ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Project In Computational Neurosciences
CNBI

EEG-Based Control of a 7-DOF Upper-Limb Prosthesis

Noémie Jaquier

Professor José del R. Millán
Supervisors Iñaki Iturrate, Ricardo Chavarriaga

January 7, 2016

Project Summary

The goal of this project is to integrate a shared-control strategy for the BMI-based control of a 7-DOF robot arm and to test it with several subjects.

The protocol consists of a 2D reaching task. The end-effector of the robot moves along a vertical 2D plane within a $5x5$ grid toward a target location as shown by Figure I. One of the following actions is performed at each time step; going up, down, left or right and goal-reached. The goal-reached action is signalled by a sound. Each action is observed by the user and assessed as correct or erroneous. After a training phase, a classifier is trained to recognise right and wrong actions using EEG Error-Related Potentials(ErrPs) generated by the subject when he/she observes an incorrect action to reach the goal. During online runs, decoding of ErrPs is used to infer the goal location by updating the probabilities of each location to be the target.

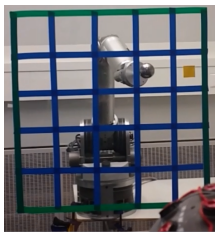


Figure I: Setup of the two-dimensional reaching task.

To control the robot and to reach the current desired position of the end-effector, the following steps are performed at each time step:

- The inverse dynamic model is updated;
- The desired position of the end-effector is set;
- Joint angles are get from the robot;
- The kinematic chain is updated;
- The jacobian is computed;
- The joints velocity limit is set;
- The inverse kinematic model is solved;
- The next movement is obtained from the BCI protocol;
- Necessary joint torques are computed by the PID.

Four subjects participated in the experiment where subjects were asked to reach two fixed and two self-chosen targets. Three subjects were able to reach successfully all targets. The remaining subject reached three of them. Based in previous studies, EEG signals were mainly measured in fronto-central channels. Figure II shows the error, correct and difference grand-averaged potentials in channel FCz averaged across all subjects.

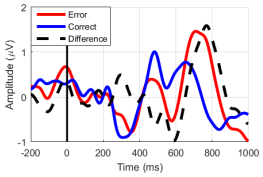


Figure II: Error, correct and difference grand averaged potentials in channel FCz.

The mean offline ten-fold accuracies across subjects were 73.1 ± 2.0 and 61.8 ± 3.9 for correct and erroneous actions, respectively. The mean online accuracies were 76.9 ± 10.6 and 56.5 ± 12.1 for correct and erroneous actions, respectively. The mean number of actions required to reach the goal was 40 ± 11. Remarkably, the device did not need to explore all possible locations to determine the target location as shown by the heat map of Figure III.

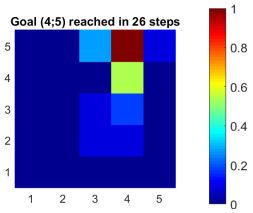


Figure III: Heat map of visited states (normalised) of subject 3 for target $(4;5)$ with start location $(4;2)$.

The experiment with the robot seems to be more demanding for the users than the one with a virtual cursor, for which offline accuracies were 74.3 ± 10.3 and 74.5 ± 7.5 for correct and erroneous actions and number of actions needed to find the target was 25 ± 13.

However, it was shown that this shared-control strategy also works in more demanding tasks and allows to control the robot toward multiple locations.

The protocol could be improved by signalling the goal-reached action with a visual stimulus. A robotic hand could be used to simulate a grasping task.

