

UNIVERSIDAD ANÁHUAC MAYAB
DIVISIÓN DE INGENIERÍA Y CIENCIAS EXACTAS



**Desarrollo de una interfaz cerebro-computadora EEG multicanal
basada en intención de movimiento**

**PROYECTO INTEGRADOR DE
INGENIERÍA MECATRÓNICA**

PRESENTA:

César Alejandro Gómez Carrillo

DIRECTOR DE PROYECTO:

M.C. Geovanny Rafael Giorgana Macedo

Mérida, Yucatán a 17 de mayo de 2018

Índice General

Capítulo 1 – Información del Proyecto.....	1
1.1 Introducción	1
1.2 Antecedentes	2
1.3 Descripción del problema.....	5
1.4 Justificación.....	8
1.5 Alcances y Limitaciones	9
1.5.1 Alcances	9
1.5.2 Limitaciones.....	10
1.6 Objetivos.....	11
Capítulo 2 – Marco Teórico.....	12
2.1 Señales cerebrales e intención de movimiento	12
2.1.1 Señales como funciones.....	12
2.1.2 Actividad cerebral y su registro por medio de EEG	13
2.1.3 Base de datos A de Graz para captura de intención de movimiento	16
2.2 Corrección de <i>artifacts</i>	18
2.2.1 Influencia del parpadeo y movimiento ocular en las señales EEG	18
2.2.2 Análisis de componentes independientes (ICA).....	20
2.3 Normalización	22
2.3.1 Introducción a la normalización para señales EEG	22
2.3.2 Normalización z-score.....	23
2.4 Selección de canales.....	24
2.4.1 Métodos de selección de canales	24
2.5 Extracción de características	28
2.5.1 Transformada de Fourier	28
2.5.2 Transformada ondícula (<i>wavelet</i>).....	29
2.5.3 Método de ventana móvil.....	30
2.6 Clasificación de intención de movimiento.....	31
2.6.1 AdaBoost	32

2.6.2	Máquinas de vectores de soporte	33
Capítulo 3	– Metodología	34
3.1	Preprocesamiento	35
3.1.1	Corrección de <i>artifacts</i> utilizando algoritmo FastICA	35
3.1.2	Normalización por Z-score	37
3.1.3	Selección de canales mediante el método de CSP	38
3.2	Extracción de características	39
3.3	Clasificación.....	40
Capítulo 4	– Desarrollo del Proyecto	42
4.1	Base de datos de Graz	42
4.2	Preprocesamiento: Corrección de <i>artifacts</i>	44
4.2.1	Algoritmo FastICA para descomposición de señales en sus componentes independientes	44
4.2.2	Parámetros para eliminación de <i>artifacts</i>	46
4.2.3	Rechazo de ICs mediante detección de datos atípicos por z-score	51
4.3	Preprocesamiento: Normalización mediante z-score	52
4.4	Preprocesamiento: Selección de canales mediante CSP	53
4.5	Generación de nuevas bases de datos combinando bloques de preprocesamiento....	55
4.6	Extracción de características y clasificación.....	56
4.6.1	AdaBoost	56
4.6.2	SVM.....	59
4.6.3	AdaSVM.....	61
Capítulo 5	– Resultados	63
5.1	Resultados del clasificador AdaBoost para la base de datos de Graz original.....	63
5.2	Resultados del clasificador AdaBoost para las señales EEG sometidas a bloques de preprocesamiento.....	64
5.2.1	Corrección de <i>artifacts</i>	65
5.2.2	Normalización.....	65
	66
5.2.3	Selección de canales	66

5.3	Resultados del clasificador SVM para la base de datos Graz original	67
5.4	Resultados del clasificador SVM para las señales EEG sometidas a bloques de preprocesamiento.....	68
5.4.1	Corrección de <i>artifacts</i>	68
5.4.2	Normalización.....	68
5.4.3	Selección de canales	69
5.5	Resultados del clasificador AdaSVM para la base de datos Graz original	70
5.6	Resultados del clasificador AdaSVM para las señales EEG sometidas a bloques de preprocesamiento.....	70
5.6.1	Corrección de <i>artifacts</i>	70
5.6.2	Normalización.....	71
5.6.3	Selección de canales	72
5.7	Mejores resultados de acuerdo con el clasificador.....	73
5.7.1	Corrección de <i>artifacts</i>	73
5.7.2	Normalización.....	73
5.7.3	Selección de canales	74
		75
5.8	Canales relevantes para la clasificación.....	75
5.9	Tiempo requerido para generación de modelos de la máquina SVM contra la máquina AdaSVM.....	77
Capítulo 6 – Conclusiones	78	
6.1	Conclusiones Técnicas	78
6.2	Conclusiones Personales	79
6.3	Recomendaciones	80
Referencias	81	

Índice de Tablas

Tabla 4.1.- Función FastICA desarrollada por Hyvarinen y Oja [34]. Esta función toma como parámetro una matrix cuyas filas se asumen como una combinación lineal de señales estadísticamente independientes, y mediante un algoritmo de punto flotante fijo, estima los componentes independientes con una alta velocidad.	44
Tabla 4.2.- Función bciRemoveArtifacts. Esta función recibe la señal EEG multidimensional y el correspondiente registro EOG como parámetros, y regresa la señal reconstruida, después de haber rechazado los ICs que, bajo el procedimiento descrito anteriormente, se identifican como 'artifacts'	47
Tabla 4.3.- Función bciGetMaxCorrCoeffs. Esta función toma una matriz de componentes independientes y la matriz EOG asociada a dichos ICs, y devuelve los máximos valores absolutos de los coeficientes de correlación entre los ICs y la información del movimiento ocular.....	48
Tabla 4.4.- Función kurtosis del toolbox Statistics and Machine Learning de Mathworks, Inc. La función devuelve el valor de curtosis de cada columna. En el caso de este proyecto, la entrada X es la matriz de componentes independientes I introducida en bciRemoveArtifacts.	49
Tabla 4.5.- Función genhurst de Aste [37]. Computa el exponente generalizado de Hurst de una serie de datos. Esta función fue llamada para cada componente independiente.	50
Tabla 4.6.- La función bciGetMeanSpectraSlope. Computa el valor medio de la densidad espectral de potencia de cada componente independiente. Para el cálculo de la densidad espectral de potencia, este algoritmo se apoya en la función pwelch del toolbox Signal Processing de MATLAB.	50
Tabla 4.7.- Función zscore del toolbox Statistics and Machine Learning integrada a MATLAB. Recibe como parámetros un arreglo de datos, una bandera que, si el	

valor es cero, devuelve la media y desviación estándar de la muestra (N-1), y la media y desviación estándar de la población (N) si se introduce un valor de 1 y la dimensión a través de la cual computará los z-scores.....	52
Tabla 4.8.- bciSelectChannels. Toma las pruebas de dos clases de imaginación de movimiento y un número de canales relevantes a seleccionar, y encuentra los canales que permiten discriminar mejor entre las dos clases utilizando el método de Common Spatial Patterns (CSP).	54
Tabla 4.9.- Función bandPassFilter de [2]. Realiza un filtrado de tipo pasabanda, de acuerdo con las especificaciones introducidas por el usuario. Para este proyecto, siempre se utilizó un filtro Butterworth 8-30 Hz.....	54
Tabla 4.10.- Función bciPreprocessingStage. Esta función toma como parámetros las variables necesarias para desempeñar los diferentes bloques de preprocesamiento, así como valores lógicos para indicar qué combinación de los bloques utilizar para la agrupación de pruebas de la variable ‘signals’....	55
Tabla 4.11.- Función bciTrain_DWT_Adaboost. Esta función construye modelos a partir de las dos variables correspondientes a una sesión de pruebas de imaginación de movimiento, donde cada variable contiene las pruebas de una sola intención de movimiento.	57
Tabla 4.12.- Función testAdaBoost. Esta función constituye la etapa de prueba que utiliza los modelos previamente construidos mediante trainAdaBoost, y, tomando un nuevo set de señales, correspondientes a las mismas clases, busca clasificarlas correctamente.....	58
Tabla 4.13.- Función fitcsvm del toolbox Statistics and Machine Learning de MATLAB. Entrena un modelo de clasificación a partir de una matriz de características. Se puede optar por un modelo construido a partir de validación cruzada si se introduce un vector de parámetros optimizables.	59

Tabla 4.14.- Función trainSVM. Filtra, selecciona canales relevantes, extrae características y entrena el modelo de clasificación de la matriz de características creada utilizando fitcsvm.	60
Tabla 4.15.- Función predict del toolbox Statistics and Machine Learning de MATLAB. Devuelve un vector de etiquetas predichas a partir del modelo de clasificación SVM y matriz de características introducido.....	60
Tabla 4.16.- Función testSVM. Esta función constituye la etapa de prueba que utiliza los modelos previamente construidos mediante trainSVM, y, tomando un nuevo set de señales, correspondientes a las mismas clases, busca clasificarlas correctamente.....	61
Tabla 4.17.- Funciones trainAdaSVM y testAdaSVM. La función trainAdaSVM filtra, selecciona canales relevantes, extrae características, selecciona características mediante AdaBoost y entrena el modelo de clasificación de la matriz de características creada utilizando fitcsvm. Previo al entrenamiento, utiliza un algoritmo basado en AdaBoost para seleccionar las mejores características para introducir al clasificador. La función testAdaSVM para por el mismo proceso, pero al final realiza las hipótesis en base al modelo generado durante el entrenamiento.....	62
Tabla 5.1.- Mejores resultados del clasificador AdaBoost con la base de datos de Graz sin ser sometida a bloques de preprocesamiento.....	63
Tabla 5.2.- Resultados de la clasificación mediante AdaBoost de la base de datos sometida al bloque de corrección de ‘artifacts’	65
Tabla 5.3.- Resultados de la clasificación mediante AdaBoost de la base de datos sometida al bloque de normalización.....	66
Tabla 5.4.- Resultados de la clasificación mediante AdaBoost de la base de datos sometida al bloque de selección de canales.	66

Tabla 5.5.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz original, utilizando el clasificador SVM.	67
.....	
Tabla 5.6.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometida a corrección de artifacts, utilizando el clasificador SVM	68
.....	
Tabla 5.7.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometida a normalización, utilizando el clasificador SVM	69
.....	
Tabla 5.8.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometida a selección de canales, utilizando el clasificador SVM	69
.....	
Tabla 5.9.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz original, utilizando el clasificador AdaSVM.	70
.....	
Tabla 5.10.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometidas a corrección de ‘artifacts’, utilizando el clasificador AdaSVM.....	71
.....	
Tabla 5.11.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometidas a normalización, utilizando el clasificador AdaSVM.....	72
.....	
Tabla 5.12.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometidas a selección de canales, utilizando el clasificador AdaSVM.....	72
.....	

Índice de Figuras

- Figura 1.1.- Gráficas de señales EEG, ECG y EOG. La gráfica en a) muestra la influencia de las señales en b) y c) sobre las señales cerebrales. Se pueden observar picos en la señal EEG, que corresponde temporalmente a los picos que se presentan naturalmente en las señales de ECG. Y la forma de la señal EEG sigue la forma de la señal EOG, debido a que la primera tiene un rango de amplitud mucho menor, lo que las vuelve altamente susceptibles a las señales cercanas provenientes del movimiento ocular. Imagen recuperada de <https://www.intechopen.com/books/adaptive-filtering-applications/noise-removal-from-eeg-signals-in-polisomnographic-records-applying-adaptive-filters-in-cascade> 5
- Figura 1.2.- Gráficas de dos pruebas correspondientes a sujetos distintos y misma intención de movimiento. Se puede observar la variación en la amplitud entre las señales, incluso cuando corresponden a la misma posición de electrodo en el cuero cabelludo, en este caso, el canal 6. Datos obtenidos de la base de datos A de Graz para la BCI Competition 2008 [13]..... 6
- Figura 1.3.- Topografía de la distribución en la importancia de cada canal para la clasificación de señales cerebrales en nueve sujetos diferentes. La distribución de importancia de los canales varía considerablemente entre cada sujeto, por lo que excluir ciertos canales de manera global en un sistema basado en BCI podría eliminar una fuente de información sustancial en ciertos sujetos. Imagen recuperada de Xu et al. [4]. 7
- Figura 1.4.- Población con limitación en la actividad según tipo de limitación. Los datos muestran que la discapacidad motriz (caminar o moverse) conforma la mayoría de la población con discapacidad. Imagen recuperada de <http://www.beta.inegi.org.mx/temas/discapacidad/> 8
- Figura 2.1.- Representación gráfica de la actividad eléctrica generada por la actividad cerebral. Imagen recuperada de:

<http://eeghacker.blogspot.mx/2013/11/waveforms-from-homemade-electrodes.html> 12

Figura 2.2.- La comunicación entre neuronas se realiza mediante impulsos eléctricos localizados en zonas de conexión denominadas ‘sinapsis’. Esta conexión puede ser eléctrica, cuando las neuronas tienen un punto de contacto físico, como se muestra en la figura. Pero también puede ser de tipo química, cuando la transmisión se realiza por un compuesto químico que se traslada de una neurona a otra. Imagen recuperada de [7]. 13

Figura 2.3.- El posicionamiento de los diferentes electrodos tiene el objetivo de obtener actividad neuronal relacionada con estado del cerebro, o imaginación de movimiento. Imagen recuperada de <https://speakingofresearch.com/tag/eeg/> 14

Figura 2.4.- a) Onda delta, comúnmente presente en un sujeto durante sueño profundo. b) Onda theta. Ésta ocurre en adulto en situaciones de estrés emocional. c) Onda alfa, presente en estado normal de relajación. d) Onda beta. Presente en sujetos dado un estímulo mental. 15

Figura 2.5.- Mapeo del posicionamiento de los 22 canales EEG (izquierda), y 3 canales EOG. Estos canales se posicionan de acuerdo al sistema internacional 10-20. Los 3 canales EOG se utilizan para capturar información del movimiento ocular, para aplicación de métodos de corrección de artifacts. Imagen de [13]. 16

Figura 2.6.- Esquema de tiempo de cada sesión: 2 minutos de registro de EOG con los ojos abiertos, seguida de 2 minutos de registro de EOG con los ojos cerrados, y 1 minuto de registro de EOG con movimiento ocular. Posterior a esto, se realizan las 6 series de 48 pruebas cada una. Imagen de [13]. 17

Figura 2.7.- Esquema temporal de cada prueba de imaginación de movimiento. En $t = 0$, se introduce un tono y una cruz de fijación. En $t = 2$, se muestra el indicador del movimiento a imaginar, y el sujeto desempeña dicha tarea por 3

segundos, hasta que la cruz de fijación desaparece en t = 6. Imagen de [13].	18
Figura 2.8.- Clasificación de los 'artifacts' que se presentan en la captura de señales EEG. Datos adaptados de Savelainen, 2010 [11].	19
Figura 2.9.- a) Señales de audio independientes. b) Combinación de las señales, observada a través de tres entradas de audio. c) Estimación de las señales originales obtenida de las señales observadas. Imágenes recuperadas de Hyvärinen, et al., 2004 [13].	22
Figura 2.10.- Función $f(x) = \cos[(2\pi 10/300)x]$ utilizando el software MATLAB de Mathworks. MATLAB utiliza la función fft para calcular la transformada discreta de Fourier. Las segunda gráfica muestra los componentes de frecuencia de la señal, es decir, el espectro de la función.	29
Figura 2.11.- Método de ventana móvil para segmentación de señales EEG. Imagen recuperada de [2].	30
Figura 2.12.- a) Ventana móvil con traslape de valor -10 ms (Traslape negativo). b) Ventana móvil con traslape de valor +10 ms (Traslape positivo). c) Ventana móvil sin traslape. Imágenes recuperadas de [2].	31
Figura 2.13.- Generación de un hiperplano utilizando SVM. El hiperplano óptimo maximiza la distancia entre los datos de entrenamiento [33].	33
Figura 3.1.- Diagrama esquemático de las etapas de operación del sistema BCI a desarrollar durante este proyecto. Inicia con el preprocesamiento de las señales provenientes de la base de datos, seguido de la extracción de características, y finaliza con la etapa de clasificación, en la que se discrimina entre dos intenciones de movimiento: mano derecha y ambos pies.	34
Figura 3.2.- Etapa de preprocesamiento del sistema BCI. Para la corrección de 'artifacts', se utilizará el algoritmo Fast ICA de Hyvarinen [32]. Se espera obtener señales limpias de 'artifacts' al final de esta etapa de preprocesamiento.	

Con la técnica Z-score, se espera establecer una referencia constante en todas las señales, observable en valores similares de amplitud en las señales entre pruebas y entre sujetos. Con CSP, se espera encontrar los canales más relevantes para la clasificación, rechazando los canales restantes, mejorando la velocidad de cada etapa de operación del sistema BCI.	35
Figura 3.3.- Corrección de la señal EEG multicanal mediante la extracción de componentes independientes con la técnica ICA. Se observa la importante eliminación de la influencia de artifacts en los canales T3 y T4, así como Fp1, Fp2, F3, F4, F7 y F8. Imagen recuperada de [36].	37
Figura 3.4.- a) La matriz de características consiste en vectores de coeficientes obtenidos por la computación de DWT en los segmentos obtenidos por ventanas móviles. b) Para cada una de las 100 señales del supuesto de este ejemplo, existe una etiqueta dependiendo de la clase correspondiente, para lo cual se presenta el vector de etiquetas.	39
Figura 3.5.- Clasificación por AdaBoost durante el proceso de entrenamiento. La máquina de clasificación construirá un modelo a partir de la matriz de características y el vector de etiquetas correspondientes a las señales extraídas.	40
Figura 3.6.- Proceso de pruebas para la máquina de clasificación AdaBoost. Utilizando el modelo generado durante el entrenamiento, una nueva matriz de características y el vector de etiquetas deseado, se comparará el desempeño del sistema mediante la generación de un vector de etiquetas por el clasificador.	40
Figura 4.1.- Formato de una prueba individual, que abarca la parte de la señal en la que se registra una intención de movimiento. Teniendo una duración de tres segundos y registrando voltaje a 250 Hz, la señal discreta cuenta con 750 datos. Las columnas representan los 22 canales usados para el registro de la	

señal EEG. La matriz EOG contiene la información del movimiento ocular registrada en la duración de la prueba.	43
Figura 4.2.-Formato de las pruebas almacenadas en cada variable. Estilo ‘página de libro’, las matrices bidimensionales corresponden a una prueba, mientras que la tercera dimensión almacena cada una de las 72 pruebas individuales.	43
Figura 4.3.- Ejemplo: si bajo procesos de análisis estadísticos, se encuentra que el componente independiente i_2 corresponde a un artifact, éste es descartado en I. También se descarta la columna que corresponde a su matriz de combinación en A.	45
Figura 4.4.- Reconstrucción de las señales EEG utilizando las matrices de recombinación y reconstrucción nuevas, A' e I' , que excluyen aquellos componentes independientes que son considerados ‘artifacts’. El resultado es una matriz S, que corresponde a la combinación lineal de las señales, como fueron registradas en los electrodos, pero la influencia del ruido en esta combinación lineal efectivamente eliminada.....	46
Figura 4.5.- Para el rechazo automático de ICs, el método FASTER mide el número de desviaciones estándar respecto de la media de un parámetro en todos los componentes independientes. Una vez que se detecta un dato atípico por la obtención de un z-score de valor mayor a 3, el IC es eliminado de la matriz I.	51
Figura 4.6.- Procedimiento de la generación de las bases de datos necesarias para la extracción de características, entrenamiento de la máquina de aprendizaje, y clasificación de intención de movimiento, utilizando la función bciPreprocessingStage.....	56
Figura 5.1.- Mejores resultados para la mejora del desempeño del sistema BCI mediante corrección de ‘artifacts’.	73

Figura 5.2.- Mejores resultados en mejora de desempeño para cada sujeto, comparando la base de datos original versus la base de datos normalizada.	74
Figura 5.3.- Mejores resultados en la mejora de desempeño del sistema BCI con los mejores canales seleccionados.....	75
Figura 5.4.- Puntuación de relevancia de cada canal a través de todos los sujetos de prueba.	76
Figura 5.5.- Mapa de calor que muestra los resultados de puntajes de relevancia de cada canal, a través de todos los sujetos.....	76
Figura 5.6.- Comparación de tiempo requerido para la generación de modelos entre SVM y AdaSVM.i	77

Resumen

En este proyecto se implementaron y analizaron técnicas de preprocessamiento para la implementación de un sistema de interfaz cerebro-computadora basada en electroencefalograma (EEG) multicanal, capaz de distinguir dos intenciones de movimiento. Utilizando la base de datos pública de la Universidad de Graz para la BCI Competition en 2008, compuesta por las pruebas desempeñadas por ocho sujetos, se desarrolló un sistema que implementara tres etapas de preprocessamiento: corrección de *artifacts*, normalización y selección de canales. Para el entrenamiento se utilizaron tres máquinas de clasificación: AdaBoost, SVM y AdaSVM, una máquina que utiliza AdaBoost para la selección de características más importantes y entrena el modelo optimizado utilizando SVM. Los resultados del proyecto muestran una mejora en el porcentaje de desempeño para las señales sometidas bloques de corrección de *artifacts* y normalización, llevando la exactitud del sistema hasta un 92.36%.

Capítulo 1 – Información del Proyecto

1.1 Introducción

En este proyecto se propone el desarrollo de una interfaz cerebro-computadora que procese señales eléctricas provenientes de un electroencefalograma (EEG) para determinar dos intenciones de movimiento. El proyecto busca principalmente explorar y analizar técnicas para eliminación de ruido, normalización y selección de canales. Con estas actividades, se busca combinar las mejores técnicas y los mejores algoritmos para crear una interfaz con mayor precisión respecto a técnicas previas con las que se ha experimentado [1, 2].

Las interfaces cerebro-computadora (BCIs, por sus siglas en inglés) son sistemas que miden la actividad cerebral asociada con la intención de un usuario y la traducen a señales de control que pueden ser procesadas y clasificadas, generalmente utilizando una computadora [3]. La intención específica que se estudiará para este proyecto es la imaginación de movimiento, puesto que las señales producidas al pensar en el movimiento de brazos, piernas, lengua o rotación de objetos, contienen patrones EEG [4] que pueden ser extraídos. Esto posibilita la construcción de sistemas no invasivos controlados únicamente por la actividad cerebral, sin involucrar movimiento muscular, los cuales resultan útiles en áreas como salud, industria, y entretenimiento.

Este proyecto es motivado por consolidar conocimientos fundamentales de ingeniería eléctrica, como el procesamiento digital de señales, y la exploración de áreas de las ciencias computacionales como el aprendizaje de máquina, para brindar avances que permitan aproximar a las BCIs a aplicaciones prácticas en todas las áreas del conocimiento humano.

1.2 Antecedentes

El uso de EEG no invasivo como base para el desarrollo de interfaces cerebro-computadora ha generado un gran interés durante los últimos 15 años [5]. El EEG es un método que ha sido adoptado ampliamente debido a su facilidad de uso, alta resolución temporal y bajo costo respecto de otros métodos no invasivos, como las imágenes por resonancia magnética funcional (fMRI), la magnetoencefalografía (MEG) y la tomografía por emisión de positrones (PET), entre otros [4].

