
Interfaz Biomédica para el Control de Sistemas Robóticos Utilizando Señales EMG

María Fernanda Girón Arevalo



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Interfaz Biomédica para el Control de Sistemas Robóticos
Utilizando Señales EMG**

Trabajo de graduación presentado por María Fernanda Girón Arevalo
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2020

Vo.Bo.:

(f) _____
Dr. Luis Alberto Rivera Estrada

Tribunal Examinador:

(f) _____
Dr. Luis Alberto Rivera Estrada

(f) _____

(f) _____

Fecha de aprobación: Guatemala, de de .

Prefacio

Esta investigación inició debido a mi interés en la ingeniería biomédica y a la oportunidad que surgió en la Universidad de Valle de pertenecer a esta línea de investigación. El interés de participar en el desarrollo de una interfaz biomédica para el control de dispositivos robóticos usando señales electromiográficas, surgió principalmente porque se espera que esta investigación contribuya en un futuro al desarrollo de herramientas que permitan el control de dispositivos de asistencia médica. Por lo que, a través de este trabajo puedo aportar al crecimiento del área científica en Guatemala y retribuir un poco a la sociedad.

Agradezco a Dios, por brindarme sabiduría, guía y esperanza en los momentos en que no creía en mi misma. A mi familia, es especial a mi madre Rosalva y a mi abuela Margoth por brindarme su apoyo incondicional. Por creer en mí, confiar en mis capacidades, ser mi soporte y motivarme siempre a luchar por mis metas. A mis hermanos Alba, Majo y Daniel por apoyarme y escuchar mis logros y frustraciones. A mis amigos y compañeros por acompañarme en este camino de aprendizaje y crecimiento. A la fundación Juan Bautista Gutiérrez por el privilegio que me brindaron de pertenecer a esta familia.

Finalmente, a la comunidad de la Universidad del Valle de Guatemala y a mi asesor Dr. Luis Albero Rivera por su dedicación, sus consejos académicos y sus críticas tan certeras que me guiaron a culminar este trabajo de graduación.

Índice

Prefacio	III
Lista de figuras	IX
Lista de cuadros	X
Resumen	XI
Abstract	XII
1. Introducción	1
2. Antecedentes	2
3. Justificación	4
4. Objetivos	6
5. Alcance	7
6. Marco teórico	8
6.1. Señales Electromiográficas EMG	8
6.2. Electrodos	9
6.2.1. Localización de los Electrodos	10
6.3. Características de las Señales Electromiográficas	10
6.3.1. Características en el Dominio del Tiempo	11
6.3.2. Características en el Dominio de la Frecuencia	12
6.3.3. Características en el Dominio de Tiempo-Frecuencia	13
6.4. Aprendizaje Automático	13
6.4.1. Validación Cruzada	14
6.4.2. Máquinas de Vectores de Soporte	15
6.4.3. Redes Neuronales	16
6.5. Dispositivos Robóticos	17
6.5.1. Manipulador Serial	17

7. Dispositivos para Adquisición de Señales sEMG	19
7.1. Familiarización con el Bitalino y <i>OpenSignals</i>	19
8. Adquisición de Señales sEMG	22
8.1. Desarrollo de Algoritmo para Lectura de Datos	22
8.2. Procesamiento de Señales	24
8.2.1. <i>Signal Processing Toolbox</i>	25
8.3. Desarrollo de Algoritmo de Identificación de Actividad	27
9. Base de Datos de Señales sEMG	29
9.1. Base de Datos Pública	29
9.2. Generación de Base de Datos Propia	31
10. Clasificación de Movimientos/Gestos	34
10.1. Extracción de Características	34
10.2. Entrenamiento de Clasificadores	36
10.3. Resultados Base de Datos Pública	38
10.4. Resultados Base de Datos Propia	52
10.5. Resultados Finales	61
11. Desarrollo de Interfaz para el Control de Dispositivos Robóticos	63
11.1. <i>Robotics Toolbox</i>	63
12. Simulaciones Finales	65
12.1. Simulación Usando la Base de Datos Pública	66
12.2. Simulación con Datos Obtenidos en Tiempo Real	69
13. Conclusiones	74
14. Recomendaciones	75
15. Bibliografía	76
16. Anexos	78
16.1. Resultados Obtenidos con la Base de Datos Pública	78
17. Glosario	98

