
Diseño e implementación de interfaz biomédica
y sensores iniciales para el control de actuadores
en tiempo real.

Santiago José Rivera Lemus



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Diseño e implementación de interfaz biomédica y sensores inerciales para el control de actuadores en tiempo real.

Trabajo de graduación presentado por Santiago José Rivera Lemus para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2025

Vo.Bo.:

(f) _____
Ing. Luis Rivera

Tribunal Examinador:

(f) _____
Ing. Luis Rivera

(f) _____
-

(f) _____
-

Fecha de aprobación: Guatemala, -de -de -.

Prefacio

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras vitae eleifend ipsum, ut mattis nunc. Pellentesque ac hendrerit lacus. Cras sollicitudin eget sem nec luctus. Vivamus aliquet lorem id elit venenatis pellentesque. Nam id orci iaculis, rutrum ipsum vel, porttitor magna. Etiam molestie vel elit sed suscipit. Proin dui risus, scelerisque porttitor cursus ac, tempor eget turpis. Aliquam ultricies congue ligula ac ornare. Duis id purus eu ex pharetra feugiat. Vivamus ac orci arcu. Nulla id diam quis erat rhoncus hendrerit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed vulputate, metus vel efficitur fringilla, orci ex ultricies augue, sit amet rhoncus ex purus ut massa. Nam pharetra ipsum consequat est blandit, sed commodo nunc scelerisque. Maecenas ut suscipit libero. Sed vel euismod tellus.

Proin elit tellus, finibus et metus et, vestibulum ullamcorper est. Nulla viverra nisl id libero sodales, a porttitor est congue. Maecenas semper, felis ut rhoncus cursus, leo magna convallis ligula, at vehicula neque quam at ipsum. Integer commodo mattis eros sit amet tristique. Cras eu maximus arcu. Morbi condimentum dignissim enim non hendrerit. Sed molestie erat sit amet porttitor sagittis. Maecenas porttitor tincidunt erat, ac lacinia lacus sodales faucibus. Integer nec laoreet massa. Proin a arcu lorem. Donec at tincidunt arcu, et sodales neque. Morbi rhoncus, ligula porta lobortis faucibus, magna diam aliquet felis, nec ultrices metus turpis et libero. Integer efficitur erat dolor, quis iaculis metus dignissim eu.

Índice

Prefacio	III
Lista de figuras	VIII
Lista de cuadros	IX
Resumen	X
Abstract	XI
1. Introducción	1
2. Antecedentes	2
2.1. Interfaces Biomédicas	2
2.2. Interfaces Biomédicas Desarrolladas en la UVG	3
2.3. Control de actuadores por medio de señales bioeléctricas	4
2.4. Mano Animatrónica UVG	5
3. Justificación	8
4. Objetivos	10
4.1. Objetivo General	10
4.2. Objetivos Específicos	10
5. Alcance	11
6. Marco teórico	12
6.1. Señales Bioeléctricas	12
6.2. Señales EMG	12
6.3. Características en el tiempo	13
6.3.1. Zero crossing	13
6.3.2. Media del Valor Absoluto	13
6.3.3. Media cuadrática	13
6.3.4. Varianza	13

6.3.5. Amplitud de Willison	14
6.3.6. Curtosis	14
6.4. BITalino	14
6.5. Myoware	14
6.6. Aprendizaje automático	15
6.6.1. Aprendizaje no supervisado	16
6.6.2. Aprendizaje supervisado	16
6.7. Interfaz HMI (Human-Machine Interface)	20
6.8. Sensores Inerciales	20
6.9. Actuadores	20
6.10. Prótesis	21
7. Recolección de señales EMG	22
7.1. Dispositivos de adquisición para señales EMG	22
7.1.1. Dispositivo BITalino	22
7.1.2. Dispositivo MyoWare	23
7.2. Algoritmos de adquisición	23
7.3. Algoritmo de captura en tiempo real	24
7.4. Procesamiento de señal	26
7.5. Interfaz de colección de señales	26
8. Clasificadores de señales EMG	29
8.1. Gestos Utilizados	29
8.1.1. Colección de 3 gestos	29
8.1.2. Colección de 4 gestos	32
8.2. <i>Classification Learner</i>	32
8.3. SVM	33
8.3.1. SVM Linear	33
8.3.2. SVM Cuadrática	34
8.3.3. SVM Cúbica	35
8.4. ANN	36
8.4.1. ANN de tres capas	37
8.4.2. ANN de dos capas	38
8.4.3. ANN de arquitectura ancha	39
8.5. KNN	40
8.5.1. KNN Pesada	40
8.5.2. KNN Fina	41
8.5.3. KNN Cúbica	42
9. Sistema de sensores inerciales	44
10. Mano antropomórfica UVG	45
10.1. Acercamiento con mano antropomórfica	45
10.2. Algoritmos de control	45
10.3. Movimientos Predeterminados	48

11. Interfaz de control de mano antropomórfica	55
11.1. Implementación de clasificadores	55
11.2. Funcionalidades	56
11.2.1. Modo Binario	57
11.2.2. Modo 1 sensor	57
11.2.3. Modo 2 sensores	57
12. Conclusiones	58
13. Recomendaciones	59
14. Bibliografía	60
15. Anexos	63

Lista de figuras

1.	Primera fase de la interfaz UVG [5].	3
2.	Segunda fase de interfaz [6].	4
3.	Placa Muscle Sensor v3 [9].	5
4.	Módulo Myoware [10].	6
5.	Mano animatrónica antropomórfica [12].	7
6.	Ejemplo de señal EMG [16].	13
7.	Kit de desarrollo BITalino [18].	15
8.	Muscle Sensor MyoWare 2.0 [19].	15
9.	Representación de SVM [24].	16
10.	Representación de funciones de Kernel y representación de hiperplanos [24]. .	17
11.	Representación SVM <i>One vs All</i> [27].	18
12.	Representación SVM <i>One vs One</i> [27].	18
13.	Representación del algoritmo KNN [30].	19
14.	Representación de capas de una Red neuronal artificial simple [23].	20
15.	Prótesis realizada en el antiguo Egipto [34].	21
16.	Esquema de circuito utilizado en el sensor MyoWare [19].	23
17.	Diagrama de flujo de algoritmo de adquisición en tiempo real.	25
18.	Interfaz de recolección de señales EMG de fases anteriores.	26
19.	Segunda versión de interfaz de recolección de señales EMG.	27
20.	Diagrama de flujo de algoritmo de interfaz de recolección EMG.	28
21.	Posición de los electrodos para la captura de los 3 gestos.	30
22.	Gesto 1: Movimiento de puño hacia arriba.	31
23.	Gesto 3: Brazo en reposo.	32
24.	Matriz de confusión con etapa de validación para SVM linear.	33
25.	Matriz de confusión con etapa de prueba para SVM linear con tres gestos. .	34
26.	Matriz de confusión con etapa de validación para SVM cuadrática con tres gestos.	34
27.	Matriz de confusión con etapa de prueba para SVM cuadrática con tres gestos. .	35
28.	Matriz de confusión con etapa de validación para SVM cúbica con tres gestos. .	35
29.	Matriz de confusión con etapa de prueba para SVM cúbica con tres gestos. .	36

30.	Matriz de confusión con etapa de validación para ANN de tres capas con tres gestos.	37
31.	Matriz de confusión con etapa de prueba para ANN de tres capas con tres gestos.	37
32.	Matriz de confusión con etapa de validación para ANN de dos capas con tres gestos.	38
33.	Matriz de confusión con etapa de prueba para ANN de dos capas con tres gestos.	38
34.	Matriz de confusión con etapa de validación para ANN ancha con tres gestos.	39
35.	Matriz de confusión con etapa de prueba para ANN ancha con tres gestos.	39
36.	Matriz de confusión con etapa de validación para KNN pesada con tres gestos.	40
37.	Matriz de confusión con etapa de prueba para KNN pesada con tres gestos.	41
38.	Matriz de confusión con etapa de validación para KNN fina con tres gestos.	41
39.	Matriz de confusión con etapa de prueba para KNN fina con tres gestos.	42
40.	Matriz de confusión con etapa de validación para KNN cúbica con tres gestos.	42
41.	Matriz de confusión con etapa de prueba para KNN cúbica con tres gestos.	43
42.	Diagrama de flujo de algoritmo de control de la mano antropomórfica.	47
43.	Conteo del número 0 en mano antropomórfica.	49
44.	Conteo del número 1 en mano antropomórfica.	50
45.	Conteo del número 2 en mano antropomórfica.	51
46.	Conteo del número 3 en mano antropomórfica.	52
47.	Conteo del número 4 en mano antropomórfica.	53
48.	Conteo del número 5 en mano antropomórfica.	54
49.	Interfaz de control de mano antropomórfica UVG, pestaña de entrenamiento.	56
50.	Interfaz de control de mano antropomórfica UVG, pestaña de control.	56

Lista de cuadros

1.	Resultados de las SVM con 3 gestos	33
2.	Resultados de ANN con 3 gestos	36
3.	Resultados de KNN con 3 gestos	40

Resumen

Abstract

This is an abstract of the study developed under the

CAPÍTULO 1

Introducción

En los últimos años, la investigación y desarrollo de prótesis se han centrado en el estudio e implementación de señales bioeléctricas para su control. Las señales electromiográficas (EMG) y su integración con algoritmos de aprendizaje automático han demostrado ser beneficiosas para mejorar la calidad de vida de personas afectadas que requieren este tipo de tecnología.

Este trabajo propone el desarrollo de una interfaz capaz de interpretar señales EMG mediante clasificadores de aprendizaje automático, así como la implementación de sensores iniciales para controlar actuadores, enfocándose en el control de prótesis.

El desarrollo de la interfaz se llevó a cabo en varias etapas. Se realizaron pruebas con diferentes sensores para optimizar la extracción de señales EMG y se implementaron algoritmos de clasificación con el fin de encontrar el clasificador más adecuado para las aplicaciones deseadas. Además, se utilizaron fundamentos de fases previas del proyecto para obtener resultados más efectivos y prácticos.

El objetivo principal fue mejorar las fases anteriores, implementando un control en tiempo real y añadiendo sensores iniciales, tanto para la clasificación como para el control de movimientos. El documento detalla la metodología empleada para la clasificación y control de una mano antropomórfica, así como la validación de los resultados a través del control de dicha prótesis.

CAPÍTULO 2

Antecedentes

2.1. Interfaces Biomédicas

En tiempos recientes, el estudio de las señales del cuerpo ha sido un campo explorado para su uso dentro de diferentes aplicaciones para ayudar y mejorar la vida de las personas que lo lleguen a necesitar. Ya sea dentro del campo médico, con la creación de prótesis o en el campo de la robótica, para lograr el control de diferentes actuadores y robots en general, para tener un control sin necesidad de un mando físico, más bien un control con las señales del cuerpo.

En la Universidad del Norte se desarrolló un sistema con la capacidad de adquirir y visualizar señales EMG [1]. Por medio del uso de electrodos y ciertos microcontroladores se logró la captación de las señales, por medio de un procedimiento específico de adquisición utilizando circuitos de amplificación y diferentes filtros para lograr tener una señal limpia. Además, el uso de una conversión A/D y el uso de USART fueron implementados para una comunicación estable con un ordenador. La interfaz utilizó LabVIEW para lograr mostrar visualmente las señales EMG producidas por una persona.

En la Universidad de Cauca se realizó el desarrollo de una interfaz gráfica Hombre – Máquina de recolección y diferenciación de señales EMG por medio de tres diodos y diferentes sensores para lograr, analizar y procesar las señales [2]. En el desarrollo se utilizó Aprendizaje Automático para lograr diferenciar las señales por el uso de características. Por ejemplo, en este proyecto se utilizaron las características de media, varianza, histograma de amplitudes y el valor RMS. Los algoritmos de control y la creación del sistema se hicieron utilizando LabVIEW 6i de National Instruments, donde se logró el control de diferentes aplicaciones como lo son un brazo robótico y su simulación, además del control de un robot móvil.

En 2019 en la Universidad Nacional de La Plata se creó una interfaz hombre – máquina con recolecciones en tiempo real de las señales y portátil [3]. Para conseguir una buena adquisición de señales, el proyecto seleccionó el microcontrolador MSP430G2553 de Texas

Instruments y un circuito integrado ADS1299 para la etapa de alimentación del sistema. Los resultados obtenidos fueron satisfactorios y se logró la creación de un dispositivo portátil con vida útil de aproximadamente 38 horas.

2.2. Interfaces Biomédicas Desarrolladas en la UVG

En la Universidad del Valle de Guatemala se han estado desarrollando herramientas en los últimos años para lograr una mejor obtención y análisis de señales bioeléctricas. Empezando en el año 2020, con el trabajo de María Fernanda Girón [4], se implementaron diferentes algoritmos de reconocimiento de señales electromiográficas (EMG) para su implementación en el control de sistemas robóticos. Realizado por medio de algoritmos de Aprendizaje Automático con Maquinas de vectores de soporte (SVM) y Redes neuronales (RNAs), obteniendo rendimientos de clasificación superiores al 80 %.

En ese mismo año Rodrigo Ralda [5], desarrolló una primera iteración de una interfaz con la habilidad de manipular sistemas robóticos con el uso de señales electroencefalográficas obteniendo resultados similares al trabajo de Girón, con porcentajes superiores a un 80 %.

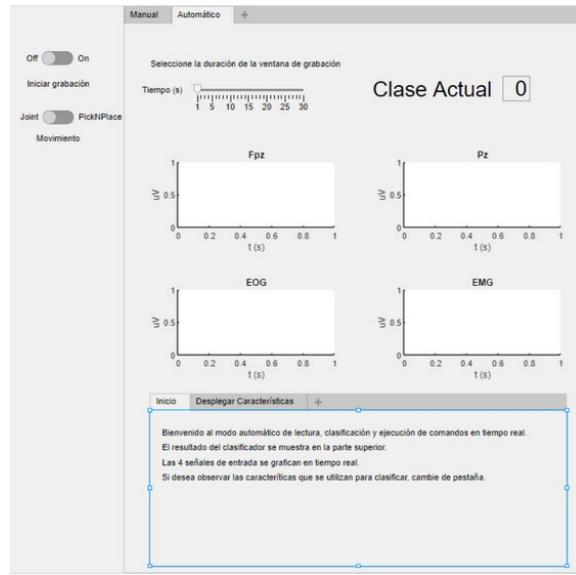


Figura 1: Primera fase de la interfaz UVG [5].

En el año 2021, se continuó la investigación y mejora de los trabajos previos dentro de la Universidad del Valle de Guatemala con el trabajo de Roberto Cáceres [6]. En ese trabajo se implementó una unificación con los trabajos previos creando una sola interfaz con el mismo objetivo de control de sistemas robóticos con la implementación de algoritmos de aprendizaje automático.

