



ESCUELA DE INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE LA LAGUNA

TRABAJO DE FIN DE GRADO

**Sensores para lectura de actividad neuronal
aplicados al control de una silla de ruedas**

Cristina Garrido Amador
Jonay Tomas Toledo Carrillo

GRADO EN INGENIERÍA INFORMÁTICA

Índice

1. Introducción	4
2. Objetivo principal	4
3. Resumen	5
4. Abstract	6
5. Antecedentes y estado actual del problema	7
5.1. Historia de la silla de ruedas	7
5.1.1. Sillas de ruedas manuales	8
5.1.2. Sillas de ruedas motorizadas	8
5.1.3. Otros tipos de sillas de ruedas	9
5.2. Estado actual del problema	10
5.3. Sillas de ruedas inteligentes	11
5.3.1. Silla de ruedas con IrisBond	11
5.3.2. Silla de ruedas controlada por la lengua	12
5.3.3. Sillas de ruedas con Inteligencia Artificial	13
6. Sensores OpenBCI Ganglion	14
6.1. Características del Hardware	14
6.2. Programa OpenBCI GUI	15
6.3. Librerías utilizadas	17
7. Captura de datos	18
7.1. Captura de datos con OpenBCI	18
8. Descripción del trabajo	19
8.1. Primeros pasos	19
8.2. Procesamiento con Octave	19
8.3. Detección de gestos	22
8.4. Pruebas y resultados	28
9. Líneas futuras	34
10. Conclusión	35
11. Conclusion	36
12. Agradecimientos	37
A. Anexo I: Códigos	39
A.1. CalculoMaxMin	39
A.2. CalculoNuevo	42
A.3. ProcesaFich	45
A.4. Outlier	47
A.5. Rmoutlier	49

Índice de figuras

1.	Primera silla de ruedas	7
2.	Comienzos del siglo XX	7
3.	Sillas de ruedas manuales	8
4.	Sillas de ruedas motorizadas	8
5.	Sillas de ruedas con accionado de un solo brazo	9
6.	Sillas de ruedas reclinables e inclinables	10
7.	Sillas de ruedas deportivas	10
8.	Logo IrisBond	11
9.	Sistema IrisBond	11
10.	Empresa creadora	12
11.	Tongue Driver System	12
12.	Logo empresa Hoobox One	13
13.	Silla con mecanismo Wheelie 7	13
14.	Marca de los sensores	14
15.	Kit de sensores OpenBCI	14
16.	Placa Ganglion	15
17.	Pantalla inicial OpenBCI.GUI	15
18.	Entorno de trabajo OpenBCI.GUI	16
19.	Producción de ondas en OpenBCI.GUI	16
20.	Fichero de datos capturados	18
21.	Fichero de datos capturados	18
22.	Señal montada en la red eléctrica	20
23.	Filtro pasa baja	20
24.	Señales de los sensores	21
25.	Eje X: frecuencia en Hz, Eje Y: energía de la frecuencia en voltios	21
26.	Señal sin aplicar la mediana	22
27.	Señal tras aplicar la mediana	23
28.	Señales de energía para el gesto derecha	24
29.	Sentencia if para hallar los rangos	24
30.	División sentencia if para hallar los rangos	25
31.	Menor número de decimales para hallar los rangos	25
32.	Resultados	25
33.	Valores máximos y mínimos	26
34.	Grafica de datos a analizar	26
35.	Diferencia de gráficas	27
36.	Procesamiento de datos. Índices: 0.9 - 1.1	27
37.	Tablas de procesamiento	29
38.	Datos tomados en diferentes días	30
39.	Datos tomados en el mismo día	30
40.	Tablas de procesamiento con límite inferior 0.9: gesto izquierda	31
41.	Tablas de procesamiento con límite inferior 1.0: gesto derecha	32
42.	Tablas de procesamiento con límite inferior 1.1: gesto subir	32
43.	Tablas de procesamiento con límite inferior 1.0: gesto bajar	33
44.	Ultracortex EEG Headset	34
45.	Esquema de trabajo de MILO	34

Índice de cuadros

1.	Tabla de análisis del gesto	28
2.	Tabla de límites para los rangos	30

1. Introducción

El objeto de la siguiente memoria es explicar el trabajo de fin de grado que se ha llevado a cabo durante el curso presente. El proyecto en cuestión se basa en analizar las ondas cerebrales producidas ante determinados gestos de la cara, captadas por los sensores EEG OpenBCI Ganglion, para que controlen el movimiento de una silla de ruedas inteligente.

La idea propuesta surge tras la observación de la población existente en la actualidad que sufre de paraplejía y el análisis del mercado, con el que se descubre el elevado coste de las sillas de ruedas eléctricas. Además, dichas sillas de ruedas no les permiten disfrutar de una autonomía completa, de forma que, cuando el usuario desea acceder a lugares abiertos, necesita de otra persona que lo acompañe.

Por esta razón, lo que se desea con este proyecto es estudiar la posibilidad de utilizar las señales neuronales que se producen al activar ciertas partes de la cara, creando así una interfaz de control para llevar a cabo el movimiento de la silla de ruedas y, a su vez, poder brindarles a las personas que sufren paraplejía la autonomía completa que anhelan.

Este objetivo se llevará a cabo con una serie de sensores, que a través de una interfaz bastante sencilla, conectarán las acciones faciales del usuario con la silla y así poder moverse este libremente por cualquier lugar, ya sea cerrado o abierto.

2. Objetivo principal

La finalidad del presente trabajo es analizar las ondas cerebrales, producidas al realizar ciertos gestos, de manera que puedan dirigir los movimientos de una silla de ruedas eléctrica. Este proyecto se ha propuesto con el fin de superar el problema actual existente con las sillas de ruedas inteligentes, del que se hablará más adelante, con el que tienen que lidiar las personas con paraplejía o con movilidad reducida, de forma que pueda aumentar su grado de autonomía al margen de estar en un espacio abierto o cerrado.

Los nuevos avances tecnológicos que se están utilizando en el sector de las sillas de ruedas eléctricas, han conseguido una mayor calidad de vida para los usuarios con paraplejía. Estos avances se basan en el empleo de cámaras, las cuales utilizan sensores con radiación infrarroja para captar la posición de los ojos y así poder percibir el movimiento de estos, que en ciertos espacios abiertos, cuando les incide la luz solar o se encuentran con demasiada luminosidad en el ambiente, pierden calidad de percepción provocando que los gestos que se realizan no puedan ser captados correctamente y, por ello, el usuario acaba perdiendo autonomía.

Para la realización del objetivo principal se van a estudiar las ondas eléctricas del cerebro que se generan ante los siguientes gestos: levantar y bajar las cejas y cerrar el ojo izquierdo y el derecho de manera independiente. Dichas ondas se analizarán con los sensores OpenBCI Ganglion, que se colocan en la frente con una banda de cuatro electrodos, de forma que se miden cuatro zonas de la cara. Dos en la parte frontal y dos en los laterales. A su vez, se utilizan para la toma de referencia unas pinzas con electrodos que se colocan en las orejas, a modo de pendientes.

Una vez se recogen las ondas de un usuario, se estudian y se comparan con una lista de gestos ya determinados, de manera que si las ondas que ha producido son similares a las ya estudiadas, estas se emparejarán con el gesto, por lo que la silla se movería tal cual ha indicado el sujeto.

Por ejemplo, el usuario desea moverse hacia delante, por tanto, deberá levantar las cejas, lo cual producirá una señal concreta que se analiza con el código y si coincide con alguna señal ya observada, el programa envía información a la silla y comienza a rodar hacia delante hasta que el individuo desee parar, teniendo que realizar el gesto de parada.

3. Resumen

El proyecto desarrollado se centra en la utilización de sensores de actividad eléctrica colocados en la cabeza de un usuario de silla de ruedas. Estos sensores son utilizados para leer ondas cerebrales con la intención de diferenciar distintos gestos. El objetivo es desarrollar un interfaz persona-computador a través de dichos sensores, de tal forma que si el usuario realiza distintos gestos, el software reconoce y diferencia el gesto realizado.

Esta detección de gestos se pretende utilizar para realizar una interfaz que, a partir de estas señales, genere comandos para mover una silla de ruedas inteligente de manera autónoma, donde la dirección de movimiento dependerá del gesto que el usuario haya efectuado.

Los datos a analizar se reciben gracias a los sensores que se colocan en la parte delantera del cráneo de la persona, quien deberá repetir los mismos gestos varias veces para poder analizar las ondas específicas asociadas a cada gesto. De esta manera se podrán traducir estos gestos a comandos de la silla de ruedas.

Una vez se han distinguido las señales y se tiene el estudio creado, se asocian acciones con dichos gestos, de manera que cada mueca se corresponda con una maniobra, es decir, si el usuario desea ir a la izquierda debe cerrar únicamente el ojo izquierdo. De esta forma, cada vez que desee hacer dicha acción deberá repetir el gesto. Para ejecutar esta parte se actualiza el código principal y se asocia el gesto con el movimiento.

Finalmente, a pesar de que el objetivo en sí mismo no se alcanzó, se ha de resaltar que se acotó notablemente la vía de investigación para lograr una silla de ruedas dirigida única y exclusivamente a través de gestos faciales.

El problema reside en que, a pesar de que cada sensor obtenía una onda, estas eran muy semejantes, por lo que resultaba muy complejo distinguirlas en el momento del análisis.

Asimismo, hay que destacar el número de sensores utilizados, pues al reducirse a cuatro, colocados en la parte frontal de la cabeza, la sensibilidad de las ondas para captar la información era muy reducida. Así pues, se considera fundamental trabajar con un mayor número de sensores dispuestos de tal forma que abarquen un número superior de zonas cerebrales.

4. Abstract

The project developed focuses on the use of electrical activity sensors placed on the head of a wheelchair user. These sensors are used to read brain waves in order to differentiate between different gestures. The objective is to develop a person-computer interface through these sensors, so that if the user makes different gestures, the software recognizes and distinguishes the gesture made.

This gesture detection is intended to be used to make an interface that, from these signals, generates commands to move an intelligent wheelchair autonomously, where the direction of movement will depend on the gesture made by the user.

The data to be analysed is received thanks to sensors placed in the front of the person's skull, who must repeat the same gestures several times in order to analyse the specific waves associated with each gesture. In this way these gestures can be translated into wheelchair commands.

Once the signals have been distinguished and the study has been created, actions are associated with these expressions, so that each grimace corresponds to a manoeuvre, i.e. if the user wishes to go to the left he must only close his left eye. In this way, every time he wants to do that action he must repeat the gesture. To execute this part, the main code is updated and the expression is associated with the movement.

Finally, despite the fact that the objective itself was not reached, it should be noted that the research path to achieve a wheelchair directed solely and exclusively through facial gestures was significantly limited.

The problem is that, although each sensor obtained a wave, these were very similar, so it was very complex to distinguish them at the time of analysis.

Also, the number of sensors used should be highlighted, since when reduced to four, placed in the front of the head, the sensitivity of the waves to capture the information was very low. Therefore, it is considered essential to work with a greater number of sensors arranged in such a way that they cover a greater number of brain areas.

5. Antecedentes y estado actual del problema

En la última década se han llevado a cabo diferentes proyectos para mejorar la autonomía de las personas parapléjicas o con movilidad reducida dado que es un problema bastante visible en la actualidad. No todas las soluciones que se encuentran en el mercado sirven para todos los tipos de discapacidades, es por ello que a continuación se van a comentar las diferentes variantes de las sillas de ruedas que se pueden encontrar hoy en día.

5.1. Historia de la silla de ruedas

Una silla de ruedas [1] es un dispositivo técnico utilizado por personas con una discapacidad motriz, la cual implica reducción o pérdida de la locomoción. Dicha reducción puede haber sido causada por una enfermedad o una lesión física, imposibilitando caminar a la persona bien por daño físico o por daño cerebral.

Las primeras anotaciones de la silla de ruedas se encuentran en China, donde se utilizaban las carretillas para poder mover no solo objetos pesados, sino también a personas. No fue hasta el año 525 A.C. cuando se llevó a cabo la distinción entre ambas funciones, comenzando a aparecer las primeras sillas de ruedas. [2]

En el año 1595 aparece la silla de ruedas en Europa gracias a un inventor, de nombre desconocido, quien crea la primera de estas para el rey Felipe II. A pesar de tener reposabrazos y reposapiernas, se encontraban en ella ciertos defectos como el hecho de no tener una propulsión eficiente requiriendo así la ayuda de otra persona para poder impulsarse.

En 1655, Stephan Ferffler, construyó la primera silla autopropulsada, la cual consistía en un chasis de tres ruedas que se movía a través de manivelas y ruedas dentadas.



Figura 1: Primera silla de ruedas



Figura 2: Comienzos del siglo XX

En los años siguientes se inventaron distintas sillas, como por ejemplo la silla Bath [3], que se trataba de un sillón, con tres o cuatro ruedas, o un carro ligero, de manera que podía ser conducido por caballos, burros o un pony.

En 1933 Hennings y Everest, dos ingenieros mecánicos, inventaron la silla de ruedas portátil y plegable, se trataba de una silla ligera gracias al uso del acero en su fabricación. En la actualidad es este el modelo al uso, en el que se han incorporado mejoras, no solo en el diseño, sino también en los materiales.

A continuación se comentarán los diferentes tipos que se encuentran hoy en día en el mercado, si bien se profundizará más en el modelo inteligente ya que es el prototipo que se tratará en este proyecto.

5.1.1. Sillas de ruedas manuales

En primer lugar, se encuentra la silla de ruedas manual, estas sillas son las más simples de todas dado que cuentan con uno o dos reposapiés, cuatro ruedas, dos grandes en la parte trasera y dos ruedas giratorias más pequeñas en la parte delantera.

Las ruedas traseras cuentan con unas llantas con un diámetro menor y sobresalen de estas, de forma que el usuario pueda mover la silla sin necesidad de estar acompañado.

Este tipo de silla también cuenta con unas manillas que se encuentran en la parte trasera cuyo objetivo es que una persona externa pueda, no solo mover la silla sino también frenarla.

Dentro de este tipo de silla de ruedas se pueden observar dos subtipos, las plegables y las rígidas. Las primeras son más simples pues su objetivo es poder plegarse para poder ser más cómodas de transportar, pero las segundas cuentan con un avance en los materiales siendo cada vez más ligeras. A su vez también presentan más innovaciones como pueden ser los amortiguadores.



Figura 3: Sillas de ruedas manuales

5.1.2. Sillas de ruedas motorizadas

Las sillas de ruedas motorizadas son iguales a las simples a excepción de que integran baterías y motores eléctricos, de forma que gracias a un joystick la silla se puede mover. Dicho joystick, se puede encontrar en diferentes puntos de la silla como puede ser el reposabrazos o la parte superior de la misma, de modo que el usuario utiliza la barbilla para poder hacer uso del joystick.

En este tipo de silla de ruedas también se puede encontrar otro subtipo, la scooter de movilidad. Estas no se consideran propias del mercado de la paraplejía dado que las personas que las utilizan pueden caminar, si bien es cierto que presentan una capacidad limitada sin que por ello requieran una silla como único medio de locomoción.

La gran ventaja de las sillas de ruedas motorizadas es que se pueden utilizar en cualquier tipo de superficie, ya sea exterior o interior. El inconveniente es que aumenta bastante el peso puesto que cuenta con un motor incorporado.



Figura 4: Sillas de ruedas motorizadas

5.1.3. Otros tipos de sillas de ruedas

En este apartado se comentarán algunas sillas de ruedas que no son tan conocidas pues su uso es más limitado, por lo que no se suelen ver a diario como ocurre con las simples o las motorizadas.

En primer lugar, se encuentran las sillas de ruedas que se accionan con un solo brazo [4]. Este tipo de sillas se utilizan cuando la persona sufre hemiplejía, es decir, parálisis en un lado del cuerpo causada por una lesión cerebral o la médula espinal.

