

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



PROYECTO FIN DE CARRERA

**DISEÑO Y CONTROL DE UN ROBOT MEDIANTE
INTERFAZ CEREBRO-MÁQUINA USANDO LA
TECNOLOGÍA SSVEP**

Manuel Marcos García

Julio 2015

Diseño y control de un robot mediante interfaz cerebro-máquina usando la tecnología SSVEP

**AUTOR: Manuel Marcos García
TUTOR: Pablo Varona Martínez**

**Grupo de Neurocomputación Biológica
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2015**

Agradecimientos

En primer lugar quiero agradecer a mi familia todo el apoyo mostrado a lo largo de estos años, por darme la educación y valores que sin duda me servirán de guía en el futuro. Llegar hasta aquí no hubiera sido posible sin vosotros: a mi padre, que ha confiado en mi incluso cuando todo parecía tambalearse; a mi madre, que ha hecho suyos todos mis sufrimientos y alegrías; y a mi hermano que siempre ha creído en mi, definiendo a capa y espada mi forma de hacer y actuar.

Gracias a ti que estás leyendo esto, porque de alguna manera estás ayudando a que trabajos como este vean la luz.

Gracias a Pablo, mi tutor, por darme la oportunidad de realizar este proyecto, por estar siempre receptivo y darme el tiempo y la libertad para trabajar. Gracias a las personas del GNB que han puesto su granito de arena en este trabajo y me han ayudado a solventar las dudas que me iban surgiendo: Paco, Jacobo, Carlos, Ana y Álvaro.

Gracias a todos los voluntarios que accedieron a someterse, a las a veces molestas, pruebas de estimulación visual. Con vuestra ayuda logramos por fin controlar el robot.

A todos los compañeros de carrera con los que he compartido clase y prácticas porque solo se va más rápido, pero en equipo se llega más lejos.

Quiero agradecer especialmente a Javier Díez Jordao y Javier Antón el apoyo incondicional, las risas y pachangas, las charlas y cañas. Gracias por estar siempre ahí. No me olvido de mis compañeros de residencia Eva y Luis, junto a los que empecé este largo camino con gran ilusión. Gracias por todo.

Para terminar, agradecer también a mis compañeros Erasmus Silvia, Fran y Elisa, porque también habéis contribuido de alguna manera a que llegue hasta aquí. Fue el frío a -40°C que nos congeló la nariz y nos despejó la mente.

Manuel Marcos García

Julio, 2015.

Resumen

El avance tecnológico así como la búsqueda de mecanismos de comunicación para personas con discapacidades severas han propiciado, hoy en día, que el control de dispositivos con señalización biológica sea un tema de investigación en auge. Este ámbito, especialmente relacionado con la medicina, ha sido el impulsor de la presente investigación cuyo objetivo es encontrar y proveer un medio de comunicación adaptado al usuario para todas estas personas con discapacidades.

En este proyecto se aborda el diseño, construcción y control de un robot básico a través de una interfaz cerebro-máquina que sea capaz de adaptarse a cada usuario. Hemos intentado optimizar el control del robot a fin de aportar un medio de comunicación a personas con discapacidades motoras severas que les impidan el movimiento de alguna parte del cuerpo o el habla. Esta adaptación al usuario se conseguirá a través de un proceso de selección de estímulos visuales que evoquen eventos específicos registrables con un casco de EEG de bajo coste. Para esta tarea, se utiliza un protocolo que busca, las cuatro frecuencias de estimulación visual más compatibles para controlar el robot. Los eventos evocados mediante esta estimulación visual se denominan SSVEP (Steady State Visually Evoked Potentials).

Se empezará realizando un estudio acerca de la situación actual de las interfaces cerebro-máquina, describiremos los distintos tipos y aplicaciones y expondremos cuáles son los sistemas de adquisición y estimulación utilizados hoy en día.

Una vez percatados del panorama actual y de la necesidad de este tipo de sistemas e interfaces, abordaremos el diseño y construcción de un robot básico, que se moverá de acuerdo a cuatro grados de libertad: adelante, parar, izquierda y derecha. Igualmente justificaremos las elecciones de los medios y del material usado para el desarrollo de este proyecto. Además, detallaremos todo el proceso, desde que adquirimos las señales cerebrales con un casco de EEG de bajo coste, su procesamiento, pasando por la transformación a un comando que enviaremos de forma inalámbrica al robot, hasta llegar a la gestión de este comando en el robot para conseguir su locomoción.

Con el sistema desarrollado se validará el funcionamiento de esta interfaz, especialmente la parte de selección de frecuencias adaptada al usuario y el control del robot. Hemos adaptado el sistema a un total de 10 sujetos voluntarios, haciendo un análisis de las frecuencias elegidas y probando el control del robot con 3 experimentos basados en la recepción de cadenas de comandos.

Para finalizar, expondremos unas conclusiones a la vista de los datos obtenidos y discutiremos algunas opciones interesantes de módulos a integrar o investigaciones futuras en este tipo de sistemas basados en la adaptación al usuario.

Abstract

The increasing technological advances as well as the pursuit of new communication mechanisms for people with severe disabilities have made biological signaling device control to be a booming subject nowadays. This area especially related to medicine has promoted the present research with the aim of finding and providing a means of communication adapted to the user for disabled persons.

In this project we approach the design, construction and control of a basic robot through a Brain Computer Interface, which is able to adjust to each user. We have tried to optimize the robot control to provide a means of communication to severe motors disabled persons, such as people with disability to move their body and speech problems. This user adaptation will be achieved through a process of visual stimuli selection which can elicit specific measurable events with a low cost EEG cap. For this task we use a protocol which looks for four compatible visual stimulation frequencies to control the robot. The elicited events of this visual stimulation are called SSVEPs (Steady State Visually Evoked Potentials).

This way, we will begin carrying out a study about the state of the art of brain computer interfaces. We will describe the different types and applications of the acquisition and stimulation systems used nowadays and we will decide which of them are more adequate for our goal in this project.

Once we have described the current scene and the need for such interfaces and systems, we will approach the design and construction of a basic robot, which will move according to four degrees of freedom: frontwards stop, leftwards and rightwards. We will justify our choices for the means and hardware used in the development of this project as well. We will detail all the process from the acquisition of brain signals with a low cost EEG cap, their processing the generation of a command that we will send wirelessly to the robot, to the management of this command in the robot to achieve its locomotion.

Once this system is developed, the operation of the interface will be validated, especially the user-specific frequency selection and the robot control phase. The system is tested in a total of 10 volunteers, including the frequency analysis and the control with three experiments based on the receptions of three command chains.

Finally, we draw several conclusions in the light of the data obtained and we discuss some interesting options to integrate extra modules or future investigations in this kind of systems based in user-specific adaptation.

Lista de palabras clave

Interfaz cerebro-máquina (BCI) adaptado al usuario, electroencefalograma (EEG), Steady State Visually Evoked Potentials (SSVEP), Arduino, WiFi, análisis de señal, artefactos, neuroprótesis, persona discapacitada, sistema motor, robot.

Keywords

User adapted Brain Computer Interface (BCI), Electroencephalogram (EEG), Steady State Visually Evoked Potentials (SSVEP), Arduino, WiFi, signal analysis, artifacts, neuroprosthesis, disable person, motor system, robot.

Índice de contenidos

1. Introducción.	1
1.1. Motivación.	1
1.2. Objetivos.	2
1.3. Estructura de la memoria.	4
2. Estado del arte.	7
2.1. BCI: Interfaces Cerebro-Máquina.	7
2.2. Registros de actividad cerebral.	9
2.2.1. Métodos invasivos y no invasivos.	10
2.2.2. La electroencefalografía (EEG).	11
2.2.3. BCI endógenos y exógenos.	13
2.2.4. Artefactos en el registro.	14
2.3. SSVEP: Steady State Visually Evoked Potentials.	15
2.3.1. Tipos de señales de control.	16
2.3.2. Interfaces basados en SSVEPs.	17
2.4. Aplicaciones.	18
3. Diseño.	25
3.1. Consideraciones generales. Introducción al diseño.	25
3.2. Captura de datos mediante el casco Emotiv EPOC	27
3.3. Diseño del robot.	29
3.3.1. Arduino Yún	30
3.3.2. Diseño de la estructura, componentes y montaje	33
3.3.3. Alimentación	36
3.3.4. Diseño de la placa auxiliar	37
4. Desarrollo	39
4.1. Adquisición de la señal de EEG	39
4.2. Procesamiento para detectar SSVEP.	41
4.3. Transformación a un comando.	45

4.4.	Comunicación de la interfaz cerebro-máquina con el robot.	47
4.5.	Movimiento el robot.....	49
4.6.	Pruebas intermedias básicas en el desarrollo.	52
5.	Experimentos y resultados.	55
5.1.	Validación del control del robot.	56
5.2.	Pruebas y resultados.....	58
5.2.1.	<i>Parte 1: Búsqueda de las mejores frecuencias de estimulación en cada sujeto para el control del robot.</i>	<i>58</i>
5.2.2.	<i>Parte 2: Estimulación del sujeto y eficiencia del sistema para el control del robot.</i>	<i>62</i>
6.	Conclusiones	69
7.	Estudio futuro.....	73
7.1.	Tabletas y Smartphones.....	73
7.2.	Sensores y Webcam de seguimiento.	74
8.	Referencias.....	77
Glosario		81
Anexos.....		I
	<i>Anexo I: Código de identificación de frecuencia y transformación a comando.</i>	<i>I</i>
	<i>Anexo II: Código del Servidor Web.</i>	<i>III</i>
	<i>Anexo III: Clase Robot.</i>	<i>IV</i>
Presupuesto		V
Pliego de condiciones		VI

Índice de Figuras

<i>Figura 1:</i> Esquema de etapas de un BCI.....	8
<i>Figura 2:</i> Actividad cerebral como respuesta a estímulos visuales evocados a diferentes frecuencias [43].	17
<i>Figura 3:</i> Entornos de desarrollo de aplicaciones BCI dentro de gráficas capacidades del usuario Vs flujo de transferencia de información. Adaptada de [1].	20
<i>Figura 4:</i> Brazo protésico realizando tareas de precisión [44].	22
<i>Figura 5:</i> Jugador usando un casco de EEG en un videojuego [45].	23
<i>Figura 13:</i> Diseño del circuito de la placa auxiliar.	37
<i>Figura 15:</i> Propagación de un SSVEP [12]	40
<i>Figura 16:</i> Esquema del análisis de señal.	42
<i>Figura 17:</i> Solape en el enventanado de señal.	43
<i>Figura 18:</i> Detección de SSVEPs con estímulos parpadeantes a 6 y 7Hz.	43
<i>Figura 19:</i> Código de inicialización de frecuencias.	45
<i>Figura 20:</i> Código de transformación a un comando.	46
<i>Figura 21:</i> Función de envío del comando.	48
<i>Figura 22:</i> Setup del programa principal del robot.	49
<i>Figura 23:</i> Extracción del comando.	50
<i>Figura 24:</i> Procesado del comando.....	50
<i>Figura 25:</i> Código de prueba de control por teclado y algunas secuencias de escape.	53
<i>Figura 26:</i> Esquema de validación del control del robot.	57
<i>Figura 27:</i> Esquema de la selección de frecuencias básica para adaptabilidad al usuario... ..	59
<i>Figura 28:</i> Estimulación del sujeto para el control del robot.	63
<i>Figura 29:</i> Circuitos 1 y 2 para la realización de los experimentos.....	64
<i>Figura 30:</i> Circuito 3 para la realización de los experimentos.	64
<i>Figura 31:</i> SBS2 para aplicaciones móviles de EEG en dispositivos Android conectados a casco Emotiv or EasyCap [37].	74
<i>Figura 32:</i> Control de un robot con cámara a través de smartphone.	75

Índice de Tablas

<i>Tabla 1:</i> Métodos de neuroimagen para detectar actividad cerebral. Adaptada de [1].....	10
<i>Tabla 2:</i> Tipos de señales cerebrales según el tipo de BCI exógeno o endógeno. Adaptada de [1]	14
<i>Tabla 3:</i> Tipos de señales de control y número posible de opciones. Adaptada de [1].....	16
<i>Tabla 4:</i> Características principales del Emotiv Epoc Adaptada de [38].	28
<i>Tabla 5:</i> Características de los micros de la tarjeta Arduino Yún. Adaptada de [28].	33
<i>Tabla 6:</i> Puntuaciones de todas las frecuencias estimuladas tras el barrido (6-20Hz).....	60
<i>Tabla 7:</i> Resultados tras la fase 1 de selección de frecuencias.	61
<i>Tabla 8:</i> Asignación de frecuencias de estimulación a comandos tras la fase 2.	61
<i>Tabla 9:</i> Resultados tras realizar los experimentos para la segunda prueba.....	66

1. Introducción.

1.1. Motivación.

Las interfaces hombre-máquina son los sistemas de comunicación que permiten a las personas comunicarse con los diferentes tipos de máquinas existentes. Sabemos que el ser humano necesita comunicarse con el medio que le rodea, por lo que será necesario incidir en sus aspectos cognitivos para diseñar modelos de interfaces adaptadas a las necesidades individuales y sociales del propio usuario. En la era actual de la información y el conocimiento, los sistemas informáticos ya no sólo establecen una comunicación usuario-máquina sino usuario-máquina-usuario, es por eso que se necesitará de sistemas que nos proporcionen una comunicación bidireccional fluida entre ellos.

En los últimos años se ha producido un gran avance en el mundo de la tecnología. Uno de los ámbitos que se ha visto más beneficiado de este avance es el de las interfaces, siendo cada vez más sofisticadas, reducidas y con más prestaciones, consiguiendo un despegue en este sector con respecto a la poca valoración que tuvieron en las últimas épocas donde evolucionaron poco. Pero no sería hasta la llegada de la primera interfaz con teclado, monitor y ratón cuando se marcaría un punto de inflexión y se iniciaría el mundo de la informática tal y como lo conocemos hoy en día. Pero lo mejor está por llegar. Las investigaciones en tecnología de reconocimiento del habla o en la robótica junto con el desarrollo de la nanotecnología abren nuevas ventanas y caminos al futuro, permitiéndonos el desarrollo de nuevas aplicaciones y dispositivos que sin lugar a dudas incidirán en nuestras vidas.

Uno de los ámbitos donde las interfaces cerebro-máquina o brain-computer interfaces (BCI) han llegado a tener una gran penetración ha sido en el de la discapacidad. Se desarrollan actualmente una gran variedad de aplicaciones basadas en BCI para este público. Concretamente para personas que poseen una discapacidad grave de su sistema motor (paraplégicos) se consiguen aplicaciones eficaces basadas en EEG (electroencefalograma). Es con esta motivación con la que se afronta nuestro proyecto, de cara satisfacer las necesidades de comunicación que un usuario de estas características pueda necesitar, haciendo hincapié en el control de dispositivos y consiguiendo una adaptación personal de la interfaz con el usuario.

El propósito de este proyecto será el de crear una interfaz que permita controlar un robot básico a través de estímulos visuales periódicos que generan potenciales evocados visuales. Estos estímulos estarán adaptados a cada usuario y generarán unos patrones determinados en el cerebro reconocibles en la señal de EEG, los cuales analizaremos e interpretaremos traduciéndolos en comandos que enviaremos al robot. De este modo conseguiremos una prueba de concepto que permita esperar que en un futuro cercano una persona parapléjica pueda controlar una silla de ruedas o pueda mover un brazo robótico mediante un BCI personalizado.

1.2.Objetivos.

El objetivo de este proyecto es el diseño y control de un robot a través de una interfaz cerebro-máquina de bajo coste basada en estimulación visual que sea capaz de adaptarse al usuario logrando de esta manera un control personalizado y eficiente del robot.

Para llegar a conseguirlo, se han realizado de manera progresiva pequeños hitos que nos llevarán a la consecución de nuestro objetivo final:

- Primeramente se ha realizado un estudio del estado del arte, informándonos acerca de las tecnologías existentes y los métodos de análisis para la adquisición de las señales cerebrales. Este estudio nos ha proporcionado una visión actual del ámbito en el que se desarrolla el proyecto para tomar las decisiones oportunas de diseño.
- Una vez informados de la panorámica de las interfaces cerebro-máquina (BCI), pasaremos a elegir nuestro robot o dispositivo que queremos controlar, el tipo de tarjeta con la que lo controlaremos, así como los medios de adquisición de la actividad cerebral.
- El siguiente paso o hito importante será el diseño y construcción de nuestro robot, integrando la tarjeta Arduino Yún para su locomoción y la placa auxiliar diseñada para el correcto funcionamiento y entrega de potencia a la placa y motores. Además se testeará la locomoción y el correcto funcionamiento del robot con pequeños programas cargados en la tarjeta.

- Otro hito de especial relevancia es el envío de comandos al robot a través de la WiFi con el protocolo adecuado. Las pruebas pertinentes para la correcta adquisición de datos se harán a través de teclado, logrando de esta manera un primer control del robot.
- Una vez controlado el robot de forma inalámbrica, nos centraremos en lograr un control de la estimulación visual del usuario por medio de 4 luces parpadeantes en pantalla que evocarán los correspondientes estímulos en la parte occipital del cerebro.
- Partiendo de la estimulación por pantalla, recogeremos la señal de EEG con un casco inalámbrico de bajo coste y fácil colocación que nos permitirá realizar experimentos con mayor comodidad respecto a otros modelos.
- Ahora toca integrar el control con la interfaz cerebro máquina, de manera que al estimular al usuario, se analizará oportunamente la señal de EEG adquirida detectando el SSVEP y se logrará enviar un comando específico al robot.
- Una vez tenemos un primer control de nuestro dispositivo, nos embarcaremos en la tarea de conseguir una adaptabilidad al usuario, esto lo lograremos a partir de un proceso de selección de frecuencias con el sistema de estimulación visual SSVEP. En este proceso se intentará seleccionar las 4 mejores frecuencias para cada usuario que le proporcionen un control eficiente del robot.
- Para finalizar validaremos la selección de frecuencias aplicada al control del robot, así como testaremos dicho control a lo largo de tres circuitos basados en un cadena de comandos que el robot deberá realizar. El tiempo de realización del circuito, así como el número de errores cometidos nos determinarán en gran medida el funcionamiento del sistema y podremos concluir si el sistema resulta adecuado para integrarlo en el uso de control de robots y, en un futuro, en dispositivos para discapacitados.

1.3. Estructura de la memoria.

La memoria se encuentra dividida en 9 capítulos, cuya temática se encuentra descrita a continuación:

- ❖ El *Capítulo 2* está dedicado al estudio del estado del arte, en el que se revisan las tecnologías y métodos actuales existentes para la adquisición de la actividad cerebral así como se hace hincapié en el tipo de señal cerebral usada (SSVEP). Por último se da una perspectiva de los campos a los que van enfocadas las aplicaciones BCI actuales.
- ❖ Llegados al *Capítulo 3* presentaremos los elementos básicos que usaremos para controlar el robot y para adquirir las señales cerebrales. Además, detallaremos cómo se ha realizado el diseño y construcción de nuestro robot.
- ❖ A lo largo del *Capítulo 4* iremos desarrollando los diferentes procesos del sistema BCI desde la estimulación, la adquisición de la señal de EEG con SSVEPs, pasando por el análisis de la señal, transformación en un comando y el envío de este al robot para producir su movimiento.
- ❖ El *Capítulo 5* está dedicado a las pruebas con diferentes sujetos para comprobar el grado de adaptabilidad del sistema al usuario, así como el control del robot. Las pruebas se basan en dos partes fundamentales: una primera en la que se intenta adaptar la interfaz al sujeto y una segunda parte en la que con esta adaptación se verificará el funcionamiento de la interfaz pidiendo al usuario que realice una cadena determinada de comandos.
- ❖ Para determinar si el sistema BCI es adecuado para extender su uso a personas discapacitadas o controlar robots, presentaremos en el *Capítulo 6* una serie de conclusiones referentes a los hitos u objetivos establecidos al diseñar el sistema.

- ❖ Finalmente, en el *Capítulo 7* se define una sección de estudio futuro donde abordaremos los posibles campos hacia donde se puede continuar la investigación de este u otros proyectos similares.
- ❖ En este *Capítulo 8* se listan las referencias usadas para el estudio y desarrollo de este trabajo.
- ❖ En la sección de *Anexos* se aportan las partes más relevantes del código utilizado para entender cómo hemos logrado el control del robot.

2. Estado del arte.

2.1. BCI: Interfaces Cerebro-Máquina.

Una interfaz cerebro-máquina es un medio de comunicación basado en un sistema hardware y software que nos permite controlar ordenadores o dispositivos externos [1]–[3]. Con este sistema una persona se puede comunicar sin tener que usar los medios que implican a nervios y músculos, lo que hace de este sistema el único medio posible de expresión para una persona que sufre por trastornos motores severos tales como la esclerosis lateral amiotrófica (ALS), derrame cerebral, lesión de médula espinal u otras enfermedades que dañen los propios músculos o las funciones principales que permiten su control. Por lo tanto un BCI proporciona un medio de comunicación a cualquier persona que padece enfermedades de parálisis debidas a problemas neurológicos o neuromotores.

Esta comunicación se hace fundamentalmente por medio de señales generadas por actividad *electroencefalográfica*. Un BCI por lo tanto crea un canal de comunicación no muscular que permite a las personas comunicar sus intenciones a dispositivos externos tales como ordenadores, sintetizadores de voz, dispositivos de asistencia, o neuroprótesis robóticas (atractivas para individuos con discapacidades severas del sistema motor). Este tipo de interfaz proporciona una calidad de vida mayor y al mismo tiempo reduce el coste derivado de la atención constante a estas personas.

Las investigaciones en interfaces cerebro-máquina no han gozado de gran interés a lo largo de la historia debido a que la idea de descifrar los pensamientos o intenciones por medio de actividad cerebral no fue bien acogida por considerarse difícil y remota. Por lo tanto, la investigación en este campo se ha limitado en gran medida al ámbito clínico o de laboratorio, analizando trastornos neurológicos o explorando las funciones cerebrales. Los BCI fueron considerados demasiado complejos debido a la limitada resolución y fiabilidad de información que era detectada en el cerebro. Además, estos sistemas requieren procesado de señal en tiempo real y hasta hace poco no existía la tecnología necesaria o era extremadamente cara. El desarrollo del hardware y software cada vez menos costosos ha permitido un análisis online más sofisticado. El uso de los BCIs como tecnología auxiliar para ayudar a las personas discapacitadas ha hecho incrementar el interés en este campo y la necesidad de acelerar su progreso [1], [4].

Algunas pequeñas compañías como Emotiv [5] o NeuroSky ya han desarrollado aplicaciones para el público general; aunque a pesar de estos avances, la mayoría de aplicaciones se encuentran limitadas a su uso en laboratorios. Para expandir el uso de estos sistemas es necesario que el coste, y los tiempos de preparación, entrenamiento y calibración se reduzcan considerablemente. La investigación de BCIs es un campo multidisciplinar relativamente joven que integra investigadores de neurociencia, fisiología, psicología, ingeniería, ciencias de la computación, rehabilitación así como otras disciplinas y técnicas de cuidados de salud.