En la investigación se obtuvieron rendimientos superiores a trabajos anteriores dentro de la universidad utilizando los dos tipos de señales bioeléctricas siendo las señales EEG como señales auxiliares. El rendimiento varió obteniendo un porcentaje de 85 % solo con características de la señal EMG, incrementando a un 87 % con el uso de las señales EEG como auxiliares.

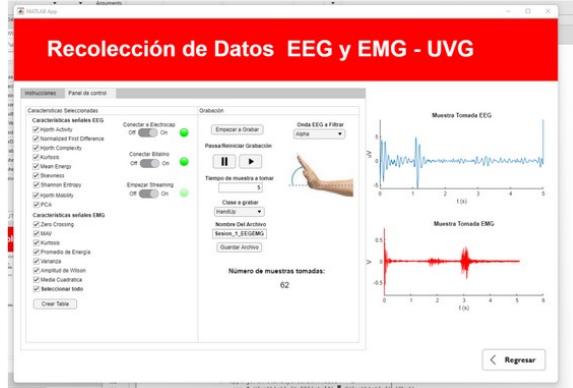


Figura 2: Segunda fase de interfaz [6].

La investigación dentro de la universidad continuó en el año 2023 con el trabajo de Fernando Sandoval [7] solventando una mejora en la disminución del tiempo computacional de los algoritmos y de extracción de señales con un cambio en el dispositivo optando por el sistema BIOPAC y con la base de datos del centro HUMANA. La mejora de los algoritmos llegó a un porcentaje de clasificación positivo mostrando un rendimiento superior a un 90 %.

Dentro de la simulación dentro de la interfaz a pesar de lograr obtener resultados positivos se mostraron porcentajes de clasificaciones erróneas, gracias a una variación en la clasificación de señales de un mismo sujeto las cuales fueron tomadas en diferentes días, esto siendo más pronunciado en las señales EEG.

En los proyectos observados en la UVG podemos analizar limitaciones que existieron con los proyectos anteriores. Estas entran a la falta de un control en tiempo real y rendimientos que pueden llegar a incrementarse para la clasificación de señales.

Las limitaciones mencionadas abren las puertas a una mejora en el reconocimiento de señales EMG y EEG para el uso de la interfaz biomédica UVG, además de una implementación física. Abriendo el campo de una implementación en tiempo real de los algoritmos para una aplicación específica dentro del campo medico con el uso de actuadores para la recreación de movimientos naturales del cuerpo.

2.3. Control de actuadores por medio de señales bioeléctricas

La investigación de la creación y programación de una prótesis o bien actuadores por medio de señales biomédicas ha sido importante durante las últimas épocas donde se han hecho varios avances con equipo capaz no solo de filtrar y obtener las señales si no que también con la capacidad de analizar y procesar las mismas.

Para poder encontrar y reconocer las diferentes señales obtenidas en el cuerpo y poder interpretarlas para un control más sistemático ha sido tema de discusión en tiempos recientes. El análisis de las mismas y la diferenciación de las mismas cae en las características de las diferentes señales.

Es necesario el uso de aprendizaje automático y el uso de características de las diferentes señales para lograr una clasificación. Ejemplos de estas pueden ser el valor medio absoluto, la longitud de forma de onda, cruces por cero, cambio de signo de pendiente, entre otros [8].

En el año 2015 en la Universidad de Zaragoza [9], se inició entorno experimental para pruebas de control con señales bioeléctricas, donde se logró una captación y lecturas de las señales EMG para crear movimientos de un brazo robótico por medio de la placa *Muscle Sensor v3* con la cual se lograron obtener movimientos simples relacionando señales tanto EMG como EEG y sus características.

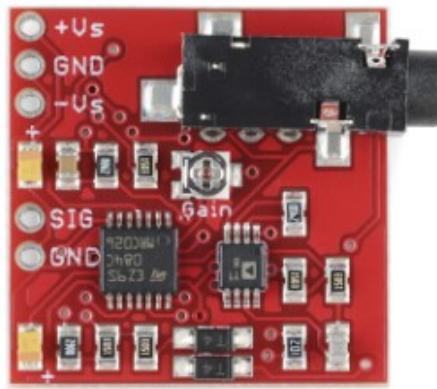


Figura 3: Placa Muscle Sensor v3 [9].

En los proyectos observados, las limitaciones que brindan recaen mucho en el análisis de las señales y el rendimiento de clasificación por medio del Aprendizaje Automático. Los sensores utilizados no brindaron una señal tan limpia y manipulable. La variedad de movimientos se ve limitada y trae consigo cierto porcentaje de error con el reconocimiento de las señales.

Las limitaciones mencionadas brindan campo para mejora de control de los diferentes dispositivos, tanto con una recolección de datos más exacta como una mejora en el procesamiento y análisis de las señales. El control de actuadores por medio de señales bioeléctricas brinda la necesidad de optimizar y mejorar la forma de obtener las señales, los algoritmos de aprendizaje automatizado se muestran esenciales para el control exacto y buscado para aplicaciones específicas.

2.4. Mano Animatrónica UVG

En la Universidad del Valle de Guatemala se han desarrollado diversas prótesis con el objetivo de replicar los movimientos naturales de la mano humana. Estas iniciativas buscan fomentar el avance tecnológico, proporcionando beneficios significativos no solo para Guatemala, sino también para todas las personas que necesitan estas prótesis en su vida diaria.

En el año 2020, se realizó un diseño y una prótesis de miembro superior en la Univer-

sidad del Valle de Guatemala hecho por Gonzalo Palarea [10], quien diseñó una prótesis electromecánica de bajo costo por medio de materiales impresos en 3D con la idea de crear una alternativa a los productos y precios de esos tiempos. Palarea logró crear un diseño bajo en costo con una programación basada en el sensor *Myoware* con el cual fue posible realizar movimientos simples reflejando movimientos naturales de la mano.



Figura 4: Módulo Myoware [10].

En el año 2021 se empezó el proyecto de la optimización del diseño y control de una mano animatrónica antropomórfica, realizado por Omar Gálvez. Quien por el uso de motores "Dynamixel" tanto AX-12A como XL-320 logró recrear los movimientos establecidos y suaves aprovechando los motores al máximo. Dentro de la programación utilizada Gálvez utilizó tanto MATLAB como el microcontrolador OpenCM9.04 C para el control de los diferentes actuadores utilizados. Además de utilizar un sensor de captura "LeapMotion" para la obtención de los diferentes movimientos deseados a utilizar en la prótesis [11].

En el año 2022 se continuó con el trabajo del proyecto gracias a Miguel García, quien optimizó aún más el trabajo de Galvez. García trabajó en la implementación del control y visualización de movimientos con el programa Unity para el procesamiento de los diferentes movimientos captados por el proyecto [12].



Figura 5: Mano animatrónica antropomórfica [12].

CAPÍTULO 3

Justificación

Hoy en día, la tecnología es parte necesaria dentro de nuestras vidas habituales, siendo dentro de diferentes campos de implementación, por ejemplo, dentro del campo médico. La tecnología dentro del ámbito médico muestra ser una clave esencial para el apoyo no solo para el personal médico, sino que brinda ayuda a personas que puedan llegar a tener alguna discapacidad o enfermedad.

El desarrollo y creación de prótesis ha sido un campo de investigación importante no solo en la actualidad, sino que a lo largo de la historia hay registros de usos de prótesis rudimentarias, siendo el primer registro histórico en el antiguo Egipto. La tecnología ha sido importante en la investigación actual de prótesis y un gran campo de desarrollo es el uso de señales bioeléctricas para el control de los mismos.

En la actualidad el desarrollo de prótesis, principalmente de extremidades, se ha enfocado en el uso de señales bioeléctricas para el control de las mismas. El estudio de las señales bioeléctricas ha traído una amplia gama de campos de trabajo no solo para el mencionado control de actuadores, se puede mencionar el análisis de los mismos para lograr un mejor entendimiento de como funciona el cuerpo humano.

El presente trabajo tiene como objetivo utilizar señales EMG y EEG para su uso dentro del control de diferentes actuadores, por medio de una interfaz biomédica y el uso de equipo de extracción de señales. Es importante mencionar que las fases previas fueron capaces de implementar una interfaz capaz de realizar un análisis y diferenciación de diferentes señales bioeléctricas para su uso dentro de simulaciones y el uso en sistemas robóticos.

Este trabajo busca implementar una mejora en el método de recolección utilizado en las fases previas de investigación y mejorar el análisis de diferenciación entre las diferentes señales creadas por el cuerpo humano. El proyecto también busca implementar una lectura y análisis en tiempo real de las señales para lograr un control en tiempo real.

El control ideal propuesto en la investigación se implementará mediante el uso de aprendizaje automático, utilizando clasificadores que se basen en las características de diversas

señales. Además de clasificar las señales de manera más precisa, se busca emplear otros tipos de sensores, como sensores de fuerza y giroscopios. Estos sensores no solo permitirán recrear movimientos exactos, sino también detectar y demostrar una señal que indique la intención de realizar un movimiento.

CAPÍTULO 4

Objetivos

4.1. Objetivo General

Mejorar las interfaces biomédicas para control de sistemas robóticos desarrollada en fases previas, incorporar sensores fuerza e iniciales, implementar un sistema de control en tiempo real y adaptar las interfaces para el control de actuadores.

4.2. Objetivos Específicos

- Mejorar las interfaces biomédicas desarrolladas anteriormente, integrando un módulo de detección, muestreo y procesamiento en tiempo real de señales EMG.
- Evaluar sensores de fuerza e iniciales y adaptarlos para su uso dentro de las interfaces.
- Aplicar algoritmos de extracción de características y aprendizaje automático a señales EMG, para el control de actuadores.
- Implementar el control de actuadores por medio de señales EMG y sensores de fuerza y dirección.
- Validar las interfaces desarrolladas reconociendo movimientos comunes y controlando prótesis de miembro superior desarrolladas anteriormente en la UVG.

CAPÍTULO 5

Alcance

Este trabajo forma parte de la línea de investigación en interfaces biomédicas, con el principal objetivo de aplicar algoritmos de clasificación basados en aprendizaje automático para el control en tiempo real de actuadores. Además, se implementó el uso de sensores inerciales para mejorar dicho control. De manera más específica, se centró en el desarrollo de una interfaz capaz de integrar estas tecnologías.

El trabajo se estructuró en cinco fases: recolección de señales EMG, desarrollo de algoritmos de clasificación, implementación de sensores inerciales, control de una mano robótica, y desarrollo de la interfaz de control.

El alcance del proyecto se vio afectado por faltas en los sensores MyoWare, por lo que no se logró la implementación deseada. Esto limitó el uso completo de las funcionalidades del sensor, impidiendo la incorporación de más sensores para obtener señales más detalladas.

A pesar de estas limitaciones, se espera que el desarrollo de este proyecto siente las bases para futuras investigaciones, ya que la implementación de este tipo de interfaces puede ofrecer un gran apoyo no solo a la tecnología de control por señales bioeléctricas, si no que también un avance en el estudio y desarrollo de prótesis.

CAPÍTULO 6

Marco teórico

6.1. Señales Bioeléctricas

Las señales bioeléctricas se refieren a las señales generadas por células nerviosas y células musculares, en específico en la membrana de potencial, las cuales producen una diferencia de potencial y corriente eléctrica para lograr diferentes procesos biológicos. Estas señales son consideradas como las señales biomédicas más importantes dentro del estudio del cuerpo humano, esto gracias a que la mayoría de sistemas dentro del cuerpo utilizan células excitables hace posible que las señales bioeléctricas sean esenciales para conocer cómo funcionan los mismos [13].

Las señales bioeléctricas se pueden clasificar principalmente en el biosistema en el que actúan los mismos, dentro de los más estudiados y utilizados en diferentes campos podemos encontrar: señales electroencefalográficas (EEG), señales electrocardiográficas (ECG) y señales electromiográficas (EMG) [14].

6.2. Señales EMG

Las señales electromiográficas son señales producidas por el sistema muscular esquelético del cuerpo humano o bien el “motor” que mueve al ser humano. Las señales EMG son generadas durante una contracción muscular y la actividad eléctrica que se genera durante este proceso la cual se diferencia en la comparación de voltajes en el proceso. Las señales EMG en lo regular se encuentran con un voltaje entre 70 a 90 mV en reposo. Estas señales son importantes en diferentes campos de investigación e innovación en el mundo científico, siendo parte clave en diferentes estudios sobre el comportamiento del cuerpo humano.

La extracción de estas señales se puede hacer por diferentes formas, principalmente podemos analizar los electrodos de superficie o bien por electrodos intramusculares [15].

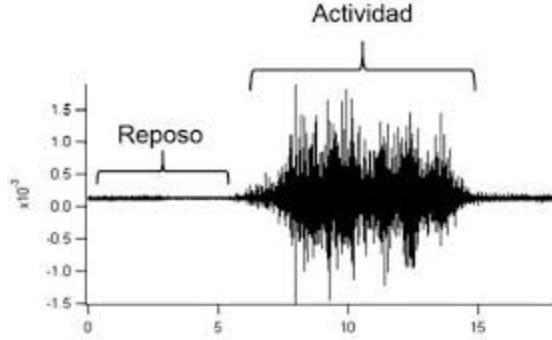


Figura 6: Ejemplo de señal EMG [16].

6.3. Características en el tiempo

6.3.1. Zero crossing

Este parámetro indica cuántas veces una onda cruza el eje cero, un procedimiento común en el procesamiento de señales. Su uso es clave para determinar la frecuencia de la señal de manera precisa [17].

6.3.2. Media del Valor Absoluto

Estima la media del valor absoluto de un grupo de datos en un tiempo específico [17].

$$\text{MAV} = \frac{1}{N} \sum_{n=1}^N |x_i| \quad (1)$$

6.3.3. Media cuadrática

Su aplicación dentro de las señales EMG, se modela como un proceso aleatorio gaussiano. En el valor calculado se relaciona con la fuerza aplicada [17].

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

6.3.4. Varianza

Es un parámetro que presenta información sobre la dispersión y la variabilidad en un conjunto de datos [17].

$$\text{VAR} = \frac{1}{N-1} \sum_{i=1}^N x_i^2 \quad (3)$$

6.3.5. Amplitud de Willison

Determina las veces que un valor absoluto de una diferencia de amplitud supera un umbral [17].

$$\text{WAMP} = \sum_{n=1}^{N-1} f(|x_{i+1} - x|), f(a) = \begin{cases} 1 & x > \varepsilon \\ 0 & \text{de lo contrario} \end{cases} \quad (4)$$

6.3.6. Curtosis

Es una medida estadística que describe la forma de como se distribuye una variable [17].

$$\text{Curtosis} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s} \right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)} \quad (5)$$

6.4. BITalino

PLUX Biosignals es una empresa cuyo objetivo es obtener soluciones para el trato y estudio de bioseñales. Cumpliendo con su objetivo, PLUX Biosignals ha desarrollado diferentes herramientas y servicios con el objetivo de apoyar a desarrollo de tecnologías basadas en el uso de bioseñales. Dentro de las herramientas y kits desarrollados podemos encontrar el BITalino siendo una plataforma de bioseñales open source diseñado para captar y analizar señales bioeléctricas [18].

