

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESCUELA DE POSGRADO



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

PROCESAMIENTO DE SEÑALES ELECTROENCEFALOGRAFICAS
EN UN SISTEMA EMBEBIDO PARA UNA INTERFAZ CEREBRO
MÁQUINA

Tesis para optar el grado de Magister en Ingeniería Mecatrónica.

Presentado por:

Kevin José Acuña Condori

Asesorado por:

M.Sc. David Ronald Achanccaray Diaz

Lima, Marzo 2017



RESUMEN

Una de las tecnologías actuales que está causando gran impacto en la vida de las personas con discapacidad motora severa es el Interfaz Cerebro-Máquina(BMI, por sus siglas en inglés), sistema que permite convertir pensamiento o intención de movimiento de una persona en medios de comunicación y comandos de control de dispositivos, logrando independencia para el usuario. Sin embargo, los equipos actuales dependen de una PC que realice el procesamiento de las señales cerebrales, lo que dificulta que el sistema sea portable y de bajo costo.

La presente tesis estudia y propone el uso de un sistema embebido (microcomputadora) como alternativa al uso de la PC en el BMI. Las microcomputadoras a diferencia de las PC comunes, son diseñadas para ciertos propósitos específicos, esto presenta una reducción de costo y mayor portabilidad del equipo. Con ello se pretende contribuir al desarrollo de esta nueva tecnología en el Perú haciéndolo accesible para personas de escasos recursos, lo que impactaría en la mejora de calidad de vida de las personas con discapacidad motora severa.

Los resultados muestran que el sistema embebido Odroid-xu4(que cuesta 20 veces menos y es 45 veces mas liviano) puede realizar el entrenamiento de los algoritmos y el procesamiento en tiempo real de señales EEG con la misma tasa de acierto que la laptop, tardando aproximadamente 9 veces más; sin embargo estos tiempos son mínimos para aplicaciones del interfaz cerebro-máquina por lo que se demuestra que el Odroid-xu4 puede ser usado como equipo de procesamiento para una BMI portable, confiable y de bajo costo.

AGRADECIMIENTOS

Agradezco al Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica (CONCYTEC) por darme la oportunidad de estudiar la presente Maestría.

Al Grupo de Investigación en Robótica Aplicada y Biomecánica (GIRAB) de la Pontificia Universidad Católica del Perú, por todo el apoyo brindado durante la realización del presente trabajo.



Índice de contenido

RESUMEN	2
AGRADECIMIENTOS	3
ÍNDICE DE TABLAS	6
ÍNDICE DE FIGURAS	7
GLOSARIO DE TÉRMINOS	9
INTRODUCCIÓN	10
1. ESTADO DEL ARTE	13
1.1. Interfaz cerebro-máquina	13
1.2. Paradigmas	14
1.2.1. Potencial relacionado a eventos	15
1.2.2. Potencial evocado por estados visuales	16
1.2.3. Potencial cortical lento	16
1.2.4. Ritmos sensorio-motores	16
1.3. Adquisición de señales cerebrales	18
1.3.1. Técnicas no invasivas y técnicas invasivas.	18
1.3.2. Electrodo Pasivos y Activos	19
1.4. Procesamiento: Extracción de características	20
1.4.1. Análisis en tiempo-frecuencia	20
1.4.2. Reducción de dimensiones	23
1.4.3. Filtros Espaciales	24
1.5. Procesamiento: Clasificador	25
1.5.1. Análisis discriminante lineal	26
1.5.2. Máquina de vectores de soporte	27
1.5.3. Vecinos más cercanos	28
1.5.4. Clasificadores Bayesianos	29
1.5.5. Redes neuronales artificiales	30
1.6. Sistemas embebidos	30
1.6.1. Microcomputadoras	30

2. DISEÑO Y CONFIGURACIÓN EXPERIMENTAL	33
2.1. Descripción del proyecto	33
2.1.1. Etapa 1: Entrenamiento	33
2.1.2. Etapa 2: Sistema online	35
2.2. Adquisición de señales EEG	36
2.2.1. Base de datos	36
2.2.2. Emulador de señales EEG (<i>Hardware in the loop</i>)	38
2.3. Equipos para el procesamiento	39
2.3.1. SBC procesador de señales EEG	39
2.3.2. Laptop basada en Linux	40
2.4. Aplicación	41
3. INTERFAZ CEREBRO-MÁQUINA	44
3.1. Etapa 1: Entrenamiento	44
3.1.1. Análisis de datos y pre-procesamiento	45
3.1.2. Extracción de características - Método 1	52
3.1.3. Extracción de características - Método 2	56
3.1.4. Clasificación	61
3.1.5. Validación	62
3.2. Etapa 2: Sistema online	63
3.2.1. Hardware in the loop	64
4. RESULTADOS Y DISCUSIÓN	66
4.1. Etapa 1: Entrenamiento	66
4.1.1. Método 1	66
4.1.2. Método 2	73
4.2. Etapa 2: Sistema online	77
CONCLUSIONES	79
RECOMENDACIONES	80
BIBLIOGRAFÍA	81

Índice de tablas

1.1. Métodos de adquisición de señales cerebrales.	19
2.1. Características de la base de datos.	36
2.2. Comparación entre los principales sistemas embebidos.	39
2.3. Características de la Laptop Asus.	40
2.4. Comparación entre los equipos de procesamiento.	41
2.5. Gestos asociados a tareas de imaginación motora.	42
3.1. Kernel considerados para optimizar el clasificador.	62
4.1. Resultado variando parámetros de procesamiento del Método 1 usando 60 canales	67
4.2. Resultados modificando el Kernel y sus parámetros usando 60 canales.	68
4.3. Resultados variando parámetros de procesamiento del Método 1 usan- do 32 canales.	69
4.4. Resultados modificando el Kernel y sus parámetros usando 32 canales.	69
4.5. Resultado variando parámetros de procesamiento del Método 1 usando 16 canales.	70
4.6. Resultados modificando el Kernel y sus parámetros usando 16 canales.	71
4.7. Resultado variando parámetros de procesamiento del Método 1 usando 8 canales.	72
4.8. Resultados modificando el Kernel y sus parámetros usando 8 canales.	72
4.9. Mejores resultados de cada configuración del Método 1.	73
4.10. Resultado de la evaluación de los parámetros de pre-procesamiento y extracción de características del Método 2.	75
4.11. Evaluación de Kernel y sus parámetros en el Método 2.	76
4.12. Mejores resultados en la etapa de entrenamiento	76
4.13. Tasa de acierto y tiempo de procesamiento en el sistema online.	78

Índice de figuras

1.1. Esquema de un Interfaz Cerebro Máquina.	14
1.2. Matriz del P300 Speller [1]	15
1.3. Método directo e indirecto de obtener la PSD [2].	21
1.4. Desplazamiento en el plano tiempo-frecuencia [3]	23
1.5. Hiperplano el cual separa 2 clases [4].	26
1.6. Selección del hiperplano óptimo.	27
1.7. Se muestran los datos de entrenamiento y como definen en dos grupos las 2 clases. A la izquierda con $k=1$, se observa un sobreajuste. A la derecha con $k=20$ una regla más generalizada [5].	28
1.8. SBC Odroid-xu4 [6].	31
1.9. SBC Raspberry Pi 3 [7].	32
1.10. SBC Banana Pi M3 [8].	32
2.1. Etapa de entrenamiento.	34
2.2. Etapa online.	35
2.3. Posición de los electrodos de la base de datos [9]	37
2.4. Temporización del paradigma.	37
2.5. Señales del protocolo asociadas a imaginación motora.	38
2.6. Secuencia del paradigma visualizado desde la pantalla.	38
2.7. Equipos a comparar en el procesamiento de señales EEG [10], [6].	41
2.8. Robot Inmoov [11].	42
2.9. Gestos del brazo robótico Inmoov.	43
3.1. Método 1 en la etapa de entrenamiento.	45
3.2. Método 2 en la etapa de entrenamiento.	45
3.3. Configuraciones dependiendo del número de electrodos a evaluar.	46
3.4. Diagrama de flujo de re-referencia CAR.	48
3.5. Efecto del filtro espacial CAR en el dominio del tiempo.	49
3.6. Comportamiento en frecuencia del filtro Butterworth (entre 8-30Hz) dependiendo del orden.	50
3.7. Efecto del filtro BT en el dominio de la frecuencia.	51
3.8. Efecto del filtro BT en el dominio de la tiempo.	51
3.9. Extracción de características del método 1 en la etapa de entrenamiento.	52

3.10. Patrones espaciales por cada clase de imaginación motora: mano izquierda, derecha, pies, lengua.	54
3.11. Diagrama de flujo de la extracción de características del método 1. . .	56
3.12. Extracción de características del método 2 en la etapa de entrenamiento.	57
3.13. Estructura de nivel de descomposición.	58
3.14. Wavelet madre.	59
3.15. Diagrama de flujo de la extracción de características del método 2. . .	61
3.16. Validación usando 10-veces validación cruzada	63
3.17. Trama a transmitir.	64
3.18. Diagrama de flujo del Servidor.	65
4.1. Comparación entre el sistema embebido y la laptop.	78



Glosario de términos

EEG: Electroencefalograma.

BMI: Brain Machine Interface. Interfaz Cerebro Máquina.

SBC: Single Board Computer.

MI: Motor Imagery. Imaginación Motora.

SMR: Sensorimotor Rhythm. Ritmos sensoriomotores.

ERP: Event Related Potential. Potencial relacionado a eventos.

SSVEP: Steady State Visually Evoked Potential.

SCP: Slow Cortical Potential. Potencial cortical lento.

ICA: Independent Component Analysis. Análisis de componente independiente.

PCA: Principal Component Analysis. Análisis de componente principal.

CAR: Common Average Reference. Referencia de promedio común.

FB: Filtro Butterworth.

PSD: Power Spectral Density. Densidad Espectral de Potencia.

WT: Wavelet Transform. Transformada Wavelet.

CSP: Common Spatial Pattern. Patrones espaciales comunes.

SVM: Support Vector Machine. Máquina de vectores de soporte.

MLP: Multi Layer Perceptron. Perceptrón Multicapa.

INTRODUCCIÓN

Según el Informe Mundial sobre la Discapacidad se estima que más de 1000 millones de personas viven con alguna discapacidad y lamentablemente esta cifra continua aumentando [12]. Esta condición impacta de forma drástica a las poblaciones económicamente vulnerables. Las personas con discapacidad no reciben el tratamiento adecuado, es muy complicado que consigan empleo y existen muchos obstáculos a nivel social y político. Por ello, brindar alternativas de solución para su desarrollo es transcendental.

Entre los tipos de discapacidades más graves, existe un grupo de enfermedades que afectan la capacidad física motriz, llegando en algunos casos a postrar a pacientes en cama y necesitar asistencia de por vida. Una de estas enfermedades es la Esclerosis Lateral Amiotrófica (ELA), la cual es una enfermedad degenerativa que causa la pérdida progresiva de la función motriz, y en estados más avanzados, pérdida del habla, debilidad y falta de control físico. En el Perú se estima que al menos 1200 personas son afectadas por esta grave enfermedad. Según Gabriela Zárate Vargas, presidenta de la Asociación de Esclerosis Lateral Amiotrófica del Perú un paciente con ELA se vuelve cada vez más dependiente de su entorno, por lo que cualquier equipo que implique un grado de independencia conlleva a grandes mejoras físicas y psicológicas para el paciente.

Otras enfermedades que conllevan a una discapacidad motora severa son: atrofia muscular espinal (AME) que compromete la capacidad de caminar, respirar y controlar movimientos y la lesión en la médula espinal (LME) que genera pérdida de sensibilidad y movilidad. También el síndrome de enclaustramiento (LIS, por sus siglas en inglés) el cual es un estado en que las personas están despiertas y conscientes de su alrededor pero no pueden hablar ni realizar movimientos faciales o corporales. Todas ellas involucran principalmente la capacidad de comunicarse o moverse por sus propios medios, por lo que requieren de un canal de comunicación alternativo que les permita exteriorizar sus pensamientos, sentimientos, emociones e interactuar con el medio físico por medio de la capacidad de controlar dispositivos externos tales como: prótesis, sillas de ruedas, brazos robóticos, etc.

Debido a la discapacidad motriz severa se requiere que las órdenes enviadas por el paciente sean registradas desde la concepción de la idea; es decir, desde el cerebro. Para ello el método más usado para registrar la actividad cerebral es el electroencefalograma (EEG), porque es una técnica no invasiva y de menor costo comparado con otras técnicas.

Con el avance tecnológico se han desarrollado dispositivos para asistencia, que si bien no devuelven la funcionalidad al paciente de manera íntegra, ofrecen alternativas para que este pueda tener una mayor autonomía. Entre este tipo de dispositivos destacan las Interfaces Cerebro Máquina (BMI, por sus siglas en inglés), las cuales se vienen usando alrededor de hace 20 años como forma alternativa de comunicación y control en pacientes con discapacidad motriz. En el Perú esta tecnología es reciente y aún no se aplica como alternativa en los hospitales y clínicas debido a que se cuenta con pocos especialistas en el tema.

Recientes estudios han demostrado que si bien las BMI fueron diseñadas para personas con discapacidad motriz, estos dispositivos aún no satisfacen las necesidades reales de estos pacientes. Diversas investigaciones muestran que las prioridades que deberían tener los BMI en pacientes con ELA son la precisión, simplicidad, funcionalidad y fácil operacionalización; debe poder ser usado no sólo en clínicas, sino también, en casa. Se resalta la necesidad que el dispositivo sea intuitivo, robusto, portable, minimalista y de bajo costo, porque debe funcionar a pesar de las restricciones cognitivas y físicas del usuario.

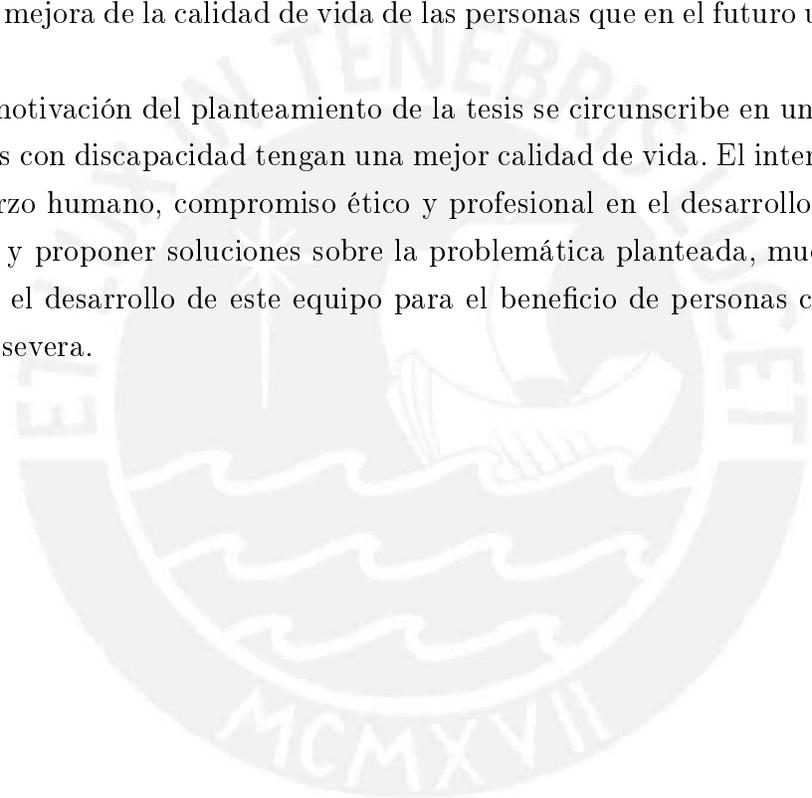
Una de las principales alternativas para subsanar a los equipos actuales que dependen de una PC para realizar el procesamiento de las señales cerebrales es el uso de sistemas embebidos o microcomputadoras. En la actualidad se cuentan con computadoras embebidas, que a diferencia de las PC comunes, son diseñadas para ciertos propósitos específicos. Debido a esto, estos sistemas embebidos son de un tamaño reducido, tienen un costo muy bajo y además cuentan con los recursos computacionales para realizar tareas de alta demanda.

El interés científico del desarrollo de la presente tesis está en el diseño e implementación de una interfaz cerebro máquina portable y de bajo costo, por medio de la ejecución de un software que controla el sistema en una microcomputadora, pro-

fundizando en el área del procesamiento de señales electroencefalográficas, que es la parte crucial del proyecto. Hay que tener en cuenta que este equipo no solo podrá ser aplicado en la interfaz cerebro-máquina, sino que puede ser de gran ayuda para diversos estudios de neurociencia.

Esta propuesta innovadora fomenta a su vez la integración de las áreas de medicina, psicología, rehabilitación e ingeniería y difunde las nuevas tecnologías biomédicas que pueden ser aplicadas en el Perú para personas con discapacidad motora severa. Debido al reducido costo y gran gama de aplicaciones (control de silla de ruedas, uso en prótesis, escritura mirando letra a letra, etc.) se espera un impacto positivo en la mejora de la calidad de vida de las personas que en el futuro usen este equipo.

La motivación del planteamiento de la tesis se circunscribe en un ideal en que las personas con discapacidad tengan una mejor calidad de vida. El interés investigativo, el esfuerzo humano, compromiso ético y profesional en el desarrollo del proyecto de indagar y proponer soluciones sobre la problemática planteada, muestra el compromiso en el desarrollo de este equipo para el beneficio de personas con discapacidad motora severa.



Capítulo 1:

Estado del arte

1.1. Interfaz cerebro-máquina

Brindar un medio de comunicación alternativo para aquellas personas que perdieron de forma parcial o total esta capacidad es uno de los objetivos más importantes de este siglo.

Interfaz Cerebro-Máquina, en inglés *Brain-Machine Interface* (BMI) se define como: “Un sistema que mide la actividad del sistema nervioso central (SNC) y la convierte en la producción artificial que sustituye, restaura, o mejora la producción natural de SNC y con ello cambia las interacciones en curso entre el SNC y de sus relaciones exteriores o ambiente interno”[13]. También es definido como: “Un sistema que se basa en el uso de actividad mental de la persona para generar comandos de control para un dispositivo” [14]. Es usado en aplicaciones tales como el control de un cursor [15] [16], el control de una silla de ruedas [17], deletreadores [18], etc.

Este es un campo de investigación activo que tiene como objetivo ayudar a las personas que sufren de discapacidad, de tal manera que se crean interfaces alternativas de comunicación con su entorno. Para evidenciar la necesidad de esto, podemos encontrar la siguiente cita que un paciente con Esclerosis Lateral Amiotrófica (ELA): “Todo el tiempo que puedo comunicar correctamente con mi voz, mis ojos o una máquina o lo que sea, yo quiero tener un respirador. Pero tan pronto como ya no pueda comunicarme, eso es todo, no quiero seguir conectado” [13].

Cuando las personas sufren de discapacidad motora, pierden autonomía y requieren de un acompañamiento permanente, incluso para el desempeño de actividades cotidianas básicas [19]. Esta pérdida progresiva de comunicación y autonomía afecta anímicamente y reduce la calidad de vida de los pacientes. Para ello, una solución

que se propone es el uso de Interfaces Cerebro-Maquina; sin embargo, aún presenta un desafío físico pues los usuarios desean llevar una vida normal sin la necesidad de ningún instrumento voluminoso [20] ya que esto dificulta la portabilidad del equipo, eleva costos y lo hace dependiente.

En la Figura 1.1 se muestra el esquema general de una Interfaz Cerebro-Máquina. Este se compone de tres fases principales: adquisición, procesamiento y aplicación. Adicionalmente una BMI cuenta con una retro-alimentación para que el usuario aprenda y mejore su desempeño con el tiempo.

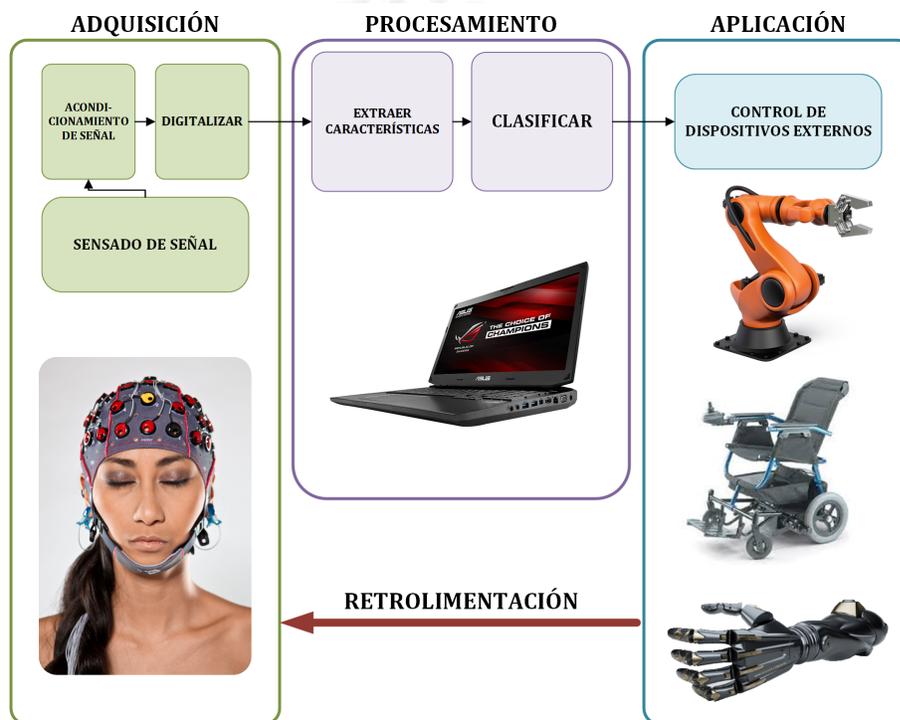


Figura 1.1: Esquema de un Interfaz Cerebro Máquina.

Fuente: Diagrama elaboración propia. Imágenes [21], [10], [22].

1.2. Paradigmas

Numerosos estudios han descrito un gran grupo de señales cerebrales que puede ser útiles como señales de control en BMI [23]. Sin embargo, los paradigmas más difundidos en la actualidad son: Potencial relacionado a eventos (ERP, por sus siglas en inglés), potencial evocado por estados visuales (SSVEP, por sus siglas en inglés), potencial cortical lento (SCP, por sus siglas en inglés) y ritmos sensorio-motores.

Los paradigmas ERP y SSVEP son clasificados como *exogenous* debido a que requieren de un estímulo externo, mientras que SCP y ritmos sensorio-motores se consideran *endogenous* ya que son evocados por el usuario sin necesidad de un estímulo externo.

1.2.1. Potencial relacionado a eventos

Potencial relacionado a eventos, en inglés Event-Related Potential, es definido como un cambio de potencial en la actividad neuronal asociada con un específico evento tal como un estímulo visual o auditivo [24]. Dentro de este paradigma, el P300 se ha convertido en uno de lo más populares en el campo de BMI ya que se caracteriza por tener una variación en la amplitud de la señal EEG, que ocurre aproximadamente 300ms después de un estímulo no frecuente. El paradigma P300 es extensamente usado para Deletreadores BMI, estos muestran una matriz de letras introducida inicialmente por Farwell y Donchin [25] (Figura 1.2), que se iluminan de forma aleatoria mientras la persona se enfoca en la letra que quiere seleccionar, para luego ser detectada y mostrada por el sistema BMI.

