

# A Brain-Controlled Exoskeleton with Cascaded Event-Related Desynchronization Classifiers

Kyuhwa Lee<sup>1</sup>, Dong Liu<sup>1,2</sup>, Laetitia Perroud<sup>1</sup>, Ricardo Chavarriaga<sup>1</sup>, and José del R. Millán<sup>1</sup>

<sup>1</sup>Defitech Chair in Brain-Machine Interface lab, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

<sup>2</sup>School of Automation Science and Electrical Engineering, Beihang University (BUAA), China

## Abstract

This paper describes a brain-machine interface for the online control of a powered lower-limb exoskeleton based on electroencephalogram (EEG) signals recorded over the user's sensorimotor cortical areas. We train a binary decoder that can distinguish two different mental states, which is applied in a cascaded manner to efficiently control the exoskeleton in three different directions: walk front, turn left and turn right. This is realized by first classifying the user's intention to walk front or change the direction. If the user decides to change the direction, a subsequent classification is performed to decide turn left or right. The user's mental command is conditionally executed considering the possibility of obstacle collision. All five subjects were able to successfully complete the 3-way navigation task using brain signals while mounted in the exoskeleton. We observed on average 10.2% decrease in overall task completion time compared to the baseline protocol.

# A Brain-Controlled Exoskeleton with Cascaded Event-Related Desynchronization Classifiers

## I. INTRODUCTION

With a potential to offer more dexterous functionalities than wheelchair, various types of exoskeletons have been actively developed in recent years [1]. Exoskeletons are wearable robots exhibiting a close physical interaction with the human user. Among them, powered upper-limb and lower-limb exoskeletons have been given attention as the potential technology to assist paraplegic or tetraplegic population. Throughout the past decade, brain-machine interface (BMI) systems have been the focus of research on improving the quality of life of people having severe motor disabilities (e.g. [2]–[4]). As an emerging technology, researchers have shown the possibility of controlling exoskeletons and neuroprosthetic devices using the non-invasive electroencephalography (EEG) which captures the brain signals from electrodes placed on a scalp (e.g. [5]–[8]). EEG is still regarded as the only practical and realistic non-invasive BMI method at present [9] because other imaging modalities such as functional magnetic resonance imaging (fMRI), magnetoencephalography (MEG) and positron emission tomography (PET) are quite expensive [10], technically demanding and not easily portable.

A BMI system can be controlled using either endogenous (spontaneous) or exogenous (evoked) signals. In exogenous BMI, evoked signals appear when a person senses external stimuli such as visual or auditory cues. The advantage of this approach includes minimal training, high bit rates up to 60 bits/min [11], [12]. However, the user needs to always attend to the stimuli which limits its applicability. Also, the user can become quickly tired due to the strong stimuli. Typical examples include modulations of steady state visually evoked potential (SSVEP) [13] and P300-based interfaces [14]. In endogenous BMI, on the other hand, control signals are generated independently from any external stimulation and can be fully operated voluntarily by the user. It is also useful for users who have sensory impairments while providing a more natural and intuitive means of interactions since users can spontaneously command the neuroprosthesis [15]. It typically requires, however, longer training sessions and the bit rate is usually lower. Examples include slow cortical potentials (SCP) [16], [17] and sensorimotor rhythms such as motor imagery (MI) [18].

Although tested with healthy subjects as a proof of concept in this work, we aim our ultimate target group as people with tetraplegia, where no or limited residual motor control is available. Even with the latest advances in non-invasive BMI, there still remains many challenging problems to achieve this goal outside of the lab environment. The low signal-to-noise ratio of EEG signals and its high sensitivity to electrical noise is a major limiting factor of decoding performance. More importantly, a higher level of cognitive load is required by the user to operate a BMI when the user is mounted in a mobile platform compared to operating a BMI in a stationary environment. Electromyography (EMG) suffers less from these problems but it requires residual motor control which makes it not suitable in our case.

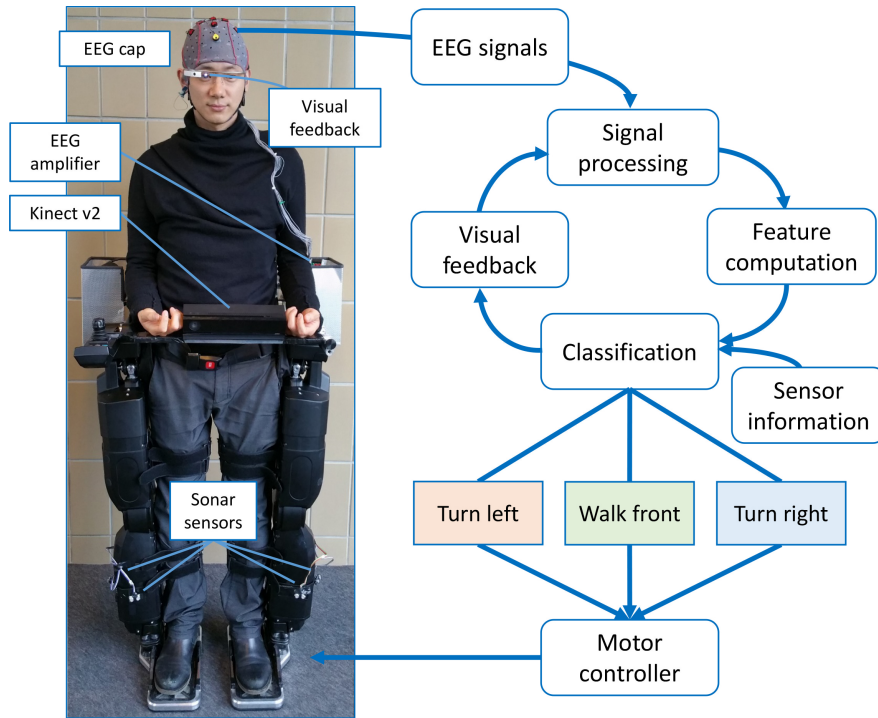


Fig. 1: The modified version of Rex exoskeleton (left) and the overview of the brain-controlled exoskeleton framework (right).

We are interested in the online control of a lower-limb exoskeleton using the endogenous brain signals as the user needs to spontaneously deliver commands based on his/her own intentions and the context of the surrounding environment. The online control of a lower-limb exoskeleton has some extra difficulties compared to other mobile BMI systems, e.g. wheelchairs, because of the higher level of fatigue the subjects experiences due to the vertical standing posture and the physical exercise. In addition, it involves a relatively higher level of risk in mechanical stability. Furthermore, the fact that its locomotion is composed of multiple atomic actions, e.g. foot lift or foot turn, a mechanism to ensure the synchronization between the user’s decision making and the robot’s movement is needed. In addition to the difficulties mentioned above, the online control of a lower-limb exoskeleton while mounted in it makes the user harder to focus and easily tired because of the non-linear body movements involved in walking. To tackle these problems, we propose a BMI decoding approach that aims at achieving: (i) high classification accuracy, (ii) short calibration time, and (iii) low cognitive workload. This approach is based on the following components:

- 1) Multiclass classification using cascaded binary classifiers.
- 2) Binary classification based on the detection of motor imagery.
- 3) The execution of mental commands considering environmental context.