BITalino es compatible con el software OpenSignals y tiene bloques de hardware con sensores capaces de captar señales electrocardiográficas, electromiográficas, electroencefalográficas, actividad electrodermica, entre otros [18].

6.5. Myoware

MyoWare 2.0 es un sensor creado por la empresa Advancer Technologies, con el objetivo de captar y visualizar señales electromiográficas por medio del uso de tres electrodos. El sensor tiene una variedad de accesorios para la mejora en la captación de señales, el sensor está diseñado para ser trabajado con un microcontrolador Arduino con la capacidad no solo de captar las señales, sino que amplificarlas y rectificarlas para facilitar el análisis y uso de las señales [19].



Figura 7: Kit de desarrollo BITalino [18].



Figura 8: Muscle Sensor MyoWare 2.0 [19].

6.6. Aprendizaje automático

El aprendizaje automático, también conocido como *Machine Learning*, es una rama de la inteligencia artificial, la cual pretende automatizar la construcción de diferentes modelos analíticos con base en series de ejemplos [20].

Gracias al uso de diferentes datos ejemplares es posible crear una mejora en el “conocimiento” de la máquina para lograr predecir o identificar diferentes series de datos, basándose

no solo por analogía o por experiencia.

El aprendizaje automático tiene diferentes ramas en las cuales se han ido creando basándose en diferentes técnicas de aprendizaje, siempre con el objetivo de mejorar con el tiempo y ser adaptable a cambios en los diferentes campos establecidos. Las dos ramas principales del aprendizaje automático se conocen como aprendizaje supervisado y el aprendizaje no supervisado [21].

6.6.1. Aprendizaje no supervisado

El aprendizaje no supervisado se refiere al tipo de aprendizaje sin el uso de etiquetas establecidas en los datos alimentados al sistema, es decir, que el sistema es el encargado de generar y establecer las diferencias entre los diferentes tipos de datos y clasificarlos por su cuenta propia. Principalmente, el aprendizaje no supervisado trata de utilizar técnicas de agrupamiento para lograr la clasificación de la serie de datos [22].

6.6.2. Aprendizaje supervisado

El aprendizaje supervisado es una rama del aprendizaje automático basado en el uso de series de datos con etiquetas establecidas para entrenar a la máquina. Este tipo de aprendizaje puede ser considerado como un aprendizaje guiado por el programador, el objetivo general de este tipo de aprendizaje es generalizar series de datos con base en la experiencia que se le ingresa a la máquina [23].

El aprendizaje automático se puede utilizar en diferentes campos, por ejemplo en la identificación y diagnóstico de una enfermedad por medio de comparación de síntomas, otro ejemplo también es la diferenciación y clasificación de señales eléctricas para su uso práctico o análisis [23].

■ Maquina de vectores de soporte

Las máquinas de vectores de soporte (SVM) es un método de aprendizaje supervisado utilizado para una clasificación estadística y basándose en un análisis de regresión [21].

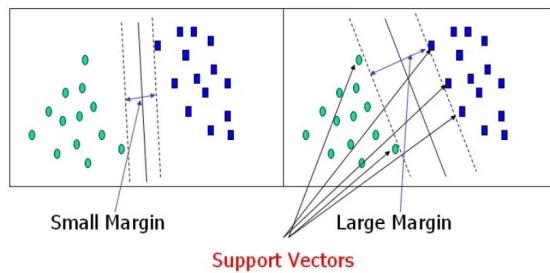


Figura 9: Representación de SVM [24].

Siendo parte del aprendizaje supervisado al momento las SVM al tener 2 o más series de datos etiquetadas, este puede actuar como un clasificador discriminativo, haciendo

uso de un hiperplano óptimo para lograr la separación de las clases. Un hiperplano bajo la definición matemática se puede establecer como un subespacio plano en razón a las dimensiones de un plano general, es decir, que para un espacio n-dimensional el hiperplano tiene dimensiones n-1 [25].

Al momento en el que las SVM no pueden dividir claramente los grupos de puntos, existe la opción de utilizar funciones de kernel las cuales ayudan al sistema a ampliar los datos en diferentes grados de dimensiones para encontrar alguna dimensión en la cual, si existe una curva establecida para separar de manera perfecta los grupos de datos, siempre haciendo uso de los hiperplanos para las diferentes y altas dimensiones que se puedan alcanzar.

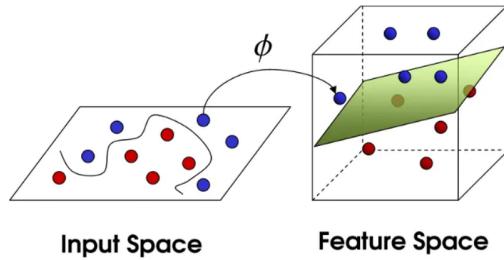


Figura 10: Representación de funciones de Kernel y representación de hiperplanos [24].

Gracias a esto, las SVM son consideradas eficaces en diferentes casos, principalmente en los grupos de datos con una dimensionalidad alta, aunque por preferencia las SVM funcionan con mejor eficacia con grupos de datos no muy extensos y que no posean muchos valores atípicos [25].

Mencionando esto, las SVM son consideradas como clasificadores binarios por lo que su implementación con una clasificación mayor a 2 clases tienden a ser utilizados de diferentes maneras que a la clasificación de 2 clases. Existen diferentes métodos de clasificación dentro de los cuales pueden entrar los métodos *One vs All*(OvA) y *One vs One*(OvO), entre otros.

- **One vs All**

El método OvA sugiere la creación de un número de máquinas de vectores de soporte equivalente al número de clases existentes. En otras palabras, se genera una SVM para cada clase, comparándola contra todas las demás. Así, el hiperplano resultante de cada modelo SVM describe cómo se diferencia cada clase en relación con el conjunto completo de clases [26].

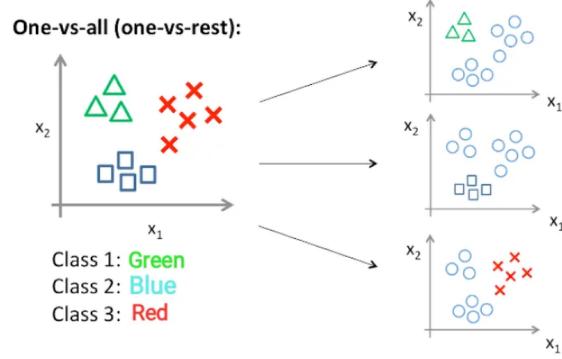


Figura 11: Representación SVM *One vs All* [27].

- One vs One

El método OvO sugiere la creación de un número de modelos SVM correspondiente a todos los pares posibles de clases en el conjunto de datos. En este caso, el número de SVM es igual a la cantidad de comparaciones uno contra uno posibles entre todas las clases del grupo de datos [26].

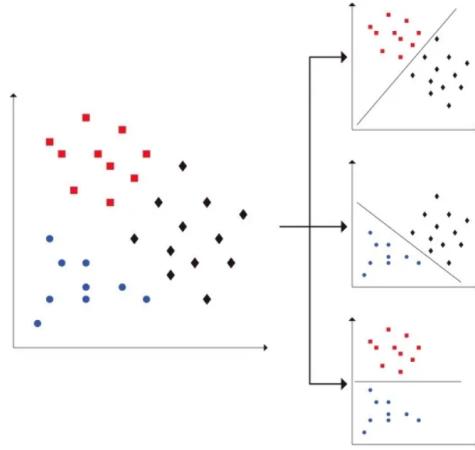


Figura 12: Representación SVM *One vs One* [27].

- K-Nearest Neighbor

El algoritmo K-Nearest Neighbor (KNN) es un método de aprendizaje supervisado ampliamente conocido, considerado uno de los algoritmos más antiguos y básicos en el campo de la clasificación. Se destaca por su simplicidad y por ser un algoritmo no paramétrico, lo que significa que no requiere de suposiciones previas sobre la distribución de los datos ni de parámetros predeterminados. En lugar de eso, KNN se basa únicamente en los datos de entrenamiento disponibles, lo que lo convierte en un modelo flexible y fácil de implementar [28].

El funcionamiento de los algoritmos KNN se fundamenta en el cálculo de distancias entre el punto que se desea clasificar y los puntos ya clasificados durante la fase de entrenamiento. En general, se utiliza la distancia euclídea para medir la proximidad

entre los puntos, aunque existen otros métodos de cálculo que se pueden implementar. Al introducir un nuevo punto o valor, el algoritmo determina a qué clase pertenece identificando los "k" vecinos más cercanos. La clasificación final se realiza en función de la clase predominante entre esos vecinos, es decir, aquella a la que pertenece la mayoría de los puntos más próximos [29].

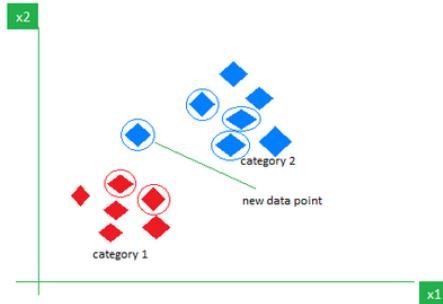


Figura 13: Representación del algoritmo KNN [30].

El uso de algoritmos KNN se limita comúnmente a aplicaciones donde los conjuntos de datos son pequeños y la precisión requerida no es extremadamente alta. Un área popular de aplicación es en los sistemas de recomendación, donde el algoritmo puede sugerir productos o contenido a los usuarios en función de similitudes con otros usuarios o productos. También es empleado en tareas como la detección de anomalías y la clasificación de imágenes [28].

■ Redes Neuronales

Las redes neuronales artificiales (RNA) o también conocidas como redes neuronales, son un método de aprendizaje supervisado que se basa en el funcionamiento de las neuronas en el cerebro humano, analizando los datos por medio de diferentes capas de procesamiento [21].

Los RNA se representan por medio de diferentes puntos o nodos, los cuales están interconectados teniendo un valor el cual se cambia mediante más datos ingresen al sistema. Este funciona gracias a los diferentes procesos que ocurren en las diferentes capas, es decir que al momento que ingresa un dato a la primera capa este lo procesa y lo manda a la segunda capa con el propósito de ir descomponiendo los datos originales en valores más abstractos dejando únicamente la información “importante” para el aprendizaje del sistema [23].

En mención, los RNA más simples solo constan de 3 capas, siendo una capa de entrada, una de procesamiento y otra de salida, aprendiendo principalmente sobre el ajuste de los parámetros que van cambiando por su cuenta gracias al algoritmo común para este método de aprendizaje. Este proceso se conoce como entrenamiento, por lo que al momento que el algoritmo termina de trabajar es posible ingresar valores reales para comprobar el funcionamiento del algoritmo [23].

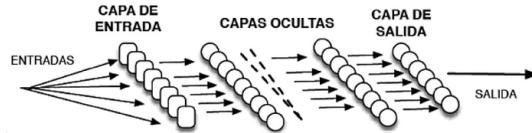


Figura 14: Representación de capas de una Red neuronal artificial simple [23].

Las redes neuronales son los algoritmos más utilizados por las ventajas que traen siendo capaces de reconocer patrones, prototipos y relaciones entre los conjuntos de datos alimentados al sistema [21].

6.7. Interfaz HMI (Human-Machine Interface)

Las interfaces humano-máquina es conocida como la interacción entre las personas con una interfaz diseñada para una interacción con un sistema o herramienta. Las interfaces HMI han sido importantes con el desarrollo de la tecnología para permitir la interacción con el usuario y el producto. Las implementaciones de las interfaces pueden tener diferentes formas, las cuales dependen de las necesidades del sistema y como se desea el control de la misma.

El rol de las interfaces HMI para facilitar las interacciones pueden caer dentro de: una interacción intuitiva del usuario, la capacidad de traer información al usuario, integrar diferentes sistemas, entre otros [31].

6.8. Sensores Inerciales

Los sensores inerciales son dispositivos que tienen la capacidad de medir aceleraciones tanto lineales como angulares. Estos dispositivos normalmente están compuestos por medio de acelerómetros, giróscopos y magnetómetros. Además muestran ser útiles en diferentes campos como lo son: la robótica, animación, estudios del movimiento humano, entre otros [32].

6.9. Actuadores

Un actuador es un dispositivo capaz de convertir energía eléctrica en movimiento o torque. Existen diferentes tipos de actuadores, clasificados según sus especificaciones y usos. Estos dispositivos son fundamentales para el desarrollo tecnológico y tienen múltiples aplicaciones, tales como el control automatizado, la robótica y el posicionamiento de componentes, entre otros [32].

6.10. Prótesis

De acuerdo a la enciclopedia A.D.A.M. una prótesis es un dispositivo diseñado para remplazar una parte faltante del cuerpo o ayudar a que una parte del cuerpo funcione mejor [33].

Las prótesis han sido parte importante dentro de la historia de la humanidad, donde se han encontrado evidencias que se han realizado prótesis a lo largo de la historia. Siendo de las más antiguas una momia con una prótesis del dedo gordo del pie hecha de cuero para ayudar al movimiento de la persona [34].



Figura 15: Prótesis realizada en el antiguo Egipto [34].

Hoy en día se muestran diferentes investigaciones y desarrollos sobre como mejorar las prótesis para la gente con la necesidad de una, con el desarrollo de prótesis biónicas con el objetivo de lograr que el miembro artificial tenga la capacidad de ser controlado por las señales naturales del cuerpo sin la necesidad de tener movimientos fijos y preestablecidos. La implementación de motores y el desarrollo de tecnologías de inteligencia artificial han permitido una evolución en la creación de prótesis [34].

CAPÍTULO 7

Recolección de señales EMG

Con el fin de adquirir un mejor entendimiento de las señales EMG, se decidió emplear distintos tipos de sensores para su captura y procesamiento. Además, se realizaron experimentos con estas señales para lograr su recopilación y procesamiento en tiempo real, con el fin de utilizarlas en el control de un actuador.

7.1. Dispositivos de adquisición para señales EMG

Para la adquisición de las señales EMG, se analizaron diferentes opciones de sensores, destacando el uso del dispositivo BITalino, previamente empleado en fases anteriores en la Universidad del Valle de Guatemala. En particular, se utilizó la placa *BITalino (r)evolution Board*, que cuenta con múltiples canales para la captura de diversas señales bioeléctricas. Con el objetivo de explorar alternativas al BITalino, se experimentó con el sensor MyoWare. Este dispositivo está diseñado específicamente para obtener señales EMG, presentándolas en diferentes modalidades. Además, su compatibilidad con microcontroladores como Arduino lo convierte en una herramienta versátil para el desarrollo de diversos proyectos. Para este estudio, se empleó el modelo *MyoWare 2.0* junto con sus accesorios para comunicación inalámbrica.