Una interfaz cerebro-máquina (BCI) requiere unos ciertos patrones en señales cerebrales siguiendo cinco etapas principales propias de cualquier BCI[6]:

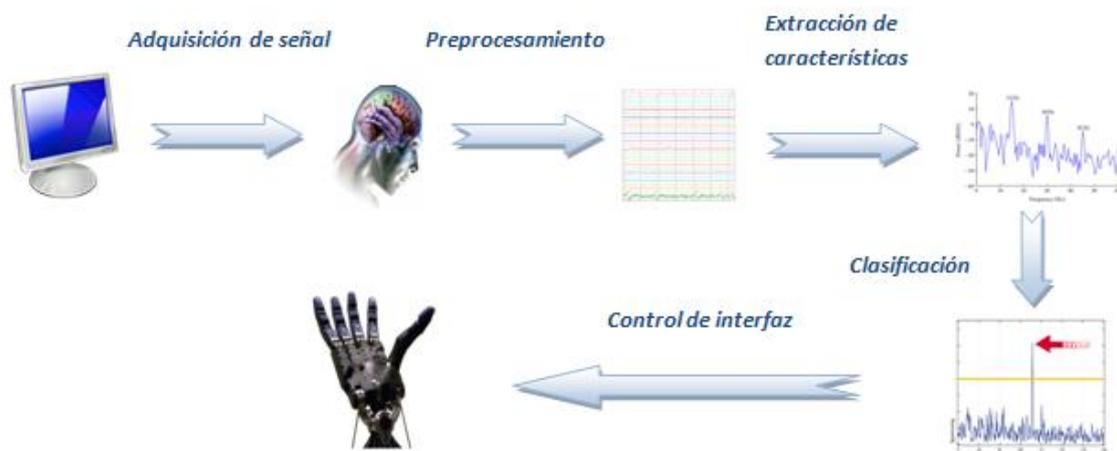


Figura 1: Esquema de etapas de un BCI.

1. **Adquisición de señal:** En esta fase se capturan las señales cerebrales y se realiza una reducción del ruido y el procesamiento de los artefactos.
2. **Preprocesamiento y realce de señal:** Aquí se preparan y acondicionan las señales de una manera adecuada para su posterior procesamiento.
3. **Extracción de características:** Es la fase donde se extrae la información específica de las señales cerebrales que han sido grabadas según el estudio a realizar. La señal es mapeada en un vector que contiene las características que hemos discriminado para extraerlas después. La extracción de la información

de interés en señales de EEG está considerada por su complejidad como una tarea ardua.

4. Clasificación: En esta etapa se clasifican las señales teniendo en cuenta los vectores de características. La elección de unas buenas características discriminativas será esencial para lograr un efectivo reconocimiento de patrones que nos permita descifrar las intenciones del usuario.

5. Control de la interfaz: Finalmente se traducen las señales clasificadas en comandos significativos a cualquier dispositivo conectado, como puede ser un ordenador, una silla de ruedas o en nuestro caso un dispositivo robótico.

2.2. Registros de actividad cerebral.

Las interfaces cerebro-máquina usan las señales cerebrales para obtener información acerca de las intenciones del usuario. Esta tarea requiere registrar y analizar online la actividad cerebral. Estos registros de actividad cerebral pueden ser de dos tipos: **electrofisiológicos** o **hemodinámicos**.

La respuesta **hemodinámica** es un proceso por el cual la sangre libera glucosa para activar neuronas. La glucosa y el oxígeno que se entrega a través de la sangre producen un excedente de oxihemoglobina en las venas y un cambio apreciable de oxihemoglobina en desoxihemoglobina. Estos cambios son captados por métodos de neuroimagen, como la resonancia magnética funcional (fMRI). Se conocen como métodos indirectos de registro por no estar directamente relacionados con la actividad cerebral.

La actividad **electrofisiológica** es generada por neuronas y suele ser medida a través de la **electroencefalografía** (EEG), electrocorticografía (ECoG) o magnetoencefalografía (MEG). Quizás el método más usado de los citados sea el de la electroencefalografía debido a su gran resolución temporal, bajo coste y conllevar menos riesgos para los usuarios [1].

2.2.1. Métodos invasivos y no invasivos.

Muchos pacientes de hoy en día que se encuentran parcial o severamente discapacitados han tenido que buscar formas de comunicarse con el medio que les rodea así como una interfaz que les permita el control de neuroprótesis o sillas de ruedas. **Métodos no invasivos** como la *Magnetoencefalografía (MEG)* o la *Electroencefalografía (EEG)* han sido satisfactoriamente usados para este propósito. Sin embargo, a pesar de la gran utilidad de estos métodos en los BCIs, el ámbito de desarrollo de BCIs para las discapacidades que implican el uso del sistema motor, se ha visto limitado debido a la necesidad de captar señales cerebrales con robustez y mayor resolución.

Los **métodos invasivos** de adquisición de señales como la *electrocorticografía (ECoG)* fueron introducidos en un esfuerzo por mejorar la calidad de las señales cerebrales monitoreadas por los BCIs. Los métodos invasivos permiten en muchos casos lograr un control eficiente y preciso de las neuroprótesis. Aunque estos métodos invasivos gozan de una gran resolución y potencia, poseen una gran desventaja al tenerse que implantar microelectrodos dentro del cráneo, lo que supone evidentemente un enorme riesgo para los usuarios quedando esta técnica relegada únicamente al ámbito experimental en hospitales para casos de discapacidad extremadamente severa. Quizás en un futuro la nanotecnología pueda desarrollar nano-detectores que se implanten en el cerebro de una forma menos invasiva para proporcionar una solución definitiva a los problemas de aplicaciones invasivas a largo plazo.

A continuación se presenta una tabla comparativa con los diferentes métodos de neuroimagen para adquirir señales cerebrales [1]:

Neuroimaging method	Activity measured	Direct/ Indirect Measurement	Temporal resolution	Spatial resolution	Risk	Portability
EEG	Electrical	Direct	~0.05 s	~10 mm	Non-invasive	Portable
MEG	Magnetic	Direct	~0.05 s	~5 mm	Non-invasive	Non-portable
ECoG	Electrical	Direct	~0.003 s	~1 mm	Invasive	Portable
Intracortical neuron recording	Electrical	Direct	~0.003 s	~0.5 mm (LFP) ~0.1 mm (MUA) ~0.05 mm (SUA)	Invasive	Portable
fMRI	Metabolic	Indirect	~1 s	~1 mm	Non-invasive	Non-portable

Tabla 1: Métodos de neuroimagen para detectar actividad cerebral. Adaptada de [1].

2.2.2. La electroencefalografía (EEG).

Una vez conocidos los dos métodos principales de adquisición de datos de las interfaces cerebro-máquina, nos decantaremos por usar una técnica no invasiva pero adaptada a cada usuario. Aunque existen varios métodos asociados a las técnicas no invasivas hablaremos principalmente de la *magnetoencefalografía (MEG)* y más ampliamente de la *electroencefalografía (EEG)* que será el método que usaremos en nuestro proyecto para detectar la actividad cerebral.

La *magnetoencefalografía* o MEG es una técnica de monitoreo no invasiva que registra la actividad magnética del cerebro por medio de inducción magnética. Las corrientes intracelulares que fluyen a través de la morfología de las neuronas producen campos magnéticos que son medibles a través de *magnetoencefalografía* fuera de la cabeza. Podríamos decir que los procesos neurofisiológicos que producen señales de tipo MEG son iguales a los que producen señales EEG. La gran ventaja de usar MEG es que los campos magnéticos son menos distorsionados por el cráneo y el cuero cabelludo que los campos eléctricos, lo que supone una gran ventaja a la hora de mejorar el registro de señales, pues el ruido introducido en el sistema es menor. Sin embargo, a pesar de estas ventajas, MEG no suele ser usado en el diseño de los BCI actuales debido al uso de una tecnología muy voluminosa y cara, resultando difícil de adquirir y poco adecuada para el uso diario. Además, requiere una gran protección contra interferencias electromagnéticas teniendo que diseñar o ser montado en habitaciones especiales para tal efecto. Estas últimas características son las que hacen que el uso de la *magnetoencefalografía* suela estar limitada a hospitales y clínicas [1].

Por todo lo expuesto anteriormente la técnica MEG queda descartada como método posible de adquisición de señales para nuestro proyecto, usaremos registros de EEG que cuentan con bastantes características ventajosas y se adecuan perfectamente a lo que necesitamos.

La *electroencefalografía (EEG)* es el método de adquisición de señal más extendido hoy en día en BCIs, ya que mide la actividad cerebral de diferentes áreas del cerebro a través de una serie de electrodos alojados en el cuero cabelludo. Esta actividad cerebral es producida por el flujo de corrientes eléctricas durante las excitaciones sinápticas en las neuronas. Las señales atraviesan el cuero cabelludo y el cráneo, lo que resulta en una pobre calidad de señal en los electrodos de registro. Además, estas señales se suelen ver afectadas por el ruido de fondo generado en el

interior del cerebro, debido a que la amplitud de señal registrada es del orden de microvoltios. Se miden como la diferencia de potencial entre la señal en un electrodo dado y en uno de referencia [1].

La interfaz electrodo-tejido se suele comportar no sólo de manera resistiva sino capacitiva, comportándose como filtro paso bajo. En la mayor parte de los casos se utiliza gel EEG o un líquido conductor para reducir la impedancia creada en esta interfaz debido a diversos factores como la temperatura, la superficie del electrodo o la propia interfaz ya citada.

Las señales de EEG pueden ser clasificadas en bandas según sea su frecuencia. Estas bandas de frecuencia ordenadas de menor a mayor frecuencia son:

- **Delta (δ):** Su rango se encuentra por debajo de los 4Hz. Suelen ser apreciables en bebés o en adultos en estado de sueño profundo. Se confunden a veces con los artefactos causados por músculos, cuello o mandíbula.
- **Theta (θ):** Las ondas theta se encuentran dentro de la banda de los 4 a los 7Hz. Se suelen dar mayormente en personas jóvenes y decrecen con la edad. Esta banda ha sido asociada a la meditación, así como a un gran número de procesos cognitivos como el cálculo mental.
- **Alpha (α):** Los ritmos alpha se encuentran en la región occipital del cerebro y están dentro del rango entre los 8 y los 12 Hz. Su amplitud se incrementa cuando se mantienen los ojos cerrados y el cuerpo se relaja y se atenúan cuando los ojos están abiertos o se produce un esfuerzo mental.
- **Beta (β):** Los ritmos beta caen dentro de la banda entre los 12 y los 30 Hz, suelen estar asociadas con actividad consciente y concentración activa.
- **Gamma (γ):** Las ondas gamma pertenecen al rango de entre los 30 y los 100 Hz. Estos ritmos no suelen ser usados en los sistemas BCI basados en EEG ya que les afectan los artefactos provenientes del movimiento de los músculos y los ojos. Pero podría ser un rango para introducir en investigaciones futuras puesto

que la actividad gamma comparada con las señales beta o alfa puede incrementar la tasa de transferencia de información así como ofrecer una alta resolución espacial. Suele estar asociada en adultos sanos a funciones motoras o de percepción.

2.2.3. BCI endógenos y exógenos.

Otro aspecto relevante relacionado con la actividad cerebral en BCIs se refiere al tipo de señales utilizadas para su control, lo que nos determinará la estructura de nuestro de BCI. Estos sistemas serán de tipo *exógeno* o *endógeno*.

Los BCI de tipo **exógeno** estimulan al sujeto con un estímulo externo consiguiendo de esta manera evocar actividad neuronal específica. Un tipo de estímulo externo muy común usado en los BCIs de tipo exógeno son los estímulos visuales que producen VEPs o potenciales visuales evocados. Los VEPs tienen la característica de que al ser visuales sabemos la zona del cerebro donde se producirá la actividad cerebral. Estos tipos de BCIs no requieren un entrenamiento demasiado exhaustivo y pueden llegar a lograr altas tasas de transferencias por encima de los 60 bits/min [1].

En contraste, los BCIs **endógenos** no requieren de un estímulo externo, sino que el usuario debe aprender la habilidad de producir voluntariamente patrones específicos que puedan ser detectados por la interfaz. Esto normalmente se implementa con retroalimentación de manera que el usuario aprenda a modular voluntariamente ciertos patrones específicos. Los entrenamientos en este tipo de BCIs suelen ser más largos y específicos que para un sistema exógeno, aun así el tiempo de un entrenamiento siempre será dependiente del sujeto y de la estrategia experimental usada que puede consistir en *condicionamiento operante* o *mantenimiento de tareas mentales específicas* [7]. Los patrones generados se decodifican mediante sistemas BCIs basados en modulaciones de los ritmos sensoriomotres o en potenciales corticales lentos (SCP) [8]. Este tipo de BCIs son especialmente útiles para usuarios en fases avanzadas de Esclerosis Lateral Amiotrófica (ALS) [9]. La ventaja de los BCIs endógenos es que el usuario puede operar libremente sobre la interfaz, por ejemplo moviendo un cursor en un espacio multidimensional, mientras que un BCI exógeno está limitado a las opciones presentadas.

Enfoque o tipo	Señales cerebrales	Ventajas	Desventajas
BCI Exógeno	SSVEP	<ul style="list-style-type: none"> Entrenamiento mínimo. Control de señal fácil Alta tasa de transferencia (60 bits/min). 	<ul style="list-style-type: none"> Atención permanente al estímulo. Puede causar cansancio en algunos usuarios.
	P300	<ul style="list-style-type: none"> Se puede grabar únicamente con un canal EEG 	
BCI Endógeno	SCPs	<ul style="list-style-type: none"> Independiente de cualquier tipo de estimulación. Puede ser operado libremente. Útil para usuarios con órganos sensoriales afectados. 	<ul style="list-style-type: none"> Tiempo de entrenamiento muy largo (semanas o meses). No todos los usuarios son capaces de obtener el control. Necesaria adquisición de señal EEG por múltiples canales. Baja tasa de bit (20-30 bit / min).
	Ritmos sensoriomotores	<ul style="list-style-type: none"> Adecuado para aplicaciones de control de cursor. 	

Tabla 2: Tipos de señales cerebrales según el tipo de BCI exógeno o endógeno. Adaptada de [1].

Los sistemas BCI tienen otras clasificaciones de relevancia como pueden ser su clasificación en *síncronos* o *asíncronos*. Los BCIs síncronos analizan señales durante tiempos de ventana predefinidos. Además el usuario es capaz de mandar comandos durante periodos específicos determinados por el sistema (este será el caso de nuestro sistema BCI, siendo además de tipo exógeno). Sin embargo los sistemas BCIs asíncronos analizan señales cerebrales sin importar cuando el usuario actúe gozando de un modo más natural de interacción entre humano y máquina. Estos sistemas asíncronos requieren un procesamiento mayor y más complejo [1].

2.2.4. Artefactos en el registro.

Los artefactos en registros de EEG son señales indeseables que se cuelan en los registros de la actividad cerebral durante su adquisición disminuyendo el rendimiento de los sistemas BCI. Los artefactos se pueden clasificar en dos grupos: *fisiológicos* y *técnicos* o *no fisiológicos*.

Los artefactos **fisiológicos** son debidos normalmente a actividad de tres tipos:

- **Muscular** : estos artefactos se deben a actividad eléctrica causada por la contracción muscular.
- **Ocular** : estos artefactos son causados por el movimiento de los ojos.
- **Cardíaca** : estos artefactos reflejan la actividad cardíaca introduciendo una señal de ritmo cardíaco en la actividad cerebral.

Algunos de estos artefactos pueden ser evitados avisando a los pacientes para que eviten movimientos del cuerpo durante los experimentos.

Los artefactos **técnicos** o **no fisiológicos** son principalmente atribuidos a ruido externo o cambios en las impedancias de los electrodos. Estos artefactos pueden ser evitados mediante técnicas apropiadas de filtrado o protección.

2.3.SSVEP: Steady State Visually Evoked Potentials.

Para la mayoría de los usuarios resulta muy difícil adaptarse a las interfaces tradicionales porque no son capaces generar actividad cerebral lo suficientemente reconocible por la interfaz, o les resulta complicado reproducir patrones suficientemente parecidos para su correcta clasificación. Por esta razón, nuestro sistema de BCI se basará en potenciales visuales evocados (SSVEPs). Y para producir la tan requerida adaptabilidad al sujeto clasificaremos cuáles son los estímulos a los que es más sensible a través de un proceso de selección de frecuencias [10].

2.3.1. Tipos de señales de control.

La finalidad de un BCI es proporcionar un medio de comunicación al usuario de manera que se interpreten sus intenciones por medio del monitoreo de su actividad cerebral. Gracias a múltiples estudios se ha descubierto un vasto grupo de señales cerebrales que pueden ser usadas como señales de control en sistemas BCI. Entre ellos destacan los *Potenciales Visuales Evocados (VEP)* [11] y los *P300* pertenecientes a BCI de tipo exógeno que será el que utilizaremos en nuestro proyecto.

Señal	Fenómenos fisiológicos	Número de opciones	Entrenamiento	Tasa de transferencia de información
VEP	Modulaciones de la señal cerebral en la corteza visual	Alto	No	60-100 bits/min
SCP	Desplazamiento de bajos voltajes en las señales cerebrales	Bajo (2 o 4)	Sí	5-12 bits/min
P300	Picos positivos debido a estímulos infrecuentes	Alto	No	20-25 bits/min
Ritmos Sensoriomotores	Modulaciones en ritmos sensoriomotores sincronizados a actividades motoras	Bajo (2,3,4,5)	Sí	3-35 bits/min

Tabla 3: Tipos de señales de control y número posible de opciones. Adaptada de [1].

Lo **VEPs** son potenciales eléctricos que ocurren en la corteza visual en la parte occipital del cerebro después de recibir un estímulo visual parpadeante. Los estímulos que generan VEPs a su vez se pueden clasificar según dos criterios:

- Morfología del estímulo: según este criterio el estímulo visual puede ser causado por *estimulación flash*, es decir, una luz uniforme parpadeante o usando *patrones combinados con luz parpadeante* como puede ser un tablero de ajedrez o un mapa de puntos aleatorios.
- Frecuencia del estímulo: según este criterio los estímulos pueden ser de *régimen transitorio (TVEP)* que ocurren por debajo de los 6 Hz o de *estado estacionario (SSVEP)* que ocurren como reacción a un estímulo de mayor

frecuencia (mayor o igual a los 6Hz). La respuesta evocada variará según el estímulo presentado. Los flash TVEPs presentan una serie de picos positivos y negativos, de los cuales los más prominentes son los N2 (90ms) y P2 (120ms) [12].

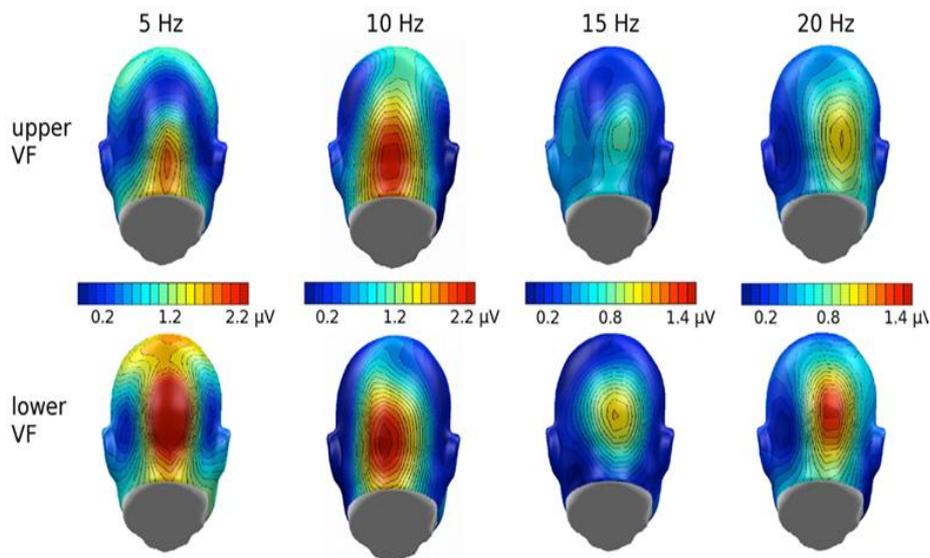


Figura 2: Actividad cerebral como respuesta a estímulos visuales evocados a diferentes frecuencias [42].

Los **P300** son otra serie de potenciales evocados consistentes en picos en la señal de EEG debido a estímulos infrecuentes de origen auditivo, visual o somatosensorial. Son señales de tipo exógenas y suelen ser excitadas en el cerebro después de 300ms de recibir el estímulo, de ahí su nombre. Sin embargo, la tasa de transferencia de información en los BCIs basados en P300 suele ser muy baja. La última característica a destacar de los P300 es que su respuesta no se ve demasiado afectada si el usuario centra su mirada en el estímulo, en contraste con los VEPs, aunque si se ve mejorada cuando se focaliza la mirada en el estímulo.

2.3.2. Interfaces basados en SSVEPs

La idea de usar SSVEPs para controlar un sistema BCI data de hace 30 años. En aquella época se propuso un sistema BCI precursor que usaba un bucle cerrado retroalimentado para controlar el contraste de estímulo patrón actuando directamente sobre la amplitud del SSVEP [13]. La aplicación real a los BCI fue introducida más tarde por Cheng y Gao en 1999 [12]. El sistema BCI procesa los SSVEPs

se producen en la parte occipital del cerebro mientras el usuario centra su atención en una de las 4 luces parpadeantes a diferentes frecuencias, cada una de ellas codifica un comando diferente, el ejemplo en nuestro caso sería: arriba, parar, izquierda y derecha. La frecuencia de un SSVEP corresponde a la del estímulo donde el sujeto concentra su atención visual, es decir, si el sujeto mira una de las luces que está, por ejemplo, parpadeando a 11Hz, el SSVEP registrado tendrá esa misma frecuencia de 11Hz. Estas señales son procesadas para controlar un robot pero también se pueden usar para controlar un cursor en la pantalla. El concepto general es asociar comandos de usuarios a respuestas cerebrales a estímulos de luz parpadeante que inducen SSVEPs a diferentes frecuencias. El usuario conseguirá mover nuestro robot o el cursor focalizando su atención en una de las luces que parpadean simultáneamente, y analizando el SSVEP generado en el cerebro el sistema BCI intentará identificar qué comando ha seleccionado el usuario [12].

Una de las razones por las que usaremos en nuestro proyecto un BCI basado en SSVEP y no en un sistema de seguimiento visual (gaze-tracking) es que este sistema posee limitaciones para algunos tipos de discapacidades como los pacientes con LIS y también en algunas aplicaciones BCI como por ejemplo el control de una silla de ruedas.

2.4.Aplicaciones.

Las aplicaciones de BCIs constituyen una parte fundamental del estudio e investigaciones en el campo de la Neurociencia. Y es que hoy en día son muchos los investigadores que centran su atención en la construcción de aplicaciones basadas en BCI principalmente con la esperanza de que puedan servir de ayuda para aquellas personas con discapacidades severas del sistema motor. Gracias a avances significativos en el área de los BCI basados en EEG, se han desarrollado recientemente una gran variedad de aplicaciones BCI [1], [4].

El público general al que van dirigidas las aplicaciones BCI suele clasificarse en tres grupos [1]:

- **Complete Lock-In State (CLIS):** pacientes que han perdido totalmente el control de su sistema motor.
- **Locked_in State (LIS):** pacientes que han perdido casi toda la movilidad del cuerpo, pudiendo sólo realizar algunos pequeños movimientos tales como mover o guiñar los ojos, contracciones labiales, etc.
- **Personas discapacitadas parcialmente:** Personas que aunque se encuentren discapacitadas poseen un control substancial sobre el sistema motor como por ejemplo el habla o el control de un brazo.

