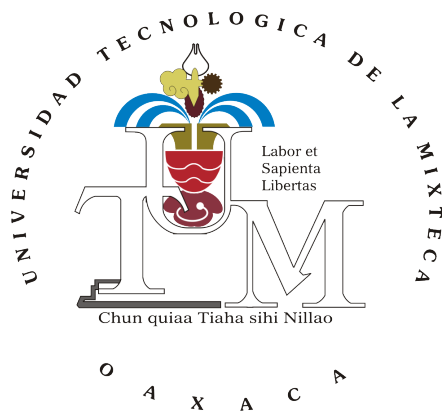




UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

“DESARROLLO DE UNA INTERFAZ CEREBRO-COMPUTADORA BASADA EN EEG PARA CONTROLAR UNA MANO ROBÓTICA”

TESIS

PARA OBTENER EL TÍTULO DE:

INGENIERO EN MECATRÓNICA

PRESENTA:

ANGEL DEL ROSARIO ZARATE RUIZ

DIRECTOR DE TESIS:

DR. MANUEL ARIAS MONTIEL

HUAJUAPAN DE LEÓN, OAXACA, MÉXICO, AGOSTO DEL 2024

Tesis presentada en Agosto de
2024 ante los sinodales:

Dr. Rosebet Miranda Luna
Dr. Enrique Guzmán Ramírez
M.C. Juan Pablo Avendaño
Santiago

Director de tesis:

Dr. Manuel Arias Montiel

Agradecimientos

En un principio, no sabía qué ni cómo agradecer sin embargo, una vez que voltee a ver hacia atrás, me dí cuenta que hay muchas cosas y sobre todo a muchas personas a las que debo agradecer.

Primeramente a mi mamá, Silvia, que con mucho esfuerzo y a pesar de los días malos, supo apoyar mis sueños y darme las herramientas para salir adelante. Nunca sabría cómo pagarle todo lo que hizo por mí pero espero algún día poder retribuir al menos una parte de todo aquello que me brindó.

A mi hermano, Germain, le agradezco la paciencia y el apoyo que me mostraba cuando ni yo sabía cómo salir adelante. Lamento los regaños innecesarios pero agradezco lo mucho que te preocupas por mí.

A mi director Manuel Arias Montiel, quien confió en mí para desarrollar este proyecto tan ambicioso y quien desde el servicio social, me ha brindado su apoyo y libertad para resolver todos y cada uno de los problemas que se han presentado.

A Fredy y Beny quienes no solo son mis amigos sino también parte de mi familia. Les agradezco que a través de las risas y peleas, siempre me hayan brindado ese apoyo y soporte que necesitaba. Han sido parte fundamental de este trabajo ya que aún a pesar de las dudas, siempre me mostraron que nunca hay que rendirse, les agradezco muchísimo chicos.

A mi novia Aleydis quien me enseñó que los sueños pueden volverse realidad y que ni siquiera el cielo puede limitarnos a soñar. Me enseñó que siempre hay una salida por más oscura que sea la situación. Te agradezco profundamente por haber llegado a mi vida y mostrarme que el horizonte no es el final del camino.

A mis amigos Rosario, Fernando, Jesús, Juan Carlos, Silvia, quienes estuvieron apoyándome a pesar de todo, que creyeron en mí y que siempre me mostraron una sonrisa ante cada logro, que siempre estuvieron ahí a pesar del tiempo y la distancia. Gracias por hacerme sentir en familia.

A la Dra. Alejandra Velarde Galván por sus consejos, regaños y apoyo que me brindó. Quien a pesar de todo, apoyó este sueño y me encaminaba cuando había perdido el sendero. Le agradezco que a pesar de mis malos ratos, me haya sabido guiar y me haya tenido tanta paciencia.

A la Dra. Esther Lugo González por apoyarme en el sueño ambicioso y con quien comparto sueños, metas y nuevas ideas. Quien nunca juzgó lo ambicioso de mis metas y que por el contrario, me mostraba que siempre se puede lograr algo nuevo.

Al Dr. Christian Eduardo Millán Hernández quien a pesar de tener poco conociéndome, me brindó apoyo y guía. Le agradezco profundamente que me haya compartido parte de su conocimiento y me mostrara que un problema es también un área de oportunidad para mejorar.

Índice

1. Introducción	1
1.1. Antecedentes	2
1.1.1. Deletreadores	2
1.1.2. Videojuegos	4
1.1.3. Prototipos móviles	6
1.2. Planteamiento del problema	9
1.3. Justificación	10
1.4. Hipótesis	10
1.5. Objetivos	10
1.5.1. Objetivo general	10
1.5.2. Objetivos específicos	10
1.6. Propuesta de solución	11
1.7. Metodología de desarrollo	12
1.8. Estructura de la tesis	13
2. Marco Teórico	15
2.1. El cerebro humano	15
2.1.1. Generación de picos	15
2.1.2. Organización del cerebro	15
2.2. Señales EEG	16
2.2.1. Bandas de frecuencia	17
2.2.2. Artefactos en EEG	18
2.2.3. SSVEP	19
2.3. Procesamiento de señales	19
2.3.1. Análisis en el dominio de la frecuencia	19
2.4. Parámetros Hjorth	21
2.5. Entropía de Shannon	21
2.6. Análisis en el dominio del tiempo	21
2.6.1. Valor Medio Absoluto (MAV)	21
2.6.2. Integral Cuadrada Simple (SSI)	22
2.6.3. Longitud de Forma de Onda (WL)	22
2.7. Técnicas de reducción de la dimensionalidad	23

2.7.1.	Análisis de Componentes Principales (PCA)	23
2.7.2.	Análisis Discriminante Lineal (LDA)	23
2.8.	Técnicas de clasificación	24
2.8.1.	Máquinas de Soporte Vectorial (SVM)	24
2.8.2.	Redes Neuronales Artificiales (ANN)	25
3.	Diseño Electroencefalográfico	29
3.1.	Diseño de la fuente	29
3.2.	Protocolo de adquisición	33
3.2.1.	Materiales	33
3.2.2.	Reclutamiento de participantes	33
3.2.3.	Métodos	34
3.3.	Señales EEG	36
3.3.1.	Adquisición	36
3.3.2.	Pre-procesamiento	38
4.	Clasificación de Señales e Integración	43
4.1.	Sistema de clasificación	43
4.1.1.	Evaluación de los modelos	44
4.1.2.	Entrenamiento ANN	45
4.1.3.	Entrenamiento SVM	50
4.1.4.	Selección de sistema y modelo de clasificación	52
4.2.	Acondicionamiento de la mano robótica virtual	54
4.2.1.	Selección de la mano robótica a utilizar	54
4.2.2.	Ajuste y acondicionamiento de la mano robótica virtual	56
4.3.	Integración de sistemas	58
4.3.1.	Comunicación GUI con Matlab	58
4.3.2.	Comunicación Adams View y Matlab	59
4.3.3.	Script de integración en Matlab	60
5.	Pruebas y Resultados	63
5.1.	Modelo de clasificación en línea	63
5.2.	Desempeño de la clasificación de la BCI	64
5.3.	Movimiento de la mano robótica	65
5.3.1.	Movimiento de saludo	66
5.3.2.	Movimiento de pinza fina	68
5.4.	Evaluación general	71
6.	Conclusiones y Trabajos Futuros	75
6.1.	Conclusiones	75
6.2.	Trabajos futuros	77

A. Documentación para la extracción de las señales EEG	79
B. Script utilizado para el pre-procesamiento y caracterización de la señal EEG	87
C. Script del modelo de clasificación utilizado	95
D. Script de comunicación e integración de sistemas	97
Bibliografía	102

Índice de figuras

1.1.	Estímulos presentados en el triple RSVP [3]	3
1.2.	Figura estereoscópica resuelta [7]	4
1.3.	GUI desarrollado para retroalimentación [8]	5
1.4.	Juego desarrollado por Filiz y Arslan. Los cuadros amarillos son los estímulos, el cuadro blanco el avatar y el cuadro rojo representa el cebo a alcanzar [9].	6
1.5.	Robot móvil operado mediante BCI [12]	9
2.1.	Sistema 10-20 internacional [16]	17
2.2.	Ejemplos de ondas EEG y su banda de frecuencias [1]	18
2.3.	Curva característica de activación de la función sigmoide.	26
2.4.	Curva característica de activación de la función escalón.	26
2.5.	Curva característica de activación de la función ReLu.	27
2.6.	Estructura básica de una ANN con arquitectura MLP.	27
3.1.	Configuración de los estímulos visuales implementados	30
3.2.	Distribución de puertos ATmega328P [26].	31
3.3.	Esquema eléctrico del estimulador visual.	32
3.4.	Diagrama de conexiones y distribución del estimulador visual.	33
3.5.	Casco Ultracortex impreso	37
3.6.	Distribución de los nodos de conexión.	37
3.7.	Configuración final casco Ultracortex.	38
3.8.	GUI para la adquisición de señales.	39
3.9.	Diagrama de procesamiento digital de las señales EEG	40
3.10.	Matriz de características final.	42
4.1.	Matriz de confusión para 2 clases	44
4.2.	Descripción de la matriz de confusión utilizada	46
4.3.	Curva de pérdida ANN con características relevantes del SSVEP	47
4.4.	Matriz de confusión ANN con características relevantes del SSVEP	47
4.5.	Curva de pérdida ANN con 5 características PCA	48
4.6.	Matriz de confusión ANN con 5 características PCA	49
4.7.	Curva de pérdida ANN con aplicación del LDA	50
4.8.	Matriz de confusión ANN con aplicación del LDA	50

4.9. Matriz de confusión de la SVM entrenada con 36 características.	51
4.10. Matriz de confusión de la SVM entrenada con 36 características.	52
4.11. Matriz de confusión de la SVM entrenada con LDA.	53
4.12. Configuración Final de la ANN para el sistema de 31 y 33 Hz.	54
4.13. Configuración Final de la SVM para el sistema de 31 y 33 Hz.	55
4.14. Mano robótica de código abierto [32].	56
4.15. Mano robótica virtual simplificada.	57
4.16. Apartado <i>Accelerometer</i> de la GUI.	59
4.17. Selección del apartado de comunicación.	59
4.18. Inicio de la comunicación LSL.	60
4.19. Exportar modelo de Adams a Matlab.	60
4.20. Configuración de parámetros de la planta a exportar.	61
4.21. Diagrama de flujo del script de control en tiempo real.	62
5.1. Matriz de confusión de validación de la SVM seleccionada.	64
5.2. Matriz de confusión de la BCI.	65
5.3. Posición inicial de la mano robótica virtual.	65
5.4. Posición final de la mano virtual con el movimiento de saludo.	66
5.5. Organización de la matriz de datos para simulación.	66
5.6. Gráficas de movimiento del dedo pulgar.	67
5.7. Gráficas de movimiento del dedo índice.	68
5.8. Gráficas de movimiento del dedo medio.	68
5.9. Gráficas de movimiento del dedo anular.	69
5.10. Gráficas de movimiento del dedo meñique.	69
5.11. Posición final de la mano virtual con el movimiento de pinza fina.	70
5.12. Gráfica de movimiento falange proximal	70
5.13. Gráficas de movimiento del dedo pulgar.	71
5.14. Gráficas de movimiento del dedo índice.	72
5.15. Gráficas de movimiento del dedo medio.	72
5.16. Prueba de integración de todos los subsistemas.	73

Índice de Tablas

1.1. Resumen del entrenamiento. Tarea Mental (MT), Tipo de Algoritmo (TA), Presición de Clasificación (ACM), Error Cuadrático Medio (MSE) y Valor de Época[10].	7
4.1. Resumen de los modelos de clasificación entrenados	53
4.2. Variables de estado de movimiento	57
4.3. Variables de estado de medición	58
5.1. Resumen de rotaciones de cada falange para cada movimiento	73

Capítulo 1

Introducción

En los últimos años se han buscado más y mejores formas de comunicación entre los seres humanos y los diversos dispositivos tecnológicos con los que interactúan en sus actividades cotidianas, debido a que los métodos comunes de interacción como el teclado, el mouse o inclusive los comandos de voz pueden ya no ser suficientes. Algunas de las investigaciones más recientes sugieren la interacción con las máquinas o computadoras mediante la activación de los músculos del cuerpo humano o a través de la interpretación de las intenciones generadas como tareas mentales. Esto último se logra al adquirir y comprender las señales generadas por el cerebro humano.

Existen diferentes técnicas que permiten detectar la actividad en las neuronas. Estas técnicas pueden ser invasivas como la electrocorticografía (ECoG) o no invasivas como lo son la magnetoncefalografía, imagen por resonancia magnética funcional (fMRI), la espectroscopia funcional del infrarrojo cercano (fNIR) o la electroencefalografía (EEG). Cada una de estas técnicas permite observar la actividad de las neuronas durante alguna tarea proporcionando diversos datos útiles para su interpretación médica sin embargo, debido a su facilidad de implementación y coste, la técnica más utilizada para el registro de la actividad cerebral es la electroencefalografía. La EEG es una técnica de neuroimagen que se encarga de obtener información electrofisiológica del cerebro, utiliza diferentes electrodos secos localizados en el cuero cabelludo en áreas de interés. Las señales EEG adquiridas reflejan la suma de los voltajes generada por la interacción de las neuronas orientadas radialmente al cuero cabelludo, este voltaje se ve transmitido hacia el cráneo hasta llegar a la superficie de la cabeza [1].

Se han desarrollado sistemas que permiten utilizar la actividad eléctrica de la corteza cerebral para convertir las señales en salidas que permitan remplazar, restaurar, aumentar, complementar o mejorar las capacidades tanto físicas como cognitivas del ser humano, a estos sistemas se les conoce como Interfaces Cerebro-Computadora (BCI, por sus siglas en inglés).

Un sistema BCI adquiere bioseñales generadas en el cerebro para posteriormente extraer sus características con algún algoritmo apropiado, las cuales son enviadas a un sistema que permita interpretar dichas características para generar señales de control adecuadas hacia algún tipo de dispositivo que permita interactuar con el ambiente y que, al mismo tiempo,

genere una retroalimentación hacia el usuario a fin de mejorar la interacción con el sistema [2].

Entre las aplicaciones más comunes que tienen las BCI en el ámbito de la robótica destacan:

- ✓ Rehabilitación de miembros en personas con problemas de movilidad.
- ✓ Recuperación de capacidades motrices en personas discapacitadas.
- ✓ Teleoperación de robots.
- ✓ Aumento de las capacidades físicas y cognitivas del ser humano.

1.1. Antecedentes

1.1.1. Deletreadores

La comunicación juega un papel importante en la sociedad pues con ella se pueden desarrollar ideas, aportar e intercambiar puntos de vista. Sin embargo, existen personas que poseen algún tipo de discapacidad que limita e incluso imposibilita su comunicación con el mundo exterior. Las BCI, a través de los deletreadores, han supuesto una mejora e incluso un medio para que personas con algún impedimento para la comunicación pueda hacerlo de manera efectiva y sin problemas. Los deletreadores o "*spellers*" son dispositivos que traducen las señales cerebrales y las convierten en letras que posteriormente conforman palabras para comunicar a las personas discapacitadas.

Un ejemplo de estos deletreadores es el propuesto por Zhimin Lin *et al.* en [3] el cual utiliza la señal P300, una respuesta involuntaria que se genera en el cerebro 300 ms después de un estímulo específico. En este trabajo se utiliza el paradigma de triple Presentación Visual en Serie Rápida (RSVP, por sus siglas en inglés) en el cual, en cada intervalo de tiempo se le presenta al usuario una matriz de caracteres en este caso, 3 caracteres por intervalo y cada caracter es presentado 3 veces al usuario en diferentes matrices para aumentar la tasa de transferencia de información (ITR), en la Figura 1.1 se muestra la configuración de los estímulos. Los resultados demostraron ser eficientes para poder ser utilizados en pantallas de baja resolución pues los principales problemas con este tipo de deletreadores se basan en el tamaño de las matrices las cuales deben ser considerables para poder evocar el estímulo en el cerebro de manera correcta, por otra parte, la propuesta de los autores tiene características independientes del espacio pues no presentaban el caracter en la misma posición sino se basaban en las matrices que se seleccionaron para hacer la clasificación del caracter.

Lei Cao *et al.* [4] utilizan el paradigma de Imaginación Motora (MI) para desarrollar un deletreador síncrono. En este trabajo utilizaron un deletreador octogonal compuesto por 26 letras, 10 dígitos y 6 símbolos divididos en los ocho bloques que componen el deletreador. Una vez que el usuario ha seleccionado el grupo de su interés, se desplegará una segunda pantalla la cual mostrará los caracteres por separado en donde el usuario pueda seleccionar el de su

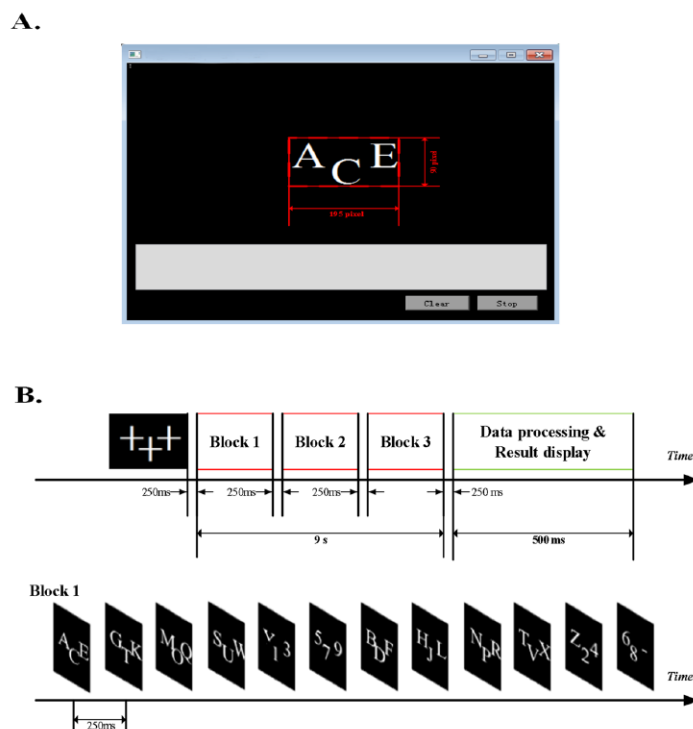


Figura 1.1: Estímulos presentados en el triple RSVP [3]

interés. La finalidad del trabajo es proponer al paradigma de MI síncrona como un método de control eficiente comparado con su homólogo asíncrono pues durante el experimento se demostró que no había diferencias significativas entre ambos paradigmas, además de obtener una precisión mayor al 70 % lo que verifica la factibilidad de estos paradigmas de control.

Por otra parte, en [5] utilizan como señales de control las señales EEG de ojos cerrados y doble parpadeo, las cuales controlan la acción de seleccionar y de deshacer respectivamente, para desarrollar un deletreador de alta tasa de deletreo, para ello se utilizó un teclado virtual de 26 caracteres mas un símbolo especial y un clasificador de máquinas de soporte vectorial de 3 clases para diferenciar entre los estados de inactividad, ojos cerrados y doble parpadeo. Los resultados obtenidos en este trabajo fueron satisfactorios pues se logró una precisión de la clasificación del 92.3 % con una tasa de deletreo media de 5 letras/min lo que permite demostrar su utilidad y factibilidad en aplicaciones reales pues utiliza señales más naturales y se puede dar una comunicación mucho más fluida y continua.

A pesar de que los deletreadores permiten la comunicación del usuario con el mundo exterior, muchos de estos deletreadores son incapaces de detectar el estado de control del paciente pues estos están diseñados para ser síncronos, utilizan tiempos de activación establecidos, o asíncronos, están a la espera de alguna clasificación correcta para poder actuar sin importar un intervalo de tiempo, lo que crea un problema para la experiencia del usuario pues puede darse el caso de tomar las señales cerebrales de manera aleatoria y ejecutar comandos sin control al no poder diferenciar entre el estado de control y uno de no control. Por lo anterior,



Figura 1.2: Figura estereoscópica resuelta [7]

Nagel y Spüler en [6], proponen un deletreador con detección robusta de estado sin control el cual integra los 3 aspectos fundamentales para un deletreador factible en la vida real: alta velocidad de comunicación, control asíncrono y detección de estado sin control. Los resultados fueron satisfactorios pues lograron un ITR de 122.7 bit/min con un error de detección de estado sin control del 0.075 errores/min por lo que se propone también como un método flexible no solo para el control de deletreadores sino también para el control de mouse y teclado en un ordenador.

1.1.2. Videojuegos

A pesar de que las BCI inicialmente fueron concebidas como una forma de neurorrehabilitación así como también para desarrollar neuroprótesis, hoy en día estas interfaces no solo se utilizan como medio de comunicación y rehabilitación en pacientes con serias discapacidades físicas sino también como un medio de investigación en personas sanas a través de videojuegos que utilizan estas interfaces como medio de control. McMahon y Schukat [7] propusieron un videojuego de realidad virtual de bajo costo basado en neurorrehabilitación con el cual plantean el movimiento de un objeto 3D para completar una figura preestablecida, como se muestra en la Figura 1.2, a través de los cambios de Sincronización y Desincronización Relacionados con Eventos (ERS/ERD, por sus siglas en inglés) dentro de la corteza motora en el cerebro. Este dispositivo fue probado en línea durante el hackaton “Hack the Brain” en la Galería de Ciencias en Dublín, donde demostró ser un dispositivo factible para la implementación de nuevas técnicas de neurorrehabilitación pues la experiencia de cada uno de los usuarios así como su desempeño, fue satisfactorio. Cabe resaltar que el objetivo de este trabajo está centrado en la interacción de las personas con los dispositivos BCI así como la respuesta de estas con el sistema.

Por su parte, Rosca y Leba en [8] proponen un videojuego de billar mental utilizando un sistema BCI para el control de fuerza y velocidad de la bola blanca. Para realizar esto último, detectaron dos estados mentales correspondientes al estado neutral, que fue utilizado como referencia para distinguir entre el estado de reposo y de pensamiento activo, y el estado de



Figura 1.3: GUI desarrollado para retroalimentación [8]

“empuje” en el cual el usuario debe imaginar el movimiento hacia adelante de la bola blanca. La velocidad y fuerza que se le imprime a la bola blanca, dependerá del estado cognitivo del usuario al realizar la tarea mental. Para controlar la dirección de la bola blanca, hicieron uso del giroscopio integrado en el casco de encefalografía. Para el desarrollo del videojuego, los autores utilizaron el software Unity3D Gaming Engine el cual es un motor de juego multiplataforma de código abierto desarrollado por Unity Technologies que permite el desarrollo de aplicaciones 2-D y 3-D para diversas plataformas, en la Figura 1.3 se puede observar el entorno desarrollado para la retroalimentación visual. Los resultados obtenidos muestran que la fuerza ejercida a la bola está directamente relacionada con el nivel de atención y concentración que debe ser mantenida a lo largo de la carga mental. Además, los autores concluyen que este videojuego puede ser utilizado como un medio de recuperación para personas que han sufrido accidentes cerebrovasculares, personas con daño cerebral o para el tratamiento de niños con Trastorno por Déficit de Atención con Hiperactividad por lo que el dispositivo BCI se convierte en un medio de asistencia útil.

Filiz y Arslan en [9] proponen un videojuego BCI utilizando el paradigma de Potenciales Visuales Evocados en Estado Estable (SSVEP, por sus siglas en inglés) para realizar el control de movimiento de un avatar. El videojuego consta de un avatar, un cebo y cuatro estímulos visuales a 6 Hz, 6.67 Hz, 8.57 Hz y 10 Hz los cuales representan los movimientos arriba, abajo, izquierda y derecha que puede realizar el avatar (Figura 1.4). La meta a lograr en el videojuego es “comer” el cebo que aparece de manera aleatoria a 45 píxeles del avatar de tal manera que sea necesario utilizar cada uno de los comandos disponibles. Para realizar la clasificación de los comandos, los autores utilizaron el Análisis de Correlación Canónica (CCA, por sus siglas en inglés) con un umbral previamente definido como 0.3. Los resultados del experimento se reportaron en dos aspectos, precisión y tiempo empleado. Para la precisión, se tomó en cuenta la razón de clasificaciones correctas de entre todas las clasificaciones obteniendo un promedio de 66.72 % siendo 75.64 % la mayor razón conseguida. Para la evaluación de tiempo,

se consideró el tiempo entre la aparición del cebo y ser comido por el avatar, siendo 1.42 segundos el mejor tiempo obtenido con un promedio de 8.57 segundos para completar la tarea.

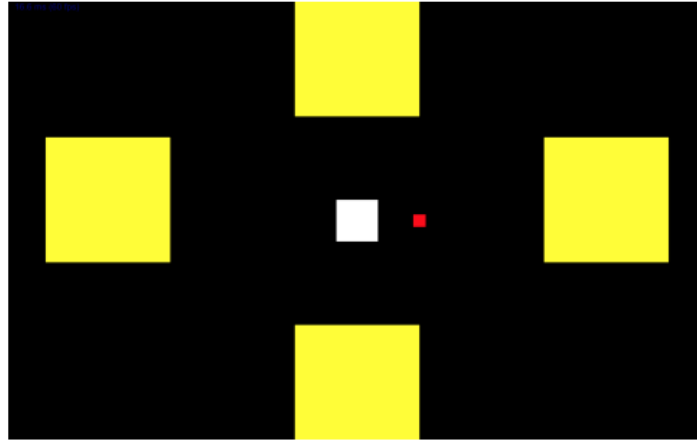


Figura 1.4: Juego desarrollado por Filiz y Arslan. Los cuadros amarillos son los estímulos, el cuadro blanco el avatar y el cuadro rojo representa el cebo a alcanzar [9].

1.1.3. Prototipos móviles

Con el fin de interactuar con el mundo físico, algunas de las investigaciones tienen por objetivo involucrar las interfaces BCI con los robots. Una de las aplicaciones robóticas está dada en [10] donde los autores utilizan el paradigma MI para controlar dos manos robóticas a través del entrenamiento de redes neuronales con los métodos de Gradiente Conjugado Escalado, Levenberg-Mcquardt y Regularización Bayesiana. Se utilizaron los parámetros de Hjort como método de extracción de características con un total de 16 características para 3 señales EEG adquiridas en las posiciones C3, C4 y FCz de acuerdo al estándar internacional 10-20. En el entrenamiento de las redes neuronales, se buscó identificar cuatro estados mentales, cierre de ambas manos, cierre de mano derecha, cierre de mano izquierda y descanso, tanto para el movimiento real como para la imaginación motora de estas actividades. Una vez realizado el entrenamiento de las redes neuronales para el movimiento real y la imaginación motora, se obtuvo que el algoritmo de Gradiente Conjugado Escalado alcanzaba el Error Cuadrático Medio (MSE, por sus siglas en inglés) con una menor cantidad de épocas en el entrenamiento (Tabla 1.1). Además, se encontró que los métodos de Regularización Bayesiana y Levenberg-Mcquart ofrecen una clasificación más precisa pero utilizan más tiempo que el método de Gradiente Conjugado Escalado por lo que para grandes bases de datos, este último método entrena a la red neuronal de manera más eficaz sin sacrificar la precisión.

Por su parte, Yang *et al.* utilizan el paradigma de SSVEP para controlar un brazo robótico retroalimentado con visión artificial [11]. Para la interfaz BCI basada en el paradigma SSVEP, se utilizaron las señales EEG obtenidas de los electrodos colocados en O1, O2, Oz, P3, P4 and Pz según el estándar internacional 10-20, utilizando estímulos visuales localizados en

Tabla 1.1: Resumen del entrenamiento. Tarea Mental (MT), Tipo de Algoritmo (TA), Presición de Clasificación (ACM), Error Cuadrático Medio (MSE) y Valor de Época[10].

Sujeto #	MT	TA	ACM	MSE	Epoch value
1	AM	SCG	99.90 %	2.79E-03	187
		BR	98.30 %	0.202	1000
		LM	99.90 %	0.191	1000
1	IM	SCG	99.50 %	4.17E-03	236
		BR	97.70 %	0.199	1000
		LM	98.30 %	0.194	1000
2	AM	SCG	100.00 %	1.31E-06	297
		BR	100.00 %	0.19	1000
		LM	100.00 %	0.19	1000
2	IM	SCG	99.70 %	3.31E-03	116
		BR	99.40 %	0.19	1000
		LM	99.40 %	0.189	1000
3	AM	SCG	94.60 %	4.32E-02	88
		BR	96.60 %	0.2	1000
		LM	98.10 %	0.194	573
3	IM	SCG	97.30 %	1.98E-02	140
		BR	99.60 %	0.189	1000
		LM	99.20 %	0.19	980
4	AM	SCG	99.50 %	2.83E-03	166
		BR	99.40 %	0.19	1000
		LM	97.80 %	0.195	1000
4	IM	SCG	100.00 %	4.59E-05	459
		BR	97.70 %	0.199	1000
		LM	99.40 %	0.2	1000
5	AM	SCG	100.00 %	3.41E-07	153
		BR	74.90 %	0.24	1000
		LM	100.00 %	0.187	997
5	IM	SCG	99.70 %	4.25E-03	105
		BR	100.00 %	0.188	1000
		LM	99.80 %	0.189	1000
6	AM	SCG	100.00 %	1.06E-07	109
		BR	100.00 %	0.188	1000
		LM	100.00 %	0.188	1000
6	IM	SCG	100.00 %	1.94E-05	191
		BR	100.00 %	0.188	1000
		LM	99.50 %	0.19	1000
7	AM	SCG	100.00 %	3.18E-05	335
		BR	99.70 %	0.189	1000
		LM	99.20 %	0.192	999
7	IM	SCG	98.80 %	9.57E-03	257
		BR	97.50 %	0.196	1000
		LM	98.40 %	0.194	1000
8	AM	SCG	93.90 %	4.32E-02	128
		BR	99.10 %	0.192	1000
		LM	99.40 %	0.191	988
8	IM	SCG	99.10 %	7.27E-03	124
		BR	96.80 %	0.197	1000
		LM	99.50 %	0.189	1000

las frecuencias de 10, 12 y 15 Hz. Para la extracción de características y posterior clasificación de las señales, los autores utilizaron la Transformada Rápida de Fourier (FFT) y CCA respectivamente. El experimento consistía en ubicar al sujeto frente a un monitor con un conjunto de objetos desplegados sobre los cuales se colocaba un diamante parpadeante a las frecuencias antes mencionadas, se generaba aleatoriamente un orden de selección de los objetos y se le indicaba al sujeto que generara el mismo orden de selección a través del dispositivo BCI. Finalmente, en las pruebas en línea, se invitaron a dos sujetos de prueba donde uno de ellos había tenido experiencia con el paradigma SSVEP mientras que el otro no poseía experiencia alguna por lo cual el sujeto con experiencia logró completar las 5 tareas propuestas mientras que el sujeto sin experiencia falló en la primera tarea y se le tuvo que explicar brevemente el experimento, logró completar las 4 tareas restantes sin dificultad. Estos resultados permitieron determinar que el paradigma SSVEP resulta fácil de aprender y requiere poca capacitación lo que permitiría su utilización en dispositivos de asistencia para personas con discapacidades sin la necesidad de un entrenamiento intenso para su utilización.

