Development of an informatic simulation for a brain-computer interface in an auditive environment

Thibaut Bertrand

Rapport de stage
04/03/19 – 30/07/19

Master Compétences Complémentaires en Informatiques
Année 2018 – 2019

Supervisors : Jérémie MATTOUT and Emmanuel MABY

Laboratory presentation

The Centre de Recherches en Neurosciences de Lyon (CRNL), led by Olivier BERTRAND is composed of 14 laboratory, with a total of 380 members from the Inserm, the CNRS or the University Lyon 1. The domains of research ranges from the bench to the patients and from the gene to the behaviour, and everything in-between (Fig. 1). The objective of this centre is to link those different level of study and to reinforce a translational research thanks to the perpetual exchanges between fundamental concepts and clinical challenges.

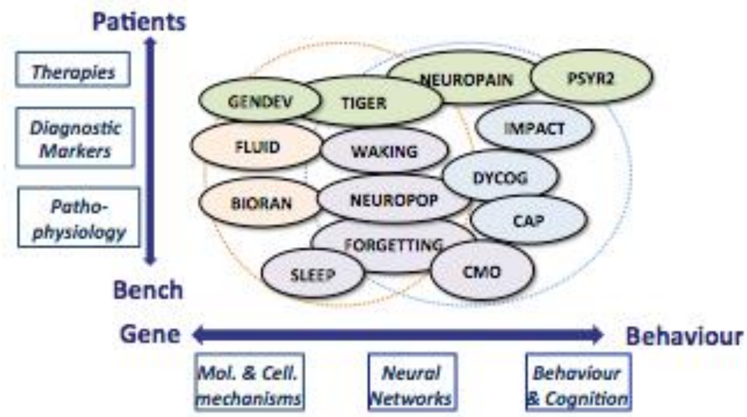


Figure 1: Team repartition inside of the CRNL. *For more details about the CRNL: https://crnl.univ-lyon1.fr/index.php/fr.*

Moreover, the CRNL possess, or is associated to, external research platforms, like the Cermep (specialized in *in-vivo* imagery) or Neuro-Immersion (virtual reality, brain signals monitoring and stimulation simultaneously) to name a few. Finally, the CRNL is of course in partnership with the hospital environment and its different services, like rehabilitation, psychiatry, neurology, neuropediatric…

This internship takes place at the Dynamique Cérébrale et Cognition (DYCOG) laboratory, also led by Olivier Bertrand. The laboratory is composed of 52 members, from doctoral students to engineers and researchers, divided into 3 teams. Those teams are: COPHY, led by Jérémie MATTOUT and Mathilde BONNEFOND, PAM, led by Anne CACLIN and Perrine RUBY, and EDUWELL, led by Jean-Philippe LACHAUX and Antoine LUTZ.

For this internship, I will be part of the COPHY team, under the supervision of Jérémie MATTOUT and Emmanuel MABY, the research engineer of the team. The team organize two weekly meetings. The first is on Monday, where one voluntary member (rarely more) can present its work in details, train for a coming conference, or present something else that can be of general interest. The second is on Tuesday, and only the part of the team managed by Jérémie MATTOUT is present. The aim of this meeting is to make a point of what was done the previous week and what will be done in the coming one. Here, everyone should make at least a very short talk about this to keep the others members informed of what's happening in the team. This is also the moment to discuss of results, encountered problems, other obstacles or organizing upcoming conferences.

Finally, a third meeting is generally organized on Thursday and is open to all the members of the CRNL. The aim of this meeting is to allow anyone to present its methodological problem in front of specialists from other teams and to discuss about a solution or a better way to do things. This can also be the moment to promote a new method or a new software that can interest other members. This also promote more interaction and dialogue between different teams.

Introduction

Our brain is the most complex organ to understand to date. Neurosciences specialise in studying it by combining the knowledge of specialists coming from different horizons: biology, informatics, linguistic, medicine, psychology… This domain contains a great variety of approaches, methods and animal models. This internship was part of a project in computational neurosciences, a branch that relies on computational approaches like statistics, mathematics and digital simulations as a model. Moreover, on an applicative point of view, the project also revolves around the development of a brain-computer interface, or BCI.

BCI encompass an ensemble of relatively new technologies and methods. They already exist since the 70s (Vidal, 1973), but only started to fully emerge in the last decade (Hochberg et al., 2012, for example). The aim of such a technology is to overcome our need of intermediate hardware, like mouse or keyboards, to interact with machines. Broadly speaking, the brain is an organ which cells, the neurons, produces ionic currents when stimulated, resulting in the production of electrical and magnetic signals. Those signals can be captured by different means, like MRI (Magnetic Resonance Imaging, for the magnetic signals) and EEG (Electroencephalography, for the electrical ones). Those signals aren't random but are localized in different brain regions and structures. Depending of the stimulation (either exogenous or endogenous), group of neurons produced signal patterns in specific brain regions. By observing those patterns, it is possible to determine which stimulation make them appear. Then, by artificial designing similar stimulation, it is possible to elicit those pattern in order to study them more precisely.

BCI uses the recorded signals from the brain as a source of input from a electrophysiological device. Generally, and as it is the case in this project, BCI uses non-invasive EEG or MEG (Magnetoencephalography) to monitor the brain activity. The subject are presented with various type of stimuli, depending of the objectives of the BCI, which will elicit particular responses in its brain. The obtained raw data then go through multiple process steps, in order to get usable and meaningful data that serve as inputs.

Typically, when brain's signals are recorded during an experiment, the data are analysed afterward, once the experiment is over. This offline analysis is the generic method used to study the brain and its mechanisms in this domain. BCI on another hand, heavily relies on what is called online analysis. This mean that the data are processed during the experiment, allowing the response of the brain to one stimulus to be meaningful for the experiment and even the next stimulus.

Moreover, BCI present feedbacks to the subject allowing him to attest the correctness of the machine. By using this method, it is thus possible to establish a form of communication between the subject and the machine. On one side, the machine present stimuli that can depend on the responses of the subject, and on the subject side (its brain) give responses depending of the presented stimuli.

The work done in this internship was heavily inspired by the work of Zander et al. (2016), as well as the work of Perrin et al. (2012). The objective of this internship was to develop an digital simulation were a virtual cursor had to find a target position in its environment, without prior knowledge, and basing its decision only by taking information from the feedback it received after each of its movement. Those feedbacks were processed EEG data that were transformed to obtain a single value. The difference with other similar BCI was that, here, instead of having a binary response ("yes" or "no"), the feedback should be accompanied by a measure of the certainty, in order to fine tune it. The aim of this simulation was to attest the feasibility of an application that uses the same method, but with real subject. Finally, BCI generally uses visual stimuli displayed on a screen as they are the easier to implement and to access. But any other type of sensory stimuli could potentially be used. This project aim to use auditory stimuli for its BCI, so there will be not visual clues. Thus, this is a feedback BCI, where the cursor will give clues about it movement by emitting sounds, and it will receive feedback from the subject in the form of processed EEG responses.

In the end, such technology, if proved to be efficient enough, could represent an alternative to the other vision-based BCI. The long-term objective of this project is to develop a method that could be used to establish a form of communication with patients in locked-in syndrome, as they are conscious of their environment and have access to auditive clues (albeit at a lesser degree than healthy subjects).

