Hybrid Brain Computer Interface to Control Modular Robots

MORET Michael

25.05.2015

Biorobotics Laboratory
Professor: Auke J. Ijspeert
Supervisors: Stéphane Bonardi, Mehmet Mutlu, Simon Hauser

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

EPFL Biorobotics Laboratory

Contents

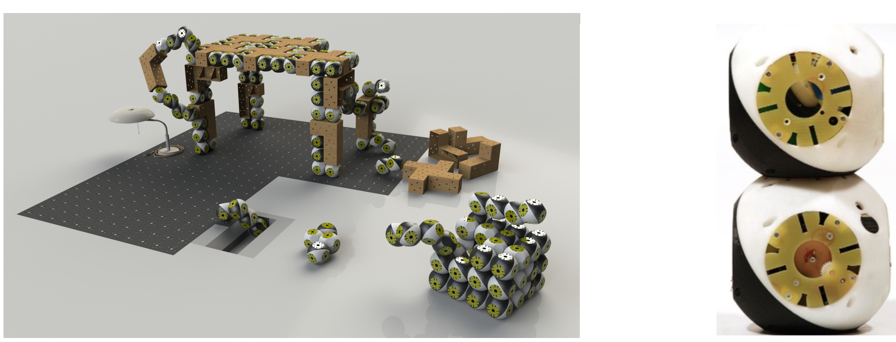
Abstract

The role of this project was to check whether an EEG based system–the Emotiv EPOC neuroheadset–would be a viable and intuitive user interface to control modular robots in the scope of the Roombots project. After a first set of tests, another modality was added to widen the range of possible actions, limited to only two with the EEG device only; speech recognition was chosen as the second modality. An evaluative user study was designed to test this system composed of two modalities to move a structure composed of passive elements and modular robots, but the EPOC did not prove to be robust enough toward this application. Finally, an internal user study with four members of the lab was conducted to test more deeply the device. The conclusion is that the EPOC device does not match the minimum requirements for robots control with its EEG signal.

1 Introduction

1.1 Goal of the project

The aim of this project is to assess if a commercially available neuroheadset device that relies on electroencephalography will be a robust and a natural user interface to control modular robots. To answer that question, we will use the EPOC neuroheadset from Emotiv systems [3] and the modular robots from the Roombots project.
This project aims to use modular robots, i.e. multiple simple robots used as building blocks than can attach and detach to each other, to form furniture that can self-reconfigure, self-assemble and move. The main application for such a technology would be to help elderly in their everyday life by, for example, providing help to recover from a fall or assist to manipulate objects.

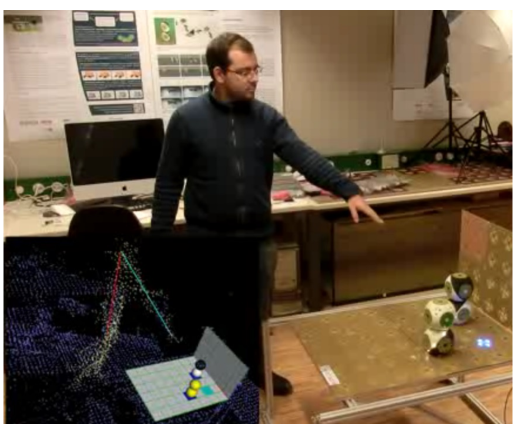


(a) Rendered picture of connected modules [20] (b) One module [20]

Figure 1: (a) Roombots modules. In the middle top, Roombots modules are connected together and with passive elements to form a table structure. In the middle, a module is getting out of the ground. On the right, a group of modules connected togheter. (b) One Roombots module.

1.2 User interface

The question of how to control such robots raises many problems that do not appear while manipulating more conventional robots such as a car or a humanoid; what would be an intuitive and effective way to control Roombots modules? There are already several projects that have covered this topic but at the moment, an ideal solution has not been found.
For example, one of them was developed to move individual Roombots by pinpointing them and the destination of arrival with body gesture [1] (figure 1a) . The user movement is detected with the help of dual depth sensor setup and a visual feedback on the location chosen as well as on the selected module is given to the user with a LED system.
A second example is a project where an ipad application was developed to help the user to interact in an intuitive manner with the Roombots [2]. It provides an augmented reality display where one can interact with virtual 3D structures made of Roombots (figure 1b). To test the system, a user study was conducted.
While both projects improved the control interface for the Roombots, there is still room for advancement. Indeed, in the second project mention, the user needs to have an ipad in his hands in order to control the Roombots while in the first example the interface is dependent of a visual recognition system to analyze the gestures. Moreover, people with physical disabilities might not be able to use those interfaces. It is where an EEG headset could have an edge by providing a new way to command the Roombots without the need of physicals inputs while also being more portable.



(a) Intelligent user interface for Roombots

(b) Mobile control interface for modular robots

Figure 2: (a) The user uses his hand to show where a selected module should move. (b) Ipad display that allow the user to see a structure within a real environnement.

2 Electroencephalography – EEG

2.1 Presentation

EEG is a way to get data from the brain by recording change of action potential that arises from large group of neurons firing in the brain, either invasively or non-invasively. For the first case, which is also called intracranial EEG (iEEG) or Electrocorticography (ECoG), the electrodes are directly placed on the surfaced of the brain. For the non-invasive case, they are placed on the scalp.
Compared to other neuroimaging technique, the non-invasive method has the advantage to have a high time resolution–about the [ms]–but have a poor spatial resolution[1]–in the order of [cm^3]. Moreover, it is also cheaper and rely on a technology that is not complex.

2.2 Electrodes position

2.2.1 Brain areas

The human central nervous system (CNS) is composed of 5 regions, one of which is called the cerebrum (sometimes also called telencephalon)[2].
The cerebrum is the part of our CNS commonly known as the brain, which is particularly developed in mammals, and especially in primates. Positioned just under the skull, it is the part from which EEG devices record most of their signals.
The cerebrum has been mapped, firstly, only in an anatomical point of view, but it has appeared that a given function in the brain resides often inside boundaries, which has allowed to make correlation between functions and brain areas. However, some functions such as decision making are not situated at one exact location, while others such as motor control are at a precise point. An example is given in figure 3.

[1]Source: Course «Neuroscience for Engineers», Bachelor cycle of Life Sciences, EPFL.
[2]Source: Course «Physiology by systems I», Bachelor cycle of Life Sciences, EPFL.

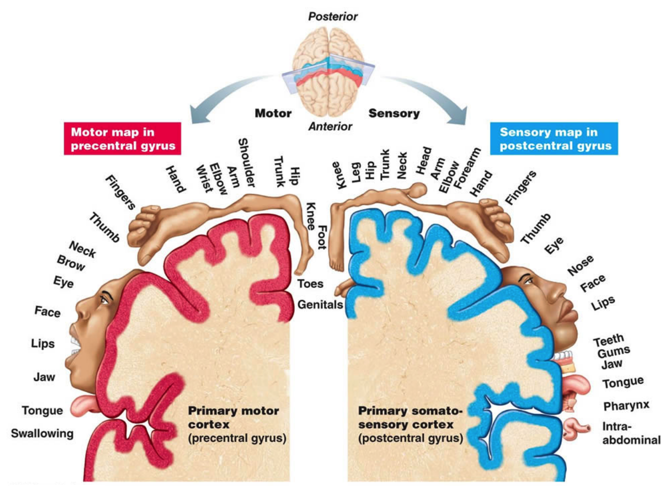


Figure 3: Visual representation of the motor and somatosensory coding on the brain surface. In red, the motor cortex–the part that control the body part voluntary actions–in blue, the sensory cortex–the part that receive inputs data [24].

2.2.2 Areas and potential detection

The position of the electrodes on the scalp as well as their number influence the signal that is recorded by an EEG system [5]. Moreover, from existing studies [4], we know that when one thinks of moving his hand, for example, the brain area linked to this task will be activated too. Therefore, the so-called brain motor imagery–where the user thinks of moving his limbs–has become one of the most efficient strategies to get voluntary action from EEG [4]. Indeed, by the bilateral symmetry of the brain, if someone thinks of moving his right hand, it will activate the related left part in the cerebrum, which can make a good basis to classify brain pattern for a system that will rely on spatial potential difference [17].

2.2.3 Nomenclature

As the position of electrodes on the scalp is important–for the reasons mentioned above as well as for the replicability of experiments–maps have been created to give researchers a common basis; the first of them is called the ”10-20 system” [6] while another one, which is in fact an extension of the first, is called the ”10-10 system” [7]. The numbers actually refer to relative distance in degree between electrodes. Typically, the 10-20 system will have 21 electrodes while the 10-10 will have 81 of them [8].

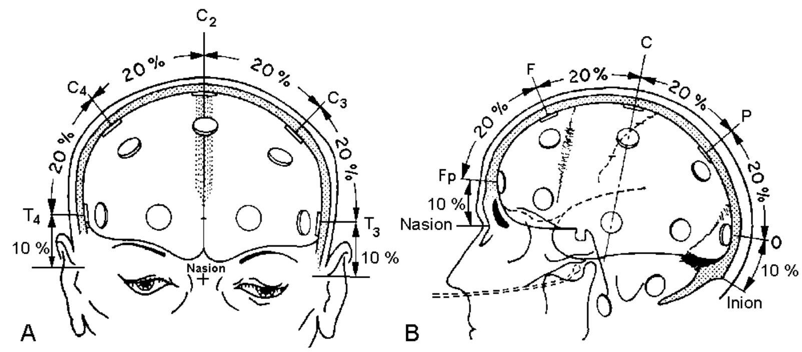


Figure 4: 10-20 system as seen frontally (A) and laterally (B) [9].

2.3 Brain computer interface – BCI

Brain computer interface is a field where data recorded from the brain are used as a modality to make an action. For example, we can cite the use of EEG signals to help someone with lock-in syndrome[3] to overcome his/her physical inability to carry any form of action and help his/her to write with a system allowing him to choose letters.