Contents

1 Introduction **1**

2 State of the Art **1**

2.1 EEG Error-Related Potentials 1

2.2 Shared-Control BMI for a Reaching Task 1

3 Control of the WAM **3**

3.1 Description of the Robot 3

3.2 Global Description of Executed Tasks 3

3.3 Detailed Implementation of the Controller 4

3.4 Velocity Profile Implementation 6

3.5 Communication Between the BMI Protocol and the Robot 6

4 Experimental Setup **7**

5 Analysis of the Results **7**

6 Conclusion and Future Work **9**

References **10**

1 Introduction

Over the last years, significant progress has been made in the field of Brain-Machine Interfaces (BMI) using both invasive and non-invasive approaches to control neuroprosthetic devices. In the non-invasive field, electroencephalogram (EEG) allows to record the electrical activity of the brain. Control of virtual cursors[1], mobile robots[2] or robotic wheelchairs[3] have been demonstrated using EEG-based BMIs. However, EEG-based systems are affected by a low information rate, so that a large amount of time is often necessary to solve a task. Moreover, the subjects need to modulate their brain activity to convey all necessary information to control the device. Thus, it requires a high concentration and the period of training is usually long.

Recently, those limitations have been reduced with the exploration of shared-control strategies, where the device is also involved in the task execution. Among the existent shared-control schemes, ErrPs-based BMIs have proved to be a feasible and efficient approach. Error-related potentials (ErrPs) are signals generated by the brain in case of an unexpected outcome, when someone makes or observes a mistake for example. In this type of schemes, they are used as feedback to supervise the execution of a task. In the example of a reaching task, the user has only to assess if the device is going or not towards the target location. Based on ErrP and on optimal motion policies, as inferred by reinforcement learning methods, the device could estimate where the target is situated.

The goal of this project is to integrate this shared-control paradigm for the BMI-based control of a 7-DOF (Degrees Of Freedom) upper limb prosthesis, here emulated by the WAM. The feasibility of the approach is also tested by an online experiment with healthy control subjects.

Section 2 of this report presents the previous work related to shared-control BMI for a two dimensional reaching task using error-related potentials. The design and implementation used to control the robotic arm is described in Section 3. Section 4 shows the experimental setup used for testing the approach. The results and their analysis are presented in Section 5. Section 6 finally concludes the report and makes suggestions about future work that could be done.

2 State of the Art

2.1 EEG Error-Related Potentials

Error-related potentials (ErrP) are event-related potentials (ERPs) evoked by actions considered wrong by the subject. They occur when a human makes a mistake or when a decision of a system, that he/she does not control, is perceived as erroneous to achieve his/her goal. Error-related potentials seem to be generated in the anterior cingulate cortex (ACC) and can be measured by scalp EEG over fronto central areas. The difference between error-related potentials for a wrong and a correct actions is characterised by a waveform with positive and negative peaks generated between approximately 250 and 500 ms after the stimuli[4],[5]. Moreover, error-related potentials have a stable waveform over long period of time, but are variable along the subjects[6].

Error-related potentials can thus be used to monitor the behaviour of a neuroprosthesis by being employed as rewards for reinforcement learning algorithms[4].

2.2 Shared-Control BMI for a Reaching Task

The shared-control strategy used in this project is the one described in[4] and[7]. The main advantages of such a method are that the user only supervises the device operations and that it is natural for him/her, as a human is used to assess right and wrong actions.

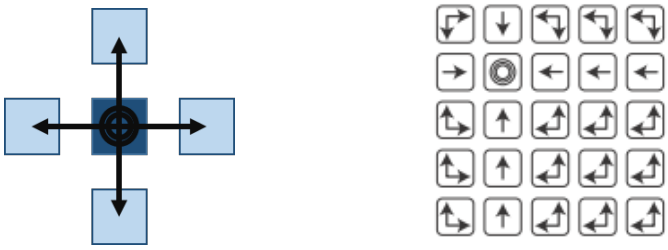
The protocol consists of a $5x5$ grid with a virtual cursor and a target location. The cursor can perform five different actions; going up, down, left or right and goal-reached. Following the reinforcement larning approach,

the correct actions that could be performed on each location to reach a specific target constitute the optimal policy for this target. Figure 1 shows the possible actions of the cursor and an example of optimal policy.