The BCI application that the project aim to develop will put the subject in the centre of a virtual environment in which an artificial agent will generate movements of an auditory cursor and gather information from feedbacks. The objective of the cursor is to reach a target close to the subject face, without prior knowledge. The sound that is used by the auditory cursor is the one of a mosquito, a noxious stimulus that will elicit a specific response from the subject's brain if it comes closer (the subject don't want the mosquito to bite him). So the feedbacks are processed EEG data coming from the subject, and will allow the agent to update its knowledge about the virtual environment. Depending of the brain response, it could determine if it come closer or not from the head.

This internship was mainly composed of a development part and simulation testing. A preliminary part will introduce the concept of electroencephalography, central to this project. Then the main report will be divided into three part, each presenting one of the aspect of the work I have done, and a final part introducing what is being worked on as a continuation of this project.

Preliminary concept

Electroencephalography

An important part of this project revolve around electroencephalography (EEG). This part will allow me to introduce its concepts and methods. EEG is a method allowing the electrophysiological monitoring of the electoral brain activity. EEG are generally non-invasive, with an electrodes net placed on the scalp of the subject, although invasive methods exist and use electrodes implanted directly into the brain.

The brain is composed of an immense amount of neurons, cells capable of creating electrical potential when they are stimulated. The activity patterns of those cells are very complex and variable, depending of the type of stimulation (either exogeneous if coming from the extern environment, or endogenous if coming from the organism), the brain regions, the brain state… And these are these electrical variations (resulting from ionic current), that are captured by the EEG electrodes. A simple example is the motor cortex: a region mainly controlling our motor movement. If we raise our arm, a typical activation pattern will appear. More interestingly, those activation pattern are very robust in the sense that they are identical for every human being.

Thus, if similar conditions are met, the patterns are also similar. This characteristic is the most important, as it will allow to precisely study brain mechanisms. One specific category of brain responses that will be of interest in this project is event-related potential (ERP), a brain response resulting of a specific sensory, cognitive and/or motor event. The response to an event will always present the same pattern.

A classic example is the P300. “P” indicates a positive amplitude wave, and “300” indicates that it appears 300 milliseconds after the stimulation. P300 appears in oddball paradigms, i.e. a low-probability target mixed with high-probability non-target items. For example, if a rare occasion of a high-pitched sound is played in a sequence of frequent low-pitched sounds, this will elicit a P300 that can clearly be observed on EEG data. Multiple other ERPs exist, all associated with specific events. Later in this report, I will talk about feedback-related negativity (FRN), which appear when a perceived feedback is worse than the expected one.

Thanks to this method, it is possible to design experiments to study specific brain behaviour, whether conscious or unconscious, and so acquiring a better understanding of how the brain works. On a side note, apart from cognitive science, psychophysiology and other neuroscience-related research domain, EEG are also widely used is medicine as a tool of diagnosis and therapies. Its main interest compared to other methods such as magnetic resonance imaging (MRI) or positron emission tomography (PET), is its portability and its great temporal resolution.

Finally, to cloture this quick presentation and introduce the rest of the report, EEG has another use since the 1970s, but which field start to develop more since only a decade ago. As explained in the introduction, brain-computer interfaces, or BCI, use EEG to extract data from the brain and transform them into input directed to a machine, which then will interpret it and complete the associated task, effectively removing the needs for intermediate hardware.

Internship mission

The main objective of this internship was to create an digital simulation to attest the feasibility of a real application in the future. The project was decomposed into three parts. The first was to create auditive stimuli that can be perceived at different spatial position around and more or less close to the head using head-related transfer function (HRTF). The second was to construct the framework of the virtual environment in which an artificial agent will generate the movements of a virtual cursor and which will receive feedbacks coming from the EEG processing. The information provided by those feedbacks are then used to update its knowledge of the environment, allowing it to “pin-point” the target it has to reach. In order to test the simulation without directly monitoring real subjects, we used EEG data coming from a previous experiment (Perrin et al., 2012), similar to the expected ones. The third part will describe how those data are used to create a classifier and how they are processed and transformed in order to give a meaningful feedback to the artificial agent.

1. Acoustic spatialization

Acoustic spatialization is the fact of giving the impression that a particular sound comes from a specific spatial position around the head while that sound is in fact only played in a stereo headphone. For that, we need two things: a mono-acoustic sound, and an Head-Related Transfer Function (HRTF) data bank that will be used to transform this sound. The result is a sound composed of two different spectral profiles, each played in one ear, thus giving the illusion of a spatialized sound.

HRTF banks are complex and take a long time to construct. Generally, an artificial head (though it can also be a real head, with the real body that comes with it) is placed on the centre of an anechoic chamber, with microphones in both ears. Then a loudspeaker will play a mono-acoustic sound at each and every positions in the space that are of interest (different azimuth, elevation and distances), multiple times. This is a lengthy process, of at least two hours in the majority of case. Then by comparing the original (mono-acoustic) sound with the two registered sounds (from the two microphones) for a specific position, an HRTF can compute for this position. And this HRTF can then be used to transform another mono-acoustic sound into a stereo-acoustic sound, that, when played in a headphone, will give the impression that it comes from the spatial position the HRTF is associated with.

We wanted to play sounds close to head of the subject (like a mosquito that wants to lands of the face). In that case, as we are way under 100 centimetres from the centre of the head, we talk about near-field HRTF, opposed to the far-field HRTF which can only be used for sound that are over 100 centimetres from the ears. When the source of a sound comes from under this distance, the size and the composition of the head and as well as the torso will alter the spectral profile of the sound wave that is receive by the two ears.

As creating an HRTF bank demands a long time and specific and expensive installations, we choose to take the pre-existing bank composed by J. Arend et al. (2016) available on www.sofaconventions.org (THK/HRIR Near-field HRTFS: http://audiogroup.web.th-koeln.de/ku100nfhrir.html). This HRTF bank was composed using an artificial head as a reference and contain HRTFs for 5 distances (25, 50, 75, 100 and 150cm) with 360 points each (360°), at a single horizontal elevation, for a total of 1800 points.

The other solution was to directly use a loudspeaker that an experimenter will move around the subject himself, but his would have create more auditive stimuli and other unwanted noises that would not have been desired during the EEG monitoring, and so this possibility was quickly excluded.

To give the impression that a sound come from a particular direction, this sound has to meet some requirement. First, it has to be mono-acoustic (one single sound stream, that is identical in both ear), so it can be transformed into a stereo-acoustic one. It also have to have characteristics similar to the sound used for the construction of the database. In this case, the frequency should be of 44100Hz and saved in a 32bit float file for a better quality (not matching those specifications would have result in an unusable, heavily distorted sound).

To transform a mono-acoustic in a stereo-acoustic one, the sound have to be convolved two time, so one time for each ear. For that, we have to use the two Impulse Responses (IR) composing the HRTF (one for each ear). Thus, if an HRTF corresponding to the position at 90° on the right were used for this convolution, the resulting sound will seem to come from that exact direction.