2.3.1 BCI training

The more common way to record a signal in this field is EEG [14]. To use a BCI based on EEG, two methods are mainly used. The first is to use an external stimuli to evoke the potential to the user. The second relies mainly on the user himself as he has to create by himself the difference of potential without any external stimuli [15]; int this case, the subject needs to train himself and the system that will recognize his thoughts. For example, one would use a motor imagery strategy and think of his right hand movement.
Of those two methods, the one relying on evoked potential is the more robust because it is less user specific and needs less training time. However, if one wants to train a system to control a robot, the need of an external stimuli would be inconvenient. This is the main reason why non evoked-potentials, even they are less reliable because the performance of distincts users may be highly variable and the training time is often very extensive, are more suitable for applications in which one has to give commands regarding a specific need at a given moment [15].
To illustrate this argumentation, we can compare the situation where one need to

[3]The locked-in syndrome (pseudocoma) describes patients who are awake and conscious but selectively deafferented, i.e., have no means of producing speech, limb or facial movements. [13]

communicate verbally through a BCI, which can be done with evoked-potential [28], with the one where someone has to drive a toy car where the will of the driver comes into play when choosing a direction.

2.3.2 Limitation of systems relying on EEG

Due to the physical technique that EEG relies on, there are barriers that limit the performance of systems relying on EEG. Indeed, as EEG is about measuring difference of potential on the scalp, there are physiological barriers that make it hard to get a signal; we can mention here the hairs, the skin and the skull. Moreover, differences of potential arising from facial muscles contraction–EMG [19]–and eyes movement–ocular artifact, EOG [18]–are also recorded, which distort often significantly the EEG signals. However, in this later case, signal processing can usually be applied to clear the wanted signals from those noises.

2.3.3 Hybrid BCI

To increase the efficiency of BCI systems, researchers in the field use the so-called hybrid BCI, which is a combination of at least one signal from the brain and one or more modalities that could work sequentially or simultaneously together[16]. The supplementary signals could either be a different modality or another brain signals.
For example, someone with a physical disability who cannot move his limbs may still have some activity in his muscles, which could be used as a modality by recording the residual electromyographic (EMG) [25].

2.4 Emotiv EPOC neuroheadset

The EPOC neuroheadset is a commercially available device from Emotiv Systems [3]. This company, as it describes itself, is «a bioinformatics company advancing understanding of the human brain using electroencephalography (EEG). Our mission is to empower individuals to understand their own brain and accelerate brain research globally.». According to them, their technology has many applications such as gaming, hands-free control system, art, transport safety, robotics and smart adaptive environnements [21], therefore their product should enter in the scope of our project.

Figure 5: Emotiv EPOC from Emotiv Systems. It possess 14 EEG channels. Its sensor technology relies on saline soaked felt pads that have to be prepared before each use. The sample rate is 2048 [Hz], filtered and downsampled to 128 [Hz]. It also possesses a 2-axis gyroscope [3].

2.4.1 Comparaison with a medical device

A typical medical EEG device has far more than 14 electrodes (figure 6) and uses gel based electrodes to increase conductivity, which constrained the user to wash his head after use. The EPOC needs far less time to be prepared as it has less electrodes, which only need to be wet with a saline solution. Thus, the user does not need to wash his head after using it. Its design as well as its wireless connection allow it to be worn out of the lab, which is usually not the case with medical devices. However, the quality of the signals recorded from the brain is lower and it does not filter out EMG and EOG noises.

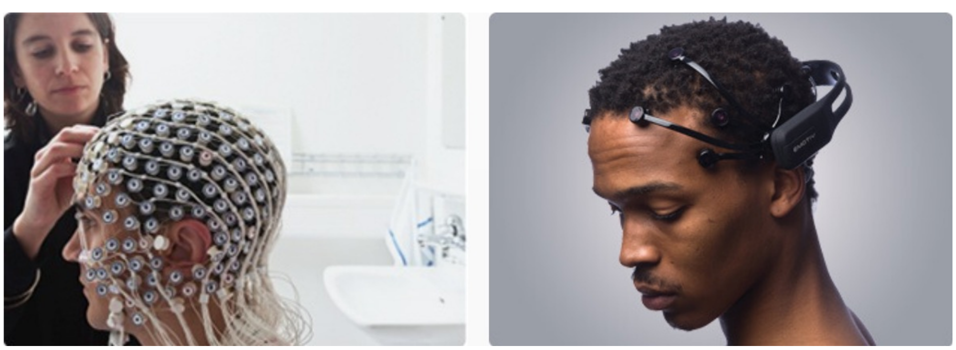


Figure 6: On the left, a conventional EEG system. On the right, the Emotiv EPOC [3].

2.4.2 EPOC and brain motor imagery

As mentioned earlier, brain motor imagery is the most commonly used technique to train a subject to use an EEG device. However, to use this training strategy, electrodes are needed above the areas that control and perceive the limbs. Unfortunately, the EPOC does not have electrodes on this location[4], which makes it difficult to use this method with this device.

2.5 First test

2.5.1 Hardware preparation

The EPOC needs some preparation before being ready to use. This first step consists in preparing the felt pads with a saline solution, which increases their conductivity–otherwise they are not usable. Afterwards, they are placed on the device one by one.

2.5.2 Emotiv control panel

Once the device is on someone's head, the quality of the signal from the electrodes has to be checked. The Emotiv control panel provides a tool which gives a color visual feedback for every electrode and a label about the quality of the signals recorded globally by the device; it informs the user if the signal is "ok" or "noisy". An example of the control panel is given in figure 7.

[4]Locations on the EPOC electrodes according to the 10-20 system: AF3, AF4, F3, F4, FC5, FC6, F7, F8, T7, T8, P7, P8, O1, O2

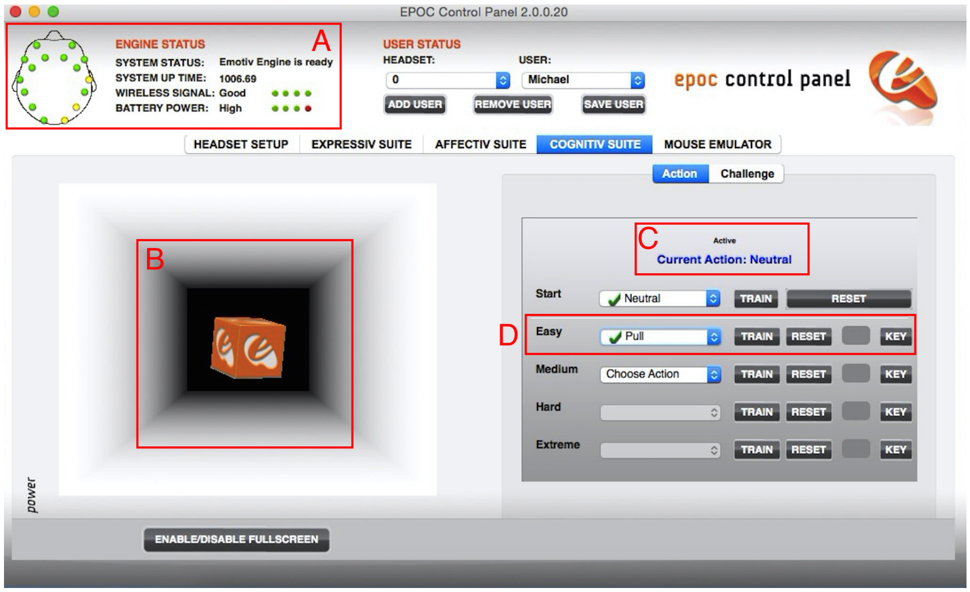


Figure 7: Emotiv software. (A) Engine status. On the left, the quality of the signal from each electrodes represented with a color feedback. On the right down, the global quality of the signal and the battery power status. (B) Visual feedback. The cube moves according to the state detected. (C) Actual state detected. (D) Control panel to choose the training and the command.

2.5.3 Training

The Emotiv software provides a training tool where one can choose to train up to four commands with a neutral state. The neutral state being the basic state of mind you are supposed to be in while not thinking of a particular action.
To train a given state, the user has to choose to do it, which will open a 8 seconds window during which he has to have a constant pattern of thoughts in order to allow the system to recognize it properly. This action can be repeated any number of time.
It has to be noted that commands can be trained in any order, while the first one is supposed to be «easy», and the fourth «very difficult» to master.
During training, the user receives a visual feedback of his action with a cube that is allowed to move accordingly to the train commands.

2.5.4 Strategy

As explained in previous sections, motor imagery could not be used to train with the EPOC. Nevertheless, we tried it as well as other well know strategy such as mental calculation [14] or thoughts related to the visual feedback of the Emotiv training software–typically imagining pushing the cube.

2.5.5 Results

The first test of the EPOC aimed to get used to the device as well as to know if it was reliable enough for our experiment. It has to be noted that the device was already the subject of a research paper that claimed it was performing above chance level with evoked potential [10].
After a total training time of 8 hours–all done by myself–the conclusions were the following: firstly, with non evoked-potential, it appears to be possible to get close to the chance level with one command if it is trained alone during approximately two hours. However, training a second command was much harder–the result of it were roughly less than half as good as the first–but more importantly, the quality response of the first command decreased with the training of the second[5].

2.5.6 Discussion

This first test opens the possibility to use one command to control a robot and hypothetically a second one. However, those results had to be taken with caution for many reasons; indeed, it appears that such a training is highly subject dependent, thus may vary a lot between different individuals. Furthermore, for some unknown reasons at the time, it is not possible to use EEG data from some people to get meaningful and reproducible actions–phenomenon described as «BCI illiteracy». Interestingly, this does not seem to depend of the technology used, but rather on the physiology of the user itself [14].
Finally and more importantly, as already said in section 2.4.1, the software used to treat the signals does not filter out the components from facial muscles contraction and eyes movements which could potentially invalidate the whole training. Also, it is arduous not to move eyes while using this device into a real world application as well as for the training itself. Similar observation can be made concerning potential arising from facial muscles contractions.
In order to further explore these limitations, a second user did a pretest to gather more data.

2.5.7 Second test

The second test was done with Mehmet. The results can be seen in section 6 (figure 11). The protocol for this test was the following: first, Mehmet trained himself with the device for 15 minutes before doing a test. The total time of the test was 2 minutes and 20 seconds. Every 20 secondes, he was asked to think about a different command in a precise order.