An action is considered as correct, if it makes the cursor going toward the goal, and as erroneous otherwise. Classification of EEG error-related potentials is used to label the actions as correct or erroneous. The classifier is trained during the training phase, where the cursor performs random actions with a determined probability of performing a wrong action given the target location. Sufficient trials should be measured with the EEG to have enough samples of evaluation of actions by the user. Features are extracted from the fronto-central channels. They are normalised and decorrelated using Principal Component Analysis (PCA). The classifier is then trained to find parameters w' and b. The probability that a trial x is classified as a correct action is given by Equation 1, where $y(x) = w'x + b$ is the output of the classifier with $y \geq 0$ for correct actions and $y < 0$ for an erroneous ones.

$$p(c = 1|x) = \frac{1}{1 + e^{-y(x)}} \tag{1}$$

Q-values $Q_i^*(s, a)$ are associated to each optimal policy $\pi_i^*(s)$ to reach to goal i from state s. The Q-value for a state-action pair represents the sum of the reward received for this pair and of future rewards resulting from new states and further actions following a particular policy leading to the goal. Q-values are computed using Q-learning reinforcement learning algorithm and are then converted into probabilities given Equation 2.



(a) Possible actions (up, down, left, right, goal reached) that can be performed by the cursor.

(b) Optimal policy for the target at location (2,2)[4].

Figure 1: Actions and example of optimal policy of the protocol.

Optimal policies for each potential target are computed offline with the calculation of Q-values and resulting probabilities. This constitute the first phase of the shared-control strategy. The second phase is the online control.

Initially, the probability of being the goal is equal for all locations. At each time step, the cursor has a state s and performs an action a evaluated by the user. The probability of Equation 1 is used to update the probability $p(\pi_i^*)$ of each optimal policy for the goal location i to be the right one. This update is given by Equation 3. The inferred target location is determined when a probability $p(\pi_i^*)$ is higher than the convergence criterion p_c.

Figure 2 shows an example of an online control run.

$$\hat{Q}_i^*(s, a) = \frac{e^{Q_i^*(s,a)/\tau}}{\sum_b e^{Q_i^*(s,b)/\tau}} \tag{2}$$

$$p(\pi_i^*|(a, s, x)_{1..t}) \propto p(a_t|\pi_i^*, (s, x)_t) \cdot p(\pi_i^*|(a, s, x)_{1..t-1}) \tag{3}$$

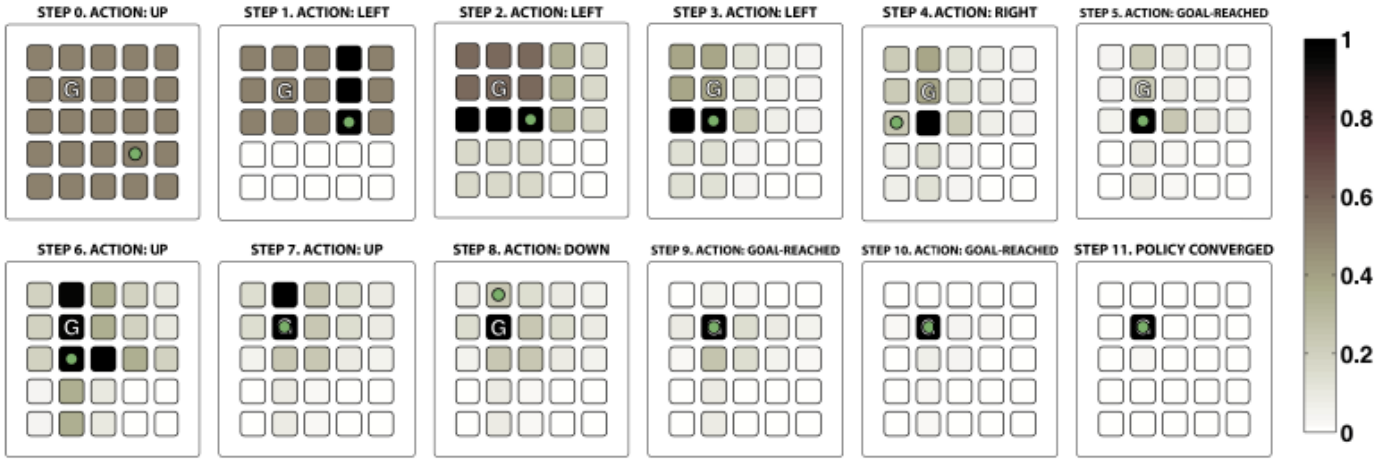


Figure 2: Online control example for goal G in position (2,2). The darker is a location, the higher is its probability to be the goal. The policy convergence criterion is reached after eleven steps[4].