Las sillas hemipléjicas pueden ser de dos tipos. La primera de ellas es un modelo simple con la excepción de que tiene dos palancas, una para direccionar la silla y otra que se conoce como palanca de marcha. La segunda de estas tiene tres posiciones, marcha atrás, neutral y hacia delante. La primera palanca, es decir, la que direcciona la silla, se mueve hacia delante y gira, de manera que si el usuario desea ir hacia la derecha, mueve la palanca hacia delante y hacia la derecha y la silla sigue el movimiento.

El segundo tipo de silla hemipléjica, tiene el control de las dos ruedas en un mismo lado, de modo que en el lado del brazo a utilizar, se encontrarán dos circunferencias de agarre, la más grande y cercana al usuario será la que controle dicha rueda, mientras que la circunferencia más alejada será la que controle la del lado contrario, gracias a un eje concéntrico interno. Para poder manejar la silla de forma que vaya hacia delante o hacia atrás, la persona deberá agarrar y mover a la misma vez ambas circunferencias. En caso de querer girar solo hacia un lado, deberá mover la circunferencia adecuada.



Figura 5: Sillas de ruedas con accionado de un solo brazo

Seguidamente se van a comentar las sillas reclinables e inclinables [5]. La diferencia entre estas no es otra que la posición del usuario.

Las sillas reclinables son aquellas en las que el respaldo de la silla se puede mover 90 grados, dejando al usuario en posición casi horizontal respecto al suelo. Esto ayuda a que el peso de la persona se distribuya mejor y, a su vez, también ayuda a la hora de tener que mover al paciente de la silla a la cama o a otros lugares.

Por otro lado, las sillas inclinables mantienen en todo momento el mismo ángulo entre cadera, rodillas y tobillos, de forma que se mueve la silla enteramente hacia atrás hasta un máximo de 50 grados.

Los beneficios que se encuentran en estas sillas es que siempre se mantiene la misma postura ayudando a aquellos pacientes que sufren escoliosis. A su vez, también tiene algunos inconvenientes, entre ellos el hecho de que al estar siempre sentado en la misma posición la vejiga se encuentra constreñida, lo cual no ayuda a las personas que sufren de vejiga o tienen un catéter.



Figura 6: Sillas de ruedas reclinables e inclinables

Otro de los tipos de sillas de ruedas que se pueden encontrar son las deportivas, estas sillas se han diseñado para que las personas que sufren paraplejía puedan practicar diferentes deportes, como por ejemplo baloncesto, rugby, fútbol, atletismo, etc.

Para cada deporte, las sillas van a ser diferentes puesto que cada uno de estos exige ciertas características para poder ser más rápido o tener más control sobre sí mismo o una pelota. Aunque también se encuentran algunas características que se dan para todas y cada una de ellas, por ejemplo, ninguna de estas sillas puede ser plegable, puesto que así cuentan con una mayor estabilidad, a su vez, las ruedas cuentan con una inclinación negativa para poder realizar giros bruscos sin perder la estabilidad.



Figura 7: Sillas de ruedas deportivas

5.2. Estado actual del problema

Hoy por hoy se pueden encontrar diferentes sillas de ruedas especializadas en la tetraplejía, desde sillas manuales que se mueven gracias a un joystick en la barbilla, hasta sillas eléctricas que utilizan dispositivos conectados a estas para que el usuario pueda contar con una mayor autonomía en el día a día. Los instrumentos que se encuentran en estas sillas son bastante innovadores, dado que lo que buscan es que los usuarios no necesiten a otra persona a su lado para poder moverse, tanto si se trata de ambientes cerrados como de ambientes abiertos.

Recientemente se han ido añadiendo mejoras a las sillas de ruedas eléctricas de forma que el usuario sea cada vez más independiente. Para realizar estas mejoras se han utilizado nuevas tecnologías como el Eye Tracking, de manera que el movimiento de la silla de ruedas lo estipule la dirección hacia donde apuntan los ojos.

A pesar de tener bastantes ventajas, estos sensores cuentan con un obstáculo bastante importante, ya que cuando el usuario sale a la calle, la luz solar puede incidir directamente en la cámara o, en caso de que haya bastante luminosidad en el ambiente, el sensor puede no captar de igual forma la mirada del usuario, robándole de esta manera esa libertad que había conseguido en espacios cerrados.

Ante la autonomía que se da solo en lugares cerrados y contando con los productos tecnológicos que se encuentran hoy en día en el mercado, el proyecto que se presenta será el estudio de estos nuevos dispositivos, de manera que el objetivo que se pretende alcanzar es que el usuario tetrapléjico no necesite la ayuda de otra persona para moverse en cualquier superficie o espacio, ya sea cerrado o abierto.

Simultáneamente se estudiará la realización de un prototipo de bajo coste, con sensores cerebrales, para ayudar de esta forma a los usuarios a obtener una silla de ruedas eléctrica que les sea más práctica puesto que podrán ser completamente autónomos y moverse libremente por cualquier tipo de ambiente y de superficie.

5.3. Sillas de ruedas inteligentes

Como se ha comentado anteriormente, se encuentran diferentes tipos de sillas de ruedas, manuales y motorizadas. En este proyecto se va a profundizar más en el tipo motorizado con mejoras electrónicas, o también conocidas como sillas de ruedas inteligentes.

Las sillas de ruedas inteligentes, son aquellas que emplean un sistema de control como interfaz de usuario. De esta forma, el usuario no tendrá que dirigir la silla de ruedas de forma física, dado que se llevará a cabo el control de esta desde un ordenador gracias al uso de sensores. Dichos sensores pueden activarse a través de diferentes métodos, según el que se desee utilizar, desde la voz, hasta los ojos, pasando por captar las ondas cerebrales de diferentes maneras.

Esta variante de sillas de ruedas está diseñada para usuarios que presentan otro tipo de problema además de la paraplejía, como pueden ser problemas cognitivos o cerebrales.

Ya que el objetivo de este proyecto es dar soporte a los usuarios que sufren tetraplejía, serán estos los tipos de sillas de ruedas que se explicarán detalladamente a continuación.

5.3.1. Silla de ruedas con IrisBond

Irisbond [6] utiliza la tecnología Eye Tracking o seguimiento ocular, proceso de medir el movimiento de los ojos en relación con la cabeza, para crear un software de inteligencia artificial que traduce los movimientos oculares de manera precisa dentro de una pantalla.

Para la utilización de esta tecnología se lleva a cabo una curva de aprendizaje [7] desarrollada por profesionales de la educación y la Comunicación Aumentativa y Alternativa (CAA), de manera que el usuario pueda avanzar de manera pautada e intuitiva.

Los pasos que se siguen en esta curva de aprendizaje serían en primer lugar la estimulación visual, donde el usuario no interactuará con el dispositivo, solo verá imágenes para atender a la estimulación visual y el seguimiento ocular.

Seguidamente, se incorpora la fase de causa-efecto, donde el usuario detectará que los movimientos oculares se corresponden con el movimiento de la pantalla y, a su vez, verá que desencadenan hechos concretos.

A continuación, se pasará a la etapa de comunicación, en la que se le enseñará a la persona el valor comunicativo a través de habilidades de nivel cognitivo.

Posteriormente, se pasará al acceso al ordenador que, tras mucha práctica el usuario habrá conseguido escribir con el comunicador teniendo total capacidad para llevar a cabo las distintas tareas en el ordenador.

Finalmente, la última etapa será la inclusión total y empleo, donde se le ha mostrado a la persona que es capaz de realizar cualquier actividad con los ojos, a la vez que se encontrará en un nivel de ilusión y motivación máxima.



Figura 9: Sistema IrisBond



Figura 8: Logo IrisBond

Este sistema cuenta con una pantalla integrada a la silla de ruedas gracias a un brazo articulado, y con un sistema IrisBond Duo conectado a la tablet para que capte el movimiento de los ojos.

La ventaja con la que cuenta este dispositivo es que permite al usuario el manejo del ordenador para realizar tareas cotidianas, a la vez que le permite trabajar de manera autónoma y comunicarse, en caso de necesitar esa función.

El aspecto negativo de este dispositivo se halla en la interacción entre la pantalla y la luz solar tal como se ha explicado anteriormente.

5.3.2. Silla de ruedas controlada por la lengua

En 2005, la Universidad de Georgia Tech comenzó a trabajar en el proyecto de crear una silla de ruedas controlada por la lengua [8], de forma que el sistema fuese más rápido e intuitivo.

El sistema consiste en un pequeño piercing magnético que se coloca en la lengua junto a unos auriculares que leen los movimientos dentro de la boca. El imán se fija en la lengua y se comunica con el auricular gracias a unos sensores magnéticos que se encuentran a ambos lados de la cara. En caso de que el usuario desee girar hacia la izquierda, solo deberá mover la lengua hacia la mejilla derecha para comunicarse con el auricular.



Figura 10: Empresa creadora

En un principio, el imán iba pegado a la lengua, pero después de pocas horas este se caía haciendo peligrar la vida de los usuarios puesto que podrían llegar a tragárselo. Por esta razón, Anne Laumann profesora de dermatología en Feinberg e investigadora principal del ensayo Northwestern, propuso la idea de perforar la lengua de manera que el imán fuese un piercing en la lengua.

La decisión de utilizar este tipo de sensores para mover la silla de ruedas, aparece como solución al hecho de que la mayoría de personas que sufre tetraplejía cuenta con el control de la lengua, músculo que se convierte en el joystick de la silla. Lo que se explica porque el tamaño de la corteza motora dedicado a la lengua y la boca en el cerebro es similar en extensión al territorio de los dedos y la mano, lo que le confiere un control motor muy sofisticado y una gran capacidad de manipulación.

A su vez, la lengua está controlada por un nervio craneal que suele escapar a los daños de lesiones graves de la médula espinal o enfermedades neuromusculares. De igual manera, sus músculos son muy resistentes a la fatiga, por lo que puede moverse rápidamente y con precisión dentro de la boca de forma intuitiva.

Gracias a este dispositivo, las personas con minusvalías graves pueden volver a tener una vida activa, independiente y productiva, es decir, pasar a tener una vida normal.

Durante la investigación se llevaron a cabo pruebas con personas sin ningún tipo de problemas y personas con diversos tipos de discapacidad y todos notaron la gran diferencia entre las sillas de ruedas normales y esta nueva con el Tongue Driver System. Se concluye pues, que estos resultados se deben a que se trata de un sistema altamente intuitivo.

Actualmente, el programa reconoce seis comandos diferentes dependiendo de la posición de la lengua, pero se podría entrenar de manera que el sistema reconozca cada diente como un comando diferente o incluso, que cada usuario tenga una serie de movimientos de lengua específicos.

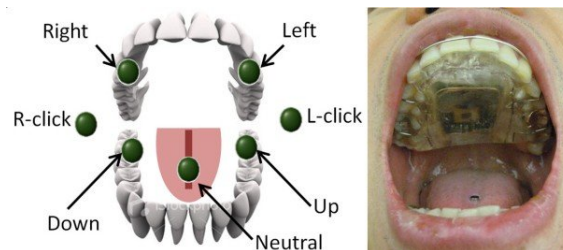


Figura 11: Tongue Driver System

5.3.3. Sillas de ruedas con Inteligencia Artificial

La empresa brasileña Hoobox Robotics [9] ha colaborado con Intel para realizar un kit adaptador que permite que casi cualquier silla de ruedas eléctrica sea controlada por las expresiones faciales del usuario, de manera que se aprovechen las habilidades del mismo y se contrarresten sus limitaciones.

Para llevar a cabo este proyecto, se utiliza un ordenador Intel RealSense para capturar y mapear en 3D las diferentes expresiones faciales y un mini ordenador en la silla con procesador Intel Core para ayudar al dispositivo a interpretar las señales que le llegan. El proceso que se realiza tiene un retraso de cien milisegundos, lo que hace que el procedimiento sea en tiempo real, gracias al kit Intel OpenVINO, Inferencia visual y optimización de redes neuronales.

Al contrario de muchas de las soluciones de alta tecnología, Wheelie 7 no incluye ningún tipo de elemento intrusivo, proporcionando a los usuarios independencia y control sobre su ubicación.

Wheelie 7, denominado de esta forma porque tan solo se tarda 7 minutos en instalarlo, es capaz de detectar hasta diez expresiones faciales, de las cuales solo necesita como mínimo cinco para poder empezar a funcionar y detecta a su vez, diez niveles de detección de dolor.



Figura 12: Logo empresa Hoobox One



Figura 13: Silla con mecanismo Wheelie 7

6. Sensores OpenBCI Ganglion

Los sensores de bajo coste que se han utilizado para la realización de este proyecto han sido, como se ha comentado anteriormente el kit de OpenBCI Ganglion [10].

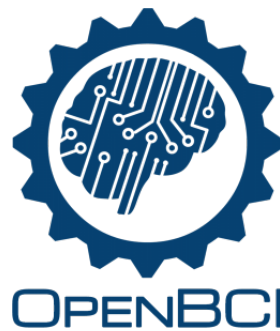


Figura 14: Marca de los sensores

6.1. Características del Hardware

El conjunto de sensores con el que se ha trabajado cuenta con una placa Ganglion [11] de cuatro canales, cinco cables electrodos a presión EMG/ECG, dos cables con electrodos de copa de oro, una banda para la cabeza donde se colocan los distintos sensores, dos pinzas sensoriales para las orejas y varios repuestos para los electrodos.

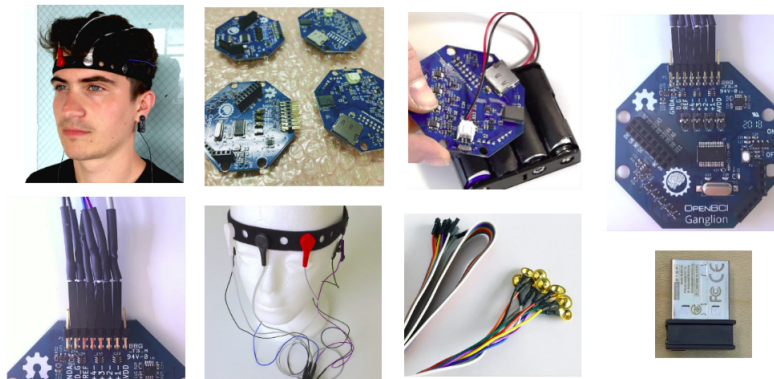


Figura 15: Kit de sensores OpenBCI

A continuación, se describe el hardware de la placa y los sensores, cómo funcionan y cómo se conectan. En primer lugar, la placa Ganglion es un dispositivo de bio-detección compatible con el software de código abierto de OpenBCI. En la parte de la entrada se encuentran cuatro entradas diferenciales de alta impedancia, una entrada a tierra, un suministro de voltaje positivo y otro de voltaje negativo.

La alimentación de la placa Ganglion se lleva a cabo a través de una batería de 3.3V, cuenta con un módulo de radio Simblee BLE, una ranura para tarjeta microSD, la cual se utiliza para poder actualizar la placa Ganglion, e interruptores para conectar o desconectar manualmente las entradas al pin REF.

Las entradas se pueden utilizar como entradas diferenciales individuales para medir EMG o ECG, o se pueden conectar individualmente a un electrodo de referencia para medir EEG.

Los datos que recoge esta placa se muestrean a 200Hz en cada uno de los cuatro canales de los que se dispone.

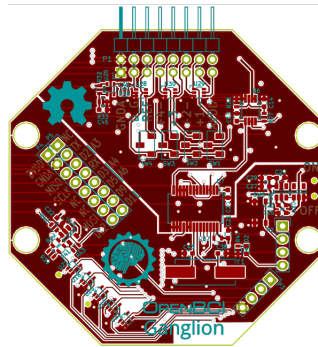


Figura 16: Placa Ganglion

6.2. Programa OpenBCI_GUI

Seguidamente, se va a explicar el programa que se utiliza para conectar la placa con los sensores y en qué consiste el proceso para recoger los datos. El programa que se ha utilizado es el que proporciona la propia plataforma, OpenBCI_GUI que se puede descargar desde la página oficial de la misma.