Lista de figuras

1.	Señal miográfica normal y con patologías neuromusculares [6].	8
2.	División de datos en conjuntos de entrenamiento, validación y prueba.	14
3.	Proceso de validación cruzada [12].	15
4.	Datos no separables linealmente [15].	16
5.	Separación de clases mediante la transformación de datos de entrada a un espacio de mayor dimensión [16].	16
6.	Estructura de un sistema de redes neuronales [13].	17
7.	Manipulador serial [19].	18
8.	Conexión Bitalino y <i>OpenSignals</i>	20
9.	Señal electromiográfica de prueba visualizada con <i>OpenSignals</i>	20
10.	Señal electromiográfica de prueba visualizada con <i>OpenSignals</i>	21
11.	Ubicación de los electrodos para adquisición de señales sEMG.	21
12.	Algoritmo de adquisición de datos Matlab.	23
13.	Algoritmo de adquisición de datos Arduino.	24
14.	Diseño filtro pasa banda tipo <i>Butterworth</i>	25
15.	Diseño filtro rechaza banda tipo <i>Butterworth</i>	26
16.	Filtrado señal de prueba.	26
17.	Ventana completa que incluye la señal sin movimiento más la señal con movimiento.	27
18.	Señal sin movimiento y detección de umbral.	28
19.	Señal con movimiento.	28
20.	Tipos de agarre incluídos en la base de datos.	30
21.	Información contenida en la base de datos pública.	31
22.	Movimientos incluídos en la base de datos: 1) empuñar, 2) arriba, 3) doble arriba y 4) movimiento de dedos.	32
23.	Información contenida en la base de datos propia correspondiente a 4 clases. .	32
24.	Visualización de señales correspondientes a cada movimiento/clase.	33
25.	Vectores de características para la base de datos pública utilizando MAV y ZC. .	36
26.	Proceso para entrenamiento de clasificadores.	36

27. Ejemplo de matriz de características utilizado para entrenamiento con redes neuronales.	37
28. Ejemplo de matriz de etiquetas para entrenamiento con redes neuronales.	38
29. Resultados obtenidos con SVM lineal para clasificación de 6 clases, 2 características.	39
30. Resultados obtenidos con SVM lineal para clasificación de 6 clases, 3 características.	39
31. Resultados obtenidos con SVM lineal para clasificación de 6 clases, 5 características.	40
32. Resultados obtenidos con SVM lineal para clasificación de 6 clases, 7 características.	40
33. Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 2 características.	41
34. Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 3 características.	41
35. Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 5 características.	42
36. Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 7 características.	42
37. Resultados obtenidos con RN para clasificación de 6 clases, 2 características. Sujeto 1: <i>female_1.mat</i>	43
38. Resultados obtenidos con RN para clasificación de 6 clases, 2 características. Sujeto 4: <i>male_1.mat</i>	44
39. Resultados obtenidos con RN para clasificación de 6 clases, 3 características. Sujeto 1: <i>female_1.mat</i>	45
40. Resultados obtenidos con RN para clasificación de 6 clases, 3 características. Sujeto 4: <i>male_1.mat</i>	46
41. Resultados obtenidos con RN para clasificación de 6 clases, 5 características. Sujeto 1: <i>female_1.mat</i>	47
42. Resultados obtenidos con RN para clasificación de 6 clases, 5 características. Sujeto 4: <i>male_1.mat</i>	48
43. Resultados obtenidos con RN para clasificación de 6 clases, 7 características. Sujeto 1: <i>female_1.mat</i>	49
44. Resultados obtenidos con RN para clasificación de 6 clases, 7 características. Sujeto 4: <i>male_1.mat</i>	50
45. Comparación de resultados entre clasificadores con 2 características.	51
46. Comparación de resultados entre clasificadores con 3 características.	51
47. Comparación de resultados entre clasificadores con 5 características.	51
48. Comparación de resultados entre clasificadores con 7 características.	52
49. Comparación de resultados de entrenamiento entre distintos tipos de clasificadores para 4 clases.	52
50. Resultados obtenidos con datos de prueba satisfactorios.	54
51. Resultados con RN para grabación ‘datos_test_1’ con 5 características.	54
52. Resultados con SVM para grabación ‘datos_test_1’ con 5 características.	55
53. Resultados obtenidos con datos de prueba no satisfactorios.	55
54. Resultados con RN para grabación ‘datos_test_4’ con 5 características.	56
55. Resultados con SVM para grabación ‘datos_test_4’ con 5 características.	56

