, process the data, train a classifier to detect MRCPs, and finally use the classifier to control a robotic soft glove. In turn, this allows us to identify which elements of the process we can contribute to or improve.

A crucial step in detecting movements involves recording electromyography (EMG) signals alongside EEG signals from a subject. The EMG signals are recorded from skeletal muscle activity, using non-invasive electrodes placed on the subject's skin over muscles and bones. The resulting signals capture the actual movements by the subject. They can then be cross-referenced with the EEG signals to distinguish between rest and movement within EEG signals and label the data to obtain a ground-truth labeled training set for a classifier.

In summary, the paper makes the following main contributions:

1) We propose an improvement to the EMG movement onset detection by introducing a clustering algorithm that uses a distance measure between groups of EMG bursts to find the desired number of movements.
2) We propose the shared local minimum (SLM) algorithm that automatically aggregates the mean peak negativity among the EEG channels and centers them at the movement onset. This minimizes the need for manual labor during the trial's data processing.
3) We propose a metric for the online MRCP detection to evaluate detection latency by calculating the mean time from a prediction to the nearest MRCP.
4) We conduct an empirical study into the proposed improvements to provide insight into their effectiveness.

The remainder of the paper is organized as follows: Section II presents the procedural steps of configuring and operating the system currently at use in Aalborg University's BCI-Lab. In Section III, we introduce the key challenges faced by the currently used system. Section IV proposes our improvements that address the challenges. Then Section V covers our data acquisition and introduces the subjects involved. In Section VI, we present and discuss our experimental findings. Section VII discusses challenges not outlined in Section III that can cause errors and affect our experimental results. Finally, Section VIII offers conclusions.

II. Experimental Setup

This section presents the current data recording setup and the current process of training and predicting movements in an online setting. We describe this procedure as a combination of the initial procedure presented by Savić et al. [5] in their research on online control of a robotic soft glove and the instructions we received in the lab from a domain expert. We highlight this procedure's key points and identify its challenges and drawbacks. First, we present the data recording setting, which deals with the instrumentation and subject motor task. This includes an offline training phase and pseudo-online analysis, followed by an online phase. Then we present the currently used methodology of preprocessing the raw data, including filtering and feature extracting, which is a part of the offline training phase. Finally, we present the technique for predicting movements from real-time data and associating those with controlling a robotic soft glove, which is a part of the online phase.

For each subject, an initial 30 movement executions are gathered for the classifier's training of movement detection. Afterward, 20 movements are gathered for a pseudo-online analysis to identify the classifier's best performing EEG channel for online prediction. During the online phase, another 20 movements are executed to validate the performance of the trained classifier on the selected EEG channel. An overview of these phases is depicted in Figure 3.

A. Instrumentation and Motor Task

The experiment uses a g.USBamp (g.tec GmbH, Austria) amplifier and a g.GAMMAbox (g.tec GmbH) to record EEG signals from nine electrode channels positioned in a g.GAMMAcap2 (g.tec GmbH) and placed over the scalp on locations F3, FC1, FC5, Cz, C3, T7, CP1, CP5, and P3 following the international standard 10-20 system, as seen in Figure 2. We refer to these channels as 1–9 in the same order they are presented. The ground electrode is placed at AFz, and the reference electrode is placed on the left earlobe. FP1 is used to record electrooculography (EOG) signals for detecting eye-movement artifacts. All signals are recorded with a sampling rate of 1200 Hz using no embedded filtering from the amplifier.

While seated in a chair in front of a table, the subject has to perform the motor task of reaching and grasping a bottle in a decisive and ballistic manner. At the same time, EMG activity is recorded using an EMG electrode placed on the forearm with a reference electrode on the wrist and the ground electrode on the lateral epicondyle of the elbow. These positions are selected as they capture the movement of the muscles that are activated when performing the grasping movement. The EMG signals can be used to identify where in the EEG signals the movement has been executed.

While the subject is in the rest position, the subject's arm is located on the table. A motor task is executed correctly if the subject initializes the movement from the rest position and then reaches, grasps, and lifts the bottle vertically off the table and finally places the bottle back on the table and returns

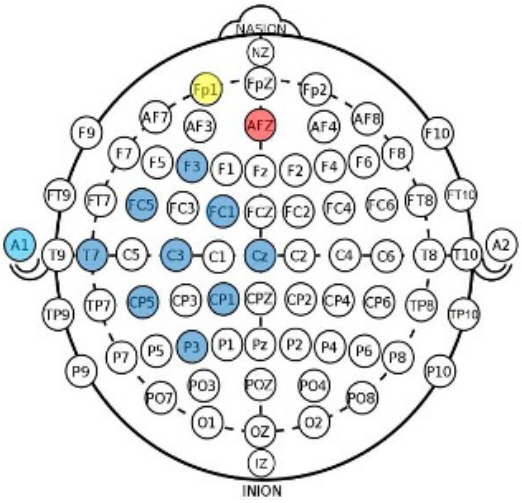


Fig. 2: Layout of the EEG electrodes from Savić et al. [5], placed over the scalp following the international standard 10–20 system. The nine EEG electrodes are marked with blue, reference electrode with cyan, EOG electrode with yellow, and ground electrode with red. These electrodes are placed according to a right-handed subject.

to the resting position. The subject is instructed to execute the movements with a pause of at least five seconds between consecutive movements, but the subject is also asked to do so in a self-paced manner without counting the seconds or focusing on maintaining a constant pace of task execution.

During the movement execution, the subject is also instructed not to blink, swallow, or exercise other facial muscles to minimize artifacts within the EEG signals while executing the movement. The subject is instead asked to perform such motions while being in the rest position between motor task movements.

B. Training Phase

The training data consisting of 30 movement executions are filtered before a sliding window is applied on top of the signal trials to collect training samples, where a trial denotes the data collected from a single channel. The EMG signals are filtered with a 2nd order high-pass zero-phase butterworth filter using a 10 Hz frequency cutoff. This minimizes baseline drift and movement artifacts within the signal. A manual threshold is set in order to find the movement onsets in the trial. A band-pass filter with a frequency range of 0.05–3 Hz is applied to the EEG signals. For the EOG signal, another band-pass filter with a frequency range of 1–10 Hz is applied to detect blinking.

After filtering, the experimenter manually inspects the EEG trials to decide whether a representative example of MRCP is present at the detected movement onset or not. A GUI aids in this process by presenting an overview of trial segments from the EEG, EOG, and EMG channels temporally aligned with the detected movement onsets. The expected outcome of the EEG trials is representative samples of MRCP. Each segment in a trial is called an epoch; hence, a trial with 30 successfully detected movement onsets yields 30 epochs. From the GUI, the experimenter is able to accept or reject epochs determined by three criteria:

1) The typical morphology of an MRCP with a readiness potential, negative peak, and rebound phase, as described in Section I, must be present in at least three EEG trials for the epoch to be accepted.
2) The EEG epoch must not contain artifacts, e.g., eye-movement or blinking artifacts.
3) There should be no discrepancy between the peak negativity of the MRCP and the EMG burst denoting the actual movement.

Before rejecting an epoch, the experimenter can adjust the alignment manually to see if it helps to satisfy the third criteria. By rejecting an epoch, a *rejected event* is annotated at the timestamp of the rejected epoch's movement onset. By accepting an epoch, an *accepted event* is annotated at the timestamp of the accepted epoch's movement onset.

The MRCP detection classifier's training set can be made when all epochs have been inspected and accepted or rejected. This is done by a 2 s sliding window moving over the EEG trials while labeling each window to be either *rest* or *movement*, as depicted in Figure 4. The sliding window starts by labeling windows as rest as this is the initial position the subjects will be in during data recording. The sliding window moves with a step size of 100 ms. Whenever the sliding window enters an event, the step size is decreased to 25 ms in order to balance the distribution of rest and movement samples within the training data. If the concerned event is annotated as accepted, the sliding window will continue to move until the event is located within its third quarter. It will then label the subsequent windows as movement until the accepted event leaves the sliding window interval and discard it otherwise. However, if the event is annotated as rejected, the concerned windows are discarded. Immediately after the event leaves the sliding window interval, the sliding window is shifted 5 s forward to introduce a refractory period in the system, and the step size is reverted to 100 ms. This process continues until the final event has been covered.

