UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

DYNAMIC M-CLASS ANGLE COMPRESSION FOR CURSOR CONTROL USING BRAIN INTERFACES

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Pere Garau Burguera

In partial fulfilment

of the requirements for the degree in

CITTEL ENGINEERING

Advisor: Deniz Erdogmus

Xavier Giró Nieto

Barcelona, October 2017

Abstract

The purpose of this project is to present an environment where a cursor can be controlled on a 2D screen by using a brain computer interface (BCI). For this purpose, and by using a probability model that determines which regions are the ones the user will eventually want to move to, and after applying information theory theorems which lead us to the desired result, the interface lets the user choose among m compressed versions of the angle at any particular point, dividing the space into m circular sectors around the current position, in an optimal way. Dynamic updates, which take into account the EEG evidence at the moment the subject selects where they want to go, are performed, in order to make the cursor get to the destination using less selection steps.

Resum

L’objectiu d’aquest projecte es presentar un entorn on és possible poder controlar un cursor damunt una pantalla bidimensional a través d’una interfície ordinador-cervell (BCI). Per aquest propòsit, i mitjançant un model probabilístic que determina a quines regions l’usuari tendrà intenció de moure’s, després d’aplicar teoremes de teoria de l’informació, la interfície deixa triar entre m versions comprimides de l’angle, a cada punt de la pantalla, dividint així l’espai entre m sectors circulars al voltant de la posició en cada moment, tot això de manera òptima. Es realitzen actualitzacions dinàmiques per tal de fer arribar el cursor al punt desitjat en el menor nombre de passes, amb l’ajuda de la informació del senyal EEG just en el moment en què l’usuari fa la selecció del sector al qual es vol moure.

Resumen

El objetivo de este proyecto es presentar un entorno donde es posible poder controlar un cursor en una pantalla bidimensional a través de una interfaz ordenador-cerebro (BCI). Para este propósito, y mediante un modelo probabilístico que determina a qué regiones el usuario va a tener intención de moverse, y después de aplicar teoremas de teoría de la información, la interfaz deja elegir entre m versiones comprimidas del ángulo, a cada punto de la pantalla, dividiendo así el espacio en m sectores circulares alrededor de la posición en cada momento, de manera óptima. Se realizan actualizaciones dinámicas para hacer llegar el cursor al punto deseado en el menor número de pasos, con la ayuda de la información de la señal EEG al momento en el que el usuario hace la selección del sector al cual quiere moverse.

Acknowledgements

I would like to thank Deniz and Xavier, and the whole CSL team at Northeastern.

Revision history and approval record

Revision	Date	Purpose
0	29/09/2016	Document creation

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Pere Garau Burguera	pere_gb@hotmail.com
Deniz Erdogmus	erdogmus@ece.neu.edu
Xavier Giró Nieto	Xavier.giro@upc.edu

Written by: Pere Garau Burguera		Reviewed and approved by:	
Date	29/09/2016	Date	
Name		Name	
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

1. Background

Controlling a computer mouse has become nowadays a task most people do on a daily basis. Therefore, offering the possibility of controlling a computer mouse (by which one is able to use most of the functions of a basic computer) to people with motor disabilities who cannot utilize a computer normally, could be an important change into improving their quality of life.

One solution which has been already used and tested in many different studies and situations is BCI. It stands for Brain Computer Interface, and allows people with mobility impairments interact with a computer and potentially control, with more or fewer functionalities, devices such as wheelchairs, personal computers, lighting control in a room… BCIs record brain activity by means of EEG, non-intrusively, by collecting the electric activity on the scalp of a person.

In the following chapters, a form of controlling a computer mouse will be presented, and in the future it will be able to be easily operated with a BCI, therefore giving people with mobility limitations an improvement in their day to day life.

2. Introduction

First of all, let us briefly present a few examples on how a cursor could be controlled (or, more specifically, being able to move around and select a specific area on a 2D screen).

The distinction of what is a cursor and what cannot be considered a cursor can be brought up. However, as far as this thesis is concerned, a cursor is considered to be any system that lets subjects select and interact with a given desired target on a 2D screen.

At this point, let us suppose that, by using a brain computer interface (BCI), the subject is able to perform the tasks required to control (for instance, all tasks or actions needed to do so in the examples below) a cursor and thus selecting and performing an action on a button/menu or any other interactive object or area of the screen.

- 2D compression

  This approach consists in, given a 2D screen, partition it (compress it) into m disjoint parts. After this, the user selects the partition where the desired destination is located. After the selection, the partitions are updated in a manner such that the subject is able to reach the desired destination in the minimum number of steps possible, finishing when the system is certain of the object the subject desires to interact with.

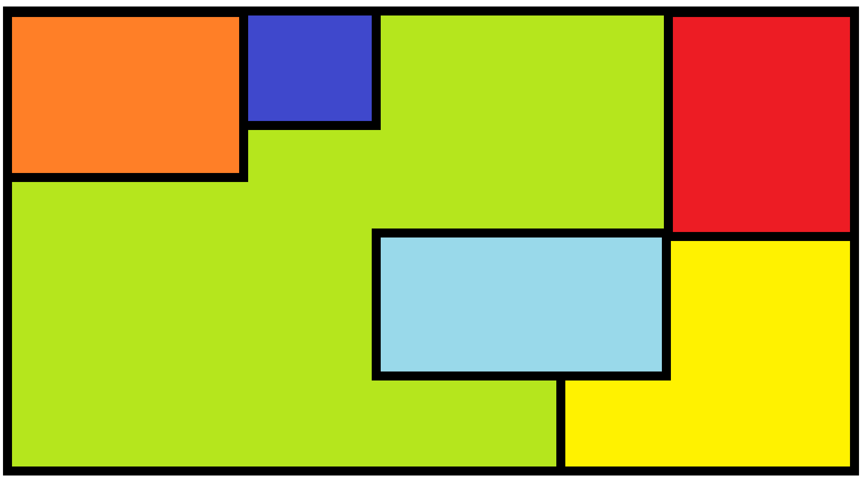

Figure 1: 2D compression.

- 4 directions (up, down, left, right)

  The cursor can be moved in any of the 2 dimensions. Then, the subject has four choices and the destination is reached by successively selecting any of these four directions. A slightly more advanced system would be rotating the perpendicular axes in a way such that it is able to reach the desired destination quicker, given the current position of the cursor and the different positions of the objects the user can interact with.