Una vez se ha instalado, se ejecuta y aparece la siguiente imagen, se selecciona la placa que se va a utilizar, en este caso Ganglion, se escoge el modo de conexión y, finalmente, la placa con la que se desee conectar.

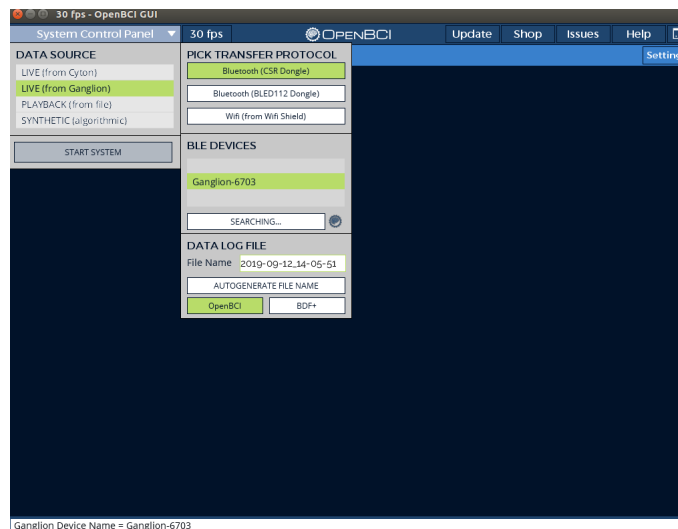


Figura 17: Pantalla inicial OpenBCI_GUI

Una vez se han seleccionado las opciones necesarias para llevar a cabo la conexión, se abrirá automáticamente la interfaz desde la cual se va a trabajar, donde se muestran cuatro ventanas que se pueden cambiar a conveniencia del usuario. En este caso, se han colocado en la primera de ellas las ondas producidas por el cerebro que van captando los diferentes sensores. En la segunda pantalla, se ha colocado la Transformada de Fourier (FFT), es decir, las ondas medidas en frecuencia y amplitud. La tercera ventana muestra la señal de la placa Ganglion y, por último, la cuarta ventana que muestra el acelerómetro.

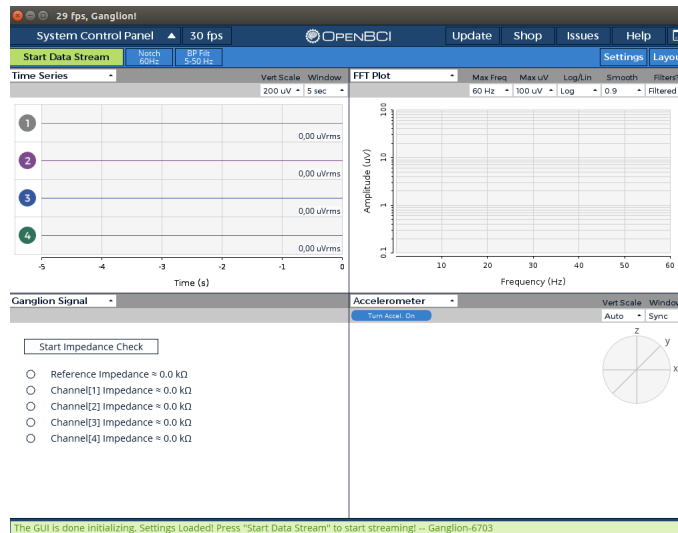


Figura 18: Entorno de trabajo OpenBCI.GUI

A continuación, cuando ya se tienen las ventanas colocadas como se desea, se elige la opción “Start Data Stream” y se lleva a cabo el análisis de las ondas. Esto haría que se viesen, en las distintas pantallas, las ondas producidas con los gestos de la cara, originando una imagen parecida a la siguiente.

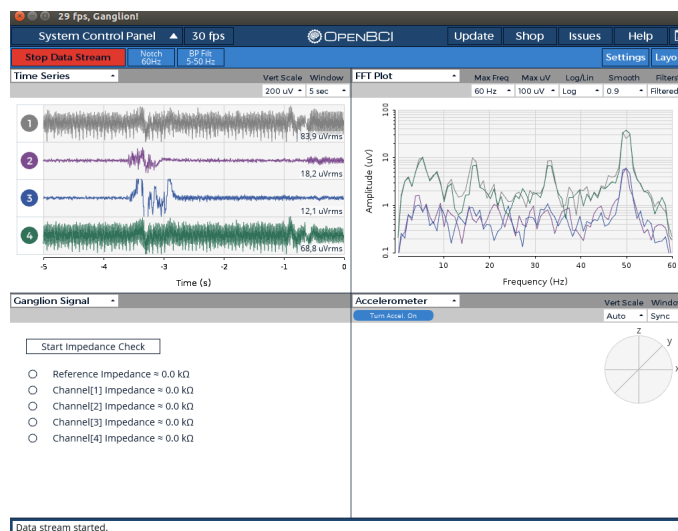


Figura 19: Producción de ondas en OpenBCI.GUI

Finalmente, cuando se ha terminado el trabajo con los sensores, se selecciona a “Stop Data Stream”, se cierra la ventana y en la carpeta de SavedData se crea un fichero de extensión txt que contiene los datos de las ondas que se han generado para poder analizarlos posteriormente.

Para concluir con este apartado, se nombrarán, a continuación, las diferentes librerías con las que se han trabajado en este proyecto. Por un lado, se encuentran las que han utilizado por defecto los propios sensores OpenBCI y, por otro lado, las que se han utilizado externamente para realizar lo deseado. A su misma vez, se nombrarán las páginas de ayuda de la página oficial.

6.3. Librerías utilizadas

Las librerías que se han empleado han sido las que se encuentran en la página de GitHub de OpenBCI [12]. Entre ellas, la librería principal es OpenBCI.GUI que se ha utilizado para captar los datos como se comentó con anterioridad. Esta librería ofrece información acerca de la interfaz gráfica de usuario de manera que se puedan visualizar todos los datos de estos dispositivos, sin importar tanto el tipo de sensor que se utilice, como el sistema operativo elegido.

Otra de las librerías utilizadas ha sido OpenBCI.Hub, se trata del software con el que se conecta el usuario a través del ordenador a la placa OpenBCI empleando el modelo TCP/IP. Dicha librería funciona como un servidor que se encuentra continuamente escuchando en el puerto definido por defecto, para que la comunicación entre placa y programa sea correcta. De esta manera, si esta librería no se encuentra instalada o no está actualizada, la conexión no es posible y, por tanto, no se puede llevar a cabo la transmisión de datos.

Estas librerías al instalarse por defecto, como se indica desde la página oficial de los sensores, no han de descargarse desde la página de GitHub. En caso de haber un error con las versiones o se necesite instalar alguna más actualizada o una versión anterior por cualquier razón, se encontrarían todas en la página oficial de GitHub.

Para finalizar, se nombrarán las páginas de ayuda que se han utilizado para resolver los distintos problemas que han ido surgiendo, entre ellas se encuentran, tanto el apartado de Preguntas Frecuentes de la página oficial de OpenBCI [13], como el Foro [14].

Por otro lado, se encuentra la página de problemas de GitHub [15] dentro del perfil de OpenBCI, en donde se puede encontrar la mayoría de los problemas que se dieron a la hora de instalar o de acceder al programa.

La última página de ayuda a la se ha accedido en caso de no tener respuesta en las dos anteriores, era una bastante conocida dentro del ámbito del programador, StackOverflow [16].

7. Captura de datos

En esta sección, se va a comentar cómo se han capturado los datos. Esta captación se ha realizado con dos plataformas diferentes. La primera de ellas ha sido el programa de los sensores OpenBCI_GUI, programa que se utilizó para tomar los datos de las ondas del cerebro a través de los captadores de ondas. La segunda plataforma utilizada fue Octave, programa que analiza, a través del código realizado, los datos capturados a través de los ficheros de tipo txt que se le han facilitado.

7.1. Captura de datos con OpenBCI

Para poder llevar a cabo el estudio de ondas y comenzar a trabajar en el proyecto, lo primero que se hizo fue la captura de datos con el programa de los sensores, OpenBCI_GUI. El funcionamiento de este programa se explicó en el apartado anterior y tal como se dijo, la captura de datos en formato txt se efectúa a la misma vez que se producen las ondas, de esta forma no hay que preocuparse por nada que no sea que el usuario realice el gesto correspondiente de la mejor forma posible.

Una vez se ha parado el programa, se puede abrir el fichero txt, que se localiza en la carpeta de SavedData. Dicho fichero se divide en dos secciones, la primera de ellas se encuentra en las primeras seis líneas, en donde, a través de comentarios se explica el tipo de sensores OpenBCI que se está utilizando, el número de canales que tiene, el ratio y, por último, los datos que se van a encontrar seguidamente.

```
%OpenBCI Raw EEG Data
%Number of channels = 4
%Sample Rate = 200.0 Hz
%First Column = SampleIndex
%Last Column = Timestamp
%Other Columns = EEG data in microvolts followed by Accel Data (in G) interleaved with Aux Data
```

Figura 20: Fichero de datos capturados

Posteriormente, se encuentra la segunda sección, que pasaría a ser la parte realmente importante, pues se trata del procesamiento de datos de las ondas capturadas. De este modo, cada columna indica datos diferentes.

La primera columna muestra el número de los datos que se encuentran a su derecha, de manera que va aumentando de uno en uno, siendo así la cantidad de datos tomados durante el periodo de tiempo que se han utilizado los sensores.

Desde la segunda columna hasta la quinta se encuentran los datos recogidos por cada uno de los sensores, de forma que la segunda columna se corresponde con los datos recogidos por el primer sensor, la tercera columna, por el segundo, etc.

A posteriori, se encuentran tres columnas que nos dan información sobre los ejes x, y, z, las cuales suelen estar a 0, debido a que no se está captando dicha información y, finalmente, las dos últimas columnas hacen referencia al tiempo y a la fecha.

```
0, -90.60, -167.00, -103.37, -39.90, 0.000, 0.000, 0.000, 16:20:56.326, 1561389656326
1, 315.18, -100.55, 293.90, 360.03, 0.000, 0.000, 0.000, 16:20:56.342, 1561389656342
2, -37.98, -157.49, -58.34, -87.21, 0.000, 0.000, 0.000, 16:20:56.342, 1561389656342
3, -481.83, -219.32, -487.28, -511.46, 0.000, 0.000, 0.000, 16:20:56.355, 1561389656355
4, -96.92, -171.59, -105.77, -38.17, 0.000, 0.000, 0.000, 16:20:56.355, 1561389656355
5, 317.47, -98.74, 301.82, 366.34, 0.000, 0.000, 0.000, 16:20:56.356, 1561389656356
6, -33.68, -132.25, -49.15, -74.34, 0.000, 0.000, 0.000, 16:20:56.356, 1561389656356
7, -474.19, -205.51, -482.10, -506.48, 0.000, 0.000, 0.000, 16:20:56.370, 1561389656370
```

Figura 21: Fichero de datos capturados

8. Descripción del trabajo

En el siguiente apartado se va a describir el proceso que se ha llevado a cabo para realizar el proyecto, comentando los pasos que se han realizado, los problemas que se han encontrado y los distintos resultados que se han obtenido.

8.1. Primeros pasos

Lo primero que se va a comentar es el proceso que se realizó para conectar las sensores con el ordenador de manera que se pudiesen empezar a captar ondas y, por tanto, comenzar a trabajar en el proyecto.

En primer lugar, se conectó con los sensores para poder captar las ondas que se estudiaron más adelante. En esta primera parte, se encontraron diferentes problemas a la hora de enlazar el ordenador con la placa Ganglion. Esto era debido a que se localizaron errores en el código, teniendo que buscar en GitHub versiones anteriores que no hubiesen suprimido algunas partes o que funcionasen de manera correcta.

Por otro lado, a la hora de conectar el adaptador bluetooth para que uniese placa y ordenador se dieron ciertos problemas, puesto que en el sistema operativo Linux, se instalaba de otra manera diferente a la que se realizaba en el sistema operativo de Windows. No se encontró la solución de manera rápida ya que las dudas que se encontraban en internet eran sobre todo con respecto a Windows o Mac, de forma que conseguir la conexión desde Linux hacia la placa, se convirtió en un proceso tedioso.

Durante el comienzo del proyecto, se intentó conectar la placa con el ordenador a través del programa OpenBCIPython, el cual funcionaba a través de la terminal de Linux, para comprobar si de esta forma se conectaba mejor y así no necesitar el programa GUI para captar las ondas. Dado que la conexión no se estaba consiguiendo por la falta de conexión del bluetooth, una vez se arregló este problema, el GUI funcionaba correctamente y era más sencillo e intuitivo que la terminal.

8.2. Procesamiento con Octave

A continuación, se va a explicar detenidamente el proceso que se ha llevado a cabo una vez se obtienen las ondas. De forma que estas sean procesadas a través del programa Octave, para conseguir la información que se desea.

En el momento en que se obtuvieron las ondas, se pasó a la utilización del programa Octave, el cual se empleó como herramienta para procesar y analizar las ondas cerebrales.

Este programa se utiliza para llevar a cabo la manipulación de matrices, representación de datos y funciones, implementación de algoritmos y la creación de interfaces de usuario.

En primer lugar, se escribió un código sencillo de forma que se viesen las ondas en frecuencia y en tiempo, dado que eran las variables que se iban a utilizar para llevar a cabo el estudio de las ondas y para poder desarrollar el programa deseado.

En un primer momento se observó que las ondas tenían una frecuencia bastante alta como consecuencia de que el ordenador, cuando se tomaron los datos, estaba conectado a la red eléctrica. Por esta razón, se volvieron a captar más ondas con la placa Ganglion, esta vez con el ordenador desconectado de la red eléctrica.

A su vez, cabe destacar que en la figura 22 se pueden apreciar todas las señales sin filtrar o procesar.

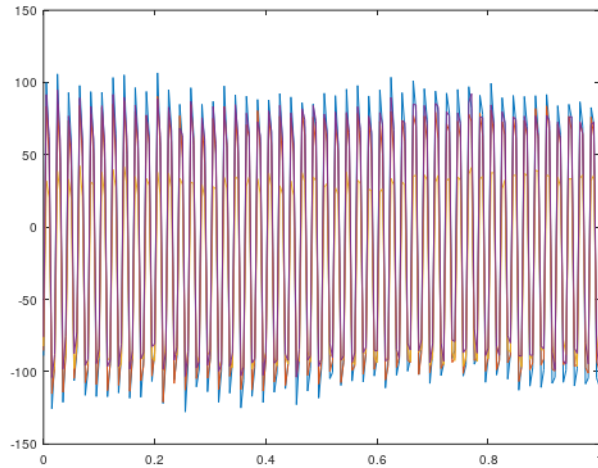


Figura 22: Señal montada en la red eléctrica

A continuación, la señal se filtró con un filtro pasa baja, de forma que eliminase los 50Hz de la red eléctrica, dejando pasar todo hasta aproximadamente 30Hz, valor en el que el filtro corta la señal.

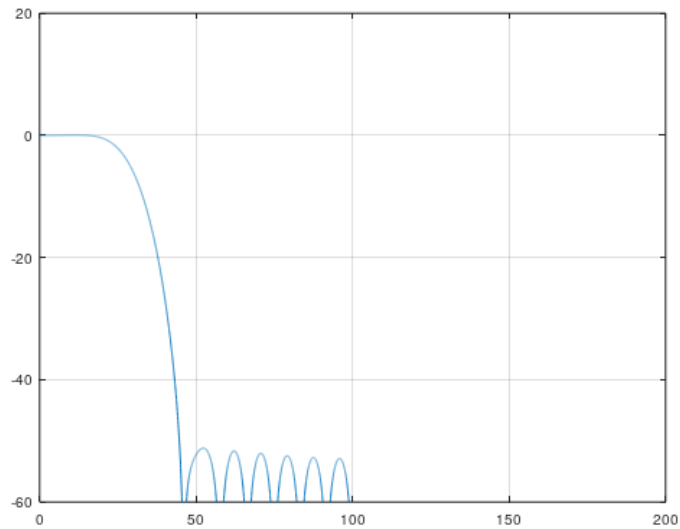


Figura 23: Filtro pasa baja

En la figura 24 se representa de forma gráfica la señal una vez ha sido filtrada en forma de ondas que produce cada uno de los sensores. Se puede distinguir que los sensores representados con los colores azul y violeta, son los que mayor amplitud tienen, por lo que son los sensores que captan más ondas al efectuar el movimiento. Por otro lado, los sensores representados con los colores amarillo y rojo, siguen el mismo estilo de onda que los dos anteriores, pero con una amplitud menor.