56.	Resultados obtenidos con datos de prueba satisfactorios, dividiendo las señales en dos secciones.	57
57.	Resultados con RN para grabación ‘datos_test_1’ con 5 características dividiendo las señales.	57
58.	Resultados con SVM para grabación ‘datos_test_1’ con 5 características dividiendo las señales.	58
59.	Resultados obtenidos con datos de prueba no satisfactorios, dividiendo las señales en dos secciones.	58
60.	Resultados con RN para grabación ‘datos_test_4’ con 5 características dividiendo las señales.	59
61.	Resultados con SVM para grabación ‘datos_test_4’ con 5 características dividiendo las señales.	59
62.	Comparación entre métodos utilizando redes neuronales.	60
63.	Resultados con RN para clasificación de 4 clases con 4 características (MAV,ZC,WL y RMS).	62
64.	Manipulador serial R17 de 5 GDL más carrera [27].	63
65.	Código para definir el manipulador R17 como objeto <i>SerialLink</i>	64
66.	Versión interactiva del manipulador R17.	64
67.	Funcionamiento del sistema final.	65
68.	Interfaz en modo manual.	66
69.	Interfaz en modo automático.	67
70.	Clase 1: Junta 4 gira 45°.	67
71.	Clase 2: Junta 3 y junta 4 giran 90°.	68
72.	Clase 3: Junta 3 gira 90°.	68
73.	Clase 4: Junta 3 gira -90°.	68
74.	Clase 5: Junta 4 gira 90°.	69
75.	Clase 6: Junta 4 gira -90°.	69
76.	Interfaz en modo manual.	70
77.	Interfaz en modo automático.	70
78.	Clase 1: Junta 4 gira 45°.	71
79.	Clase 2: Junta 3 y junta 4 giran 90°.	72
80.	Clase 3: Junta 3 gira 90°.	72
81.	Clase 4: Junta 4 gira 90°.	73
82.	Resultados obtenidos con SVM lineal para clasificación de 6 clases, 2 características.	78
83.	Resultados obtenidos con SVM lineal para clasificación de 6 clases, 3 características.	79
84.	Resultados obtenidos con SVM lineal para clasificación de 6 clases, 5 características.	80
85.	Resultados obtenidos con SVM lineal para clasificación de 6 clases, 7 características.	81
86.	Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 2 características.	82
87.	Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 3 características.	83

88.	Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 5 características.	84
89.	Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 7 características.	85
90.	Resultados obtenidos con RN para clasificación de 6 clases, 2 características. Sujeto 2: <i>female_2.mat</i>	86
91.	Resultados obtenidos con RN para clasificación de 6 clases, 2 características. Sujeto 3: <i>female_3.mat</i>	87
92.	Resultados obtenidos con RN para clasificación de 6 clases, 2 características. Sujeto 5: <i>male_2.mat</i>	88
93.	Resultados obtenidos con RN para clasificación de 6 clases, 3 características. Sujeto 2: <i>female_2.mat</i>	89
94.	Resultados obtenidos con RN para clasificación de 6 clases, 3 características. Sujeto 3: <i>female_3.mat</i>	90
95.	Resultados obtenidos con RN para clasificación de 6 clases, 3 características. Sujeto 5: <i>male_2.mat</i>	91
96.	Resultados obtenidos con RN para clasificación de 6 clases, 5 características. Sujeto 2: <i>female_2.mat</i>	92
97.	Resultados obtenidos con RN para clasificación de 6 clases, 5 características. Sujeto 3: <i>female_3.mat</i>	93
98.	Resultados obtenidos con RN para clasificación de 6 clases, 5 características. Sujeto 5: <i>male_2.mat</i>	94
99.	Resultados obtenidos con RN para clasificación de 6 clases, 7 características. Sujeto 2: <i>female_2.mat</i>	95
100.	Resultados obtenidos con RN para clasificación de 6 clases, 7 características. Sujeto 3: <i>female_3.mat</i>	96
101.	Resultados obtenidos con RN para clasificación de 6 clases, 7 características. Sujeto 5: <i>male_2.mat</i>	97