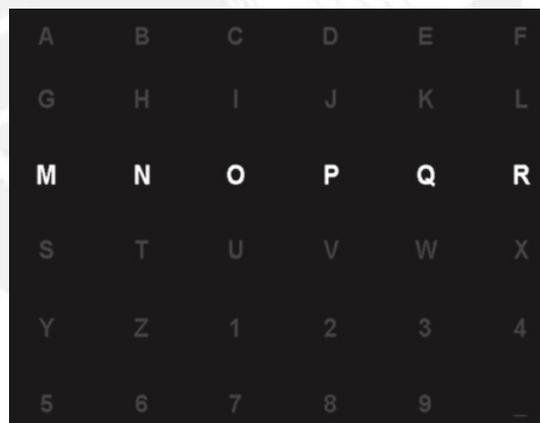


Figura 1.2: Matriz del P300 Speller [1]

El deletreador basado en P300 es especialmente útil para personas con esclerosis lateral amiotrófica, accidente cerebro-vascular, lesión en la médula espinal o el cerebro, parálisis cerebral, distrofias musculares, esclerosis múltiple y otras discapacidades que impiden comunicarse de una forma normal [1]. Mejoras del uso del P300 Speller han sido propuestas, por ejemplo como navegador de páginas web [23].

1.2.2. Potencial evocado por estados visuales

Potencial evocado por estados visuales, en inglés Steady State Visual Evoked Potential (SSVEP), es una respuesta a una estimulación externa repetitiva visual; por lo general, oscila entre las frecuencias fundamentales y armónicas del estímulo.

En una configuración típica de un BMI que utiliza el paradigma SSVEP, los objetivos se representan generalmente al usuario de una manera parpadeante sea una luz LED, un patrón sobre una pantalla LCD, etc. El tipo de dispositivo de estimulación, el color y la forma del estímulo, la banda de frecuencia de la velocidad de oscilación y la fase entre los estímulos son características de los estímulos de luz que pueden afectar a la SSVEP.

Debido a que SSVEP es una respuesta a un estímulo visual repetitivo, esta es prominente y observable a través de los lugares occipitales cerca de la corteza visual primaria.

1.2.3. Potencial cortical lento

Potencial cortical lento, en inglés Slow Cortical Potential (SCP), es un paradigma que utiliza los cambios de voltaje en el EEG que se producen lentamente con el tiempo sobre la corteza cerebral, por ejemplo, entre 0,5-10 segundos a frecuencias menores a 1Hz. Está asociado a funciones que implique la activación cortical. Debido a que estos potenciales lentos pueden ser conscientemente regulados por las personas sanas y paralizadas, SCP es una opción para el diseño de un BMI.

Utilizar el paradigma SCP requiere de entrenamiento y el tiempo que está sujeto a condiciones psicológicas y físicas del paciente, según esto, no todas las personas tienen la misma tasa de acierto bajo un sistema BMI con SCP.

1.2.4. Ritmos sensorio-motores

Ritmos sensorio-motores, en inglés Sensorimotor Rhythms (SMR), es un paradigma en el cual se usa las señales mu (9-12Hz) y beta (13-40Hz) que tienen ritmos prominentes sobre la corteza sensorio-motora. Estos cambios ocurren cuando el suje-

to realiza, imagina u observa un movimiento [26]. Los términos SMR e imaginación motora (MI, por sus siglas en inglés) se pueden utilizar indistintamente para definir este paradigma.

Los trabajos están orientados a las personas que padecen daños motores severos, por lo que es idóneo trabajar con paradigmas *endogenous* o espontáneos (generados internamente por la actividad mental del usuario). Entre las aplicaciones ampliamente difundidas está el control de un robot asistencial, el cual representa un reto debido al número de grados de libertad del robot.

El número de tareas mentales y la tasa de acierto son factores importantes en las investigaciones, se ha desarrollado un interfaz para el movimiento en el plano 2D del brazo robótico PuParm usando un control jerárquico de 2 tareas mentales (MI: mano derecha o mano izquierda) seleccionando inicialmente el eje a moverse y luego la dirección de dicho movimiento con 88 % de precisión [20]. Arnil propone el uso de 3 tareas mentales para el movimiento de un brazo robótico compuesto de 3 servomotores, desarrollado como una prótesis básica, donde se usa el movimiento del brazo izquierdo para giro anti horario, el del brazo derecho para giro horario y el de ambos para cambiar de articulación [27]. Se puede manipular la trayectoria de un brazo robot asistente [19] y para el movimiento de piezas efectuada por un brazo robótico implementado con RFID usando 2 tareas mentales de imaginación motora para desplazarse a la derecha o izquierda y el estado mental de reposo para agarrar o soltar los objetos situados en el experimento [28]. El control de un brazo robótico modelo LR-Mate 200iB con 6 grados de libertad es realizado con el uso de 4 tareas mentales, para el movimiento en un plano con objetivos se usan las tareas de imaginación motora para moverlo hacia la derecha o izquierda: contar de forma descendente desde el 20 hace ir hacia adelante el brazo y recitar el alfabeto lo hace ir hacia atrás, todo con el objetivo de poder llevar un objeto de un punto a otro en un determinado tiempo [29].

Los métodos usados para la clasificación y extracción de características para este paradigma de imaginación motora son diversos y depende del número de clases. Para distinguir entre cuatro clases de imaginación motora (mano izquierda, mano derecha, ambas manos y reposo) se ha usado el algoritmo de patrones espaciales comunes (CSP, por sus siglas en inglés) y redes neuronales (NN, por sus siglas en inglés) [26], otras cuatro clases de imaginación motora (mano izquierda, mano derecha, pies y

lengua) son clasificadas usando CSP y NN [30]. Se ha demostrado que la elección de un conjunto de cuatro clases de MI (mano izquierda, mano derecha, ambas mano y reposo) tiene el mismo desempeño que otro conjunto de cuatro clases de MI (mano izquierda, mano derecha, ambos pies y reposo) usando el algoritmo CSP para extraer características y el análisis discriminante lineal (LDA, por sus siglas en inglés) para clasificar [31].

El ganador de la competición de interfaz cerebro máquina, que empleó la base de datos II, aplicó para la clasificación de cuatro clases de MI (mano izquierda, mano derecha, pies y lengua) el algoritmo de banco de filtros en patrones espaciales comunes (FBCSP, por sus siglas en inglés) y un clasificador bayesiano [32], estas mismas cuatro clases son clasificadas usando la Transformada de Fourier de tiempo corto, CSP y máquina vectores de soporte (SVM, por sus siglas en inglés) [33], en otro enfoque se usó parámetros autoregresivos para extraer las características y se clasificó usando diversos métodos: k-vecinos más cercanos, LDA, SVM y distancia Mahalanobis, eligiendo como el mejor clasificador a SVM [34].

1.3. Adquisición de señales cerebrales

Existen diversos enfoques para la adquisición de señales cerebrales, la elección del método y el tipo de electrodo queda a criterio de cada investigación. Para ello, se toma en cuenta principalmente: la calidad de señal requerida, la usabilidad, principios éticos y económicos.

1.3.1. Técnicas no invasivas y técnicas invasivas.

Las técnicas no invasivas se caracterizan por no requerir ningún tipo de cirugía. Entre estas técnicas tenemos: Electroencefalografía (EEG) es la modalidad más utilizada debido a su alta resolución temporal, bajo costo y portabilidad [23], Magnetoencefalografía (MEG) mide campos magnéticos producidos por la actividad eléctrica del cerebro, *Near-infrared spectroscopy* (NIRS) mide el flujo sanguíneo en el cerebro asociado a actividad cerebral [35]. Otra técnica no invasiva es la imagen por resonancia magnética funcional (fMRI, por sus siglas en inglés) el cual detecta el cambio de volumen local de sangre en el cerebro, el flujo sanguíneo y el nivel de oxígeno.

Las técnicas invasivas hacen uso de tecnología invasiva de sensores, que implantan un arreglo de microelectrodos dentro del cerebro, lo cual representa un riesgo médico. Dos métodos son usados en BMI: Electrococtografía(EECoG), que coloca electrodos en la superficie de la corteza y Grabación Neuronal Intracortical, que implanta electrodos dentro de la corteza cerebral. Estas son usadas con el fin de tomar medidas de mejor calidad en mayor resolución. Además son aplicadas en mediciones de señales cerebrales de seres humanos con el fin de controlar prótesis [36].

En la Tabla 1.1 se muestran las principales características de los métodos de medición tanto invasivos como no invasivos.

Tabla 1.1: Métodos de adquisición de señales cerebrales.

Método	Variable medida	Técnica	Portabilidad
EEG	Eléctrica	No invasiva	Portable
MEG	Magnética	No invasiva	No portable
NIRS	Metabólica	No invasiva	Portable
fMRI	Metabólica	No invasiva	No portable
EECoG	Eléctrica	Invasiva	Portable
Intracortical	Eléctrica	Invasiva	Portable

Fuente: Elaboración Propia

1.3.2. Electrodo Pasivos y Activos

Los electrodos pasivos son discos de metal diminutos basado en materiales como: estaño, oro, plata o de plata/cloruro de plata (Ag/AgCl) los cuales requieren estar conectados a un equipo amplificador a través de cables eléctricos. La señal que adquieren es amplificada en el circuito amplificador externo. Dado que las señales cerebrales adquiridas a través del cuero cabelludo tienen amplitudes que oscilan entre 10-100 μ V [37], las fuentes de ruido de mayor amplitud como movimientos de la cabeza, los factores ambientales y ruido en la línea eléctrica tiene el riesgo de distorsionar las señales. Para evitar este problema tanto como sea posible, los cables de electrodos deben ser cortos, apantallados y fijos. Un buen contacto entre el cuero cabelludo y los electrodos también es crucial para mejorar la relación señal a ruido (SNR) de las señales adquiridas. Una capa de pasta de gel conductor es generalmente aplicado antes de grabaciones para reducir la impedancia de la piel-electrodo.

Los electrodos activos hacen uso de un pequeño circuito de amplificación incorporado en el electrodo para elevar la amplitud de la señal del electrodo. Las señales del cerebro pre-amplificadas son por lo tanto más robustas frente a las fuentes de ruido externas mencionadas anteriormente, ya que los ruidos aditivos no están pre-amplificados.

Los electrodos pasivos son generalmente el tipo de electrodos más utilizados en los registros de señales EEG, ya que son fáciles de diseñar y no son costosos.

1.4. Procesamiento: Extracción de características

Existen diversos métodos para extraer características procesando las señales EEG. En esta sección se presentan los métodos de análisis en tiempo-frecuencia, reducción de dimensiones y filtros espaciales.

1.4.1. Análisis en tiempo-frecuencia

Diversos patrones en las señales EEG se presentan en el dominio de la frecuencia, por lo que es importante realizar un análisis que no solo contemple las características temporales sino también las de frecuencia.

Densidad espectral de potencia

La densidad espectral de potencia (Power Spectral Density, PSD) es una función que nos indica como está distribuida la potencia de una señal sobre un rango de frecuencias. Howard [38] define dos distintas pero equivalentes modos de hallar PSD, la primera es vía descomposición, esta dada por la transformada de Fourier; la segunda, es a través de la transformada de Fourier de una función de autocorrelación. Kay *et al* [2] desarrolla y detalla los métodos directo e indirecto para obtener la PSD, esto se muestra en la Figura 1.3.

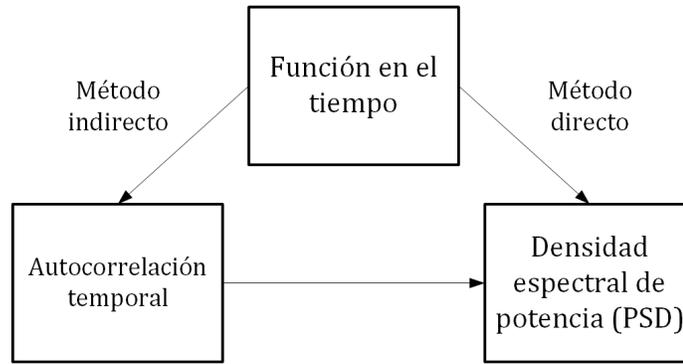


Figura 1.3: Método directo e indirecto de obtener la PSD [2].

Técnicas como el Classical Periodogram, Classical Blackman-Tukey, Moving Average (MA), Autoregressive (AR), Autoregressive-moving average (ARMA), maximum likelihood, Prony y Pisarenko son ampliamente desarrolladas para estimar la densidad espectral de potencia (PSD) [2].

Los métodos paramétricos (modelo AR, AM, ARMA) de la estimación del espectro de potencia evitan el problema de las fugas y proporcionan una mejor resolución en frecuencia que los métodos no paramétricos basados en el algoritmo FFT (Classic Periodogram, Blackman-Tukey, Bartlett, Welch), especialmente cuando hay un registro de datos corto.

La estimación del PSD usando la técnica auto regresiva fue exitosamente implementada en un sistema embebido (Beagle Bone Black), este emplea la PSD para extraer características de las señales EEG, las cuales fueron estimuladas usando el paradigma SSVEP [39].

Componentes autoregresivos

Componentes autoregresivos (AutoRegressive Components, AR) es un método de estimación espectral ampliamente usado para modelar señales. AR modela las señales EEG como salida de un filtro, donde los coeficientes del filtro obtenidos pueden ser usados como características de la señal [4]. Matemáticamente se puede representar este método de estimación con la siguiente ecuación:

$$y(t) = a_1y(t - 1) + a_2y(t - 2) + a_3y(t - 3) + \dots + a_p y(t - p) + n(t) \quad (1.1)$$

Donde a_i es en i -ésimo coeficiente del filtro, p es en orden del filtro y $n(t)$ representa el ruido. Para hallar a_1, a_2, \dots, a_p existen algoritmos como Yule-Walker, Burg, covariance o forward-backward [2]. Con estos coeficientes se puede dar la siguiente estimación de PSD de la señal $y(w)$:

$$y(w) = \frac{1}{|1 - \sum_{k=1}^p a_k e^{-jkw}|^2} \quad (1.2)$$

Estimación espectral AR es preferida a la transformada de Fourier porque da una mejor resolución para segmentos pequeños de tiempo.[40].

Transformada de Wavelet

La transformada de Wavelet (Wavelet Transform, WT) es una herramienta matemática útil para señales no estacionarias. Debido a que las señales EEG son no estacionarias en general, es más apropiado el uso de dominios tiempo-frecuencia tal como la WT [41].

Las WT son funciones de diferentes frecuencias y duración finita que permite simultáneamente un estudio en dominio tiempo-frecuencia, a diferencia de Fourier Transform (FT), Short Time Fourier Transform (STFT). WT propone una mejora a FT, sin embargo presenta algunas limitaciones como la contrapartida entre resolución de escalas tiempo y frecuencia. Una comparación entre FT, STFT y WT se muestra en la Figura 1.4.

La WT se basa en el desplazamiento y dilatación de una Wavelet madre (Mother Wavelet), diferentes tipos de familias de Mother Wavelet son usados: Haar, Daubechies, Symlets, Coiflets, BiorSplines, Meyer, Mexican hat, Morlet, Shannon, entre otros. Esta Wavelet madre es desplazada en el tiempo y escalada en amplitud como se muestra en la siguiente ecuación:

$$\Psi_{s,\tau}(t) = \frac{\Psi\left(\frac{t-\tau}{s}\right)}{\sqrt{s}} \quad (1.3)$$

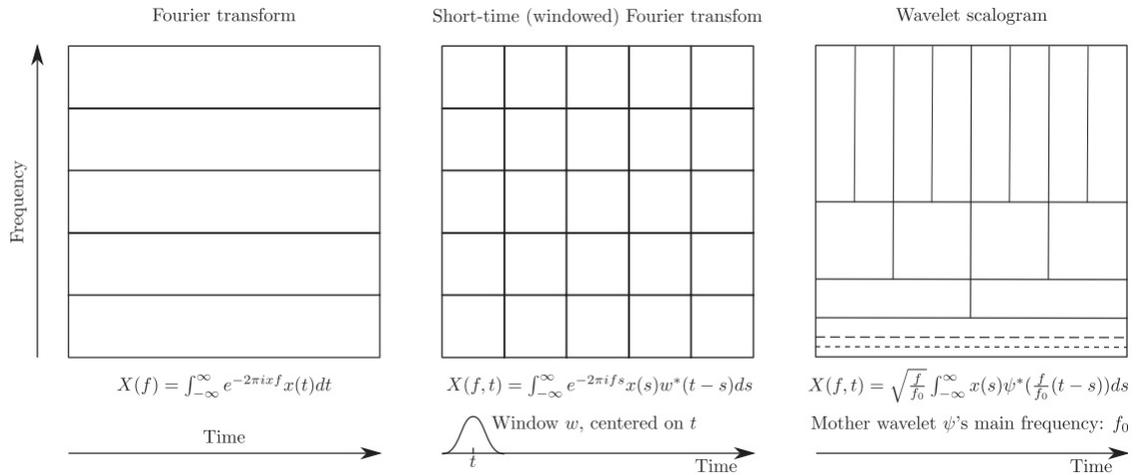


Figura 1.4: Desplazamiento en el plano tiempo-frecuencia [3]

Donde Ψ simboliza la Wavelet madre, τ es el desplazamiento en el tiempo y s el escalamiento de la Wavelet madre.

Por otro lado, existen dos tipos de análisis de WT: Continuous Wavelet Transform (CWT) y Discrete Wavelet Transform (DWT). DWT es generalmente más eficiente al reducir la complejidad de CWT con un costo computacional menor [42], por lo cual es empleado en sistemas embebidos. Para la implementación en hardware se han propuesto arquitecturas en VLSI y FPGA [43]. Así mismo, ha sido aplicado en la microcomputadora PandaBoard como parte de la etapa de extracción de características de un sistema BMI [44].

1.4.2. Reducción de dimensiones

Análisis de componentes independientes

Análisis de componentes independientes (Independent Component Analysis, ICA) es un procedimiento estadístico que divide un conjunto de señales mixtas en sus fuentes, asume que las fuentes son mutuamente independientes, como en el EEG que provienen de múltiples actividades cognitivas y artefactos.

$$x(t) = f(s(t)) + n(t) \tag{1.4}$$

Donde $\mathbf{x}(\mathbf{t})$ representan los datos, cuya dimensión es el número de canales medidos, \mathbf{f} es una señal de mezcla desconocida, $\mathbf{s}(\mathbf{t})$ es la señal que depende de las fuentes y $\mathbf{n}(\mathbf{t})$ es un ruido aditivo.

La resolución del problema de ICA consiste en el cálculo de la función inversa de \mathbf{f} , para obtener una estimación de $\mathbf{s}(\mathbf{t})$, que tiene una menor dimensión que nuestra entrada $\mathbf{x}(\mathbf{t})$. Para lo cual se hacen ciertas simplificaciones del modelo, tal como la asunción de una función textbff lineal y un ruido débil, para luego calcular la función y la fuente a partir de la entrada usando algoritmos tal como Infomax [45].

ICA puede ser útil para remover la interferencia eléctrica tal como el movimiento de los ojos o el parpadeo. Señales más limpias pueden ser construidas usando ICA y análisis en tiempo-frecuencia [46].

Análisis de componente principal

Análisis de componente principal (Principal Component Analysis, PCA) es un método de extracción de características estadístico utilizado ampliamente para el análisis de los datos.

PCA usa una transformación lineal que convierte un conjunto de puntos observados correlacionados a un conjunto de variables no correlacionadas llamadas componentes principales. PCA busca una óptima representación de los datos, transformándolos a un nuevo sistema de coordenadas donde se analizan los componentes principales para quedarse solo con las que más información aportan al sistema, reduciendo así la dimensión de características.

PCA logra esto mediante el cálculo de los autovectores y autovalores de la matriz de covarianza de los datos de EEG [47].

1.4.3. Filtros Espaciales

Además de la información en tiempo y frecuencia, las señales EEG también presentan información espacial la cual tiene que ser analizada. Los filtros espaciales buscan capturar esta información y utilizarla para representar las señales EEG.

Patrones espaciales comunes

Patrones espaciales comunes (Common Spatial Pattern, CSP) es un método de extracción de características que procesa señales de múltiples canales de EEG para diseñar filtros espaciales óptimos que maximizan la varianza para un conjunto de datos y minimizan la varianza para el otro, resaltando las diferencias y minimizando las similitudes. Esta siendo usado además para la selección y reducción de canales EEG [48] [49].

El método empleado por el algoritmo de CSP se basa en la diagonalización simultánea de dos matrices de covarianza. La señal filtrada espacialmente Z de un ensayo E se da como:

$$Z = WE \quad (1.5)$$

Donde E es una matriz de $N \times T$, donde N es el número de muestras por canal y T es el número de canales. W es la CSP matriz de proyección [50].

CSP incrementa la precisión para BMI síncronos, sin embargo no ofrece la misma mejora para BMI asincronos [23]. Por lo que existen diversos algoritmos mejorados basados en CSP como Wavelet Common Spatial Pattern (WCSP), Common Spatio-Spectral Pattern (CSSP), Common Sparse Spectral Spatial Pattern (CSSSP), Filter Bank Common Spatial Pattern (FBCSP) [50], Analytic Common Spatial Pattern (ACSP) [51].

1.5. Procesamiento: Clasificador

Los clasificadores lineales son algoritmos discriminantes que usan funciones lineales para distinguir clases. Dos tipos de clasificadores lineales son ampliamente usados para BMI: Linear Discriminant Analysis (LDA) y Support Vector Machine (SVM) [4] [23]. Clasificadores no lineales como Nearest Neighbor, Bayesian Classifiers y Neural Network también son usados para diversos tipos de aplicaciones en BMI.

1.5.1. Análisis discriminante lineal

Análisis discriminante lineal (LDA, por sus siglas en inglés) es un clasificador que brinda un porcentaje de éxito aceptable con un bajo costo computacional, su objetivo es usar hiperplanos para separar los datos en diferentes clases [4], como se observa en la Figura 1.5.

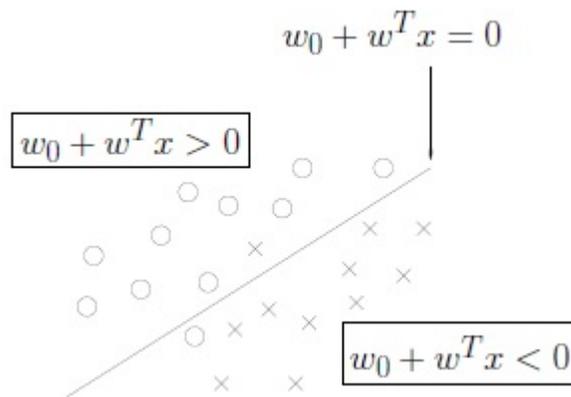


Figura 1.5: Hiperplano el cual separa 2 clases [4].