Millan *et al.*[12] utilizan el análisis asíncrono de señales EEG para detectar tres estados mentales con la finalidad de lograr la navegación de un robot móvil en una habitación, Figura 1.5. Para esto, los autores hicieron uso de 8 canales EEG ubicados en F3, F4, C3, Cz, C4, P3, Pz, y P4 para realizar un estimado del espectro de potencia en la banda de 8-30 Hz a través del algoritmo del Periodograma de Welch en 3 ventanas de 0.5 segundos con un solapamiento del 50 % para posteriormente utilizar el resultado como entrada para un clasificador estadístico Gaussiano integrado en el robot que permite identificar cada uno de los estados mentales. El experimento consistió en dos tipos de entrenamientos previos que permitieron a los sujetos adaptarse a los comandos que generan cada uno de los estados mentales así como optimizar el clasificador Gaussiano para cada uno de los sujetos. El primer entrenamiento consistía en generar cada uno de los estados de manera aleatoria de tal forma que cada uno de los sujetos se fuera familiarizando con la generación de los comandos mentales de manera eficaz, en el segundo entrenamiento, los sujetos debían controlar el robot móvil a través de los comandos mentales de tal manera que los usuarios pudieran ligar cada comando mental con los movimientos del robot. Posteriormente, se realizaron pruebas de funcionamiento del sistema en su totalidad en donde se le pedía al usuario pasar de un punto inicial a un cuarto meta, pasando a través de algún cuarto intermedio de manera aleatoria. Para determinar el rendimiento del sistema, en primer lugar se realizó una prueba mediante comandos mentales para posteriormente realizar el mismo experimento a través de comandos manuales a fin de obtener una comparativa en cuanto a rendimiento. Los resultados arrojaron que el control mental posee un desempeño menor comparado con el control manual sin embargo, este sistema de comunicación podría ser escalable para permitir el control de sillas de ruedas y prótesis en personas con discapacidad pues se logra un desempeño del 83 % en relación con el control manual en el mejor de los casos.

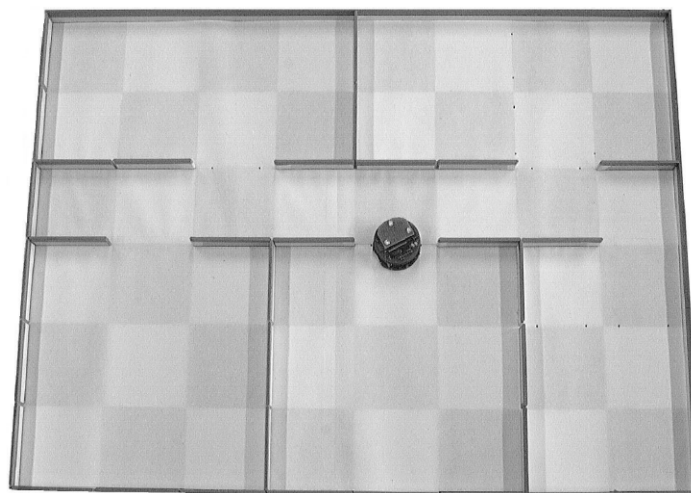


Figura 1.5: Robot móvil operado mediante BCI [12]

1.2. Planteamiento del problema

De acuerdo con el Censo de Población y Vivienda 2020 del INEGI, en México el 4.9 % de la población total tienen algún tipo de discapacidad. Del total de personas con discapacidad, el 48 % padece algún tipo de discapacidad motriz lo que le impide caminar, subir, bajar o desplazarse en general [13], ya sea por factores musculoesqueléticos y/o neuromotrices. Para estas personas, el realizar actividades cotidianas como encender o apagar las luces, abrir o cerrar puertas e incluso tomar algún objeto resulta ser un reto por lo que para superarlo, requieren de dispositivos especiales, rehabilitación intensiva o asistencia de otras personas. En estas situaciones, dispositivos que permitan recuperar o sustituir las capacidades motrices de las que carece la persona, puede ser de gran utilidad pues mejora la calidad de vida al otorgarle mayor autonomía sobre sus acciones.

Debido a lo anterior, se han desarrollado interfaces de asistencia para las personas con discapacidades entre los que se encuentran los sistemas BCI, las cuales utilizan bioseñales producidas por la actividad eléctrica del cerebro mediante sensores ubicados en áreas específicas para traducirlas en señales de control ya sea para realizar el control de algún dispositivo en especial (sillas de ruedas, brazos articulados, carros móviles, etcétera) o para realizar la selección de algún carácter o acción (deletreadores y videojuegos en general). Gran parte de las BCI utilizan electrodos superficiales secos ubicados en el cuero cabelludo del paciente en áreas específicas pues de esta manera se evitan las complicaciones de métodos más invasivos como los electrodos intracraneales que proveen una mejor señal y mayor confiabilidad en ésta, pero que requieren de mayores cuidados y de una intervención quirúrgica para colocarse, por lo que el coste de estos sensores se eleva en demasía además, siendo más asequibles los sensores superficiales pues estos pueden ser colocados en cualquier paciente tan solo asegurando que se ubiquen en las mismas áreas de interés.

En este proyecto se propone el desarrollo de un sistema BCI para el control de una

mano robótica virtual mediante la adquisición de las señales a través de EEG para generar acciones específicas. Para el control de la mano robótica virtual, se implementará una interfaz de conexión que permita la comunicación bidireccional entre la mano robótica y el sistema de adquisición de señales así como contar con un sistema de clasificación que permita su implementación en tiempo real.

1.3. Justificación

A través de la integración de la interfaz con la prótesis de mano, el usuario podrá realizar diversas tareas, lo que le permitirá ampliar su rango de movilidad y autonomía en la realización de actividades cotidianas.

Dentro de la BCI, se podrán implementar diferentes paradigmas para realizar la adquisición de datos lo que facilitará las investigaciones de diversos algoritmos de control y procesamiento de datos.

Las BCI brindan la capacidad de potenciar, sustituir y restaurar las capacidades físicas y mentales del ser humano por lo que son de suma importancia para el área de la robótica siendo además, un precedente dentro de la Universidad Tecnológica de la Mixteca para la expansión en las líneas de investigación de las interfaces humano-máquina.

1.4. Hipótesis

La implementación de un sistema BCI utilizando del paradigma SSVEP y el dispositivo de adquisición de señales EEG OpenBCI de código abierto, permite controlar una mano robótica virtual para generar posturas fijas.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar una interfaz cerebro-computadora para controlar una mano robótica virtual utilizando señales adquiridas por medio del dispositivo EEG OpenBCI de código abierto comercial en conexión con MATLAB.

1.5.2. Objetivos específicos

Para el logro del objetivo general, se formulan los siguientes objetivos específicos:

- ✓ Analizar y seleccionar las técnicas de identificación y clasificación de señales EEG adecuadas al proyecto.
- ✓ Acondicionar el prototipo de mano robótica virtual a utilizar.

- ✓ Definir las tareas a realizar con el prototipo de mano robótica.
- ✓ Desarrollar la interfaz BCI y de comunicación con la mano robótica.
- ✓ Realizar pruebas de funcionamiento on-line.
- ✓ Analizar los resultados obtenidos.

1.6. Propuesta de solución

Para el logro de los objetivos planteados previamente, se propone implementar el sistema que se describe a continuación:

El sistema BCI está integrado por la tarjeta de adquisición Cyton en conjunto con el casco Ultracortex Mark IV, ambos de la empresa OpenBCI, y la mano robótica virtual, que se adecuará para la integración del sistema BCI. La tarjeta de adquisición y el casco, se encargarán adquirir y enviar las señales EEG, una computadora se encargará de recibir las señales EEG provenientes de la tarjeta Cyton, realizará el pre-procesamiento de los datos para, posteriormente, segmentarlos y realizar la extracción de características. Posteriormente, las características obtenidas se enviarán a un algoritmo de clasificación basado en Machine Learning, el cual devolverá una salida acorde a la tarea a realizar. Para la utilización del algoritmo de clasificación, se realizará el entrenamiento a través de una base de datos, generada mediante un procedimiento similar al descrito anteriormente, la cual contendrá las características extraídas de las señales EEG de diversos usuarios.

Una vez obtenida la salida del algoritmo de clasificación, esta será enviada al sistema de control el cual será el encargado de interpretarla para generar la salida de control adecuada hacia la mano robótica para realizar el gesto seleccionado por el usuario. Una vez realizado el gesto deseado, el sistema de la mano robótica virtual enviará una señal al sistema de control para realizar una nueva postura, permitiendo realizar una nueva clasificación. El procedimiento en su totalidad se repetirá hasta que el sistema sea apagado.

1.7. Metodología de desarrollo

Para el desarrollo del presente proyecto se propone la siguiente metodología:

- ✓ Documentación del problema y las necesidades a satisfacer con el prototipo.
 - Investigación bibliográfica de las características del problema.
 - Investigación bibliográfica de soluciones propuestas con enfoque en sistemas BCI.
 - Investigación de los antecedentes en el control de dispositivos móviles.
- ✓ Acondicionamiento de la mano robótica virtual para la comunicación con la interfaz de control.
 - Selección de la mano virtual a utilizar.
 - Configuración de parámetros necesarios para el control.
 - Definición de la interfaz de comunicación con la mano robótica
 - Configuración y programación de la interfaz de comunicación.
- ✓ Adquirir señales EEG mediante el dispositivo OpenBCI.
 - Definición del paradigma a utilizar.
 - Definición del protocolo de adquisición de señales EEG y cantidad de participantes.
 - Creación de la base de datos preliminar.
- ✓ Implementación de algoritmos para la detección y reducción de artefactos en las señales EEG.
 - Evaluación de los distintos algoritmos para la detección y eliminación de artefactos.
- ✓ Implementación de algoritmos para la detección de comandos cerebrales encontrados en la literatura (P300, SSVEP, Imagen Motora).
- ✓ Realización de pruebas y con ello generar un banco de datos que sirva para encontrar el límite de comandos diferentes a detectar y la realización de pruebas off-line.
- ✓ Establecer la comunicación entre el dispositivo OpenBCI y MATLAB para realizar las pruebas on-line.
 - Definición del método de comunicación entre la GUI OpenBCI y MATLAB.
 - Programación del sistema de adquisición de datos en tiempo real.
- ✓ Determinación de las tareas a realizar con la mano robótica, dependiendo del número de comandos.
 - Análisis de las distintas posturas efectuadas por la mano robótica.
 - Selección de las posturas a realizar.

- ✓ Integración del dispositivo EEG con la computadora y la mano robótica para la realización de pruebas.
 - Evaluación del desempeño de la interfaz de comunicación y el sistema de adquisición en tiempo real.
 - Optimización del sistema en general.
- ✓ Realización de pruebas y análisis de resultados finales.
 - Realización de pruebas finales.
 - Evaluación del desempeño final del sistema BCI completo.
 - Reporte de resultados.

1.8. Estructura de la tesis

Este trabajo de tesis está organizado de la siguiente manera: en el Capítulo 1 se presenta una introducción y los antecedentes al tema, se establece el planteamiento y la justificación del problema a abordar, se especifican los objetivos a cumplir y, por último, se describe la metodología a seguir durante el desarrollo del trabajo. En el Capítulo 2 se enlistan características fundamentales de las señales electroencefalográficas, teoría acerca de las técnicas de reducción de la dimensionalidad para conjuntos de datos así como conceptos fundamentales de los clasificadores basados en Machine Learning. En el Capítulo 3 se presenta el diseño del experimento, el protocolo de adquisición de señales a seguir así como el pre-procesamiento de las mismas. El Capítulo 4 muestra el entrenamiento de los sistemas de clasificación a implementar así como el acondicionamiento de la mano robótica virtual para generar las salidas correspondientes y la integración de cada uno de los sistemas desarrollados previamente. En el Capítulo 5 se presentan las pruebas realizadas y los resultados obtenidos durante dichas pruebas, así mismo, se presenta una evaluación tanto del desempeño del sistema BCI, el movimiento de la mano robótica virtual y una evaluación general de todos los sistemas integrados. Finalmente, en el Capítulo 6 se enlistan las conclusiones a las que llegó toda vez se ha realizado el proyecto, de la misma manera se presentan las posibles mejoras y trabajos futuros para la implementación de nuevas funciones y esquemas de trabajo para un mejor desempeño de todo el sistema.

Capítulo 2

Marco Teórico

2.1. El cerebro humano

El cerebro transforma señales de millones de sensores localizados a través de todo el cuerpo en comandos musculares apropiados para generar un comportamiento adecuado a la tarea requerida. La capacidad de procesamiento del cerebro surge de su manera de realizar operaciones de manera paralela y distribuida. El componente principal del cerebro es una célula llamada neurona, un complejo mecanismo electroquímico que recibe información de cientos de otras neuronas, la procesa, y transmite su resultado a cientos de otras neuronas. Además, la conexión entre neuronas es plástica, permitiendo a las redes neuronales adaptarse a nuevas entradas y circunstancias cambiantes [1].

2.1.1. Generación de picos

Cuando la neurona recibe entradas suficientemente fuertes de otras neuronas, una cascada de eventos es provocada: el rápido flujo de iones Na^+ que entran a la célula causan que el potencial de la membrana incremente rápidamente, hasta que la apertura de canales K^+ desencadenan el flujo de iones K^+ en la salida, causando una caída en el potencial de la membrana. Este rápido incremento y caída del potencial de la membrana se conoce como potencial de acción o pico, y representa el modo dominante de comunicación entre una neurona y otra. El pico es un evento de tipo todo o nada con poca o nula información en la forma del pico, en lugar de eso, se piensa que la información es transmitida mediante la tasa de disparos (números de picos por segundo) y/o el tiempo entre los picos. Las neuronas por eso son usualmente modeladas como emisores de salidas digitales 0 o 1 [1].

2.1.2. Organización del cerebro

Las secciones del cerebro humano son principalmente llamadas como cerebro anterior, mesencéfalo y cerebro posterior. Sin embargo, la parte más importante del cerebro en la cual se concentran las interfaces BCI es la corteza cerebral debido a que envuelve al cerebro, donde la mayor cantidad de señales eléctricas son generadas y emanadas [14].

La corteza cerebral se dice que tiene naturaleza contra-lateral debido a que el hemisferio izquierdo del cerebro monitorea el lado derecho del cuerpo y viceversa. La mayoría de las funciones del cuerpo son iniciadas en la corteza cerebral, incluidas la planificación, visión, audición, movimiento, sentido y control.

La corteza cerebral se encuentra particionada en cuatro subsecciones llamadas lóbulos [14]:

- ✓ **Lóbulo frontal:** Relacionada con el razonamiento, la planificación, emociones, juicio, resolución de problemas y funciones motoras. Daños en esta zona pueden generar una notable disminución en las capacidades relacionadas con la personalidad y la conducta.
- ✓ **Lóbulo parietal:** Asociado con la percepción de formas o reconocimiento, percepción de estímulos, sensación, escritura, movimiento y orientación, incluido el control de músculos del torso, pierna, brazo, mano, cadera, cara, lengua y cuerdas vocales. Un trauma en los lóbulos parietales genera a menudo anomalías en imagen del cuerpo y relaciones espaciales.
- ✓ **Lóbulo occipital:** Se encuentra en la parte posterior de la cabeza, está principalmente relacionado con el sistema de procesamiento visual del cerebro. Las personas con un trastorno en este lóbulo pueden percibir imágenes distorsionadas además, puede repercutir en ceguera parcial o total dependiendo de la magnitud del daño.
- ✓ **Lóbulo temporal:** Está asociada con la memoria, interpretación y reconocimiento de estímulos auditivos. Más específicamente, la sección izquierda del lóbulo incorpora la audición, lenguaje y memoria del habla, mientras que el reconocimiento auditivo y musical están asociados a la sección derecha. Afectaciones en esta zona pueden generar sordera total o parcial, trastornos del lenguaje, pérdida de la memoria y de reconocimiento.

2.2. Señales EEG

La electroencefalografía (EEG) es una técnica no invasiva para adquirir señales del cerebro utilizando electrodos colocados en el cuero cabelludo. Los picos o potenciales de acción de las neuronas causan que neurotransmisores sean liberados en las sinapsis, causando potenciales post-sinápticos dentro de las dendritas de las neuronas receptoras. Las señales EEG reflejan la sumatoria de los potenciales post-sinápticos de miles de neuronas que están orientadas radialmente al cuero cabelludo. Las corrientes tangenciales al cuero cabelludos no son detectadas por la EEG, además de las corrientes originadas profundamente dentro del cerebro debido a que los campos de voltaje decaen con el cuadrado de la distancia de la fuente. Por lo cual, la EEG predominantemente captura actividad eléctrica en la corteza cerebral. La resolución espacial de la EEG es típicamente pobre (en el rango de centímetros cuadrados) pero la resolución temporal es buena (en el rango de milisegundos)[1].

La colocación de los electrodos en el cuero cabelludo corresponde al sistema 10-20 internacional aceptado por la Federación Internacional de Neurofisiología Clínica [15]. Este sistema

determina la distancia entre los electrodos de manera porcentual, de ahí el nombre de 10-20 ya que los electrodos se encuentran separados entre sí por un 10 % o 20 % de una distancia determinada como se muestra en la Figura 2.1. La razón principal de la utilización de porcentajes en lugar de longitudes se encuentra en el hecho de que el tamaño del cráneo varía de persona a persona. Además, se utiliza una combinación de letras y números para identificar el lóbulo y la ubicación del hemisferio. Las letras F, T, C, P y O corresponden a los lóbulos frontal, temporal, central, parietal y occipital mientras que los números pares corresponden al hemisferio derecho y los números impares corresponden al hemisferio izquierdo, para hacer referencia a la parte media de la cabeza, se utiliza la letra “z” en cada uno de los lóbulos.

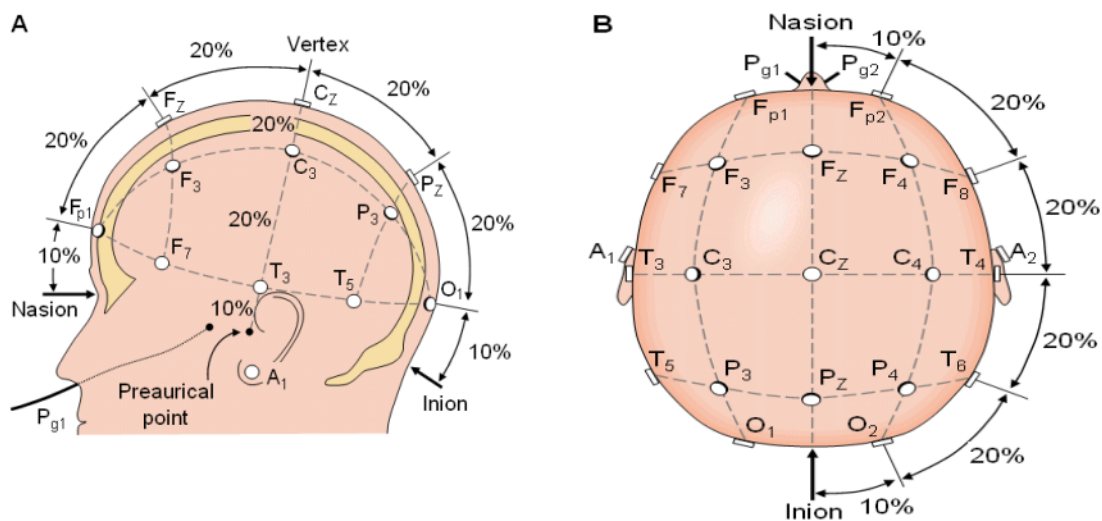


Figura 2.1: Sistema 10-20 internacional [16]

2.2.1. Bandas de frecuencia

Los registros EEG son adecuados para capturar actividad cerebral oscilatoria a una variedad de frecuencias (Figura 2.2). Estas ondas tienen rangos de frecuencia característicos y características espaciales que son frecuentemente correlacionados con diferentes estados funcionales del cerebro.

- ✓ Ondas Delta: Comprenden el rango de 0.5-4 Hz, son detectables en bebés y durante el sueño de onda lenta.
- ✓ Ondas Theta: Comprendidos en el rango de 4-8 Hz, asociadas a estados de somnolencia u “ocio”.
- ✓ Ondas Alpha: Comprendidos entre los 8 y 13 Hz, estas formas de onda pueden ser detectables en la región occipital en personas despiertas cuando estas están relajadas o con los ojos cerrados. Un ritmo particular de estas ondas es el ritmo Mu (8-12 Hz) el cual se presenta en el área sensorimotora y se decrementa o suprime cuando el sujeto imagina o realiza algún movimiento.

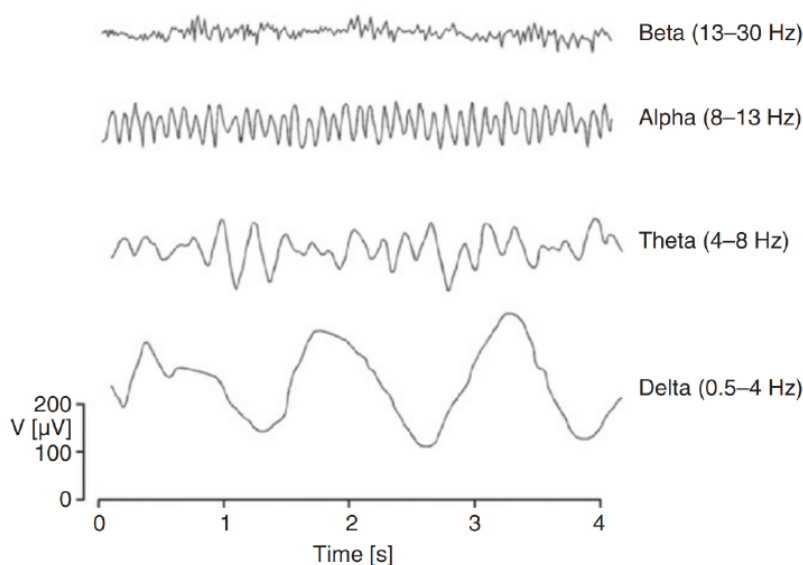


Figura 2.2: Ejemplos de ondas EEG y su banda de frecuencias [1]

- ✓ Ondas Beta: Son ondas comprendidas entre los 13 y 30 Hz, detectables en el lóbulo parietal y frontal. Estas ondas se asocian con el estado de alerta o concentración activa.

2.2.2. Artefactos en EEG

Los artefactos en EEG son cualquier señal indeseable que altera la señal original y no permiten el correcto análisis de la misma. Existen dos tipos de artefactos, los artefactos de origen externo como el ruido debido a la línea de alimentación y los artefactos de origen internos como los causados por el movimiento de ojos y músculos.

Los artefactos de origen externo, particularmente el originado por el ruido de la línea de alimentación, pueden ser reducidos por medio de métodos físicos como la utilización de una jaula de Faraday o por medio de software utilizando técnicas de filtrado [1].

Los artefactos de origen interno, son originados por movimientos involuntarios o voluntarios dentro del cuerpo del sujeto. Los artefactos más comunes son:

- ✓ Artefactos rítmicos debidos a la respiración o latidos
- ✓ Distorsión de la señal debido al cambio de conductancia de la piel debido a la sudoración entre otros motivos.
- ✓ Movimiento de los ojos o parpadeos que aparecen como desviaciones de amplitud en las señales EEG.
- ✓ Artefactos musculares causados por el movimiento de la cabeza, mandíbula, cara, lengua, entre otras partes del cuerpo.

Para reducir la influencia de cada uno de estos artefactos, existen diversas técnicas como el Thresholding el cual rechaza cualquier dato contaminado, el Filtro de Muesca que atenúa los componentes de la señal en una banda de frecuencia específica, la Modelación Lineal en la que se asume que el efecto de los artefactos es aditivo con lo que se puede reconstruir la señal sin artefactos, el Análisis de Componentes Principales (PCA) o el Análisis de Componentes Independientes (ICA).

2.2.3. SSVEP

Los potenciales evocados visualmente (VEP), son modulaciones que se presentan en la corteza visual del cerebro ubicada en el lóbulo occipital del cerebro la cual es la encargada del procesamiento de la información visual. Estos potenciales se generan en respuesta a un estímulo visual repetitivo los cuales pueden variar en cuanto a su frecuencia los cuales pueden ir desde 1 hasta los 100 Hz [17]. Los SSVEP son modulaciones presentadas ante estímulos mayores a los 5 Hz siendo estas fácilmente identificables debido a que la respuesta generada en el lóbulo occipital es una onda sinusoidal con frecuencia fundamental acorde al estímulo presentado lo que las hace atractivas para investigaciones tales como las interfaces cerebro-computador [18].

2.3. Procesamiento de señales

2.3.1. Análisis en el dominio de la frecuencia

La señal registrada mediante EEG captura únicamente las actividades correlacionadas de grandes poblaciones de neuronas tales como la actividad oscilatoria. Debido a la naturaleza oscilatoria de las señales EEG, el análisis en el dominio de la frecuencia es particularmente útil. Algunas de las técnicas aplicables son:

- ✓ Transformada de Fourier
- ✓ Transformada Discreta de Fourier (DFT)
- ✓ Transformada Rápida de Fourier (FFT)

Transformada de Fourier (TF)

La TF es un procedimiento matemático que permite descomponer una función, en este caso una señal, en las diferentes frecuencias que la componen. Esta descomposición es una de las herramientas para el procesamiento de señales fisiológicas más utilizada pues permite caracterizar de manera frecuencial una señal en el tiempo. Las señales EEG son mayormente caracterizadas por medio de la TF puesto que es posible determinar la banda de frecuencias que predomina en la muestra, posteriormente se utiliza dicha información para poder detectar anomalías en los ritmos, realizar la interpretación de actividades cerebrales, entre otros.

La TF se define como una transformación que pasa una señal del tiempo al dominio de la frecuencia, matemáticamente puede describirse como [19]:

$$X(F) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad (2.1)$$

Donde:

- ✓ $x(t)$ es la señal de prueba.
- ✓ f es la variable de la frecuencia.
- ✓ t es la variable de tiempo.
- ✓ $X(F)$ es la función transformada al dominio de la frecuencia.

Transformada Discreta de Fourier (DFT) y Transformada Rápida de Fourier (FFT)

En las aplicaciones BCI, las señales generadas por el cerebro son captadas mediante sensores que realizan un muestreo de la señal en un intervalo de tiempo discreto. La TF entonces debe modificarse para poder realizar la transformación ahora no de una señal continua en el tiempo sino de una señal discreta. Esta modificación da como resultado la DFT cuya entrada será ahora la señal discretizada en el tiempo y cuya salida será la transformación descrita como [19]:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{\frac{-j2\pi kn}{N}} \quad k = 0, 1, 2, \dots, N-1 \quad (2.2)$$

Donde:

- ✓ $x(n)$ es la señal de entrada discreta.
- ✓ k es la variable de la frecuencia.
- ✓ N es el número de muestras.
- ✓ $X(k)$ es la señal de salida en frecuencia discreta.

Por su parte, la FFT es un algoritmo que permite calcular la DFT de una manera más eficaz disminuyendo el coste computacional puesto que realiza la división de la señal en varias señales con el fin de reducir la cantidad de cálculos que se deben realizar.

2.4. Parámetros Hjorth

Los parámetros Hjorth son características estadísticas de una señal variante en el tiempo que permiten determinar características del dominio de la frecuencia [1]. Estos parámetros fueron propuestos por B. Hjorth en 1970 y son también llamados como descriptores normalizados de la pendiente puesto que pueden ser calculados a través de la primera y segunda derivada de la señal. Matemáticamente, los parámetros Hjorth se pueden definir como sigue:

$$A = \text{var}(x(t)) \quad (2.3)$$

$$M = \sqrt{\frac{\text{var}(x'(t))}{\text{var}(x(t))}} \quad (2.4)$$

$$C = \sqrt{\frac{\text{var}(x''(t))}{\text{var}(x(t))}} \quad (2.5)$$

Siendo A (ecuación 2.3) el parámetro conocido como actividad el cual mide la varianza de la señal que está relacionado con la potencia media en el dominio de la frecuencia. El parámetro M (ecuación 2.4) conocido como movilidad, representa la proporción de desviación estándar con respecto al espectro de poder correlacionado con la frecuencia media en el dominio de la frecuencia. Por último se encuentra el parámetro C (ecuación 2.5) el cuál mide la complejidad de la señal relacionado a los cambios de frecuencia, este parámetro compara la señal con una onda sinusoidal convergiendo a 1 cuando la señal es similar.

2.5. Entropía de Shannon

La Entropía de Shannon (SE) fue propuesta por C. E. Shannon como una medida de la cantidad de información contenida en un mensaje dentro del ámbito de la Informática. Estadísticamente, la SE se puede definir como una medida cuantitativa de la incertidumbre asociada a una variable aleatoria [20]. La SE se define como:

$$H(X) = - \sum_{i=1}^M P(x_i) \log_2(P(x_i)) \quad (2.6)$$

Siendo X una variable aleatoria, M el número de estados posibles, $P(x_i)$ la probabilidad del estado x_i y $H(X)$ la entropía asociada a la variable aleatoria X .

2.6. Análisis en el dominio del tiempo

2.6.1. Valor Medio Absoluto (MAV)

El MAV estima el valor medio absoluto a través de la sumatoria de los valores absolutos de cada uno de los puntos de la señal contenidos en un segmento para posteriormente realizar

la división entre el número de elementos que conforman dicho segmento. Matemáticamente se define como:

$$MAV_k = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (2.7)$$

donde:

- ✓ N es el número de muestras del segmento a analizar
- ✓ x_i es la i -ésima muestra del segmento
- ✓ MAV_k es el Valor Medio Absoluto del canal k

2.6.2. Integral Cuadrada Simple (SSI)

La SSI permite calcular la energía de una señal EEG. Matemáticamente se define como.

$$SSI_k = \sum_{i=1}^N (|x_i|)^2 \quad (2.8)$$

donde:

- ✓ N es el número de muestras del segmento a analizar
- ✓ x_i es la i -ésima muestra del segmento
- ✓ SSI_k es la Integral Cuadrada Simple del canal k

2.6.3. Longitud de Forma de Onda (WL)

Como su nombre lo indica, este valor es la suma acumulada de la longitud de onda a lo largo del segmento. Este valor posee información de la amplitud, frecuencia y duración de la onda. La WL se describe como:

$$WL_k = \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (2.9)$$

donde:

- ✓ N es el número de muestras del segmento a analizar
- ✓ x_i es la i -ésima muestra del segmento
- ✓ WL_k es la Longitud de la Forma de Onda del canal k

2.7. Técnicas de reducción de la dimensionalidad

Las técnicas de reducción de la dimensionalidad son herramientas que permiten, como su nombre lo indica, reducir la dimensionalidad de un conjunto de datos para obtener aquellas características o combinación de estas que permitan describir de mejor manera el conjunto, logrando así separar las clases contenidas en este. Dentro de las técnicas de reducción de la dimensionalidad se encuentran las de aprendizaje no supervisado y las de aprendizaje supervisado, siendo las primeras aquellas cuyo objetivo es encontrar los descriptores adecuado sin tomar en cuenta las clases mientras que las de aprendizaje supervisado intentan encontrar descriptores adecuados tomando en cuenta las clases e intentando separarlas entre sí.