Features for the classifier are then extracted from movement and rest samples by introducing a new 500 ms sliding window with 50% overlap, i.e., a shift of 250 ms, within each labeled 2 s window. We refer to these as sub-windows. However, the number of data-points samples are down-sampled with a factor of 100. Three temporal features are extracted within each sub-windows, namely the mean, slope, and standard deviation of the EEG amplitude. The slope feature is computed using linear regression and is the slope of a line fitted through the data-points of each sub-window.

When the classifier training is completed, the pseudo-online analysis can commence, where the classifier is given the next set of 20 movements to identify the best three performing candidate channels for the final online phase. From the GUI, the experimenter can see the classifier's predictions on the 20 MRCPs, on each EEG channel. In the GUI, a detection can fall into three categories; true (true positive), false (false

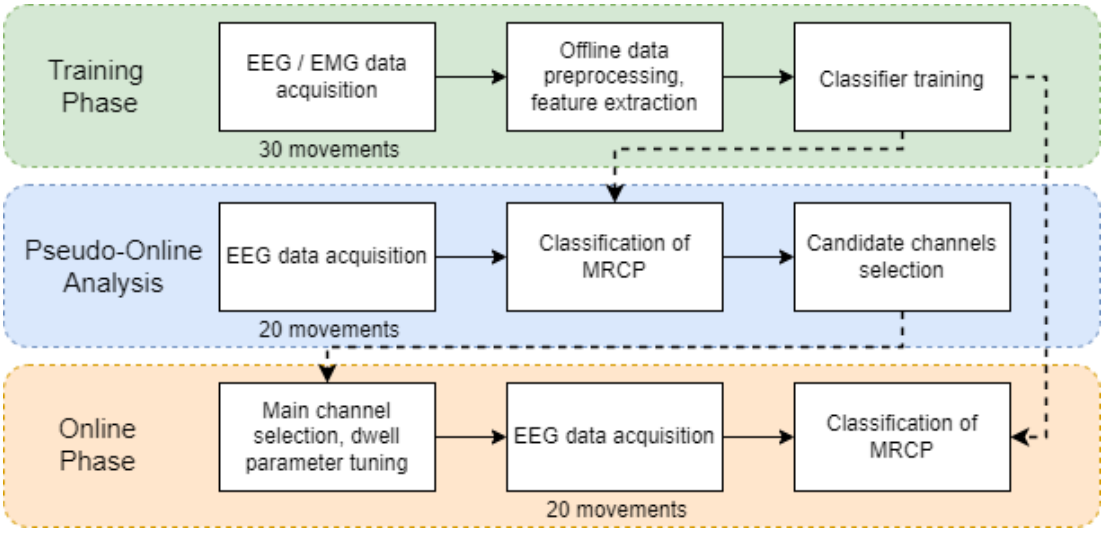


Fig. 3: Overview of the procedural steps of the training, pseudo-online, and online phase presented to us in the BCI-lab.

positive), and missed (false negative). The selection criteria for the candidate channels is the least false-positive detections.

C. Online Phase

The best-performing candidate channel is selected as the classifier's prediction channel, and EEG signals are now given directly to the classifier. The input format is a window with a 2 s interval to correspond with the training data that contains the newest captured EEG signals. Any new data is then appended in blocks of 100 ms.

Before commencing the final online assessment, an additional parameter called the *dwell* parameter needs to be tuned. The dwell parameter denotes the required number of consecutive windows classified as MRCP for a positive detection to be declared, hence introducing a trade-off between sensitivity and specificity. A low dwell parameter makes the system susceptible to false positive detections during rest periods, while a high dwell parameter might increase the system's number of false negative detections. Thus, the goal is to find a minimum dwell value without introducing too much sensitivity to the system.

As an initial value, the dwell parameter is set to three. In order to tune the dwell parameter, the subjects have to remain in the rest position for around 20 s while the experimenter monitors the system for any false positive detections. If any false positive detections are made, which the experimenter verifies, the dwell value is increased, and another evaluation is made until the subject completes a period without any such detections. Sometimes the detection of MRCPs requires a decrease of the dwell value for successful detection, but this consequently introduces too much sensitivity during rest periods. In that case, the experimenter can switch to another candidate channel and go through the procedure again until a satisfactory state is reached.

The subject must then perform the motor task identically to the previous phases, which is another set of 20 movements in the online assessment. The subject also needs to wear the robotic soft glove. The rest period between movements is crucial in this phase due to the refractory period of 5 s, where the glove will close, and the classifier will temporarily stop to ensure that the movement is completed and the subject has returned to the resting position.

For each movement execution, the experimenter manually logs the detection of the classifier as either correct (true positive), missed (false negative), false detections (false positive), or delayed. A detection falls into the delayed category if it is detected in the interval of two seconds prior to three seconds after ([-2 s, 3 s]) the actual movement execution of the subject but is not classified as a true positive. A delayed detection counts more towards a correct than a missed detection as the system still manages to pick up on the movement that is the readiness potential or rebound phase of the MRCP.

III. Key Challenges

Throughout the investigation of the existing procedures and learning about the problem domain from the experts in the field, we deduce some key challenges when recording and processing EEG signals to perform the classification of MRCP.

The current process is domain expert dependent and requires the experimenter's attention to different criteria and parameters during the training phase, pseudo-online analysis, and online phase to detect inaccuracies and validate results, as described in Section II. Savić et al. [5] highlight that this process can take an experimenter up to 20 minutes and relies on the experimenter's ability to identify faulty samples and adjust the dwell parameter adequately. Results are also affected by the subject and external factors such as the environment and the hardware. We highlight important areas that could influence the results, which we will use as motivation for our methodology presented in Section IV.

- The EMG signals' movement onset detection often fails to capture all the movements executed during data record-