Compared to the multiclass MI-based classification methods commonly used in endogenous BMI [5], [19]–[21], e.g. left versus right hand imagery or hand versus feet imagery, our classifier discriminates the imagery of moving both hands versus resting. We exploit the well-known event-related desynchronization (ERD) effect which can be observed in the EEG signal immediately after the user starts imagining the movement of one or more body parts

[22]. We show that as few as 10 trials of motor imagery are sufficient to train a classifier compared to the typical lengthy training required for multi-class MI decoders [11]. The binary classification is performed in a cascaded manner to make the final decision. In a 3-way MI classification protocol, the robot turns only when the left or right motor command is delivered and the implicit motor command of moving front is continuously executed while the user deliberately avoids sending any motor command. We show through our preliminary experiments that our protocol is well suited for controlling an exoskeleton online.

To the best of our knowledge, this is the first work on the online control of walking and turning of the lower-limb exoskeleton for navigation based only on EEG signals. The only comparable work is NeuroRex [23] in which the online control was used to detect the intention of a user to start or stop walking. Here, we show 5 healthy subjects successfully controlling the exoskeleton to complete a navigation task that requires the decoding of 3 different motor commands online – walk front, turn left and turn right – in order to reach a target with high accuracy while avoiding obstacles.

## II. BACKGROUND

### A. State-of-the-art lower-limb exoskeletons

With the advances in materials, actuators, sensors and computer size, many wearable exoskeletons have become real-life products. Some examples of lower-limb exoskeletons include Rex (Rex Bionics) [24], ReWalk (ReWalk Robotics) [25], Ekso (Ekso Bionics) [26], HAL (Cyberdyne) [27], ALEX (University of Delaware) [28], X1 (NASA) [29], and Indego (Vanderbilt University) [30]. We use Rex as our testing platform which can ultimately be used on people with severe motor disabilities.

Rex is a self-balancing, battery-powered lower-limb exoskeleton that can perform basic functions with a joystick such as walking front, back and side, turning left and right, sitting down and standing up. Aimed for paraplegic users, although potentially it can be also used by people with tetraplegia, it is the only lower-limb robotic exoskeleton currently available that can move independently without the support of the user. Compared to Rex, ReWalk is a lighter exoskeleton which is more suited for people who can balance and stand themselves with intact hands, arms and shoulders. Ekso is an exoskeleton that is also aimed for paraplegia who can balance his or her upper body and shift weights while walking. It is equipped with a remote controller so that a physical therapist can control the device. ALEX is a gait rehabilitation device which can apply just the right amount of force on the leg to help it move along the desired path. However, ALEX is limited to be used on a treadmill. HAL combines a voluntary control system based on surface electromyography (EMG) with autonomous control to assist the user in performing various types of actions. It requires upper body function. Indego is a modular exoskeleton weighing less than 13kg which allows users to stand and walk. Nasa's X1 assists or resists human movement through the use of actuators positioned at the hips and knees. It was primarily developed as an assistive exercise technology, which requires an upper body functionality.

### B. Brain-machine interfaces for assistive technologies

The Electroencephalography (EEG) is a classical non-invasive method for measuring a person's brain signals. The electrodes placed on the scalp capture the signals that have the amplitude in the range of microvolts in real time. In the development of EEG-based BMIs, individually tuned parameters for characterizing signals are computed from several training trials, which are used to perform online signal decoding. As the new research in the recent years has improved the decoding capability of real-time brain signals, there is an increasing interest in combining BMI technology with the existing assistive technologies [3], [31].

Many BMI systems have been used to provide control in computer applications. The real-world applications which involve the interaction with physical devices still need more thorough investigation. On the one hand, there is a need for more reliable, wearable EEG recording technologies [32]. On the other hand, there is the need to develop and test interaction approaches robust enough for practical applications. Tavella *et al.* [6] demonstrated the possibility of healthy subjects mentally controlling a non-invasive BMI neuroprosthesis for the restoration of grasping while performing multiple tasks in a hand-writing application. Leeb *et al.* [33] showed how users can mentally control a telepresence robot with BMI to perform a navigation task in daily environments. Carlson *et al.* [5] and Millán *et al.* [21] demonstrated the control of an intelligent wheelchair with BMI that can navigate in a room by jointly utilizing the mental command of the user and the environment information. Müller-Putz *et al.* demonstrated a patient controlling a neuroprosthesis to perform a grasp action [7]. NeuroRex [23] studies the potential of controlling Rex using brain signals. In their work, the online control was performed for delivering *walk front* vs *stop walk* actions with a single paraplegic user while the turning classification was tested in offline analysis.

## III. METHODS

In the training phase, the preprocessed EEG signals are transformed into feature representation, which are then used to build a classifier (Section III-B). During the testing phase, the likelihoods computed by the classifier are accumulated over time, represented as the final decision score (Section III-B). The real-time visual feedback is given to the user through a see-through screen (Google Glass, Figure 1) so that the user knows how his/her brain signals are being decoded and what will be the next action executed by the robot.

For feature computation, we exploit the ERD explained in Section I which can be reliably observed in EEG signals when the subject performs MI. The details can be found in Section III-B. For classification, we use Random Forests (RF) classifiers [34] which are resistant to over-fitting even at low trials-to-features ratio, robust to outliers, and capable to build a complex model for capturing non-linear relationships while running in (near) real-time. RF classifiers have been recently applied successfully in decoding EEG signals for both classification [35]–[37] and data mining purposes [38] as well as in other fields of science such as data mining [39], gene analysis [40] and computer vision [41]–[43].