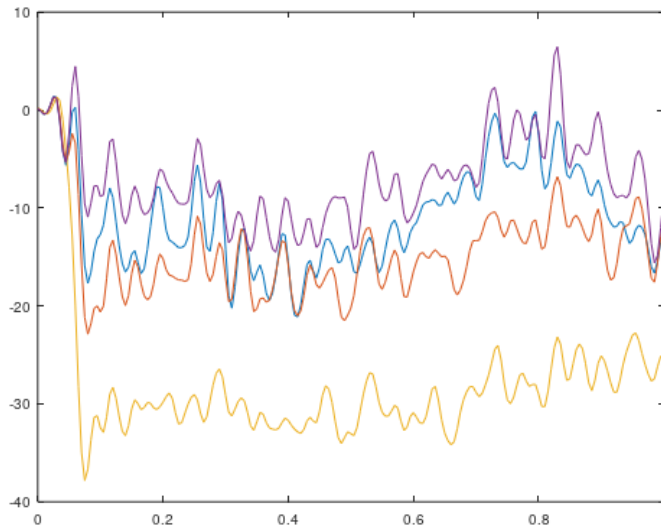


Figura 24: Señales de los sensores

Después de realizar varios cambios en el código a modo de perfeccionamiento en cuanto a frecuencia y visualización de las imágenes, se pasó al análisis de las ondas, es decir, a crear un código que, a partir de unas normas, identificase el gesto a través de herramientas como la FFT (Transformada de Fourier), amplitud, etc.

Lo primero que se hizo fue crear un código que, después de filtrar las señales tal y como se comentó anteriormente, aplicara la FFT para que la señal se transformara desde el dominio del tiempo al dominio de la frecuencia o viceversa. Por último, se calculó la energía que producían las ondas, produciendo unas imágenes como las que pueden observar en la figura 25.

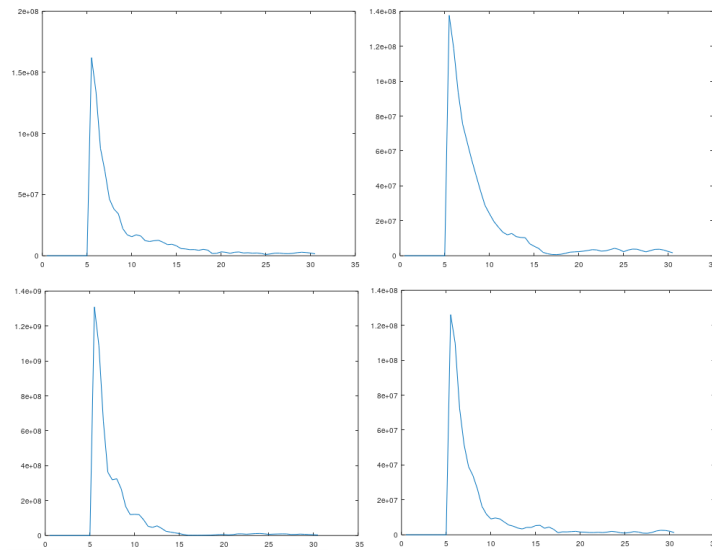


Figura 25: Eje X: frecuencia en Hz, Eje Y: energía de la frecuencia en voltios

8.3. Detección de gestos

En este apartado, se va a describir el proceso que se ha llevado a cabo para relacionar los gestos efectuados por el usuario con las ondas captadas por los sensores, tal como se ha explicado anteriormente.

Tras aplicar la FFT a las ondas se calcula la mediana, valor que representa la suma de todos los datos divididos entre número de valores que se tienen, obteniendo una cifra media para caracterizar las nuevas medidas.

De esta forma, tras aplicar la mediana a todos los ficheros de datos por cada gesto, se alcanzan valores cercanos entre sí, con los que se puede definir una banda de frecuencia y así caracterizar las medidas, es decir, determinar un gesto, en caso de que los datos se encuentren dentro de esta banda, por sus frecuencias.

Tras aplicar la mediana, se consigue para cada uno de los gestos y sensores un único punto, figura 27, mientras que, cuando no se le aplicaba la mediana, se obtenían múltiples puntos por sensor, figura 26.

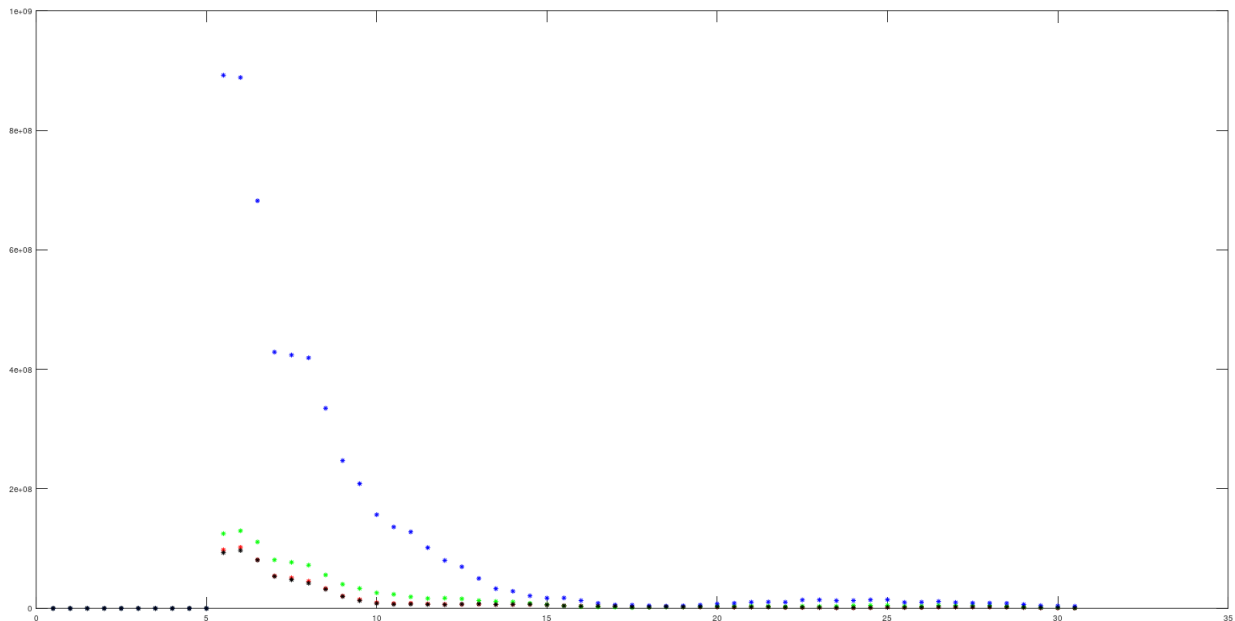


Figura 26: Señal sin aplicar la mediana

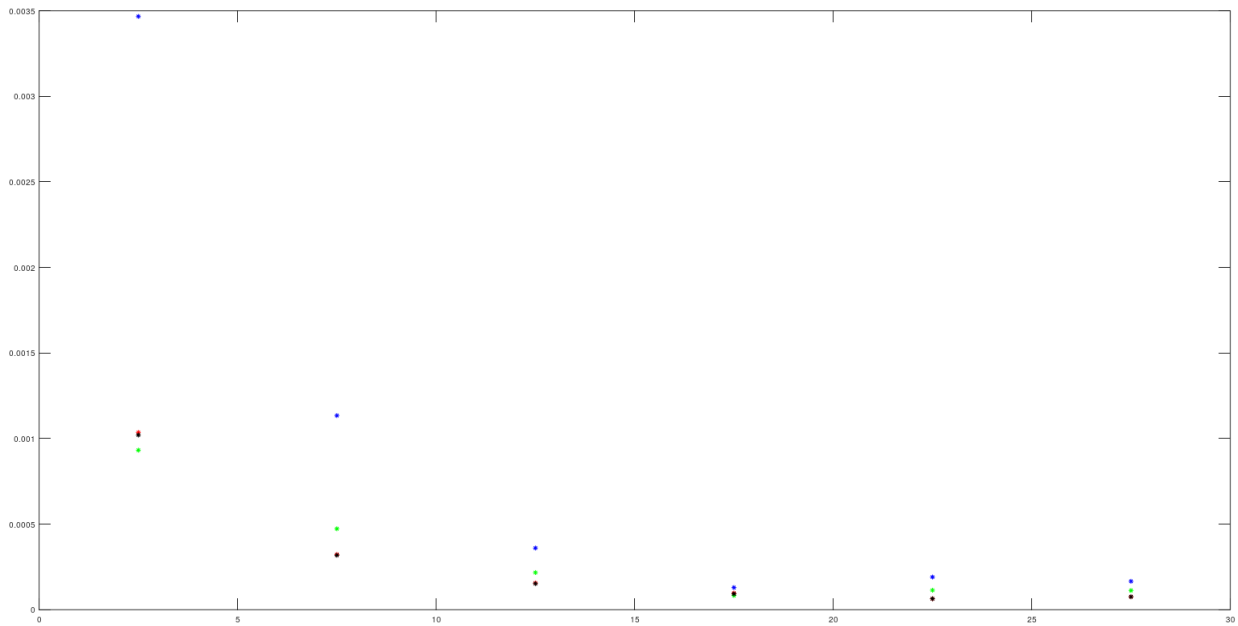


Figura 27: Señal tras aplicar la mediana

De esta manera, al pasar de tener diferentes valores por sensor a tener un solo valor, es posible definir un rango de frecuencias teniendo en cuenta los puntos de cada sensor por cada fichero de gestos. De este modo, al considerar todos los valores obtenidos con la mediana se caracterizan las señales, es decir, en caso de que la energía de la señal se encuentre dentro del rango, entonces se podría presuponer que se trata de la misma señal.

En la figura 28 se pueden ver cinco imágenes, cada una de ellas representa un fichero de datos del mismo gesto y en cada imagen se encuentran las señales de energía que producen los distintos sensores. En esta ocasión se trata del análisis del gesto de cerrar el ojo derecho, por lo que se puede ver que el sensor representado con el color azul es el que más energía tiene, pudiendo presuponer que es el que se ha colocado encima de la ceja derecha o en el lado derecho de la cabeza.

En este caso, para seleccionar correctamente el gesto, se define un rango de valores para cada uno de los sensores. Para los sensores 2, 3 y 4, el rango de valores tiene unos límites más pequeños que los que se alcanzan con sensor 1. A su vez, la amplitud del sensor 1 es bastante más amplia que la de los sensores 2, 3 y 4, dado que es el que más energía tiene.

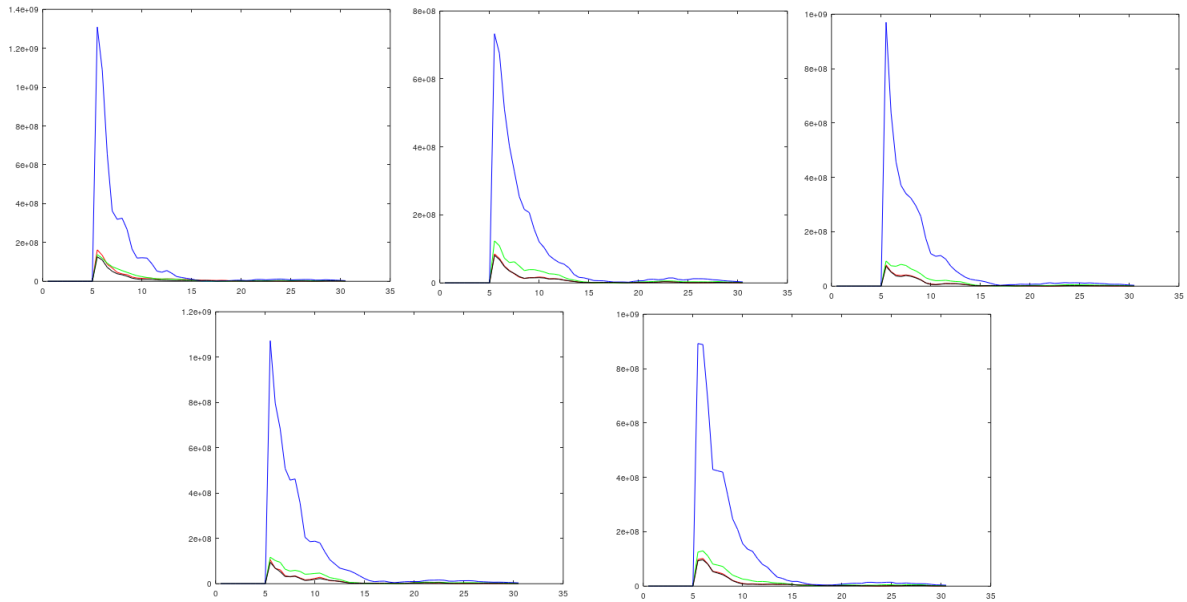


Figura 28: Señales de energía para el gesto derecha

Una vez se tuvo la mediana para cada uno de los gestos y de los sensores, se normalizaron los datos. La normalización [17] se basa en ajustar los valores medidos en diferentes escalas respecto a una escala común, de esta forma, si hubiera un valor que fuese mucho más pequeño o mucho más grande que los que se encontraban dentro del rango, el programa le cambiaría el valor quedando dentro de los valores y no haciendo que el rango tuviese que ser más grande, pudiendo reconocer así un gesto que no es el correcto.

En este caso, la normalización se ha realizado para tratar de reducir el efecto de la diferencia de amplitud en las señales que se puede dar por la colocación de los sensores.

Lo siguiente que se hizo fue calcular unos valores de máximo y mínimo, es decir, establecer numéricamente el rango de frecuencias en el que se encontraba cada uno de los gestos. De esta forma, cuando se analiza un fichero, se estudian los valores de los diferentes sensores, en caso de que todos den positivo, se determina el gesto, obteniendo así un análisis correcto.

```

if ( n==1 )
    if ( ( 0.0013757 <= acumulada_0_5 >= 0.0034679 ) && ( 3.1966e-04 <= acumulada_5_10 >= 4.5563e-04 )
        && ( 1.1695e-04 <= acumulada_10_15 >= 1.9304e-04 ) && ( 7.1717e-05 <= acumulada_15_20 >= 1.0636e-04 )
        && ( 4.5410e-05 <= acumulada_20_25 >= 8.9153e-05 ) && ( 3.1991e-05 <= acumulada_25_30 >= 7.9118e-05 ) )
        sensor1 = 1;
    endif
endif

```

Figura 29: Sentencia if para hallar los rangos

Durante la realización de las comprobaciones, se dieron ciertos problemas con los que se estuvo trabajando la mayor parte del tiempo.