[5] The method used to quantify the results from this first test was a «metric game» which is provided with the Emotiv software. However, it does not provide an absolute scale, which make it difficult to present and compare different results, which is the reason why no data are provided in this section. Nonetheless, this was corrected in the second preliminary test (see section 2.5.7).

The summary of this test[6] can be seen below in table 1 with the related sensitivity test corresponding to each interval. The sensitivity was calculated with the following formula:

$$Sensitivity = \frac{True\ positive}{True\ positive + False\ negative} \tag{1}$$

True positive: period of time during which the command asked appeared.
False negative: period of time during which the command asked did not appear.

The sensitivity was chosen as the statistical indication to analyze the data regarding our goal: as we want to have a given command when we ask for it, it is important to know quantitatively how many time the asked command will happen–true positive–and more importantly how many time another command will happen–false negative. A sensitivity of 1 indicates that the command asked was always happening during the time period it was asked. On the opposite, a sensitivity of 0 indicates that it was never happening.

	0 to 20''	20'' to 40''	40'' to 1'	1' to 1'20''	1'20'' to 1'40''	1'40'' to 2'	2' to 2'20''
cmd asked	neutral	push (cmd 1)	neutral	pull (cmd 2)	neutral	push (cmd 1)	pull (cmd 2)
Sensitivity (TPR)	1	0.5	0.7	0.3	0.5	0.1	0.6

Table 1: Summary of the protocol and sensitivity–also called true positive rate– of the second test for every interval of time. The test was done with Mehmet.

Those results, even if they should also be taken with caution, show us that in this case, unlike for my training, that the second command could potentially get better results than the first. Indeed, the first command has a mean sensitivity of 0.3 against a mean of 0.45 for the second command.
Moreover, the mean sensitivity of all the commands asked–which does not take into account the neutral state as we cannot use it as a command–equal 0.375, which is higher than the chance level[7]
To conclude, those results suggested that the device could potentially be used for the control task we envisioned.

[6]This protocol was designed to test how well a user can keep up with a command during a period of time which would be needed to move the Roombots in a room as well as to test the capacity of the user to switch between commands.

[7]Indeed, for every command asked, either the neutral state, the push state or the pull state could occur, therefore the chance level in a given period of time corresponds to a sensivity of 0.33.

3 Interface

3.1 First strategy

Considering the first results, we decided to create an interface which will allow a user to move modular robots on the ground of a room and also to command them to execute a particular pre-constructed form, such as a table. We choose to implement our interface in simulation as the Roombots hardware was not available at the time.
In order to have enough degrees of freedom, two commands being the maximum we could get from the EPOC, we decided to add another modality to our interface, namely speech recognition. This will allow us to use both commands from the EPOC to move the modular robots in a 2D plan[8] and use speech recognition to choose the desired form. Therefore, by adding this modality, we enter in the scope of hybrid BCI[9].
To implement this strategy, we planned to use Webots [26] as a simulation platform where cube bots, whose movements would have been simulated with the help of the Million module march Algorithm [11], would have been our modular robots[10].

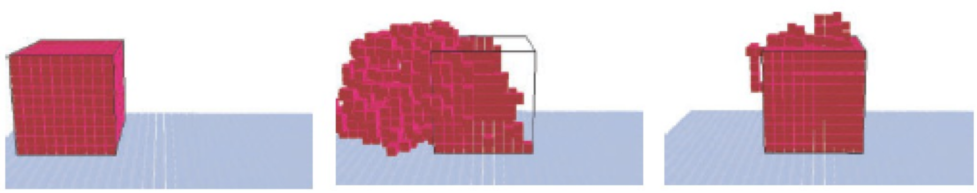


Figure 8: Simulation of a 1000-module robot moving over flat ground: From left to right, the movement from the original state to the goal location [11].

Moreover, we wanted to give the user an immersive experience despite being in a simulated environnement; this would have been achieved with an Oculus Rift. However, as the EPOC and the Oculus Rift could not be worn together, we had to abandon this idea.

[8]This could be achieved by giving to one command the «forward» direction and to the other the «rotation» while when no state is detected, i.e. neutral state, the robots would stop.
[9]It has to be noted that, as the modular robots are moving with the help of an algoritm, we were technicaly already in the scope of an hybrid BCI
[10]The Million module march would have been implemented by the supervisors of the project.

3.2 Second strategy

Afterwards, as two Roombots modules were available, we decided to change our strategy to go into something realistic with those modules and a passive elements, namely an ikea table. Thus, it solved the major issues of the first strategy, which was the lack of immersion and the tangibility of the interface.

Figure 9: Four Roombtos modules connected the Lack ikea table [20, 27].

To implement this strategy, we used MATLAB to control the Roombots according to EEG input and a smartphone connected to a computer for the speech recognition input. The flow of information was as follows:

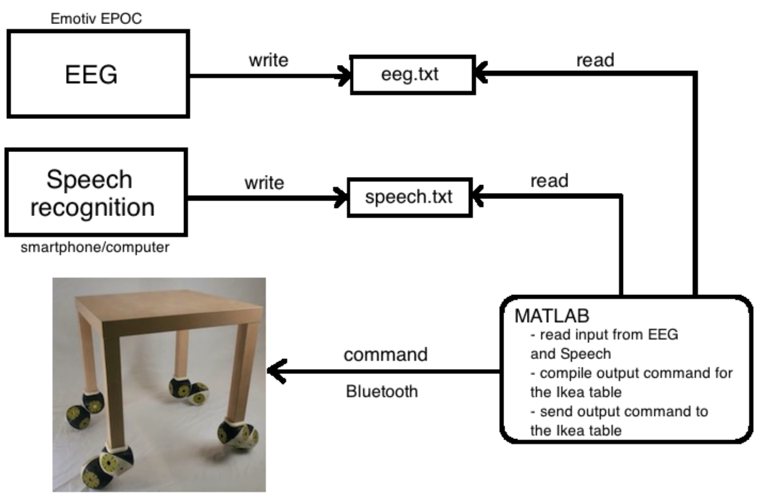


Figure 10: Commands from the EPOC and the smartphone are sent to the computer and written into .txt file. Those are read by MATLAB which then computes the action corresponding to the input and sends the command via bluetooth to the Roombots module.

3.3 Tools

The major challenge in this part was to create an interface that could be used with the EPOC input considering we could only get commands back from the Emotiv software by linking a given command to a letter that would be written sequentially in a .txt file.
The MATLAB code used was an existing one. Mehmet modified it to be used for our application and I implemented the necessary conditions for our project. The code written can be found in section 7.1 at the end of the report.
To use speech recognition, OpenEars [12], a free speech recognition framework using CMUSphinx [22] was chosen. It has the advantage to be implemented in Objective-C, and thus could be built on a iphone, which was convenient considering our main objective to allow a user to use commands from the EPOC or speech recognition once entering a room to give order to the Roombots without being constraint to stay at a given location. The code can also be found at the end of the report in section 7.2.

3.4 Protocol testing

When we tested our protocol–which can be seen in section 7.3–to validate it, several serious problems appeared, not from the experiment design itself, but from the reliability of the EPOC. Indeed, it was almost impossible to control the ikea table with this modality while the speech recognition system worked fine.
As the user who tested the protocol never tested the EPOC before, it was a sign that our previous results from the two first tests were not sufficient to proceed with an external user study.
Therefore, it was decided at this point that an extensive test with more subjects should be done with the EPOC.

4 Evaluative user study

We designed a new internal user study from which the training was defined precisely. Stéphane, Mehmet, Simon and Florin volunteered to be part of this study.
The protocol was the following: each subject would train four times over the course of one week. The training was defined as described in table 2:

neutral	neutral	push	push	push	push	neutral	neutral
push	push	push	push	second	second	neutral	neutral
second	second	push	push	push	push	second	second
second	second	second					

Table 2: Training sequence for the internal user study. Each of the name corresponds to a training of 8 seconds. The sequence is execute left-to-right, from the top to the bottom

Each subject was asked to train the push state as the first command, but was given the choice of the seconds. Also, the mental strategy was left to them. After the first and the fourth training session, they undergo a quantitative test– the same that was done for the second pretest with Mehmet.

4.1 Results

Simon choose as a second command the «drop» state. His strategy was to think of a music playing for the first command while he tried to induce fear by imagining being on a platform and looking down for the «drop» state. The neutral state was just about thinking about nothing in particular.
Stéphane choose the «lift» as the second command. He did arithmetic mentally to train the neutral state and imagined for command one and two the cube moving accordingly. He also thought at the same time of the word corresponding to the action.
Mehmet chose the «pull» state for his second command. As Simon, he chose nothing particular to train the neutral state. To train the two other states, he imagine the cube moving according to the given state.
Florin also chose the «pull» state for his second command. To train the pull state, he tried to feel it coming to him, and to feel it leaving for the push state.

Below are the results of those tests:

	0 to 20''	20'' to 40''	40'' to 1'	1' to 1'20''	1'20'' to 1'40''	1'40'' to 2'	2' to 2'20''
cmd asked	neutral	push (cmd 1)	neutral	lift (cmd 2)	neutral	push (cmd 1)	lift (cmd 2)
Sensitivity (TPR)	0.85	0.025	0.88	0.05	1	0.15	0

Table 3: Summary of the protocol and sensitivity–also called true positive rate– of the first test with Stéphane. Mean sensitivity: 0.06.

	0 to 20''	20'' to 40''	40'' to 1'	1' to 1'20''	1'20'' to 1'40''	1'40'' to 2'	2' to 2'20''
cmd asked	neutral	push (cmd 1)	neutral	lift (cmd 2)	neutral	push (cmd 1)	lift (cmd 2)
Sensitivity (TPR)	1	0.775	0.45	0.575	0.875	0.65	0.5

Table 4: Sensitivity of the second test with Stéphane. Mean sensitivity: 0.63.

	0 to 20''	20'' to 40''	40'' to 1'	1' to 1'20''	1'20'' to 1'40''	1'40'' to 2'	2' to 2'20''
cmd asked	neutral	push (cmd 1)	neutral	pull (cmd 2)	neutral	push (cmd 1)	pull (cmd 2)
Sensitivity (TPR)	0.95	0.075	0.95	0.1	0.875	0.15	0.05

Table 5: Sensitivity of the first test with Mehmet. Mean sensitivity: 0.09.