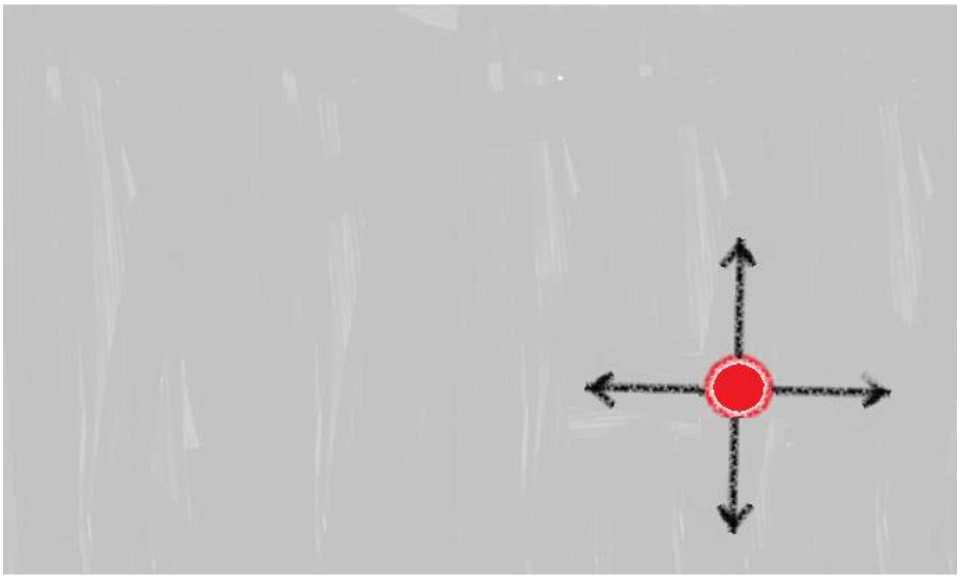

Figure 2: 4 directions approach.

- Stop the bar

In this approach, a bar which scans the screen in a given dimension is shown on the screen. The user focuses on the desired destination, and at the moment the bar scan passes by it, the user stops the bar, and then the scan is performed in the orthogonal direction relative to the previous scan. The user stops the bar at the desired point again, and thus the area where both bars cross is the intended area. An additional improvement would be letting the user choose the directions of the scan, or choose them in a way that allows the user to select their intended destination faster.

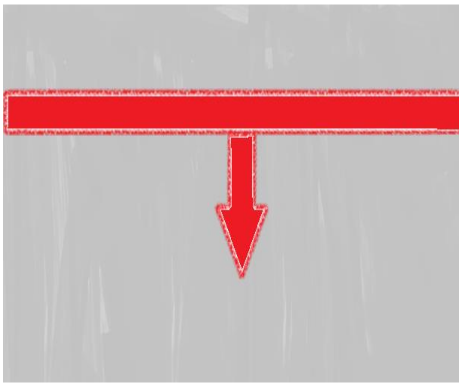

Figure 3: First step, vertical scan.

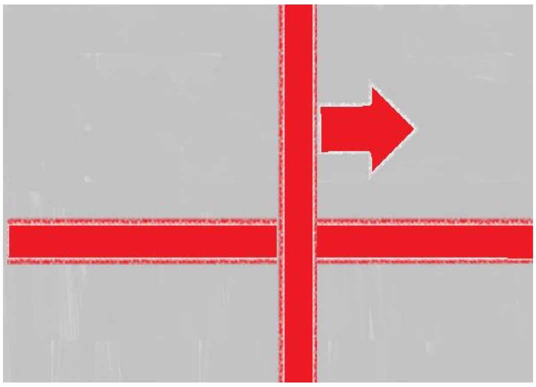

Figure 4: Second step, horizontal scan.

- Angle compression

This case is similar to 2D compression. In this case, however, instead of partitioning the screen into m different sections, it is the angle which is compressed, so given the current position of the cursor, m sectors are presented to the subject. The subject then selects the sector he/she wants to move to, and the cursor makes a move towards a direction inside the sector. After each move, the way partitions are calculated will get the user to reach the destination in the fewer number of steps possible.

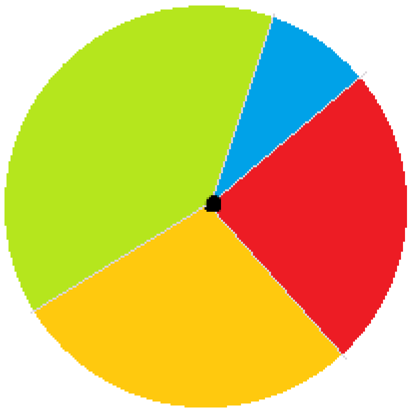

Figure 5: Angle compression.

For this thesis we will focus on how to, given a screen which consists of areas of particular interest, such as buttons, menus,… and other areas where no action can be done, calculate the optimal angle compression such that the subject's wished destination can be reached in the least amount of successive steps possible, from any starting position of the cursor. In the following chapters, a step by step explanation on how to solve the problem will be presented.

3. State of the art of the technology used or applied in this thesis:

A background, comprehensive review of the literature is required. This is known as the Review of Literature and should include relevant, recent research that has been done on the subject matter.

No project that uses angle compression as a paradigm to move a cursor has been found. However, there exist different approaches on the topic, for instance in [10] they use the 4 directions approach, similar to figure 2.

In [11] they use an interesting P300 and SSVEP combination to control both the speed and the direction of the cursor movements.

While there are many different examples and variations of SSVEP being used to control cursors, angle compression, as explained in chapter 2 (figure 5) is perhaps slightly less intuitive or less simple at a first impression, but at the same time it provides more flexibility and it could be easily expanded to other applications.

Moreover, the literature reviewed focuses more on the technical aspects of EEG data collection. In this thesis, the focus is more on the probabilistic part of the problem, in other words how to partition the space so that a desired point can be reached in the minimum number of steps. In some examples there is no probability involved, which makes it easier to use, but at the cost of reducing the great variability and adaptability to both the user and the application that probability introduces.

4. Methodology / project development:

4.1. Probability model

In order to make the optimal angle partition, we first need to define a model that shows where the areas the subject is more likely willing to move towards are. For this purpose, it is reasonable to assume that the subject will only desire to move towards any of the areas which contain objects on which an action can be performed (e.g. clicking a button).

In particular, a probability density function in 2 dimensions will be defined so that it matches with the premise that the more probable areas are the ones the subject's intended destinations are going to be located.