En primer lugar, a la hora de analizar los valores de cada uno de los sensores frente al rango asignado, aun siendo los mismos datos que se habían utilizado para determinarlos, el programa no los identificaba correctamente. Por esta razón, lo primero que se hizo fue dividir la sentencia if, de forma que, los valores del rango quedasen divididos de la siguiente manera.

```

if ( n==1 )
    if ((( 0.0013757 <= acumulada_0_5) && ( acumulada_0_5 >= 0.0034679 )) && (( 3.1966e-04 <= acumulada_5_10) && (acumulada_5_10 >= 4.5563e-04 ))
        && (( 1.1695e-04 <= acumulada_10_15) && ( acumulada_10_15 >= 1.9304e-04 )) && (( 7.1717e-05 <= acumulada_15_20) && ( acumulada_15_20 >= 1.0636e-04 ))
        && (( 4.5410e-05 <= acumulada_20_25) && (acumulada_20_25 >= 8.9153e-05 )) && (( 3.1991e-05 <= acumulada_25_30) && (acumulada_25_30 >= 7.9118e-05 ))
            sensor1 = 1;
    endif
endif

```

Figura 30: División sentencia if para hallar los rangos

Seguidamente, se probó a poner los rangos con menos decimales, de manera que los extremos fuesen más abiertos y que de esa forma cogiese los valores, sin importar el resto de decimales que tuviesen.

```

if ( n==1 )
    if ((( 0.0013 <= acumulada_0_5) && ( acumulada_0_5 >= 0.0034 )) && (( 3.19e-04 <= acumulada_5_10) && (acumulada_5_10 >= 4.55e-04 ))
        && (( 1.16e-04 <= acumulada_10_15) && ( acumulada_10_15 >= 1.93e-04 )) && (( 7.17e-05 <= acumulada_15_20) && ( acumulada_15_20 >= 1.06e-04 ))
        && (( 4.54e-05 <= acumulada_20_25) && (acumulada_20_25 >= 8.91e-05 )) && (( 3.19e-05 <= acumulada_25_30) && (acumulada_25_30 >= 7.91e-05 ))
            sensor1 = 1;
    endif
endif

```

Figura 31: Menor número de decimales para hallar los rangos

De igual forma, se seguían obteniendo resultados como los que se pueden apreciar en la figura 30, es decir, no se estaban analizando debidamente los gestos y por tanto, no se estaba consiguiendo el resultado esperado.

```

sensor1 = 0
sensor2 = 0
sensor3 = 0
sensor4 = 0
ans = El gesto que ha realizado NO ha sido: CERRAR EL OJO DERECHO

```

Figura 32: Resultados

Dado que ninguna de estas soluciones daba resultado, se probó a escribir un código que calculase los valores máximos y mínimos que se daban para cada uno de los gestos. Este cálculo se realizó con los cinco ficheros de datos que se crearon para llevar a cabo el estudio.

```

# name: Maximos
# type: matrix
# rows: 4
# columns: 6
0.0013572162716540691 0.00035541510119075476 0.00015630941091155433 0.00010090193006605334 8.4170650694611476e-05 7.5484359468993754e-05
0.0011034445123343202 0.00047301692906878988 0.00021706219818783256 8.2980444410351071e-05 0.00012217302853465212 0.00011222607782645385
0.004316347247653389 0.0011345389778215473 0.00040660017112761307 0.00015327535058596507 0.00019182823837142046 0.00017064837328101794
0.0012952856219528882 0.00033462617775902715 0.00015216137507091443 9.2428961931082264e-05 8.7965748065328475e-05 7.5622866227337962e-05

# name: Minimos
# type: matrix
# rows: 4
# columns: 6
0.0010110359779387731 0.0002592303309111283 0.00013498941246798464 5.6415695430604915e-05 5.8867135750037565e-05 5.9898831829124348e-05
0.00092875510605583671 0.00038366862872044613 0.00018048866582758265 4.9444764186197933e-05 9.4986882378929175e-05 9.4107259157709329e-05
0.0033738144502760258 0.00089919767197252484 0.00029892577786301414 7.4634307718607056e-05 0.00015278589416702187 0.00014448731508089123
0.00097972579130434592 0.00027310623440743179 0.00011525612438794541 6.0758883222529398e-05 5.3821736327670927e-05 6.0593554799306296e-05

```

Figura 33: Valores máximos y mínimos

Una vez se tuvieron todos los valores correctamente calculados, se pasó a comprobar que el código analizase adecuadamente los gestos. Puesto que el problema persistía, lo que se hizo fue ampliar el rango. De esta forma, el valor mínimo se multiplicó por 0.9, disminuyendo así el límite inferior y el valor máximo se multiplicó por 1.1 aumentando el límite superior. De esta manera el rango se amplió por ambos extremos un 10 %.

Gracias a haber ampliado el rango, los valores empezaron a ser más positivos en cuanto al análisis, por lo que se decidió establecer así los límites.

A continuación, tras observar gráficamente que la señal dejaba de ser plana únicamente en un intervalo concreto de datos, figura 34, se decidió aplicar el código solo a esos valores, de manera que, cuando se fuese a procesar el gesto, solo se tuviesen en cuenta los datos en los que se producía.

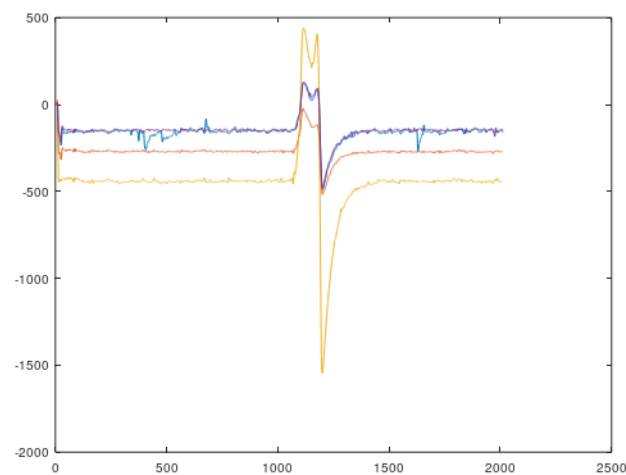


Figura 34: Grafica de datos a analizar

Para realizar correctamente el procesamiento, se calculó el rango de valores para cada uno de los gestos de forma precisa. Para ello, se fue cogiendo cada una de las gráficas que se producen al calcular el máximo y el mínimo y se fueron acotando los ejes. De esta manera, se pasó de tener una gráfica de casi 2000 datos a tener una gráfica de entre 70 y 80 datos.

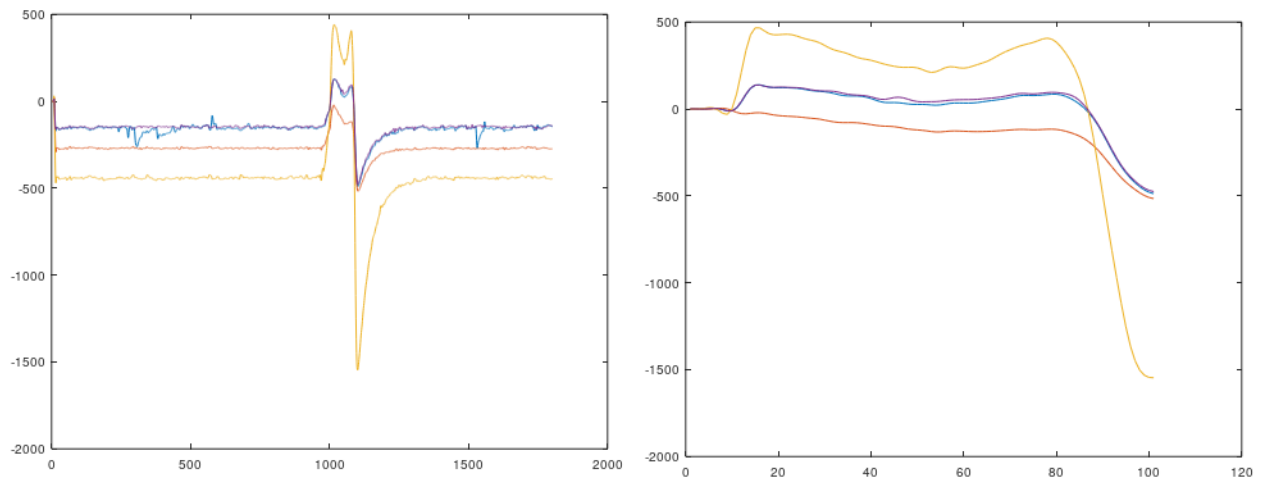


Figura 35: Diferencia de gráficas

Lo siguiente que se realizó fue un nuevo código que, a partir de los resultados obtenidos al calcular el máximo y el mínimo, procesase los datos de los gestos a través de una ventana deslizante. Esta ventana deslizante, lo que hace es coger una primera muestra del tamaño del rango calculado y procesarlos utilizando la lógica. [18]

Se toman los primeros 70 datos, suponiendo que el rango es de dicha anchura, y se guarda esa información en la memoria. A continuación, pasa a la siguiente muestra abarcando esta desde el dato 2 hasta el 71. La ventana deslizante, lo que hace es analizar los datos de uno en uno pero con la anchura indicada, de esta manera no va avanzando de 70 en 70, sino que se va avanzando según se van analizando los datos.

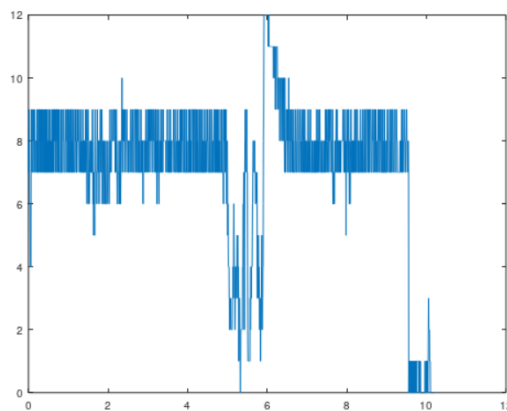


Figura 36: Procesamiento de datos. Índices: 0.9 - 1.1

Para llevar a cabo la Figura 36, se utilizó la siguiente tabla para determinar si un gesto era correcto o si, por el contrario, se trataba de una señal diferente.

Dicha tabla se crea al dividir el número de datos de un gesto frente a las bandas necesarias para analizarlo.

Sensor	0 - 0.7	0.7 - 15.4	15.4 - 23.1
Sensor1	1	1	1
Sensor2	1	1	1
Sensor3	1	1	1
Sensor4	1	1	1

Cuadro 1: Tabla de análisis del gesto

Una vez se tienen todos los datos, se pasa a procesar todos los gestos, de modo que el procedimiento que se sigue es el siguiente. Se calcula el rango de máximos y mínimos del gesto derecha utilizando los cinco ficheros de datos que se han capturado y se procesa para tener una gráfica como la vista en la figura 36. Este procesamiento se hará con los mismos datos que se utilizaron para calcular el rango. De forma que por cada fichero de datos, el resultado obtenido sea un 12.

Dicho valor se obtiene de la siguiente manera: por un lado se encuentran los cuatro sensores, que se muestran en la tabla 1, en la columna de la izquierda y, por otro lado, las bandas para analizarlos, que figuran en las columnas siguientes. En caso de que el valor de cada sensor se encuentre dentro del rango de las bandas correspondientes, se representará con el valor 1, de manera que si los valores de los sensores figuran en los rangos propuestos, el valor total es de 12, lo que se traduce como un análisis exitoso.

Dado que no se daban cinco 12 con ningún gesto, se decidió utilizar la librería de outlier para comprobar que el problema no se debía a un valor atípico, es decir, un valor que se encuentra numéricamente distante al resto. De esta forma, lo que se hace es tomar el dato atípico y cambiarle el valor a uno más cercano al resto, como por ejemplo, un valor medio. Gracias a esta librería, el valor resultante de la mediana se encuentra en el núcleo de datos y no en un resultado más cercano al valor atípico. Sin embargo, a pesar de que la solución mejora, no alcanza el objetivo perseguido debido a que no se obtienen los cinco 12 necesarios para determinar correctamente el gesto.

8.4. Pruebas y resultados

Finalmente, se explicarán las pruebas que se realizaron junto con los pequeños cambios que se desarrollaron para obtener unos resultados correctos. Seguidamente, se interpretarán las soluciones obtenidas presentandolas de forma gráfica.

Por último, se analizan los siguientes gestos con los valores de máximo y mínimo de derecha para poder obtener una tabla de falsos positivos. Por lo que, gracias a dichas tablas, se puede saber si el gesto está siendo analizado correctamente o no.

Para que un gesto esté bien analizado, el valor que debe aparecer es 12 en el mismo gesto que se esté estudiando y un valor menor a 12 en el resto de gestos.

En la figura 37 se puede observar una tabla de falsos positivos, esta tabla se utiliza para analizar en profundidad el gesto, de manera que si todos los resultados son correctos, se está analizando acertadamente el gesto. En cambio, si se encuentran valores incorrectos, entonces se estaría dando un falso positivo, es decir, un valor que a priori puede entenderse como correcto, pero que al obtener dicho resultado con un fichero de datos diferente al gesto que se está analizando, es incorrecto.

En la tabla de falsos positivos se representan diferentes datos, en la primera columna se encuentra el valor por el que se multiplican los límites de máximo y mínimo. Seguidamente, la segunda columna contiene el nombre del fichero que se va a analizar. La tercera columna representa el gesto y el valor que se ha obtenido al estudiar dicho gesto. Finalmente, en la cuarta columna se encuentra el resultado de manera que en la casilla se pondrá en rojo un "No" si el gesto está dando un resultado erróneo, mientras que la casilla verde "Sí" refleja que el resultado sea correcto.

Por esta razón, en las tablas que se ven en la figura 37, ningún gesto está siendo analizado correctamente, dado que los ficheros de derecha no dan todos 12 y los ficheros de otros gestos, en algunas ocasiones dan 12, por lo que se encuentra dentro de los límites de cerrar el ojo derecho y por tanto, se podría dar una equivocación a la hora de distinguir el gesto.

0.9 – 1.0	Gestos a reconocer / Ficheros	Derecha	¿Detectado Correctamente?	0.9 – 0.9	Gestos a reconocer / Ficheros	Derecha	¿Detectado Correctamente?	1.0 – 1.1	Gestos a reconocer / Ficheros	Derecha	¿Detectado Correctamente?
	CalculoMaxMin				CalculoMaxMin				CalculoMaxMin		
	Dcha1	11	No		Dcha1	11	No		Dcha1	12	Sí
	Dcha2	11	No		Dcha2	11	No		Dcha2	12	Sí
	Dcha3	11	No		Dcha3	11	No		Dcha3	12	Sí
	Dcha4	11	No		Dcha4	11	No		Dcha4	11	No
	Dcha5	11	No		Dcha5	11	No		Dcha5	12	Sí
	Izq1	10	Sí		Izq1	10	Sí		Izq1	9	Sí
	Izq2	10	Sí		Izq2	10	Sí		Izq2	10	Sí
	Izq3	11	Sí		Izq3	10	Sí		Izq3	10	Sí
	Izq4	11	Sí		Izq4	11	Sí		Izq4	10	Sí
	Izq5	11	Sí		Izq5	10	Sí		Izq5	10	Sí
	Subir1	12	No		Subir1	12	No		Subir1	11	Sí
	Subir2	12	No		Subir2	11	Sí		Subir2	9	Sí
	Subir3	11	Sí		Subir3	11	Sí		Subir3	10	Sí
	Subir4	11	Sí		Subir4	11	Sí		Subir4	9	Sí
	Subir5	10	Sí		Subir5	10	Sí		Subir5	9	Sí
	Bajar1	10	Sí		Bajar1	10	Sí		Bajar1	9	Sí
	Bajar2	10	Sí		Bajar2	10	Sí		Bajar2	9	Sí
	Bajar3	12	No		Bajar3	12	No		Bajar3	9	Sí
	Bajar4	10	Sí		Bajar4	10	Sí		Bajar4	9	Sí
	Bajar5	11	Sí		Bajar5	11	Sí		Bajar5	8	Sí
	ProcesaFich				ProcesaFich				ProcesaFich		
	Dcha1	11	No		Dcha1	10	No		Dcha1	9	No
	Dcha2	10	No		Dcha2	9	No		Dcha2	11	No
	Dcha3	11	No		Dcha3	11	No		Dcha3	11	No
	Dcha4	12	Sí		Dcha4	9	No		Dcha4	11	No
	Dcha5	11	No		Dcha5	10	No		Dcha5	11	No
	Izq1	11	Sí		Izq1	11	Sí		Izq1	9	Sí
	Izq2	11	Sí		Izq2	11	Sí		Izq2	11	Sí
	Izq3	11	Sí		Izq3	11	Sí		Izq3	11	Sí
	Izq4	10	Sí		Izq4	10	Sí		Izq4	9	Sí
	Izq5	9	Sí		Izq5	9	Sí		Izq5	9	Sí
	Subir1	10	Sí		Subir1	9	Sí		Subir1	10	Sí
	Subir2	10	Sí		Subir2	10	Sí		Subir2	9	Sí
	Subir3	10	Sí		Subir3	9	Sí		Subir3	8	Sí
	Subir4	10	Sí		Subir4	9	Sí		Subir4	8	Sí
	Subir5	9	Sí		Subir5	8	Sí		Subir5	8	Sí
	Bajar1	11	Sí		Bajar1	11	Sí		Bajar1	10	Sí
	Bajar2	11	Sí		Bajar2	11	Sí		Bajar2	11	Sí
	Bajar3	12	No		Bajar3	12	No		Bajar3	12	No
	Bajar4	12	No		Bajar4	12	No		Bajar4	12	No
	Bajar5	12	No		Bajar5	12	No		Bajar5	10	Sí

Figura 37: Tablas de procesamiento

Finalizadas las tablas de falsos positivos para todos los gestos y viendo que los resultados no eran los esperados, se optó por llevar a cabo diversas alternativas.