Lista de cuadros

1. Descripción de las grabaciones almacenadas en la base de datos pública. 30
2. Combinación de características utilizadas para pruebas de entrenamiento de clasificadores. 53
3. Descripción de las grabaciones utilizadas para evaluar los clasificadores. 53
4. Comparación de resultados de entrenamiento con diferentes combinaciones y cantidad de características temporales. 61

Resumen

Este trabajo consistió en el desarrollo de una interfaz biomédica para el control de dispositivos robóticos mediante la adquisición, filtrado y análisis de señales electromiográficas de superficie (sEMG). Con el fin de separar las señales sEMG según su clase para el control de dichos dispositivos, se implementó un algoritmo de clasificación mediante el entrenamiento de una máquina de vectores de soporte (SVM) y de una red neuronal (RN), utilizando vectores de características extraídos de la señal. Al haber obtenido un modelo de clasificación que presenta un alto rendimiento, se hizo uso de la simulación de un sistema robótico y un algoritmo capaz de traducir los resultados del clasificador en comandos, para visualizar la manipulación del sistema según la señal sEMG adquirida.

Los mejores resultados de entrenamiento de clasificadores se obtuvieron a partir del uso de redes neuronales, con rendimientos superiores al 90 % al clasificar cuatro movimientos/gestos, resultados que superan los presentados por las máquinas de vectores de soporte. Además, se demostró que el uso de características únicamente en el dominio del tiempo es suficiente para generar clasificadores que presenten porcentajes de rendimiento mayores a 80 %.

En un futuro se espera que la implementación de esta interfaz biomédica contribuya al desarrollo de nuevas metodologías que promuevan el aprendizaje enfocado en *Machine Learning* y al desarrollo de futuras investigaciones sobre el control de dispositivos de asistencia médica utilizando señales bioeléctricas.

Abstract

This work consisted of the development of a biomedical interface for the control of robotic devices by acquiring, filtering and analyzing surface electromyographic signals (sEMG). In order to classify sEMG signals according to their class for the control of such devices, a classification algorithm was implemented by training a support vector machine (SVM) and a neural network (RN), using feature vectors extracted from the signal. After obtaining a high performing classification model, a robotic system simulator was developed. An algorithm capable of translating the results of the classifier into commands was coupled with the simulator to visualize the manipulation of the system according to the acquired sEMG signal.

In terms of classification accuracy the Neuronal Networks outperformed the SVM. Accuracies greater than 90 % were achieved when classifying four different gestures. The results showed that using time domain features alone can yield classification accuracies greater than 80 %.

In the future, the implementation of this biomedical interface may contribute to the development of new methodologies that promote the use of Machine Learning and the development of future research on the control of medical assistance devices using bioelectric signals.

CAPÍTULO 1

Introducción

El uso de señales bioeléctricas ha contribuido significativamente al desarrollo de dispositivos de asistencia médica, cuyo principal fin es mejorar el bienestar humano tanto físico como psicológico, de personas con discapacidades motrices. Las señales electromiográficas de superficie forman parte importante en estos estudios, dado que permiten adquirir información de la actividad muscular de manera no invasiva.

A través de esta investigación se plantea el desarrollo de una interfaz biomédica que permita el control de dispositivos robóticos disponibles en la Universidad del Valle de Guatemala, mediante la adquisición y filtrado en tiempo real de señales electromiográficas de superficie. También se espera que este trabajo contribuya a fomentar el aprendizaje enfocado en técnicas de *Machine Learning*, al desarrollar algoritmos para clasificación de movimientos basados en reconocimiento de patrones y extracción de características.

Esta investigación se desarrolló en una serie de etapas principales que se detallan en este documento. Se presenta información sobre la etapa de adquisición y procesamiento de señales EMG en donde se expone el proceso de filtrado de señales en tiempo real y se explica el funcionamiento del algoritmo de adquisición y detección de actividad. Posteriormente se presenta información sobre las bases de datos utilizadas para entrenamiento de clasificadores. Más adelante, se detalla el proceso de extracción de características y entrenamiento de clasificadores, así como los resultados obtenidos después de una serie de pruebas de entrenamiento. La última etapa de esta investigación expone el proceso de desarrollo de la interfaz gráfica, complementado con el desarrollo de simulaciones que integran las etapas expuestas anteriormente.