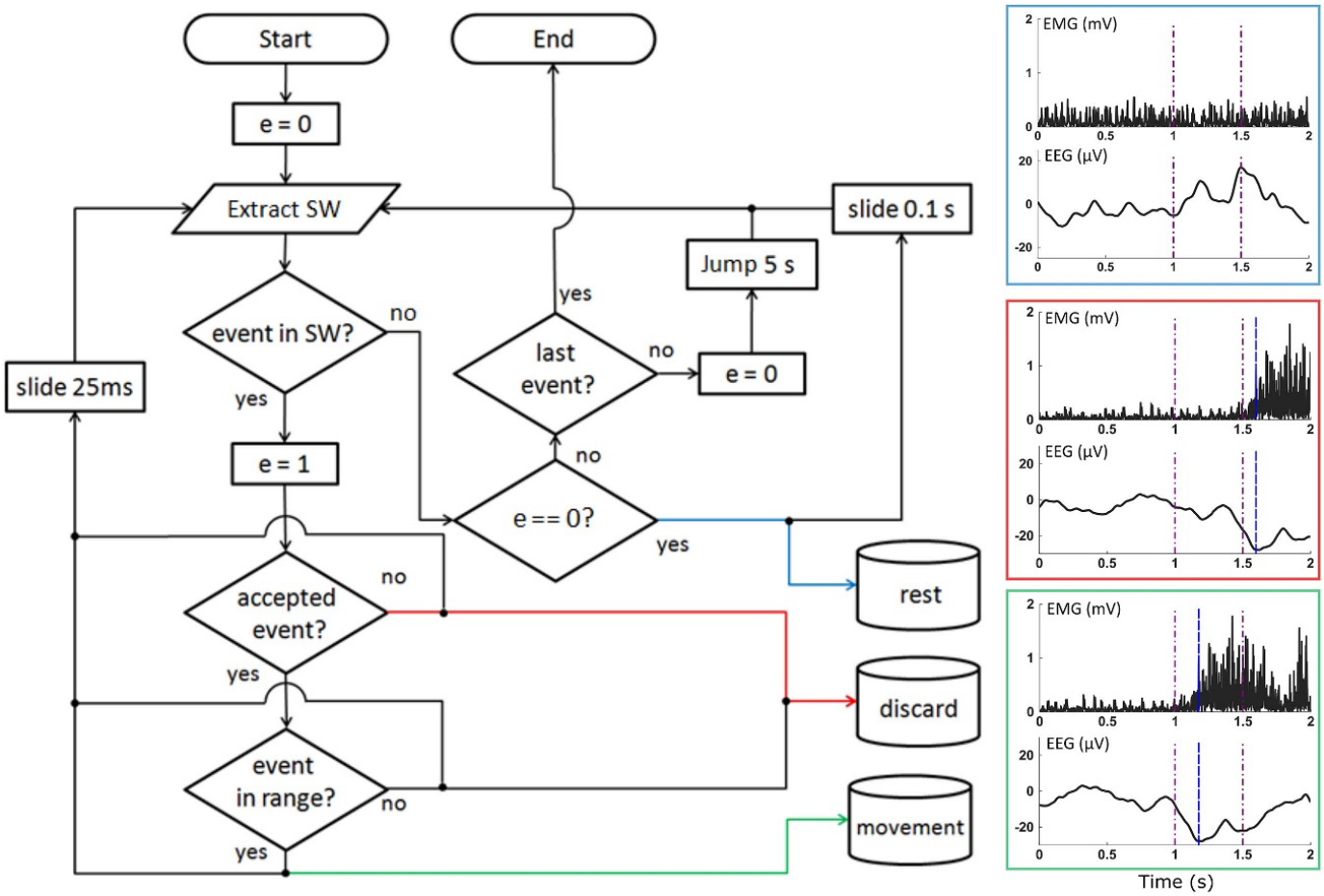


Fig. 4: Flowchart of the algorithm used to extract and label training samples for the movement detection classifier presented by Savić et al. [5]. In the flowchart, e=0 denotes that the sliding window has not entered an event, and e=1 means it has. The three panels on the right side represent the three categories a sliding window sample can be designated. All panels depict the EMG graph above and the EEG graph below. Both graphs are aligned in the time frame of the depicted sliding window. The blue panel depicts a sample labeled as rest. The red panel depicts a sample to be discarded, as the accepted event is located outside the defined range, which is marked by the purple dashed and dotted lines. The green panel shows a sample labeled as movement due to the accepted event being located inside the range. The blue dashed line annotates the accepted event.

ing. Consequently, any MRCPs in the EEG data that corresponds with the missed detections will not be labeled.

- Curating the training data samples in the latter part of the training phase poses a challenge. Due to the nature of MRCP in EEG signals, where the peak negativity can be located either before or after the movement onset, and the peak negativity of all channels is not always aligned, labeling cannot be done automatically in a trivial way. The current solution involves the experimenter using domain knowledge and informed judgment to manually adjust the movement onset closer to the peak negativity of the samples to make it adhere to the third sample criteria described in Section II-B.
- The existing procedure identifies the single best channel for MRCP detection and thus disregards potentially useful data within the other channels.
- Balancing the distribution of rest and MRCP samples within the training data, where the sliding window is decreased to 25 ms as opposed to the original 100 ms, might also have undesirable consequences on the training data. A smaller shift between MRCP-windows might cause the data to be too identical without sufficient divergence between consecutive MRCP-labeled samples.
- The current process to validate the system's performance is loosely defined. For the online testing, if the system reacts within [-2, 3] of the intended movement, it is considered a correct prediction, albeit without accurately measuring the time.

To sum up, the challenges we highlight mainly revolve around the high level of domain expert dependency and manual data processing. We also focus on some technical parts of the data preprocessing step involving improving movement onset detection and curation of training samples, which currently only uses a single channel for the classification task, ignoring potentially useful data from the remaining channels. We aim to define a way to consistently label the dataset

while capturing as much diversity as possible since the current process only includes samples considered 'good' by human-eye-measure. Finally, we focus on the evaluation metrics used and the lack thereof in the aforementioned areas, and the final online validation.

IV. Methodology

To address the challenges outlined in Section III, we present our methods to alleviate the manual labor required from the experimenter.

To improve the existing EMG movement onset detection, we propose an automatic clustering algorithm. Furthermore, we automate the process of improving the MRCP sample's quality by introducing a weighted algorithmic approach for aligning MRCPs with movement onsets. We also define a metric to quantify the performance of our classifier in an online environment.

A. Data Preprocessing

We apply a 5th order butterworth highpass filter with zero phases and 80 Hz cutoff to the EMG data to isolate the signal bursts above 80 Hz, making subject movement in the EMG data more apparent. Zero-phase filtering aids in preserving features precisely where they transpire in the unfiltered signal, but as a drawback, it cannot by default be used in real-time, as it is non-causal, meaning it relies on future input [15].

We show an example of a filtered EMG trial in Figure 5. The EMG peaks denoting movement onsets are easier to identify in the filtered data than the raw data showing significant baseline drift.

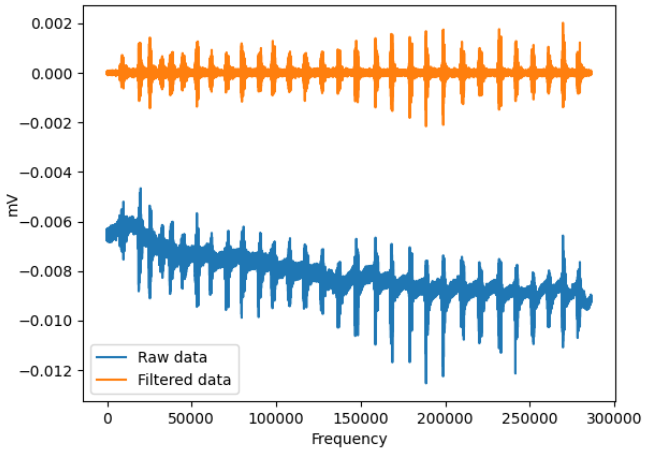


Fig. 5: Comparison of the raw data and the filtered data after applying the highpass filter on the EMG channel.

1) Onset Detection and Clustering: We use a python library for biosignal processing, BioSPPy [16], that provides an adaptive threshold for EMG signal onset detection. The threshold used in the onset detection is predefined in the library. Following a full-wave rectification of the filtered signal, a signal smoother is applied. Subsequently, the threshold is calculated by:

$$T = 1.2 \cdot \frac{1}{n} \sum_{i=1}^{n} |s_i| + 2 \cdot \sigma(|s|) \tag{1}$$

where s is the smoothed signal vector, n is the number of data points in s, and σ is the standard deviation, which uses one degree of freedom. We refrain from using BioSPPy's implementation for cluster creation, as it performs inconsistently on the various datasets recorded in the laboratory. In many cases, BioSPPy's cluster creation finds fewer clusters than what exists in the datasets.

To group the data points above the threshold *T*, we use a distance measure where we make data points close together into a cluster:

$$y > T \wedge |y - y_{\text{next}}| < d \tag{2}$$

where y and y_{next} are data points larger than *T*, and *d* is the maximum distance to the next value. We then consider y and y_{next} in the same cluster. Initially, Equation 2 forms many clusters, but not all represent the EMG onset clusters we are looking for. Some clusters represent abnormalities or artifacts in the data not related to the movement executions of the motor task. The undesired clusters are often small, and we filter them out based on their size relative to the mean size of all detected clusters. We base the heuristic of this approach on how each movement of the motor task is executed homogeneously in terms of force and duration.

We implement an optional parameter to specify a requested number of clusters. We use the cluster parameter when we know the number of movements that are present in the data. When utilizing the cluster parameter, the algorithm gradually increases the default value *d* until the number of clusters corresponds to the cluster parameter after removing unwanted clusters. If the cluster parameter is below the number of detected onset clusters, the algorithm increases the distance measure *d* significantly. If the cluster parameter exceeds the algorithm's number of detected clusters, it does not change the distance measure *d* and returns the detected clusters.