	0 to 20''	20'' to 40''	40'' to 1'	1' to 1'20''	1'20'' to 1'40''	1'40'' to 2'	2' to 2'20''
cmd asked	neutral	push (cmd 1)	neutral	pull (cmd 2)	neutral	push (cmd 1)	pull (cmd 2)
Sensitivity (TPR)	0.9	0	0.725	0.175	0.6	0	0.15

Table 6: Sensitivity of the second test with Mehmet. Mean sensitivity: 0.08.

	0 to 20''	20'' to 40''	40'' to 1'	1' to 1'20''	1'20'' to 1'40''	1'40'' to 2'	2' to 2'20''
cmd asked	neutral	push (cmd 1)	neutral	down (cmd 2)	neutral	push (cmd 1)	down (cmd 2)
Sensitivity (TPR)	0.95	0	0.8	0.1	0.775	0	0.05

Table 7: Sensitivity of the first test with Simon. Mean sensitivity: 0.08.

	0 to 20''	20'' to 40''	40'' to 1'	1' to 1'20''	1'20'' to 1'40''	1'40'' to 2'	2' to 2'20''
cmd asked	neutral	push (cmd 1)	neutral	down (cmd 2)	neutral	push (cmd 1)	down (cmd 2)
Sensitivity (TPR)	1	0	1	0	1	0	0

Table 8: Sensitivity of the second test with Simon. Mean sensitivity: 0.

	0 to 20''	20'' to 40''	40'' to 1'	1' to 1'20''	1'20'' to 1'40''	1'40'' to 2'	2' to 2'20''
cmd asked	neutral	push (cmd 1)	neutral	pull (cmd 2)	neutral	push (cmd 1)	pull (cmd 2)
Sensitivity (TPR)	0.8	0	0.85	0.7	0.65	0	0.55

Table 9: Sensitivity of the first test with Florin. Mean sensitivity: 0.31.

	0 to 20''	20'' to 40''	40'' to 1'	1' to 1'20''	1'20'' to 1'40''	1'40'' to 2'	2' to 2'20''
cmd asked	neutral	push (cmd 1)	neutral	pull (cmd 2)	neutral	push (cmd 1)	pull (cmd 2)
Sensitivity (TPR)	1	0	1	0	1	0	0

Table 10: Sensitivity of the second test with Florin. Mean sensitivity: 0.

4.2 Discussion

As suspected during the protocol testing, it appears that it is not possible to get two commands out of the EPOC device only with EEG signals, that is without EMG and EOG components so that we can use it for our application.
The four subjects were asked to be careful not to move their head, neither their eyes if possible. Of course, this is a very difficult and non intuitive task to do. Even though, we suspect strongly that the best results–for example the second test of Stéphane, figure 13–was purely the result of potential arising from muscle contraction, as it shows almost perfect correlation to voluntary muscle contraction.

Moreover, it was noticed that the visual feedback of the cube going up and down also negatively impact the truthfulness of the results. Indeed, when the second command chosen was lift or down, when the cube started to move randomly in those directions, the subject would follow automatically it with his eyes, thus provoking a difference of potential that could be used as positive feedback loop by the system, which would appear to the naive user as a success–i.e. a true positive.

Concerning the training itself, in the case of Simon, Mehmet and Florin (respectively figure 14, 15; 16, 17; 18, 19), the mean sensitivity has even decreased from the first test to the second, i.e. after three more training sessions. For Florin and Simon, it even went to zero (figure 15 and 19). This might be explained by the training method chosen, which was to train in parallel both commands instead of sequentially.
Also, it appeared for everyone that either for the first command–push state–or for the second, that the false negative were mostly coming from the neutral state. Moreover, false negative were mostly coming from the second command

when the neutral state was asked. As far as we can say, this is certainly due to the classifier used by the Emotiv software.

5 Conclusion and future work

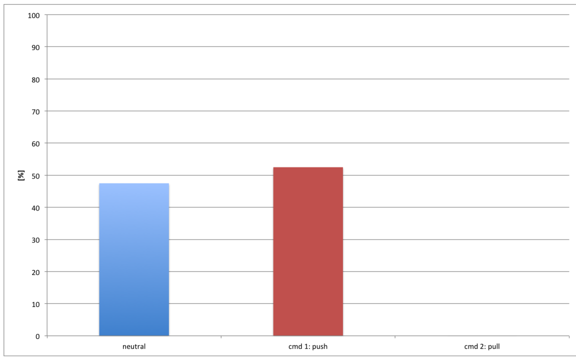
During this project, we tested an hybrid BCI to move a structure composed of modular robots connected to a table. Our modalities were EEG from the Emotiv EPOC neuroheadset and speech recognition, implemented in Objectiv-C with OpenEars. To test our system, we designed an external user study. After a pretest of the protocol, we decided not to run it because the modality provided by the EPOC did not match our minium requirements.
We further investigated the EPOC device and concluded that it was not suitable to control robot only with its EEG signal.

It has to be noted that after the first tests with the EPOC (see section 2.5 and 2.5.7) , we thought it would be reliable enough toward our needs. Moreover, even if we had some suspicions of biased results, we proceeded further in the project; at this point, we should have done the evaluative user study (see section 4) to remove all doubt. This would have prevent the cancellation of the external user study.

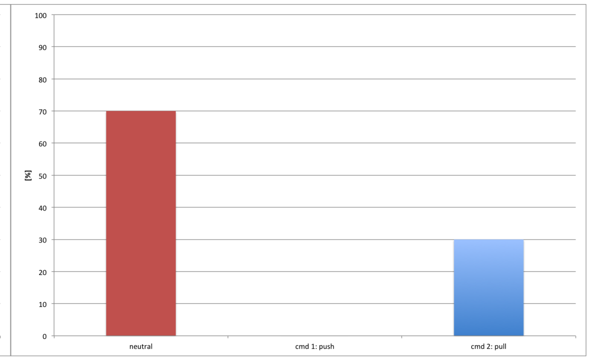
The next step would be to access the raw data from the EPOC and use a custom classifier which would filter out the EMG and EOG components, therefore allowing for a precise conclusion on the quality of the signal recorded by this device.
In the case where a good signal could be acquired, other mental strategies could be tested to train the user; for example, we explored one strategy where the user would induce himself a minor physical pain to train and then tried to remember it to activate commands. We also tested a strategy where the user think of a memories which bring back strong emotion to him. Both may be worth exploring according to our tests.

Nonetheless, EEG seems to be in itself a modality that is very limited. Indeed, even with medical devices, it appears to be very hard to get several outputs, not to mention the fact that those outputs are not reliable at 100%. As this issue does not seem to come from a technological problem, but rather from the limitation of the technique used (see section 2.1 and 2.3.2), EEG system are probably far from being an ideal modality for modular and usual robot control. We could suggest to explore further speech recognition to control modular robots–which worked fine according to our needs in the scope of this project–in combination with user movement or augmented reality (see section 1.2). For this later case, the so-called «smart glasses» could be worth exploring instead of the ipad [23].