Aunque hoy en día existen muchos tipos de aplicaciones BCI, la mayoría se suelen diseñar sólo con propósitos de entrenamiento o demostración. Su explicación la encontramos en que todavía quedan muchos retos para que el control BCI se pueda emplear en tareas dentro del mundo real. Es en estos puntos donde se debe incidir para lograr una inserción cada vez mayor de estos sistemas en nuestra vida diaria [14].

- La tasa de transferencia proporcionada por los BCIs es demasiado baja.
- La alta tasa de error todavía complica la comunicación en algunos casos.
- Los sistemas BCI no pueden ser usados de manera autónoma por discapacitados pues requieren de una persona que les asista a la hora de la colocación del casco y los electrodos.
- El manejo de las aplicaciones BCI demanda una alta carga cognitiva que puede ser lograda por usuarios en un laboratorio pero no en el mundo real.

Como ya hemos citado existen numerosas aplicaciones BCI y de diferentes tipos según su aplicación. Tiene sentido que destaquemos los campos de aplicación más importante y evaluemos que uso se puede hacer de estos sistemas dependiendo de las capacidades del usuario y la tasa de transferencia de información del BCI.

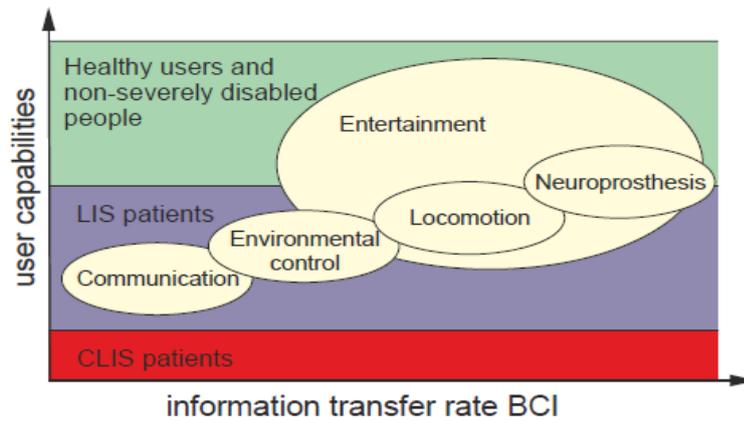


Figura 3: Entornos de desarrollo de aplicaciones BCI dentro de gráfica capacidades del usuario Vs flujo de transferencia de información. Adaptada de [1].

Como se puede apreciar en la figura para los pacientes con una discapacidad total del sistema motor (CLIS) no existen aplicaciones puesto que es difícil encontrar una forma de comunicación en este caso.

Para los pacientes que conservan algún tipo de movimiento de su sistema motor se desarrollan aplicaciones enfocadas sobre todo a su comunicación y al control de su entorno (domótica). Además también se suelen desarrollar aplicaciones de locomoción, neuroprotésicas y de ocio, aunque estas últimas suelen ser creadas más especialmente para personas con discapacidades no severas y usuarios sanos.

➤ **Comunicación a través de un BCI.**

La comunicación es una actividad esencial en la vida del ser humano es por esta razón que este campo es en el que más se incide a la hora de investigar y desarrollar aplicaciones BCI.

La idea general de las aplicaciones BCI desarrolladas para este campo se basa en una operación que despliega un teclado virtual, donde el usuario es capaz de seleccionar una letra de un alfabeto existente. Lo que distingue cada uno de estos BCIs será el tipo de señal de control y el BCI.

Otro tipo de aplicación dentro de este entorno es el de los navegadores, debido a que internet ha llegado a ser una parte muy importante en nuestra vida diaria. "Descartes" es considerado el primer navegador controlado por EEG. Su interfaz está basada en una ordenación alfabética en un árbol de decisión donde el usuario selecciona o rechaza cada ítem, produciendo de este modo picos positivos o negativos que producen desplazamientos SCP [1].

➤ **Control del entorno.**

Uno de los principales objetivos de las aplicaciones basadas en BCI es la adaptabilidad del usuario al entorno, así como conseguir un cierto grado de independencia a pesar de sus discapacidades motoras. Las personas que padecen graves discapacidades motoras a menudo suelen estar confinadas en su casa, es por esta razón que las aplicaciones de control del entorno resultan tan importantes. Tareas como encender o apagar una luz, el control de un termostato, el manejo del televisor o subir y bajar persianas deben estar al alcance de cualquier persona independientemente de su discapacidad. Micrófonos reconocedores de voz o seguidores de movimientos de cabeza están disponibles para personas con extremidades deterioradas pero tienen intactos los músculos del cuello y disponen de la capacidad del habla. Para casos de discapacidad total el sistema puede ser controlado medio de un BCI basado en EEG. El sistema permite el control domótico de luces, TV y equipos de música, cama monitorizada, alarma acústica, puertas, teléfonos o videocámaras inalámbricas para controlar el entorno [15], [16].

➤ **Restauración del sistema motor.**

La *lesión de médula espinal* o *Spinal Cord Injury (SCI)* u otros problemas asociados a la pérdida de funciones del sistema motor disminuyen dramáticamente la calidad de vida de las personas que los padecen y les crean una dependencia de por vida. La restauración motora alivia de alguna manera su padecimiento psicológico. Se ha probado que la estimulación funcional eléctrica (FES) junto con sistemas BCI basados en EEG puede usarse para controlar neuroprótesis de manera efectiva. FES compensa la pérdida de las funciones vitales generando contracciones musculares artificiales. Las corrientes eléctricas generan potenciales de acción por medio de despolarización en los nervios motores que inervan el músculo seleccionado y causan su contracción muscular [17].



Figura 4: Brazo protésico realizando tareas de precisión [43].

Más recientemente, los Events-Related Potentials (ERPs) o Potenciales relacionados con Evento basados en respuestas cerebrales a eventos específicos sensoriales, cognitivos o motrices [18], han sido usados para proporcionar restauración motora. Muller [19] presentó un novedoso dispositivo neuroprotésico para restauración funcional de personas con lesión de médula espinal. Este dispositivo consiste en una mano protésica controlada por un BCI basado en cuatro clases de SSVEPs. De esta manera el usuario podía seleccionar sólo cuatro movimientos correspondientes a los 4 LEDs parpadeantes a diferentes frecuencias. Una luz a 6Hz en el dedo índice y otra luz en el dedo meñique a 7Hz servían para rotar la mano. Las dos luces restantes parpadeando a 8 y 13 Hz en la muñeca permitían abrir y cerrar la mano. Como apreciamos en esta neuroprótesis, muy cercano al sistema diseñado en nuestro proyecto, predomina el uso de las bajas frecuencias en las luces [20].

➤ **Locomoción.**

Este campo de aplicación de los BCIs es uno de los más importantes y resultará básico en nuestro proyecto, pues nuestro objetivo es el de controlar la locomoción de un robot utilizando un BCI basado en EEG. Quizás la aplicación que más se desarrolla en este ámbito sea la del control de una silla de ruedas, pues la movilidad resulta básica en nuestras vidas y la portabilidad es una necesidad inherente en este tipo de aplicaciones. El uso de EEG se ha extendido en estas aplicaciones. El problema de este tipo de señales es que resultan bastante ruidosas y variables, lo que implica que se produzca un intervalo largo entre comandos. Además, a menudo no gozan de la precisión suficiente para el control de la locomoción debido normalmente a una baja tasa de transferencia de información. Una de las tareas en este proyecto será conseguir una mejora minimizando estas desventajas para un control más eficiente de nuestro robot.

➤ **Entretenimiento.**

Con respecto al ámbito del entretenimiento, podríamos decir que el diseño de aplicaciones BCI no goza de gran prioridad. Sin embargo su interés ha aumentado en la última década debido a los avances de esta tecnología; sus mejoras de rendimiento han hecho que el uso de los BCI se extienda a personas no discapacitadas consiguiendo así más penetración en el mercado y llegar a todo el público. Este tipo de aplicaciones BCI crean una nueva forma de interacción que puede convertir a los videojuegos en experiencias más atractivas y desafiantes. Además estas aplicaciones de BCI pueden llegar a detectar si el jugador se siente aburrido, ansioso o frustrado, lo que puede llegar a ser de información de gran interés para los desarrolladores de aplicaciones y videojuegos [21].



Figura 5: Jugador usando un casco de EEG en un videojuego [44].

Los potenciales evocados han llegado a utilizarse también en este campo. Middendorf [22] diseñó una simple luz estimuladora controlada por BCI basada en *SSVER (Steady State Visually Evoked Response)*. Esta aplicación era muy modesta pues sólo gozaba de dos grados de libertad con dos acciones posibles para controlar un aeroplano. Por un lado se hicieron experimentos en los que el sistema de detección se basaba en la fuerza del estímulo, y por otro lado experimentos en los que la selección de un comando u otro se hacía teniendo en cuenta a la frecuencia a la que se detectaba el estímulo. Los resultados mostraron que era preferible el segundo tipo de detección pues requería menor entrenamiento. También se llegaron a desarrollar todo tipo de juegos y aplicaciones basadas en SSVEPs como el “MindBalance” o basadas en P300 como el “MindGame” [1].

Todos estos experimentos han sido llevados a cabo de manera experimental y en contextos de investigación, aunque actualmente ya existen compañías como Emotiv o NeuroSky preparando juegos comerciales con sus respectivos cascos de adquisición de señal cerebral. Sin duda el campo del entretenimiento es un camino que se abre en el mundo de las aplicaciones para BCI y que puede desembocar más allá en sistemas que nos ayuden a entender mejor el funcionamiento del cerebro o a interactuar con el mundo de forma diferente para no tomar riesgos como puede ser el caso de los avatares o robots no tripulados. En este proyecto se utilizará el casco de EEG inalámbrico de bajo coste de Emotiv.

Por lo que vemos, existen diversos campos para los que se están realizando aplicaciones BCI; pero hay una particularidad crucial en cuanto al manejo de muchas de ellas (habitualmente): la adaptabilidad al usuario. En el caso de nuestro proyecto se implementa con SSVEPs haciendo una selección de las mejores frecuencias compatibles entre sí para cada sujeto, lo que resulta en un mejor control del BCI y del robot.

3. Diseño

3.1. Consideraciones generales. Introducción al diseño.

El objetivo de este proyecto es el del diseño de una interfaz cerebro-máquina basada en SSVEPs y adaptada al usuario cuya principal función será la del control de un dispositivo, en este caso, un robot. El hecho de que sea un dispositivo de tipo robótico supone una prueba de concepto para un posible uso futuro en personas con discapacidades severas; de modo que el control de este tipo de “robots” le sea lo más cómodo posible al usuario y le sirva de medio de interacción con su entorno. Dicho esto, nos percatamos de que una parte esencial del diseño y desarrollo de esta interfaz debe ir encaminada a la adaptabilidad al usuario, así como a su sencillez de manejo. Teniendo en consideración lo anterior se han tomado una serie de decisiones intentado que el desarrollo de este proyecto sea lo más viable posible y de bajo coste.

Antes de pasar a la justificación de algunas elecciones, se ha de recalcar que la importancia e interés del desarrollo de este proyecto no radica simplemente en el control de un robot o en evaluar un tipo de señalización biológica para un grupo de sujetos, sino en la unión de diferentes tecnologías, por un lado las encargadas de interpretar una serie de estímulos producidos por el cerebro de manera adecuada y por otro el envío inalámbrico de los datos obtenidos para el control de un dispositivo robótico, que en este caso es muy básico pero que de forma avanzada puede llegar a tener aplicaciones en el mundo de la discapacidad y en el de la robótica.

En relación al diseño del robot como ya hemos citado se ha optado por un modelo de robot muy simple que pueda realizar acciones tales como avanzar, retroceder, parar, girar a la izquierda o a la derecha. Respecto a los giros se ha de comentar que serán sobre el mismo eje vertical de rotación y no giros con avance. En cuanto al control del robot, se ha optado por utilizar una placa base Arduino, ya que se vio que estaba empezando a introducirse en el mercado con muy buenos resultados y su programación era sencilla a la vez que versátil para utilizarse con múltiples lenguajes de programación, concretamente se ha usado lenguaje C y C++ para la creación de librerías y código, aunque se debe citar que el propio Arduino tiene algunas peculiaridades a tener en cuenta.

El problema de qué protocolo de comunicaciones usar y cómo adaptarlo a la tarjeta Arduino y a la comunicación con el PC, se abordó teniendo en cuenta el lenguaje usado y las librerías soportadas por Arduino. Finalmente se decidió usar un protocolo con peticiones HTTP creando un servidor Web en la placa Arduino. El uso de navegadores de internet hoy en día está muy extendido ya sea en PC's, tablets o smartphones; esta decisión nos permite hacer extensible este proyecto a múltiples plataformas y explorar las posibilidades futuras de generalización que tiene este enfoque.

Otro de los puntos a considerar a la hora de establecer el diseño del proyecto es el tipo de señales con las que trabajaremos y qué opciones tenemos para ello. Por una parte, en cuanto a la captura de las señales cerebrales obviamente hemos optado por un modelo no invasivo. Dentro de los métodos no invasivos, la mejor opción para este proyecto es el de registro de EEG, que nos ofrece mejor resolución temporal y la mayor portabilidad y comodidad de uso de los sistemas de adquisición de datos. La adquisición de los datos se ha realizado mediante un casco inalámbrico de bajo coste con electrodos de fácil colocación y preparación con respecto a otros modelos. Como podemos observar todo irá enfocado a la usabilidad y sobre todo adaptabilidad al usuario.

Otro punto importante es el de tipo de señales utilizadas en el registro de EEG para implementar los comandos del usuario. Los SSVEPs (Steady State Visually Evoked Potentials) pertenecen a un grupo de señales de fácil captación ya que sabemos que se producen en la parte occipital del cerebro donde tenemos alojados algunos sensores del casco. Como se ha comentado en el capítulo anterior, este tipo de señales nos ofrecen una tasa de transferencia mayor que otros como pueden ser los SCP (Slow Cortical Potentials in brain signals), el P300 (utilizado para estímulos infrecuentes) o los basados en modulaciones en ritmos sensoriomotores (SRM). Quizás una de las características más importante a la hora de la elección de los SSVEP como el tipo de señales a utilizar es que pertenecen al grupo de señales cerebrales con un enfoque exógeno (al igual que P300), esto quiere decir que son señales que se producen a partir de una estimulación externa y no interna (sistemas endógenos), lo que nos da también mayores facilidades, ya que las señales no van contaminadas con tanto ruido y son más fácilmente detectables. Los artefactos que se pueden generar debido al movimiento de los ojos o movimientos faciales se pueden evitar o subsanar.

Como ya hemos citado el tipo de estimulación a usar en este tipo de interfaz es de tipo visual, lo que nos deja con dos opciones de estimulación: exterior con LEDs o por pantalla. Para este trabajo utilizaremos los estímulos visuales por una pantalla con frecuencia de refresco de 60Hz por la versatilidad que nos ofrece: portabilidad y diseño. La frecuencia de las señales SSVEP va a depender de la frecuencia del estímulo en el sistema evocador, que está limitado por la frecuencia de refresco de nuestra pantalla. Para este trabajo se parte del control de estímulos en una pantalla desarrollado en Qt previamente en el Grupo de Neurocomputación Biológica [23]. Este control se ha integrado con el resto del software.

El control óptimo del robot, va a depender en gran medida de la selección de frecuencias de estimulación, puesto que cada sujeto es más o menos sensible a determinadas frecuencias de parpadeo [24], [25]. Por lo tanto, podemos concluir que la elección de las frecuencias óptimas será crucial para un control preciso del robot.

3.2. Captura de datos mediante el casco Emotiv EPOC

Para poder adquirir las señales cerebrales producidas por los estímulos visuales necesitamos un casco de EEG con electrodos capaces de registrar los SSVEPs que se producen cuando el usuario concentra su atención visual en una luz parpadeante. Aunque el GNB dispone de un casco de licra para la adquisición de estas señales, su colocación y preparación es bastante ardua y lleva tiempo: para cada electrodo hay que raspar un poco la cabeza en el lugar del sensor aplicar un gel conductor y su entorno de desarrollo y sistema de adquisición es algo rudimentario. Por estas razones decidimos usar el casco de registro de EEG comercial Emotiv EPOC diseñado por la empresa del mismo nombre [5].

El casco Emotiv es un dispositivo multicanal que nos ofrece una alta resolución y la gran ventaja de comunicarse de forma inalámbrica y ser compatible con Windows, Linux o MAC. Su coste comercial es relativamente bajo (alrededor de unos 700 euros con el entorno SDK de investigación). Sus características completas se exponen en la siguiente tabla:

	EEG HEADSET
Number of channels	14 (plus CMS/DRL references, P3/P4 locations)
Channel names (International 10-20 locations)	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4
Sampling method	Sequential sampling. Single ADC
Sampling rate	128 SPS (2048 Hz internal)
Resolution	14 bits 1 LSB = 0.51 μ V (16 bit ADC, 2 bits instrumental noise floor discarded)
Bandwidth	0.2 - 45Hz, digital notch filters at 50Hz and 60Hz
Filtering	Built in digital 5th order Sinc filter
Dynamic range (input referred)	8400 μ V (pp)
Coupling mode	AC coupled
Connectivity	Proprietary wireless, 2.4GHz band
Power	LiPoly
Battery life (typical)	12 hours
Impedance Measurement	Real-time contact quality using patented system

Tabla 4: Características principales del Emotiv Epoc Adaptada de [37].

El sistema conjunto cuenta con 14 sensores de adquisición de datos y dos referencias. Se encuentran distribuidos según el sistema internacional 10-20, llamado así porque las distancias entre sensores son de un 10% y un 20% desde el plano que une las dos referencias [26]. La distribución se puede observar en la figura.

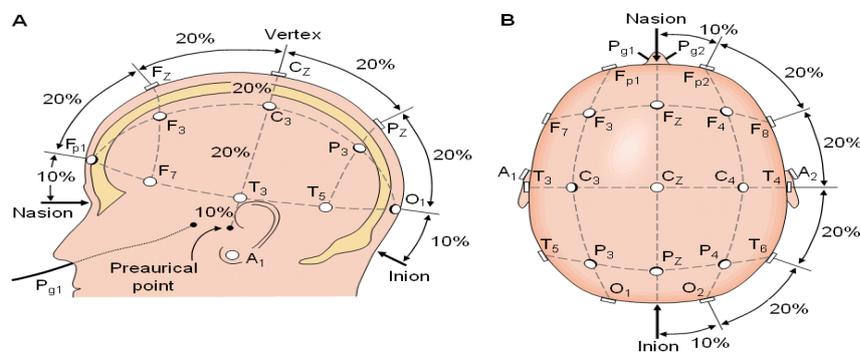


Figura 6: Distribución de sensores según el sistema internacional 10-20 [26].

El casco Emotiv consta de 14 en las posiciones según indica la figura, donde F significa Frontal, AF Anterior Frontal, FC FrontoCentral, T Temporal, P Parietal y O Occipital. CMS y DRL son los electrodos de referencia de izquierda y derecha respectivamente. El software de adquisición proporciona un diagrama que representa cada electrodo en un color: estas representaciones deben estar de color verde para confirmar que la conductancia es la adecuada y que los electrodos funcionen adecuadamente. Si están en rojo significa que no están colocados adecuadamente en la cabeza (ver Figura 14).

Cada electrodo del casco es un canal que registra actividad cerebral en ese punto. El entorno de desarrollo del casco SDK nos permite guardar los datos que registra cada canal de manera que se puedan extraer las características adecuadas según el estudio que se quiera realizar. Durante las pruebas con sujetos haremos uso de esta herramienta para la adquisición de la señal y poder saber con certeza cuándo se detectan SSVEP. Con la herramienta TestBench del SDK vemos una imagen como la de la figura donde se monitoriza la posición de los sensores que se mostrarán de color verde si la señal es perfecta, amarillo si es buena, naranja si es media, roja si es mala o insuficiente y sin color si no hay señal. Posteriormente la SDK nos permitirá acceder en tiempo real a la señal y detectar los SSVEPs para el control online del robot.

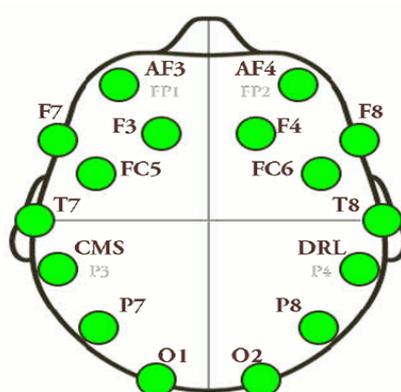


Figura 7: Distribución de los sensores del casco Emotiv EPOC [45].

Este tipo de casco cuenta con una desventaja de cara a la adquisición de señal para este proyecto al no contar con un electrodo en la posición Oz. Las señales que estamos intentando estimular SSVEP se producen en la parte occipital del cerebro justo en el punto Oz del que no disponemos electrodo. Aun así, contamos con dos electrodos occipitales que combinándolos linealmente entre ellos y con otros conseguiremos detectar el tipo de señal deseada. Estas combinaciones además nos servirán para minimizar parte del ruido que introduzcan los canales.

3.3. Diseño del robot.

Una vez comentadas las características de los registros con los que vamos a trabajar y de haber explicado algunas consideraciones acerca de los componentes y métodos utilizados para el diseño y desarrollo del proyecto, debemos explicar los componentes utilizados para el diseño y montaje del robot. Para llevar a cabo la

gestión del control se ha usado una placa Arduino Yún y para el diseño de las piezas del robot una impresora 3D.

Las primeras preguntas que pueden surgir son ¿qué es Arduino? Y ¿por qué Arduino?

3.3.1. Arduino Yún

3.3.1.1. ¿Qué es Arduino?

Arduino es una plataforma electrónica de hardware y software libre de reducidas dimensiones y de bajo coste creada especialmente para programadores, diseñadores, aficionados y cualquiera interesado en crear objetos o entornos interactivos. Se basa en una placa con un microcontrolador y un entorno de desarrollo para la creación de programas y su carga en la placa [27].

Según el uso que vayamos a darle necesitaremos un tipo de placa Arduino u otra, pero en modo general todas tienen algunas características a destacar sobre otro tipo de placas[27]:

- **Multi-plataforma:** El software de Arduino es capaz de trabajar tanto Windows como en Linux y Mac. La mayoría de las plataformas están limitadas al sistema operativo Windows.
- **Lenguaje de programación intuitivo y similar a C++:** Arduino dispone de un entorno de programación simple e fácil de usar basado en Processing que es un lenguaje similar a C++. Además, Arduino usa la transmisión serie de datos soportada por la mayoría de lenguajes de programación lo que hace a este tipo de plataforma bastante versátil: C, C++, Python, Matlab, Java, PHP,...
- **Gran número de entradas y salidas:** Aunque este aspecto dependerá de la placa que usemos, podemos afirmar que en general casi todas las placas disponen de unas dimensiones reducidas pero también de bastantes entradas y salidas analógico-digitales, ya que es como Arduino obtiene la información del entorno.

3.3.1.2. ¿Por qué Arduino Yún?

