

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Interfaz para el Control de Sistemas Robóticos utilizando
Combinaciones de Señales Biomédicas**

Trabajo de graduación presentado por Roberto José Cáceres García
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



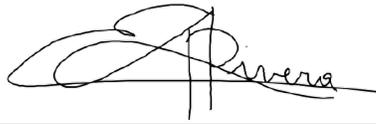
**Interfaz para el Control de Sistemas Robóticos utilizando
Combinaciones de Señales Biomédicas**

Trabajo de graduación presentado por Roberto José Cáceres García
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

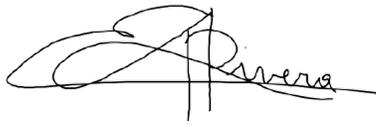
2022

Vo.Bo.:



(f) _____
Dr. Luis Alberto Rivera Estrada

Tribunal Examinador:



(f) _____
Dr. Luis Alberto Rivera Estrada



(f) _____
M. Sc. Carlos Esquit



(f) _____
MAEB. Pablo Mazariegos

Fecha de aprobación: Guatemala, 23 de Junio de 2022.

Esta investigación nace de mi interés en la investigación científica y en el campo de la ingeniería biomédica. También a la oportunidad que se presentó en la Universidad del Valle para poder pertenecer a esta línea de investigación. El interés de desarrollar interfaces que permitan la recolección y clasificación de señales biomédicas, así como experimentar con la combinación de estas señales, surge principalmente para contribuir en el estudio de las señales biomédicas contando con bases de datos propias de la Universidad del Valle. Pudiendo así fomentar el desarrollo de sistemas robóticos para la asistencia médica utilizando las interfaces clasificadoras de señales. A través de este trabajo, espero aportar herramientas útiles para investigaciones futuras y poder así aportar un poco a la sociedad.

Agradezco principalmente a Dios, por el amor y la sabiduría con la que me acompañó en este proceso. A Ferlandy y Dalila, mis padres, por siempre creer en mí, por su amor incondicional. A mis abuelos que siempre han creído en cada proyecto que he emprendido. A mis hermanos, tíos, primos y amigos que siempre tuvieron palabras de apoyo y la confianza siempre puesta en mí. Al Centro Universitario Ciudad Vieja, donde siempre creyeron en mí, me apoyaron en cada aspecto de mi vida y me formaron como persona y profesional. Gracias Ruth, Hee, Diego, Edin, Ronaldo, Javier, Olga, Annie, Alex, Hans, José David, Miguel, Josué, Daniela, Paola, Francis y a todas esas personas que fueron parte de esto.

Por último agradezco a toda la comunidad de la Universidad del Valle de Guatemala. A todos los profesores y personal que aportó en mi formación y aprendizaje. Un especial agradecimiento a mi asesor el Dr. Luis Alberto Rivera por su pasión y dedicación, cada uno de sus consejos, experiencias y críticas permitieron la culminación de este trabajo de graduación.

Prefacio	v
Lista de figuras	xii
Lista de cuadros	xiv
Resumen	xvi
Abstract	xvii
1. Introducción	1
2. Antecedentes	3
3. Justificación	5
4. Objetivos	7
4.1. Objetivo general	7
4.2. Objetivos específicos	7
5. Alcance	9
6. Marco teórico	11
6.1. Señales electromiográficas (EMG)	11
6.1.1. Electromiografía	11
6.1.2. Posición de electrodos en los músculos	11
6.1.3. Características en el dominio del tiempo	12
6.1.4. Características en el dominio del tiempo-frecuencia	14
6.2. Señales electroencefalográficas (EEG)	14
6.2.1. Electroencefalografía	14
6.2.2. Ondas cerebrales	14
6.2.3. Sistema de posicionamiento 10-20	15
6.2.4. Características en el dominio del tiempo	15
6.2.5. Características en el dominio de la frecuencia	16

6.3. Aprendizaje automático	17
6.3.1. Redes neuronales	17
6.3.2. Máquina de soportes de vectores	18
6.4. Interacción Hombre-Máquina (HMI)	18
6.5. Interfaz Cerebro-Computadora (BCI)	19
6.6. Robot R17	19
7. Colecciones de señales EEG y EMG	21
7.1. Base de datos pública de 11 tareas de movimiento intuitivas desde una sola extremidad superior	21
7.1.1. Paradigma experimental	22
7.1.2. Procesamiento de señales	23
7.1.3. Extracción de características	24
8. Clasificadores para colección de señales	25
8.1. Clasificadores para señales EMG	25
8.1.1. Redes neuronales señales EMG (2 clases y 1 Sujeto)	25
8.1.2. Máquina de soporte de vectores señales EMG (2 clases y 1 sujeto)	29
8.2. Clasificadores señales EEG	30
8.2.1. Redes neuronales señales EEG (2 clases y 1 sujeto)	30
8.2.2. Máquina de soporte de vectores señales EEG	32
8.3. Clasificadores señales EEG-EMG	33
8.3.1. Redes neuronales combinación de señales EMG-EEG (2 clases)	33
8.3.2. Máquina de soporte de vectores señales combinación de señales EMG- EEG (2 clases)	34
8.3.3. Redes neuronales combinación de señales EMG-EEG (4 clases)	35
8.3.4. Máquina de soporte de vectores señales combinación de señales EMG- EEG (4 clases)	36
9. Adquisición y recopilación de señales EMG	37
9.1. Dispositivo de adquisición de señales EMG	37
9.1.1. Primeros acercamientos a dispositivo bitalino	38
9.2. Adquisición y procesamiento de señales EMG	38
9.2.1. Algoritmo de adquisición	38
9.2.2. Procesamiento de señal	40
9.2.3. Algoritmo de identificación de actividad	40
9.3. Interfaz para la recolección de datos	40
9.4. Paradigma experimental para la recolección de datos	41
10. Interfaz clasificadora de señales EMG	43
10.1. Criterios y métricas de evaluación para la interfaz EMG de fases previas	43
10.2. Clasificación de señales EMG	44
10.2.1. <i>Classification Learner MATLAB</i>	44
10.2.2. <i>Neuronal net Fitting MATLAB</i>	44
10.2.3. Redes neuronales en Python	45
10.2.4. Resultados de entrenamientos	45
10.2.5. Interfaz clasificadora de señales EMG	48

11. Adquisición y recopilación de señales EEG	51
11.1. Dispositivo de adquisición de señales EEG	51
11.1.1. Primeros acercamientos a dispositivo cyton board	51
11.2. Adquisición y procesamiento de señales EEG	52
11.2.1. Algoritmo de adquisición	52
11.2.2. Procesamiento de señal	53
11.2.3. Algoritmo para la identificación de actividad	54
11.3. Interfaz para la recolección de datos EEG	54
11.4. Paradigma experimental para la recolección de datos EEG	54
12. Interfaz clasificadora de señales EEG	57
12.1. Criterios y métricas de evaluación para la interfaz EEG de fases previas	57
12.2. Clasificación de señales EEG	58
12.2.1. Resultados de entrenamientos	58
12.2.2. Interfaz clasificadora de señales EEG	63
13. Combinación de señales EMG y EEG	65
13.1. Interfaz recolectora de datos EEG y EMG	65
13.2. Paradigma experimental primer enfoque	66
13.3. Clasificación de señales EEG-EMG, primer enfoque	67
13.3.1. Resultados de entrenamientos	67
13.4. Paradigma experimental segundo enfoque	70
13.5. Clasificación de señales EEG-EMG, segundo enfoque	70
13.5.1. Resultados de entrenamientos	70
13.6. Paradigma experimental, tercer enfoque	71
13.7. Clasificación de señales EEG-EMG, tercer enfoque	72
13.7.1. Resultados de entrenamientos	72
13.7.2. Interfaz clasificadora de señales EEG-EMG	77
14. Conclusiones	79
15. Recomendaciones	81
16. Bibliografía	83
17. Anexos	87
17.1. Paradigma experimental señales EMG	87
17.2. Paradigma experimental señales EEG-EMG	89

Lista de figuras

1. Sistema de posicionamiento 10-20 [12].	15
2. Sistema de posicionamiento 10-20 [15].	18
3. Robot R17 [19].	19
4. Robot R17 grados de libertad [19].	20
5. Posición de electrodos en el estudio [20].	22
6. Movimientos realizados en la base de datos [20].	23
7. Movimientos realizados por los sujetos, detallados [20].	23
8. Matriz de confusión para RNA con 9 características y 1 canal	26
9. Matriz de confusión para RNA con 4 características y 3 canales	27
10. Matriz de confusión para RNA con 5 características y 4 canales	28
11. Matriz de confusión para SVM con 8 características y 1 canal	29
12. Matriz de confusión para SVM con 8 características y 4 canales	30
13. Matriz de confusión para redes neuronales con 2 características y un canal.	31
14. Matriz de confusión para redes neuronales con 5 características y un canal.	31
15. Matriz de confusión para redes neuronales con 4 características y un canal.	32
16. Matriz de confusión para redes neuronales con 8 características y 4 canales.	32
17. Matriz de confusión para redes neuronales con 8 características, 2 canales y 2 clases	33
18. Matriz de confusión para SVM con 8 características, 2 canales y 2 clases	34
19. Matriz de confusión para redes neuronales con 8 características, 2 canales y 4 clases	35
20. Matriz de confusión para SVM con 8 características, 2 canales y 4 clases	36
21. Diagrama de bloques interconectados dispositivo bitalino [23].	37
22. Algoritmo de adquisición de datos bitalino	39
23. Interfaz recolectora de datos	41
24. Paradigma experimental	42
25. Learning App	44
26. Neuronal net fitting	45
27. Interfaz EMG	48
28. Clase puño arriba	49

29. Clase palma arriba	49
30. Paquete electro-cap [29]	52
31. Cyton board [28]	52
32. Algoritmo de adquisición de señales EEG	53
33. Interfaz recolectora de datos EEG	54
34. Paradigma experimental	55
35. Interfaz clasificadora EEG	63
36. Interfaz recolectora de datos EEG-EMG	66
37. Paradigma experimental primer enfoque	66
38. Paradigma experimental segundo enfoque	70
39. Paradigma experimental tercer enfoque	72
40. Resultados de los pesos en de pruebas de chi-cuadrado.	76
41. Interfaz clasificadora EEG-EMG	78
42. Grabación clase puño arriba	87
43. Grabación clase puño arriba.	88
44. Grabación clase palma arriba señales EEG-EMG.	89
45. Proyecto fisiología 2 utilizando interfaz recolectora de señales EMG (Grupo 3).	89
46. Primera prueba interfaz clasificadora señales EEG-EMG.	90
47. Proyecto fisiología 2 utilizando interfaz clasificadora de señales EMG (Grupo 1).	91
48. Proyecto fisiología 2 utilizando interfaz clasificadora de señales EMG (Grupo 2).	91
49. Posicionamiento de electrodos y toma de datos (Grupo2).	92
50. Posicionamiento de electrodos y toma de datos (Grupo2).	93
51. <i>Live script</i> realizado en el software MATLAB para ayudar a la selección de características.	93
52. Manual para el uso del dispositivo bitalino y uso de las interfaces para señales EMG.	94
53. Manual para el uso del dispositivo electrocap y uso de las interfaces para señales EEG.	95

Lista de cuadros

1. Tabla de ondas cerebrales.	15
2. Tabla de lóbulos.	15
3. Cuadro de resultados clasificación con algoritmos distintos de clasificación señales EMG.	46
4. Cuadro de resultados clasificación con algoritmos distintos de clasificación señales EMG.	46
5. Cuadro de resultados clasificación con algoritmos distintos de clasificación señales EMG.	47
6. Cuadro de resultados clasificación con algoritmos distintos de clasificación señales EMG.	47
7. Cuadro de resultados clasificación primera sesión con algoritmos distintos de clasificación señales EEG.	59
8. Cuadro de resultados clasificación segunda sesión con algoritmos distintos de clasificación señales EEG.	60
9. Tabla de resultados clasificación tercera sesión con algoritmos distintos de clasificación señales EEG.	61
10. Cuadro de resultados clasificación cuarta sesión con algoritmos distintos de clasificación señales EEG.	62
11. Cuadro de resultados clasificación primera sesión con algoritmos distintos de clasificación señales EEG-EMG.	68
12. Cuadro de resultados clasificación segunda sesión con algoritmos distintos de clasificación señales EEG-EMG.	69
13. Cuadro de resultados clasificación primera con algoritmos distintos de clasificación señales EEG-EMG.	71
14. Cuadro de resultados clasificación primera sesión con algoritmos distintos de clasificación señales EEG-EMG.	73
15. Cuadro de resultados clasificación segunda sesión con algoritmos distintos de clasificación señales EEG-EMG.	74
16. Cuadro de resultados clasificación tercera sesión con algoritmos distintos de clasificación señales EEG-EMG.	75

17. Cuadro de resultados clasificación segunda sesión con algoritmos distintos de clasificación señales EEG-EMG.	77
--	----

Este trabajo consistió en el desarrollo de tres interfaces biomédicas para la recolección de señales electromiográficas (EMG), señales electroencefalográficas (EEG) y ambas señales combinadas. Estas tres interfaces se desarrollaron con el fin de tener una forma asequible para poder obtener muestras y así poder contar con bases de datos propias. Estas bases de datos pueden ser utilizadas para el desarrollo de modelos de clasificación o post-procesamiento de las señales para su estudio. Al contar con bases de datos propias se procedió a desarrollar otras tres interfaces biomédicas, esta vez para el control de dispositivos robóticos por medio de la adquisición, procesamiento y análisis de las señales EMG y EEG. Usando extracción de características a las señales procesadas se realizaron diversos entrenamientos con técnicas de aprendizaje automático para obtener modelos de clasificación de clases. Con los vectores de características se realizaron entrenamientos mediante máquina de soporte de vectores (SVM) y redes neuronales (RN). Se hicieron diversas pruebas con los algoritmos hasta obtener los mejores resultados para la clasificación de señales EMG, EEG y ambas señales combinadas. Todos los clasificadores con los que se lograron buenos rendimientos se probaron en las interfaces de cada señal. Cada interfaz interactúa directamente con el robot R17 de la Universidad del Valle de Guatemala, dependiendo del gesto y como lo clasifiquen los algoritmos, se enviará cierto comando predeterminado al robot.

Uno de los objetivos principales del proyecto era la combinación de señales al momento de clasificar gestos o señales. Se combinó las señales de dos formas diferentes. La primera forma fue realizando un entrenamiento con 4 clases, que de cierta manera funcionaban como dos clasificadores distintos, ya que dos clases eran muy distintivos para las señales EMG y las otras dos eran muy distintivas para las señales EEG logrando una clasificación con rendimientos superiores al 91 %. La segunda forma era grabando dos clases distintivas para señales EMG utilizando las señales EEG como una señal auxiliar y así brindar más peso a la clasificación, se determinó que en efecto al combinar las señales se logra un mejor rendimiento, obteniendo un 87 % de rendimiento, mientras solo con las características de las señales EMG se logró un 85 %

En el futuro se espera que estas interfaces puedan ser utilizadas para la recolección de datos y creación de bases de datos que aporten material para futuras investigaciones utilizando técnicas mejoradas en algoritmos de clasificación y también a las investigaciones relacionadas con los dispositivos de asistencia médica utilizando estas señales bioeléctricas.

This work consisted in the development of three biomedical interfaces for the collection of electromyographic (EMG) signals, electroencephalographic (EEG) signals and a combination of both. These interfaces were developed in order to have an affordable way to obtain samples and thus be able to have an own database.

These databases can be used for the development of classification or post-processing models of the signals for their study. Using this data, another three biomedical interfaces were developed, this time for the control of robotic devices through the acquisition, processing and analysis of EMG and EEG signals. Using feature extraction to the processed signals, various trainings were performed with machine learning techniques to obtain class classification models. With the feature vectors, training was performed using support vector machines (SVM) and neural networks (RN). Various tests were carried out with the algorithms until the best results were obtained for the classification of EMG and EEG signals and both signals combined. All classifiers with good performance were tested at the interfaces of each signal. Each interface interacts directly with the R17 robot from the Universidad del Valle de Guatemala, depending on the gesture and how the algorithms classify it, a certain predetermined command will be sent to the robot.

One of the main objectives of the project was the combination of signals when classifying gestures or signals. The signals were combined in two different ways. The first way was to carry out training with four classes, which in a certain way functioned as two different classifiers, since two classes were very distinctive for EMG signals and the other two were very distinctive for EEG signals, achieving a classification with performances higher than 91 %. The second way was recording two distinctive classes for EMG signals using EEG signals as an auxiliary signal and thus provide more weight to the classification, it was determined that in effect combining the signals achieves better performance, obtaining 87 % of performance, while only with the characteristics of the EMG signals an 85 % was achieved

In the future, it is expected that these interfaces can be used for data collection and creation of databases that provide material for future research using improved techniques in classification algorithms and also for research related to medical assistance devices using these signals. bioelectric.

El estudio de las señales biomédicas ha logrado numerosos aportes a la ciencia en general, logrando una diversidad de dispositivos de asistencia médica, con el fin de mejorar el bienestar físico de personas que padecen de discapacidades motrices. Además ha permitido acercamientos a poder utilizar chips cerebrales con fines de controlar enfermedades neurológicas. Las señales tanto electromiográficas como electroencefalográficas, de superficie juegan un papel importante de los estudios para poder realizar todos estos avances. Esto de una forma segura, ya que se realizan de forma no invasiva.

En esta investigación se planteó el desarrollo tres interfaces biomédicas que permitan el control de sistemas robóticos físicos disponibles en la Universidad del Valle de Guatemala. Esto mediante la adquisición y filtrado en tiempo real de las señales electroencefalográficas, electromiográficas y de ambas señales combinadas. El trabajo espera sentar bases para el desarrollo de trabajos donde se utilicen algoritmos de aprendizaje automático ya que además de las tres interfaces para el control de sistemas robóticos, se desarrollan tres interfaces para la adquisición de datos de señales electroencefalográficas, electromiográficas y ambas señales combinadas. Esto permitirá ahorrar curvas de aprendizaje a los usuarios para indagar más en el estudio de las señales como tal.

El trabajo constó de varias etapas principales. Se hicieron pruebas de clasificación con algoritmos de aprendizaje automático con una base de datos pública, para poder experimentar con diferentes vectores de características y así poder contar con otra referencia de resultados. Para cada tipo de señal, se realizó una etapa de adquisición y procesamiento de señal, para cada señal se presenta la forma en que se adquieren los datos, ya que se utilizaron dos dispositivos distintos para esta tarea, electro cap para señales EEG y dispositivo bitalino para señales EMG. Se desarrollaron los filtrados y conversiones de cada tipo de señal. Para cada señal se desarrolló una etapa de detección de actividad y se explica los métodos utilizados. Se desarrollaron los paradigmas experimentales que se utilizaron para grabar cada clase en cada entrenamiento. Se presentan las interfaces recopiladoras de datos, donde se integran todas las etapas que tienen que ver con como se reciben las señales. Al tener una base de datos propia, se explica el proceso de extracción de características y todos

los algoritmos de clasificación que se utilizaron, también se comparan los rendimientos de cada algoritmo. Por último, al escoger los entrenamientos más óptimos se procedió al desarrollo de las interfaces clasificadoras que le envían comandos al robot R17, para verificar y comprobar el funcionamiento de las interfaces.

Por último, cabe mencionar que este trabajo tiene como objetivo sentar una base para la experimentación con señales biomédicas utilizando técnicas de aprendizaje automático. Esto logra acotar curvas de aprendizaje a la continuación de este tipo de trabajos de investigación, esto se logra gracias a que las interfaces están diseñadas para que sean completamente útiles para los usuarios de las mismas. Se espera que en un futuro estas interfaces sean utilizadas y puedan contribuir a toda la comunidad científica de la Universidad del Valle de Guatemala, así como al estudio de señales biomédicas utilizadas en dispositivos de asistencia médica.

A lo largo de las últimas décadas el estudio de las señales bioeléctricas han determinado un crecimiento en la forma de afrontar los problemas relacionados a la carencia de ciertas partes del cuerpo, esto ha mejorado la forma de vida que por cualquier causa no cuenten con alguna parte de su cuerpo. En su mayoría las investigaciones y trabajos que se han hecho en base a estos estudios toman a las señales Electroencefalográficas (EEG) y a las señales Electromiográficas (EMG) como base para estudios enfocados a las prótesis en humanos. Esto se debe a lo práctico de los métodos para estudiar las señales ya que este es un método no invasivo y permite al investigador realizar pruebas sin la necesidad de contar con supervisión médica.

Tanto para señales EEG como para señales EMG se han realizado estudios sobre la clasificación y extracción de características de cada una de las señales. La extracción de características en señales EMG tiene que tratar de ser lo menos redundante posible al momento de buscar un vector de características para la clasificación de las señales como se muestra en [1] donde se tratan de agrupar las señales por medio de propiedades matemáticas de las señales. Las señales específicamente se pueden agrupar en cuatro grupos: energía y complejidad, frecuencia, modelo de predicción y dependencia del tiempo, además de encontrar las características en el dominio de la frecuencia. Para las señales EEG sucede lo mismo al momento de clasificar las señales, aparte de la clasificación de señales por sus características se tiene que escoger un método para la extracción de estas señales como en [2]. En este se explican diversos métodos para la extracción de características como el método de descomposición empírica.

Hay una diversa cantidad de investigaciones con respecto a la clasificación de cada una de los tipos de señales biomédicas, por lo que se tiene que indagar más con respecto a trabajos que utilicen diferentes tipos de señales biomédicas, pero que logren combinarlas de modo que éstas puedan dar un mejor control a ciertas tareas que se necesiten implementar. En [3] se explora la fusión de ambas señales módulo de detección para identificar movimientos en las extremidades inferiores. Las señales se clasificaron en paralelo y en cascada mientras que el rendimiento de la combinación de las señales fue evaluado por medio de la precisión en la predicción y el tiempo computacional del procesamiento.

En la Universidad del Valle de Guatemala se ha investigado la clasificación y selección tanto de señales EMG como EEG para traducirlas en comandos que puedan aplicarse en sistemas robóticos como en [4] y [5]. Utilizando técnicas de aprendizaje automático, se lograron extraer las características de ambos tipos de señales y se logró desarrollar dos interfaces capaces de traducir los dos tipos de señales en comandos para sistemas robóticos simulados por computadora. Estas fases previas cuentan con la particularidad de haberse desarrollado en una época singular, ya que no se logró tener un acceso total a todas las herramientas y equipo con los que cuenta la universidad, debido a la pandemia del COVID-19. Por esta razón las interfaces presentan carencias, no en cuanto a la calidad de los clasificadores, sino a la poca cantidad de pruebas físicas que se lograron realizar. Esto es un punto de partida para el presente trabajo, ya que estos clasificadores se tienen que poder implementar en sistemas robóticos físicos.

A lo largo de los años la innovación y desarrollo en la tecnología han permitido que la vida de millones de personas cambie, logrando dar una calidad de vida mayor, así como facilitar el día a día de las personas. Todos los dispositivos electrónicos a lo largo de los años han cambiado su forma de interactuar con los humanos, pasando desde un sistema de botones convencionales hasta los asistentes inteligentes de diferentes *smartphones*, el desarrollo en la HMI no se ha dado en solo estos aspectos. Se pueden encontrar muchos trabajos de la comunidad científica que tratan el estudio de las señales biomédicas implementadas por medio de interfaces para poder controlar sistemas robóticos o prótesis.

Este proyecto busca a través de la extracción de características de las señales biomédicas como lo son las señales EEG y EMG lograr implementar un sistema de control combinado, es decir, poder tener un control más fiel sobre los sistemas robóticos como el brazo articulado R17 teniendo ahora dos referencias, las cuales son las dos señales utilizadas para el análisis en tiempo real. El presente trabajo también busca la implementación de las fases previas en sistemas físicos y no solo en simulaciones. Es importante mencionar esto ya que las fases previas a este trabajo que implementaban interfaces que traducían las señales biomédicas provenientes de movimientos y gestos preestablecidos en comandos para robots, haciendo que estos realizarán una acción ya establecida. Por la pandemia del COVID-19 probar las interfaces en los sistemas robóticos con los que contaba la Universidad del Valle fue imposible, por lo que en este trabajo se busca la implementación de estas interfaces en sistemas robóticos físicos. También se busca poder evaluar y optimizar estas interfaces, pudiendo así plantear oportunidades de mejora en el entrenamiento de los clasificadores que se utilizan en estas interfaces.