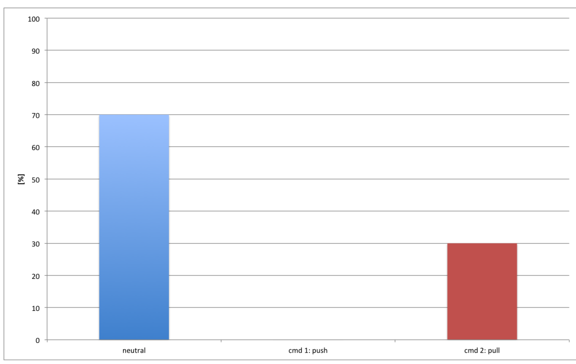
6 Figures



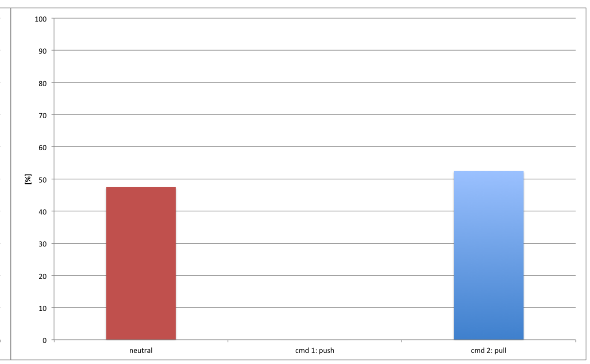
(a) Push



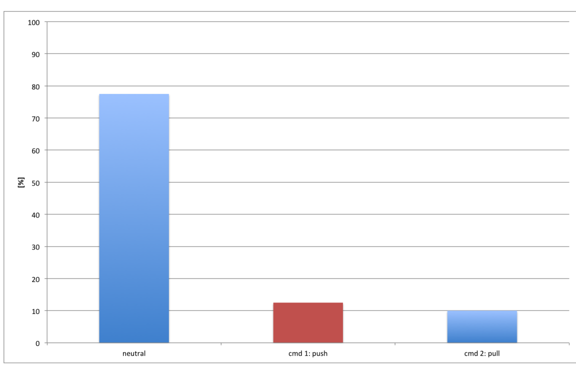
(b) Neutral



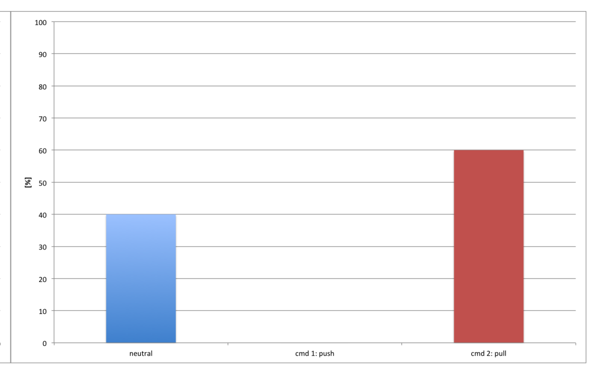
(c) Pull



(d) Neutral

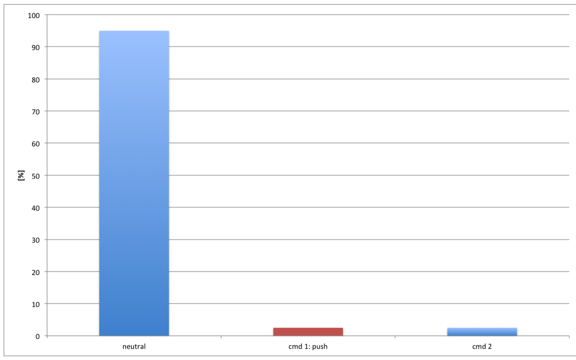


(e) Push

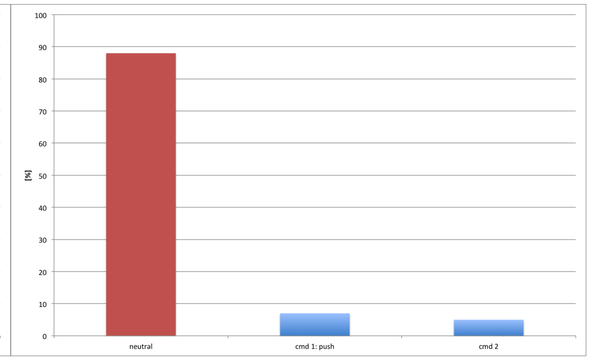


(f) Pull

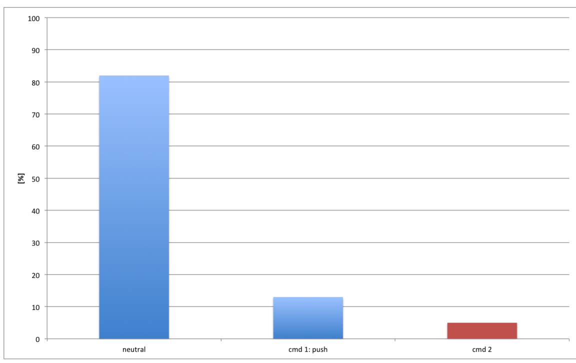
Figure 11: Results from the first test with Mehmet. Each figure represents a period of 20 seconds where one specfic command was asked to the user; in red, the command asked. The first 20 seconds–always neutral state asked–are not shown to keep all figures from a test on one page; results are given numerically in the legend: here, 100% in neutral state.



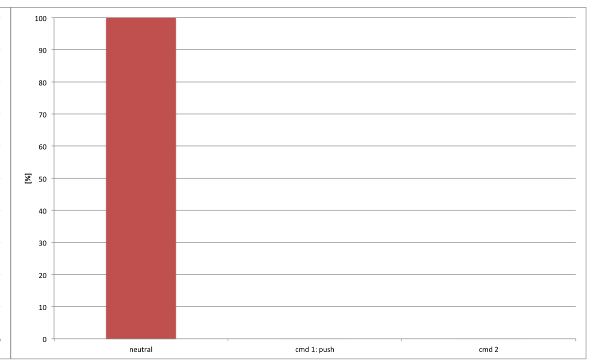
(a) Push



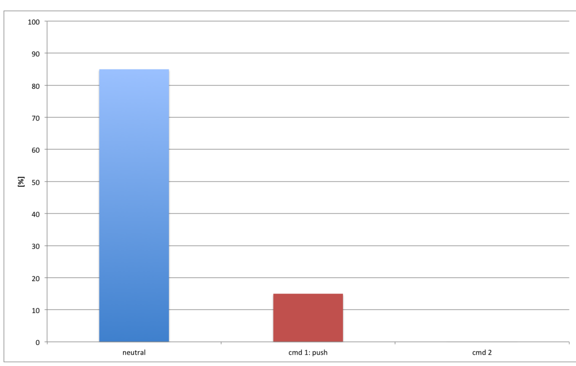
(b) Neutral



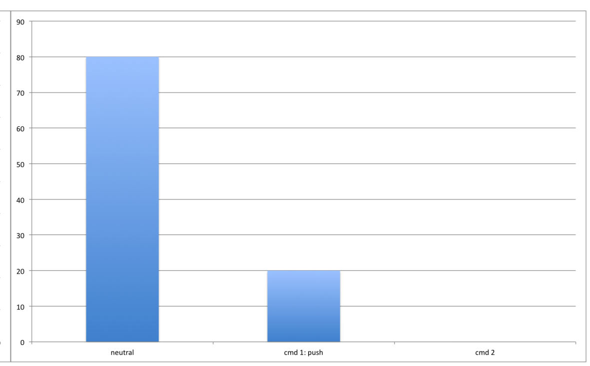
(c) Lift



(d) Neutral

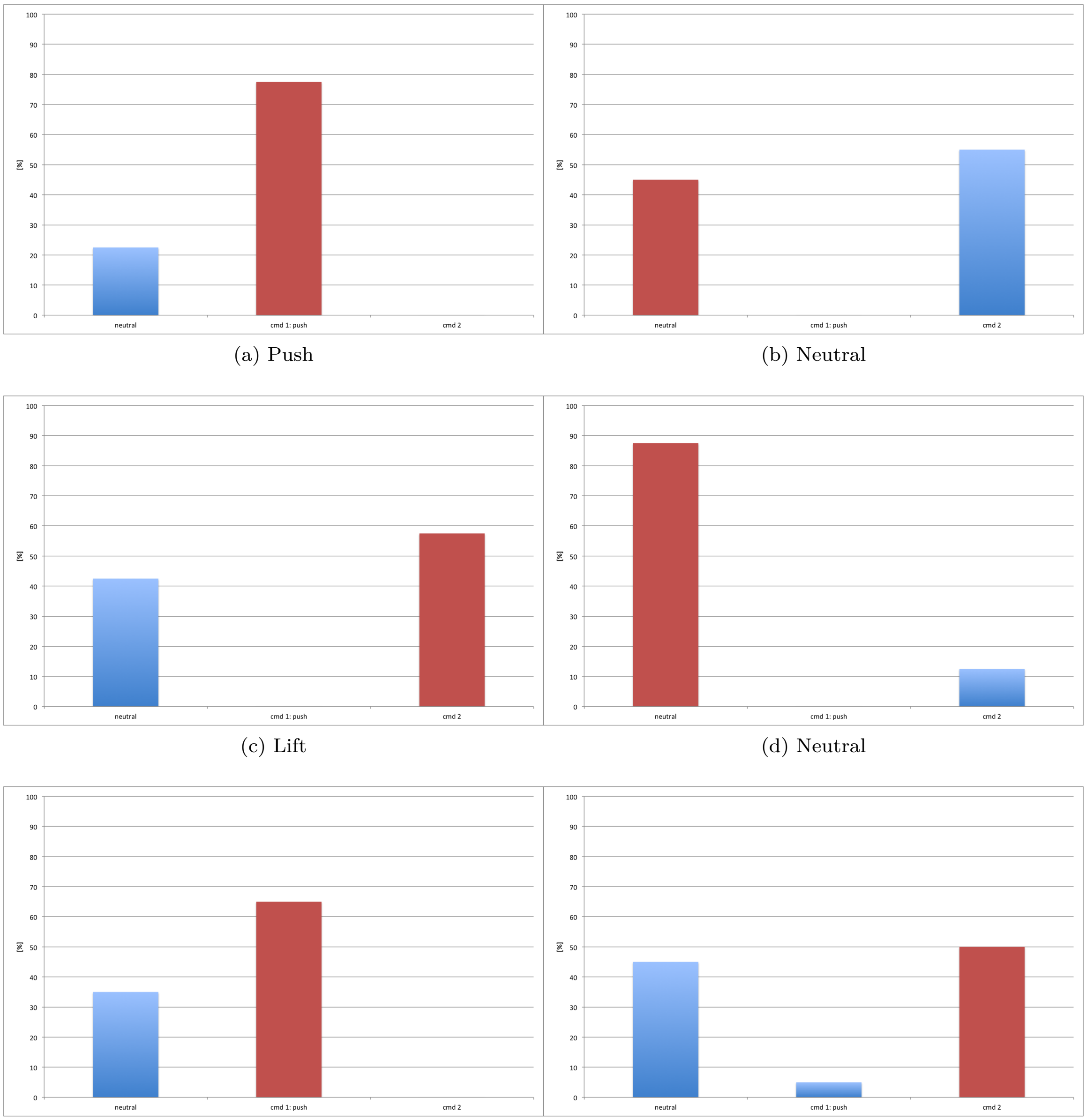


(e) Push



(f) Lift

Figure 12: Results from test one with Stéphane. The second command is «lift». First 20 seconds not shown–85% in neutral state, 10% in push state, 5% in lift state.



(a) Push

(b) Neutral

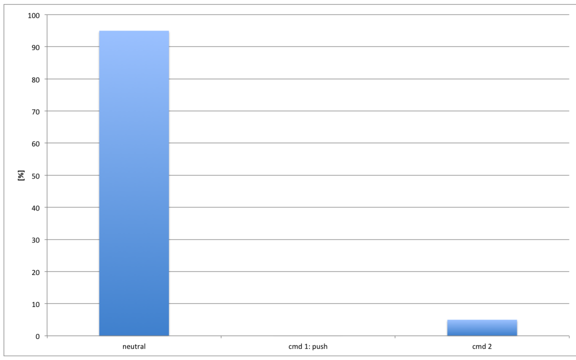
(c) Lift

(d) Neutral

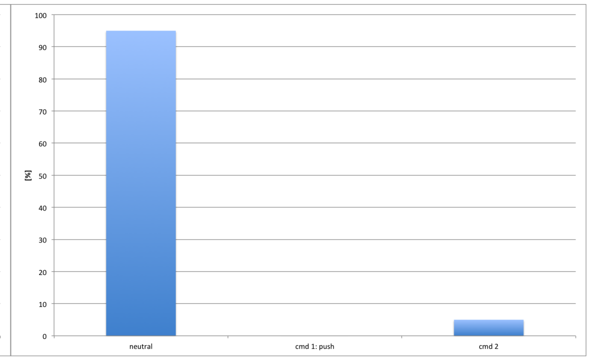
(e) Push

(f) Lift

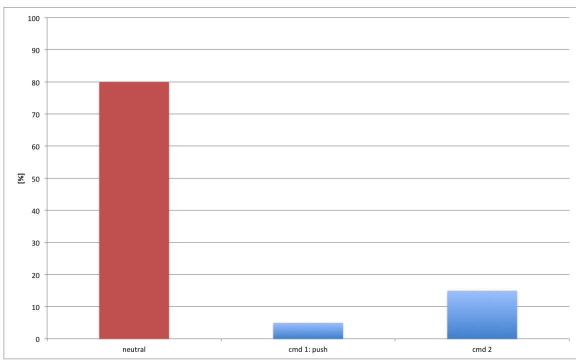
Figure 13: Results from test two with Stéphane. The second command is «lift». First 20 seconds not shown–100% in neutral state