Comprendido qué es Arduino y las ventajas que ofrece, investigamos acerca de los diferentes tipos de placas existentes en el mercado y nos decantamos finalmente por una tarjeta *Arduino Yún*, pues era la única que nos ofrecía el conjunto de características que buscábamos para llevar a cabo nuestro proyecto:

- **Fácil programación y compatibilidad con C y C++:** Nos facilita mucho las cosas que el entorno de programación de Arduino sea sencillo de manejar, esté basado en C++ y acepte lenguajes similares, ya que el código está escrito en C y C++.
- **Comunicación inalámbrica:** Este es uno de los requisitos que buscamos especialmente. Queremos que nuestro robot se pueda mover libremente dentro de un área determinada. El alcance de dicha conectividad no será tan determinante para las pruebas de control del robot como lo será la propia conectividad, puesto que el área de movimiento del robot será cercana al emisor.



Figura 8: Tarjeta Arduino Yún.

- **Acoplable al diseño del robot:** Otro de los requisitos de bastante importancia es que la placa se pueda acoplar fácilmente a nuestro diseño del robot. Hay un conjunto de tarjetas que no tendrían mucho problema en encajar: UNO, MEGA, Leonardo. Nos interesa una placa fácil de acoplar a la estructura del robot y que no pese mucho y desequilibre el robot.
- **Bajo consumo para la movilidad del robot:** Aunque para el desarrollo de las pruebas de este proyecto el tiempo de uso del robot no superaba la hora, es importante tener en cuenta la autonomía. Este robot está diseñado para que se

pueda controlar durante varias horas, pudiéndose incorporar otros elementos como sensores o algún otro motor.

- **Entorno de desarrollo (IDE) abierto:** Al trabajar con Linux, necesitamos que el entorno de desarrollo de la placa donde crear el código, sea compatible con esta plataforma y Arduino nos ofrecía esta posibilidad al ser multi-plataforma.

Por estas razones, la placa Arduino que hemos elegido ha sido la Arduino Yún, por otro lado la única que cuenta actualmente en el mercado con conectividad WiFi.

La característica destacable de Arduino Yún es que dispone además del microprocesador ATmega32u4 de otro microprocesador Atheros AR9331 con sistema Linux embebido (Linino). Combinados, estos sistemas ofrecen las ventajas de un ordenador en red y la simplicidad de programación de Arduino. Además Arduino consta de: 20 pines digitales de entrada/salida (de los cuales 7 pueden ser usados como salida PWM y 12 como entradas analógicas), un ICSP header (In Circuit Serial Programming) que sirve para programar el BootLoader del microcontrolador ATmega32u4, 3 botones de Reset.

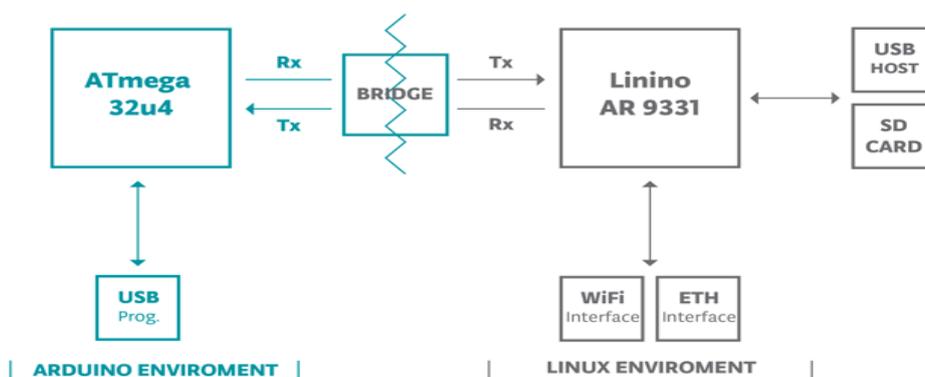


Figura 9: Esquema interno de los procesadores de la tarjeta Arduino Yún [27].

El entorno de Linux, basado en OpenWrt, es el que maneja el host USB, una tarjeta interna SD, así como Ethernet y la interfaz WiFi que se utilizará para enviarle los comandos pertinentes al robot. Este micro dispone además de una memoria RAM de 64MB y 16 MHz de memoria Flash. Por otra parte el ATmega32u4 se encarga de la gestión de los programas creados en el entorno de desarrollo de Arduino, con el cual se comunica a través de un microUSB.

Los procesadores se comunican a través de la librería Bridge, que permite controlar el Linino a los sketches que se cargan en el ATmega32u4. En las siguientes tablas se han detallado sus características:

Microcontrolador Arduino AVR		Microprocesador Linux	
Microcontrolador	ATmega32u4	Procesador	Atheros AR9331
Voltaje de operación	5V	Arquitectura	MIPS @400MHz
Voltaje de entrada	5V	Voltaje de operación	3.3V
Pines digitales de E/S	20	Ethernet	IEEE 802.3 10/100Mbit/s
Canales PWM	7	WiFi	IEEE 802.11b/g/n
Canales usados como entradas analógicas	12	USB Tipo A	2.0 Host
Corriente DC máxima por el pin de E/S	40 mA	Lector de tarjetas	Sólo Micro-SD
Corriente DC máxima para los pines a 3.3V	50 mA	Memoria RAM	64 MB DDR2
Memoria Flash	32 KB (de los cuales 4 KB son usados por el bootloader)	Memoria Flash	16 MB
Memoria SRAM	2.5 KB		
Memoria EEPROM	1 KB		
Velocidad de reloj	16 MHz		

Tabla 5: Características de los micros de la tarjeta Arduino Yún. Adaptada de [27].

Destacar en cuanto a características físicas que esta placa dispone de cuatro perforaciones para el anclaje a otra placa o soporte. Aunque dispone de un conjunto de luces como la de encendido en verde y la L13 que encenderemos en rojo un segundo como confirmación de recepción del comando del casco de EEG. Sus dimensiones son de 68 mm x 53 mm x 15 mm y pesa alrededor de 41 g. Finalmente, se ha de destacar que esta placa posee un coste relativamente bajo.

3.3.2. Diseño de la estructura, componentes y montaje

El robot que vamos a diseñar y montar como ya hemos citado será un robot con capacidad para realizar 4 movimientos básicos: avanzar, girar a izquierda o derecha y retroceder, además de parar. El movimiento del robot lo conseguimos haciendo girar sus dos ruedas a través de servomotores conectados a la tarjeta principal Arduino.

Para la elección del modelo nos hemos basado en un Miniskybot 2 [28]. Se plantearon otras dos opciones de robots posibles como pueden ser un robot tipo gusano y un robot con patas. Pero teniendo en cuenta el objetivo de control que era lograr el movimiento del robot con respecto a 4 grados de libertad a partir de registros con un casco de EEG de bajo coste y con estimulación parpadeante en una pantalla LED se optó por el Miniskybot 2 del que ya disponíamos de bastante documentación al haberse utilizado en otros proyectos en el grupo de Neurocomputación Biológica.

El primer paso para obtener nuestro robot de locomoción será diseñar las piezas principales del mismo: el chasis y las ruedas. El portabaterías no lo diseñaremos sino que lo pediremos junto con los componentes restantes a través de las tiendas online de RS o Farnell [29], [30]. Para diseñar el chasis y las ruedas utilizaremos OpenScad [31], herramienta software para el diseño de piezas 3D y que tiene un lenguaje de programación bastante fácil de usar. A veces basta con describir la pieza a crear y especificar sus dimensiones, de manera tan sencilla como `cube([width,depth,height], center)` o `sphere(radius | d=diameter)`. Una vez tenemos los diseños creados con OpenScad (formato .scad) basta con pasarlos a formato .stl para que la impresora 3D los entienda.

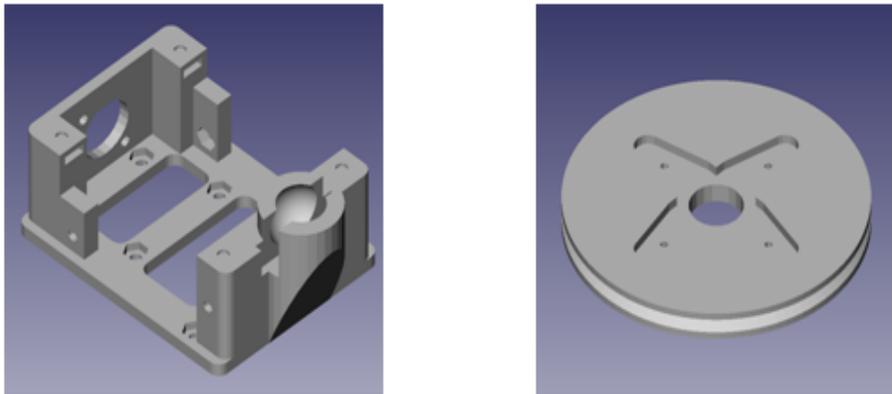


Figura 10: Diseño del chasis y las ruedas del robot mediante OpenScad [31]

Los modelos los imprimiremos con la impresora 3D del laboratorio del GNB en plástico PLA blanco. El PLA (ácido poli-láctico) es un polímero permanente, inodoro y biodegradable derivado del ácido láctico. Es un material altamente versátil, que se hace a partir de recursos renovables 100%, como son el maíz, la remolacha, el trigo y otros productos ricos en almidón. Este ácido tiene muchas características equivalentes e incluso mejores que muchos plásticos derivados del petróleo, lo que hace que sea eficaz para una gran variedad de usos.

Como podemos observar el chasis lleva unos agujeros taladrados para que se le pueda acoplar la tarjeta Arduino que lleva agujeros también en sitios similares. También apreciamos una cavidad especialmente creada para una canica que se alojará a presión y quedará suelta en su interior a modo de ruda loca trasera. Las muescas en las ruedas tiene el dibujo exacto para acoplar el modelo de servomotores Futaba 3003.

Las piezas restantes para montaje del robot: servomotores, tornillos, tuercas, alzadores para la placa y gomas para las ruedas se adquirieron a través de las tiendas de componentes ya citadas (*ver presupuesto al final de la memoria*).

Antes de proceder al montaje debemos prestar especial atención a los servomotores. Los servos que hemos adquirido tienen el ángulo de giro limitado a 180 grados, lo que nos limita el movimiento del robot en cuanto a los giros. Queremos un robot que podamos manejar libremente en cualquier dirección. Además, los sujetos o usuarios que controlen el dispositivo no tendrán un gran control sobre este, por las limitaciones que impone la decodificación de la actividad neuronal registrada con un casco de bajo coste, por lo que nos conviene que la rotación de los servomotores sea continua. Para poder tener servomotores de rotación continua debemos *trucarlos*, ya que el coste de unos íntegramente de rotación continua era mayor.

Para trucar los servos los abrimos con un destornillador y les quitamos la parte superior de manera que quede a la vista el mecanismo de engranajes. Debemos coger el engranaje del servo (de color negro) y cortar y limar el tope de plástico.

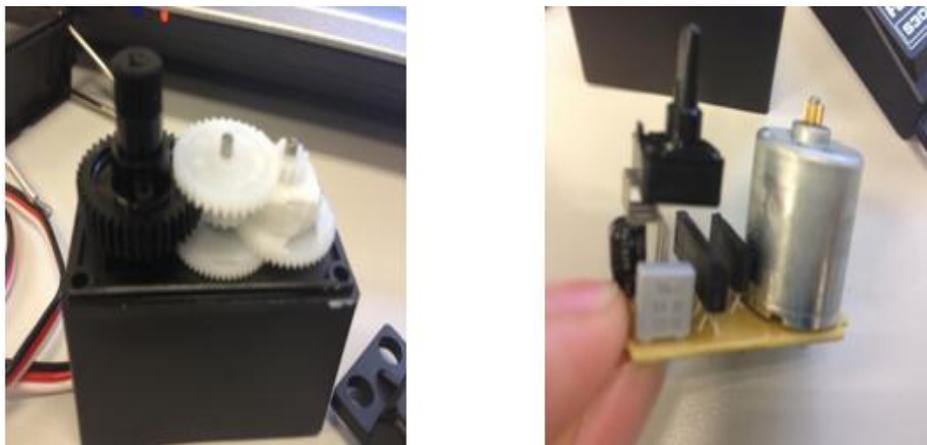


Figura 11: Fases del trucado de los

Una vez hecho esto extraemos con cuidado el circuito integrado con los componentes y desoldamos el potenciómetro existente. Como no disponemos de un potenciómetro multivuelta utilizaremos un divisor resistivo con dos resistencias del mismo valor y tolerancia 1% que soldaremos en el lugar del antiguo potenciómetro. Montamos con cuidado otra vez los engranajes y cerramos el servo atornillando de nuevo. En nuestro caso las resistencias se soldaron por el lado de la tapa, por lo que tendremos cuidado de no apretar en exceso al cerrar para que las patas de las resistencias no hagan cortocircuito.

Para finalizar, realizaremos el montaje del robot. Ajustamos los servos al chasis principal atornillándolos como muestra la imagen. Le ponemos las juntas tóricas a las ruedas que harán de bandas de rodadura y se las acoplamos a los servos. En la imagen también se ve un portabaterías y una cavidad delantera para alojar un posible sensor que no se encuentran en nuestro diseño. Además acoplaremos la placa arduino Yún con unos separadores para alzarla sobre el chasis, en medio quedará un hueco donde se alojará el portabaterías comprado [29].

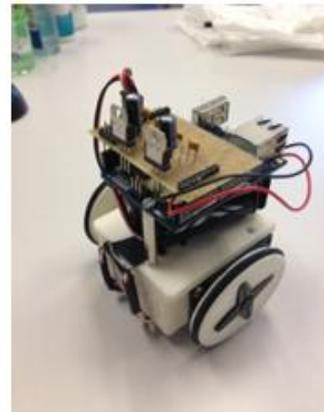
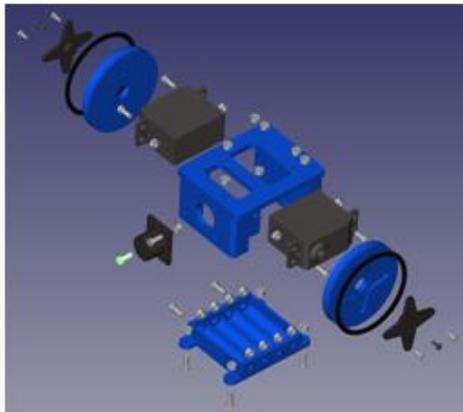


Figura 12: Plano de ensamblaje de piezas y robot montado.

3.3.3. Alimentación

Para que el robot se mueva es necesario que los servomotores sean alimentados al igual que la tarjeta Arduino, que recibirá los comandos y los gestionará produciendo el movimiento. Su alimentación y la de los motores, debe ser de 5V.

Para poder alimentar ambas partes hemos optado por utilizar 4 pilas AA Eneloop [29] ancladas en un portabaterías conectado con un clip a la placa auxiliar. El portabaterías tiene dimensiones 57 mm x 30 mm x 26 mm y lo fijaremos entre la tarjeta y el chasis con cinta aislante para poder cambiar las baterías con facilidad cuando se gasten.

3.3.4. Diseño de la placa auxiliar

Como debemos alimentar la tarjeta Arduino y los motores con 5V necesitaremos regular la tensión, puesto que la proporcionada por la pila será mayor. Para regularla en ambos casos utilizaremos reguladores 7805 a los que hay que incorporar condensadores de desacoplo. Necesitaremos diseñar una placa auxiliar que cumpla esta función.

Para el diseño de la placa auxiliar utilizaremos una matriz metalizada por una cara y troquelada con agujeros separados entre sí 2.54 mm para soldar los pines. La placa auxiliar irá acoplada mediante pines macho-macho a nuestra tarjeta; algunos para controlar los servos y otros simplemente de apoyo. El diseño del circuito implementado en la tarjeta está implementado en la siguiente figura.

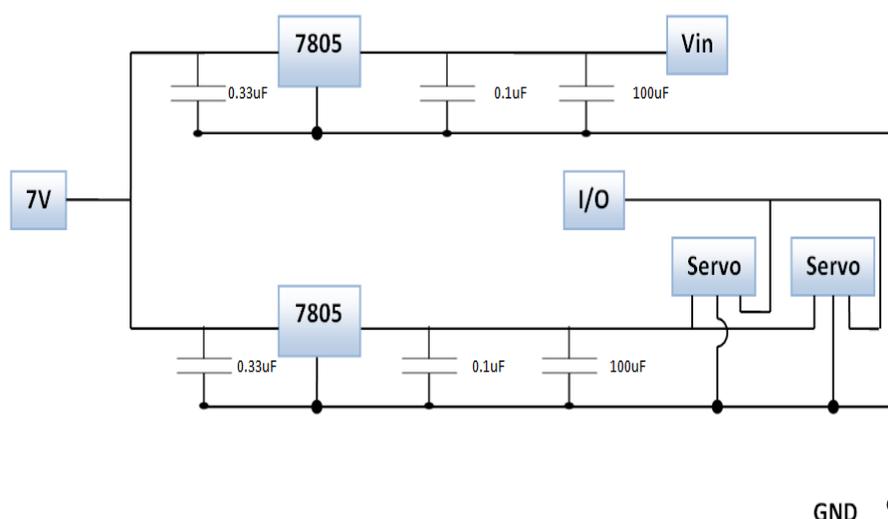


Figura 6: Diseño del circuito de la placa auxiliar.

También deberemos proporcionar una toma de tierra a los reguladores y a los servos, así como al polo negativo de la batería. Esta toma la conectaremos al pin GND de la tarjeta. Las salidas de los reguladores se conectarán a los polos de alimentación de los servos y al pin de alimentación 5V de la tarjeta respectivamente. Los pines destinados a los servos se configurarán a través de las salidas analógicas A0 y A1. El LED 13 rojo de la tarjeta lo usaremos como indicador de confirmación de comando recibido.

El aspecto de la placa auxiliar una vez soldados los componentes es el que se muestra en las imágenes destacando el lugar de las conexiones a la placa y los reguladores.

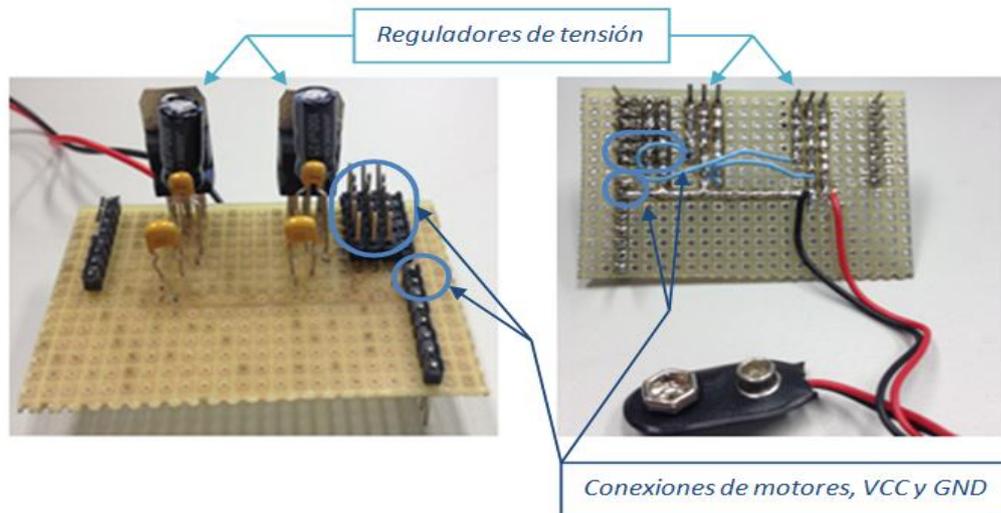


Figura 14: Vista de placa auxiliar superior (izq.) e inferior (der.).

Como apreciamos los reguladores se han conectado con la parte de disipación hacia afuera de modo que no calienten otros componentes del circuito y por si en un futuro fuera necesario acoplar algún pequeño disipador. En las primeras pruebas de locomoción del robot se ajustó una batería de 9V, pero nos dimos cuenta de que el regulador que alimentaba la Arduino se sobrecalentaba, debido a que el voltaje suministrado era excesivo, lo que nos llevó a usar el modelo actual de bloque de 4 pilas AA.

4. Desarrollo

4.1. Adquisición de la señal de EEG

Como ya hemos citado en la sección anterior el casco Emotiv EPOC es la herramienta que vamos a utilizar para la adquisición de datos de EEG en este proyecto. Cuenta con 14 electrodos de detección de señales EEG y dos referencias a través de los cuales se lee la actividad cerebral a una frecuencia de 128 Hz comunicándose con el dispositivo donde esté instalado el software de manera inalámbrica. El estado de calidad de señal de los sensores se muestra a través de la aplicación TestBench del SDK que además es la encargada de grabar los datos. Esta aplicación será fundamental a la hora de la adquisición de los datos, puesto que cuanto mejor sea la calidad de señal del electrodo mayor precisión obtendremos. Aunque el uso de este casco resulta una ventaja de cara a la poca preparación que requiere y la comodidad para el usuario, es fundamental empapar bien las almohadillas de los electrodos con el líquido iónico (líquido de lentillas normal) e incluso humedecerse bien el pelo con agua para mejorar considerablemente la calidad de señal.

Los tipos de señales que se necesitan detectar en esta interfaz son de tipo SSVEP (Steady State Visually Evoked Potentials). Este tipo de señales visuales se generan en la parte occipital del cerebro cuando el sujeto focaliza su atención en una de las luces parpadeantes de la pantalla, por lo que el electrodo en posición Oz sería el más adecuado para recoger este tipo de señales. Dado que nuestro casco no posee un electrodo en esta posición, se ha buscado para el análisis de señal la mejor combinación posible de entre los 14 canales disponibles. Ya que las señales se generan en la parte occipital del cerebro, los electrodos en las posiciones O1 y O2 serán los más importantes, y de obligado uso. Además se utilizarán 4 más de la parte frontal de la cabeza para atenuar el ruido procedente de estas zonas frontales que no nos interesan. Aunque con la combinación lineal de estos electrodos atenuemos un poco el ruido de cierta parte del cerebro, siguen existiendo artefactos, que no son más que eventos como parpadeos, guiños o pequeños movimientos involuntarios. Para realizar los registros de SSVEP en cada usuario con la mayor capacidad de discriminación, así como para monitorizar la actividad de fondo del cerebro se ha grabado una línea de *baseline* de 60s sin estimulación visual. Esta línea de base nos permitirá realizar posteriormente una mejor protección frente al ruido. Una vez se termina de grabar el *baseline* se procede al entrenamiento del sujeto con sesiones de estimulación largas de 20s con eventos parpadeantes a bajas frecuencias (entre 6Hz y 20Hz) de cara a

evaluar y seleccionar las mejores frecuencias para cada sujeto (se hablará de este tema en el punto 4.2).