En su estudio de tesis, Szachewicz [6] presenta un algoritmo para la clasificación de segmentos de señales de EEG. Con este algoritmo, busca implementar métodos de procesamiento de señales para optimizar el tiempo del procesamiento, reducir el ruido de las señales y extraer características. Sus resultados fueron utilizados para comparar el desempeño entre dos diferentes máquinas de clasificación, análisis discriminante lineal y máquina de vectores de soporte (LDA y SVM, por sus siglas en inglés, respectivamente), y obtuvo resultados aceptables al ser comparados con los mejores algoritmos de clasificación de la BCI Competition, un concurso abierto sobre clasificación de datos para BCIs. SVM es la máquina de clasificación que se utilizará para el desarrollo de este proyecto.

Wang, Miao y Blohm [4] desarrollaron un *framework* que procesa señales EEG obtenidas de una base de datos pública mediante cinco etapas de procesamiento: segmentación de señales, corrección de *artifacts* (i.e., eliminación de ruido que contamina las señales), normalización, selección de canales y extracción de características. Sus resultados mostraron que la intención de movimiento a partir de las señales contaminadas con *artifacts* puede ser correctamente clasificada si se utilizan sus métodos.

Gollahalli [7] presenta un *framework* que utiliza una red neuronal para obtener y clasificar datos EEG en tiempo real. Utilizando el *framework*, Gollahalli logra implementar una metodología para controlar un Quadcopter, que es un dron utilizado para llevar cargas pesadas. El autor logra enviar comandos a un dron virtual de manera correcta utilizando la red neuronal DeSNN para la clasificación de datos.

Kachenoura et al. [8] realizaron un estudio comparativo de diferentes métodos para implementar la técnica de análisis de componentes independientes (ICA, por sus siglas en inglés), una técnica que permite separar las señales cerebrales del ruido proveniente de fuentes como la actividad ocular, muscular o cardiaca [8]. En base a sus resultados, los autores concluyen que la selección apropiada del algoritmo de ICA puede mejorar el desempeño de los sistemas BCI.

Zhou, Mazon y Birch [9] implementaron un método de normalización denominado transformada de normalización de energía (ENT, por sus siglas en inglés) con el que lograron mejorar el desempeño de un sistema BCI al reducir la variabilidad de la amplitud de las señales provenientes de seis canales, y pruebas realizadas en seis sujetos. Sus resultados mostraron una mejora en el desempeño de su sistema BCI en sujetos cuya precisión se encontraba por debajo de 90%.

Wang, Gao S. y Gao X. [10] desarrollaron un método de selección de canales basado en *common spatial patterns* (CSP). Al mejorar la discriminación entre diferentes clases de imaginación de movimiento utilizando CSP, el objetivo de los autores fue mejorar la practicidad de las BCIs mediante la reducción del número de canales. Sus resultados mostraron una precisión de 93.45% y 91.88% para dos sujetos utilizando 4 canales óptimos.

Los resultados de estas investigaciones muestran que existe un gran interés por mejorar el desempeño de las interfaces cerebro computadora. El objetivo de

estos avances es llegar a un sistema robusto que pueda integrarse a aplicaciones prácticas. Un sector de gran interés para aplicaciones de BCI es el sector salud, puesto que pacientes con discapacidad motriz [11] podrían comunicarse o controlar sistemas electromecánicos. Pero existen otras áreas en las que sistemas de esta clase pueden ser implementados, como el sector industria o para aplicaciones de entretenimiento.

1.3 Descripción del problema

Las BCIs prometen establecerse como una tecnología capaz de mejorar la calidad de vida del ser humano en diversos ámbitos. Sin embargo, existen ciertos retos que aún no han sido completamente superados, y que previenen el uso de sistemas de esta clase en aplicaciones prácticas. La sección 1.2 menciona algunos de los trabajos dirigidos a mejorar el desempeño de las BCIs mediante la implementación y análisis de diferentes técnicas de procesamiento y clasificación de señales cerebrales. A continuación, se menciona la problemática que éstos abordan.

Diversos estudios han demostrado la alta susceptibilidad de las señales de EEG a ser contaminadas por *artifacts*, es decir, ruido en las señales, como señales producidas por el movimiento de los ojos (EOG), ritmo cardíaco (ECG) y músculos craneales (EMG) [4, 6, 8, 12]. Como muestra la Figura 1.1 en la que se observan tres gráficas de señales eléctricas de EEG, ECG y EOG respectivamente, el canal EEG registra parcialmente las señales del corazón y del movimiento ocular,

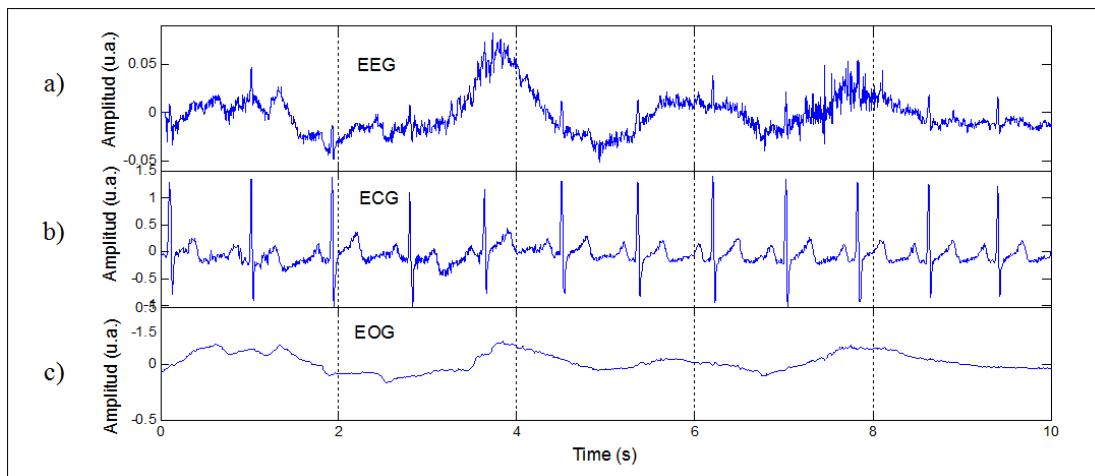


Figura 1.1.- Gráficas de señales EEG, ECG y EOG. La gráfica en a) muestra la influencia de las señales en b) y c) sobre las señales cerebrales. Se pueden observar picos en la señal EEG, que corresponde temporalmente a los picos que se presentan naturalmente en las señales de ECG. Y la forma de la señal EEG sigue la forma de la señal EOG, debido a que la primera tiene un rango de amplitud mucho menor, lo que las vuelve altamente susceptibles a las señales cercanas provenientes del movimiento ocular. Imagen recuperada de <https://www.intechopen.com/books/adaptive-filtering-applications/noise-removal-from-eeg-signals-in-polisomnographic-records-applying-adaptive-filters-in-cascade>

afectando la forma de la señal cerebral, que se presenta en un rango de amplitud mucho menor.

Al mismo tiempo, los resultados del procesamiento de las señales EEG varían de sujeto a sujeto, lo cual dificulta la construcción de un dispositivo controlado por la mente que funcione de manera homogénea para una población. La Figura 1.2 muestra dos señales EEG de la base de datos A de Graz para la BCI Competition en 2008 [13], obtenidas del sujeto 2 y 8, respectivamente. Se muestran pruebas correspondientes a la misma intención de movimiento. Como se observa en las señales, a pesar de que la información obtenida entre los usuarios corresponde al mismo posicionamiento de electrodo (canal 6), la amplitud de la señal del sujeto 2 oscila aproximadamente entre -17 μ V y 25 μ V, mientras que el rango es entre -50 μ V y 50 μ V para la señal del sujeto 8. La variación en la amplitud entre sujetos dificulta la clasificación de intención de movimiento y, por tanto, representa un obstáculo para implementar un sistema BCI multisujeto.

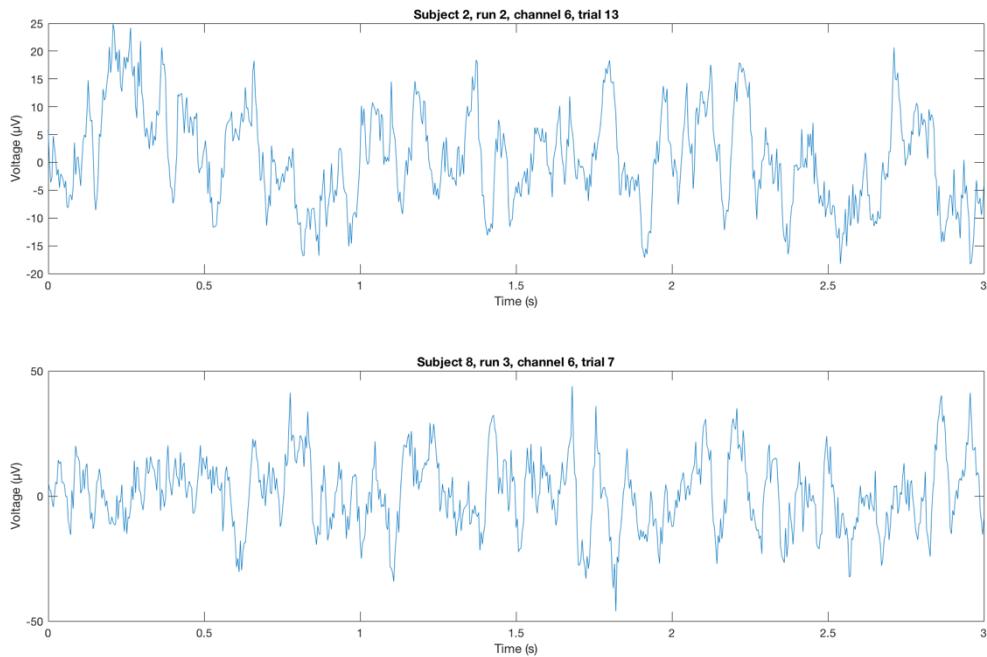


Figura 1.2.- Gráficas de dos pruebas correspondientes a sujetos distintos y misma intención de movimiento. Se puede observar la variación en la amplitud entre las señales, incluso cuando corresponden a la misma posición de electrodo en el cuero cabelludo, en este caso, el canal 6. Datos obtenidos de la base de datos A de Graz para la BCI Competition 2008 [13].

Finalmente, los sistemas BCIs usualmente requieren de múltiples canales para una correcta clasificación de la intención de movimiento. Sin embargo, el uso de un gran número de canales puede volver impráctico a un sistema, sobre todo si se quieren desarrollar dispositivos domésticos con cierta facilidad de colocación para el usuario, además de la elevación de la complejidad computacional de las tareas de clasificación, lo cual introduce latencia, así como mayores fuentes de *artifacts*. La Figura 1.3 muestra una topografía de la distribución en importancia de los canales en diferentes sujetos de prueba obtenida en un estudio por Xu et al. [4], mostrando una gran diferencia en ésta de sujeto a sujeto.

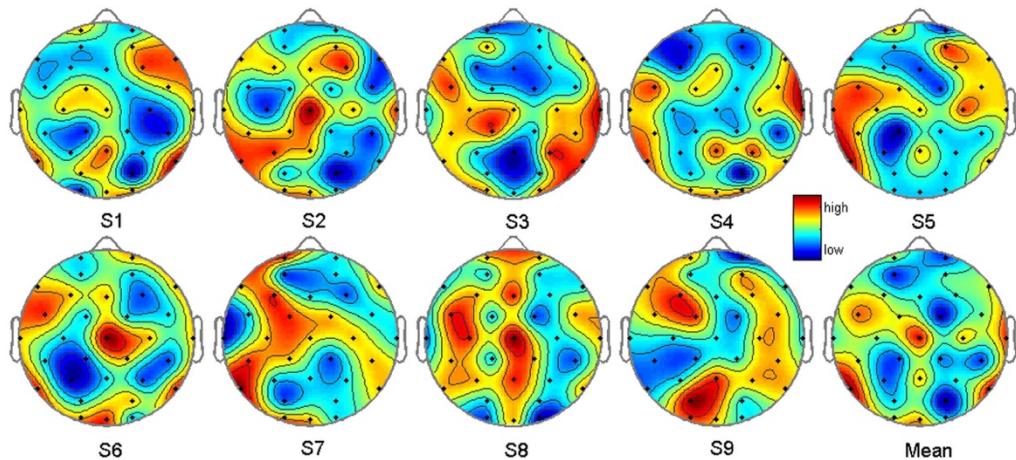


Figura 1.3.- Topografía de la distribución en la importancia de cada canal para la clasificación de señales cerebrales en nueve sujetos diferentes. La distribución de importancia de los canales varía considerablemente entre cada sujeto, por lo que excluir ciertos canales de manera global en un sistema basado en BCI podría eliminar una fuente de información sustancial en ciertos sujetos. Imagen recuperada de Xu et al. [4].

En conclusión, la contaminación de las señales por ruido no deseado, así como la variabilidad de la amplitud de las señales entre sujetos impide que los sistemas tengan el grado de confiabilidad que se espera en la mayoría de las aplicaciones prácticas. Y la considerable latencia de la mayoría de las BCIs debida al elevado número de canales requerido para la clasificación vuelve imposibles muchas tareas que requieren respuesta en tiempo real, tales como el control de vehículos no tripulados, o prótesis neurológicas que respondan de manera inmediata a la intención de movimiento que corresponde a la señal específica emitida por el cerebro.

1.4 Justificación

La Figura 1.4 muestra datos del INEGI [15] que revelan que 2,437,397 personas en México padecen una limitación en su habilidad para caminar o moverse.

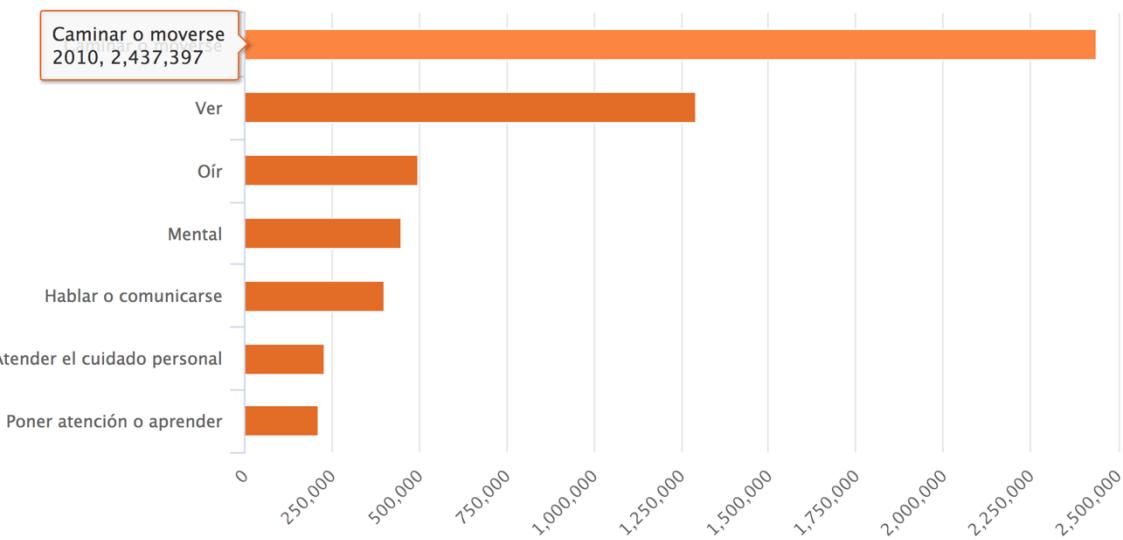


Figura 1.4.- Población con limitación en la actividad según tipo de limitación. Los datos muestran que la discapacidad motriz (caminar o moverse) conforma la mayoría de la población con discapacidad. Imagen recuperada de <http://www.beta.inegi.org.mx/temas/discapacidad/>

Estos datos justifican la enorme importancia del desarrollo de proyectos que busquen implementar tecnologías de vanguardia para la mejora de la calidad de vida de millones de personas. Tan sólo en México, más de 2 millones de personas se verían beneficiadas por un sistema que les permitiera controlar múltiples dispositivos electrónicos en tiempo real a partir de procesos mentales.

La problemática mencionada anteriormente aborda el hecho de que estos sistemas aún no pueden ser utilizados en aplicaciones prácticas debido a las limitaciones actuales de las tecnologías de interfaces cerebro-computadora, algunas de las cuales corresponden a los retos explicados en la sección 1.3. Este proyecto tiene como fin, brindar avances que permitan llegar a aplicaciones prácticas en un futuro cercano mediante la exploración, implementación y análisis de técnicas que enfrenten los retos mencionados.

1.5 Alcances y Limitaciones

1.5.1 Alcances

- Procesamiento de base de datos de EEG
- Exploración y análisis de técnicas de corrección de ‘artifacts’
- Implementación de técnicas de normalización y selección de canales
- Entrenamiento y pruebas
- Análisis de resultados
- Detección de los beneficios y desventajas de las técnicas implementadas para el procesamiento de la base de datos EEG pública.
- Sistema multisujeto y multicanal

1.5.2 Limitaciones

- No se realizarán pruebas físicas.
- No se desarrollará un dispositivo para la obtención de señales.
- Se tendrá un enfoque a dos intenciones de movimiento.

El motivo por el cual no se realizarán pruebas físicas o se desarrollará un dispositivo para la obtención de las señales es que se cuenta con una base de datos EEG pública, la cual ha sido estudiada y procesada por proyectos anteriores y de terceros. El objetivo es comparar las técnicas que se implementen durante este proyecto con aquellas realizadas por estudios previos, con el fin de aportar un conjunto de técnicas optimizadas.

1.6 Objetivos

General:

Desarrollar un sistema BCI capaz de determinar intención de movimiento, robusto, y multisujeto.

Específicos:

- 1) Implementar y analizar los 3 bloques fundamentales del proyecto:
 - a) Corrección/eliminación de 'artifacts'
 - b) Técnicas de normalización
 - c) Selección de canales
- 2) Extracción de características
- 3) Entrenamiento de la máquina SVM para la clasificación de dos intenciones de movimiento.

Capítulo 2 – Marco Teórico

2.1 Señales cerebrales e intención de movimiento

2.1.1 Señales como funciones

Lathi (1998) [16] define a una señal como “un conjunto de información o datos”. En ingeniería eléctrica, se define a las señales como funciones matemáticas que mapean pares de elementos de manera que:

$$f = \mathbb{R} \rightarrow \mathbb{R} \quad (1)$$

$$f = \mathbb{R} \rightarrow \mathbb{C} \quad (2)$$

$$f = \mathbb{C} \rightarrow \mathbb{C} \quad (3)$$

Las ecuaciones 1, 2 son mapeos de valores en el dominio, reales \mathbb{R} , con valores reales o complejos \mathbb{C} en el rango, la ecuación 3 mapea valores complejos con complejos, y f es la regla que especifica este mapeo, es decir, la función. Esto implica que las señales tienen una representación matemática y, por tanto, sus datos pueden ser estudiados a partir de ésta, habilitando herramientas como el cálculo, ciencias computacionales y procesamiento digital de señales para ejercer dicho análisis. En muchos casos, estas señales son representaciones de un valor en una escala temporal, es decir, el valor está en función del tiempo, como las señales de electroencefalograma en la Figura 2.1.

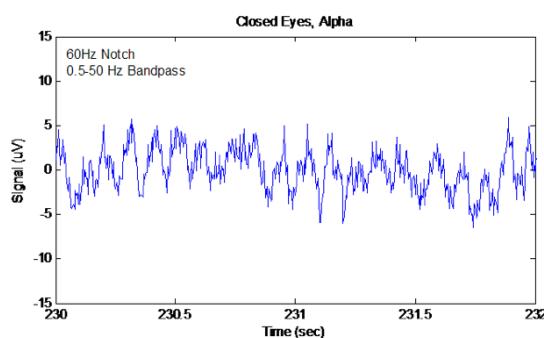


Figura 2.1.- Representación gráfica de la actividad eléctrica generada por la actividad cerebral.
Imagen recuperada de: <http://eeghacker.blogspot.mx/2013/11/waveforms-from-homemade-electrodes.html>

2.1.2 Actividad cerebral y su registro por medio de EEG

Las componentes del cerebro que realizan la transmisión o intercambio de información son las neuronas. Las neuronas son células capaces de transmitir impulsos eléctricos entre ellas, a través de zonas de conexión conocidas como sinapsis [7], como lo muestra la Figura 2.2. Estos impulsos eléctricos componen la actividad cerebral a gran escala, generada por una compleja red de comunicación entre las más de 100 mil millones de neuronas presentes en el cerebro [7].

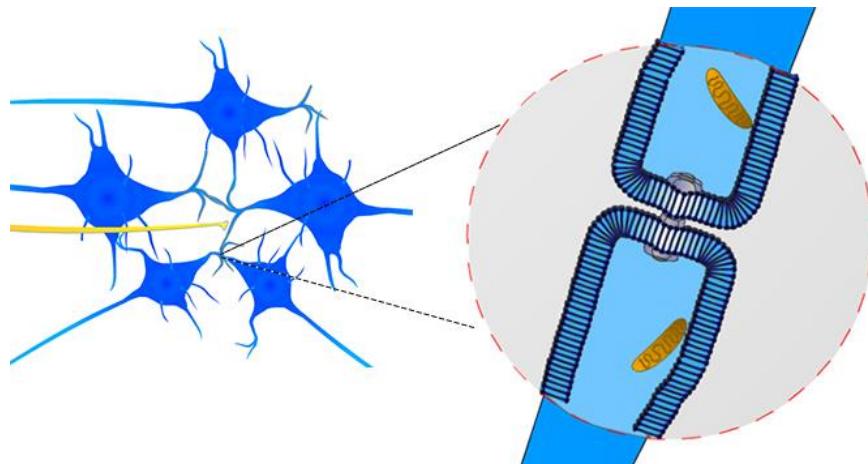


Figura 2.2.- La comunicación entre neuronas se realiza mediante impulsos eléctricos localizados en zonas de conexión denominadas 'sinapsis'. Esta conexión puede ser eléctrica, cuando las neuronas tienen un punto de contacto físico, como se muestra en la figura. Pero también puede ser de tipo química, cuando la transmisión se realiza por un compuesto químico que se traslada de una neurona a otra. Imagen recuperada de [7].

Los impulsos eléctricos debidos a la actividad cerebral se encuentran presentes en el cuero cabelludo a baja escala, pero detectable y medible, siendo este hecho descubierto por Hans Berger en 1924 [17]. Mediante el registro del potencial eléctrico – esto es, voltaje – en zonas particulares del cuero cabelludo en diferentes instantes, el electroencefalograma (EEG) representa gráficamente la actividad cerebral en función del tiempo, como se muestra en la Figura 2.1.