La forma de lograr el control de prótesis o sistemas robóticos por medio de señales combinadas, es creando clasificadores con características combinadas. El objetivo de crear una interfaz que combine señales biomédicas es poder contribuir al estudio de las señales en aplicaciones para prótesis de personas que tenga, por ejemplo, amputaciones que lleguen hasta el codo. Esto implica que, al no tener una parte de su brazo, al utilizar señales EEG las señales no logren funcionar bien en los clasificadores por falta de correlación entre las

características. Las señales se combinan de manera que las señales EEG funcionen como un complemento a las señales EMG para identificar de forma correcta los movimientos de la mano o muñeca. Esto funcionaría de buena manera, ya que las señales EEG no son dependientes de las condiciones de amputación con las que pueda contar un paciente. En otras palabras, este trabajo busca que las señales EEG funcionen como un tipo de señal auxiliar a las señales EMG, esto con el objetivo de poder mejorar los patrones de reconocimiento. Este trabajo puede sentar las bases para futuros estudios en los que se pueda experimentar directamente en el diseño de prótesis que estén enfocadas en pacientes con condición de amputación.

4.1. Objetivo general

Mejorar las interfaces biomédicas para sistemas robóticos simulados desarrolladas en fases previas, desarrollar una interfaz que permita el uso una o varias señales biomédicas, y adaptar las interfaces para controlar sistemas físicos.

4.2. Objetivos específicos

- Desarrollar interfaces que permitan la detección, muestreo y almacenamiento de señales EEG y EMG.
- Mejorar la interfaz biomédica que utiliza señales EEG para controlar robots simulados desarrollada en la fase anterior, y adaptarla a sistemas físicos de la UVG.
- Mejorar la interfaz biomédica que utiliza señales EMG para controlar robots simulados desarrollada en la fase anterior, y adaptarla a sistemas físicos de la UVG.
- Desarrollar una interfaz que permita el control de los sistemas físicos por medio de combinaciones de señales EEG y EMG.

Este proyecto forma parte de la segunda fase de la línea de investigación de interfaces biomédicas. Tiene como fin el aplicar técnicas de clasificación de aprendizaje automático para poder implementar estos modelos en interfaces que permitan el control de sistemas robóticos físicos. Específicamente se desea controlar los sistemas robóticos con los que cuenta la Universidad del Valle de Guatemala. Más específicamente está investigación consiste en el desarrollo de 6 interfaces biomédicas. Las primeras tres interfaces son las interfaces recolectoras de datos de señales EMG, EEG y EEG-EMG. Las otras tres interfaces son las interfaces clasificadoras de señales EMG, EEG y EEG-EMG.

El proyecto está dividido básicamente por los tipos de señal y para cada tipo de señal existen cuatro módulos principales. El primero es la conexión con los dispositivos y la adquisición y procesamiento de las señales, donde se puede escoger la banda de frecuencia a filtrar tanto para las señales EMG como EEG. El siguiente módulo es el desarrollo de algoritmos de detección de actividad, donde se plantean para cada tipo de señal, formas de detectar actividad, esto para evitar enviarle comandos a los sistemas robóticos todo el tiempo. El tercer módulo es la recolección de los datos, este consta básicamente en unir los módulos 1 y 2 en una interfaz recolectora de datos, donde se pueden seleccionar las características que se obtendrán de cada muestra, así como la data en bruto de cada una de las muestras. El siguiente módulo es la clasificación y comparación de modelos de aprendizaje automático, donde se entrenó variando los vectores de características y los algoritmos de entrenamiento, permitiendo esto poder obtener los mejores modelos para la clasificación. El último módulo consta en el desarrollo de las interfaces clasificadoras donde se pusieron a prueba los modelos escogidos y el funcionamiento en conjunto con el robot R17.

El desarrollo de este proyecto sienta las bases para el desarrollo y optimización de próximas investigaciones, ya que las interfaces están en una etapa donde el uso es factible por parte de los usuarios que necesiten experimentar con estas.

6.1. Señales electromiográficas (EMG)

Este tipo de señales se generan por la contracción de fibras musculares y en su mayoría los métodos para medir estas señales son métodos no invasivos como la aplicación superficial de los electrodos en la piel [6]. Estas señales son producidas por la contracción muscular, esta contracción muscular provoca actividad eléctrica, que se puede traducir en una señal, estas señales son esencia unidimensionales y esto es lo que permite la extracción de características para el reconocimiento de patrones [6].

6.1.1. Electromiografía

Es una técnica que puede evaluar la función muscular midiendo la diferencia de potencial generada por los músculos en la superficie de la piel, esto se logra por medio de electrodos que se pueden localizar en la superficie muscular. Una unidad motora se define por una moto neurona y todas las fibras que esta inerva. Cuando un impulso nervioso es enviado por el cerebro a través de las interneuronas hasta llegar a la moto neurona, este impulso es mejor conocido como potencial de acción, este activa las fibras musculares y provoca la activación de las células musculares y esto pone en funcionamiento la maquinaria contráctil del músculo, esto se traduce a cambios en las propiedades eléctricas de la membrana de las células musculares y es donde se puede medir la señal por medio de los electrodos. En otras palabras las señales EMG son las sumas de los potenciales de acción de las unidades motoras que se detectan a través de los electrodos [7].

6.1.2. Posición de electrodos en los músculos

La distancia interelectrodos se define como la distancia centro a centro del área conductiva de electrodos. Como se menciona en [8] la medición de señales EMGs necesita de

un conjunto de por lo menos 2 electrodos individuales, en la mayoría de casos se utiliza una única señal EMG como un diferencial de potencial eléctrico entre electrodos colocados separados en el musculo.

- **Forma del electrodo:** En [8] no se recomienda un estándar en la forma del electrodo que sea defendible ya que no se espera una influencia importante en la señal EMG porque se tomen diferentes formas en estos, sin embargo sí se menciona que se tiene que dejar claro que forma se está utilizando.
- **Tamaño de electrodos:** El tamaño del electrodo varía entre 1 mm hasta un poco más de 2 cm, el tamaño del electrodo es mucho más importante que la forma ya que el tamaño si influye en la señal directamente.
- **Posición del sensor:** Esta posición se deriva del principio: con respecto a la ubicación longitudinal en dirección de la fibra, en [8] se recomienda colocar el sensor en la mitad de la zona más distal de la placa terminal motora y el tendón distal y con respecto a la ubicación transversal del sensor en el músculo, se recomienda ubicar el sensor en la superficie lejos del borde con otras subdivisiones de manera que la distancia a otros músculos sea maximizada.

6.1.3. Características en el dominio del tiempo

- **Integración de la señal** Este valor se obtiene calculando la sumatoria de los valores absolutos de una señal EMG. Puede ser tratada como un estimador de poder.

$$IEMG_k = \sum_{i=1}^N |x_i| \quad (1)$$

- **Media del valor absoluto** Estima la media del valor absoluto, sumando el valor absoluto de x_i en un segmento k y dividido por la longitud del segmento.

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

- **Media cuadrática** Se modela como un proceso aleatorio gaussiano modulado en amplitud cuyo RMS es relacionado con la fuerza constante y la contracción no fatigante.

$$RMS_k = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (3)$$

- **Varianza** Está dada por:

$$VAR_k = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (4)$$

- **Waveform length**

Es la longitud acumulada en forma de onda sobre un segmento escogido. Los valores resultantes indican una medida de amplitud, frecuencia y duración en un solo valor.

$$WL_k = \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (5)$$

- **Zero crossings** Es el número de veces que la forma de onda cruza cero, es bueno incluir un umbral para detectar los ceros, ya que por el ruido pueden haber ceros falsos.

$$\{x_i > 0 \text{ and } x_{i+1} < 0\} \text{ or } \{x_i < 0 \text{ and } x_{i+1} > 0\} \text{ and } |x_i - x_{i+1}| \geq \epsilon \quad (6)$$

- **Willison Amplitude (WAMP)**

Calcula el número de veces que el valor absoluto de la diferencia entre la amplitud de la señal EMG de dos muestras consecutivas supera un umbral predeterminado.

$$WAMP_k = \sum_{i=1}^{N-1} f(|x_i - x_{i+1}|) \quad (7)$$

$$f(x) = \begin{cases} 1 & x > \epsilon \\ 0 & \text{de lo contrario} \end{cases}$$

- **Simple Square Integral (SSI)** Usar la energía de una señal EMG como una característica.

$$SSI_k = \sum_{i=1}^N (|x_i^2|) \quad (8)$$

Características en el dominio de la frecuencia

- **Coefficientes autorregresivos** Describe cada muestra de la señal EMG como una combinación lineal de muestras anteriores más un término de error de ruido blanco. Los coeficientes AR se utilizan como características en el reconocimiento de patrones EMG.

$$x_k = - \sum_{i=1}^N a_i x_{k-i} + e_k \quad (9)$$

- **Mediana de frecuencia** Divide la densidad del espectro de potencia en dos partes iguales:

$$F_{MD} = \frac{1}{2} \sum_{i=1}^M PSD_i \quad (10)$$

6.1.4. Características en el dominio del tiempo-frecuencia

- **Transformada de Fourier de tiempo corto** Se amplía la aplicabilidad del método de la transformada de Fourier dividiendo la señal de entrada en segmentos; al hacer esto, se puede suponer que la señal en cada ventana es estacionaria.

$$STFT_x(t, w) = \int W^*(\tau - t)x(\tau)e^{-jw\tau} d\tau \quad (11)$$

- **Transformada Wavelet** Es una transformación en la que se integra una señal con una función de ondícula madre desplazada y escalada. La transformada de ondículas continua está representada por:

$$W_x(a, b) = \int x(t) \left(\frac{1}{\sqrt{a}} \right) \Psi^* \left(\frac{t - b}{a} \right) dt \quad (12)$$

6.2. Señales electroencefalográficas (EEG)

Una señal EEG es una medida de las corrientes que fluyen durante las excitaciones sinápticas de las dendritas de muchas neuronas en forma piramidal en la corteza cerebral, cuando se activan las células del cerebro (neuronas), las corrientes sinápticas se producen dentro de las dendritas. Esto provoca un campo magnético medible, siendo estas señales EMG, pero también producen un campo eléctrico secundario sobre el cuero cabelludo que es medible por sistemas EEG. [9].

6.2.1. Electroencefalografía

Es una técnica de exploración que tiene como base el registro de la actividad eléctrica cerebral. Las diferencias de potenciales son causados por la suma de potenciales posinápticos de las células piramidales que crean dipolos eléctricos entre el soma y dendritas apicales, que se ramifican a partir de neuronas. La corriente en el cerebro se genera principalmente bombeando los iones positivos de sodio, N_a+ , potasio $K+$, calcio Ca^{++} y el ion negativo de cloro Cl^- , a través de las membranas de las neuronas en la dirección que determine el potencial de la membrana [9].

La cabeza humana tiene varias capas en las que se puede encontrar el cuero cabelludo, el cráneo, el cerebro y dentro de estas hay varias capas intermedias. Solo el cráneo atenúa las señales cien veces más los tejidos blandos como la piel, esto quiere decir que solo las grandes poblaciones de neuronas activas para ser grabadas usando los electrodos del cuero cabelludo [10].

6.2.2. Ondas cerebrales

El sistema bioeléctrico de potenciales transmembranales producen en los grandes grupos celulares ondas electromagnéticas que se perciben en el córtex. Son voltajes eléctricos

oscilantes en el cerebro que miden apenas unas millonésimas de voltio, hay cinco ondas cerebrales ampliamente reconocidas como se puede observar en el Cuadro 1 [11].

Banda de frecuencia	Frecuencia	Estado del cerebro
Ondas Gamma γ	$> 35Hz$	Concentración
Ondas Beta β	12-35 Hz	Ansiedad dominante, Atención externo, relajado
Ondas alpha α	8-12 Hz	Atención relajada
Ondas Theta θ	4-8 Hz	Profundamente relajado
Ondas Delta δ	0.5-4 Hz	Durmiendo

Cuadro 1: Tabla de ondas cerebrales.

6.2.3. Sistema de posicionamiento 10-20

El sistema de posicionamiento 10-20 es un método reconocido internacionalmente por describir la localización de cada uno de los electrodos. Los números 10-20 se refieren al porcentaje (10 y 20 respectivamente) de distancia en la que deben de estar colocados los electrodos en la parte frontal, occipital del lado izquierdo y derecho del cráneo. Se tiene una letra para identificar los lóbulos y el número para ubicar el hemisferio.

Electrodo	Lóbulo
F	Frontal
T	Temporal
C	Central
P	Parietal
O	Occipital

Cuadro 2: Tabla de lóbulos.

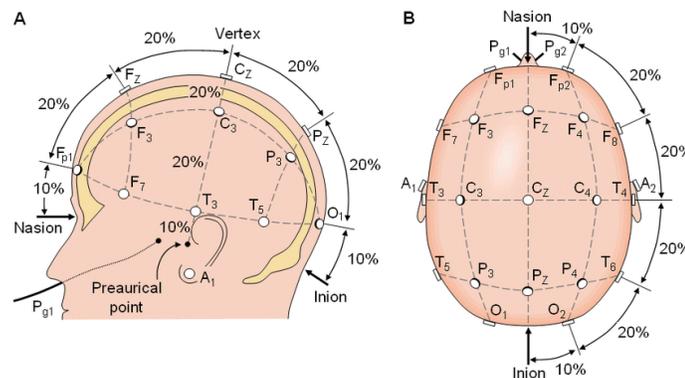


Figura 1: Sistema de posicionamiento 10-20 [12].

6.2.4. Características en el dominio del tiempo

- **Valor Medio** Dado que la suma del potencial EEG (positivo y negativo) suele ser del orden de unos pocos microvoltios cuando el tiempo de análisis no es demasiado

corto, la media es esencialmente una constante, aunque de un valor pequeño. Cualquier cambio en los valores de la media,

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (13)$$

- **Desviación estándar** Donde \bar{x} es el valor medio.

$$x_{std} = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}} \quad (14)$$

- **Oblicuidad** La oblicuidad mide el grado de desviación de la simetría de una distribución normal o gaussiana. Esta medida tiene el valor de cero cuando la distribución es completamente simétrica y asume algún valor distinto de cero cuando las formas de onda del EEG son asimétricas con respecto a la línea de base (como es el caso de algunos patrones de sueño característicos, ritmos mu, husos de morfina, etc.)

$$S_{kmc} = \frac{\sum_{i=1}^N \frac{(x_i - \bar{x})^3}{N}}{\left[\sum_{i=1}^N \frac{(x_i - \bar{x})^2}{N} \right]^{3/2}} \quad (15)$$

- **Curtois** Esta medida revela el pico o la planitud de una distribución. En electroencefalografía clínica, cuando se analiza EEG con poca frecuencia y modulación de amplitud, se observan valores negativos de curtois; por otro lado, los valores positivos altos de curtois están presentes cuando el EEG contiene picos transitorios,

$$K_{mc} = \frac{\sum_{i=1}^N \frac{(x_i - \bar{x})^4}{N}}{\sum_{i=1}^N \left[\frac{(x_i - \bar{x})^2}{N} \right]^2} - 3 \quad (16)$$

- **Correlación cruzada** Una secuencia de correlación cruzada entre dos señales de energía mide el grado de similitud entre estas dos señales. Si una señal se correlaciona consigo misma, la secuencia resultante se denomina secuencia de autocorrelación.

$$\hat{R}_{xy}(m) = \begin{cases} \sum_{n=0}^{N-m-1} x_{n+m} y_n & m \geq 0 \\ \hat{R}_{yx}(-m) & m < 0 \end{cases} \quad (17)$$

6.2.5. Características en el dominio de la frecuencia

- **Coefficientes auto-regresivos** Como se puede observar también puede ser usada en las señales EEG como en la EMG.

$$x_k = - \sum_{i=1}^N a_i x_{k-i} + e_k \quad (18)$$

- **Densidad del espectro de potencia** Muestra en qué punto las variaciones de frecuencia son fuertes y débiles. La densidad del espectro de potencia (P SD) se calcula mediante:

$$PSD = \left| \sum_{i=0}^{N-1} x_i e^{-j2\pi ki/N} \right|^2 \quad (19)$$

- **Poder de la banda** Las señales EEG contiene diferentes componentes de frecuencia específicos, que llevan la información discriminativa. Este tipo de característica refleja la energía en varias bandas ($\alpha, \beta, \delta, \gamma$ and θ). Una vez filtradas las bandas, se puede aplicar la densidad del espectro de potencia a cada una para obtener características importantes.
- **Relación de asimetría** Este valor se deriva de la densidad del espectro de potencia y se puede utilizar como una característica, también esta característica es útil para el análisis de EEG cuando las actividades mentales que se estudian exhiben diferencias inter-hemisféricas.

$$AS_{PSD} = \left[\frac{PSD_1 - PSD_2}{PSD_1 + PSD_2} \right] \quad (20)$$

6.3. Aprendizaje automático

El aprendizaje automático, *Machine Learning* (ML), se podría entender como automatizar las tareas manuales, pero este en vez de automatizar las tareas manuales, este realiza tareas frecuentes de volumen grande y que se computarizan de manera confiable y sin fatiga. El aprendizaje automático sigue dependiendo del humano para configurar el sistema y que este se haga las preguntas correctas. La inteligencia artificial (IA) se adapta a través de algoritmos de aprendizaje progresivo para permitir que los datos hagan la programación. La IA encuentra estructura y regularidades en los datos para que el algoritmo adquiera una habilidad entonces, vuelve a estos algoritmos clasificadores o predictores [13].

6.3.1. Redes neuronales

Las neuronas como las conocemos comúnmente son células del cerebro cuya función principal es la recolección, procesamiento y esparcimiento de señales eléctricas. Pero cuando se habla de una neurona artificial esta se refiere a una unidad de procesamiento de información fundamental para el funcionamiento de una red neuronal. Estas neuronas reciben entradas x y pesos w , dentro de estas neuronas se realiza una suma aritmética seguida de una función de activación. La suma aritmética es la siguiente:

$$net = \sum_{i=0}^n w_i x_i$$

Se asocia un *function fact* que en resultado da la salida de la neurona, la función en sí limita la salida de la neurona, ya que esta tiene dos condiciones, una es “activar” la neurona cuando las entradas a esta son las correctas y “desactivar” la neurona cuando las entradas son las incorrectas. Hay varias funciones de activación como la del umbral lineal, escalonada, escalonada arbitraria etc [14].

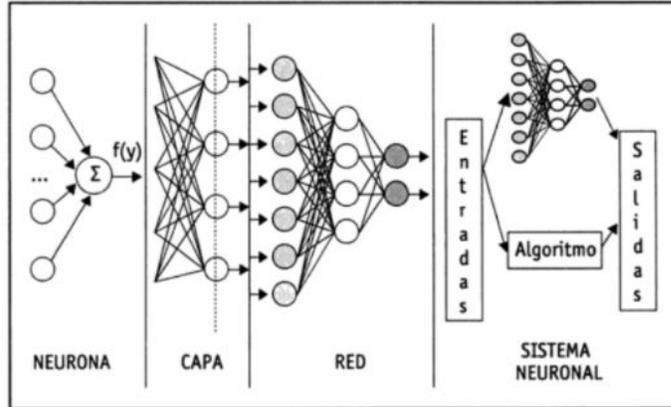


Figura 2: Sistema de posicionamiento 10-20 [15].

6.3.2. Máquina de soportes de vectores

Las máquinas de soporte de vectores (SVM) fueron desarrolladas en el marco de la Teoría de Aprendizaje Estadístico (SLT). Son entrenadas por algoritmos de optimización convexa (que existe una única solución) y construidas a partir de una estructura que depende de un subconjunto de Vectores Soporte (VS) estas ayudan a la interpretación del modelo [16].

Clasificación de patrones con SVM

Durante las tareas de clasificación existen dos fases, la fase del aprendizaje automático y la fase de reconocimiento, en la primera se seleccionan todas las características y atributos del espacio de entrada y así se logra entrenar el clasificador, el entrenamiento da como resultado un conjunto de parámetros llamados pesos w que definen al clasificador y la función discriminante que representa la frontera entre clases o regiones, en la segunda fase el modelo obtenido del clasificador ya entrenado se encarga de asignar los nuevos datos de entrada a una de las clases según la similitud de sus características [16].

Hiperplano de separación óptimo y margen máximo

Las SVM poseen la capacidad de aprender a partir de un conjunto de N muestras experimentales el cual se llama conjunto de entrenamiento. Donde cada muestra (\vec{x}_i, y_i) para $i = 1 \dots N$ está formada por un vector de n características con una etiqueta que indica la clase $\{\pm 1\}$ a la clase que pertenece cada muestra. El objetivo de esto es encontrar la regla de clasificación o función subyacente que clasifique correctamente las nuevas muestras pertenecientes al conjunto de prueba [16].

6.4. Interacción Hombre-Máquina (HMI)

Es un método mediante el cual un usuario ordena a un sistema mecánico-eléctrico un estado deseado. Esta tiene que tener la capacidad de comunicar los movimientos del usuario previstos al sistema robótico. Las *Human Machine Interface* pueden detectar los gestos naturales del usuario y los traduce en acciones de robot deseadas mientras este solo hace transiciones seguras, las HMI pueden ser invasivas y no-invasivas [17].

6.5. Interfaz Cerebro-Computadora (BCI)

La idea principal de las interfaces cerebro-computadora es la de capturar las señales eléctricas de la actividad cerebral de los deseos de comunicación del usuario y traducirlas en comandos que se interpretan y se ejecutan por una computadora. La interfaz examina las señales cerebrales utilizadas, tiene diferentes tipos de adquisición y de procesamiento para así poder traducir las señales del usuario [18]

6.6. Robot R17

El robot R17 es un brazo robótico de formato articulado verticalmente. Este es el formato que más se parece al brazo humano. Sus partes llevan el nombre del brazo humano, por esta misma razón. Es más versátil que otros formatos, pero tiene el inconveniente de que el contragolpe y el cumplimiento en cada articulación se suma articulación por articulación. R17 tiene un alcance nominal de 750 mm, lo que quiere decir 750 mm desde la articulación del hombro hasta la muñeca, más otros 100 mm más o menos para la mano, esto dependiendo del efector final que se le instale al R17 [19].



Figura 3: Robot R17 [19]

El robot es impulsado por motores paso a paso en micropasos, de 0,18 grados por paso, lo que da como resultado una resolución nominal de 0.1 mm o mejor en el efector final después del engranaje. El sistema del robot R17 consta de 3 unidades principales: el robot, el controlador y una computadora o terminal [19].

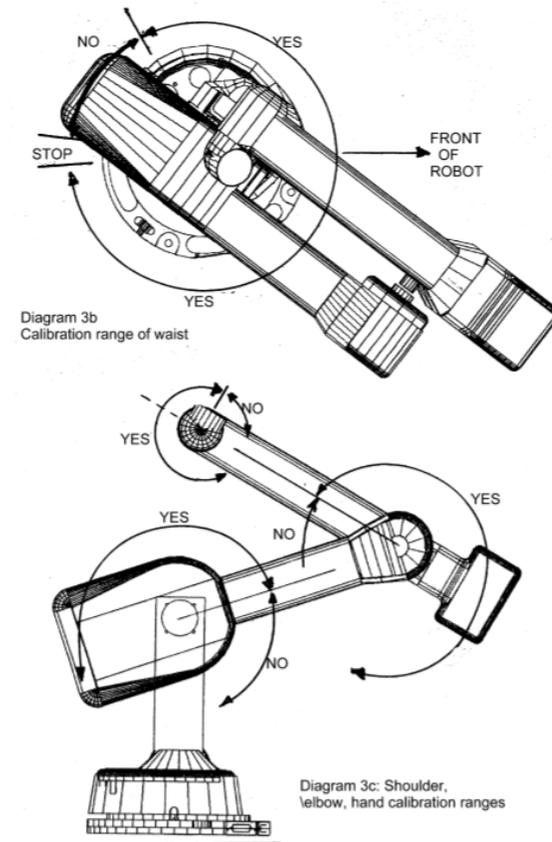


Figura 4: Robot R17 grados de libertad 19

Con fines de tener un acercamiento a experimentar con señales EMG y EEG se buscó una base de datos que cumpliera con los requisitos para poder cumplir los objetivos de este trabajo. La base de datos que se encontró, cuenta con señales EMG y EEG grabadas al mismo tiempo, pudiendo así poder tener un acercamiento a experimentar con combinación de señales para el control de sistemas robóticos. Al obtener los datos de esta base de datos, se procedió a realizar experimentos con distintos clasificadores y experimentando con diferentes tipos de características.