However, simply using the IR of a particular direction isn't enough. To add the impression of distance, the signal have to undergo a last simple modification that is not given by the IR. The inverse square law dictates that each time the distance from the source is doubled, the sound level *roughly* decreases by 6dB. Here, the sounds situated at 25cm are considered as the sources. This mean that the sounds situated at 50cm have their intensity reduced of 6dB. In a same manner, sounds at 75, 100 and 150cm have their intensity reduced by 9, 12 and 15dB respectively. Thus, after being convolved by an IR corresponding to a point at 90° and 50cm, the sound also have to have its intensity reduced by 6dB. After those modifications, the sound will seem to come from the chosen couple of direction and distance. The convolution were done using MATLAB, by following the instructions given on www.sofaconventions.org. The intensity modification were done with Python by using the pydub package.

For this project, I kept the five distances and choose 16 directions from 0° to 360° with steps of 22,5° (rounded down). This gives 80 possible positions evenly divided around the head, with a greater spatial precision the closer to the head (Fig. 2). Of course, it was possible to use more positions, or even all of them, but the trajectories of the cursor to reach the target (see next part) would have been too long. Also, the difference of two sounds separated by just a few degree is barely perceivable for untrained humans.

Finally, the sound had a length of 200 milliseconds (sufficient to elicit an EEG response) and was the one of a mosquito. This choice was due to the fact that a mosquito sound is something noxious, that we actively want to avoid (if a mosquito comes too close, it can bite us). As explained before, the auditory cursor (the "mosquito") have to reach the head at the centre of the environment using feedbacks. And a noxious stimuli eliciting a particular brain response when coming closer, this can be used a the feedback for the cursor.

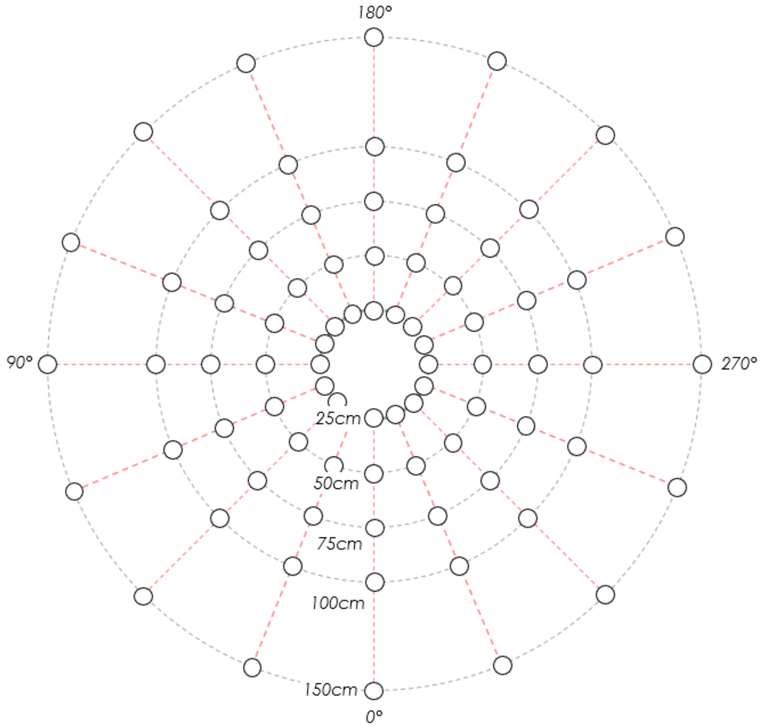


Figure 2: Representation of the auditive map. *Five distances from the centre of the head: 25, 50, 75, 100 and 150cm. Sixteen directions around the head, from 0° to 360° with steps of 22,5°. Eighty possible positions in total. The circular plan is centred on the head and is at eyes height.*

2. Framework environment

The second part was to create an environment in which the "mosquito" will move. This part was heavily inspired by the work of Zander et al. (2016). In their work, they use a visual grid in which a cursor moves without any prior knowledge of its environment. When the cursor make a movement, the subject's brain will produce a response depending of whether the cursor come closer or not from the target. This response is then use by the cursor to know if its movement was correct or not. And depending of this binary response, the probability associated with this particular movement is either reinforced or decreased.

My first prototype for the environment was coded under Java before being ported to Python, the reason being that I had more experience in this language at this moment, and this gave me time to learn Python. In both case, the program was coded using an oriented-object method. This first prototype of the environment was a simple grid represented as a two-dimensional list and with the target being the bottom line. The directions were simple, the cursor could go to the top, bottom, right, left, top-right, top-left, bottom-right and bottom-left (Fig. 3).

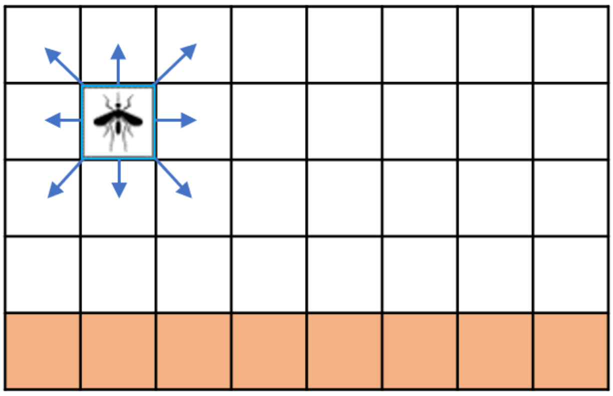


Figure 3: Spatial layout prototype. *The first spatial layout was under the form of a simple two-dimensional grid. The mosquito was able to chose up to eight direction at each position. The left and right sides were connected in such a way that the mosquito can go from the left side to the right and vice-versa. The target was the entire bottom line, and the starting position was a random one from the top line.*

However, this was quickly converted from Java to Python. The improved and final virtual environment is represented in Fig. 4. As explained before, there is 16 different directions, each with 5 distances. This gives a good compromise between the number of possible positions, and the length of the trajectories to reach the target. For each given position, the mosquito could choose up to eight possible directions (five if it was on the inner or outer border).

The simple directions (top, bottom, right, left…), were replaced by angle. Each direction will create an angle in relation to the target (target – original position – destination). So, each position in the spatial layout possesses up to eight angles associated to their respective directions. The angles were computed using simple trigonometry: $((a^2 + b^2 + c^2)/(2ab))$, with a = distance between the actual position and the target ; b = distance between the actual position and the destination ; c = distance between the destination and the target. An angle close to 0° represents a movement toward the target, and inversely the more the grater the angle is.

The mosquito, at the beginning, don't have any information about its environment or the position of the target, nor its own position relatively to them. When on a specific point, it has the choice between 8 directions (5 if on one of the border), each associated with an angle. The mosquito entire knowledge is represented by a list of angle intervals. This list is composed of 12 intervals, from 0° to 180° by step of 15°. Giving the circular nature of the environment, only the absolute value of the angle are used. Moreover, a weight, or a share (Zander et al., 2016), is attributed to each intervals. Those shares are all equal to 100 at the beginning. Finally the probability for an interval to be chosen correspond to its number of share compared to the sum of the total shares. So, at the beginning, each interval have a probability of 1/12 chance to be chosen.