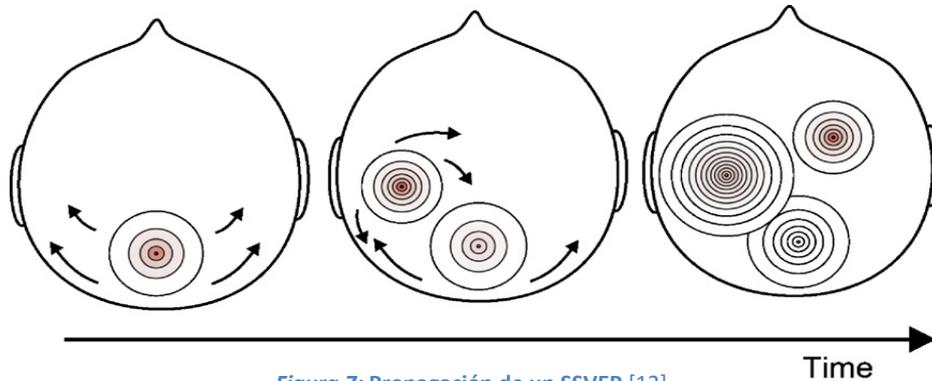


Figura 7: Propagación de un SSVEP [12].

La imagen representa la propagación de un SSVEP (observable con los electrodos de EEG) por la combinación de fuentes distribuidas y locales. Los círculos concéntricos de color rojo representan dipolos, y sus flechas su propagación. Inicio de la propagación (imagen izquierda); actividad de la propagación, al girar, activa fuentes secundarias (imagen central); El VEP alcanza su estado estacionario con una sucesión de amplios dipolos locales (imagen derecha). Estos dipolos dependen de las características del estímulo, lo cual explica las características de las señales observados en EEG [12].

Se ha hablado con anterioridad de que uno de los objetivos de este proyecto es la adaptabilidad al usuario; sin embargo en lo que se refiere al dispositivo de adquisición de señal se pretende que nuestro BCI funcione de la manera más universal posible para la adquisición de este tipo de señales. Se investigaron algunas combinaciones con otros electrodos temporales T7 y T8 (intentando realzar el protagonismo de hemisferio izquierdo y derecho respectivamente), pero los resultados no fueron muy esclarecedores, por lo que finalmente se descartó la lateralidad y se optó por una combinación de ambos hemisferios. La opción que mejores resultados nos proporcionó fue: (O1 + O2) – (F3 + F4).

Durante las pruebas realizadas en los sujetos se observó que a medida que se incrementaba la frecuencia de estimulación la importancia de los armónicos y subarmónicos crecía, incluso cuando la frecuencia fundamental resultaba casi indetectable. Es relevante por tanto la utilización de los armónicos en el análisis de señal y detección de SSVEP de manera que nos permitan ampliar el rango de frecuencias en las que detectar SSVEPs.

Según lo comentado podemos deducir que la relación señal a ruido no es todo lo buena que nos gustaría, lo que nos puede llevar a falsas detecciones de SSVEPs

(especialmente si se utilizan ventanas pequeñas). Intentaremos paliar estos efectos en el análisis de señal.

4.2. Procesamiento para detectar SSVEP.

Una vez que el casco adquiere los datos y los graba en los correspondientes archivos de texto *.txt*, es necesario realizar un análisis de señal de modo que se procese adecuadamente para la posterior extracción de características y detección de SSVEP. Para llevar a cabo esta tarea se ha hecho uso de las librerías y funciones de la API del SDK del Emotiv junto con otros archivos proporcionados por el Grupo de Neurocomputación Biológica. La mayoría del código se encuentra escrito en C / C++ y en algunos casos se utilizó MATLAB para el análisis de frecuencia offline para usar la Transformada de Fourier [25] durante las pruebas de detección de SSVEPs.

Aunque no vamos a entrar en detalle en describir el funcionamiento de todo el sistema interior haciendo un análisis exhaustivo del análisis de señal [23], sí que conviene tener una idea general de cómo se lleva a cabo el proceso para poder después extraer la información precisada.

Nuestro sistema debe funcionar sin una gran latencia a la vez que las detecciones de SSVEP han de ser precisas de modo que el sistema se equivoque lo menos posible y el comando recibido en el robot sea el adecuado (en el caso como es el de este proyecto de un robot básico un error en recepción no sería una catástrofe, pero en otros casos como el control de una silla de ruedas, de un robot que realice una tarea de precisión o uno enviado al espacio sí que sería crucial una óptima recepción). Para ello, va a ser indispensable realizar un análisis de señal online de forma rápida y evitando tamaños de ventanas muy grandes. Esto entrará en contradicción con el uso de ventanas no demasiado pequeñas para evitar que se cuele el posible ruido. Debemos encontrar pues una solución de enventanado de modo que exista un equilibrio entre estos dos puntos.

Finalmente se adoptó una solución con las siguientes condiciones:

- Se utilizarán ventanas de 2s de duración a 128 muestras por segundo, existiendo solape del 50% entre ventanas, es decir, tomando 1s de la ventana anterior.
- El enventanado será de tipo Hanning para darle importancia al lóbulo principal y no atenuar demasiado los armónicos.
- Tendremos en consideración los armónicos y subarmónicos para la puntuación final de cada frecuencia.

La primera parte del análisis consiste en eliminar las tendencias lineales existentes (*de-trending*) que se cuelan como ruido en bajas frecuencias. Estas tendencias lineales vienen generadas al enventanar señales discretas con componentes lineales y obtener su FFT (algoritmo para implementar la DFT), pues producen nuevas componentes frecuenciales que no se encontraban en la señal original.



Figura 8: Esquema del análisis de señal.

Una vez eliminadas estas tendencias haremos un enventanado de tipo Hanning, que es uno de los tipos más utilizados de ventanas para un análisis de estas características y nos permite obtener una buena resolución en frecuencia [25]. A continuación describiremos el proceso: Primeramente accedemos a los sensores del casco para recoger la información (esta información se envía cada segundo a 128 muestras por segundo). Al pasar el primer segundo se han recogido 128 muestras, pero la ventana no está completa, por lo que guardaremos las 128 muestras y esperaremos a que pasen 2s. A los dos segundos el casco vuelve a enviar información con otras 128 muestras, que concatenaremos (solape) con las guardadas en el segundo anterior para obtener nuestra ventana completa a analizar de 256 muestras. De esta forma se va procesando y enviando información cada segundo.

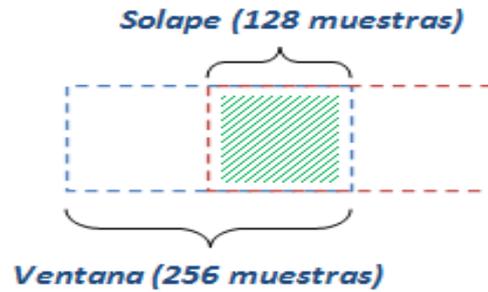


Figura 9: Solape en el enventanado de señal.

Posteriormente realizaremos la DFT para la obtención del espectro en frecuencia, implementada a partir de las librerías que realicen el algoritmo FFT.

Con el objetivo de comprobar las detecciones en señales registradas previamente, representaremos con MATLAB [32] los espectros en frecuencia de las ventanas a analizar y nos fijaremos a qué frecuencia se produce un pico de amplitud considerable respecto a los demás. Esta frecuencia debería coincidir con aquella a la que se esté estimulando el sujeto. El encontrar picos de mayor o menor amplitud para diferentes frecuencias estimuladas y para diferentes sujetos es totalmente lógico, debido a que el sistema está planteado desde un punto de vista de adaptabilidad al usuario y cada persona reacciona de manera diferente a cada estímulo, siendo más o menos sensibles a un estímulo en concreto. Además al mostrarse por pantalla estímulos simultáneos es posible que se generen interferencias entre ellos por lo que pondremos un especial cuidado en este análisis de señal considerando los armónicos y las derivas frecuenciales que se dan en las señales evocadas utilizando un rango de frecuencias de entre 5 Hz y 41 Hz, suficientemente grande como para abarcar el barrido de frecuencias de estimulación de los 6 a los 20Hz, sus armónicos y los adyacentes de todos ellos. Este punto será de vital importancia al realizar la selección de frecuencias (*punto 4.2.1 y 5.2.1*).

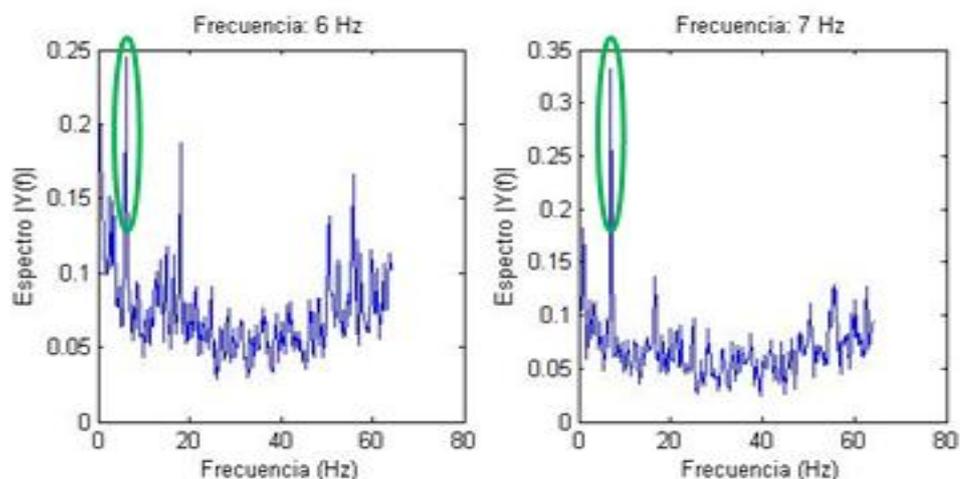


Figura 10: Detección de SSVEPs con estímulos parpadeantes a 6 y 7Hz.

Ya sabemos que nuestro sistema es capaz de detectar SSVEPs. Nuestra tarea ahora consiste en seleccionar las frecuencias que con mayor probabilidad producirán estímulos de este tipo en cada usuario. Este punto será de vital importancia a la hora de probar nuestro sistema, ya que este se debe adaptar a cada usuario de manera eficiente, para optimizar de la mejor manera el control del robot. Seleccionaremos un conjunto de frecuencias de estimulación válidas presentándolas una a una al sujeto, definidas por un umbral mínimo de relación señal ruido en los correspondientes SSVEPs. Finalmente, en este conjunto de frecuencias obtenidas, buscaremos las 4 frecuencias compatibles para el control del robot, mediante un protocolo de selección en ciclo-cerrado de su compatibilidad. Este protocolo de adaptación al usuario personalizado se propuso por primera vez en [24] y se adaptó para el casco de Emotiv en [23]).

Para obtener este conjunto de calcularemos una puntuación para cada frecuencia estimulada; teniendo en cuenta las derivas frecuenciales. La puntuación de cada frecuencia f vendrá dada según la expresión:

$$P_f = X(f) + X(2f) + \frac{1}{2} + [X(f - 1) + X(f + 1) + X(2f - 1) + X(2f + 1)]$$

Según la expresión, consideramos para el cálculo de la puntuación los valores de la transformada en la frecuencia fundamental $X(f)$ y en el primer armónico $X(2f)$, siendo f la frecuencia de parpadeo del estímulo. Para calcular los adyacentes hemos limitado su peso a la mitad en la puntuación final, de modo que se tienen en cuenta pero se penalizan ligeramente (*Proyecto Fin de Carrera EPS-UAM [23]*).

Obtenida la puntuación para todo el barrido de frecuencias estimuladas, se guardarán los datos en el archivo *analisisAlgoritmo_FASE1.txt*. Formaremos ahora el conjunto de estas frecuencias que con mayor probabilidad el usuario podrá detectar, a este conjunto de frecuencias las denominaremos “frecuencias válidas”. Los resultados de esta selección de frecuencias, se guardarán en el archivo *resultadosAlgoritmo.txt* ordenando estas frecuencias de mayor a menor según su puntuación. De estas “frecuencias válidas” el algoritmo desarrollado en (*Proyecto Fin de Carrera EPS-UAM [23]*), seleccionará 4 como las más compatibles para el control del robot. Este proceso de selección de frecuencias se describirá y desarrollará con detalle en la sección de Experimentos y resultados (5.2.1).

4.3.Transformación a un comando.

Una vez realizada la detección de un SSVEP, se ha de detectar el correspondiente comando y enviarlo al robot para su posterior gestión en la generación del movimiento.

Una vez que sabemos cómo el sistema detecta los SSVEPs y se ha realizado la selección de las 4 frecuencias que servirán de estímulos para el usuario a la hora de controlar el robot, pasamos a analizar cómo conseguimos transformar estas frecuencias que detectamos a un comando que enviar al robot.

Lo primero que hace nuestro programa es inicializar los datos referentes a las direcciones de control, esto es, asignar a cada una de las direcciones o grados de libertad las frecuencias seleccionadas por el sistema de entrenamiento. Estas frecuencias fueron previamente cargadas en un archivo de texto que usará nuestro programa para pasar por argumento las frecuencias seleccionadas, así como un identificador, que nos servirá para reconocer la cola por la que recibiremos los datos online del casco.

```
//Inicialización de las frecuencias a partir de los datos de entrada
int idCola = atoi(argv[1]);

int frecuenciaAvanzar = atoi(argv[2]);
int frecuenciaParar = atoi(argv[3]);
int frecuenciaIzquierda= atoi(argv[4]);
int frecuenciaDerecha = atoi(argv[5]);
```

Figura 11: Código de inicialización de frecuencias.

Además, inicializaremos a 0 una serie de contadores que nos permitirán gestionar las detecciones de eventos SSVEPs, así como una variable para guardar la última frecuencia detectada.

El primer paso del algoritmo es la recepción de los datos a través de la cola. Estos datos se encuentran dentro de una estructura que contiene: un campo que nos dice si nos llega un dato de información o de cierre del programa (fin de estimulación) y un campo en el que almacena la frecuencia de estimulación detectada.

Antes de la identificación del comando compararemos la variable *ultima_frecuencia* con la frecuencia actual detectada, si son distintas se inicializarán los

contadores a 0. La razón del uso de esta variable radica el hecho de querer evitar falsas detecciones, obligando a que existan dos detecciones seguidas de la misma frecuencia, si esto no ocurre, inicializaremos los contadores, de manera que la parte de identificación del comando no procese su envío.

Con este algoritmo nos aseguramos de que con una gran certeza el sujeto estará mirando a un estímulo determinado, evitando posibles interferencias de los estímulos o detecciones incorrectas. Además, con este algoritmo también se reduce en parte la desventaja de que se detecten estímulos incorrectos cuando el usuario no se encuentran mirando a los estímulos, pues se hace una gestión del control del robot limitando el envío de comandos en cada detección y optimizando su control. Además, se ha de tener en cuenta, como ya mencionamos en la sección anterior, que se procesarán datos cada segundo según el análisis de señal.

Si el dato que recibimos es de información, es decir, se detecta una frecuencia entonces pasaremos a identificar la frecuencia y transformarla en comando. Identificaremos de qué dirección se trata comparando la frecuencia detectada con la de inicialización. Una vez sabemos de qué frecuencia se trata incrementamos el contador asociado a esa frecuencia, si el contador no ha llegado a 2 entonces asignamos esta frecuencia a la variable *ultima_frecuencia* y volvemos a la cola para recibir el dato de la siguiente detección. Si el contador asociado a la frecuencia detectada ha llegado a 2 entonces enviamos el comando al robot siguiendo el siguiente proceso:

```

if (comunicacionRobot.ComandoRobot == frecuenciaAvanzar)
{
    cont_Avanzar++;
    if (cont_Avanzar >= 2)
    {
        printf("\nHas elegido la tecla de direccion 'arriba'\n\n");
        page = "/arduino/arriba/1";
        process_http(sockfd, page);
        cont_Avanzar = 0;
        cont_Parar = 0;
        cont_Izquierda = 0;
        cont_Derecha = 0;
    }

    ultima_frecuencia = frecuenciaAvanzar;
}

```

Figura 12: Código de transformación a un comando.

1. Mostramos un mensaje en pantalla con la dirección o comando asociada a la frecuencia detectada.
2. Guardamos en una variable `char*` la cadena de caracteres correspondiente al comando asociado (Ejemplo: `page = "/arduino/arriba/1";`) y que será parte de una URL. Según la frecuencia detectada variaremos la opción entre barras: *arriba, izquierda, derecha, parar*.
3. Llamamos a la función que gestiona el envío del comando por petición HTTP (Ejemplo: `process_http(sockfd, page);`).
4. Inicializamos todos los contadores a 0.
5. Actualizamos la variable `ultima_frecuencia` con la última frecuencia detectada.
6. Volvemos a la cola para recibir una nueva detección.

En caso de que el dato que nos llegue sea de fin de estimulación, se pondrá la variable `ultima_frecuencia` a cero, se mostrará un mensaje por pantalla y se terminará del programa.

4.4. Comunicación de la interfaz cerebro-máquina con el robot.

El objetivo principal de este proyecto es el del control de un robot a través de una interfaz-cerebro-máquina, pero para controlarlo será necesario enviarle los comandos adecuados, lo que supone que exista una comunicación entre la interfaz y el robot.

Para llevar a cabo esta comunicación se ha optado por utilizar el WiFi que nos proporciona la tarjeta Arduino Yún. Se ha descartado por tanto la utilización de puerto Serie que tienen las placas Arduino, al poseer la desventaja de tener que entrar al IDE de Arduino, cargar el programa y abrir dicho puerto cada vez que se quiera utilizar el robot. Ya comentamos en la parte de diseño la ventaja que supone el utilizar una comunicación vía WiFi en nuestro proyecto, por tanto será esta característica la que usemos en el tipo de comunicación que implementemos.

Nuestro robot se mueve al recibir comandos a través de la WiFi de Arduino. Estos comandos se van a enviar en peticiones HTTP a través de nuestra interfaz, pero también se pueden recibir a través de cualquier navegador instalado en cualquier sistema operativo. El uso y control de ciertos dispositivos a través de navegadores de internet es cada vez más frecuente, y es que hoy en día nos resulta intuitivo manejarlos y estamos totalmente acostumbrados al conectarnos diariamente a internet a través de ellos. En nuestro caso basta con estar conectado a la red WiFi que genera la placa, abrir cualquier navegador en cualquier tipo de dispositivo e introducir una URL del tipo <http://robotico.local/arduino/arriba/1>, donde “robotico” es el nombre de la placa, “.local/arduino” es parte de la configuración del sistema cliente-servidor de Arduino, “arriba” es el comando a enviar y “1” implica encender el LED 13. A continuación, pasaremos a detallar dicha comunicación.

Cada vez que la interfaz quiere enviar un comando, se llama a la función que envía la petición HTTP añadiendo una URL como dato a enviar, como esto se tiene que hacer de manera automática sin utilizar navegadores se utilizan sockets, especialmente en este caso el comando **send** dentro de la función que gestiona el envío.

```
int process_http(int sockfd, char *page)
{
    char sendline[MAXLINE + 1];

    snprintf(sendline, MAXSUB,
             "POST %s HTTP/1.0\r\n"
             "Host: robotico.local\r\n"
             "Content-length: %d\r\n\r\n", page, 30);
    send(sockfd, sendline, strlen(sendline), 0);
}
```

Figura 13: Función de envío del comando.

De esta forma se forma la cadena: <http://robotico.local/x>, donde x será la subcadena que se encuentra dentro de ***page**. Una vez lista se envía a través del comando **send** dentro de **sendline** y se recibe en el servidor web de nuestro programa Arduino.

Dentro de nuestro programa principal (cargado en la placa) en Arduino que tendrá un nombre de *Servidor.ino*, declararemos aparte de las dos librerías principales (*Arduino.h* y *Bridge.h*) dos más que nos permiten gestionar el Servidor y aceptar las

peticiones (*YunServer.h* y *YunClient.h*). Lo primero que debemos hacer para poder comunicarnos con el servidor es instanciar un servidor de modo que habilitemos a la tarjeta Yun para que pueda escuchar a los clientes conectados (*YunServer server*).

En la parte de configuración se inicia la comunicación de puerto serie y se configura el LED 13 como salida y se pone a rojo mientras se inicia el Bridge bloqueando el proceso durante 2s. Cuando termina se apaga el LED. En la segunda parte de la configuración se le dice a la instancia creada que permanezca a la escucha sólo para conexiones procedentes del localhost. Las conexiones hechas al Linino pasarán al procesador 32U4 para poder controlar los pines. Después se inicia el servidor.

1ª parte del setup

```
void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);
  Bridge.begin();
  digitalWrite(13, LOW);
}
```

2ª parte del setup

```
server.listenOnLocalhost();
server.begin();
}
```

Figura 14: Setup del programa principal del robot.

Una vez configura nuestro servidor en el proceso *void loop()* que no es nada más que un bucle, instanciamos un cliente en el bucle para aceptar cada petición que llegue. Comprobamos que es de tipo *Client* y si es así llamamos a la función *process* que recibe como argumento la instancia cliente, donde lo primero que se hará será extraer de la cadena en un String el comando para gestionarlo y generar el movimiento del robot.

4.5. Movimiento el robot.

Ya disponemos de una comunicación WiFi para poder recibir los comandos enviados desde la interfaz. Ahora explicaremos cómo se gestiona el comando recibido en el programa para conseguir mover el robot en las diferentes direcciones.

Lo primero que debemos agregar a nuestro programa principal son las librerías a utilizar. En este caso aparte de las librerías ya citadas para la gestión del servidor y los clientes, agregaremos *Robot.h* y *Servo.h* que contendrán la definición de funciones para el manejo del robot y los servos. El fichero *Servo.h* se encuentra ya en el software de Arduino, sin embargo *Robot.h* así como el archivo *Robot.cpp* han tenido que ser creadas para conseguir un código limpio y bien estructurado.

Con las librerías agregadas, instanciamos dos variables tipo *Servo* (*ServoD* y *ServoI*) que nos servirán para controlar el servomotor derecho e izquierdo respectivamente. También crearemos un objeto tipo *Robot* que tendrá como argumentos las variables servos instanciadas para poder controlarlos.

Como ya dijimos una vez que se ha aceptado la petición del cliente en el servidor, se llama a la función *void process (YunClient client)*. En la función se extrae el comando por medio de:

```
String command = client.readStringUntil('/');
```

Figura 15: Extracción del comando.

Posteriormente se identifica el comando según sea “arriba”, “parar”, “izquierda” o “derecha” por medio de sencillos comandos de control *if*. Cuando se detecta cualquiera de los comandos encendemos el LED13 rojo durante 1s y lo apagamos introduciendo otro *delay* de 1s antes de realizar el movimiento seleccionado. Para decirle al robot que se mueva en la dirección adecuada llamaremos al método adecuado según la instrucción recibida mandando como argumentos los dos servos a controlar. Tras la llamada al método dejaremos un tiempo de 2s *delay* (2000); (el tiempo viene dado en ms) durante los que el robot se moverá en la dirección seleccionada. Tras estos dos segundos el robot se parará con una llamada al método *Parar* y se volverá al bucle *void loop()* para aceptar una nueva petición.

```
if (command == "arriba")
{
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);

    Robot.Adelante(ServoD, ServoI);
    delay (3000);

    Robot.Parar(ServoD, ServoI);
}
}
```

Figura 16: Procesado del comando.

Mientras el programa se encuentre gestionando un comando este no podrá procesar o aceptar peticiones hasta que no vuelva al bucle **loop**. Esto puede ser motivo de un procesamiento lento y pérdida de peticiones, incluso de buenas detecciones. Se intentará mejorar el sistema con el algoritmo adecuado de procesamiento de comandos, eliminando y ajustando los retardos del programa Arduino.