We use g.Tec USBamp to sample the signals in 512 Hz from 16 active Ag/AgCl electrodes placed on the scalp according to the extended 10/20 system [44], which normally takes 10-15 minutes to prepare. See Figure 7 for their physical locations. Our classification runs in approximately 15 Hz ( $\sim 62.5$  ms), where the most of the time is spent on computing power spectral density of EEG signals. We use a laptop configured with core i7 (2.5 GHz)

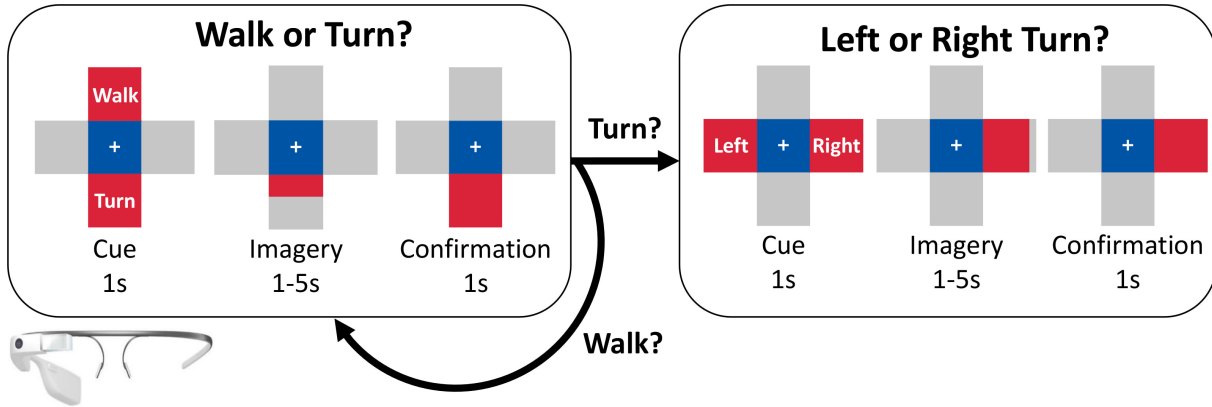


Fig. 2: A cascaded binary classification protocol. This is the visual feedback that a user sees on the see-through screen. The *Walk vs Turn* classification is performed during the first stage. If the red bar reaches either the top or remains in the upper area after timeout (5s), a walk motor command is delivered to the robot. Otherwise, *Turn* intention is detected and *Left vs Right* classification is performed. It has the same behavior as in the first classification step except that the bar moves left or right.

and 16 GB RAM. Visual processing, Rex control and ultrasonic sensor processing modules are implemented in C++ with multithreading, while signal acquisition, signal processing, machine learning and visualization modules are implemented in Python with multiprocessing support. Google Glass visualization app is implemented in Java.

We made some adaptations to the original Rex to perform our experiments. First, the shoulder component was removed and a metal plate was added to mount a laptop on the back and two baskets on each side for holding an amplifier, cables and batteries. Similar to previous developments in neuroprosthetics [5], the exoskeleton was endowed with sensors that provide information about the environment. For this purpose, another plate for mounting Kinect v2 camera in the front was added. 4 pairs of ultrasonic sensors controlled by Arduino Mega [45] have been attached to the body of Rex. Rex Bionics provided a USB wireless control box which can be used to control the Rex wirelessly from a computer implementing the BMI. Figure 1 shows our modified version of Rex.

#### A. Protocol

While mounted inside Rex, a subject follows the visual cue shown on the see-through wearable screen of Google Glass. During the training stage, 10 trials of each of the following two tasks are performed for 5 seconds each, one after the other:

- 1) Move: Imagine moving both hands (5 seconds)
- 2) Relax: Imagine relaxing both hands (5 seconds)

During the *Move* state, the subjects are instructed not to move any body part including eyes. During the *Relax* state, they are told not to move any body part and focus only on relaxation of the muscles. Subjects are given a resting period of 10 seconds between these trials to freely move their body and rest.

During the online testing stage, the subject can deliver one of three mental commands to the robot: Walk front, Turn left, and Turn Right, which is then immediately executed by the robot. The robot makes one atomic action for each mental command and the classification is repeated again. However, the robot ignores the delivered motor

command if its execution will cause the robot to hit a nearby object or wall detected by the on-board sensors (See Section III-C).

Our cascaded protocol is defined as following, c.f., Figure 2.

- 1) Cue (1s): Indicate which actions are available.
- 2) Imagery (1-5s): Classification of *Walk vs Turn* is performed and feedback is provided in the top and bottom bars. The bar length represents the decoding confidence and the direction represents a corresponding class. Section III-B explains how the confidence is computed. Decision is made if the bar reaches either end of the cell or timeout (5s) occurs. *Walk* action is executed as soon as the bar reaches the end of the upper cell or remains in it. Otherwise, *Turn* intention is detected and a second binary classification, *Left vs Right*, is performed. During the *Left vs Right* classification, bar moves either left or right, where left and right cells correspond to *Left turn* and *Right turn*, respectively.

- 3) Command feedback (1s): Indicate the decision.

We compare our protocol with another commonly used 3-way classification protocol, similar to [5], as a baseline. In this baseline protocol, the robot makes a left turn if the bar reaches the left end and right turn if the bar reaches the right end before the timeout occurs. If the bar is intentionally kept from reaching either side of the ends until timeout occurs, i.e. intentional non-control, the robot executes a *Walk* action after the timeout. In our protocol, instead of using an intentional non-control to distinguish three mental commands, we use 2 layers of cascaded binary classifications to distinguish them.

## B. BMI Decoder

**Signal preprocessing.** The EEG signals, typically ranging in an amplitude of microvolts, are captured from active electrodes and amplified through an EEG amplifier and digitized. We apply common average reference (CAR) spatial filtering on the raw input signals to improve the signal-to-noise ratio [46]. For every sample time, CAR subtracts the mean value of all electrodes, which minimizes the uncorrelated random noise with a zero mean through the averaging process [47].

**Feature computation.** The conventional ERD analysis measures the amount of power decrease in the band-passed signals of a single electrode [22] compared to a short time in the past, i.e. a baseline. In our work, however, we use signals from all electrodes and compute the averaged power spectral density (PSD) from each electrode between 14 and 19 Hz. For both training and testing, a sliding window of length 0.5 s is used. We use multitaper PSD [48] implemented in Python MNE library [49]. Multitaper PSD is highly effective in reducing the estimation bias by averaging over multiple independent estimates from a given sample [48]. The averaged PSD from all electrodes are concatenated to form a feature vector of dimension equivalent to the number of electrodes. It is important to note that although we do not explicitly perform feature selection, the importance of features are ranked and taken into account while training the Random Forest classifier based on Gini index scores of tree nodes. A similar feature selection approach is reported in [38].