7.1.1. Dispositivo BITalino

Para la utilización del sensor BITalino se hizo uso del software de MATLAB ya que cuenta con una toolbox la cual posee diferentes funciones para facilitar el uso del sensor y obtener los mejores resultados con el sensor. La función a utilizar fue la función *Read()*, la cual permite la lectura de los n valores anteriores guardados en el buffer permitiendo una

lectura rápida y exacta de los valores leídos en el sensor.

En fases enteriores, se determinó que las frecuencias de muestreo en las que funciona el sensor son de 1, 10, 100 y 1000 Hz.

7.1.2. Dispositivo MyoWare

Para utilizar el dispositivo MyoWare, fue necesario integrar un microcontrolador ESP32 para establecer la comunicación entre el sensor y MATLAB mediante comunicación serial. Entre los accesorios disponibles, el accesorio de comunicación inalámbrica fue el más utilizado. Este microcontrolador funciona a través del módulo ESP32-WROOM, el cual utiliza comunicación BLE (Bluetooth Low Energy) para enlazar el accesorio con el dispositivo ESP32. El sensor MyoWare ofrece diferentes modos de representación de la señal capturada, gracias a su diseño que incorpora diversas configuraciones y filtros, como se puede observar en la Figura 16.

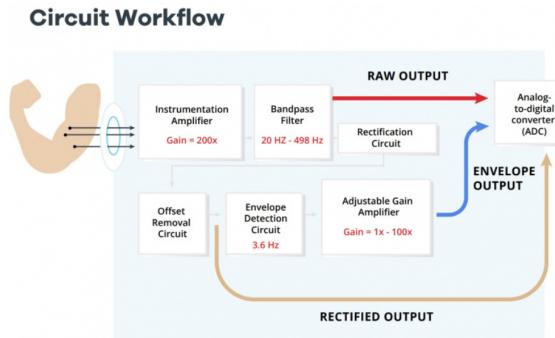


Figura 16: Esquema de circuito utilizado en el sensor MyoWare [19].

Se encontraron diferentes problemas del sensor por lo que se optó por la funcionalidad “Envelope” la cual representa la señal como un pulso. Para obtener este tipo de señal, es necesario cambiar las configuraciones del sensor a las de tipo *ENVELOPE*.

7.2. Algoritmos de adquisición

Como se mencionó anteriormente, se utilizó la función *read()* del toolbox de BITalino para la lectura de señales, las cuales se almacenan inicialmente en el buffer del sensor. Una vez capturados los valores y gracias a la comunicación con MATLAB, estos se almacenan en una variable. Consecuentemente el sensor vacía el buffer y se prepara para recibir nuevos datos.

Para gestionar esta interacción entre MATLAB y los sensores, es necesario realizar una configuración que mantenga al sensor a la espera de nuevas lecturas. Esto se logra mediante la función *startBackground(Objeto)*, que inicializa el sensor y le permite leer datos en segundo plano sin interrumpir la ejecución de otros comandos o funciones en MATLAB.

Para implementar una funcionalidad de lectura en tiempo real, es esencial que el sensor esté constantemente capturando valores. Por esta razón, se implementó un ciclo *while* en el cual el sensor lee datos cada 0.25 segundos, esperando la activación de una bandera para almacenar los valores obtenidos a través de la función *read()*.

Es importante destacar que el toolbox de BITalino despliega las señales capturadas en múltiples canales. Esto gracias a la existencia de diferentes tipos de señales las cuales puede capturar el dispositivo. Sin embargo, en esta implementación, únicamente se seleccionó el canal correspondiente a las señales EMG.

7.3. Algoritmo de captura en tiempo real

Con el objetivo de obtener lecturas en tiempo real el BITalino debe de estar leyendo constantemente, por lo que es necesario un algoritmo que tenga la capacidad de detectar cuando se desea una lectura.

Para implementar el funcionamiento de lectura en tiempo real, se utilizó el sensor MyoWare y su funcionalidad *Envelope*. Primero se inicializó la comunicación entre el sensor y sus diferentes accesorios de comunicación con el ESP32 y la misma entre el ESP32 y MATLAB. Utilizando la librería de MyoWare en el ambiente de programación Arduino IDE se debe inicializar el sensor en una propiedad de *ENVELOPE* y realizar el procedimiento de comunicación con el ESP32 dentro de un ciclo *while* para que el MyoWare siempre esté enviando valores. Al recibir la información en el ESP32 se realizó un límite para convertir la señal en una bandera, siendo esta la que recibe MATLAB. Consecuentemente recibida la bandera, MATLAB ejecuta los comandos para recibir la señal del BITalino.

Este procedimiento se puede observar en la Figura 17.

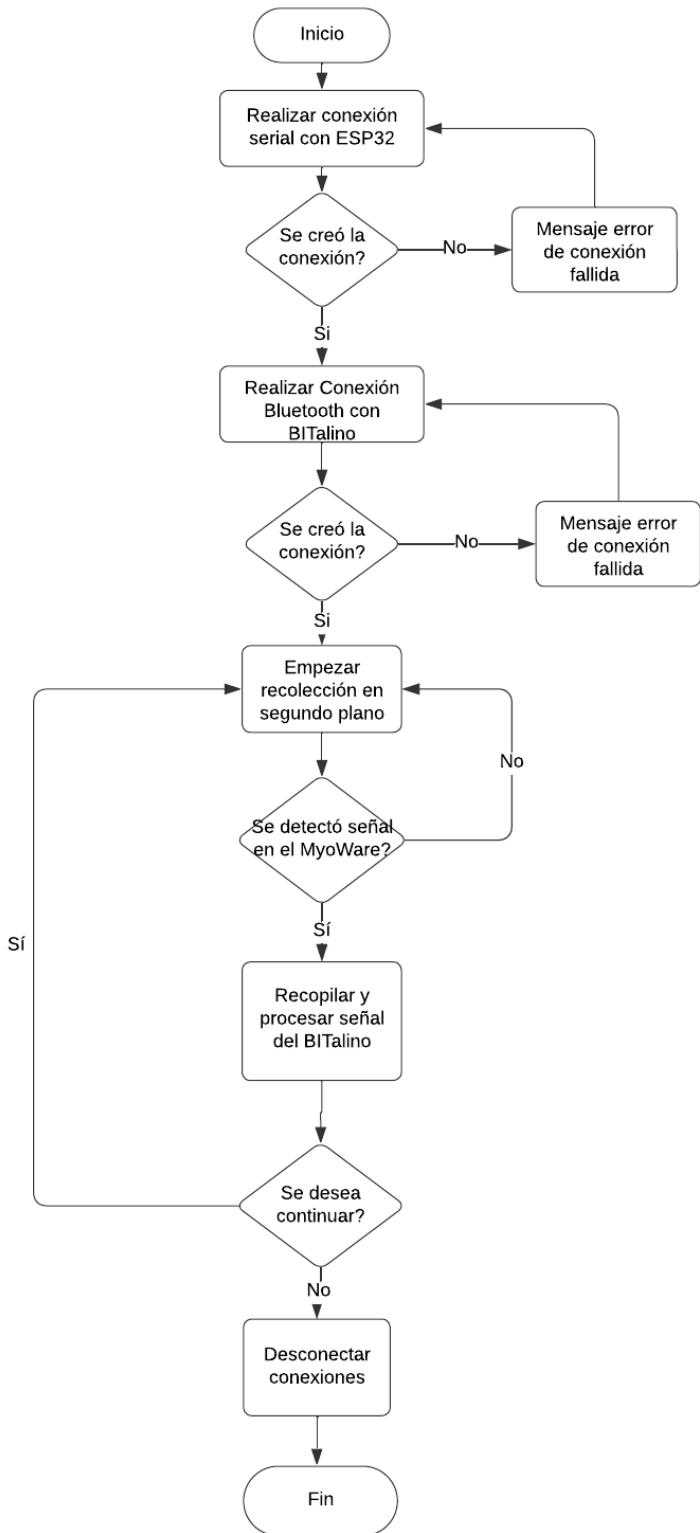


Figura 17: Diagrama de flujo de algoritmo de adquisición en tiempo real.

7.4. Procesamiento de señal

Al tener la señal en bruto deseada, se emplearon dos filtros con el objetivo de obtener la mejor señal posible. Con el objetivo de evitar problemas al momento de su clasificación. En investigaciones anteriores se determinó que las señales EMG se encuentran entre el rango de frecuencia de 10 Hz y 350 Hz [4]. En fases anteriores del proyecto se diseñó un filtro IIR de tipo Butterworth pasa bandas entre los rangos de 20 Hz y 450 Hz. Además, se implementó un filtro rechaza bandas entre 59 Hz y 61 Hz para eliminar el ruido causado por la red eléctrica [6].

7.5. Interfaz de colección de señales

Antes de iniciar el entrenamiento y clasificación de señales, se implementó una interfaz desarrollada en fases anteriores del proyecto. Esta interfaz fue creada utilizando la aplicación integrada de *App Designer* en MATLAB. La interfaz incluye diversas opciones con el objetivo de determinar las características de extracción de las señales. Además de permitir la conexión con un dispositivo BITalino. La aplicación también muestra las señales adquiridas en gráficos, representando los valores en voltaje. Gracias a esto la interfaz es capaz de mostrar las señales adquiridas en tiempo real durante el tiempo establecido [6].

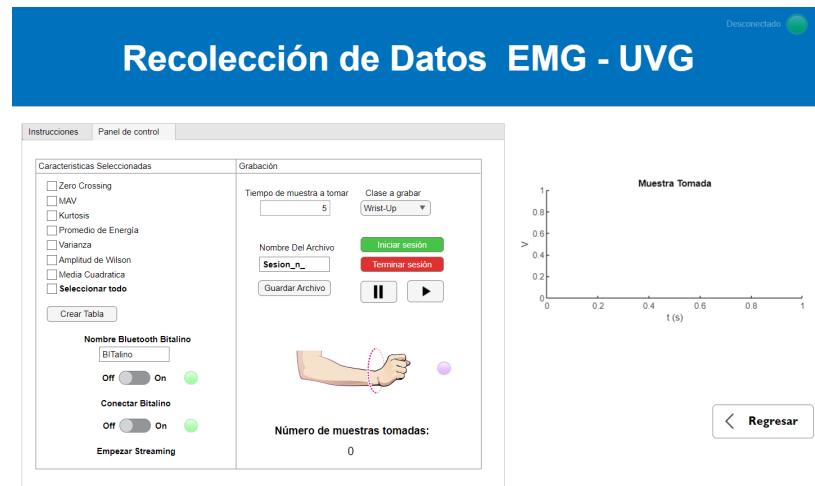


Figura 18: Interfaz de recolección de señales EMG de fases anteriores.

Aunque se contaba con la interfaz de fases anteriores, fue necesario realizar ajustes para adaptarla a esta implementación. En primer lugar, se eliminó la restricción de clases, ya que el objetivo de este trabajo era recopilar distintos tipos de señales, no solo un grupo específico. Además, se incorporó el uso de un segundo sensor BITalino, lo que requirió implementar un segundo sistema de conexión y añadir un nuevo modo de uso en la interfaz: uno para operar con un sensor y otro para dos sensores.

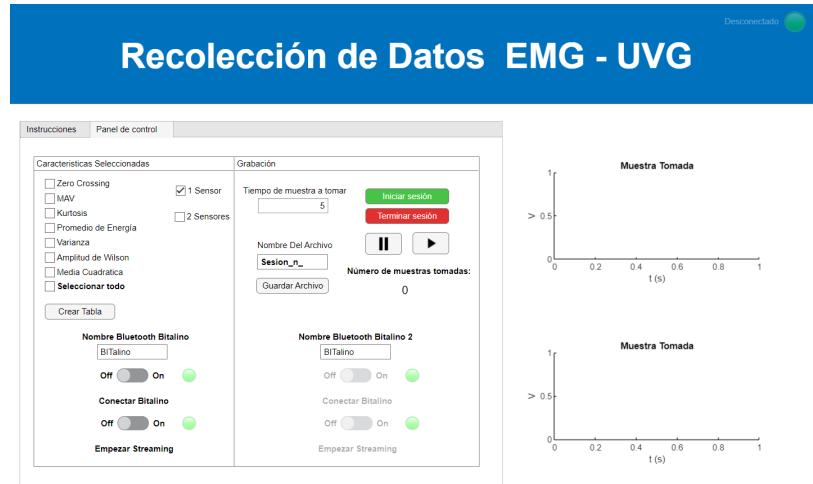


Figura 19: Segunda versión de interfaz de recolección de señales EMG.

En la siguiente figura se puede observar el diagrama de flujo del algoritmo implementado, para la interfaz de recolección EMG.

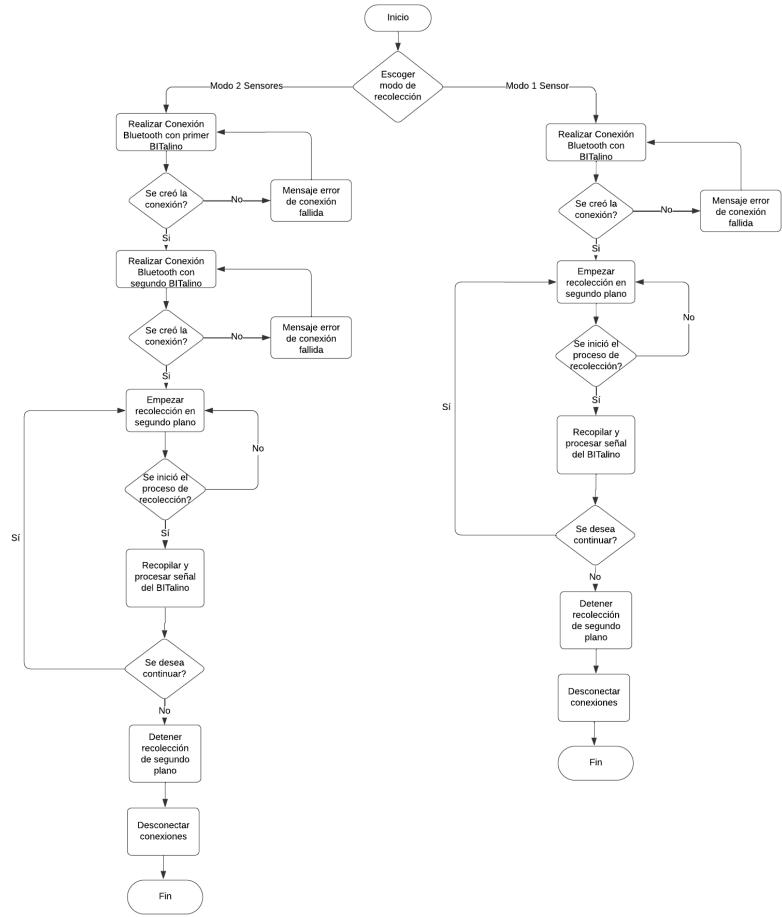


Figura 20: Diagrama de flujo de algoritmo de interfaz de recolección EMG.