7.1. Base de datos pública de 11 tareas de movimiento intuitivas desde una sola extremidad superior

Al ser uno de los objetivos el poder combinar las señales EMG y EEG para poder utilizarlas de manera combinada, se procedió a buscar una base de datos que permitiera encontrar clasificadores que entregaran resultados congruentes y que lograran un *performance* aceptable. La base de datos tenía que contar con la característica que las señales, tanto las EMG's y las EEG's fueran capturadas al mismo tiempo y de la misma manera. Se encontró una base de datos que cumplía con todas las características necesarias para poder realizar los clasificadores. Esta es una base de datos encontrada en GigaDB se basa en el estudio de señales biomédicas y esta buscaba en efecto ampliar el estudio de las señales biomédicas combinadas, por eso el conjunto de datos incluye electroencefalografía de 60 canales (EEG), electromiografía de 7 canales (EMG) y electrooculografía de 4 canales (EOG) de 25 sujetos sanos recopilados durante sesiones de 3 días para un total de 82,500 ensayos en todos los sujetos [20]. En la Figura 5 se puede observar los sistemas de posicionamiento utilizados en el estudio de la base de datos.

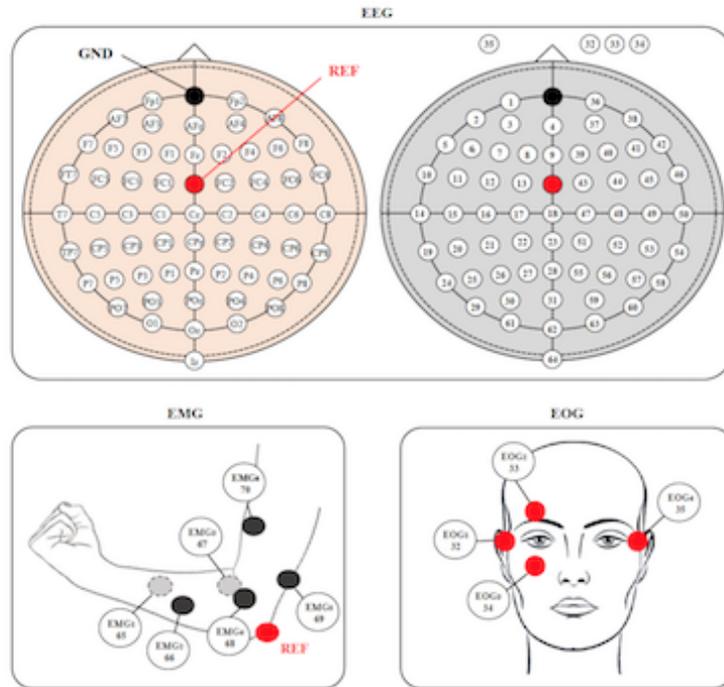


Figura 5: Posición de electrodos en el estudio [20].

7.1.1. Paradigma experimental

Como se mencionó antes la base de datos fue diseñada para adquirir datos cuantitativamente relacionados con los 11 movimientos diferentes de las extremidades superiores. Se pueden observar en la Figura 6 los diferentes movimientos que se realizaron por cada uno de los individuos. Cada muestra constaba de 4 segundos de descanso, 3 segundos de visualización sobre el movimiento a realizar y los últimos 4 segundos para realizar el movimiento en cuestión.

Extender el brazo en 6 direcciones

En este, se le pidió a los sujetos que realizaran tareas multidireccionales de estiramiento dirigidas desde el centro de sus cuerpos hacia afuera. Se llevaron a cabo estas tareas a lo largo de 6 direcciones diferentes en el plano 3D: adelante, atrás, izquierda, derecha, arriba y abajo.

Agarrar tres objetos con la mano

En este se pidió a los sujetos que tomaran tres objetos de uso diario mediante los correspondientes movimientos de agarre asignado, los movimientos fueron *card*, *ball*, and *cup*, corresponding to *lateral*, *spherical*, and *cylindrical grasp* que se pueden observar de mejor manera en la Figura 4.

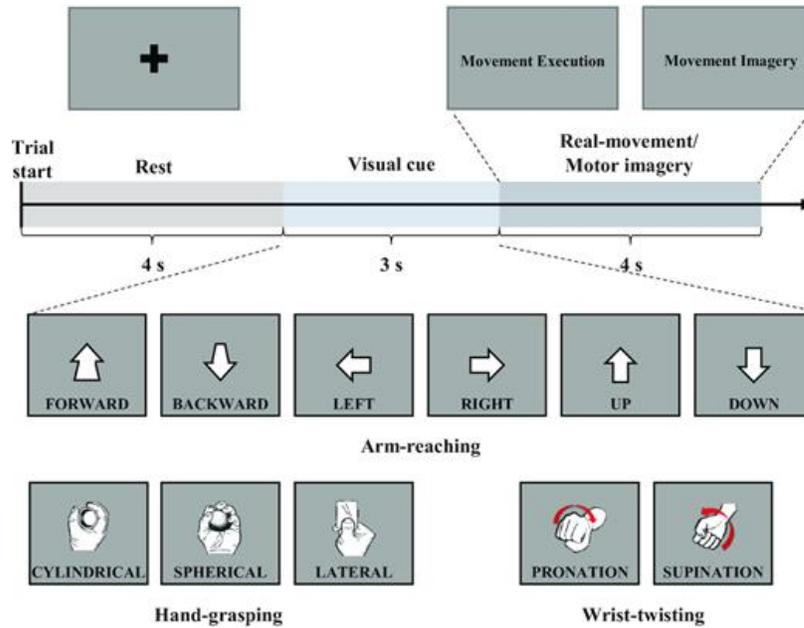


Figura 6: Movimientos realizados en la base de datos [20].

Giro de muñeca con dos movimientos diferentes

Para las tareas de giro con muñeca, los sujetos rotaron sus muñecas hacia la izquierda (pronación) y hacia la derecha (supinación). Estos dos tipos de movimiento son acciones complejas que más que todo se utilizan para decodificar las intenciones del usuario a partir de las señales cerebrales.

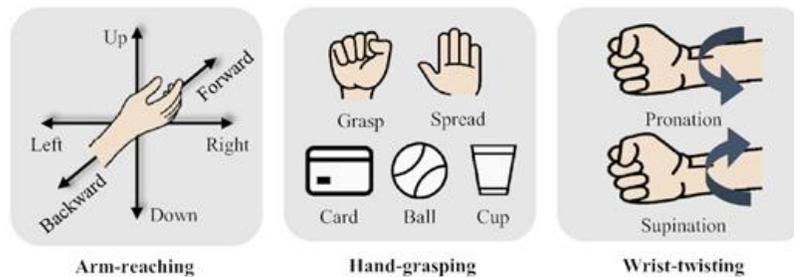


Figura 7: Movimientos realizados por los sujetos, detallados [20].

7.1.2. Procesamiento de señales

La base de datos que se encontró, ya contaba con *scripts* para el correcto uso de los datos. Los datos venían en forma de estructuras, por lo que el procesamiento para obtener los datos ordenados por clases y filtrados correctamente se hicieron con estos scripts. Los datos obtuvieron se obtuvieron de un repositorio en línea de la base de datos por medio del software “FileZilla”, esto porque la base de datos al ser tan grande está guardada en un servidor. Estos procesos se hicieron de manera independiente para cada tipo de señal.

7.1.3. Extracción de características

Al tener los datos ordenados por clases, se procedió a realizar un algoritmo que permitiera obtener características de las señales ya filtradas y ordenadas. Este algoritmo es diferente tanto para señales EMG y señales EEG, ya que no se utilizaron las mismas características para ambas señales. El algoritmo permite obtener un vector de características de 2 clases y 10 características. El algoritmo entrega este vector en forma de celda. Para poder utilizar los datos en los clasificadores, se llegó a la conclusión que el número de canales utilizados para obtener los datos tiene que ser igual para ambas señales. Esto permite que el número de muestras sea el mismo sin importar el número de características utilizado para cada tipo de señal. En el caso de las señales EEG, se utilizaron los canales F3, AF3, F4, AF4 esto porque según [21] son los canales con más correlación a la corteza motora, ya que ofrecen un mayor potencial. En [21] se determina que por motivo de reducir la cantidad de datos y que él maneja de estos sean más prácticos los canales más significativos para realizar la clasificación son AF3 y AF4. Para las señales EMG, se determinó que cualquiera de los 7 canales utilizados brindan un buen rendimiento al momento de clasificar, por lo que estos solo tienen que cumplir con ser la misma cantidad que los canales de las señales EEG.

Clasificadores para colección de señales

Los resultados de los clasificadores se representarán en forma de matrices de confusión. Las matrices de confusión son una medida muy popular utilizada para resolver problemas de clasificación. Se puede aplicar tanto para la clasificación binaria como problemas de clasificación multiclase como se menciona en [22].

Es importante mencionar que las matrices de confusión en el caso de las redes neuronales, muestra 4 matrices y no solo una como en el caso de las máquinas de soporte de vectores. Esto se debe a que el paquete utilizado en MATLAB para redes neuronales entrega las matrices de confusión de los datos de entrenamiento, los datos de validación y los de prueba. La matriz de la esquina inferior derecha es la que se compara con las matrices de confusión de SVM.

8.1. Clasificadores para señales EMG

Para las pruebas con dos clases forma en la que se obtenía el vector de características para cada prueba que se hizo es la misma, lo único que variaba en el algoritmo al momento de obtenerlo es la cantidad de características obtenidas, ya que si agregábamos más canales, se agregaban más características. El vector para señales EMG cuenta con las siguientes características: Varianza (VAR), Cruces por Zero (ZC), Valor Absoluto promedio (MAV), Media cuadrática (RMS), Longitud de forma de onda (WF), Amplitud de Wilson (WA), Entropía de Shannon, Oblicuidad, Energía y Entropía logarítmica. Con este vector de características es fácil realizar todo tipo de pruebas, clasificando de distintas maneras para poder obtener el mejor rendimiento.

8.1.1. Redes neuronales señales EMG (2 clases y 1 Sujeto)

Se hicieron diversas pruebas de las señales EMG, tanto en redes neuronales como en máquinas de soporte de vectores. Desde utilizar un solo canal, una característica y dos clases, hasta utilizar 7 canales, 10 características y 2 clases. Los mejores resultados en el caso de las señales EMG se dieron cuando se utilizaron 5 características y 4 canales. Para el entrenamiento de las redes neuronales se utilizó un tamaño de capa oculta de 10, la función de entrenamiento SCG debido al balance de consumo de memoria y calidad de resultados

que presenta. Se utilizó un 80 % de las muestras para entrenar, 10 % para validar y 10 % para probar. Las dos clases utilizadas son los dos tipos de *wrist-twisting*.

En la Figura 8 se puede observar un entrenamiento utilizando 9 características y solamente un canal, se obtuvo un 89 % de rendimiento. En la Figura 9 se observa la matriz de confusión para 4 características y 3 canales, así mismo el número de muestras subió, se obtuvo un rendimiento de 85 % de rendimiento, esto no es favorable, ya que en teoría el agregar más datos y más canales al entrenamiento, debería de ayudar a clasificar mejor las muestras. Por último en la Figura 10 se muestra una matriz de confusión de un entrenamiento con 5 características y 4 canales, con un rendimiento del 92 %. Posiblemente el aumentar los canales mejore el entrenamiento.



Figura 8: Matriz de confusión para RNA con 9 características y 1 canal



Figura 9: Matriz de confusión para RNA con 4 características y 3 canales

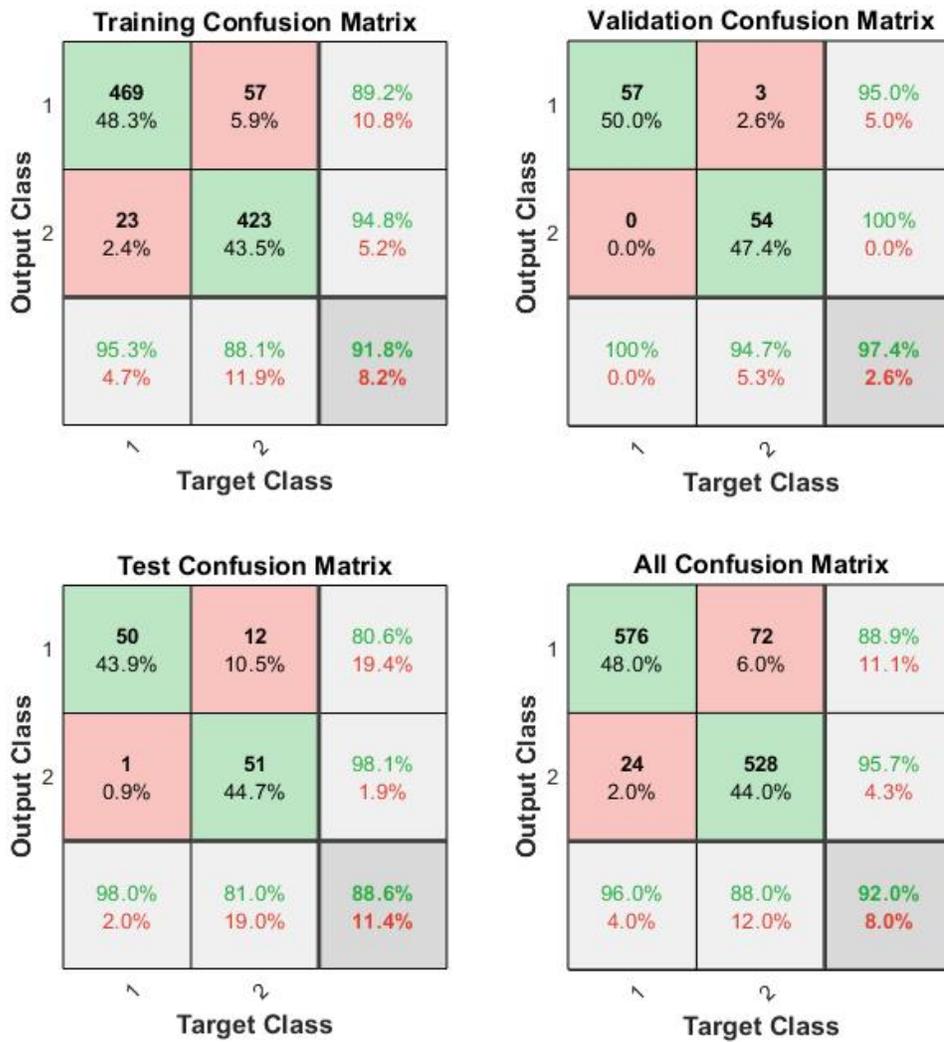


Figura 10: Matriz de confusión para RNA con 5 características y 4 canales

8.1.2. Máquina de soporte de vectores señales EMG (2 clases y 1 sujeto)

Para el caso de la máquina de soporte de vectores, se implementaron 4 tipos de entrenamientos. En el primer modelo se utilizó un Kernel con la función lineal, con una escala de Kernel automático. El segundo modelo fue un Kernel con una función polinomial, con una escala de kernel polinomial grado 2. En el tercer modelo se utilizó un kernel con la función de base radial, con una escala automática. Por último se utilizó un modelo también con una función de kernel de base radial, pero con un tamaño de kernel estandarizado. Para cada corrida se utilizó un 70 % para realizar el entrenamiento y el 30 % para validar. En cada prueba se tomó el mejor de los 4 modelos.

Se puede observar en la Figura 11 la matriz de confusión de de un modelo que utiliza 8 características de un solo canal, se obtiene un 92.2% de rendimiento. En la Figura 12 se muestra un modelo con 8 características y 4 canales, se obtiene un mejor resultado con un 98.9% de rendimiento. Mostrando en este caso que seleccionando 4 canales y mejores características por canal se logra discriminar mejor las clases.

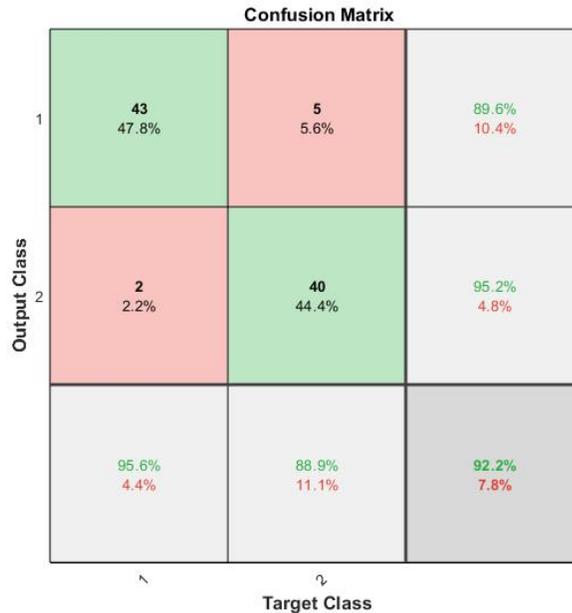


Figura 11: Matriz de confusión para SVM con 8 características y 1 canal

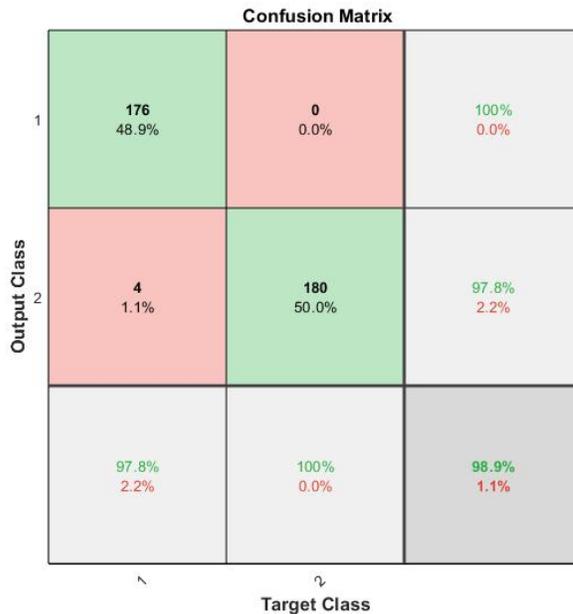


Figura 12: Matriz de confusión para SVM con 8 características y 4 canales

Es bastante aceptable que el mejor resultado se haya obtenido mediante máquina de soporte de vectores. Esto es porque para redes neuronales, se necesita una mayor cantidad de datos. SVM funciona mejor con pocas muestras y funciona muy bien como clasificador binario. Es importante enfatizar en que estos entrenamientos realizados cumplen con el objetivo de obtener una idea general para los entrenamientos que se realizarán en los siguientes capítulos, no es una comparación directa entre los tipos de modelos.

8.2. Clasificadores señales EEG

Al igual que para las señales EMG en las señales EEG para las pruebas con dos clases la forma en la que se obtenía el vector de características para cada prueba que se hizo es la misma, lo único que variaba en el algoritmo al momento de obtenerlo es la cantidad de características obtenidas, ya que si agregábamos más canales, se agregaban más características. El vector para señales EEG cuenta con las siguientes características: Varianza (VAR), Cruces por Zero (ZC), Valor Absoluto promedio (MAV), Media cuadrática (RMS), Longitud de forma de onda (WF), Amplitud de Wilson (WA), Entropía de Shannon, Oblicuidad, Energía y Entropía logarítmica. Con este vector de características es fácil realizar todo tipo de pruebas, clasificando de distintas maneras para poder obtener el mejor rendimiento. De igual forma la dos clases utilizadas son las dos formas de *wrist-twisting* al igual que para las señales EMG.

8.2.1. Redes neuronales señales EEG (2 clases y 1 sujeto)

En el caso de las señales EEG, se variaron un poco los parámetros en el entrenamiento de las redes neuronales ya se utilizó un tamaño de capa oculta de 20, se siguió utilizando la función de entrenamiento SCG debido al balance de consumo de memoria y calidad de

resultados que presenta. Se utilizó un 70 % de las muestras para entrenar, 15 % para validar y 15 % para probar.

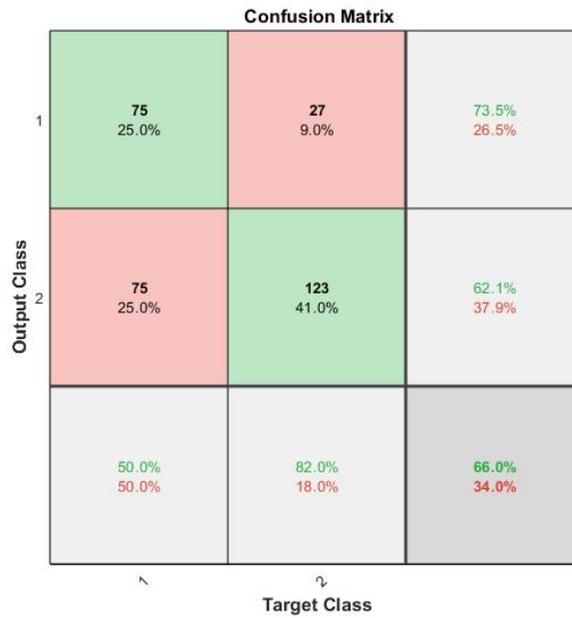


Figura 13: Matriz de confusión para redes neuronales con 2 características y un canal.



Figura 14: Matriz de confusión para redes neuronales con 5 características y un canal.

Se puede observar en las figuras 13 y 14 estos dos modelos mostraron los mejores resul-

tados obtenidos con redes neuronales, se utilizó el canal AF3 para realizar estas pruebas. El hecho de obtener un rendimiento menos óptimo con redes neuronales en señales EEG puede ser por las características obtenidas de las señales EEG, normalmente en una red neuronal, se tienen que normalizar las características y la selección de las características en esta parte de los entrenamientos no fue un factor importante, solo se buscaba obtener un buen resultado a partir de tener características iguales para ambas señales por separado. Tampoco se tomaron en cuenta los canales a utilizar y se seleccionó el AF3 porque era el más cercano a la corteza motora.

8.2.2. Máquina de soporte de vectores señales EEG

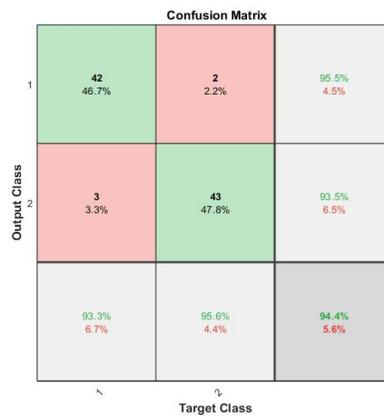


Figura 15: Matriz de confusión para redes neuronales con 4 características y un canal.

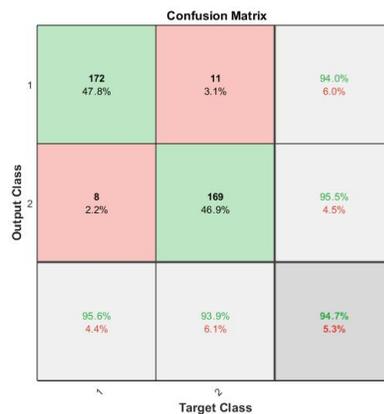


Figura 16: Matriz de confusión para redes neuronales con 8 características y 4 canales.

Se puede observar en las matrices de confusión de las figuras [15](#) y [16](#) los resultados obtenidos en SVM son mucho mejores que los entrenamientos con RN.

8.3. Clasificadores señales EEG-EMG

Uno de los paradigmas experimentales que se desarrollan en los siguientes capítulos, es el de la combinación de señales para el desarrollo de modelos de AI. Para las primeras pruebas se utilizaron 8 características, 4 por cada canal y dos clases. También se hicieron pruebas con las mismas características, pero utilizando 4 clases. La base de datos asegura una sincronía al momento de tomar los datos, tanto de las señales EEG como de las señales EMG. Para seleccionar los canales a utilizar, se tomó como base, las secciones anteriores. Esto ayudó a determinar que los canales con más peso al momento de clasificar, eran el C3 y el FC5, en las clases: *forward*, *left*, *right* y *backward*. También se determinaron las 4 características con más pesos en los entrenamientos anteriores y se utilizaron en estos entrenamientos. En este caso se decidió mostrar el rendimiento de cada una de las pruebas en todos los modelos. Los datos utilizados corresponde a un set de 25 individuos, con 50 muestras por cada clase de parte de cada uno de ellos. El objetivo de utilizar 25 sujetos, es buscar la generalización del modelo y así poder observar cómo se comportan los modelos cuando se utilizan sujetos variados y en diferentes condiciones.