**Classification.** For training, we run the sliding window with a stepping time of 62.5 ms (16 overlapping windows) in the time intervals of interest where a PSD is computed from each electrode. For both classes of *Move* and *Relax*,

we use the time interval between 1 and 4 seconds relative to the onset cue. We ignore the data up to 1 second because it usually takes some time for a subject to start the imagery after confirming the cue. A RF classifier with 1000 trees is trained from these features with a depth limit 100. During the online testing stage, we apply the trained RF classifier after a feature is computed from the current window. Since a single window is not reliable due to the highly noisy nature of the EEG signal, the computed likelihoods  $P_c(t)$  of a class  $c \in \{Move, Relax\}$  at current sampling time  $t$  are accumulated over time with a scoring function  $S_c(t)$ :

$$S_c(t) = \alpha S_c(t-1) + (1-\alpha)P_c(t), \quad S_c(0) = 0.5 \quad (1)$$

where  $0 \leq \alpha < 1$  is a damping factor to smooth out the noisy output of the classifier, which is obtained heuristically [50]. It is commonly set between 0.8 and 0.9 in practical applications, depending on the reliability of a single prediction. We set  $\alpha = 0.8$  in our experiments.  $S_{Move}$  and  $S_{Relax}$  denote the likelihoods of being in the *Move* and the *Relax* states, respectively. Now we define our decision function  $D(t)$ :

$$D(t) = D(t-1) + \beta(S_{Move}(t) - S_{Relax}(t)), \quad D(0) = 0 \quad (2)$$

A decision is made and delivered to the robot if  $|D(t)| \geq L$  or  $t \geq t_{timeout}$ , where  $L$  is the length of a bar. The speed parameter  $\beta$  controls the bar speed which is adapted individually to each user in the training stage as a good BMI user can move the bar more quickly than an average user. While  $\alpha$  is used to reflect the reliability of the classifier,  $\beta$  is used to control the length of visual feedback duration that a user feels most comfortable with. We set  $t_{timeout} = 5$  seconds. The final decision is labeled as *Move* if  $D(t) > 0$ , or *Relax* otherwise. For example, in Figure 2,  $L$  corresponds to the length of a gray bar in pixels, and  $D(t)$  corresponds to the length of a growing red bar in pixels which gradually fills the gray bar from the center.

### C. Obstacle detection

For safety reasons, we implemented obstacle avoidance based on the added sensors so that the robot does not crash into any object. We estimate the likelihoods of obstacles around Rex using Kinect v2 and 4 pairs of ultrasonic sensors. Each ultrasonic sensor pair (transmitter/receiver) is mounted as in Figure 1. We apply median filtering to remove occasional outliers in the ultrasonic readings. The ultrasonic sensors are mainly used to detect close obstacles ( $\sim 20$ cm for each side,  $\sim 60$ cm for front) that cannot be detected by the visual sensor. We use the depth camera of Kinect v2 to detect obstacles in the surrounding space of the exoskeleton. This camera has a coverage between 40cm to 450cm and has a resolution of 521 x 424 pixels with a field-of-view of  $70.6^\circ \times 60^\circ$ .

**Preprocessing.** We obtain the intrinsic camera parameters using Zhang’s method [51] for reducing the image distortion using OpenCV library [52]. The depth image is downsampled to 0.4 times the original size and 2D Gaussian filtered to reduce noise.

**Floor and object detection.** We first detect the floor area and mark any other segmented object as obstacle. At the time of writing, Kinect SDK’s native floor plane estimation does not work well when the robot is walking. The reason that Kinect SDK function is not suitable for us is because it keeps recalibrating whenever there is a tilt



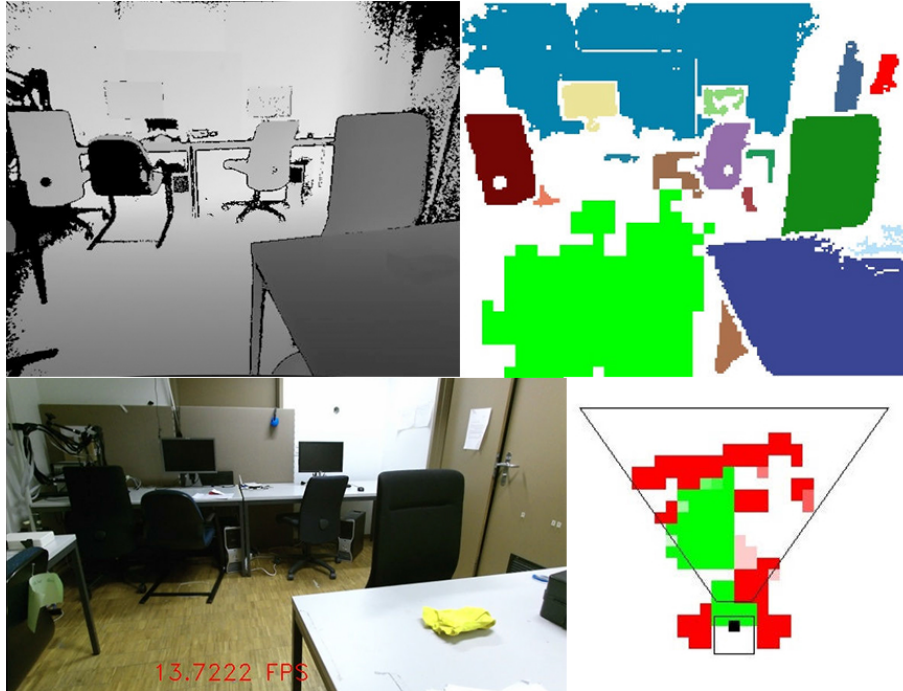


Fig. 3: Depth map acquired from Kinect v2 (top-left) and the segmented result (top-right). The bright green color represents the detected ground region. RGB image (bottom-left) is shown for reference. The occupancy grid (bottom-right) shows the detected obstacles in red color where higher color intensity means higher certainty. The depth camera can detect obstacles in the region marked by the inverse triangle, while the obstacles on the sides of the robot are detected by the ultrasonic sensors.

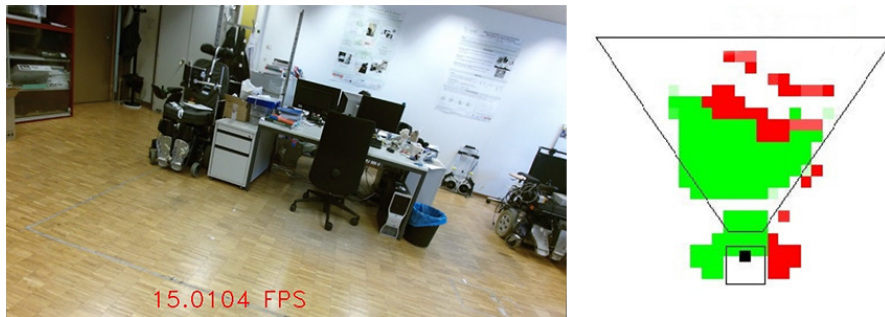


Fig. 4: Ground detection while the robot is walking. It can be seen that the tilt angle of the camera is high during the exoskeleton's locomotion which makes the problem more challenging.

motion on Kinect and it takes several seconds to settle. Due to this reason, we implement the floor detection using the following method.