En primer lugar, algo que podría estar pasando era que, como los ficheros de datos que se utilizaban para calcular los rangos y los que se utilizaban para comprobar los gestos, se habían tomado en momentos diferentes, podría estar pasando que la colocación de los sensores en la cabeza y la conexión entre sensores y placa Ganglion no fuese la misma.

Por esta razón, se crearon diez ficheros de datos por cada gesto simultáneo, de manera que cada gesto contase con cinco ficheros para el cálculo de rangos y cinco ficheros para la comprobación del gesto.

En las siguientes figuras se puede ver que tanto de una manera como de otra, los resultados no son correctos. Igualmente, también se puede comprobar que cuando los ficheros se toman en diferentes días, los resultados son sensiblemente mejores.

Gestos a reconocer / Ficheros	Izquierda	¿Detectado Correctamente?
Fecha: 3-10		
Izq1	12	Si
Izq2	12	Si
Izq3	12	Si
Izq4	12	Si
Izq5	12	Si
Dcha1	12	No
Dcha2	12	No
Dcha3	12	No
Dcha4	12	No
Dcha5	12	No
Subir1	11	Si
Subir2	12	No
Subir3	11	Si
Subir4	12	No
Subir5	12	No
Bajar1	11	Si
Bajar2	12	No
Bajar3	12	No
Bajar4	11	Si
Bajar5	12	No
Fecha: 06-02		
Izq1	10	No
Izq2	11	No
Izq3	12	Si
Izq4	12	Si
Dcha1	11	Si
Dcha2	9	Si
Dcha3	10	Si
Dcha4	9	Si
Subir1	11	Si
Subir2	10	Si
Subir3	9	Si
Subir4	9	Si
Bajar1	10	Si
Bajar2	9	Si
Bajar3	10	Si
Bajar4	10	Si

Figura 38: Datos tomados en diferentes días

Gestos a reconocer / Ficheros	Izquierda	¿Detectado Correctamente?
CalculoMaxMin		
Izq1	12	Si
Izq2	12	Si
Izq3	12	Si
Izq4	12	Si
Izq5	12	Si
Dcha1	12	No
Dcha2	12	No
Dcha3	12	No
Dcha4	12	No
Dcha5	12	No
Subir1	12	No
Subir2	12	No
Subir3	12	No
Subir4	12	No
Subir5	12	No
Bajar1	12	No
Bajar2	12	No
Bajar3	12	No
Bajar4	12	No
Bajar5	12	No
ProcesaFich		
Izq1	12	Si
Izq2	12	Si
Izq3	12	Si
Izq4	12	Si
Izq5	12	Si
Dcha1	12	No
Dcha2	12	No
Dcha3	12	No
Dcha4	12	No
Dcha5	12	No
Subir1	12	No
Subir2	12	No
Subir3	12	No
Subir4	12	No
Subir5	12	No
Bajar1	12	No
Bajar2	12	No
Bajar3	12	No
Bajar4	12	No
Bajar5	12	No

Figura 39: Datos tomados en el mismo día

Seguidamente, al no estar conformes con los resultados que se estaban obteniendo, se probó a modificar los límites del rango, de forma que se fuesen cambiando los valores comprobando así cuál era el que mejor resultados daba.

Para ello se crearon diferentes tablas por cada gesto, en las que se iban cambiando los límites tal y como se aprecia en las siguientes tablas.

0.9 - 1.1	0.9 - 1.0	0.9 - 0.9
1.0 - 1.1	1.0 - 1.0	1.0 - 0.9
1.1 - 1.1	1.1 - 1.0	1.1 - 0.9

Cuadro 2: Tabla de límites para los rangos

Con estos nuevos cambios en el código, se volvieron a procesar los datos y se obtuvieron las siguientes tablas. Como se puede observar, no hay unos límites en los que algún gesto resulto completamente correcto. Aunque sí es cierto, que para los límites 1.0 - 1.1 en el gesto derecha, los resultados son bastante positivos; sin embargo, no se puede decir lo mismo de los otros ficheros de datos tomando los mismos límites.

Por esta razón se ha llegado a la conclusión de que con este tipo de sensores, en los que solo se obtiene información de cuatro puntos del cerebro y estando colocados tan cerca entre sí, no se puede obtener el objetivo deseado al inicio del proyecto, lo que era el objetivo del mismo.

0.9 - 1.1	Gestos a reconocer / Ficheros	Izquierda	¿Detectado Correctamente?	0.9 - 1.0	Gestos a reconocer / Ficheros	Izquierda	¿Detectado Correctamente?	0.9 - 0.9	Gestos a reconocer / Ficheros	Izquierda	¿Detectado Correctamente?
	CalculoMax/Min				CalculoMax/Min				CalculoMax/Min		
	Izq1	12	Sí		Izq1	12	Sí		Izq1	12	Sí
	Izq2	12	Sí		Izq2	12	Sí		Izq2	12	Sí
	Izq3	12	Sí		Izq3	12	Sí		Izq3	12	Sí
	Izq4	12	Sí		Izq4	12	Sí		Izq4	12	Sí
	Izq5	12	Sí		Izq5	12	Sí		Izq5	12	Sí
	Dcha1	12	No		Dcha1	12	No		Dcha1	12	No
	Dcha2	12	No		Dcha2	12	No		Dcha2	12	No
	Dcha3	12	No		Dcha3	12	No		Dcha3	12	No
	Dcha4	12	No		Dcha4	12	No		Dcha4	11	Sí
	Dcha5	12	No		Dcha5	12	No		Dcha5	12	No
	Subir1	12	No		Subir1	12	No		Subir1	12	No
	Subir2	12	No		Subir2	12	No		Subir2	12	No
	Subir3	12	No		Subir3	12	No		Subir3	12	No
	Subir4	12	No		Subir4	12	No		Subir4	12	No
	Subir5	12	No		Subir5	12	No		Subir5	12	No
	Bajar1	12	No		Bajar1	12	No		Bajar1	12	No
	Bajar2	12	No		Bajar2	12	No		Bajar2	12	No
	Bajar3	12	No		Bajar3	12	No		Bajar3	12	No
	Bajar4	12	No		Bajar4	12	No		Bajar4	12	No
	Bajar5	12	No		Bajar5	12	No		Bajar5	12	No
	ProcesaFich				ProcesaFich				ProcesaFich		
	Izq1	12	Sí		Izq1	12	Sí		Izq1	12	Sí
	Izq2	12	Sí		Izq2	12	Sí		Izq2	12	Sí
	Izq3	12	Sí		Izq3	12	Sí		Izq3	12	Sí
	Izq4	12	Sí		Izq4	12	Sí		Izq4	12	Sí
	Izq5	12	Sí		Izq5	12	Sí		Izq5	11	No
	Dcha1	12	No		Dcha1	12	No		Dcha1	12	No
	Dcha2	12	No		Dcha2	12	No		Dcha2	12	No
	Dcha3	12	No		Dcha3	12	No		Dcha3	12	No
	Dcha4	12	No		Dcha4	12	No		Dcha4	12	No
	Dcha5	12	No		Dcha5	12	No		Dcha5	12	No
	Subir1	12	No		Subir1	12	No		Subir1	12	No
	Subir2	12	No		Subir2	12	No		Subir2	12	No
	Subir3	12	No		Subir3	12	No		Subir3	12	No
	Subir4	12	No		Subir4	12	No		Subir4	12	No
	Subir5	12	No		Subir5	12	No		Subir5	12	No
	Bajar1	12	No		Bajar1	12	No		Bajar1	12	No
	Bajar2	12	No		Bajar2	12	No		Bajar2	12	No
	Bajar3	12	No		Bajar3	12	No		Bajar3	12	No
	Bajar4	12	No		Bajar4	12	No		Bajar4	12	No
	Bajar5	12	No		Bajar5	12	No		Bajar5	12	No

Figura 40: Tablas de procesamiento con límite inferior 0.9: gesto izquierda

1.0 – 1.1	Gestos a reconocer / Ficheros	Derecha	¿Detectado Correctamente?	1.0 – 1.0	Gestos a reconocer / Ficheros	Derecha	¿Detectado Correctamente?	1.0 – 0.9	Gestos a reconocer / Ficheros	Derecha	¿Detectado Correctamente?
	CalculoMaxMin				CalculoMaxMin				CalculoMaxMin		
	Dcha1	12	Si		Dcha1	11	No		Dcha1	11	No
	Dcha2	12	Si		Dcha2	11	No		Dcha2	11	No
	Dcha3	12	Si		Dcha3	11	No		Dcha3	11	No
	Dcha4	11	No		Dcha4	11	No		Dcha4	11	No
	Dcha5	12	Si		Dcha5	11	No		Dcha5	11	No
	Izq1	9	Si		Izq1	9	Si		Izq1	9	Si
	Izq2	10	Si		Izq2	8	Si		Izq2	8	Si
	Izq3	10	Si		Izq3	10	Si		Izq3	8	Si
	Izq4	10	Si		Izq4	10	Si		Izq4	10	Si
	Izq5	10	Si		Izq5	9	Si		Izq5	8	Si
	Subir1	11	Si		Subir1	10	Si		Subir1	10	Si
	Subir2	9	Si		Subir2	8	Si		Subir2	8	Si
	Subir3	10	Si		Subir3	10	Si		Subir3	8	Si
	Subir4	9	Si		Subir4	8	Si		Subir4	8	Si
	Subir5	9	Si		Subir5	8	Si		Subir5	7	Si
	Bajar1	9	Si		Bajar1	8	Si		Bajar1	7	Si
	Bajar2	9	Si		Bajar2	8	Si		Bajar2	7	Si
	Bajar3	9	Si		Bajar3	9	Si		Bajar3	9	Si
	Bajar4	9	Si		Bajar4	8	Si		Bajar4	7	Si
	Bajar5	8	Si		Bajar5	8	Si		Bajar5	7	Si
	ProcesaFich				ProcesaFich				ProcesaFich		
	Dcha1	9	No		Dcha1	11	No		Dcha1	10	No
	Dcha2	11	No		Dcha2	9	No		Dcha2	8	No
	Dcha3	11	No		Dcha3	11	No		Dcha3	11	No
	Dcha4	11	No		Dcha4	11	No		Dcha4	9	No
	Dcha5	11	No		Dcha5	11	No		Dcha5	10	No
	Izq1	9	Si		Izq1	9	Si		Izq1	9	Si
	Izq2	11	Si		Izq2	10	Si		Izq2	10	Si
	Izq3	11	Si		Izq3	10	Si		Izq3	10	Si
	Izq4	9	Si		Izq4	9	Si		Izq4	8	Si
	Izq5	9	Si		Izq5	9	Si		Izq5	8	Si
	Subir1	10	Si		Subir1	8	Si		Subir1	8	Si
	Subir2	9	Si		Subir2	7	Si		Subir2	8	Si
	Subir3	8	Si		Subir3	7	Si		Subir3	7	Si
	Subir4	8	Si		Subir4	8	Si		Subir4	9	Si
	Subir5	8	Si		Subir5	8	Si		Subir5	7	Si
	Bajar1	10	Si		Bajar1	10	Si		Bajar1	9	Si
	Bajar2	11	Si		Bajar2	10	Si		Bajar2	9	Si
	Bajar3	12	No		Bajar3	12	No		Bajar3	11	Si
	Bajar4	12	No		Bajar4	12	No		Bajar4	11	Si
	Bajar5	10	Si		Bajar5	10	Si		Bajar5	9	Si

Figura 41: Tablas de procesamiento con límite inferior 1.0: gesto derecha

1.1 – 1.1	Gestos a reconocer / Ficheros	Subir	¿Detectado Correctamente?	1.1 – 1.0	Gestos a reconocer / Ficheros	Subir	¿Detectado Correctamente?	1.1 – 0.9	Gestos a reconocer / Ficheros	Subir	¿Detectado Correctamente?
	CalculoMaxMin				CalculoMaxMin				CalculoMaxMin		
	Subir1	7	No		Subir1	6	No		Subir1	5	No
	Subir2	11	No		Subir2	9	No		Subir2	7	No
	Subir3	8	No		Subir3	8	No		Subir3	7	No
	Subir4	8	No		Subir4	7	No		Subir4	5	No
	Subir5	12	Si		Subir5	8	No		Subir5	6	No
	Bajar1	8	Si		Bajar1	7	Si		Bajar1	5	Si
	Bajar2	8	Si		Bajar2	7	Si		Bajar2	6	Si
	Bajar3	10	Si		Bajar3	9	Si		Bajar3	8	Si
	Bajar4	9	Si		Bajar4	8	Si		Bajar4	7	Si
	Bajar5	9	Si		Bajar5	8	Si		Bajar5	7	Si
	Izq1	8	Si		Izq1	7	Si		Izq1	7	Si
	Izq2	8	Si		Izq2	7	Si		Izq2	6	Si
	Izq3	8	Si		Izq3	7	Si		Izq3	6	Si
	Izq4	8	Si		Izq4	7	Si		Izq4	5	Si
	Izq5	8	Si		Izq5	7	Si		Izq5	6	Si
	Dcha1	8	Si		Dcha1	7	Si		Dcha1	5	Si
	Dcha2	8	Si		Dcha2	7	Si		Dcha2	5	Si
	Dcha3	8	Si		Dcha3	7	Si		Dcha3	5	Si
	Dcha4	8	Si		Dcha4	7	Si		Dcha4	5	Si
	Dcha5	8	Si		Dcha5	7	Si		Dcha5	5	Si
	ProcesaFich				ProcesaFich				ProcesaFich		
	Subir1	12	Si		Subir1	8	No		Subir1	6	No
	Subir2	10	No		Subir2	8	No		Subir2	5	No
	Subir3	10	No		Subir3	8	No		Subir3	5	No
	Subir4	8	No		Subir4	7	No		Subir4	5	No
	Subir5	8	No		Subir5	7	No		Subir5	6	No
	Bajar1	8	Si		Bajar1	7	Si		Bajar1	6	Si
	Bajar2	8	Si		Bajar2	7	Si		Bajar2	5	Si
	Bajar3	8	Si		Bajar3	7	Si		Bajar3	6	Si
	Bajar4	9	Si		Bajar4	7	Si		Bajar4	6	Si
	Bajar5	8	Si		Bajar5	7	Si		Bajar5	6	Si
	Izq1	8	Si		Izq1	7	Si		Izq1	5	Si
	Izq2	8	Si		Izq2	7	Si		Izq2	5	Si
	Izq3	8	Si		Izq3	7	Si		Izq3	5	Si
	Izq4	8	Si		Izq4	7	Si		Izq4	5	Si
	Izq5	8	Si		Izq5	7	Si		Izq5	5	Si
	Dcha1	8	Si		Dcha1	7	Si		Dcha1	5	Si
	Dcha2	8	Si		Dcha2	7	Si		Dcha2	5	Si
	Dcha3	8	Si		Dcha3	7	Si		Dcha3	5	Si
	Dcha4	8	Si		Dcha4	7	Si		Dcha4	5	Si
	Dcha5	8	Si		Dcha5	7	Si		Dcha5	5	Si