Por otra parte, este documento tiene como objetivo sentar las bases y fundamentos para la continuación y desarrollo de nuevas fases para este tipo de proyectos. Y se espera que en un futuro los resultados de esta investigación puedan contribuir a la comunidad científica de la Universidad del Valle y al desarrollo de dispositivos de asistencia médica, controlados mediante señales bioeléctricas.

CAPÍTULO 2

Antecedentes

El estudio de señales bioeléctricas ha contribuido al avance de distintos proyectos que buscan mejorar el bienestar humano. En particular, las señales electromiográficas de superficie (EMG) representan una herramienta muy importante y muy utilizada debido a que se pueden obtener fácilmente de forma no invasiva y sin necesidad de supervisión médica [1]. El objetivo de estudio de estas señales radica principalmente en el desarrollo de metodologías que permitan el control de dispositivos de rehabilitación, como se realizó en [2] y [3], en donde se implementa una prótesis de mano cuyos movimientos se controlan mediante señales EMG. Así como el desarrollo de algoritmos de reconocimiento de patrones en aplicaciones en tiempo real, como en [4] donde mediante el reconocimiento de patrones en señales EMG se identifican seis tipos de movimientos de la mano, entre otros.

Las señales electromiográficas proveen información acerca de la actividad neuromuscular que se produce en la fibra muscular durante la contracción o relajación de un músculo. Esta información es interpretada mediante la extracción de características en el dominio del tiempo y frecuencia. En investigaciones anteriores se ha estudiado el proceso de extracción de características con el fin de determinar cuáles son más relevantes y efectivas para el reconocimiento de patrones en señales EMG. En el caso de investigaciones como la realizada en [5], se establece que la extracción de características temporales utiliza un menor tiempo de procesamiento que las características en el dominio de la frecuencia y las de tiempo-frecuencia. También se recomienda el uso de características en tiempo-frecuencia sobre las que se basan únicamente en el dominio de la frecuencia, ya que estas últimas presentan una limitación al trabajar con señales no estacionarias, cuyas propiedades cambian a lo largo del tiempo como lo son las señales EMG.

Finalmente, la ubicación de los electrodos para la extracción de señales electromiográficas también se ha estudiado con el fin de determinar la mejor ubicación según el tipo de movimiento a estudiar. Como en [2], donde se realiza un estudio de las formas correctas de ubicación de electrodos para reducir el ruido que registran los electrodos de superficie y poder obtener registros más selectivos.

Cabe mencionar que en la universidad del Valle de Guatemala se cuenta con una línea

de investigación en el área de biomédica, en donde ya se han realizado estudios sobre señales bioeléctricas. Sin embargo, no se ha realizado una investigación enfocada específicamente en el estudio y procesamiento de señales electromiográficas para futuras aplicaciones.

CAPÍTULO 3

Justificación

En la comunidad científica se puede encontrar una variedad de investigaciones relacionadas con el estudio de señales biomédicas, realizadas con el fin de mejorar el bienestar humano. El enfoque principal que toman estas investigaciones se basa en implementar metodologías para el control de dispositivos como prótesis. Tomando como apoyo la información recolectada en estas investigaciones, se planteó el desarrollo de una interfaz biomédica que, mediante la interpretación de señales electromiográficas, permita el control de sistemas robóticos con los que cuenta la Universidad del Valle de Guatemala.

A través de este proyecto se busca fomentar el aprendizaje enfocado en *Machine Learning* para desarrollar distintas técnicas de reconocimiento de patrones. La importancia de impulsar esta disciplina radica principalmente en que facilita el manejo de grandes series de datos. Ya que estas técnicas permiten la creación de sistemas que mediante la extracción de información significativa y el análisis sistemático de datos, descubren comportamientos y establecen patrones, dando paso al desarrollo de modelos predictivos que posibilitan reaccionar a situaciones futuras y llegar a la resolución rápida de problemas complejos.

El enfoque principal del proyecto es implementar una herramienta que permita el control de robots y dispositivos disponibles en la Universidad del Valle, mediante la adquisición de señales EMG y métodos de clasificación basados en reconocimiento de patrones. Ejemplos de estos dispositivos son el brazo robótico articulado R17 o el robot humanoide NAO.