Los métodos de la clase Robot están contenidos en *Robot.cpp* y definidos en el fichero *Robot.h*. Si bien todos ellos son casi idénticos variando ligeramente algunas variables definiremos los tres más representativos:

- **Setup:** En este método se configuran los servos asociando cada uno a un pin determinado de la tarjeta por medio del método **attach**, de esta manera el servo quedaría “conectado” al pin.
- **Adelante:** En este método primeramente se hace una llamada a **attach** para “conectar” los servos a los pines y después cada servo llama al método **write** para mover el motor. Dependiendo del número que se mande como argumento el motor se moverá más o menos deprisa y girará en un sentido u otro. El número que se manda correspondería con el ángulo de giro, por lo que si con un **0** un motor gira en un sentido con **180** giraría en el sentido opuesto y con **90** debería estar parado, pero esto último no es tan trivial, pues no concuerda exactamente la parada del motor con ese ángulo y habrá que buscarlo a base de pruebas. Para este caso el servo derecho se le manda un argumento de **0** y para el servo izquierdo un argumento de **180**. Puede resultar extraño que para que el robot deba moverse hacia adelante y tener que girar los dos motores en el mismo sentido se envíen argumentos con ángulos opuestos; pero recordemos, que los servos están orientados en sentidos opuestos.
- **Parar:** En este método lo lógico sería buscar el ángulo de giro en que cada servo se queda parado para mandarlo como argumento. Sin embargo se optó por utilizar el método **detach** y desconectar cada servo del pin, puesto que era el único modo de parar el robot completamente.

Para llegar a este código final se realizaron pruebas básicas de movimiento de los servos y para encontrar los ángulos de giro adecuados. Se buscó también el ángulo

de giro donde el servo se encuentra parado, pero tras unas pequeñas pruebas observamos que aunque el robot realmente seguía funcionando como si estuviera en ralentí, lo que no nos proporcionaba ningún tipo de estabilidad, por lo que finalmente se decidió usar el método **detach** para parar el robot.

Finalmente, al probar el sistema global conjuntamente, apreciamos que el robot procesaba demasiado lentamente las peticiones perdiendo la mayoría de ellas y produciendo movimientos erróneos. Tras repasar el código Arduino de la tarjeta nos dimos cuenta de que era debido a que teníamos algunos *delays* repetidos y demasiado altos, se eliminaron los sobrantes y se ajustaron los tiempos afinando el sistema de procesamiento de datos en el robot lo máximo posible.

4.6. Pruebas intermedias básicas en el desarrollo.

En esta sección de desarrollo se han comentado y explicado cómo se han llevado a cabo los diferentes apartados del proyecto desde el punto de vista del funcionamiento del sistema global. No se han tenido en cuenta sin embargo los procesos y pruebas intermedias que en ocasiones están incluso considerados como hitos o pasos importantes dentro del proyecto pues llevan a la consecución de una parte importante de los objetivos de este trabajo. Comentaremos en este apartado algunos de estos procesos.

Antes de desarrollar el código comentado en el punto 4.3, se hicieron pruebas con pequeños hitos consistentes en el reconocimiento de ciertas teclas del teclado mostrando por pantalla un mensaje mostrando la dirección asociada (ARRIBA, ABAJO, IZQUIERDA Y DERECHA).

Primeramente se investigó acerca del control mediante las flechas de dirección del teclado con las que queríamos controlar el robot como un paso intermedio antes de la integración con la interfaz. Hicimos un programa básico que tiene en cuenta que estas teclas de dirección no poseen un código ASCII asociado directamente imprimible, sino que están formadas por un comando de escape (**ESC**) + el carácter '[' + una de las letras mayúsculas '**A**', '**B**', '**C**' o '**D**' según la tecla de dirección sea **arriba**, **abajo**, **izquierda** o **derecha** respectivamente. El programa reconocía las teclas asociadas a los caracteres A, B, C Y D. En un principio se hizo gestionando el código ASCII asociado a

cada letra y después se vio que utilizando el comando **getchar ()** se podía reconocer el carácter por lo que se utilizó esta opción. Conociendo entonces las secuencias de escape de las teclas de dirección se consiguió construir un programa más sólido para el reconocimiento de las teclas que después se integraría con la parte de comunicación de la interfaz para el control del robot por teclado.

A continuación se observa parte del código del reconocimiento de las teclas de dirección siendo **ch==27** el carácter de escape (**ESC**):

Key	Código de rastreo	"ASCII"	Secuencia de Escape
Home	71	0x101	ESC[H
Cursor Arriba	72	0x103	ESC[A
Pg Up	73	0x107	ESC[V
-	74	0x10A	ESC[S
Cursor Izquierda	75	0x105	ESC[D
5	76	0x109	ESC[G
Cursor Derecha	77	0x106	ESC[C
+	78	0x10B	ESC[T
End	79	0x102	ESC[Y
Cursor Abajo	80	0x104	ESC[B
Pg Dn	81	0x108	ESC[U
Ins	82	0x10C	ESC[@

```
do{
  ch = getchar();
}while(ch==10);
if(ch==27){
  getchar();
  ch = getchar();
  switch(ch){
    case 'A':
      ch=21;
      break;
    case 'B':
      ch=22;
      break;
    case 'C':
      ch=23;
      break;
    case 'D':
      ch=24;
      break;
  }
}
```

Figura 17: Código de prueba de control por teclado y algunas secuencias de escape.

Una de las partes más complicadas del proyecto fue la de establecer un sistema de comunicación entre el PC y el robot. Aunque ya se había tomado la decisión de hacerlo creando un servidor web en la tarjeta y enviar los comandos vía peticiones HTTP en formato URL, había que conseguir enviar la URL sin la utilización de un navegador, es decir, desde nuestro programa. El comando POST permite realizar esta tarea y para la conexión es indispensable el uso de sockets. Se elaboró un programa que manda una petición, confirmando su funcionamiento encendiendo el LED13 en el robot.

Conseguido esto, se integró el reconocimiento de las teclas de dirección con la parte de comunicación WiFi, formando un primer sistema de control inalámbrico del robot por teclado que después se integraría con la parte de procesamiento de la señal y de la generación de estímulos en pantalla, dando lugar a la interfaz cerebro-máquina actual.

5. Experimentos y resultados.

En este capítulo vamos a comprobar el funcionamiento de la interfaz cerebro máquina basada en SSVEP, adaptada a cada usuario, así como su validez para el control inalámbrico adecuado del robot. Para ello, se realizarán pruebas a un grupo de 10 sujetos dentro del laboratorio del Grupo de Neurocomputación Biológica con un equipo que cuenta con el software adecuado para realizarlas, en este caso con el SDK Enviroment para la adquisición y tratamiento de los datos con el casco Emotiv EPOC, así como el IDE Arduino para la comunicación WiFi con la tarjeta que controla el robot. Las condiciones ambientales se ha procurado que se mantengan constantes e invariables para cada usuario. A continuación se describen las características del PC con el que se han llevado a cabo las pruebas:

- Procesador Intel Core Duo i7
- CPU 950 de 3.07 GHz
- 4 núcleos reales e hyperthreading
- 8GB de RAM
- Pantalla Ancor Communications 27” con 1920x1080 de resolución y 60Hz de refresco de pantalla
- Sistema Operativo Ubuntu 12.04 LTS de 32 bits
- Tarjeta Gráfica NVIDIA GeForce GT 220

El sistema de pruebas utilizado se ha adaptado para controlar el robot. La fiabilidad de la interfaz de estimulación por pantalla y el análisis de señal para detectar SSVEP ya fue evaluada con otra serie de pruebas, al formar parte del proyecto de fin de carrera de Álvaro Morán García [23]. Esta parte se ha integrado en este proyecto para realizar las pruebas de control del robot.

5.1. Validación del control del robot.

Partiendo del sistema de pruebas relacionado con el barrido y selección de frecuencias de estimulación ya validados en proyectos anteriores [23], el paso siguiente será validar si se puede controlar el robot a través de este tipo de estimulación exógena (SSVEP). Hay que destacar que la adaptación al sujeto consiste en dos fases previas a las pruebas de validación del control: la primera, un barrido en frecuencias, donde se seleccionan las frecuencias válidas a priori de forma individual para cada usuario y la segunda la selección de las cuatro frecuencias más compatibles entre las frecuencias válidas individualmente para su uso en el control del robot.

Para poder describir las pruebas es recomendable recordar e identificar las principales etapas de un BCI con este tipo de aplicación para este proyecto. La primera etapa es la de adquisición de la señal utilizando la estimulación por pantalla y el casco Emotiv para registrar la actividad cerebral que produce la respuesta al estímulo. Después, para validar que se detecta adecuadamente el SSVEP observaremos en el análisis de FFT el valor de pico de mayor amplitud. Una vez hecho el barrido de frecuencias detectando los SSVEPs y determinando las frecuencias de estimulación válidas para cada sujeto, se seleccionan las 4 más compatibles para el control del robot. A continuación se procede a evaluar la tarea del control de robot en la que el usuario concentra su atención visual en el estímulo adecuado para mover el robot. En este proceso se detecta cuando hay un SSVEP a cierta frecuencia y se transformará esta frecuencia en el comando correspondiente mediante un código de identificación. Una vez que tenemos identificado el comando, se envía al robot de forma inalámbrica a través de la WiFi creada por la tarjeta Arduino Yún, donde por último se gestionará el comando para realizar el movimiento deseado.

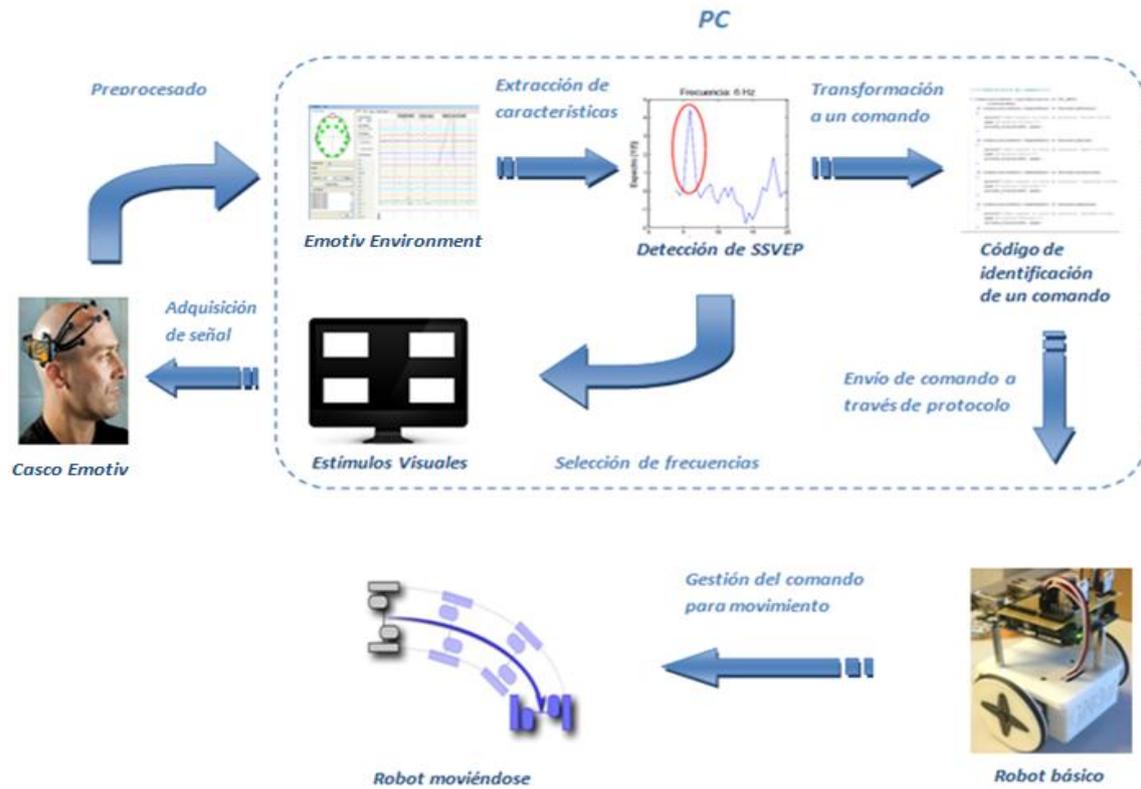


Figura 18: Esquema de validación del control del robot.

Las pruebas realizadas se han llevado a cabo dentro del laboratorio del Grupo de Neurocomputación Biológica a un número total de 10 sujetos voluntarios intentando mantener unas condiciones estables y similares de iluminación en todos los sujetos de manera que estén no molesten e incidan para la prueba y el sujeto se encuentre lo más relajado posible. La única restricción para la realización de la prueba era la de no poseer problemas graves en la vista o de epilepsia, ya que la exposición a la luz parpadeante podría provocarlos en personas que padecen esta enfermedad.

La realización total del experimento consiste en una primera parte donde se somete al sujeto a la estimulación visual con el fin de seleccionar las 4 mejores frecuencias de control (adaptación personalizada de la interfaz a la particularidad de cada sujeto), y de una parte de validación o pruebas de control donde el sujeto deberá hacerle llegar al robot una secuencia de comandos específicos (8 o 16 comandos según el caso).

Se ha estimado un tiempo total para la realización del experimento completo, incluyendo los tres circuitos, de modo que no sobrepase la hora, ya que llega a ser cansado e incluso molesto estar prestando atención a luces parpadeantes, sobre todo durante la primera parte de la prueba, la que se refiere a la selección de frecuencias para el sujeto. Para algunos sujetos este proceso es más largo que para otros, debido

al número diferente de frecuencias válidas para cada usuario. A mayor número de frecuencias más largo es el proceso. Además se han establecido períodos de descanso adaptados a la necesidad del sujeto entre los diferentes experimentos.

Los resultados obtenidos diferirán entre sujetos, al responder de forma diferente a la estimulación con luz y registrarse una respuesta diferente en el EEG. Al realizárselos experimentos en un número reducido de sujetos los datos obtenidos pueden no resultar determinantes a la hora de tomar la decisión de implantar cierto sistema para un uso extendido.

5.2. Pruebas y resultados

Se realizarán una serie de experimentos con la interfaz cerebro-máquina creada para el control del robot de cara a evaluar diferentes aspectos:

- Personalización de la interfaz mediante la elección de frecuencias para el control del robot (4 grados de libertad).
- Eficiencia del sistema de control del robot (a través de la evaluación del control del robot en tres circuitos).

5.2.1. Parte 1: Búsqueda de las mejores frecuencias de estimulación en cada sujeto para el control del robot.

Una vez ajustado el casco Emotiv adecuadamente y después de comprobar que todos los sensores se encuentran en un nivel aceptable de funcionamiento, pasamos a la fase de selección de frecuencias de estimulación, para lograr el posterior control del robot.

5.2.1.1. Descripción de la prueba.

Dentro de esta primera parte lo primero será la obtención del baseline que no es más que una grabación de la actividad de fondo del cerebro de cada usuario para que sirva como referencia para la detección de SSVEPs. Para obtener este baseline el usuario deberá observar durante 60 segundos la pantalla del monitor en negro, es decir, sin estimulación visual. Esta grabación se guardará en un archivo de texto baseline.txt

Se pedirá a los sujetos que eviten realizar movimientos musculares faciales como guiñar los ojos, hablar o rechinar los dientes, hacer fuerza con alguna parte del cuerpo,... En conclusión se le pedirá al sujeto que esté lo más relajado posible evitando sobre todo cualquier gesticulación facial y se centró únicamente en la tarea asignada. Aunque no es un requisito muy estricto para el baseline, ya que el tiempo es bastante grande 60s, se le pedirá al sujeto que evite parpadear en la medida de lo posible.

Terminada la grabación del baseline se pasa a la búsqueda de las 4 mejores frecuencias para el usuario que serán las que se utilicen como estímulos para el control del robot. Para esta búsqueda de frecuencias se utilizará un algoritmo de selección de frecuencias basado en el protocolo de optimización ACL (Assisted Closed Loop) desarrollado por el Grupo de Neurocomputación Biológica [24], con una serie de cambios para su utilización con el Emotive Epoc [23], [33].

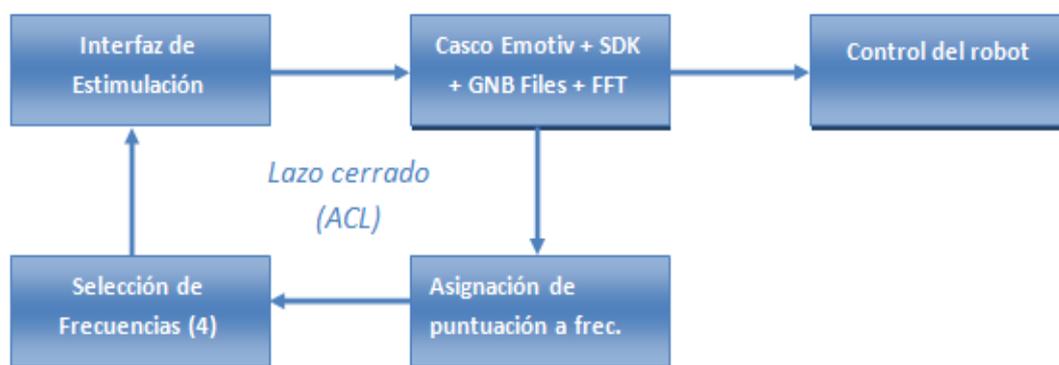


Figura 19: Esquema de la selección de frecuencias básica para adaptabilidad al usuario.

Esta selección de frecuencias consta de dos fases:

- ❖ **Fase 1:** En esta fase se determinan las frecuencias válidas de partida para cada usuario. Para esta tarea se realiza un barrido de frecuencias desde 6Hz hasta 20Hz con saltos de 1Hz, mostrando un único cuadro parpadeante blanco durante 10s a la frecuencia indicada. Pasados esos 10s el usuario dispondrá de

2s de descanso en los que podrá parpadear y gesticular antes de que se muestre un nuevo estímulo a 1Hz mayor. Una frecuencia de estimulación se considera válida si la señal de SSVEP registrada supera un umbral de relación señal/ruido. Este umbral se ha elegido basándonos en la experimentación y en análisis de las pruebas, y en nuestro caso está fijado en 7. A cada frecuencia se le asignará una puntuación obtenida a partir de la transformada de Fourier según la fórmula expuesta en la sección 4.2.1 (Selección de frecuencias). Con estas puntuaciones y el umbral se seleccionarán el grupo de frecuencias válidas. Adicionalmente se ordenarán estas frecuencias válidas de mayor a menor puntuación.

- ❖ **Fase 2:** En esta fase se evalúa la compatibilidad entre frecuencias mostradas simultáneamente para evocar un único SSVEP claro cuando el sujeto concentra su atención visual en un único icono parpadeante. En esta fase se determinarán las 4 mejores frecuencias de estimulación para cada usuario del grupo de frecuencias válidas obtenidas en la primera fase.

5.2.1.2. Análisis de resultados.

En una primera tabla se muestra el barrido en frecuencia realizado para cada sujeto de 6 a 20 Hz con los valores de detección o puntuaciones asignadas a cada frecuencia.

#Us	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	421.3	47.5	11.6	7.8	11.7	10.2	7.3	7.7	6.1	4.9	5.03	5.04	4.1	5.2	5.4
2	440.7	59.4	15.5	11.5	22.4	9.3	6.1	6.8	5.7	5.7	7.1	7	12.6	13.3	34.9
3	427.7	57	17.1	11.6	20.9	9.4	8.7	10.3	8.2	11	8.1	9.7	6.6	8.8	7.4
4	426.2	47.4	12	6.5	13.1	6.7	5.9	6	4	5.3	5.5	6.7	5.2	7.8	5.3
5	441.8	48.1	9.9	5.9	5.2	2.4	7.4	7.6	10.6	16.8	10.2	51.6	16.7	12.9	13.3
6	425.6	49	16.3	11.9	21.8	11.5	7.9	10.8	6.1	5.9	8.6	43.7	9.9	6.9	4.9
7	442	47.4	9.2	6.2	6.9	6.4	3.6	3.7	6.6	8.6	11.9	7.5	7.6	17.6	55.3
8	433.7	53.3	15.8	7.2	6.1	5.2	4.5	6.9	4.8	3.6	4.4	3.8	4	5.8	7.2
9	435.6	55.6	17.5	14	20.5	27.7	33.4	26.4	33	24.6	27.6	34.4	34.9	29.9	23.5
10	498.6	55.4	15.7	6.4	19.9	7.8	6.21	7.6	7.1	5.5	5.3	7.3	5.9	8.8	13.2

Tabla 6: Puntuaciones de todas las frecuencias estimuladas tras el barrido (6-20Hz).

Estas puntuaciones se obtienen a partir de la transformada de Fourier en la frecuencia fundamental y en el primer armónico. Para el cálculo de los adyacentes

estamos limitando su peso en la puntuación final a la mitad, de forma que se tienen en cuenta pero están ligeramente penalizados. El objetivo de esta puntuación es lograr la mejor compatibilidad de frecuencias de estimulación entre sí. Este cálculo de puntuaciones viene dado por la fórmula descrita en el apartado 4.2 (*Procesamiento para detectar un SSVEP*):

Para cada uno de los sujetos y de cara a evaluar cómo de adecuadas son las frecuencias asignadas por el programa de entrenamiento se ha confeccionado la siguiente tabla que refleja el número de frecuencias válidas que tiene cada sujeto (esto es, aquellas que superan el umbral de puntuación escogido en 7) y las frecuencias válidas para cada sujeto ordenadas por orden de puntuación. Se encuentran resaltadas en azul las cuatro mejores frecuencias para cada usuario.

#Us	#f.det	Frecuencias detectadas de mejor a peor
1	7	6, 7, 10, 8, 11, 9 y 13
2	8	6, 7, 20, 8, 19, 18, 11 y 17
3	10	6, 7, 10, 8, 9, 15, 13, 17, 11 y 19
4	5	6, 7, 10, 8 y 19
5	9	6, 17, 7, 15, 18, 20, 19, 16 y 13
6	8	6, 7, 17, 10, 8, 9, 11 y 13
7	8	6, 20, 7, 19, 16, 15, 18 y 17
8	5	6, 7, 8, 9 y 20
9	10	6, 7, 18, 17, 19, 11, 16, 13, 15 y 20
10	8	6, 7, 10, 8, 19, 11, 13 y 17

Tabla 7: Resultados tras la fase 1 de selección de frecuencias.

Tras la fase 2 de análisis de compatibilidad entre frecuencias durante su presentación simultánea en pantalla, el programa asigna las cuatro mejores frecuencias a cada una de las direcciones de control.

#Us	↑	STOP	←	→
1	10	6	11	7
2	20	7	18	8
3	10	7	17	9
4	19	7	10	6
5	15	6	20	17
6	10	6	11	9
7	7	20	15	16
8	20	6	8	7
9	13	16	18	7
10	10	7	19	6

Tabla 8: Asignación de frecuencias de estimulación a comandos tras la fase 2.

Podemos observar a la vista de los resultados de la tabla 8 como el programa asigna las 4 frecuencias seleccionadas en cada sujeto a cada dirección, su asignación se

establece siguiendo el orden: adelante, parar, izquierda y derecha. Estas frecuencias son asignadas según el criterio de máxima compatibilidad de la fase 2 en el que se mide la compatibilidad entre conjuntos de dos y cuatro frecuencias hasta encontrar el grupo de las 4 más compatibles. En algunos sujetos en los que se observó que un comando se detectaba peor que otro se modificó la asignación de frecuencias a direcciones manualmente, dando prioridad a los comandos más utilizados en los circuitos, de modo que una frecuencia que se detectará mejor estuviera asociada al comando más utilizado.