Para realizar la medición de señales eléctricas en el cuerpo humano, se utilizan múltiples electrodos, también denominados canales, colocándolos de

manera estratégica en diferentes zonas del cuerpo, de acuerdo con la actividad que se desee monitorear. La Figura 2.3 muestra un ejemplo del posicionamiento de los electrodos en un paciente para capturar actividad cerebral, es decir, la captura de un electroencefalograma. El registro de señales EEG se ha destinado desde sus inicios a aplicaciones clínicas y de investigación, tales como la detección de trastornos en el cerebro, monitoreo de estado cerebral, probar efectos de alguna droga, entre otros [17]. El surgimiento de los sistemas BCI introducen la utilización de la información sobre la actividad cerebral capturada por medio de EEG como un método de comunicación directa entre un humano y un sistema computacional o dispositivo electrónico.

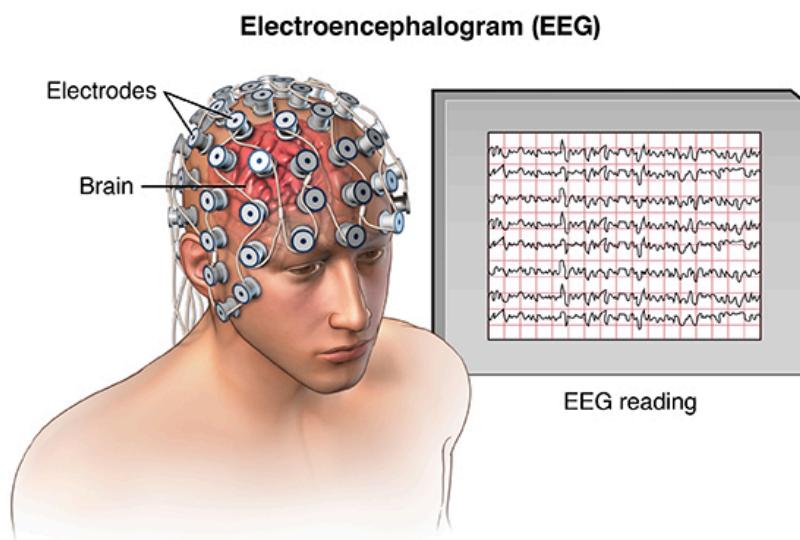


Figura 2.3.- El posicionamiento de los diferentes electrodos tiene el objetivo de obtener actividad neuronal relacionada con estado del cerebro, o imaginación de movimiento. Imagen recuperada de <https://speakingofresearch.com/tag/eeg/>

La actividad cerebral está asociada con distintos tipos de ondas cerebrales, que se observan como patrones de amplitud y frecuencia. Es decir, las ondas que componen a las señales EEG tenderán a mostrar que el comportamiento neuronal se puede categorizar en base a patrones específicos de voltaje en función del tiempo [18]. La clasificación de las ondas cerebrales de acuerdo a dichos patrones establece cuatro categorías principales [19]:

- Ondas *Delta*: Todas las ondas por debajo de 3.5 Hz, que ocurren durante el sueño profundo, durante la infancia y durante alguna enfermedad orgánica cerebral.
- Ondas *Theta*: Frecuencias entre 4 y 7 Hz, que ocurren durante la infancia, o durante estrés emocional en adultos.
- Ondas *Alfa*: Frecuencias entre 8 y 13 Hz, que ocurren durante un estado normal de relajación.
- Ondas *Beta*: Frecuencias de baja amplitud y frecuencia entre 13 y 30 Hz, que son afectadas por la actividad mental.

Adicional a estas cuatro ondas, existe una onda denominada Onda Mu, cuyo rango de frecuencias es entre los 8 Hz y 13 Hz y entre los 15 Hz y 25 Hz [16]. Esta onda se presenta en zonas del cuero cabelludo que se encuentran sobre la corteza sensorial y motora, y tienen ocurrencia durante la ejecución de un movimiento o la imaginación o visualización de éste [16]. Dado que el usuario no necesita ejecutar un movimiento físico, sino sólo imaginar que lo realiza, este tipo de señales crean una forma de comunicación viable entre un usuario y una computadora. La Figura 2.3 muestra ejemplos de los distintos tipos de ondas cerebrales que se pueden medir mediante dispositivos EEG.

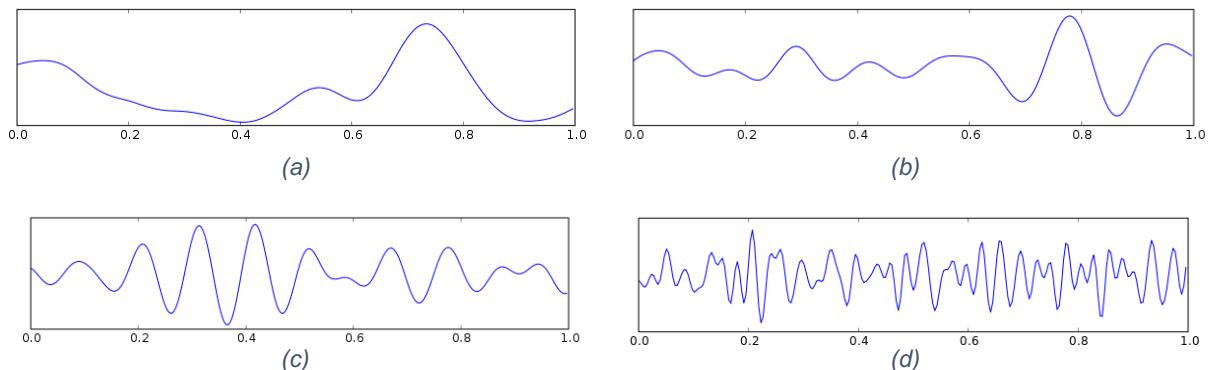


Figura 2.4.- a) Onda delta, comúnmente presente en un sujeto durante sueño profundo. b) Onda theta. Ésta ocurre en adulto en situaciones de estrés emocional. c) Onda alfa, presente en estado normal de relajación. d) Onda beta. Presente en sujetos dado un estímulo mental.

2.1.3 Base de datos A de Graz para captura de intención de movimiento

Para la BCI Competition en 2008 (mencionada en la sección 1.2), Brunner et al. registraron señales EEG posicionando 22 canales en 9 sujetos para registrar patrones relacionados con la imaginación de movimiento de cuatro partes diferentes del cuerpo, también denominadas clases: 1) mano izquierda, 2) mano derecha, 3) ambos pies, 4) lengua [13]. El mapeo de los canales se muestra en la Figura 2.3, y fue realizado de acuerdo con el sistema internacional 10-20. Este sistema tiene como objetivo establecer un estándar de posicionamiento de electrodos que permita equidistancia entre canales [20]. Su nombre se deriva del porcentaje de la distancia medida entre el nasión (depresión entre los ojos que une el hueso de la nariz con el hueso frontal del cráneo) y el inion (protuberancia notable en la parte inferior trasera del cráneo) respecto a la cual se posicionan los canales: el canal 1 se posiciona 10% de la distancia respecto del nasión, y los electrodos 4 al 20 se posicionan de manera que el 20% de la distancia total sea el espacio entre ellos, posicionando el canal 22 en el 10% restante respecto del inion. Este método se repite para posicionar los electrodos horizontales, iniciando con las referencias en las protuberancias detrás de las orejas.

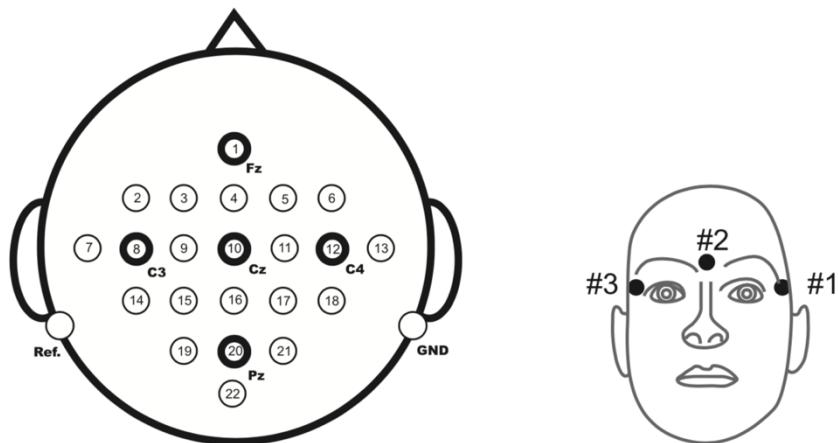


Figura 2.5.- Mapeo del posicionamiento de los 22 canales EEG (izquierda), y 3 canales EOG. Estos canales se posicionan de acuerdo al sistema internacional 10-20. Los 3 canales EOG se utilizan para capturar información del movimiento ocular, para aplicación de métodos de corrección de artífactos. Imagen de [13].

La captura de datos para cada uno de los 9 sujetos fue realizada en dos sesiones llevadas a cabo en diferentes días. Cada sesión inicia con 5 minutos de registro de señales EOG: 2 minutos con los ojos abiertos, 2 minutos con los ojos cerrados y 1 minuto de movimiento ocular. Posterior al registro de EOG, se realizan 6 series de 48 pruebas cada una, 12 para cada clase, consistiendo una sesión en 288 pruebas. Las 6 series fueron separadas por breves periodos de descanso. La Figura 2.6 muestra el esquema de tiempo de cada sesión. Una excepción a este esquema se da con el sujeto 4 en la primera sesión, que, debido a problemas técnicos, el registro de EOG sólo tiene los datos del movimiento ocular. La primera sesión contiene información sobre la clase que fue registrada durante cada prueba, y su propósito es usarse para entrenar a la máquina de clasificación, mientras que la segunda sesión se utiliza para la evaluación de esta. La sección 2.6 cubre a detalle la clasificación de intención de movimiento.

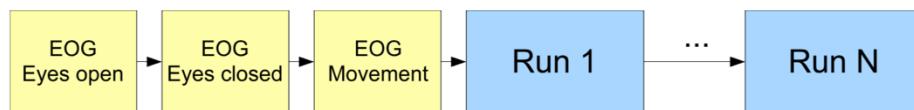


Figura 2.6.- Esquema de tiempo de cada sesión: 2 minutos de registro de EOG con los ojos abiertos, seguida de 2 minutos de registro de EOG con los ojos cerrados, y 1 minuto de registro de EOG con movimiento ocular. Posterior a esto, se realizan las 6 series de 48 pruebas cada una. Imagen de [13].

Cada prueba corresponde a la imaginación del movimiento de una sola parte del cuerpo. Cada sujeto se encontraba sentado en un sillón enfrente de una pantalla de computadora. Al inicio de cada prueba, se presentaba un tono acústico breve y una cruz de fijación en la pantalla de la computadora. Despues de 2 segundos, un indicador en forma de flecha instruía al sujeto a imaginar un movimiento. La flecha se presentaba apuntando en una de cuatro direcciones: izquierda, derecha, abajo y arriba, correspondiendo al movimiento de la mano izquierda, mano derecha, ambos pies y lengua, respectivamente. El sujeto realizaba la imaginación de movimiento hasta que la cruz de fijación desapareciera de la pantalla en el segundo 6 a partir del inicio de la prueba. La Figura 2.7 muestra el esquema temporal de cada prueba.

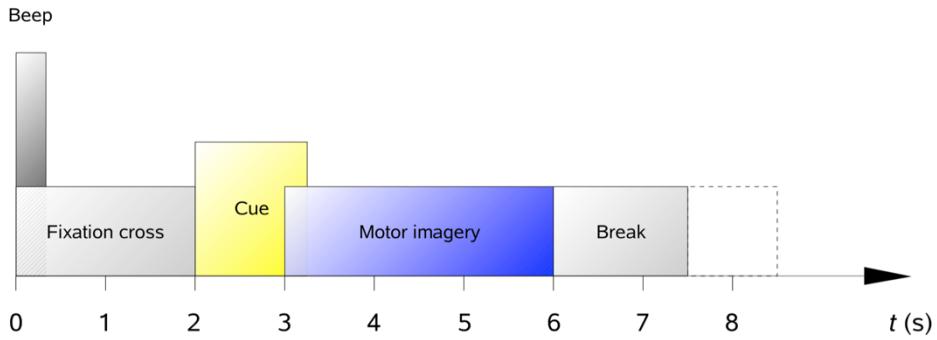


Figura 2.7.- Esquema temporal de cada prueba de imaginación de movimiento. En $t = 0$, se introduce un tono y una cruz de fijación. En $t = 2$, se muestra el indicador del movimiento a imaginar, y el sujeto desempeña dicha tarea por 3 segundos, hasta que la cruz de fijación desaparece en $t = 6$. Imagen de [13].

Los datos de las pruebas fueron obtenidos con una frecuencia de muestreo de 250 Hz, y se les aplicó un filtro pasa-banda entre 0.5 Hz y 100 Hz, además de un filtro de rechazo de banda de 50 Hz para eliminar el ruido presente en las conexiones [13].

2.2 Corrección de *artifacts*

2.2.1 Influencia del parpadeo y movimiento ocular en las señales EEG

Al ejecutar una medición de EEG, se busca obtener las señales eléctricas que se asocian con la actividad neuronal. No obstante, al obtener las señales, éstas usualmente se encuentran contaminadas por señales provenientes de fuentes diferentes al cerebro. A estas señales no deseadas se les denomina *artifacts* [12]. Los tipos de *artifacts* de señales EEG de acuerdo a su origen se muestran en la Figura 2.1. Los *artifacts* se dividen en dos grupos principales: externos, es decir, los que se originan por los instrumentos de medición, y los internos, los cuales se originan en el cuerpo. El enfoque de este proyecto en cuanto a la eliminación de

artifacts es a los internos causados por el movimiento ocular, es decir, las señales por electrooculograma (EOG) [4].

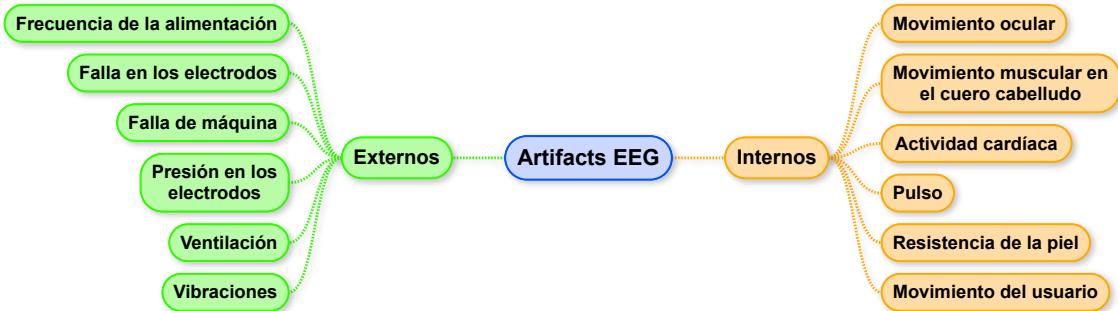


Figura 2.8.- Clasificación de los '*artifacts*' que se presentan en la captura de señales EEG. Datos adaptados de Savelainen, 2010 [11].

La Figura 2.9 muestra dos gráficas de amplitud de potencial eléctrico vs tiempo. Éstas son representaciones de las señales EEG obtenidas por dos electrodos ubicados en la parte frontal izquierda y frontal derecha del cuero cabelludo, correspondientes al canal 2 y canal 6 del mapeo de electrodos de la base de datos de Graz, respectivamente (ver sección 2.1.3) [11]. Como se indica en la figura, existen perturbaciones o picos en el potencial de las señales capturadas por los electrodos debido al movimiento de los globos oculares (Figura 2.9a), así como parpadeo (Figura 2.9b). La amplitud de las señales EOG puede ser órdenes de magnitud más grande que la de las señales EEG, inhibiendo la capacidad de

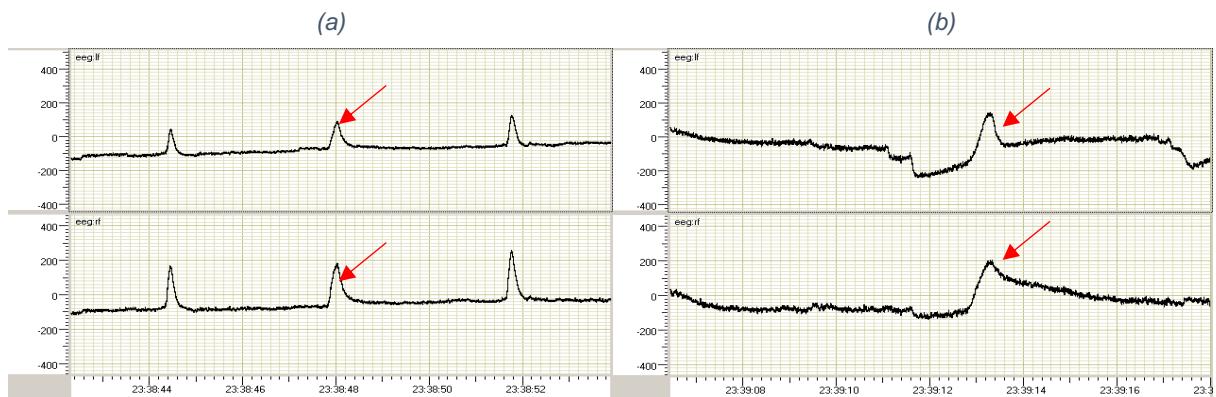


Figura 2.8.- Gráficas de señales EEG de voltaje (μ V en el eje 'y') en el tiempo. a) Los picos en las señales fueron causados por parpadeo. b) Los picos en las señales fueron causados por movimiento ocular. Imágenes recuperadas de Savelainen, 2010 [11].

interpretarlas correctamente [22]. Por esta razón, el desarrollo de técnicas para eliminar *artifacts* EOG con un alto grado de desempeño es importante para la aplicación práctica de las BCIs.

2.2.2 Análisis de componentes independientes (ICA)

El análisis de componentes independientes (ICA, por sus siglas en inglés) es un modelo que describe cómo una señal observada se compone de una mezcla de componentes considerados estadísticamente independientes (abreviados ICs, por sus siglas en inglés) [23]. Esto es, el valor medido de cada componente no depende del valor de los demás. Además, los componentes son variables latentes, esto es, no pueden ser observados de manera directa [23].

Para la comprensión del funcionamiento de la técnica ICA se presenta el concepto del “efecto de fiesta de coctel”. El ser humano tiene la capacidad de oír y distinguir una conversación entre una cacofonía de conversaciones y ruido de fondo [24]. Se han realizado diferentes estudios sobre esta habilidad con el fin de poder extraer las características principales que permiten el reconocimiento de las conversaciones y usarlas para construir dispositivos capaces de realizar esta misma tarea [24]. Considerando cada voz como una señal independiente, y asumiendo que se utilizan múltiples micrófonos para grabar la combinación de las voces, se pueden implementar métodos para separar las señales independientes originales a partir de las combinaciones obtenidas por los micrófonos [23]. A estos métodos se les conocen como métodos de separación ciega de fuentes (BSS, por sus siglas en inglés).

El método de ICA está basado en la BSS [23], puesto que describe la generación de los datos que se observan como una combinación de las fuentes originales, que, por ser variables latentes, no pueden ser observadas de manera directa. La Figura 2.9 ilustra el proceso de ICA como un método de BSS. La Figura

2.9a muestra tres señales de audio independientes en función del tiempo. La Figura 2.9b muestra tres señales de audio capturadas con tres micrófonos en tres ubicaciones distintas, que se asumen como una combinación lineal de las señales originales, denotadas como $s_1(t)$, $s_2(t)$ y $s_3(t)$. Así, las señales observadas se pueden expresar como funciones de tiempo [23]:

$$x_1 = a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t) \quad (1)$$

$$x_2 = a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t) \quad (2)$$

$$x_3 = a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t) \quad (3)$$

Los parámetros a_{ij} afectan la magnitud de las señales independientes, y están relacionadas con la distancia entre las señales observadas y los micrófonos [14]. En la práctica, no existe una manera directa de conocer la proporción de estos parámetros. Hyvarinen (2004) [23] establece que “el método de ICA usa la información sobre la independencia de las señales originales para encontrar los parámetros a_{ij} ”, basándose en tres restricciones fundamentales:

- 1) Los ICs se asumen como estadísticamente independientes.
- 2) Los ICs deben tener una distribución no gaussiana.
- 3) El número de parámetros incógnitos es igual al número de señales observadas.

Con estas tres restricciones, es posible obtener una estimación de los parámetros a_{ij} . Si se considera el sistema lineal en forma matricial, se tiene:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\mathbf{As} = \mathbf{x} \quad (4)$$

Tomando la matriz inversa de A , finalmente se pueden obtener las señales originales $s_1(t)$, $s_2(t)$ y $s_3(t)$, a partir de las combinaciones lineales $x_1(t)$, $x_2(t)$ y $x_3(t)$.

$$s = A^{-1}x \quad (5)$$

La Figura 2.9c muestra la obtención de una estimación de las señales originales por medio del método ICA. Es importante destacar que las magnitudes de las señales no coinciden con las originales, debido a que son multiplicadas por los escalares incógnitos correspondientes a_{ij} .

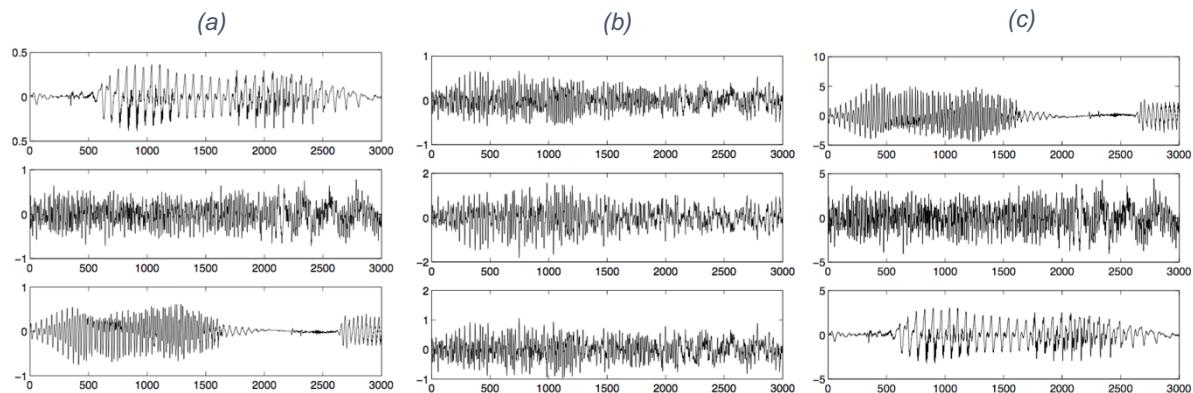


Figura 2.9.- a) Señales de audio independientes. b) Combinación de las señales, observada a través de tres entradas de audio. c) Estimación de las señales originales obtenida de las señales observadas. Imágenes recuperadas de Hyvärinen, et al., 2004 [13].