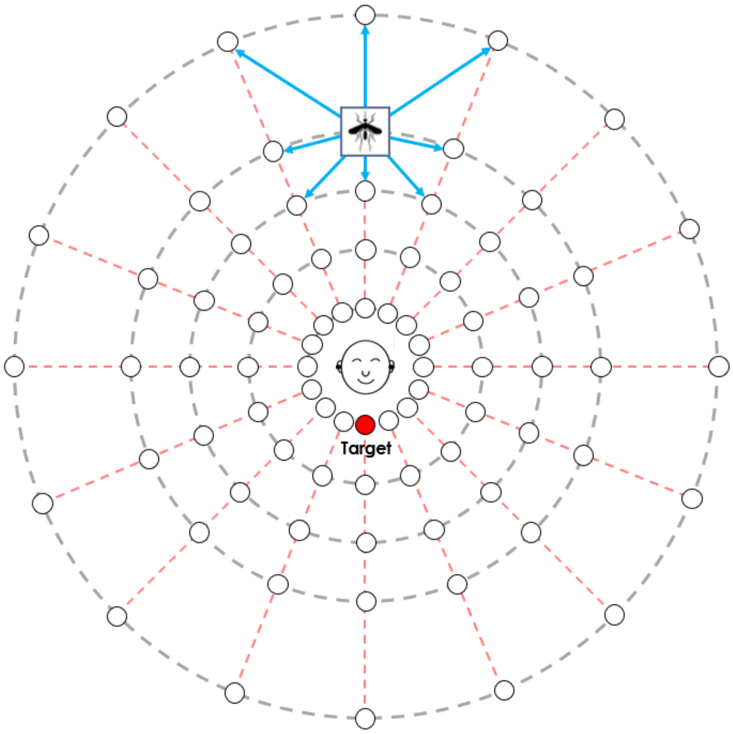


Figure 4: Representation of the spatial layout. *Five distances and sixteen directions around the head, for eighty possible positions in total. The circular plan is centred on the head and is at eyes level. The red circle represents the target that the cursor (the "mosquito") have to reach. Up to eight directions (blue arrows, five if on the inner or outer circle) are possible from each position.*

Now, to choose a direction over another, the mosquito will generate a random float number between 0 and 1, and will run through the interval list. It starts at 0 and adds the probability of each interval until this number is superior to the randomly generated one. Then, from the eight (or five) available angles of its current position, it will choose the one that corresponds to the interval. In the case of no interval corresponds to the chosen one (as there is twelve possibilities), it will start again by generating another random number. The mosquito will then move to its new destination.

After the movement, the mosquito, via the artificial agent, will receive a feedback from the subject (see next part) that it will use to update its probability table (the ones associated to the angle intervals). If the feedback indicates that its movement was correct, then the shares of the chosen interval will be increased, as well as the ones from the four adjacent intervals in the list (Table1). If the feedback is a negative one, the shares of the chosen interval the adjacent ones will be decreased. For example, if the fifth interval was chosen, then the shares of the third, fourth, fifth, sixth and seventh are increased. The degree of increase/decrease is stronger for the chosen interval (100% of the bonus/malus), lower for the close adjacent one (75% of bonus/malus), and even lower for the two farthest adjacent interval (only 50% of the bonus/malus).

Moreover, the intensity of the shares alteration is fine-tuned by the certainty of the feedback: if totally certain, the alteration intensity is maximum (100%), and if totally uncertain, the alteration intensity is null (0%).

In short, the following formula is used to compute the shares modification: $S_{new} = S_{actual} + (S_{actual} * M * C)$, with S = Share, M = Multiplier (1, 0.75 and 0.5 for a favourable feedback ; 0.5, 0.375 and 0.25 for an unfavourable feedback) and C = Certainty (between 0 for totally uncertain and 1 or -1 for totally certain or uncertain). This way, a movement considered as correct see its shares multiplied by two, and divided by two if it is incorrect (and supposing the certainty is maximum).

Interval	Probabilities	Shares
0° – 15°	8.3%	100
15° – 30°	8.3%	100
30° – 45°	8.3%	100
45° – 60°	8.3%	100
60° – 75°	**8.3%**	**100**
75° – 90°	8.3%	100
90° – 105°	8.3%	100
105° – 120°	8.3%	100
120° – 135°	8.3%	100
135° – 150°	8.3%	100
150° – 165°	8.3%	100
165° – 180°	8.3%	100

Interval	Probabilities	Shares
0° – 15°	6.45%	100
15° – 30°	6.45%	100
30° – 45°	9.67%	150
45° – 60°	11.29%	175
60° – 75°	**12.9%**	**200**
75° – 90°	11.29%	175
90° – 105°	9.67%	150
105° – 120°	6.45%	100
120° – 135°	6.45%	100
135° – 150°	6.45%	100
150° – 165°	6.45%	100
165° – 180°	6.45%	100

Interval	Probabilities	Shares
0° – 15°	9.75%	100
15° – 30°	9.75%	100
30° – 45°	7.31%	75
45° – 60°	6.09%	62.5
60° – 75°	**4.87%**	**50**
75° – 90°	6.09%	62.5
90° – 105°	7.31%	75
105° – 120°	9.75%	100
120° – 135°	9.75%	100
135° – 150°	9.75%	100
150° – 165°	9.75%	100
165° – 180°	9.75%	100

Table 1: Example of a probability update after a movement. Here you can see the way of the probabilities evolves after a mosquito movement. For this example, the movement corresponded to an angle from the 60°-75° interval, and the certainty is equal to one. Left: Shares and probabilities at the beginning of the experiment, when the cursor don't have any information. Centre: Shares and probabilities increasing after the first move if it was considered as correct. Right: Shares and probabilities decreasing after the first move if it was considered as incorrect. The share are directly modified by the formulae $S_{new} = S_{actual} + (S_{actual} * M * C)$.

Finally, by this method of trials and errors, the mosquito can pin-point the position of the target. The probabilities will evolve in accordance to the feedbacks, leading to choices that are more and more probable to lead to correct movements, that is, toward the head. And this until the mosquito reach the target.

3. Electroencephalography

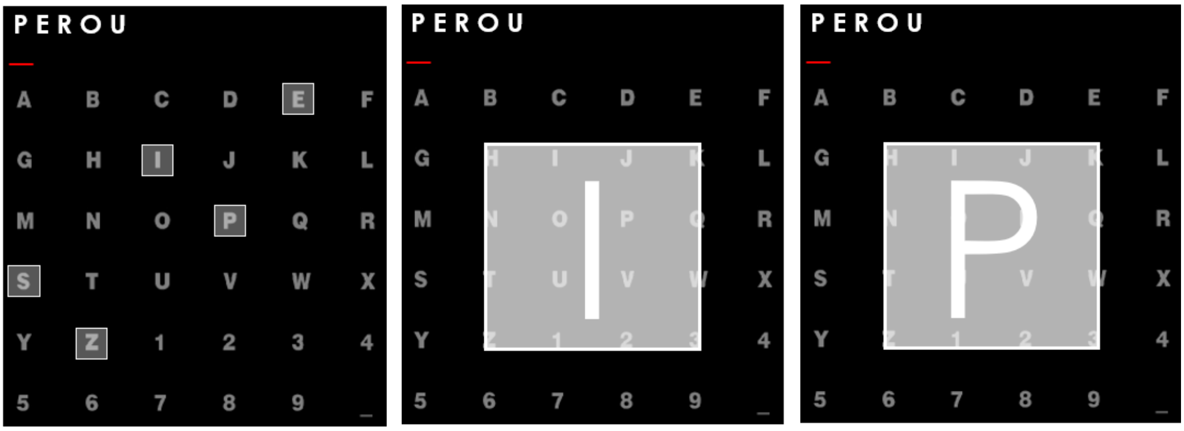
Finally, the last part was to add simulated Electroencephalography (EEG) data feedbacks to the "mosquito". Those EEG data are transformed into a single value: the Certainty, as enounced previously. The sign of this value indicates if the movement is correct (toward the head) or incorrect (away from the head), and its absolute value (between 0 and 1) indicates if the feedback is more or less certain.