Para la matriz de confusión en el modelo entrenado en la sección 8.3.4 la matriz de confusión es diferente ya que se utilizó la aplicación *classification learner* del software MATLAB ya que los códigos utilizados en las secciones anteriores para los modelos SVM solo permiten dos clases.

8.3.1. Redes neuronales combinación de señales EMG-EEG (2 clases)

En este caso, para las redes neuronales, se utilizó un tamaño de capa oculta de 15, la función de entrenamiento SCG, se utilizó un 80 % de las muestras para entrenar, un 10 % para validar y 10 % para probar.

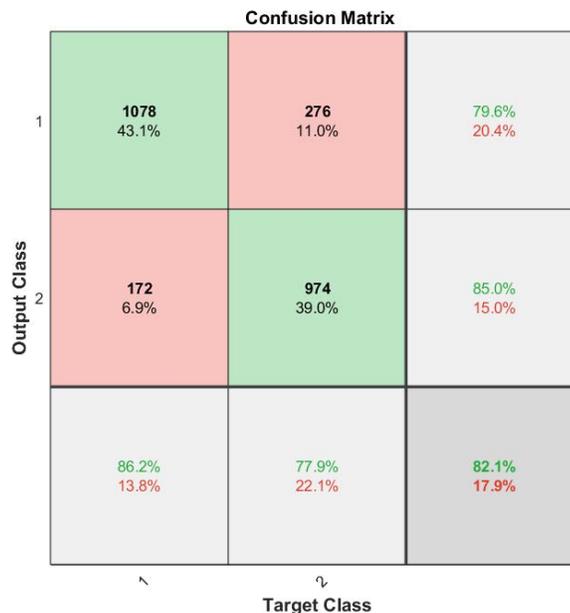


Figura 17: Matriz de confusión para redes neuronales con 8 características, 2 canales y 2 clases

8.3.2. Máquina de soporte de vectores señales combinación de señales EMG-EEG (2 clases)

Para este caso, en máquina de de soporte de vectores, se decidió utilizar la aplicación *Classification learner* del software MATLAB. Esta aplicación permite entrenar diferentes variantes de SVM. Se obtuvo un 65% de rendimiento utilizando SVM utilizando kernel cuadrático.

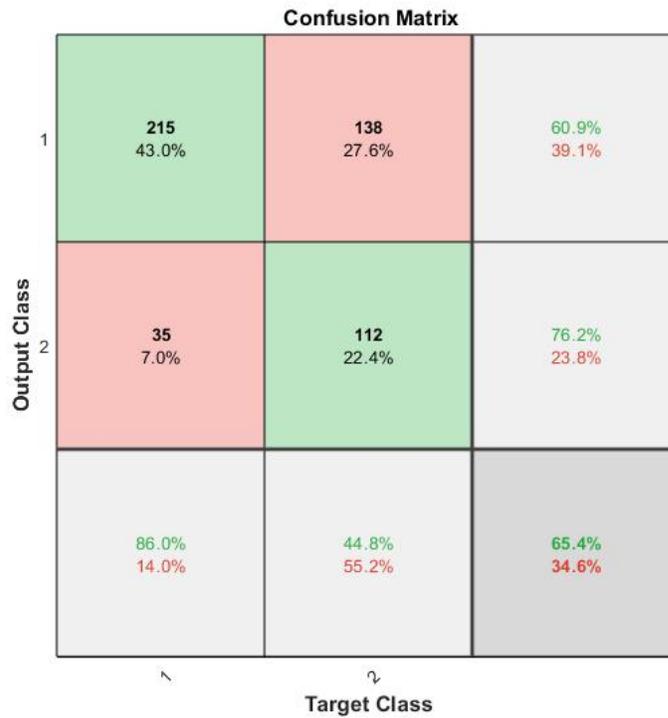


Figura 18: Matriz de confusión para SVM con 8 características, 2 canales y 2 clases

8.3.3. Redes neuronales combinación de señales EMG-EEG (4 clases)

En este caso, para las redes neuronales, se utilizó un tamaño de capa oculta de 15, la función de entrenamiento SCG, se utilizó un 80 % de las muestras para entrenar, un 10 % para validar y 10 % para probar. Los resultados fueron de un 40.1 % de rendimiento para 4 clases.

Confusion Matrix

Output Class	1	273 5.5%	135 2.7%	226 4.5%	221 4.4%	31.9% 68.1%
	2	556 11.1%	886 17.7%	192 3.8%	693 13.9%	38.1% 61.9%
	3	292 5.8%	100 2.0%	708 14.2%	197 3.9%	54.6% 45.4%
	4	129 2.6%	129 2.6%	124 2.5%	139 2.8%	26.7% 73.3%
		21.8% 78.2%	70.9% 29.1%	56.6% 43.4%	11.1% 88.9%	40.1% 59.9%
	Target Class					

Figura 19: Matriz de confusión para redes neuronales con 8 características, 2 canales y 4 clases

8.3.4. Máquina de soporte de vectores señales combinación de señales EMG-EEG (4 clases)

Para este caso, en máquina de de soporte de vectores, se obtuvo un 59.6 % de rendimiento utilizando SVM utilizando kernel cuadrático.

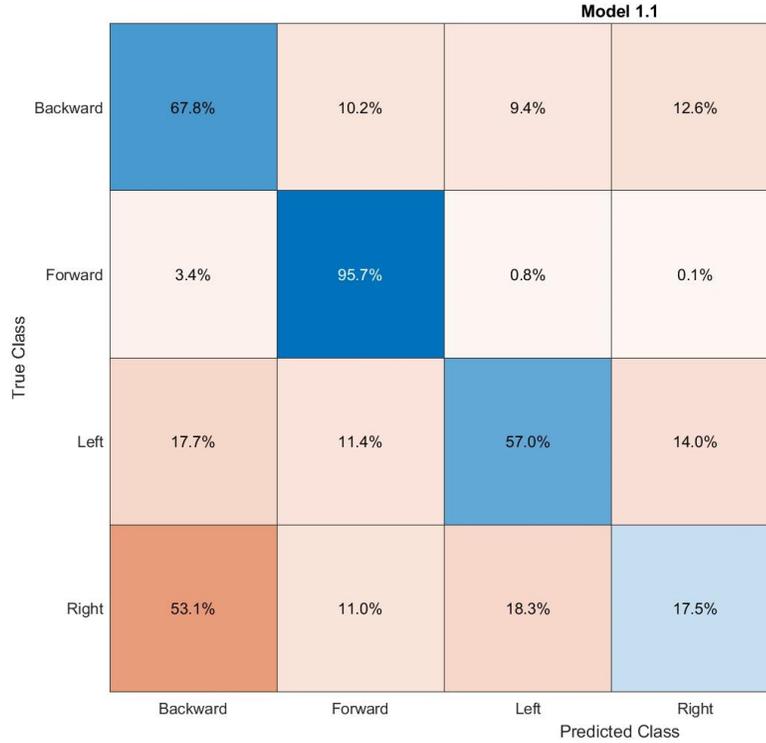


Figura 20: Matriz de confusión para SVM con 8 características, 2 canales y 4 clases

Se puede observar que para cada modelo, al agregar dos clases más, los rendimientos bajan. Esto tiene sentido cuando se sabe que los datos utilizados es de 25 individuos diferentes con solo 50 muestras por cada una de las clases. Al utilizar dos clases muy distintivas entre si como *forward* y *left*, no es necesaria tanta información en los vectores de características y los resultados son buenos. Al momento de utilizar 4 clases, se necesitan muchas más muestras por cada una y así poder tener un modelo más general. Estos resultados ayudarán a determinar los paradigmas experimentales que se utilizarán en los siguientes capítulos.

Adquisición y recopilación de señales EMG

En este capítulo se tratarán todos los temas que tienen que ver con la adquisición de las señales EMG, así como los paradigmas experimentales con los que se recopilaban las señales. También se introduce a la primera interfaz utilizada para la adquisición de datos.

9.1. Dispositivo de adquisición de señales EMG

La interfaz de fases anteriores para la adquisición de señales, utilizaba el dispositivo bitalino para extraer las señales y un arduino uno para adquirir esas señales y poder procesarlas en la computadora. El dispositivo con el que cuenta la Universidad del Valle de Guatemala es específicamente el *BITalino (r)evolution Board* que tiene un diseño de hardware todo en uno, todos los bloques están preconectados y los datos se adquieren de forma bruta y esto permite que se pueda utilizar en aplicaciones para *Human-Computer Interaction* [23].

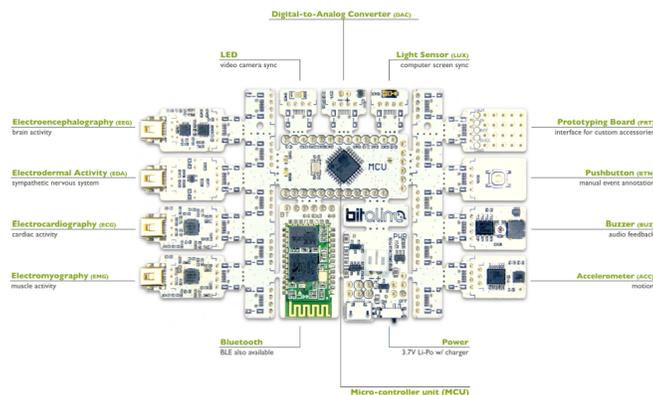


Figura 21: Diagrama de bloques interconectados dispositivo bitalino [23]

El principal reto con el dispositivo bitalino, fue encontrar una mejor manera de comunicación entre el y la interfaz o computadora. La mejor forma para utilizar el dispositivo fue por medio de comunicación bluetooth utilizando *BITalino Toolbox* en el software MATLAB.

Este *toolbox* se comunica por medio de comandos con el dispositivo bitalino, de esta forma desarrolla diferentes formas de obtener los datos en bruto [24].

9.1.1. Primeros acercamientos a dispositivo bitalino

Se desarrollaron varios scripts en MATLAB, para poder familiarizarse con las funciones del *toolbox* y poder encontrar la mejor manera de obtener las señales. El *toolbox* permite obtener las señales de dos formas, recolectando los datos actuales del *buffer* (*ReadCurrentValue()*) del bitalino y recolectando los últimos n valores del *buffer* (*Read()*) que nosotros le asignemos. Luego de hacer varias pruebas, se determinó que la mejor manera de obtener los datos es esperando que el *buffer* se llene con el tiempo que le asignemos. Esto porque al tratar de usar la función *ReadCurrentValue()* existía un desfase de 3 segundos aproximadamente, esto seguro se debe a configuraciones internas del dispositivo bitalino.

Se determinó que el dispositivo bitalino funciona con frecuencias de muestreo de; 1, 10, 100, 1000 Hz. La batería del bitalino es de 500 mAh, unas 10 horas de uso.

9.2. Adquisición y procesamiento de señales EMG

9.2.1. Algoritmo de adquisición

Como se mencionó en la sección anterior, la mejor forma de obtener la señal mediante el dispositivo bitalino fue por medio de la función *read()*. Esta lee los datos del búfer interno del bitalino, luego que se llenó con el tiempo asignado, toma los datos y limpia el búfer. El algoritmo empieza estableciendo la conexión entre MATLAB y el dispositivo bitalino. La variable donde se guardó la conexión como objeto. Este puede utilizarse con las funciones del *toolbox*. Al establecer la conexión con el dispositivo bitalino, se le ordena al bitalino que empiece la adquisición de datos en segundo plano con la propiedad *startBackground(obj)* que permite la adquisición de data sin bloquear la ejecución de los comandos de MATLAB [24]. Luego de estar recibiendo datos en segundo plano, se realiza el procedimiento para guardar los datos. Se tiene que entrar a un ciclo continuo en el que cada n segundos (depende de la ventana de tiempo que quiera el usuario) se tomen datos, en ese ciclo de espera, se tuvo que realizar el movimiento o movimientos requeridos, luego se reinicia el búfer y empieza a llenarse en n segundos. Cuando el búfer está lleno, se utiliza la función *read(obj)* y guarda los últimos n datos.

Una consideración importante es que la forma en la que el *toolbox* recibe los datos en todas sus funciones, es en una matriz de $n \times 11$ donde las 11 columnas representan a los 4 módulos de sensores con los que cuenta el dispositivo bitalino: EMG; ECG; EDA; EEG; ACC [23]. En este caso en particular y como se determinó en las pruebas del capítulo anterior, un canal es muy efectivo para señales EMG. Solo se toma la columna 6 de la matriz que representa la señal EMG.

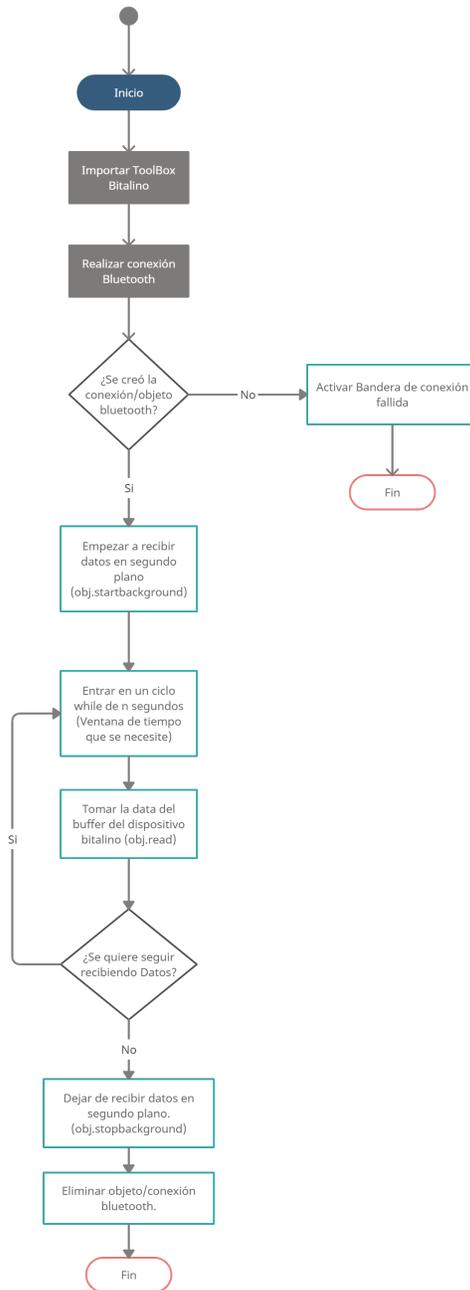


Figura 22: Algoritmo de adquisición de datos bitalino

9.2.2. Procesamiento de señal

Al contar con la señal directamente se procede a amplificar y convertir la señal, como se menciona en [23] el voltaje al que trabaja el dispositivo bitalino es de 3.3 V, por lo que se hizo la conversión. Luego de esta etapa, se tiene que proceder a realizar un filtrado de paso de banda primario como se menciona en [25]. En las investigaciones previas se determinó que las grabaciones de señales EMG suelen estar entre el rango de banda de 10Hz a 350Hz. Se procedió a utilizar unos filtros como los utilizados en [5]. Unos filtros IIR de tipo *Butterwoth* pasa banda, con frecuencia de corte entre 20 Hz y 450 Hz, esto porque la grabación intramuscular debe formarse con el aumento adecuado del corte de alta frecuencia a un punto bajo de 450 Hz según [25]. Luego se utilizó un filtro quita banda entre 59 Hz y 61 Hz para eliminar el ruido de la red eléctrica. No se utilizó un filtro *notch* en este caso, para asegurarse contener un buen rango de la frecuencia que se necesitaba eliminar.

9.2.3. Algoritmo de identificación de actividad

Por la forma del algoritmo diseñado y por el objetivo principal en el que está enfocado, es necesario implementar un algoritmo para la identificación de actividad muscular. Esto nos permite tener una lectura continua de datos sin la necesidad de estar pasando la señal por el modelo de *machine learning* construido y estar dando una clase constantemente pudiendo así dañar el sistema robótico al que se le están enviando comandos.

Para implementar este algoritmo se tiene que determinar un umbral de actividad. En [5] se utiliza un umbral con el voltaje máximo de una primera muestra. En este caso, se decidió utilizar otro método para la identificación de actividad. Se utilizó el método de *Matched Continuous Wavelet Transform*(CWT), esto porque al usar el método con un umbral de voltaje, no se consideran cosas como la distancia de los electrodos cuando se generan los MUAP's en diferentes sesiones [26]. Como se muestra en [26] lo que hace la CWT es realizar un análisis sub-banda de la señal original. Para determinar el umbral, se realizó una sesión de pruebas de actividad. Se determinó que los electrodos importan mucho al momento de determinar este umbral, ya que el ruido provocado por la mala posición de la referencia, la distancia entre los dos electrodos y la conducción de estos puede afectar.

Al obtener la señal filtrada, se aplica la función $CWT(X)$ que entrega la CWT, esto es una matriz bidimensional donde cada fila corresponda a una escala y las columnas corresponden a la magnitud de X . Luego de esto se tuvo que obtener el valor absoluto de esta matriz y aplicar dos veces la función $max(CWT)$ para obtener el máximo de la matriz bidimensional. El resultado de estas operaciones será el umbral para detectar actividad. Luego se planteó un algoritmo revisa si se supera este umbral, si una bandera de activación está apagada y si este no sobre pasa el voltaje nominal que es de 3.3 V.

9.3. Interfaz para la recolección de datos

Antes de empezar a entrenar y crear una interfaz para la clasificación de señales, se procedió a realizar una interfaz para poder recolectar datos de una forma más intuitiva y fácil. La interfaz fue creada con la aplicación integrada en el software MATLAB, llamada *App Designer*. Se utilizaron los algoritmos mencionados con anticipación en este capítulo para poder recolectar los datos necesarios para poder crear una base de datos propia.

La interfaz que se muestra en la Figura [23] busca ser muy intuitiva para el usuario y busca ser una herramienta útil al momento de querer recopilar datos. Esta se conecta con el

Recolección de Datos EMG - UVG

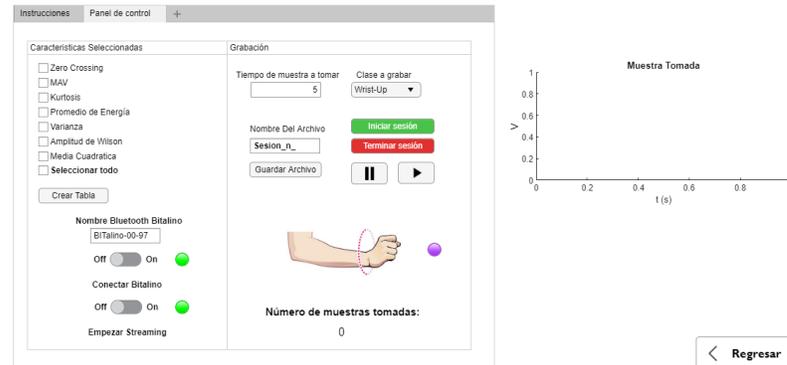


Figura 23: Interfaz recolectora de datos

dispositivo bitalino y permite capturar las señales brindadas por el canal EMG del mismo. Si se desea cambiar el canal a recibir, se debe de cambiar el código interno. Cuando se guarda la recopilación de datos al terminar la sesión, se guarda de dos formas, en forma de los datos con los vectores de características seleccionados para cada prueba realizada, así como los datos en bruto de estas. El uso detallado e instrucciones de esta interfaz se puede encontrar en el manual de usuario del Anexo [52](#).

En el Anexo [45](#) se puede observar el funcionamiento de esta interfaz, esta fue probada por alumnos del curso de fisiología cuantitativa 2 de la Universidad del Valle de Guatemala, como parte de su proyecto final.

9.4. Paradigma experimental para la recolección de datos

Para la recolección de datos se estableció el paradigma experimental con el cual se tomaron las muestras de las clases con las que se entrenarían los modelos para la interfaz clasificadora. El experimento se diseñó para adquirir los datos cualitativamente relacionados con los dos movimientos diferentes del brazo izquierdo. Los experimentos se realizaron con un solo sujeto y con la misma extremidad. Se hicieron 6 sesiones distintas, donde cada sesión se recopilaban 120 muestras. Al principio de cada prueba el sujeto descansaba 2 segundos y se proyectaba una ayuda visual del movimiento a grabar. Después de estos 2 segundos se activó una luz en la interfaz por 1 segundo, que indicaba al usuario que tenía que realizar el movimiento. Al terminar de realizar el movimiento, el sujeto descansaba 2 segundos antes de empezar la siguiente prueba.

- **Palma arriba** - Este movimiento constaba de mover la palma de la mano izquierda hacia arriba por 1 segundo. Este tenía que ser un movimiento bastante rápido y suave.
- **Puño arriba**-Este movimiento constaba de mover el puño de la mano izquierda hacia arriba por 1 segundo. Este tenía que ser un movimiento bastante lento y fuerte.

Se recopilaban alrededor de 920 muestras por cada uno de los movimientos o clases propuestas en el paradigma experimental. Estas muestras se guardaron tanto en vectores

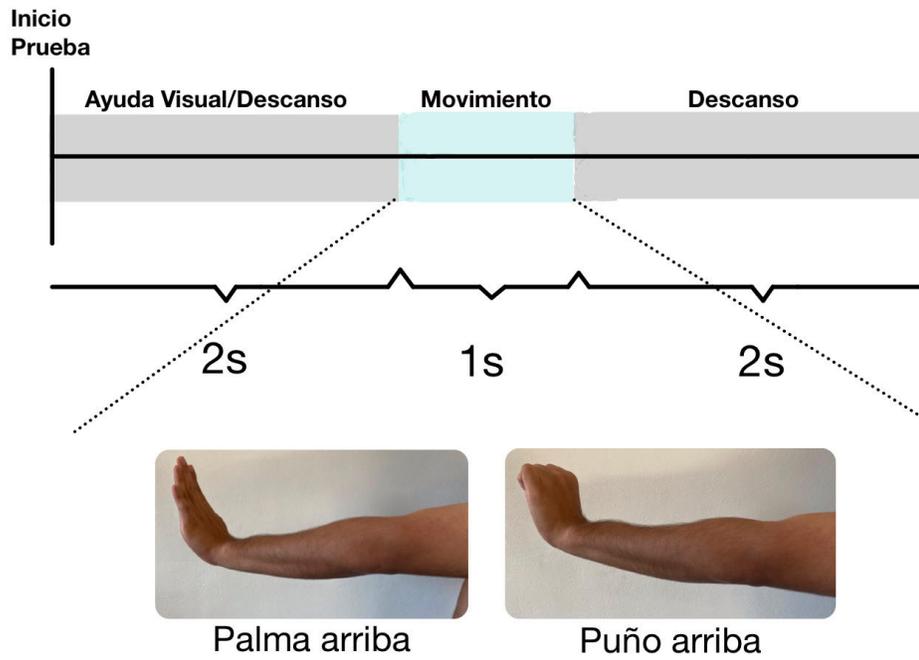


Figura 24: Paradigma experimental

de características, como las señales en bruto sin procesar de cada una de las pruebas. Los resultados y clasificación de resultados se verán en el siguiente capítulo.

Interfaz clasificadora de señales EMG

Al existir fases anteriores que ya habían hecho tangible el hecho de tener una interfaz clasificadora pero solo para simulaciones de sistemas robóticos. Se tenían que plantear criterios y métricas de evaluación para estas fases previas. Esto con el fin de obtener mejoras en las próximas interfaces. En este capítulo también se muestran los resultados de los entrenamientos para los datos recopilados que se mencionó en capítulos anteriores. También se muestra el funcionamiento de la interfaz con entrenamientos propios utilizando el robot R17 de la Universidad del Valle de Guatemala. El uso detallado e instrucciones de esta interfaz se puede encontrar en el manual de usuario del Anexo [52](#).

10.1. Criterios y métricas de evaluación para la interfaz EMG de fases previas

Para poder obtener un área de mejora en la interfaz para la adquisición de señales EMG, se hicieron varias pruebas con la interfaz de fases previas y se determinaron las áreas de mejora en las que se tenía que enfocar está parte del desarrollo de la interfaz.