Es importante destacar que en un futuro los resultados de este proyecto pueden contribuir en el desarrollo de herramientas que permitan el control de sistemas de asistencia médica. Por ejemplo, el accionamiento y manejo de prótesis mediante la actividad eléctrica generada en un músculo específico, como se menciona en [2] y [5]. La gran ventaja que presenta el control mioeléctrico tanto en prótesis, como en otro tipo de dispositivos, radica en que ofrece un control automático. Es decir, que no se requiere la activación o control manual mediante interruptores como ocurre con dispositivos de asistencia o prótesis que requieren control mecánico. Por otra parte, la adquisición de señales de manera no invasiva mediante electrodos de superficie, contribuye a la comodidad de los usuarios que controlan dichos dispositivos. Lo que en conjunto ayuda a mejorar la calidad de vida de las personas que

requieran de dispositivos de asistencia médica.

CAPÍTULO 4

Objetivos

Objetivo General

Diseñar e implementar una interfaz para el manejo de sistemas robóticos usando señales electromiográficas.

Objetivos Específicos

- Establecer un mecanismo de adquisición y filtrado en tiempo real de señales electromiográficas (EMG) de superficie.
- Seleccionar características adecuadas para el reconocimiento de patrones en señales EMG.
- Evaluar e implementar algoritmos para la clasificación de movimientos o gestos, a partir de características y patrones de señales EMG.
- Desarrollar una interfaz que traduzca los resultados de la clasificación de señales en comandos para un sistema robótico.

CAPÍTULO 5

Alcance

Este proyecto forma parte de una fase inicial de investigación, que tiene como fin el desarrollo de una interfaz biomédica que mediante la adquisición y procesamiento de señales sEMG permita el control de sistemas robóticos. En específico, sistemas robóticos con los que cuenta la Universidad del Valle de Guatemala, como el brazo robótico articulado R17.

El desarrollo de este proyecto se divide en cuatro etapas principales. La primera consiste en el desarrollo de un algoritmo de adquisición y filtrado en tiempo real de señales electromiográficas de superficie, en esta etapa se implementa también un algoritmo de detección de actividad. La segunda etapa consiste en la extracción y selección de características adecuadas que permitan un uso eficiente de algoritmos de identificación de patrones. Derivado de esta etapa, la tercera utiliza como base el aprendizaje automático para la implementación de dos tipos de clasificadores, máquinas de vectores de soporte y redes neuronales, que permitan diferenciar distintos movimientos/gestos especificados con anterioridad, a partir de características y patrones. Por último, la cuarta etapa tiene como fin la creación de una interfaz, en la cual los resultados de la clasificación se traduzcan a comandos de control para el sistema robótico.

Con el desarrollo de este proyecto se espera establecer las bases para futuros proyectos o fases enfocadas en el uso de señales bioeléctricas para el control de sistemas robóticos físicos. Dado que, como resultado de la pandemia de coronavirus, el alcance del proyecto se vio limitado al control de dispositivos robóticos únicamente mediante simulaciones.

CAPÍTULO 6

Marco teórico

6.1. Señales Electromiográficas EMG

Este tipo de señales biomédicas representan el campo de potencial eléctrico que se crea por la despolarización de la membrana de fibra muscular externa.

Estas señales bioeléctricas se generan debido a la actividad eléctrica que se produce en la fibra muscular durante la contracción o relajación del músculo. Siendo de suma utilidad para la detección de patologías relacionadas con la actividad neuromuscular y enfermedades como la distrofia muscular, inflamación de los músculos, daños en los nervios de las manos y pies, entre otras.

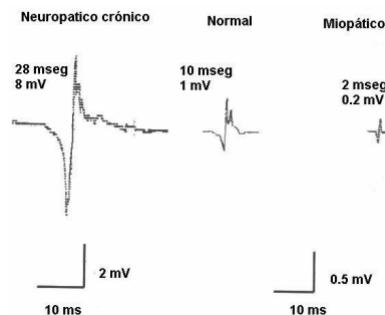


Figura 1: Señal miográfica normal y con patologías neuromusculares [6].

Para detectar este tipo de señales usualmente se utilizan electrodos intramusculares o de superficie. Estos se colocan a cierta distancia de las fuentes que generan la señal. Estas fuentes se ubican en las zonas despolarizadas de las fibras musculares. Al tejido biológico que separa estas zonas de los electrodos se le llama conductor de volumen y las propiedades del mismo son de gran importancia para determinar ciertas características de la señal EMG detectada. Entre ellas el contenido de frecuencia y la distancia a la cual la señal deja de detectarse [7].