Figura 42: Tablas de procesamiento con límite inferior 1.1: gesto subir

0.9 – 1.0	Gestos a reconocer / Ficheros	Bajar	¿Detectado Correctamente?	1.0 – 1.1	Gestos a reconocer / Ficheros	Bajar	¿Detectado Correctamente?	1.1 – 0.9	Gestos a reconocer / Ficheros	Bajar	¿Detectado Correctamente?
	CalculoMax/Min				CalculoMax/Min				CalculoMax/Min		
	Bajar1	12	Si		Bajar1	12	Si		Bajar1	10	No
	Bajar2	12	Si		Bajar2	12	Si		Bajar2	9	No
	Bajar3	12	Si		Bajar3	12	Si		Bajar3	11	No
	Bajar4	12	Si		Bajar4	12	Si		Bajar4	9	No
	Bajar5	12	Si		Bajar5	12	Si		Bajar5	9	No
	Subir1	12	No		Subir1	12	No		Subir1	11	Si
	Subir2	12	No		Subir2	12	No		Subir2	11	Si
	Subir3	12	No		Subir3	12	No		Subir3	11	Si
	Subir4	11	Si		Subir4	12	No		Subir4	10	Si
	Subir5	11	Si		Subir5	11	Si		Subir5	10	Si
	Izq1	12	No		Izq1	12	No		Izq1	11	Si
	Izq2	12	No		Izq2	12	No		Izq2	11	Si
	Izq3	12	No		Izq3	12	No		Izq3	11	Si
	Izq4	12	No		Izq4	12	No		Izq4	11	Si
	Izq5	11	Si		Izq5	11	Si		Izq5	8	Si
	Dcha1	12	No		Dcha1	12	No		Dcha1	11	Si
	Dcha2	12	No		Dcha2	12	No		Dcha2	11	Si
	Dcha3	12	No		Dcha3	12	No		Dcha3	11	Si
	Dcha4	12	No		Dcha4	12	No		Dcha4	11	Si
	Dcha5	12	No		Dcha5	12	No		Dcha5	9	Si
	ProcesaFich				ProcesaFich				ProcesaFich		
	Bajar1	11	No		Bajar1	10	No		Bajar1	8	No
	Bajar2	12	Si		Bajar2	12	Si		Bajar2	10	No
	Bajar3	12	Si		Bajar3	11	No		Bajar3	10	No
	Bajar4	12	Si		Bajar4	12	Si		Bajar4	11	No
	Bajar5	12	Si		Bajar5	12	Si		Bajar5	10	No
	Subir1	12	No		Subir1	11	Si		Subir1	11	Si
	Subir2	11	Si		Subir2	11	Si		Subir2	10	Si
	Subir3	11	No		Subir3	11	Si		Subir3	10	Si
	Subir4	12	No		Subir4	12	No		Subir4	11	Si
	Subir5	12	No		Subir5	11	Si		Subir5	10	Si
	Izq1	8	Si		Izq1	8	Si		Izq1	5	Si
	Izq2	12	No		Izq2	12	No		Izq2	11	Si
	Izq3	12	No		Izq3	12	No		Izq3	10	Si
	Izq4	9	Si		Izq4	8	Si		Izq4	8	Si
	Izq5	11	Si		Izq5	8	Si		Izq5	5	Si
	Dcha1	11	Si		Dcha1	10	Si		Dcha1	11	Si
	Dcha2	11	Si		Dcha2	11	Si		Dcha2	10	Si
	Dcha3	11	Si		Dcha3	12	No		Dcha3	10	Si
	Dcha4	11	Si		Dcha4	11	Si		Dcha4	10	Si
	Dcha5	12	No		Dcha5	12	No		Dcha5	9	Si

Figura 43: Tablas de procesamiento con límite inferior 1.0: gesto bajar

9. Líneas futuras

En un futuro, se podría plantear el mismo proyecto de distintas formas. Por un lado, se podrían colocar los sensores en otros puntos de la cabeza, para probar si uno de los problemas era la colocación de estos. Ya que, en primer lugar, los sensores se encontraban dispuestos de la siguiente manera: dos situados encima de ambas cejas y los otros dos en los laterales de la cabeza, por lo que se pondrían los otros dos en los laterales de la cabeza, por lo que se podría deducir, viendo las tablas de procesamiento, que las ondas son bastantes parecidas en cuanto a los valores resultantes y por esa razón, no se distinguen entre sí.

Otra de las cuestiones que se abordan cuando se termina el proyecto, es la similitud de los gestos, dado que se trata de subir y bajar las cejas y de cerrar el ojo izquierdo y el derecho. Es por ello, que al realizar gestos tan parecidos entre sí, se pueden obtener los mismos valores y de esta forma, no se distinguen correctamente.

Por otro lado, otra de las soluciones que se pueden proponer para llevar a cabo este proyecto, es utilizar otro tipo de sensores que capten información no solo de cuatro puntos, sino de toda la cabeza, de manera que faciliten la obtención de ondas desde cualquier punto del cerebro.

Este proyecto, ya se ha sido desarrollado con otro producto de OpenBCI, Ultracortex EEG Headset. Dicho proyecto, fue llevado a cabo por un grupo de estudiantes americanos, que se dividían en varios departamentos: recopilación de datos, procesamiento de la señal, machine learning, software y hardware.

Cada departamento contaba con un grupo numeroso de estudiantes que se centraban única y exclusivamente en la tarea asignada.

El proyecto en cuestión, MILO (Mind-Controlled Locomotive), se basa en desarrollar una silla de ruedas que se mueve a través de simples pensamientos, es decir, si la persona desea moverse a la derecha, se imagina el movimiento y la silla lo realiza. [19]

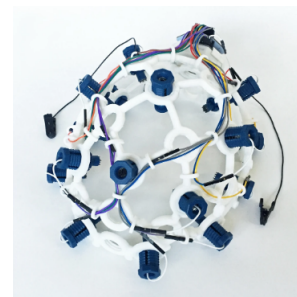


Figura 44: Ultracortex EEG Headset

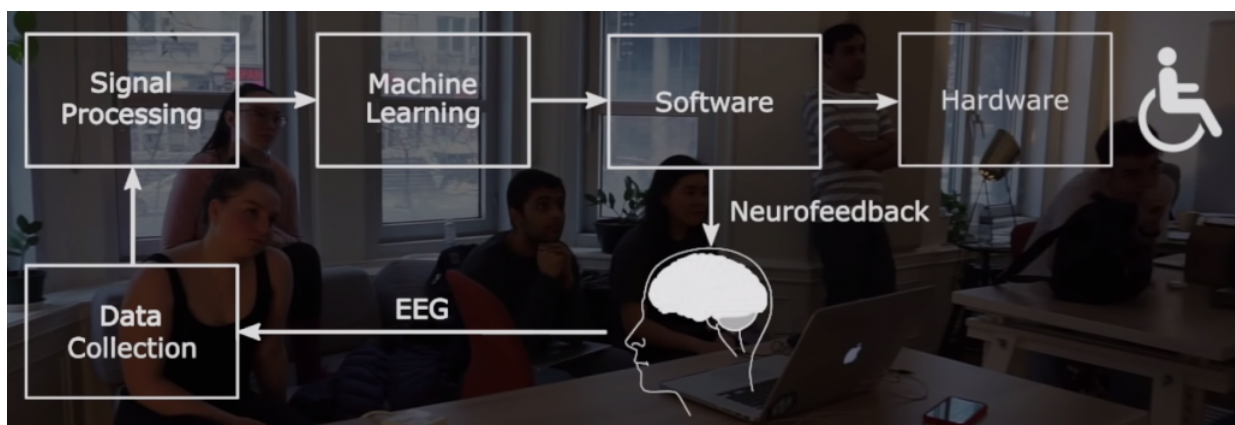


Figura 45: Esquema de trabajo de MILO

10. Conclusión

Por último, pese a que el objetivo del trabajo no se ha alcanzado, se han logrado otras tareas que han sido importantes en la realización del Trabajo de Fin de Grado.

En primer lugar, se ha conseguido conectar el ordenador a los sensores a pesar de los problemas iniciales que dieron lugar a que el comienzo del proyecto se retrasase. En parte por la utilización de Ubuntu como sistema operativo y, por otra parte, por las distintas versiones que se encuentran colgadas en la página, ya que, la más actual solo permite la conexión con el sistema operativo IOS.

En segundo lugar, se logró obtener datos a partir de las ondas producidas con los sensores y se consiguió analizar las diferentes columnas de valores de los ficheros. De esta forma, se sabía en todo momento los datos que se estaban tratando y que se iban a procesar a continuación.

Seguidamente, se diseñó un programa que procesase las ondas, con el cual, a pesar de los múltiples cambios y enfoques aplicados, se logró obtener información de las ondas analizadas y se pudieron reconocer los gestos realizados, si bien no de la forma que se pretendía, pues el reconocimiento de estos no era totalmente correcto.

A pesar de que el objetivo del proyecto, no se ha podido lograr, se han estudiado las posibles causas. Por una parte, puede haber sido el hecho de que los sensores se encontrasen tan próximos los unos a los otros, de forma que al realizar los gestos, un sensor podía captar ondas que no le correspondieran.

Por otro lado, se puede suponer, que el hecho de realizar gestos tan parecidos entre sí puede dificultar la tarea de distinguirlos, dado que las ondas son bastante parecidas entre sí. Aunque bien es cierto, que cuando se analiza el gesto derecha y se utilizan los datos de subir o de bajar, los resultados obtenidos son igual de insatisfactorios que se si comparan con los de izquierda.

Por último, se llega a la conclusión final de que, lo idóneo para poder terminar este proyecto, sería contar con otro tipo de sensores, de manera que en vez de captar información solo de cuatro puntos, se obtuviera de diferentes puntos del cráneo, es decir, que se utilizase, por ejemplo, el Ultracortex EEG Headset o modelos similares.

11. Conclusion

Finally, despite the fact that the objective of the work has not been achieved, other tasks have been accomplished that have been important in the completion of the End of Degree work.

First of all, the computer has been successfully connected to the sensors despite initial problems that caused the start of the project to be delayed. This is partly due to the use of Ubuntu as the operating system, and partly due to the different versions that are posted on the page, since the most current one only allows connection to the IOS operating system.

Secondly, it was possible to obtain data from the waves produced by the sensors and to analyse the different columns of values in the files. In this way, we knew at all times what data was being processed and what was going to be processed next.

Then, a program was designed to process the waves, with which, in spite of the multiple changes and approaches applied, it was possible to obtain information from the analysed waves and to recognise the gestures made, although not in the way that was intended, since the recognition of these was not totally correct.

Although the objective of the project has not been achieved, the possible causes have been studied. On the one hand, it may have been the fact that the sensors were so close to each other that when making the gestures, a sensor could pick up waves that did not correspond to it. On the other hand, it can be assumed that making gestures so similar to each other can make the task of distinguishing them difficult, given that the waves are quite similar to each other.

Although it is certain, that when the right gesture is analyzed and the data of raising or lowering is used, the obtained results are as unsatisfactory as if they were compared with those of the left.

Finally, the final conclusion is that the ideal way to finish this project would be to have another type of sensor, so that instead of capturing information from only four points, it would be obtained from different points of the skull, that is, using, for example, the Ultracortex EEG Headset or similar models.

12. Agradecimientos

En primer lugar, quiero agradecer a mis padres y hermanas por haberme acompañado durante la carrera, por haberme apoyado con cada una de mis decisiones, dejando que fuese yo quien las tomase ya fuesen acertadas o erróneas. Gracias por haberme ayudado cada vez que lo he necesitado, por haberme entendido en cada uno de los momentos que he pasado y por haber estado siempre a mi lado.

En segundo lugar, me gustaría agradecer a mis amistades por haber creído en mí cuando ni yo misma lo hacía, dándome palabras de aliento y ánimo, y por haberme ofrecido ayuda en todo momento.

Asimismo, me gustaría dar las gracias a Jonay, mi tutor de Trabajo de Fin de Grado, quien confió en mí para llevar a cabo este proyecto, resolviendome todas las dudas y ayudándome hasta el final.

Finalmente agradecer también a la Universidad de La Laguna por haber financiado mi Trabajo de Fin de Grado, permitiéndome realizar el estudio que deseaba y de esta forma, poder descubrir el mundo de la Biomedicina.

Referencias

- [1] <https://ortopedia1.com/silla-de-ruedas>
- [2] <https://en.m.wikipedia.org/wiki/Wheelchair>.
- [3] https://en.m.wikipedia.org/wiki/Bath_chair
- [4] <https://www.smpcares.com/wp-content/uploads/2013/05/9-One-Arm-Drive-Wheelchair-1.pdf>
- [5] http://atwiki.assistivetech.net/index.php/Tilt-in-space_and_Recline_Wheelchairs
- [6] <https://www.irisbond.com/tecnologia/>

- [7] <https://www.irisbond.com/soporte-y-formacion/primeros-pasos-con-mi-sistema-de-comunicacion>
- [8] <https://mobilitymgmt.com/Articles/2018/09/01/Tongue-Drive-System.aspx?Page=1>
<https://es.gizmodo.com/crean-una-silla-de-ruedas-para-tetraplejicos-controlada-1473498217>
<https://www.20minutos.es/noticia/1992709/0/piercing-lengua/control/silla-ruedas>
<https://www.abc.es/sociedad/20131127/abci-silla-ruedas-conducir-lengua-201311271900.html?>
<https://www.20minutos.es/noticia/394347/0/silla/ruedas/lengua/>
<https://www.news.gatech.edu/features/tongue-drive-wheelchair>
- [9] <https://hoobox.one/>
<https://www.technowize.com/kit-wheelie-7-a-kit-to-control-wheelchair-with-facial-expressions/>
<https://www.fastcompany.com/90275037/people-can-now-control-their-wheelchairs-with-just-their-facial-expressions>
<https://strammer.com/en/wheelchairs-human-expression-controlled/>

- [10] <https://shop.openbci.com/collections/frontpage/products/biosensing-classroom-bundle?variant=13043163824200>
- [11] <https://shop.openbci.com/collections/frontpage/products/pre-order-ganglion-board?variant=13461804483>
- [12] <https://github.com/OpenBCI>
- [13] <https://docs.openbci.com/docs/08FAQ/FAQLanding>
- [14] <https://openbci.com/forum/>
- [15] https://github.com/OpenBCI/OpenBCI_GUI/issues
- [16] <https://stackoverflow.com/>
- [17] [https://es.wikipedia.org/wiki/Normalizaci%C3%B3n_\(estad%C3%ADstica\)](https://es.wikipedia.org/wiki/Normalizaci%C3%B3n_(estad%C3%ADstica))
- [18] https://es.wikipedia.org/wiki/L%C3%B3gica_difusa
- [19] https://www.youtube.com/watch?v=_46AoSnHCRo