- **Conectividad:** Para poder utilizar el dispositivo bitalino como instrumento de adquisición de señales, se tenía que utilizar un Arduino para poder extraer por medio de canales internos de la placa las señales deseadas.
- **Adquisición de señales:** La señal obtenida al utilizar un cable directo de la placa bitalino a una de las entradas analógicas del Arduino, no termina de venir directamente del completo procesamiento que hace la placa bitalino con los filtros internos para cada señal.
- **Manual de usuario:** La interfaz no cuenta con un manual de usuario completo, para poder determinar los fallos que pueda tener y como se solucionan. Este manual de usuario tendría que tener en cuenta consideraciones como que los umbrales para la detección de actividades, cambian para los diferentes usuarios de la interfaz.

10.2. Clasificación de señales EMG

Como se mencionó en el capítulo anterior, se recolectaron alrededor de 480 pruebas por cada clase de movimiento que se planteó en el paradigma experimental. Se utilizaron los conocimientos adquiridos en el Capítulo 8 para poder hacer varias pruebas con los datos recolectados y poder determinar el mejor modelo y entrenamiento para la interfaz clasificadora.

10.2.1. Classification Learner MATLAB

Para entrenar los modelos se decidió utilizar la aplicación interna con la que cuenta el software MATLAB, esta permite poder clasificar en paralelo 6 algoritmos de clasificación diferentes. Por ejemplo variantes de SVM, algoritmos de árboles de decisión, Análisis de discriminante, clasificadores de regresión logística y clasificadores de vecino más cercano. La aplicación solo necesita los datos ordenados en vectores de característica y estar etiquetada. Con esto la aplicación permite empezar a entrenar, entregando una validación cruzada de las características seleccionadas y entrega las matrices de confusión de cada modelo entrenado.

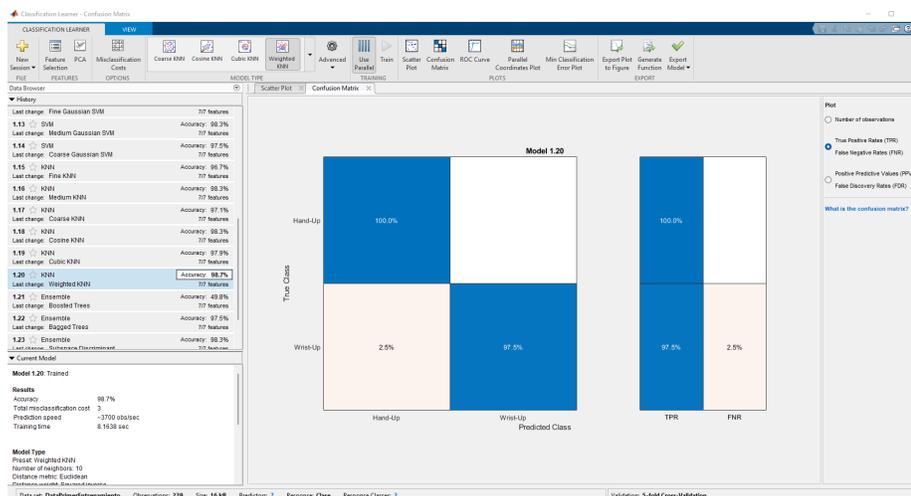


Figura 25: Learning App

La clasificación se dividió en cuatro conjuntos de datos a clasificar, el primer conjunto de datos fueron las primeras 120 muestras por clase de cada sesión, en el siguiente set se agregaban los nuevos 120 datos de la siguiente sesión, y así sucesivamente.

10.2.2. Neuronal net Fitting MATLAB

Ya que el *classification learner* no cuenta con el algoritmo de clasificación de redes neuronales, se procedió a utilizar otra aplicación con la que cuenta el software MATLAB para clasificar, pero esta vez solo serviría para redes neuronales. El app entrega el algoritmo necesario para poder cambiar parámetros sencillos de redes neuronales, como la función de entrenamiento, el número de neuronas internas y los porcentajes de data de entrenamiento y data para validación.

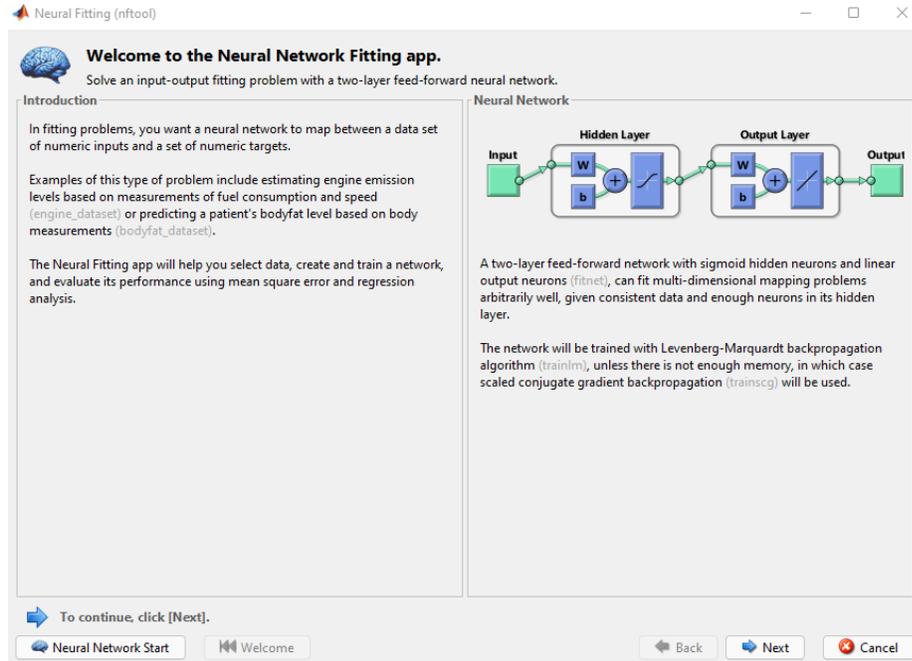


Figura 26: Neuronal net fitting

10.2.3. Redes neuronales en Python

Con objeto de experimentación se decidió utilizar otro lenguaje de programación para poder comparar, no solamente los modelos, sino también los lenguajes. Esto recordando que las redes neuronales no solo son modelos complejos, sino que tienen un exceso de parámetros. Para poder hacer más tangible los entrenamientos con redes neuronales se decidió utilizar python, que cuenta con librerías como *sklearn*, *keras* y *tensorflow* que permiten la construcción de redes neuronales de una forma sencilla y bastante efectiva. En este algoritmo también se agregaron conceptos importantes para evitar por ejemplo, el *overfitting*. El *overfitting* se caracteriza por formar parte de modelos que representan un buen rendimiento cuando se evalúa con el conjunto de entrenamiento, pero un muy mal rendimiento cuando se prueba con un conjunto de prueba separado al entrenamiento [27]. Utilizando técnicas de normalización de los datos se pudo dar un agregado al entrenamiento pudiendo obtener mejores resultados. También se utilizaron técnicas para mezclar los datos y volverla una muestra más homogénea.

10.2.4. Resultados de entrenamientos

Se realizaron diversas pruebas para poder determinar las mejores características para entrenar, para esto se hicieron validaciones cruzadas de los vectores de características. De cada uno de los algoritmos de clasificación mencionados con anterioridad se tomaron los entrenamientos con mejores resultados y rendimiento. Para los algoritmos de redes neuronales se usaron los mismos parámetros. Función de activación *sigmoid*, 10 capas ocultas de 10 neuronas cada una. Para el algoritmo programado en Python, se utilizaron los parámetros de *learning rate* de 0.0001 y una tolerancia de $1e^{-9}$ para todos los casos. Se decidió utilizar únicamente un solo modelo para pruebas y en cada sesión de entrenamiento se colocó la matriz del modelo a utilizar. Por cada prueba se obtuvo un vector de características de

7 dimensiones. Las características seleccionadas fueron *Cruces por cero*(ZC), *Valor absoluto medio* (MAV), *Curtosis* (KUR), *Varianza*(VAR), *Amplitud de Wilson* (AW) y *Media cuadrática*(SMR).

Entrenamientos primera sesión

Para esta sesión de entrenamiento se utilizaron 119 muestras por cada clase. Los resultados de cada entrenamiento se pueden observar en el Cuadro 3.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	98.7%
Árbol de decisión fino	MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	97.5%
Medium KNN	MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	98.7%
Regresión logística	MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	97.5%
Discriminante lineal	MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	97.5%
SVM (3er grado)	AE,VAR,AW,SMR	Palma arriba, Puño arriba	98.3%
Árbol de decisión fino	AE,VAR,AW,SMR	Palma arriba, Puño arriba	95.8%
Medium KNN	AE,VAR,AW,SMR	Palma arriba, Puño arriba	97.5%
Regresión logística	AE,VAR,AW,SMR	Palma arriba, Puño arriba	97.5%
Discriminante lineal	AE,VAR,AW,SMR	Palma arriba, Puño arriba	96.7%
RN (Python)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	100%
RN (MATLAB)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	96.6

Cuadro 3: Cuadro de resultados clasificación con algoritmos distintos de clasificación señales EMG.

Entrenamientos segunda sesión

Para esta sesión de entrenamiento se utilizaron 238 muestras por cada clase. En estas 238 muestras se incluyen las 119 muestras de la sesión anterior. En el Cuadro 4 se pueden observar los resultados de los entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	95.2%
Árbol de decisión fino	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	92.9%
Medium KNN	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	94.2%
Regresión logística	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	95%
Discriminante lineal	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	93.7%
SVM (3er grado)	AE,VAR,AW,SMR	Palma arriba, Puño arriba	93.5%
Árbol de decisión fino	AE,VAR,AW,SMR	Palma arriba, Puño arriba	90.8%
Medium KNN	AE,VAR,AW,SMR	Palma arriba, Puño arriba	93.5%
Regresión logística	AE,VAR,AW,SMR	Palma arriba, Puño arriba	92.7%
Discriminante lineal	AE,VAR,AW,SMR	Palma arriba, Puño arriba	90.8%
RN (Python)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	100%
RN (MATLAB)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	94.6%

Cuadro 4: Cuadro de resultados clasificación con algoritmos distintos de clasificación señales EMG.

Entrenamientos tercera sesión

Para esta sesión de entrenamiento se utilizaron 357 muestras por cada clase. En estas 357 muestras se incluyen las 238 muestras de las sesiones anteriores. En el Cuadro 5 se pueden

observar los resultados de este entrenamiento.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	89.3 %
Árbol de decisión fino	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	88.2 %
Medium KNN	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	87.2 %
Regresión logística	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	84.8 %
Descriminante linear	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	82.1 %
SVM (3er grado)	AE,VAR,AW,SMR	Palma arriba, Puño arriba	86.1 %
Árbol de decisión fino	AE,VAR,AW,SMR	Palma arriba, Puño arriba	78.6 %
Medium KNN	AE,VAR,AW,SMR	Palma arriba, Puño arriba	86.1 %
Regresión logística	AE,VAR,AW,SMR	Palma arriba, Puño arriba	80.4 %
Descriminante linear	AE,VAR,AW,SMR	Palma arriba, Puño arriba	80.3 %
RN (Python)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	99.2 %
RN (MATLAB)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	87.2 %

Cuadro 5: Cuadro de resultados clasificación con algoritmos distintos de clasificación señales EMG.

Entrenamientos cuarta sesión

Para esta sesión de entrenamiento se utilizaron 476 muestras por cada clase. En estas 479 muestras se incluyen las 357 muestras de las sesiones anteriores. En el Cuadro 6 se pueden observar los resultados de estos entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	88.4 %
Árbol de decisión fino	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	81 %
Medium KNN	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	83.3 %
Regresión logística	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	79.9 %
Descriminante linear	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	79 %
SVM (3er grado)	AE,VAR,AW,SMR	Palma arriba, Puño arriba	54.5 %
Árbol de decisión fino	AE,VAR,AW,SMR	Palma arriba, Puño arriba	76.2 %
Medium KNN	AE,VAR,AW,SMR	Palma arriba, Puño arriba	79.9 %
Regresión logística	AE,VAR,AW,SMR	Palma arriba, Puño arriba	79.9 %
Descriminante linear	AE,VAR,AW,SMR	Palma arriba, Puño arriba	79 %
RN (Python)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	99 %
RN (MATLAB)	ZC,MAV,KUR,AE,VAR,AW,SMR	Palma arriba, Puño arriba	80.7 %

Cuadro 6: Cuadro de resultados clasificación con algoritmos distintos de clasificación señales EMG.

Luego de cuatro sesiones de entrenamientos y utilizando dimensiones diferentes en los vectores de características se pueden determinar ciertas cosas. Por ejemplo, los entrenamientos al contrario de lo esperado, entre más data se le incluía menor eran los porcentajes de rendimiento de clasificación. Esto seguramente es porque los datos recolectados en cada sesión, es bastante variada, esto representa una muestra muy variada de las clases. Recordando que el objetivo del ML no es clasificar exactamente las muestras con las que se entrenó, sino tener un buen rendimiento de clasificación en una muestra bastante variada. Los resultados se consideran aceptables y el modelo que se escogió para probar con la interfaz clasificadora es el modelo de SVM grado 3 con un vector de características de 7 dimensiones. Esto porque presento un buen rendimiento en todos los entrenamientos. Otra cuestión a considerar es que las RN en Python, que presentaron resultados muy superiores a los que presentan las RN

en MATLAB. Esto es seguramente porque en Python se pueden manejar más parámetros y construir las redes neuronales desde un nivel más bajo.

10.2.5. Interfaz clasificadora de señales EMG

Al tener la interfaz recopiladora de datos del Capítulo 9 se procedió a reutilizarla y crear la interfaz, pero estaba vez incluyendo un panel de control para conectarse al robot R17 y así poder probar los modelos entrenados. Los algoritmos de adquisición en esta interfaz son los mismos que se describen en el Capítulo 9, con la diferencia que en esta interfaz no se guardan los datos y solo se clasifica cada muestra. La interfaz sólo clasifica si se detecta actividad, si no no manda ningún comando al robot R17.

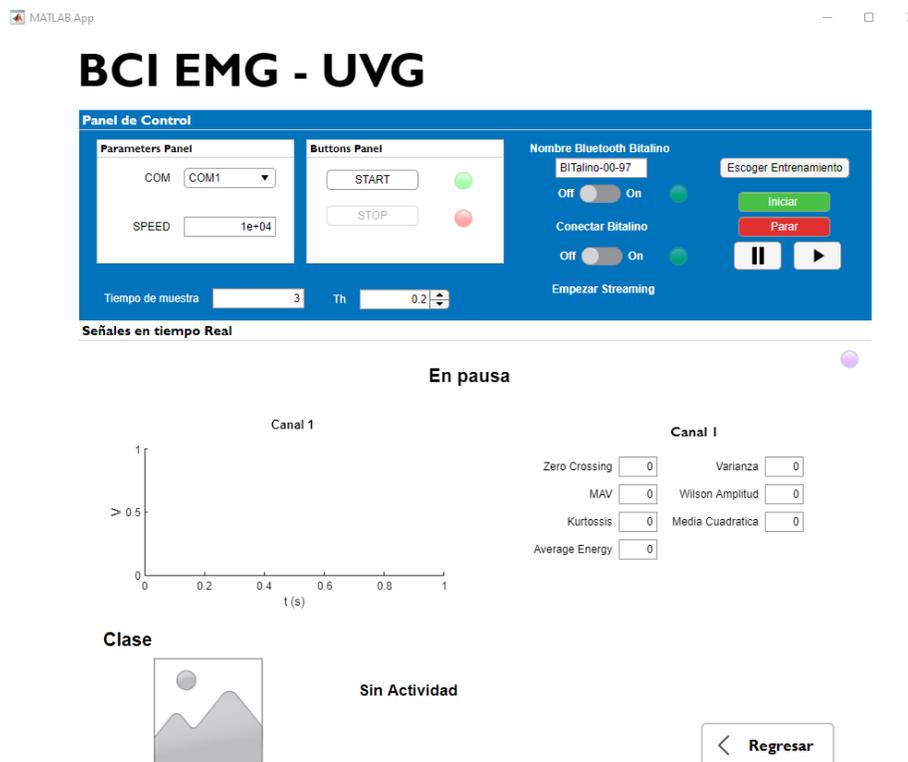


Figura 27: Interfaz EMG

La interfaz permite al usuario escoger el entrenamiento que desee probar. Cada vez que la interfaz clasifica, aparece una imagen ilustrativa y el nombre de la clase que detectó. Dependiendo de la clase, esta envía por serial al Robot R17 un comando. En este caso, se programo de forma que cada vez que la interfaz detecta que es la clase “palma arriba” el robot recorrerá en un paso de 0.2 metros el eje x. Si la interfaz detecta la clase “puño arriba” el robot recorrerá a un paso de 45 grados su eje Z.



Figura 28: Clase puño arriba



Figura 29: Clase palma arriba

El uso de la interfaz se explica en el manual de usuario de la interfaz. En los Anexos [47](#), [48](#), [48](#) y [47](#) se puede encontrar la interfaz “BCI EMG-UVG” siendo utilizada por alumnos del curso de Fisiología 2 de la Universidad del Valle de Guatemala, como parte de su proyecto final del curso, probando así su funcionamiento.

Adquisición y recopilación de señales EEG

En este capítulo se tratarán los temas que tienen que ver con la adquisición y procesamiento de las señales EEG, también sobre los paradigmas experimentales con los que se recopilaron las señales. Se introduce a la interfaz que sirve para la adquisición de señales EEG.

11.1. Dispositivo de adquisición de señales EEG

En la interfaz de la fase previa, por motivos de la pandemia COVID-19, no se pudo utilizar ningún dispositivo para recolectar señales EEG y solo se basó en simulaciones. En este caso se decidió utilizar el dispositivo *Electro-Cap System I* con el que cuenta la Universidad del Valle de Guatemala. Este sistema cuenta con adaptador de placa de electrodos, electrodos de inserción rápida y electrodos de oído. Este cuenta con el sistema de posicionamiento 10-20. Para la adquisición de las señales se utiliza la placa Cyton Board [28], que se conecta con el adaptador de placa de los electrodos del electrocap. La placa se conecta de forma inalámbrica con la computadora a través de un *dongle* USB.

El principal reto que se encontró con el dispositivo electrocap, fue el poder obtener los datos de los 8 canales que puede recibir el cyton board en forma streaming directamente en MATLAB. Esto se solucionó utilizando *lab Streaming Layer* (LSL) una herramienta que permite recibir los datos del Cyton Board en MATLAB de forma *streaming*.

11.1.1. Primeros acercamientos a dispositivo cyton board

Para poder visualizar y hacer un análisis de las señales previo a establecer un paradigma de experimentación, se utilizó la *OpenBCI GUI* para poder tener los primeros acercamientos para la visualización de las señales. Esta interfaz también proporciona la opción al usuario de poder grabar señales. Para poder obtener los datos en MATLAB se utilizaron las funciones con las que cuenta la librería de OpenBCI para MATLAB así como los ejemplos con los que cuenta el repositorio. Para poder recibir los datos se tiene que cargar la librería y luego empezar a recibir los datos, se determinó que con la función `lsl_inlet(result{1})` se recibe cada muestra enviada por el cyton board.



Figura 30: Paquete electro-cap [29]

Se determinó que el cyton board trabaja a una frecuencia máxima de muestreo de 250 Hz, cuenta con un voltaje de operación de 3.3 V y un voltaje de entrada entre 3.3-12 V.

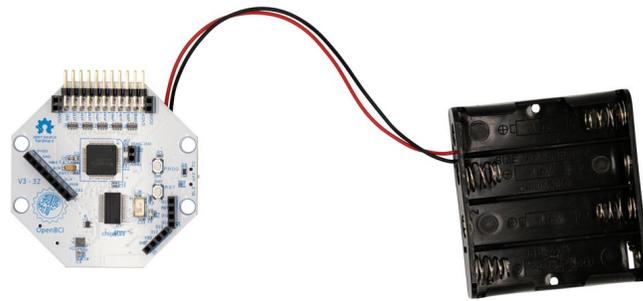


Figura 31: Cyton board [28]

11.2. Adquisición y procesamiento de señales EEG

11.2.1. Algoritmo de adquisición

Como se mencionaba ya en la sección anterior, se utilizó la librería de OpenBCI para la obtención de los datos. En este caso, la obtención de la señal EEG, difiere a la forma en la que se obtenían las señales EMG en el Capítulo 9, ya que caso contrario al dispositivo bitalino, donde se obtienen los datos directamente del búfer, acá se obtiene muestra por muestra. El primer paso para empezar a recibir los datos, es correr el código en Python que permite obtener los datos en tiempo real en MATLAB. Desde la ventana de comandos de la computadora se tienen que correr el código *openbci_matlab.py* y se empezarán a recibir los datos en MATLAB. Estos pasos se detallan mejor en el Manual de usuario para interfaces EEG. En MATLAB al tener cada muestra con la función *lsl_inlet* se procede a recibir los datos y se tienen que recibir los 8 canales $[vec,ts] = inlet.pull_sample()$. Al tener los 8 canales se tiene que hacer una conversión de los datos. Esto se hace con un factor de escala

para la conversión entre datos brutos (recuentos) y potenciales de voltaje. Este factor es de 0.0224 según [28]. Al tener los datos convertidos se empieza a llenar el búfer circular, hasta que se cumpla con el número de épocas establecidas, esto depende directamente de la frecuencia de muestreo que en este caso es de 250 Hz como se mencionó anteriormente. Por ejemplo si se quiere tener 5 épocas, serían 1250 datos en esos 5 segundos. El algoritmo almacena la muestra y obtiene el valor para darle un *offset* a la siguiente llenada del búfer y así poder centrar la señal y no perder datos. Algo importante a considerar es que de los 8 canales para este caso de estudio, se puede utilizar cualquiera, ya que los entrenamientos hacen el supuesto de un individuo que focaliza su concentración, por lo que los diferenciales de voltaje en cualquier sector son parecidos. En este caso en particular se decidió utilizar el canal número 1.

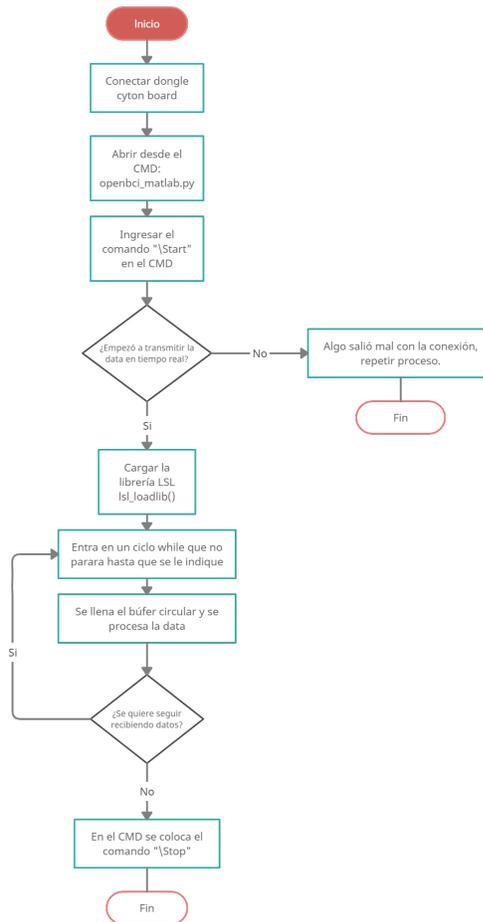


Figura 32: Algoritmo de adquisición de señales EEG

11.2.2. Procesamiento de señal

Contando son la señal a analizar, como se mencionó en la sección anterior, se le da un factor de escalamiento y además realizar un filtrado a esta señal. Al contrario de las señales EMG donde se tiene una banda de frecuencias donde dichas señales trabajan en la señal EEG se tiene algo distinto. Se tiene que seleccionar la banda de frecuencia según el objeto

de estudio. En este caso en específico se utilizaron las ondas Alpha (8-12 Hz) para filtrar la señal. Para esto se utilizó el *Signal Processing Toolbox* de MATLAB para crear un filtro butter pasa banda de grado 2 con frecuencias de corte en 7 y 14 Hz. Luego esta señal se volvió a filtrar, pero esta vez con un filtro butter quita banda de grado 2 con frecuencias de corte entre 59 y 61 Hz para eliminar el ruido de la red eléctrica.