This purpose is accomplished by modeling our probability distribution with a Gaussian mixture function, as follows.

$$\mathrm{f}_o(\mathrm{x}_1,\mathrm{x}_2) \sim \sum_i w_i \mathcal{N}(\mu_i, \Sigma_i)$$

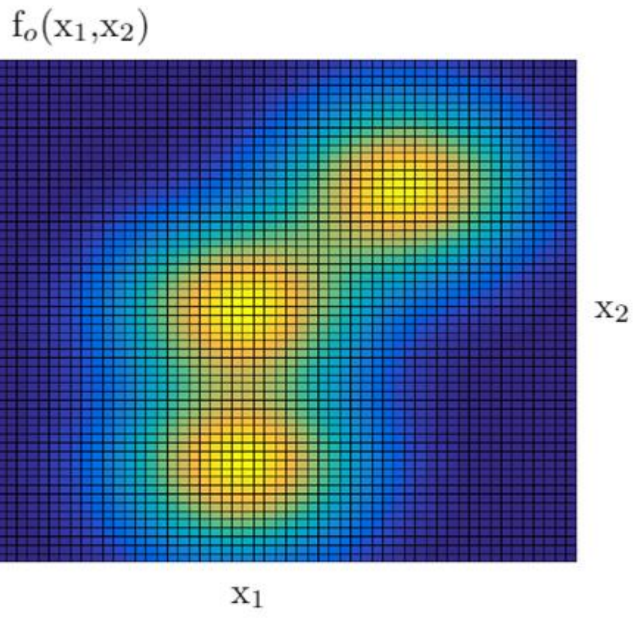

Figure 6: Probability density function in the 2 dimensions.

In figure 6, we can observe three peaks (yellow/red), which have higher probability, whereas the blue areas have a lower probability. Thus, the subject will most likely want to move towards one of these three areas with a higher probability.

After defining a probability density function in two dimensions, and since we are interested in angle compression given the current position of the cursor, a probability density function with respect to the angle at a certain position on the screen is needed. After performing these operations, we will have a probability density function that varies with the angle, calculated around the point x0 on the screen.

$$\mathrm{f}_{x0} = f(\overrightarrow{x} + \overrightarrow{x_0})$$
$$\mathrm{f}_{x0}(r,\theta) = f_{x0}(r\cos\theta, r\sin\theta)$$
$$\mathrm{P}_{x0}(\theta) = \frac{\int_0^{\infty} f_{x0}(r,\theta)\, r\, \mathrm{d}r}{\int_0^{\infty}\int_0^{2\pi} f_{x0}(r,\theta)\, \mathrm{d}\theta \mathrm{d}r}$$

4.2. <u>Description of the problem</u>

At this point, given a probability density function with respect to the angle at a particular point on the two-dimensional space, the optimal partition of such angle must be found.

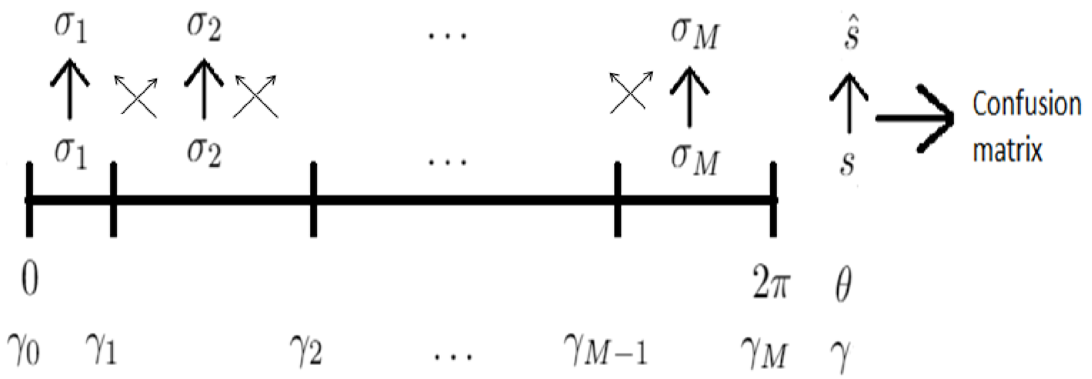

Figure 7: Notation used for this problem.

The objective is to find ɣ, which represents the borders of the partitions along the angle that are going to be made, so that an optimal solution, which makes it easier to the subjects to reach their intended destinations, is found.

s represents the compressed version of the angle, which consists of M different sectors.

ŝ is the estimated version of s. During calibration, a confusion matrix can be calculated, so we can know the probability of the system estimating a given σ_l when the true intended one was σ_k.

Another way of looking at the relation between s and ŝ is considering s as the input of a channel (and let σ_1 to σ_M be the different symbols each one with its probability), and ŝ as the output of that channel, so depending on the confusion matrix (channel model), the probabilities of the different symbols at the output are then:

$$\mathrm{P}(\mathrm{s}{=}\sigma_k) = \int_{\gamma_{k-1}}^{\gamma_k} P_{x0}(\vartheta)\mathrm{d}\vartheta$$
$$\mathrm{P}(\hat{s} = \sigma_l) = \sum_k P(s=\sigma_k)P(\hat{s}=\sigma_l|s=\sigma_k)$$

The problem to be solved is the following: find ɣ such that the mutual information between the estimated sector and the actual angle is maximized.

$$\text{Find } \gamma \text{ such that } \gamma = \max_\gamma I(\theta;\hat{s})$$

$$\max_\gamma I(\theta;\hat{s}) = \max_\gamma(H(\theta)-H(\theta|\hat{s})) = \max_\gamma(H(\hat{s})-H(\hat{s}|\theta))$$

$$\mathrm{H}(\hat{s}|\theta) = \int_0^{2\pi} P(\theta=\vartheta)H(\hat{s}|\theta=\vartheta)d\vartheta$$
$$\mathrm{H}(\hat{s}|\theta=\vartheta) = -\sum_l P(\hat{s}=\sigma_l|\theta=\vartheta)log(P(\hat{s}=\sigma_l|\theta=\vartheta))$$
$$\mathrm{P}(\hat{s}{=}\sigma_l|\theta=\vartheta) = \sum_k P(\hat{s}=\sigma_l|s=\sigma_k)P(s=\sigma_k|\theta=\vartheta)$$
$$= \mathrm{P}(\hat{s}{=}\sigma_l|s=\sigma_{\kappa\vartheta})$$

$$\mathrm{H}(\hat{s}|\theta) = -\int_0^{2\pi} P(\theta=\vartheta)\sum_l P(\hat{s}=\sigma_l|s=\sigma_{k\vartheta})log(P(\hat{s}=\sigma_l|s=\sigma_{k\vartheta}))\mathrm{d}\vartheta$$
$$= -\sum_k \int_{\gamma_{k-1}}^{\gamma_k} P(\theta=\vartheta)\sum_l P(\hat{s}=\sigma_l|s=\sigma_{k\vartheta})log(P(\hat{s}=\sigma_l|s=\sigma_{k\vartheta}))\mathrm{d}\vartheta$$
$$= -\sum_k \int_{\gamma_{k-1}}^{\gamma_k} P(\theta=\vartheta)\sum_l P(\hat{s}=\sigma_l|s=\sigma_k)log(P(\hat{s}=\sigma_l|s=\sigma_k))\mathrm{d}\vartheta$$
$$= -\sum_k\sum_l P(\hat{s}=\sigma_l|s=\sigma_k)log(P(\hat{s}=\sigma_l|s=\sigma_k))\int_{\gamma_{k-1}}^{\gamma_k} P(\theta=\vartheta)\mathrm{d}\vartheta$$