3 Control of the WAM

3.1 Description of the Robot

The WAM is a 7-DOF arm robot. The WAM is backdrivable, meaning that forces and motions applied at the output end of mechanical transitions are accurately reproduced at the input end. Its mechanical transmissions are based on cable drives. Motors of the robot are Neodymium Iron Boron, brushless, DC servo-motors. Specially designed motor encoders are mounted directly on the motors. They convert a digital torque command in continuous torque for the motor and serve as an encoder for the angle.

Figures 3 shows the WAM structure in zero position. As shown by Figure 4, a gimbal is fixed to the end-effector and allows to point precisely the grid location corresponding to the position of the virtual cursor in the simulated reaching task. Joints positive and negative limits determining the workspace of the robot are given by Table 1.

Joint	Positive joint limit (°)	Negative joint limit (°)
1	150	-150
2	113	-113
3	157	-157
4	180	-180
5	71	-71
6	90	-90
7	172	-172

Table 1: Joints limits defining the range of motion of the WAM[8].

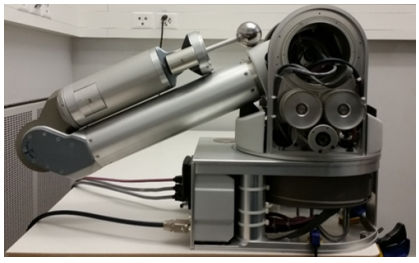


Figure 4: Picture of the WAM.

3.2 Global Description of Executed Tasks

The robot operates in a two dimensional vertical space. A 5x5 grid is drawn on a plexiglas board. Each location can be pointed by the gimbal, as shown by Figure 5.

The behaviours of the robot during offline and online runs are respectively given by Algorithms 1 and 2. The initial joint position of the robot is chosen so that the gimbal points the central location and all others locations can be pointed without mechanical problems. During an offline run, the protocol selects new start and target locations each time the goal is reached until the required number of actions has been performed. In this case, several targets can be reached. On the contrary, only one start-target pair is defined for the online

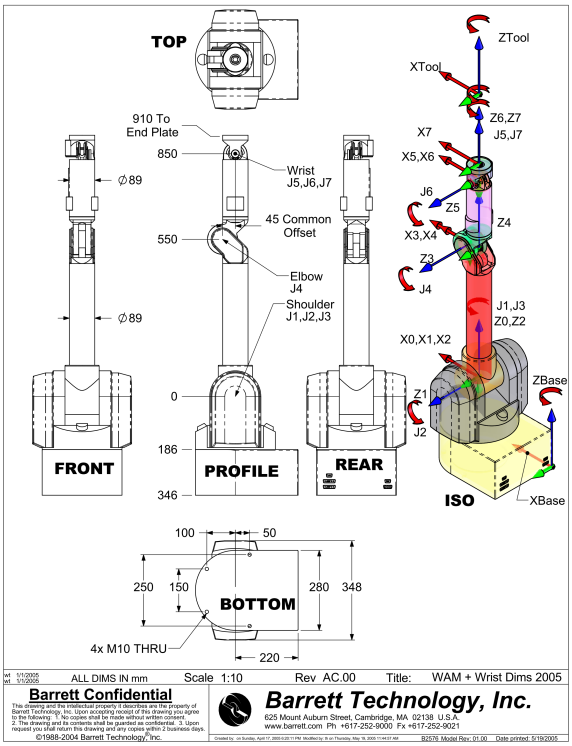


Figure 3: 7-DOF WAM system in zero position[8].

Figure 5: Setup of the two-dimensional reaching task with the real robot.

part. The trial is finished when the policy convergence criterion is reached. To show that it converged, the robot moves so that the gimbal goes back from the board and return pointing at the supposed goal. To signal a goal-reached action, a ”beep” is emitted, while the robot does not move.