Estas señales producidas por las neuronas motoras al activar los músculos del cuerpo son de naturaleza aleatoria y sus voltajes se encuentran usualmente entre 0 y 6 mV. Con frecuencias entre 0 a 500 HZ, destacando en el rango de 50 a 150 Hz las señales de mayor intensidad [6].

La medición de estas señales se puede realizar de manera superficial o intramuscular. De manera superficial los electrodos se colocan sobre la superficie de la piel que recubre el músculo de interés. Mientras que la manera intramuscular implica insertar electrodos de aguja a través de la piel en el músculo que se desea examinar. Además, existen cinco tipos de electromiografía basadas en el tipo de electrodo e instrumento que se utiliza, estos tipos son:

- Electromiografía Convencional
- Electromiografía Cuantitativa
- Electromiografía de Fibra Simple
- Macro EMG
- EMG de superficie y estudio del espectro de frecuencias

Entre las mediciones importantes de una electromiografía destacan los potenciales de unidad motora (PUM). Ya que la amplitud y duración de estos potenciales varía según la actividad eléctrica muscular. Por lo que, al presentarse alguna enfermedad muscular, suelen presentarse anomalías en los PUM aunque el número de unidades motoras permanezca normal [8].

6.2. Electrodo

Los electrodos son un instrumento que permite transformar las corrientes iónicas del cuerpo humano en corrientes eléctricas, por lo que en electromiografía se utilizan para detectar la actividad eléctrica en la fibra muscular. Los electrodos se componen de una superficie metálica y un electrolito, el cual entra en contacto con la piel. Por lo que la señal bioeléctrica transita por dos interfaces, una es la que relaciona el electrolito y la parte mecánica del electrodo y la otra corresponde al contacto entre la piel y el electrolito [2].

Los electrodos más utilizados en electromiografía son los electrodos de aguja y los de superficie. La ventaja de los electrodos de aguja, usada en electromiografía invasiva, es que permiten registrar el potencial eléctrico directamente de las unidades motoras, por lo que se minimiza la presencia de ruido en la señal. Con este tipo de electrodos la amplitud de la señal depende tanto del área que ocupan, como de la distancia entre el electrodo y la fuente de la señal. La desventaja de los electrodos de aguja radica en que, al ser insertados directamente en el músculo, se requiere de supervisión médica y el proceso de inserción resulta doloroso.

Por otra parte, los electrodos de superficie son más comunes debido a que se colocan en contacto con la piel se forma rápida y sencilla, además de ser más cómodos de usar

y no requerir supervisión médica. Estos electrodos registran como actividad eléctrica una superposición de señales provenientes de varias unidades motoras presentes en músculos superficiales, por lo que presentan dificultades para detectar señales de forma selectiva en músculos pequeños. Otra desventaja de este tipo de electrodos es el alto nivel de ruido que presentan los registros obtenidos [5].

6.2.1. Localización de los Electrodos

Las características de una señal electromiográfica, como su amplitud y ancho de banda, dependen en parte del tipo, tamaño, orientación y espaciamiento entre electrodos. Con el fin de normalizar estos factores, en 1996 surgió la iniciativa SENIAM (*Surface Electromyography for Noninvasive Assessment of Muscles*), que presenta una técnica popular para la estandarización de características de los electrodos.

Tamaño de los electrodos: La SENIAM define el tamaño de un electrodo como el tamaño de su zona conductora. Este debe ser lo suficientemente grande para detectar un número considerable de unidades motoras y a la vez lo suficientemente pequeño para evitar la diafonía de otros músculos. Por lo que se recomienda un tamaño máximo de 10mm en el sentido de las fibras.

Distancia entre electrodos: El espaciamiento entre electrodos es de gran importancia para determinar el volumen de registro o recepción del tejido, por lo que menores distancias resultan en registros más selectivos. Este espaciamiento, conocido como distancia interelectrodo, se define como la distancia entre centros de las áreas conductivas de los electrodos.

La normativa SENIAM recomienda que para electrodos bipolares la distancia interelectrodo sea entre 20 y 30mm. En caso de que los electrodos se ubiquen sobre músculos relativamente pequeños, el espaciamiento no debe superar 1/4 de la longitud de la fibra muscular.

Posición de los electrodos: Es importante colocar los electrodos en una ubicación estable donde se pueda obtener una señal EMG de calidad, la ubicación se puede realizar de manera longitudinal o transversal [2].