$$\mathrm{H}(\hat{s}|\theta) = -\sum_k \sum_l P(\hat{s} = \sigma_l | s = \sigma_k) log(P(\hat{s} = \sigma_l | s = \sigma_k)) P(s = \sigma_k)$$

$$\mathrm{H}(\hat{s}|s) = \sum_k H(\hat{s}|s = \sigma_k) P(s = \sigma_k)$$

$$= -\sum_k \sum_l P(\hat{s} = \sigma_l | s = \sigma_k) log(P(\hat{s} = \sigma_l | s = \sigma_k)) P(s = \sigma_k)$$

The problem can be simplified if we note the following:

If θ is given, then s is determined. There is no uncertainty in s if we know theta, we know the particular sigma for any given theta with a probability of one, and all the other sigmas have probability 0. With this in mind, the problem can be simplified, and at the end, the problem gets reduced to the following:

$$\mathrm{H}(\hat{s}|s) = H(\hat{s}|\theta)$$

$$\max_{\gamma} I(\theta; \hat{s}) = \max_{\gamma} I(s; \hat{s})$$

The maximum mutual information for s and ŝ (with a channel in between) depending on the distribution of probabilities of s (the input of the channel) is what is usually referred to as channel capacity. This problem then becomes easy to solve with an algorithm that calculates the theoretical channel capacity (and the distribution of probabilities at the input of the channel) e.g. Blahut-Arimoto algorithm.

This algorithm will take the channel (i.e. the confusion matrix) and, by performing recursive operations, will find the distribution of probabilities at the input (s) which maximizes the mutual information between the input (s) and the output (ŝ).

$$\text{Get } P(s = \sigma_i) \forall i = 1, ..., M-1$$

$$\text{Calculate } \gamma \ s.t. \int_{\gamma_{i-1}}^{\gamma_i} P(\vartheta) d\vartheta = P(s = \sigma_i)$$

Once the distribution of s is found, then ɣ needs to be adjusted so each of the sectors has the probabilities calculated by the algorithm.

After ɣ is found, the consequent solution will look like the following:

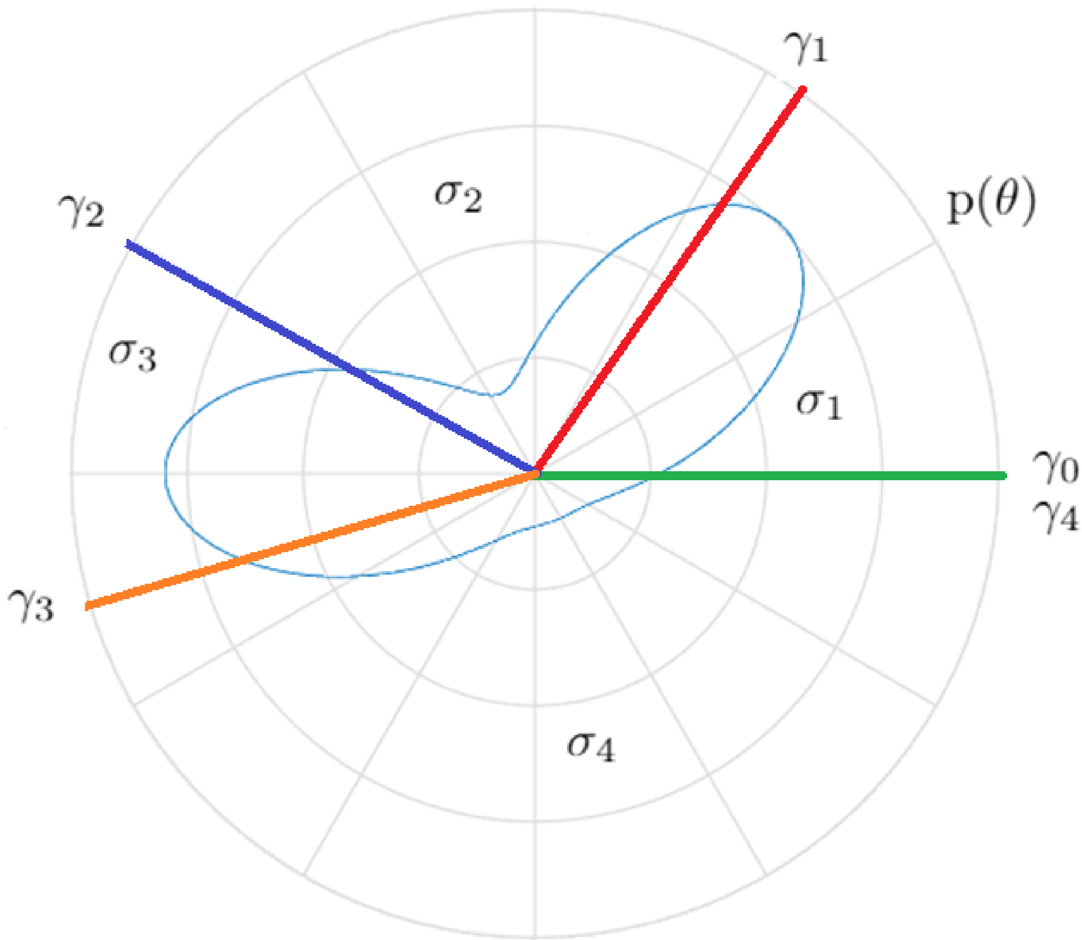

Figure 8: Optimal angle partition.

Note that θ = 0 is always the start point for finding the optimal gammas, so we will always have a border at θ = 0. This is because of the way the coding is done, and as far as finding the maximum mutual information is concerned, this starting point could be located at any given angle, or randomly changed at every step.

4.3. Moving the cursor

The cursor is located at a certain point on the screen. The optimal compression of the angle has been calculated, given a previously found confusion matrix. Next step is make the subject select a sector, and move in a particular angle inside that sector. Two questions may arise; the first one has to do with the actual angle of movement, that is, when a sector is selected, in what angle the cursor will move. The second one is, after the subject has selected one of the M options, how the probability density function changes so that the system can make it easier to reach a particular destination.