Es trabajado en BMI multiclase, síncrono y ampliamente usado para P300 Speller (LDA, FLDA, BLDA, SWLDA) [1].

Para un problema de dos clases, el plano de decisión puede ser representado como:

$$g(x) = w^T x + w_0 \quad (1.6)$$

Donde x es el vector de características, w es el vector de pesos y w_0 el offset. Dependiendo del signo de la función la entrada x es asignada a una clase.

Otros algoritmos de clasificación que se basan en LDA son:

- Fisher Linear Discriminant Analysis (FLDA): Es una modificación robusta del LDA, maximiza la distancia entre dos clases proyectando los datos a un espacio de menor dimensión.
- Regularized Fisher's LDA (RFLDA): Introduce un parametro de regulación C para penalizar errores de clasificación.

- Bayesian Linear Discriminant Analysis (BLDA): Es una extensión del FLDA, introduce un método estadístico llamado regularización, que previene el sobreajuste. Usa un análisis bayesiano para la estimación del vector discriminante. Provee mayor exactitud de clasificación que FLDA por un ligero tiempo de computo mayor.

1.5.2. Máquina de vectores de soporte

Máquina de vectores de soporte (SVM, por sus siglas en inglés) es un clasificador que construye hiperplanos y elige el que maximiza los márgenes, para así incrementar sus capacidades de generalización. El resultado del hyperplano óptimo es una combinación de algunos puntos conocidos como vectores de soporte que se muestra en la Figura 1.6. Usa un parámetro de regularización C , el cual previene al clasificador de acomodar posibles datos ruidosos y previene el sobreentrenamiento. Es ampliamente usado para BMI binario y multiclase.

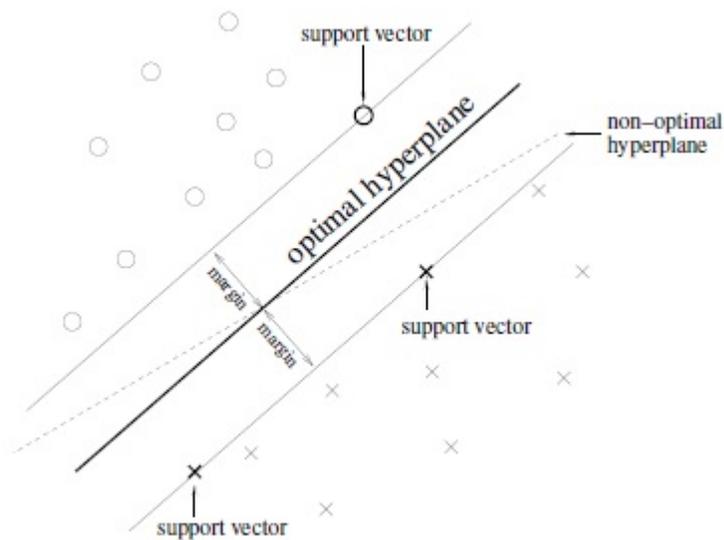


Figura 1.6: Selección del hiperplano óptimo.

Fuente: [4]

Es posible crear un SVM con límites de decisión no lineales, esto incrementa las opciones para optimizar el algoritmo con un bajo incremento en la complejidad. Para esto se usa los kernel, los cuales se basan en diversas funciones tal como la polinomial, gaussiana o Radial Basis Function (RBF). Estas pueden entrenarse con vectores de

características con altas dimensiones y son lo suficientemente rápido para un BMI en tiempo real.

1.5.3. Vecinos más cercanos

Es un clasificador no lineal, supervisado, relativamente simple, que busca hallar distancias mínimas para pertenecer a una clase. Esta distancia puede ser Euclidiana o de Mahalanobis.

K-vecinos más cercanos (kNN)

Tiene por premisa el principio de que las características que pertenecen a diferentes clases usualmente se separan en diferentes grupos en el espacio de características, por lo que un vector a clasificar estará más cercano a los vectores que pertenecen a su clase, que a vectores que pertenecen a otras clases.

Es necesario suficientes datos de entrenamiento y un k que define el número de vecinos a encontrar produciendo límites de decisión no lineales como se muestra en la Figura 1.7. Este algoritmo es sensible a la dimensionalidad del vector de características, siendo efectivo a bajo número de dimensiones para el uso en BMI.

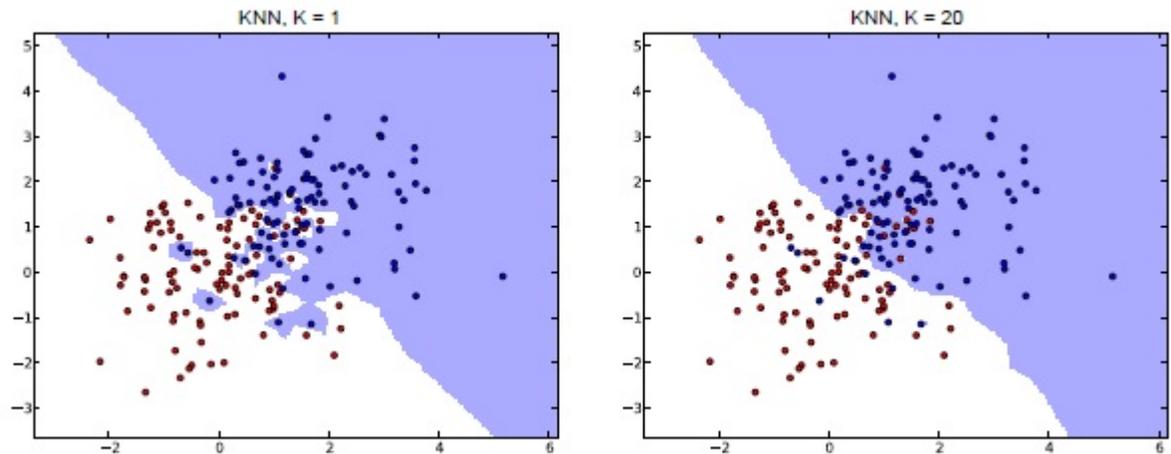


Figura 1.7: Se muestran los datos de entrenamiento y como definen en dos grupos las 2 clases. A la izquierda con $k=1$, se observa un sobreajuste. A la derecha con $k=20$ una regla más generalizada [5].

Mahalanobis Distance

A diferencia de kNN no usa una distancia métrica; sino, la distancia de Mahalanobis. Ésta es aplicada desde el vector característico hasta el prototipo de clase c , que se hace por cada clase. Mahalanobis, ideo esta distancia para considerar la importancia de cada característica, teniendo en cuenta la matriz de varianza covarianza. Esta distancia es definida como:

$$d_c(x) = \sqrt{(x - u_c)M_c^{-1}(x - u_c)^T} \quad (1.7)$$

Donde cada prototipo de clase tiene su media u y su matriz de varianza covarianza M . Este clasificador es usado para BMI multiclase y asíncrono.

1.5.4. Clasificadores Bayesianos

Son clasificadores no lineales estadísticos que predicen la probabilidad de pertenecer a una clase [52]; su objetivo es clasificar un vector de características \mathbf{x} dependiendo de cual es la probabilidad de que pertenezca a una determinada clase \mathbf{y} . Usa en enfoque de probabilidad *a posteriori*, usando la regla de Bayes:

$$P(y|x) = \frac{P(y)P(x|y)}{P(x)} \quad (1.8)$$

En el cálculo es usado Expectation Maximization (EM) sobre prototipos Gaussianos, obtenidó de los datos de entrenamiento, con el objetivo de garantizar un Maximum a Posteriori (MAP) para luego clasificar el vector de características.

Basado en este clasificador, tenemos Bayes Quadratic, aplicado con éxito para Motor Imagery; Hidden Markov Model(HMM) que para BMI usa probabilidades de un Gaussian Mixture Model(GMM), logrando clasificar incluso con data cruda de EEG y la versión de HMM para multiclase Input Output Hidden Markov Model (IOHMM).

A pesar de que en BMI no sean ampliamente usados; en un estudio comparativo, han resultado con un porcentaje de éxito aceptable superando a otros clasificadores más conocidos [53].

1.5.5. Redes neuronales artificiales

Las redes neuronales artificiales (Artificial Neural Networks, ANN) son clasificadores no lineales inspirados en cómo el cerebro procesa la información. Se caracterizan por ser flexibles y aplicables en problemas multiclase. En conjunto con los clasificadores lineales son los clasificadores más usados en BMI. Una red neuronal, es considerada como un ensamble de muchas neuronas artificiales con capacidad de decisión no lineal [54]. La facultad de aprender a través de ejemplos es una de las más importantes propiedades de una red neuronal [23].

Un ANN comprende un conjunto de nodos y conexiones que se modifican durante el proceso de entrenamiento. Si una predicción es incorrecta internamente se ajusta los pesos para minimizar la diferencia; mientras que si ésta es correcta, no los modifica.

Una de las redes neuronales más usadas es el MultiLayer Perceptron(MLP). Usando DWT y MLP se puede clasificar si una señal EEG es normal o de epilepsia con gran precisión [55].

1.6. Sistemas embebidos

Un sistema embebido, es un sistema de computación orientado al uso en funciones específicas. Existen diversos sistemas embebidos basados en microcontroladores, FPGA, PLD. Sin embargo, el estudio de interés se centra en las microcomputadoras debido a su gran versatilidad, capacidad de computo y bajo coste.

1.6.1. Microcomputadoras

Una microcomputadora, Single Board Computer (SBC) en inglés, es un sistema embebido multiuso que corre bajo sistemas operativos como Android, Linux Debian, Linux Ubuntu, Windows 10, entre otros. Posee diversos periféricos tales como USB 2.0, USB 3.0, puerto GPIO, Ethernet, HDMI, Bluetooth, IR, micrófono, entre otros. Estas plataformas se basan en un System On Chip (SoC), generalmente de arquitectura ARM, se encuentran actualmente en prestaciones de 1 a 8 núcleos. A diferencia con las PC de escritorio, estas consumen mucho menor cantidad de energía, tienen

tamaño reducido, son de bajo costo y tienen un buen desempeño en aplicaciones específicas. Gracias a su sistema operativo es la plataforma embebida con más versatilidad. [44]

Odroid-xu4

Odroid-xu4 (Figura 1.8) es una *32-bits Single Board Computer (SBC)* de bajo costo y alto desempeño realizada en Corea del Sur, disponible desde febrero de 2015. Este dispositivo se caracteriza por un procesador *octa-core Exynos 5422 big.LITTLE* de arquitectura ARM basado en procesadores ARM-Cortex (4xCortexTM-A15 2Ghz y 4xCortexTM-A7), una unidad de procesador gráfico (GPU por sus siglas en inglés) Mali-T628, una memoria RAM LPDDR3 2Gbyte .

Este sistema embebido posee los siguientes puertos: dos USB 3.0, un USB 2.0, un 10/100/1000Mbps RJ45 Ethernet y el *General Porpouse Input-Output (GPIO)*: 30 GPIO/IRQ/SPI/ADC pines y 12 GPIO/I2S/I2C pines a 1.8v. La SBC se alimenta con una fuente de 5V/4A DC. Corre con diferentes sistemas operativos tales como Ubuntu 15.04, Android 4.4 KitKat o 5.0 Lollipop. Tiene la ventaja que a diferencia de otras SBC es compatible con un eMMC, el cual es módulo de almacenamiento con un tiempo de acceso 3 veces más rápido que una SD Card de clase 10.



Figura 1.8: SBC Odroid-xu4 [6].

A su vez los sistemas embebidos de la compañía Odroid HardKernel son ampliamente usados en el área de investigación y cuentan con una revista donde se publica mensualmente diversas aplicaciones [56].

Raspberry Pi 3

Raspberry Pi 3 (Figura 1.9) es una microcomputadora ampliamente usada para prototipos de bajo costo. Fue desarrollada en Reino Unido con la intención de enseñar electrónica y programación en las escuelas. Ha sido aplicado para el control de robot móviles [57], en internet de las cosas (IoT) [58], reconocimiento de imágenes faciales [59] y otros. Posee un procesador Quad Core, ARM A7, basado en un SoC Broadcom BCM2836 y un procesador gráfico Broadcom VideoCore IV.

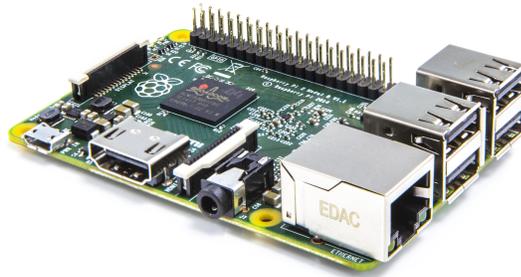


Figura 1.9: SBC Raspberry Pi 3 [7].

Banana Pi M3

Banana Pi M3 (Figure 1.10) es un dispositivo de \$70 compatible con Android y diversas distribuciones de Linux. Esta microcomputadora posee un procesador de Allwinner 83T compuesto por 8 nucleos ARM Cortex A7 y un GPU PowerVR SGX544MP1. Así mismo usa un RAM de 2GB compartida entre el CPU y el GPU. Adicionalmente cuenta con 40 pines GPIO, un módulo WiFi, un modulo Bluetooth, 2 puertos USB 2.0, un puerto ethernet 10/100/1000Mbps y salida HDMI.



Figura 1.10: SBC Banana Pi M3 [8].

Capítulo 2:

Diseño y configuración experimental

En el presente capítulo se describe el proyecto, las etapas que comprende y los equipos a ser empleados en la fase de adquisición, procesamiento y aplicación, los cuales serán usados a lo largo de la tesis.

2.1. Descripción del proyecto

El proyecto contempla la elaboración de una interfaz cerebro máquina usando un sistema de bajo costo orientado a la asistencia de pacientes con discapacidad motora. Por ello, se decide emplear el **paradigma de imaginación motora** (MI, por sus siglas en inglés). En consecuencia, se trabaja con una base de datos de MI que contenga registro de cuatro diferentes clases de movimiento.

Con el objetivo de realizar esta BMI y evaluar la factibilidad del uso de un sistema embebido como sistema de procesamiento, se compara con una PC diversos aspectos del desempeño. Para evaluar este desempeño, se consideran la siguientes etapas en el proyecto:

- Etapa 1: Entrenamiento del algoritmo.
- Etapa 2: Sistema online.

En la aplicación para la interfaz cerebro-máquina, se usa la prótesis de brazo Inmoov. Los cuatro comandos asociados a cada imaginación de movimiento en la base de datos son usados para controlar este equipo.

2.1.1. Etapa 1: Entrenamiento

Como se vio en el estado del arte, una BMI está compuesta por las fases de adquisición de señales EEG, procesamiento y aplicación. Para una BMI con un adecuada

tasa de acierto y tiempo de procesamiento, es necesario evaluar diferentes configuraciones en la fase del procesamiento.

La etapa de entrenamiento tiene como objetivo encontrar una adecuada configuración de los parámetros de un conjunto de algoritmos con que comprenden un método y así entrenar el clasificador. Para ello, esta etapa sigue la metodología propuesta en la Figura 2.1. Se usa una base de datos de señales EEG como entrada, se realiza el procesamiento en una laptop y en un sistema embebido; finalmente se envían los comandos para controlar la aplicación.

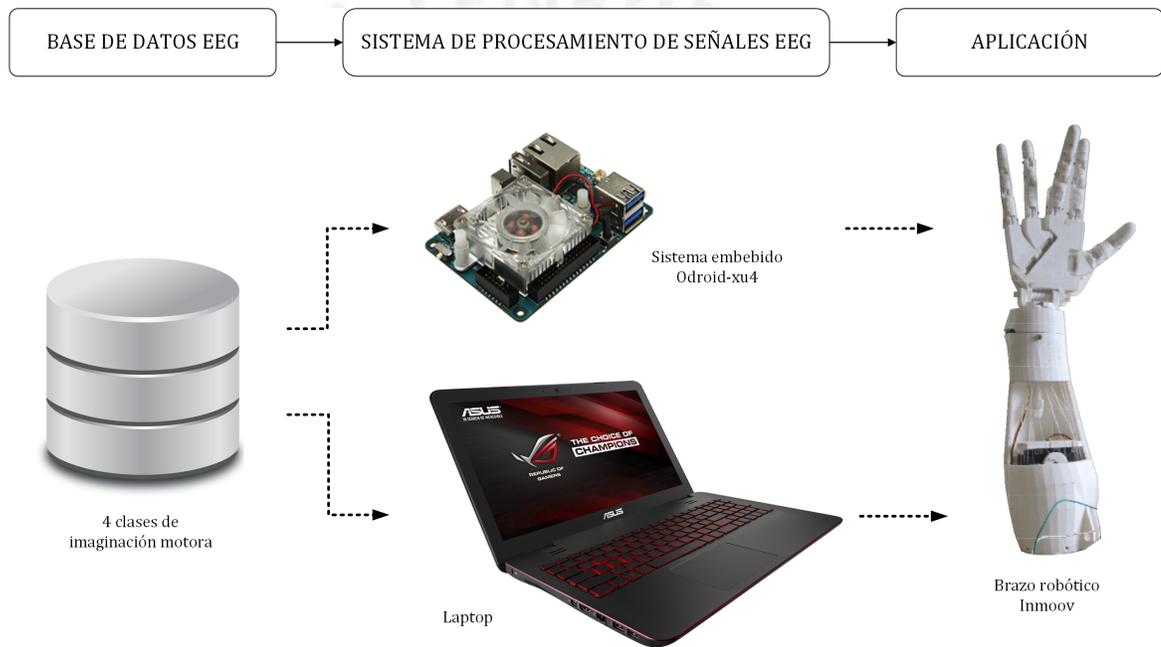


Figura 2.1: Etapa de entrenamiento.

Fuente: Diagrama Elaboración Propia. Imágenes [10], [6]

El conjunto de parámetros en el procesamiento y clasificación son probados en ambos equipos de procesamiento con el fin de evaluar el desempeño del sistema embebido a comparación de la laptop. Se escoge el conjunto de algoritmos que muestran una alta tasa de acierto y bajo tiempo de procesamiento, el cual será usado en la siguiente etapa.

2.1.2. Etapa 2: Sistema online

La segunda etapa se caracteriza por el uso de señales EEG en tiempo real. A este sistema se le denomina *online* y se efectúa con los algoritmos y parámetros definidos en la etapa de entrenamiento.

En la Figura 2.2 se muestra las fases que contemplan esta etapa. En la fase de adquisición se usa las señales EEG de la misma base de datos que en la etapa uno, se concatenan y son transmitidas de forma inalámbrica por medio de un equipo emulador de señales cerebrales, efectuandose así una simulación de *Hardware in the loop* para comprobación del sistemas en tiempo real.

La fase del procesamiento es realizada completamente por el sistema embebido Odroid-xu4, el cual procesa la señal, extrae las características y las clasifica; finalmente se envía un comando que controla movimiento de la prótesis de brazo Inmoov en la fase de aplicación.

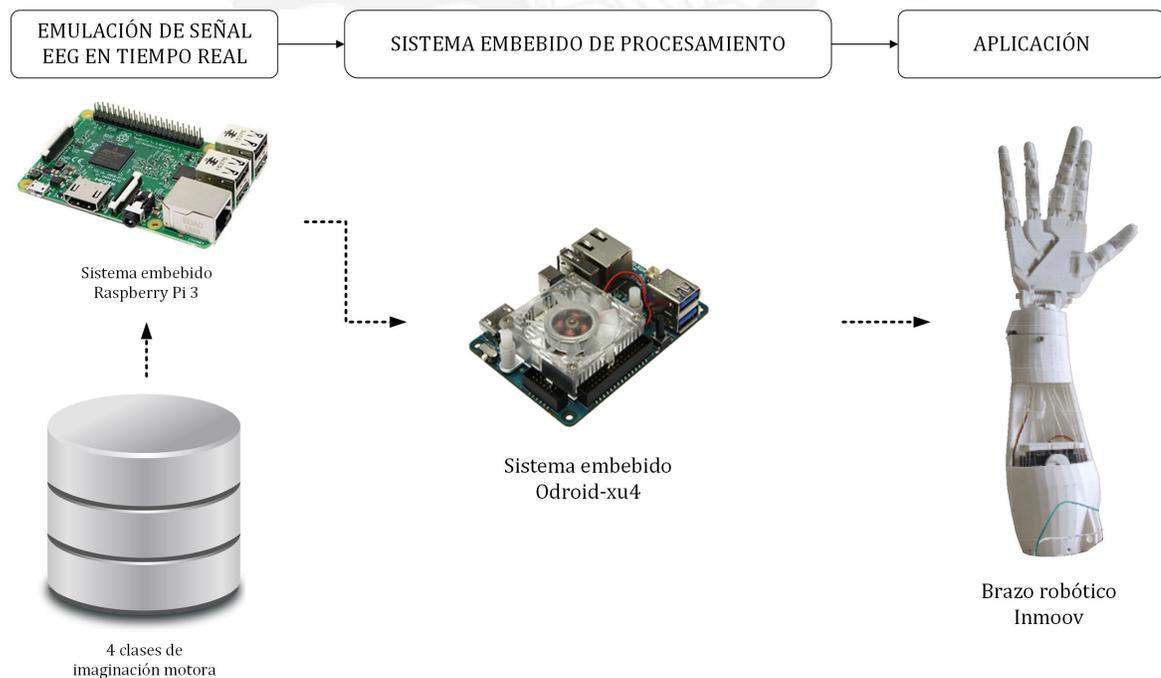


Figura 2.2: Etapa online.

Fuente: Diagrama Elaboración Propia. Imagenes [7], [6]

2.2. Adquisición de señales EEG

En las etapas uno y dos del proyecto, se hace uso de una base de datos en la fase de adquisición de señales EEG. A su vez, en la segunda etapa se usa una microcomputadora Raspberry Pi 3 para transmitir estas señales EEG y efectuar un *Hardware in the loop*, emulando el envío de señales en tiempo real. Ambos tópicos (base de datos y equipo emulador de señales EEG) son tratados a mayor detalle en esta sección.

2.2.1. Base de datos

En la presente tesis se usa la base de datos de imaginación motora multiclase proporcionada por el Laboratorio de interfaces cerebro-máquina (BCI-Lab) de la Universidad Tecnológica de Graz, Austria, publicada en la competencia BCI 2005, bajo el nombre de *Data sets IIIa: (motor imagery, multi-class)*, disponible en [60].

El paradigma usado se basa en la imaginación del movimiento de:

- Mano izquierda.
- Mano derecha.
- Pies.
- Lengua.

La base de datos contiene el registro de señales EEG, adquiridas con el amplificador EEG Neuroscan. Las principales características de la base de datos se encuentran en la Tabla 2.1.

Tabla 2.1: Características de la base de datos.

Característica	Descripción
Número de electrodos EEG	60.
Frecuencia de muestreo	250 Hz.
Posición del electrodo de referencia	Mastoideo izquierdo.
Posición del electrodo de tierra	Mastoideo derecho.
Filtro pasabanda	1 - 50Hz.
Filtro Notch	50Hz.