For this purpose, I used an EEG dataset coming from a previous experiment (Perrin et al., 2012). Participants wore an EEG headset, and their task was to write words by using a grid containing thirty-six characters, displayed on a screen (Fig.4). All the words were composed of five letters, and the experiment was divided into five blocks: the first four contained twelve words each (sixty letters each), and the fifth contained twenty words (one hundred letters). To write a word, they were tasked to look at the letter of interest, and focus his attention on it without doing anything else. Then, groups of characters of the grid were randomly flashed (with each character being part of different group). When the fixed character is flashed, we talk about a target stimulus, and when not flashed, this is a non-target stimulus. The target stimulus will elicit an event-related potential (ERP) in the subject's brain. This particular event is called a P300, as it is a positive spike that happen 300ms after the flash. From the totality of the resulting EEG data, only a portion of the signal is processed, that is the part following the flash. A part of a signal associated with a precise time (the apparition of a stimulus) is what's called an epoch. Then each epoch is compared to a template to determine whether it correspond to a target response or a non-target response. The template comes from a calibration phase (the first four blocks, with a total of forty-eight words), and the epochs are processed online, that is, during the experiment.

Then it will compare the data obtained by each of the character groups. As the group shares some characters, the online process can isolate the groups where a P300 spike appears. It is possible that multiple characters are associated to a P300 spike (most notably the character adjacent to the fixed one). The online process will then create a list ordering the characters based on the associated brain response: the first letter has the highest probability to be the one fixed by the participant, the second character in the list is the second most probable…

At this moment, the software will present the first letter of this list to the subject. In the 750ms following this feedback, the subject brain can elicit another ERP. If the presented letter is not the expected one (it is a "bad feedback), the ERP should appear on the epoch following the feedback (Miltner et al., 1997). This ERP is called feedback-related negativity (or FRN). If the letter is the expected one (it is a "good feedback"), the FRN should not appear. The epochs are processed online and a classifier will determine if the feedback perceived by the subject is considered as a good or a bad feedback. This ERP processing indicates if the presented letter is the correct one or not, allowing a correction in the latter case (Fig. 5).Finally, the experiment was divided into two main blocks. In the first block, the subject have to write word as explained previously, by fixing the letters of interest. This block was composed of forty-eight words of five letters each. The epochs following the feedback (presentation of the letter), that is the epoch characteristics to the presence or absence of the FRN were used to train a classifier. This classifier is then use in the second block, where the responses to the feedback are used to apply a correction.

A.



B.

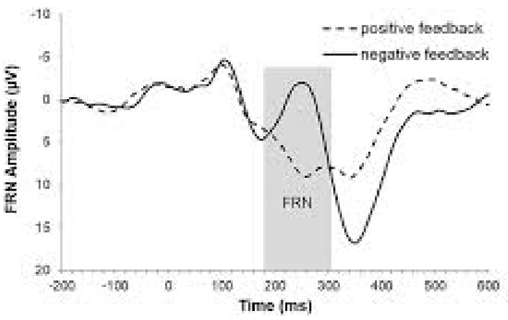


Figure 5: Character grid displayed during the experiment. A: Left: Random character group flashing during which P300 will be used to determine which character is fixed. Centre: Bad feedback presentation, FRN should be present. Right: Good feedback presentation, FRN should not be present. B: Example of the presence (full line) or absence (dotted line) of a FRN spike in the 750ms following the presentation of a stimulus (feedback). For more information, see Perrin et al., 2012.

A important point to keep in mind for this experiment is that the classifier is not infallible. For example, the noise present in the processed signals or the subject state (fatigue, attention…), can lead to bad quality epochs. This means that for a given epoch, the classifier could categorize a FRN response in the no-FRN category and inversely. Thus giving a false classification that could change a correct letter or keep an incorrect one.

Now, the data from this experiment are the one I used to train my own classifier and simulate the EEG inputs and mosquito's feedback. The classifier code was written by Emmanuel MABY, and lightly adapted by myself to correspond to my need.

The process of those data was composed of multiple steps. First, the simulated feedback represents the virtual participant brain response when the mosquito moves. As the mosquito represents something noxious, if it comes closer to the head, a FRN should appear and thus can be detect by the classifier, and if it goes away, there should not be a FRN present. However, as the classifier isn't infallible, a FRN response could be categorized as no-FRN, and inversely. Thus, the FRN being the component of interest, only the data corresponding to 750ms following a letter presentation (the time were a FRN could potentially be present) were analysed.

During the experiment, each event (character flashing, feedback presentation…) was associated with a number. The first step was to only keep the events corresponding to the feedback presentation (events 14 and 15 in this case), with a total of 60 events by subjects. Then, the data are filterer by using a 1/20 low/high-pass frequency filter, meaning that frequencies outside that rang were rejected.