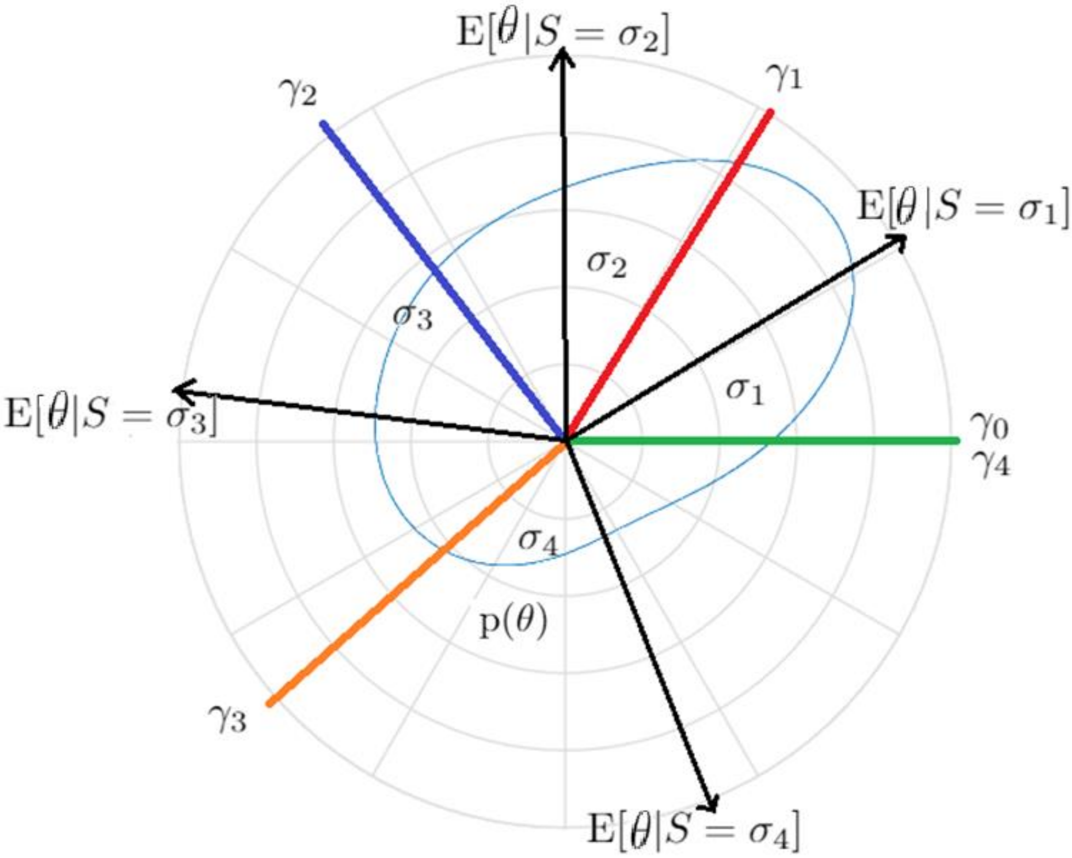

Figure 9: Optimal angle partition showing the actual possible directions of movement.

The actual directions of movement are given by the expected value inside the selected sector. We assume the estimated sector is the one the user has actually chosen, even when it is not the case. The reason for this is that the cursor should always move in the direction given by an angle that is actually inside the sector that has been estimated, not necessarily true if that condition is not imposed (and it would be confusing for the subject).

After the user has selected a sector, the 2D function that determines the probability density in the 2 dimensions of the screen is going to be updated by using the EEG data collected when making that decision.

$$f_t(r,\theta) = f_{t-1}(r,\theta) \cdot g(e_t|\mathcal{M}_t, r, \theta)$$

e_t corresponds to the EEG evidence at time t, M is the selected sector at time t.

g is a function that corresponds to the normal distribution of the SSVEP data.

This dynamic update, after each time the user selects and the cursor moves can be seen in the following figure:

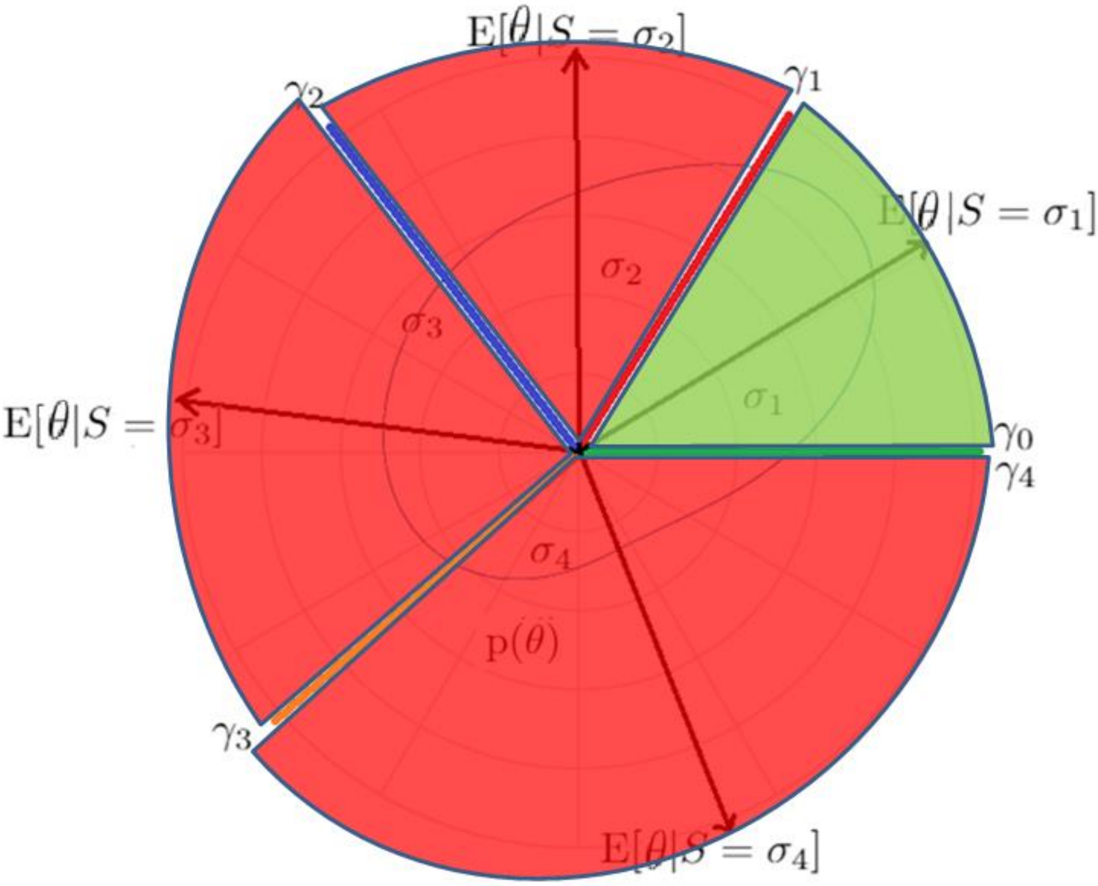

Figure 10: Updates after selecting the sector in green.