Fuente: Elaboración Propia

El esquema de posicionamiento y numeración de los electrodos se muestra en la Figura 2.3.

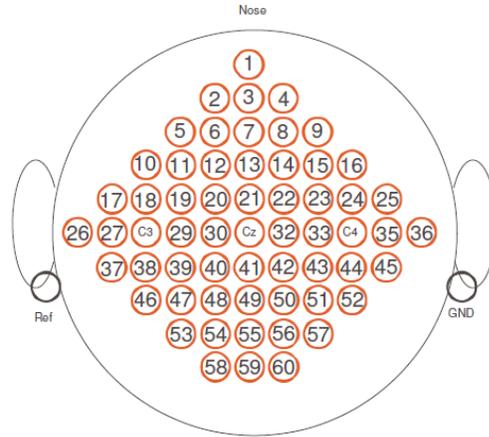


Figura 2.3: Posición de los electrodos de la base de datos [9]

El protocolo para el registro de datos consiste en series de 40 ensayos, cada ensayo dura 7 segundos. Se muestran 10 veces y de forma aleatoria cada una de las 4 clases en cada serie.

Para el registro de las señales EEG se sigue el diagrama de tiempo que se muestra en la Figura 2.4. Los dos primeros segundos del ensayo se muestra una pantalla blanca. En el segundo 2, un estímulo acústico indica el comienzo de la prueba; a su vez, aparece en el centro de la pantalla una cruz fija que el paciente debe observar. A continuación, desde el tercer segundo hasta el cuarto, una flecha apuntando hacia la izquierda, derecha, arriba o abajo es mostrada, el sujeto debe imaginar un movimiento de la mano izquierda, la mano derecha, la lengua o el pie respectivamente, esto se muestra en la Figura 2.5. La imaginación del movimiento debe continuar hasta que la cruz desaparezca en el segundo 7.

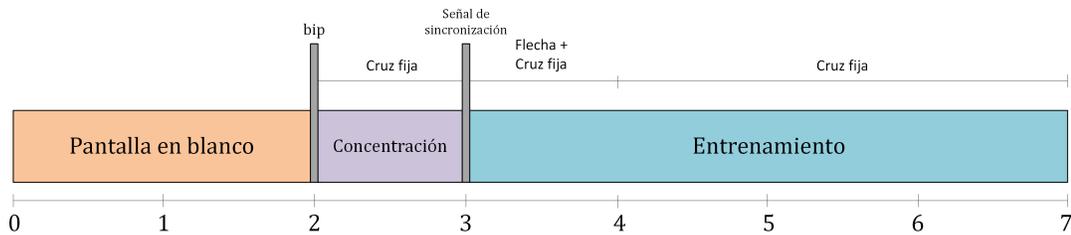


Figura 2.4: Temporización del paradigma.



Figura 2.5: Señales del protocolo asociadas a imaginación motora.

La interpretación del paradigma usado para el registro de señales EEG visto desde la pantalla del paciente se muestra en la Figura 2.6.

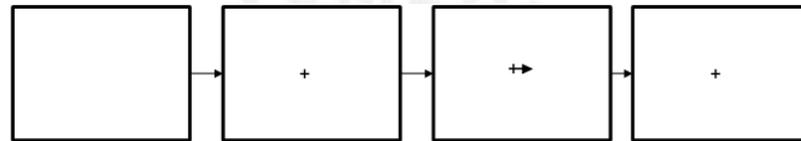


Figura 2.6: Secuencia del paradigma visualizado desde la pantalla.

2.2.2. Emulador de señales EEG (*Hardware in the loop*)

Se emplea un sistema embebido para emular la transmisión de señales EEG. El objetivo de este dispositivo es el envío de las señales EEG en tiempo real para la etapa *Online* del sistema.

La microcomputadora a seleccionar requiere cumplir con las siguiente especificaciones:

- Conexión tipo TCP/IP para la transmisión de datos.
- Almacenamiento de 450MBytes para la base de datos.
- Velocidad de transmisión de 4mseg. ($f=250\text{Hz}$).
- Portabilidad para el uso en diversos ambientes de trabajo.

Con estos requerimientos se escoge el sistema embebido Raspberry Pi 3. Se considera a su vez el bajo costo, disponibilidad y desempeño para esta aplicación específica. Se usa el sistema operativo Raspbian Jessie que se basa en distribución de Linux Debian Jessie disponible en la página oficial de Raspberry Pi [61] instalado en una tarjeta SD de clase 10 de 8Gb.

2.3. Equipos para el procesamiento

Se describen los componentes que forman parte del diseño en la fase de procesamiento de la interfaz cerebro-máquina propuesta en la presente tesis y se resalta el uso de un sistema embebido y una laptop:

- Sistema embebido para realizar el procesamiento y clasificación.
- Laptop para comparar y evaluar el desempeño del sistema embebido como procesador y clasificador.

Según la revisión del estado del arte se realiza la Tabla 2.2. En esta tabla se muestra un resumen de las características de los más destacados sistemas embebidos y posibles candidatos para la fase de procesamiento en la BMI.

Tabla 2.2: Comparación entre los principales sistemas embebidos.

	Odroid-xu4	BananaPi M3	CubieBoard C80	RaspberryPi 3
SoC	Samsung Exynos-5422	Allwinner 83T	Allwinner A80	Broadcom BCM2837
Arquitectura	ARM Cortex A15x4/A7x4	ARM Cortex A7	ARM Cortex A15x4/A7x4	ARM Cortex A53
Núcleos	8	8	8	4
Frecuencia	2.1GHz	1.8GHz	2.1GHz	1.2GHz
RAM	2Gb	2Gb	2Gb	1Gb
Tamaño(cm)	8.2 x 5.8 x 2.2	9.2 x 6 x 2.2	14.6 x 14.2 x 1.8	8.6 x 5.4 x 2.2
Costo	\$ 74	\$ 70	\$ 124	\$ 35

Fuente: Elaboración Propia

2.3.1. SBC procesador de señales EEG

Es el dispositivo encargado de la recepción, procesamiento y clasificación de señales EEG, por lo que es el más importante en el marco del desarrollo de la tesis. En este dispositivo se implementa y evalúan los algoritmos relacionado a BMI por lo que debe cumplir los siguientes requerimientos:

- Alta capacidad de procesamiento.
- Dimensiones pequeñas que garanticen la portabilidad del equipo.
- Conexión tipo TCP/IP para la recepción de datos.
- Bajo costo.
- Documentación publicada de aplicaciones relacionadas a procesamiento.

Después de analizar diferentes aspectos de las SBC se decide adquirir el sistema embebido Odroid-xu4. En este sentido y para el desarrollo de la tesis se escoge la distribución Linux Ubuntu 16.04 liberada para Odroid-xu3/xu4 en julio 2016 [62]. Este sistema operativo ha sido instalado en una tarjeta embebida MultiMediaCard (eMMC) de 16Gb.

2.3.2. Laptop basada en Linux

Con el fin de comparar y a su vez evaluar el desempeño de la SBC que procesa las señales EEG, se requiere de una laptop orientada a realizar las mismas tareas, debido a que un típico BMI contempla el uso de este equipo para el procesamiento de EEG.

Las características técnicas de la laptop usada en las pruebas experimentales se muestran en la Tabla 2.3.

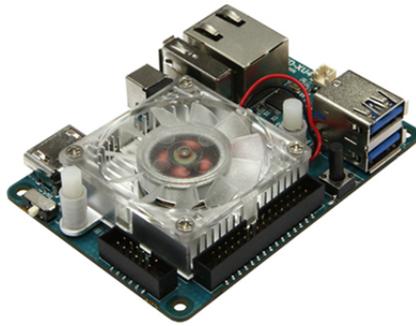
Tabla 2.3: Características de la Laptop Asus.

Características	Descripción
Marca	Asus
Procesador	Intel Core i7
Frecuencia de trabajo	3.4GHz
RAM	8GBytes
Disco duro	2Tb
Precio	Alrededor de \$1500.

Fuente: Elaboración Propia

La laptop corre bajo el sistema operativo Linux Ubuntu 16.04 y es empleada debido a la disponibilidad y confiabilidad de desempeño para el procesamiento.

En la Figura 2.7 se muestran los dos equipos a usarse a lo largo de la tesis en la fase de procesamiento de señales EEG: el SBC procesador de señales EEG y la Laptop basada en Linux. En la Tabla 2.4 se muestra las principales características de estos equipos, ambos serán evaluados a lo largo de la tesis para comparar el desempeño y factibilidad del uso de una SBC en la fase de procesamiento de un BMI.



(a) SBC Odroid-xu4



(b) Laptop

Figura 2.7: Equipos a comparar en el procesamiento de señales EEG [10], [6].

Tabla 2.4: Comparación entre los equipos de procesamiento.

SBC Odroid-xu4	Laptop Asus
S.O. Linux Ubuntu	S.O. Linux Ubuntu
8 core procesador a 2.1GHz	Core i7 Procesador a 3.4 GHz
2 GB RAM	16 GB RAM
\$ 74	\$ 1500
60 g.	2.7 Kg.

Fuente: Elaboración Propia

2.4. Aplicación

Como se aprecia en el estado del arte, existen diversos dispositivos externos que son controlados en una BMI. Tales dispositivos están a la espera de los comandos enviados al predecir un estado de MI. En la presente tesis se hace uso del brazo robótico Inmoov.

Inmoov [11] (Figura 2.8) es un robot humanoide construido a partir de filamentos de impresión en 3D y controlado por microcontroladores Arduino. Este proyecto de código abierto fue iniciado en enero de 2012 por Gael Langevin, un escultor y diseñador francés. El robot se ha utilizado como una plataforma para la enseñanza de las universidades, laboratorios, aficionados y fabricantes.

Para el propósito de esta investigación, el brazo del robot fue impreso en 3D. Seis servomotores (cinco para el control de cada dedo y uno para la rotación de la muñeca) comandados por un Arduino permiten el control de múltiples gestos. Los

gestos que el brazo robótico Inmoov realiza respecto a cada tarea de imaginación motora se muestra en la Tabla 2.5.

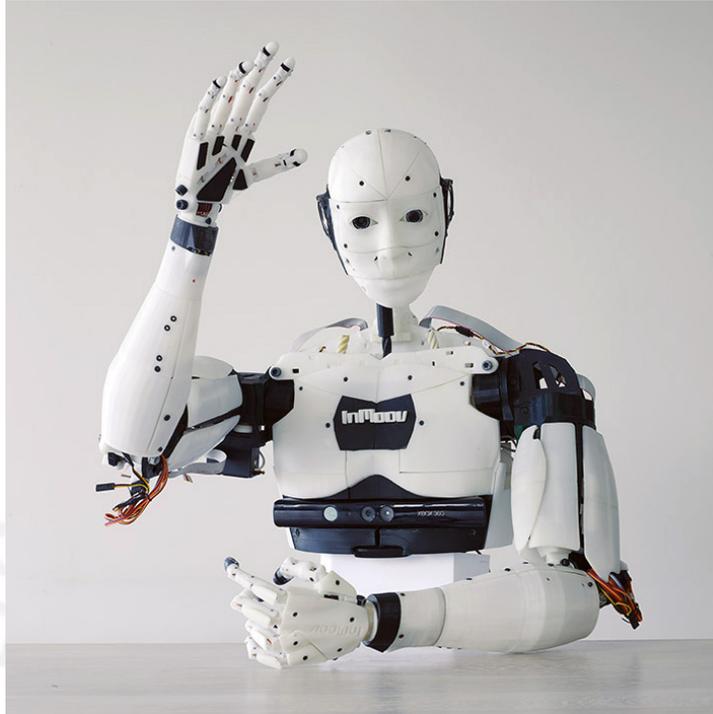


Figura 2.8: Robot Inmoov [11].

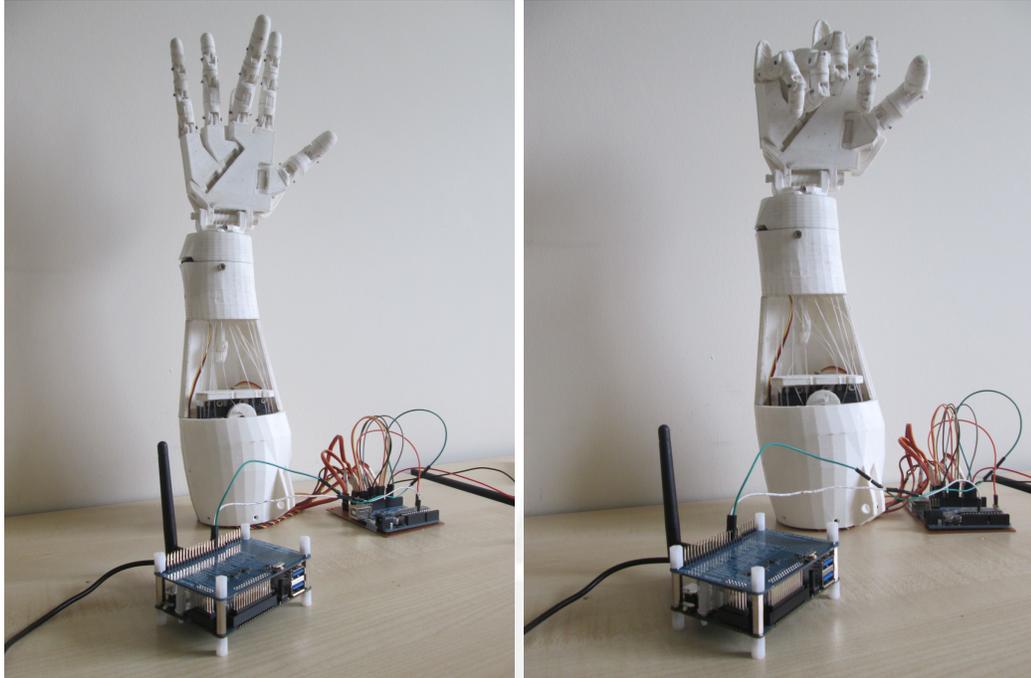
Tabla 2.5: Gestos asociados a tareas de imaginación motora.

Imaginación motora	Comando	Gesto
Mano izquierda	1	Rota la muñeca 90° en sentido horario
Mano derecha	2	Rota la muñeca 90° en sentido antihorario
Pies	3	Gesto de cerrar la mano
Lengua	4	Gesto de abrir la mano

Fuente: Elaboración Propia

En la Figura 2.9 se muestra dos gestos ejecutados por el brazo robótico Inmoov cuando el sistema embebido Odroid-xu4 detecta el estado de imaginación motora de lengua y pies.

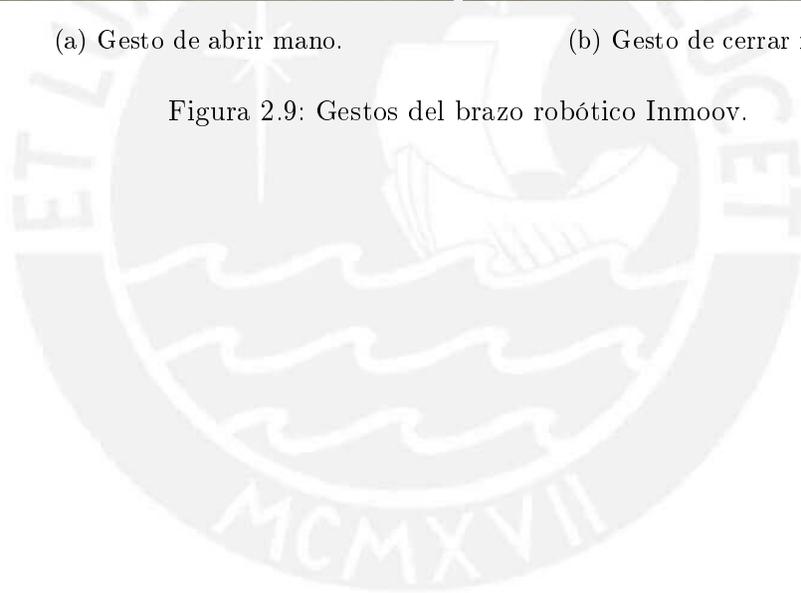
El enfoque de la presente tesis está relacionado con el procesamiento de las señales EEG usando el sistema embebido Odroid-xu4, por lo que la implementación paso a paso del brazo robótico Inmoov es visto solo como potencial aplicación.



(a) Gesto de abrir mano.

(b) Gesto de cerrar mano.

Figura 2.9: Gestos del brazo robótico Inmoov.



Capítulo 3:

Implementación de la interfaz cerebro-máquina

En el presente capítulo se describe el desarrollo e implementación de las dos etapas de la interfaz cerebro máquina propuestos. En la primera etapa se detallan los algoritmos que se usan para el procesamiento de la señal EEG para el entrenamiento. En la segunda etapa se muestran los procedimientos para llevar las señales a tiempo real y las consideraciones que se tienen que hacer en los algoritmos.

3.1. Etapa 1: Entrenamiento

En la etapa de entrenamiento se proponen dos métodos. Cada método comprende los algoritmos para la adquisición, pre-procesamiento, extracción de características, clasificación y validación. La idea es implementar estos dos métodos en los dos equipos de procesamiento (sistema embebido Odroid-xu4 y laptop) y evaluar el desempeño de cada sistema. A su vez se modifican los parámetros para ajustar cada algoritmo teniendo como criterio la tasa de acierto que proporciona la validación.

El método 1 se muestra en la Figura 3.1. Este método tiene como entrada la señal EEG (S_{EEG}), la cual pasa por un primer filtro espacial que efectúa una re-referencia y luego por un filtro pasabanda. Los datos son procesados por un algoritmo de Patrones Espaciales Comunes (CSP, por sus siglas en inglés). Finalmente se extrae las principales características de las señales usando el logaritmo de la varianza. Estas características componen un vector que entrena al clasificador SVM. La tasa de acierto es evaluado usando 10-veces validación cruzada.

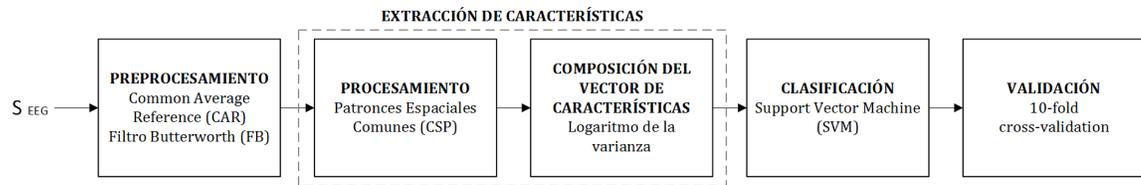


Figura 3.1: Método 1 en la etapa de entrenamiento.

El método 2 se muestra en la figura 3.2. La principal diferencia con el método 1 se encuentra en el algoritmo de procesamiento y extracción de características. Para procesar la señal usa la transformada Wavelet (WT), la cual genera señales de descomposición y una principal, usando estas señales se calcula la potencia y energía que componen el vector de características.

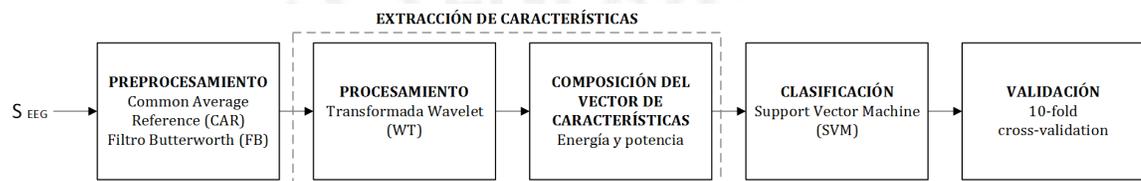


Figura 3.2: Método 2 en la etapa de entrenamiento.

A continuación se detalla la implementación de los algoritmos correspondientes a cada método.

3.1.1. Análisis de datos y pre-procesamiento

Conjunto de datos EEG

Las características de la base de datos de señales EEG elegida se encuentran detalladas en la Sección 2.2.1.

Si bien es cierto que se cuenta con información de 60 electrodos en la base de datos, no es necesario el uso de todos estos para realizar una BMI. Típicamente los sistemas de adquisición de bajo costo contienen una menor cantidad de canales; por lo tanto se considera evaluar el desempeño del sistema utilizando 8, 16 y 32 de los 60 canales.

La actividad relacionada a la imaginación motora es prominente en el área sensorimotora ubicada en la parte central del cerebro, debido a esto se escogen los

electrodos alrededor de esta área. En la Figura 3.3 se muestran las 4 configuraciones que se proponen y evalúan en la tesis. Las posiciones de los electrodos para cada configuración se representan por el número de electrodos.

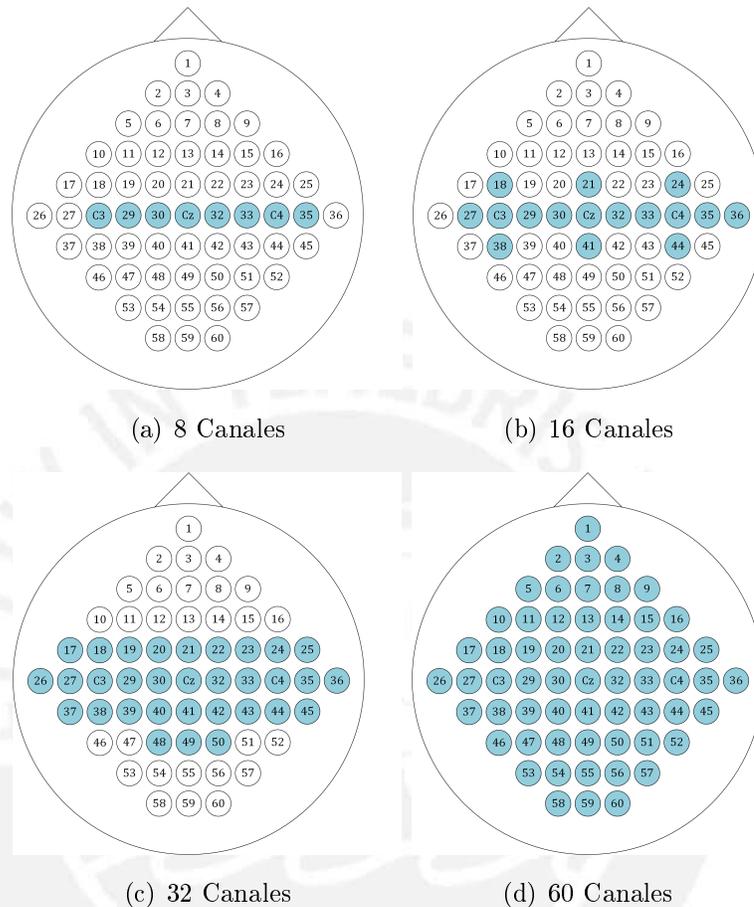


Figura 3.3: Configuraciones dependiendo del número de electrodos a evaluar.