11.2.3. Algoritmo para la identificación de actividad

En este caso, el algoritmo de detección de actividad ya no es tan fácil de implementar, es más complejo detectar a través de las señales EEG si hay o no hay actividad. Recordando que esto sirve mucho por el caso particular que se está tratando acá que es el del que se estarán utilizando sistemas robóticos físicos, por lo que el clasificador que se entrenó más adelante no tendría que enviar comandos si no se está haciendo nada. Una solución fue utilizar el parámetro de actividad de *HJORTH* como un umbral de actividad, ya que representa el poder de la señal, la varianza en función del tiempo. La otra solución que se encontró fue agregar a los entrenamientos una clase de relajamiento, donde el usuario no esté haciendo ninguna actividad física.

11.3. Interfaz para la recolección de datos EEG

Al igual que con las señales EMG se decidió crear una interfaz que permitiera la recolección de datos de una forma más intuitiva y fácil. La interfaz también fue creada en MATLAB con *App Designer*. Se utilizaron los algoritmos descritos en las secciones anteriores de este capítulo. Al igual que la interfaz para recolectar datos EMG, esta permite recolectar los datos en vector de características y también los datos en bruto. El uso detallado e instrucciones de esta interfaz se puede encontrar en el manual de usuario del Anexo [53](#).

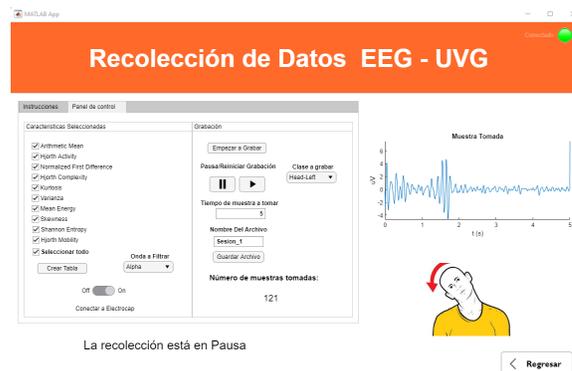


Figura 33: Interfaz recolectora de datos EEG

11.4. Paradigma experimental para la recolección de datos EEG

Para la recolección de datos se estableció el paradigma experimental con el cual se tomaron las muestras de las clases con las que se entrenarían los modelos para la interfaz clasificadora. El experimento se diseñó para adquirir los datos cualitativamente relacionados con los dos movimientos de la cabeza y con no realizar ningún movimiento y estar en modo

relajación. Se realizaron 4 sesiones de 120 pruebas con cada clase. Al principio de cada prueba el sujeto descansa 2 segundos y en esos 2 segundos se proyectaba una ayuda visual del movimiento o clase a grabar. Después de estos 2 segundos se activa una luz en la interfaz por 1 segundo, que indica que es momento de realizar el movimiento. Al terminar de realizar el movimiento, el sujeto descansa por 2 segundos antes de empezar la siguiente prueba.

- **Inclinación derecha** - Este movimiento constaba de inclinar la cabeza hacia la derecha viendo hacia en frente durante 1 segundo.
- **Giro izquierda** - Este movimiento constaba de girar la cabeza hacia la izquierda durante 1 segundo.
- **Relajado**- En esta clase, no se realizaba ningún tipo de movimiento, se escuchaba música, se veían vídeos o se utilizaba el celular.

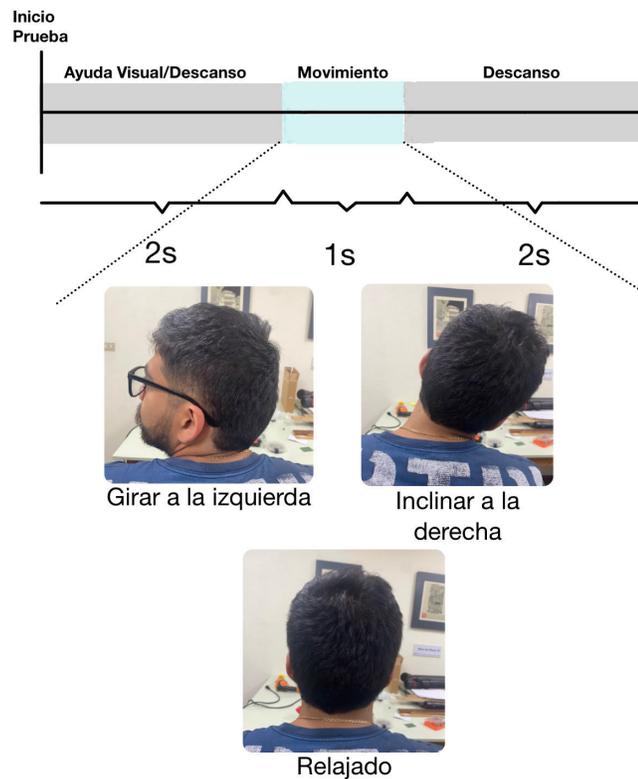


Figura 34: Paradigma experimental

Interfaz clasificadora de señales EEG

Para este caso, también existen fases anteriores donde se probaron clasificadores en interfaces, pero nunca usando el casco electro cap, ya que todo había sido hecho por simulaciones y datos recopilados para otros proyectos de graduación. En este capítulo se plantean criterios y métricas de evaluación para estas fases previas. En este capítulo también se muestran los resultados de los entrenamientos con los datos recopilados, mencionada en el Capítulo [11](#). También se muestra el funcionamiento de la interfaz con entrenamientos propios utilizando el robot R17 de la Universidad del Valle de Guatemala. El uso detallado e instrucciones de esta interfaz se puede encontrar en el manual de usuario del Anexo [53](#).

12.1. Criterios y métricas de evaluación para la interfaz EEG de fases previas

Para poder obtener un área de mejora en la interfaz para la adquisición de señales EEG, se hicieron varias pruebas con la interfaz de fases previas y se determinaron las áreas de mejora en las que se tenía que enfocar esta parte del desarrollo de la interfaz.

- **Conexión con electro cap** Una de las deficiencias más grandes que tenía la interfaz de fases previas, es que está no se conectaba al dispositivo electro cap, ya que por motivos de la pandemia COVID-19, no se pudo utilizar este dispositivo en la interfaz. Por lo que este era un punto que se tenía que cubrir obligatoriamente.
- **Conexión con el robot R17** Otra carencia con la que contaba la interfaz para señales EEG, era que no tenía opción para poder conectarse con el robot R17 y así poder probar los modelos entrenados.
- **Amigable para el usuario** La interfaz tiene que ser de mucha utilidad para los usuarios que quieran probar los modelos entrenados, por lo que se tienen que mejorar la forma en la que el usuario interactuaba con esta. Por ejemplo agregar opciones de filtrado de ondas, obtener información valiosa como si existe actividad o no, seleccionar el entrenamiento sin tener que tocar el código interno de la interfaz y una forma más interactiva de realizar la conexión con el electro cap. Todo esto para que el usuario no tenga que modificar muchas cosas en el código fuente.

- **Manual de usuario** La interfaz no cuenta con un manual de usuario completo, para poder determinar los fallos que pueda tener y como se solucionan. Este manual de usuario tendría que tener en cuenta consideraciones como que los umbrales para la detección de actividades, cambian para los diferentes usuarios de la interfaz.

12.2. Clasificación de señales EEG

Se recolectaron alrededor de 480 muestras por cada clase que se planteó en el paradigma experimental del Capítulo 11 en 4 sesiones distintas. Para entrenar los modelos se decidió utilizar la misma metodología utilizada en Capítulo 10 donde se utilizaron *classification learner MATLAB*, *neural net fitting MATLAB* y las redes neuronales en python.

12.2.1. Resultados de entrenamientos

Se realizaron diversas pruebas para poder determinar las mejores características para entrenar, para esto se hicieron validaciones cruzadas de los vectores de características. Además, en el caso de las señales EEG, se decidió utilizar la transformada de wavelet utilizando el paquete de descomposición de wavelet de MATLAB. Se aplicó la descomposición sobre las señales EEG, de los nodos (5 2), (5 3), (5 4), (5 5), (5 6) y (5 7) que representan las frecuencias de 8Hz a 32 Hz. De cada uno de los algoritmos de clasificación mencionados con anterioridad se tomaron los entrenamientos con mejores resultados y rendimiento. Para los algoritmos de redes neuronales se usaron los mismos parámetros. Función de activación *sigmoid*, 10 capas ocultas de 10 neuronas cada una. Para el algoritmo programado en Python, se utilizaron los parámetros de *learning rate* de 0.0001 y una tolerancia de 1×10^{-9} para todos los casos. Por cada prueba se obtuvo un vector de características en el dominio del tiempo de 10 dimensiones. Las características seleccionadas fueron *HjorthActivity*(HA), *Normalized First Difference* (NFD), *Hjorth Complexity* (HC), *Kurtosis*(KUR), *Mean Energy* (ME), *Skewness*(Sk), *Shannon Entropy*(SE), *HjorthMobility*(HM) y *Principal Component Analysis*(PCA). El vector de características utilizado con la transformada de wavelet fue *Arithmetic Mean* (AM), *Shannon Entropy* (SE) y *Standard Deviation* (SD)

Entrenamientos primera sesión

Para esta sesión de entrenamiento, se recolectaron 119 datos por cada una de las clases. Se hicieron entrenamientos utilizando las clases “Inclinación derecha”, “Giro izquierda”. También se hicieron entrenamientos utilizando las tres clases propuestas en el paradigma experimental, “Inclinación derecha”, “Giro izquierda” y “Relajado”. En el Cuadro 7 se pueden encontrar los resultados de estos entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (2do grado)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	89.3 %
Árbol de decisión fino	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	91.6 %
Medium KNN	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	87.2 %
Regresión logística	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	80.7 %
Discriminante lineal	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	81.1 %
SVM (2do grado)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	90.2 %
Árbol de decisión fino	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	88.2 %
Medium KNN	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	84.9 %
Discriminante lineal	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	81.5 %
RN (Python)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	81.7 %
RN (MATLAB)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	79.4 %
SVM (2do grado)	Wavelet packet decomposition	Inclinación derecha, Giro izquierda	88.3 %

Cuadro 7: Cuadro de resultados clasificación primera sesión con algoritmos distintos de clasificación señales EEG.

Entrenamientos segunda sesión

Para esta sesión de entrenamiento, se recolectaron 119 datos por cada una de las clases. Se agregaron al entrenamiento los 119 datos por clase que se recopilaban en la primera sesión, dando un total de 238 datos por clase. Se hicieron entrenamientos utilizando las clases Inclinación derecha, Giro izquierda. También se hicieron entrenamientos utilizando las tres clases propuestas en el paradigma experimental, Inclinación derecha, Giro izquierda y Relajado. En el Cuadro 8 se pueden encontrar los resultados de estos entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (2do grado)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	92.9 %
Árbol de decisión fino	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	89.9 %
Medium KNN	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	85.7 %
Regresión logística	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	80.7 %
Discriminante lineal	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	72.9 %
SVM (2do grado)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	90.5 %
Árbol de decisión fino	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	88.1 %
Medium KNN	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	86.3 %
Discriminante lineal	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	76.5 %
RN (Python)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	88 %
RN (MATLAB)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	85.3 %
SVM (2do grado)	Wavelet packet decomposition	Inclinación derecha, Giro izquierda	93.5 %

Cuadro 8: Cuadro de resultados clasificación segunda sesión con algoritmos distintos de clasificación señales EEG.

Entrenamientos tercera sesión

Para esta sesión de entrenamiento, se recolectaron 119 datos por cada una de las clases. Se agregaron al entrenamiento los 238 datos por clase que se recopilaron en las primeras sesiones, dando un total de 357 datos por clase. Se hicieron entrenamientos utilizando las clases Inclinación derecha, Giro izquierda. También se hicieron entrenamientos utilizando las tres clases propuestas en el paradigma experimental, Inclinación derecha, Giro izquierda y Relajado. En el Cuadro 9 se pueden encontrar los resultados de estos entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (2do grado)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	86.8 %
Árbol de decisión fino	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	86.1 %
Medium KNN	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	82.4 %
Regresión logística	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	75.4 %
Descriminante linear	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	74.1 %
SVM (2do grado)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	86.6 %
Árbol de decisión fino	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	82.6 %
Medium KNN	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	83.6 %
Descriminante linear	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	74.2 %
RN (Python)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	81 %
RN (MATLAB)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	77.5 %
SVM (2do grado)	Wavelet packet decomposition	Inclinación derecha, Giro izquierda	85.5 %

Cuadro 9: Tabla de resultados clasificación tercera sesión con algoritmos distintos de clasificación señales EEG.

Entrenamientos cuarta sesión

Para esta sesión de entrenamiento, se recolectaron 119 datos por cada una de las clases. Se agregaron al entrenamiento los 357 datos por clase que se recopilaron en las primeras sesiones, dando un total de 476 datos por clase. Se hicieron entrenamientos utilizando las clases “Inclinación derecha”, “Giro izquierda”. También se hicieron entrenamientos utilizando las tres clases propuestas en el paradigma experimental, “Inclinación derecha”, “Giro izquierda” y “Relajado”. En el Cuadro 10 se pueden encontrar los resultados de estos entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (2do grado)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	81.1 %
Árbol de decisión fino	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	82.2 %
Medium KNN	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	80.6 %
Regresión logística	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	71.7 %
Descriminante linear	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	72.3 %
SVM (2do grado)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	86 %
Árbol de decisión fino	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	84 %
Medium KNN	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	84.1 %
Descriminante linear	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda, Relajado	74.5 %
RN (Python)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	79 %
RN (MATLAB)	HA,NFD,HC,KUR, ME,SK,SE,HM,PCA	Inclinación derecha, Giro izquierda	76.5 %
SVM (3er grado)	Wavelet packet decomposition	Inclinación derecha, Giro izquierda	89.4 %

Cuadro 10: Cuadro de resultados clasificación cuarta sesión con algoritmos distintos de clasificación señales EEG.

Luego de cuatro sesiones de entrenamientos y utilizando un vector de características en el dominio del tiempo de 10 dimensiones, también utilizando el *Wavelet packet decomposition* para entrenar se pueden concluir ciertas cosas. Por ejemplo, los entrenamientos al contrario de lo esperado, entre más data se le incluía menor eran los porcentajes de rendimiento de clasificación. Esto seguramente es porque los datos recolectados en cada sesión, es bastante variada, esto representa una muestra muy variada de las clases. Al contrario de lo que paso con las señales EMG donde las RN en Python tuvieron un alto rendimiento comparado con todos los otros modelos en general, se notó un menor rendimiento cada vez que se agregaban más datos. En general como se puede observar en los cuadros 7, 8, 9 y 10, entre más datos se agregaban a los modelos, el rendimiento baja. Esto pasó con casi todos los modelos y utilizando 2 y 3 clases. A excepción del modelo donde se usó el *Wavelet packet decomposition* para entrenar, que presentó un mejor rendimiento en la última sesión de entrenamiento. Esto da pauta a poder escoger entre dos de los mejores entrenamientos, siendo estos SVM

de segundo grado con el vector de características de 10 dimensiones y utilizando el SVM de tercer grado usando *Wavelet packet decomposition*. Para escoger entre ambos entrenamientos, se tuvo que determinar si el rendimiento de clasificación es lo suficientemente bueno como para aceptar ese tiempo extra que generaba usar *Wavelet packet decomposition*, ya que su procesamiento es caro computacionalmente como para implementarse en tiempo real.

12.2.2. Interfaz clasificadora de señales EEG

Al tener la interfaz recopiladora de datos del capítulo 11 se procedió a reutilizarla y crear la interfaz, pero estaba vez incluyendo un panel de control para conectarse al robot R17 y así poder probar los modelos entrenados. Los algoritmos de adquisición en esta interfaz son los mismos que se describen en el Capítulo 11, con la diferencia que en esta interfaz no se guardan los datos y solo se clasifica cada muestra. La interfaz solo clasifica si se detecta actividad, si no no manda ningún comando al robot R17.

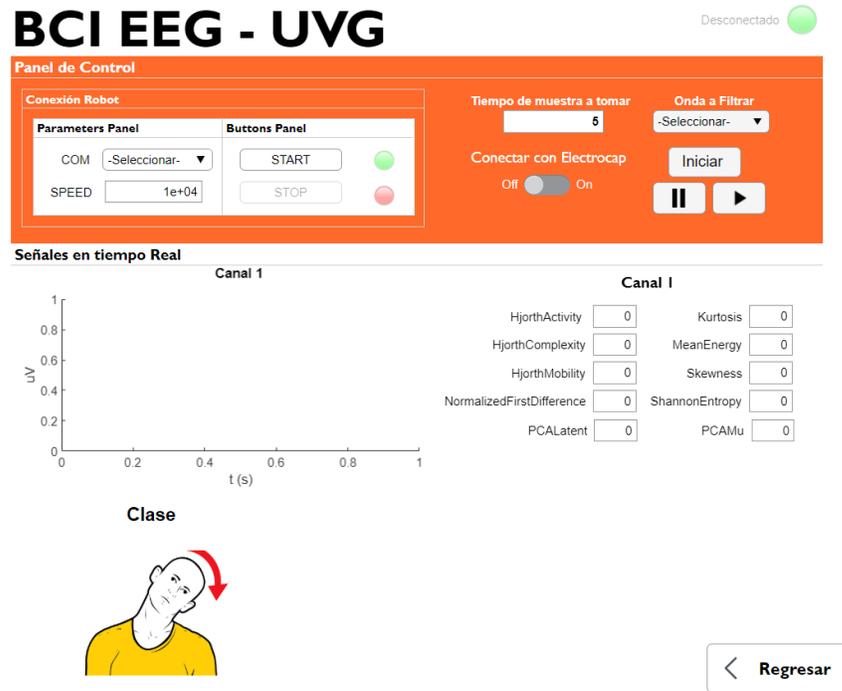


Figura 35: Interfaz clasificadora EEG

Combinación de señales EMG y EEG

En este capítulo se tratarán las interfaces de clasificación desde una perspectiva que amplíe la visión de investigación sobre la utilidad que tienen las señales biomédicas. En resumen se indagará en el peso o en el valor que agrega incluir a un vector de características las señales EEG, cuando se está llevando a cabo un evento que involucre las señales EMG como principal entrada a los vectores de características. También se tratará la combinación de señales como una forma de tener dos dispositivos para aumentar la cantidad de clases y así poder tener opción a tener una mayor cantidad de comandos que enviar.

13.1. Interfaz recolectora de datos EEG y EMG

Para desarrollar esta interfaz, se utilizaron los algoritmos desarrollados en los Capítulos 9 y 11 para poder implementar la interfaz recolectora de datos. Por la forma en la que funcionaban los algoritmos de las interfaces recolectoras de datos EEG y EMG, implementarlo en una interfaz donde se recolectarán ambas señales fue sencillo. Esto se debe a que el que dicta en qué momento terminan de guardarse los datos es el algoritmo de adquisición de la señal EEG, ya que al llenarse el búfer guarda todos los datos. Esto implica que cuando este se llene, se tienen que tomar los datos recolectados en el búfer del dispositivo bitalino, como se indicaba en el algoritmo de adquisición de las señales EMG. En cuanto a los algoritmos de adquisición depende del enfoque. Si es el primer enfoque, la detección de actividad la dicta la señal EMG. Si es el segundo enfoque, la detección de actividad es una combinación de ambos algoritmos y funciona como una compuerta OR. En cuanto al tercer enfoque se parece mucho al primero, pero se utiliza una zona diferente del brazo para poder recolectar las señales.

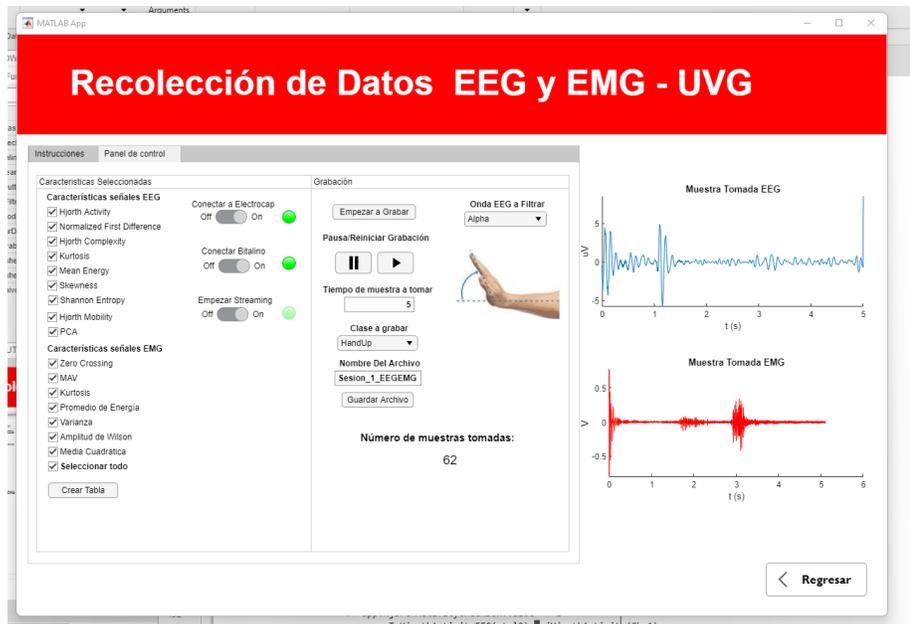


Figura 36: Interfaz recolectora de datos EEG-EMG

13.2. Paradigma experimental primer enfoque

El primer enfoque utilizado es prácticamente el mismo que se utilizó en el Capítulo 9. Otra vez el sujeto descansa los primeros 2 segundos con una ayuda visual proyectada. Luego de estos dos segundos la interfaz encendía una luz indicando que era momento de realizar el movimiento, luego el individuo descansa otros 2 segundos.

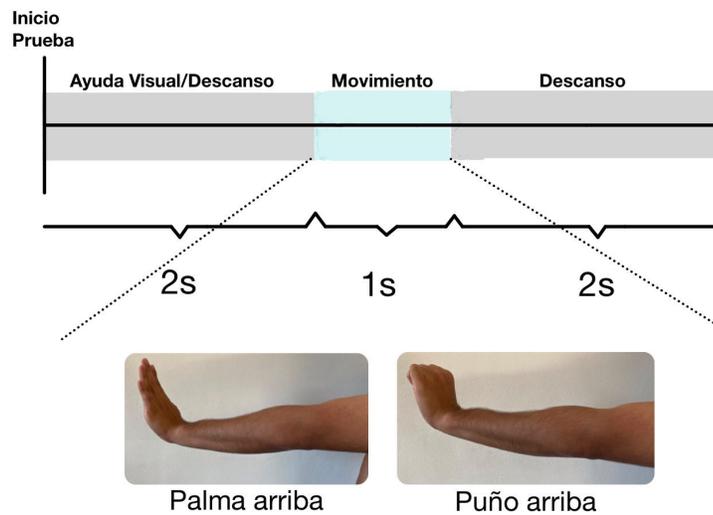


Figura 37: Paradigma experimental primer enfoque

13.3. Clasificación de señales EEG-EMG, primer enfoque

Para esta clasificación se recolectaron alrededor de 438 muestras por cada clase que se planteó en el paradigma experimental planteado en la sección anterior. Para entrenar los modelos se decidió utilizar la misma metodología utilizada en el Capítulo 10 pero esta vez solo utilizando *classification learner MATLAB*.

13.3.1. Resultados de entrenamientos

En este caso se decidió utilizar para las señales EEG el mismo vector de características de 10 dimensiones que ya se usaba en el Capítulo 12 que son *HjorthActivity*(HA), *Normalized First Difference* (NFD), *Hjorth Complexity* (HC), *Kurtosis*(KUR), *Mean Energy* (ME), *Skewness*(Sk), *Shannon Entropy*(SE), *HjorthMobility*(HM) y *Principal Component Analysis*(PCA). También se decidió utilizar el paquete de descomposición de wavelet como vector de características de las señales EEG. En el caso de las señales EMG también se decidió utilizar el vector de características de 7 dimensiones utilizado en el Capítulo 10 que son *Zero Crossing*(ZC), *Mean Absolute Value* (MAV), *Kurtosis* (KUR), *Variance*(VAR), *Wilson Amplitude* (AW) y *Root Mean Square*(SMR).