2.7.1. Análisis de Componentes Principales (PCA)

PCA es una técnica de reducción lineal cuyo objetivo es simplificar un conjunto de datos reduciendo la cantidad de características necesarias para describir al mismo preservando información importante. Esta técnica es una forma de aprendizaje no supervisado puesto que ignora las clases a las que pertenecen los datos e intenta encontrar las direcciones en la cual la varianza del conjunto de datos es mayor. Esta técnica se basa en la búsqueda de los eigen-vectores y eigen-valores de la matriz de covarianza representando las direcciones de mayor varianza y la cantidad de información que puede ser representada respectivamente [21][22]. La matriz de covarianza se calcula y representa como:

$$Cov(x, y) = \frac{1}{n-1} \sum_i (x_i - \bar{x})(y_i - \bar{y}) \quad (2.10)$$

Suponiendo 2 características, la matriz queda como sigue:

$$C = \begin{pmatrix} cov(x, x) & cov(x, y) \\ cov(y, x) & cov(y, y) \end{pmatrix} \quad (2.11)$$

2.7.2. Análisis Discriminante Lineal (LDA)

El LDA es una técnica de reducción de dimensionalidad cuyo objetivo es simplificar un conjunto de datos manteniendo la separabilidad entre clases. La principal diferencia entre el LDA y el PCA es que el LDA forma parte del aprendizaje supervisado puesto que toma en cuenta las etiquetas de clase para lograr su objetivo. En esta técnica, se intentan encontrar las direcciones (discriminantes lineales) que maximizan la separabilidad entre clase. Este método de reducción de dimensionalidad se caracteriza por utilizar la dispersión intra-clase y la dispersión inter-clase para calcular los ejes de separabilidad [21][22]. Las dispersiones inter-clase e intra-clase se calculan siguiendo las ecuaciones (2.12) y (2.13):

$$S_B = \sum_{c=1}^{|C|} n_c (\bar{x}_c - \bar{x})(\bar{x}_c - \bar{x})^T \quad (2.12)$$

$$S_W = \sum_{c=1}^{|C|} \sum_{x_i \in c} (x_i - \bar{x}_c)(x_i - \bar{x}_c)^T \quad (2.13)$$

donde:

- ✓ $|C|$ es el número de clases.
- ✓ n_c es el número de ejemplos de entrenamiento en la clase c .
- ✓ $\bar{x}_c$ es a media de la clase c .
- ✓ $\bar{x}$ es la media de todos los ejemplos de entrenamiento.

2.8. Técnicas de clasificación

La tarea de un clasificador es asignar etiquetas de clase $y \in Y$ a un vector de características p -dimensional x .

2.8.1. Máquinas de Soporte Vectorial (SVM)

Las SVM son una técnica de clasificación de datos muy útil. Las SVM usan un hiperplano o un conjunto de hiperplanos en un espacio dimensional alto o un espacio dimensional infinito a fin de distinguir entre objetos de diferentes clases. El hiperplano o hiperplanos seleccionados son aquellos que maximizan la separación entre clases pues se sabe que al aumentar los márgenes de separación entre clases aumenta la capacidad de generalización [23]. El rendimiento de las SVM se ve directamente afectado con la elección del kernel pudiendo ser estos lineales, Gaussianos o Función de Base Radial (RBF) [24] por lo que es necesario seleccionar aquel que mejor se ajuste al problema a resolver.

Kernel de las SVM

La función kernel de una SVM es una función que proyecta los datos desde una dimensión baja a un espacio de mayor dimensión con la esperanza que en el nuevo espacio los datos sean separables. Las funciones kernel más conocidas son la Lineal, Función de Base Radial (RBF) y Función Polinomial.

La función de kernel Lineal es la más simple de todas, se describe directamente como el producto punto de los vectores involucrados. La Ecuación 2.14 describe la función de kernel lineal siendo x e y dos puntos de datos en el espacio de entrada. Los puntos también pueden ser vectores de características que describen esos puntos.

$$K(x, y) = x \cdot y^T \quad (2.14)$$

La función de kernel RBF es uno de los más populares debido a que esta función añade una “protuberancia” o “relieve” a cada uno de los datos. Esta función de kernel se usa

principalmente cuando no hay conocimiento previo de los datos. La Ecuación 2.15 describe el comportamiento del kernel.

$$K(x, y) = \exp^{-\sigma \|x-y\|^2} \quad \sigma > 0 \quad (2.15)$$

La función de kernel Polinomial, es una función direccional ya que la salida depende directamente de la dirección de los vectores en el espacio de baja dimensión, esto debido al producto punto utilizado dentro del kernel. La Ecuación 2.16 describe el comportamiento del kernel siendo d el grado de la función.

$$K(x, y) = (1 + x \cdot y^T)^d \quad (2.16)$$

2.8.2. Redes Neuronales Artificiales (ANN)

Las neuronas artificiales son modelos matemáticos que fueron diseñados para “emular” las características del funcionamiento básico de las neuronas biológicas, para esto, utilizan como elemento de procesamiento a la neurona artificial que, al igual que su homónimo biológico, responde a una excitación y comunica su respuesta con las demás neuronas formando así conexiones sinápticas. Siendo así que una red neuronal artificial o ANN (por sus siglas en inglés) está compuesta por varias neuronas artificiales permitiendo crear conexiones que se traducen en límites de decisión no lineales [23].

Funciones de activación

La función de activación de una neurona artificial es una función definida en $a(x)$ y define el comportamiento de la salida de la neurona.

La función sigmoide o logística, Ecuación 2.17, es una de las funciones más utilizada pues está limitada entre 0 y 1 lo que representa la probabilidad de que la neurona se active.

$$f(a) = \frac{1}{1 + e^a} \quad (2.17)$$

La curva característica de esta función de activación se encuentra en la Figura 2.3

La función escalonada (Figura 2.4) es otra de las funciones de activación que pueden utilizarse y cuya función se representa en la Ecuación 2.18, siendo U el umbral de activación deseado.

$$f(a) = \begin{cases} 0 & \text{si } a \leq U \\ 1 & \text{si } a > U \end{cases} \quad (2.18)$$

La función ReLu, una función de activación similar a la función escalonada que únicamente difiere en su límite superior pues esta es limitada por la entrada a como se define en la Ecuación 2.19 donde U es el umbral de activación deseado.

$$f(a) = \begin{cases} 0 & \text{si } a \leq U \\ a & \text{si } a > U \end{cases} \quad (2.19)$$

La curva característica de esta función de activación se encuentra en la Figura 2.5

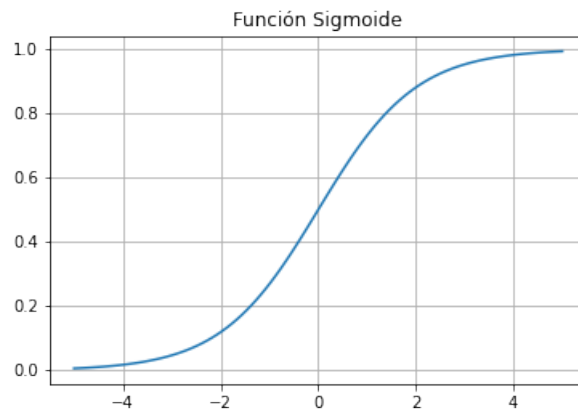


Figura 2.3: Curva característica de activación de la función sigmoide.

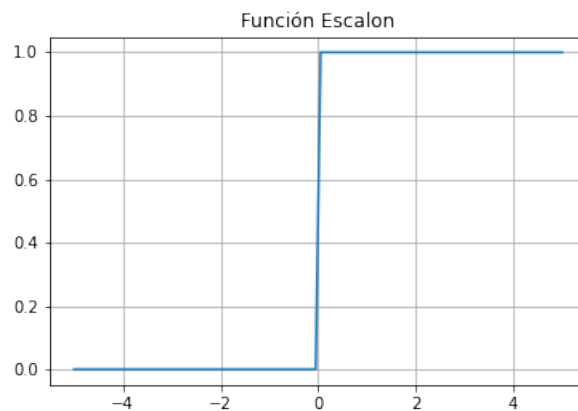


Figura 2.4: Curva característica de activación de la función escalón.

Arquitectura

La arquitectura más conocida de las ANN es la del Perceptrón Multicapa (MLP, por sus siglas en inglés) [23]. La arquitectura base de un MLP está conformada por una capa de entrada (que recibe el vector de características x), una cantidad arbitraria de capas ocultas, una capa de salida y un conjunto de pesos y bias entre cada capa, w y b respectivamente. La representación de la estructura básica una RNA con arquitectura MLP se puede observar en la Figura 2.6.

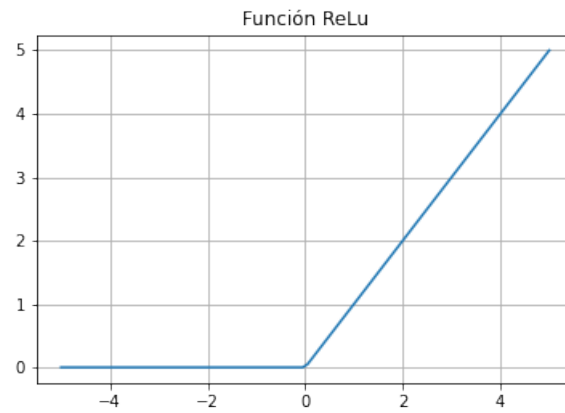


Figura 2.5: Curva característica de activación de la función ReLu.

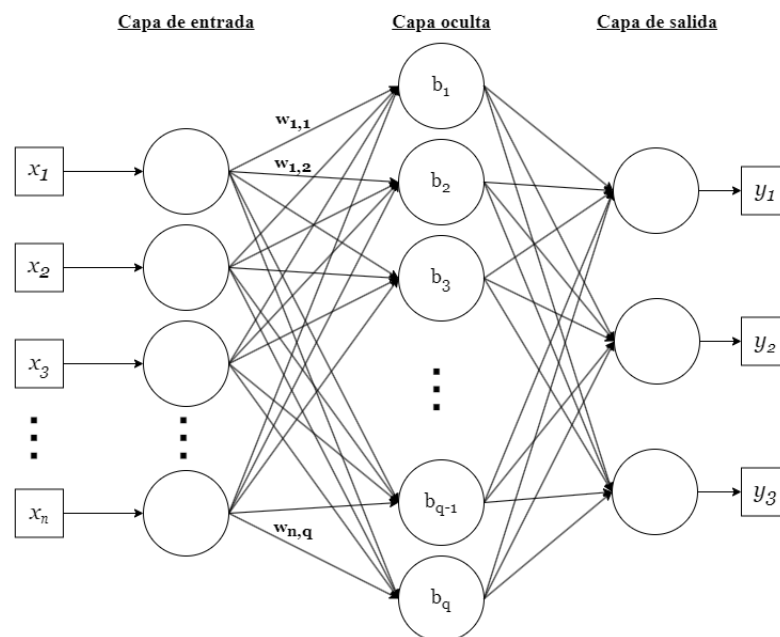


Figura 2.6: Estructura básica de una ANN con arquitectura MLP.

Capítulo 3

Diseño Electroencefalográfico

En este apartado se describen cada una de las partes correspondientes al desarrollo del sistema de adquisición y pre-procesamiento de las señales utilizadas en la BCI. Este apartado se encuentra dividido en 3 partes, la primera describe el diseño de la fuente de estímulos utilizada, en la segunda parte se describe el método utilizado para la adquisición de las señales EEG y, finalmente, se describe la secuencia a seguir para la adquisición y construcción del banco de datos.

3.1. Diseño de la fuente

Dentro de la literatura existen diversos paradigmas que permiten obtener y caracterizar las señales del cerebro para posteriormente ser utilizadas como una señal de control. Como se pudo observar en la Sección 1.1, alguno de los paradigmas más utilizados en la actualidad son el paradigma P300 el cual aprovecha la respuesta involuntaria del cerebro que se genera 300 ms después de que el usuario sea sometido a un estímulo ya sea auditivo, visual o somatosensorial.

El paradigma IM, como su nombre indica, utiliza las señales EEG obtenidas durante el experimento de imaginar el movimiento de alguna de las extremidades ya sea de manera controlada o de manera aleatoria, este paradigma resulta ser de los más complejos pues exige que el usuario tenga un entrenamiento exhaustivo con el sistema para poder adaptarse de mejor manera a él, por lo que aumenta el tiempo necesario para poder utilizar el sistema de manera fiable.

Por último, se encuentra el paradigma SSVEP el cual utiliza estímulos visuales los cuales provocan una respuesta en el cerebro con base en la frecuencia del estímulo ya que en la corteza visual del individuo, se genera una respuesta con la misma frecuencia que el estímulo visualizado siendo fácilmente detectada. Las BCI basadas en este sistema requieren de fuentes de luz que parpadeen a una frecuencia en el rango de los 5-50 Hz puesto que la corteza visual solo puede manejar este rango de frecuencias [17], dichas fuentes de luz pueden ser realizadas ya sea mediante una pantalla LCD en la que se generen patrones de estímulos (cuadros de ajedrez con y sin inversión de patrón, círculos, cuadros simples, entre otros) o mediante luces

LED [25] siendo estos últimos los más sencillos y versátiles pues la simpleza del dispositivo permite controlar tanto la frecuencia como la intensidad del estímulo, de tal manera que pueda contrastar con el entorno, garantizando la generación de los potenciales en estado estable.

Después de una revisión en la literatura y evaluando los pros y contras de los diferentes paradigmas encontrados, se optó por utilizar el paradigma SSVEP por la versatilidad y facilidad de implementación que ofrece, además de que es de los sistemas que requieren el menor entrenamiento para ser usados siendo escalable a diferentes áreas de aplicación. Una vez definido el paradigma a utilizar, se procedió a diseñar la fuente de los estímulos visuales tomando en cuenta algunos puntos importantes de diseño

- ✓ Que fuera reutilizable.
- ✓ Fácil de modificar.
- ✓ De bajo costo.
- ✓ Que posea componentes accesibles.
- ✓ Que sea replicable.

El diseño a implementar se compone de un microcontrolador Atmega328p, una fuente de poder de 5V, el estimulador visual y una fuente de poder regulable. El diseño sigue el diagrama de la Figura 3.1 a fin de generar los estímulos visuales adecuados al experimento.

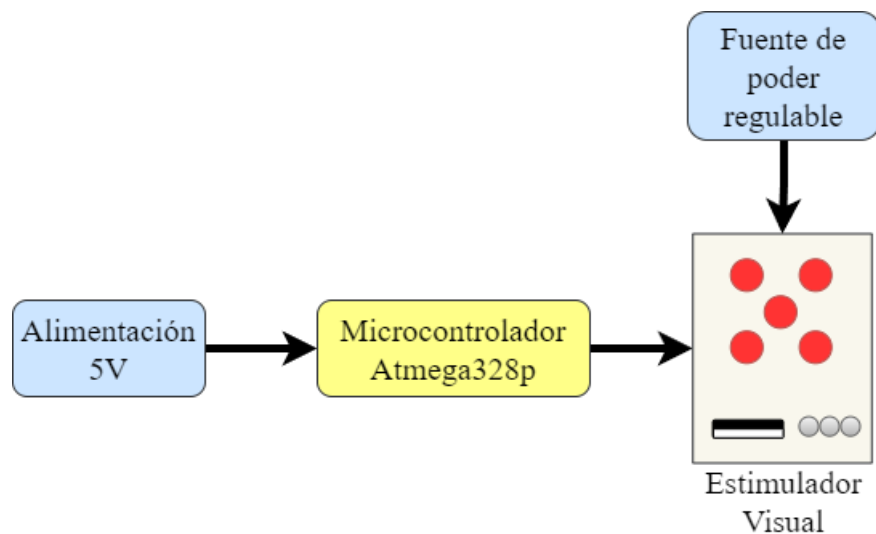


Figura 3.1: Configuración de los estímulos visuales implementados

Atmega328P

El ATmega328P (Figura 3.2) es un microcontrolador de 8 bits de bajo consumo basado en la arquitectura RISC de AVR [26]. Las características principales de este microcontrolador son:

- ✓ 32 Kbytes de memoria Flash programable
- ✓ 1 Kbyte de memoria EEPROM
- ✓ 23 líneas de E/S de propósito general
- ✓ 3 temporizadores/contadores con modos de comparación y Modulación de Ancho de Pulso (PWM)
- ✓ 1 convertidos Analógico-Digital (ADC) de 6 canales y 10 bits
- ✓ 2 pines de interrupción externa por hardware (INT0 e INT1)
- ✓ 23 interrupciones por cambio de pin (PCINT0:14 y PCINT16:23)

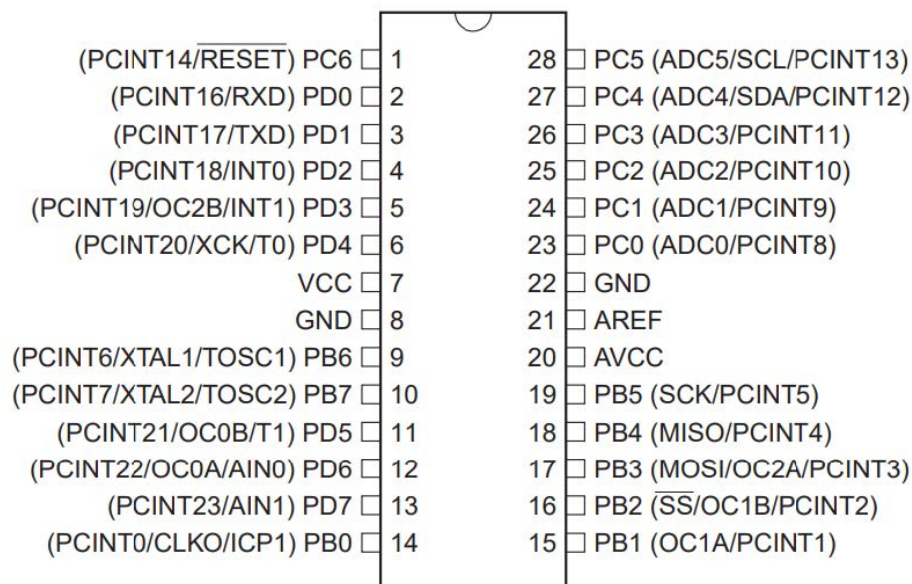


Figura 3.2: Distribución de puertos ATmega328P [26].

Se utilizó un microcontrolador ATmega329P para la generación de la señal PWM que controlará la frecuencia del estimulador visual. La señal PWM generada se realizó mediante la respuesta automática de los temporizadores para asegurar una señal estable y con la frecuencia correcta y sin retrasos. Se implementaron un total de 6 frecuencias en pares dentro del microcontrolador, las cuales fueron 8 Hz, 12 Hz, 13 Hz, 15 Hz, 31 Hz y 33 Hz. Las primeras dos frecuencias se eligieron dado que aquellos estímulos cercanos a 10 Hz (frecuencia fundamental del cerebro) generan una respuesta SSVEP mucho mayor que frecuencias más alejadas [17]. En el caso de 13 y 15 Hz, se tomaron en cuenta dado que a pesar de estar alejadas de la frecuencia fundamental, éstas no se ven tan afectadas por la frecuencia fundamental de 10 Hz permitiendo de esta manera obtener señales de respuesta en el cerebro con una mejor claridad y con una buena potencia en el espectro de la frecuencia. Finalmente, las frecuencias más altas de 31 y 33 Hz fueron seleccionadas para minimizar la influencia de la fatiga visual provocada

por los estímulos y del mismo modo aumentar la comodidad para el usuario puesto que estas frecuencias son menos perceptibles en el parpadeo que las frecuencias antes mencionadas.

Estimulador Visual

Para la construcción del estimulador visual, se optó por utilizar la estimulación mediante luces LED ya que permiten una respuesta SSVEP mayor que con una aplicación de escritorio que se reproduce mediante una pantalla LCD tal como lo demuestra Zhu *et al.* en [27]. En la Figura 3.3 se puede observar el esquema eléctrico utilizado.

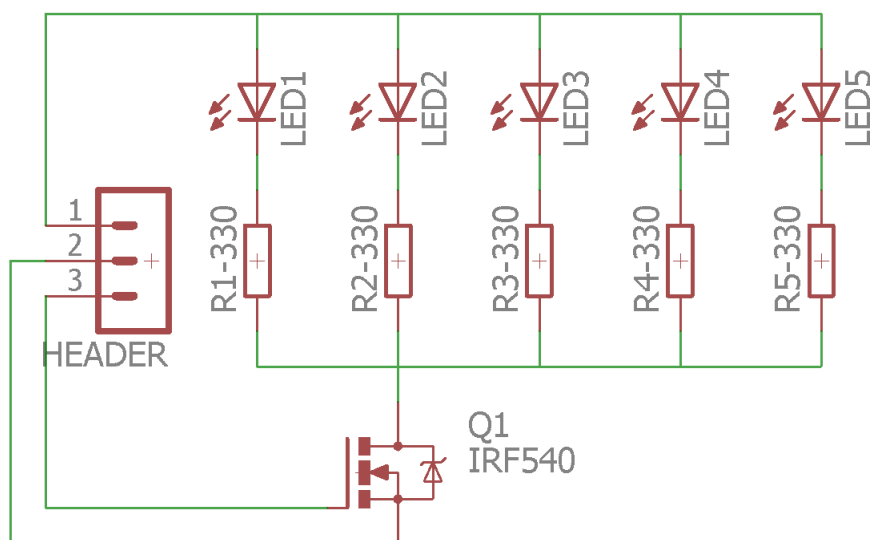


Figura 3.3: Esquema eléctrico del estimulador visual.

Para el estimulador visual, se utilizaron 5 LEDs de color rojo puesto que este color genera una respuesta SSVEP mayor comparado con otros colores [28] además de que permite contrastar de buena manera con la luz ambiente. Para el control de encendido y apagado simultáneo de todos los LEDs, se utilizó un MOSFET como interruptor, en este caso se utiliza el IRF540 pues es un componente altamente comercial y de bajo costo. Para la conexión de alimentación y disparo del MOSFET se utilizó un header de 3 pines en donde irán conectados la alimentación regulable de los LEDs (pines 1 y 2) y el disparo (pin 3). La alimentación de este circuito se propuso variable ya que permitirá variar la intensidad lumínica de los LEDs de tal manera que pueda ser ajustable al entorno donde se utilice. Una vez definido el circuito eléctrico, se realizó la distribución y conexiones para el maquinado de la PCB quedando como se muestra en la Figura 3.4

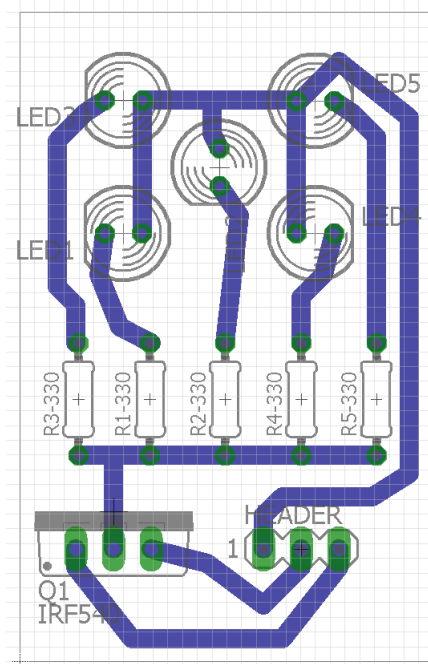


Figura 3.4: Diagrama de conexiones y distribución del estimulador visual.

3.2. Protocolo de adquisición

En este apartado se describen los materiales, métodos y procedimientos a seguir para la adquisición de las señales EEG.

3.2.1. Materiales

A continuación se detallan los materiales a utilizar para la adquisición de las señales EEG.

- ✓ Tarjeta de adquisición Cyton de la empresa OpenBCI
- ✓ Casco UltraCortex Mark IV.
- ✓ 4 electrodos secos superficiales para cuero cabelludo de AgCl colocados en el casco cubriendo la sección de la corteza visual acorde al estándar 10-20.
- ✓ GUI del usuario para la conexión tarjeta-computadora.
- ✓ Estimuladores visuales.

3.2.2. Reclutamiento de participantes

Se reclutaron participantes que cumplieran con los siguientes requisitos de elegibilidad:

- ✓ Edad entre los 18 -25 años.

- ✓ Con buena salud en general que se evaluará a través de una encuesta rápida.
- ✓ Sin historial de episodios epilépticos.
- ✓ Estudiantes de la Universidad Tecnológica de la Mixteca.

Se definieron algunos criterios de exclusión para el experimento, esto con la finalidad de garantizar un buen desempeño del sistema así como delimitar a la población que puede participar en el experimento. Los criterios de exclusión son:

- ✓ Que tengan problemas de la vista tales como ceguera total o parcial, así como una disminución grave de la vista.
- ✓ Con enfermedades oculares como la retinopatía diabética o degeneración macular.
- ✓ Con afecciones del sistema nervioso central tales como la epilepsia o cualquier otra enfermedad convulsiva.
- ✓ Que padezcan esclerosis múltiple o con problemas de Parkinson.
- ✓ Que hayan sufrido de algún ACV o apoplejía.
- ✓ Enfermedades cardiovasculares como la hipertensión arterial, uso de marcapasos entre otros.
- ✓ Que se encuentren en tratamiento con drogas o medicamentos que afecten el sistema nervioso central como los tranquilizantes, ansiolíticos, estimulantes, analgésicos opiáceos.
- ✓ Con problemas de atención o de concentración como el autismo, trastorno de déficit de atención con hiperactividad, entre otras que afecten la capacidad de atención del sujeto.

Los participantes seleccionados fueron personas que cumplieran con los requisitos previamente mencionados y que, además, no poseían discapacidades motrices. Esto último es esencial ya que a pesar de que el proyecto esté enfocado a este sector de la población, se encontraron dificultades para el traslado de las personas al lugar del experimento además que dentro de la comunidad universitaria no se identificó población con discapacidad motriz que cumpliera con los requisitos propuestos.

3.2.3. Métodos

Antes de la realización de cualquier registro, a cada participante se le entregó una forma de consentimiento informado la cual se le pidió leer y firmar. En esta forma, se detallan el número de sesiones, riesgos, beneficios y consideraciones dentro del experimento. Al mismo tiempo, a cada participante se le pidió contestar de manera fiel un cuestionario de identificación demográfica, esto con el fin de obtener datos estadísticos acerca de los participantes. A cada participante se le notificó que puede negarse a la participación garantizando así la participación voluntaria en el experimento sin coacción alguna. Cabe resaltar que tanto el protocolo de adquisición de datos así como las formas otorgadas a cada participante, fueron

revisadas y aprobadas por el Comité de Ética de la carrera de Medicina de la Universidad de la Sierra Sur (UN SIS) esto con el objetivo de garantizar y proteger los derechos de los participantes al tratarse de un experimento en donde se extraen señales electrobiológicas de los participantes (Apéndice A).

La obtención de las señales se realizará en el Laboratorio de Detección y Corrección de Fallas Industriales dentro de los laboratorios de posgrado esto con la intención de tener un ambiente controlado. La necesidad de un ambiente controlado es imperativo ya que en aproximaciones previas del experimento, se detectaron diversas condiciones tanto ambientales como técnicas que se deben cumplir para obtener los mejores resultados. Las condiciones ambientales necesarias para realizar el experimento son: ambiente con iluminación natural sin exposición directa a la luz del sol, ruido bajo, temperatura ambiente agradable de entre 20-25 °C. Las condiciones técnicas incluyen: distancia al estimulador de entre 50-100 cm, silla sin reposabrazos, estimulador colocado justo al centro del campo visual tomando como referencia el LED central del estimulador.

Tomando en cuenta las condiciones anteriormente mencionadas, a continuación se describe el método de adquisición de las señales EEG:

1. Preparación

- ✓ Se realiza la medición de la longitud entre nasion e inion del usuario para determinar el punto Cz.
- ✓ Se coloca el casco de EEG realizando la coincidencia del punto Cz del casco con el punto Cz encontrado.
- ✓ Los electrodos de EEG se ubican en las posiciones Poz, O1, Oz y O2 de acuerdo al estándar internacional 10-20 para cubrir de manera correcta la corteza visual del cerebro ubicado en el lóbulo occipital siendo ésta el área de generación de los SSVEP como se menciona en la Sección 2.2.3.
- ✓ Se verifica el contacto de los electrodos y el cuero cabelludo mediante una inspección visual y a través del indicador de la GUI.
- ✓ Se verifica la calidad de las señales de EEG a través del indicador de impedancias y la detección de la onda alfa, de tal manera que se garantice la calidad de las señales obtenidas.

2. Tarea

- ✓ Visualización del estímulo ubicado a 70 cm del usuario. Se le indica al usuario estar en una posición relajada con las manos en las piernas.
- ✓ Se realizan 4 sesiones con 5 repeticiones por sesión.
- ✓ Cada repetición consta de dos fases: Relajación y Atención. La fase de Relajación consta de 15 segundos mientras que la de Atención consta de 15 segundos. Cada repetición se realiza después de 2 minutos mínimos de descanso para el usuario.

- ✓ Durante toda la adquisición se vigilará al usuario de manera visual para actuar ante cualquier malestar o incomodidad percibida. Así mismo se realizará el registro de movimientos involuntarios y factores externos que pudieran comprometer la adquisición de señales y su interpretación.

3.3. Señales EEG

En este apartado se describe la secuencia a seguir para la adquisición y construcción del banco de datos a utilizar para la BCI. Se incluyen las definiciones de los materiales así como las técnicas aplicadas para procesar los datos obtenidos.

3.3.1. Adquisición

Para la adquisición de las señales EEG, tal como se menciona en la Sección 3.2.1, se utilizará la tarjeta Cyton de la empresa OpenBCI. La tarjeta Cyton es una tarjeta que posee 8 canales para la adquisición de datos y un procesador de 32-bits. Dentro de la tarjeta se implementa un microcontrolador PIC32MX250F128B lo que le confiere una gran memoria así como una velocidad de procesamiento alta. Este dispositivo se comunica de manera inalámbrica mediante un emisor-receptor USB Bluetooth, también le es posible comunicarse con otros dispositivos compatibles con Bluetooth de baja energía (BLE) [29]. Las características más relevantes de la tarjeta se enlistan a continuación:

- ✓ Alimentación con baterías de 3-6 V únicamente
- ✓ Microcontrolador PIC32MX250F128B con cargador de arranque chipKIT UDB32-MX2-DIP
- ✓ ADS1299 Interfaz analógica
- ✓ Acelerómetro de 3 ejes LIS3DH
- ✓ Radio RFduino BLE
- ✓ Ranura para tarjeta micro SD
- ✓ Regulación de voltaje (3.3V, + 2.5V, -2.5V)
- ✓ Frecuencia de muestreo de 250 Hz

Para el montaje de la tarjeta Cyton, se hizo uso del casco Ultracortex Mark IV el cual es un casco de código libre cuyas partes pueden ser impresas mediante impresión 3D. Este casco posee 35 nodos para la conexión de los electrodos superficiales secos, en general, este casco está diseñado para trabajar de manera directa con todos los sistemas de la empresa OpenBCI [30].