Re-referencia

La señal EEG de los electrodos registrada en la base de datos usa como referencia el electrodo posicionado en el mastoideo izquierdo (A1), esto quiere decir que las señales captadas por cada electrodo se restan con el de referencia y se registra este valor. Sin embargo, este método no siempre es el más óptimo, por lo que en diversos casos se modifica la referencia a usarse. A este proceso se le denomina re-referencia y tiene por objetivo mejorar la calidad de la señal.

Common Average Reference (CAR) es un filtro espacial usado para realizar la re-referencia entre cada electrodo y el valor promedio en ese instante de tiempo,

mejorando así la actividad local de cada canal. La fórmula que enmarca el concepto matemático de CAR se muestra en la ecuación 3.1 y es tomada de Rao [63].

$$e\tilde{e}g_i = eeg_i - \frac{1}{N} \sum_{i=1}^N eeg_i \quad (3.1)$$

Para la implementación en el sistema embebido, se procede a realizar un análisis matemático con el fin de encontrar una notación matricial, debido a que los procesos iterativos son de alto costo computacional mientras que las operaciones matriciales están optimizadas para ser ejecutadas en un menor tiempo. Para ello, se define la matriz E de la señal EEG tal que $E \in \mathbb{R}^{ch \times T}$ donde ch es el número de canales y T es el número de muestras. Remplazando y evaluando E para la ecuación 3.1 obtenemos:

$$\begin{pmatrix} \tilde{E}_1 = E_1 - \frac{1}{N_{ch}}(E_1 + E_2 + \dots + E_{ch}) \\ \tilde{E}_2 = E_2 - \frac{1}{N_{ch}}(E_1 + E_2 + \dots + E_{ch}) \\ \vdots \\ \tilde{E}_{ch} = E_{ch} - \frac{1}{N_{ch}}(E_1 + E_2 + \dots + E_{ch}) \end{pmatrix} \quad (3.2)$$

La ecuación 3.2 puede ser escrita como:

$$\begin{pmatrix} \tilde{E}_1 \\ \tilde{E}_2 \\ \vdots \\ \tilde{E}_{ch} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_{ch} \end{pmatrix} - \frac{1}{N_{ch}} \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & & & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & & & 1 & 1 \\ 1 & 1 & \dots & 1 & 1 \end{pmatrix} \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_{ch} \end{pmatrix} \quad (3.3)$$

Simplificando la notación de la ecuación 3.3 se obtiene:

$$E_{ch \times T}^{\tilde{}} = I_{ch \times ch} E_{ch \times T} - \frac{1}{N_{ch}} O_{ch \times ch} E_{ch \times T} \quad (3.4)$$

Donde $I_{ch \times ch}$ es la matriz identidad y $O_{ch \times ch}$ es una matriz cuadrada de unos. Finalmente agrupando términos se escribe la ecuación 3.4 como:

$$\tilde{E} = (I - \frac{1}{N} O) E \quad (3.5)$$

Se define la matriz de transformación S que realiza el filtro espacial denominado

Common Average Reference (CAR) tal que $S \in \mathbb{R}^{ch \times ch}$ y se calcula como:

$$S = I - \frac{1}{N}O \quad (3.6)$$

Finalmente para realizar el proceso de re-referencia se tiene que aplicar el filtro espacial CAR a la señal cerebral E obteniendo una señal re-referenciada denominada \tilde{E} y se calcula como:

$$\tilde{E} = S * E \quad (3.7)$$

El procedimiento es traducido a un diagrama de flujo, que contempla como entrada a la matriz E , que contiene la señal de los electrodos y como salida a la matriz E_r , que contiene la señal de los electrodos re-referenciada. El algoritmo se implementa en el sistema embebido y la laptop basándose en el planteamiento matemático y el diagrama de flujo que se muestra en la Figura 3.4.

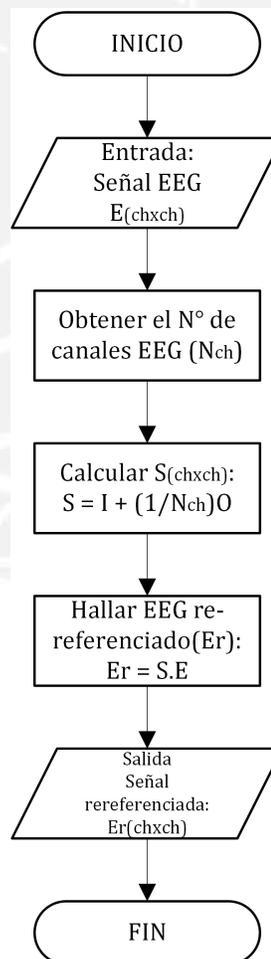


Figura 3.4: Diagrama de flujo de re-referencia CAR.

En la Figura 3.5 se muestra la señal EEG relativa a los canales C3, Cz y C4 de un ensayo de imaginación motora de mano izquierda. La gráfica de cada canal antes de usar el filtro CAR es de color azul, mientras que la señal de salida después de usar la re-referencia CAR es graficada de color verde.

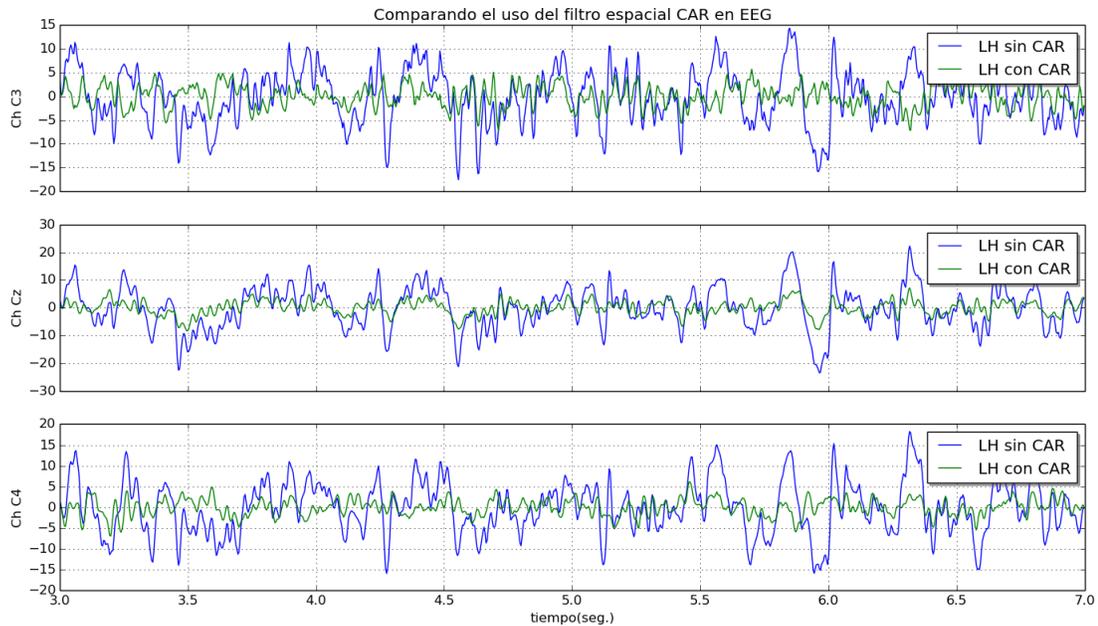


Figura 3.5: Efecto del filtro espacial CAR en el dominio del tiempo.

Filtro en frecuencia

El análisis de las señales EEG en el dominio de la frecuencia es importante para una óptima extracción de características que permitan diferenciar la imaginación motora de cada clase. En ese sentido, los ritmos asociados a MI que presentan mayores cambios en amplitud y potencia se encuentran en las bandas μ (8-12 Hz) y β (13-30 Hz), y se muestran de forma prominente en las áreas premotora y sensorimotora primaria del cerebro.

Por tanto se busca atenuar las bandas que no sean de interés haciendo uso de un filtro pasabanda. Debido a su amplio uso en señales EEG, se propone el uso de un filtro IIR: Butterworth. Este filtro tiene una respuesta en frecuencia que depende del orden del filtro. Estas se comparan en la Figura 3.6.

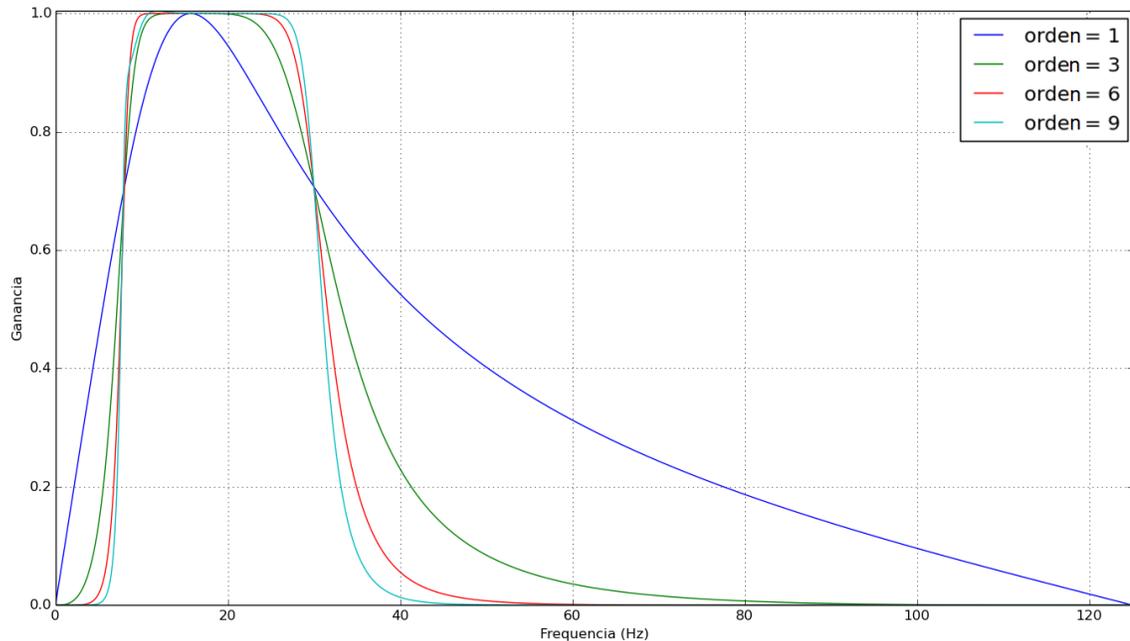


Figura 3.6: Comportamiento en frecuencia del filtro Butterworth (entre 8-30Hz) dependiendo del orden.

Como se observa en la Figura 3.6, utilizando un Filtro Butterworth (FB) de orden 3, se obtiene una respuesta en frecuencia adecuada con un menor orden, el cual se ve reflejado en un menor costo computacional. Por lo tanto, proponemos que para esta fase de pre-procesamiento se utilice un filtro pasabanda Butterworth de tercer orden de 8 a 30 Hz.

La implementación de este filtro en el sistema embebido y en la laptop se realiza basándose en la librería de procesamiento de señales “signal” que pertenece al paquete “scipy” de python. Se usan las funciones *butter* y *lfilter* de este paquete para implementar el filtro con los requerimientos solicitados.

En la Figura 3.7 se muestra el efecto del filtro BT en el dominio de la frecuencia. Para ello se usa la señal EEG después de la re-referencia CAR de los canales C3, Cz y C4 relativos a un ensayo usando imaginación motora de la mano izquierda. Se grafica la respuesta en frecuencia antes y después del filtro BT. Como se observa en la gráfica de color verde, las frecuencia fuera del rango deseado (8-30Hz) se atenúan y eliminan.

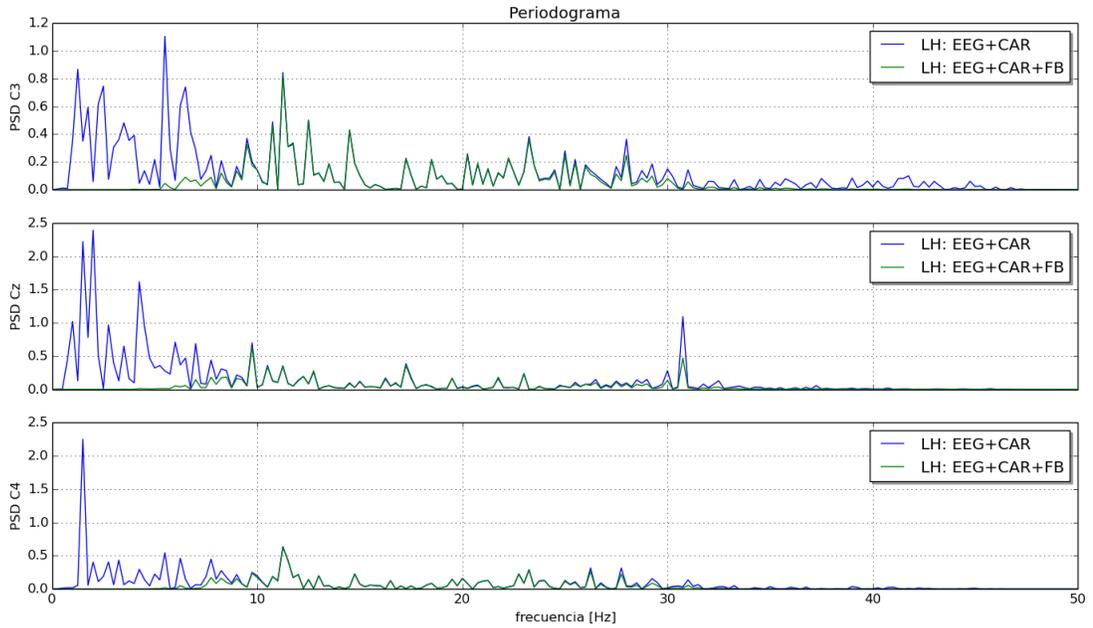


Figura 3.7: Efecto del filtro BT en el dominio de la frecuencia.

Del mismo modo en la Figura 3.8 se muestra el efecto del filtro en estas señales en el dominio del tiempo. Como se puede apreciar la señal se ve suavizada debido a que se eliminan las frecuencias superiores a 30Hz.

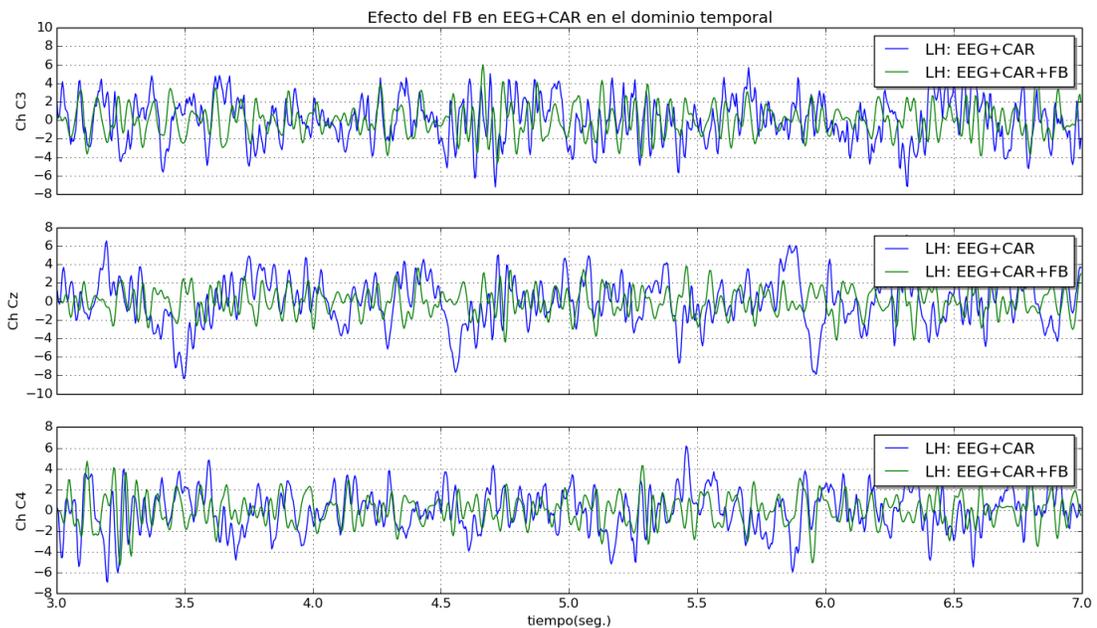


Figura 3.8: Efecto del filtro BT en el dominio de la tiempo.

3.1.2. Extracción de características - Método 1

La extracción de características es una etapa fundamental en el procesamiento de señales EEG. Esta permite recolectar la información más representativa de un conjunto de señales asociadas a un estado de MI usando pocos valores.

En la Figura 3.9 se resalta los algoritmos que se usan en esta etapa de extracción de características vistos en la propuesta del método 1.

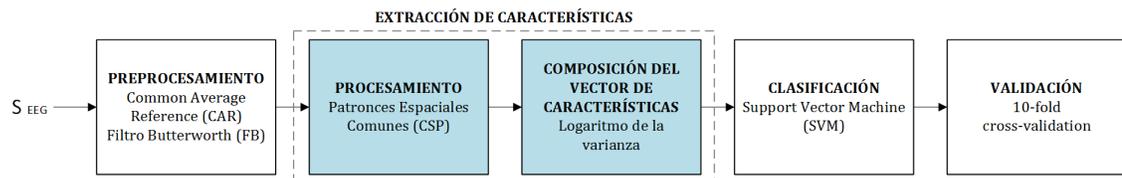


Figura 3.9: Extracción de características del método 1 en la etapa de entrenamiento.

Se procede a detallar el algoritmo de CSP y la composición del vector de características. Ambos algoritmos son orientados a la extracción de características del conjunto de señales relativas a cuatro clases de imaginación motora (movimiento de mano derecha, mano izquierda, pies y lengua) que se encuentran en la base de datos.

Patrones espaciales comunes (CSP)

El algoritmo CSP se basa en la elección de las columnas con las mayores y menores varianzas después de la transformación espacial. Es ampliamente usado en el área de BMI. Busca la máxima separación entre clases.

Este método fue propuesto para la separación de dos clases [64] [65]; sin embargo, diversos enfoques han sido propuestos como la extensión de este método para n-clases. Una de las extensiones del algoritmo CSP que considera múltiples clases es *One versus the Rest Common Spatial Pattern* (OVR-CSP). A continuación se detalla el proceso que se sigue y las ecuaciones necesarias para implementar el algoritmo expuesto en [66].

Se halla la matriz de covarianza para cada clase.

$$C_i = \frac{E_i E_i^T}{\text{trace}(E_i E_i^T)}, i = 1, 2, 3, 4 \quad (3.8)$$

La matriz de covarianza compuesta es la suma de las matrices de covarianza:

$$C_c = C_1 + C_2 + C_3 + C_3 \quad (3.9)$$

De modo similar a CSP 2-clases, la matriz C_c puede es escrita como:

$$C_c = U_c \lambda_c U_c^T \quad (3.10)$$

Siendo λ_c la matriz diagonal de autovalores y U_c la matriz de autovectores. Se define la variable P , para posteriormente ser usado en la transformación blanca:

$$P = \sqrt{\lambda_c^{-1}} U_c^T \quad (3.11)$$

A continuación se elige una clase C_n , el resto se suma para representar una nueva clase C'_n . Se calcula CSP de 2-clases entre C_n y C'_n . Este mismo procedimiento se realiza para cada una de las n-clases. En nuestro caso se realiza 4 veces (4 clases). Se detalla el caso para C_1 :

Se calcula C'_1 :

$$C'_1 = C_2 + C_3 + C_4 \quad (3.12)$$

Se halla S_1 y S'_1 como:

$$S_1 = PC_1P^T \quad y \quad S'_1 = PC'_1P^T \quad (3.13)$$

Cada una puede ser escrita como:

$$S_1 = B\lambda_1B^T \quad y \quad S'_1 = B\lambda'_1B^T \quad (3.14)$$

Donde B es el autovector y λ el autovalor de cada clase. Y cumple que $\lambda_1 + \lambda'_1 = I$.

La matriz W_1 se calcula como:

$$W_1 = B_1^T P \quad (3.15)$$

Finalmente se halla Z que contiene el conjunto de datos ordenados de forma óptima para la separación entre clases. Se elige las p-primeras columnas y las p-ultimas columnas del resultado.

$$Z_1 = W_1 E \quad (3.16)$$

La implementación de este algoritmo en el sistema embebido procesador (Odroid-

xu4) y la laptop se realiza en python basado en las ecuaciones planteadas. Se orienta esta programación al análisis de CSP para cuatro clases.

Una de las grandes ventajas del uso de CSP como método de extracción de características es la generación de los patrones espaciales, los cuales representan la esencia del comportamiento en la ejecución de alguna clase de imaginación de movimiento.

Usando los 60 canales de nuestra base de datos se calcula la matriz “W” generada por cada clase vs. el resto (OVR). La primera columna de cada uno de los cuatro “W” asociados a cada imaginación de movimiento es registrado y escalado de 0 a 100. Finalmente se asocia el valor de cada elemento de la primera columna a la intensidad producida en su posición correspondiente. Se realiza una interpolación lineal basada en estos valores y se obtiene un mapa de la actividad cerebral relativa a cada imaginación de movimiento. Esto se muestra en la Figura 3.10.

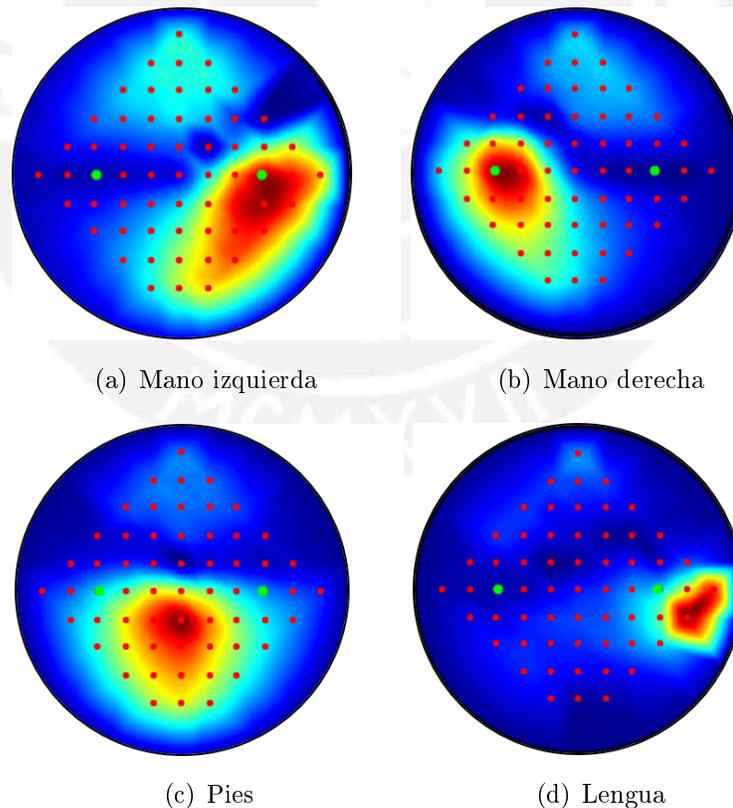


Figura 3.10: Patrones espaciales por cada clase de imaginación motora: mano izquierda, derecha, pies, lengua.