2) Sample Optimization: As explained in Section II-B, the existing procedure relies on a domain expert to manually inspect samples from each epoch to either accept or reject them. This process is cumbersome; thus, we propose a shared local minimum (SLM) method to derive a mutual peak negativity location for each epoch compliant with all EEG channels. We base SLM on a voting approach. In each epoch, to align the peak negativity of the MRCP with the window's center, every EEG channel expresses what direction it should move to achieve this. As a result, we keep all samples instead of rejecting a part of them.

Shibasaki et al. [8] define MRCP as the peak negativity following a slow decline in amplitude. To locate the morphology of MRCPs, the EEG data is filtered using a butterworth bandpass filter with zero phases and 0.05–5Hz cutoff.

Figure 6 exhibits an MRCP sample where the EMG movement onset at the top aligns with the associated peak negativity

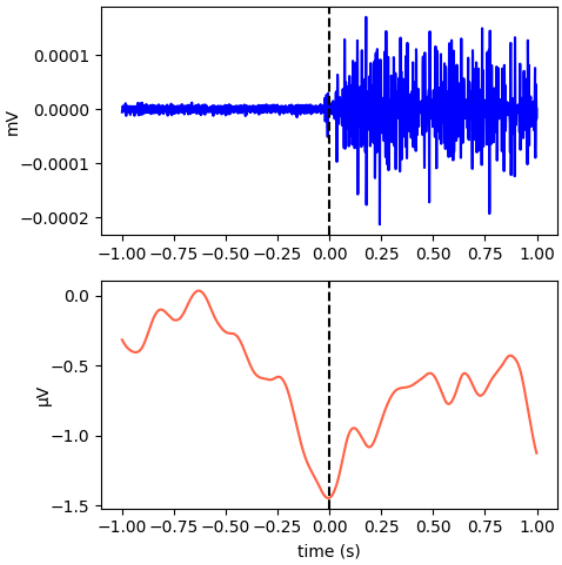


Fig. 6: A sample of filtered EMG data is shown in the top graph and filtered EEG data in the bottom graph. We center the samples at the same timestamp (0 s) denoted by the dashed line.

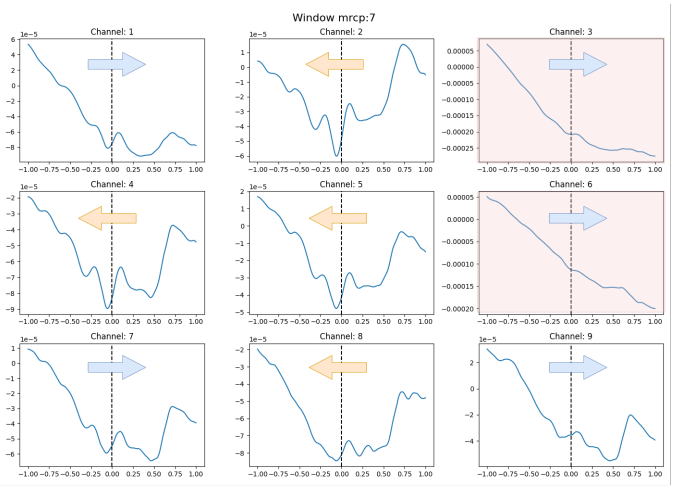


Fig. 7: Overview of an epoch showing MRCP samples from each EEG channel. Each sample is a window of 2 s centered at the same movement onset (0 s), which we denote by a dashed line. The peak negativity of each MRCP sample is located differently between the channels, making the process of labeling difficult. The arrows indicate which direction a channel should move for the peak negativity to be at the window's center.

of an MRCP sample at the bottom. Such an occurrence ties the MRCPs peak negativity to the start of the movement execution, which we want to our classifiers to learn. However, not all EEG channels present such alignment, and we can locate the peak negativity of MRCPs at different time points around the movement onsets, which complicates labeling the EEG data for training. Figure 7 shows MRCP samples from the nine different EEG channels. We center each sample at the associated movement onset denoted by the dashed line. The peak negativity of channels 2, 4, 5, and 8 occurs before the movement onset, while we observe the remaining channels' peak negativity after the movement onset.

To locate the MRCP's peak negativity and use it as a measure, we define a local minimum distance *lmd*:

$$lmd = \arg\min_{x \in F} f(x) - \frac{||F||}{2} \tag{3}$$

where F is the set of frequencies within the range of the given MRCP window. *lmd* calculates the location of the peak negativity relative to the movement onset derived from EMG data. We must keep the directional information to identify whether the local minimum occurs before or after the movement onset.

We apply the SLM voting algorithm to each epoch, and it consists of six steps:

1) The algorithm calculates the *lmd* of each channel, which acts as the channel's vote regarding the desired direction to align the peak negativity with the window's center.
2) The algorithm chooses the direction with the majority of votes.
3) The channels voting in the opposite direction have their votes invalidated.
4) The remaining channels are sorted by *lmd*, and we find the median.
5) The algorithm does not include channels that exceed $1.5 \cdot$median or are less than $0.5 \cdot$median, as the required distance to move the peak negativity closer to the window's center would be too extreme.
6) Adjust the onset by the median of the remaining channels.

By default, there are two issues with the voting procedure, which affect the algorithm's robustness. The first issue is that all EEG channels attain equal voting rights regardless of their placement on the subject's head. The second issue is that samples do not always exhibit the desired morphology of MRCP, hence giving a vote that pollutes the final election. We lack the domain knowledge to tackle the first issue, which would require assigning weights reflecting the channel's expected MRCP quality. However, we introduce weighted voting in step 2 of the voting algorithm, where the algorithm determines the direction, to address the second issue. We calculate the average signal of all MRCP samples for each channel, and then we calculate the $|lmd|$. Higher weights are assigned to channels with a low $|lmd|$ and vice versa. We calculate the local minimum distance vector $\vec{lmd}$, consisting of the collective distances of all the channels $c_1, \ldots, c_n$.

$$\vec{lmd} = (|lmd|_{c_1}, \ldots, |lmd|_{c_n}) \tag{4}$$

Equation 4 is z-score normalized and we calculate the $Softmax$ of the normalized values.

$$\vec{w} = 1 - Softmax(z(\vec{lmd})) \tag{5}$$

The result of Equation 5 is the weight vector $\vec{w}$, which makes the channels with a high $|lmd|$ receive lower voting weight.

Following the example illustrated in Figure 7, the votes from channel 3 and 6 are invalidated by weighting as their samples

do not exhibit morphology of MRCP, and hence yields a high $|lmd|$.

B. Labeling

MRCP samples are only labeled as a movement in the existing procedure if the peak negativity lies within the third quarter of the sliding window. We disregard this boundary to allow labeling samples as movement as long as the peak negativity lies within the sliding window. This approach captures more of the MRCP's rebound phase and also yields more training samples of movement.

The existing procedure also ignores the five seconds of data occurring after MRCP windows, which it deems as refractory periods, hence starving the classifier of this particular data. If the classifier is untrained on data in the refractory periods between movements, it can cause the classifier to act unpredictably in real-time. The existing online phase compensates by introducing a five-second freeze period after any detected movement but leaves the classifier vulnerable if it makes a false negative prediction. This ties into the starvation mentioned earlier. If the classifier does not detect an MRCP, it will not freeze and consequently receive data on which it has not previously trained. To minimize the aforementioned problems, we include the five seconds of data after the MRCP windows in our labeling procedure. By including the five seconds after the MRCP, we also attain the side-effect of labeling continuous movement as rest. Nonetheless, we deem introducing the model to unseen data more severe.

1) Label Balancing: We predominantly record the data with the subjects in the rest periods between movements. The initial recording of the datasets contains $20.1\% \pm 8.2\%$ movement, where the remaining is rest. This creates a skewed distribution of labels with a majority of rest labels, which can lead to overfitting. To achieve a uniform distribution of movement and rest samples, we resample the data by randomly removing rest samples until there is a 50% representation of both.