The rescaled depth image is divided into cells of size  $5 \times 5$  pixels. Starting from several cells at the bottom line, a cell is marked as ground if it satisfies two conditions: 1) the depth value continuously increases compared to the cells in the lower line, 2) the normal of the cell points towards the y-axis. We use RANSAC for computing the normal similar to [53] but with Least Trimmed Squares [54] instead of Minimum Description Length as it was simpler to implement. Within each cell, 10 sets of 3 points are randomly selected. For each set of 3 points, the

normal is computed by first finding the corresponding 3D world coordinates and then taking the cross product of two vectors formed by these 3 points. Hence, each set of points determines its own plane. To find which one is the most representative of the whole cell, the residual squared error is computed on each pixel of the cell and the lowest  $h/2$  errors are summed together ( $h$  is the total number of pixels in the cell) to obtain the trimmed sum of the error. The normal of the plane with the lowest trimmed error sum is chosen and normalized.

If the normalized vector is close to  $(0, 1, 0)^T$ , the cell passes the criteria. In practice, however, we allow some error to compensate the roll movement of the camera due to the robot's movement. Some examples of ground detection can be seen in Figure 4. Once the ground is detected, the objects are segmented from the downscaled depth map by grouping regions that share the similar depth with smooth continuity. Segments that are too small are ignored. Since we do not aim to do an object recognition, this segmentation is sufficient for our purpose.

**Obstacle occupancy grid.** We represent a 2D occupancy grid having 44 x 44 cells, where each cell represents a square of 23.5cm in real world. We chose this cell size to map with the single step of Rex. The Kinect is mapped to the center of the grid.

#### IV. EXPERIMENTS AND ANALYSIS

##### A. Experiment Scenario

In our work, we run both protocols in two different scenarios having different difficulty levels as shown in Figure 5. Due to the slow speed of the Rex (4-8 seconds per action), we design scenarios such that each trial lasts between 4 and 8 minutes which requires 20-30 motor commands to finish. In the first scenario (scenario A), there are two obstacles placed on each side but no obstacle in the center. Although it may seem straightforward, it requires around 20 correct decisions to reach the goal following the optimal path. The second scenario (scenario B) has an additional obstacle in the center which requires the user to pass between 3 obstacles placed 1.5m apart from each other, where the user decides which path to take. After running a few practice runs until the user feels comfortable, each subject performs one trial each for both protocols in two different scenarios (4 trials in total).

##### B. Evaluation and Analysis

We test our protocol followed by the baseline protocol explained in Section III-A with 5 healthy subjects (1 female/4 males, mean age  $29.2 \pm 6.2$ ) on our modified version of Rex (Figure 1). We ensured that we have enough resting period between two protocols. Our classification method is used in both protocols. All subjects had previous experiences with the standard motor imagery (*left hand vs right hand movement* or *hand vs feet movement*) but none of them had any experience with our protocol (*both hands movement vs relaxing*). The electrode placement is shown in Figure 7.

Figure 6 shows the completion times for each subject. It is important to note that the cascaded protocol has potential overhead in time because it requires two consecutive classifications (*Walk vs Turn* and *Left vs Right*). Nevertheless, the average completion time for scenario A was 285 seconds with the cascaded protocol, which is almost 1 minute difference compared to 341 seconds with the baseline protocol. This is due to the lower error rate during the classification. Once an error is made, e.g. Rex turns right where it had to go straight, it needs at least

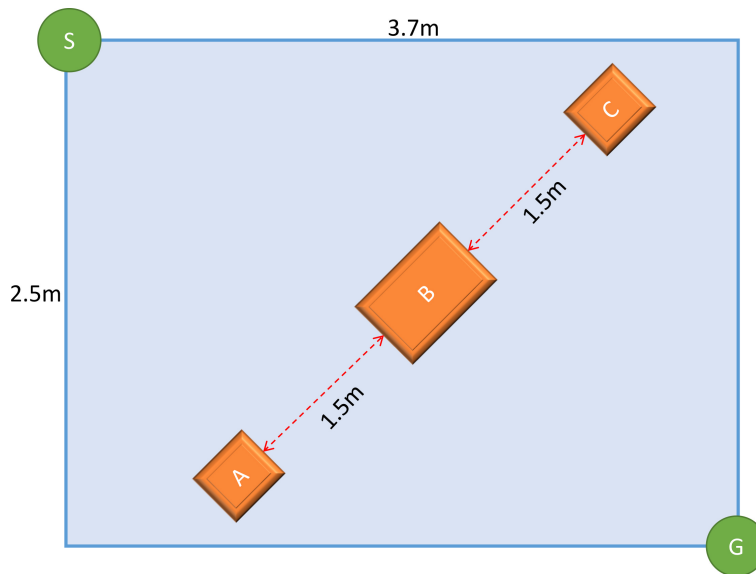


Fig. 5: Experiment scenarios. The user starts a trial in the circle marked as “S” while facing towards the goal circle “G”. A trial is finished when one of the feet of Rex touches the goal circle which has a diameter of 30cm. Three obstacle positions are marked as “A”, “B” and “C”. In scenario A, obstacles A and C are present. In scenario B, obstacle B is also added.

TABLE I: Error estimates for scenario A. The optimal path requires 16 correct steps.

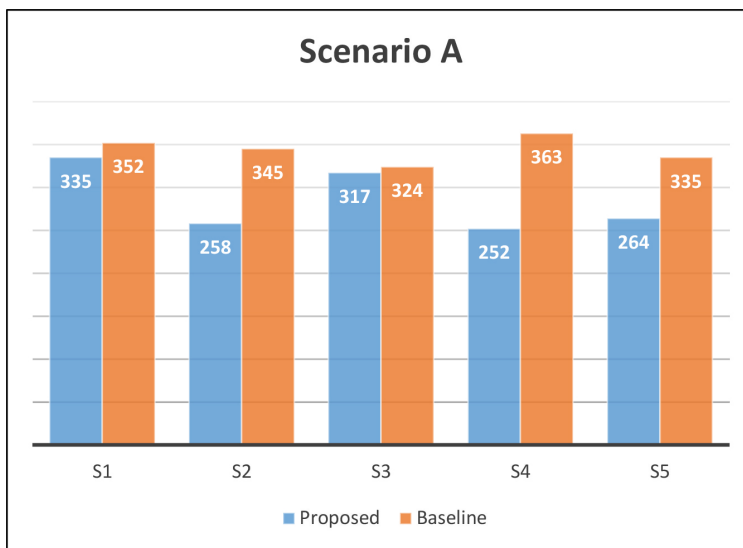
ID	Proposed			Baseline		
	Errors	Decisions	Rate	Errors	Decisions	Rate
s1	2	19	0.11	2	21	0.10
s2	0	16	0.00	3	21	0.14
s3	3	20	0.15	4	21	0.19
s4	0	16	0.00	6	21	0.29
s5	1	18	0.06	3	20	0.15
Total	6	89	0.07	18	104	0.17

TABLE II: Error estimates for scenario B. The optimal path requires 20 correct steps.