Composición del vector de características

Después de haber procesado la señal, se procede a componer el vector de características. El conjunto de estos se usa como entrada para entrenar el algoritmo de clasificación. El tamaño de este vector varía en este método dependiendo del número de CSP que escogemos.

Al emplear el algoritmo OVR-CSP se obtienen cuatro matrices (una por cada clase). Las matrices se denominan: Z_n tal que $Z_n \in \mathbb{R}^{2pxT}$, donde $2p$ es el número de patrones seleccionados del OVR-CSP (p -primeras y p -últimas columnas), T es la trama de señal escogida y n es la clase a la que pertenece. Estas matrices Z_n que corresponden a una trama T de señal EEG son representadas usando la normalización logarítmica de su varianza, la cual se calcula usando la siguiente ecuación:

$$f_i = \log\left(\frac{VAR_i}{\sum_{i=1}^4 VAR_i}\right), i = 1, 2, 3, 4 \quad (3.17)$$

Donde $VAR_i \in \mathbb{R}^{2px1}$ es varianza de cada fila de Z_i . Esto se realiza para cada clase obteniendo f_1, f_2, f_3 y f_4 donde $f_i \in \mathbb{R}^{2px1}$.

Finalmente la composición del vector de características se muestra en la siguiente ecuación:

$$F = [f_1, f_2, f_3, f_4] \quad (3.18)$$

Donde $F \in \mathbb{R}^{8px1}$. Esto quiere decir que en un vector de $8p$ valores se representa toda una trama T de N canales EEG. Este procedimiento se repite para cada ensayo etiquetado y se usa para entrenar el clasificador.

En la Figura 3.11 se muestra el diagrama de flujo de la extracción de características del método 1.

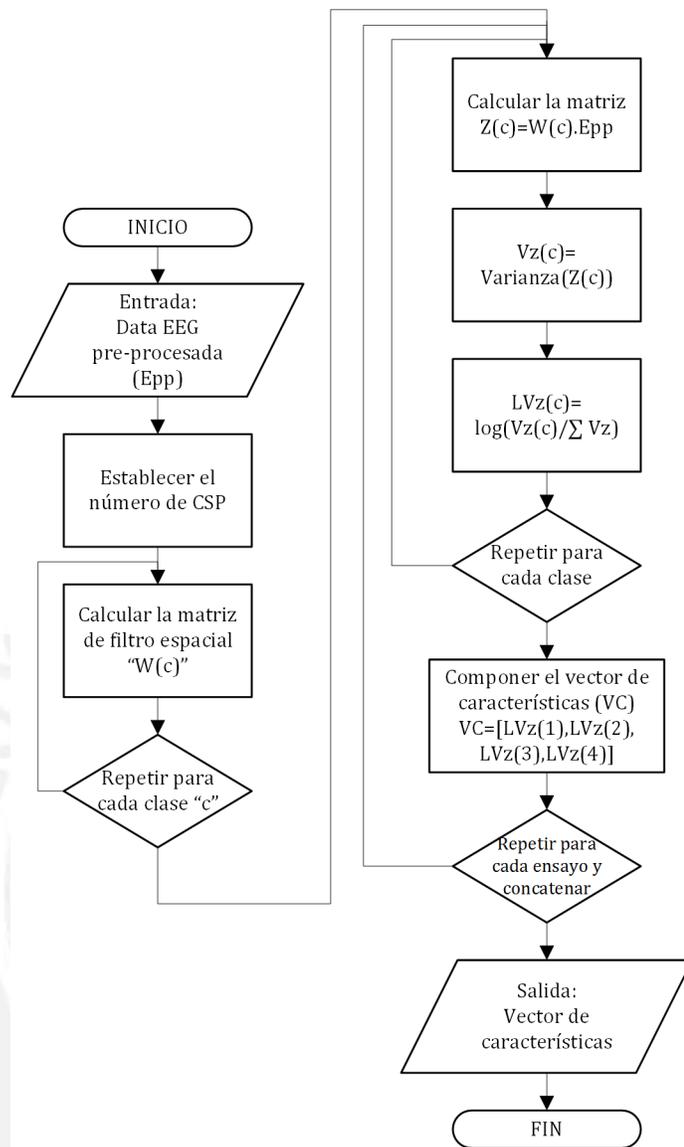


Figura 3.11: Diagrama de flujo de la extracción de características del método 1.

3.1.3. Extracción de características - Método 2

La extracción de características permite recolectar la información más representativa de un conjunto de señales asociadas a un estado de MI usando pocos valores que componen un vector de características.

Como ya se ha detallado, el uso del paradigma de imaginación motora se relaciona con la variación de energía en las bandas μ y β , estas bandas son analizadas usando la transformada Wavelet que caracteriza este segundo método.

En la Figura 3.12 se resalta los algoritmos que se usan la etapa de extracción de características vistos en la propuesta del método 2.

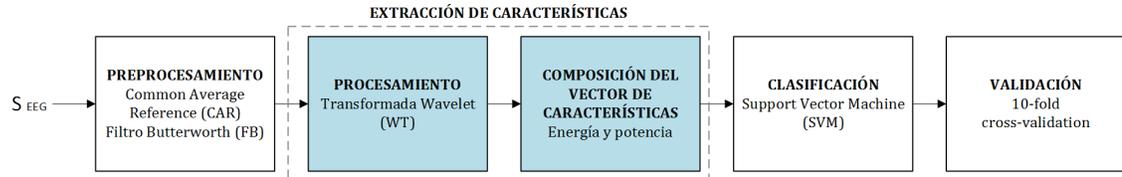


Figura 3.12: Extracción de características del método 2 en la etapa de entrenamiento.

Se procede a detallar el algoritmo de WT y la composición del vector de características. Como en el método anterior, ambos algoritmos son orientados a la extracción de características del conjunto de señales EEG de la base de datos seleccionada.

Transformada Wavelet

La transformada Wavelet (Wavelet Transform, WT) es un método de análisis en tiempo-frecuencia útil para extraer información de bandas específicas.

Los hiperparámetros a configurar son:

- Wavelet madre: Es la función que se desplaza en el tiempo y se escala para descomponer la señal de entrada.
- Nivel de descomposición: Representa cuantas veces dividirá la señal sucesivamente en rangos de frecuencia relativos al ancho de banda.

Para hallar el nivel de descomposición requerido se necesita conocer el ancho de banda (BW). Considerando que la frecuencia de muestreo usada en la base de datos es de 250Hz y que $BW = f_m/2$ entonces $BW = 125Hz$. La banda de interés se encuentra entre 8 y 30Hz; por lo tanto, se procede a dividir el ancho de banda sucesivamente hasta encontrar el nivel de descomposición que contenga el rango deseado. Esto se muestra en la Figura 3.13, donde a $cD1$, $cD2$, $cD3$ y $cD4$ se les denomina señales de descomposición y a la $aD4$ se le denomina señal principal.

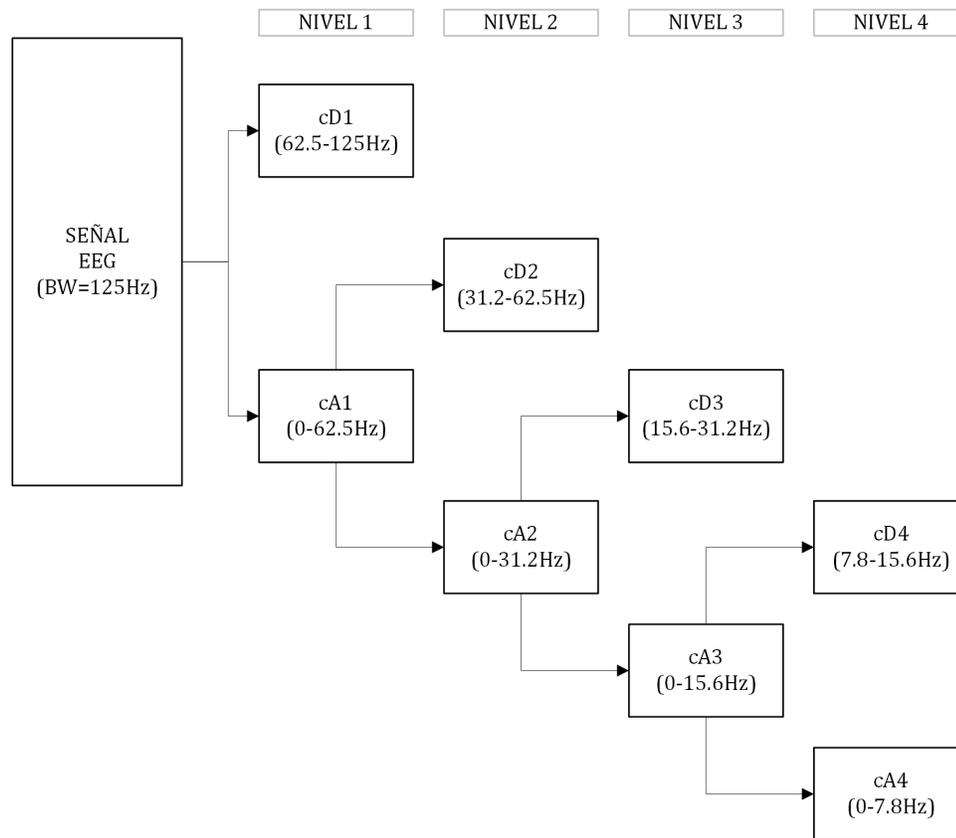


Figura 3.13: Estructura de nivel de descomposición.

Debido a que en el nivel de descomposición igual a cuatro obtenemos entre $cD4$ y $cD3$ un rango de frecuencias entre 7.8 y 31.2Hz, la cual se aproxima a la deseada, se decide trabajar con este nivel de descomposición.

Con respecto a la elección de una wavelet madre, no existe una que *a priori* sea óptima para todas las señales EEG. Por lo que se experimenta con diversas wavelet madre y se analiza la tasa de acierto resultante para escoger la más adecuada.

Las familias de wavelet que se consideran en las pruebas son: Daubichies, Haar, Coiflet y Symlet. Algunas de las wavelet madre probadas y que pertenecen a estas familias se muestran en la Figura 3.14.

Para la implementación de la transformada Wavelet en python se usa la librería PyWavelet, disponible en [67].

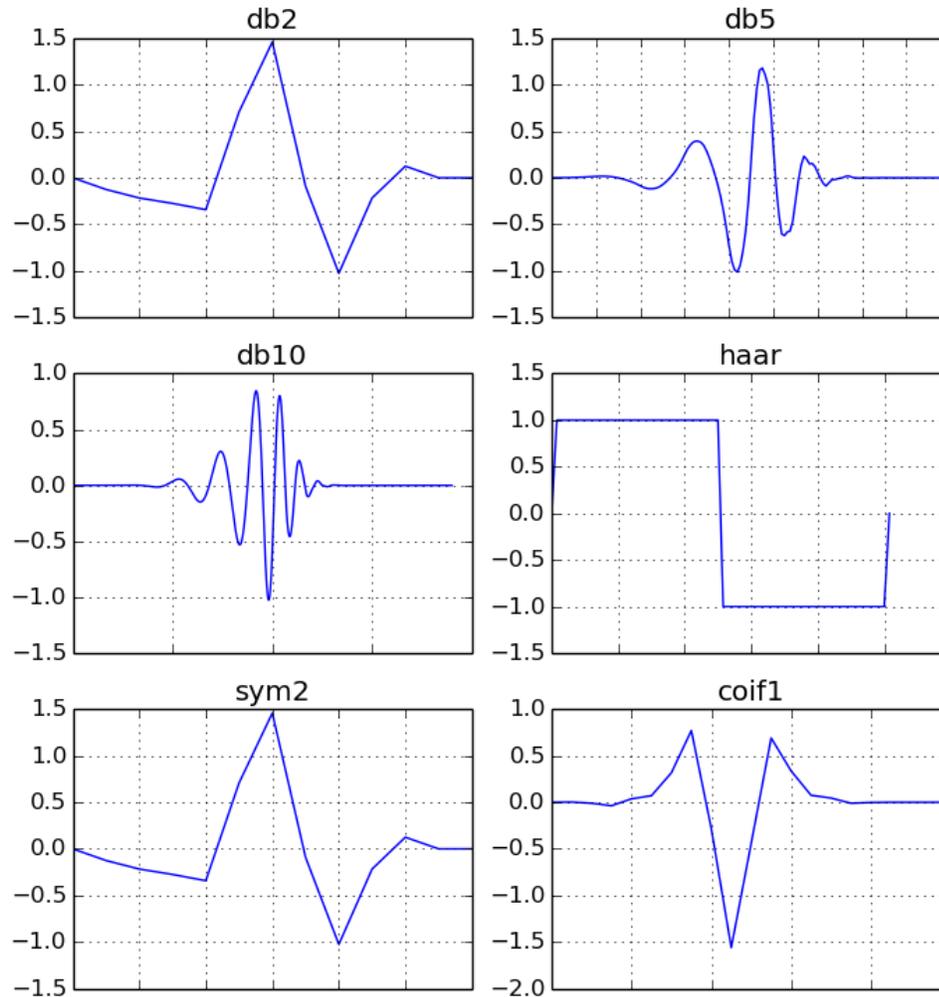


Figura 3.14: Wavelet madre.

Composición del vector de características

La composición del vector de características requiere de un previo cálculo de los valores representativos de la señal procesada.

De este modo, usando la WT en un canal, las señales $cD1$, $cD2$, $cD3$, $cD4$ y $cA4$ son obtenidos. Estas señales son empleadas para el cálculo de la energía en cada banda deseada: $cD3$ y $cD4$ para las ondas μ y β y $cA4$ para conocer las energía de la componente principal respecto de la total.

El cálculo de la energía de $cD3$ se deriva de:

$$E(cD3) = (cD3).(cD3)^T \quad (3.19)$$

Del mismo modo, la energía de $cD4$ se calcula como:

$$E(cD4) = (cD4).(cD4)^T \quad (3.20)$$

La energía relativa de la componente principal $cA4$ respecto de la energía total se obtiene de:

$$E(cA4) = \frac{(cA4).(cA4)^T}{\sum_1^4 (cD_i).(cD_i)^T} \quad (3.21)$$

Con estos tres valores por cada uno de los N canales, se obtiene el vector de características (fv) compuesto por $3N$ valores. Tal que fv es igual a:

$$fv = [E_1(cD3), E_1(cD4), E_1(cA4), E_2(cD3), E_2(cD4), E_2(cA4), \dots, E_N(cD3), E_N(cD4), E_N(cA4)] \quad (3.22)$$

Este procedimiento es aplicado a todos los ensayos y se concatenan de modo que se obtiene finalmente un $fv \in \mathbb{R}^{M \times 3N}$, donde M es el número de ensayos y N el número de canales.

Antes de entrenar el clasificador usando el vector de características fv este es normalizado. El nuevo vector de características normalizado F se calcula como:

$$F = \frac{fv - \mu}{\sigma} \quad (3.23)$$

Donde μ y σ son respectivamente la media y desviación estándar de fv sobre cada columna.

En la Figura 3.15 se muestra el diagrama de flujo de la extracción de características del método 2.

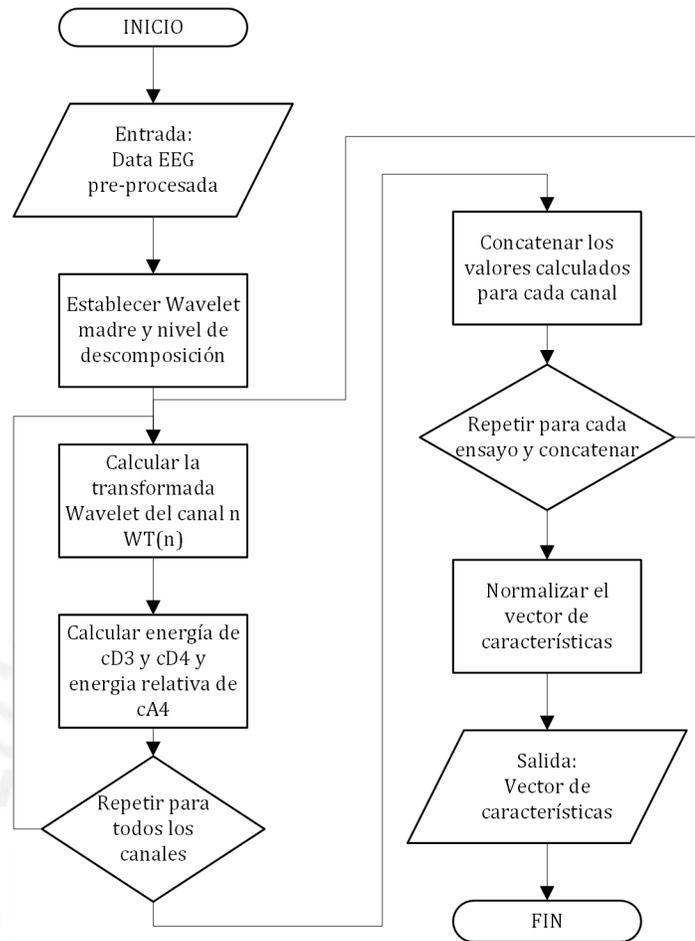


Figura 3.15: Diagrama de flujo de la extracción de características del método 2.

3.1.4. Clasificación

El objetivo de la clasificación es predecir correctamente la clase a la que pertenece un vector de características determinado. La clase 1, 2, 3 y 4 corresponden respectivamente a la imaginación del movimiento de la mano izquierda, derecha, de pies y lengua.

Para la clasificación se usan algoritmos de aprendizaje de máquina supervisado, los cuales se caracterizan por contar con una etiqueta. La etiqueta indica a qué clase pertenece cada vector de características. Con estas etiquetas y vectores el algoritmo entrena al clasificador.

Debido a su amplio uso en BMI y buen desempeño, se propone *Support Vector Machine* en la fase de clasificación de los métodos 1 y 2.

Support Vector Machine (SVM)

Este clasificador construye hiperplanos de separación y elige el que maximiza los márgenes entre clases. El resultado del hiperplano óptimo es una combinación de algunos puntos conocidos como vectores de soporte.

El clasificador SVM está orientado a la clasificación binaria. Para múltiples clases se propone el uso de la extensión *One-Versus-One Support Vector Machine* (OVR-SVM). Los parámetros a configurar dentro del clasificador corresponden a la elección del *Kernel*, que genera los límites de decisión, y la penalización “C” del clasificador.

Se prueban diferentes Kernel con diversos parámetros del mismo para encontrar el más adecuado al grupo de datos que se tiene, estos se muestran en la Tabla 3.1.

Tabla 3.1: Kernel considerados para optimizar el clasificador.

Kernel	Parámetro	Función
Lineal	-	$K(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}'$
Polinomial	Grado = d	$K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x} \cdot \mathbf{x}')^d$
Radial Basis Function (RBF)	Gamma = γ	$K(\mathbf{x}, \mathbf{x}') = \exp^{-\gamma \ \mathbf{x} - \mathbf{x}'\ ^2}$

Para la implementación de este algoritmo se emplea la librería *svm* del paquete *scikit-learn* [68]. Scikit-learn es una herramienta *open-source* ampliamente usada para análisis de datos y *data mining* que cuenta con librerías para clasificación, regresión, entre otras.

3.1.5. Validación

La validación es una técnica para evaluar el desempeño de un clasificador y la capacidad de generalización de este para predecir un conjunto de datos independientes. Como resultado de esta evaluación se obtiene la tasa de acierto, el cual es uno de los principales criterios para la elección de un conjunto de parámetros en el procesamiento y el clasificador.

K-veces validación cruzada

K-veces validación cruzada o *K-fold cross-validation*, es un técnica de validación ampliamente utilizada en el área de aprendizaje de máquina. Este consiste en dividir la base de datos en K -grupos, se entrena el algoritmo con $K - 1$ grupos y se evalúa la precisión de predecir la clase del grupo restante. Este proceso se realiza K -veces variando los grupos de entrenamiento y prueba.

Debido a la cantidad de ensayos con que se cuenta en la base de datos se elige trabajar con $K = 10$. En la Figura 3.16 se muestra como se dividen los grupos de la base de datos usando 10-veces validación cruzada.

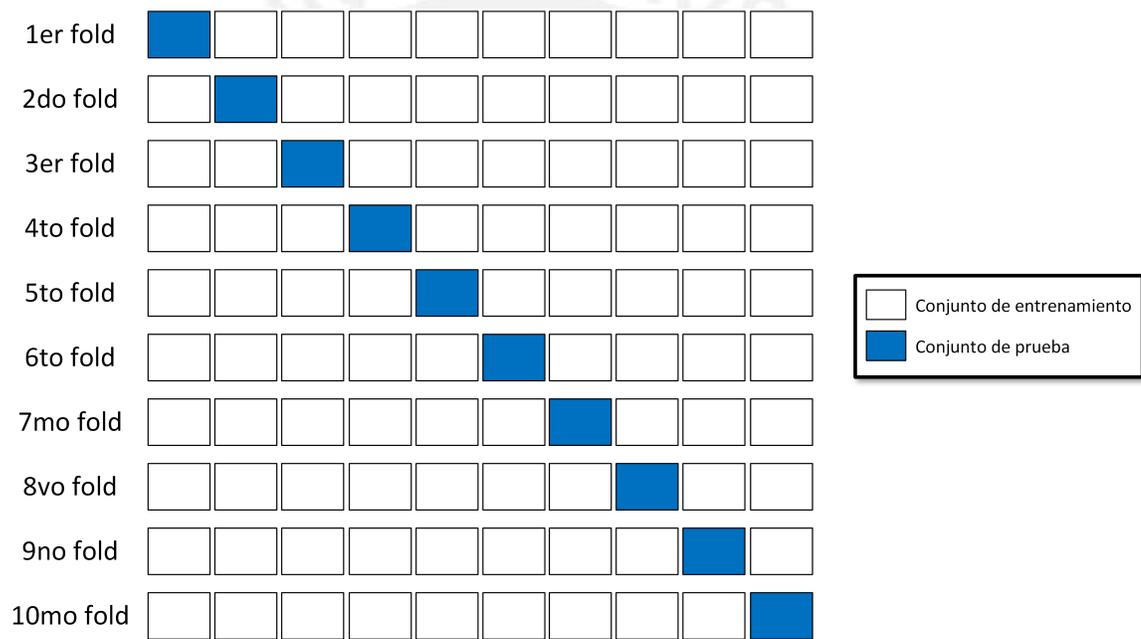


Figura 3.16: Validación usando 10-veces validación cruzada

3.2. Etapa 2: Sistema online

En esta etapa se considera poner a prueba en modo online el método de procesamiento seleccionado de la etapa de entrenamiento. Por lo que se utiliza los algoritmos expuestos en la etapa 1; sin embargo, la principal diferencia está en el desarrollo del *hardware in the loop* que permite emular la transmisión de la señal EEG almacenada en la base de datos en tiempo real.