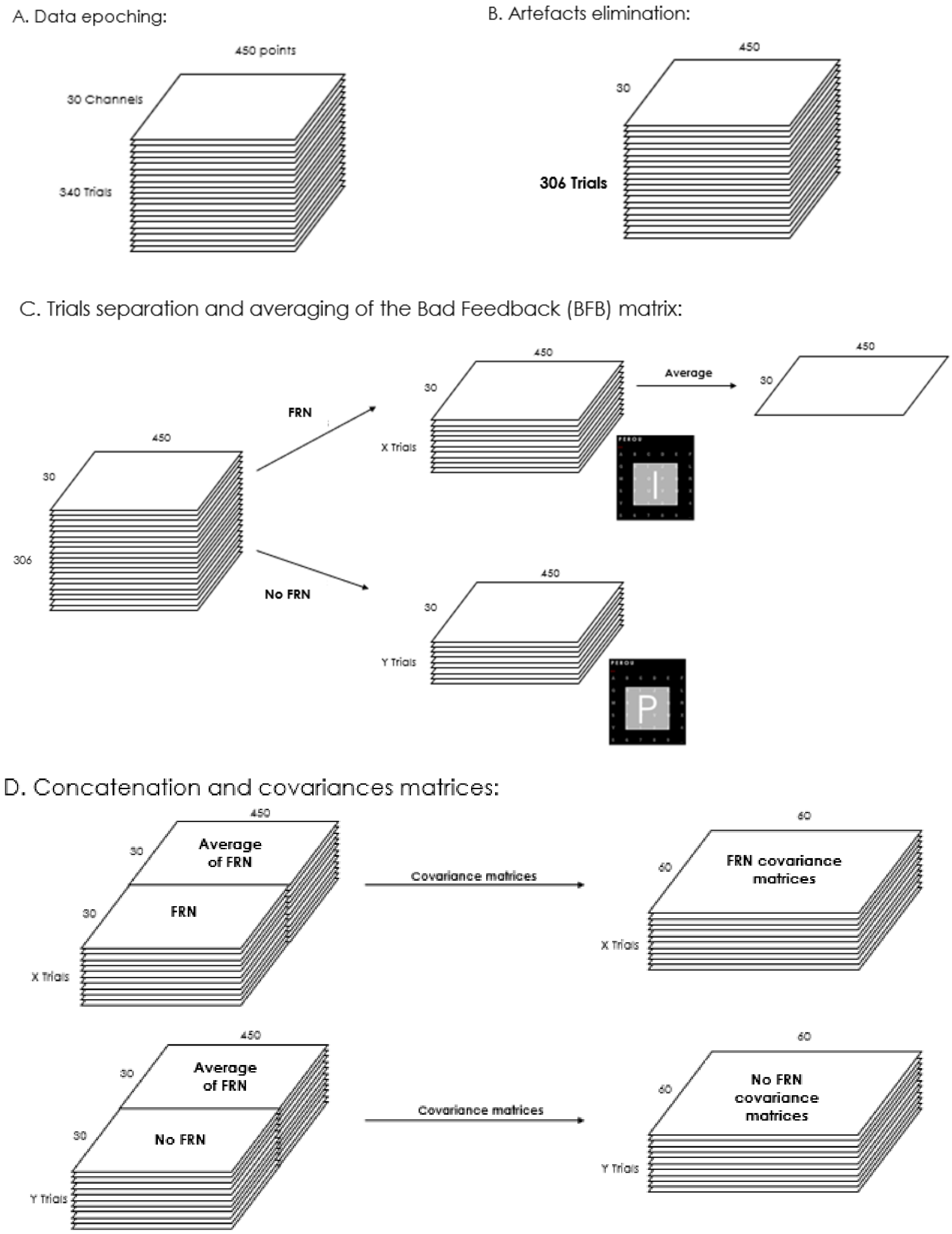
Second, only the 750ms following the events were kept: this is what is called data epoching (Fig. 6 A), with an epoch being an event timestamp associated to a filtered data. As the EEG headset was composed of 30 registering electrodes (channels), and the capture frequency being of 600Hz (so there is 450 points in 750ms), each epoch (i.e. an event, a letter presentation) was a matrix of dimension 30 * 450. Also, each subject having 340 trials (four blocks containing twelve words and one block with twenty words, each words being five letters long), the data of one subject was contained in a matrix of dimensions 340 * 30 * 450. From this matrix, the epochs that are higher than 90% of the maximum value (artefacts) are eliminated, so the dimension is now 306 * 30 * 450 for each subjects (Fig. 6 B).

In the next step, this matrix is separated into two smaller matrices: one containing the "good" feedback (no FRN present), and the other containing "bad" feedback (FRN present): the two matrices are of dimensions X * 30 * 450, with X being the number of corresponding trials (variable for each subjects). Once the matrices separated in two, one called "noFRN", the other, "FRN", the "FRN" matrix was averaged (resulting dimension: 30 * 450) (Fig. 6 C). This average matrix was then concatenated to each epoch of both BFB and GFB matrices, now of dimension X * 60 * 450 (Fig. 6 D).

Both matrices will next be transformed in two covariance matrices of dimension X * 60 * 60. And those covariance matrices will undergo a Riemann averaging, creating two average covariance matrices of dimension 60 * 60 (Fig. 6 E). Those two average covariance matrices, "noFRN" and "FRN", are what constitute the parameters of our classifier.

Two different classifiers were constructed using this method. Both are made using a leave-one-out paradigm, which mean that part of the data is used for the setting of the classifiers and the other part is used for the testing. The two classifiers were either inter-subjects or intra-subjects.

For the first one, all subject expect one are used to construct a classifier that was tested on the one put aside. For the intra-subject classifiers, the four first blocks of one subject are used for the construction, and the fifth for testing. As there was 22 subjects, this mean that there was 22 inter-subject classifier, and 22 intra-subject classifiers.



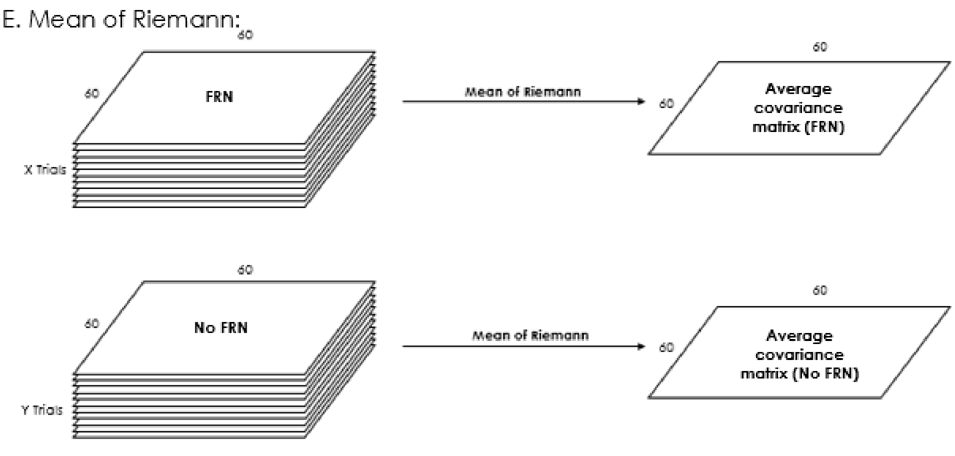


Figure 6: Classifier construction steps. A: Data epoching: only the data in the following 750ms (450 points for a 600Hz frequency) after the stimulus presentation are kept as the FRN spike appear during this interval. B: Artefacts elimination: the trial higher than the 90% of the maximum value are eliminated. C: Trial separation: the trial presenting FRN or not are separated. The matrix containing the FRN is averaged. D: Concatenation of the FRN average to each epochs and computation of the corresponding covariance matrices. E: Average covariance matrices after the two group of covariance matrices undergone an averaging of Riemann.

Two datasets were available for this part, but both were coming from similar experiment. The difference is that one of them contained training blocks with a controlled rate of errors (as explained previously), while the other didn't. The first one was the one used as it gives a constant rate of error. In the second set, some subjects presented only FRN, or not at all, which would have biased the classifiers.

Once the classifiers created, it's time to use the leaved-out data to simulate the feedbacks. These data are processed in the same way as previously explained to obtain covariance matrices of dimension X * 60 * 60, with X being the totality of the trials of the subject (340 trials for the inter-subject classifier) or of the last block (100 trials for the intra-subject classifier). One by one, each trial will be compared to the two average covariance matrices of the classifier ("noFRN" and "FRN"). This comparison will give two distances of Riemann for each trial: d_{FRN} and d_{noFRN}. The higher the distance, the farther the trial is from the given matrix, and so is different.