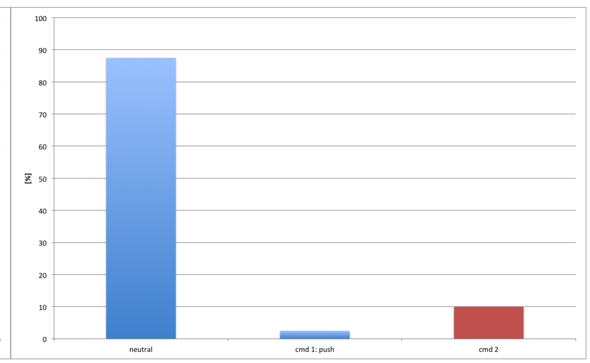
(a) Push



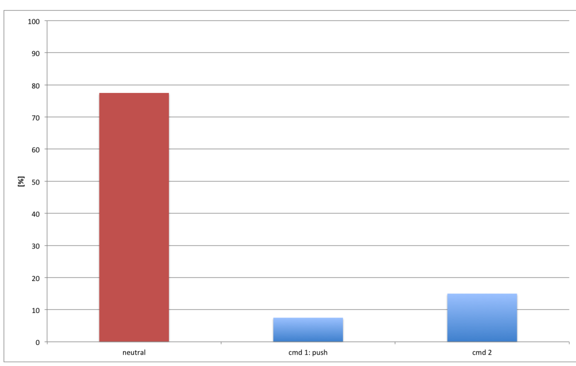
(b) Neutral



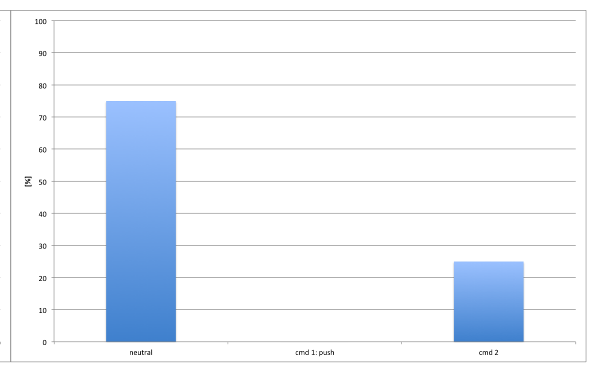
(c) Down



(d) Neutral

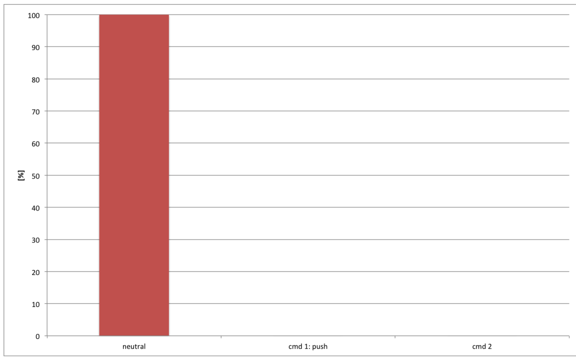


(e) Push

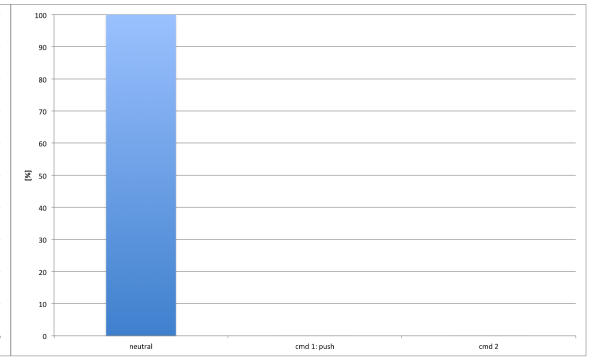


(f) Down

Figure 14: Results from test one with Simon. The second command is «down». First 20 seconds not shown–95% in neutral state, 5% in down state.

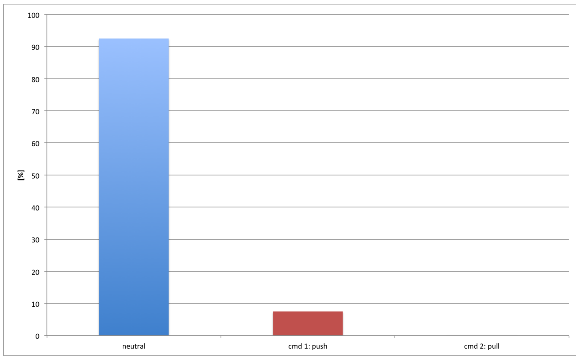


(a) Neutral

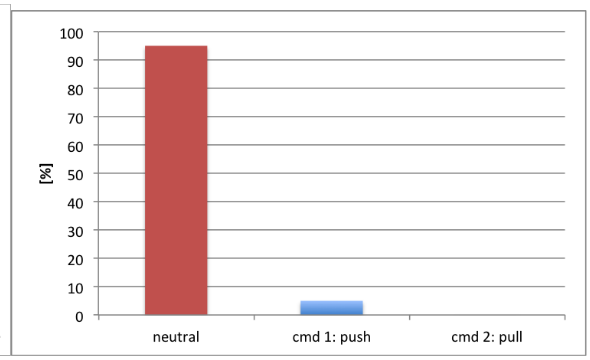


(b) Push & Down

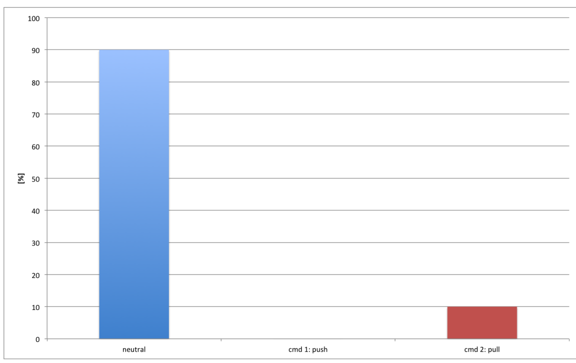
Figure 15: Results from test two with Simon. The second command is «down». a) all figures when neutral state was aked, b) all figure when push or down state were asked.



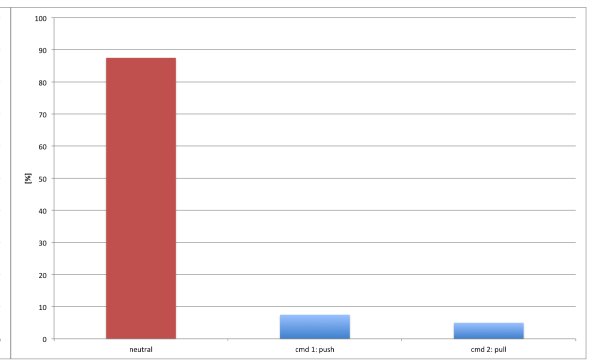
(a) Push



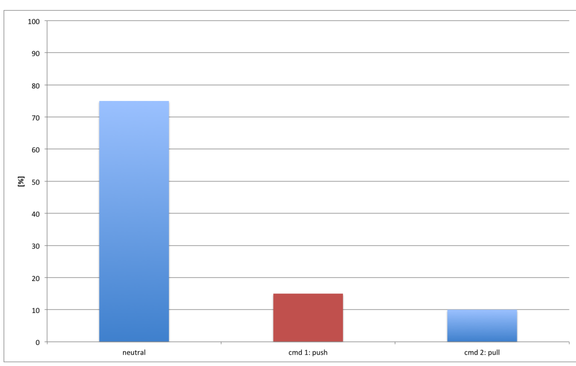
(b) Neutral



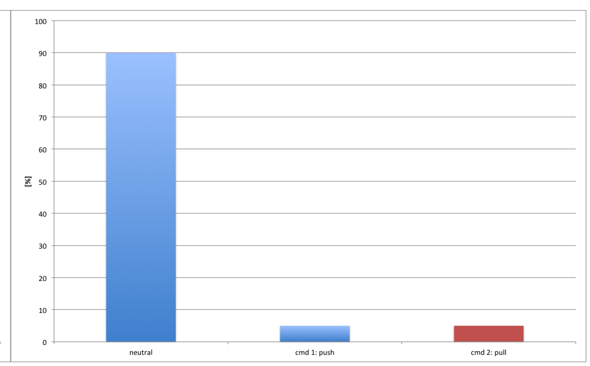
(c) Pull



(d) Neutral

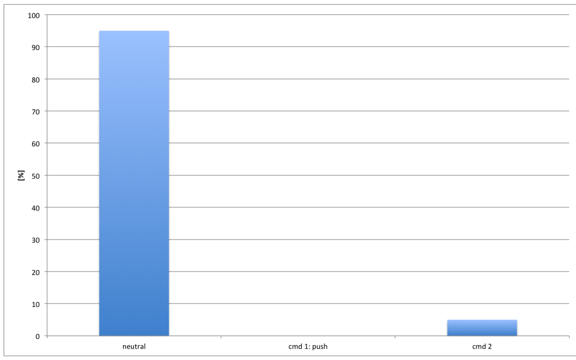


(e) Push

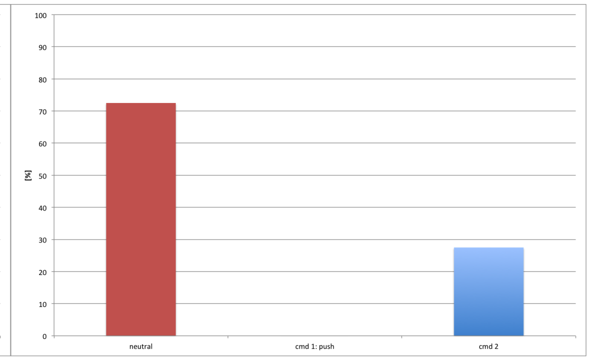


(f) Pull

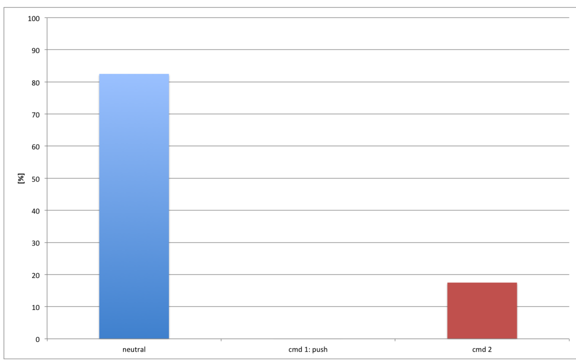
Figure 16: Results from test one with Mehmet. The second command is «pull». First 20 seconds not shown–95% in neutral state, 5% in push state.