In figure 10 the subject has selected σ. Therefore, the cursor will move towards the direction that corresponds to σ_1. The 2D function will get updated, so the area that corresponds to the selected sector will get boosted, and the remaining probability (in red) will get dimmed down, all of this depending on the EEG evidence that has been generated in order to make the selection. After the update, all the probability will be normalized so it adds up to one again.

In addition to giving the subject M choices to choose from, a reset/click option will be added. Whenever the user wants to click, the probability density function over the screen will get back to the original one, forgetting all the successive previous updates. This also works if the subject decides to change their intended destination, for the previous updates may play against them because the probability of the new intended destination can be rather low.

Another important thing to have in mind is step size. In other words, how far the cursor moves after every step. The easiest way would be having a fixed step size, so every time the cursor would move the same amount. Other solutions with dynamic step size can be more interesting, for example one that depends on how certain the system is about the user actually willing to move in that direction.

5. Results: Environment

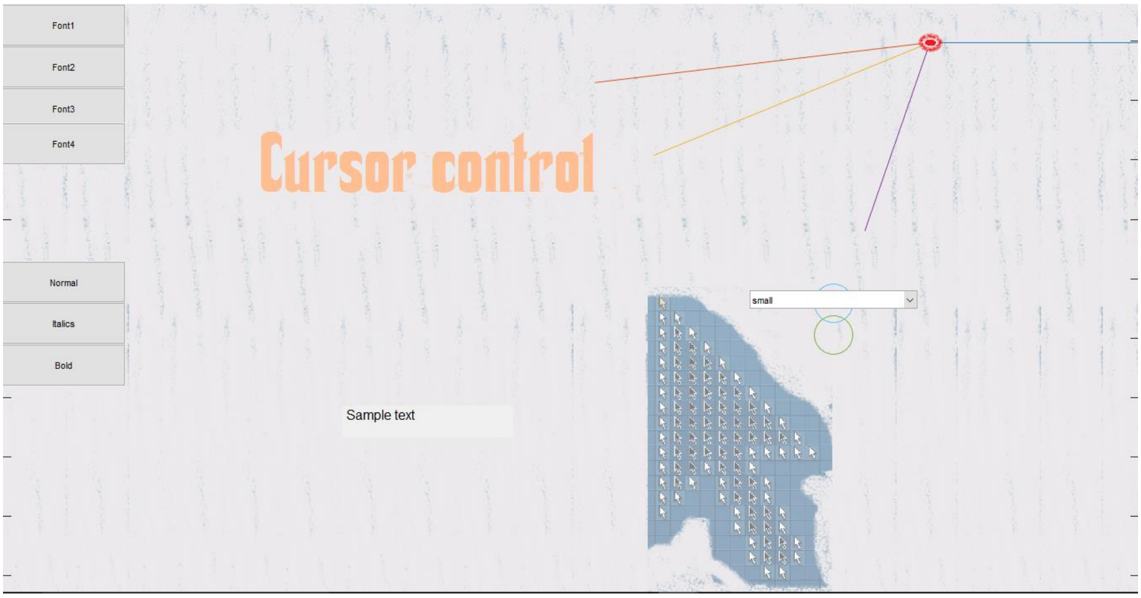

Figure 11: General view of the cursor control environment.

In this chapter the environment used to test the cursor control system will be presented.

This environment consists of a text “sample text”, which can change its size, font, and style, by having the subject perform some actions. The areas the subject will most likely want to go to are the 7 buttons on the left side of the screen, and the menu on the right side.

The red dot near the top right corner represents the current position of the cursor. Coming out of it, 4 lines are drawn. They represent the borders between the sectors. Therefore, the user has 4 sectors to choose from. In other words, the angle is compressed into four parts. The subjects can identify which sector they want to move easily, knowing that the count starts at θ = 0 (horizontal-right), so sectors will be assigned numbers 1-4 in a counterclockwise manner. To select any of the sectors, the user will need to press the numbers 1-4 on the keyboard, and to perform a mouse click (or/and reset back to the original probability density function), key 0 must be pressed.

The step size has two different values, small and large. The small step size is used whenever the cursor is close to any of the peaks that form the probability density function. While the cursor is not close to any of the peaks, the step size changes to its largest value. This way, when the cursor has to be more precise in its movements, for example when trying to select among the different options of the dropdown menu, a smaller step size can be more convenient. This way of having two different values for the step size is one of the most basic ones, and a fixed value is set in order to make the distinction of when the cursor is close to the peaks (and small step size used), and when it is further away so the step size switches back to its large value.

When a sector is selected, an artificial EEG signal corresponding to the selected sector is generated. That signal is the one used to update the probability density function of the bi-dimensional space. After this, the system decides which class that particular sequence belongs to and moves towards the corresponding angle.

In particular, SSVEP 4 class data is used, in response to an m-sequence based flickering at a rate of 60 Hz with sequences of 31 bit, which results in 133 samples.