With the two distances, we had the choice to compute two different values. The first one was the log ratio. The log ratio of this distances ($\log(d_{FRN}/d_{noFRN})$) will indicate from which of the two classifier's matrices the trial is closer to. If it's positive, it's closer to the GFB matrix, and if it's negative, it's closer to the BFB one. But this value don't indicate if the trial is very close or very far from one or the other matrix. It just indicates the relative proximity. The second value is obtained using the following formula: $((d_{FRN} - d_{noFRN})/(d_{FRN} + d_{noFRN}))$. There again, the sign indicates from which it is closer. However, here, the closer the value is to 1 (or -1), the closer it is from one or the other matrix. And this fact indicates that this value can be use as a measure of the certainty of the classification. Thus, the more certain (either 1 or -1), the more probable a FRN is present or not.

The trials were then separated into two groups: the first contain the trials considered as "correct", and the second, the trials considered as "incorrect".

Important note: in the simulation, the presence of a FRN indicate a "correct" movement for the mosquito (it comes closer to the head), but this is a bad thing for the subject. While in the experiment from which the data comes, a "correct" feedback is a good thing for the subject.

Finally, another solution was proposed, but was rejected as it wasn't differentiating true and false positive and true and false negative. This solution used the log ratio values and sorted them in two lists: either positive or negative. Box-plot were then computed for the two lists. During the simulation, when a random trial were selected, its log ratio were compared to the corresponding box plot (negative or positive). It the log ratio were higher than the third quartile (for the positive plot), or lower than the first quartile (for the negative plot), then the certainty was equal to one. If it was between those values and 0, the certainty were between 0 and 1, using the following formula: X/Max, with X = log ratio ; Max = third or first quartile (Fig. 7).

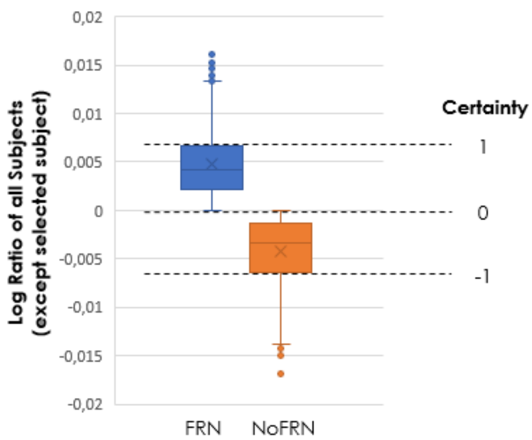


Figure 7: Rejected certainty computation design. The randomly selected log ratio were compared to the log ratio of all other subject to compute a level of certainty.

4. Final simulation

Fig. 8 represents the loop the application do during the simulation.

Before the beginning, one subject is randomly selected, as well as it associated classifier (the one from which he was excluded for the inter-subject condition, on the one containing its four first block for the intra-subject condition). The mosquito starts at a random position in the environment, then it makes a random movement. The angle formed by the target, the actual position and the destination will dictate which feedback it will receive. If the angle is inferior to 60°, a trial coming from the "correct" feedback is randomly selected (expected FRN with a chance of no FRN). If the angle is superior to 60°, it's a trial from the "incorrect" feedbacks that is randomly selected (no FRN expected, with a chance of FRN). The level of Certainty is computed by using the two distances of Riemann of the selected trial in the following formula: $((d_{FRN} - d_{noFRN})/(d_{FRN} + d_{noFRN}))$. This value is then used to adjust the weight of the interval to which the angle is part of (positively or negatively depending of the sign), thanks to the following formula: $S_{new} = S_{actual} + (S_{actual} * M * C)$. Then the mosquito makes another movement based on the new probability table, and so on until it reach the target.

On top of that, in the real condition, subject needs a feedback too. That is where the sound localization methods enter in action. Each position is associated with a specific sound that has been modified with HRTF as explained in the first part. At the beginning, the mosquito plays a sound at its starting position, and after each of its movement. This auditory clue is perceived by the participant. As the subject is equipped with an EEG headset monitoring its brain activity, real EEG data are sent to the application which will process them and extract a certainty value from the epoch following the auditory feedback. The rest doesn't change as the mosquito use this feedback to update its probability table.

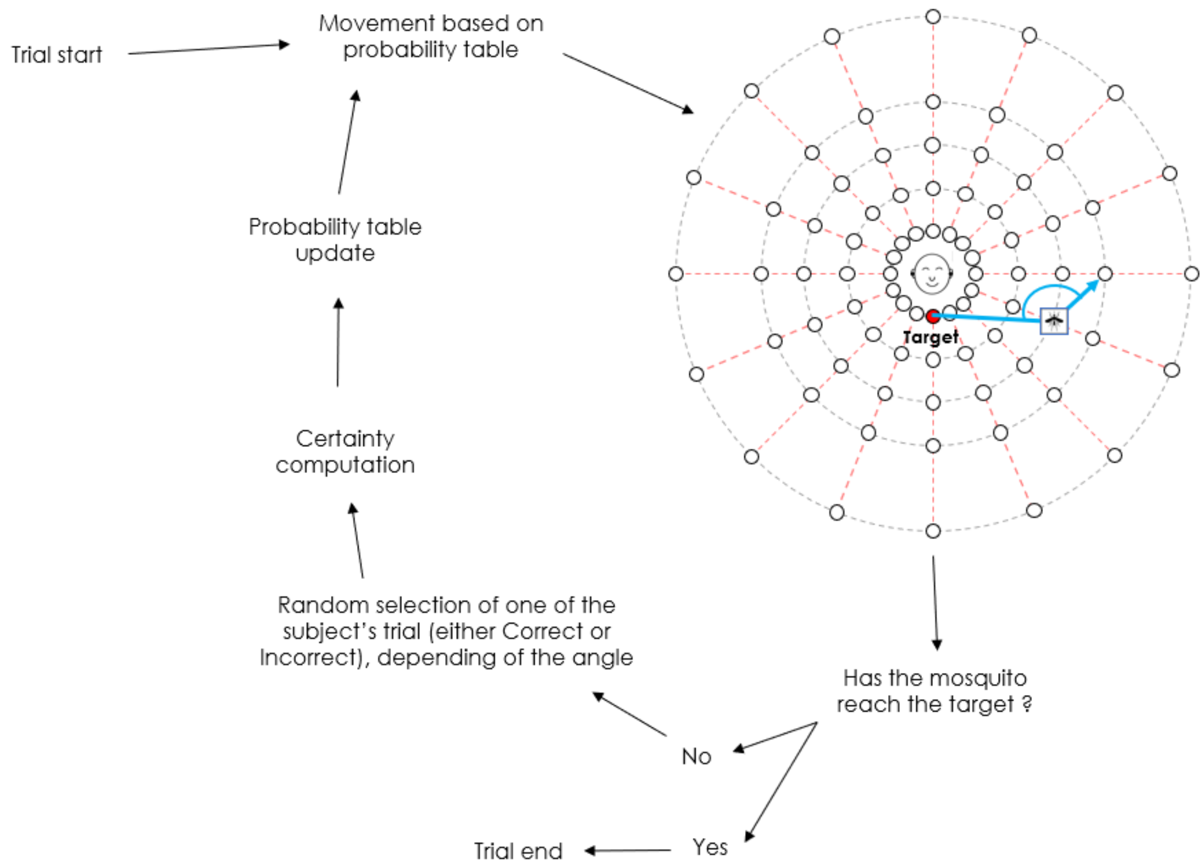


Figure 8: Application loop. The selected trial depend of the angle. If inferior to 60°, a "correct" feedback is sent (FRN expected), and if superior, an "incorrect" feedback is sent (no FRN expected).