C. Feature Extraction

We utilize three features for each epoch to use for prediction to distinguish MRCP windows from idle windows. The three features we extract from each MRCP epoch are the mean amplitude, the amplitude variability, and the slope given by linear regression.

We extract the features from the sub-windows of size 500 *ms* with a 50% overlap of each epoch. Given an epoch of 2 s duration, this creates seven sub-windows from which we extract the aforementioned features. In total, this creates a vector of $7 \cdot 3 = 21$ features. The mean amplitude, slope, and method of extracting them are consistent with what domain experts use for similar data [6], but we choose amplitude variability instead of standard deviation.

D. Channel and Model Selection

Savić et al. [5] select a single EEG channel for online prediction. We hope to capture additional information by evaluating all channels. We select three classifiers, a kNN, SVM, and LDA trained for all channels selected for each dataset. Aliakbaryhosseinabadi et al. report subject-specific classifier tuning to increase the performance of hand movement detection [6]. To accommodate this, we also train individual classifiers for each subject. Our strategy involves evaluating all channels and using an ensemble approach to find the optimal subset of channels that yielded the best results during the offline evaluation. Our ensemble approach combines the predictions of multiple models, one for each channel, and predicts the majority. Given that an uneven number of channels determines the majority vote, we only optimize over the odd-sized powerset of channels. Although we evaluate all channels, a single channel could potentially be the best performing subset.

E. Simulation

To test our models in an online setting, we create a simulation suite to mimic a real-time procedure with our proposed improvements presented in Section IV. The suite makes us independent of the recording equipment in the BCI-Lab. The suite encapsulates an offline environment, which can simulate the procedure without time constraints. It also includes an online environment to simulate how an online system behaves under real-time constraints, and data is only given to the model progressively as if we recorded it. The time constraints also require us to implement our proposed methods without introducing additional latency to the system, which can impede movement detection and the overall usage of the system by a subject.

1) Online Environment: The purpose of the online simulation is to predict when to close the robotic soft glove and subsequently stop all predictive behavior to allow for a refractory period where, under real circumstances, the robotic soft glove has time to reopen, and the subject can return to the resting position. We depict the steps of our online simulation in Figure 8. After loading a set of models together with the associated dataset, the data is sequentially fed to the model to mimic a sliding window with a step size of 100 ms. A data buffer is built from the sliding window until it reaches its maximum capacity of 20 seconds and is constantly updated as long as the simulation runs. This means we always keep the latest sliding window within the newest part of the data buffer. The data buffer is necessary to apply filters to the data as zero-phase filtering requires more data than a single window to function. We apply a 2nd order bandpass filter with zero phases and a 0.05–5 Hz cutoff to the data buffer and extract features from the latest sliding window in the buffer. We give the extracted features to the classifier, which predicts whether the current sliding window contains MRCP or not. We introduce our freeze heuristic to determine if the system should execute the clasping movement of the robotic soft glove. The freeze heuristic acts similar to the dwell parameter used in the existing procedure. However, instead of using a consecutive number of predicted movements to trigger the movement of the robotic soft glove, we let the freeze heuristic tally if four out of the five latest sliding windows are predicted as movement

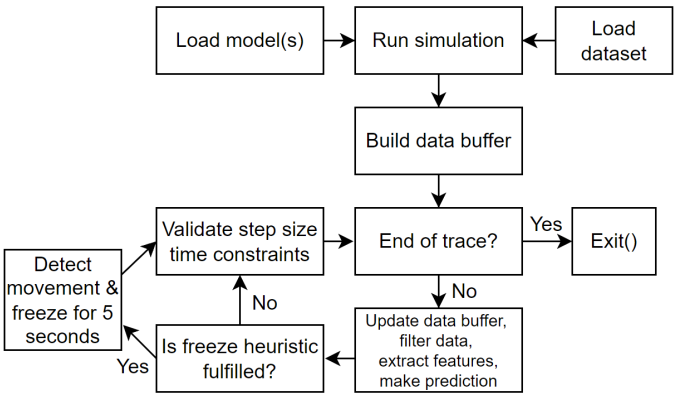


Fig. 8: Workflow of the online simulation environment. We load pretrained or newly trained models for the simulation into the system. After building the data buffer, the simulation will loop through the data until we reach the end. Each epoch will have its features extracted and predicted upon. If four out of the five last predictions are movements, we announce a movement and freeze the system for five seconds. Before the next iteration begins, the system validates whether the time constraints are fulfilled. If not, we increase the step size accordingly.

regardless of order. We cannot tune the freeze heuristic in the same manner as the dwell parameter from the existing procedure. In the lab, there is a period between the training phase and the online phase where the experimenter can view the classifier's predictions and in real-time tune the dwell parameter accordingly. In contrast, we simulate on prerecorded data where this period is not present. Tuning can reduce the amount of false positive detections and thus increase the accuracy, but it also requires the experimenter's involvement, which can be cumbersome. We hope to find a good balance between sensitivity and specificity with our freeze heuristic despite not having the ability to change how many detected movements are required for the robotic soft glove to activate. The duration of the freeze period can be changed but is kept at five seconds to correspond with the rest period between movements during data recording.

We keep track of each step's duration in the simulation to ensure we can perform the calculations within the default step size's time limit of 100 ms. If the duration of a step exceeds the time limit, we increase the step size to accommodate the added latency. We need to be aware of this issue as the number of selected channels can vary for each dataset. Additional channels mean increasingly complex feature extraction and data buffer updates, which cause increased latency.

F. Online Evaluation Metric

We define a metric to evaluate the performance of the dataset in the online simulation. Understanding how distant a prediction is to the nearest MRCP epoch is valuable, as predictions close to MRCPs can still be helpful. In the existing procedure, the experimenter manually counts and annotates the movements of the robotic soft glove in the online phase. Missed predictions can manually be classified as delayed predictions instead if the experimenter deems it fitting, which is not a robust way of evaluating the system in an online setting.

We propose a new post-analysis metric called mean time to nearest MRCP (MTNM), a more sophisticated method for evaluating the system during online simulation. We define the metric as:

$$MTNM = \frac{1}{n}\sum_{i=1}^{n} \min_{x \in X} |\hat{y}_i - y_x| \tag{6}$$

where, for each prediction, $\hat{y}$, we find the minimum distance in seconds to the nearest ground truth, y. X is the set of ground truth variables present in the trial. If the MTNM is low, the system has an overall low response time, and it has positioned few predictions outside of MRCP windows. Using the metric for all predictions gives an indication of the system's overall response time, but it can be valuable to look solely at the false positive predictions. When evaluating the simulation, we will apply the MTNM specifically to false positive predictions.

V. Experiments

We have conducted three sessions of data recording in the BCI-Lab at Aalborg University with three subjects (SUB1, 25 yrs, male), (SUB2, 25 yrs, male), and (SUB3 27 yrs, male). All participants were right-handed and healthy without hearing or vision abnormalities and did not have any known neurological disorders. The participants were all acquainted with BCI systems but had no prior experience with the data collection procedure, which we presented in Section II. From each participant we recorded two datasets: One consisting of 30 movements from the training-phase and a second consisting of 20 movements from the pseudo-online-phase.

Additionally, we were provided with two prerecorded datasets consisting of 30 and 31 movements by the Department of Health Science and Technology at Aalborg University. Both datasets were recorded from the same subject with diagnosed ALS. The age and gender of the subject is unknown due to privacy reasons. We refer to this subject as SUB0. The instrumentation, i.e., amplifier, number of electrodes, electrode placement, and sampling rate used for the data acquisition are identical to ours as presented in Section II-A. However, the motor task differs from our own as the subject had to execute a ballistic dorsiflexion, i.e., an upwards movement of the foot. The subject also had to execute the movements timed to a visual cue where the other three subjects executed their movements in a self-paced manner.

VI. Results

This section presents the comparative results of the movement onset clustering algorithm and the existing method. Furthermore, we compare the SLM algorithm's performance to a baseline approach without sample improvement. Lastly, we evaluate the performance of all proposed methods in an online simulation. Hereto, we use the MTNM as an additional evaluation metric.