En las tablas anteriores se puede analizar qué frecuencias estimulan mejor a los sujetos en esta interfaz. A la vista de los resultados representados en la primera tabla podemos ver cómo las bajas frecuencias son las más seleccionadas durante el barrido inicial (las marcadas en azul son las cuatro mejores). Como vemos la frecuencia más utilizada entre todos los sujetos es la de 6Hz, la más baja en el barrido, la segunda mejor casi siempre es la de 7Hz y aunque la tercera y la cuarta mejor ya empiezan a variar según el sujeto cabe destacar la frecuencia de 8Hz o 10Hz y los saltos a frecuencias mayores de final de barrido como 20Hz. La segunda adaptación de compatibilidad casi siempre resulta en la inclusión de alguna frecuencia válida distinta a las cuatro mejores del barrido. Hay que destacar que estos resultados pueden ser muy dependientes del casco/amplificador de EEG que se utilice y la elección de la localización de los electrodos. En cualquier caso el proceso de selección de frecuencias resulta en un conjunto de cuatro frecuencias compatibles para cada sujeto.

5.2.2. Parte 2: Estimulación del sujeto y eficiencia del sistema para el control del robot.

En esta segunda parte se estimulará al sujeto a través de la interfaz visual en la que se mostrarán los 4 estímulos a las frecuencias óptimas elegidas según el protocolo descrito en la sección anterior. Cada uno de estas frecuencias hace referencia a un tipo de comando para el control del robot: adelante, parar, izquierda y derecha.

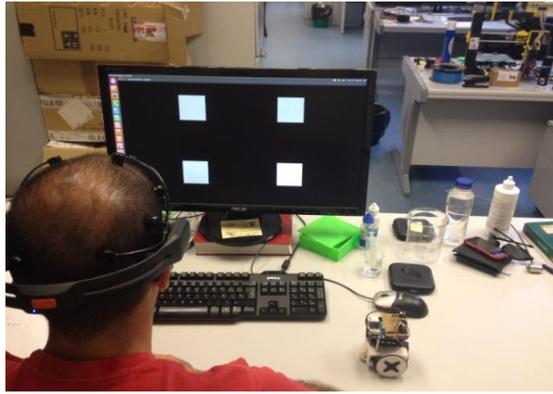


Figura 20: Estimulación del sujeto para el control del robot.

5.2.2.1. Descripción de la prueba.

En primer lugar, le pediremos al usuario que centre su atención en un estímulo durante el tiempo necesario para cerciorarnos de que el robot recibe el comando asociado. Este paso lo efectuaremos igualmente con los otros tres estímulos y lo repetiremos para todos dos o tres veces. Esto permitirá al usuario tener una primera toma de contacto con el control del robot; así como nos ayudará a evaluar cómo de bien el robot recibe cada comando asociado (acierta, falla, se retrasa,...) y tener un primer conocimiento acerca del control usuario-robot, o lo que es lo mismo evaluar a priori el funcionamiento del sistema o interfaz hombre-máquina-robot.

Una vez el usuario se ha relacionado con la interfaz de estímulos así como con el control del robot, pasamos a realizar las pruebas o experimentos, a partir de los cuales obtendremos unos resultados que nos permitirán evaluar el sistema así como su fiabilidad. Los experimentos desarrollados se basan en el envío de una cadena de comandos al robot implementada a través de 3 circuitos; en cada circuito se evalúa una cadena de comandos diferente. Para la realización de los experimentos se han tomado los siguientes datos observables y medibles:

- *Comandos totales realizados por el robot*
- *Errores cometidos por el robot cuando el comando realizado es incorrecto.*
- *Tiempo de recorrido del circuito.*

Los dos primeros circuitos son similares, su diferencia básicamente radica en la permutación de los comandos izquierda-derecha. El objetivo de estos dos

experimentos es el de observar si existen diferencias por la diferente secuencialidad de los comandos.

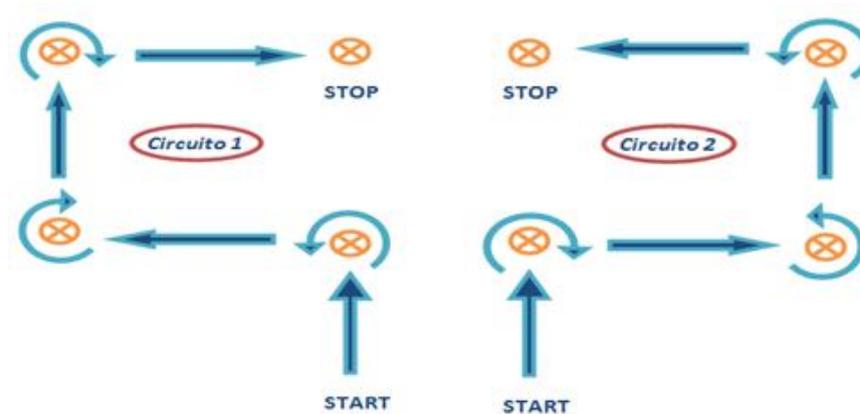


Figura 21: Circuitos 1 y 2 para la realización de los experimentos.

Para desarrollar este tercer experimento se han unido las cadenas de comandos anteriores (cambiando el comando parar intermedio por un giro) resultando un tercer circuito el doble de largo que los dos anteriores.

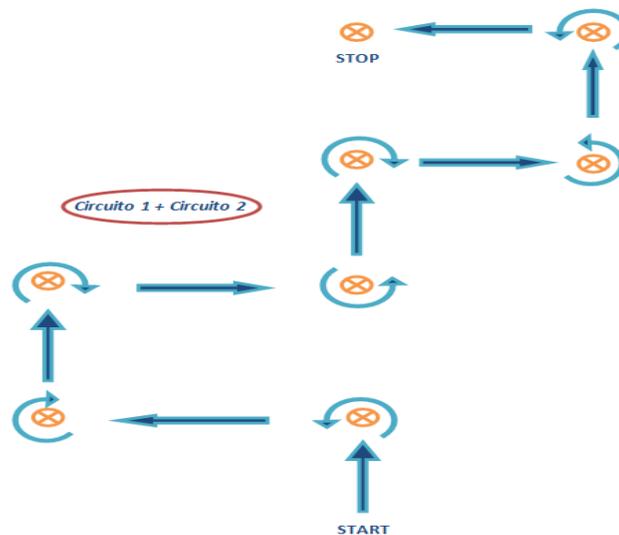


Figura 22: Circuito 3 para la realización de los experimentos.

5.2.2.2. Análisis de resultados.

Para evaluar la eficiencia del control del robot se han realizado pruebas consistentes en hacerle llegar al robot una serie de comandos seguidos contabilizando el tiempo y los errores al realizar la tarea. Para esta tarea cada sujeto iba focalizando su atención en un estímulo de los 4 presentados en pantalla según la dirección deseada. Esta secuencia de comandos se le proporcionaba al sujeto previamente al

recorrido, aun así se le iba recordando en voz alta al sujeto cuál era la dirección o comando que debía hacer llegar al robot, por lo que el sujeto sólo debía relacionar qué estímulo presentado en pantalla (superior izquierda, superior derecha, inferior izquierda o inferior derecha) era el que correspondía a la dirección (correspondencia que también se le proporcionaba a priori). Para mejorar esta parte en el futuro se podría modificar la interfaz gráfica de cara a que en vez de cuadros fueran flechas y estuvieran colocadas en pantalla según la dirección.

Para evaluar los resultados se han tomado datos durante la realización de esta secuencia de comandos en cada uno de los tres circuitos. Se le ha asignado una secuencia de 8 comandos al primer y segundo circuito y una secuencia de 16 comandos, el doble, al tercer circuito unión del primero y segundo. Estos datos son:

- Número de errores cometidos: Es el número de errores que se comete durante el recorrido del circuito. Tomando como error cuando se está mirando a un estímulo pero el comando que recibe el robot es distinto al esperado. No es considerado error si se ha producido ya uno y en los sucesivos el comando recibido sigue siendo erróneo, en este caso, los comandos recibidos serán tomados como comandos extras para la realización del circuito.
- Número de comandos totales: Es el número de comandos que el robot ha recibido. Este número de comandos será la suma del número de comandos del circuito más el número de comandos extras realizados durante su recorrido.
- Tiempo: Es el tiempo que tarda el robot en realizar la cadena de comandos asignada para cada circuito para el sujeto dado.
- Success - Rate o Tasa de aciertos: Es la capacidad de acierto del sistema o capacidad para resolver el circuito sin fallos. Esta tasa de aciertos se ha calculado a partir de:

$$\text{Acierto (\%)} = \left[1 - \frac{\text{nº de errores}}{\text{nº de comandos del circuito}} \right] \times 100$$

#Us	C1					C2					C1 + C2				
	#Co m. Cir.	#Err	#Com Real	Time	Acierto (%)	#Com . Circ.	#Err	#Com Real	Time	Acierto (%)	#Com. Circ.	#Err ores	#Com Real	Time	Acierto (%)
1	8	0	8	44s	100	8	3	13	1m 20s	62.5	16	2	19	1m 57s	75
2	8	0	8	45s	100	8	0	8	46s	100	16	2	18	1m 52s	75
3	8	3	15	1m 20s	62.5	8	4	12	1m 30s	62.5	16	4	27	2m 30s	75
4	8	3	20	1m 58s	62.5	8	3	15	1m 38s	62.5	16	5	38	>5m	68.75
5	8	1	9	1m 20s	87.5	8	1	9	56s	87.5	16	2	19	2m	93.75
6	8	2	11	1m 18s	75	8	3	12	2m 56s	62.5	16	4	23	2m 56s	75
7	8	4	23	2m 15s	50	8	4	20	2m 53s	50	16	10	64	>5m	37.5
8	8	1	9	50s	87.5	8	2	10	42s	75	16	3	19	1m 35s	81.25
9	8	2	17	2m 34s	75	8	2	14	1m 34s	75	16	4	35	3m 12s	75
10	8	2	14	1m 10s	75	8	1	9	1m 6s	87.5	16	3	23	1m 56s	81.25

Tabla 9: Resultados tras realizar los experimentos para la segunda prueba.

Las secuencias de comandos para realizar los circuitos son las siguientes:

- **Circuito 1:** ADELANTE, IZQUIERDA, ADELANTE, DERECHA, ADELANTE, DERECHA, ADELANTE, PARAR. Como podemos observar en este circuito existen dos giros a la derecha y uno a la izquierda, por lo que predomina el comando DERECHA sobre el comando IZQUIERDA.
- **Circuito 2:** ADELANTE, DERECHA, ADELANTE, IZQUIERDA, ADELANTE, IZQUIERDA, ADELANTE, PARAR. Como podemos observar en este circuito existen dos giros a la izquierda y uno a la derecha, por lo que predomina el comando IZQUIERDA sobre el comando DERECHA.
- **Circuito 1 + Circuito 2:** ADELANTE, IZQUIERDA, ADELANTE, DERECHA, ADELANTE, DERECHA, ADELANTE, IZQUIERDA, ADELANTE, DERECHA, ADELANTE, IZQUIERDA, ADELANTE, IZQUIERDA, ADELANTE, PARAR.

El diseño de estos circuitos se ha hecho de cara a comprobar el control del robot para cualquiera de los cuatro direcciones y poder detectar de este modo cuál de los comandos se recibe peor y qué frecuencia es la asociada a este comando y si convendría cambiarla.

Cabe destacar el caso del usuario 7 que es el que peor resultados ha dado. Este sujeto tiene un % de acierto del 50% para circuitos por separados y del 37.5% para el circuito conjunto. Observando las puntuaciones de las frecuencias finales asignadas (7Hz (47.4), 20Hz (55.3), 15Hz (8.6) y 16Hz (11.9)) vemos que la de mayor puntuación está asignada al comando parar que se utiliza sólo una vez al final en todos los circuitos y la puntuación para el comando izquierda es la más baja y el que se detectará peor, por lo tanto es uno de los casos en los que conviene reordenar estas frecuencias.

El caso del usuario 4 también es de mencionar puesto que su tiempo en el circuito conjunto es superior a los 5 minutos. Normalmente los usuarios con un tiempo mayor suelen llevar asociada alguna frecuencia con pobres niveles de detección para alguna de sus direcciones.

También mencionar a los sujetos 1, 2 y 8 cuyos resultados son muy buenos. Esto, en parte, puede deberse a que los sujetos posean una especial adaptación al sistema de estímulos, pues uno de ellos estaba bastante entrenado ya (sujeto 2).

Para finalizar este apartado, podemos observar que el algoritmo de control y el funcionamiento del sistema son bastante buenos ya que tenemos unas tasas de acierto bastante aceptables y buenas en promedio (superiores a las reportadas en [2], cuyos experimentos se hicieron con un casco de EEG de mejores prestaciones, pero sin adaptación al usuario).

A continuación se encuentra un enlace a un vídeo donde se puede ver a uno de los sujetos realizando un circuito de prueba:

<https://youtu.be/05xK1WYllsU>

6. Conclusiones

El objetivo principal de este proyecto era el del diseño y control de un robot a través de una interfaz cerebro máquina basado en SSVEPs con capacidad de adaptación al usuario para lo que se hizo un proceso de selección de frecuencias. Todos los dispositivos utilizados se caracterizan por ser de bajo coste.

Repasando los diferentes hitos enumerados en la sección de Objetivos podemos decir que los dos primeros, referentes al estudio de documentación y la elección de dispositivos hardware, se realizaron sin complicaciones por no entrañar ninguna dificultad especial. El tercer punto de diseño y construcción del robot no tuvo muchos problemas, más allá del sobrecalentamiento inicial solucionado con el cambio de baterías.

El punto referente a la comunicación inalámbrica fue quizás el que mayor problema entrañó por el hecho, primeramente de la amplia variedad de posibilidades referentes al protocolo a utilizar. Una vez elegido no fue fácil conseguir tampoco enviar el comando integrado en una URL sin usar un navegador. Además el uso de sockets desentraña siempre cierta complejidad.

Los siguientes tres puntos se pueden unir en uno sólo, que tiene que ver con el análisis de señal e integración de todo el sistema. La parte del análisis de señal, en parte ya nos venía dada por el trabajo previo desarrollado en el Grupo de Neurocomputación Biológica, básicamente estudiamos en detalle este análisis para poder integrarla con nuestra parte del control elaborada en este proyecto. Al integrarla nos dimos cuenta de que el casco enviaba información mucho más rápido de lo que la tarjeta Arduino conseguía procesar (debido en parte al tipo de inventariado realizado). Este problema lo resolvimos con el algoritmo analizado en la sección de desarrollo con el que además de minimizar el envío de datos conseguimos cerciorarnos de que el sujeto estaba respondiendo a un estímulo determinado.

En lo referente a la adaptabilidad al usuario, consistente en un proceso de selección de frecuencias con una fase inicial de barrido seleccionando las mejores en relación a la SNR y otra fase de compatibilidad, conseguimos unos resultados mayormente satisfactorios, pues se pretendía que la interfaz se adaptase al usuario de

modo que radicara en un mejor control del robot. Dentro de estos resultados cabe las mejores detecciones se producen en general en los sujetos en respuesta a estímulos a bajas frecuencias.

Para el control del robot realizado en los tres circuitos expuestos se obtuvieron unos resultados ligeramente variables en cuanto a recepción del comando (factor determinante en el control). Esta variabilidad viene dada por tener una frecuencia válida pero no óptima asociada a uno de los estímulos visuales, haciendo más complicada la generación del SSVEP a esa frecuencia y por tanto que se reciba ese comando. Esto trasladado a una cadena del circuito, conlleva un tiempo mayor simplemente por el hecho de recibir mal un comando determinado, aunque, como ya hemos comentado en la sección anterior, este problema se soluciona optimizando aún más la selección de frecuencias. Como apreciamos en la tabla de resultados de los circuitos, aunque los tiempos varían ligeramente, vemos como la tasa de aciertos en todos los circuitos se encuentra por encima del 50% salvo en un caso, lo que nos hace pensar en un sistema bastante fiable. Es destacable que 8 de cada 10 usuarios tienen un rendimiento mayor o igual del 75% en el recorrido del circuito 3, lo que nos hace pensar ya en un sistema exportable para controlar algunos tipos de neuroprótesis. Por lo expuesto concluimos que el sistema diseñado es una opción bastante eficiente para el control de robots con pocos grados de libertad. Aunque con este proyecto se ha dado un paso en este sentido convendría mejorar el sistema en términos de rendimiento, rango de frecuencias y dotando de más acciones al dispositivo a controlar.

Comentaremos ahora algunas pequeñas cuestiones a la vista de los resultados como puede ser el que el casco Emotiv no disponga de un electrodo en la posición Oz, lo que sin lugar a dudas haría que nuestros estímulos SSVEPs se detectaran con mayor facilidad y resolución, ya que sabemos que estos estímulos se producen en la parte occipital del cerebro y que al adquirir la señal, esta viene con ruido de fondo al tener que atravesar el cráneo y el cuero cabelludo, por lo que cuanto mayor sea la recepción de señal mejor.

Otra pequeña cuestión es la referente a la detección de estímulos cuando el usuario no se encuentra mirando a ninguno. En esta situación el casco sigue enviando información y detectando estímulos enviando datos de detección cada segundo esté el usuario mirando o no a un estímulo. El algoritmo implementado evita que parte de

estas detecciones se envíen pues se centra en detecciones correctas; pero aun así el sistema no está exento de error y convendría mejorar esta cuestión.

Cabe destacar una característica de este tipo de interfaz, en la que basta con entrenar al sujeto una vez para poder controlar el robot sin tener que volver a hacerlo en un largo periodo de tiempo, pues los datos quedan almacenados y las variaciones de las frecuencias a las que un usuario es más o menos sensible pueden variar a lo largo de los meses pero se mantienen para pocos días e incluso durante semanas.

Finalmente, se debería estudiar la posible inclusión de más grados de libertad dotando de esta manera al robot de más acciones, teniendo en cuenta que esto estaría limitado por el número de frecuencias válidas así como por el barrido en frecuencias realizado.

7. Estudio futuro.

7.1. Tabletas y Smartphones.

En apartados anteriores hemos citado la importancia que tiene en nuestro proyecto la conectividad WiFi, el por qué buscamos un hardware que tuviera capacidad inalámbrica tanto como para la tarjeta del robot como para el casco de adquisición Emotiv Epoc. Gracias a estos dispositivos inalámbricos se facilitaron diferentes tareas como la adquisición de los datos de la actividad cerebral y tiempo de preparación, así como se favoreció la libertad de movimiento tanto para el individuo como para el robot. Además estos dispositivos tienen la característica de ser multiplataforma pudiéndose adaptar a cualquier sistema operativo (Windows, Linux o Mac) y aceptando también diferentes lenguajes de programación, aunque en el caso de este proyecto los programas se escribieron principalmente en C++. Gracias a estas características se nos brinda la oportunidad de poder disponer de un BCI sin la necesidad de estar en un puesto fijo de PC.

Como ya explicamos anteriormente el control del robot se puede llegar a hacer desde cualquier dispositivo tableta o Smartphone gracias al tipo de protocolo implementado de peticiones HTTP enviando los comandos por URL desde un navegador, aunque sería ideal la existencia de un software que pudiera gestionar estos comandos y hacer el análisis de señal online en el mismo dispositivo. Como hemos visto algunas compañías como Emotiv y NeuroSky ya han dado un paso adelante en este sentido haciendo cascos (*neuroheadsets*) inalámbricos y multiplataforma, aunque aún no se ha investigado su uso en Android, para *tabletas* y *smartphones*. Esto nos presenta un cambio de paradigma haciendo posible el diseño de nuevos tipos de experimentos que puedan caracterizar señales cerebrales en una interacción normal durante largos periodos de tiempo. Además a estos sistemas o aplicaciones se les pueden llegar a añadir más grados de libertad con la inclusión de alguna luz más dando lugar a nuevas opciones de control sobre el dispositivo en cuestión [34].



Figura 23: SBS2 para aplicaciones móviles de EEG en dispositivos Android conectados a casco Emotiv or EasyCap [36].

Tomando en consideración la idea de llevar los sistemas BCI y sus aplicaciones un paso más lejos el *Smartphone Brain Scanner (SBS2)* de software libre [35], hace posible el uso de aplicaciones de reconstrucción en 3D por monitoreo cerebral en tiempo real. Para ello se combina un casco inalámbrico con un *Smartphone* o *tableta* Android. Este software se ha validado inicialmente con tres experimentos relacionados con: el control motor a través de BCI, y la incorporación de semántica e interfaces de retroalimentación para ilustrar la fiabilidad de la adquisición de actividad cerebral en un entorno móvil. Además este software se encuentra en C++ y sus aplicaciones principales de escritorio están disponibles en modo multiplataforma (Windows, Linux o Mac) [36]. El sistema desarrollado en este PFC podría combinarse con estas otras iniciativas.

7.2. Sensores y Webcam de seguimiento.

Cuando surgió la idea de este proyecto enseguida aparecieron diversas mejoras para complementarlo, en el contexto de la adaptación a cada usuario sobre todo centrándonos en las personas discapacitadas. Algunas como las ya mencionadas trasladan el sistema de control visual a una *tableta* o *Smartphone* y otras como las que analizaremos ahora nos permiten un seguimiento remoto del dispositivo a controlar.

La primera idea que nos vino a la mente, fue la de incorporar una cámara de video WiFi en el chasis que nos permitiera tener una perspectiva del campo visual del propio robot, de manera que el usuario pueda decidir el mejor camino a seguir evadiendo los obstáculos existentes. A medida que avanzaba el proyecto esta posible mejora fue tomando forma y al final decidimos utilizar una cámara no inalámbrica en

modo ambiente de modo que podamos visualizar el campo donde el robot se estaba moviendo, dado que la WiFi del robot no nos permitía el control a grandes distancias. Hay que decir que finalmente se utilizó sólo en los inicios de las pruebas de control como referencia. Sería interesante investigar sobre su incorporación con la interfaz de estimulación visual en sujetos comprobando su compatibilidad con los requisitos de la estimulación parpadeante para generar los SSVEPs.

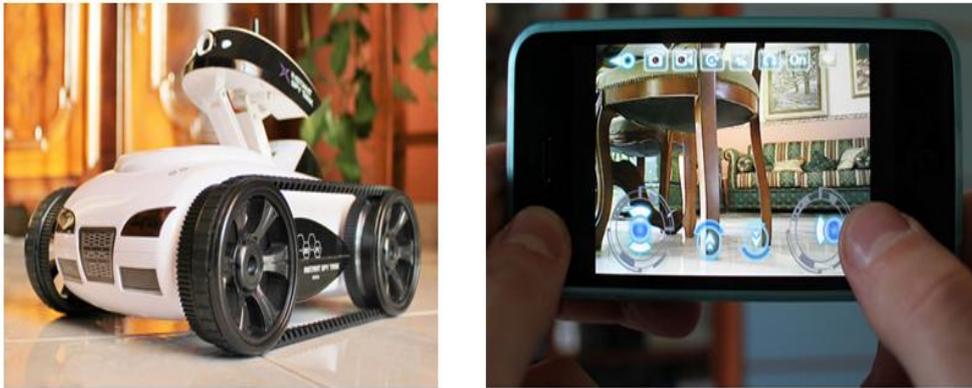


Figura 24: Control de un robot con cámara a través de smartphone.