3.3 Detailed Implementation of the Controller

When the protocol is started, the following elements related to the robot are initialised:

- Sensors and actuators lists;
- Kinematic chain;
- Inverse kinematic model;
- Inverse dynamic model;
- PID controller.

Sensors and actuators are determined by the robot type. Their measurements and values are transmitted through the communication between the WAM and the computer.

Algorithm 1: Offline run

Initialisation at joint position $[0, 0.69, 0, 1.57, 0, -0.69, 0]$;
while *Resting time* **do**
    Wait;
end
Go to Start location;
while *Required number of actions not reached* **do**
    Execute action commanded by the protocol;
    if *action == goal-reached* **then**
        Sound emission;
    end
    if *target location reached* **then**
        Go to Start location;
        Wait during resting time;
    end
end

Algorithm 2: Online run

Initialisation at joint position $[0, 0.69, 0, 1.57, 0, -0.69, 0]$;
while *Resting time* **do**
    Wait;
end
Go to Start location;
repeat
    Execute action commanded by the protocol;
    if *action == goal-reached* **then**
        Sound emission;
    end
until *Policy has converged*;
Go back and forth;

The kinematic chain is a mathematical model in which rigid bodies, or links, are connected by joints. Kinematic chains of serial robots, like the WAM, are formed by links connected in series. The joint variables $(q_1, q_2, ..., q_n)$ of the robot are the linear or angular inputs sent to the actuators. The operational variables define the position (x, y, z) and the orientation (ϕ, ψ, θ) of the end-effector. The forward kinematics express the operational variables $(x, y, z, \phi, \psi, \theta)$ in function of the joint variables. Conversely, the inverse kinematics express the joint variables in function of the operational variables. For serial robots, the inverse kinematics have more than one solution. There are thus more than one configuration for some position of the end-effector and inverse kinematics are complex to solve.

The robot Jacobian relates joint velocities $(\dot{q})$ to the linear and angular velocities (v, w) of the end-effector. Singular configurations of the robot correspond to those where $det(J) = 0$. The Jacobian can also be used to solve the inverse kinematics problem.

The inverse dynamic model computes the torques that must be delivered by the motors in function of joint positions, velocities and accelerations $(q, \dot{q}, \ddot{q})$. The inverse dynamics depend also from the inertial properties of the system.

A PID controller is used to compute the torques of the robot's motors necessary to move from the actual joint positions to the desired ones.

The following steps are thus performed at each time step to reach the desired position of the end-effector:

- The inverse dynamic model is updated in order to compensate the gravity;
- The desired position of the end-effector is set;
- Joint angles are get from the WAM;
- The kinematic chain is updated;
- The jacobian is computed;
- The joints velocity limit is set;
- The inverse kinematic model is solved;
- The amplitude of the next movement is calculated in function of the maximal velocity and of Δt;
- Necessary joint torques are computed by the PID controller.

This closed-loop approach allows to adapt the command in function of the current state of the robot at each time step. Singular configurations of the robot are also avoided.

In this case, the possible desired positions (x, y, z) of the end-effector are defined in the centre of each location of the grid. The positions used to signal that the policy converged are the same, except that they are further from the plexiglas board and closer to the robot base.

3.4 Velocity Profile Implementation

To have smoother movements of the robot, a parabolic velocity profile was implemented. This type of profile is adapted to follow trajectories. In contrast to a trapezoidal velocity profile, it has no jump of acceleration, which reduce solicitations on the mechanic parts of the system. An example of parabolic velocity profile is shown by Figure 6.

3.5 Communication Between the BMI Protocol and the Robot

Figure 7 summarises the system and the communication process between the different elements. The robot is connected to its controller through CAN communication. This controller receives instructions (i.e., next positions of the end-effector) from the BMI protocol via sockets. The EEG acquisition system transmits EEG measurements to the protocol, which uses them to classify the movements of the robot as correct or erroneous and to consequently send the orders to the robot controller.