A. Movement Onset Detection

Table I shows our automated clustering method to detect movement onsets compared to the already existing method on eight different datasets. The metric we use for determining the success of our movement onset detection is the amount of movements detected in comparison to how many actual movements were executed during recording.

TABLE I: A comparison between how many movements our method and the already existing method of movement onset detection found on eight different datasets from four different subjects.

Sub ID	Data-set	Our Method	Existing Method	Expected
SUB0	01	30	36	30
	02	31	31	31
SUB1	03	30	29	30
	04	19	22	20
SUB2	05	30	24	30
	06	20	19	20
SUB3	07	28	34	30
	08	20	19	20

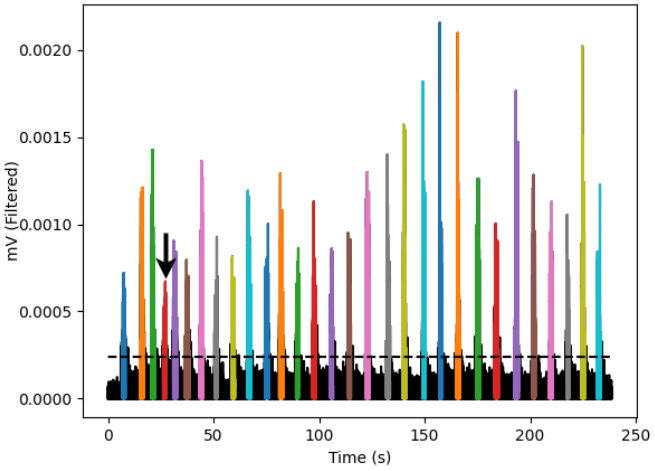
We observe lower deviance between the expected and detected movements in our method from these tests. Our method also does not detect a higher number of movements than is expected due to our cluster parameter described in Section IV-A1, which prevents the detection of more than the defined number of movements.

Figure 9a shows the movement onsets detected by our clustering algorithm, and Figure 9b shows the existing method from SUB1, Dataset 03. The dataset contains 30 expected movements, and our clustering method succeeds at detecting 30 movements, while the existing method detects 29. We show the difference in Figure 9. We can see the missed detection of the existing method at the fourth burst in Figure 9b, whereas we detect the same burst as a cluster using our method, as shown in Figure 9a. The existing method may consider the missed burst to be associated with one of the adjacent bursts, denoting it as one instead of two. Onset detection is a fundamental step of preprocessing as it annotates all samples for training. Missed onset detections are therefore costly on account of the resources involved in data recording.

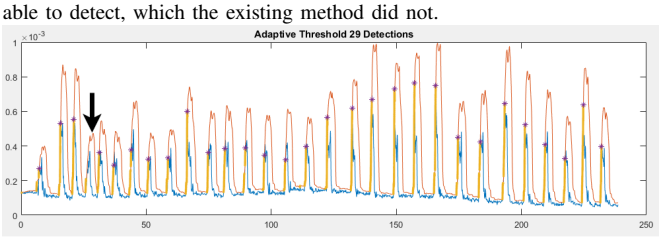
B. Impact of SLM Alignment

We compare our SLM approach to the baseline approach where no optimization is applied to the MRCP samples besides filtering. We use leave-*one*-out-cross-validation due to the low amount of samples in each dataset we are training on. Table II shows *test* accuracy, precision, recall, and F_1-score from our kNN, SVM, and LDA classifiers.

We present the average results of the individual channels and the ensemble results. The average results encapsulate the results averaged across all selected channels. The ensemble results come from the selected channels' majority vote for each prediction. We observe that our SLM approach improves the performance of the SVM and LDA in both aspects of the average channel performance and the ensemble approach. The kNN model improves the performance of the individual channel average, while the ensemble retains the same performance.



(a) Our movement onset detection method detects 30 out of the 30 expected movements. Each cluster of detected movement is colorized and the threshold is shown by a dashed horizontal line. An arrow indicates the cluster we were able to detect, which the existing method did not.



(b) The existing onset detection method captures 29 out of the 30 expected movements. Each detected onset is annotated by a purple + sign. The missed detection is annotated by an arrow.

Fig. 9: Comparison between (a) our onset detection method and (b) the existing onset detection method. Both methods are applied on Dataset 03 from SUB1 with 30 expected movements.

TABLE II: Results of dataset 01 recorded from SUB0. All results are based on the test scores after validating the samples using leave-*one*-out-cross-validation. The ensemble approach yields higher improvement for the SVM and the LDA while the kNN does not see any performance increase with this specific dataset.

Classifier/ Channels	Dataset 01	Average		Ensemble	
		Baseline	SLM	Baseline	SLM
kNN [1, 4, 5]	Accuracy	0.696	0.783	0.816	0.816
	Precision	0.661	0.747	0.756	0.756
	Recall	0.813	0.855	0.933	0.933
	F1	0.728	0.797	0.835	0.835
SVM [3, 4, 5]	Accuracy	0.522	0.761	0.733	0.883
	Precision	0.503	0.830	0.694	0.870
	Recall	0.600	0.644	0.833	0.900
	F1	0.547	0.656	0.757	0.885
LDA [3, 4, 6, 7, 9]	Accuracy	0.664	0.686	0.733	0.766
	Precision	0.645	0.659	0.718	0.710
	Recall	0.723	0.780	0.766	0.900
	F1	0.680	0.713	0.741	0.794

We refrain from presenting the results in the same manner as in Table II for all eight datasets. As an alternative, we summarize all ensemble scores of dataset 1–8 in Table III by showing how each dataset and model perform. Ensemble scores will always yield better results than average.

Table III reports improvements in scores across the majority of the datasets. We see a correlation between our

more consistent labeling technique and higher performance in classification. We observe Dataset 08 with kNN exhibits the biggest decrease in performance, while Dataset 03 with LDA shows the best overall gain.

C. Online Simulation

In Table IV, we present the online simulation experiment's results. For each subject, we use the first dataset recorded for calibration and the second for simulation, which we show in the table's second column. Calibration involves the procedure presented in Section IV-D, where the best-performing channels are selected, and we train the models on subject-specific data.

True Positives presents how many times our freeze heuristic have been correctly fulfilled during simulation, i.e., how many times the model have correctly predicted four out of the five latest predictions as movement. *Hit rate* describes how large a percentage of the expected MRCPs that are detected during simulation. *Total positives* are based on how many times our freeze heuristic have been fulfilled in total, which also includes the number of false positive detections. *False Positive rate* denotes the percentage of the total positives that are false positive. The last two columns present the *MTNM* metric, which we use to determine the mean time of how far a detected movement is from the nearest MRCP, which is presented in Section IV-F. This is calculated for both the total number of detections and the false positive detections.

The number of expected predictions is affected by the data buffer's size, which will consume the first 20 s and skip any movement in this period. Also, if the time between movement executions is shorter than five seconds, the simulation will be unable to detect the movement, as the system will be frozen.

To evaluate the simulation performance of the datasets recorded from each subject, one must consider multiple factors at once. A high hit rate is not a sufficient metric alone to determine if a simulation is good or bad. For instance, SUB3 detects 16 out of 17 expected MRCPs with the SVM classifier, thus reaching a 94.1% hit rate, but at the same time, it also has a significant false positive rate, which is undesirable. The MTNM measures of 2.52 s (total) and 5.39 (false positive) also display high detection response times. Our best-resulting simulation arguably originates from SUB2 using the LDA classifier, where we observe a decent hit rate, and the lowest false positive rate and MTNM (total) of all simulations. The simulation of SUB0's datasets using the SVM classifier also shows promising results with an 85.7% hit rate, a 22.5% false positive rate, and a 0.55 s MTNM (total). SUB0's results are slightly worse than SUB2 with LDA, but SUB0 has nearly double the expected MRCPs in the data than SUB2.

Our approach to filtering in online simulation can lead to inaccurate results. As the data buffer moves across the dataset, repeatedly filtering the same data can cause the data to be represented differently between epochs.