CAPÍTULO 8

Clasificadores de señales EMG

En este capítulo se presentan los clasificadores utilizados junto con sus respectivos resultados. Para ello, los resultados se ilustran mediante matrices de confusión, las cuales permiten evaluar la precisión de los algoritmos. Estos resultados demuestran el desempeño de los distintos métodos frente a diversas pruebas. Para validar su funcionalidad, se realizaron ensayos utilizando una variedad de gestos y diferentes cantidades de sensores.

8.1. Gestos Utilizados

Como se mencionó anteriormente, se llevaron a cabo pruebas utilizando distintas cantidades de gestos, con el objetivo de evaluar el rendimiento de los clasificadores en diferentes situaciones. Estas pruebas brindan una perspectiva más amplia de la capacidad para manejar datos con diferentes números de clases. En particular, se realizaron experimentos con colecciones de señales que incluyen 3 y 4 gestos, lo que permitió analizar la eficacia de los algoritmos al adaptarse a estas variaciones. Esto proporciona información valiosa sobre la robustez y versatilidad de los clasificadores frente a distintas configuraciones de señales.

8.1.1. Colección de 3 gestos

Para obtener los mejores resultados en la recolección de señales, la posición ideal de los electrodos consiste en colocar dos de ellos sobre el músculo principal, mientras que un tercer electrodo se sitúa a cierta distancia para actuar como referencia. Esta disposición se muestra en la Figura 21.



Figura 21: Posición de los electrodos para la captura de los 3 gestos.

El primer gesto fue el movimiento del puño para arriba. Este se puede observar en la Figura 22.



Figura 22: Gesto 1: Movimiento de puño hacia arriba.

El segundo gesto fue el mismo que el primer gesto solo que realizado dos veces continuas.

El tercer gesto fue tomar un movimiento de referencia manteniendo el brazo en reposo. Este movimiento se puede observar en la Figura 23.



Figura 23: Gesto 3: Brazo en reposo.

8.1.2. Colección de 4 gestos

8.2. *Classification Learner*

Para comparar y seleccionar los clasificadores más adecuados, se utilizó la aplicación integrada *Classification Learner* de MATLAB, que permite entrenar una amplia variedad de modelos de clasificación. La aplicación ofrece métodos como: Máquinas de Vectores de Soporte (SVM), Redes Neuronales Artificiales (ANN), y *k-Nearest Neighbor* (KNN), entre otros.

Con el objetivo de evaluar diferentes opciones, se seleccionaron tres métodos principales: SVM, ANN y KNN. El entrenamiento se centró en estos modelos, eligiendo aquellos que mostraron el mayor porcentaje de precisión en la validación. Además, se evaluaron los porcentajes de validación en fase de prueba, con el objetivo de comprobar su utilidad con datos diferentes a los que se evaluaron. La aplicación solo requiere una base de datos estructurada según las características deseadas, junto con las etiquetas correspondientes a cada dato.

Para la evaluación, se utilizó una recolección de 180 muestras, distribuidas en 60 para cada clase. Adicionalmente, se creó una base de datos de prueba con 60 muestras, divididas en 20 por clase, para verificar el rendimiento de los clasificadores.

8.3. SVM

La aplicación ofrece varios tipos de SVM en su biblioteca para el entrenamiento. En este caso, se seleccionaron tres SVM con distintos tipos de hiperplanos: Lineal, Cúbica y Cuadrática.

El porcentaje de validación obtenido se puede observar en la Tabla 1.

SVM	Número de Clases	Porcentaje de Validación
SVM Linear	3	95.7 %
SVM Cuadrática	3	93.83 %
SVM Cúbica	3	93.83 %

Cuadro 1: Resultados de porcentaje de validación para las SVM seleccionadas.

Entrando en mayor detalle podemos observar las matrices de confusión de los diferentes clasificadores SVM.

8.3.1. SVM Linear

Resultados SVM linear en fase de validación

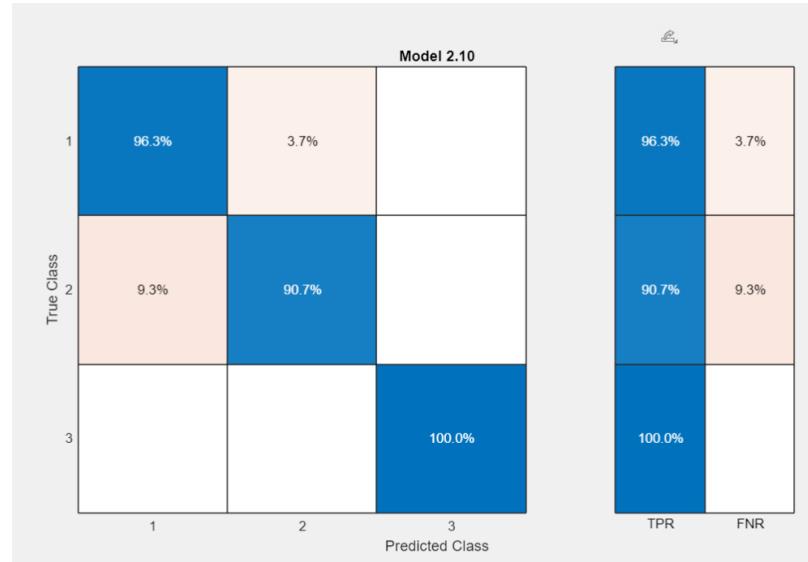


Figura 24: Matriz de confusión con etapa de validación para SVM linear.

Resultados SVM linear en fase de prueba

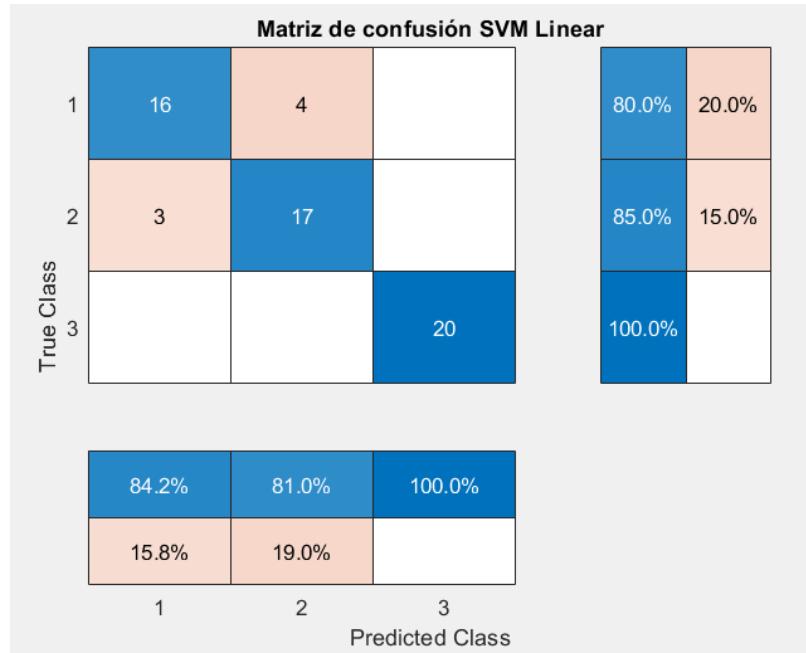


Figura 25: Matriz de confusión con etapa de prueba para SVM linear con tres gestos.

8.3.2. SVM Cuadrática

Resultados SVM cuadrática en fase de validación

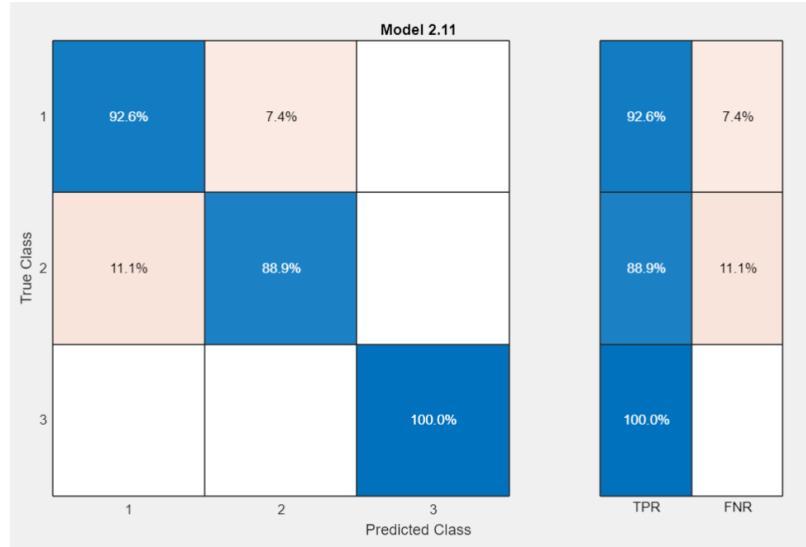


Figura 26: Matriz de confusión con etapa de validación para SVM cuadrática con tres gestos.

Resultados SVM cuadrática en fase de prueba

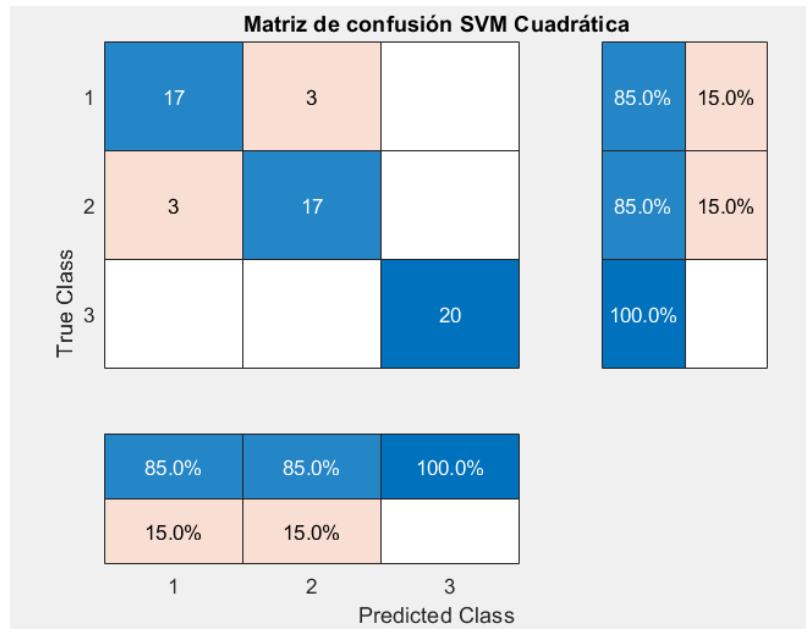


Figura 27: Matriz de confusión con etapa de prueba para SVM cuadrática con tres gestos.

8.3.3. SVM Cúbica

Resultados SVM cúbica en fase de validación

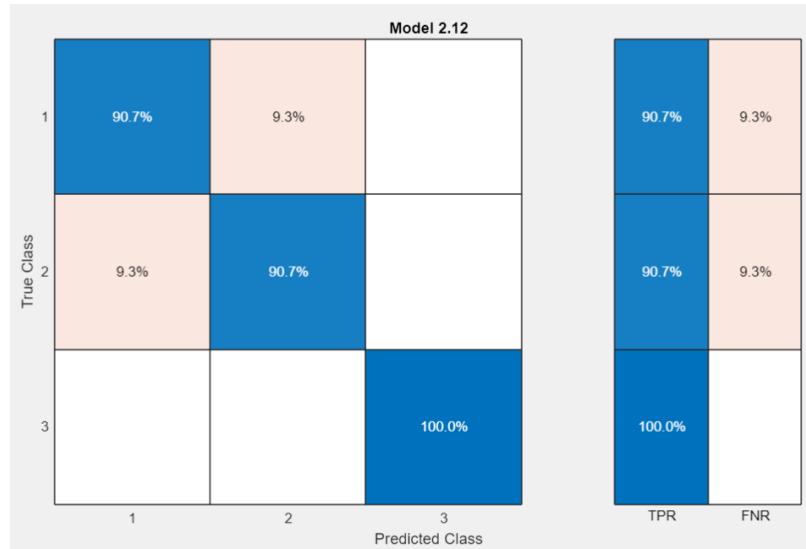


Figura 28: Matriz de confusión con etapa de validación para SVM cúbica con tres gestos.

Resultados SVM cúbica en fase de prueba

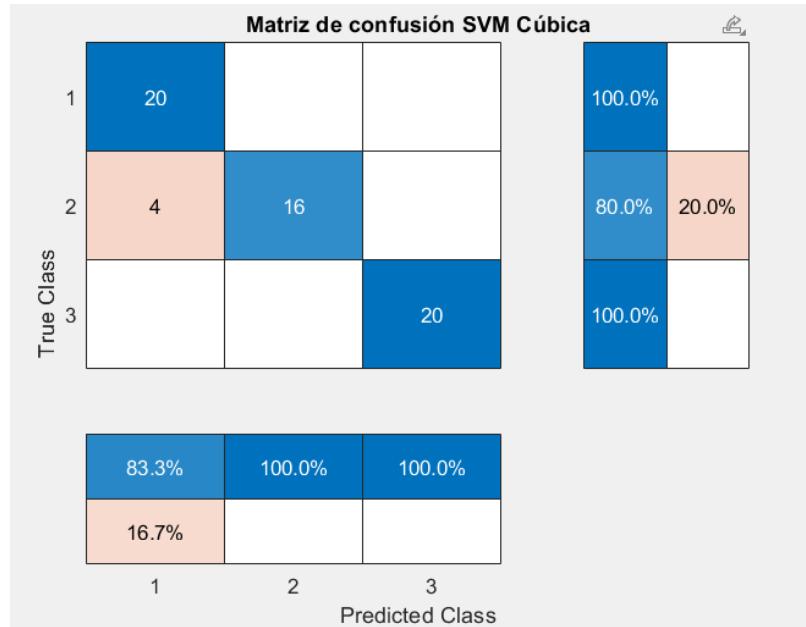


Figura 29: Matriz de confusión con etapa de prueba para SVM cúbica con tres gestos.

Mencionando los resultados obtenidos podemos observar como los porcentajes de prueba resultan ser mejores para la SVM Cúbica. En comparación a los porcentajes de validación la SVM cúbica presenta mejores resultados en la práctica, como se observa en la Figura 29.

8.4. ANN

Al igual que con los modelos SVM, se seleccionaron tres tipos de redes neuronales para evaluar su efectividad con los datos de entrenamiento. Estos modelos incluyen una red neuronal de dos capas, una red neuronal de tres capas y una red neuronal de arquitectura ancha.

El mejor porcentaje de validación calculado fue el de la red neuronal de tres capas como se puede observar en la Tabla 2.

Red Neuronal Artificial	Número de Clases	Porcentaje de Validación
ANN tres capas	3	92.6 %
ANN dos capas	3	92 %
ANN ancha	3	92 %

Cuadro 2: Resultados de porcentaje de validación para las redes neuronales seleccionadas.