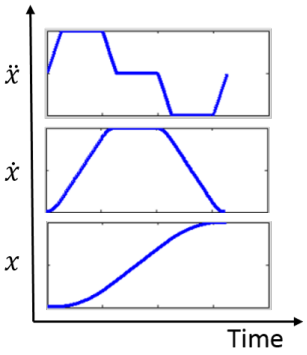


Figure 6: Position, velocity and acceleration in function of time for a parabolic velocity profile.

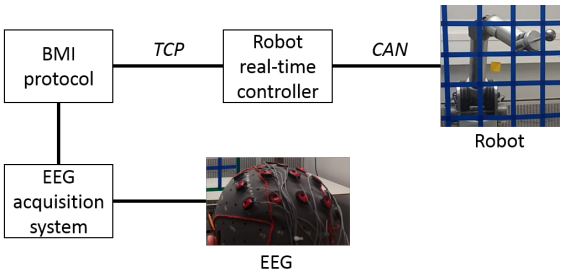


Figure 7: Schema of the communication systems between different componants of the protocol.

4 Experimental Setup

Electroencephalographic (EEG) activities were recorded using a gTec USBamp system with 16 channels. Four subjects participated in the experiments. They were seated one meter away from the plexiglas panel situated in front of the robot with their eyes approximately at the height of the grid central location. The time between two actions of the robot was 1.5s. The role of subjects was to assess the actions of the robot as correct or erroneous. They should limit artifacts, as blinking or moving the eyes, only to the resting periods.

The first part of the experiment consisted of four offline runs used to train a classifier. Each offline run was composed of 100 actions. An online run was then executed and, if the target was not reached, the classifier was retrained with the five first runs. During the control phase, two fixed and two self-chosen locations were tested. The fixed targets were the same for all subjects.

EEG signals were sampled at 512 Hz and bandpass filtered at $[1, 10]$ Hz. A Common-Averaged-Reference (CAR) filter was also applied on EEG signals.

5 Analysis of the Results

Figure 8 shows the error, correct and difference grand averaged potentials for channel Fz averaged across all subjects. The vertical line is the time 0 ms indicating when the action is sent to the robot controller by the protocol. Signals showed larger modulations in those two fronto-channels. The grand averaged difference is characterised by a positive peak around 350 ms, a negative peak around 600 ms and a positive peak around 750 ms.

Table 2 shows the offline ten-fold accuracy for correct and erroneous actions, the number of fixed and self-chosen reached targets, the number of actions necessary to reach the targets and the online accuracy for correct and erroneous movements for each subject. The device was able to reach all the target locations for three subjects out of four. The protocol was not able to find the target location for the last target of subject 4. As the experiment was demanding, the tiredness of the subject at the end can explain that the device was not able to reach the last location.

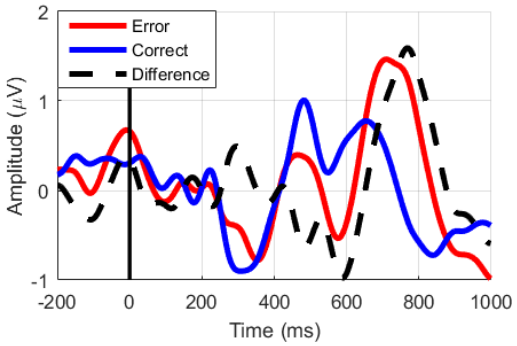


Figure 8: Error (red), correct (blue) and difference (black) grand averaged potentials in channel Fz averaged across all subjects.

The mean offline accuracies across subjects were 73.1 ± 2 and 61.8 ± 3.9 for correct and erroneous actions. Similarly, the mean online accuracies across subjects were 76.9 ± 10.6 and 56.2 ± 12.1. The mean number of actions of the robot necessary to reach the target was 40 ± 11.

	s1	**s2**	**s3**	**s4**	**mean±std**
Offline ten-fold accuracy correct/error (%)	75.8/67.1	72/61.2	71.2/57.6	73.3/61.4	$73.1 \pm 2.0/61.8 \pm 3.9$
Fixed/self-chosen reached targets	2/2	2/2	2/2	1/2	$2 \pm 0/2 \pm 1$
Number of actions per target	47 ± 6	47 ± 14	31 ± 6	33 ± 6	40 ± 11
Online accuracy correct/error (%)	79.3/68.8	69.2/57.1	68.2/60.0	90.9/40.0	$76.9 \pm 10.6/56.5 \pm 12.1$