3.2.1. Hardware in the loop

Se considera la implementación de un sistema que emule el envío de señales EEG. Esta técnica es conocida como *hardware in the loop* y es ampliamente utilizada para el desarrollo y comprobación de sistemas embebidos en tiempo real. La configuración del sistema se realiza de acuerdo a las especificaciones de la base de datos a emular (Dataset IIIa BCI Competition 2005).

Se implementa la comunicación en una red ad-hoc inalámbrica (WANET, por sus siglas en inglés) bajo el protocolo TCP/IP ya que provee una comunicación punto a punto, inalámbrica y portable. Una red ad hoc es un tipo de red inalámbrica descentralizada, donde cada nodo participa en la comunicación y no depende de una infraestructura (routers y access point).

Como se trató en la Sección 2.2.2, se usa una microcomputadora Raspberry Pi como servidor para la transmisión de datos. El servidor implementado está configurado para ejecutarse apenas se energice la Raspberry Pi y se mantenga a la espera de la petición del cliente (receptor).

Cuando se establece la comunicación a el cliente, el servidor concatena el valor de todos los canales correspondientes una muestra y la respectiva etiqueta a la que pertenece el conjunto de datos de MI para comparar con la precedida por el clasificador, esta trama a ser transmitida se muestra en la Figura 3.17. Se repite este proceso cada 4 mseg., correspondiente a la frecuencia de muestreo de la base de datos (250 Hz). El diagrama de flujo del proceso se muestra en la Figura 3.18.

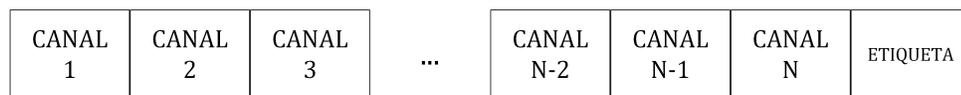


Figura 3.17: Trama a transmitir.

El desarrollo del programa ha sido realizado en *python* y se basa en el uso del paquete *socket* para las conexiones. Para ejecutar el programa con el arranque del sistema se edita el archivo `/etc/rc.local` agregando la dirección del ejecutable.

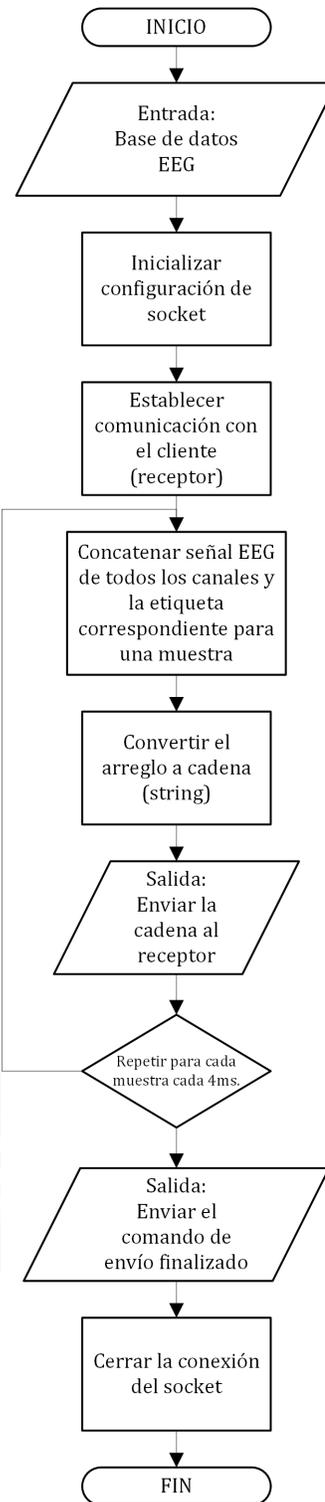


Figura 3.18: Diagrama de flujo del Servidor.

Capítulo 4:

Resultados y discusión

En este capítulo se expone los resultados de las etapas de entrenamiento y transmisión en tiempo real; y el modo en que el algoritmo seleccionado en el entrenamiento fue implementado para las pruebas usando señales EEG en tiempo continuo.

4.1. Etapa 1: Entrenamiento

En esta etapa se evalúa el desempeño del sistema embebido y la laptop para el procesamiento de señales EEG. A su vez, se evalúa el desempeño de los algoritmos empleados en cada método propuesto. El diagrama de bloque de cada método se encuentran en la Figura 3.1 y Figura 3.2.

4.1.1. Método 1

Se realizaron diversos experimentos para encontrar el conjunto más adecuado de parámetros que entregue una mayor tasa de acierto con menor tiempo de entrenamiento.

Los parámetros a modificar en las fases de pre-procesamiento y extracción de características son:

- Número de electrodos: La base de datos proporciona 60 electrodos, sin embargo es deseable el uso de una menor cantidad. Se experimenta con 60, 32, 16 y 8 canales. La posiciones de estos canales se muestran en la Figura 3.3.
- Re-referencia : Se evalúa el uso de CAR en la etapa de pre-procesamiento.
- Número de componentes CSP: Representa el número de columnas a utilizar de la matriz CSP.

Debido a previos experimentos y resultados aceptables se decide trabajar inicialmente con un clasificador SVM con *kernel* lineal y parámetro de penalización $C = 50$. Por cada configuración de canales se escoge el resultado más resaltante y se procede a modificar los parámetros del clasificador con el objetivo optimizarlo y obtener mejores resultados.

A. Configuración usando 60 canales

En la Tabla 4.1 se muestra los resultados de la tasa de acierto (accuracy = acc) y tiempo de entrenamiento usando 60 canales. Como se puede observar conforme aumenta el número de CSP en la extracción de características la tasa de acierto aumenta así como el tiempo de procesamiento.

El uso del filtro CAR implica un pequeño tiempo adicional de entrenamiento mientras que la tasa de acierto no aumenta de forma consistente; sin embargo, se advierte que este conduce a una mejor calidad de señal.

Tabla 4.1: Resultado variando parámetros de procesamiento del Método 1 usando 60 canales

n° Ch.	Pre-proces.		Ext.Car	Clasif.	Validación			
	Re-referencia	F.B (Hz)	CSP (n°)	SVM Lineal C	Acc(%)		Tiempo(seg.)	
					PC	SBC	PC	SBC
60	CAR	8 - 30	8	50	90	90	0.51	10.02
60	-	8 - 30	8	50	90	90	0.38	8.73
60	CAR	8 - 30	6	50	88	88	0.52	9.99
60	-	8 - 30	6	50	88	88	0.38	8.60
60	CAR	8 - 30	4	50	79	79	0.49	9.89
60	-	8 - 30	4	50	86	86	0.37	8.16
60	CAR	8 - 30	2	50	87	87	0.47	9.55
60	-	8 - 30	2	50	84	84	0.36	7.97

Fuente: Elaboración Propia

Se selecciona la primera configuración (resaltada en letras negritas) y se procede a evaluar los resultados modificando diversos parámetros solo en el clasificador con el objetivo de optimizar el algoritmo. En la Tabla 4.2 se muestra los resultados obtenidos.

Tabla 4.2: Resultados modificando el Kernel y sus parámetros usando 60 canales.

SVM		Acc.(%)		Tiempo(seg.)	
Kernel	Parametro	PC	SBC	PC	SBC
Linear	C=1000	90 %	90 %	0.50	10.17
	C=500	90 %	90 %	0.50	10.00
	C=100	90 %	90 %	0.50	10.21
	C=50	90 %	90 %	0.50	10.00
	C=10	90 %	90 %	0.50	10.22
	C=1	90 %	90 %	0.51	10.26
	C=0.1	89 %	89 %	0.51	10.23
	C=0.01	89 %	89 %	0.52	10.32
	C=0.001	83 %	83 %	0.52	10.56
Polynomial	Degree=2	89 %	89 %	0.53	10.21
	Degree=3	89 %	89 %	0.51	10.27
Radial Basis Function (RBF)	Gamma=0.01	90 %	90 %	0.51	10.27
	Gamma=0.16	90 %	90 %	0.51	10.30
	Gamma=1	90 %	90 %	0.51	10.32
	Gamma=5	90 %	90 %	0.51	10.26

Fuente: Elaboración Propia

Como se muestra en la Tabla 4.2 la tasa de acierto en la laptop (PC) y en el sistema embebido (SBC) es la misma. Ambos sistemas predicen del mismo modo cada estado de imaginación motora.

Se selecciona para esta configuración de 60 electrodos el uso del *kernel* lineal con un parámetro de penalización $C = 50$. El tiempo de entrenamiento en el sistema embebido es aproximadamente 20 veces el de la laptop.

B. Configuración usando 32 canales

En la Tabla 4.3 se muestra los resultados de la tasa de acierto y tiempo de entrenamiento usando 32 canales. La configuración seleccionada se resalta en negrita, esta no usa CAR para el pre-procesamiento y usa 6 columnas de la matriz CSP. El tiempo de entrenamiento del sistema embebido es 16 veces al de la laptop, menor que en el caso anterior.

Tabla 4.3: Resultados variando parámetros de procesamiento del Método 1 usando 32 canales.

n° Ch.	Pre-proces.		Ext.Car	Clasif.	Validación			
	Re-referencia	F.B (Hz)	CSP (n°)	SVM Lineal	Acc(%)		Tiempo(seg.)	
				C	PC	SBC	PC	SBC
32	CAR	8 - 30	8	50	89	89	0.38	4.36
32	-	8 - 30	8	50	89	89	0.27	3.98
32	CAR	8 - 30	6	50	85	85	0.40	4.25
32	-	8 - 30	6	50	88	88	0.23	3.87
32	CAR	8 - 30	4	50	81	81	0.38	4.17
32	-	8 - 30	4	50	80	80	0.24	3.78
32	CAR	8 - 30	2	50	82	82	0.37	4.06
32	-	8 - 30	2	50	86	86	0.24	3.65

Fuente: Elaboración Propia

Se optimiza el algoritmo luego de modificar los parámetros del clasificador. Se encuentra un mejor resultado usando un *Kernel* lineal con un parámetro $C=1$. Los experimentos realizados se muestran en la Tabla 4.4.

Tabla 4.4: Resultados modificando el Kernel y sus parámetros usando 32 canales.

SVM		Accuracy		Tiempo(seg.)	
Kernel	Parametro	PC	SBC	PC	SBC
Linear	C=1000	88 %	88 %	0.26	3.90
	C=500	88 %	88 %	0.23	3.88
	C=100	88 %	88 %	0.23	3.90
	C=50	88 %	88 %	0.23	3.90
	C=10	88 %	88 %	0.23	3.88
	C=1	89 %	89 %	0.23	3.88
	C=0.1	89 %	89 %	0.22	3.90
	C=0.01	83 %	83 %	0.22	3.87
	C=0.001	81 %	81 %	0.22	3.90
Polynomial	Degree=2	89 %	89 %	0.23	3.90
	Degree=3	89 %	89 %	0.23	3.90
Radial Basis Function	Gamma=0.01	88 %	88 %	0.23	3.91
	Gamma=0.16	88 %	88 %	0.24	3.87
	Gamma=1	88 %	88 %	0.25	3.90

Fuente: Elaboración Propia

C. Configuración usando 16 canales

Del mismo modo en la Tabla 4.5 se muestra los resultados de la tasa de acierto y tiempo de entrenamiento usando 16 canales. Como se puede apreciar, los resultados de *accuracy* son idénticos debido a que usan el mismo algoritmo como base del procesamiento. Se elige la configuración resaltada en negrita.

El tiempo de entrenamiento en el sistema embebido (SBC) respecto al de la laptop en esta configuración, es de 1 a 13, una relación menor que la presentada en la configuración de 60 canales. El tiempo de entrenamiento más largo es de 2.19 seg.; sin embargo, este tiempo es bastante pequeño considerando que la etapa de entrenamiento solo se realiza una vez.

Tabla 4.5: Resultado variando parámetros de procesamiento del Método 1 usando 16 canales.

n° Ch.	Pre-proces.		Ext.Car (n°)	Clasif. SVM Lineal C	Validación			
	Re-referencia	F.B (Hz)			Acc(%)		Tiempo(seg.)	
			PC	SBC	PC	SBC		
16	CAR	8 - 30	8	50	86	86	0.17	2.19
16	-	8 - 30	8	50	85	85	0.16	2.07
16	CAR	8 - 30	6	50	86	86	0.17	2.13
16	-	8 - 30	6	50	83	83	0.16	2.00
16	CAR	8 - 30	4	50	81	81	0.16	2.08
16	-	8 - 30	4	50	88	88	0.15	1.96
16	CAR	8 - 30	2	50	82	82	0.16	2.02
16	-	8 - 30	2	50	82	82	0.15	1.92

Fuente: Elaboración Propia

A diferencia de los casos anteriores, en esta configuración se encuentra un mejor resultado usando un *kernel* polinomial de 3er grado. La tasa de acierto hallada con esta configuración es de 91%. Los resultados se muestran en la Tabla 4.6. Como se puede observar en esta tabla, usando un *kernel RBF* se encuentran resultados similares, sin embargo el tiempo de procesamiento es mayor y se presume que pueda no ser generalizable debido a la complejidad de su función de decisión. Con respecto a los tiempo de procesamiento, se observa en promedio de 1.96 seg. para el sistema embebido, mientras que 0.15 seg. para la laptop, una relación de 1 a 13 aproxima-

damente. El tiempo de entrenamiento es el adecuado considerando que este solo se realizará una vez y con el clasificador entrenado se procede a predecir los estados de MI en la etapa del sistema online.

Tabla 4.6: Resultados modificando el Kernel y sus parámetros usando 16 canales.

SVM		Accuracy		Tiempo(seg.)	
Kernel	Parametro	PC	SBC	PC	SBC
Linear	C=1000	88 %	88 %	0.155	1.96
	C=500	88 %	88 %	0.154	1.96
	C=100	88 %	88 %	0.155	1.97
	C=50	88 %	88 %	0.155	1.96
	C=10	88 %	88 %	0.153	1.95
	C=1	88 %	88 %	0.153	1.95
	C=0.1	87 %	87 %	0.153	1.95
	C=0.01	82 %	82 %	0.153	1.96
	C=0.001	80 %	80 %	0.155	1.97
Polynomial	Degree=2	88 %	88 %	0.153	1.96
	Degree=3	91 %	91 %	0.153	1.95
Radial	Gamma=0.01	91 %	91 %	0.154	1.96
Basis	Gamma=0.16	91 %	91 %	0.155	1.96
Function	Gamma=1	91 %	91 %	0.154	1.96
(RBF)	Gamma=5	91 %	91 %	0.154	1.96

Fuente: Elaboración Propia

D. Configuración usando 8 canales

Se experimenta con diferentes configuraciones. Los resultados se muestran en la Tabla 4.7. Si bien es cierto que la tasa de acierto ha disminuido con relación a las anteriores configuraciones que utilizan más electrodos, la ventaja del uso de solo 8 electrodos es bastante importante debido a que un sistema de adquisición de señales EEG de bajo costo tiene entre 8 y 16 canales EEG.

En esta configuración el sistema embebido tarda aproximadamente 10 veces en entrenar el algoritmo que la laptop.

Tabla 4.7: Resultado variando parámetros de procesamiento del Método 1 usando 8 canales.

n° Ch.	Pre-proces.		Ext.Car	Clasif.	Validación			
	Re-referencia	F.B (Hz)	CSP (n°)	SVM Lineal	Acc(%)		Tiempo(seg.)	
				C	PC	SBC	PC	SBC
8	CAR	8 - 30	6	50	84	84	0.13	1.31
8	-	8 - 30	6	50	85	85	0.13	1.29
8	CAR	8 - 30	4	50	82	82	0.13	1.28
8	-	8 - 30	4	50	84	84	0.12	1.25
8	CAR	8 - 30	2	50	82	82	0.12	1.25
8	-	8 - 30	2	50	83	83	0.11	1.21

Fuente: Elaboración Propia

Al modificar los parámetros del clasificador bajo la misma configuración en la etapa de pre-procesamiento y extracción de características, encontramos que con un Kernel RBF y parámetro $\gamma = 1$ se obtiene la mayor tasa de acierto, siendo esta de 90%. Los resultados se muestran en la Tabla 4.8.

Tabla 4.8: Resultados modificando el Kernel y sus parámetros usando 8 canales.

SVM		Accuracy		Tiempo(seg.)	
Kernel	Parametro	PC	SBC	PC	SBC
Linear	C=1000	86 %	86 %	0.127	1.26
	C=500	86 %	86 %	0.127	1.26
	C=100	84 %	84 %	0.126	1.25
	C=50	84 %	84 %	0.126	1.26
	C=10	89 %	89 %	0.124	1.26
	C=1	89 %	89 %	0.124	1.25
	C=0.1	86 %	86 %	0.124	1.26
	C=0.01	81 %	81 %	0.127	1.26
Polynomial	Degree=2	87 %	87 %	0.126	1.26
	Degree=3	85 %	85 %	0.126	1.26
Radial Basis Function	Gamma=0.01	90 %	90 %	0.129	1.26
	Gamma=0.16	90 %	90 %	0.129	1.26
	Gamma=1	90 %	90 %	0.128	1.25

Fuente: Elaboración Propia

En la Tabla 4.9 se muestran los mejores resultados por cada configuración de 8, 16, 32 y 60 electrodos.

Como se puede observar, usando una configuración con 16 electrodos EEG se encuentra una mayor tasa de éxito, un tiempo de entrenamiento adecuado y una cantidad de electrodos recomendada. Esta configuración es la seleccionada como representante del método 1.

Tabla 4.9: Mejores resultados de cada configuración del Método 1.

n° Ch.	Pre-proces.		Ext.Car	Clasif.		Validación			
	Re-referencia	F.B (Hz)	CSP (n°)	Acc(%)		Acc(%)		Tiempo(seg.)	
				Kernel	Par.	PC	SBC	PC	SBC
60	CAR	8 - 30	8	Lineal	C=50	90	90	0.50	10.00
32	-	8 - 30	6	Lineal	C=1	89	89	0.23	3.88
16	-	8 - 30	4	Poli.	d=3	91	91	0.15	1.95
8	-	8 - 30	4	RBF	g=1	90	90	0.12	1.25

Fuente: Elaboración Propia

4.1.2. Método 2

Al igual que el método 1, se realizaron diversos experimentos para encontrar el conjunto más adecuado de parámetros que nos entregue una mayor tasa de acierto con menor tiempo de entrenamiento.

Los parámetros a modificar en las fases de pre-procesamiento y extracción de características son:

- Número de electrodos: La base de datos proporciona 60 electrodos, sin embargo es deseable el uso de una menor cantidad. Se experimenta con 60, 32, 16 y 8 canales. La posiciones de estos canales se muestran en la Figura 3.3.
- Re-referencia : Se evalúa el uso de CAR en la etapa de pre-procesamiento.
- Wavelet madre: Herramienta principal de la transformada Wavelet para descomponer la señal.

De modo similar que en el método 1, para la realización de las pruebas, inicialmente se trabaja con un clasificador SVM de *kernel* Lineal y parámetro $C=50$. Las pruebas consisten en modificar los parámetros en la fase de pre-procesamiento y la extracción de características. Con la configuración más adecuada se procede a optimizar el clasificador probando con diferentes *kernel* y parámetros.

En la Tabla 4.10 se muestra los resultados de diversas configuraciones. Como se puede apreciar, el porcentaje de éxito no cambia; esto, debido a que los algoritmos que se implementan en la laptop y el sistema embebido (SBC) tienen el mismo desarrollo matemático. Por otro lado, el tiempo de entrenamiento más largo fue de escasos 7 segundos. Debido a que este entrenamiento solo se realiza una vez y puede ser desempeñado sin necesidad de que el usuario esté presente, se considera aceptable.

La configuración seleccionada se resalta en letras negritas. Esta contempla el uso de 16 electrodos, el filtro espacial CAR y la wavelet madre *db5* de la familia Daubechie.

La mayor tasa de acierto vista en la Tabla 4.10 es 80% la cual es menor a la mostrada en el método 1; por otro lado, el mayor tiempo de entrenamiento encontrado con este método es de 6.56seg., menor al del método 1. El efecto del filtro CAR en este método evidencia una mejora en la tasa de acierto y un menor tiempo de entrenamiento. En ambos sistemas de procesamiento se obtuvo la misma tasa de acierto.

El tiempo de entrenamiento en el sistema embebido fue de 1.8 segundos, 5 veces más que en la laptop.

Para optimizar el resultado de usando 16 canales, se modifican parámetros en el clasificador dejando los del procesamiento con la anterior configuración. Los resultados se muestran en la Tabla 4.11. Se observa que los resultados aumentaron en un 10% modificando el Kernel de lineal a uno RBF con $\gamma = 1$. Este resultado representa al Método 2.

Tabla 4.10: Resultado de la evaluación de los parámetros de pre-procesamiento y extracción de características del Método 2.

n° Ch.	Pre-proces.	Ext.Car.	Clasif.	Validación			
	Re-referencia	WT	SVM Lineal	Acc.(%)		Tiempo(seg.)	
		W.Madre	C	PC	SBC	PC	SBC
60	CAR	db2	50	76	76	0.84	6.08
60	CAR	db10	50	80	80	1.13	7.02
60	CAR	sym2	50	76	76	0.84	6.27
60	CAR	sym6	50	76	76	1.01	6.56
60	-	sym6	50	76	76	0.94	4.95
60	-	coif1	50	70	70	0.81	4.75
60	-	haar	50	68	68	0.72	4.69
32	CAR	db2	50	78	78	0.52	3.06
32	CAR	db5	50	79	79	0.58	3.19
32	CAR	db10	50	80	80	0.67	3.50
32	CAR	sym2	50	78	78	0.50	3.10
32	-	db5	50	67	67	0.50	2.88
32	-	db10	50	72	72	0.62	3.15
32	-	sym2	50	69	69	0.43	2.78
32	-	sym6	50	72	72	0.50	2.90
16	CAR	db2	50	74	74	0.26	1.77
16	CAR	db5	50	75	75	0.30	1.83
16	CAR	sym2	50	74	74	0.27	1.77
16	CAR	haar	50	72	72	0.26	1.79
16	-	db10	50	68	68	0.37	1.97
16	-	sym2	50	63	63	0.34	1.99
16	-	sym6	50	67	67	0.33	1.85
16	-	coif1	50	66	66	0.33	2.00
8	CAR	db2	50	72	72	0.25	1.55
8	CAR	db5	50	70	70	0.35	1.75
8	CAR	db10	50	72	72	0.29	1.63
8	CAR	sym2	50	72	72	0.25	1.55
8	-	db2	50	62	62	0.59	2.21
8	-	db5	50	64	64	0.40	2.13
8	-	db10	50	63	63	0.44	2.21
8	-	coif1	50	67	67	0.50	2.61

Fuente: Elaboración Propia

Tabla 4.11: Evaluación de Kernel y sus parámetros en el Método 2.