Entrando en mayor detalle podemos observar las matrices de confusión de las diferentes redes neuronales creadas.

8.4.1. ANN de tres capas

Resultados ANN de tres capas en fase de validación

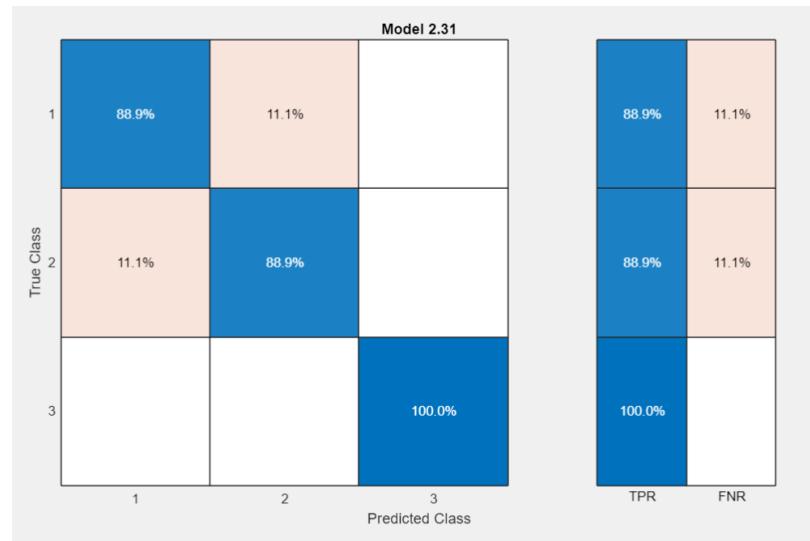


Figura 30: Matriz de confusión con etapa de validación para ANN de tres capas con tres gestos.

Resultados ANN de tres capas en fase de prueba

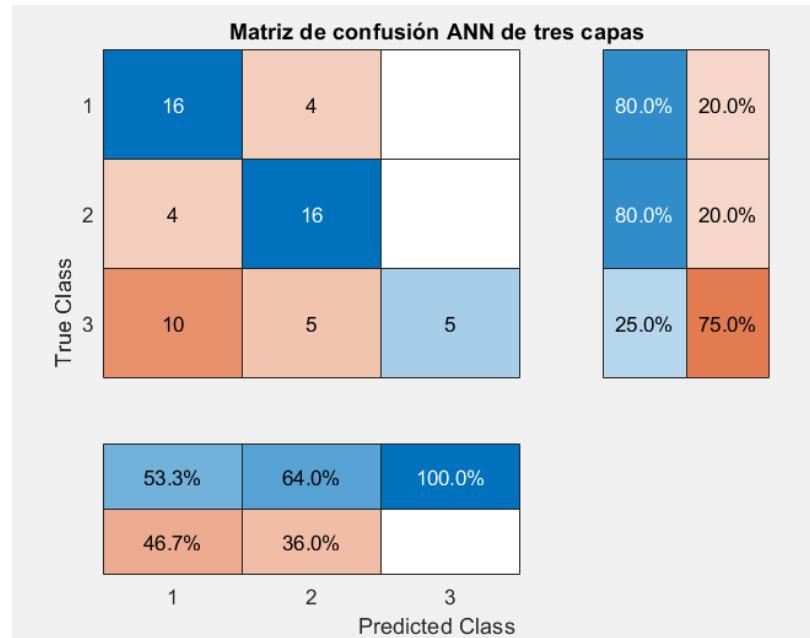


Figura 31: Matriz de confusión con etapa de prueba para ANN de tres capas con tres gestos.

8.4.2. ANN de dos capas

Resultados ANN de dos capas en fase de validación

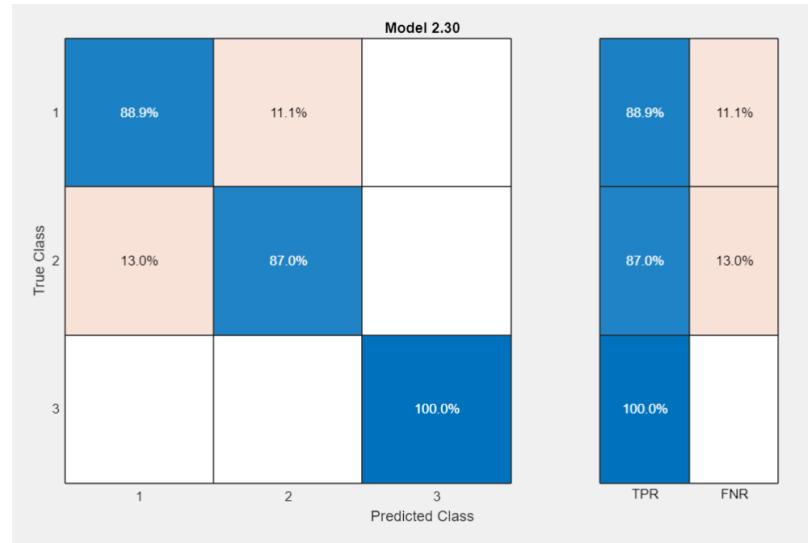


Figura 32: Matriz de confusión con etapa de validación para ANN de dos capas con tres gestos.

Resultados ANN de dos capas en fase de prueba

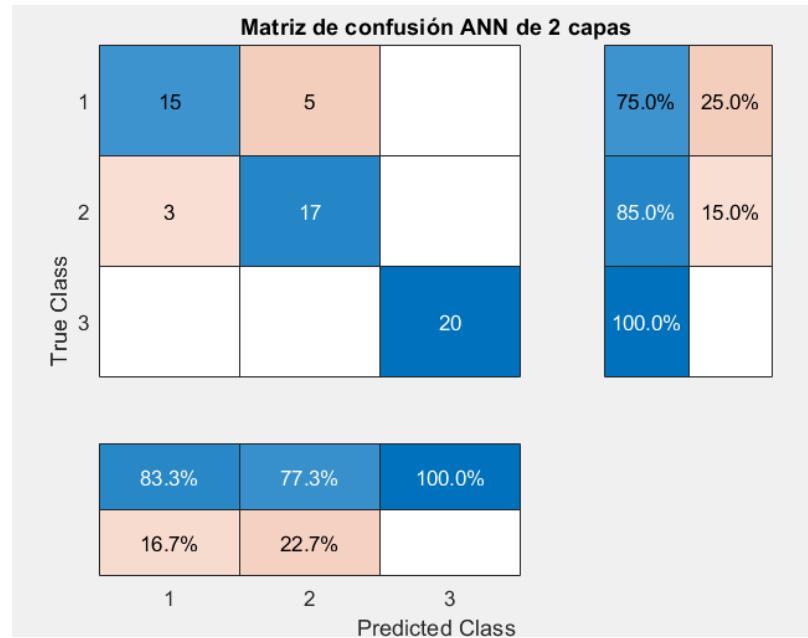


Figura 33: Matriz de confusión con etapa de prueba para ANN de dos capas con tres gestos.

8.4.3. ANN de arquitectura ancha

Resultados ANN de arquitectura ancha en fase de validación

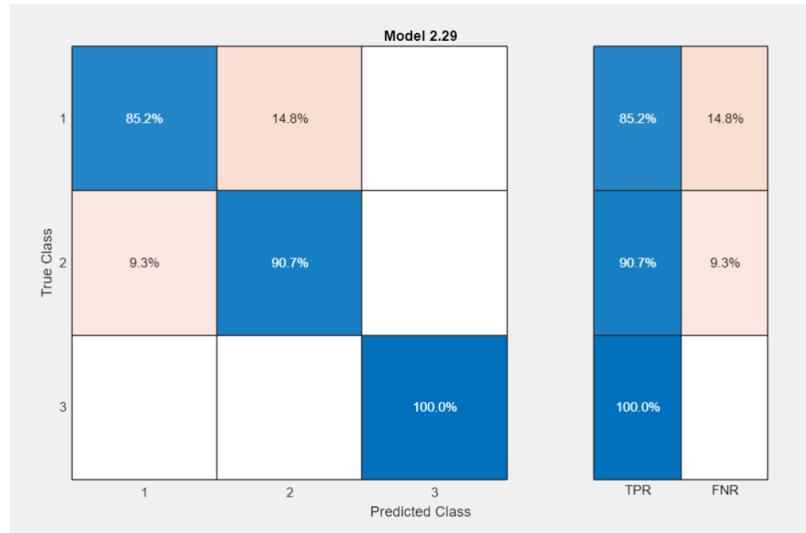


Figura 34: Matriz de confusión con etapa de validación para ANN ancha con tres gestos.

Resultados ANN de arquitectura ancha en fase de prueba

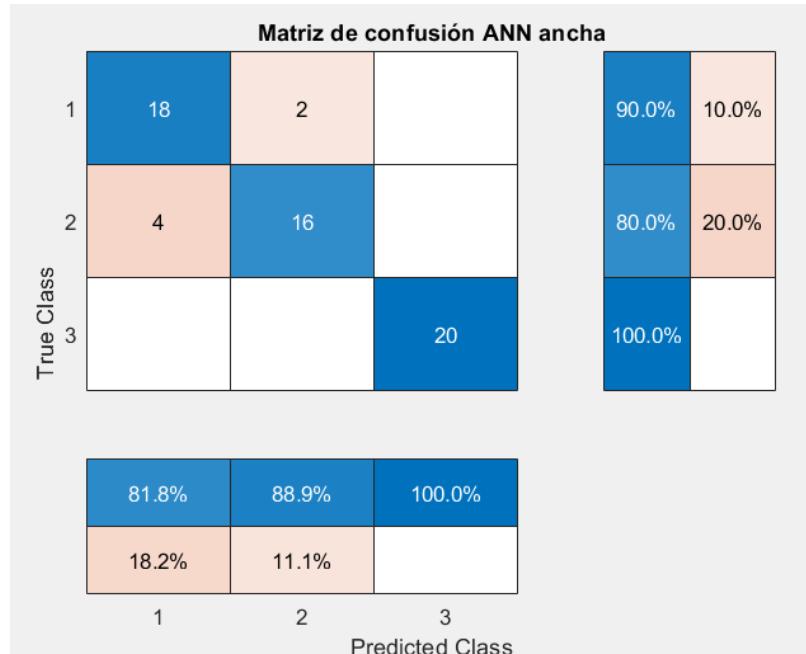


Figura 35: Matriz de confusión con etapa de prueba para ANN ancha con tres gestos.

Mencionando los resultados obtenidos podemos observar como los porcentajes de prueba resultan ser mejores para la ANN de tres capas. En comparación a los porcentajes de validación la ANN de arquitectura ancha presenta mejores resultados en la práctica, vistos en la Figura 35.

8.5. KNN

De igual forma que los modelos anteriores, se seleccionaron 3 algoritmos KNN para observar su efectividad con los valores de entrenamiento. Se seleccionaron los algoritmos KNN Pesada, KNN Fina y KNN Cúbica.

El mejor porcentaje de validación calculado fue el del algoritmo KNN Pesada como se puede observar en la Tabla 3.

k-Nearest Neighbor	Número de Clases	Porcentaje de Validación
KNN Pesada	3	92 %
KNN Fina	3	89.5 %
KNN Cúbica	3	88.9 %

Cuadro 3: Resultados de porcentaje de validación para las KNN seleccionadas.

Entrando en mayor detalle podemos observar las matrices de confusión de los diferentes algoritmos creados.

8.5.1. KNN Pesada

Resultados del algoritmo KNN Pesada en fase de validación

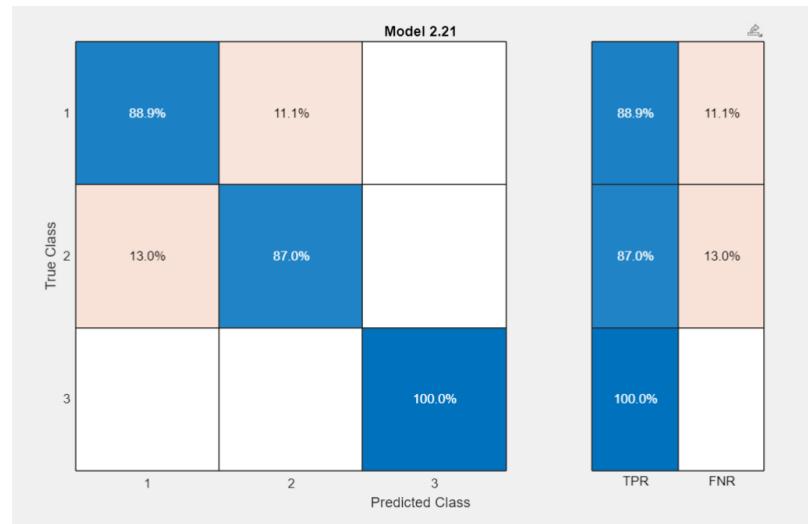


Figura 36: Matriz de confusión con etapa de validación para KNN pesada con tres gestos.

Resultados del algoritmo KNN Pesada en fase de prueba

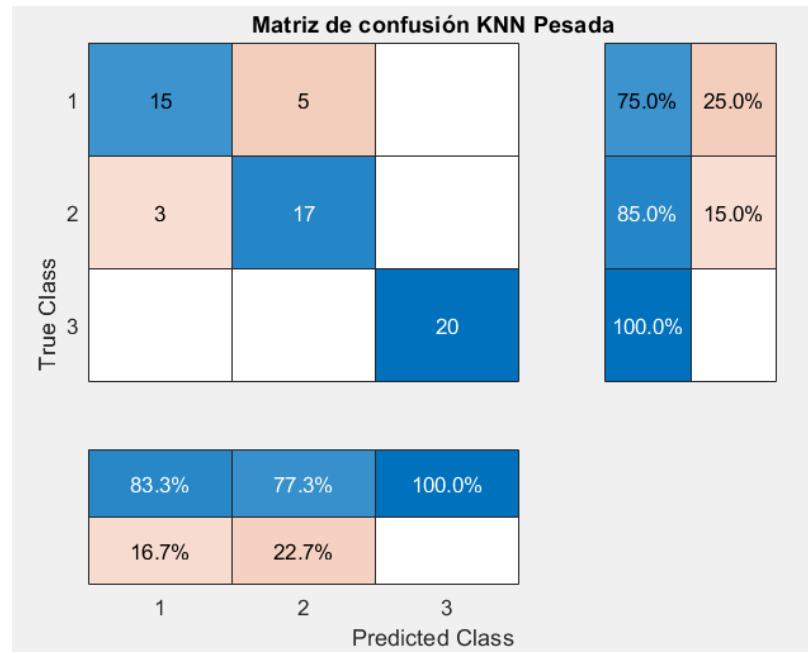


Figura 37: Matriz de confusión con etapa de prueba para KNN pesada con tres gestos.