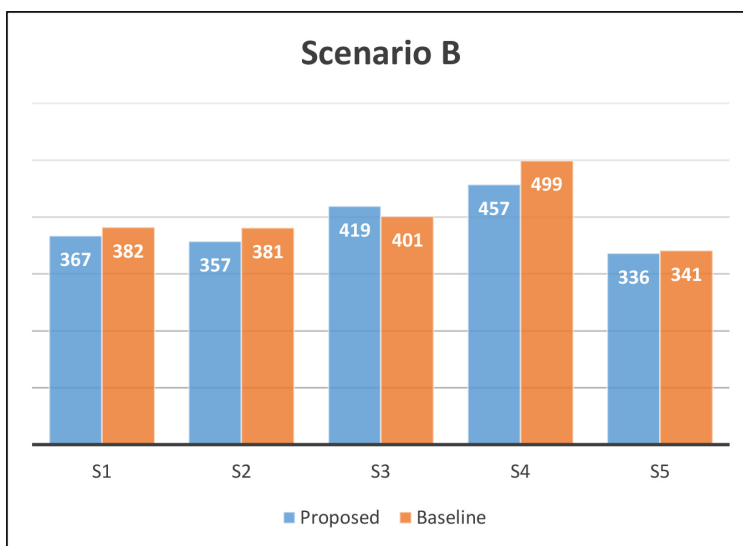
ID	Proposed			Baseline		
	Errors	Decisions	Rate	Errors	Decisions	Rate
s1	1	22	0.05	4	25	0.16
s2	2	22	0.09	4	25	0.16
s3	3	23	0.13	5	24	0.21
s4	3	25	0.12	5	26	0.19
s5	1	21	0.05	1	22	0.05
Total	10	113	0.09	19	122	0.16

one or more actions to recover. In scenario B, which has a longer path than scenario A, the cascaded protocol again showed lower average completion time than the baseline protocol (385 and 404 seconds, respectively).

It should be noted that we do not know the ground truth label of each decision the user had made as each motor command is voluntarily chosen by the user. So instead we estimate a BMI classification error indirectly by counting how many times the chosen action led the robot to move farther from the goal, i.e. assuming the user



(a)



(b)

Fig. 6: Task completion time in seconds for each subject between two protocols.

TABLE III: Three most informative electrode locations in order for each subject during the training phase. This ranking was computed using Gini index of tree nodes of the Random Forest.

s1	s2	s3	s4	s5
C2, C4, FC2	C4, C2, FC4	C1, CP2, C4	C2, FC2, CP1	C4, C1, CP4

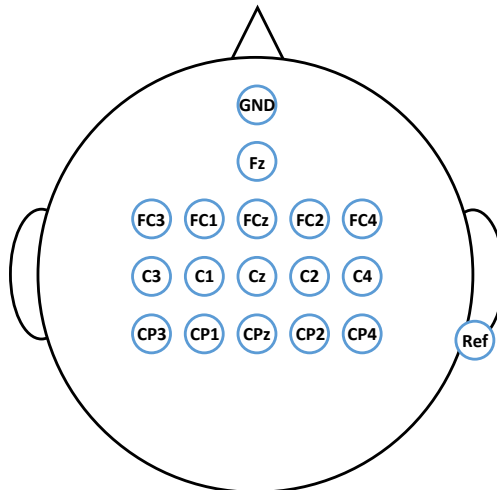


Fig. 7: The placement of electrodes on the scalp when seen from the top. We use g.GAMMAcap having Ag/AgCl wet electrodes.

was always delivering the optimal command. It is, however, not uncommon that the user gets confused on what motor command needs to be delivered, e.g., because of the loss of attention or due to the error on path planning. Therefore, the estimated errors may be interpreted as an upper bound on the classification error.

In general, as can be observed in Tables I and II, the estimated error rate is lower in the proposed protocol. Note that the total number of decisions can vary even with the same number of errors, as it depends on how long it took for the user to steer back to the optimal path after making an error. For example, the user can make an error in the beginning and can gradually diverge from the optimal path until he or she realizes that the robot needs to get back to the optimal path. It is also worth mentioning that the time does not entirely depend on the number of errors or decisions made because the classification time can vary between 1 to 5 seconds, depending on the situation. In addition, it takes a shorter time for Rex to turn ( $\sim 4s$ ) than to walk ( $\sim 8s$ ).

We report the three most informative features, i.e. electrode locations, of each subject in Table III. Although most of the electrodes are placed around the center of scalp to minimize the chances of having artifact contamination (see Figure 7 for electrode placement), there is still a possibility of having a performance degrade due to an artifact. For example, electrode Fz can be easily affected by unconscious eye movements, whereas central electrode (Cz, C1, C2...) are much less likely. The results in Table III show that central electrodes (C1, C2, C4) dominate among the selected top features. Electrodes FC2 and FC4 are always associated with C2 or C4, respectively, which suggests that they are not caused by a random artifact but intentional modulation of sensorimotor rhythms. It is worth noting that the subjects were explicitly instructed to not move their eyes as much as possible during the classification.

In post-experimental interviews, subjects reported that they had to put more attention on the visual feedback to check the bar location while executing the baseline protocol. It is because to walk front, they had to continuously balance the position of the bar to be centered around in the middle to keep it from reaching the border. Furthermore, since turning the robot requires sustained focusing to make the bar reach the end of a cell, users were more likely

to get tired. In the cascaded protocol, however, since the bar just needs to be kept within a cell (not reaching the end), there seems to be less burden for the user even though it can potentially take longer time due to overhead.

Compared to a wheelchair which allows the user to move continuously, each stepping action of Rex is discrete and atomic in a sense that an action cannot be stopped or modified until the current action is finished. This means that a single error can lead to a large amount of time delay, potentially increasing the user frustration. Due to this reason, a protocol that provides higher accuracy even though it has time overhead seems to fit better for the purpose of realizing brain-controlled lower-limb exoskeletons.

## V. CONCLUSIONS & FUTURE WORK

We have shown a working protocol for the purpose of navigating with a powered lower-limb exoskeleton using only EEG signals. It is also endogenous BMI, which can be voluntarily controlled in full by a user compared to exogenous BMIs which require external stimuli. We have demonstrated the effectiveness of the cascaded binary classifiers enabling users to efficiently deliver the three motor commands by thinking: walk front, turn left and turn right. This is realized by first detecting a user's intention to keep walking front or turn the direction based on a binary classification. Once the user's intention of turning is detected, a subsequent binary classification is performed to turn the robot left or right.