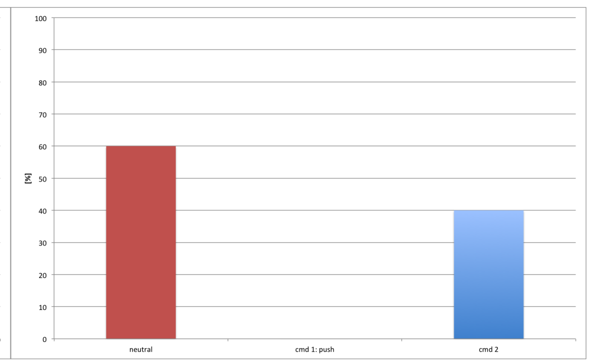
(a) Push



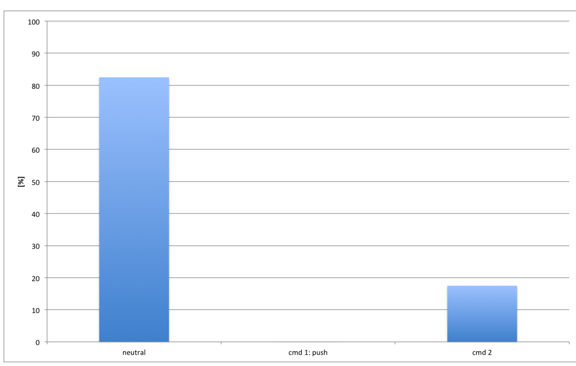
(b) Neutral



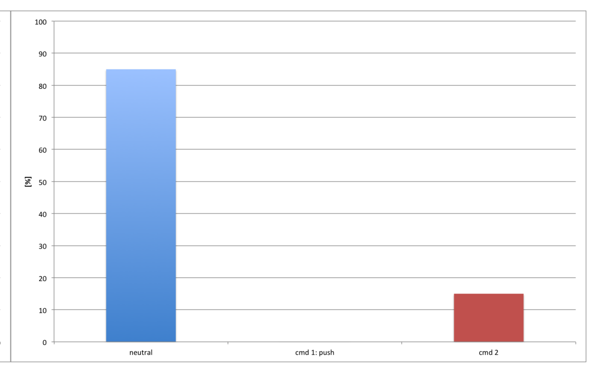
(c) Pull



(d) Neutral

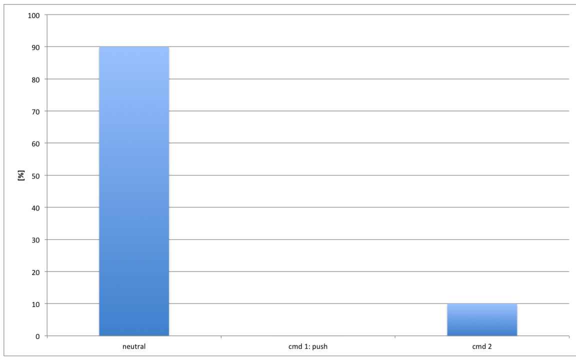


(e) Push

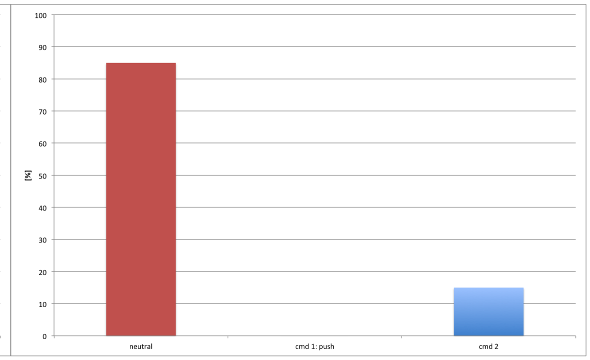


(f) Pull

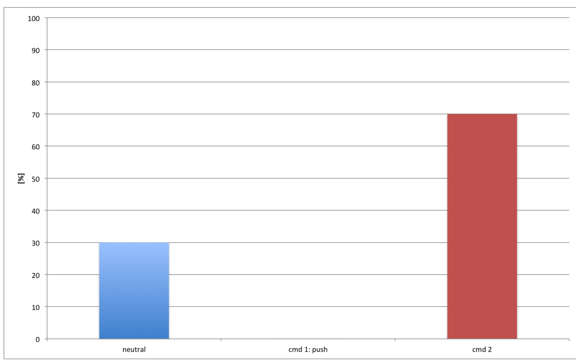
Figure 17: Results from test two with Mehmet. The second command is «pull». First 20 seconds not shown–90% in neutral state, 10% in push state.



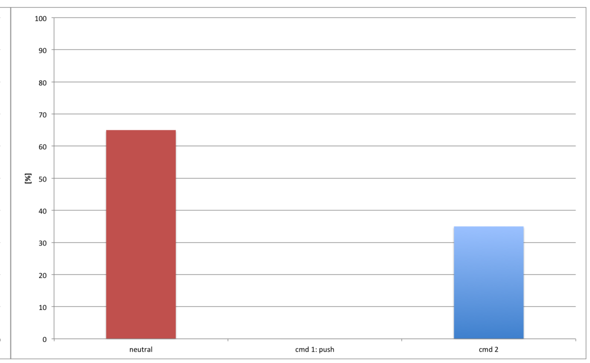
(a) Push



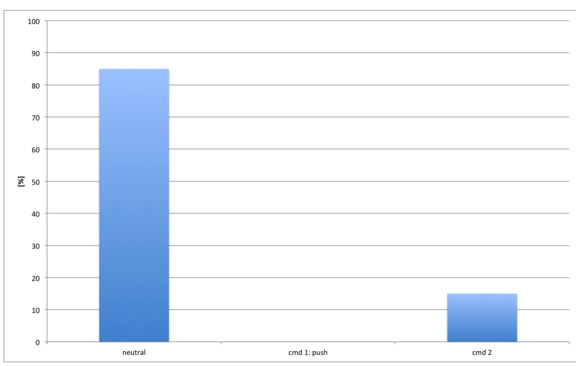
(b) Neutral



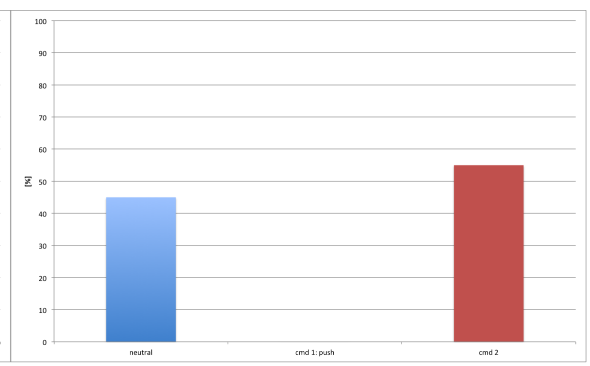
(c) Pull



(d) Neutral

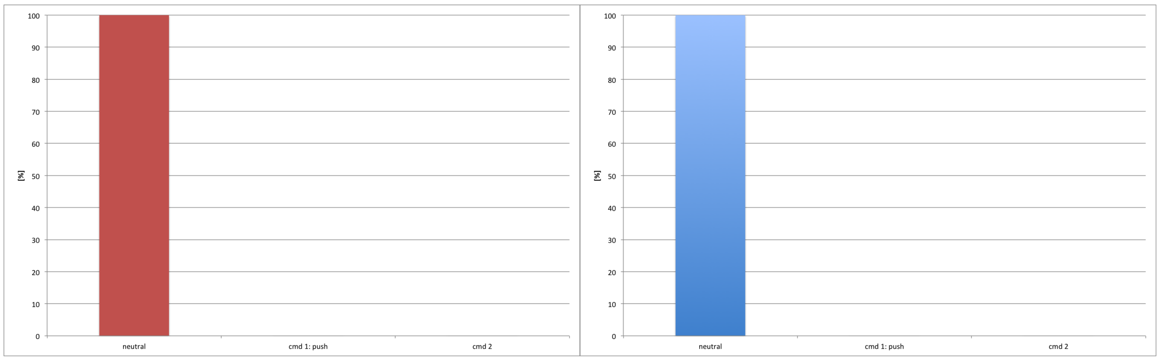


(e) Push



(f) Pull

Figure 18: Results from test one with Florin. The second command is «pull». First 20 seconds not shown–80% in neutral state, 5% in push state, 15% in pull state.



(a) Neutral

(b) Push & Pull

Figure 19: Results from test two with Florin. The second command is «pull». a) all figures when neutral state was aked, b) all figure when push or down state were asked.

Acknowledgements

I would like to express my special thanks to my supervisors, Stéphane Bonardi, Mehmet Mutlu and Simon Hauser for their great support as well as to Florin Dzeladini for his implication in the internal study and for the loan of the Emotiv EPOC.
Also, I would like to thanks Professor Auke J. Ijspeert and Stéphane Bonardi for having accepted the proposition of this project.