8.5.2. KNN Fina

Resultados del algoritmo KNN Fina en fase de validación

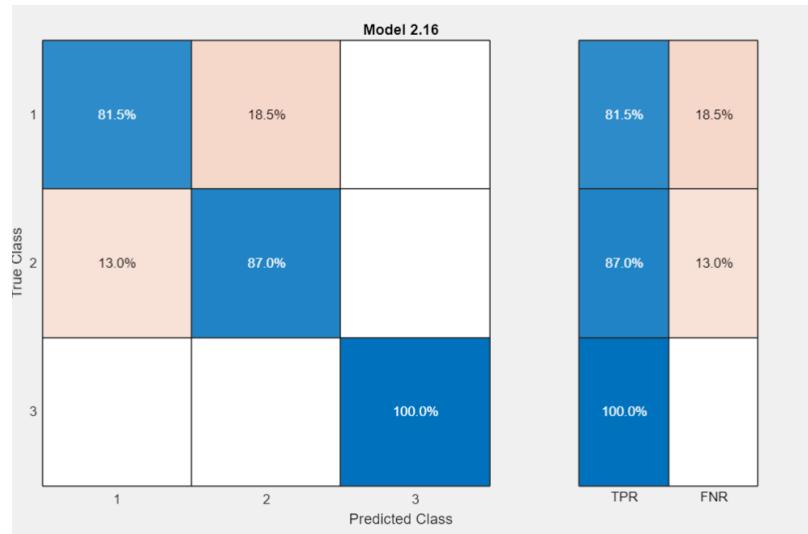


Figura 38: Matriz de confusión con etapa de validación para KNN fina con tres gestos.

Resultados del algoritmo KNN Fina en fase de prueba

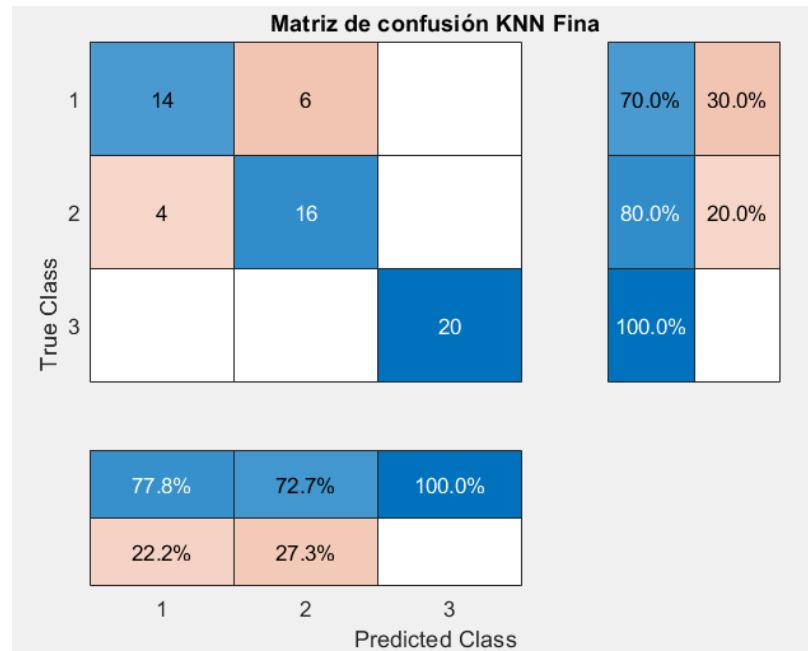


Figura 39: Matriz de confusión con etapa de prueba para KNN fina con tres gestos.

8.5.3. KNN Cúbica

Resultados del algoritmo KNN cúbica en fase de validación

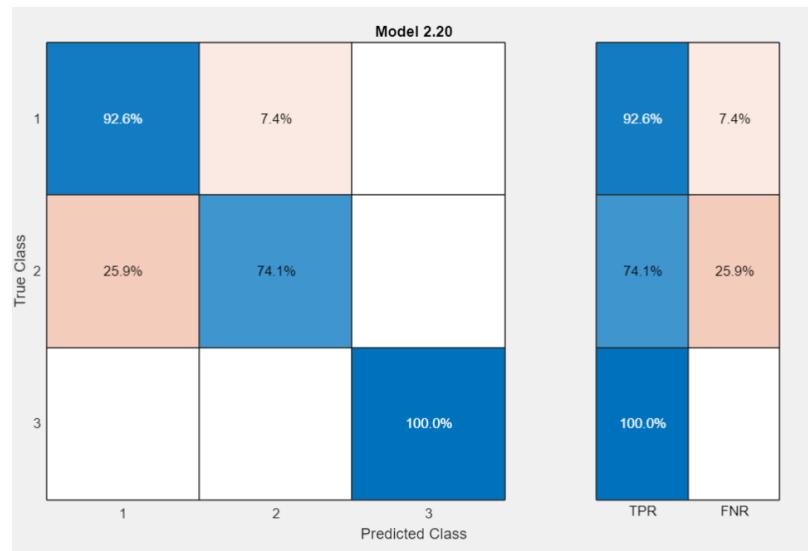


Figura 40: Matriz de confusión con etapa de validación para KNN cúbica con tres gestos.

Resultados del algoritmo KNN cúbica en fase de prueba

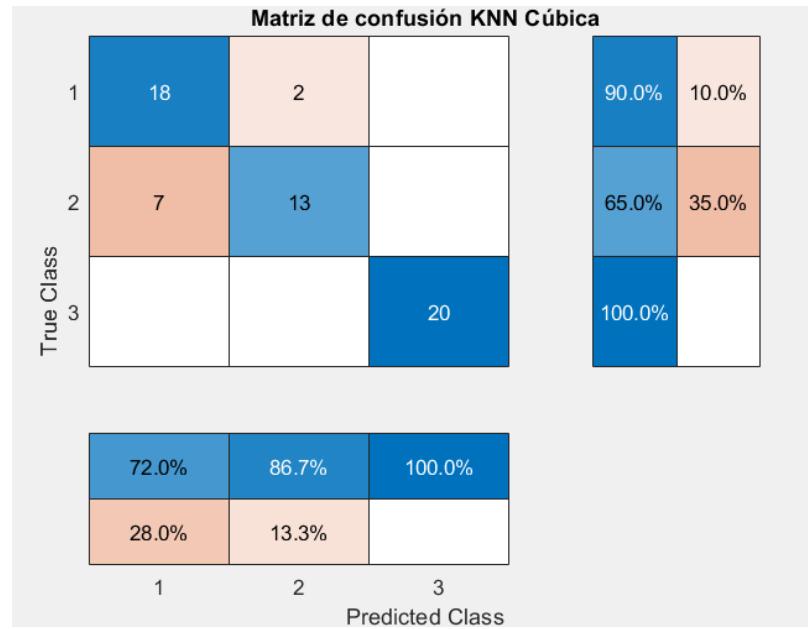


Figura 41: Matriz de confusión con etapa de prueba para KNN cúbica con tres gestos.

En el caso de los algoritmos KNN podemos observar que en porcentajes de validación el modelo KNN Pesado tiene el más alto. Cabe mencionar que los porcentajes de predicción en la etapa de testeо fueron mejores para la KNN cúbica como se puede ver en la Figura 41.

CAPÍTULO 9

Sistema de sensores iniciales

CAPÍTULO 10

Mano antropomórfica UVG

Con el fin de presentar el control de actuadores se utilizó la mano antropomórfica desarrollada en la Universidad del Valle de Guatemala [11]. En el siguiente capítulo se detallará el uso de la misma y la comunicación entre los diferentes componentes.

10.1. Acercamiento con mano antropomórfica

En el caso de la mano antropomórfica UVG, se emplean dos tipos de servomotores para controlar el movimiento de las distintas partes del proyecto. Estos servomotores, de la marca Dynamixel, corresponden a las series AX-12A y XL-320.

Para operar la mano, se utiliza el microcontrolador OpenCM, diseñado específicamente para el control de motores de alta potencia, en particular los servomotores mencionados. El programa para este microcontrolador se basa en el software de Arduino, lo que permite compartir muchas de sus funciones y modos de trabajo, facilitando la programación y control de los motores.

10.2. Algoritmos de control

En investigaciones previas, se desarrolló un programa específico para el control de los servomotores utilizados, diseñado especialmente para integrarse con el software Unity y generar los movimientos deseados en el entorno virtual [12]. El código convierte los ángulos percibidos en Unity a ángulos reconocidos por el microcontrolador, lo que permite reproducir el movimiento exacto en la mano antropomórfica. Además, el programa se comunica con MATLAB, que actúa como intermediario entre la computadora y el controlador.

Para implementar este programa en el contexto de la interfaz biomédica, fue necesario modificar algunas partes del código, optimizando el uso de la mano antropomórfica y eliminando la dependencia de Unity. Aunque la mayor parte del código se mantuvo, se ajustaron los ángulos de entrada para adaptarlos al controlador. A través de pruebas con diferentes ángulos, se identificaron los movimientos adecuados para controlar la mano. En dichas pruebas, se encontró que los dedos operan dentro de un rango de ángulos entre 127° y 190° , por lo que los movimientos se basaron en la apertura y cierre de las distintas partes de la mano.

Dado que ya no es necesaria la integración con Unity, se simplificó el sistema al eliminar la librería y los comandos específicos para la comunicación entre Unity y el dispositivo. Ahora, el control se gestiona únicamente mediante comunicación serial entre MATLAB y el controlador OpenCM. El control se implementa mediante una estructura condicional tipo *switch-case*, donde cada caso corresponde a un conjunto de ángulos de movimiento que se envían al controlador cuando se cumplen las condiciones correspondientes. A través de esta comunicación serial, el controlador ejecuta los movimientos precisos en los servomotores.

Este procedimiento se puede observar en la Figura 42

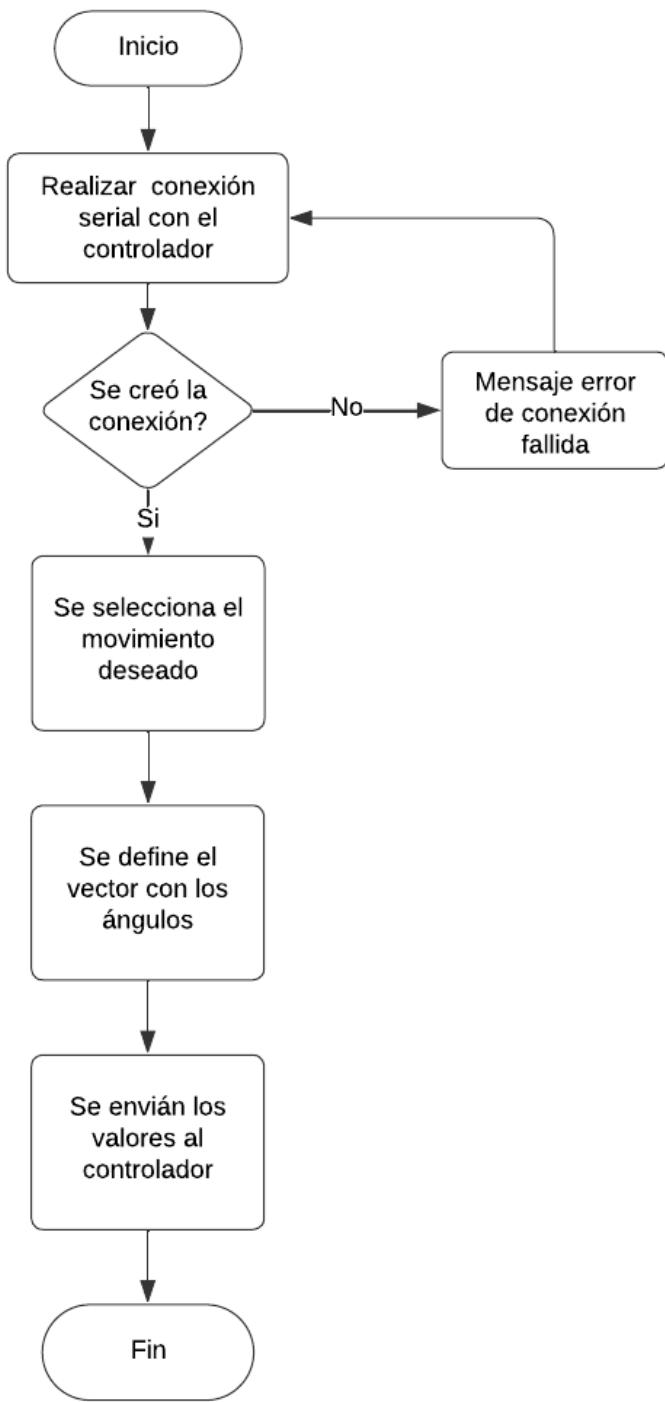


Figura 42: Diagrama de flujo de algoritmo de control de la mano antropomórfica.

10.3. Movimientos Predeterminados

Dado que el proyecto requiere la clasificación de diferentes gestos y con el objetivo de recrear movimientos comunes de la mano, se tomó la decisión de implementar el gesto de conteo con los dedos.

Con esto en mente, y utilizando los ángulos de apertura y cierre de los dedos previamente identificados, se definieron variables en MATLAB que contienen los ángulos específicos para cada servomotor de la mano, logrando así el movimiento deseado. Estos movimientos se pueden observar en las siguientes figuras.