2.3 Normalización

2.3.1 Introducción a la normalización para señales EEG

Considerando la medición de señales a través de electrodos, es necesario tomar en cuenta la variabilidad de las señales obtenidas dado el equipo de medición, así como la propia naturaleza de las pruebas. Cuando se capturan señales provenientes del cuerpo humano, éstas pueden variar de sujeto de prueba a otro y entre pruebas con un mismo sujeto, debido a múltiples factores, entre los cuales, se

encuentra el posicionamiento de los electrodos, preparación de la piel, transpiración y temperatura [25]. Cuando se habla de señales eléctricas, siempre se debe considerar un punto de referencia para medir el voltaje que compone a la señal. Los factores mencionados tienden a causar que dicho valor de referencia varíe considerablemente [25]. Por esta razón, para la mejora en el desempeño del procesamiento de señales en cuanto a pruebas en distintos tiempos, o para la creación de sistemas que funcionen con más de un sujeto, se deben implementar técnicas de normalización.

Según Halaki y Ginn (2012) [25], “normalización se refiere a la conversión de una señal a una escala relativa a un valor conocido y repetible” (p. 175). Así, los métodos de normalización se pueden utilizar para mantener un desempeño constante a través de un sistema, convirtiendo todas las señales observadas a una referencia constante, con el fin de eliminar la influencia de factores externos que pudieran influir en las observaciones realizadas.

2.3.2 Normalización z-score

Según Abdi et al. [26] “constituye una medida sin unidades que se puede utilizar para comparar observaciones medidas con diferentes unidades”. Las señales normalizadas se pueden expresar como un vector dado por:

$$Z_n = \frac{Y_n - M}{\hat{S}} \quad (6)$$

donde Y_n representa un vector, M su valor medio y \hat{S} su desviación estándar. Con esta técnica se transforma el vector de datos a uno con promedio de cero y desviación estándar de 1.

2.4 Selección de canales

2.4.1 Métodos de selección de canales

El uso de múltiples canales para obtener señales EEG posibilita la obtención de suficientes datos para poder clasificar la intención de movimiento. Sin embargo, el uso indiscriminado de múltiples canales para ejercer esta clasificación puede resultar en una disminución en el desempeño del sistema BCI, debido a la introducción de ruido (*artifacts*), redundancia, y demanda de mayores recursos computacionales [27]. Debido a que los canales óptimos para obtener datos EEG puede variar de sujeto a sujeto [27], no es posible resolver este problema reduciendo el número de electrodos en el cuero cabelludo. Por este motivo, es deseable implementar métodos para seleccionar los canales que producen los datos más útiles de acuerdo con cada sujeto. A continuación, se presentan algunos métodos utilizados para las BCIs:

- 1) Criterios de Fisher (FC por sus siglas en inglés): Este método determina la magnitud con la que cierta característica en una señal se asocia con dos clases (intenciones de movimiento) diferentes [27], y se define como:

$$R_j(X) = \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2} \quad (7)$$

μ y s^2 representan la media y la varianza, respectivamente. El objetivo de este método de clasificación busca minimizar la varianza dentro de cada clase, así como maximizar la media entre las dos clases [27].

- 2) Información Mutua (MI por sus siglas en inglés): El método de información mutua se basa en posicionar las mejores características de acuerdo con la cantidad de información que comparte un conjunto de datos (información de los canales)

respecto a previos datos de entrenamiento [24]. En su estudio, Arvaneh et al. (2010) [27] implementan este método en tres pasos:

- 1) Inicializar un conjunto de características
 - 2) Computar la MI de las características con los valores obtenidos.
 - 3) Seleccionar las características que maximicen la MI
- 3) *Common Spatial Patterns* (CSP): el método de CSP, al igual que FC, se basa en la discriminación entre dos clases diferentes de datos EEG mediante la filtración de una señal de tal manera que se maximice la varianza de una clase, al mismo tiempo que se minimiza la varianza de la otra clase [27]. Tomando como objetivo la clasificación de dos intenciones de movimiento diferentes, como se establece en la sección 1.6, se definen \mathbf{X}_1 y \mathbf{X}_2 como dos matrices correspondientes a dos clases diferentes, de dimensiones $N \times T$, donde N denota el número de canales y T denota el número de muestras en cada canal. La covarianza espacial normalizada del electroencefalograma se define como [10]:

$$R_1 = \frac{\mathbf{X}_1 \mathbf{X}_1^T}{\text{trace}(\mathbf{X}_1 \mathbf{X}_1^T)} \quad R_2 = \frac{\mathbf{X}_2 \mathbf{X}_2^T}{\text{trace}(\mathbf{X}_2 \mathbf{X}_2^T)} \quad (8)$$

\mathbf{X}^T es la matriz transpuesta de \mathbf{X} y *trace* denota la suma de los elementos diagonales de la matriz introducida como su argumento [10]. La covarianza espacial compuesta se obtiene como:

$$R = \overline{R_1} + \overline{R_2} = \mathbf{U}_0 \Sigma \mathbf{U}_0^T \quad (9)$$

$\overline{R_1}$ y $\overline{R_2}$ son las covarianzas normalizadas promedio obtenidas a partir de la covarianza de todas las pruebas para cada clase. \mathbf{U}_0 es una matriz de vectores propios y Σ es una matriz diagonal de valores propios. Factorizando \mathbf{U}_0 , la transformación lineal P se define como:

$$P = \Sigma^{-\frac{1}{2}} \mathbf{U}_0^T \quad (10)$$

Esta matriz de transformación se puede utilizar para obtener las matrices \mathbf{S}_1 y \mathbf{S}_2 :

$$\mathbf{S}_1 = P \bar{\mathbf{R}}_1 P^T \quad \mathbf{S}_2 = P \bar{\mathbf{R}}_2 P^T \quad (11)$$

\mathbf{S}_1 y \mathbf{S}_2 comparten vectores propios y la suma de valores propios correspondientes siempre será la unidad [10]:

$$\mathbf{S}_1 = U \Sigma_1 U^T \quad \mathbf{S}_2 = U \Sigma_2 U^T \quad \Sigma_1 + \Sigma_2 = I \quad (12)$$

Se define a la matriz de proyección W como:

$$W = U^T P \quad (13)$$

A partir de la matriz de proyección, se puede transformar la matriz de las señales originales de EEG en valores no correlacionados:

$$Z = W X \quad (14)$$

X es la señal original, y al ser multiplicada por la matriz de proyección, se obtiene la matriz Z filtrada espacialmente. La inversa de la matriz de proyección, W^{-1} , se puede utilizar para reconstruir la señal original a partir de la señal filtrada Z :

$$X = W^{-1} Z \quad (15)$$

Las columnas de W^{-1} son denominadas patrones espaciales. Es deseable obtener aquellos que establecen mayor varianza en la primera clase y menor varianza en la segunda clase. Mediante la información de los patrones espaciales, se pueden seleccionar los canales con máxima correlación con las

tareas de imaginación de movimiento [10], específicos al sujeto de prueba, lo cual disminuirá el número de datos a procesar por el sistema BCI.

Para la construcción de la matriz de proyección W , si se ordena el vector propio U de acuerdo con los valores numéricos de la matriz diagonal de sus valores propios correspondientes, $\Sigma_{1,2}$, entonces las primeras y últimas filas corresponderán a los filtros espaciales más significativos para la discriminación entre clases [28]. Para obtener la importancia de cada canal, se calcula la norma-1 de cada columna de W , y se divide ésta respecto a la norma-1 de la matriz W completa [28]. Así, el puntaje de rango [0 1] de cada canal se puede obtener:

$$p(i) = \frac{\|W_{:,i}\|_1}{\|W\|_1} \quad (16)$$

Con estos puntajes, se pueden seleccionar aquellos canales con los valores de $p(i)$ más altos, bajo el supuesto de su mayor relevancia respecto de los canales con puntajes más bajos, dadas las características que presentan, y que permiten una mejor discriminación entre dos clases de intención de movimiento.

2.5 Extracción de características

Como se mencionó en la sección 2.1, las señales que se relacionan con intención de movimiento comparten ciertos valores de frecuencia y amplitud. A menudo es deseable obtener ciertas características de una señal particular. La información que nos puede dar la señal en cuanto a la variación de su amplitud en función de su frecuencia tiende a ser útil, como los picos característicos y sus componentes de frecuencia [31].

2.5.1 Transformada de Fourier

La transformada de Fourier tiene la forma:

$$F(k) = \int_{-\infty}^{\infty} f(x) \exp(-i2\pi kx) dx \quad (17)$$

Este sistema permite convertir una función de tiempo a una función de frecuencia, conocida como espectro de la función, lo que permite obtener información sobre las características ya mencionadas, así como implementar filtros de frecuencias no deseadas (métodos de corrección de ruido). Además, dada su forma discreta, la transformada discreta de Fourier (DFT, por sus siglas en inglés), es posible implementar algoritmos en una computadora que aproximen a la transformada de Fourier continua:

$$DFT\{f\}_m = \sum_{n=0}^{N-1} f_n \exp\left(-\frac{i2\pi mn}{N}\right) \quad (18)$$

La Figura 2.10 muestra la representación de una función coseno en el dominio del tiempo, así como su transformada discreta de Fourier, utilizando una técnica que optimiza recursos computacionales denominada transformada rápida de Fourier (FFT, por sus siglas en inglés). Como se puede observar en la figura, cuando existe una combinación de frecuencias, no es posible utilizar la información del espectro para determinar el instante en el que se presentan ciertos componentes de frecuencia.

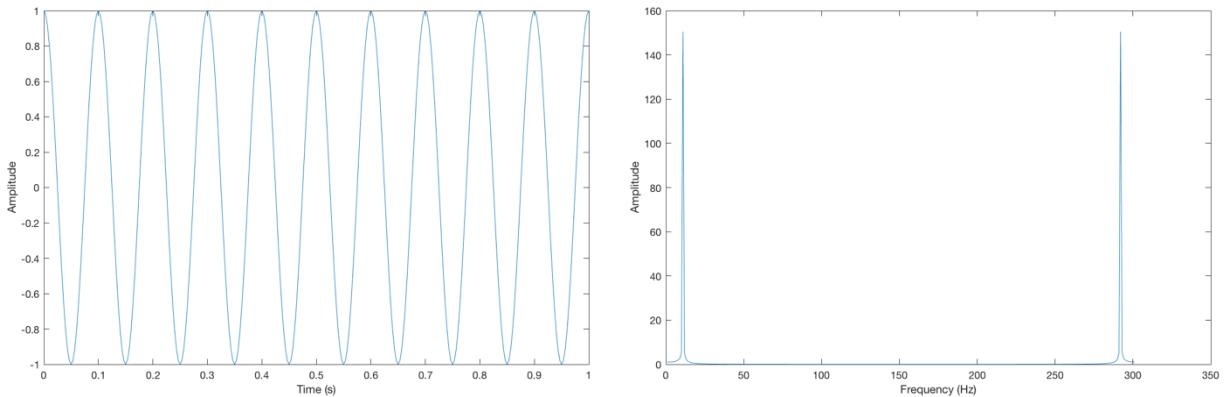


Figura 2.10.- Función $f(x) = \cos[(2\pi 10/300)x]$ utilizando el software MATLAB de Mathworks. MATLAB utiliza la función `fft` para calcular la transformada discreta de Fourier. Las segunda gráfica muestra los componentes de frecuencia de la señal, es decir, el espectro de la función.

2.5.2 Transformada ondícula (wavelet)

La transformada ondícula, o transformada *wavelet*, es un sistema diseñado para proveer información de tiempo y de frecuencia [31], ofreciendo una técnica efectiva para obtención de características al compararse con la transformada de Fourier. Su forma es:

$$Wf(s, u) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt \quad (19)$$

ψ es denominada la wavelet madre, una función delimitada en el tiempo, de la cual se obtiene el nombre wavelet, que significa pequeña onda [32]. Como se observa de la Ecuación 18, la función resultante de la aplicación de una transformada wavelet es una función de dos parámetros: un parámetro u conocido como de traslación y el parámetro s para escalado. Con la transformada wavelet continua se puede mapear a una frecuencia y tiempo específicos. La transformada wavelet tiene la característica que el parámetro s es inversamente proporcional a la frecuencia, por lo que, si se desea analizar frecuencias bajas, es necesario utilizar valores elevados para s , y valores elevados de frecuencias requieren valores pequeños de s [32].

2.5.3 Método de ventana móvil

El método de ventana móvil consiste en obtener segmentos de duración predeterminada de una señal mediante una ventana que se ‘desliza’ a través de la totalidad de la señal. A estos segmentos se les conoce como ventanas, y al método se le denomina de “ventana móvil” debido a que esta ventana se desliza en el tiempo para obtener diferentes segmentos de la señal. La Figura 2.11 muestra la aplicación del método de segmentación de una señal EEG usando ventanas móviles.

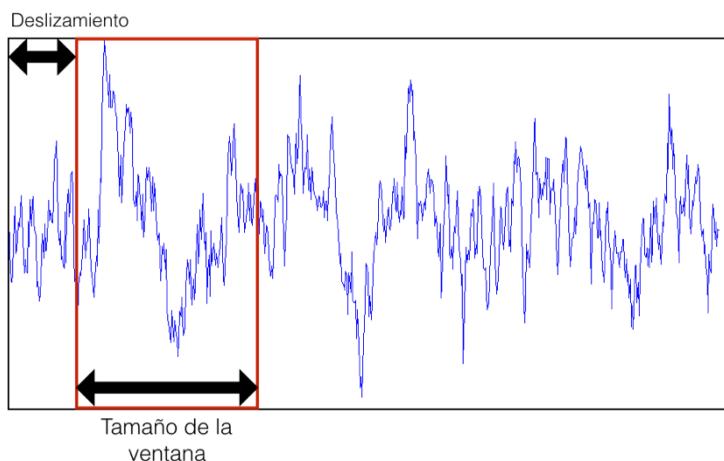


Figura 2.11.- Método de ventana móvil para segmentación de señales EEG.
Imagen recuperada de [2].

La latencia del inicio de cada segmento respecto del inicio de la señal puede ser mayor o menor a la duración de cada segmento, lo cual resulta en un traslape negativo o positivo, siendo negativo cuando los segmentos de las señales coinciden en secciones menores, y positivo cuando existe un pequeño espacio entre cada segmento, el cual no se considera durante la segmentación. Si la latencia se compone de múltiplos de la duración completa del tamaño de cada ventana, no existe traslape. Ejemplos de estas segmentaciones se muestran en la Figura 2.12.

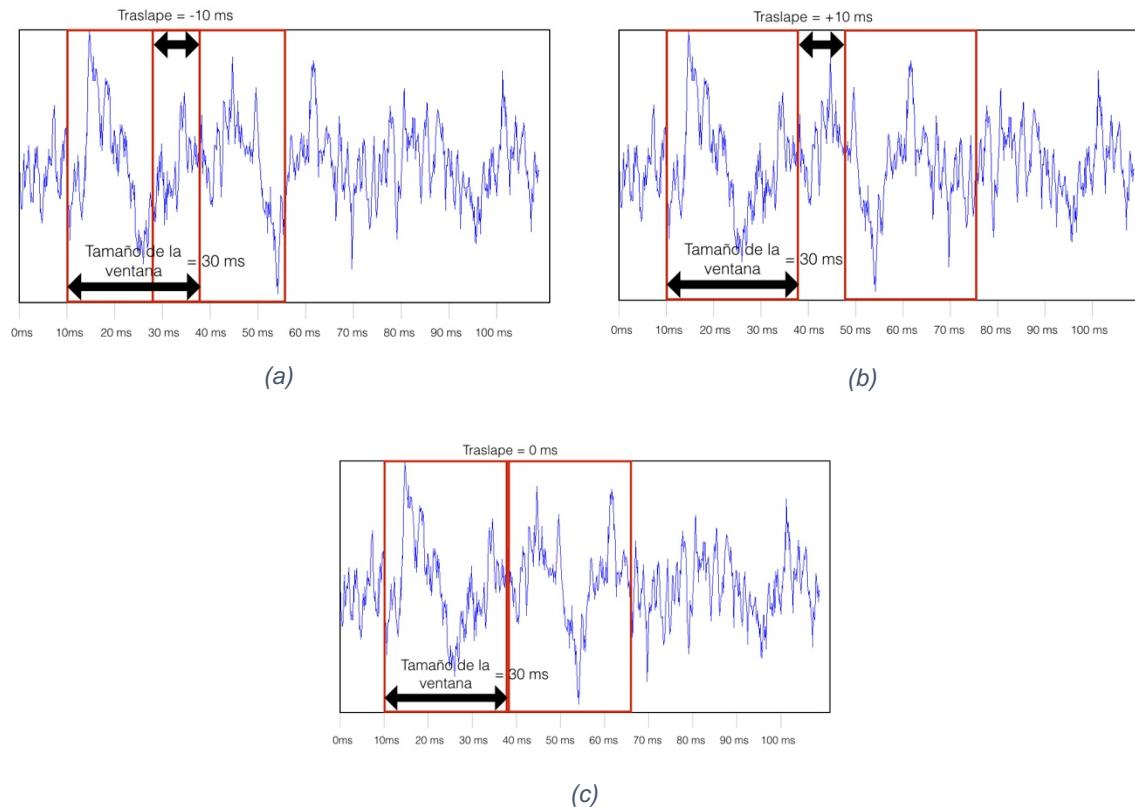


Figura 2.12.- a) Ventana móvil con traslape de valor -10 ms (Traslape negativo). b) Ventana móvil con traslape de valor +10 ms (Traslape positivo). c) Ventana móvil sin traslape. Imágenes recuperadas de [2].

2.6 Clasificación de intención de movimiento

Para que un dispositivo con una BCI pueda ejecutar una tarea en base a un comando producido por una intención, debe poder discriminar entre diferentes intenciones que el usuario pueda generar. Para realizar esta tarea, se depende de métodos implementados en algoritmos a ejecutar por una computadora. Como se ha mencionado en la sección 2.5, la actividad cerebral asociada con intención de movimiento presenta características que pueden ser extraídas. Los algoritmos permiten a un sistema computacional identificar patrones en las señales EEG [33] y asignarlas a clases identificables por medio de estas características.

2.6.1 AdaBoost

El algoritmo de clasificación AdaBoost utiliza la idea de construir una regla de predicción robusta, o de alta precisión a partir de un conjunto de reglas débiles [33]. En base a esta idea, AdaBoost crea un clasificador fuerte mediante la introducción de un número de clasificadores débiles, donde se asigna un valor escalar a cada clasificador débil en base al margen de error encontrado para cada hipótesis. El algoritmo para AdaBoost se describe a continuación, adaptado de [33]:

Sean: datos de entrenamiento $(x_1, y_1) \dots (x_m, y_m)$ donde $x_i \in \mathcal{H}$, $y_i \in \{-1, +1\}$.

Iniciar: $D_1(i) = \frac{1}{m}$ para $i = 1, \dots, m$.

Desde $t = 1, \dots, T$:

- Entrenar un algoritmo de aprendizaje débil utilizando la distribución D_t .
- Obtener clasificador débil $h_t: \mathcal{H} \rightarrow \{-1, +1\}$.
- Enfoque: encontrar h_t con bajo porcentaje de error:

$$\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i] \quad (20)$$

- Escoger $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\varepsilon_t}{\varepsilon_t} \right)$
- Actualizar, desde $i = 1, \dots, m$:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \quad (21)$$

donde Z_t es un actor de normalización escogido de tal manera que D_{t+1} sea una distribución.

1. Computar hipótesis final:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right) \quad (22)$$

2.6.2 Máquinas de vectores de soporte

Los algoritmos de las máquinas de vectores de soporte buscan generar un hiperplano a partir de datos de entrenamiento. Como se muestra en la Figura 2.9, datos correspondientes a dos clases diferentes son separados por un hiperplano, el cual es generado de tal manera que los márgenes entre datos de entrenamiento pertenecientes a dos clases distintas sean maximizados [34].

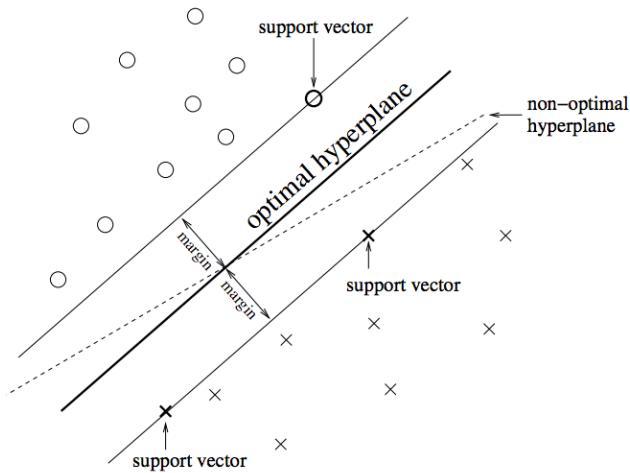


Figura 2.13.- Generación de un hiperplano utilizando SVM. El hiperplano óptimo maximiza la distancia entre los datos de entrenamiento [33].

Capítulo 3 – Metodología

La operación del sistema BCI de este proyecto está dividida en tres etapas: preprocessamiento, extracción de características y clasificación. Estas etapas se ilustran en la Figura 3.1. La etapa de preprocessamiento concentra los tres bloques fundamentales mencionados en la sección 1.6: corrección de *artifacts*, normalización y selección de canales. Después del preprocessamiento, el sistema realizará una extracción de características de las señales. La etapa de clasificación se divide en dos procesos: entrenamiento y pruebas. Durante el entrenamiento, las características se introducirán a la máquina de clasificación. En el proceso de pruebas, la máquina utilizará las características introducidas en el entrenamiento para distinguir entre las dos clases de imaginación de movimiento con un conjunto nuevo de datos, evaluando el desempeño comparando la clasificación con la clase correspondiente a la señal. Las señales de entrenamiento y las señales de prueba se seleccionarán de acuerdo con la estructura de la base de datos de Graz (véase la sección 2.1.3). A continuación, se describe el procedimiento de la implementación de las etapas de este proyecto.

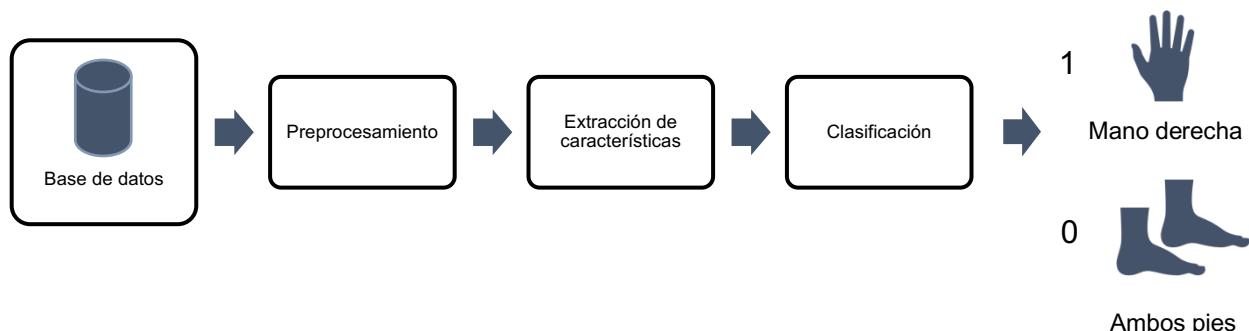


Figura 3.1.- Diagrama esquemático de las etapas de operación del sistema BCI a desarrollar durante este proyecto. Inicia con el preprocessamiento de las señales provenientes de la base de datos, seguido de la extracción de características, y finaliza con la etapa de clasificación, en la que se discrimina entre dos intenciones de movimiento: mano derecha y ambos pies.