7 Annexes

7.1 Code MATLAB

```
% --- Executes on button press in togglebutton_speech_eeg.
function togglebutton_speech_eeg_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton_speech_eeg (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%to do: open first the files and write in them z if they are empty
%to do: prevent the robot to get crazy we a bouton to be push in case it
%isaa
%not able to detect the stop
char importDataString;
char importDataStringEmotiv;
char previousEntrySpeech;
char previousEntryEmotiv;

%init of the variable
previousEntrySpeech = 'z';
previousEntryEmotiv = 'z';
command = 'z';

% Hint: get(hObject,'Value') returns toggle state of togglebutton_speech_eeg
while get(hObject,'Value')
    A = importdata('dataUnput.txt'); %A is a cell value if the file is not empty
    importDataString = A{1}; %change format form cell to a string

    sizeImportDataStringEmotiv = [1 Inf];
    FID = fopen('dataUnputEmotiv.txt');
    importDataStringEmotiv = fscanf(FID,'%s',sizeImportDataStringEmotiv); %is a already a char
    fclose(FID);

    %loop only take the last character of the import datas, and
    %change the value of the command if there was a change of the datas,
    %which mean that a new command has been given either from speech for
    %the if just behind, or from the emotiv for the one after

    if length(importDataString) > length(previousEntrySpeech)
    isDifferentSpeech = strcmp(previousEntrySpeech(end), importDataString(end));

        if (isDifferentSpeech == 0)
        previousEntrySpeech = importDataString(end);

        %Forward = a, stop b, turn c, init z
```

```
    %Boucle pour le speech
        if strcmp(previousEntrySpeech, 'a')
            command = 'FORWARD';
        elseif strcmp(previousEntrySpeech, 'b')
            command = 'STOP';
        elseif strcmp(previousEntrySpeech, 'c')
            command = 'TURN';
        elseif strcmp(previousEntrySpeech, 'z')
            command = 'STOP';
        else
            command ='BUG';
        end
    end
end

if length(importDataStringEmotiv) > length(previousEntryEmotiv)
isDifferentEmotiv = strcmp(previousEntryEmotiv(end), importDataStringEmotiv(end));

   if (isDifferentEmotiv == 0)
    previousEntryEmotiv = importDataStringEmotiv(end);

        %Forward = a, stop b, turn c, init z/ Boucle for Emotiv
        if strcmp(previousEntryEmotiv, 'a')
            command = 'FORWARD';
        elseif strcmp(previousEntryEmotiv, 'b')
            command = 'STOP';
        elseif strcmp(previousEntryEmotiv, 'c')
            command = 'TURN';
        elseif strcmp(previousEntrySpeech, 'z')
            command = 'STOP';
        else
            command ='BUG';
        end
   end
end

speed_eeg = 0;          % Scale to 2500 and -2500
direction_eeg = 0;          % Scale to 10 and -10

if strcmp(command,'FORWARD')
    speed_eeg = 1000;         % Scale to 2500 and -2500
    direction_eeg = 0;          % Scale to 10 and -10
elseif strcmp(command,'TURN')
    speed_eeg = 1000;         % Scale to 2500 and -2500
    direction_eeg = 5;          % Scale to 10 and -10
elseif strcmp(command,'STOP')
     speed_eeg = 0;         % Scale to 2500 and -2500
    direction_eeg = 0;          % Scale to 10 and -10
else
    speed_eeg = 0;          % Scale to 2500 and -2500
    direction_eeg = 0;          % Scale to 10 and -10
    disp('command is set wrong');
end

set(handles.slider_speed_demo2, 'Value', speed_eeg);
set(handles.slider_direction_demo2, 'Value', direction_eeg);
```

```
        speed = round(get(handles.slider_speed_demo2,'Value'));
        direction = round(get(handles.slider_direction_demo2,'Value'));

        direction_string = num2str(direction);
        disp(['Direction value will be set to ' direction_string ', speed ' num2str(speed)]);

        demo2_set_speed_direction( hObject, handles, speed, direction );

        pause(0.1)
    end
```

7.2 Code Objective-C

```
- (void) pocketsphinxDidReceiveHypothesis:(NSString *)hypothesis recognitionScore:(NSString *)
    recognitionScore utteranceID:(NSString *)utteranceID {
    NSLog(@"The received hypothesis is %@ with a score of %@ and an ID of %@", hypothesis, recognitionScore,
        utteranceID);

    NSString *stringToOutput = @"";

    //code to have the same written output as the one from emotiv
    if ([hypothesis isEqualToString:@"FORWARD"])
        {
            stringToOutput = @"a";
            self.myImageView.image = [UIImage imageNamed:@"forward.JPG"];
        }
    else if ([hypothesis isEqualToString:@"STOP"])
        {
            stringToOutput = @"b";
            self.myImageView.image = [UIImage imageNamed:@"stop1.JPG"];
        }
    else if ([hypothesis isEqualToString:@"ROTATE"])
        {
            stringToOutput = @"c";
            self.myImageView.image = [UIImage imageNamed:@"rotate.JPG"];
        }

    //code which keeps history
    NSString *entryToAdd = [NSString stringWithFormat:@"%@", stringToOutput];

    NSFileHandle *fileHandle = [NSFileHandle fileHandleForWritingAtPath:@"/Users/michaelmoret/Desktop/
        Roombots Demo with Gamepad MATLAB 2/DataUnput.txt"];
    [fileHandle seekToEndOfFile];
    [fileHandle writeData:[entryToAdd dataUsingEncoding:NSUTF8StringEncoding]];
    [fileHandle closeFile];

    /*code which doesn't keep history

    [entryToAdd writeToFile:@"/Users/michaelmoret/Desktop/DataUnput.txt" atomically:YES
        encoding:NSUTF8StringEncoding error:nil];*/
}
```

Figure 20: Objective-C code that write into .txt file word detected from speech recognition.

7.3 Protocol of the external user study

Research question What is the most efficient interaction strategy for a user to control a moving robotic structure in a room with obstacles.

Hypothesis We hypothesize that, using an EEG headset, it would be possible to move efficiently a robotic structure into a room with the help of a second modality due to the constraint of DOF (degree of freedom) that arises from such a technology.

Independant variables We will compare three modalities: computer keyboard buttons, speech recognition and a combination of speech recognition and EEG.

Control group The control group will be the computer keyboard as it is a very common and intuitive way to move objects into a 2D plan, like in video games.

Task definition As this system is design to help elderly and people with physical disabilities, we imagine a scenario where one will have to move a robotic structure, in our case Roombots modules with a passive element, a table, in order to bring from a point B to a point A, close to the subject, an object that would not be reachable otherwise.

Survey Two short surveys will be given to the subjects of this experiment. One will be completed prior to the experiment in order to assess the background of the user. The second will be given at the end of the experiment to collect meaningful informations to compare our three groups.

Scenario When the subject enters the room, it will start one of the three tasks after having been introduce to the goal of the experiment. To avoid order effects, we will change the order of the modalities between subjects and each of them will test all of them. We can therefore resume in three parts the experiment:

I. EEG and speech recognition First, the user will be introduce to the Emotiv control panel and then we will place on him the headset. Then, the user will follow the protocol in section 4. After this, we will conduct a short test (see section 4.1) in order to assess, with the help of the specificity (true positive over false negative) how well one has perform with the headset.
The subject will then be asked to either control the table to bring it to him (i.e. object on the table reachable by hand) with both commands from the EEG or with one from the EEG and one with the help of speech recognition.

II. Speech recognition The user will be introduce to the command available to him, and will test them to see how well the system can recognize what he said . As for the previous part, he will be asked to bring to table to him.

III. Computer keyboard buttons The user will be introduced to this modality and will then be asked to bring the table to him as for the previous section.

Dependant variables During the training period for each group: time, true positives, false negatives.
During the experiment: time, number of collisions, success or failure of the task (was the user able to grab the object in the given time period?).

References

[1] Ayberk Özgur: Intelligent User Interface for Roombots, semester project, 2013-2014.
`http://biorob.epfl.ch/page-104432.html`

[2] Jérémy Blatter: Mobile control interface for modular robots, master project, 2011-2012.
`http://biorob.epfl.ch/page-75451.html`

[3] `https://emotiv.com/epoc.php`

[4] Thinking Penguin: Multimodal Brain-Computer Interface Control of a VR Game, R. Leeb et al., 2013.

[5] Temporal and spatial complexity measures for electroencephalogram based brain-computer interfacing, S. J. Roberts et al., 1999.

[6] Cerebral location of international 10-20 system electrode placement, R. W. Homan et al., 1987.

[7] Automated cortical projection of EEG sensors: Anatomical correlation via the international 10-10 system, Koessler et al., 2009.

[8] 10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems, V. Jurcak et al., 2006.

[9] The ten twenty electrode system of the international federation, Jasper H. H., 1958.

[10] Performance of the Emotiv EPOC headset for P300-based applications, M. Duvinage et al., 2013.

[11] Fitch, R., & Butler, Z. (2008). Million module march: Scalable locomotion for large self-reconfiguring robots. The International Journal of Robotics Research, 27(3-4), 331-343.

[12] `http://www.politepix.com/openears/`

[13] The locked-in syndrome : what is it like to be conscious but paralyzed and voiceless? Laureys S. et al., 2005.

[14] Brain-Computer Interfacing for Intelligent Systems, A. Nijholt & D. Tan, 2008.

[15] EEG-Based Brain-Controlled Mobile Robots: A Survey, Luzheng Bi et al., 2013.

[16] The hybrid BCI, G. Pfurtscheller et al., 2010.

[17] Motor imagery and direct brain-computer communication, G. Pfurtscheller et al., 2001.

[18] Removal of ocular artifact from the EEG: a review, R. J. Croft & R. J. Barry, 2000.

[19] EMG contamination of EEG: spectral and topographical characteristics, I.I Goncharova et al., 2003.

[20] http://biorob.epfl.ch/cms/page-36376.html

[21] https://emotiv.com/company.php

[22] http://cmusphinx.sourceforge.net

[23] http://www.laforgeoptical.com

[24] 2013 Pearson Eduction, Inc.

[25] A hybrid brain–computer interface based on the fusion of electroencephalographic and electromyographic activities, R .Leeb et al., 2011.

[26] https://www.cyberbotics.com/overview

[27] http://www.ikea.com/us/en/catalog/products/40104270/

[28] A Brain–Computer Interface Controlled Auditory Event-Related Potential (P300) Spelling System for Locked-In Patients, A. Kubler et al., 2009.