5. Qualitative study and real application

The simulation is just the first step toward the application of this method with real subjects. And as this report is being written, a qualitative study is being constructed in this perspective. The aim of this short study is to attest the good functioning of the application. More specifically, the sound spatialization using HRTF (moreover HRTF coming from artificial head) aren't always correctly perceived. Some participants could be very efficient while other potentially couldn't tell were the sound seems to be played.

This study would be composed of two blocks. The first will aim to see if the participants can detect from which direction the mosquito comes from. The second aim to see if participants can detect when the mosquito is about to reach its target (the front of the head), that, if they are able to detect if the sound seems to be played in their close peri-personal space. In the two blocks, for the sake of simplicity, only the front environment is used (Fig. 9, with 0° being the right ear, 180° the right ear, and 90° in front of the head).

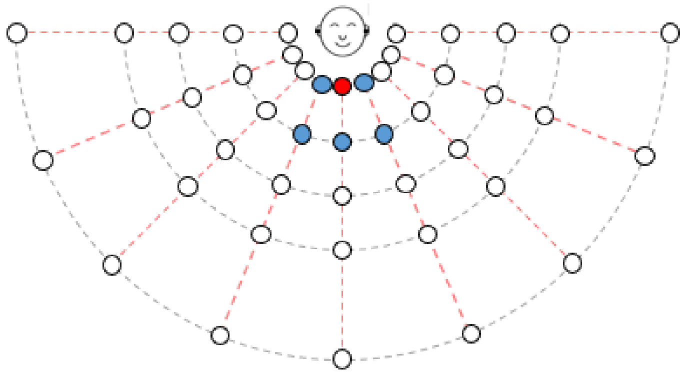


Figure 9: Environment used for the qualitative study. Five distances, nine directions, nine starting positions. The red circle represents the target aimed by the mosquito. The blue circles represents the peri-personal space of the subject for the second block.

The first block will ask the participant to indicate the path borrowed by the mosquito. For that, he will be able to give its response by clicking on an similar image to the one in Fig. 8 (without the blue circle).

In the second block, the subject won't have access to a visual representation of the environment. Instead he will have to give an input (push a button) when he thinks the mosquito is in its peri-personnel space (the blue circles on Fig. 8). The block will contain training trials at the beginning, were the subject won't have to give input. A sound will be played at the moment the mosquito enter the peri-personal space. During the testing trials, the subject will have to give input and will receive feedback depending if he gives a correct answer (the mosquito is on the blue circles), or an incorrect one (input too soon or too late).

The stimuli will be 200 milliseconds long, with an interval of 2 seconds between them. The experiment should last about 30 minutes or more, so pause will be proposed each 20 minutes. In half of the trials, the mosquito will directly goes to the target, either by using the most direct path, or by using a path with one or two more jumps. Each starting positions will be chosen the same number of time, in a random fashion.

Each trial will be characterized by its starting position, its path (direct or not) and by the angle formed by the target – head – starting position.

In the end, this study should give us insight on whether the sound movements are correctly perceived or not by the subjects. In the case it is not, the collected variables will indicate which characteristics isn't well perceived (the direction, the proximity, the type of movement…). Furthermore, participants will be asked to fill out a questionnaire in which they will be able to give their impression about the task and its difficulty, as well as what they think about the sounds and the way it moves in the environment around them.

Conclusion

This internship had the development of an digital simulation as a main objective. This could be divided into three parts, each with its role in the final product. The sound spatialization allows the positioning and movement of an auditory stimulus in the space around the head of the subject. The virtual environment saw a lot of modification during the course of its construction. Most notably, the way the mosquito learns how it was organized (feedback, probability and certainty computation…) changed the most from the beginning to the end. Then, the third part explained which EEG data were selected and processed, and how they were used to train a classifier. This classifier were then used to categorize the remaining trials in order to have a meaningful information used as feedback, which in turn is used as a variable in the formula for the updating of the probability table.

In the end, two main issues remain. First, extracting meaningful value from EEG data, especially for the computation of certainty, is challenging, and the global efficiency isn't up to our expectation. Second, the sound spatialization and the perception of it, though this is being worked on as this report is written.

In any case, an auditory-based BCI could not only extends the possibilities of BCI technology, but could also show whether locked-in patients still have the capacities to interact with their environment. And if proven true, this method could be used to re-establish a form of communication with these patients, and even push toward potential new treatments. In a more general point of view, developing those methods could lead to faster and more efficient interactions with machines. But this will also lead to a new vision of our technology, potentially creating new usage or concepts that are still unforeseen today, much like the evolution of computer between their invention and today. Thus, the domains of application are theoretically only limited by our own imagination. Entertainment, communication, vehicle piloting, artistic are some examples, but the most studied today is certainly the medical one. This technology could gave back people their lost members or senses, helps to retrieve the usage of them, or even improves them. Inserting it in our daily life is also a great challenge, asking to rethink how we interact with our environment through our technology.

Finally, to conclude on a personal note, this internship teach me a lot in the domain of BCI and the possibilities it gives. This topic was of a great interest to me for many years, and I am glad to have learnt so much in so little time. And I think it would have been possible, or at least very difficult, without the computer science background that I gain during this year.

References

Arend JM, Neidhart A, Pörschmann C (2016). Measurement and perceptual evaluation of a spherical near-field HRTF set. *VDT International Convention* 29: 356-363.

Hochberg LR, Bacher D, Jarosiewicz B, Masse NY, Simeral JD, Vogel J, Haddadin S, Liu J, Cash SS, van der Smagt P, Donoghue JP (2012). Reach and grasp by people with tetraplegia using a neutrally controlled robotic arm. *Nature* 485: 372-375.

Miltner WHR, Braun CH, Coles MGH (1997). Event-related brain potentials following incorrect feedback in a time-estimation task: evidence for a "generic" neural system for error detection. *Journal of Cognitive Neuroscience* 9: 788-798.

Perrin M, Maby E, Daligault S, Bertrand O, Mattout J (2012). Objective and subjective evaluation of online error correction during P300-based spelling. *Advances in Human-Computer Interactions* 6.

Vidal JJ (1973). Toward direct brain-computer communication. *Annual Review of Biophysics and Bioengineering* 2: 187-180.

Zander TO, Krol LR, Birbaumer NP, Gramann K (2016). Neuroadaptative technology enables implicit cursor control based on medial prefrontal cortex activity. *PNAS* 113 (52): 14898-14903.