Se descargaron los archivos necesarios para la impresión del casco de la página web del fabricante. Se realizó la impresión de las partes y se armó siguiendo las indicaciones provistas

en la página web del fabricante. En la Figura 3.5 se observa el casco impreso.

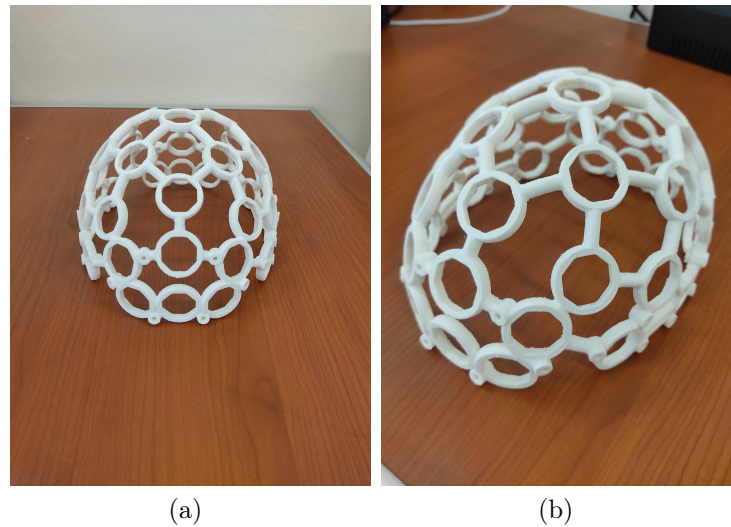


Figura 3.5: Casco Ultracortex impreso

Para el posicionamiento de los electrodos, se utilizó la guía provista por la empresa para la colocación de electrodos siguiendo el estándar 10-20. En la Figura 3.6 se aprecia la localización de los nodos de conexión que posee el casco de acuerdo al estándar 10-20. Siguiendo el esquema, se optó por utilizar 4 nodos de conexión para los electrodos secos de tal manera que se cubrieran puntos importantes de la corteza visual, los electrodos se posicionaron en los puntos O1, Oz, O2 y POz siguiendo el estándar.

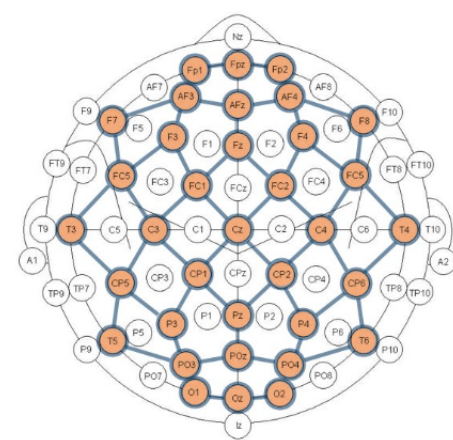


Figura 3.6: Distribución de los nodos de conexión.

En la Figura 3.7 se observa la configuración final del casco. La alimentación de la tarjeta se realizó mediante 4 baterías recargables de 1.2V. Los electrodos de referencia fueron tomados en el lóbulo de la oreja izquierda para el BIAS y el lóbulo de la oreja derecha para la referencia

de los electrodos con el fin de atenuar las alteraciones miográficas.

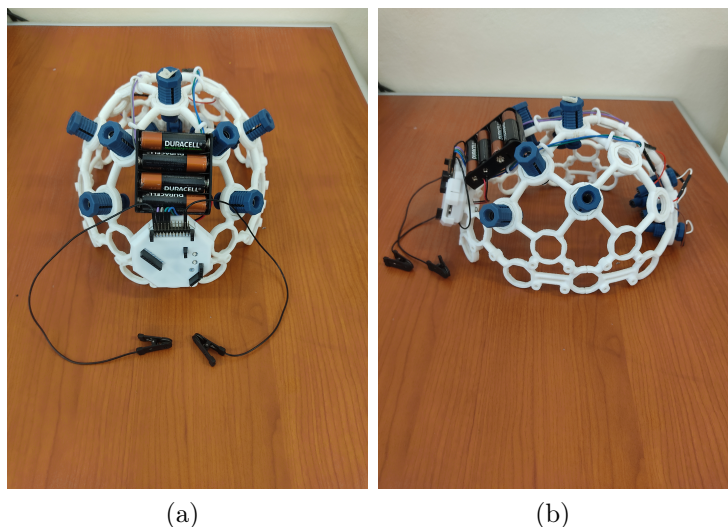


Figura 3.7: Configuración final casco Ultracortex.

Para la recepción de las señales EEG de la tarjeta a la computadora, se utilizó el receptor USB Bluetooth que proporciona el fabricante. La recepción y guardado de las señales se realizaron mediante la interfaz de usuario (GUI) integrado al dispositivo (Figura 3.8). Esta GUI es proporcionada directamente por el fabricante, permite realizar la adquisición y visualización de las señales a través del receptor USB (Figura 3.8b) así como guardarlas en formato original (RAW) en una carpeta con el nombre establecido en la interfaz de configuración (Figura 3.8a).

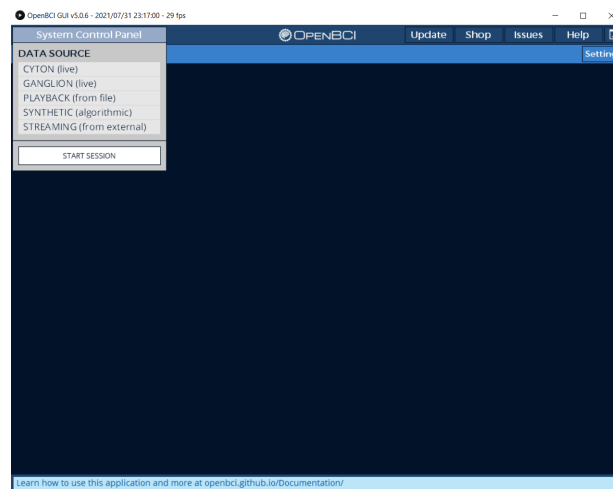
3.3.2. Pre-procesamiento

Para el pre-procesamiento de las señales adquiridas y la construcción del banco de datos, se realizaron dos fases de procesamiento. La primera constó de un pre-procesamiento que implicó adecuar los archivos obtenidos al formato y forma necesarios para la segunda fase en la que se procesa la señal de manera digital para eliminar artefactos y ruido no deseado así como segmentar las señales para obtener las características deseadas.

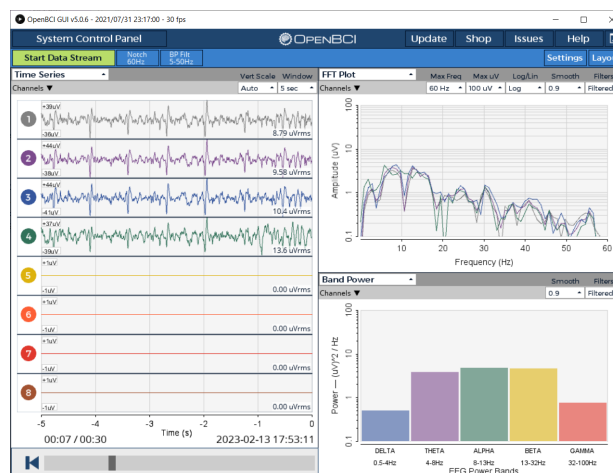
Conversión y adecuación de los archivos

En esta primera fase se realizó la conversión y adecuación de los archivos obtenidos para posteriormente realizar el procesamiento digital. Los pasos a seguir son los siguientes:

1. Conversión de formato: los archivos guardados por la GUI se encuentran en tipo *txt* por lo que es necesario pasarlos a un formato que permita ser manipulado de manera más sencilla. Se optó por utilizar el formato *csv* aprovechando que los datos guardados se encuentran delimitados por comas.



(a) Interfaz de configuración.



(b) Interfaz de funcionamiento.

Figura 3.8: GUI para la adquisición de señales.

2. Renombre del archivo: Como se mencionó en el apartado anterior, los datos obtenidos se guardan en una carpeta con el nombre definido por el usuario sin embargo, los archivos siguen un formato de nombre que no puede modificarse siendo necesario renombrar los archivos, de tal manera que se pueda establecer un orden además de permitir la rápida identificación de los mismos. Se propone utilizar el código *SN_FE_Trial* donde:
 - ✓ N: Identificador numérico del sujeto.
 - ✓ FE: Frecuencia de estimulación.
 - ✓ Trial: Número de prueba de la sesión.
3. Eliminación de encabezados: Los datos guardados por la GUI poseen un encabezado en el que se nombran las columnas así como algunos datos extra, sin embargo, no son útiles

para el procesamiento por lo que se eliminaron las primeras 5 filas correspondientes a los encabezados.

4. División de etapas: En este apartado se realizó la segmentación de las señales para obtenerlas en sus dos etapas (Relajación y Atención), esta segmentación se realizó mediante un script en Matlab y, adicionalmente se realizó una segmentación extra que contiene el punto de cambio entre ambas fases para fines de investigación posteriores.

Filtrado y caracterización

En esta fase se realiza el procesamiento de las señales a fin de mejorar la calidad así como la extracción de características relevantes para el sistema BCI. La secuencia a seguir se describe en la Figura 3.9.

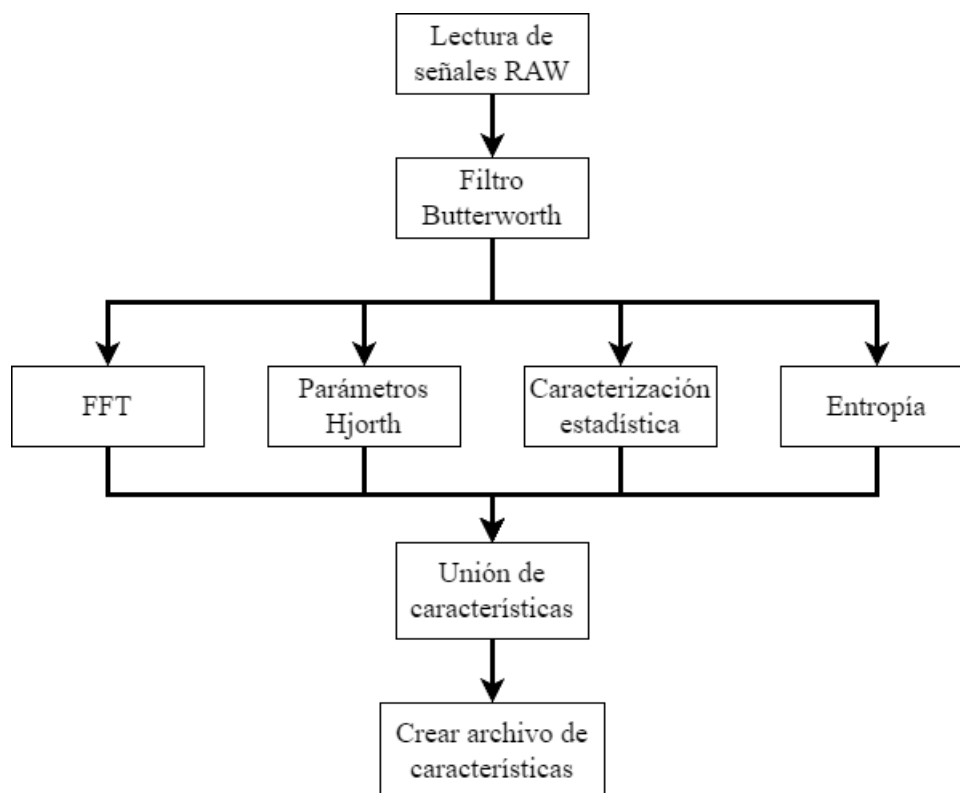


Figura 3.9: Diagrama de procesamiento digital de las señales EEG .

El procesamiento digital de las señales se realiza mediante un script de Matlab. Para la lectura de las señales, se hace uso de la función “*textscan*” puesto que esta función permite leer diferentes tipos de archivos incluido el formato *csv* y requiriendo el nombre, el formato de las columnas y el delimitador para la lectura. Para el filtrado de la señal, se optó por utilizar un filtro pasa bandas con el fin de atenuar las frecuencias innecesarias para la caracterización, de este modo se implementó un filtro pasa bandas tipo Butterworth.

Para el diseño del filtro, se tomaron en cuenta las características que se listan a continuación:

- ✓ Frecuencia de corte 1:
 - Frecuencias de estimulación de 8 y 12 Hz: 5Hz
 - Frecuencias de estimulación de 12 y 15 Hz: 11 Hz
 - Frecuencias de estimulación de 31 y 33 Hz: 26 Hz
- ✓ Frecuencia de corte 2:
 - Frecuencias de estimulación de 8 y 12 Hz: 15Hz
 - Frecuencias de estimulación de 12 y 15 Hz: 17 Hz
 - Frecuencias de estimulación de 31 y 33 Hz: 40 Hz
- ✓ Orden del filtro: 4to orden
- ✓ Frecuencia de muestreo: 250 Hz

De esta manera se obtiene un filtro pasa bandas tipo Butterworth cuya frecuencia central se encuentra justo a la mitad de las frecuencias de interés. Además, la implementación de este tipo de filtros permiten suavizar la señal y mejorar los componentes de las frecuencias contenidas en la señal. La implementación del filtro pasa bandas con frecuencia central en la mitad de las frecuencias de interés, permite eliminar artefactos externos como el ruido eléctrico en los 60 Hz así como la gran mayoría de artefactos mioeléctricos y oculares.

Para el cálculo de características, se realizó una segmentación de la señal a través de ventanas de 3 segundos con una superposición del 80 %. Tomando en cuenta que se tiene una señal de 15 segundos de duración, se obtuvieron en total 20 ventanas de 3 segundos. Para el cálculo de la FFT, se hizo uso de la función “*fft()*” la cual toma como parámetro una señal y entrega la FFT de dicha señal. Para el análisis de la frecuencia fundamental de la señal, se toma únicamente la mitad de la FFT entregada por la función debido a la propiedad de simetría que posee la FFT. Posteriormente, se utiliza la función “*find()*” que devuelve el o los índices en donde la evaluación del argumento sea verdadera, dicho argumento se definió como el máximo de la mitad de la mitad de la FFT. Aplicando este procedimiento a cada uno de los canales de la señal EEG, se obtienen un total de 4 características en frecuencia (una por cada canal) además, se optó también por guardar la potencia de la señal en esa frecuencia obteniendo 4 características extra (una por cada canal). En el caso de los parámetros Hjorth, se siguió el siguiente pseudocódigo:

- ✓ Se define el tamaño del paso para la derivada aproximada.
- ✓ Se calcula la varianza de la señal a través de la función “*var()*” de Matlab
- ✓ Se calcula la derivada aproximada de la señal mediante la función “*diff()*”
- ✓ Se calcula la derivada del resultado anterior

- ✓ Se calculan los parámetros Hjorth de acuerdo a las ecuaciones (2.3), (2.4) y (2.5)

Obteniendo así 3 características por canal, siendo 12 características por ventana. Para la caracterización estadística únicamente se implementan las funciones de acuerdo a las ecuaciones (2.7), (2.8) y (2.9) descritas en el capítulo anterior. Finalmente, para el cálculo de la entropía, en primer lugar se calcula la probabilidad relativa de la señal mediante la función “*histcounts()*”, se eliminan las probabilidades 0 y se cambian por 1, posteriormente se calcula la entropía de la señal utilizando la ecuación (2.6) descrita en el capítulo anterior. Se obtiene una característica por canal siendo 4 características por ventana.

Una vez obtenidas las características deseadas de la señal, estas se unen en una matriz de Nx36 donde cada columna corresponde a una característica y cada fila a una ventana como se observa en la Figura 3.10. Cabe resaltar que este proceso se realizó para todos los archivos obtenidos en un mismo script puesto que los nombres siguen un código y patrón, pudiéndose automatizar para que en una sola ejecución del script se obtuvieran las características de cada uno de los archivos obtenidos. Finalmente, la matriz de características se guarda en el archivo **CaracteristicasFinales.csv**.

	Frecuencia principal en la FFT				Potencia de la frecuencia principal				Parámetros Hjorth												Entropía de Shannon				Longitud de Forma de Onda				Integral Cuadrada Simple				Valor Medio Absoluto			
	C1	C2	C3	C4	C1	C2	C3	C4	ACT: C1-C4				MOV: C1-C4				COM: C1-C4				C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
Ventana 1																																				
Ventana 2																																				
Ventana 3																																				
⋮																																				
Ventana n																																				

Figura 3.10: Matriz de características final.

El script utilizado para el filtrado y caracterización de las señales se encuentra en el Anexo B.

Capítulo 4

Clasificación de Señales e Integración

4.1. Sistema de clasificación

El sistema de clasificación se compone de dos etapas: entrenamiento del sistema y la clasificación en tiempo real. En la primera etapa se realiza la adquisición de las señales EEG a fin de conformar un banco de datos con el cual se pueda entrenar el sistema, posteriormente se realiza la caracterización de las señales obtenidas para finalmente entrenar el sistema con el banco de características generado. La etapa de clasificación en tiempo real, consiste en la adquisición continua de las señales EEG, realizar el preprocesamiento, extracción de características para finalmente ser enviadas al sistema previamente entrenado el cual se encargará de clasificar dicha señal para con ello generar la salida adecuada de movimiento.

Para el sistema de clasificación, se realizó una selección dentro de las referencias consultadas y que tengan una similitud con el proyecto propuesto. Dentro de las referencias se encontraron 2 sistemas de clasificación populares, las SVM y las ANN por lo cual se optó por entrenar y comparar ambas a fin de seleccionar de mejor manera el sistema de clasificación adecuado al proyecto.

Para el entrenamiento de los clasificadores, se desarrollaron 4 fases, entrenamiento con las características totales, entrenamiento con características importantes dentro del SSVEP, entrenamiento utilizando PCA y entrenamiento utilizando LDA. La primera fase consiste en entrenar el modelo y tratar de ajustar los hiperparámetros de los modelos los cuales recibieron las 36 características para el entrenamiento. En la segunda fase, se utilizaron solo aquellas características que fueran relevantes para el paradigma SSVEP que, para este caso, son la frecuencia principal dentro de la FFT, la potencia de la frecuencia principal y la actividad de los parámetros Hjorth dado que todas estas características se encuentran ligadas a la frecuencia de estimulación que se busca generar en el cerebro. Para la tercera fase, del conjunto de datos, se buscó poder describir al menos un 70 % de la información relevante mediante los componentes principales, de esta manera se podían elegir desde 4 componentes a más. Por último, se aplicó el LDA para el entrenamiento a fin de mejorar la separabilidad entre clases. Cada una de estas fase se aplicaron a los 3 pares de frecuencias elegidas previamente y se eligieron aquellos modelos que presentaran el mejor rendimiento en cuanto a precisión

de entrenamiento y precisión en la validación para posteriormente comparar cada uno de los sistemas con su desempeño en la clasificación en línea.

4.1.1. Evaluación de los modelos

Para la evaluación de los modelos de clasificación existen diversos criterios, uno de los más utilizados son las matrices de confusión ya que estas concentran y detallan la distribución de errores de un modelo de clasificación. Una matriz de confusión para el caso de la clasificación de dos clases se puede visualizar como en la Figura 4.1. Para la evaluación de los modelos de clasificación propuestos en este proyecto, se usarán las matrices de confusión y las métricas relacionadas a este para la selección del mejor modelo.

		Clase Predicha	
		1	0
Clase Real	1	Verdadero Positivo (VP)	Falso Negativo (FN)
	0	Falso Positivo (FP)	Verdadero Negativo (VN)

Figura 4.1: Matriz de confusión para 2 clases

Además de la visualización de la distribución de errores y aciertos de una clasificación, las matrices de confusión también nos permiten obtener algunas métricas importantes que son de utilidad para la evaluación e implementación de los modelos de clasificación, a continuación se describen algunas de las métricas más importantes.

Exactitud

Se mide como la proporción de resultados positivos (Verdaderos Positivos y Verdaderos Negativos) con respecto al total de mediciones tal como se muestra en la Ecuación 4.1. Para efectos prácticos, la Exactitud es la proporción de predicción que fueron correcta.

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN} \quad (4.1)$$

Precisión

Es la proporción de Verdaderos Positivos entre todos los resultados positivos, es decir, tanto Verdaderos Positivos como Falso Positivo (Ecuación 4.2). En la práctica, se define como

la razón de Verdaderos Positivos obtenidos en la prueba.

$$Precision = \frac{VP}{VP + FP} \quad (4.2)$$

Sensibilidad

También conocida como Tasa de Verdaderos Positivos, esta métrica determina la proporción de casos positivos que fueron identificadas correctamente por el algoritmo. La Ecuación 4.3 describe el cálculo de esta métrica.

$$Sensibilidad = \frac{VP}{VP + FN} \quad (4.3)$$

Especificidad

Conocido también como Tasa de Verdaderos Negativos, determina la proporción de casos negativos identificados correctamente por el algoritmo. Esta métrica se calcula como la razón entre los casos Verdadero Negativos y la suma de los Verdaderos Negativos con los Falsos Positivos (Ecuación 4.4).

$$Especificidad = \frac{VN}{VN + FP} \quad (4.4)$$

Valor Predictivo Negativo (VPN)

Determina la proporción de los casos negativos correctamente identificados entre la cantidad de predicciones negativas realizadas (Ecuación ??). En la práctica, determina la probabilidad de que, al obtener un resultado negativo, realmente sea negativo.

$$VPN = \frac{VN}{VN + FN} \quad (4.5)$$

Tomando en cuenta las métricas antes mencionas y las ecuaciones correspondientes para su cálculo, se define la matriz de confusión a utilizar como se muestra en la Figura 4.2 de tal manera que se muestre la información relacionada al modelo de clasificación de manera más concentrada y visual.

4.1.2. Entrenamiento ANN

Para cada par de frecuencias se entrenaron diversas redes implementando cada una de las 4 fases a fin de encontrar aquella red que presentara el mejor desempeño con respecto al entrenamiento y la validación de datos. En cada una de las fases de entrenamiento y para cada par de frecuencias, se ajustaron cada uno de los hiperparámetros de las redes neuronales a fin de generar una ANN que permitiera generalizar correctamente los datos. Tomando en cuenta lo anterior, se entrenaron un aproximado de 200 ANN en total de las cuales en esta Sección solo se describe aquella que mostró las mejores características para la clasificación.

		Clase Predicha		
		1	0	
Clase Real	1	Verdadero Positivo (VP)	Falso Negativo (FN)	Sensibilidad
	0	Falso Positivo (FP)	Verdadero Negativo (VN)	
		Precisión	VPN	Exactitud

Figura 4.2: Descripción de la matriz de confusión utilizada

Estímulos: 8 y 12 Hz

Para este par de frecuencias de estimulación, se seleccionó el modelo entrenado con las características más relevantes del paradigma SSVEP. Los parámetros de entrenamiento utilizados para el modelo son los siguientes:

- ✓ Tamaño del conjunto de prueba: 20 %
- ✓ Semilla del split: 3
- ✓ Cantidad de capas ocultas: 2
- ✓ Neuronas por capa oculta: 15 en ambas
- ✓ Función de activación de las capas ocultas: *relu*
- ✓ Factor de regularización α : 0,001
- ✓ Solver: *adam*
- ✓ Razón de aprendizaje: 0,004
- ✓ Semilla de la red: 5
- ✓ Tamaño de batch: 5

La curva de pérdida obtenida del entrenamiento se encuentra en la Figura 4.3.

Los resultados obtenidos para el entrenamiento de esta red neuronal fueron:

- ✓ Precisión del entrenamiento: 90.56 %
- ✓ Precisión del test de validación: 87.5 %

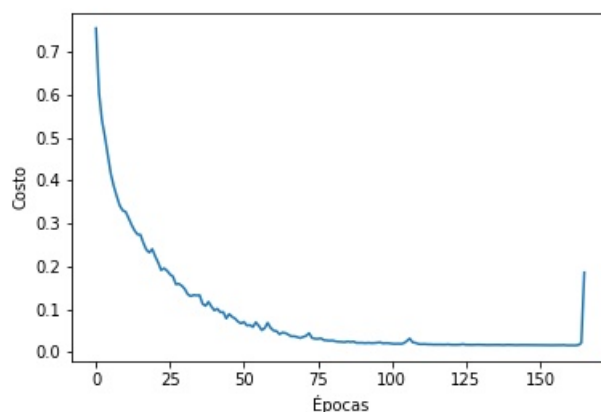


Figura 4.3: Curva de pérdida ANN con características relevantes del SSVEP

La matriz de confusión obtenida del modelo (Figura 4.4) muestra una generalización adecuada al no presentar un desbalance en la clasificación de las clases que podría traducirse en una “preferencia” hacia alguna frecuencia en específico.

		Clase Predicha		
		8 Hz	12 Hz	
Clase Real	8 Hz	20	2	90.9%
	12 Hz	3	15	83.33%
		86.95%	88.23%	87.5%

Figura 4.4: Matriz de confusión ANN con características relevantes del SSVEP

Estímulos: 13 y 15 Hz

Para estas dos frecuencias, se seleccionó el modelo entrenado con 5 características correspondientes a la fase de aplicación del PCA. Los parámetros de entrenamiento utilizados son los siguientes:

- ✓ Tamaño del conjunto de prueba: 20 %
- ✓ Semilla del split: 3
- ✓ Cantidad de capas ocultas: 2

- ✓ Neuronas por capa oculta: 15 en ambas
- ✓ Función de activación de las capas ocultas: *relu*
- ✓ Factor de regularización α : 0,001
- ✓ Solver: *adam*
- ✓ Razón de aprendizaje: 0,01
- ✓ Semilla de la red: 5
- ✓ Tamaño de batch: 5

Debido a la dificultad presentada en la separación de las clases ligado a un solapamiento entre las frecuencias dentro de las muestras obtenidas, se entrenaron una gran cantidad de redes eligiendo aquella que presentara un mejor comportamiento. Por lo anterior, la curva de pérdida de la Figura 4.5 muestra un aprendizaje un tanto accidentado pues se buscaba que la red entrenada presentara el menor sobreajuste y con un balance en la precisión.

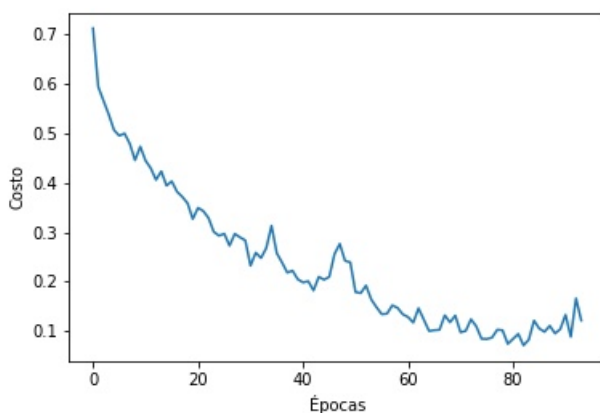


Figura 4.5: Curva de pérdida ANN con 5 características PCA

Las precisiones obtenidas en el entrenamiento para el modelo fueron:

- ✓ Precisión del entrenamiento: 96,85 %
- ✓ Precisión del test de validación: 72,5 %

La matriz de confusión asociada al modelo se encuentra en la Figura 4.6 en la cual se observa un comportamiento poco adecuado en el test de validación puesto que a pesar que la precisión del entrenamiento haya sido mayor al 90 %, realmente se puede apreciar un sobreajuste al analizar la matriz de confusión puesto que la precisión en la clasificación cae al 72.5 %.

		Clase Predicha		
		13 Hz	15 Hz	
Clase Real	13 Hz	15	7	68.18%
	15 Hz	4	14	77.77%
		78.94%	66.66%	72.5%

Figura 4.6: Matriz de confusión ANN con 5 características PCA

Estímulos: 31 y 33 Hz

En este par de frecuencias, se seleccionó el modelo que utiliza la técnica de reducción de la dimensionalidad LDA pues fue el que mejor se ajustó a los datos. Los parámetros de entrenamiento fueron los siguientes:

- ✓ Tamaño del conjunto de prueba: 20 %
- ✓ Semilla del split: 3
- ✓ Cantidad de capas ocultas: 2
- ✓ Neuronas por capa oculta: 8 en ambas
- ✓ Función de activación de las capas ocultas: *relu*
- ✓ Factor de regularización α : 0,001
- ✓ Solver: *adam*
- ✓ Razón de aprendizaje: 0,01
- ✓ Semilla de la red: 5
- ✓ Tamaño de batch: 5

La curva de aprendizaje del modelo (Figura 4.7) muestra un entrenamiento bastante rápido puesto que utiliza menos épocas comparado con los modelos anteriores.

Los resultados obtenidos con el modelo en cuanto a precisión fueron:

- ✓ Precisión del entrenamiento: 95 %
- ✓ Precisión del test de validación: 94.87 %

La matriz de confusión asociada al modelo se encuentra en la Figura 4.8.

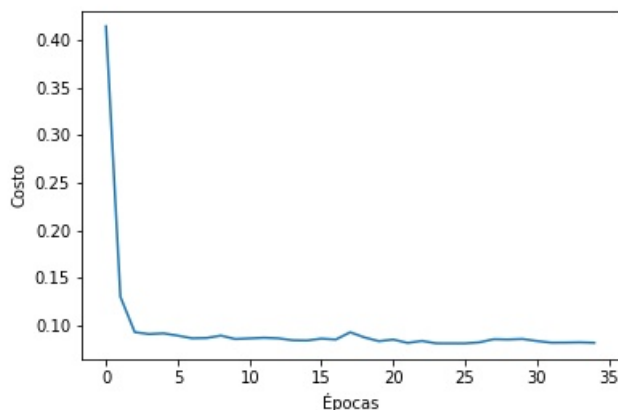


Figura 4.7: Curva de pérdida ANN con aplicación del LDA

		Clase Predicha		
		31 Hz	33 Hz	
Clase Real	31 Hz	21	1	95.45%
	33 Hz	1	17	94.44%
		95.45%	94.44%	95%

Figura 4.8: Matriz de confusión ANN con aplicación del LDA

4.1.3. Entrenamiento SVM

Dado que para las SVM no se tiene un sistema de entrenamiento similar al de las ANN, se optó por dividir el conjunto de datos en entrenamiento y prueba para poder determinar las características de salida de la SVM. Para todas las frecuencias se entrenaron 8 SVM las cuales se dividen en cada una de las 4 fases previamente descritas y utilizando uno de los dos kernel disponibles, lineal y RBF.