SVM		Accuracy		Tiempo(seg.)	
Kernel	Parametro	PC	SBC	PC	SBC
Linear	C=1000	75 %	75 %	0.127	1.26
	C=500	75 %	75 %	0.127	1.26
	C=100	75 %	75 %	0.126	1.25
	C=50	75 %	75 %	0.126	1.26
	C=10	75 %	75 %	0.124	1.26
	C=1	73 %	73 %	0.124	1.25
	C=0.1	78 %	78 %	0.124	1.26
	C=0.01	67 %	67 %	0.127	1.26
	C=0.001	56 %	56 %	0.127	1.25
Polynomial	Degree=2	46 %	46 %	0.126	1.26
	Degree=3	61 %	61 %	0.126	1.26
Radial Basis Function	Gamma=0.01	76 %	76 %	0.129	1.26
	Gamma=0.16	76 %	76 %	0.129	1.26
	Gamma=1	85 %	85 %	0.128	1.25

Fuente: Elaboración Propia

La Tabla 4.12 muestra los mejores resultados en cada uno de los dos métodos abordados. En ambos métodos el número de electrodos utilizados de la base de datos es de 16. El método 1 presenta 6 % más tasa de acierto que el método 2. En tema de tiempos, el método 2 es 0.7 segundos más rápido; sin embargo, para temas de entrenamiento, este tiempo no es significativo.

Debido a que el criterio que pesa más es que el algoritmo pueda predecir con exactitud la clase a la que pertenece un conjunto de señales EEG y debido a que los tiempos son pequeños, se decide trabajar con la mejor configuración propuesta en el método 1.

Tabla 4.12: Mejores resultados en la etapa de entrenamiento

Configuración	Resultados			
	Acc(%)		Tiempo(seg.)	
	PC	SBC	PC	SBC
Método 1	91	91	0.15	1.95
Método 2	85	85	0.13	1.25

Fuente: Elaboración Propia

4.2. Etapa 2: Sistema online

En esta etapa se evalúa el desempeño del sistema embebido y la laptop en el procesamiento de señales EEG en tiempo real. Para ello, se usa la configuración seleccionada en la etapa 1. Esta comprende el uso de 16 electrodos situados alrededor de la parte central del cerebro, el filtro Butterworth de 8 a 30Hz en la fase de pre-procesamiento, 4 columnas de la matriz CSP para la extracción de características y un clasificador SVM con *kernel* polinomial de tercer grado.

Se realizaron las pruebas experimentales usando el concepto de *hardware in the loop* efectuada en la microcomputadora Raspberry Pi 3. La duración de cada estado en la base de datos es de 4 seg y se transmite de forma continua cada estado. El sistema embebido Odroid-xu4 recibe la señal en tiempo real y la procesa en el momento en que la bandera se activa. La bandera se activa cuando se cumple la ventana de tiempo fijada. La ventana de tiempo indica cuantos segundos de recepción de datos se usa para procesar la señal.

En la tabla 4.13 se muestra la tasa de acierto y el tiempo de procesamiento (en milisegundos) en el sistema online. Se modifica la ventana de tiempo; mientras más corta sea la ventana de tiempo la frecuencia de envío de comandos aumenta, pero la tasa de acierto disminuye.

Se puede observar que ambos sistemas mantienen la misma tasa de acierto al igual que en la etapa de entrenamiento. De ello se demuestra que el sistema embebido Odroid-xu4 puede realizar la etapa de procesamiento en tiempo real sin diferencias en la predicción de los estados con respecto a la laptop.

Tabla 4.13: Tasa de acierto y tiempo de procesamiento en el sistema online.

Ventana de tiempo	Resultados			
	Acc(%)		Tiempo(mseg.)	
	PC	SBC	PC	SBC
4 seg.	90	90	1.9	15.9
2 seg.	83	83	1.2	9.1
1 seg.	81	81	0.7	5.9
0.8 seg.	80	80	0.6	5.3

Fuente: Elaboración Propia

En la Figura 4.1 se muestra la comparación entre los tiempo de procesamiento de cada sistema dependiendo de la ventana de tiempo seleccionada.

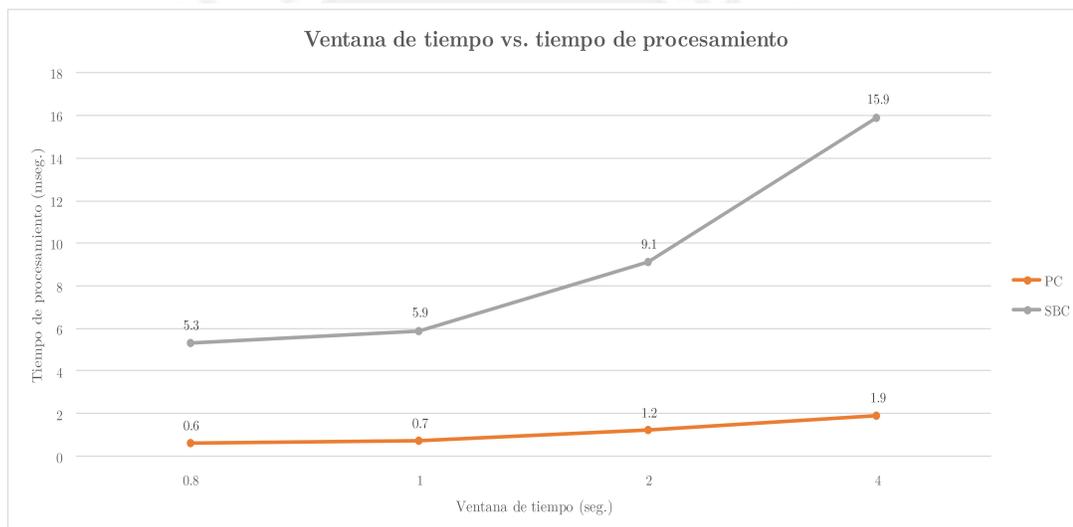


Figura 4.1: Comparación entre el sistema embebido y la laptop.

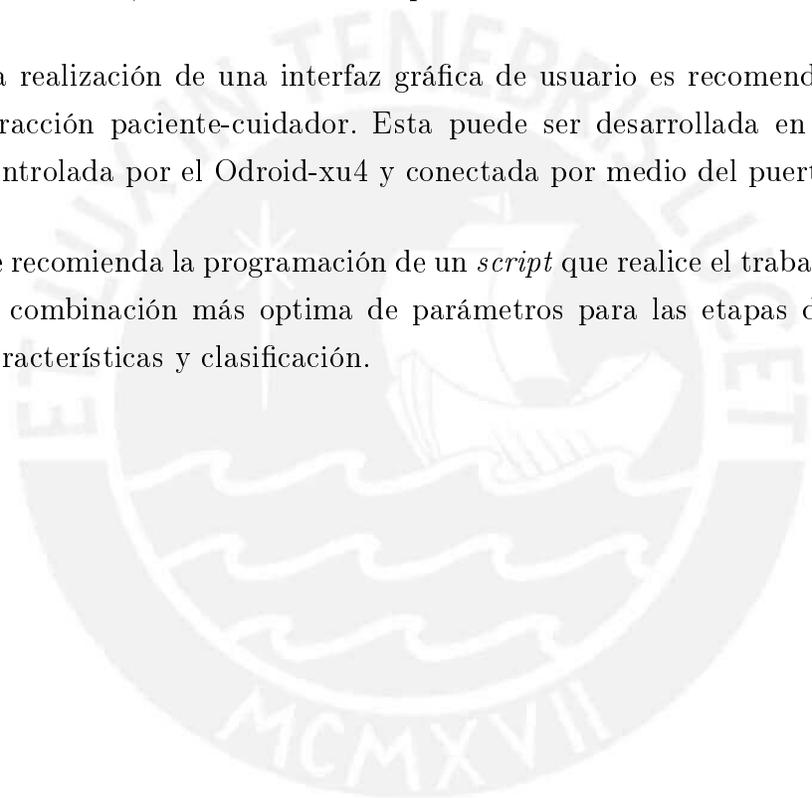
Fuente: Elaboración Propia

CONCLUSIONES

- La tasa de acierto resultante de cada experimento, utilizando la misma configuración de electrodos y algoritmos, es la misma tanto para el sistema embebido Odroid-xu4 como para la laptop. Esto demuestra que el sistema embebido puede ser usado como equipo de procesamiento ya que este va a predecir las mismas clases que predice la laptop ante una entrada de conjunto de señales EEG.
- El tiempo de entrenamiento en la configuración escogida en el método 1 es 13 veces mayor en el Odroid-xu4 (1.95s) que en la laptop (0.15s). En el método 2 se encontramos que el Odroid-xu4 (1.25s) tarda 9 veces lo que la laptop (0.13s). Sin embargo, estos tiempos son mínimos y el algoritmo de entrenamiento solo es necesario ejecutarlos una única vez para entrenar el clasificador. Por lo tanto el Odroid-xu4 también puede ser usado para entrenar sus propios algoritmos.
- El tiempo de procesamiento en la etapa 2 (Online) de la laptop (1.2 mseg.) es aproximadamente 7 veces más corto que el del Odroid-xu4(9.1 mseg.) usando una ventana de tiempo de 2 seg. Estos tiempos son del rango del tiempo de muestreo (4 mseg.). Por lo tanto puede ejecutarse sin problemas ya que se ejecutan de forma paralela a la toma de datos.
- El sistema embebido Odroid-xu4 posee todas las características y cumple con todos los requisitos para cumplir la función de un equipo procesador de señales EEG. Se ha demostrado con experiencias de entrenamiento y sistema online la factibilidad del Odroid-xu4 para realizar un interfaz cerebro-máquina accesible, portable y de bajo costo, ya que el sistema embebido Odroid-xu4 cuesta 20 veces menos que la laptop Asus con la que se experimento.

RECOMENDACIONES

- Realizar un sistema complemente embebido requiere de hardware para la adquisición de señal EEG. Se recomienda el uso del OpenBCI como plataforma para el registro de señales cerebrales, el cual remplace la función tomada por la base de datos. Este se conecta con la PC embebida Odroid-xu4 de forma inalámbrica, lo cual conserva la portabilidad del sistema.
- La realización de una interfaz gráfica de usuario es recomendable para la interacción paciente-cuidador. Esta puede ser desarrollada en el software Qt, controlada por el Odroid-xu4 y conectada por medio del puerto HDMI.
- Se recomienda la programación de un *script* que realice el trabajo de seleccionar la combinación más óptima de parámetros para las etapas de extracción de características y clasificación.



Bibliografía

- [1] R. Fazel-rezai and W. Ahmad, “Interface Paradigm Design,” *Design*, 2009.
- [2] S. M. Kay and S. L. Marple, “Spectrum analysis - A modern perspective,” *Proceedings of the IEEE*, vol. 69, no. 11, pp. 1380–1419, 1981. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1456452>
- [3] N. Brodu, F. Lotte, and A. Lécuyer, “Comparative study of band-power extraction techniques for Motor Imagery classification,” *IEEE SSCI 2011 - Symposium Series on Computational Intelligence - CCMB 2011: 2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain*, pp. 95–100, 2011.
- [4] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, “A review of classification algorithms for EEG-based brain-computer interfaces.” *J Neural Eng*, vol. 4, no. 2, pp. R1–R13, 2007.
- [5] O. Kramer, *Dimensionality Reduction with Unsupervised Nearest Neighbors*, 2013, vol. 51.
- [6] R. Roy and V. Bommankanti, “User manual odroid-xu4,” 2015.
- [7] R. P. Fondation, “Raspberry Pi 3 Model B.” [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [8] BPI-Sinovoip, “Banana Pi M3,” 2016. [Online]. Available: <http://www.banana-pi.org/m3.html>
- [9] A. Schlögl, “Dataset IIIa : 4-class EEG data.” [Online]. Available: <http://www.bbc.de/competition/iii/desc{ }IIIa.pdf>
- [10] “ASUS ROG G751JY,” 2016. [Online]. Available: <https://www.asus.com/ROG-Republic-Of-Gamers/ROG-G751JY/>
- [11] G. Langevin, “Inmoov,” 2014. [Online]. Available: <http://www.inmoov.fr/project>
- [12] World Health Organization, “World Report on Disability - Summary,” *World Report on Disability 2011*, no. WHO/NMH/VIP/11.01, pp. 1–23, 2011.

- [13] J. J. Daly and J. E. Huggins, “Brain-Computer Interface: Current and Emerging Rehabilitation Applications,” *Archives of Physical Medicine and Rehabilitation*, vol. 96, no. 3, pp. S1—S7, 2015. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0003999315000209>
- [14] K.-R. M. G. Dornhege, J.R. Millán, T. Hinterberger, D.J. McFarland, “Towards Brain-Computer Interfacing,” *MIT Press, Cambridge, Massachusetts*, 2007.
- [15] Y. Li, J. Long, T. Yu, Z. Yu, C. Wang, H. Zhang, and C. Guan, “A hybrid BCI system for 2-D asynchronous cursor control.” *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, vol. 2010, pp. 4205–8, 2010.
- [16] Y. Li, J. Long, Z. Yu, T. Yu, C. Wang, H. Zhang, and C. Guan, “An EEG-based BCI system for 2-D cursor control by combining Mu/Beta rhythm and P300 potential,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 10 PART 1, pp. 2495–2505, 2010.
- [17] S. M. T. Müller, T. F. Bastos, and M. S. Filho, “Using a SSVEP-BCI to command a robotic wheelchair,” *Journal of Control, Automation and Electrical Systems*, vol. 24, no. 1-2, pp. 97–105, 2011.
- [18] T. Castermans, M. Duvinage, M. Petieau, T. Hoellinger, C. D. Saedeleer, K. Seetharaman, A. Bengoetxea, G. Cheron, and T. Dutoit, “Optimizing the Performances of a P300-Based Brain-Computer Interface in Ambulatory Conditions,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 4, pp. 566–577, 2011.
- [19] E. Iáñez, J. M. Azorín, A. Úbeda, J. M. Ferrández, and E. Fernández, “Mental tasks-based brain-robot interface,” *Robotics and Autonomous Systems*, vol. 58, no. 12, pp. 1238–1245, 2010.
- [20] A. Ubeda, J. M. Azorin, N. Garcia, J. M. Sabater, and C. Perez, “Brain-machine interface based on EEG mapping to control an assistive robotic arm,” *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 1311–1315, 2012.
- [21] “g.tec medical engineering,” p. 5, 2012. [Online]. Available: <https://gtecmedical.wordpress.com/>

- [22] “Proyectos tecnológicos y ejes temáticos,” 2015. [Online]. Available: <http://proyectostecnologicosmarth.blogspot.pe/2015/09/01/archive.html>
- [23] L. F. Nicolas-Alonso and J. Gomez-Gil, “Brain computer interfaces, a review,” *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012.
- [24] E. Bash, *An Introduction to the Event-Related Potential Technique*, 2014, vol. 1.
- [25] Farwell-Donchin, “Talking off the top of your head - toward a mental prosthesis utilizing event-related brain potentials,” 1988.
- [26] C. R. Hema, M. P. Paulraj, S. Yaacob, A. H. Adom, and R. Nagarajan, “Single trial motor imagery classification for a four state brain machine interface,” *Proceedings of 2009 5th International Colloquium on Signal Processing and Its Applications, CSPA 2009*, pp. 39–41, 2009.
- [27] J. Arnil, D. Anopas, M. Horapong, K. Luangrat, Y. Punsawad, and Y. Wong-sawat, “Bci-based assistive robot arm,” pp. 208–212, 2013.
- [28] A. Úbeda, E. Iáñez, and J. M. Azorín, “Shared control architecture based on RFID to control a robot arm using a spontaneous brain-machine interface,” *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 768–774, 2013.
- [29] E. Hortal, D. Planelles, A. Costa, E. Iáñez, A. Úbeda, J. M. Azorín, and E. Fernández, “SVM-based Brain–Machine Interface for controlling a robot arm through four mental tasks,” *Neurocomputing*, vol. 151, pp. 116–121, 2015.
- [30] L. Wang and X. Wu, “Classification of Four-Class Motor Imagery EEG Data Using Spatial Filtering,” *Engineering, 2008. ICBBE 2008. The 2nd*, no. 070412038, pp. 2153–2156, 2008.
- [31] H. S. Kim, M. H. Chang, H. J. Lee, and K. S. Park, “A comparison of classification performance among the various combinations of motor imagery tasks for brain-computer interface,” *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*, pp. 435–438, 2013.
- [32] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, “Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b,” *Frontiers in Neuroscience*, vol. 6, no. MAR, pp. 1–9, 2012.

- [33] S. Ge, R. Wang, and D. Yu, "Classification of four-class motor imagery employing single-channel electroencephalography." *PloS one*, vol. 9, no. 6, p. e98019, jan 2014.
- [34] A. Schlögl, F. Lee, H. Bischof, and G. Pfurtscheller, "Characterization of four-class motor imagery EEG data for the BCI-competition 2005." *Journal of neural engineering*, vol. 2, no. 4, pp. L14–L22, 2005.
- [35] D. Devlaminck, B. Wyns, L. Boullart, P. Santens, and G. Otte, "Brain-Computer Interfaces : from theory to practice," *Neural Networks*, no. April, pp. 22–24, 2009.
- [36] J. S. Brumberg, F. H. Guenther, and P. R. Kennedy, *Brain-Computer Interface Research*, 2013. [Online]. Available: <http://link.springer.com/10.1007/978-3-642-36083-1>
- [37] S. Sanei and J. a. Chambers, *EEG Signal Processing*, 2007, vol. 1, no. 11.
- [38] R. M. Howard, *Principles of Random Signal Analysis and Low Noise Design*, 2002.
- [39] O. Çaglayan, "A portable and embedded SSVEP BCI system: emBCI," Ph.D. dissertation, 2014.
- [40] D. J. Krusienski, D. J. Mcfarland, and J. R. Wolpaw, "An Evaluation of Auto-regressive Spectral Estimation Model Order for Brain-Computer Interface Applications.pdf," pp. 1323–1326, 2006.
- [41] H. Ocak, "Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy," *Expert Systems with Applications*, vol. 36, no. 2 PART 1, pp. 2027–2036, 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.eswa.2007.12.065>
- [42] P. Ghorbanian, D. M. Devilbiss, A. J. Simon, A. Bernstein, T. Hess, and H. Ashrafiun, "Discrete wavelet transform EEG features of Alzheimer'S disease in activated states," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 2937–2940, 2012.
- [43] H. Olkkonen, *Discrete Wavelet Transforms: Algorithms and Applications.*, 2011.

- [44] A. C. Gonzalez, “Procesamiento embebido de señales cerebrales relacionadas con la imaginación de movimientos para aplicaciones de BCI Procesamiento embebido de señales cerebrales relacionadas con la imaginación de movimientos para aplicaciones de BCI,” 2013.
- [45] I. Computing, “Mental EEG Analysis based on Independent Component Analysis,” no. 60271024, 2003.
- [46] N. Ye, X. Wang, and Y. Sun, “Independent component analysis and time-frequency method for noisy EEG signal analysis,” *International Conference on Signal Processing Proceedings, ICSP*, vol. 4, no. 3, 2007.
- [47] Y. Padmasai, K. Subba Rao, C. Raghavendra Rao, S. Sita Jayalakshmi, K. SubbaRao, C. R. Rao, and S. S. Jayalakshmi, “Analysis of EEG Using Principal Component Approach,” *Proc. 14th IEEE Int. Conf. on Electronics, Circuits and Systems*, pp. 134–137, 2007.
- [48] Y. Wang, S. Gao, and X. Gao, “Common Spatial Pattern Method for Channel Selection in Motor Imagery Based Brain-computer Interface,” *Engineering In Medicine And Biology*, vol. 5, pp. 5392–5395, 2005.
- [49] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller, “Optimal spatial filtering of single trial EEG during imagined hand movement,” *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 4, pp. 441–446, 2000.
- [50] Kai Keng Ang, Zheng Yang Chin, Haihong Zhang, and Cuntai Guan, “Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface,” *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 2390–2397, 2008.
- [51] L. F. Nicolas-Alonso, R. Corralejo, D. Alvarez, and R. Hornero, “Analytic common spatial pattern and adaptive classification for multiclass motor imagery-based BCI,” *International IEEE/EMBS Conference on Neural Engineering, NER*, pp. 1084–1087, 2013.
- [52] P. A. Flach and N. Lachiche, “Naive Bayesian classification of structured data,” *Machine Learning*, vol. 57, no. 3, pp. 233–269, 2004.
- [53] S. Bhattacharyya, A. Khasnobish, A. Konar, D. N. Tibarewala, and A. K. Nagar, “Performance analysis of left/right hand movement classification from EEG sig-

- nal by intelligent algorithms,” *IEEE SSCI 2011 - Symposium Series on Computational Intelligence - CCMB 2011: 2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain*, pp. 1–8, 2011.
- [54] C. M. Bishop, “Neural Networks for Pattern Recognition,” p. 504, 1995.
- [55] S. Vani, S. Member, and G. R. Suresh, “Performance Analysis of Lifting based DWT and MLPNN for Epilepsy seizure From EEG.”
- [56] Odroid HardKernel, “Odroid Magazine,” 2016. [Online]. Available: <http://magazine.odroid.com/es/>
- [57] M. Selvalakshmi and R. Selvarasu, “Monitoring and Controlling of Mobile Robot Via Internet Through Raspberry PI Board,” 2016.
- [58] P. H. Kulkarni, P. D. Kute, and V. N. More, “IoT Based Data Processing for Automated Industrial Meter Reader using Raspberry Pi,” pp. 107–111, 2016.
- [59] P. Suja and S. Tripathi, “Real-Time Emotion Recognition from Facial Images using Raspberry Pi II,” pp. 666–670, 2016.
- [60] G. Pfurtscheller and A. Schlögl, “BCI Competition III.” [Online]. Available: <http://bbci.de/competition/iii/>
- [61] R. P. Foundation, “Raspbian Jessie,” 2016. [Online]. Available: www.raspberrypi.org/downloads/raspbian/
- [62] O. HardKernel, “Linux Ubuntu 16.04,” 2016. [Online]. Available: <http://odroid.com/dokuwiki/doku.php?id=en:xu3{ }release{ }linux{ }ubuntu>
- [63] R. P. N. Rao, *Brain-computer interfacing: An introduction*, 2011.
- [64] Z. J. Koles, M. S. Lazar, and S. Z. Zhou, “Spatial patterns underlying population differences in the background EEG,” *Brain Topography*, vol. 2, no. 4, pp. 275–284, 1990.
- [65] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K. R. Müller, “Optimizing spatial filters for robust EEG single-trial analysis,” *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41–56, 2008.
- [66] W. Wu, X. Gao, and S. Gao, “One-Versus-the-Rest(OVR) Algorithm: An Extension of Common Spatial Patterns(CSP) Algorithm to Multi-class Case.” *Conference proceedings : Annual International Conference of the IEEE Engineering*

in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference, vol. 3, pp. 2387–2390, 2005.

- [67] The PyWavelets Developers, “PyWavelets - Wavelet Transforms in Python.” [Online]. Available: <http://www.pybytes.com/pywavelets/>
- [68] “Scikit Learn,” 2016. [Online]. Available: <http://scikit-learn.org/stable/modules/svm.html>