Our contribution is twofold. First, we experimented with a classification method based on the binary classification of *Move* and *Relax* instead of a more standard motor imagery control, e.g. *Left hand vs Right hand*. Second, we presented a BMI framework based on cascaded binary classifiers for controlling a lower-limb exoskeleton. We have demonstrated that our approach is well-suited for controlling exoskeletons by achieving high accuracy as shown in Tables I and II, while reducing the cognitive burden of a user.

There are some interesting possible extensions to our current preliminary work. First, error-related potential [55], [56], which is observed when the user gets frustrated due to the wrong decision, can be exploited as a verification tool before delivering a motor command to the robot for more robust classification. Second, instead of a synchronous classification we used in this experiment, an asynchronous classification which continuously classifies the user's intention *during* the motor execution can be considered. It is an interesting but challenging problem since several types of movement artifact should be thoroughly dealt to achieve this goal. Third, the performance may increase by individually tuning the frequency range used for feature computation. Last but not least, testing with a larger number of subjects and experimental conditions would be helpful to better confirm the effectiveness of our protocol.

## ACKNOWLEDGMENT

We acknowledge Rex Bionics and EPFL mechanical workshop for helping on the hardware modification of Rex. This work was supported by the Swiss National Center of Competence in Research (NCCR) Robotics project and the China Scholarship Council under Contract 201406020025. This paper only reflects the authors' view and funding agencies are not liable for any use that may be made of the information contained herein.

## REFERENCES

- [1] R. Bogue, “Robotic exoskeletons: a review of recent progress,” *Industrial Robot: An International Journal*, volume 42, number 1, 2015.
- [2] I. Iturrate, R. Chavarriaga, L. Montesano, J. Mínguez, and J. d. R. Millán, “Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control,” *Scientific reports*, volume 5, 2015.
- [3] J. d. R. Millán, R. Rupp, G. R. Müller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kübler, R. Leeb, *et al.*, “Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges,” *Frontiers in Neuroscience*, volume 4, 2010.
- [4] S. Perdikis, R. Leeb, J. Williamson, A. Ramsay, M. Tavella, L. Desideri, E.-J. Hoogerwerf, A. Al-Khodairy, R. Murray-Smith, and J. d. R. Millán, “Clinical evaluation of BrainTree, a motor imagery hybrid BCI speller,” *Journal of Neural Engineering*, volume 11, number 3, page 036003, 2014.
- [5] T. Carlson and J. d. R. Millán, “Brain-controlled wheelchairs: a robotic architecture,” *IEEE Robotics and Automation Magazine*, volume 20, pages 65–73, 2013.
- [6] M. Tavella, R. Leeb, R. Rupp, and J. d. R. Millán, “Towards natural non-invasive hand neuroprostheses for daily living,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2010, pages 126–129.
- [7] G. R. Müller-Putz, R. Scherer, G. Pfurtscheller, and R. Rupp, “EEG-based neuroprosthesis control: a step towards clinical practice,” *Neuroscience Letters*, volume 382, number 1, pages 169–174, 2005.
- [8] J. L. Contreras-Vidal and R. G. Grossman, “NeuroRex: A clinical neural interface roadmap for EEG-based brain machine interfaces to a lower body robotic exoskeleton,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2013, pages 1579–1582.
- [9] G. Onose, C. Grozea, A. Anghelescu, *et al.*, “On the feasibility of using motor imagery EEG-based brain–computer interface in chronic tetraplegics for assistive robotic arm control: a clinical test and long-term post-trial follow-up,” *Spinal Cord*, volume 50, number 8, pages 599–608, 2012.
- [10] J. R. Wolpaw, D. J. McFarland, and T. M. Vaughan, “Brain-computer interface research at the Wadsworth Center,” *Rehabilitation Engineering, IEEE Transactions on*, volume 8, number 2, pages 222–226, 2000.
- [11] L. F. Nicolas-Alonso and J. Gomez-Gil, “Brain computer interfaces, a review,” *Sensors*, volume 12, number 2, pages 1211–1279, 2012.
- [12] A. Ramos-Murguialday, M. Schürholz, V. Caggiano, M. Wildgruber, A. Caria, E. M. Hammer, S. Halder, and N. Birbaumer, “Proprioceptive feedback and brain computer interface (BCI) based neuroprostheses,” *PLoS One*, volume 7, number 10, e47048, 2012.
- [13] J. Polich, “Updating P300: an integrative theory of P3a and P3b,” *Clinical Neurophysiology*, volume 118, number 10, pages 2128–2148, 2007.
- [14] D. Regan, “Human brain electrophysiology: evoked potentials and evoked magnetic fields in science and medicine,” *Elsevier*, 1989.

- [15] J. d. R. Millán, P. W. Ferrez, F. Galán, E. Lew, and R. Chavarriaga, “Non-invasive brain-machine interaction,” *International Journal of Pattern Recognition and Artificial Intelligence*, volume 22, number 05, pages 959–972, 2008.
- [16] T. Elbert, B. Rockstroh, W. Lutzenberger, and N. Birbaumer, “Biofeedback of slow cortical potentials. I,” *Electroencephalography and Clinical Neurophysiology*, volume 48, number 3, pages 293–301, 1980.
- [17] G. Garipelli, R. Chavarriaga, and J. del R Millán, “Single trial analysis of slow cortical potentials: a study on anticipation related potentials,” *Journal of Neural Engineering*, volume 10, number 3, page 036014, 2013.
- [18] G. Pfurtscheller and C. Neuper, “Motor imagery and direct brain-computer communication,” *Proceedings of the IEEE*, volume 89, number 7, pages 1123–1134, 2001.
- [19] L. Tonin, T. Carlson, R. Leeb, and J. d. R. Millán, “Brain-controlled telepresence robot by motor-disabled people,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2011, pages 4227–4230.
- [20] J. d. R. Millán, F. Renkens, J. Mouriño, and W. Gerstner, “Noninvasive brain-actuated control of a mobile robot by human EEG,” *IEEE Transactions on Biomedical Engineering*, volume 51, number 6, pages 1026–1033, 2004.
- [21] F. Galán, M. Nuttin, E. Lew, P. W. Ferrez, G. Vanacker, J. Philips, and J. d. R. Millán, “A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots,” *Clinical Neurophysiology*, volume 119, number 9, pages 2159–2169, 2008.
- [22] G. Pfurtscheller and F. L. Da Silva, “Event-related EEG/MEG synchronization and desynchronization: basic principles,” *Clinical Neurophysiology*, volume 110, number 11, pages 1842–1857, 1999.
- [23] A. Kilicarslan, S. Prasad, R. G. Grossman, and J. L. Contreras-Vidal, “High accuracy decoding of user intentions using EEG to control a lower-body exoskeleton,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2013, pages 5606–5609.
- [24] <http://www.rexbionics.com>.
- [25] <http://www.argomedtec.com>.
- [26] E. Strickland, “Good-bye, wheelchair,” *Spectrum, IEEE*, volume 49, number 1, pages 30–32, 2012.
- [27] H. Kawamoto and Y. Sankai, “Power assist method based on phase sequence and muscle force condition for HAL,” *Advanced Robotics*, volume 19, number 7, pages 717–734, 2005.
- [28] S. K. Banala, S. H. Kim, S. K. Agrawal, and J. P. Scholz, “Robot assisted gait training with active leg exoskeleton (ALEX),” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, volume 17, number 1, pages 2–8, 2009.
- [29] R. Rea, C. Beck, R. Rovekamp, M. Diftler, and P. Neuhaus, “X1: A robotic exoskeleton for in-space countermeasures and dynamometry,” in *AIAA SPACE Conference and Exposition*, 2013.
- [30] H. A. Quintero, R. J. Farris, and M. Goldfarb, “Control and implementation of a powered lower limb orthosis to aid walking in paraplegic individuals,” in *IEEE International Conference on Rehabilitation Robotics (ICORR)*, IEEE, 2011, pages 1–6.