We have not tested the size of our data buffer during our online simulation enough to guarantee optimal performance. A buffer size of ten consecutive windows was sufficient to get results comparable to the offline simulation, where we apply zero-phase filtering on the entire dataset and hence chosen as our buffer size. The data buffer introduces a 20 second wait time when the system initializes and builds the buffer; however, we have not recognized other ways of achieving data comparable to the offline data on which we train the model.

We chose to include all samples found during the MRCP detection procedure in our tests. Samples are often discarded due to poor quality in the existing procedure, either caused by signal artifacts, e.g., eye movement, or discrepancies between MRCP and movement onset. Removing the poor quality

TABLE III: The results of the offline tests; here, we test each dataset on all three classifiers and, in general, they show an increase in performance. Dataset 08 with kNN shows the biggest decrease in performance, while Dataset 03 with LDA shows the best overall gain.

Sub ID	Dataset	Model	Accuracy	Precision	Recall	F1
SUB0	01	kNN	0.81	0.75	0.93	0.83
		SVM	0.88 (+15%)	0.87 (+18%)	0.90 (+7%)	0.88 (+13%)
		LDA	0.73 (+3%)	0.71	0.90 (+14%)	0.79 (+5%)
	02	kNN	0.83 (+11%)	0.78 (+9%)	0.93 (+13%)	0.85 (+11%)
		SVM	0.79	0.76 (+1%)	0.83 (-4%)	0.80
		LDA	0.74 (-1%)	0.71 (+1%)	0.80 (-10%)	0.75 (-3%)
SUB1	03	kNN	0.70 (+12%)	0.65 (+8%)	0.83 (+17%)	0.73 (+12%)
		SVM	0.73 (+5%)	0.67 (+3%)	0.90 (+10%)	0.77 (+6%)
		LDA	0.80 (+15%)	0.78 (+16%)	0.83 (+10%)	0.80 (+13%)
	04	kNN	0.58 (+2%)	0.56 (+1%)	0.76 (+3%)	0.64 (+2%)
		SVM	0.73 (+12%)	0.67 (+7%)	0.90 (+24%)	0.63 (+14%)
		LDA	0.73 (+8%)	0.69 (+8%)	0.83 (+3%)	0.75 (+6%)
SUB2	05	kNN	0.62 (-3%)	0.60 (-2%)	0.70	0.65 (-1%)
		SVM	0.65 (+10%)	0.66 (+12%)	0.60 (-5%)	0.63 (+4%)
		LDA	0.70 (+3%)	0.72 (+7%)	0.65 (-10%)	0.68 (-1%)
	06	kNN	0.65 (-8%)	0.62 (-8%)	0.73 (-7%)	0.67 (-8%)
		SVM	0.73 (+3%)	0.70 (+2%)	0.80 (+7%)	0.75 (+5%)
		LDA	0.68 (-2%)	0.66 (-1%)	0.73 (-7%)	0.69 (-2%)
SUB3	07	kNN	0.80 (+13%)	0.77 (+11%)	0.85 (+15%)	0.81 (+13%)
		SVM	0.75 (+3%)	0.69	0.90 (+10%)	0.78 (+4%)
		LDA	0.72 (+7%)	0.69 (+6%)	0.80 (+10%)	0.74 (+8%)
	08	kNN	0.64 (-14%)	0.60 (-12%)	0.82 (-10%)	0.69 (-12%)
		SVM	0.71 (+4%)	0.67 (+2%)	0.82 (+7%)	0.74 (+4%)
		LDA	0.82 (+9%)	0.80 (+6%)	0.85 (+14%)	0.82 (+10%)

TABLE IV: The results of the online simulation where the initial dataset from each subject calibrates the models used to predict the movements in the second dataset. Calibration implies selecting the best-performing channels and training the model using subject-specific data. *True Positives* is the number of times the model fulfilled the freeze heuristic in the correct period. The *False Positive Rate* is the percentage of times the model incorrectly executed the movement of the robotic glove. MTNM is the mean time of the total positive predictions' distance to the nearest MRCP label. SUB2's LDA model achieved the lowest false positive rate and the lowest MTNM, while SUB1 had the three worst simulation runs.

SUB ID	Cal. dataset → sim. dataset	Classifier	Expected MRCPs	True Positives	Hit rate	Total Positive	False Positive rate	MTNM (Total)	MTNM (False Positives)
SUB0	01 → 02	kNN	28	23	82.1%	52	55.7%	1.44 s	2.59 s
		SVM		24	85.7%	31	22.5%	0.55 s	2.42 s
		LDA		14	50.0%	39	64.1%	1.31 s	2.05 s
SUB1	03 → 04	kNN	14	10	71.4%	45	77.7%	6.0 s	7.94 s
		SVM		8	57.1%	41	80.4%	6.09 s	7.81 s
		LDA		11	78.5%	45	75.5%	5.72 s	7.81 s
SUB2	05 → 06	kNN	15	10	66.6%	14	28.5%	0.37 s	1.72 s
		SVM		8	53.3%	17	52.9%	0.44 s	0.94 s
		LDA		11	73.3%	12	8.3%	0.26 s	3.14 s
SUB3	07 → 08	kNN	17	13	76.4%	39	66.6%	2.31 s	3.46 s
		SVM		16	94.1%	30	46.6%	2.52 s	5.39 s
		LDA		8	47.0%	26	69.2%	2.68 s	3.87 s

samples could improve the overall dataset quality and model performance but also require extensive domain knowledge. Our purpose was not to achieve the best possible result on each dataset but to show that the system can function with no human interaction in the training and online phases. Furthermore, we did not tune the freeze heuristic specifically for each dataset. The high false positive rate in the simulation results of SUB1 and SUB3 reflects that the system is too sensitive, which we could remedy by adjusting the freeze heuristic during simulation in the same manner as we adjust the dwell parameter in the existing procedure. However, tuning the dwell parameter is considered labor intensive and would introduce undesired manual labor to our simulation approach.

VII. Discussion

In Section II, we presented the existing procedure used in Aalborg University's BCI-Lab and subsequently identified a set of key challenges in Section III. However, there are still sources of error involved in the procedure, which we have not highlighted, that are important to be cognizant of, as they impact the movement classification. The sources of error can be associated with the data collection and data processing steps of the existing procedure. During data collection, we attribute certain faults and inconsistencies to the recording setup, including the subject, which increases data discrepancies and decreases the data's quality and viability.

Poor placement of electrodes affects the results, e.g., insufficient gel appliance can cause a poor connection between the electrodes and the scalp, but this is often detected and rectified prior to recording. However, electrodes can move during the experiment, altering the captured signal, often caused by the subject moving. This will impact data quality severely if it occurs during data acquisition. The movement can also happen between phases, e.g., between the training and online phases, which will impact the classifier's performance as it might not recognize the data captured by the shifted electrode(s). The setup consistency between sessions is also a significant challenge, also regarding the same subject. This includes the g.GAMMAcap2 placement, which can alter the signals if the electrodes are not placed in the exact location on the scalp between sessions.

There is a possibility of high modulation noise from session to session, which interferes with the amplifier. The electrical current from the ceiling lights or other electrical appliances connected to the same electrical network can be a cause of this. As a result, the modulation needs to be kept consistent during all sessions and to achieve this, we should, preferably, turn off all electrical appliances. However, we choose to keep the lights on to mimic a more real-world scenario where the system would be applied.

The existing system relies heavily on filtering to identify the MRCP's morphology. However, the filter parameters are not static across the literature, and many choose to filter different frequencies, increasing the parameters that the domain expert manages during the experiment.

Subject-related consistencies such as the subject's physical and psychological state, e.g., level of focus, also affect the signals. Eye blinks, eye movement, swallowing, and other facial movements disrupt the EEG signals, and if they occur during the movement execution, it can be troublesome to identify the morphology of the MRCPs. If the subject does not execute the motor task in a decisive ballistic fashion, it impedes the process of locating exactly when the movements began.