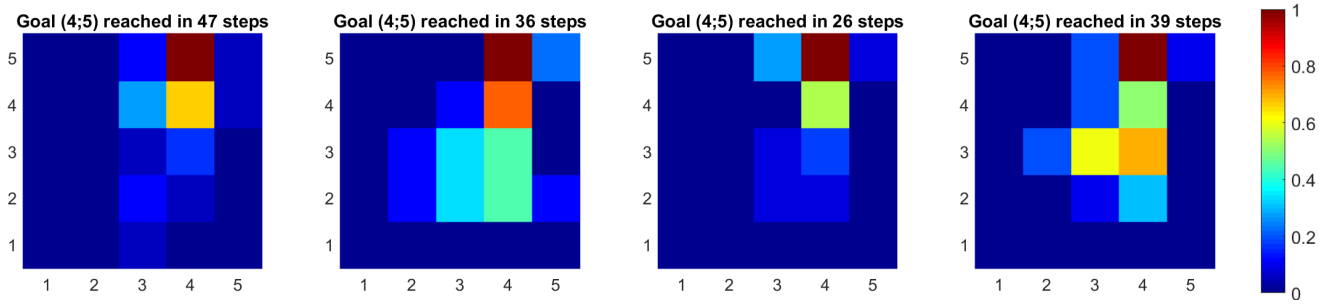
Table 2: Results of the reaching task.

Figure 9 shows the heat maps for all the subjects for the fixed target location $(4, 5)$. The range is normalised with 1 corresponding to the most visited state of the run, indicated with dark red. Less visited locations are represented in blue. The start location was $(4, 2)$ for all subjects. Not all the states were visited to reach the goal. Moreover, locations situated around the target and the target itself are generally the most visited ones. This indicate that this control scheme is rapidly able to find the zone of the grid where the goal is located and needs then to visit neighbours locations to converge to the goal. As reported in [4], a random walk strategy selecting random actions at each step would require around 150 actions. The selected approach is thus much more efficient, as each step allows to update information about all possible goal locations.

The offline accuracies obtained with a 2D virtual cursor [4] were 74.27 ± 10.32 and 74.5 ± 7.45 for respectively correct and erroneous actions. Moreover, the mean number of actions needed to reach the target was 25 ± 13. The accuracies obtained with the end-effector of the robot as cursor are slightly lower and the number of actions needed is higher. A possible explanation is that assessing the actions of a real device is more demanding for subjects than assessing actions of a virtual cursor.

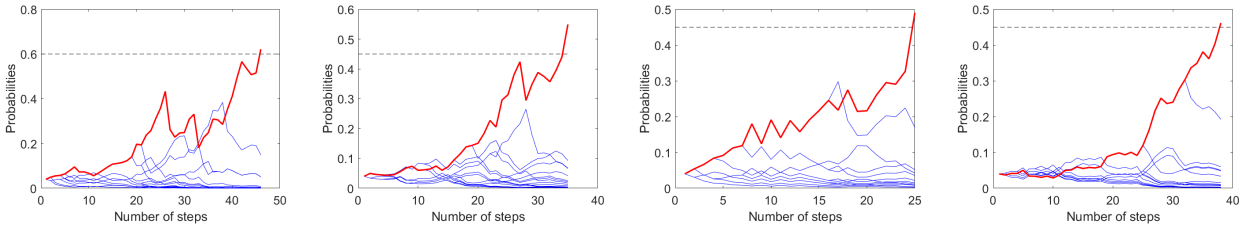
Figure 10 shows the evolution of the probability of policies to be the correct one through one run for each subject. Each policy corresponds to a given location being the target. In this case, the target was at location $(4, 5)$ and the start location was $(4, 2)$. For all subjects, most of the policies have a low probability from the first steps. Only a few ones have relatively high probabilities. At the end of the runs, two policies remain probable and the right one, corresponding to the target, is finally chosen.