3.1 Preprocesamiento

La etapa de preprocesamiento es el enfoque principal de este proyecto. La Figura 3.2 muestra la ramificación de esta etapa en los tres bloques fundamentales (ver sección 1.6). Los métodos por utilizar para implementar los tres bloques se muestran, así como los resultados que se esperan obtener de la implementación de cada bloque. El resultado general esperado de esta etapa es la elevación del porcentaje de aciertos durante el proceso de evaluación de la máquina de clasificación.

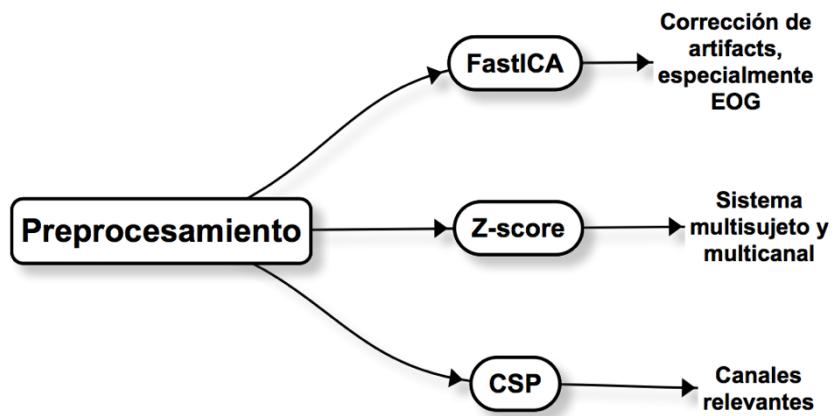


Figura 3.2.- Etapa de preprocesamiento del sistema BCI. Para la corrección de ‘artifacts’, se utilizará el algoritmo Fast ICA de Hyvärinen [32]. Se espera obtener señales limpias de ‘artifacts’ al final de esta etapa de preprocesamiento. Con la técnica Z-score, se espera establecer una referencia constante en todas las señales, observable en valores similares de amplitud en las señales entre pruebas y entre sujetos. Con CSP, se espera encontrar los canales más relevantes para la clasificación, rechazando los canales restantes, mejorando la velocidad de cada etapa de operación del sistema BCI.

3.1.1 Corrección de *artifacts* utilizando algoritmo FastICA

Para la corrección de *artifacts*, se utilizará el algoritmo FastICA, introducido por Hyvärinen y Oja [35] para aplicar BSS (ver sección 2.2.2) mediante el método de análisis de componentes independientes. Este algoritmo utiliza iteraciones de punto fijo que convergen a los componentes independientes de un conjunto de señales observadas. Se ha demostrado su alta eficiencia, obteniendo convergencia

entre 10 y 100 veces más rápido que algoritmos ICA preexistentes [35]. Para este proyecto, se establece la suposición de que los componentes independientes de las señales EEG de la base de datos de Graz corresponden a las señales cerebrales presentes en cada canal, así como las señales que corresponden a los *artifacts* por EOG, y las señales observadas son una combinación lineal de estas componentes. El algoritmo FastICA opera de la manera descrita por Hyvärinen y Oja [35], adaptada a continuación.

Para la estimación de un componente independiente:

1. Tomar un vector inicial aleatorio $\mathbf{w}(0)$ de norma vectorial 1. Sea $k = 1$.
2. Sea $\mathbf{w}(k) = E\{\mathbf{x}(\mathbf{w}(k-1)^T \mathbf{x})^3\} - 3\mathbf{w}(k-1)$. E es el valor esperado de $\mathbf{x}(\mathbf{w}(k-1)^T \mathbf{x})^3$.
3. Dividir $\mathbf{w}(k)$ por su norma.
4. Si $|\mathbf{w}(k)^T \mathbf{w}(k-1)|$ no se acerca lo suficiente a 1, definir $k = k + 1$ y volver al paso 2. De otra manera, definir la salida como $\mathbf{w}(k)$.

Para la estimación de n componentes independientes:

Ejecutar el algoritmo descrito anteriormente n veces, agregando una proyección de ortogonalización dentro del ciclo, en el paso 3:

3. Sea $\mathbf{w}(k) = \mathbf{w}(k) - \bar{\mathbf{B}}\bar{\mathbf{B}}^T\mathbf{w}(k)$, donde \mathbf{B} representa las columnas de la matriz de mezcla encontradas en iteraciones previas. Dividir $\mathbf{w}(k)$ por su norma.

El algoritmo FastICA se implementará en segmentos de señales multicanal a través de la función `fastica`, el programa escrito por Hyvärinen y Oja para la ejecución del algoritmo de punto flotante introducido en [35]. Para esto, se extraerá la base de datos de Graz, segmentando las señales en pruebas, y ejecutando la función `fastica` para obtener los componentes independientes (ICs). Se rechazarán los ICs que correspondan a *artifacts*, y se reconstruirán las señales EEG con los componentes independientes que corresponden a la actividad cerebral, como lo muestra la Figura 3.2. Mediante un método de análisis estadístico

denominado *FASTER*, desarrollado por Nolan et al. [35], se eliminarán los ICs correspondientes a *artifacts*. A través de la observación de las gráficas de las señales después del proceso de corrección de *artifacts*, así como su porcentaje de desempeño después de la etapa de clasificación, se determinará la efectividad de esta parte del preprocesamiento para mejorar el sistema BCI.

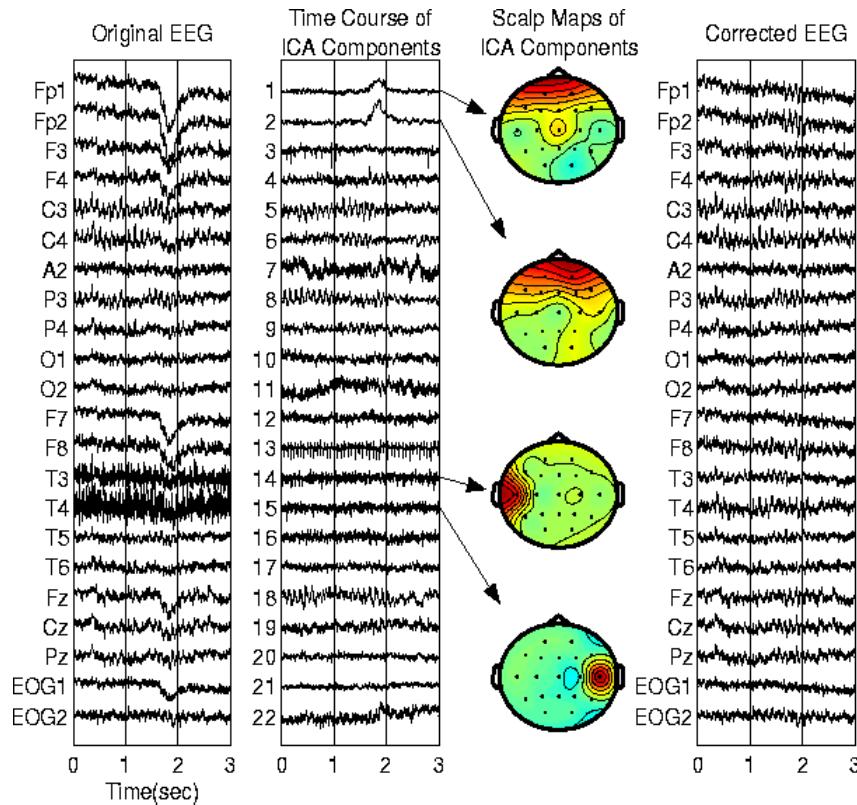


Figura 3.3.- Corrección de la señal EEG multicanal mediante la extracción de componentes independientes con la técnica ICA. Se observa la importante eliminación de la influencia de artifacts en los canales T3 y T4, así como Fp1, Fp2, F3, F4, F7 y F8. Imagen recuperada de [36].

3.1.2 Normalización por Z-score

Para aplicar la normalización Z-score, se implementa la ecuación 22 para obtener una matriz de señales normalizadas s' , a partir de las señales obtenidas de la base de datos:

$$s' = \frac{s - \mu}{\sigma} \quad (22)$$

Donde $\mu = \frac{1}{P} \sum_{k=1}^P s$, $\sigma = \left(\frac{1}{P} \sum_{k=1}^P s\right)^{\frac{1}{2}}$ y P denota el número de pruebas ejecutadas.

Se espera que la implementación de este bloque permita disminuir la variación de amplitud de las señales; tanto entre sujetos como entre sesiones de pruebas con un mismo sujeto.

3.1.3 Selección de canales mediante el método de CSP

El tercer bloque del proyecto es la selección de canales. Después de la corrección de *artifacts* y normalización de las señales, se buscará obtener los canales de la matriz de señales s' que posean mayor correlación con las tareas de imaginación de movimiento. De la ecuación 15, aplicada a las señales normalizadas:

$$s' = W^{-1}Z \quad (23)$$

Esta matriz se utilizará para seleccionar los canales relevantes cuyos datos se utilizarán para las etapas subsecuentes de la operación del sistema BCI. Se espera que el rechazo de canales irrelevantes reduzca significativamente el número de operaciones realizadas, así como una menor influencia externa de factores externos a las señales EEG al desempeño de la clasificación. De igual manera, se espera un mínimo impacto negativo del desempeño de los clasificadores ante un reducido número de canales, debido a que los canales más importantes contienen características suficientes para la discriminación entre dos intenciones de movimiento.

Para estudiar los efectos de la selección de canales, se realizarán pruebas de este bloque específico, utilizando los porcentajes de desempeño de las máquinas de clasificación y comparándolos con los resultados obtenidos

3.2 Extracción de características

Para la segmentación y extracción de características de las señales EEG, se utilizarán los métodos previamente implementados por Marrufo y Giorgana [2], en los que se utiliza el método de ventana móvil en conjunto con la transformada wavelet discreta (DWT por sus siglas en inglés) para obtener la matriz de características que será introducida a la máquina de clasificación, junto con un vector de etiquetas correspondientes a las dos clases a discriminar, cuya información lógica se presenta en la base de datos de Graz. La Figura 3.4 muestra el ejemplo de un supuesto para la extracción de características de las señales, en las que se obtienen 100 señales EEG de prueba, de las cuales 50 son para la clase 1 y 50 son para la clase 0. Las filas de la matriz mostrada en la Figura 3.3a son los vectores de características de cada señal dividida en M segmentos, donde N es el número de coeficientes devueltos por la computación de DWT al extraer características de la porción de señal segmentada por la ventana de duración T. Además, se asume cada señal está compuesta por 22 canales. El vector de etiquetas se genera a partir del valor conocido de la clase de la señal con las señales de entrenamiento.

$$\begin{array}{c}
 (a) \\
 \begin{matrix}
 1 & \left[\begin{matrix} 1 \times 22N \end{matrix} \right] \\
 2 & \left[\begin{matrix} 1 \times 22N \end{matrix} \right] \\
 & \vdots \\
 M & \left[\begin{matrix} 1 \times 22N \end{matrix} \right] \\
 & \vdots \\
 2M & \left[\begin{matrix} 1 \times 22N \end{matrix} \right] \\
 & \vdots \\
 & \left[\begin{matrix} 1 \times 22N \end{matrix} \right] \\
 & \vdots \\
 100M & \left[\begin{matrix} 1 \times 22N \end{matrix} \right]
 \end{matrix} \\
 \text{Matriz de características}
 \end{array}
 \qquad
 \begin{array}{c}
 (b) \\
 \begin{matrix}
 1 \\
 1 \\
 1 \\
 1 \\
 & \vdots \\
 0 \\
 0 \\
 0 \\
 0 \\
 0
 \end{matrix} \\
 \text{Vector de etiquetas}
 \end{array}$$

Figura 3.4.- a) La matriz de características consiste en vectores de coeficientes obtenidos por la computación de DWT en los segmentos obtenidos por ventanas móviles. b) Para cada una de las 100 señales del supuesto de este ejemplo, existe una etiqueta dependiendo de la clase correspondiente, para lo cual se presenta el vector de etiquetas.

3.3 Clasificación

Habiendo obtenido la matriz de características y el vector de etiquetas, se procederá a generar un modelo utilizando la máquina de clasificación AdaBoost. La Figura 3.5 muestra el proceso de generación del modelo durante la etapa de entrenamiento.

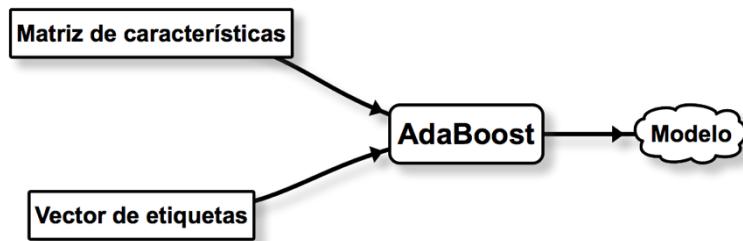


Figura 3.5.- Clasificación por AdaBoost durante el proceso de entrenamiento. La máquina de clasificación construirá un modelo a partir de la matriz de características y el vector de etiquetas correspondientes a las señales extraídas.

El modelo generado será reintroducido a la máquina de clasificación durante el proceso de pruebas. Este proceso de pruebas involucra una nueva matriz de características, con señales que no han sido previamente utilizadas en el entrenamiento. La Figura 3.6 muestra el proceso de pruebas de la máquina de clasificación AdaBoost.

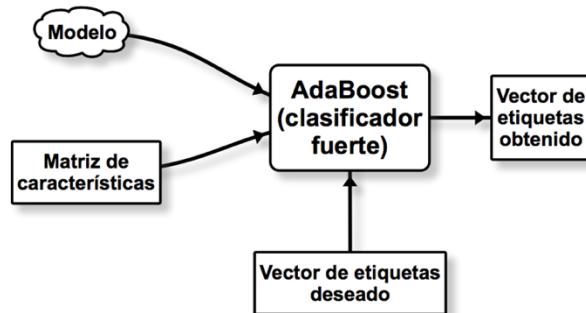


Figura 3.6.- Proceso de pruebas para la máquina de clasificación AdaBoost. Utilizando el modelo generado durante el entrenamiento, una nueva matriz de características y el vector de etiquetas deseado, se comparará el desempeño del sistema mediante la generación de un vector de etiquetas por el clasificador.

El desempeño del sistema BCI se basará en la capacidad de discriminar correctamente entre las dos clases establecidas. Este valor es cuantificable mediante el porcentaje de aciertos dado por el vector de etiquetas correspondientes a la clase, y el vector generado por la máquina de clasificación. El proceso experimental consiste en la ejecución de pruebas con las que se evaluará el desempeño del sistema BCI, principalmente basándolo en el porcentaje de aciertos obtenidos de la Ecuación 24, después de cada etapa del preprocesamiento, combinando las etapas conforme se realizan pruebas individuales de cada etapa.

$$\% \text{ acierto} = \frac{1 - \sum |actual \text{ label vector} - desired \text{ label vector}|}{size \text{ of } actual \text{ label vector}} \quad (24)$$

Capítulo 4 – Desarrollo del Proyecto

En este capítulo se describe el desarrollo del proyecto de acuerdo con los objetivos establecidos en la Sección 1.6 y la metodología propuesta en el capítulo 3. Se inicia con la descripción de la obtención de las señales obtenidas de la base de datos correspondiente. A continuación, se describen los pasos para crear los algoritmos necesarios para ejecutar los tres bloques de preprocesamiento, para generar bases de datos nuevas, a partir de las cuales se ejecutaron las etapas de extracción de características y clasificación anteriormente descritas.

4.1 Base de datos de Graz

Para cargar la información necesaria de la base de datos de Graz en el software MATLAB, se optó por generar dos variables por sujeto. Cada variable contiene las pruebas correspondientes a una clase (intención de movimiento). Cada prueba consiste en una matriz de 750 filas y 22 columnas. Las filas representan los registros de amplitud de voltaje en los electrodos en un punto de tiempo específico y las columnas representan cada uno de los 22 canales usados para registrar la señal multidimensional. De acuerdo con la descripción de la base de datos (ver Sección 2.1.3), el registro de la intención de movimiento en cada prueba ocurre durante 3 segundos. Tomando en cuenta la frecuencia de muestreo de 250 Hz usada para el registro de las señales EEG, así como la duración mencionada, se requirieron 750 datos temporales para la segmentación correcta de la señal relacionada con intención de movimiento. Las pruebas desempeñadas en cada sesión fueron registradas sin interrupción, por lo que la base de datos contiene los índices del inicio de cada prueba realizada, así como etiquetas para identificar la intención de movimiento que fue registrada en dicha prueba. La base de datos también contiene la información ocular registrada en cada prueba, utilizando 3 canales EOG para registrar señales eléctricas a la misma frecuencia de muestreo.

Con esta información, se puede construir la información de cada prueba, con la matriz mostrada en la Figura 4.1.

$$x = \begin{bmatrix} x_{1,1} & \cdots & x_{1,22} \\ \vdots & \ddots & \vdots \\ x_{750,1} & \cdots & x_{750,22} \end{bmatrix} \quad EOG = \begin{bmatrix} e_{1,1} & e_{1,2} & e_{1,3} \\ \vdots & \vdots & \vdots \\ e_{750,1} & e_{750,2} & e_{750,22} \end{bmatrix}$$

Figura 4.1.- Formato de una prueba individual, que abarca la parte de la señal en la que se registra una intención de movimiento. Teniendo una duración de tres segundos y registrando voltaje a 250 Hz, la señal discreta cuenta con 750 datos. Las columnas representan los 22 canales usados para el registro de la señal EEG. La matriz EOG contiene la información del movimiento ocular registrada en la duración de la prueba.

Para el registro de las 72 señales correspondientes a una clase, se construyó un arreglo tridimensional, como lo muestra la Figura 4.2, en el que se organizaron las 72 matrices de pruebas, del formato descrito anteriormente, en el que la tercera dimensión contiene dichas pruebas, que se pueden visualizar en forma gráfica como si fueran ‘páginas de libro’.

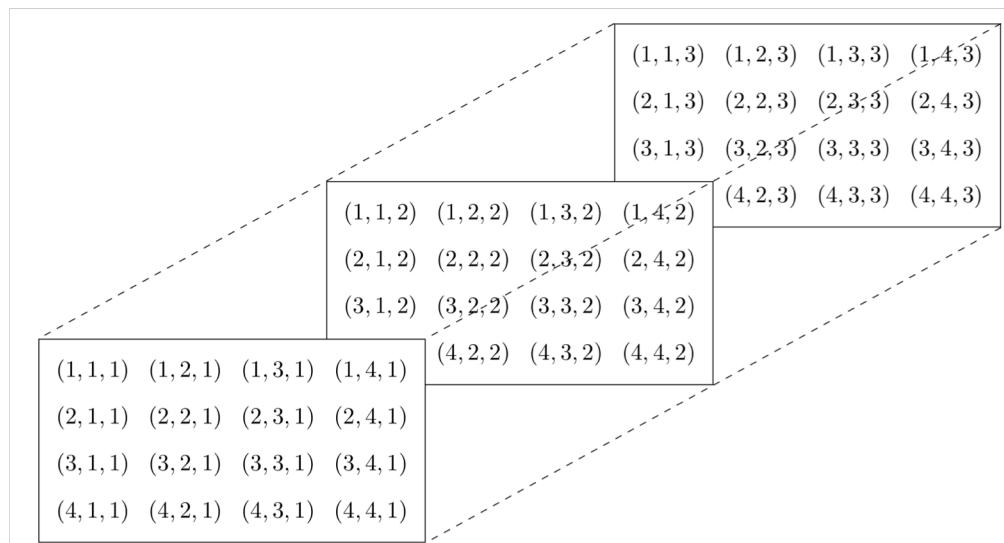


Figura 4.2.-Formato de las pruebas almacenadas en cada variable. Estilo ‘página de libro’, las matrices bidimensionales corresponden a una prueba, mientras que la tercera dimensión almacena cada una de las 72 pruebas individuales.

4.2 Preprocesamiento: Corrección de *artifacts*

4.2.1 Algoritmo FastICA para descomposición de señales en sus componentes independientes

Para la corrección de *artifacts*, se inició con la descomposición de la señal multidimensional en sus componentes independientes utilizando el algoritmo FastICA de Hyvarinen [32]. Dados 22 canales, considerados como combinaciones lineales de las señales fuentes, se establece el supuesto que las señales EEG están combinadas con fuentes de voltaje no provenientes del cerebro, y las señales independientes pueden ser separadas por la aplicación de un método de ICA. La documentación de las entradas y salidas de FastICA se describen en la Tabla 4.1.

Tabla 4.1.- Función FastICA desarrollada por Hyvarinen y Oja [34]. Esta función toma como parámetro una matriz cuyas filas se asumen como una combinación lineal de señales estadísticamente independientes, y mediante un algoritmo de punto flotante fijo, estima los componentes independientes con una alta velocidad.

FastICA		
Parámetros	mixedsig	Una matriz correspondiente a la señal EEG con múltiples canales.
Salidas	I	Los componentes independientes estimados por el algoritmo.
	A	La matriz de mezcla, que multiplica a los ICs para obtener la matriz original.
	W	La matriz de separación, que multiplica a mixedsig para obtener los componentes independientes.

Como lo indica la Tabla 4.1, al introducir una matriz como parámetro, que en este caso corresponde a cada una de las pruebas individuales segmentadas como en la Figura 4.2, la función FastICA devuelve tres salidas: la matriz I, con las mismas dimensiones que la señal de entrada, contiene los componentes independientes estimados. El algoritmo asume que la señal contiene las combinaciones lineales en 44

las filas, a diferencia de las columnas, por lo que las matrices correspondientes a las pruebas fueron introducidas como transpuestas. Es importante destacar que el algoritmo de FastICA no siempre converge al número de dimensiones que recibe, por lo que se tuvo que establecer un ciclo de repetición para esta función para asegurar la convergencia de 22 componentes independientes.

Bajo el supuesto de la combinación lineal de las señales EEG, se asume que los componentes que corresponden a ondas cerebrales, así como los componentes que corresponden a ruido, se encontrarán separados y ubicados en la matriz I . La implicación de esta hipótesis, de ser verdadera, es que se puede proceder a identificar las fuentes de ruido en alguno de los componentes independientes, y en el evento de identificarlos, eliminarlos, y recombinar la matriz EEG original, utilizando la matriz de recombinación A obtenida por FastICA.

La Figura 4.3 muestra el proceso de eliminación de ICs correspondientes a *artifacts* y la posterior reconstrucción utilizando los componentes de la matriz I' , la cual excluye dichos componentes. La matriz de recombinación A' excluye la columna correspondiente a los coeficientes de combinación lineal de dicho componente independiente. Así, la matriz de señales originales EEG, denominada

$$I' = \begin{bmatrix} i_{1,1} & i_{1,2} & \dots & i_{1,750} \\ i_{2,1} & \cancel{i_{2,2}} & & i_{2,750} \\ \vdots & & & \\ i_{22,1} & i_{22,2} & \dots & i_{22,750} \end{bmatrix}$$

$$A' = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,22} \\ a_{2,1} & \cancel{a_{2,2}} & \dots & a_{2,22} \\ \vdots & & \dots & \\ a_{22,1} & a_{22,2} & \dots & a_{22,22} \end{bmatrix}$$

IC₂ es un artifact

Figura 4.3.- Ejemplo: si bajo procesos de análisis estadísticos, se encuentra que el componente independiente i_2 corresponde a un *artifact*, éste es descartado en I . También se descarta la columna que corresponde a su matriz de combinación en A .