Entrenamientos primera sesión

Para esta sesión de entrenamiento, se recolectaron 219 datos por cada una de las clases. Se hicieron entrenamientos utilizando las clases “palma arriba” y “puño arriba”. En el Cuadro 11 se pueden observar los resultados de los entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	87.2 %
Árbol de decisión fino	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	84.2 %
Regresión logística	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	89.3 %
SVM (3er grado)	EEG: Wavelet packet decomposition EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	83.8 %
Árbol de decisión fino	EEG: Wavelet packet decomposition EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	79.9 %
Regresión logística	EEG: Wavelet packet decomposition EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	89.3 %
SVM (3er grado)	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	51.1 %
Árbol de decisión fino	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	84.2 %
Regresión logística	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	87.4 %

Cuadro 11: Cuadro de resultados clasificación primera sesión con algoritmos distintos de clasificación señales EEG-EMG.

Entrenamientos segunda sesión

Para esta sesión de entrenamiento, se recolectaron 219 datos por cada una de las clases y se agregaron los primeros 219 datos. Se hicieron entrenamientos utilizando las clases palma arriba y puño arriba. En el Cuadro 12 se pueden observar los resultados de los entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	94.6 %
Árbol de decisión fino	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	87.9 %
Regresión logística	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	92.4 %
SVM (3er grado)	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	91.3 %
Árbol de decisión fino	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	89.3 %
Regresión logística	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Palma arriba, Puño arriba	88.1 %

Cuadro 12: Cuadro de resultados clasificación segunda sesión con algoritmos distintos de clasificación señales EEG-EMG.

Partiendo de los supuestos en los que se mencionaba que las señales EEG aportan un valor agregado a las señales EMG, dándole mejor rendimiento a los clasificadores entrenados utilizando ambas señales. Se puede deducir de los resultados de los entrenamientos que se pueden observar en los Cuadros 11 y 12, que en efecto cuando uno combina las señales EMG y EEG al tener una cantidad de datos menores, un vector de características de las señales EEG grabas en ese momento, si aportan un peso y mejoran el rendimiento de los clasificadores. Cuando el número de muestras aumentan, los vectores de características de las señales EEG dejan de mejorar el rendimiento de la clasificación de los entrenamientos. Incluso el rendimiento sin los vectores de características extraídos de las señales EEG el rendimiento de los clasificadores mejoran.

13.4. Paradigma experimental segundo enfoque

El segundo enfoque utilizado es prácticamente unir los paradigmas experimentales que se utilizaron en el Capítulo 9 y el Capítulo 11. Otra vez el sujeto descansa los primeros 2 segundos con una ayuda visual proyectada. Luego de estos dos segundos la interfaz encendía una luz indicando que era momento de realizar el movimiento, luego el individuo descansa otros 2 segundos.

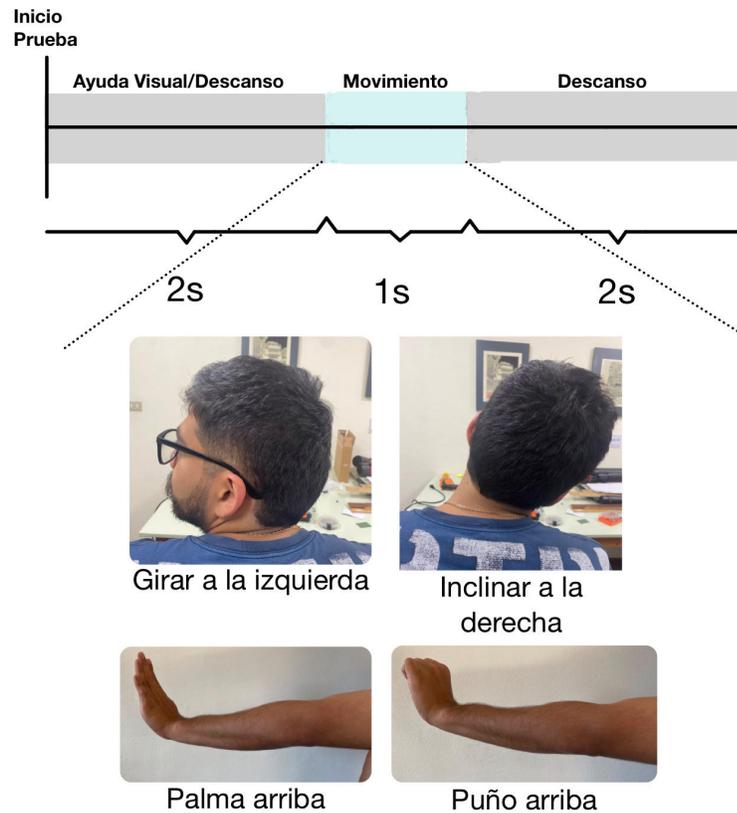


Figura 38: Paradigma experimental segundo enfoque

13.5. Clasificación de señales EEG-EMG, segundo enfoque

Para esta clasificación se recolectaron alrededor de 438 muestras por cada clase que se planteó en el paradigma experimental planteado en la sección anterior. Para entrenar los modelos se decidió utilizar la misma metodología utilizada en el Capítulo 10 pero esta vez solo utilizando *classification learner MATLAB*.

13.5.1. Resultados de entrenamientos

En este caso se decidió utilizar para las señales EEG el mismo vector de características de 10 dimensiones que ya se usaba en el Capítulo 12 que son *HjorthActivity*(HA), *Normalized First Difference* (NFD), *Hjorth Complexity* (HC), *Kurtosis*(KUR), *Mean Energy*

(ME), *Skewness*(Sk), *Shannon Entropy*(SE), *HjorthMobility*(HM) y *Principal Component Analysis*(PCA). En el caso de las señales EMG también se decidió utilizar el vector de características de 7 dimensiones utilizado en el Capítulo 10 que son *Zero Crossing*(ZC), *Mean Absolute Value* (MAV), *Kurtosis* (KUR), *Variance*(VAR), *Wilson Amplitude* (AW) y *Root Mean Square*(SMR). En el Cuadro 13 se pueden observar los resultados de los entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Inclinación derecha, Giro izquierda, Palma arriba, Puño arriba	89.5 %
Árbol de decisión fino	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Inclinación derecha, Giro izquierda, Palma arriba, Puño arriba	86.6 %
Medium KNN	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Inclinación derecha, Giro izquierda, Palma arriba, Puño arriba	87.1 %
Discriminante linear	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Inclinación derecha, Giro izquierda, Palma arriba, Puño arriba	84.9 %
Bagged Trees	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Inclinación derecha, Giro izquierda, Palma arriba, Puño arriba	90.5 %

Cuadro 13: Cuadro de resultados clasificación primera con algoritmos distintos de clasificación señales EEG-EMG.

En general se presentó un buen rendimiento para cada uno de los entrenamientos, se confirma que la combinación de señales EEG y EMG se puede llevar a cabo, partiendo desde el hecho de tener 4 clases distintas donde 2 clases dependen directamente del sensor de señales EMG (bitalino) y las otras 2 clases dependen directamente del sensor de señales EEG (electro cap).

13.6. Paradigma experimental, tercer enfoque

Para la recolección de datos en el tercer enfoque se estableció un paradigma experimental con el cual se tomaron las muestras de dos nuevas clases. El experimento se diseñó para adquirir los datos cualitativamente relacionados con dos movimientos asociados a los músculos

que se encuentran por arriba del codo. La razón de realizar este paradigma es para poder realizar un acercamiento a un caso real de gente con extremidades amputadas por encima del hombro como se realiza en [30]. Se realizaron 3 sesiones diferentes, con 130 muestras cada una de las sesiones. Al igual que los otros paradigmas experimentales, se utilizaron los primeros 2 segundos de descanso, después de esos 2 segundos, se tiene que hacer el movimiento por aproximadamente 1 segundo, luego de eso se tiene 2 segundos de descanso antes de empezar la siguiente prueba. A continuación se explican las dos nuevas clases.

- **Girar brazo** - Este movimiento constaba de girar el tríceps a favor de las agujas del reloj durante 1 segundo.
- **Contraer tríceps** - Este movimiento constaba de apretar el tríceps durante aproximadamente 1 segundo.

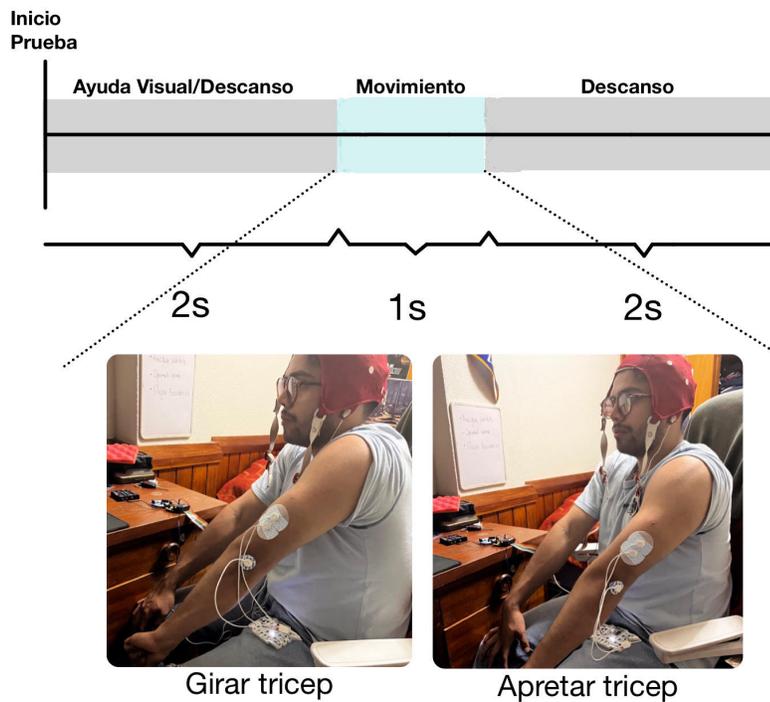


Figura 39: Paradigma experimental tercer enfoque

13.7. Clasificación de señales EEG-EMG, tercer enfoque.

Para esta clasificación se recolectaron alrededor de 390 muestras por cada clase que se planteó en el paradigma experimental de la sección anterior. Para entrenar los modelos se decidió utilizar la misma metodología utilizada en el Capítulo 10 pero esta vez solo utilizando *classification learner MATLAB*.

13.7.1. Resultados de entrenamientos

En este caso se decidió utilizar para las señales EEG el mismo vector de características de 10 dimensiones que ya se usaba en el Capítulo 12 que son *HjorthActivity(HA)*, *Nor-*

malized First Difference (NFD), Hjorth Complexity (HC), Kurtosis(KUR), Mean Energy (ME),Skewness(Sk), Shannon Entropy(SE), HjorthMobility(HM) y Principal Component Analysis(PCA). En el caso de las señales EMG también se decidió utilizar el vector de características de 7 dimensiones utilizado en el Capítulo 10 que son Zero Crossing(ZC), Mean Absolute Value (MAV), Kurtosis (KUR), Variance(VAR), Wilson Amplitude (AW) y Root Mean Square(SMR).

Entrenamientos primera sesión

Para esta sesión de entrenamiento, se recolectaron 130 datos por cada una de las clases y se agregaron los primeros 130 datos. Se hicieron entrenamientos utilizando las clases Contraer tríceps y Girar brazo . En el Cuadro 14 se pueden observar los resultados de los entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	98.1 %
Árbol de decisión fino	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	97.7 %
Medium KNN	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	96.2 %
Discriminante linear	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	95.4 %
Bagged Trees	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	98.5 %

Cuadro 14: Cuadro de resultados clasificación primera sesión con algoritmos distintos de clasificación señales EEG-EMG.

Entrenamientos segunda sesión

Para esta sesión de entrenamiento, se recolectaron 130 datos por cada una de las clases y se agregaron los primeros 130 datos, dando un total de 260 datos. Se hicieron entrenamientos utilizando las clases Girar brazo y Contraer tríceps. En el Cuadro 15 se pueden observar los resultados de los entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	92.3 %
Árbol de decisión fino	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	85.2 %
Medium KNN	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	89.0 %
Discriminante linear	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	90.4 %
Bagged Trees	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	91.5 %

Cuadro 15: Cuadro de resultados clasificación segunda sesión con algoritmos distintos de clasificación señales EEG-EMG.

Entrenamientos tercera sesión

Para esta sesión de entrenamiento, se recolectaron 130 datos por cada una de las clases y se agregaron las primeras 260 pruebas, teniendo un total de 390 pruebas por clase. Se hicieron entrenamientos utilizando las clases Contraer tríceps y Girar brazo. En el Cuadro 16 se pueden observar los resultados de los entrenamientos. Para esta última sesión también se entrenaron los modelos únicamente con las características obtenidas de las señales EMG, para poder realizar una comparación entre entrenamientos.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	90.9 %
Árbol de decisión fino	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	86.5 %
Medium KNN	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	86.4 %
Discriminante linear	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	85.3 %
Bagged Trees	EEG: HA,NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	89.6 %
SVM (3er grado)	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	85.4 %
Árbol de decisión fino	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	77.7 %
Medium KNN	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	81.9 %
Discriminante linear	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	83.2 %
Bagged Trees	EMG: ZC,MAV,KUR,AE, VAR,AW,SMR	Girar, Contraer	83.7 %

Cuadro 16: Cuadro de resultados clasificación tercera sesión con algoritmos distintos de clasificación señales EEG-EMG.

Como se puede observar en el Cuadro 16 que todos los entrenamientos con las dos señales combinadas, representan un mejor resultado que los entrenamientos con solo la señal EMG. Es importante considerar que cuando se usan ambas señales, se tiene un vector de 17 características, esto puede representar un problema de dimensionalidad o *curse of dimensionality*, que como se menciona en [31], se refiere a la dificultad de encontrar una estructura oculta

cuando el número de características es grande. Para solucionar esto, se procedió a utilizar un método de selección de características. El método fue la “Clasificación de características univariante para la clasificación mediante pruebas de chi-cuadrado” como se hizo en [32]. Se realizó un *Live script* en el software MATLAB, para que pueda ser utilizado con cualquier conjunto de datos, se utilizaron los paquetes de análisis de datos brindados por el software MATLAB y se procedió a hacer el análisis de los pesos de las características del conjunto de datos utilizado en la tercera sesión. En el *Live script* del Anexo 51 se puede encontrar el funcionamiento completo.

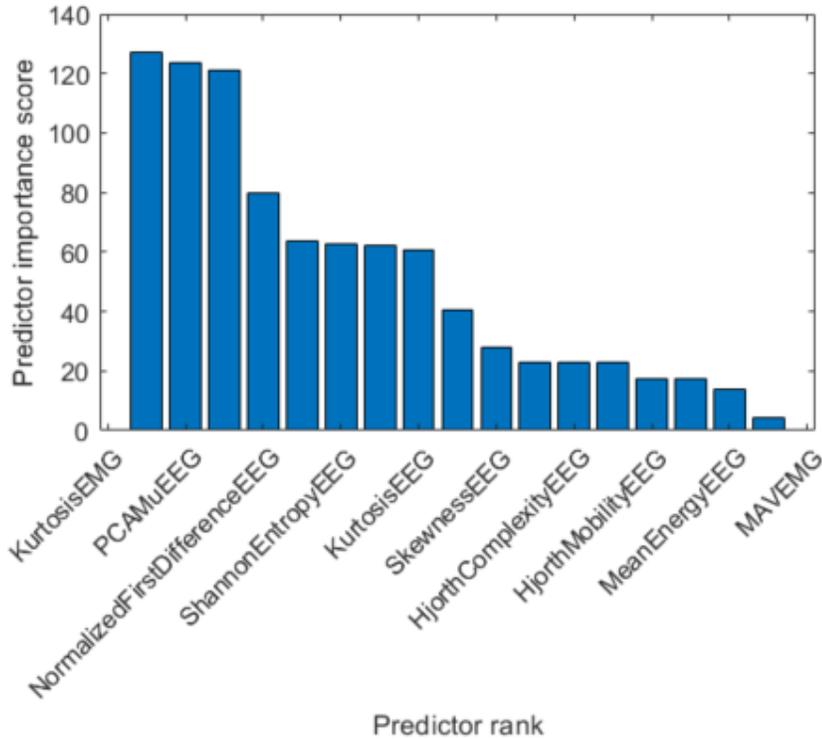


Figura 40: Resultados de los pesos en de pruebas de chi-cuadrado.

Al obtener estos resultados, se procedió a realizar los mismos entrenamientos, esta vez con un vector de 10 características seleccionadas, siendo estas 10 características, las que tenían mayor pesos en las pruebas de chi-cuadrado.

Algoritmo	Vector de características	Clases	Rendimiento
SVM (3er grado)	EEG: NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: MAV,KUR	Girar, Contraer	90.6 %
Árbol de decisión fino	EEG: NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: MAV,KUR	Girar, Contraer	84.7 %
Medium KNN	EEG: NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: MAV,KUR	Girar, Contraer	87.1 %
Discriminante linear	EEG: NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: MAV,KUR	Girar, Contraer	80.5 %
Bagged Trees	EEG: NFD,HC,KUR, ME,SK,SE,HM,PCA EMG: MAV,KUR	Girar, Contraer	88.6 %

Cuadro 17: Cuadro de resultados clasificación segunda sesión con algoritmos distintos de clasificación señales EEG-EMG.

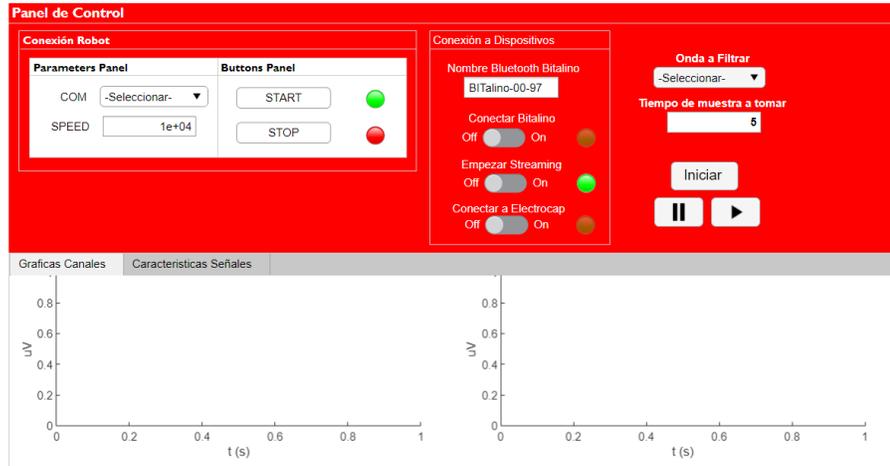
Si se comparan los Cuadros 16 y 17 se puede observar que a pesar de que todos los modelos que se entrenaron utilizando un vector 17 de características tienen un porcentaje mayor de rendimiento, esta diferencia no es tan significativa. Por ejemplo utilizando SVM de tercer grado y un vector de 17 características para entrenar se obtiene un 90.9% de rendimiento y utilizando un vector de 10 características para entrenar, se obtiene un 90.6% de rendimiento. Es preferible utilizar modelos de menor dimensionalidad ya que esto evita tener problemas como por ejemplo el denominado efecto frontera. La idea de este problema es que en la “frontera” del rango de valores de una variable explicativa se va a disponer de pocos datos y los errores de predicción van a tener gran variabilidad (se están haciendo extrapolaciones de los datos, más que interpolaciones, y van a ser menos fiables) 33.

13.7.2. Interfaz clasificadora de señales EEG-EMG

Al tener las interfaces clasificadoras del Capítulo 10 y Capítulo 12 se procedió a reutilizarlas y crear la interfaz y así poder probar los modelos entrenados para la combinación de señales. Los algoritmos de adquisición en esta interfaz son los mismos que se describen en los Capítulos 9 y Capítulo 11, con la diferencia que en esta interfaz no se guardan los datos y solo se clasifica cada muestra. La interfaz solo clasifica si se detecta actividad, si no, no manda ningún comando al robot R17. La detección de actividad en esta interfaz, la determina la señal EMG.

BCI EEG EMG - UVG

Desconectado 



< Regresar

Figura 41: Interfaz clasificadora EEG-EMG

- Los algoritmos de adquisición de las señales EMG y EEG, cumplen de manera correcta y eficiente con la tarea de capturar las señales.
- Al incluir más datos a los entrenamientos, los porcentajes de rendimiento en la mayoría de los clasificadores, disminuye. Esto debido a que las muestras se vuelven más variadas, las sesiones de recopilación de datos no pudieron ser exactamente iguales por lo que las condiciones presentarán variaciones.
- Se determinó que para señales EMG usar un vector de características de menor dimensión presentan porcentajes de rendimiento aceptables.
- Les redes neuronales desarrolladas en Python son más eficientes que las redes neuronales desarrolladas en MATLAB, ya que Python si permite una configuración de más bajo nivel que en MATLAB.
- El algoritmo de detección de actividad para señales EMG desarrollado utilizando la transformada continua de wavelet demostró ser eficiente para detectar estos picos de actividad.
- Los clasificadores desarrollados para las señales EMG, demostraron clasificar de una manera correcta las clases entrenadas.
- La conexión y envío de comandos al robot R17 funciono de manera eficaz, demostrando que la implementación de las interfaces funciona.
- Utilizando Wavelet packet decomposition como vector de características se logró un mejor rendimiento en los clasificadores que utilizando el vector de características de 10 dimensiones en el dominio del tiempo, para señales EEG.
- Utilizar las señales EEG extrayendo características y utilizándolas como vector de características auxiliar para clasificar señales EMG demostró funcionar cuando se tiene una poca cantidad de muestras. Al aumentar las muestras, ese vector deja de aportar significativamente a los entrenamientos.

- Combinar clases que dependan directamente de un sensor y del otro en un solo entrenamiento, resultó ser efectivo y funciona de forma adecuada.

- Considerar el uso del lenguaje Python para el desarrollo de los algoritmos e interfaces desarrollados en este trabajo. Esto nace de todos los inconvenientes con los que se cuenta al utilizar MATLAB. Los dos dispositivos o sensores utilizados para la obtención de las señales, funcionan con Python, incluso el microcontrolador del dispositivo bitalino, está programado en Python y PHP. Mientras el dispositivo electro cap utiliza un código basado en Python para poder recibir las señales en MATLAB.
- El uso del dispositivo bitalino presentó resultados satisfactorios y permitió el cumplimiento de los objetivos. Pero se recomienda indagar en otros dispositivos que sirvan para la recolección de señales EMG, ya que el dispositivo bitalino presenta deficiencias de conexión y tiempos.
- Al contar con las interfaces para cada tipo de señal que cuentan con diferentes dispositivos de adquisición, se recomienda el desarrollo de un módulo encargado de la sujeción de los sensores de una manera en la que el usuario no afecte las mediciones con algún movimiento no deseado.
- Probar todas las interfaces clasificadoras con otros sistemas robóticos que no sean el R17, para dar pauta al inicio del desarrollo de sistemas robóticos que sirvan para la asistencia médica.
- Se recomienda el uso de características en el dominio de la frecuencia para los entrenamientos con señales EEG, para poder observar su rendimiento.
- Se recomienda en siguientes trabajos, el desarrollo de entrenamientos que incluyan posicionamiento distinto de electrodos, gestos diferentes y el uso de más de dos clases en un solo entrenamiento.

-
- [1] P. Angkoon, P. Pornchai y L. Chusak, “Feature reduction and selection for EMG signal classification,” Artículo Científico, Prince of Songkla University, 2012.
 - [2] R. Farhan, H. Ali, R. Saad, N. Imran Khan y D. Kim, “EMD based Temporal and Spectral Features for the Classification of EEG Signals Using Supervised Learning,” Artículo Científico, National University of Sciences y Technology, Islamabad, 2016.
 - [3] H. Neha, D. Ratan y K. Neelesh, “Fusion of EEG and EMG signals for classification of unilateral foot movements,” Artículo Científico, Biomedical Instrumentation Unit, CSIR-Central Scientific Instruments Organisation, Chandigarh, 160030, India, 2020.
 - [4] R. Ralda, “Interfaz Biomédica para el Control de Sistemas Robóticos Utilizando Señales EEG,” Tesis de licenciatura, Universidad del Valle de Guatemala, 2020.
 - [5] M. F. Girón, “Interfaz Biomédica para el Control de Sistemas Robóticos Utilizando Señales EMG,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2020.
 - [6] J. D. Avila y F. Fajardo, “Análisis Entre señales electromiográficas en los musculos cervicales y el ángulo de inclinación de la cabeza en niños con PCI entre 5 y 10 años aplicada a terapia CERVICAL,” Tesis de licenciatura, Universidad Politécnica Salesiana Sede Cuenca, 2019.
 - [7] J. Basmajian, *Muscles Alive THEIR FUNCTIONS REVEALED BY ELECTROMYOGRAPHY*. Baltimore: The williams & Wilkins Company, 1967.
 - [8] D. Stegeman y H. Hermens, “Standards for surface electromyography: The European project Surface EMG for non-invasive assessment of muscles (SENIAM),” vol. 1, ene. de 2007.
 - [9] S. Sanei y J. A. Chambers, *EEG signal processing*. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England: John Wiley & Sons, Ltd, 2007.
 - [10] J. Mosher, P. Lewis y R. Leahy, *Multiple dipole modeling and localization from spatio-temporal MEG data*. California: IEEE Transactions on Biomedical Engineering, 1992.
 - [11] P. Abhang, B. Gawali y S. Mehrotra, *Introduction to EEG- and Speech-Based Emotion Recognition*. India: Ambedkar Marathwada University, Aurangabad, India, 2009.