The challenges outlined above are often related to consistency when recording the training samples. However, they also highlight the difficulty of applying the system for online execution; if we do not mirror the conditions used in the data collection during online usage, the results may vary and likely deteriorate. To minimize factors that adversely affect the data acquisition and bring unwanted noise and inconsistencies to the signals, the data acquisition is performed in a lab's sterile environment and thus does not mimic real-world usage. However, we are cognizant of this being a baseline environment for early testing and development.

We have learned that there is a close connection between the physiological effects of the brain and how it affects the

data on which we apply our methods. In many cases, there are details requiring extensive domain expert knowledge. We have made our decisions to the best of our knowledge and based on the guidance and advice from domain experts.

It would be optimal to make an equal comparison between the existing system used in the BCI-Lab and our improved version to evaluate its effectiveness. Nevertheless, due to compatibility issues between our system and the existing system, we have not directly compared the two as we did with the movement onset detection in Section VI-A. As an alternative, we have compared our SLM approach to a baseline as we describe Section VI-B and use our metrics to determine the performance of our simulation in Section VI-C. Despite not being directly compared to the existing system, we still consider it feasible to use the SLM approach's results and our simulation to estimate their performance.

VIII. Conclusion

We propose alternatives to the existing procedure in use at Aalborg University Department of Health Science and Technology's BCI-Lab. We implement a clustering algorithm to achieve more consistent and automatic labeling of MRCPs in the data. We demonstrate that our algorithm detects several movements closer to the expected number and thus increases the number of training samples. Implementing an automatic labeling approach for the EEG data reduces the supervision required from domain experts and allows more data to be collected seamlessly. The same applies to our SLM method, where we propose a weighted-voting algorithm to enhance the MRCP samples of each epoch, which, in turn, also increases the performance of our models. We see a correlation between our more consistent labeling technique and higher classifier performance, which verifies our algorithm's robustness. We show that our methods work as part of a BCI by simulating an in-lab, online environment.

IX. Acknowledgments

We would like to thank the Health Science and Technology department at Aalborg University for access to their BCI-Lab, and especially Susan Aliakbary Hosseinabadi and Strahinja Dosen for their guidance throughout this project. We would also like to thank Dalin Zhang and Christian S. Jensen for their excellent supervision during the project.

References

[1] R. Abiri, S. Borhani, E. W. Sellers, Y. Jiang, and X. Zhao, "A comprehensive review of EEG-based brain–computer interface paradigms," *Journal of Neural Engineering*, vol. 16, no. 1, p. 011001, jan 2019. [Online]. Available: https://doi.org/10.1088/1741-2552/aaf12e

[2] S. Aliakbaryhosseinabadi, N. Jiang, A. Vuckovic, K. Dremstrup, D. Farina, and N. Mrachacz-Kersting, "Detection of movement intention from single-trial movement-related cortical potentials using random and non-random paradigms," *Brain-Computer Interfaces*, vol. 2, no. 1, pp. 29–39, 2015. [Online]. Available: https://doi.org/10.1080/2326263X.2015.1053301

[3] S. Aliakbaryhosseinabadi, D. Farina, and N. Mrachacz-Kersting, "Real-time neurofeedback is effective in reducing diversion of attention from a motor task in healthy individuals and patients with amyotrophic lateral sclerosis," *Journal of Neural Engineering*, vol. 17, no. 3, p. 036017, jun 2020. [Online]. Available: https://doi.org/10.1088/1741-2552/ab909c

[4] N. Jiang, L. Gizzi, N. Mrachacz-Kersting, K. Dremstrup, and D. Farina, "A brain–computer interface for single-trial detection of gait initiation from movement related cortical potentials," *Clinical Neurophysiology*, vol. 126, no. 1, pp. 154–159, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1388245714002521

[5] A. M. Savić, S. Aliakbaryhosseinabadi, J. U. Blicher, D. Farina, N. Mrachacz-Kersting, and S. Došen, "Online control of an assistive active glove by slow cortical signals in patients with amyotrophic lateral sclerosis," *Journal of Neural Engineering*, vol. 18, no. 4, p. 046085, jun 2021. [Online]. Available: https://doi.org/10.1088/1741-2552/ac0488

[6] S. Aliakbaryhosseinabadi, S. Dosen, A. M. Savic, J. Blicher, D. Farina, and N. Mrachacz-Kersting, "Participant-specific classifier tuning increases the performance of hand movement detection from EEG in patients with amyotrophic lateral sclerosis," *Journal of Neural Engineering*, vol. 18, no. 5, p. 056023, sep 2021. [Online]. Available: https://doi.org/10.1088/1741-2552/ac15e3

[7] I. K. Niazi, N. Jiang, M. Jochumsen, J. F. Nielsen, K. Dremstrup, and D. Farina, "Detection of movement-related cortical potentials based on subject-independent training," *Medical & Biological Engineering & Computing*, 2013.

[8] H. Shibasaki and M. Hallett, "What is the bereitschaftspotential?" *Clinical Neurophysiology*, vol. 117, no. 11, pp. 2341–2356, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138824570600229X

[9] G. Pfurtscheller and C. Neuper, "Future prospects of erd/ers in the context of brain–computer interface (bci) developments," in *Event-Related Dynamics of Brain Oscillations*, ser. Progress in Brain Research, C. Neuper and W. Klimesch, Eds. Elsevier, 2006, vol. 159, pp. 433–437. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079612306590284

[10] Y. Jeon, C. S. Nam, Y.-J. Kim, and M. C. Whang, "Event-related (de)synchronization (erd/ers) during motor imagery tasks: Implications for brain–computer interfaces," *International Journal of Industrial Ergonomics*, vol. 41, no. 5, pp. 428–436, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169814111000540

[11] Z. Tang, S. Sun, S. Zhang, Y. Chen, C. Li, and S. Chen, "A brain-machine interface based on erd/ers for an upper-limb exoskeleton control," *Sensors*, vol. 16, no. 12, p. 2050, 2016.

[12] G. Pfurtscheller and F. Lopes da Silva, "Event-related eeg/meg synchronization and desynchronization: basic principles," *Clinical Neurophysiology*, vol. 110, no. 11, pp. 1842–1857, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1388245799001418

[13] G. Pfurtscheller, "Functional brain imaging based on erd/ers," *Vision Research*, vol. 41, no. 10, pp. 1257–1260, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0042698900002352

[14] A. Seeland, L. Manca, F. Kirchner, and E. A. Kirchner, "Spatio-temporal comparison between erd/ers and mrcp-based movement prediction." in *BIOSIGNALS*, 2015, pp. 219–226.

[15] MATHLAB. filtfilt zero-phase digital filtering. Accessed: 15-10-2021. [Online]. Available: https://se.mathworks.com/help/signal/ref/filtfilt.html

[16] BioSPPy. Biosppy biosignal processing in python. Accessed: 14-10-2021. [Online]. Available: https://biosppy.readthedocs.io/en/stable/index.html

[17] NeuroKit2. Introduction. Accessed: 28-10-2021. [Online]. Available: https://neurokit2.readthedocs.io/en/latest/introduction.html

[18] F. Fahimi, S. Dosen, K. K. Ang, N. Mrachacz-Kersting, and C. Guan, "Generative adversarial networks-based data augmentation for brain–computer interface," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 9, pp. 4039–4051, 2021.

[19] D. Zhang, L. Yao, X. Zhang, S. Wang, W. Chen, and R. Boots, "Cascade and parallel convolutional recurrent neural networks on eeg-based intention recognition for brain computer interface," 2021.

[20] J. Meng, S. Zhang, A. Bekyo, J. Olsoe, B. Baxter, and B. He, "Non-invasive electroencephalogram based control of a robotic arm for reach and grasp tasks," *Scientific Reports*, vol. 6, no. 1, pp. 1–15, 2016.

[21] S. Rimbert, O. Avilov, and L. Bougrain, "Discrete motor imageries can be used to allow a faster detection," in *7th Graz Brain-Computer Interface Conference 2017*, 2017.