S , reconstruida excluyendo los componentes que corresponden a *artifacts* resulta como en la Figura 4.4.

$$S = A'I' = \begin{bmatrix} a_{1,1} & a_{1,3} & \dots & a_{1,22} \\ a_{2,1} & a_{2,3} & \dots & a_{2,22} \\ \vdots & & & \\ a_{22,1} & a_{22,3} & \dots & a_{22,22} \end{bmatrix} \begin{bmatrix} i_{1,1} & i_{1,2} & \dots & i_{1,750} \\ i_{3,1} & i_{3,2} & \dots & i_{3,750} \\ \vdots & & & \\ i_{22,1} & i_{22,2} & \dots & i_{22,750} \end{bmatrix}$$

$$= \begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,750} \\ s_{2,1} & s_{2,2} & \dots & s_{2,750} \\ \vdots & & & \\ s_{22,1} & s_{22,2} & \dots & s_{22,750} \end{bmatrix}$$

Figura 4.4.- Reconstrucción de las señales EEG utilizando las matrices de recombinación y reconstrucción nuevas, A' e I' , que excluyen aquellos componentes independientes que son considerados 'artifacts'. El resultado es una matriz S , que corresponde a la combinación lineal de las señales, como fueron registradas en los electrodos, pero la influencia del ruido en esta combinación lineal efectivamente eliminada.

4.2.2 Parámetros para eliminación de *artifacts*

Para la eliminación de *artifacts*, se procedió a realizar un análisis estadístico de cinco características de cada componente independiente que conforma las pruebas individuales, de acuerdo con el método *Fully Automated Statistical Tresholding for EEG artifact Rejection (FASTER)*, desarrollado e implementado por Nolan, Whelan y Reilly [36]. De este método se seleccionaron cinco parámetros relacionados con características que distinguen a las señales cerebrales de los *artifacts*:

1. El máximo valor absoluto de los coeficientes de correlación de los ICs con los canales EOG.

$$\max (r_{c_t}, x_{EOG1,2,3})$$

2. Curtosis de la información espacial de cada componente independiente.

$$\frac{\mu_4}{\mu_2^2} - 3$$

3. La pendiente media de la densidad espectral de potencia de cada componente independiente.

$$\left| \frac{dF(x_{c_t})}{df} \right| \quad f_{LP1} < f < f_{LP2}$$

4. El exponente de Hurst de cada componente independiente.

$$H_{x_{c_t}}$$

5. La mediana del gradiente de cada componente independiente.

$$\text{median} \left(\frac{d(x_{c_t})}{dt} \right)$$

De esta manera, se obtuvieron cinco parámetros para cada uno de los 22 componentes independientes, los cuales fueron sometidos a un análisis estadístico dado por la obtención de su z-score (ver Ecuación 6). El z-score de cada parámetro fue computado utilizando como población el mismo parámetro para cada uno de los componentes independientes. Por definición, un valor de z-score de 3, indica una

Tabla 4.2.- Función bciRemoveArtifacts. Esta función recibe la señal EEG multidimensional y el correspondiente registro EOG como parámetros, y regresa la señal reconstruida, después de haber rechazado los ICs que, bajo el procedimiento descrito anteriormente, se identifican como ‘artifacts’.

bciRemoveArtifacts		
Parámetros	signal	Registro multidimensional EEG de una prueba individual de imaginación de movimiento desempeñada por un usuario.
	eog	Registro de movimiento ocular correspondiente a la misma prueba de imaginación de movimiento, que se utilizará para corregir <i>artifacts</i> .
Salidas	cleanEEG	Registro multidimensional EEG, reconstruido después de rechazar cualquier IC que se determine como relacionado a <i>artifacts</i> por medio de este algoritmo.

distancia de tres desviaciones estándar respecto de la media, es decir, un valor atípico. Aquellos ICs con algún parámetro cuyo z-score cumpliera con esta característica fueron clasificados como *artifacts* y removidos de la matriz I. Así, se desarrolló la función bciRemoveArtifacts, con la que se ejecutó este proceso, y se describe en la Tabla 4.2.

El algoritmo bciRemoveArtifacts inicia con la señal EEG obtenida como parámetro, y realiza el ciclo de repeticiones de FastICA mencionado en la sección anterior para obtener los 22 ICs necesarios para la obtención de parámetros. Para obtener el primer parámetro de cada IC, el máximo valor absoluto de los coeficientes de correlación del IC con cada uno de los canales EOG, se desarrolló la función bciGetMaxCorrCoeffs, mostrada en la Tabla 4.3, y se implementó dentro de bciRemoveArtifacts para obtener el primer parámetro para el análisis estadístico.

Tabla 4.3.- Función bciGetMaxCorrCoeffs. Esta función toma una matriz de componentes independientes y la matriz EOG asociada a dichos ICs, y devuelve los máximos valores absolutos de los coeficientes de correlación entre los ICs y la información del movimiento ocular.

bciGetMaxCorrCoeffs		
Parámetros	ics	Matriz que contiene los componentes independientes de una prueba individual de imaginación de movimiento, computada usando un algoritmo de ICA.
	eog	Registro de movimiento ocular correspondiente a la misma prueba de imaginación de movimiento, que se utilizará para corregir <i>artifacts</i> .
Salidas	maxcoeffs	El máximo valor absoluto de los coeficientes de correlación entre los ICs y los canales EOG.

La función presentada en la tabla logra relacionar la información del movimiento ocular del sujeto con los componentes independientes, calculando la máxima correlación entre cada IC con alguno de los tres canales EOG. De esta manera, una alta correlación de un IC con algún canal de registro EOG, es un indicador confiable de la presencia de un *artifact* EOG, que al ser relacionado a los

mismos coeficientes de los ICs restantes mediante z-score, puede ser detectado y rechazado de manera automática.

A continuación, se utilizó la función kurtosis del *toolbox Statistics and Machine Learning* integrado a MATLAB para obtener el segundo parámetro, el valor de curtosis. Este algoritmo se describe en la Tabla 4.4, y fue implementado en *bciRemoveArtifacts*, tomando como parámetro la matriz de componentes independientes calculada por el ciclo de FastICA.

Tabla 4.4.- Función kurtosis del toolbox Statistics and Machine Learning de Mathworks, Inc. La función devuelve el valor de curtosis de cada columna. En el caso de este proyecto, la entrada X es la matriz de componentes independientes I introducida en bciRemoveArtifacts.

kurtosis		
Parámetros	X	Matriz cuyas columnas forman variables, a las cuales se les computará el valor de curtosis individualmente.
Salidas	K	El cuarto momento central de X, dividido por la cuarta potencia de su desviación estándar. Si X es una matriz, devuelve el valor de curtosis de cada columna.

Para el tercer parámetro, se desarrolló el algoritmo *bciGetMeanSpectraSlope*, que toma como parámetros los componentes independientes previamente calculados, y computa la media de la pendiente de su densidad espectral de potencia. Esta función se describe en la Tabla 4.5. Para computar la densidad espectral de potencia, *bciGetMeanSpectraSlope* se apoya de la función *pwelch* del *toolbox Signal Processing* de MATLAB.

El cuarto parámetro es el exponente de Hurst, computado utilizando la función genhurst de Aste [37], y que se describe en la Tabla 4.6.

Tabla 4.6.- La función bciGetMeanSpectraSlope. Computa el valor medio de la densidad espectral de potencia de cada componente independiente. Para el cálculo de la densidad espectral de potencia, este algoritmo se apoya en la función pwelch del toolbox Signal Processing de MATLAB.

bciGetMeanSpectraSlope		
Parámetros	ics	Matriz que contiene los componentes independientes de una prueba individual de imaginación de movimiento, computada usando un algoritmo de ICA.
Salidas	meanSpectra	El valor medio del espectro de potencia de cada componente independiente.

Tabla 4.5.- Función genhurst de Aste [37]. Computa el exponente generalizado de Hurst de una serie de datos. Esta función fue llamada para cada componente independiente.

genhurst		
Parámetros	S	Serie de datos
Salidas	H	El exponente generalizado de Hurst de la serie de datos introducida como parámetro.

La función genhurst fue computada a través de todos los componentes independientes computada al inicio de bciRemoveArtifacts.

El quinto parámetro, la mediana del gradiente de cada componente IC, se computó mediante una derivada discreta centrada en cada IC, donde los puntos de

datos representan cada uno de los puntos de una función discreta dada por los valores de tiempo y el valor de amplitud de voltaje medido en dicho punto.

4.2.3 Rechazo de ICs mediante detección de datos atípicos por z-score

Como se mencionó en la sección 4.2.2, una vez computados los cinco parámetros de cada componente independiente, se procedió al rechazo de componentes independientes mediante un análisis estadístico basado en z-score, de acuerdo con el método FASTER de remoción automática de *artifacts* [36]. La Figura 4.5 muestra el último paso del algoritmo implementado en bciRemoveArtifacts para el rechazo de los componentes independientes que regresaran un z-score superior a 3. La siguiente sección describe la función utilizada para calcular el z-score en los parámetros de los componentes independientes, así como en el bloque de normalización del preprocesamiento. Este proceso fue

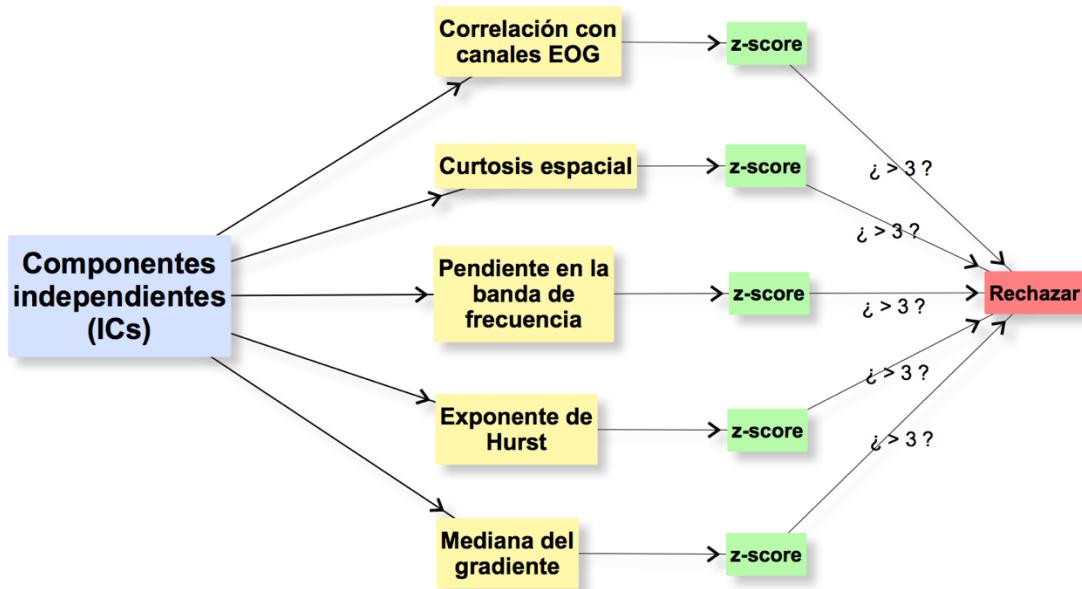


Figura 4.5.- Para el rechazo automático de ICs, el método FASTER mide el número de desviaciones estándar respecto de la media de un parámetro en todos los componentes independientes. Una vez que se detecta un dato atípico por la obtención de un z-score de valor mayor a 3, el IC es eliminado de la matriz I .

repetido a través de todas las pruebas de ambas clases en cada sujeto, en ambas sesiones.

4.3 Preprocesamiento: Normalización mediante z-score

Para introducir las señales a un proceso de normalización, se precisó utilizar la misma función implementada previamente en `bciRemoveArtifacts` para la detección automática de *artifacts*. La función `zscore` pertenece al toolbox Statistics and Machine Learning de MATLAB, y se describe en la Tabla 4.7. Esta función acepta como parámetros vectores, matrices o arreglos multidimensionales, y computa el z-score de cada elemento. Como lo indica la teoría de la sección 2.3.2, los valores de z-score conforman una estandarización de los datos introducidos, ya que la media de los z-scores es cero, y la desviación estándar de los z-scores es 1.

Tabla 4.7.- Función `zscore` del toolbox Statistics and Machine Learning integrada a MATLAB. Recibe como parámetros un arreglo de datos, una bandera que, si el valor es cero, devuelve la media y desviación estándar de la muestra ($N-1$), y la media y desviación estándar de la población (N) si se introduce un valor de 1 y la dimensión a través de la cual computará los z-scores.

zscore		
Parámetros	X	Vector, matriz, o arreglo multidimensional de datos.
	FLAG	Bandera para indicar si la media y desviación estándar se calculan utilizando N, o $N-1$, siendo N el número de datos totales a través de los que se obtienen estas mediciones estadísticas.
	DIM	Dimensión del arreglo a través del cual se estandariza X.
Salidas	Z	Una versión centrada y escalada de X, de las mismas dimensiones que X.

Para el z-score, se realizó la estandarización de cada valor temporal, en el mismo canal y en el mismo instante de tiempo, a través de todas las pruebas desempeñadas por un sujeto. Para esto, se tomó ventaja del formato con el que se

registraron las pruebas, es decir, el formato de ‘páginas de libro’ descrito anteriormente, para indicar como parámetro DIM la dimensión 3. Además, se introdujo como parámetro FLAG el valor 1, para indicar que se desea una media y desviación estándar de la población. Así, la introducción de los datos se realiza con el formato con el que se generaron las variables, y mediante una simple indicación de la dimensión a través de la cual se debe ejecutar el algoritmo, se genera la normalización de 72 pruebas correspondientes a una intención de movimiento por un sujeto. Este proceso fue repetido a través de las dos variables por sesión de cada sujeto.

4.4 Preprocesamiento: Selección de canales mediante CSP

Para implementar la selección de canales, las pruebas relacionadas a ambas clases se sometieron a un proceso de filtrado de señales, implementando la función bandPassFilter desarrollada por Marrufo y Giorgana [2], la cual se muestra en la Tabla 4.8. Como parámetros, siempre se eligieron el tipo de filtro pasabanda Butterworth, con frecuencias de corte de 8 y 30 Hz.

Una vez que se obtuvieron las señales filtradas, se procedió a seleccionar los canales relevantes utilizando el método de CSP descrito en la Sección 2.4.1. En primer lugar, se diseñó un algoritmo que tuviera en cuenta el número de canales que el usuario indique que desea obtener de las pruebas. De esta manera, para las matrices de pruebas con el formato especificado en la Figura 4.1, se modificará el tamaño de columnas al número de canales indicados por el usuario. La implementación de estos algoritmos para recuperar las pruebas con número de canales modificados de acuerdo con los canales relevantes indicados por su norma 1, como se describe en la Sección 2.4.1, se implementó en la función bciSelectChannels, descrita en la Tabla 4.9.

Tabla 4.9.- Función `bandPassFilter` de [2]. Realiza un filtrado de tipo pasabanda, de acuerdo con las especificaciones introducidas por el usuario. Para este proyecto, siempre se utilizó un filtro Butterworth 8-30 Hz.

bandPassFilter		
Parámetros	nSubjects	Número de sujetos.
	nChannels	Número de canales EEG.
	windows	Las ventanas para ser filtradas.
	nWindows	Número de ventanas.
	typeOfFilter	El tipo de filtro a utilizar. Puede ser 'windowed_sync' o 'butterworth'.
	lcf	Frecuencia de corte inferior a utilizar.
	hcf	Frecuencia de corte superior a utilizar.
	Fs	Frecuencia de muestreo de la señal.
Salidas	filtered_signals	Celda con las ventanas filtradas.

Tabla 4.8.- `bciSelectChannels`. Toma las pruebas de dos clases de imaginación de movimiento y un número de canales relevantes a seleccionar, y encuentra los canales que permiten discriminar mejor entre las dos clases utilizando el método de Common Spatial Patterns (CSP).

bciSelectChannels		
Parámetros	signalsClass1	Señales de la primera clase.
	signalsClass2	Señales de la segunda clase.
	nChannelsToSel	Número de canales a seleccionar.
Salidas	newSignalsClass1	Pruebas de clase 1 con dimensión de canales reducida.
	newSignalsClass2	Pruebas de clase 2 con dimensión de canales reducida.
	sortedSelectedChannelNumbers	Vector con el índice de los mejores canales.

4.5 Generación de nuevas bases de datos combinando bloques de preprocessamiento

Con base en las funciones descritas anteriormente, y con el objetivo de probar la efectividad de cada bloque de preprocessamiento para la mejora de la clasificación de intención de movimiento, se generaron nuevas bases de datos a partir de la utilización de bloques de preprocessamiento, de tal manera que cada bloque individual, generaran una nueva base de datos a la cual introducir a las etapas subsecuentes del sistema BCI. El desarrollo de esta generación se basó en el desarrollo de una función en la que se pudieran introducir las variables necesarias para realizar el preprocessamiento, así como la instrucción de desempeñar uno o más bloques específicos de esta etapa. Esta función fue denominada `bciPreprocessingStage`, y se describe en la Tabla 4.10. Al inició, se compuso la función para incluir los tres bloques de preprocessamiento, pero dado que el filtrado

Tabla 4.10.- Función `bciPreprocessingStage`. Esta función toma como parámetros las variables necesarias para desempeñar los diferentes bloques de preprocessamiento, así como valores lógicos para indicar qué combinación de los bloques utilizar para la agrupación de pruebas de la variable ‘signals’.

bciPreprocessingStage		
Parámetros	signals	Un arreglo de 750-por-22-por-72 datos, donde cada matriz a lo largo de la tercera dimensión compone las pruebas de imaginación de movimiento.
	eog	Registros de movimiento ocular de las pruebas de imaginación de movimiento.
	doArtifCorr	Valor lógico para indicar si se desea implementar el bloque de corrección de <i>artifacts</i> en la señal multidimensional.
	doNorm	Valor lógico para indicar si se desea implementar el bloque de corrección de <i>artifacts</i> en la señal multidimensional.
Salidas	preprocessedEEG	El arreglo de ‘signals’, después de haber pasado por los bloques de preprocessamiento seleccionados.

sería implementado en las etapas de entrenamiento y clasificación, solamente se generaron bases de datos para corrección de *artifacts* y normalización.

De esta manera, para facilitar la introducción de las señales sometidas a diferentes bloques de preprocesamiento, se crearon nuevas bases de datos para cada sujeto, a partir de la salida preprocessedEEG. El procedimiento de esta etapa del proyecto se ilustra en la Figura 4.6.

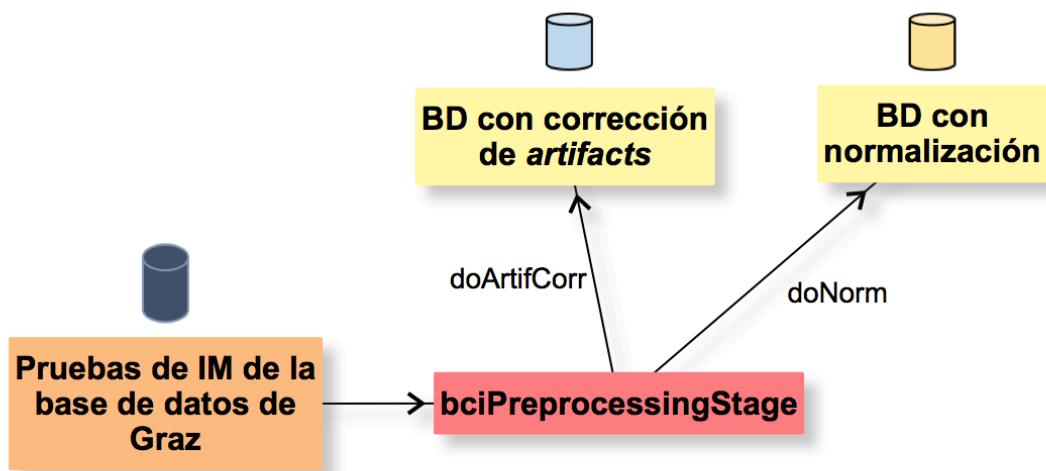


Figura 4.6.- Procedimiento de la generación de las bases de datos necesarias para la extracción de características, entrenamiento de la máquina de aprendizaje, y clasificación de intención de movimiento, utilizando la función bciPreprocessingStage.

4.6 Extracción de características y clasificación

4.6.1 AdaBoost

Para las etapas de extracción de características y clasificación, se desarrollaron funciones basadas en los algoritmos desarrollados por Marrufo y Giorgana para el proyecto “Clasificación de intención de movimiento para un sistema BCI no invasivo” [2]. De acuerdo con los algoritmos implementados en dicho proyecto, se utilizó la DWT para obtener los coeficientes de detalle y aproximación, utilizando wavelets de tipo Coiflet, Symlet y Daubechies y el nivel DWT 7. Se

obtuvieron características de todos los canales EEG mediante sus coeficientes DWT, y la longitud de ventana fue de 3 segundos, es decir, la longitud total de la imaginación de movimiento de cada usuario. Las tablas 4.11 y 4.12 describen las funciones trainAdaBoost y testAdaBoost, utilizadas para el entrenamiento y pruebas del algoritmo de clasificación AdaBoost.

Para el entrenamiento de la máquina de clasificación AdaBoost, todas las pruebas de ambas intenciones de movimiento fueron utilizadas (72 pruebas por clase). Se generaron los modelos mediante trainAdaBoost, utilizando los siguientes intervalos de clasificadores débiles para el entrenamiento:

- De 2 a 30 clasificadores débiles, en intervalos de 2.
- De 35 a 90 clasificadores débiles, en intervalos de 5.
- De 100 a 400 clasificadores débiles, en intervalos de 100.

Tabla 4.11.- Función bciTrain_DWT_AdaBoost. Esta función construye modelos a partir de las dos variables correspondientes a una sesión de pruebas de imaginación de movimiento, donde cada variable contiene las pruebas de una sola intención de movimiento.

trainAdaBoost		
Parámetros	subject	Número de sujeto al cual corresponden las pruebas introducidas.
	processStage	Secuencia de caracteres que se utilizará para ubicar el directorio correspondiente a la etapa de preprocesamiento a la cual fueron sometidas las señales.
	signalsClass1	72 pruebas correspondientes a la clase 1 (mano derecha).
	signalsClass2	72 pruebas correspondientes a la clase 2 (ambos pies).
	dwt_type	Tipo de wavelet para extracción de características.
	numChansToSel	Número de canales a seleccionar si se decide seleccionar canales.