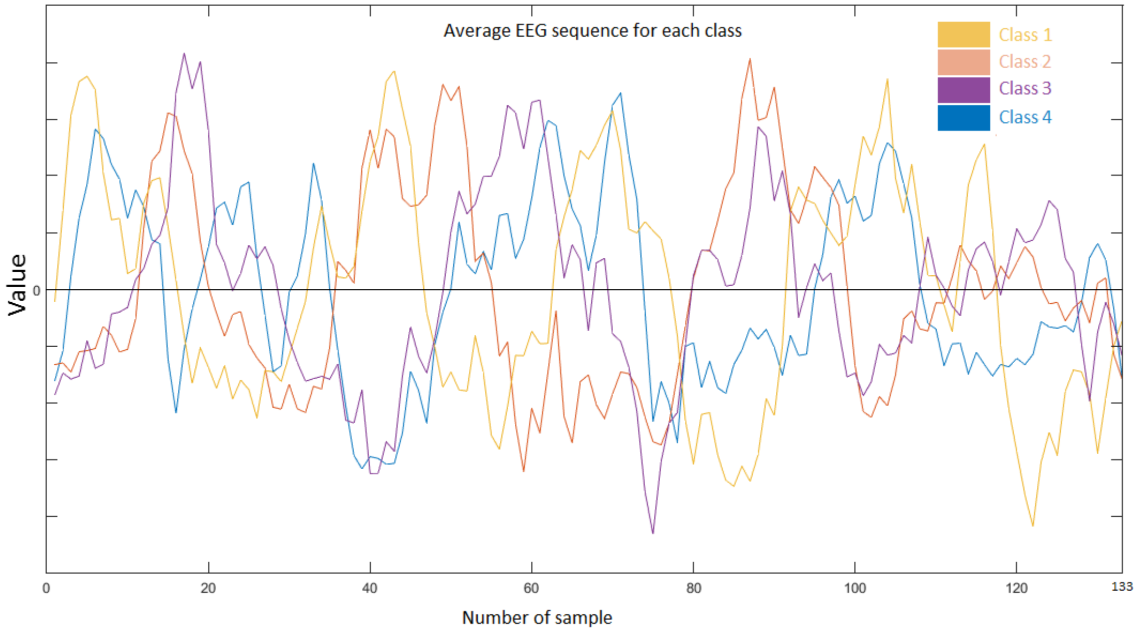

Figure 12: Comparison between all 4 classes of SSVEP signals used.

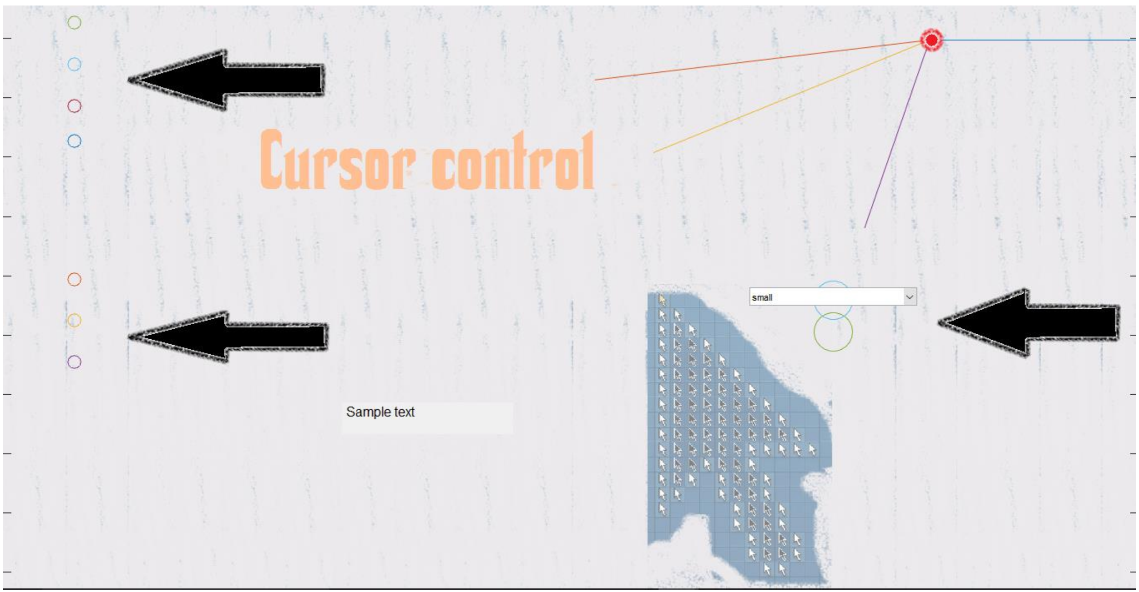

Figure 13: Arrows showing the locations of the objects.

The black arrows point to where the peaks of the probability density function are located. Each circle is centered at one peak, and the size of it represents how high it is. In this case, we have a total of 7 peaks that would correspond to the 7 buttons on the left side, and 2 larger peaks that correspond to the dropdown menu on the right side (there is a peak which is slightly below the dropdown menu, which corresponds to when, after the subject clicking on it, it is displaying all the options below).

Let us suppose that, with the cursor being at the position shown in figure 13, we want to move to the dropdown menu on the right side of the screen. In this situation, and provided that the cursor is located where the red dot is (top right), the subject could either choose sectors 3 or 4, since the menu spans inside these two sectors.

Figure 14: Detail of how the partitions are shown around the cursor position.

Let's suppose the subject selects sector #3 to reach the menu.

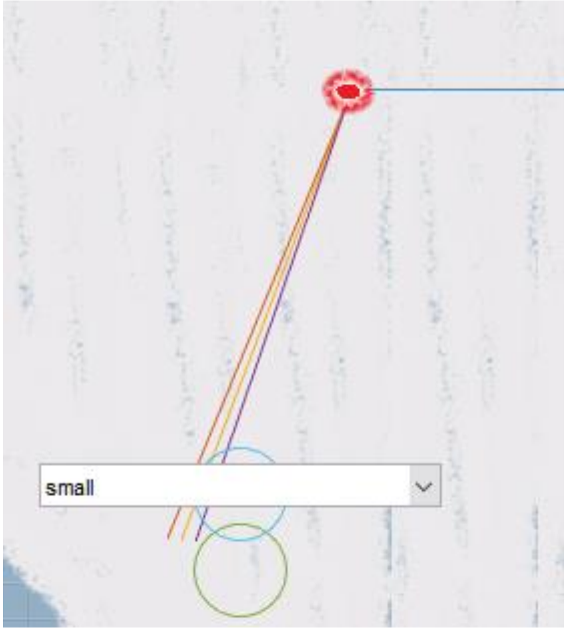

Figure 15: Sectors after moving towards the destination.

Figure 15 corresponds to the cursor after selecting sector #3 for 3 consecutive steps. Note that all the probability has been concentrated between sectors 2 and 3 after the updates, leaving sectors 1 and 4 (the larger ones) with a low probability inside.

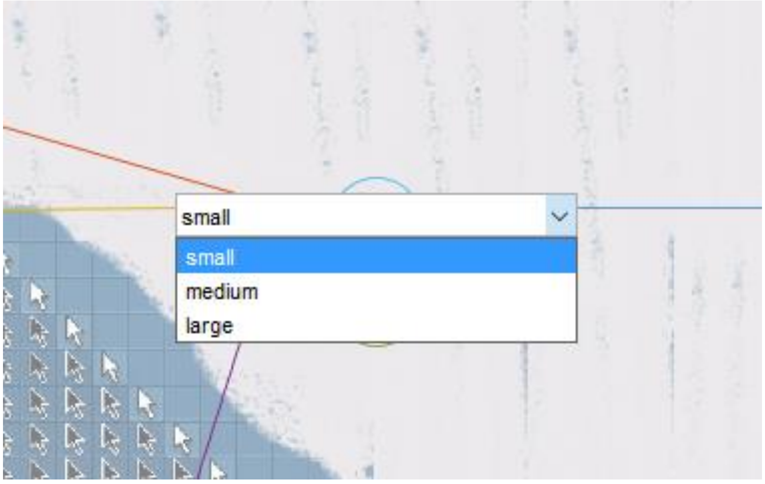

Figure 16: Sectors reset after clicking and opening menu.