Estímulos: 8 y 12 Hz

El modelo SVM con mejor rendimiento para este par de frecuencias fue el entrenado con la totalidad de las características disponibles (36 características). Los parámetros de entrenamiento son los siguientes:

- ✓ Tamaño del conjunto de prueba: 20 %

- ✓ Semilla del split: 42
- ✓ Kernel: RBF
- ✓ Cantidad de características: 36 (Primera fase)

Al aplicar el conjunto de prueba para la validación del entrenamiento, se obtuvo un porcentaje de precisión del 92,5 %. La matriz de confusión asociada al modelo se muestra en la Figura 4.9 en la cual puede observarse un desbalance en la precisión entre las clases puesto que existe una preferencia hacia una de estas.

		Clase Predicha		
		8 Hz	12 Hz	
Clase Real	8 Hz	18	3	85.71%
	12 Hz	0	19	100%
		100%	86.36%	92.5%

Figura 4.9: Matriz de confusión de la SVM entrenada con 36 características.

Estímulos: 13 y 15 Hz

El modelo con mejor rendimiento para este par de frecuencias fue el entrenado con la totalidad de las características disponibles. Los parámetros de entrenamiento son los siguientes:

- ✓ Tamaño del conjunto de prueba: 20 %
- ✓ Semilla del split: 15
- ✓ Kernel: Linear
- ✓ Cantidad de características: 36 (Primera fase)

Una vez aplicado el conjunto de prueba al modelo obtenido para la validación del entrenamiento, el porcentaje de precisión obtenido fue del 80 %. La matriz de confusión del modelo (Figura 4.10) muestra una mejora relativa con el modelo ANN presentado anteriormente.

		Clase Predicha		
		13 Hz	15 Hz	
Clase Real	13 Hz	13	4	76.47%
	15 Hz	4	19	82.6%
		76.47%	82.6%	80%

Figura 4.10: Matriz de confusión de la SVM entrenada con 36 características.

Estímulos: 31 y 33 Hz

Para este par de frecuencias, el modelo con mejor rendimiento obtenido fue el entrenado utilizando la técnica de reducción de la dimensionalidad LDA. Los parámetros de entrenamiento se enlistan a continuación:

- ✓ Tamaño del conjunto de prueba: 20 %
- ✓ Semilla del split: 3
- ✓ Kernel: rbf
- ✓ Cantidad de características: 1 LDA (Cuarta Fase)

Después de aplicar el conjunto de prueba para la validación del entrenamiento, se obtuvo un porcentaje de precisión del 95 %. La matriz de confusión del modelo que se encuentra en la Figura 4.11, muestra un equilibrio en la clasificación de las clases así como una precisión alta.

4.1.4. Selección de sistema y modelo de clasificación

Una vez realizado el entrenamiento de los distintos modelos y tomando en cuenta las precisiones de entrenamiento y validación así como las características utilizadas, se construyó la Tabla 4.1 la cual muestra de manera resumida estos parámetros.

De la Tabla 4.1 se observó que, en general, las SVM produjeron mejores resultados de clasificación comparado con las ANN compuestas de 2 capas ocultas. Dado que para la implementación es necesario definir el mejor modelo, se tomaron en cuenta los porcentajes de desempeño tanto en entrenamiento como en clasificación así como las matrices de confusión

		Clase Predicha		
		31 Hz	33 Hz	
Clase Real	31 Hz	21	1	95.45%
	33 Hz	1	17	94.44%
		95.45%	94.44%	95%

Figura 4.11: Matriz de confusión de la SVM entrenada con LDA.

Tabla 4.1: Resumen de los modelos de clasificación entrenados

Estímulos	Clasificador	Precisión En-trenamiento	Precisión Validación	Características Utilizadas
8 Y 12 Hz	ANN	90.56 %	87.5 %	12 SSVEP
	SVM	No aplica	92.5 %	36 Totales
13 y 15 Hz	ANN	96.85 %	72.5 %	5 PCA
	SVM	No aplica	80.0 %	36 Totales
31 y 33 Hz	ANN	95.00 %	95.0 %	1 LDA
	SVM	No aplica	95.0 %	1 LDA

de cada sistema para evitar preferencias en la clasificación que comprometan el buen funcionamiento. Por lo anterior, se eligió al sistema que utiliza las frecuencias de estimulación de 31 y 33 Hz como sistema a implementar puesto que las matrices de confusión tanto de la ANN como de la SVM no muestran preferencias de clasificación así como ser las que obtuvieron el mejor desempeño tanto en entrenamiento como en validación.

La configuración final del modelo de clasificación para la ANN (Figura 4.12) en el sistema elegido consta de 2 capas ocultas con 8 neuronas por capa utilizando una sola característica proporcionada por la técnica de PCA. Por su parte, la configuración final del modelo de clasificación para la SVM (Figura 4.13) en el sistema elegido consta de una característica de entrada proporcionada por la PCA y la SVM con el kernel RBF implementado.

Para la selección del modelo de clasificación a implementar en el sistema seleccionado, se tomaron en cuenta algunos aspectos extra con la finalidad de simplificar el ajuste del modelo a nuevos datos y por ende, a nuevos usuarios así como la reducción en cuanto al coste computacional y el tiempo requerido para realizar los nuevos ajustes. Los puntos a tomar en cuenta para la selección son los siguientes:

1. Facilidad de entrenamiento.

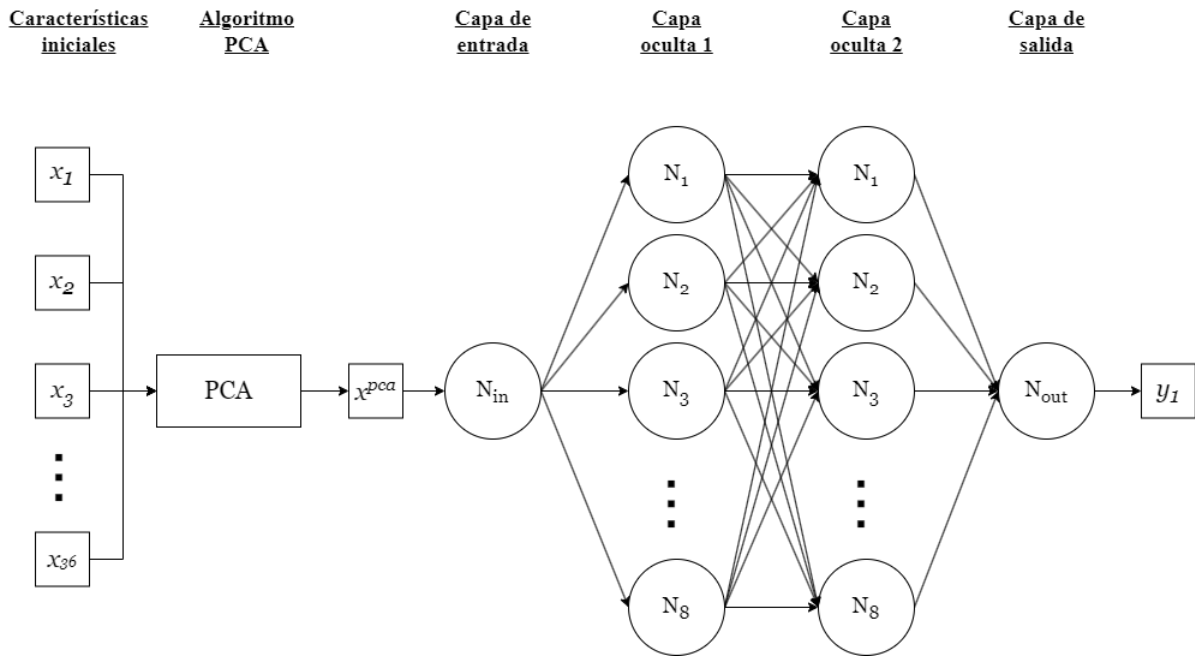


Figura 4.12: Configuración Final de la ANN para el sistema de 31 y 33 Hz.

2. Facilidad de ajuste con respecto a nuevos datos.
3. Costo computacional.

Tomando en cuenta los puntos anteriores, el modelo que mejor se ajusta al sistema es el SVM; debido a que se requieren menos datos heurísticos para el entrenamiento lo que se traduce en un menor tiempo de entrenamiento siendo además menos costosos computacionalmente, eliminando la posibilidad de sobre ajuste.

El script con el modelo implementado en Python se encuentra en el Anexo C.

4.2. Acondicionamiento de la mano robótica virtual

En esta sección se detallan la selección, ajuste y acondicionamiento de la mano robótica virtual a utilizar para la implementación de la BCI.

4.2.1. Selección de la mano robótica a utilizar

En un principio, se propuso utilizar la mano robótica desarrollada en la universidad [31] por lo que se comenzó con la identificación y acondicionamiento de la mano para poder integrarla al proyecto. Sin embargo, al realizar la identificación tanto electrónica como de software, se detectaron algunas discrepancias reportadas en el trabajo y las implementadas en el prototipo por lo que se realizó una comparativa entre lo reportado y lo implementado para identificar aquellos componentes o códigos que fuesen distintos dando como válida la opción

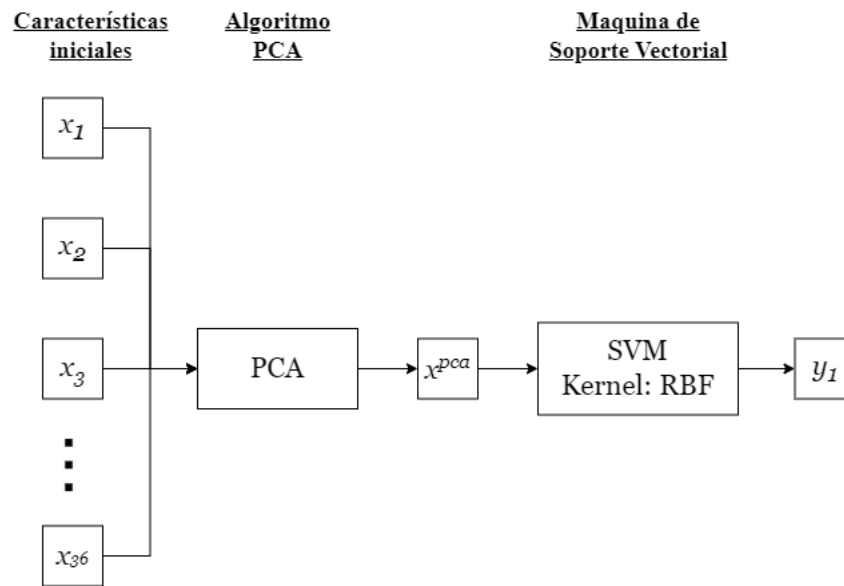


Figura 4.13: Configuración Final de la SVM para el sistema de 31 y 33 Hz.

implementada.

Asimismo, dentro del acondicionamiento de la mano robótica propuesta, se detectaron algunos problemas en cuanto a la electrónica implementada. A continuación se describen las partes y problemas identificados:

- ✓ Sensores resistivos de fuerza: Las conexiones de estos sensores son delicadas además de implementar láminas de cobre para la unión de los sensores con el cableado lo que lo vuelve sensible a los falsos contactos. También se identificaron algunas láminas de cobre que presentaban desgaste en la soldadura producto del movimiento de la mano.
- ✓ Cristales osciladores y transistores: Se detectaron fallas en el funcionamiento de estos componentes ya que en ocasiones mostraban un comportamiento anómalo pudiendo cambiar los cristales sin embargo, no se encontraron transistores comerciales asequibles que cumplieran con las especificaciones requeridas. además de que las conexiones de estos componentes presentaban fallas en las uniones.
- ✓ Falsos contactos entre tarjetas de control de motores y los motores: Se detectaron problemas en cuanto al cableado correspondiente a la comunicación hacia y desde los motores con las tarjetas de control. La falla principal del cableado consistía en rupturas en el cable lo que ocasionaba que se generaran falsos contactos así como una mala comunicación de las tarjetas dando como resultado fallas al momento de realizar los movimientos con la mano robótica.

Tomando en cuenta los problemas identificados, se determinó que la opción viable sería la realización de un nuevo circuito para la comunicación y control con la mano lo que requeriría un ajuste completo del prototipo así mismo se determinó que era necesario realizar un cambio

total en el cableado del circuito electrónico, sin embargo, tomando en cuenta los objetivos planteados y las limitaciones inherentes al proyecto en cuanto a tiempo y alcances, se optó por sustituir el prototipo físico por un prototipo virtual a fin de generar la retroalimentación hacia el usuario. Además, este prototipo virtual podría ser utilizado no solo para este proyecto sino también, se podría utilizar todo el sistema diseñado a modo de plataforma de experimentación para la detección y corrección de fallas en sistemas BCI antes de llevarlas al prototipo físico.

Tomando en cuenta lo anterior y a fin de tener un sistema de retroalimentación visual para el usuario, se determinó el uso del prototipo virtual por lo que fue necesario realizar una selección para el prototipo a utilizar. Para la selección de la mano robótica a utilizar se optó por una de código abierto, que permitiera su implementación en físico para futuros proyectos con bajo costo así como permitir la ejecución de diversas posturas. De la bibliografía consultada, se seleccionó la presentada por Krausz *et al.* en [32] (Figura 4.14) la cual posee 6 grados de libertad y 10 juntas móviles los cuales permiten alcanzar diversas posturas.

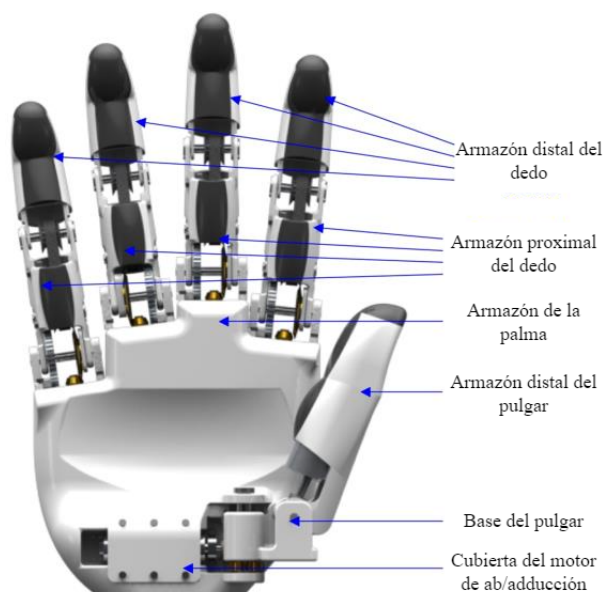


Figura 4.14: Mano robótica de código abierto [32].

4.2.2. Ajuste y acondicionamiento de la mano robótica virtual

Para poder utilizar la mano robótica en tiempo de ejecución del sistema BCI, se tuvieron que realizar algunos ajustes y acondicionamiento al modelo original. Inicialmente, el ensamble de la mano robótica se encontraba realizado en el programa de SolidWorks sin embargo, para poder realizar la ejecución de movimiento así como el análisis de las trayectorias de los dedos se tuvo que exportar el archivo al programa de Adams View. Al realizar la exportación del archivo al programa de Adams View, se tuvo que simplificar el modelo a menos de 40 piezas dado que

la licencia estudiantil otorgada por la empresa permite hasta un máximo de 40 piezas. Para realizar dicha simplificación, se tuvieron que eliminar todos los componentes relacionados con la representación de motores, engranes, bandas y tornillos así como la realización de subensamblajes para que el modelo se comportara como un sistema sólido y unificado en los dedos pero permitiendo el movimiento en las juntas. En la Figura 4.15 se puede observar la simplificación realizada.

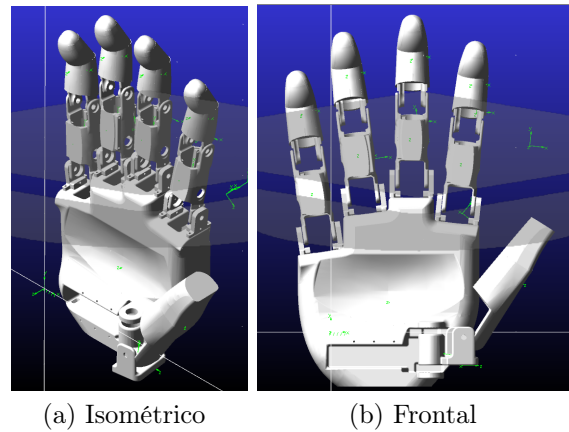


Figura 4.15: Mano robótica virtual simplificada.

Para el acondicionamiento, se agregaron en Adams View la conexión de junta revoluta en cada una de las juntas de la mano, así mismo se añadieron los motores correspondientes a cada una de las conexiones para poder realizar los movimientos necesarios. Finalmente se agregaron los sensores de medición para cada uno de las falanges de los dedos así como las variables de estado ligadas al movimiento y medición del mismo. Las variables de estado quedaron como se muestran en las Tablas 4.2 y 4.3.

Tabla 4.2: Variables de estado de movimiento

Variable de estado	Definición
mPA	Motor Pulgar Distal
mPB	Motor Pulgar Proximal
mIA	Motor Índice Distal
mIB	Motor Índice Proximal
mMA	Motor Medio Distal
mMB	Motor Medio Proximal
mAA	Motor Anular Distal
mAB	Motor Anular Proximal
mMqA	Motor Meñique Distal
mMqB	Motor Meñique Proximal

Tabla 4.3: Variables de estado de medición

Variable de estado	Definición
rPA	Rotación Pulgar Distal
rPB	Rotación Pulgar Proximal
rIA	Rotación Índice Distal
rIB	Rotación Índice Proximal
rMA	Rotación Medio Distal
rMB	Rotación Medio Proximal
rAA	Rotación Anular Distal
rAB	Rotación Anular Proximal
rMqA	Rotación Meñique Distal
rMqB	Rotación Meñique Proximal

4.3. Integración de sistemas

En esta sección se describen los pasos que se siguieron para realizar la integración de los distintos módulos para crear un solo sistema que funcione de manera coordinada sin generar problemas entre sí.

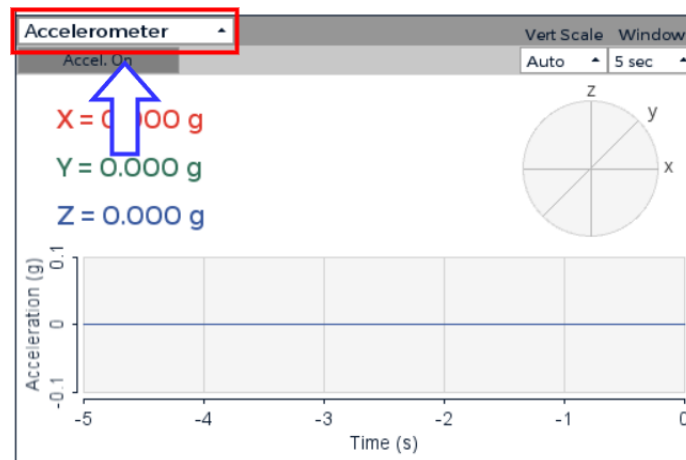
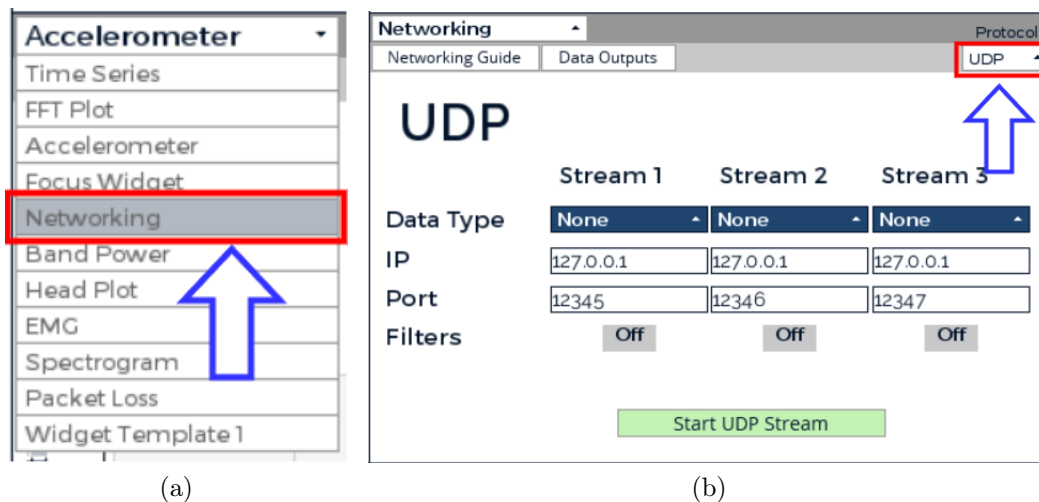
4.3.1. Comunicación GUI con Matlab

Para realizar la comunicación entre la GUI con el programa de Matlab, la GUI ofrece distintas maneras de comunicarse como son el Protocolo de Datagramas de Usuario (UDP), el protocolo Serial, Lab Streaming Layer (LSL) y el protocolo Open Sound Control (OSC). De los protocolos mencionados anteriormente, se seleccionó el LSL debido a que éste permite enviar/recibir flujos de datos de series temporales reduciendo la pérdida de datos además de simplificar la transmisión de datos electrofisiológicos entre diversos dispositivos.

Para generar la transmisión de datos mediante LSL dentro de la GUI, primero, en el apartado de *Accelerometer* se hace clic en esa etiqueta (Figura 4.16).

Posteriormente se abrirá un menú del cual se seleccionará el apartado *Networking* (Figura 4.17a), se desplegará la pantalla de comunicación UDP por defecto, se dará clic en la pestaña de *Protocol* que se encuentra en la parte superior derecha del apartado *Networking* (Figura 4.17b).

Del menú desplegable se selecciona *LSL* (Figura 4.18a), se desplegará la pantalla de comunicación LSL. Se selecciona *TimeSeries* en la sección *Data Type*. Finalmente se inicia la transmisión de datos haciendo clic en el botón *Start LSL Stream* (Figura 4.18b), no es necesario modificar ningún parámetro de la comunicación extra puesto que por defecto se transmiten los datos obtenidos del casco sin procesar (formato RAW).

Figura 4.16: Apartado *Accelerometer* de la GUI.

(a)

(b)

Figura 4.17: Selección del apartado de comunicación.

Para realizar recepción de datos entre la GUI y Matlab mediante LSL, se utilizó la librería de LSL para Matlab que se encuentra en el repositorio de Github por el usuario *LabStreamingLayer submodules*[33]. Una vez instalada la librería siguiendo los pasos que se describen en el documento *README.md* disponible en el repositorio, se incluye en el *path* del directorio de trabajo en Matlab la carpeta *liblsl-Matlab* ya que dentro de ella se encuentran los módulos y funciones que se utilizarán dentro del script para la recepción de datos.

4.3.2. Comunicación Adams View y Matlab

Para poder realizar la comunicación entre Adams View y Matlab, es necesario crear una planta de trabajo desde Adams hacia Matlab, para ello, toda vez se han definido todos los parámetros necesarios para el control de la mano robótica así como las variables de control y salida del modelo, es necesario exportar el modelo como una planta de Matlab. Para exportar el

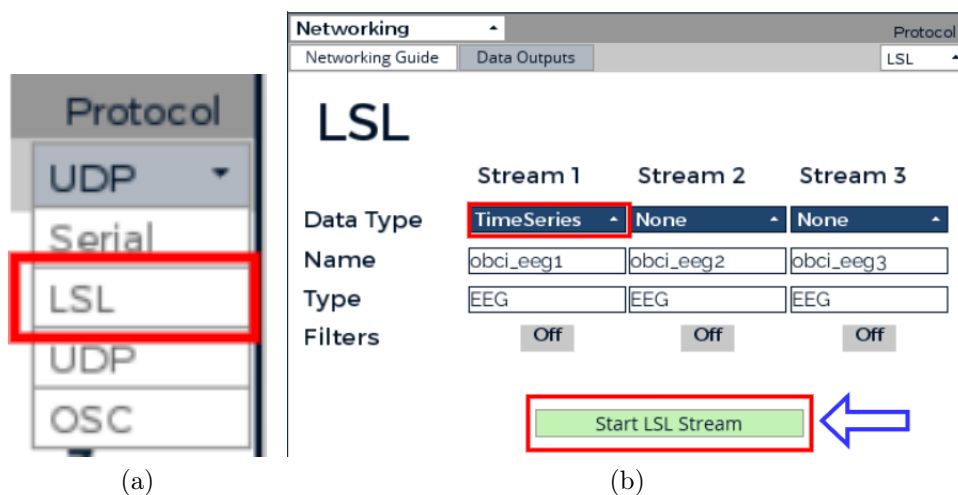


Figura 4.18: Inicio de la comunicación LSL.

modelo como una planta de Matlab, desde la barra de herramientas de Adams, se selecciona en *Plugins*, posteriormente en *Controls*, se desplegará una pestaña desde donde se elige la opción *Plant Export* (Figura 4.19).

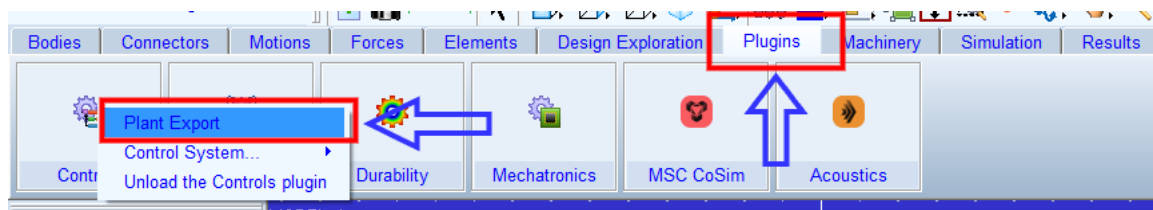


Figura 4.19: Exportar modelo de Adams a Matlab.

Posteriormente se abrirá una ventana en donde se configurarán el nombre de la planta a exportar, las variables de entrada y salida del modelo, la aplicación objetivo y el tipo de *solver* a utilizar (Figura 4.20). Una vez configurados estos parámetros, se da clic en *OK* y se creará la planta en formato *.xml* de Simulink.

4.3.3. Script de integración en Matlab

Una vez obtenidas las interfaces de comunicación hacia Matlab de la GUI y Adams View, se desarrolló un script de control para la realización del proceso completo de adquisición de señales, procesamiento, clasificación y movimiento de la mano robótica. En la Figura 4.21 se observa el proceso que sigue el script realizado para ejecutar las tareas antes mencionadas.

A continuación se describen los puntos más relevantes del script.

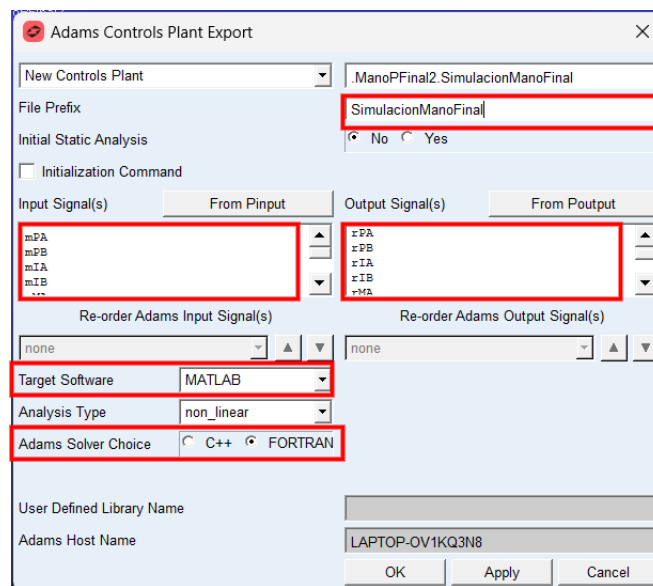


Figura 4.20: Configuración de parámetros de la planta a exportar.

Inicio de la comunicación por LSL

Se realiza la comunicación por LSL instanciando la librería por medio de la función *lsl_loadlib()* asignada a la variable *lib* para posteriormente realizar la conexión de datos con la función *lsl_resolve_byprop()* asignada a la variable *result*, a la función se le pasa como argumentos:

- ✓ *lib*: la instancia de la librería.
- ✓ 'type', 'EEG': el cual define la comunicación como una transmisión de datos de tipo EEG.

Finalmente, se crea un bus de entrada a través de la función *lsl_inlet()* con el argumento *result{1}* asignado a la variable *inlet*.

Adquisición de señales en formato RAW

Una vez generada la comunicación mediante LSL, se adquieren los datos a través del método *pull_sample()* de la variable *inlet* la cual recibe un vector de 8 columnas correspondientes a cada canal dentro de la GUI. Se adquieren 3 ventanas de datos de 3 segundos (750 datos) con una superposición del 80 % para asegurar la adquisición de las frecuencias de interés solo de los primeros 4 canales, dando un tiempo de adquisición de 4.2 segundos.

Clasificación

Para la clasificación, se creó un archivo en lenguaje Python el cual contiene el modelo y las transformaciones de los datos necesarios para la clasificación de los datos. Para la llamada del archivo, se hace uso de la sentencia *py* dentro del script de Matlab, siguiendo la estructura **py.NombreArchivo.Definicion** por lo que dentro del script quedó como

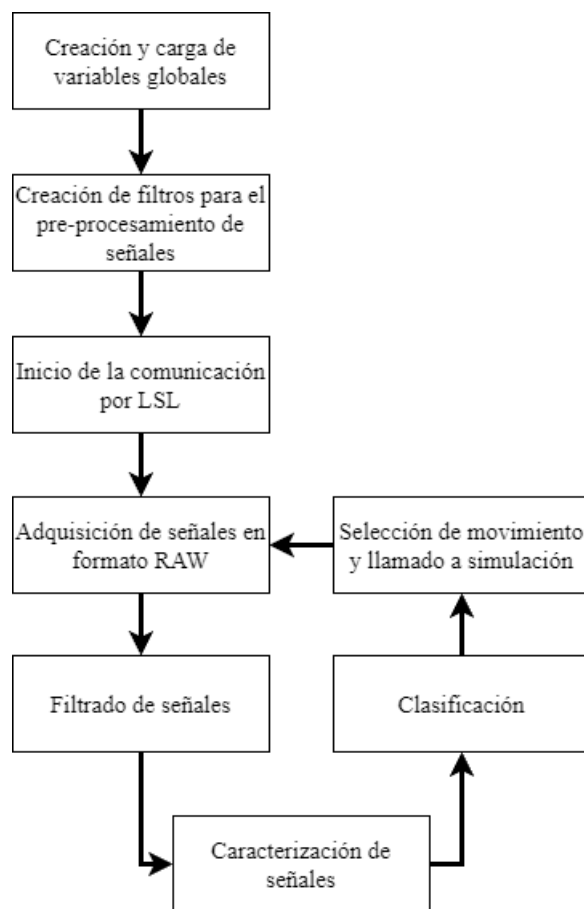


Figura 4.21: Diagrama de flujo del script de control en tiempo real.

`py.ClasificadorFinal.neural()` y mandando como argumento el vector de características obtenido de la misma forma que en la creación del banco de datos. La salida de esta sentencia se guarda en un vector de 3 posiciones puesto que se obtiene una clasificación por ventana. Finalmente, a través de la función `mode()`, se obtiene la clasificación que más se repite la cual se guarda en la variable `postura`.