Por último, mencionar como posibles estudios o proyectos futuros la inclusión de un sistema de sensores de seguimiento, así como añadir algunos grados más de libertad en el robot de modo que le dotemos con más capacidades. Este sistema de sensores nos permitiría tener una idea más clara de la situación del robot y al igual que la videocámara inalámbrica se podría integrar como pantalla en miniatura con el sistema de estímulos visuales. Esto conllevaría resolver, aparte del problema ya citado, una problemática en el rango de alcance del sistema de comunicación inalámbrico, dado que la tarjeta Arduino utilizada actualmente no dispone de un gran alcance y se pierde la señal a los pocos metros. Se debería estudiar el caso con otro sistema inalámbrico de más alcance.

Sin duda vemos que existen posibles campos para continuar el estudio en esta misma línea, incluyendo también el estudio de otras vertientes similares como el control de otros tipos de dispositivos robóticos como pueden ser las neuroprótesis. Debo mencionar que esta última área de investigación sería de especial interés ya que implicaría la inclusión a nivel más sensorial de los algoritmos en lazo cerrado (*“close loop”*) tan en auge hoy en día, así como supondría grandes avances de cara a proveer a personas con discapacidades motoras de un medio de comunicación adaptado a sus circunstancias que les haga más independientes y minimizando los cuidados que necesiten.

8. Referencias.

- [1] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain Computer Interfaces, a Review," *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012.
- [2] R. Ortner, C. Guger, R. Prueckl, E. Grünbacher, and G. Edlinger, "SSVEP Based Brain-Computer Interface for Robot Control," *Comput. Help. People with Spec. Needs, Part II 12th Int. Conf. ICCHP 2010, Vienna, Austria, July 14-16, 2010. Proc.*, pp. 85–90, 2010.
- [3] M. Ahn and S. C. Jun, "Performance variation in motor imagery brain–computer interface: A brief review," *J. Neurosci. Methods*, vol. 243, pp. 103–110, 2015.
- [4] A. M. Green and J. F. Kalaska, "Learning to move machines with the mind," *Trends Neurosci.*, vol. 34, no. 2, pp. 61–75, 2011.
- [5] "Emotiv." [Online]. Available: <https://emotiv.com/>.
- [6] D. E. Thompson, S. Blain-Moraes, and J. E. Huggins, "Performance assessment in brain-computer interface-based augmentative and alternative communication.," *Biomed. Eng. Online*, vol. 12, no. 1, p. 43, 2013.
- [7] G. Müller-Putz, "Game-like training to learn single switch operated neuroprosthetic control," *BRAINPLAY 07 Brain- ...*, pp. 41–43, 2007.
- [8] Grupo de Ingeniería Biomédica (GIB). Universidad de Valladolid, "Brain-Computer Interface (BCI) aplicado al entrenamiento cognitivo y control domótico para prevenir los efectos del envejecimiento." [Online]. Available: http://www.fgcsic.es/lychnos/es_es/articulos/Brain-Computer-Interface-aplicado-al-entrenamiento-cognitivo.
- [9] "Esclerosis Lateral Amiotrófica (ALS)." [Online]. Available: https://es.wikipedia.org/wiki/Esclerosis_lateral_amiotr%C3%B3fica.
- [10] eurekaalert, "Una interfaz adaptativa para controlar el ordenador con el pensamiento," 2006. [Online]. Available: <http://www.eurekaalert.org/staticrel.php?view=ef06142>.
- [11] C. Kapeller, C. Hintermuller, M. Abu-Alqumsan, R. Pruckl, a. Peer, and C. Guger, "A BCI using VEP for continuous control of a mobile robot," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 5254–5257, 2013.
- [12] F.-B. Vialatte, M. Maurice, J. Dauwels, and A. Cichocki, "Steady-state visually evoked potentials: focus on essential paradigms and future perspectives.," *Prog. Neurobiol.*, vol. 90, no. 4, pp. 418–38, May 2010.

- [13] D. Regan, "Electrical responses evoked from the human brain.," *Sci. Am.*, vol. Vol 241(6), pp. 134–146, 1979.
- [14] E. M. Holz and L. Botrel, "Bridging Gaps : Long-Term Independent BCI Home-Use by a Locked-In End-User," *1 Inst. Psychol. Univ. Würzburg, Ger. 1 Inst. Psychol. Univ. Würzburg, Ger.*, vol. 21, pp. 65–66, 2013.
- [15] Jarod T. Giger* & Martha Markwardb, "The Need to Know Caregiver Perspectives Toward Using Smart Home Technology," *1Department Soc. Work. Univ. South Dakota, Vermillion, South Dakota 57069, USA. Jarod.Giger@usd.edu*, pp. 96–114, 2011.
- [16] M. Jiménez, P. Sánchez, F. Rosique, B. Álvarez, and A. Iborra, "A tool for facilitating the teaching of smart home applications," *Comput. Appl. Eng. Educ.*, vol. 22, no. 1, pp. 178–186, 2014.
- [17] E. Mikołajewska and D. Mikołajewski, "Neuroprostheses for increasing disabled patients' mobility and control," *Adv. Clin. Exp. Med.*, vol. 21, no. 2, pp. 263–272, 2012.
- [18] E. Farwell, L.A. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr. Clin. Neurophysiol. Dept. Psychol. Cogn. Psychophysiol. Lab. Univ. Illinois Urbana-Champaign, Champaign, 61820, U.S.A.*, pp. 510–23, 1988.
- [19] G. R. Müller-Putz and G. Pfurtscheller, "Control of an electrical prosthesis with an SSVEP-based BCI.," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 1, pp. 361–364, 2008.
- [20] P. J. Grahn, G. W. Mallory, B. M. Berry, J. T. Hachmann, D. a. Lobel, and J. L. Lujan, "Restoration of motor function following spinal cord injury via optimal control of intraspinal microstimulation: toward a next generation closed-loop neural prosthesis," *Front. Neurosci.*, vol. 8, no. September, pp. 1–12, 2014.
- [21] M. Ahn, M. Lee, J. Choi, and S. C. Jun, "A Review of Brain-Computer Interface Games and an Opinion Survey from Researchers, Developers and Users.," *Sensors (Basel)*, vol. 14, no. 8, pp. 14601–14633, 2014.
- [22] M. Middendorf, J. K. Calhoun G, and M. G, "Brain-computer interfaces based on the steady-state visual-evoked response.," *Middend. Sci. Serv. Inc., Medway, OH 45341, USA.*, p. 211, 2000.
- [23] Á. Morán García, "PFC_Álvaro_Morán_García," *Proy. fin carrera*, p. 105, 2015.
- [24] J. Fernandez-Vargas, H. U. Pfaff, F. B. Rodriguez, and P. Varona, "Assisted closed-loop optimization of SSVEP-BCI efficiency," *Front. Neural Circuits*, vol. 7, no. 27, 2013.

- [25] A. Jacobo and F. Vargas, "Optimization of SSVEP - BCIs through an assisted closed - loop," *Trab. Fin Máster*, pp. 1–47, 2013.
- [26] "Eeg_10_20." [Online]. Available: <http://www.bem.fi/book/13/13.htm>.
- [27] "arduino." [Online]. Available: <http://www.arduino.cc/>.
- [28] J. González, "Miniskybot 2." [Online]. Available: http://www.learobotics.com/wiki/index.php?title=Miniskybot_2#Robot_Miniskybot:_P.C3.A1gina_principal.
- [29] "Tienda RS." [Online]. Available: <http://es.rs-online.com/web/>.
- [30] "Tienda Farnell." [Online]. Available: <http://es.farnell.com/>.
- [31] "OpenScad." [Online]. Available: <http://www.openscad.org/index.html>.
- [32] Mathworks, "Signal Processing Toolbox™ User ' s Guide R 2015 a," 2015.
- [33] J. Tong and D. Zhu, "Multi-phase cycle coding for SSVEP based brain-computer interfaces," *Biomed. Eng. Online*, vol. 14, no. 1, p. 5, 2015.
- [34] A. Guneyasu and H. L. Akin, "An SSVEP based BCI to control a humanoid robot by using portable EEG device," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 6905–6908, 2013.
- [35] "Smartphone Brain Scanner (SBS2)." [Online]. Available: <https://github.com/SmartphoneBrainScanner>.
- [36] A. Stopczynski, C. Stahlhut, M. K. Petersen, J. E. Larsen, C. F. Jensen, M. G. Ivanova, T. S. Andersen, and L. K. Hansen, "Smartphones as pocketable labs: Visions for mobile brain imaging and neurofeedback.," *Int. J. Psychophysiol.*, pp. 54–66, Aug. 2013.
- [37] Emotiv Systems, "Emotiv EPOC," 2011.
- [38] N. I. of N. D. and Stroke and N. I. of Health, "Esclerosis lateral amiotrófica." [Online]. Available: http://espanol.ninds.nih.gov/trastornos/esclerosis_lateral_amiotrofica.htm.
- [39] V. B. Plaza, B. M. Pacheco, C. M. Aguilar, J. F. Valenzuela, and J. J. Z. Pérez, "Lesión de la médula espinal. Actualización bibliográfica: Fisiopatología y tratamiento inicial," *Coluna/ Columna*, vol. 11, no. 1, pp. 73–76, 2012.
- [40] RFC HTTP, "Definición Peticiones HTTP." [Online]. Available: <http://www.ietf.org/rfc/rfc2616.txt>.

- [41] R. J. Ross, "Microelectronics Failure Analysis: Desk Reference," *ASM Int. 2004*, no. ISBN 0871708043, 9780871708045, p. 800, 2011.
- [42] H. Neuroscience, L. F. Gruss, M. J. Wieser, S. R. Schweinberger, and A. Keil, "Face-evoked steady-state visual potentials : effects of presentation rate and face inversion," vol. 6, no. November, pp. 1–10, 2012.
- [43] T. Padir and J. Schaufeld, "Design of a Human Hand Prosthesis," *Fac. Worcester Polytech. Inst.*, p. 74, 2012.
- [44] "Emotiv videogame." [Online]. Available: <http://www.techfresh.net/wp-content/uploads/2008/08/emotiv-epoc-headset.jpg>.
- [45] P. Teich, "Master ' s Thesis : Designing a Brain Computer Interface Using an Affordable EEG Headset .," 2015.

Glosario

Interfaz Cerebro Máquina (BCI): Canal de comunicación que permite a una persona interactuar con un ordenador incluso controlar dispositivos a partir de señales cerebrales [1], [2].

Electroencefalograma (EEG): Exploración neurofisiológica basada en el registro de actividad bioeléctrica cerebral en condiciones de reposo, vigilia o sueño, y durante diversas actividades (por ejemplo, estimulación lumínica intermitente) mediante un equipo de electroencefalografía [1].

Steady-State Visually Evoked Potentials (SSVEP): Señales generadas como respuestas a la estimulación visual a diferentes frecuencias. Cuando la retina es excitada por un estímulo visual entre 3.5 Hz y 75Hz, el cerebro genera actividad eléctrica a la misma frecuencia (o múltiplos) del estímulo visual [12].

Arduino: Plataforma de hardware libre, de bajo coste y altas prestaciones, está basada en una tarjeta con un microcontrolador y en entorno de desarrollo sencillo. Está diseñada para usar la electrónica en entornos multidisciplinares [27].

Neuroprótesis: Una neuroprótesis es un dispositivo que reemplaza la función de una parte dañada del cuerpo, creando una interfaz con el sistema nervioso [4].

Emotiv Epoc: Casco comercial de bajo coste para registrar EEG que actúa como una interfaz personal. De este modo las personas pueden interactuar con ordenadores a través de ella. Es capaz de realizar un registro de electroencefalografía de alta resolución multicanal, utilizando sensores que transforman la actividad cerebral en señales eléctricas [37].

SDK: Kit de desarrollo software dotado con herramientas útiles para que el programador pueda crear aplicaciones de control de dispositivos o sistemas.

Esclerosis Lateral Amiotrófica (ALS): Enfermedad degenerativa de tipo neuromuscular. Provoca una parálisis muscular progresiva debido a la disminución gradual de células motoneuronas en el sistema nervioso [38].

Baseline: Será la línea de base que grabaremos cuando el individuo se encuentra en un estado de reposo de estimulación visual, esto es, mirando un fondo completamente negro sin luces parpadeantes.

Electrodo: Conductor eléctrico utilizado para hacer contacto con una parte no metálica del circuito. En el caso de los cascos usados para electroencefalografía, los electrodos sirven de contacto entre el circuito y la cabeza del usuario.

Asisted Close loop (ACL): Proceso que permite a la interfaz adaptar las frecuencias de estimulación para un usuario concreto y teniendo en cuenta su compatibilidad cuando se exponen varios estímulos simultáneamente [24], [25].

Filtro paso bajo: Corresponde a un filtro electrónico caracterizado por permitir el paso de las frecuencias más bajas y atenuar las frecuencias más altas.

Artefactos: Esta palabra está asociada a dispositivos que registran actividad cerebral y está referida a toda variación de señal no originada por el órgano cuya actividad se desea registrar.

Gaze-tracking: Técnica para evaluar el punto de fijación de la mirada o el movimiento del ojo en relación al resto de la cabeza.

Spinal Cord Injury (SCI) o lesión de médula espinal: Alteración de la médula espinal que puede provocar una pérdida de la sensibilidad o movilidad [39].

Peticiones HTTP: Peticiones del protocolo usado para transacciones con la World Wide Web. En este proyecto se usarán para comunicarnos con el servidor web creado en el programa de la tarjeta Arduino por medio del comando Post [40].

Servomotor (Servo): Es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación y mantenerse estable en dicha posición. Motor eléctrico que puede ser controlado tanto en velocidad como en posición.

Regulador de tensión: Dispositivo electrónico diseñado para mantener un nivel de tensión constante. En este proyecto se usará el 7805 que mantiene una tensión constante de 5V [41].

Transformada de Fourier: Transformación matemática empleada para transformar señales entre el dominio del tiempo y el dominio de la frecuencia, que tiene muchas aplicaciones en ingeniería. Además es reversible. El propio término se refiere tanto a la operación de transformación como a la función que produce.

Enventanado: Enventanar consiste en limitar una señal continua en el tiempo para su análisis. A esta señal acotada se le suele aplicar una función para limitar los efectos y distorsiones derivados de haber acotado una señal continua. Al enventanar se pueden dar efectos como el solapamiento de ventanas en el cual la parte solapada será el número de muestras que una ventana y la anterior comparten.

Anexos

Anexo I: Código de identificación de frecuencia y transformación a comando.

(Ver sección 4.3 Transformación a un comando para una explicación más detallada).

```

/*****
/*      CONTADORES      */
*****/
int cont_Avanzar = 0;
int cont_Parar = 0;
int cont_Izquierda = 0;
int cont_Derecha = 0;
int ultima_frecuencia;
ultima_frecuencia = 0;

do
{
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(80);
    inet_pton(AF_INET, str, &servaddr.sin_addr);
    connect(sockfd, (SA *) & servaddr, sizeof(servaddr));

    printf("\nMira a las flechas papadeantes para mover el robot:");

    //Algoritmo de mejora del control del robot.

    errorTransmision = msgrcv(idCola, &comunicacionRobot, sizeof(comunicacionRobot), 0, MSG_NOERROR);
    if (errorTransmision < 0)
    {
        printf("    \e[1;31m[ERROR]\e[0m \e[0;37m Transmisión de datos fallida\e[0m");
        return 1;
    }

    if (comunicacionRobot.ComandoRobot != ultima_frecuencia)
    {
        cont_Avanzar = 0;
        cont_Parar = 0;
        cont_Izquierda = 0;
        cont_Derecha = 0;
    }

    /*****IDENTIFICACION DEL COMANDO*****/

    if (comunicacionRobot.tipoComunicacion == IPC_DATO)
    {
        if (comunicacionRobot.ComandoRobot == frecuenciaAvanzar)
        {
            cont_Avanzar++;
            if (cont_Avanzar >= 2)
            {
                printf("\nHas elegido la tecla de direccion 'arriba'\n\n");
                page = "/arduino/arriba/1";
                process_http(sockfd, page);
                cont_Avanzar = 0;
                cont_Parar = 0;
                cont_Izquierda = 0;
                cont_Derecha = 0;
            }

            ultima_frecuencia = frecuenciaAvanzar;
        }
    }
}

```

```

        if (comunicacionRobot.ComandoRobot == frecuenciaParar)
        {

            cont_Parar++;
            if (cont_Parar >= 2)
            {
                printf("\nHas elegido la tecla de direccion 'abajo'\n\n");
                page = "/arduino/parar/1";
                process_http(sockfd, page);
                cont_Avanzar = 0;
                cont_Parar = 0;
                cont_Izquierda = 0;
                cont_Derecha = 0;
            }

            ultima_frecuencia = frecuenciaParar;
        }
        if (comunicacionRobot.ComandoRobot == frecuenciaIzquierda)
        {

            cont_Izquierda++;
            if (cont_Izquierda >= 2)
            {
                printf("\nHas elegido la tecla de direccion 'izquierda'\n\n");
                page = "/arduino/izquierda/1";
                process_http(sockfd, page);
                cont_Avanzar = 0;
                cont_Parar = 0;
                cont_Izquierda = 0;
                cont_Derecha = 0;
            }

            ultima_frecuencia = frecuenciaIzquierda;
        }

        if (comunicacionRobot.ComandoRobot == frecuenciaDerecha)
        {

            cont_Derecha++;
            if (cont_Derecha >= 2)
            {
                printf("\nHas elegido la tecla de direccion 'derecha'\n\n");
                page = "/arduino/derecha/1";
                process_http(sockfd, page);
                cont_Avanzar = 0;
                cont_Parar = 0;
                cont_Izquierda = 0;
                cont_Derecha = 0;
            }

            ultima_frecuencia = frecuenciaDerecha;
        }
    }

    if (comunicacionRobot.tipoComunicacion == IPC_FIN)
    {

        ultima_frecuencia = 0;

        printf (" \e[1;31m[INFO]\e[0m \e[0;32m FINALIZANDO RECEPCIÓN DE DATOS\e[0m");
        return 0;
    }

}

close(sockfd);
}while(1);

```

Anexo II: Código del Servidor Web.

```

void setup() {
  Serial.begin(9600);

  pinMode(13,OUTPUT);
  digitalWrite(13, HIGH);
  Bridge.begin();
  digitalWrite(13, LOW);

  /*In the second part of setup(),tell the instance of YunServer
  to listen for incoming connections only coming from localhost.
  Connections made to Linino will be passed to the 32U4 processor
  for parsing and controlling the pins.
  This happens on port 5555. Start the server with server.begin().*/

  server.listenOnLocalhost();
  server.begin();
}

void loop() {
  YunClient client = server.accept();

  if (client)
  {
    process(client);
    //client.stop();
  }
}

/*Create a function named process that accepts the YunClient as its argument
Read the command by creating a string to hold the incoming information.
Parse the REST commands by their functionality (digital, analog, and mode)
and pass the information to the appropriately named function.*/

void process(YunClient client)
{
  String command = client.readStringUntil('/');

  if (command == "arriba")
  {
    digitalWrite(13, HIGH);
    delay(10);
    digitalWrite(13, LOW);
    delay(10);

    Robot.Parar(ServoD, ServoI);

    Robot.Adelante(ServoD, ServoI);
    delay(1000);
    Robot.Parar(ServoD, ServoI);
  }
  if (command == "parar")
  {
    digitalWrite(13, HIGH);
    delay(10);
    digitalWrite(13, LOW);
    delay(10);

    Robot.Parar(ServoD, ServoI);
    delay(1000);
  }
  ...
}

```

Anexo III: Clase Robot.

```

Robot.cpp - Biblioteca para el envío de clase Robot.
Creado por Manuel Marcos, el 13 de Marzo de 2014.
Dominio privado.
*/
#include <Arduino.h>
#include <Servo.h>
#include <Robot.h>

Robot::Robot(Servo ServoD, Servo ServoI)
{
    ServoD=ServoD;
    ServoI=ServoI;
}

void Robot::Setup(Servo ServoD, Servo ServoI)
{
    ServoD.attach(A0);
    ServoI.attach(A1);
}

void Robot::Parar(Servo ServoD, Servo ServoI)
{
    ServoD.detach();
    ServoI.detach();
}

void Robot::Adelante(Servo ServoD, Servo ServoI)
{
    ServoD.attach(A0);
    ServoI.attach(A1);
    ServoD.write(0);
    ServoI.write(180);
}

void Robot::Atras(Servo ServoD, Servo ServoI)
{
    ServoD.attach(A0);
    ServoI.attach(A1);
    ServoD.write(180);
    ServoI.write(0);
}

void Robot::Derecha(Servo ServoD, Servo ServoI)
{
    ServoD.attach(A0);
    ServoI.attach(A1);
    ServoD.write(180);
    ServoI.write(180);
}

void Robot::Izquierda(Servo ServoD, Servo ServoI)
{
    ServoD.attach(A0);
    ServoI.attach(A1);
    ServoD.write(0);
    ServoI.write(0);
}

```

```

Robot.h - Biblioteca para el envío de clase Robot.
Creado por Manuel Marcos, el 13 de Marzo de 2014.
Dominio privado.
*/
#ifndef Robot_h
#define Robot_h

#include <Arduino.h>
#include <Servo.h>

class Robot{
public:
    Robot(Servo ServoD, Servo ServoI);
    void Setup(Servo ServoD, Servo ServoI);
    void Parar(Servo ServoD, Servo ServoI);
    void Adelante(Servo ServoD, Servo ServoI);
    void Atras(Servo ServoD, Servo ServoI);
    void Derecha(Servo ServoD, Servo ServoI);
    void Izquierda(Servo ServoD, Servo ServoI);
private:
    Servo ServoD;
    Servo ServoI;
};
#endif

```

Presupuesto

Hardware

- Ordenador (PC)..... 500 €
- Pantalla 100 €
- Casco de EEG Emotiv Epoc 314 €
- 4 Servomotores Futaba 3000 28 €
- Componentes varios para montaje del robot y placa auxiliar 20 €

Software

- SDK del Emotiv Epoc..... 236 €
- Licencia para el uso educativo del Emotiv 1412 €

Ejecución material y Honorarios

- Subtotal de ejecución material (hardware y software)..... 2610 €
- Gastos generales (16% del subtotal) 418 €
- Beneficio industrial (6% del subtotal) 157 €
- Honorarios (790 horas a 15 €/hora) 11850 €

Material fungible

- Agua ionizada 10 €
- Electrodos para el casco Emotiv (x2)..... 100 €
- Fotocopias e impresión de artículos..... 60 €
- Impresión de tomos..... 180 €
- Encuadernación 50 €

Total presupuesto

- Subtotal del presupuesto 15435 €
- I.V.A. Aplicable (21%)..... 3241 €
- Total presupuesto..... 18676 €

Madrid, Julio de 2015

El Ingeniero Jefe de Proyecto

Fdo.: *Manuel Marcos García*

Ingeniero de Telecomunicación

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un *diseño y control de un robot mediante interfaz cerebro-máquina usando la tecnología SSVEP*. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder

de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.