In figure 16, after reaching the menu and clicking, the function has been reset (it resets every time key 0 is pressed) and now the sectors are restored so it's easier to move to any new direction, for example move to one of the seven buttons on the left side of the screen.

6. Budget

This thesis has been completed after 6 months. If we consider working full time for 6 months (most of the time programming, rest for literature review, meetings, thinking new ideas…). 10,000€ would be a sensible price for the time worked.

The environment has been made using MATLAB, whose student license is 69€. It already includes most of the toolboxes.

The potential final clients this finished product (and many others that use BCIs) would have are persons with mobility impairments. Specifically, people suffering from locked-in syndrome. Patients suffer from complete paralysis in nearly all muscles of the body, except for eye movements or blinking. Therefore, the product will have to be adapted to every individual, depending on their condition and their needs. Also, it will be important to collect data from the experience of all users, so that changes can be made to further adapt to each one's needs.

Medical research centers (or similar) are going to be the potential clients of the prototype described in this thesis, complete knowledge of the medical condition a person suffers is very important in order to slowly introduce the final product into the daily life of a person. This will lead to the development of a more suitable product not just for testing but for daily use.

7. Conclusions and future development:

7.1. Possible improvements

Several possible improvements could be made to the environment in order to provide a better user experience and translate the angle compression method for other uses other than mouse control.

- Collect actual data of how the mouse is moved and where are the most and least important areas on the screen, for a given application/program and subject. This way, a more realistic probability model could be used so subjects could reach their desired destinations faster.

- Have a more dynamic form of controlling the step size so the subject is not slowed down by a short step size when not needed and vice versa. Additionally, take the uncertainty or, in other words, how spread the probability density is inside the selected sector, in order to control the step size.

- Be able to present the user a better way of identifying the sectors, especially if more classes are added (more than 4).

- Consider sizes of objects on the screen so that one single object can only be reached by selecting one partition (unlike what was happening in figure 13), in other words the partition doesn't need to have a smaller angle than the one that spans the complete object. This way, there are no close narrow partitions that point towards only one object, and these additional partitions can be pointing to other potentially desired areas of the screen.

- Make the system take into consideration, not only intended destinations after each step, but also consider a longer term set of possible different upcoming destinations so that trajectories can be optimized in advance when moving across those destinations.

7.2. Additional considerations

- The environment, explained in chapter 5 is programmed in Matlab 2015a. Other versions may not work.

- To use it, read the readme document inside the documentation folder. As explained, select the different sectors by pressing keys 1 to 4 and perform clicks and reset back to the original probability density function with the 0 key.

Bibliography:

[1] Bos, D. P. O., Reuderink, B., van de Laar, B., Gurkok, H., Muhl, C., Poel, M., ... & Nijholt, A. (2010, October). Human-computer interaction for bci games: Usability and user experience. In *Cyberworlds (CW), 2010 International Conference on* (pp. 277-281). IEEE.

[2] Cheng, M., Gao, X., Gao, S., & Xu, D. (2002). Design and implementation of a brain-computer interface with high transfer rates. *IEEE transactions on biomedical engineering*, *49*(10), 1181-1186.

[3] Ding, J., Sperling G., and Srinivasan, R. (2006). Attentional modulation of SSVEP power depends on the network tagged by the flicker frequency. *Cereb Cortex.* 16, 1016-1029.

[4] Gürkök, H. (2012). *Mind the sheep! User experience evaluation & brain-computer interface games*. University of Twente.

[5] Larsen, E. A. (2011). Classification of EEG signals in a brain-computer interface system.

[6] Müller-Putz, G. R., Scherer, R., Brauneis, C., & Pfurtscheller, G. (2005). Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components. *Journal of neural engineering*, *2*(4), 123-130.

[7] Nezamfar, H., Orhan, U., Erdogmus, D., Hild, K. E., Purwar, S., Oken, B., & Fried-Oken, M. (2011, May). On visually evoked potentials in eeg induced by multiple pseudorandom binary sequences for brain computer interface design. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2044-2047). IEEE.

[8] Nezamfar, H., Salehi, S. S. M., Moghadamfalahi, M., & Erdogmus, D. FlashType (TM): A Context-Aware c-VEP based BCI Typing Interface using EEG Signals.

[9] Oken, B. S., Orhan, U., Roark, B., Erdogmus, D., Fowler, A., Mooney, A., ... & Fried-Oken, M. B. (2014). Brain–computer interface with language model–electroencephalography fusion for locked-in syndrome. *Neurorehabilitation and neural repair*, *28*(4), 387-394.

[10] L. J. Trejo, R. Rosipal and B. Matthews, "Brain-computer interfaces for 1-D and 2-D cursor control: designs using volitional control of the EEG spectrum or steady-state visual evoked potentials," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 225-229, June 2006.

[11] Luzheng Bi, Jinling Lian, Ke Jie, Ru Lai, Yili Liu, A speed and direction-based cursor control system with P300 and SSVEP, In Biomedical Signal Processing and Control, Volume 14, 2014, Pages 126-133.

Glossary

- BCI: Brain Computer Interface
- EEG: Electroencephalography
- SSVEP: SSVEP stands for Steady State Visual Evoked Potentials. A Visually Evoked Potential (VEP) is the response of the visual cortex to a flashing stimulus. If such stimulus is flashing in a steady manner, then they become SSVEPs. The simplest method (and one that could be used for this problem) consists in a flickering stimulus at a fixed frequency. These stimuli induce a response in corresponding frequencies (that of stimulus and higher harmonics).

  In this example, the user will have to choose among 4 different choices. Thus, 4 conveniently chosen visual stimuli will be presented to the subject. While ignoring others, the subject will focus on the stimulus that corresponds to his/her desired option while ignoring the others. The way the environment presents all four choices and maps them to a particular stimulus should be such that the subject can easily and quickly identify what stimulus to focus on. A reasonable way of presenting these stimuli would be placing them one at every corner of the screen, so that they are sufficiently spaced out to avoid interferences among each other.

  Artificially generated SSVEP is used for this example, when the subject clicks a key (1 to 4), the system generates a SSVEP sequence, taking into account the mean and covariance of the corresponding class. The next reasonable step to take would be implementing a BCI interface that takes and processes actual SSVEP evidence which is generated by the subjects after focusing their sight at the flickering lights/patterns in order to make the corresponding selections.