Selección de movimiento y llamado a simulación

Una vez obtenida la postura a realizar, se llama la planta de simulación en simulink a través de la función `sim()` con los parámetros:

- ✓ modelo: Nombre del archivo de simulink
- ✓ Tiempo de ejecución: Se define a través de “StopTime” y “4”

Finalmente se realiza la ejecución del movimiento de la mano robótica dentro del entorno de Adams View para de nuevo iniciar a adquirir datos de la GUI.

El script final de comunicación se encuentra en el Anexo D.

Capítulo 5

Pruebas y Resultados

En este capítulo se detallan las pruebas realizadas a las distintas partes de la BCI así como a la BCI en conjunto.

Las pruebas son las siguientes:

- ✓ Prueba de desempeño de la SVM en línea.
- ✓ Prueba de clasificación del sistema BCI.
- ✓ Prueba de movimiento de la mano robótica virtual.
- ✓ Prueba de integración de los distintos sistemas.

5.1. Modelo de clasificación en línea

Para el modelo de clasificación en línea se utilizaron dos señales EEG completamente nuevas, una para cada frecuencia, que no fueron utilizadas para el entrenamiento del modelo de clasificación. Además, dichas señales fueron adquiridas bajo las mismas condiciones y momento que las utilizadas para el entrenamiento del modelo así como ser tratadas de la misma manera para evitar ingresar incertidumbres al sistema. Las condiciones de adquisición se describen en la Sección 3.2.3 las cuales permiten asegurar una buena adquisición de señales así como una pauta para la repetibilidad del proyecto.

Una vez obtenidas y filtradas, estas señales se dividieron en ventanas de 3 segundos con un solapamiento del 80 % dando un total de 39 ventanas en total, con 19 muestras para el estímulo de 31 Hz y 20 para el estímulo de 33 Hz. Estas muestras fueron tratadas como la señal de validación dado que en esta prueba únicamente se evaluó el modelo de clasificación obtenido en el capítulo anterior.

Al aplicar la señal de validación al modelo de clasificación en la fase en línea, se obtuvo un porcentaje de desempeño de 92,3 %. La matriz de confusión mostrada en la Figura 5.1

describe el comportamiento de las clasificaciones realizadas por el modelo.

		Clase Predicha		
		31 Hz	33 Hz	
Clase Real	31 Hz	18	1	94.73%
	33 Hz	2	18	90%
		95.45%	94.73%	92.3%

Figura 5.1: Matriz de confusión de validación de la SVM seleccionada.

Tomando en cuenta la matriz de confusión de la Figura 5.1, se observó un comportamiento similar al mostrado en el entrenamiento puesto que para estos nuevos datos, la diferencia de desempeño de la SVM varía con respecto del entrenamiento en un 2,7 % lo que comprueba la correcta generalización del modelo. De la misma manera, el comportamiento de la clasificación en línea no muestra preferencias hacia ninguna clase por lo que se valida el modelo SVM elegido como óptimo para el sistema.

5.2. Desempeño de la clasificación de la BCI

Para esta prueba, se utilizará la BCI para validar el desempeño y comprobar la correcta integración entre la adquisición, filtrado y caracterización con la SVM. Como se mencionó en el apartado 4.3.3 en los subtemas de adquisición de señales y clasificación del capítulo anterior, dentro de la BCI se utilizan 3 ventanas de 3 segundos con un solapamiento del 80 % con el cual se asegura la correcta adquisición de las señales de interés siendo cada ventana filtrada, caracterizada y clasificada de manera independiente. Para seleccionar la salida adecuada de las 3 clasificaciones, se utiliza la moda de éstas puesto que de esta manera se evitan los falsos positivos para la frecuencia que se está analizando.

Tomando en cuenta lo anterior y la duración de la señal para cada frecuencia de estimulación, se obtuvieron 4 muestras de 4.2 segundos para cada una de las frecuencias siendo un total de 8 muestras para la señal de validación de la BCI. Esta señal fue aplicada a la BCI obteniendo un porcentaje de desempeño en la clasificación del 100 % y cuya matriz de confusión se muestra en la Figura 5.2.

De la matriz de confusión se comprobó que se mejora el desempeño en la clasificación de las clases dentro de la BCI al tomar la moda de las 3 clasificaciones como se mencionó

		Clase Predicha		
		31 Hz	33 Hz	
Clase Real	31 Hz	4	0	100%
	33 Hz	0	4	100%
		90%	100%	100%

Figura 5.2: Matriz de confusión de la BCI.

anteriormente. Además, se comprueba la correcta integración entre los distintos módulos que conforman la BCI así como una correcta adquisición de los datos provenientes de la GUI.

5.3. Movimiento de la mano robótica

En esta prueba se analizaron los movimientos realizados por la mano al ejecutar dos movimientos: saludo y pinza fina. Cada una de las posiciones se consideran para este proyecto como posiciones fijas por lo que no hay retroalimentación por parte del modelo virtual siendo así que cada movimiento se ejecuta de la misma manera en cada prueba realizando un recorrido predefinido previamente. La posición inicial de la mano robótica virtual es con la palma abierta con todos los dedos completamente extendidos tal como se muestra en la Figura 5.3, todos los movimientos realizados por la mano, se ejecutarán desde esta posición inicial considerada como punto 0. Para los dedos índice, medio, anular y meñique, el eje de rotación es el X mientras que para la falange proximal del pulgar será en el eje Y y únicamente la falange distal del pulgar puede tener rotación en los 3 ejes de acuerdo al movimiento del mismo.

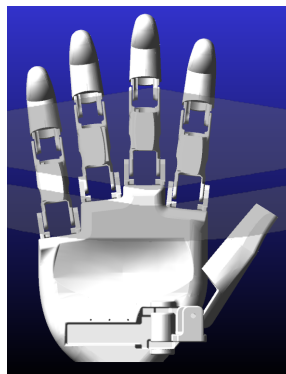


Figura 5.3: Posición inicial de la mano robótica virtual.

5.3.1. Movimiento de saludo

Para este movimiento, se considera el movimiento únicamente de los dedos meñique, anular y pulgar. La posición final a alcanzar se muestra en la Figura 5.4.

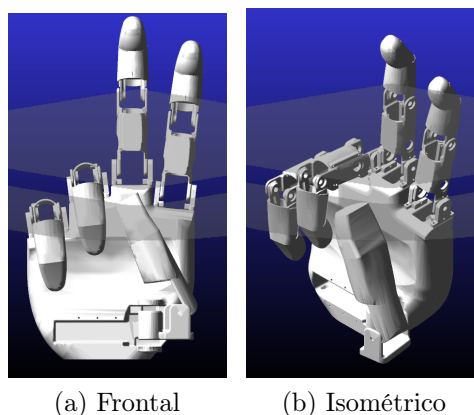


Figura 5.4: Posición final de la mano virtual con el movimiento de saludo.

Para la realización de esta prueba, se ejecutó la cosimulación entre Matlab y Adams View de tal manera que los datos de entrada para el modelo en Adams sean enviados desde el entorno de Matlab a través de la planta generada en la Sección 4.3.2 del capítulo anterior. Dado que cada dedo está dividido en dos partes, para realizar el movimiento, es necesario una matriz de 41x11 donde las filas representan el tiempo de ejecución mientras las columnas la posición de cada una de las partes de cada dedo comenzando desde el pulgar. La primera columna, representa al tiempo de ejecución del movimiento el cual comienza en 0 y culmina en 4 segundos con saltos de 0.1 segundos mientras que las columnas 2 a la 11 representan la posición en radianes de cada una de las partes de los dedos comenzando con la falange distal del pulgar (PA) seguido de la falange proximal del mismo (PB) para continuar con el índice siguiendo el mismo patrón. La Figura 5.5 muestra la organización de la matriz utilizada para la simulación.

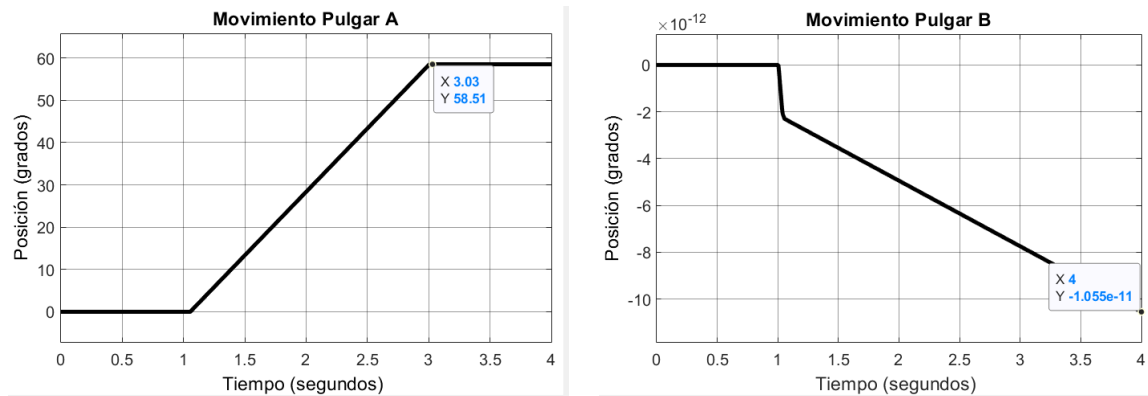
Tiempo	PA	PB	IA	IB	MA	MB	AA	AB	MqA	MqB
0	0	0	0	0	0	0	0	0	0	0
0.1	0	0	0	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
4	-1.0472	0	0	0	0	0	-1.5708	-1.5708	-1.5708	-1.5708

Figura 5.5: Organización de la matriz de datos para simulación.

Una vez definidos los parámetros para la simulación, se ejecuta el movimiento. Los resultados obtenidos para cada dedo durante el movimiento fueron los siguientes.

Dedo Pulgar

Las gráficas de movimiento obtenidas para este dedo se encuentran en la Figura 5.6. En este movimiento, la rotación de la falange distal se encuentra sobre el eje Z.



(a) Gráfica de movimiento de la falange distal. (b) Gráfica de movimiento de la falange proximal

Figura 5.6: Gráficas de movimiento del dedo pulgar.

De la Figura 5.6a se puede observar que la falange distal del pulgar realizó un recorrido de 0 a 58.51 grados en un tiempo aproximado de 2 segundos. Por su parte, la falange proximal (Figura 5.6b), no realizó ningún movimiento dando un resultado cercano a cero debido a redondeo en la simulación.

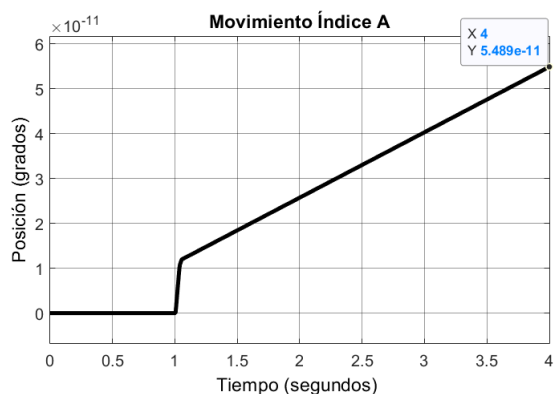
Dedo Índice y dedo Medio

En el movimiento de saludo, los dedos índice y medio no participan activamente en el movimiento. Las Figuras 5.7 y 5.8, obtenidas corroboran el nulo movimiento de estos dedos en ambas falanges.

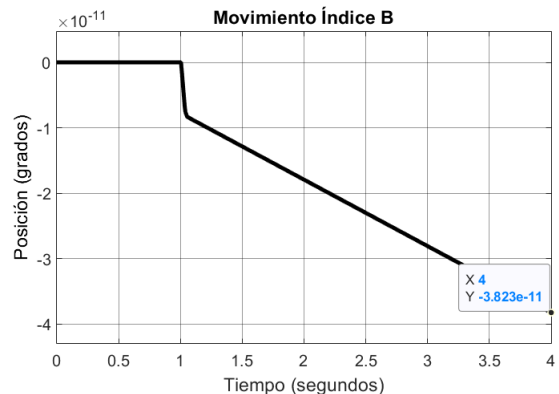
Dedos Anular y Meñique

En esta prueba de movimiento tanto el dedo anular como el meñique realizan el mismo movimiento de retracción o cierre. Las gráficas obtenidas de la simulación fueron las siguientes.

De las Figuras 5.9 y 5.10 se puede observar el comportamiento similar entre ambos recorridos con un desplazamiento de 0 a 174.2 grados para la falange distal del dedo anular (Figura 5.9a) y de 173.8 grado para la correspondiente en el meñique (Figura 5.10a), todo en un tiempo aproximado de 2 segundos. Para las falanges proximales, el recorrido se realiza de 0 a 87.12 grados para el dedo anular y de 0 a 86.92 grados para el meñique en el mismo tiempo de 2 segundos (Figuras 5.9a y 5.10b).

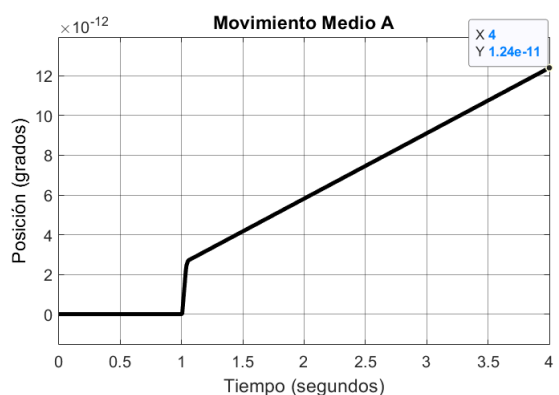


(a) Gráfica de movimiento de la falange distal.

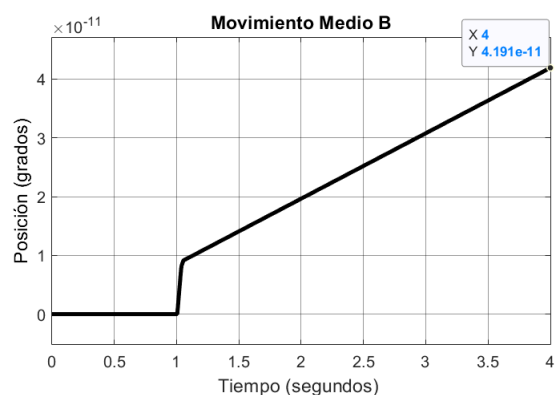


(b) Gráfica de movimiento de la falange proximal

Figura 5.7: Gráficas de movimiento del dedo índice.



(a) Gráfica de movimiento de la falange distal.



(b) Gráfica de movimiento de la falange proximal

Figura 5.8: Gráficas de movimiento del dedo medio.

5.3.2. Movimiento de pinza fina

Para este movimiento, se considera el movimiento de todos los dedos. La posición final a alcanzar se muestra en la Figura 5.11. Puesto que en este movimiento el recorrido de los dedos meñique y anular son exactamente el mismo que en la sección anterior, se obviará en esta prueba, así también, debido a que en este movimiento se ve involucrado el movimiento combinado de ambas falanges en el dedo pulgar, la falange proximal posee rotación en los 3 ejes.

Una vez ejecutada la simulación de movimiento de la mano robótica se obtuvieron los resultados siguientes.

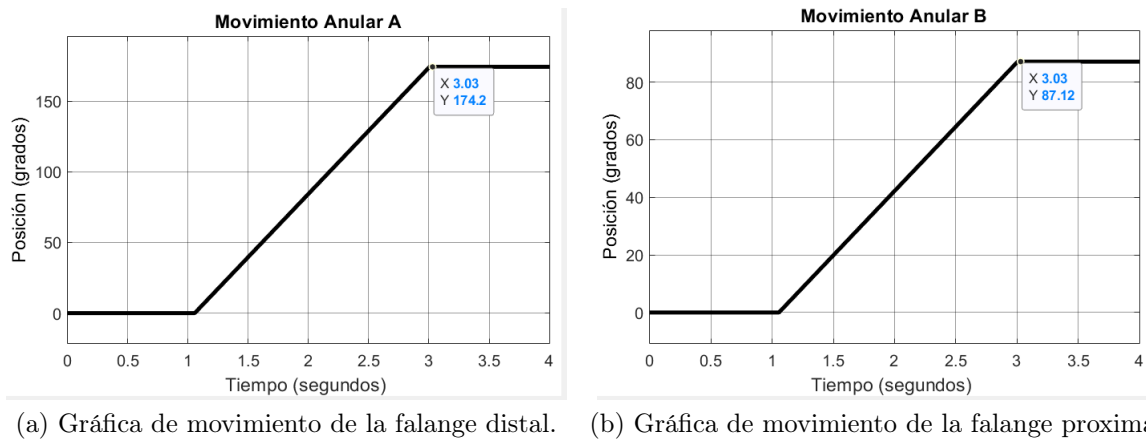


Figura 5.9: Gráficas de movimiento del dedo anular.

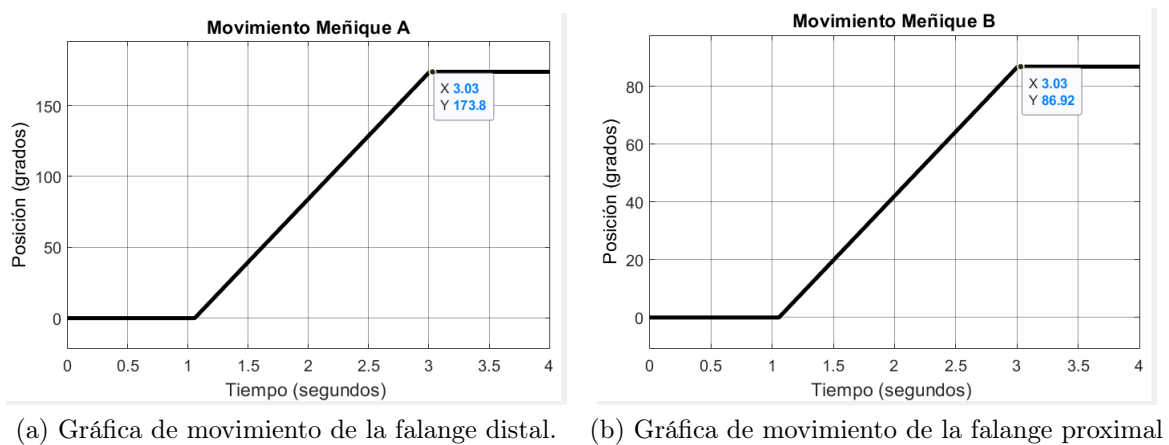


Figura 5.10: Gráficas de movimiento del dedo meñique.

Dedo Pulgar

Para el análisis de este movimiento se dividirá dos, movimiento de la falange proximal y movimiento de la falange distal. Para la falange proximal se obtuvo la gráfica de la Figura 5.12 donde se puede apreciar que la rotación de esta falange fue de 53.36 grados en sentido horario en un tiempo aproximado de 2 segundos.

Para el movimiento de la falange distal, las gráficas de movimiento obtenidas se encuentran en la Figura 5.13.

Para la rotación de la falange distal respecto al eje X, se obtuvo un giro de 7.319 grados en sentido horario mientras que para el eje Y, obtuvo una rotación de 53.36 grados en el mismo sentido, coincidente con la rotación mostrada por la falange proximal, esto debido a la dependencia que posee la falange distal sobre la falange proximal. Por último, la rotación

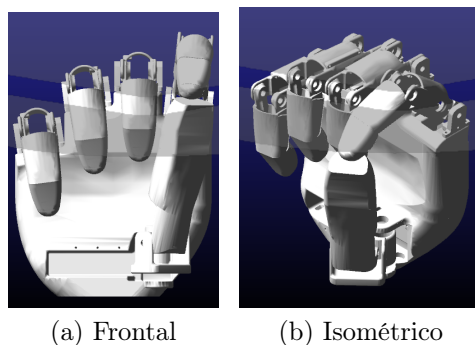


Figura 5.11: Posición final de la mano virtual con el movimiento de pinza fina.

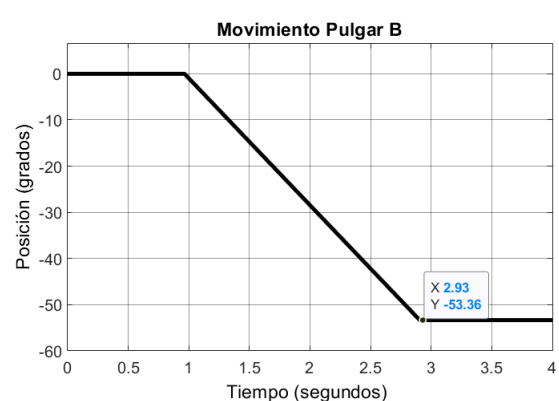


Figura 5.12: Gráfica de movimiento falange proximal .

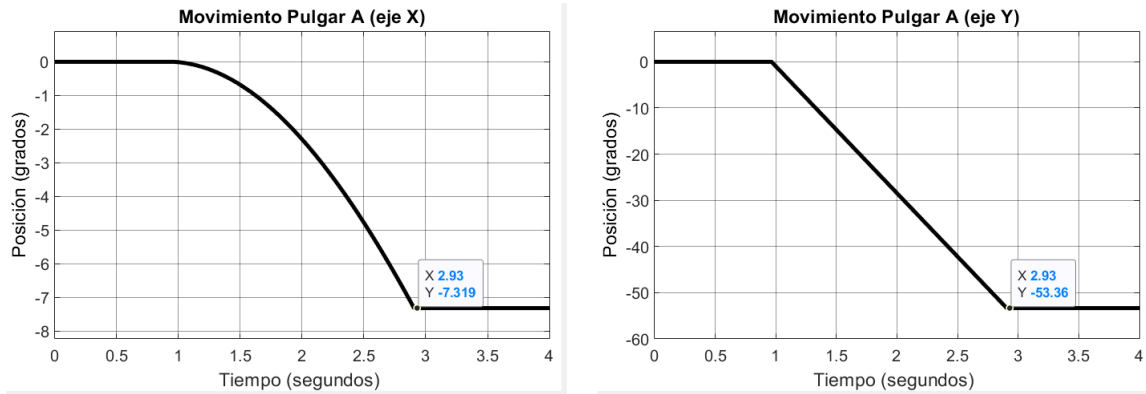
mostrada sobre el eje Z, fue de 13.61 grados al final del recorrido, todas las rotaciones realizadas por esta falange se realizaron en un tiempo aproximado de 2 segundos.

Dedo Índice

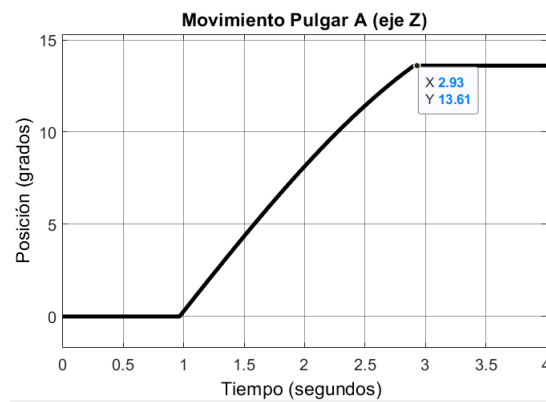
Las gráficas de movimiento obtenidas para el dedo índice después de la simulación se encuentran en la Figura 5.14. Para este movimiento, la falange distal presenta un recorrido de 0 a 108.4 grados en un aproximado de 2 segundos (Figura 5.14a) mientras que el movimiento de la falange proximal se realiza de 0 a 50.51 grados en la mitad del tiempo (Figura 5.14b).

Dedo Medio

El recorrido realizado de este dedo para el movimiento solicitado, es exactamente igual que el de los dedos anular y medio. Los resultados obtenidos de la simulación se encuentran en la Figura 5.15 donde se puede apreciar que el recorrido de la falange distal (Figura 5.15a) presenta una alta similitud con los recorridos de los dedos anular y medio en la falange distal que se muestran en la sección anterior con una rotación de 174.6 grados de la misma manera que para la falange proximal, la similitud entre los recorridos son muy cercanas con una rotación de 87.32 grados para el dedo medio.



(a) Gráfica de movimiento falange distal eje X. (b) Gráfica de movimiento falange distal eje Y.



(c) Gráfica de movimiento falange distal eje Z.

Figura 5.13: Gráficas de movimiento del dedo pulgar.

Finalmente, en la Tabla 3 se encuentra el resumen de las rotaciones de cada falange para cada uno de los movimientos realizados comprobando así que se generan de manera adecuada cada una de las posiciones solicitadas desde la interfaz de Matlab.

5.4. Evaluación general

Una vez se han integrado todos los subsistemas diseñados, en esta prueba se verificará la integración completa y funcional de ellos. Para esta prueba se hará uso del script de integración realizado en el capítulo anterior (Sección 4.3.3). Para realizar esta prueba se siguieron los siguientes pasos:

1. Abrir el script de integración (*ClasificacionActivaFinal.m*).
2. Abrir la GUI para la lectura de datos.
3. Cargar los datos de validación en la GUI a través de la lectura de datos por medio de archivos (*PLAYBACK (from file)*).

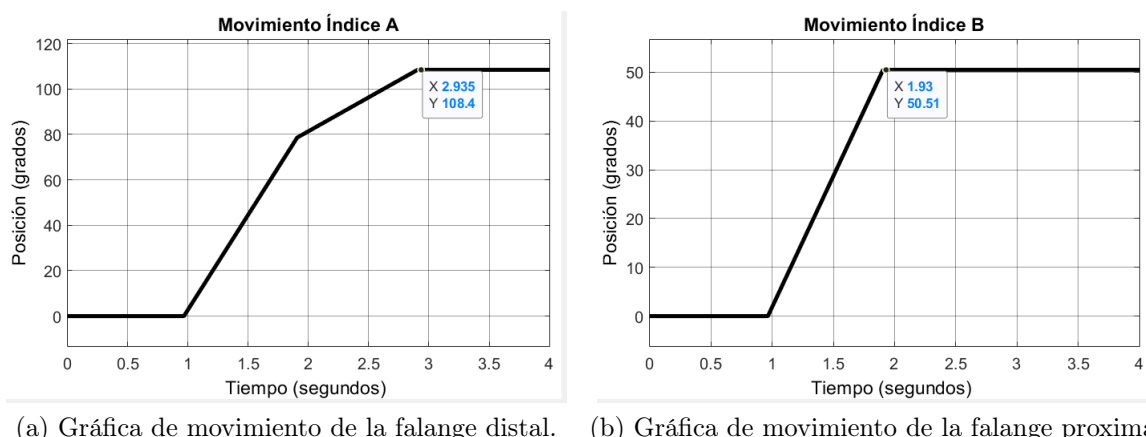


Figura 5.14: Gráficas de movimiento del dedo índice.

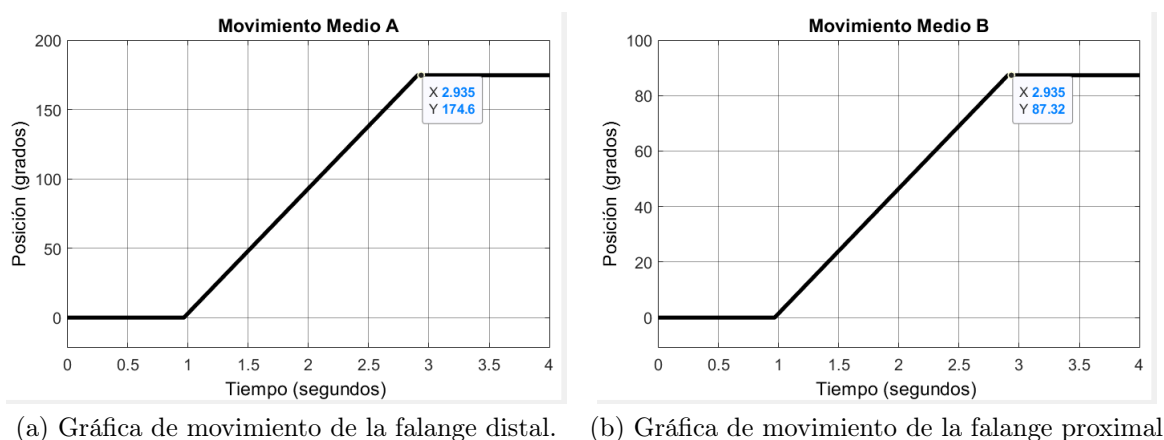


Figura 5.15: Gráficas de movimiento del dedo medio.

4. Iniciar la comunicación por LSL desde la GUI.
5. Iniciar la captura de datos EEG desde la GUI
6. Ejecutar el script desde Matlab.

Una vez realizado lo anterior, se espera la ejecución del programa. Se realizaron 3 pruebas de ejecución integral para validar la integración de todos los subsistemas así como la ejecución correcta de las simulaciones. En la Figura 5.16 se muestra una de las pruebas realizadas en donde se aprecia el correcto funcionamiento del sistema tanto de captura como de simulación.

En cada una de las pruebas realizadas se verificó la correcta clasificación de las señales así como el correcto desarrollo de los movimientos a realizar. Para cada una de las pruebas, se definió la postura a realizar dependiendo de la frecuencia identificada por el clasificador. Para

Tabla 5.1: Resumen de rotaciones de cada falange para cada movimiento

	Recorrido Final de cada falange en grados (A:distal B:Proximal)									
Postura	PA	PB	IA	IB	MA	MB	AA	AB	MqA	MqB
Saludo	58.51	0	0	0	0	0	174.2	87.12	173.8	86.92
Pinza Fina	7.319 (X)	53.36	108.4	50.51	174.6	87.32	174.2	87.12	173.8	86.92
	53.36 (Y)									
	13.61 (Z)									

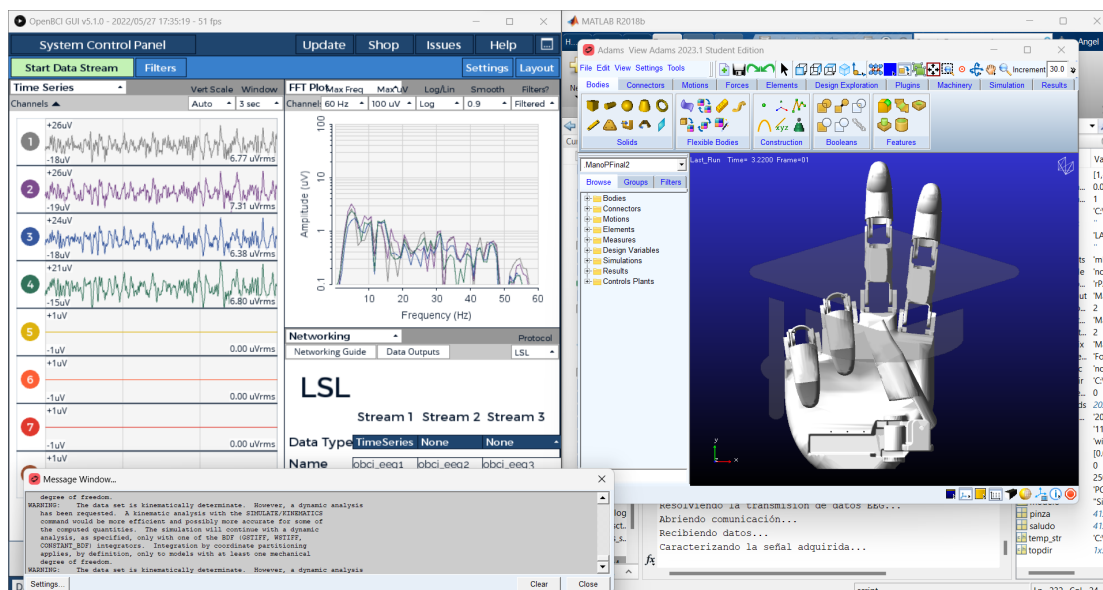


Figura 5.16: Prueba de integración de todos los subsistemas.

la postura de "pinza", se asignó a la frecuencia de 31 Hz (salida 0 del clasificador) mientras que para la postura de "saludo", se asignó a la frecuencia de 33 Hz (salida 1 del clasificador) siendo así que de acuerdo a la frecuencia identificada dentro de la señal EEG aplicada al sistema sería la postura a realizar por la mano virtual. De la misma manera se propuso que al identificarse dos veces la misma frecuencia, se realizara el regreso de la mano a su posición inicial para poder realizar una nueva postura esto con el fin de no limitar la postura a un tiempo determinado por software sino estar a libre uso del usuario.