A sudden decrease in the probability of the correct policy (marked in red) is observed between steps 27 and 28 for subject 2. This is due to the sound emission for the goal-reached action that surprised the subject. This sound was also surprising for some of others subjects due to the difference of stimuli between the goal-reached actions (auditory) and the actions of displacement (visual). Future protocols may use a different stimulus to denote the goal-reached action.



(a) Subject 1 (b) Subject 2 (c) Subject 3 (d) Subject 4

Figure 9: Heat maps for target $(4;5)$ with start location $(4;2)$. The range is normalised with 1 corresponding to the most visited state of the run, indicated with dark red. Blue colors correspond to less visited locations.



(a) Subject 1 (b) Subject 2 (c) Subject 3 (d) Subject 4

Figure 10: Probabilities history for target $(4;5)$ with start location $(4;2)$. The optimal policy is the bold red line while the other ones are the blue lines. The horizontal black line is the value of the convergence criterion.

6 Conclusion and Future Work

A shared-control BMI based on Error-Related Potentials was implemented on a 7-DOF arm robot. The end-effector of the robot was used to point the current location on a $5x5$ grid. The controller of the robot used inverse kinematic and dynamic models, the jacobian and a PID controller to perform the necessary operations to reach the desired position of the end-effector. A parabolic velocity profile was also implemented to have smooth movements.

Four subjects participated in the experiment. The device was able to reach two self-chosen and two fixed targets for three subjects and two self-chosen and one fixed target for the fourth one. The mean number of actions necessary to find the target was 40. Moreover, the device did not need to explore all possible locations to determine the goal.

The experiment with the robot seems to be more demanding for subjects compared to previous work where the cursor was virtual in two dimensions. Indeed, the number of actions needed to reach the goal was slightly higher and the offline accuracies of the classifier were also slightly lower using the end-effector of the robot as cursor.

To improve the protocol, the goal-reached action should be indicated with a visual stimulus. Moreover, a robotic hand could be used to simulate a reaching and grasping task. The movements of the robot could also become continuous to simulate a real reaching task similar to those executed by humans. Finally, the approach could be extended to 3D movements.

The control of the robot and the experimental part used to test the approach with the robot were done for this project. The BCI protocol was already available and taken from previous work[4], [5] and[6].

References

[1] N. Birbaumer, A. Kubler, N. Ghanayim, T. Hinterberger, J. Perelmouter, J. Kaiser, I. Iversen, B. Kotchoubey, N. Neumann, and H. Flor. The thought translation device (ttd) for completely paralyzed patients. *Rehabilitation Engineering, IEEE Transactions on*, 8(2):190–193, Jun 2000.

[2] C. Escolano, J.M. Antelis, and J. Minguez. A telepresence mobile robot controlled with a noninvasive brain computer interface. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(3):793–804, June 2012.

[3] I. Iturrate, J.M. Antelis, A. Kubler, and J. Minguez. A noninvasive brain-actuated wheelchair based on a p300 neurophysiological protocol and automated navigation. *Robotics, IEEE Transactions on*, 25(3):614–627, June 2009.

[4] I. Iturrate, L. Montesano, and J. Minguez. Shared-control brain-computer interface for a two dimensional reaching task using eeg error-related potentials. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 5258–5262, July 2013.

[5] I. Iturrate, R. Chavarriaga, L. Montesano, J. Minguez, and J. del R. Millán. Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control. *Scientific Reports*, 5(13893), 2015.

[6] R. Chavarriaga and J.R. del Millan. Learning from EEG error-related potentials in noninvasive brain-computer interfaces. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(4):381–388, Aug 2010.

[7] R. Chavarriaga, I. Iturrate, Q. Wannebroucq, and J. del R. Millán. Decoding fast-paced error-related potentials in monitoring protocols. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 1111–1114, Aug 2015.

[8] Barrett Technology Inc. WAM Arm, 2015-2016. User's Manual.

[9] P. Ferrez and J. del R. Millán. Error-related EEG potentials generated during simulated brain-computer interaction. *IEEE Trans. on Biomedical Engineering*, 55(3), 0 2008.