Movimiento 0

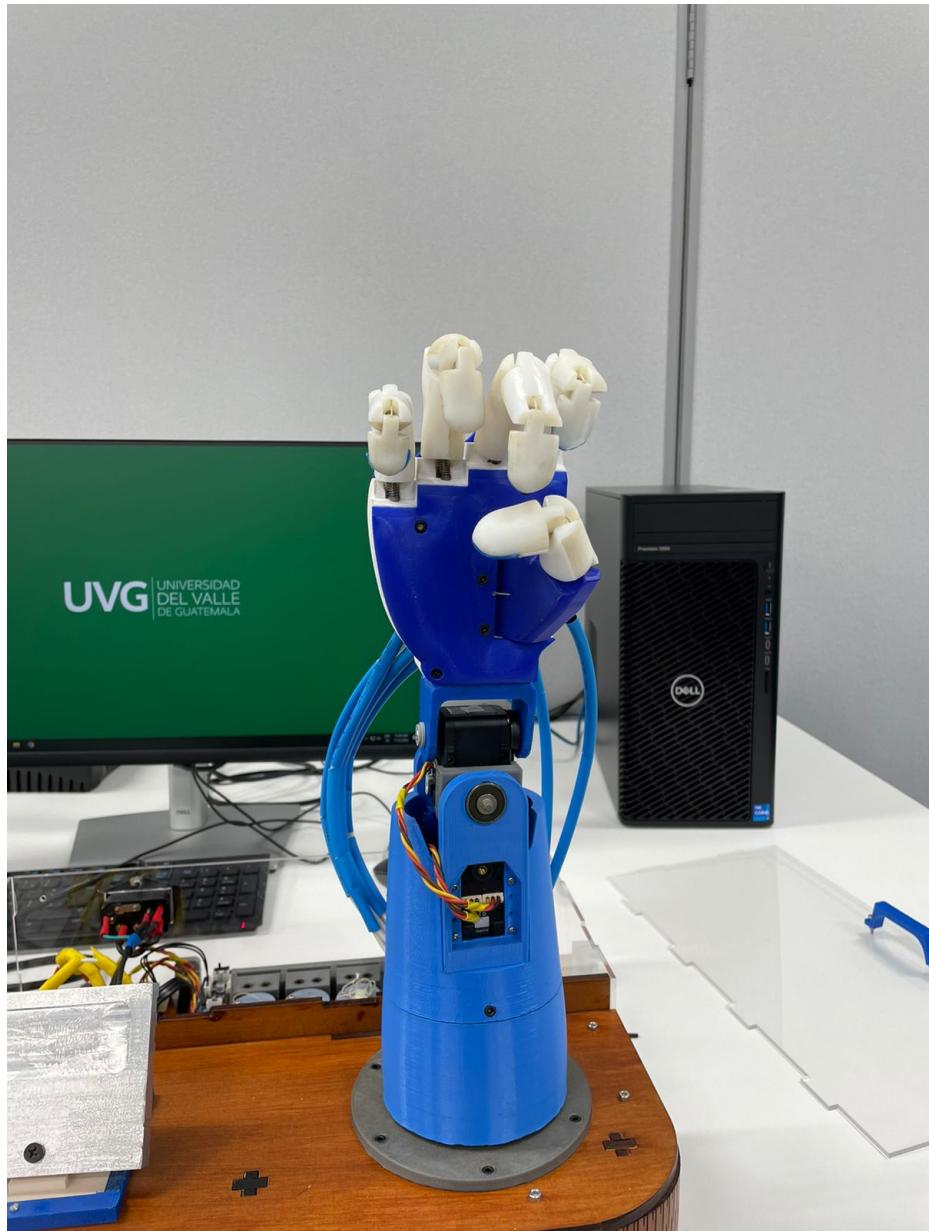


Figura 43: Conteo del número 0 en mano antropomórfica.

Movimiento 1



Figura 44: Conteo del número 1 en mano antropomórfica.

Movimiento 2



Figura 45: Conteo del número 2 en mano antropomórfica.

Movimiento 3



Figura 46: Conteo del número 3 en mano antropomórfica.

Movimiento 4



Figura 47: Conteo del número 4 en mano antropomórfica.

Movimiento 5



Figura 48: Conteo del número 5 en mano antropomórfica.

CAPÍTULO 11

Interfaz de control de mano antropomórfica

Es importante destacar que existen interfaces, realizadas en fases anteriores, capaces de realizar control mediante señales EMG, aunque se basan únicamente en bases de datos preestablecidas. Esto resalta la necesidad de continuar investigando y comprobar su funcionamiento con un enfoque en el control en tiempo real. En este capítulo se presenta la implementación de una interfaz con la capacidad de controlar actuadores en tiempo real, por medio de señales EMG.

11.1. Implementación de clasificadores

Debido a la variedad de clasificadores disponibles en la aplicación de *Classification Learner*, se decidió ofrecer al usuario diferentes opciones. En la interfaz se implementaron los algoritmos de clasificación más efectivos observados en el capítulo 8, seleccionados principalmente por sus porcentajes de validación en la etapa de prueba. Entre estos se encuentran la SVM cónica, la ANN ancha y la KNN cónica.

Si se habilitan las tres opciones, la interfaz selecciona automáticamente el clasificador con el mejor porcentaje de rendimiento durante el entrenamiento, aunque el usuario también tiene la posibilidad de elegir el clasificador de su preferencia.

Además, la aplicación permite cargar una base de datos personalizada para el entrenamiento. Al ingresar un archivo '.mat', la interfaz selecciona automáticamente una variable llamada 'BaseEntrenamiento', que se utiliza para entrenar los tres clasificadores.

Esta parte de la interfaz se puede observar en la Figura 49, estando separada en la parte de la selección del archivo y una parte para los clasificadores.

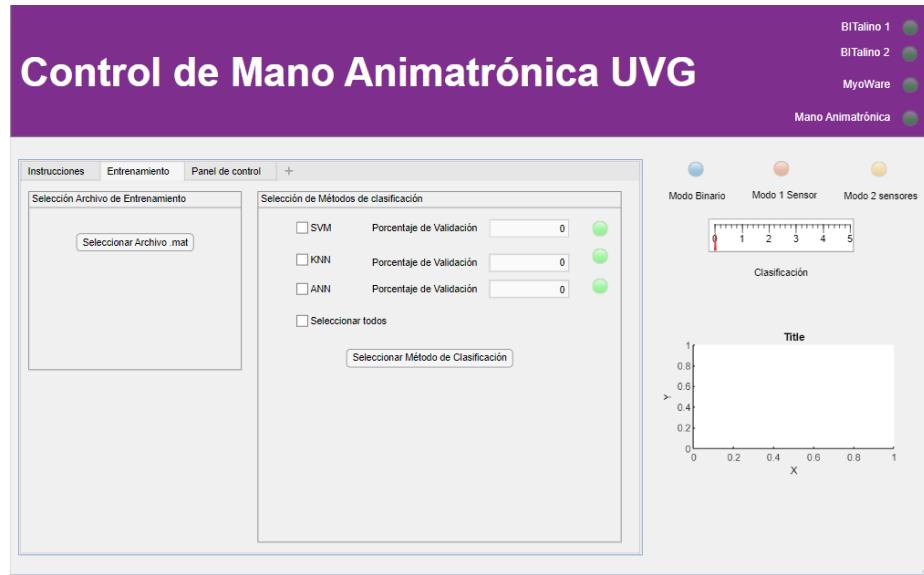


Figura 49: Interfaz de control de mano antropomórfica UVG, pestaña de entrenamiento.

11.2. Funcionalidades

Con la disponibilidad de diferentes sensores, se implementaron tres funcionalidades en la interfaz: modo binario, modo de 1 sensor y modo de 2 sensores. Estas funcionalidades permiten utilizar diversos tipos de sensores y conectarlos a la interfaz de manera efectiva.

Esta implementación se puede observar en la Figura 50, donde se muestran y habilitan las conexiones necesarias para cada modo.

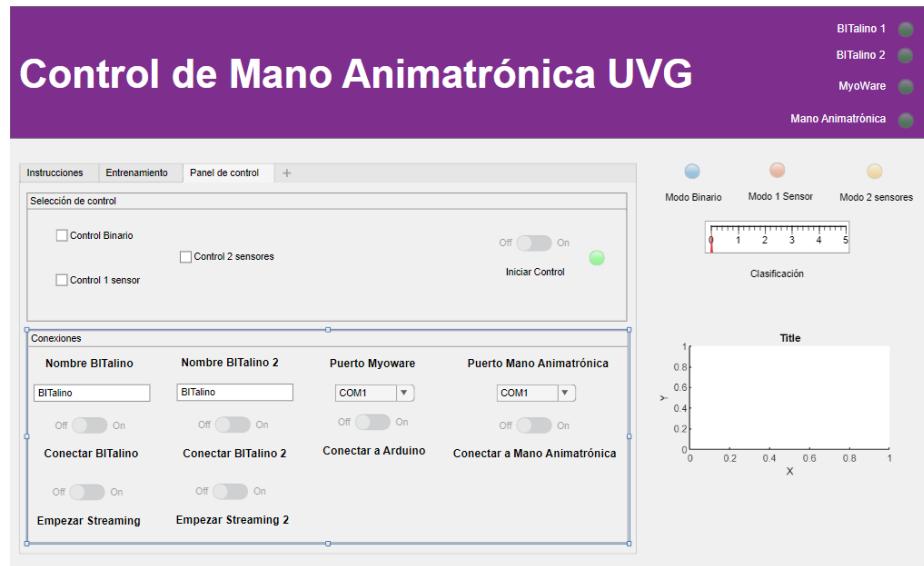


Figura 50: Interfaz de control de mano antropomórfica UVG, pestaña de control.

11.2.1. Modo Binario

Esta funcionalidad de la interfaz opera exclusivamente con el sensor MyoWare, utilizando el algoritmo de captura descrito en el capítulo 7. Al recibir una señal del sensor, la interfaz clasifica entre dos tipos de movimiento. Debido a la velocidad de los movimientos, se implementó un retardo en las funcionalidades de la interfaz para facilitar la visualización clara de los movimientos en la mano.

11.2.2. Modo 1 sensor

Similar al modo binario, este modo también se basa en la señal del sensor MyoWare. Sin embargo, agrega el uso del sensor BITalino para capturar señales EMG crudas directamente, las cuales son clasificadas mediante los algoritmos implementados en la interfaz. Posteriormente, se selecciona una posición predeterminada en el algoritmo, que luego se envía a la mano.

Este modo se diseñó con el objetivo de distinguir entre tres clases de movimientos, aunque también es funcional con un número menor o mayor de clases.

11.2.3. Modo 2 sensores

Este modo es similar al modo de 1 sensor, con la única diferencia de que admite un segundo sensor BITalino. Esta funcionalidad se implementó con el objetivo de clasificar entre 4 y 6 clases de movimiento, aprovechando la versatilidad que ofrece la incorporación de dos señales extraídas, lo que permite manejar un mayor número de variables.

CAPÍTULO 12

Conclusiones

CAPÍTULO 13

Recomendaciones

CAPÍTULO 14

Bibliografía

- [1] J. F. Delgado, E. Vallejo y J. Torres, “Diseño y Construcción de un Sistema de adquisición y Visualización de Señales Electromiográficas,” *Universidad del Norte*, 2007.
- [2] E. Muñoz, O. H. Paruma y J. F. Florez, “Aplicaciones de las Señales Mioeléctricas para el Control de Interfaces Hombre-Maquina,” *Universidad del Cauca*, 2004.
- [3] D. Capdevila, E. Spinelli y A. Veiga, “Plataforma Vestible para Interfaces Hombre-Máquina,” *Universidad Nacional de La Plata*, 2019.
- [4] M. F. Girón, “Interfaz Biomédica para el Control de Sistemas Robóticos Utilizando Señales EMG,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2020.
- [5] R. Ralda, “Interfaz Biomédica para el Control de Sistemas Robóticos Utilizando Señales EEG,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2020.
- [6] R. J. Cáceres, “Interfaz para el Control de Sistemas Robóticos Utilizando Combinaciones de Señales Biomédicas,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.
- [7] F. J. Sandoval, “Aplicación sistemática de algoritmos de aprendizaje automático y reconocimiento de patrones a señales bioeléctricas obtenidas de HUMANA y con el sistema Biopac,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2023.
- [8] H. A. Romo, J. C. Realpe y P. E. Jojoa, “Análisis de Señales EMG Superficiales y su Aplicación en Control de Prótesis de Mano,” *Avances en Sistemas e Informática*, 2007.
- [9] C. Alquézar, “Proyecto Roujin: Entorno experimental para pruebas de control con señales bioeléctricas,” Tesis de licenciatura, Universidad de Zaragoza, 2015.
- [10] G. Palarea, “Diseño y Fabricación de una Prótesis Electromecánica de Mano Humana, Controlada por Señales EMG de Superficie,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.
- [11] O. Gálvez, “Optimización de diseño de una mano y muñeca animatrónica antropomórfica de la fase tres e implementación de un control interactivo.,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2021.

- [12] M. García, “Optimización del proyecto de una mano y muñeca animatrónica antropomórfica y su sistema de control y operación.,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2022.
- [13] J. D. Bronzino, *The Biomedical Engineering Handbook*. CRC Press, 2000.
- [14] L. Sörnmo y P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. Elsevier Science, 2005.
- [15] P. A. Parker y R. Merletti, *Electromyography Physiology, Engineering, and Non-Invasive Applications*. Wiley, 2004.
- [16] E. Guzmán-Muñoz y G. Mendéz-Rebolledo, “Electromiografía en las Ciencias de la Rehabilitación.,” *Revista Salud Uninorte*, 2018.
- [17] R. Bagherian, M. Esmaeili y M. Nourani, “EMG-Based Feature Extraction and Classification for Prosthetic Hand Control,” 2021.
- [18] PLUX Biosignals, *BITalino: Learning and exploring biosignals everywhere*, <https://www.pluxbiosignals.com/pages/bitalino>, Accessed: 2024-05-11, 2024.
- [19] MyoWare, *MyoWare 2.0 Muscle Sensor*, <https://myoware.com/products/muscle-sensor/>, Accessed: 2024-05-11, 2024.
- [20] J. Adamssen, *Inteligencia Artifical: Cómo el aprendizaje automático, la robótica y la automatización han dado forma a nuestra sociedad*. Efalon Acies, 2020.
- [21] J. Muñoz, *Inteligencia Computacional Inspirada en la Vida*. SPICUM - Universidad de Málaga, 2010.
- [22] C. Pineda, *Aprendizaje automático y profundo en python*. Ediciones de la U, 2022.
- [23] E. Sorias, M. Sánchez-Montañes, R. G. anf Borja Castillo y P. Cano, *Sistemas de Aprendizaje Automático*. Ra-Ma S.A. Editorial y Publicaciones, 2023.
- [24] C. Chique, *Maquina de Soporte Vectorial*, <https://medium.com/@csarchiquerodriguez/maquina-de-soporte-vectorial-svm-92e9f1b1b1ac>, Accessed: 2024-05-12, 2020.
- [25] L. Wang, *Support Vector Machines: Theory and Applications*. Springer, 2005.
- [26] H. Rhys, *Machine Learning with R, the Tidyverse, and Mlr*. Manning, 2020.
- [27] A. Band, *Multi class Classification One vs All One-vs-One*, <https://towardsdatascience.com/multi-class-classification-one-vs-all-one-vs-one-94daed32a87b>, Accessed: 2024-07-27, 2020.
- [28] F. Sabry, *K Nearest Neighbor Algorithm: Fundamentals and Applications*. One Billion Knowledgeable, 2023.
- [29] Y. Akbulut, A. Sengur, Y. Guo y F. Smarandache, “NS-k-NN: Neutrosophic Set-Based k-Nearest Neighbors Classifier,” 2023.
- [30] G. for Geeks, *K-Nearest Neighbor Algorithm*, <https://www.geeksforgeeks.org/k-nearest-neighbours/>, Accessed: 2024-09-11, 2024.
- [31] C. Ltd., *Mastering HMI*. Cybellium Ltd., 2023.
- [32] L. Corona, G. Abarca y J. Mares, *Sensores y Actuadores*. Grupo Editorial Patria, 2014.
- [33] MedlinePlus, *Prothesis*, <https://medlineplus.gov/ency/article/002286.htm>, Accessed: 2024-05-12, 2023.

[34] J.-M. Seeber, *Reconstruyendo el cuerpo*. Teacher Created Materials, 2022.

CAPÍTULO 15

Anexos