Con la prueba anterior, se validó la integración completa y funcional de todos los subsistemas diseñados, así mismo se determinó que el tiempo de ejecución final para cada prueba tiene un promedio de 1-2 minutos debido a la alta carga computacional que se crea entre la adquisición de datos, caracterización y simulación. Debe tomarse en cuenta que las ejecuciones de cada prueba pueden variar dependiendo de las características del equipo en donde se realicen las pruebas. Las características de la unidad de procesamiento utilizada son:

- ✓ CPU: Ryzen 7 4800.
- ✓ GPU: NVIDIA GeForce GTX 1650 Ti.
- ✓ Memoria RAM: 8 GB.

Capítulo 6

Conclusiones y Trabajos Futuros

En este capítulo se presentan las conclusiones a las que se llegó tras desarrollar y evaluar los sistemas que componen la BCI como lo son la adquisición de señales, procesamiento, clasificación así como la conexión con la mano robótica virtual. Finalmente, se brindan sugerencias para mejorar el desempeño de la BCI y sus sistemas.

6.1. Conclusiones

En este proyecto de tesis se desarrolló una interfaz BCI basada en EEG para el control de una mano robótica utilizando el dispositivo de código abierto comercial OpenBCI para la adquisición de las señales EEG y su integración con el programa de Matlab para el procesamiento de las señales, cálculo de características y clasificación on-line.

Se desarrolló un sistema de limpieza y caracterización de las señales EEG obtenidas a través de un filtro pasa bandas tipo Butterworth el cual se encargaba de reducir el ruido inherente a la adquisición de las señales a través de electrodos superficiales para cuero cabelludo así como reducir y mitigar los artefactos involucrados en la señal EEG obtenida.

El paradigma utilizado para la adquisición de las señales EEG fue el de SSVEP siendo este un paradigma que requiere menos tiempo de entrenamiento con el paciente comparado con sus homólogos IM y P300. Además, el paradigma SSVEP permite implementar una mayor cantidad de comandos ya que utiliza un sistema de estimulación visual que, para el desarrollo del proyecto, fue una matriz LED debido a la versatilidad, bajo costo, simpleza y fácil implementación en sistemas de asistencia de tamaño reducido.

Las frecuencias de estimulación utilizadas en el proyecto comprenden 3 bandas de interés: la banda de mayor resonancia según la literatura, la banda media de frecuencias y la banda alta de frecuencias de estimulación visual. Para la banda de mayor resonancia según la literatura, se seleccionaron las frecuencias de 8 y 12 Hz, para la banda media, se seleccionaron las frecuencias de 12 y 15 Hz, finalmente para la banda de frecuencias altas, las frecuencias de 31 y 33 Hz fueron las implementadas. El objetivo de seleccionar 3 pares de frecuencias

localizadas en diferentes bandas de frecuencias es para poder observar el comportamiento de las señales adquiridas y comparar la facilidad de clasificación de cada uno de ellos. Tomando en cuenta los resultados obtenidos en el capítulo anterior, los pares de frecuencias con mejor rendimiento fueron las de 8 y 12 Hz y la de 31 y 33 Hz puesto que en ambas, los sistemas de clasificación implementados obtuvieron los mejores desempeños.

Se implementaron dos sistemas de clasificación para las señales EEG: una ANN y una SVM, siendo estos los sistemas que mayormente se utilizan dentro de los sistemas BCI revisados en la literatura. La implementación de ambos sistemas permitió definir el modelo que mejor se ajusta a las características del proyecto siendo el modelo de SVM perteneciente a las frecuencias de 31 y 33 Hz el que mejor se adecua debido a la facilidad de entrenamiento y el menor costo computacional requerido frente a la ANN que obtuvo el mismo porcentaje de precisión en la validación on-line del 95 %. A través del análisis de desempeño de la clasificación de la BCI mostrado en la Sección 5.2 se pudo observar que al realizar modificaciones al sistema de clasificación se pudo obtener un porcentaje de clasificación del 100 % para las 8 pruebas realizadas en clasificación on-line.

En un inicio, se había planteado el reacondicionamiento de la mano robótica realizada en [31] misma que se encuentra en la Universidad y que está disponible para su utilización sin embargo, al momento de realizar las adecuaciones y pruebas necesarias para la utilización dentro del proyecto, se detectaron algunos detalles técnicos, de construcción e integración que no permitieron el uso de la mano robótica de manera física. Tomando en cuenta que las limitaciones y objetivos planteados dentro de este proyecto de tesis, se optó por la utilización de una mano robótica virtual.

Se reacondicionó la mano robótica propuesta por Krauz en [32] de manera virtual con el objetivo de generar una retroalimentación visual hacia el usuario. Se determinaron las posturas a ejecutar con la mano robótica virtual siendo saludo y pinza fina los movimientos seleccionados puesto que cada uno representa un comando útil para la vida diaria de cualquier paciente. Además, se generó la planta del modelo virtual para poder realizar la comunicación con Matlab integrando las entradas que accionan cada una de las falanges de los dedos así como las salidas que representan los recorridos realizados por cada falange a lo largo del tiempo tal como se ve en la Sección 5.3.

Finalmente se comprobó que la implementación de un sistema BCI basado en EEG utilizando el paradigma SSVEP permite controlar una mano robótica virtual obteniendo resultados aceptables en la adquisición al permitir obtener datos de la tarjeta de código abierto OpenBCI por medio del protocolo de transmisión de datos LSL, procesamiento de las señales reduciendo el ruido integrado en la señal EEG adquirida así como la mitigación de artefactos además de una correcta caracterización y segmentación de datos tanto para entrenamiento como para las pruebas on-line, así como en la clasificación pues en esta última siendo la más importante, se obtuvieron resultados favorables con un 92.3 % en la la clasificación on-line solo para el modelo de clasificación mientras que para la adecuación implementada como parte de la BCI, se obtuvieron resultados del 100 % en la clasificación tomando en cuenta que el número de muestras para este último son comparativamente bajas a las muestras utilizadas para el

análisis de desempeño del modelo de clasificación SVM.

6.2. Trabajos futuros

Con el fin de darle seguimiento a la interfaz cerebro-computadora desarrollada en esta tesis, se realizan las siguientes propuestas de mejora.

- ✓ Implementación de la interfaz cerebro-computadora en personas con discapacidades motoras y adecuar la cantidad de frecuencias y gestos a realizar de acuerdo a las necesidades del individuo.
- ✓ Rediseño del casco de adquisición de señales EEG para un mejor posicionamiento de los electrodos.
- ✓ Construcción de un nuevo casco con materiales flexibles que permitan adecuarse a las distintas formas y tamaños de cráneos de los pacientes.
- ✓ Implementar algoritmos de mejoramiento de señales EEG así como mejorar los protocolos de adquisición de las señales.
- ✓ Realizar pruebas de desempeño de la interfaz cerebro-computadora en un sistema protésico de mano robótica adecuando las salidas obtenidas para el nuevo dispositivo.

Apéndice A

Documentación para la extracción de las señales EEG

Cuestionario rápido para evaluación general de salud

Código: _____

Fecha: _____

	SÍ	NO
Ha padecido de algún episodio epiléptico de 10 años a la fecha.		
Ha sufrido de algún accidente con lesión craneoencefálica.		
Padece de alguna enfermedad cardiovascular ejemplo hipertensión arterial, uso de marcapasos, entre otros.		
Se encuentra en tratamiento con medicamentos ansiolíticos, tranquilizantes o analgésicos opiáceos.		
Padece de algún tipo de ceguera total o parcial o deficiencia visual grave		
Ha sufrido de algún ACV o apoplejía		



Universidad de la Sierra Sur

COMITÉ DE ÉTICA

Formato muestra de la Carta de Consentimiento Informado

CARTA DE CONSENTIMIENTO INFORMADO

Título de proyecto: DESARROLLO DE UNA INTERFAZ CEREBRO-COMPUTADORA BASADA EN EEG PARA CONTROLAR UNA MANO ROBÓTICA

Nombre del Investigador Principal: Ángel del Rosario Zárate Ruiz

Folio de registro del proyecto: _____

Fecha aprobación por el Comité de Ética: _____

Buenos días/tardes:

Su servidor Ángel del rosario Zárate Ruiz soy alumno en la Universidad Tecnológica de la Mixteca, usted ha sido invitado a participar en el presente proyecto de investigación, que se realizará en: Laboratorio de Postgrado

Si Usted decide participar en el estudio, es importante que considere la siguiente información. Siéntase libre de preguntar cualquier asunto que no le quede claro.

El propósito de la investigación es el desarrollo de una interfaz cerebro-computadora que permita, mediante la interpretación de las señales cerebrales, controlar un prototipo de mano robótica. A través de la integración de la interfaz con el prototipo, el usuario podrá realizar diversas tareas, permitiéndole ampliar su rango de movilidad y autonomía en la realización de tareas cotidianas.

Las interfaces cerebro computadora brindan la capacidad de potenciar, sustituir y restaurar las capacidades físicas y mentales del ser humano por lo que el desarrollo de esta interfaz está orientada a la resolución del problema de movilidad que poseen las personas con discapacidades motrices, específicamente en las manos.

Solicitamos su participación en este estudio puesto que cumple con los requisitos de elegibilidad establecidos, recordando que su participación es libre y voluntaria por lo que no habrá represalia ni daño hacia su persona y estado actual.

Su participación consistirá en:

- Visualizar un estímulo durante algunos segundos mientras se realiza la adquisición de su actividad cerebral mediante un casco de electroencefalografía. El estímulo será colocado frente a usted a aproximadamente 70 cm. Durante toda la sesión, usted deberá permanecer sentado en una posición cómoda. Mientas se realizan las pruebas, se le pedirá no realizar movimiento alguno y



Universidad de la Sierra Sur

COMITÉ DE ÉTICA

Formato muestra de la Carta de Consentimiento Informado

permanecer lo más enfocado posible, así mismo, después de cada prueba podrá relajarse y moverse para evitar cualquier incomodidad.

- La sesión durará alrededor de 20 minutos y abarcará varias pruebas en donde se repetirá la visualización del estímulo, pero variando la frecuencia del mismo.
- La entrevista será realizada en el lugar, día y hora determinada por los investigadores, tomando en cuenta la disponibilidad de horarios de los participantes. El lugar será el Laboratorio de Detección y Corrección de Fallas Industriales ubicado en el edificio de los Laboratorios de Postgrado.
- Para facilitar el análisis, esta sesión será grabada. En cualquier caso, usted podrá interrumpir la grabación en cualquier momento y retomarla cuando quiera.

Beneficios: No hay un beneficio directo por su participación en el estudio, sin embargo, con su participación estará contribuyendo al estudio de un problema que afecta la salud.

Confidencialidad: Toda la información que usted nos proporcione para el estudio será estrictamente confidencial, será utilizada únicamente por el equipo de investigación del proyecto y no estará disponible para ningún otro propósito. Usted quedará identificado(a) con un número y no con su nombre. Los resultados de este estudio serán publicados con fines científicos, pero se presentarán de tal manera que no podrán ser identificados los participantes.

Participación Voluntaria/Retiro: Su participación en este estudio es absolutamente voluntaria. Usted está en plena libertad de negarse a participar o de retirar su participación en cualquier momento. Su decisión de participar o no en el estudio no implicará ningún tipo de consecuencia para usted. Podrá solicitar también que se retiren sus muestras del estudio sin que ello implique ningún tipo de consecuencia.

Riesgos Potenciales: Los riesgos potenciales que implican su participación en este estudio son: Fatiga visual, sensación de incomodidad y hormigueo en el cuero cabelludo. Si alguna de las preguntas le hicieran sentir un poco incomodo(a), tiene el derecho de no responderla. En el remoto caso de que ocurriera algún daño como resultado de la investigación, los investigadores se comprometen a brindarle atención oportuna acorde al daño recibido siendo en primera instancia el servicio de enfermería con el que cuenta la universidad y, en caso de ser necesario, se remitirá a la unidad médica del IMSS más cercana por lo que durante toda su participación se le estará evaluando y realizando preguntas de control para garantizar su seguridad. Usted recibirá atención médica oportuna pero no recibirá pago o retribución alguna, salvo en caso de negligencia plenamente comprobada.



Universidad de la Sierra Sur

COMITÉ DE ÉTICA

Formato muestra de la Carta de Consentimiento Informado

Aviso de Privacidad: El responsable institucional del estudio es el Dr. Manuel Arias Montiel, quien tendrá bajo su resguardo la información y los datos personales que nos proporcione, los cuales serán protegidos conforme a lo dispuesto por la Ley General de Protección de Datos Personales en Posesión de Sujetos Obligados. Los datos personales que le solicitaremos serán utilizados exclusivamente para las finalidades expuestas en este documento. Usted puede solicitar la corrección de sus datos o que sus datos se eliminen de nuestras bases o retirar su consentimiento para su uso.

Contacto institucional: Si usted tiene alguna pregunta, comentario o preocupación con respecto al proyecto, por favor comuníquese con el investigador responsable del proyecto: Dr. Manuel Arias Montiel al siguiente número de teléfono 953 53 202 14 ext. 759 o si lo prefiere puede escribir a la siguiente dirección de correo electrónico mam@mixteco.utm.mx

Si usted acepta participar en el estudio, le entregaremos una copia de este documento que le pedimos sea tan amable de firmar después de darle lectura al siguiente Texto:

- Yo _____ hago constar que se me ha leído la presente Carta de Consentimiento Informado, por medio de la cual me han explicado el proyecto de investigación incluyendo el objetivo, los posibles riesgos y beneficios, y otros aspectos sobre mi participación en el estudio. También hago constar que he podido hacer preguntas relacionadas a mi participación en el estudio, y me han respondido satisfactoriamente mis dudas.

Por lo tanto, ACEPTO participar voluntariamente en este estudio y, por este medio, AUTORIZO que la información y los datos que sean proporcionados puedan ser difundidos y publicados, con EXCEPCIÓN de los datos personales.

PARTICIPANTE:

Nombre: _____

Firma: _____

TESTIGO 1:

Nombre: _____

Firma: _____



Universidad de la Sierra Sur

COMITÉ DE ÉTICA

Formato muestra de la Carta de Consentimiento Informado

TESTIGO 2:

Nombre: _____

Firma: _____

Nombre y firma del investigador o persona que obtiene el Consentimiento Informado:

Nombre: _____

Firma: _____

Constancia de entrega de una copia al Participante:

Nombre de quien recibe: _____ Firma: _____

Lugar _____ . Fecha: a ____ de _____ de _____

Censo demográfico simplificado

Edad: _____ Sexo: M F Otro: _____ Semestre: _____

Carrera: _____

¿Practica algún deporte? Sí No ¿Cuál? _____



Universidad de la Sierra Sur

COMITÉ DE ÉTICA

Asunto: Dictamen

Folio número: CEI-01/2023.

C. ÁNGEL DEL ROSARIO ZÁRATE RUIZ (Tesisista)
DR. MANUEL ARIAS MONTIEL (Director)
UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA
PRESENTE:

La Comisión de Ética en Investigación de la Universidad de la Sierra Sur, es un órgano colegiado, constituido por acuerdo del H. Consejo Académico en sesión extraordinaria de fecha 27 de febrero de 2017, que tiene como propósito evaluar y dictaminar los protocolos de investigación en seres humanos desde el punto de vista ético y salvaguardar la dignidad, derechos y seguridad de las personas involucradas en el proceso; así como vigilar la aplicación de la legislación en materia de salud, la normatividad vigente, los lineamientos y los criterios que establezca la Comisión Nacional de Bioética (CONBIOÉTICA) en la materia.

Con relación a la solicitud de revisión del protocolo de investigación titulado "*Desarrollo de una interfaz cerebro-computadora basada en EEG para controlar una mano robótica*" recibida por esta comisión. Con base y fundamento en el artículo 109 del Reglamento de la Ley General de Salud en Materia de Investigación para la Salud y con apego a los principios de transparencia, neutralidad, legalidad y profesionalismo, ésta Comisión, con fecha 07 de febrero de 2022 realizó la revisión y valoración del proyecto de investigación, emitiendo el siguiente dictamen:

APROBATORIO

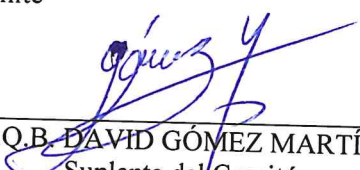
Con una vigencia de dos años, comprendido del 07 de Febrero de 2023 al 05 de Febrero de 2025. El periodo de vigencia podrá ser prorrogado a solicitud de los interesados y al concluir el proyecto el responsable institucional deberá entregar un reporte final.

Se extiende el presente dictamen a solicitud de los interesados para los usos y fines descritos con anterioridad, en la Ciudad de Miahuatlán de Porfirio Díaz Oaxaca con fecha 07 de febrero de 2023.

ATENTAMENTE


DRA. ROXANA GRISEL CÁSAREZ SANTIAGO
Presidenta del Comité


L.E. SILVIA MERCEDES COCA
Vocal del Comité


M.C.Q.B. DAVID GÓMEZ MARTÍNEZ
Suplente del Comité

Apéndice B

Script utilizado para el pre-procesamiento
y caracterización de la señal EEG

```

clear all;
%Frecuencia de muestreo de la señal
fs = 250;
%%Filtro butterworth
[B,A] = butter(2,[26 40]/(fs/2));
%Base del nombre del archivo
filenameBase = 'L_33_S';
extension = '.csv';
numArc = 6;
for k = 1:numArc
    %Lectura del archivo RAW EEG en una sola ventana de tiempo
    filename = strcat(filenameBase,int2str(k));
    filename = strcat(filename,extension);
    file = fopen(filename);
    out=textscan(file, '%f%f%f%f', 'delimiter', ',',');
    %v1 = out{2};
    fclose(file);
    REEGc1 = out{2};
    REEGc2 = out{3};
    REEGc3 = out{4};
    REEGc4 = out{5};
    clear out;
    %Filtrado de las señales
    %Canal 1
    REEGc1 = filtfilt(B,A,REEGc1);
    %Canal 2
    REEGc2 = filtfilt(B,A,REEGc2);
    %Canal 3
    REEGc3 = filtfilt(B,A,REEGc3);
    %Canal 4
    REEGc4 = filtfilt(B,A,REEGc4);
    %Corte de la ventana total en 3 secciones: Descanso (15s), Actividad
    % (15s), Especial (segundo del 13.5 al 16.5)
    %Corte para descanso
    DEEGc1 = REEGc1(1:3750);
    DEEGc2 = REEGc2(1:3750);
    DEEGc3 = REEGc3(1:3750);
    DEEGc4 = REEGc4(1:3750);
    %Corte para actividad
    AEEGc1 = REEGc1(3751:7500);
    AEEGc2 = REEGc2(3751:7500);
    AEEGc3 = REEGc3(3751:7500);
    AEEGc4 = REEGc4(3751:7500);
    %Corte para especial del segundo 13.5 al 16.5
    SEEGc1 = REEGc1(3376:4125);
    SEEGc2 = REEGc2(3376:4125);
    SEEGc3 = REEGc3(3376:4125);
    SEEGc4 = REEGc4(3376:4125);
    clear REEGc1;
    clear REEGc2;
    clear REEGc3;
    clear REEGc4;
    %Ventaneo de cada las señales
    ven = 750; %Tamaño de ventana equivalente a 3s
    for m = 0:3
        %Canal 1
        DV1c1 = DEEGc1((ven*m+1):(ven*m+750));
        DV2c1 = DEEGc1((ven*m+151):(ven*m+900));
        DV3c1 = DEEGc1((ven*m+301):(ven*m+1050));
        DV4c1 = DEEGc1((ven*m+451):(ven*m+1200));
        DV5c1 = DEEGc1((ven*m+601):(ven*m+1350));
        DVc1(m*5+1,:)= DV1c1;
        DVc1(m*5+2,:)= DV2c1;
        DVc1(m*5+3,:)= DV3c1;
        DVc1(m*5+4,:)= DV4c1;
        DVc1(m*5+5,:)= DV5c1;
        %Canal 2
        DV1c2 = DEEGc2((ven*m+1):(ven*m+750));
        DV2c2 = DEEGc2((ven*m+151):(ven*m+900));
        DV3c2 = DEEGc2((ven*m+301):(ven*m+1050));
        DV4c2 = DEEGc2((ven*m+451):(ven*m+1200));
        DV5c2 = DEEGc2((ven*m+601):(ven*m+1350));
        DVc2(m*5+1,:)= DV1c2;
        DVc2(m*5+2,:)= DV2c2;
        DVc2(m*5+3,:)= DV3c2;
        DVc2(m*5+4,:)= DV4c2;
        DVc2(m*5+5,:)= DV5c2;
        %Canal 3
        DV1c3 = DEEGc3((ven*m+1):(ven*m+750));

```

```

DV2c3 = DEEGc3((ven*m+151):(ven*m+900));
DV3c3 = DEEGc3((ven*m+301):(ven*m+1050));
DV4c3 = DEEGc3((ven*m+451):(ven*m+1200));
DV5c3 = DEEGc3((ven*m+601):(ven*m+1350));
DVc3(m*5+1,:)= DV1c3;
DVc3(m*5+2,:)= DV2c3;
DVc3(m*5+3,:)= DV3c3;
DVc3(m*5+4,:)= DV4c3;
DVc3(m*5+5,:)= DV5c3;
%Canal 4
DV1c4 = DEEGc4((ven*m+1):(ven*m+750));
DV2c4 = DEEGc4((ven*m+151):(ven*m+900));
DV3c4 = DEEGc4((ven*m+301):(ven*m+1050));
DV4c4 = DEEGc4((ven*m+451):(ven*m+1200));
DV5c4 = DEEGc4((ven*m+601):(ven*m+1350));
DVc4(m*5+1,:)= DV1c4;
DVc4(m*5+2,:)= DV2c4;
DVc4(m*5+3,:)= DV3c4;
DVc4(m*5+4,:)= DV4c4;
DVc4(m*5+5,:)= DV5c4;
end
clear DV1c1 DV2c1 DV3c1 DV4c1 DV5c1;
clear DV1c2 DV2c2 DV3c2 DV4c2 DV5c2;
clear DV1c3 DV2c3 DV3c3 DV4c3 DV5c3;
clear DV1c4 DV2c4 DV3c4 DV4c4 DV5c4;
DVc1 = DVc1.';
DVc2 = DVc2.';
DVc3 = DVc3.';
DVc4 = DVc4.';
for l = 0:3
    %Canal 1
    AV1c1 = AEEGc1((ven*l+1):(ven*l+750));
    AV2c1 = AEEGc1((ven*l+151):(ven*l+900));
    AV3c1 = AEEGc1((ven*l+301):(ven*l+1050));
    AV4c1 = AEEGc1((ven*l+451):(ven*l+1200));
    AV5c1 = AEEGc1((ven*l+601):(ven*l+1350));
    AVc1(l*5+1,:)= AV1c1;
    AVc1(l*5+2,:)= AV2c1;
    AVc1(l*5+3,:)= AV3c1;
    AVc1(l*5+4,:)= AV4c1;
    AVc1(l*5+5,:)= AV5c1;
    %Canal 2
    AV1c2 = AEEGc2((ven*l+1):(ven*l+750));
    AV2c2 = AEEGc2((ven*l+151):(ven*l+900));
    AV3c2 = AEEGc2((ven*l+301):(ven*l+1050));
    AV4c2 = AEEGc2((ven*l+451):(ven*l+1200));
    AV5c2 = AEEGc2((ven*l+601):(ven*l+1350));
    AVc2(l*5+1,:)= AV1c2;
    AVc2(l*5+2,:)= AV2c2;
    AVc2(l*5+3,:)= AV3c2;
    AVc2(l*5+4,:)= AV4c2;
    AVc2(l*5+5,:)= AV5c2;
    %Canal 3
    AV1c3 = AEEGc3((ven*l+1):(ven*l+750));
    AV2c3 = AEEGc3((ven*l+151):(ven*l+900));
    AV3c3 = AEEGc3((ven*l+301):(ven*l+1050));
    AV4c3 = AEEGc3((ven*l+451):(ven*l+1200));
    AV5c3 = AEEGc3((ven*l+601):(ven*l+1350));
    AVc3(l*5+1,:)= AV1c3;
    AVc3(l*5+2,:)= AV2c3;
    AVc3(l*5+3,:)= AV3c3;
    AVc3(l*5+4,:)= AV4c3;
    AVc3(l*5+5,:)= AV5c3;
    %Canal 4
    AV1c4 = AEEGc4((ven*l+1):(ven*l+750));
    AV2c4 = AEEGc4((ven*l+151):(ven*l+900));
    AV3c4 = AEEGc4((ven*l+301):(ven*l+1050));
    AV4c4 = AEEGc4((ven*l+451):(ven*l+1200));
    AV5c4 = AEEGc4((ven*l+601):(ven*l+1350));
    AVc4(l*5+1,:)= AV1c4;
    AVc4(l*5+2,:)= AV2c4;
    AVc4(l*5+3,:)= AV3c4;
    AVc4(l*5+4,:)= AV4c4;
    AVc4(l*5+5,:)= AV5c4;
end
clear AV1c1 AV2c1 AV3c1 AV4c1 AV5c1;
clear AV1c2 AV2c2 AV3c2 AV4c2 AV5c2;
clear AV1c3 AV2c3 AV3c3 AV4c3 AV5c3;
clear AV1c4 AV2c4 AV3c4 AV4c4 AV5c4;

```

```

AVc1 = AVc1.';
AVc2 = AVc2.';
AVc3 = AVc3.';
AVc4 = AVc4.';

%Cálculo del Valor Medio Absoluto
%Actividad
AVMac1 = sum(abs(AVc1))/size(AVc1,1);
AVMac2 = sum(abs(AVc2))/size(AVc2,1);
AVMac3 = sum(abs(AVc3))/size(AVc3,1);
AVMac4 = sum(abs(AVc4))/size(AVc4,1);
%Descanso
DVMac1 = sum(abs(DVc1))/size(DVc1,1);
DVMac2 = sum(abs(DVc2))/size(DVc2,1);
DVMac3 = sum(abs(DVc3))/size(DVc3,1);
DVMac4 = sum(abs(DVc4))/size(DVc4,1);
%Cálculo de la Integral Cuadrada Simple
%Actividad
for p=1:20
    auxICSc1 = abs(AVc1(:,p));
    auxICSc2 = abs(AVc2(:,p));
    auxICSc3 = abs(AVc3(:,p));
    auxICSc4 = abs(AVc4(:,p));
    for q = 1:750
        auxICSc1(q) = auxICSc1(q)^2;
        auxICSc2(q) = auxICSc2(q)^2;
        auxICSc3(q) = auxICSc3(q)^2;
        auxICSc4(q) = auxICSc4(q)^2;
    end
    AICSc1(p) = sum(auxICSc1);
    AICSc2(p) = sum(auxICSc2);
    AICSc3(p) = sum(auxICSc3);
    AICSc4(p) = sum(auxICSc4);
end
%Descanso
for p=1:20
    auxICSc1 = abs(DVc1(:,p));
    auxICSc2 = abs(DVc2(:,p));
    auxICSc3 = abs(DVc3(:,p));
    auxICSc4 = abs(DVc4(:,p));
    for q = 1:750
        auxICSc1(q) = auxICSc1(q)^2;
        auxICSc2(q) = auxICSc2(q)^2;
        auxICSc3(q) = auxICSc3(q)^2;
        auxICSc4(q) = auxICSc4(q)^2;
    end
    DICSc1(p) = sum(auxICSc1);
    DICSc2(p) = sum(auxICSc2);
    DICSc3(p) = sum(auxICSc3);
    DICSc4(p) = sum(auxICSc4);
end
clear auxICSc1 auxICSc2 auxICSc3 auxICSc4;

%Longitud de forma de onda
%Actividad
aux_restac1 = 0;
aux_sumac1 = 0;
aux_restac2 = 0;
aux_sumac2 = 0;
aux_restac3 = 0;
aux_sumac3 = 0;
aux_restac4 = 0;
aux_sumac4 = 0;
for p=1:20 %1 a la cantidad de ventanas
    for q=1:(749) %1 a la cantidad de datos -1
        aux_restac1 = abs(AVc1(q+1,p) - AVc1(q,p));
        aux_sumac1 = aux_sumac1 + aux_restac1;
        aux_restac2 = abs(AVc2(q+1,p) - AVc2(q,p));
        aux_sumac2 = aux_sumac2 + aux_restac2;
        aux_restac3 = abs(AVc3(q+1,p) - AVc3(q,p));
        aux_sumac3 = aux_sumac3 + aux_restac3;
        aux_restac4 = abs(AVc4(q+1,p) - AVc4(q,p));
        aux_sumac4 = aux_sumac4 + aux_restac4;
    end
    AWFLc1(p) = aux_sumac1;
    aux_sumac1 = 0;
    aux_restac1 = 0;
    AWFLc2(p) = aux_sumac2;
    aux_sumac2 = 0;