A. Anexo I: Códigos

A.1. CalculoMaxMin

```
close all;
clear all;
clc;
fichero = "\\Bajar"
s5 = "Bajar";
pkg load signal;

j = 1;
rangos = [0,7.7;7.7,15.4;15.4,23.1];
TGesto = [ 1075,1100,1100,1080,1100;
           1200,1225,1225,1205,1225]

function [energia] = energia_calculation(signal, freq,range) %Se calcula la energia
    for j=1:4
        index = find((freq>range(1)) & (freq < range(2))); %Se definen los indices del intervalo
        amplitud = abs(signal(index(:)));
        energia = median(amplitud); %Mediana de la energia
    end
endfunction

while j <= 5

    s0 = "C:\\Users\\Cris\\Downloads\\Octave\\Datos\\Calculo";
    s1 = fichero;
    s2 = int2str(j);
    s3 = ".txt";
    disp([s0,s1,s2,s3]);
    file = fopen(strcat(s0, s1, s2, s3), 'r+');
    matriz = dlmread(file);
    d = matriz(7:end,1:5);
    size(d)

    f1 = 30;
    f2 = 45;
    delta_f = f2-f1;
    Fs = 200; %Define la frecuencia de muestreo, la mitad de esta frecuencia es la frecuencia maxima a
              representar
    dB = 40; %Se define para darle unidades
    N = dB*Fs/(22*delta_f); %Define el orden del filtro --> es un parametro que te dice cuantos baches va
                             a tener tu filtro, el cual es una banda que cae a una frecuencia determinada (f) y luego tiene como
                             varios montoncitos.
```



```

f = [f1 ]/(Fs/2); %Esta es la frecuencia maxima que estipula tu tutor que tienes unicamente la seal
que deseas.
hc = fir1(round(N)-1, f,'low'); %Define un filtro pasa baja, es decir, solamente pasan las frecuencias
por debajo de lo que el considere

%Filtra los datos, pasa los datos por el filtro
df = filter(hc,1,d(TGesto(1,j):TGesto(2,j),2));
df(:,2) = filter(hc,1,d(TGesto(1,j):TGesto(2,j),3));
df(:,3) = filter(hc,1,d(TGesto(1,j):TGesto(2,j),4));
df(:,4) = filter(hc,1,d(TGesto(1,j):TGesto(2,j),5));

disp(['Gesto en el fichero',s1,s2,s3])
plot(df)
pause()

t=0:1/200:1/200*length(df)-1/200; %Crea una variable como eje x de los tiempos

%La y es la FFT
clear y
for i=1:4
    y(:,i)=abs(fft(df(:,i))); %Realiza la transformada de fourier de las seales
end

freqHz = (0:1:length(y)-1)*200/length(y); %length --> devuelve la longitud del vector. La frecuencia va a
ser desde 0 hasta 1, cada length(y)-1 por 200 entre length(y)

max_y = -1.0;
for i=1:4
    if( norm(y(:,i),1) > max_y)
        max_y = norm(y(:,i),1);
    end
end

clear y_n

for n=1:4
    y_n(:,n) = y(:,n)/max_y;
    for k=1:length(rangos)
        acumulada(j,n,k) = energia_calculation(y_n(:,n),freqHz(:),rangos(k,:));
    end
    f_remuestreada = mean(rangos'); %Se pone desde 0.5 aunque no salgan datos hasta 5.5
end

j=j+1;
end

[A,B,C] = size(acumulada);

clf

```

```

for i=1:4
    figure(i)
    hold on

    plot(f_remuestreada,acumulada(1,i,:),'*');
    legenda=[[fichero , int2str(1), "txt "]];
    for j=2:A
        plot(f_remuestreada,acumulada(j,i,:),'*');
        legenda=[legenda; [fichero , int2str(j), "txt "]];
    end
    title(["Sensor" , int2str(i)]);
    legend(legenda);
end

for i=1:4
    MaximosS(i,:) = max(squeeze(acumulada(:,i,:)));
    MinimosS(i,:) = min(squeeze(acumulada(:,i,:)));
end

save (sprintf ("%s.mat", s5), "MaximosB", "MinimosB")

```

A.2. CalculoNuevo

```
close all;
clear all;
clc;
fichero = "\\Bajar"
s5 = "Bajar";
pkg load signal;

j = 1;
rangos = [0,7.7;7.7,15.4;15.4,23.1];
TGesto = [ 1075,1100,1100,1080,1100;
           1200,1225,1225,1205,1225]

function [energia] = energia_calculation(signal, freq,range) %Se calcula la energia
    for j=1:4
        index = find((freq>range(1)) & (freq < range(2))); %Se definen los indices del intervalo
        amplitud = abs(signal(index(:)));
        energia = median(amplitud); %Mediana de la energia
    end
endfunction

while j <= 5

    s0 = "C:\\Users\\Cris\\Downloads\\Octave\\Datos\\Calculo";
    s1 = fichero;
    s2 = int2str(j);
    s3 = ".txt";
    disp([s0,s1,s2,s3]);
    file = fopen(strcat(s0, s1, s2, s3), 'r+');
    matriz = dlmread(file);
    d = matriz(7:end,1:5);
    size(d)

    f1 = 30;
    f2 = 45;
    delta_f = f2-f1;
    Fs = 200; %Define la frecuencia de muestreo, la mitad de esta frecuencia es la frecuencia maxima a
    representar
    dB = 40; %Se define para darle unidades
    N = dB*Fs/(22*delta_f); %Define el orden del filtro --> es un parametro que te dice cuantos baches va
    a tener tu filtro, el cual es una banda que cae a una frecuencia determinada (f) y luego tiene como
    varios montoncitos.

    f = [f1 ]/(Fs/2); %Esta es la frecuencia maxima que estipula tu tutor que tienes unicamente la seal
    que deseas.
```

```

hc = fir1(round(N)-1, f,'low'); %Define un filtro pasa baja, es decir, solamente pasan las frecuencias
    por debajo de lo que el considere

%Filtra los datos, pasa los datos por el filtro
df = filter(hc,1,d(TGesto(1,j):TGesto(2,j),2));
df(:,2) = filter(hc,1,d(TGesto(1,j):TGesto(2,j),3));
df(:,3) = filter(hc,1,d(TGesto(1,j):TGesto(2,j),4));
df(:,4) = filter(hc,1,d(TGesto(1,j):TGesto(2,j),5));
df2 = filter(hc,1,d(:,2));
df2(:,2) = filter(hc,1,d(:,3));
df2(:,3) = filter(hc,1,d(:,4));
df2(:,4) = filter(hc,1,d(:,5));

disp(['Gesto en el fichero ',s1,s2,s3])

plot(TGesto(1,j):TGesto(2,j),df, '* ',1:length(df2),df2)
title([s1,s2,s3])
pause

t=0:1/200:1/200*length(df)-1/200; %Crea una variable como eje x de los tiempos

%La y es la FFT
clear y
for i=1:4
    y(:,i)=abs(fft(df(:,i))); %Realiza la transformada de fourier de las seales
end

freqHz = (0:1:length(y)-1)*200/length(y); %length --> devuelve la longitud del vector. La frecuencia
    va a ser desde 0 hasta 1, cada length(y)-1 por 200 entre length(y)

plot(freqHz,y)
title(['Freq: ', s1, s2, s3]);
pause;

max_y = -1.0;
for i=1:4
    if( norm(y(:,i),1) > max_y)
        max_y = norm(y(:,i),1);
    end
end

clear y_n

for n=1:4
    y_n(:,n) = y(:,n)/max_y;
    for k=1:length(rangos)
        acumulada(j,n,k) = energia_calculation(y_n(:,n),freqHz(:),rangos(k,:));
    end
    f_remuestreada = mean(rangos'); %Se pone desde 0.5 aunque no salgan datos hasta 5.5

```

```

end

j=j+1;
end

[A,B,C] = size(acumulada);

clf
for i=1:4
    for j=1:3
        acumulada(:,i,j) = rmoutlier(acumulada(:,i,j),1);
    end
end

for i=1:4 # Para cada sensor
    figure(i)
    hold on

    plot(f_remuestreada,acumulada(1,i,:), '*');
    legenda=[[fichero , int2str(1), " txt "]];
    for j=2:A
        plot(f_remuestreada,acumulada(j,i,:), '*');
        legenda=[legenda; [fichero , int2str(j), " txt "]];
    end
    title([" Sensor" ,int2str(i) ]);
    legend(legenda);
end

for i=1:4
    MaximosB(i,:) = max(squeeze(acumulada(:,i,:)));
    MinimosB(i,:) = min(squeeze(acumulada(:,i,:)));
end

save ( sprintf (" %s.mat", s5), "MaximosB", "MinimosB")

```

A.3. ProcesaFich

```
load Dcha.mat
load Subir.mat
load Bajar.mat
load Izq.mat

pkg load signal;

RangoExaminar = 125;

fich = "C:\\Users\\Cris\\Downloads\\Octave\\Datos\\Prueba\\Dcha5.txt"
rangos = [0,7.7;7.7,15.4;15.4,23.1];

function [energia] = energia_calculation(signal, freq,range) %Se calcula la energia

    for j=1:4
        index = find((freq>range(1)) & (freq < range(2))); %Se definen los indices del intervalo
        amplitud = abs(signal(index(:)));
        energia = median(amplitud); %Mediana de la energia
    end
endfunction

file = fopen(fich, 'r+');
matriz = dlmread(file);
d = matriz(7:end,1:5);
size(d)

f1 = 30;
f2 = 45;
delta_f = f2-f1;
Fs = 200; %Define la frecuencia de muestreo, la mitad de esta frecuencia es la frecuencia maxima a
representar
dB = 40; %Se define para darle unidades
N = dB*Fs/(22*delta_f); %Define el orden del filtro --> es un parametro que te dice cuantos baches va
a tener tu filtro, el cual es una banda que cae a una frecuencia determinada (f) y luego tiene como
varios montoncitos.

f = [f1]/(Fs/2); %Esta es la frecuencia maxima que estipula tu tutor que tienes unicamente la seal
que deseas.
hc = fir1(round(N)-1, f,'low'); %Define un filtro pasa baja, es decir, solamente pasan las frecuencias
por debajo de lo que el considere

for l=1:length(d)-RangoExaminar
    %Filtra los datos, pasa los datos por el filtro
    df = filter(hc,1,d(1:l+RangoExaminar,2));
    df(:,2) = filter(hc,1,d(1:l+RangoExaminar,3));
    df(:,3) = filter(hc,1,d(1:l+RangoExaminar,4));
```

```

df(:,4) = filter(hc,1,d(1:1+RangoExaminar,5));

t=0:1/200:1/200*length(df)-1/200; %Crea una variable como eje x de los tiempos

%La y es la FFT
clear y
for i=1:4
    y(:,i)=abs(fft(df(:,i))); %Realiza la transformada de fourier de las seales
end

freqHz = (0:length(y)-1)*200/length(y); %length --> devuelve la longitud del vector. La frecuencia va
    a ser desde 0 hasta 1, cada length(y)-1 por 200 entre length(y)

max_y = -1.0;
for i=1:4
    if( norm(y(:,i),1) > max_y)
        max_y = norm(y(:,i),1);
    end
end

clear y_n

for n=1:4
    y_n(:,n) = y(:,n)/max_y;
    for k=1:length(rangos)
        acumulada(n,k) = energia_calculation(y_n(:,n),freqHz(:),rangos(k,:));
    end
    f_remuestreada = mean(rangos'); %Se pone desde 0.5 aunque no salgan datos hasta 5.5
end

[C,D] = size(acumulada);
coincidencias = 0;
for j=1:C
    for k=1:D
        if ((acumulada(j,k) > MinimosB(j,k)*1.1) && (acumulada(j,k) < MaximosB(j,k)*0.9))
            coincidencias = coincidencias + 1;
        end
    end
end

resultado(1) = coincidencias;
end

tresultado=0:1/Fs:(length(resultado)-1)*(1/Fs);
plot(tresultado,resultado)

```

A.4. Outlier

```
## Copyright (C) 2007 Lukasz Komsta, http://www.komsta.net/
##
## This program is free software; you can redistribute it and/or modify
## it under the terms of the GNU General Public License as published by
## the Free Software Foundation; either version 2 of the License, or
## (at your option) any later version.
##
## This program is distributed in the hope that it will be useful,
## but WITHOUT ANY WARRANTY; without even the implied warranty of
## MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
## GNU General Public License for more details.
##
## You should have received a copy of the GNU General Public License
## along with this program; If not, see <http://www.gnu.org/licenses/>.

## Find value with largest difference from the mean
##
## Description:
##
##     Finds value with largest difference between it and sample mean,
##     which can be an outlier.
##
## Usage:
##
##     [out] = outlier(x,opposite, logical)
##
## Arguments:
##
##     x: a data sample, vector in most cases. If argument is a
##     matrix, each column is treated as independent dataset.
##
## opposite: if set to 1 (default 0), gives opposite value (if largest value has
##     maximum difference from the mean, it gives smallest and vice
##     versa)
##
## logical: if set to 1 (default 0), gives vector of logical values, and possible
##     outlier position is marked by 1, others are 0
##
## Value:
##
##     A vector of value(s) with largest difference from the mean.
##
## Author(s):
##
##     Lukasz Komsta, ported from R package "outliers".
## See R News, 6(2):10–13, May 2006
##
```



```
##
```

```
function [out] = outlier(x,opposite, logical )
```

```
    if nargin<3  
        logical =0;
```

```
    end
```

```
    if nargin<2  
        opposite=0;
```

```
    end
```

```
    if ~isvector(x) && ismatrix(x)
```

```
        out = [];
```

```
        for i=1:columns(x)
```

```
            oo = outlier(x(:,i),opposite, logical );
```

```
            out = [out oo];
```

```
        end
```

```
        elseif isvector(x)
```

```
            if (xor(((max(x) - mean(x)) < (mean(x) - min(x))), opposite))
```

```
                if ~logical
```

```
                    out = min(x);
```

```
                else
```

```
                    out = (x == min(x));
```

```
                end
```

```
                else
```

```
                    if ~logical
```

```
                        out = max(x);
```

```
                    else
```

```
                        out = (x == max(x));
```

```
                    end
```

```
            end
```

```
        else
```

```
            error("x must be a vector or a matrix");
```

```
        end
```

```
end
```

A.5. Rmoutlier

```
## Copyright (C) 2007 Lukasz Komsta, http://www.komsta.net/
##
## This program is free software; you can redistribute it and/or modify
## it under the terms of the GNU General Public License as published by
## the Free Software Foundation; either version 2 of the License, or
## (at your option) any later version.
##
## This program is distributed in the hope that it will be useful,
## but WITHOUT ANY WARRANTY; without even the implied warranty of
## MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
## GNU General Public License for more details.
##
## You should have received a copy of the GNU General Public License
## along with this program; If not, see <http://www.gnu.org/licenses/>.

## Remove the value(s) most differing from the mean
##
## Description:
##
##   This function can remove the outliers or replace by sample mean or median.
##
## Usage:
##
##   [res] = rmoutlier(x, fill ,median,opposite)
##
## Arguments:
##
##   x: a dataset, most frequently a vector. If argument is a
##       matrix, each column is treated as independent dataset.
##
##   fill : If set to 1 (default 0), the median or mean is placed instead of
##           outlier . Otherwise, the outlier (s) is /are simply removed.
##
##   median: If set to 1 (default 0), median is used instead of mean in outlier
##            replacement.
##
##   opposite: if set to 1 (default 0), replaces opposite value (if largest value has
##             maximum difference from the mean, it replaces smallest and vice
##             versa)
##
## Value:
##
##   A dataset of the same type as argument, with outlier(s) removed or
##   replaced by appropriate means or medians.
##
## Author(s):
```

```
##
##   Lukasz Komsta, ported from R package "outliers".
## See R News, 6(2):10–13, May 2006
##
##
```

```
function [res] = rmoutlier(x, fill ,median,opposite)
```

```
    if nargin<4
        opposite=0;
    end
    if nargin<3
        median=0;
    end
    if nargin<2
        fill =0;
    end
```

```
if ~isvector(x) && ismatrix(x)
    res = [];
    for i=1:columns(x)
        rr = rmoutlier(x(:,i), fill ,median,opposite);
        res = [res rr];
    end
```

```
elseif isvector(x)
    wo = x;
    ou = find(x == outlier(x,opposite));
    for i=1:length(ou)
        wo(ou(i)) = [];
    end
```

```
        if ~ fill
            res = wo
        else
            res = x;
            for i=1:length(ou)
                if median
                    res(ou(i)) = median(wo);
                else
                    res(ou(i)) = mean(wo);
                end
            end
        end
    end
end
end
```