Tabla 4.12.- Función `testAdaBoost`. Esta función constituye la etapa de prueba que utiliza los modelos previamente construidos mediante `trainAdaBoost`, y, tomando un nuevo set de señales, correspondientes a las mismas clases, busca clasificarlas correctamente.

testAdaBoost		
Parámetros	subject	Número de sujeto al cual corresponden las pruebas introducidas.
	processStage	Secuencia de caracteres que se utilizará para ubicar el directorio correspondiente a la etapa de preprocesamiento a la cual fueron sometidas las señales.
	signalsClass1	72 pruebas correspondientes a la clase 1 (mano derecha).
	signalsClass2	72 pruebas correspondientes a la clase 2 (ambos pies).
	dwt_type	Tipo de wavelet para extracción de características.
	numChansToSel	Número de canales a seleccionar si se decide seleccionar canales.

Estos modelos fueron almacenados en directorios correspondientes al sujeto de prueba, así como a las etapas de procesamiento a las cuales se sometieron las variables que contienen sus registros EEG. Los mismos fueron utilizados por `testAdaboost` para ejecutar la clasificación de todas las pruebas correspondientes a la segunda sesión de registro de señales, cuyas señales no fueron utilizadas para el entrenamiento del algoritmo AdaBoost. Para evaluar la exactitud del clasificador en su discriminación de intención de movimiento, se introdujo el vector de etiquetas que corresponde a las clases correctas de cada prueba al algoritmo, y se compara con el vector de etiquetas generadas por éste, generando porcentajes de acierto como en la Ecuación 24 (ver Sección 3.3). La función `bciTest_DWT_AdaBoost` genera archivos denominados ‘accuracies’, en los que se registran estos porcentajes para cada número de clasificadores débiles.

4.6.2 SVM

La segunda máquina de clasificación que fue utilizada para entrenamiento y clasificación de las pruebas fue la máquina de vectores de soporte, y para esto se utilizó la función fitcsvm del toolbox Statistics and Machine Learning de MATLAB, descrita en la Tabla 4.13. La función recibe como principales parámetros la matriz de características que se produce como en la Sección 3.2, así como el vector de etiquetas que indican la clase a la que pertenece cada vector fila de la matriz de características. Adicionalmente, se introducen ciertos parámetros para optimizar el entrenamiento. Para ejecutar la optimización del modelo mediante validación cruzada, se introdujo un vector de parámetros con los valores de BoxConstrait y KernelScale en un rango entre 1×10^{-5} y 1×10^5 , y un KFold de valor 5.

Tabla 4.13.- Función fitcsvm del toolbox Statistics and Machine Learning de MATLAB. Entrena un modelo de clasificación a partir de una matriz de características. Se puede optar por un modelo construido a partir de validación cruzada si se introduce un vector de parámetros optimizables.

fitcsvm		
Parámetros	X	Matriz de valores numéricos, con cada fila correspondiendo a una observación y cada columna a una característica.
	Y	Vector de etiquetas.
	'KernelFunction', function name	Tipo de función para computar el modelo, que puede ser 'linear', 'gaussian', 'rbf', o 'polynomial'.
	'OptimizeHyperparameters', OptimizableVariable vector	Vector con hiperparámetros a optimizar.
Salidas	Mdl	Modelo de entrenamiento para clasificación.

Para el entrenamiento de las pruebas, que incluye el filtrado, la selección de canales relevantes si se elige el bloque de selección de canales como preprocessamiento, la extracción de características y la introducción de las

características y etiquetas en fitcsvm para generar el modelo de entrenamiento, se desarrolló la función trainSVM, descrita en la Tabla 4.14.

Tabla 4.14.- Función trainSVM. Filtra, selecciona canales relevantes, extrae características y entrena el modelo de clasificación de la matriz de características creada utilizando fitcsvm.

trainSVM		
Parámetros	subject	Número de sujeto al cual corresponden las pruebas introducidas.
	processStage	Secuencia de caracteres que se utilizará para ubicar el directorio correspondiente a la etapa de preprocessamiento a la cual fueron sometidas las señales.
	signalsClass1	72 pruebas correspondientes a la clase 1 (mano derecha).
	signalsClass2	72 pruebas correspondientes a la clase 2 (ambos pies).
	dwt_type	Tipo de wavelet para extracción de características.
	numChansToSel	Número de canales a seleccionar si se decide seleccionar canales.

La generación de hipótesis de la máquina SVM utilizando los modelos generados se realiza mediante la función predict del toolbox Statistics and Machine Learning de MATLAB, descrito en la Tabla 4.15.

Tabla 4.15.- Función predict del toolbox Statistics and Machine Learning de MATLAB. Devuelve un vector de etiquetas predichas a partir del modelo de clasificación SVM y matriz de características introducido.

predict		
Parámetros	SVMModel	Modelo de clasificación SVM.
	X	Matriz de características a clasificar.
Salidas	label	Vector de etiquetas predicho basado en el modelo de clasificación introducido.

Para realizar la preparación de las matrices de características como en las fases de entrenamiento, y ejecutar predict en la matriz de características con las señales de evaluación de las bases de datos, se desarrolló la función testSVM, descrita en la Tabla 4.16.

Tabla 4.16.- Función testSVM. Esta función constituye la etapa de prueba que utiliza los modelos previamente construidos mediante trainSVM, y, tomando un nuevo set de señales, correspondientes a las mismas clases, busca clasificarlas correctamente.

testSVM		
Parámetros	subject	Número de sujeto al cual corresponden las pruebas introducidas.
	processStage	Secuencia de caracteres que se utilizará para ubicar el directorio correspondiente a la etapa de preprocesamiento a la cual fueron sometidas las señales.
	signalsClass1	72 pruebas correspondientes a la clase 1 (mano derecha).
	signalsClass2	72 pruebas correspondientes a la clase 2 (ambos pies).
	dwt_type	Tipo de wavelet para extracción de características.
	numChannsToSel	Número de canales a seleccionar si se decide seleccionar canales.

4.6.3 AdaSVM

Se utilizó un tercer clasificador: un híbrido entre AdaBoost y SVM, basado en los algoritmos desarrollados por Marrufo y Giorgana [2], conocido como AdaSVM. Se utilizaron las funciones auxiliares desarrolladas por [2] para seleccionar características relevantes utilizando AdaBoost, y entrenar la matriz de características reducidas utilizando la máquina SVM cuya función fitcsvm se describe en la Tabla 4.13. Para la implementación de estos algoritmos, se presentan las funciones trainAdaSVM y testAdaSVM en las Tablas 4.17 y 4.18.

Tabla 4.17.- Funciones trainAdaSVM y testAdaSVM. La función trainAdaSVM filtra, selecciona canales relevantes, extrae características, selecciona características mediante AdaBoost y entrena el modelo de clasificación de la matriz de características creada utilizando fitcsvm. Previo al entrenamiento, utiliza un algoritmo basado en AdaBoost para seleccionar las mejores características para introducir al clasificador. La función testAdaSVM para por el mismo proceso, pero al final realiza las hipótesis en base al modelo generado durante el entrenamiento.

trainAdaSVM/ testAdaSVM		
Parámetros	subject	Número de sujeto al cual corresponden las pruebas introducidas.
	processStage	Secuencia de caracteres que se utilizará para ubicar el directorio correspondiente a la etapa de preprocesamiento a la cual fueron sometidas las señales.
	signalsClass1	72 pruebas correspondientes a la clase 1 (mano derecha).
	signalsClass2	72 pruebas correspondientes a la clase 2 (ambos pies).
	dwt_type	Tipo de wavelet para extracción de características.
	numChansToSel	Número de canales a seleccionar si se decide seleccionar canales.

Capítulo 5 – Resultados

Los resultados del desarrollo del proyecto se presentan en este capítulo, comprendiendo los porcentajes de acierto de la clasificación de intención de movimiento generados descritos en la sección 4.5. Para la exposición del desempeño del clasificador AdaBoost para cada base de datos generada usando diferentes bloques de preprocesamiento (ver Figura 4.6), se exponen tablas con los mejores resultados para cada sujeto.

5.1 Resultados del clasificador AdaBoost para la base de datos de Graz original

La Tabla 5.1 muestra los resultados obtenidos al ejecutar el algoritmo de clasificación AdaBoost utilizando los modelos generados a partir de la extracción de características de las pruebas de la base de datos de Graz original. Los resultados

Tabla 5.1.- Mejores resultados del clasificador AdaBoost con la base de datos de Graz sin ser sometida a bloques de preprocesamiento.

Sujeto	Filtro	Número de clasificadores débiles	Tipo de extractor de características	Porcentaje de acierto
1	Butterworth	200	FFT-DWT	87.5%
2	Butterworth	70	FFT-DWT	63.89%
3	Butterworth	26	FFT-DWT	84.72%
5	Butterworth	60	FFT-DWT	63.89%
6	Butterworth	16	FFT-DWT	65.28%
7	Butterworth	10	FFT-DWT	78.47%
8	Butterworth	6	FFT-DWT	66.67%
9	Butterworth	12	FFT-DWT	60.42%

que se muestran son aquellos cuyos parámetros concuerdan con los de los mejores resultados de Marrufo y Giorgana [2], con el fin de compararlos con los resultados obtenidos al clasificar las bases de datos generadas durante el desarrollo de este proyecto.

5.2 Resultados del clasificador AdaBoost para las señales EEG sometidas a bloques de preprocesamiento

Los resultados obtenidos de la ejecución del clasificador AdaBoost en las pruebas sometidas a bloques de preprocesamiento se muestran en esta sección. Cada tabla de resultados muestra los mejores porcentajes de desempeño del clasificador AdaBoost de acuerdo con el número de clasificadores débiles utilizados por dicho algoritmo. Los resultados por cada bloque de procesamiento muestran la efectividad de dicho bloque para mejorar la calidad de las señales, y por consecuencia la mejora del sistema BCI al realizar una comparación con los resultados mostrados en la Tabla 5.1 con aquellos pertenecientes a los procesos desarrollados en este proyecto. Los resultados de la Tabla 5.1 serían efectivamente similares a los resultados obtenidos por Marrufo y Giorgana en [2], dado que se ejecutaron algoritmos basados en sus mejores resultados para obtener nuevamente el desempeño del clasificador para la base de datos sin corrección de *artifacts*, normalización o selección de canales. Sin embargo, es importante resaltar que Marrufo y Giorgana rechazaron aquellas pruebas que fueron etiquetadas por los autores de la base de datos como contaminadas por *artifacts* [2][13].

5.2.1 Corrección de *artifacts*

La Tabla 5.2 muestra los resultados de la base de datos con el bloque de corrección de *artifacts*. Los resultados iniciales son aquellos en los que la extracción de características fue mediante el algoritmo FFT-DWT, utilizando 5 tipos de wavelet madre (Coiflet5, Daubechies6, Symlet4, Symlet5, Symlet6), filtrado Butterworth y un tamaño de ventana de 3 segundos.

Tabla 5.2.- Resultados de la clasificación mediante AdaBoost de la base de datos sometida al bloque de corrección de 'artifacts'.

Sujeto	Número de clasificadores débiles	Tipo de extractor de características	Porcentaje de acierto
1	300	FFT-DWT	90.28%
2	300	FFT-DWT	61.81%
3	2	FFT-DWT	81.94%
5	22	FFT-DWT	59.03%
6	6	FFT-DWT	66.67%
7	2	FFT-DWT	75%
8	6	FFT-DWT	66.67%
9	24	FFT-DWT	61.11%

5.2.2 Normalización

La Tabla 5.3 muestra los resultados de la base de datos con el bloque de normalización. Los resultados iniciales son aquellos en los que la extracción de características fue mediante el algoritmo DWT, utilizando una wavelet madre Daubechies4, sin filtro y un tamaño de ventana de 3 segundos.

Tabla 5.3.- Resultados de la clasificación mediante AdaBoost de la base de datos sometida al bloque de normalización.

Sujeto	Número de clasificadores débiles	Tipo de extractor de características	Porcentaje de acierto
1	40	FFT-DWT	80.56%
2	6	FFT-DWT	60.42%
3	12	FFT-DWT	79.17%
5	50	FFT-DWT	61.11%
6	18	FFT-DWT	62.5%
7	400	FFT-DWT	75%
8	40	FFT-DWT	66.67%
9	90	FFT-DWT	69.44%

5.2.3 Selección de canales

Tabla 5.4.- Resultados de la clasificación mediante AdaBoost de la base de datos sometida al bloque de selección de canales.

Sujeto	Número de clasificadores débiles	Tipo de extractor de características	Porcentaje de acierto
1	24	FFT-DWT	86.82%
2	16	FFT-DWT	54.86%
3	28, 30	FFT-DWT	84.03%
5	14	FFT-DWT	65.97%
6	12	FFT-DWT	64.58%
7	55	FFT-DWT	80.56%
8	12	FFT-DWT	68.06%
9	100	FFT-DWT	65.28%

La Tabla 5.4 muestra los resultados de la base de datos con el bloque de selección de canales. Los resultados iniciales son aquellos en los que la extracción de características fue mediante el algoritmo DWT, utilizando una wavelet madre Daubechies4, sin filtro y un tamaño de ventana de 3 segundos.

5.3 Resultados del clasificador SVM para la base de datos Graz original

Los resultados presentados en esta sección corresponden a aquellos generados por la máquina de clasificación SVM, cuya implementación se describe en la sección 4.6.2. De la misma manera que con la máquina AdaBoost, se obtienen los resultados de la máquina SVM con la base de datos original, con el fin de comparar dichos resultados con aquellos obtenidos de las pruebas sometidas a cada bloque de preprocesamiento. Los resultados de SVM para la base de datos original se presenta en la Tabla 5.5.

Tabla 5.5.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz original, utilizando el clasificador SVM.

Sujeto	Tipo de extractor de características	Porcentaje de acierto
1	FFT-DWT	88.19%
2	FFT-DWT	61.11%
3	FFT-DWT	82.64%
5	FFT-DWT	58.33%
6	FFT-DWT	62.5%
7	FFT-DWT	77.78%
8	FFT-DWT	72.22%
9	FFT-DWT	68.05%

5.4 Resultados del clasificador SVM para las señales EEG sometidas a bloques de preprocesamiento

5.4.1 Corrección de *artifacts*

La Tabla 5.6 muestra los resultados de la máquina de clasificación SVM para la base de datos con señales sometidas al bloque de corrección de *artifacts* mediante FastICA. Los parámetros de optimización son los mismos que para la base de datos original.

Tabla 5.6.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometida a corrección de artifacts, utilizando el clasificador SVM.

Sujeto	Tipo de extractor de características	Porcentaje de acierto
1	FFT-DWT	88.89%
2	FFT-DWT	68.75%
3	FFT-DWT	80.55%
5	FFT-DWT	55.56%
6	FFT-DWT	65.28%
7	FFT-DWT	77.78%
8	FFT-DWT	72.92%
9	FFT-DWT	73.61%

5.4.2 Normalización

La Tabla 5.7 muestra los resultados de la máquina de clasificación SVM para la base de datos con señales sometidas al bloque de normalización vía z-score. Los parámetros de optimización utilizados para la función fitcsvm fueron los mismos que para las señales originales y las señales sometidas a corrección de *artifacts*.

Tabla 5.7.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometida a normalización, utilizando el clasificador SVM.

Sujeto	Tipo de extractor de características	Porcentaje de acierto
1	FFT-DWT	78.47%
2	FFT-DWT	52.08%
3	FFT-DWT	75.69%
5	FFT-DWT	55.56%
6	FFT-DWT	55.56%
7	FFT-DWT	52.08%
8	FFT-DWT	59.72%
9	FFT-DWT	65.97%

5.4.3 Selección de canales

La Tabla 5.8 muestra los resultados de las pruebas de clasificación de las señales EEG con canales seleccionados por el método de CSP, utilizando el clasificador SVM para entrenamiento y evaluación.

Tabla 5.8.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometida a selección de canales, utilizando el clasificador SVM.

Sujeto	Tipo de extractor de características	Porcentaje de acierto
1	FFT-DWT	78.47%
2	FFT-DWT	52.08%
3	FFT-DWT	75.69%
5	FFT-DWT	55.56%
6	FFT-DWT	55.56%
7	FFT-DWT	52.08%
8	FFT-DWT	59.72%
9	FFT-DWT	65.97%

5.5 Resultados del clasificador AdaSVM para la base de datos Graz original

Los resultados que se muestran en esta sección muestran los resultados de la implementación del clasificador AdaSVM para entrenar y clasificar intención de movimiento a partir de las pruebas de la base de datos de Graz original. Estos resultados, mostrados en la Tabla 5.9, se compararán con aquellos obtenidos de las señales cuando son sometidas a cada bloque de preprocesamiento.

Tabla 5.9.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz original, utilizando el clasificador AdaSVM.

Sujeto	Número de clasificadores débiles	Tipo de extractor de características	Porcentaje de acierto
1	90, 100, 200, 300	FFT-DWT	91.67%
2	30	FFT-DWT	68.06%
3	40	FFT-DWT	83.33%
5	30, 40, 60	FFT-DWT	55.56%
6	50	FFT-DWT	66.67%
7	60	FFT-DWT	77.08%
8	30	FFT-DWT	67.36%
9	100, 200, 300, 400	FFT-DWT	68.75%

5.6 Resultados del clasificador AdaSVM para las señales EEG sometidas a bloques de preprocesamiento

5.6.1 Corrección de artifacts

La Tabla 5.10 muestra los resultados del clasificador AdaSVM para determinar intención de movimiento en señales EEG sometidas al bloque de

corrección de *artifacts*, utilizando el método de descomposición por ICA y rechazo por análisis mediante *FASTER*.

Tabla 5.10.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometidas a corrección de ‘*artifacts*’, utilizando el clasificador AdaSVM.

Sujeto	Número de clasificadores débiles	Tipo de extractor de características	Porcentaje de acierto
1	40, 55	FFT-DWT	88.19%
2	85	FFT-DWT	70.83%
3	2	FFT-DWT	82.64%
5	18	FFT-DWT	59.72%
6	4,6	FFT-DWT	67.36%
7	4	FFT-DWT	85.42%
8	75	FFT-DWT	70.14%
9	60, 65, 85, 200	FFT-DWT	69.44%

5.6.2 Normalización

La Tabla 5.11 muestra los resultados del clasificador AdaSVM para determinar intención de movimiento en señales EEG sometidas al bloque de corrección de *artifacts*, utilizando el método de normalización por z-score.

Tabla 5.11.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometidas a normalización, utilizando el clasificador AdaSVM.

Sujeto	Número de clasificadores débiles	Tipo de extractor de características	Porcentaje de acierto
1	40, 55	FFT-DWT	88.19%
2	85	FFT-DWT	70.83%
3	2	FFT-DWT	82.64%
5	18	FFT-DWT	59.72%
6	4,6	FFT-DWT	67.36%
7	4	FFT-DWT	85.42%
8	75	FFT-DWT	70.14%
9	60, 65, 85, 200	FFT-DWT	69.44%

5.6.3 Selección de canales

La Tabla 5.11 muestra los resultados del clasificador AdaSVM para determinar intención de movimiento en señales EEG sometidas al bloque de selección de canales, utilizando el método de CSP.

Tabla 5.12.- Mejores resultados de la clasificación de intención de movimiento para las pruebas de la base de datos de Graz sometidas a selección de canales, utilizando el clasificador AdaSVM.

Sujeto	Número de clasificadores débiles	Tipo de extractor de características	Porcentaje de acierto
1	40, 55	FFT-DWT	88.19%
2	85	FFT-DWT	70.83%
3	2	FFT-DWT	82.64%
5	18	FFT-DWT	59.72%
6	4,6	FFT-DWT	67.36%
7	4	FFT-DWT	85.42%
8	75	FFT-DWT	70.14%
9	60, 65, 85, 200	FFT-DWT	69.44%

5.7 Mejores resultados de acuerdo con el clasificador

Los diagramas mostrados a continuación muestran los mejores resultados obtenidos para cada sujeto, tomando aquellos en los que hubo la mayor mejora de desempeño de la máquina dado cierto bloque de preprocesamiento.

5.7.1 Corrección de *artifacts*

La Figura 5.1 muestra los mejores resultados para mejorar el desempeño del sistema BCI independientemente del clasificador utilizado. El clasificador que presentó mejora en el desempeño se encuentra etiquetado para cada sujeto.

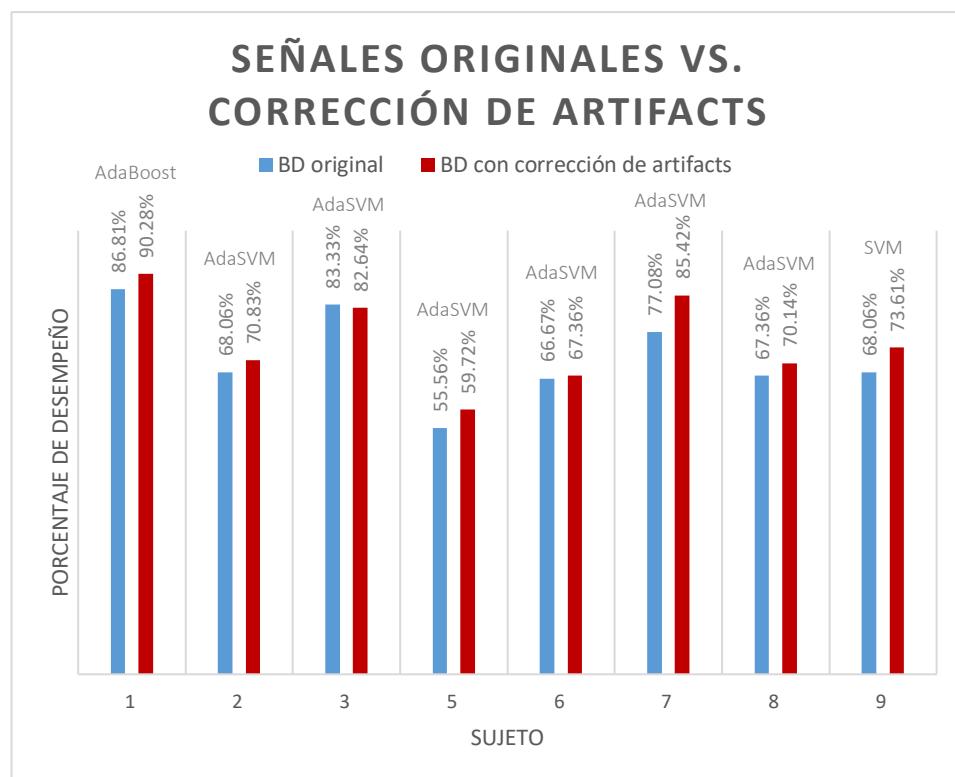


Figura 5.1.- Mejores resultados para la mejora del desempeño del sistema BCI mediante corrección de 'artifacts'.

5.7.2 Normalización

Los mejores resultados para cada sujeto con las pruebas sometidas al bloque de normalización se muestran en la Figura 5.2.