```

```

    aux_restac2 = 0;
    AWFLc3(p) = aux_sumac3;
    aux_sumac3 = 0;
    aux_restac3 = 0;
    AWFLc4(p) = aux_sumac4;
    aux_sumac4 = 0;
    aux_restac4 = 0;
end
%Descanso
aux_restac1 = 0;
aux_sumac1 = 0;
aux_restac2 = 0;
aux_sumac2 = 0;
aux_restac3 = 0;
aux_sumac3 = 0;
aux_restac4 = 0;
aux_sumac4 = 0;
for p=1:20 %1 a la cantidad de ventanas
    for q=1:(749) %1 a la cantidad de datos -1
        aux_restac1 = abs(DVc1(q+1,p) - DVc1(q,p));
        aux_sumac1 = aux_sumac1 + aux_restac1;
        aux_restac2 = abs(DVc2(q+1,p) - DVc2(q,p));
        aux_sumac2 = aux_sumac2 + aux_restac2;
        aux_restac3 = abs(DVc3(q+1,p) - DVc3(q,p));
        aux_sumac3 = aux_sumac3 + aux_restac3;
        aux_restac4 = abs(DVc4(q+1,p) - DVc4(q,p));
        aux_sumac4 = aux_sumac4 + aux_restac4;
    end
    DWFLc1(p) = aux_sumac1;
    aux_sumac1 = 0;
    aux_restac1 = 0;
    DWFLc2(p) = aux_sumac2;
    aux_sumac2 = 0;
    aux_restac2 = 0;
    DWFLc3(p) = aux_sumac3;
    aux_sumac3 = 0;
    aux_restac3 = 0;
    DWFLc4(p) = aux_sumac4;
    aux_sumac4 = 0;
    aux_restac4 = 0;
end
clear aux_sumac1 aux_sumac2 aux_sumac3 aux_sumac4;
clear aux_restac1 aux_restac2 aux_restac3 aux_restac4;

%Transformada Rapida de Fourier para obtener las frecuencias presentes
%en el EEG
%Actividad
for t=1:20
    ftc1=fft(AVc1(:,t));
    ftc2=fft(AVc2(:,t));
    ftc3=fft(AVc3(:,t));
    ftc4=fft(AVc4(:,t));
    %Se toma la mitad de la FFT puesto que es simétrica
    ptfc1 = abs(ftc1(79:121));
    ptfc2 = abs(ftc2(79:121));
    ptfc3 = abs(ftc3(79:121));
    ptfc4 = abs(ftc4(79:121));
    %Se localiza el índice en donde se encuentra el máximo de la FFT y se
    %transforma a frecuencia
    Frec1 = (find(ptfc1==max(ptfc1))+77)*125/375;
    Frec2 = (find(ptfc2==max(ptfc2))+77)*125/375;
    Frec3 = (find(ptfc3==max(ptfc3))+77)*125/375;
    Frec4 = (find(ptfc4==max(ptfc4))+77)*125/375;
    AFc1(t) = Frec1;
    AFc2(t) = Frec2;
    AFc3(t) = Frec3;
    AFc4(t) = Frec4;
    %Se toma el valor en potencia de la FFT
    Frec1 = find(ptfc1==max(ptfc1));
    Frec2 = find(ptfc2==max(ptfc2));
    Frec3 = find(ptfc3==max(ptfc3));
    Frec4 = find(ptfc4==max(ptfc4));
    APScl(t) = ptfc1(Frec1);
    APSc2(t) = ptfc1(Frec2);
    APSc3(t) = ptfc1(Frec3);
    APSc4(t) = ptfc1(Frec4);
end
%Descanso
for t=1:20

```

```

ftc1=fft(DVc1(:,t));
ftc2=fft(DVc2(:,t));
ftc3=fft(DVc3(:,t));
ftc4=fft(DVc4(:,t));
%Se toma la mitad de la FFT puesto que es simétrica
ptfc1 = abs(ftc1(79:121));
ptfc2 = abs(ftc2(79:121));
ptfc3 = abs(ftc3(79:121));
ptfc4 = abs(ftc4(79:121));
%Se localiza el índice en donde se encuentra el máximo de la FFT y se
%transforma a frecuencia
Frec1 = (find(ptfc1==max(ptfc1))+77)*125/375;
Frec2 = (find(ptfc2==max(ptfc2))+77)*125/375;
Frec3 = (find(ptfc3==max(ptfc3))+77)*125/375;
Frec4 = (find(ptfc4==max(ptfc4))+77)*125/375;
DFc1(t) = Frec1;
DFc2(t) = Frec2;
DFc3(t) = Frec3;
DFc4(t) = Frec4;
%Se toma el valor en potencia de la FFT
Frec1 = find(ptfc1==max(ptfc1));
Frec2 = find(ptfc2==max(ptfc2));
Frec3 = find(ptfc3==max(ptfc3));
Frec4 = find(ptfc4==max(ptfc4));
DPSc1(t) = ptfc1(Frec1);
DPSc2(t) = ptfc1(Frec2);
DPSc3(t) = ptfc1(Frec3);
DPSc4(t) = ptfc1(Frec4);
end
%Calculo de la Entropía de Shannon
%Actividad
for t=1:20
    auxc1 = AVc1(:,t);
    auxc2 = AVc2(:,t);
    auxc3 = AVc3(:,t);
    auxc4 = AVc4(:,t);
    %Calcular la distribución de la probabilidad relativa
    probc1 = histcounts(auxc1, 'Normalization', 'probability');
    probc2 = histcounts(auxc2, 'Normalization', 'probability');
    probc3 = histcounts(auxc3, 'Normalization', 'probability');
    probc4 = histcounts(auxc4, 'Normalization', 'probability');
    idx1 = find(probc1==0);
    idx2 = find(probc2==0);
    idx3 = find(probc3==0);
    idx4 = find(probc4==0);
    probc1(idx1) = 1;
    probc2(idx2) = 1;
    probc3(idx3) = 1;
    probc4(idx4) = 1;
    %Calcular la Entropía
    ASEc1(t) = -sum(probc1.*log2(probc1));
    ASEc2(t) = -sum(probc2.*log2(probc2));
    ASEc3(t) = -sum(probc3.*log2(probc3));
    ASEc4(t) = -sum(probc4.*log2(probc4));
end
%Descanso
for t=1:20
    auxc1 = DVc1(:,t);
    auxc2 = DVc2(:,t);
    auxc3 = DVc3(:,t);
    auxc4 = DVc4(:,t);
    %Calcular la distribución de la probabilidad relativa
    probc1 = histcounts(auxc1, 'Normalization', 'probability');
    probc2 = histcounts(auxc2, 'Normalization', 'probability');
    probc3 = histcounts(auxc3, 'Normalization', 'probability');
    probc4 = histcounts(auxc4, 'Normalization', 'probability');
    idx1 = find(probc1==0);
    idx2 = find(probc2==0);
    idx3 = find(probc3==0);
    idx4 = find(probc4==0);
    probc1(idx1) = 1;
    probc2(idx2) = 1;
    probc3(idx3) = 1;
    probc4(idx4) = 1;
    %Calcular la Entropía
    DSEc1(t) = -sum(probc1.*log2(probc1));
    DSEc2(t) = -sum(probc2.*log2(probc2));
    DSEc3(t) = -sum(probc3.*log2(probc3));
    DSEc4(t) = -sum(probc4.*log2(probc4));
end

```

```

end
%Cálculo de los parámetros Hjorth
%Actividad
for t=1:20
    diffc1 = diff(AVc1(:,t));
    diff2c1 = diff(diffc1);
    varc1 = var(AVc1(:,t));
    var_diffc1 = var(diffc1);
    var_diff2c1 = var(diff2c1);
    AACTc1(t) = varc1;
    AMOBc1(t) = sqrt(var_diffc1 / varc1)*250/(2*pi);
    ACOMc1(t) = sqrt(var_diff2c1 / var_diffc1) / AMOBc1(t);
    diffc2 = diff(AVc2(:,t));
    diff2c2 = diff(diffc2);
    varc2 = var(AVc2(:,t));
    var_diffc2 = var(diffc2);
    var_diff2c2 = var(diff2c2);
    AACTc2(t) = varc2;
    AMOBc2(t) = sqrt(var_diffc2 / varc2)*250/(2*pi);
    ACOMc2(t) = sqrt(var_diff2c2 / var_diffc2) / AMOBc2(t);
    diffc3 = diff(AVc3(:,t));
    diff2c3 = diff(diffc3);
    varc3 = var(AVc3(:,t));
    var_diffc3 = var(diffc3);
    var_diff2c3 = var(diff2c3);
    AACTc3(t) = varc3;
    AMOBc3(t) = sqrt(var_diffc3 / varc3)*250/(2*pi);
    ACOMc3(t) = sqrt(var_diff2c3 / var_diffc3) / AMOBc3(t);
    diffc4 = diff(AVc4(:,t));
    diff2c4 = diff(diffc4);
    varc4 = var(AVc4(:,t));
    var_diffc4 = var(diffc4);
    var_diff2c4 = var(diff2c4);
    AACTc4(t) = varc4;
    AMOBc4(t) = sqrt(var_diffc4 / varc4)*250/(2*pi);
    ACOMc4(t) = sqrt(var_diff2c4 / var_diffc4) / AMOBc4(t);
end
%Descanso
for t=1:20
    diffc1 = diff(DVc1(:,t));
    diff2c1 = diff(diffc1);
    varc1 = var(DVc1(:,t));
    var_diffc1 = var(diffc1);
    var_diff2c1 = var(diff2c1);
    DACTc1(t) = varc1;
    DMOBc1(t) = sqrt(var_diffc1 / varc1)*250/(2*pi);
    DCOMc1(t) = sqrt(var_diff2c1 / var_diffc1) / DMOBc1(t);
    diffc2 = diff(DVc2(:,t));
    diff2c2 = diff(diffc2);
    varc2 = var(DVc2(:,t));
    var_diffc2 = var(diffc2);
    var_diff2c2 = var(diff2c2);
    DACTc2(t) = varc2;
    DMOBc2(t) = sqrt(var_diffc2 / varc2)*250/(2*pi);
    DCOMc2(t) = sqrt(var_diff2c2 / var_diffc2) / DMOBc2(t);
    diffc3 = diff(DVc3(:,t));
    diff2c3 = diff(diffc3);
    varc3 = var(DVc3(:,t));
    var_diffc3 = var(diffc3);
    var_diff2c3 = var(diff2c3);
    DACTc3(t) = varc3;
    DMOBc3(t) = sqrt(var_diffc3 / varc3)*250/(2*pi);
    DCOMc3(t) = sqrt(var_diff2c3 / var_diffc3) / DMOBc3(t);
    diffc4 = diff(DVc4(:,t));
    diff2c4 = diff(diffc4);
    varc4 = var(DVc4(:,t));
    var_diffc4 = var(diffc4);
    var_diff2c4 = var(diff2c4);
    DACTc4(t) = varc4;
    DMOBc4(t) = sqrt(var_diffc4 / varc4)*250/(2*pi);
    DCOMc4(t) = sqrt(var_diff2c4 / var_diffc4) / DMOBc4(t);
end
%Se integran los canales en una sola matriz
%Frecuencias de por TF
AF = [AFc1.' AFc2.' AFc3.' AFc4.'];
DF = [DFc1.' DFc2.' DFc3.' DFc4.'];
%Potencia de la TF
APS = [APSc1.' APSc2.' APSc3.' APSc4.'];
DPS = [DPSc1.' DPSc2.' DPSc3.' DPSc4.'];

```

```

%Parametros Hjorth
AACT = [AACTc1.' AACTc2.' AACTc3.' AACTc4.'];
AMOB = [AMOBc1.' AMOBc2.' AMOBc3.' AMOBc4.'];
ACOM = [ACOMc1.' ACOMc2.' ACOMc3.' ACOMc4.'];
DACT = [DACTc1.' DACTc2.' DACTc3.' DACTc4.'];
DMOB = [DMOBc1.' DMOBc2.' DMOBc3.' DMOBc4.'];
DCOM = [DCOMc1.' DCOMc2.' DCOMc3.' DCOMc4.'];
%Entropía de Shannon
ASE = [ASEc1.' ASEc2.' ASEc3.' ASEc4.'];
DSE = [DSEc1.' DSEc2.' DSEc3.' DSEc4.'];
%Longitud de forma de onda
AWFL = [AWFLc1.' AWFLc2.' AWFLc3.' AWFLc4.'];
DWFL = [DWFLc1.' DWFLc2.' DWFLc3.' DWFLc4.'];
%Integral cuadrada simple
AICS = [AICScl.' AICScl2.' AICScl3.' AICScl4.'];
DICS = [DICScl.' DICScl2.' DICScl3.' DICScl4.'];
%Valor medio absoluto
AVMA = [AVMAc1.' AVMAc2.' AVMAc3.' AVMAc4.'];
DVMA = [DVMAc1.' DVMAc2.' DVMAc3.' DVMAc4.'];
%Se integran todas las características en una sola matriz
if(k==1)
    AFeatures = [AF APS AACT AMOB ACOM ASE AWFL AICS AVMA];
    DFeatures = [DF DPS DACT DMOB DCOM DSE DWFL DICS DVMA];
    EFeatures = [ASE DSE];
else
    AFeatures = [AFeatures; AF APS AACT AMOB ACOM ASE AWFL AICS AVMA];
    DFeatures = [DFeatures; DF DPS DACT DMOB DCOM DSE DWFL DICS DVMA];
    EFeatures = [EFeatures; ASE DSE];
end
clear DVc1 DVc2 DVc3 DVc4;
clear AVc1 AVc2 AVc3 AVc4;
end
dlmwrite('AFeatures_L_E33C.csv', AFeatures, 'precision',11);
dlmwrite('DFeatures_L_E33C.csv', DFeatures, 'precision',11);
dlmwrite('TestFinal33.csv', AFeatures(101:120,:), 'precision',11);

```

Apéndice C

Script del modelo de clasificación utilizado

```
1  # Importar bibliotecas necesarias
2  import numpy as np
3  from pickle import load
4
5  def neural(feature):
6      svm = load(open('modeloSVM_Krbf_Final.pkl', 'rb'))
7      scaler = load(open('escaladorSVM_Krbf_Final.pkl', 'rb'))
8      lda = load(open('modeloSVM_LDA_Final.pkl', 'rb'))
9      feature = np.array([feature])
10     X = feature
11     X_scaled = scaler.transform(X)
12     X_transform = lda.transform(X_scaled)
13     movPredicted = svm.predict(X_transform)
14
15     return movPredicted
```

Apéndice D

Script de comunicación e integración de sistemas

```

clear

%%Cargar modulo de Adams
load("ParametrosAdams.mat");
load("posturasMano.mat");
modelo = "SimulacionManoFinal";
%open_system(modelo);

%%Creación del filtro
%Frecuencia de muestreo de la señal
fs = 250;
%Filtro butterworth
[B,A] = butter(2,[26 40]/(fs/2));

%%Conexión LSL
% instanciar la librería
disp('Cargando la librería...');
lib = lsl_loadlib();
% resolviendo la transmisión de datos
disp('Resolviendo la transmisión de datos EEG...');
result = {};
while isempty(result)
    result = lsl_resolve_byprop(lib, 'type', 'EEG');
end
% Creando un nuevo bus de entrada de datos
disp('Abriendo comunicación...');
inlet = lsl_inlet(result{1});
disp('Recibiendo datos...');

%%Inicio de recepción de datos
pruebas = 1;
while pruebas<=1
    contador = 1;
    v1=750;
    v2=900;
    v3=1050;
    while contador<=1050
        % obtener datos del bus de entrada
        [vec,ts] = inlet.pull_sample();
        % Para la ventana 1 de 750 elementos
        if(contador<=v1)
            REEGv1c1(contador) = vec(1);
            REEGv1c2(contador) = vec(2);
            REEGv1c3(contador) = vec(3);
            REEGv1c4(contador) = vec(4);
        end
        %Para la ventana 2
        if(contador>=151 && contador<=v2)
            REEGv2c1(contador-150) = vec(1);
            REEGv2c2(contador-150) = vec(2);
            REEGv2c3(contador-150) = vec(3);
            REEGv2c4(contador-150) = vec(4);
        end
        %Para la ventana 3
        if(contador>=301 && contador<=v3)
            REEGv3c1(contador-300) = vec(1);
            REEGv3c2(contador-300) = vec(2);
            REEGv3c3(contador-300) = vec(3);
            REEGv3c4(contador-300) = vec(4);
        end
        contador = contador+1;
    end

    disp('Caracterizando la señal adquirida...');

    %%Filtrado de la señal o preprocesamiento y corrección de vector 1x750
    %a 750x1
    %Ventana 1
    FEEGv1c1 = filtfilt(B,A,REEGv1c1.);
    FEEGv1c2 = filtfilt(B,A,REEGv1c2.);
    FEEGv1c3 = filtfilt(B,A,REEGv1c3.);
    FEEGv1c4 = filtfilt(B,A,REEGv1c4.);
    %Ventana 2
    FEEGv2c1 = filtfilt(B,A,REEGv2c1.);
    FEEGv2c2 = filtfilt(B,A,REEGv2c2.);
    FEEGv2c3 = filtfilt(B,A,REEGv2c3.);
    FEEGv2c4 = filtfilt(B,A,REEGv2c4.);
    %Ventana 3

```

```

FEEGv3c1 = filtfilt(B,A,REEGv3c1.);
FEEGv3c2 = filtfilt(B,A,REEGv3c2.);
FEEGv3c3 = filtfilt(B,A,REEGv3c3.);
FEEGv3c4 = filtfilt(B,A,REEGv3c4.);

%Integracion de los canales en una sola matriz de 750x4 para cada
%ventana
EEGv1 = [FEEGv1c1 FEEGv1c2 FEEGv1c3 FEEGv1c4];
EEGv2 = [FEEGv2c1 FEEGv2c2 FEEGv2c3 FEEGv2c4];
EEGv3 = [FEEGv3c1 FEEGv3c2 FEEGv3c3 FEEGv3c4];

%%Caracterización de las señales
% Valor Medio Absoluto VMA
VMAv1 = sum(abs(EEGv1))/size(EEGv1,1);
VMAv2 = sum(abs(EEGv2))/size(EEGv2,1);
VMAv3 = sum(abs(EEGv3))/size(EEGv3,1);

%Integral Cuadrada Simple ICS
auxICSv1 = EEGv1.^2;
auxICSv2 = EEGv2.^2;
auxICSv3 = EEGv3.^2;
ICSv1 = sum(auxICSv1);
ICSv2 = sum(auxICSv2);
ICSv3 = sum(auxICSv3);

%Longitud de Forma de Onda WFL
aux_restav1 = 0;
aux_sumav1 = 0;
aux_restav2 = 0;
aux_sumav2 = 0;
aux_restav3 = 0;
aux_sumav3 = 0;
for p=1:4 %1 a la cantidad de canales
    for q=1:(749) %1 a la cantidad de datos -1 (750-1=749)
        aux_restav1 = abs(EEGv1(q+1,p) - EEGv1(q,p));
        aux_sumav1 = aux_sumav1 + aux_restav1;
        aux_restav2 = abs(EEGv2(q+1,p) - EEGv2(q,p));
        aux_sumav2 = aux_sumav2 + aux_restav2;
        aux_restav3 = abs(EEGv3(q+1,p) - EEGv3(q,p));
        aux_sumav3 = aux_sumav3 + aux_restav3;
    end
    WFLv1(p) = aux_sumav1;
    aux_sumav1 = 0;
    aux_restav1 = 0;
    WFLv2(p) = aux_sumav2;
    aux_sumav2 = 0;
    aux_restav2 = 0;
    WFLv3(p) = aux_sumav3;
    aux_sumav3 = 0;
    aux_restav3 = 0;
end

%Transformada Rapida de Fourier para obtener las frecuencias presentes
%en el EEG
for t=1:4
    ftv1=fft(EEGv1(:,t));
    ftv2=fft(EEGv2(:,t));
    ftv3=fft(EEGv3(:,t));
    %Se toma la mitad de la FFT puesto que es simétrica y tomando la
    %banda de frecuencias de 26-40 Hz
    ptfv1 = abs(ftv1(79:121));
    ptfv2 = abs(ftv2(79:121));
    ptfv3 = abs(ftv3(79:121));
    %Se localiza el índice en donde se encuentra el máximo de la FFT y se
    %transforma a frecuencia
    Frec1 = (find(ptfv1==max(ptfv1))+77)*125/375;
    Frec2 = (find(ptfv2==max(ptfv2))+77)*125/375;
    Frec3 = (find(ptfv3==max(ptfv3))+77)*125/375;
    Fv1(t) = Frec1;
    Fv2(t) = Frec2;
    Fv3(t) = Frec3;
    %Se toma el valor en potencia de la FFT
    Frec1 = find(ptfv1==max(ptfv1));
    Frec2 = find(ptfv2==max(ptfv2));
    Frec3 = find(ptfv3==max(ptfv3));
    PSv1(t) = ptfv1(Frec1);
    PSv2(t) = ptfv1(Frec2);
    PSv3(t) = ptfv1(Frec3);
end

```

```

%Calculo de la Entropía de Shannon
for t=1:4
    auxc1 = EEGv1(:,t);
    auxc2 = EEGv2(:,t);
    auxc3 = EEGv3(:,t);
    %Calcular la distribución de la probabilidad relativa
    probc1 = histcounts(auxc1, 'Normalization', 'probability');
    probc2 = histcounts(auxc2, 'Normalization', 'probability');
    probc3 = histcounts(auxc3, 'Normalization', 'probability');
    %Eliminar las probabilidades relativas 0 y cambiarlas por 1 ya que
    %no afectan al cálculo
    idx1 = find(probc1==0);
    idx2 = find(probc2==0);
    idx3 = find(probc3==0);
    probc1(idx1) = 1;
    probc2(idx2) = 1;
    probc3(idx3) = 1;
    %Calcular la Entropía
    SEv1(t) = -sum(probc1.*log2(probc1));
    SEv2(t) = -sum(probc2.*log2(probc2));
    SEv3(t) = -sum(probc3.*log2(probc3));
end

%Cálculo de los parámetros Hjorth
for t=1:4
    %Ventana 1
    diffv1 = diff(EEGv1(:,t));
    diff2v1 = diff(diffv1);
    varv1 = var(EEGv1(:,t));
    var_diffv1 = var(diffv1);
    var_diff2v1 = var(diff2v1);
    ACTv1(t) = varv1;
    MOBv1(t) = sqrt(var_diffv1 / varv1)*250/(2*pi);
    COMv1(t) = sqrt(var_diff2v1 / var_diffv1) / MOBv1(t);
    %Ventana 2
    diffv2 = diff(EEGv2(:,t));
    diff2v2 = diff(diffv2);
    varv2 = var(EEGv2(:,t));
    var_diffv2 = var(diffv2);
    var_diff2v2 = var(diff2v2);
    ACTv2(t) = varv2;
    MOBv2(t) = sqrt(var_diffv2 / varv2)*250/(2*pi);
    COMv2(t) = sqrt(var_diff2v2 / var_diffv2) / MOBv2(t);
    %ventana 3
    diffv3 = diff(EEGv3(:,t));
    diff2v3 = diff(diffv3);
    varv3 = var(EEGv3(:,t));
    var_diffv3 = var(diffv3);
    var_diff2v3 = var(diff2v3);
    ACTv3(t) = varv3;
    MOBv3(t) = sqrt(var_diffv3 / varv3)*250/(2*pi);
    COMv3(t) = sqrt(var_diff2v3 / var_diffv3) / MOBv3(t);
end

%Se integran todas las características en una sola matriz siguiendo el
% mismo orden que en el entrenamiento de las RNA

Featuresv1 = [Fv1 PSv1 ACTv1 MOBv1 COMv1 SEv1 WFLv1 ICSv1 VMav1];
Featuresv2 = [Fv2 PSv2 ACTv2 MOBv2 COMv2 SEv2 WFLv2 ICSv2 VMav2];
Featuresv3 = [Fv3 PSv3 ACTv3 MOBv3 COMv3 SEv3 WFLv3 ICSv3 VMav3];

%Se envían las características a la red neuronal
predictv1 = py.ClasificadorFinal.neural(Featuresv1);
res(1) = double(predictv1);
predictv2 = py.ClasificadorFinal.neural(Featuresv2);
res(2) = double(predictv2);
predictv3 = py.ClasificadorFinal.neural(Featuresv3);
res(3) = double(predictv3);
postura = mode(res);
if(postura == 0)
    posMov = pinza;
    sim(modelo, "StopTime", "4");
else
    posMov = saludo;
    sim(modelo, "StopTime", "4");
end
pruebas = pruebas + 1;
end

```

```
res
postura
disp('Prueba exitosa 1x36 car...')
```


Bibliografía

- [1] Rajesh P. N. Rao. *Brain-Computer Interfacing: An Introduction*. Cambridge University Press, 2013. DOI: [10.1017/CB09781139032803](https://doi.org/10.1017/CB09781139032803).
- [2] Clemens Brunner y col. “BNCI Horizon 2020: Towards a roadmap for the BCI Community”. En: *Brain-Computer Interfaces* 2.1 (2015), págs. 1-10. DOI: [10.1080/2326263x.2015.1008956](https://doi.org/10.1080/2326263x.2015.1008956).
- [3] Zhimin Lin y col. “A novel P300 BCI speller based on The triple rsvp paradigm”. En: *Scientific Reports* 8.1 (2018), págs. 1-9. DOI: [10.1038/s41598-018-21717-y](https://doi.org/10.1038/s41598-018-21717-y).
- [4] Lei Cao y col. “A synchronous motor imagery based neural physiological paradigm for Brain Computer Interface speller”. En: *Frontiers in Human Neuroscience* 11 (2017), pág. 274. DOI: [10.3389/fnhum.2017.00274](https://doi.org/10.3389/fnhum.2017.00274).
- [5] Trung-Hau Nguyen, Da-Lin Yang y Wan-Young Chung. “A high-rate BCI speller based on eye-closed EEG signal”. En: *IEEE Access* 6 (2018), págs. 33995-34003. DOI: [10.1109/access.2018.2849358](https://doi.org/10.1109/access.2018.2849358).
- [6] Sebastian Nagel y Martin Spüler. “Asynchronous non-invasive high-speed BCI speller with robust non-control state detection”. En: *Scientific Reports* 9.1 (2019), págs. 1-9. DOI: [10.1038/s41598-019-44645-x](https://doi.org/10.1038/s41598-019-44645-x).
- [7] Michael McMahon y Michael Schukat. “A low-cost, open-source, BCI- VR game control development environment prototype for game based neurorehabilitation”. En: *2018 IEEE Games, Entertainment, Media Conference (GEM)* (2018). DOI: [10.1109/gem.2018.8516468](https://doi.org/10.1109/gem.2018.8516468).
- [8] Sebastian Rosca y Monica Leba. “Design of a brain-controlled video game based on a BCI system”. En: *MATEC Web of Conferences* 290 (2019), págs. 1-9. DOI: [10.1051/mateconf/201929001019](https://doi.org/10.1051/mateconf/201929001019).
- [9] Elifsu Filiz y Reis Burak Arslan. “Design and implementation of steady state visual evoked potential based brain computer interface video game”. En: *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*. IEEE. 2020, págs. 335-338.
- [10] Adam Reust, Jaydip Desai y Louis Gomez. “Extracting motor imagery features to control two robotic hands”. En: *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)* (2018), págs. 118-122. DOI: [10.1109/isspit.2018.8642627](https://doi.org/10.1109/isspit.2018.8642627).

- [11] Chenguang Yang y col. "Mind control of a robotic arm with Visual Fusion Technology". En: *IEEE Transactions on Industrial Informatics* 14.9 (2018), págs. 3822-3830. DOI: [10.1109/tii.2017.2785415](https://doi.org/10.1109/tii.2017.2785415).
- [12] J. del R. Millan y col. "Noninvasive brain-actuated control of a mobile robot by human EEG". En: *IEEE Transactions on Biomedical Engineering* 51.6 (2004), págs. 1026-1033. DOI: [10.1109/tbme.2004.827086](https://doi.org/10.1109/tbme.2004.827086).
- [13] INEGI. *Discapacidad en México*. <http://cuentame.inegi.org.mx/poblacion/discapacidad.aspx>. Fecha de consulta: 18/11/2021. 2020.
- [14] Swagata Das, Devashree Tripathy y Jagdish Lal Raheja. *Real-time BCI system design to control Arduino based speed controllable robot using EEG*. 1.^a ed. Springer Verlag, 2019. DOI: [10.1007/978-981-13-3098-8](https://doi.org/10.1007/978-981-13-3098-8).
- [15] Marc R Nuwer y col. "IFCN standards for digital recording of clinical EEG". En: *Electroencephalography and clinical Neurophysiology* 106.3 (1998), págs. 259-261.
- [16] Avenilde Romo-Vázquez, Rebeca Romo-Vázquez y Hugo Velez-Pérez. "De la ingeniería Biomédica al aula de Matemáticas - Biomedical engineering to mathematics classroom". En: *ReCIBE, Revista electrónica de Computación, Informática, Biomédica y Electrónica* 1.1 (2017), pág. IV. DOI: [10.32870/recibe.v1i1.6](https://doi.org/10.32870/recibe.v1i1.6). URL: <http://recibe.cucei.udg.mx/index.php/ReCIBE/article/view/6>.
- [17] Christoph S Herrmann. "Human EEG responses to 1–100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena". En: *Experimental brain research* 137 (2001), págs. 346-353.
- [18] Zheng-Hua Wu y De-Zhong Yao. "A study on SSVEP-based BCI". En: *Journal of Electronic Science and Technology* 7.1 (2009), págs. 7-11.
- [19] John G. Proakis y Dimitris G. Manolakis. *Tratamiento Digital De Señales*. Pearson Educación, 2009.
- [20] Saeed V Vaseghi. *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.
- [21] Benyamin Ghogh y col. "Feature selection and feature extraction in pattern analysis: A literature review". En: *arXiv preprint arXiv:1905.02845* (2019).
- [22] Muhamed Farooq y Omid Dehzangi. "High accuracy wearable SSVEP detection using feature profiling and dimensionality reduction". En: *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE. 2017, págs. 161-164.
- [23] Fabien Lotte y col. "A review of classification algorithms for EEG-based brain-computer interfaces". En: *Journal of neural engineering* 4.2 (2007), R1.
- [24] Fernando Flórez y col. "Development of a Low-cost SVM-based Spontaneous Brain-computer Interface." En: *IJCCI (NCTA)*. 2011, págs. 415-421.
- [25] Danhua Zhu y col. "A survey of stimulation methods used in SSVEP-based BCIs". En: *Computational intelligence and neuroscience* 2010 (2010), págs. 1-12.

- [26] Microchip. *ATmega328P: 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash - Datasheet*. <https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf>. Fecha de consulta: 18/01/2021. 2018.
- [27] Danhua Zhu y col. “A survey of stimulation methods used in SSVEP-based BCIs”. En: *Computational intelligence and neuroscience* 2010 (2010), págs. 1-12.
- [28] David Regan. “An effect of stimulus colour on average steady-state potentials evoked in man”. En: *Nature* 210.5040 (1966), págs. 1056-1057.
- [29] *Cyton Specs: Openbci documentation*. Jul. de 2022. URL: <https://docs.openbci.com/Cyton/CytonSpecs/>.
- [30] *Ultracortex Mark IV: Openbci documentation*. Sep. de 2022. URL: <https://docs.openbci.com/AddOns/Headwear/MarkIV/#the-complete-ultracortex>.
- [31] A. Matías. “CONTROL DE PRÓTESIS ACTIVA DE MIEMBRO SUPERIOR MEDIANTE INTELIGENCIA ARTIFICIAL Y RETROALIMENTACIÓN DE FUERZA”. Universidad Tecnológica de la Mixteca, mar. de 2022.
- [32] Nili E Krausz, Ronald AL Rorrer y col. “Design and fabrication of a six degree-of-freedom open source hand”. En: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24.5 (2015), págs. 562-572.
- [33] Labstreaminglayer. *MATLAB bindings for libls*. Dic. de 2020. URL: <https://github.com/labstreaminglayer/libls-Matlab>.