- [12] C. Novo-Olivas, L. Guitiérrez y J. Bribiesca, “Mapeo Electroencefalográfico y Neurofeedback,” en. feb. de 2010, págs. 371-412, ISBN: 978-970-764-911-8.
- [13] J. Adamssen, *Inteligencia artificial cómo el aprendizaje automático, la robótica y la automatización han dado forma a nuestra sociedad*. Brasil: Efalon Acies, 2020.
- [14] V. Bennett, K. Bowman y S. Wright, “Stages for Developing Control Systems using EMG and EEG Signals: A survey,” School of Computer Science y Electronic Engineering University of Essex, United Kingdom, inf. téc. CES-513 ISSN 1744-8050, jun. de 2011.
- [15] R. López, J. Fernandez y J. Fernández, *Las Redes Neuronales Artificiales*, ép. Metodología y Análisis de Datos en Ciencias Sociales. Netbiblo, 2008, ISBN: 9788497452465. dirección: <https://books.google.com.gt/books?id=XOuLwi1Ap4QC>.
- [16] R. González, A. Barrientos, M. Toapanta y J. Del Cerro, “Aplicación de las Máquinas de Soporte Vectorial (SVM) al diagnóstico clínico de la Enfermedad de Párkinson y el Temblor Esencial,” *Revista Iberoamericana de Automática e Informática industrial*, vol. 50, n.º 14, págs. 394-405, 2017.
- [17] K. A. Strausser y H. Kazerooni, “The development and testing of a human machine interface for a mobile medical exoskeleton,” en *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, págs. 4911-4916. DOI: [10.1109/IROS.2011.6095025](https://doi.org/10.1109/IROS.2011.6095025).
- [18] J. Gutiérrez, J. Cantillo, R. Cariño y E. Viñas, “Los sistemas de interfaz cerebro-computadora: una herramienta para apoyar la rehabilitación de pacientes con discapacidad motora,” *medigraphic*, n.º 2, págs. 62-69, 2013.
- [19] S. Robotics, *R17 DEUCALEON ROBOT SYSTEM User Manual*, 1.ª ed., Mail: support@strobotics.com, ST Robotics, The address of the publisher, abr. de 2017.
- [20] J.-H. Jeong, J.-H. Cho, K.-H. Shim, B.-H. Kwon, B.-H. Lee, D.-Y. Lee, D.-H. Lee y S.-W. Lee, “Multimodal signal dataset for 11 intuitive movement tasks from single upper extremity during multiple recording sessions,” *GigaScience*, vol. 9, n.º 10, oct. de 2020, g1aa098, ISSN: 2047-217X. DOI: [10.1093/gigascience/g1aa098](https://doi.org/10.1093/gigascience/g1aa098), eprint: https://academic.oup.com/gigascience/article-pdf/9/10/g1aa098/33842213/g1aa098_reviewer_2_report_original_submission.pdf, dirección: <https://doi.org/10.1093/gigascience/g1aa098>.
- [21] A. Buerkle, W. Eaton, N. Lohse, T. Bamber y P. Ferreira, “EEG based arm movement intention recognition towards enhanced safety in symbiotic Human-Robot Collaboration,” *Robotics and Computer-Integrated Manufacturing*, vol. 70, pág. 102137, 2021, ISSN: 0736-5845. DOI: <https://doi.org/10.1016/j.rcim.2021.102137>, dirección: <https://www.sciencedirect.com/science/article/pii/S0736584521000223>.
- [22] A. Kulkarni, D. Chong y F. A. Batarseh, “5 - Foundations of data imbalance and solutions for a data democracy,” en *Data Democracy*, F. A. Batarseh y R. Yang, eds., Academic Press, 2020, págs. 83-106, ISBN: 978-0-12-818366-3. DOI: <https://doi.org/10.1016/B978-0-12-818366-3.00005-8>, dirección: <https://www.sciencedirect.com/science/article/pii/B9780128183663000058>.
- [23] BITalino, *BITalino (r)evolution Board Kit Data Sheet*, Wireless Biosignals, S.A, 1050-059 Lisbon, Portugal, 2020.

- [24] M. I. C. T. Team. (2017). “BITalino Toolbox,” dirección: <https://www.mathworks.com/matlabcentral/fileexchange/53983-bitalino-toolbox> (visitado 12-10-2021).
- [25] A. Subasi, *Practical guide for biomedical signals analysis using machine learning techniques : a MATLAB based approach / Abdulhamit Subasi*. eng, First edition. London, United Kingdom: Academic Press, an imprint of Elsevier, 2019, ISBN: 0-12-817673-3.
- [26] A. Merlo, D. Farina y R. Merletti, “A fast and reliable technique for muscle activity detection from surface EMG signals,” *IEEE transactions on bio-medical engineering*, vol. 50, págs. 316-23, abr. de 2003. DOI: [10.1109/TBME.2003.808829](https://doi.org/10.1109/TBME.2003.808829).
- [27] J. Subramanian y R. Simon, “Overfitting in prediction models – Is it a problem only in high dimensions?” *Contemporary Clinical Trials*, vol. 36, n.º 2, págs. 636-641, 2013, ISSN: 1551-7144. DOI: <https://doi.org/10.1016/j.cct.2013.06.011>, dirección: <https://www.sciencedirect.com/science/article/pii/S1551714413001031>.
- [28] O. BCI. (2021). “Cyton Biosensing Board (8-channels),” dirección: <https://shop.openbci.com/products/cyton-biosensing-board-8-channel?variant=38958638542> (visitado 21-11-2021).
- [29] I. Electro-Cap International. (2017). “systems,” dirección: <https://electro-cap.com/index.cfm/systems/> (visitado 21-11-2021).
- [30] C. Pulliam, J. Lambrecht y R. Kirsch, “Electromyogram-based neural network control of transhumeral prostheses,” *Journal of rehabilitation research and development*, vol. 48, págs. 739-54, jul. de 2011. DOI: [10.1682/JRRD.2010.12.0237](https://doi.org/10.1682/JRRD.2010.12.0237).
- [31] D. L. Banks y S. E. Fienberg, “Data Mining, Statistics,” en *Encyclopedia of Physical Science and Technology (Third Edition)*, R. A. Meyers, ed., Third Edition, New York: Academic Press, 2003, págs. 247-261, ISBN: 978-0-12-227410-7. DOI: <https://doi.org/10.1016/B0-12-227410-5/00164-2>, dirección: <https://www.sciencedirect.com/science/article/pii/B0122274105001642>.
- [32] S. Goswami y A. Chakrabarti, “Feature selection: A practitioner view,” *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 6, n.º 11, pág. 66, 2014.
- [33] Fernandez, C. Ruben, O. Julián y Manuel, *Aprendizaje Estadístico*. Vigo: Universidad de Santiago de Compostela, 2021.

17.1. Paradigma experimental señales EMG

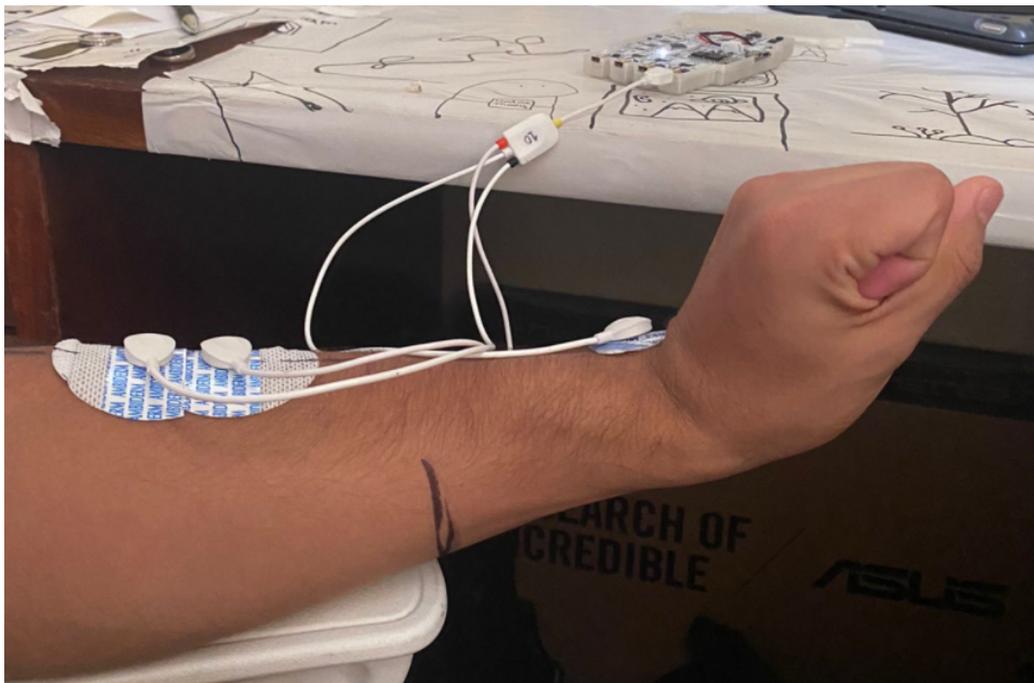


Figura 42: Grabación clase puño arriba

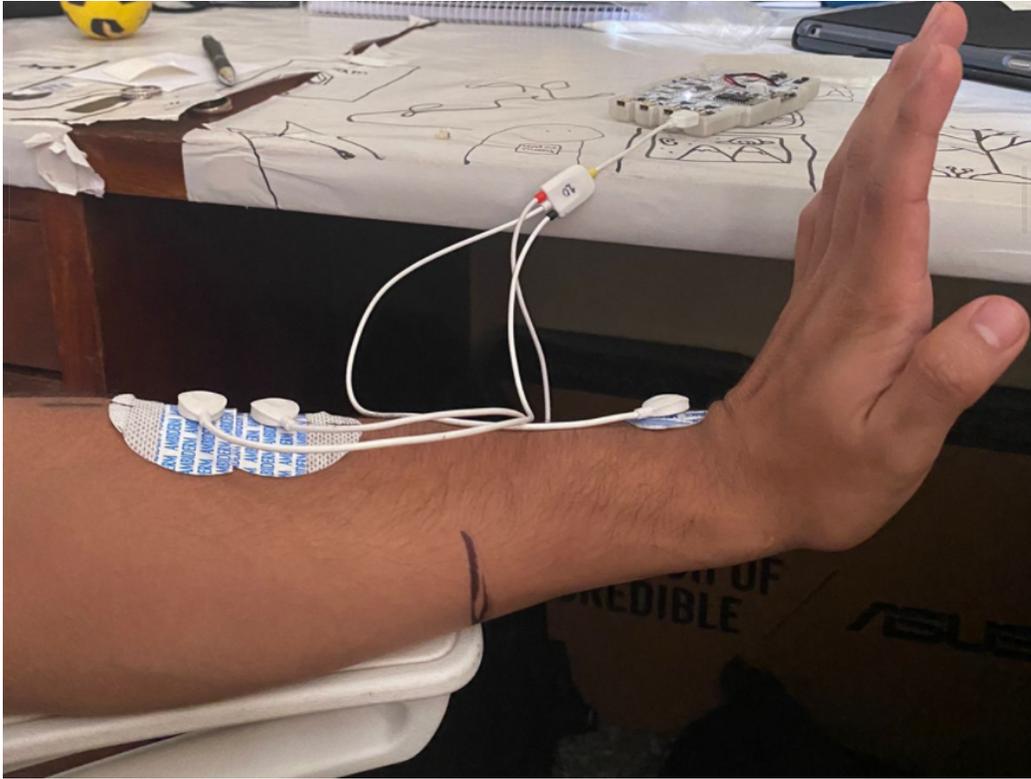


Figura 43: Grabación clase puño arriba.

17.2. Paradigma experimental señales EEG-EMG



Figura 44: Grabación clase palma arriba señales EEG-EMG.

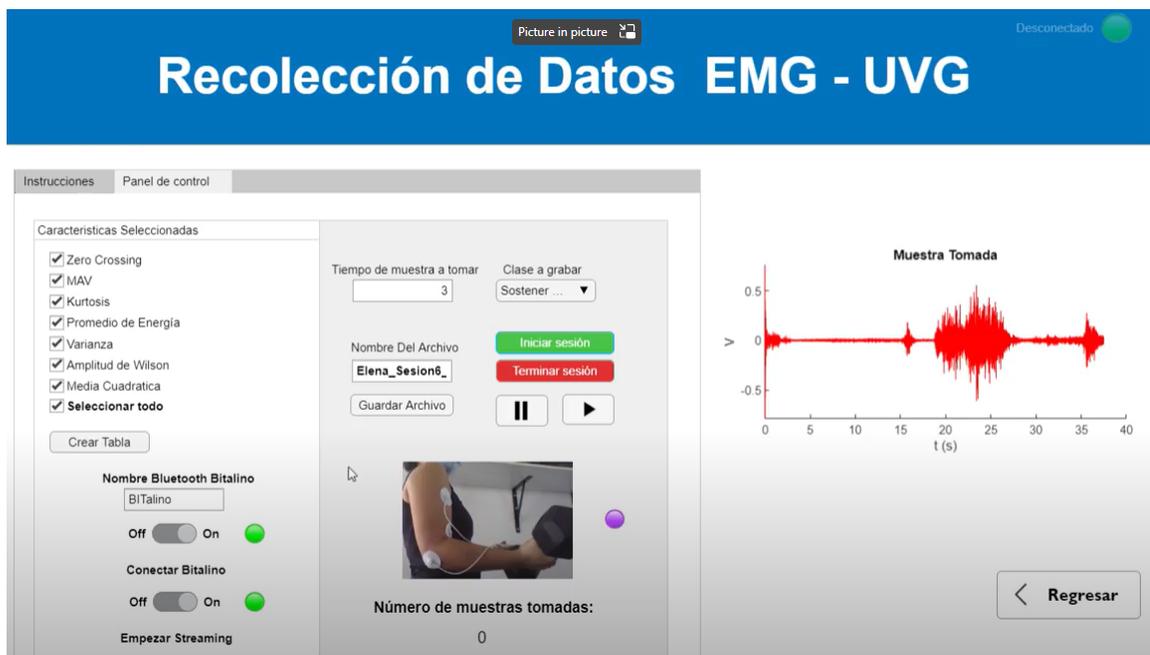


Figura 45: Proyecto fisiología 2 utilizando interfaz recolectora de señales EMG (Grupo 3).



Figura 46: Primera prueba interfaz clasificadora señales EEG-EMG.

BCI EMG - UVG

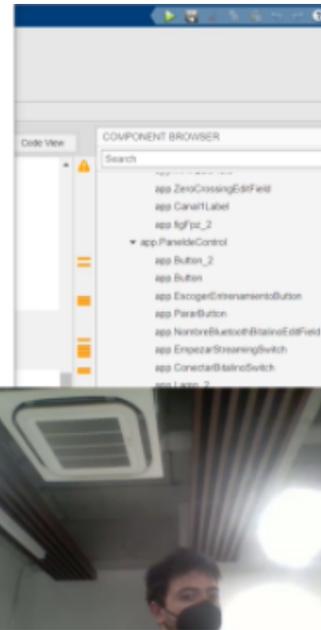


Figura 47: Proyecto fisiología 2 utilizando interfaz clasificadora de señales EMG (Grupo 1).

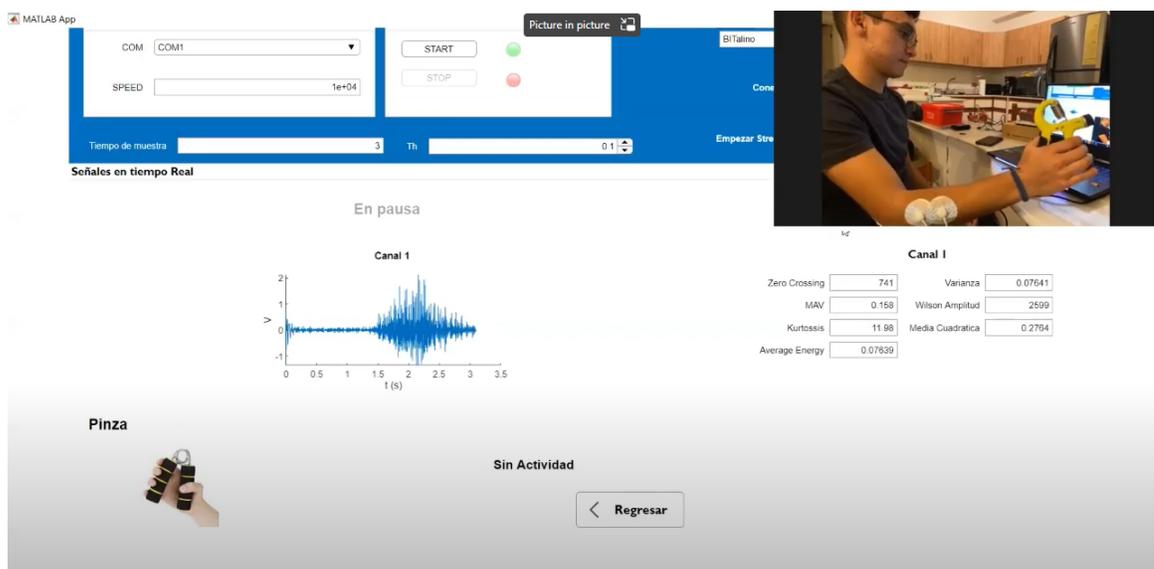


Figura 48: Proyecto fisiología 2 utilizando interfaz clasificadora de señales EMG (Grupo 2).



Figura 49: Posicionamiento de electrodos y toma de datos (Grupo2).



Figura 50: Posicionamiento de electrodos y toma de datos (Grupo2).

Selección de características

Roberto Cáceres.

Clasificación de características univariante para la clasificación mediante pruebas de chi-cuadrado

Este metodo lo podemos utilizar para observar cual es el peso que tienen las características para las señales obtenidas. con la función fscchi2, podemos obtener los caracteristicas y el peso que tiene cada una de las variables en la clase en este caso. Este es un análisis hecho para una sola clase y sus características y lo podemos utilizar para apoyar la selección de nuestras características.

```
%load ("Data.mat")
[idx,scores] = fscchi2(Primeras3Sesiones,"Clase")
find(isinf(scores))
bar(scores(idx))
xlabel('Predictor rank')
ylabel('Predictor importance score')
xticklabels(strrep(Primeras3Sesiones.Properties.VariableNames(idx),'_','\'))
xtickangle(45)
idx(1:10)
```

En este ejemplo podemos observar que tomamos de la tabla **NombreDeLaTabla** la etiqueta que es la respuesta a esta clasificación "**Clase**". Tomamos los pesos que tienen las variables en la clasificación de la "**Clase**". Para más información sobre esta función se puede consultar el siguiente enlace.

[Univariate feature ranking for classification using chi-square tests - MATLAB fscchi2 \(mathworks.com\)](#)

Feature selection using neighborhood component analysis for classification

Definimos nuestra data de entrenamiento y nuestra data para la validación

```
Xtrain = XtrainYtrain(1:540,1:17);
ytrain = XtrainYtrain(1:540,18);
```

```
idx = 1x17
    13    10     2     7     4     6     3     8     5    12    14    15    17 ...
scores = 1x17
    17.3593    120.8838    62.2220    63.5434    40.3167    62.8581    79.9714    60.9018 ...
ans =
    1x0 empty double row vector
```

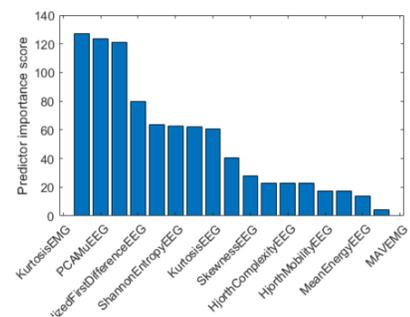


Figura 51: Live script realizado en el software MATLAB para ayudar a la selección de características.

Manual de usuario Interfaces EMG

Roberto Cáceres, Estudiante
Contacto: +502 5411-6850
Correo: cac17163@uvg.edu.gt

April 3, 2022

Contents

1	Introducción	1
1.1	Sobre las interfaces	1
1.2	Sobre Dispositivo BITalino	2
2	Dispositivo Bitalino	2
2.1	BITalino Toolbox	2
2.2	Conexión con Bitalino	6
2.3	Ejemplos	8
2.4	Consideraciones	9
3	Interfaces EMG	10
3.1	Interfaz recolectora de datos	10
3.2	Interfaz clasificadora de datos	14

1 Introducción

Se ha creado este documento para realizar un resumen para estudiantes e investigadores que quieran hacer uso de las interfaces de recolección y clasificación de señales electromiográficas (EMG). En este manual de usuario se explicará al interesado los requerimientos e implicaciones que conlleva el uso de la interfaz. Desde el uso del dispositivo Bitalino hasta los recursos necesarios para utilizar este con el software MATLAB. Este Manual de usuario no solo pretende guiar al interesado en los recursos creados en este trabajo de graduación, también pretende guiar al interesado en las experiencias e inconvenientes encontrados con cada uno de las etapas de este.

1.1 Sobre las interfaces

Se cuenta con dos interfaces, una de las interfaces cuenta con cumplir con la necesidad de recolectar señales EMG, que pueden ser utilizadas para investigaciones posteriores o como base de datos basada en diferentes experimentos. La segunda interfaz es la interfaz clasificadora de gestos, donde con entrenamientos previos y utilizando inteligencia artificial (IA) se mandan comandos al robot R17 de la Universidad del Valle de Guatemala.

Figura 52: Manual para el uso del dispositivo bitalino y uso de las interfaces para señales EMG.

Manual de usuario Interfaces EEG

Roberto Cáceres, Estudiante
Contacto: +502 5411-6850
Correo: cac17163@uvg.edu.gt

April 3, 2022

Contents

1	Introducción	1
1.1	Sobre las interfaces	1
1.2	Sobre electrocap	2
2	Dispositivo Electrocap	2
2.1	Brain flow Library	2
2.2	Conexión con electrocap	7
2.3	Ejemplos	9
2.4	Consideraciones	10
3	Interfaces EEG	11
3.1	Interfaz recolectora de datos	11
3.2	Interfaz clasificadora de datos	19

1 Introducción

Se ha creado este documento para realizar un resumen para estudiantes e investigadores que quieran hacer uso de las interfaces de recolección y clasificación de señales electroencefalográficas (EEG). En este manual de usuario se explicará al interesado los requerimientos e implicaciones que conlleva el uso de la interfaz. Desde el uso del dispositivo electrocap hasta los recursos necesarios para utilizar este con el software MATLAB. Este Manual de usuario no solo pretende guiar al interesado en los recursos creados en este trabajo de graduación, también pretende guiar al interesado en las experiencias e inconvenientes encontrados con cada uno de las etapas de este.

1.1 Sobre las interfaces

Se cuenta con dos interfaces, una de las interfaces cuenta con cumplir con la necesidad de recolectar señales EEG, que pueden ser utilizadas para investigaciones posteriores o como base de datos basada en diferentes experimentos. La segunda interfaz es la interfaz clasificadora de gestos, donde con entrenamientos previos y utilizando inteligencia artificial (IA) se mandan comandos al robot R17 de la Universidad del Valle de Guatemala.

Figura 53: Manual para el uso del dispositivo electrocap y uso de las interfaces para señales EEG.