- [31] G. R. Müller-Putz, C. Breitwieser, F. Cincotti, R. Leeb, M. Schreuder, F. Leotta, M. Tavella, L. Bianchi, A. Kreiling, A. Ramsay, *et al.*, “Tools for brain-computer interaction: a general concept for a hybrid BCI,” *Frontiers in Neuroinformatics*, volume 5, 2011.
- [32] A. J. Casson, D. Yates, S. Smith, J. S. Duncan, and E. Rodriguez-Villegas, “Wearable electroencephalography,” *Engineering in Medicine and Biology Magazine, IEEE*, volume 29, number 3, pages 44–56, 2010.
- [33] R. Leeb, L. Tonin, M. Rohm, L. Desideri, T. Carlson, and J. d. R. Millán, “Towards independence: a BCI telepresence robot for people with severe motor disabilities,” *Proceedings of the IEEE*, volume 103, number 6, pages 969–982, 2015.
- [34] L. Breiman, “Random forests,” *Machine Learning*, volume 45, number 1, pages 5–32, 2001.
- [35] D. Steyrl, R. Scherer, J. Faller, and G. R. Müller-Putz, “Random forests in non-invasive sensorimotor rhythm brain-computer interfaces: a practical and convenient non-linear classifier,” *Biomedical Engineering*, 2015.
- [36] F. Akram, S. M. Han, and T.-S. Kim, “An efficient word typing P300-BCI system using a modified T9 interface and random forest classifier,” *Computers in Biology and Medicine*, volume 56, pages 30–36, 2015.
- [37] M. Bentlemsan, E.-T. Zemouri, D. Bouchaffra, B. Yahya-Zoubir, and K. Ferroudji, “Random Forest and Filter Bank Common Spatial Patterns for EEG-Based Motor Imagery Classification,” in *5th International Conference on Intelligent System Modeling and Simulation ISMS'14*, 2014.
- [38] D. Steyrl, R. Scherer, and G. R. Müller-Putz, “Random forests for feature selection in non-invasive brain-computer interfacing,” in *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, Springer, 2013, pages 207–216.
- [39] A. Verikas, A. Gelzinis, and M. Bacauskiene, “Mining data with random forests: A survey and results of new tests,” *Pattern Recognition*, volume 44, number 2, pages 330–349, 2011.
- [40] R. Díaz-Uriarte and S. A. De Andres, “Gene selection and classification of microarray data using random forest,” *BMC Bioinformatics*, volume 7, number 1, page 3, 2006.
- [41] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, “Real-time human pose recognition in parts from single depth images,” *Communications of the ACM*, volume 56, number 1, pages 116–124, 2013.
- [42] A. Doumanoglou, A. Kargakos, T.-K. Kim, and S. Malassiotis, “Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pages 987–993.
- [43] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, “Latent regression forest: Structured estimation of 3d articulated hand posture,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2014, pages 3786–3793.
- [44] H. H. Jasper, “The ten twenty electrode system of the international federation,” *Electroencephalography and Clinical Neurophysiology*, volume 10, pages 371–375, 1958.
- [45] <http://www.arduino.cc>.

- [46] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller, “Optimal spatial filtering of single trial EEG during imagined hand movement,” *IEEE Transactions on Rehabilitation Engineering*, volume 8, number 4, pages 441–446, 2000.
- [47] O. Bertrand, F. Perrin, and J. Pernier, “A theoretical justification of the average reference in topographic evoked potential studies,” *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, volume 62, number 6, pages 462–464, 1985.
- [48] D. B. Percival, *Spectral analysis for physical applications*. Cambridge University Press, 1993.
- [49] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, *et al.*, “MEG and EEG data analysis with MNE-Python,” *Frontiers in Neuroscience*, volume 7, 2013.
- [50] R. Leeb, S. Perdakis, L. Tonin, A. Biasiucci, M. Tavella, M. Creatura, A. Molina, A. Al-Khodairy, T. Carlson, and J. d. R. Millán, “Transferring brain–computer interfaces beyond the laboratory: successful application control for motor-disabled users,” *Artificial intelligence in medicine*, volume 59, number 2, pages 121–132, 2013.
- [51] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 22, number 11, pages 1330–1334, 2000.
- [52] G. Bradski, “The OpenCV Library,” *Doctor Dobbs Journal*, volume 25, number 11, pages 120–126, 2000.
- [53] M. Y. Yang and W. Förstner, “Plane detection in point cloud data,” in *Proceedings of the 2nd int conf on machine control guidance, Bonn*, volume 1, 2010, pages 95–104.
- [54] K. Pulli and M. Pietikäinen, “Range image segmentation based on decomposition of surface normals,” in *Proceedings of the Scandinavian conference on image analysis*, PROCEEDINGS PUBLISHED BY VARIOUS PUBLISHERS, volume 2, 1993, pages 893–893.
- [55] R. Chavarriaga and J. d. R. Millán, “Learning from EEG error-related potentials in noninvasive brain-computer interfaces,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, volume 18, number 4, pages 381–388, 2010.
- [56] R. Chavarriaga, A. Sobolewski, and J. d. R. Millán, “Errare machinale est: the use of error-related potentials in brain-machine interfaces,” *Frontiers in Neuroscience*, volume 8, 2014.