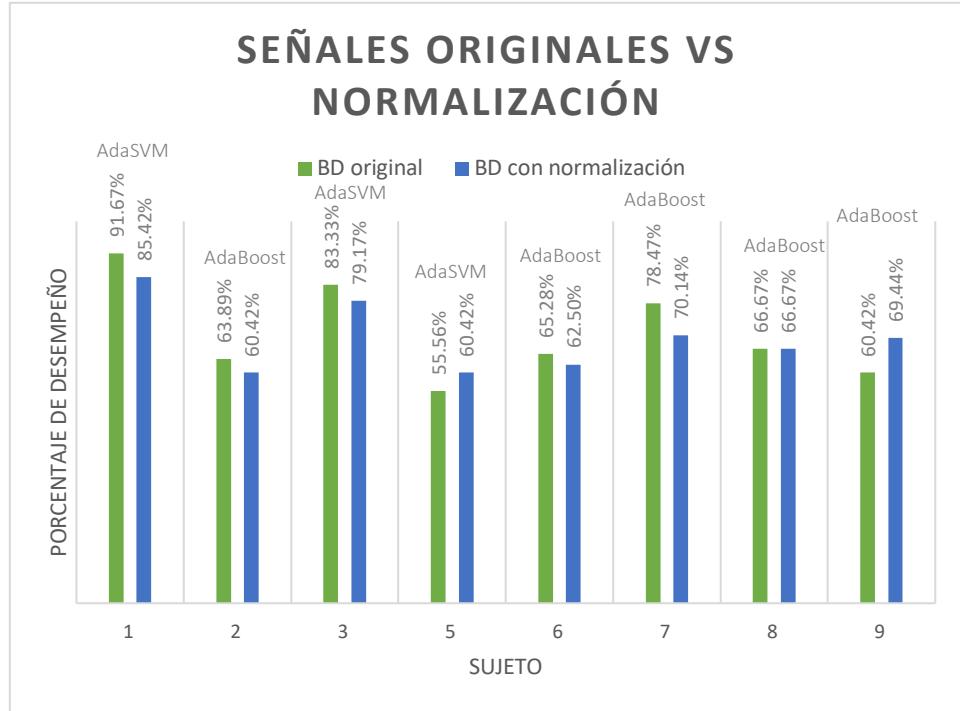


Figura 5.2.- Mejores resultados en mejora de desempeño para cada sujeto, comparando la base de datos original versus la base de datos normalizada.

5.7.3 Selección de canales

La Figura 5.3 muestra los mejores resultados en cuanto a mejora de desempeño del sistema comparando la base de datos de Graz original versus las señales con los mejores canales seleccionados.

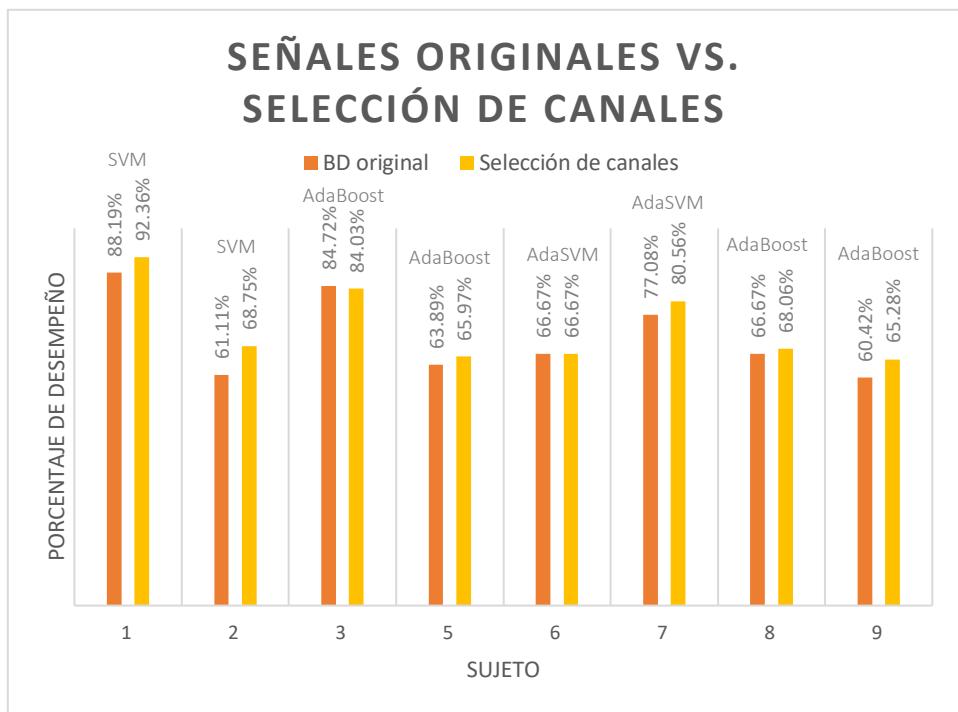


Figura 5.3.- Mejores resultados en la mejora de desempeño del sistema BCI con los mejores canales seleccionados.

5.8 Canales relevantes para la clasificación

De acuerdo con la descripción del desarrollo del método CSP explicado en la sección 4.4, se presenta la combinación de mejores puntuajes para cada canal, combinando puntuaciones obtenidas por cada canal, en cada sujeto, dando una puntuación de 10 al canal más relevante de los diez canales seleccionados, hasta una puntuación de 1 al menos relevante de ellos. Estos resultados se muestran en las Figuras 5.4 y 5.5.

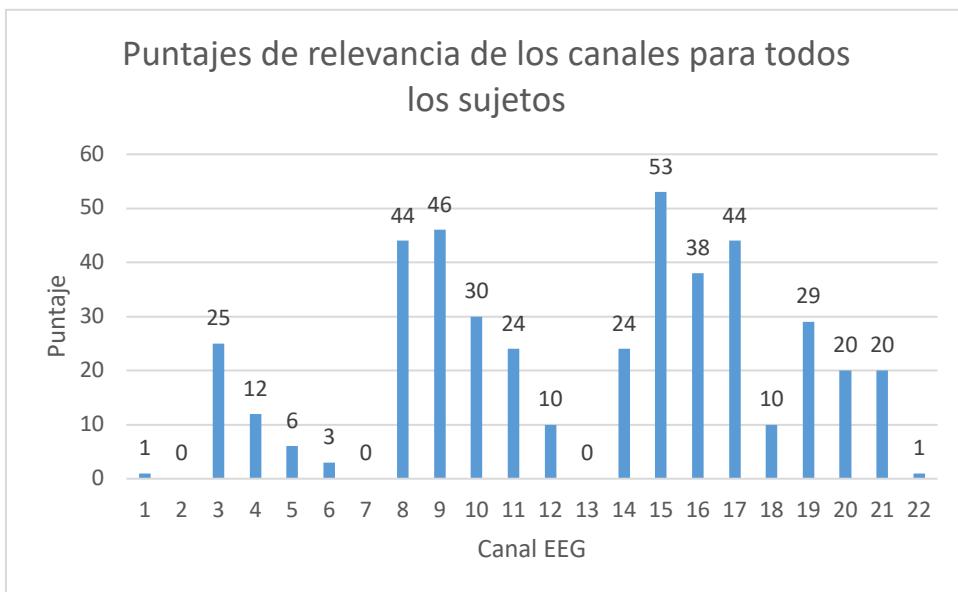


Figura 5.4.- Puntuación de relevancia de cada canal a través de todos los sujetos de prueba.

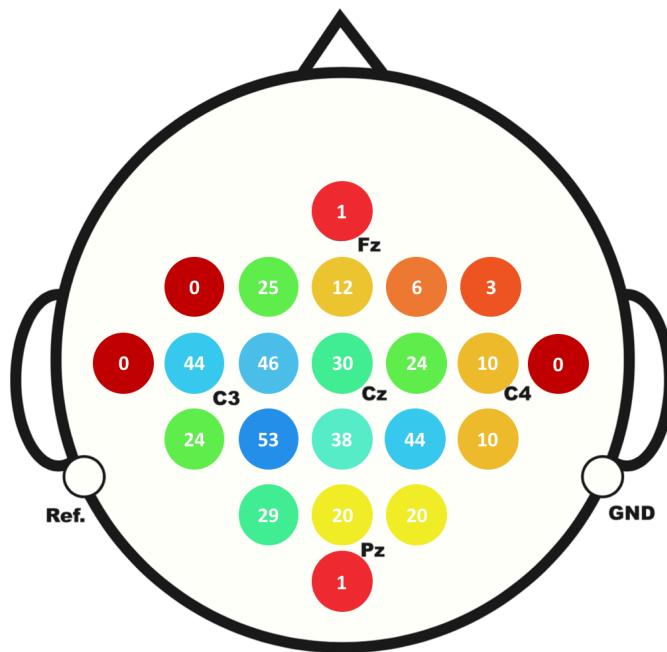


Figura 5.5.- Mapa de calor que muestra los resultados de puntajes de relevancia de cada canal, a través de todos los sujetos.

5.9 Tiempo requerido para generación de modelos de la máquina SVM contra la máquina AdaSVM.

La Figura 5.6 muestra la comparación de tiempos para la generación de modelos de entrenamiento de dos clasificadores: SVM y AdaSVM. El eje horizontal muestra el tiempo, en segundos, para generar los modelos que generaron los mejores resultados para clasificar las señales de la base de datos de Graz original. Los modelos de AdaSVM fueron generados tomando hasta 400 características relevantes, es decir, el caso en el que se selecciona el mayor número de características para la generación de modelos de clasificación.

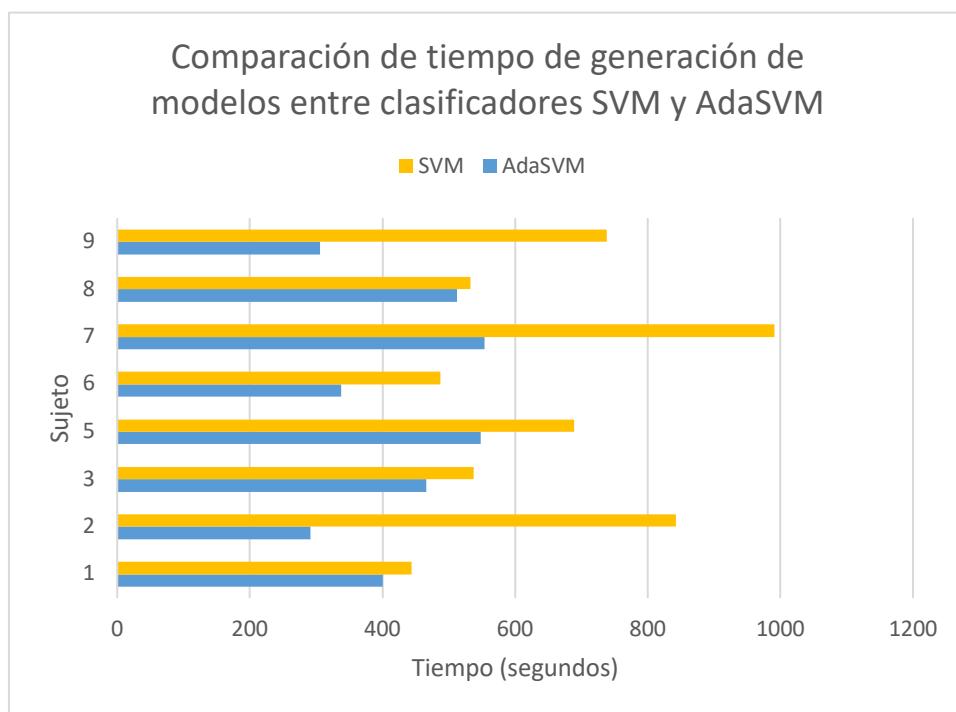


Figura 5.6.- Comparación de tiempo requerido para la generación de modelos entre SVM y AdaSVM.i

Capítulo 6 – Conclusiones

6.1 Conclusiones Técnicas

Los resultados que brindaron conclusión a este proyecto revelan que existe una mejora en el desempeño de los clasificadores basados en algoritmos de aprendizaje de máquina cuando se someten las señales multicanal EEG relacionadas con intención de movimiento a distintos bloques de preprocesamiento. Cada bloque de preprocesamiento implementado en este proyecto mostró una variación en su efectividad, dependiendo del sujeto al cual pertenecen las señales. Esto muestra que cada sujeto, al momento de registrar intención de movimiento, puede verse sujeto a diferentes problemas que impiden el correcto desempeño de las máquinas de clasificación. Dependiendo del sujeto, un bloque o una combinación de distintos bloques de preprocesamiento pueden resultar efectivos para alcanzar el máximo desempeño posible. En la implementación de un sistema BCI práctico, esto implicaría el desarrollo de una etapa de calibración y adaptación. Esto implica un registro preliminar de señales, con el fin de que el sistema determine implementar bloques de preprocesamiento de señales, con el fin de maximizar el desempeño del sistema para recibir comandos. Esto es especialmente importante para el bloque de normalización, ya que, para la implementación de éste, se requiere computar una referencia adecuada, respecto a la cual convertir los voltajes registrados, para eliminar correctamente datos de amplitud atípicos, tanto entre pruebas como entre sujetos.

Respecto al bloque de corrección de *artifacts*, se demostró una efectividad para mejorar el porcentaje de desempeño, y los registros por número y tipo de *artifacts* muestran que, para esta base de datos, los componentes de ruido prevalentes fueron precisamente los de tipo ocular. Sin embargo, también se registró un número importante de componentes relacionados con ruido de alta frecuencia, y aquellos relacionados con el movimiento de los electrodos, que causa cambios en su impedancia y, por tanto, un cambio en su registro de voltaje. El

método FASTER para corrección de *artifacts* mostró efectividad para rechazo automático de *artifacts*, lo cual propone un método que podría resultar práctico para un sistema BCI online.

El bloque de normalización no mostro contundencia en la mejora del desempeño del sistema BCI. Solamente los sujetos 5 y 9 vieron mejora en los resultados de la clasificación versus las señales originales. Esto parece implicar que las variaciones en amplitud de las pruebas de estos sujetos fueron significativas para inhibir la extracción de características y clasificación, si bien no fue así para los demás sujetos.

Los resultados de la selección de canales muestran que es posible clasificar intención de movimiento utilizando un número de canales considerablemente menor al disponible. Más aún, el rechazo de canales irrelevantes para esta base de datos resultó en una mejoría en el porcentaje de desempeño para casi todos los sujetos de prueba. Esto implica que seleccionar los canales más importantes no sólo posibilita la obtención de suficiente información con menor número de electrodos, sino que también parece funcionar para rechazar canales que poseen información perjudicial para la clasificación.

El desempeño de la máquina AdaSVM en el desarrollo de este proyecto muestra no sólo una mayor rapidez para generar modelos de entrenamiento, sino que fue consistentemente superior a las otras máquinas de clasificación para discriminar intención de movimiento. Este resultado implica que la selección de características más importantes también contribuye a modelos de clasificación más confiables.

6.2 Conclusiones Personales

El desarrollo de este proyecto implicó la combinación de distintas áreas de la ciencia e ingeniería, como lo son la parte de procesamiento de señales, así como el aprendizaje de máquina y el conocimiento sobre registro de actividad cerebral. La

combinación de áreas de conocimiento implicó mucha preparación previa al inicio de la implementación del sistema de este proyecto. Sin embargo, el conocimiento generado y aportado también se distribuye a estas distintas áreas, por lo que considero que los resultados han sido fructíferos. El área de procesamiento de señales ha mostrado ser un área de enorme interés para mí, y considero que lo que se hizo para ella en este proyecto, ha terminado por consolidar conocimientos que deben ser considerados fundamentales en la Ingeniería Mecatrónica.

6.3 Recomendaciones

Debido a los resultados mostrados en este proyecto, considero de suma importancia el continuar con la búsqueda de mejora de técnicas de preprocesamiento, incluso poniendo como meta futura para este proyecto, generar técnicas innovadoras que permitan alcanzar un desempeño aún mayor que el logrado en este proyecto. Alcanzado ese objetivo, podría contemplarse la inversión en el desarrollo de un sistema BCI físico para poder elevar la sofisticación del proyecto de una metodología que involucre experimentación con bases de datos, a la experimentación usando datos registrados con nuevos sujetos de prueba, así como la demostración de la efectividad de los métodos de innovación en un sistema BCI online. De igual manera, es importante experimentar con algoritmos de aprendizaje de máquina, incluyendo la posibilidad de ramificar el alcance de este proyecto de Prácticum en una sección dedicada a la experimentación y posible desarrollo de algoritmos de aprendizaje.

Referencias

- [1] Leyva Bustillo R. A. & Giorgana Macedo G. R. (2016) "Clasificación de señales EEG para un sistema BCI de un solo canal basado en wavelets y Adaboost". Universidad Anáhuac Mayab, Yucatán, México.
- [2] Marrufo Peón P. P. & Giorgana Macedo G. R. (2017) "Clasificación de intención de movimiento para un sistema BCI no invasivo". Universidad Anáhuac Mayab, Yucatán, México.
- [3] Graimann, B., Allison, B., & Pfurtscheller, G. (2009). Brain-computer interfaces: A gentle introduction. In *Brain-Computer Interfaces*. Springer Berlin Heidelberg. Pág. 1-27
- [4] Wang D, Miao D and Blohm G (2012) Multi-class motor imagery EEG decoding for brain-computer interfaces. *Front. Neurosci.* **6**:151. doi: 10.3389/fnins.2012.00151
- [5] Shih, J. J., Krusienski, D. J., & Wolpaw, J. R. (2012). Brain-Computer Interfaces in Medicine. *Mayo Clinic Proceedings*, 87(3), 268–279. <http://doi.org/10.1016/j.mayocp.2011.12.008>
- [6] Szachewicz, P. (2013). *Classification of motor imagery for brain computer interfaces* (Master's thesis, Poznan University of Technology, 2013) (p. 1). Poznań.
- [7] Gollahalli, Akshay Raj. (2015). Brain-Computer Interfaces for Virtual Quadcopters based on a spiking-neural network architecture - NeuCube. . 10.13140/RG.2.1.3563.9768.
- [8] Kachenoura, A., Albera, L., Senhadji, L., & Comon, P. (2008). ICA: a potential tool for BCI systems. *IEEE Signal Processing Magazine*, 25(1), 57-68.
- [9] Zhou, Y., Mason, S. G., & Birch, G. E. (2004). Impact of an energy normalization transform on the performance of the LF-ASD brain computer interface. In *Advances in Neural Information Processing Systems* (pp. 725-732).
- [10] Wang, Y., Gao, S., & Gao, X. (2006, January). Common spatial pattern method for channel selelction in motor imagery based brain-computer interface. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the* (pp. 5392-5395). IEEE.

- [11] Anupama, H.S. & Cauvery, N.K. & Lingaraju, G.M.. (2012). Brain computer interface and its types-a study. International Journal of Advances in Engineering & Technology. 3. 739-745.
- [12] Savelainen, Antti. (2010). An Introduction to EEG artifacts, Independent research projects in applied mathematics, Mat-2.4108
- [13] Brunner, C., Leeb, R., Müller-Putz, G.R., Schloegl, A., and Pfurtscheller, G. (2008). *BCI Competition 2008 – Graz data set A*. Disponible en <http://bnci-horizon-2020.eu/database/data-sets>
- [14] Xu, M., Qi, H., Ma, L., Sun, C., Zhang, L., Wan, B., Ming, D. (2013). Channel Selection Based on Phase Measurement in P300-Based Brain-Computer Interface. *PLoS ONE*, 8(4), e60608. <http://doi.org/10.1371/journal.pone.0060608>
- [15] INEGI (2013). Población con limitación en la actividad según tipo de limitación [Imagen Digital]. Recuperado el 15 de septiembre de 2017 de <http://www.beta.inegi.org.mx/temas/discapacidad/>.
- [16] Lathi, B. P. (1998). *Signal processing and linear systems* (pp. 319-325). New York: Oxford university press.
- [17] Teplan, M. (2002). Fundamentals of EEG measurement. *Measurement science review*, 2(2), 1-11.
- [18] Budzynski, T. (2006). The clinical guide to sound and light. In *Proceedings of Brainwave Entrainment Symposium*.
- [19] Sawant, Hemant K., and Zahra Jalali. "Detection and classification of EEG waves." *Oriental Journal of Computer Science and Technology* 3, no. 1 (2010): 207-213.
- [20] Morley, A., Hill, L., & Kaditis, A. G. (n.d.). *10-20 system EEG Placement*. Lecture. Retrieved November 8, 2017, from <http://www.ers-education.org/lrmedia/2016/pdf/298830.pdf>
- [21] Pineda, J. A. (2005). The functional significance of mu rhythms: translating “seeing” and “hearing” into “doing”. *Brain Research Reviews*, 50(1), 57-68.
- [22] Zeng, H., & Song, A. (2014). Removal of EOG artifacts from EEG recordings using stationary subspace analysis. *The Scientific World Journal*, 2014.
- [23] Hyvärinen, A., Karhunen, J., & Oja, E. (2004). *Independent component analysis* (Vol. 46). John Wiley & Sons.

- [24] Arons, B. (1992). A review of the cocktail party effect. *Journal of the American Voice I/O Society*, 12(7), 35-50.
- [25] Halaki, M., & Ginn, K. (2012). Normalization of EMG signals: To normalize or not to normalize and what to normalize to?. In *Computational intelligence in electromyography analysis-a perspective on current applications and future challenges*. Intech.
- [26] Abdi, H. (2010). Normalizing Data. In *Encyclopedia of Research Design*. Sage.
- [27] Arvaneh, M., Guan, C., Ang, K. K., & Quek, H. C. (2010). EEG channel selection using decision tree in brain-computer interface. In *Proceedings of the Second APSIPA Annual Summit and Conference* (pp. 225-230).
- [28] Meng, J., Liu, G., Huang, G., & Zhu, X. (2009, December). Automated selecting subset of channels based on CSP in motor imagery brain-computer interface system. In *Robotics and biomimetics (ROBIO), 2009 IEEE international conference on* (pp. 2290-2294). IEEE.
- [29] Guyon, I., & Elisseeff, A. (2006). An introduction to feature extraction. *Feature extraction*, 1-25.
- [30] Haque, A. F. (2012). FFT and Wavelet-Based Feature Extraction for Acoustic Audio Classification. *International Journal of Advance Innovations, Thoughts & Ideas*, 1(1).
- [31] Chun-Lin, L. (2010). A tutorial of the wavelet transform. *NTUEE, Taiwan*.
- [32] Schapire, R. E. (2013). Explaining adaboost. In *Empirical inference* (pp. 37-52). Springer Berlin Heidelberg.
- [33] Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., & Arnaldi, B. (2007). A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of neural engineering*, 4(2), R1.
- [34] A. Hyvärinen and E. Oja. A Fast Fixed-Point Algorithm for Independent Component Analysis. *Neural Computation*, 9(7):1483-1492, 1997.
- [35] Jung, T.P., Makeig, S., Humphries, C., Lee, T.W., McKeown, M. J., Iragui, V., & Sejnowski, T.J. (2000). Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37(2), 163-178.

- [36] Nolan, H., Whelan, R., & Reilly, R. B. (2010). FASTER: fully automated statistical thresholding for EEG artifact rejection. *Journal of neuroscience methods*, 192(1), 152-162.
- [37] Aste, T. (2013). Generalized Hurst exponent (<https://la.mathworks.com/matlabcentral/fileexchange/30076-generalized-hurst-exponent?focused=3776833&tab=function>) MATLAB Central File Exchange. Retrieved May 18, 2006.