- Longitudinal: El electrodo se ubica en la zona media del músculo, es decir, entre la terminación de la neurona motora que envía el impulso eléctrico y el tendón distal.
- Transversal: El electrodo se ubica en la zona media del músculo, de forma que la línea que uno los electrodos sea paralela al eje longitudinal del músculo.

6.3. Características de las Señales Electromiográficas

Existen diferentes técnicas para la extracción de características de señales EMG, de esta forma se puede obtener información relevante e identificar patrones en las señales. Se pueden obtener características mediante un análisis en el dominio del tiempo, en el dominio de la frecuencia y en el dominio de tiempo-frecuencia [5].

6.3.1. Características en el Dominio del Tiempo

En este tipo de análisis temporal la información es extraída en cada segmento de tiempo y no se requiere de ninguna transformación anticipada de los datos. Razón por la cual es uno de los métodos más utilizados en aplicación que requieren análisis de datos en tiempo real[9].

Entre las características más importantes en el dominio del tiempo se encuentran las siguientes:

Valor Medio Absoluto (MAV)

Estima el promedio del valor absoluto en un conjunto de datos obtenidos en un tiempo determinado. El valor N representa el tamaño del segmento y x_i el valor de cada elemento del segmento.

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

Valor Integrado de EMG (IEMG)

Tiene como fin integrar la señal mediante la sumatoria de cada elemento en un segmento de tiempo específico.

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

Cruces por Cero (ZC)

Este parámetro expresa el número de veces que la señal pasa por cero en un segmento dado. Usualmente se incluye un umbral de amplitud para evitar contar como cruces por cero el ruido de la señal. Este parámetro representa un forma simple de estimar la frecuencia de la señal.

Raíz Media Cuadrada (RMS)

Esta característica modela a la señal electromiográfica como un proceso aleatorio Gaussiano, en donde el valor de la raíz media cuadrada se relaciona a la fuerza aplicada bajo condiciones de no fatiga.

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

Desviación Estándar (STD)

Este parámetro se utiliza para cuantificar la variación o dispersión de un grupo de datos, representando qué tan separados están los mismos respecto a la media de la señal (\bar{x}).

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

Varianza (VAR)

Este parámetro al igual que la desviación estándar otorga información acerca de la dispersión y variabilidad de una serie de datos respecto a la media (\bar{x}).

$$VAR = \frac{1}{N} \sum_{i=1}^{N-1} (x_i - \bar{x})^2 \quad (5)$$

Longitud de Forma de Onda (WL)

Esta característica hace referencia a la longitud de onda de la señal EMG acumulada en un segmento de tiempo. Su importancia radica en que ofrece un estimado de la media de la amplitud, frecuencia y duración de la señal.

$$WL = \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (6)$$

6.3.2. Características en el Dominio de la Frecuencia

Debido a la naturaleza no estacionaria de las señales electromiográficas las características en el dominio de la frecuencia no son las más recomendadas para este estudio. Ya que se requiere de alguna transformación para realizar un análisis espectral, lo que origina perdidas de información de la señal en el dominio del tiempo, limitando la capacidad de detectar cuando un nuevo evento se lleva a cabo [5].

Sin embargo, durante cierto tipo de contracciones de bajo nivel las señales electromiográficas pueden ser consideradas como estacionarias por un corto período de tiempo. Por lo que se pueden considerar las siguientes características en el dominio frecuencial [10].

Frecuencia Mediana (FMD)

En donde PSD es el espectro de densidad de potencia y M la longitud del PSD.

$$FMD = \frac{1}{2} \sum_{i=1}^M PSD_i \quad (7)$$

Frecuencia Media (FMN)

$$FMN = \frac{\sum_{i=1}^M f_i PSD_i}{\sum_{i=1}^M PSD_i} \quad (8)$$

$$f_i = \frac{(i * \text{samplingrate})}{(2 * M)} \quad (9)$$

6.3.3. Características en el Dominio de Tiempo-Frecuencia

Es importante tener una herramienta que proporcione información de la señal en el dominio de tiempo-frecuencia, debido a las limitaciones del análisis espectral para señales estacionarias como las señales EMG. Estas características permiten realizar un análisis más fiable y completo. Sin embargo, debido a su complejidad matemática, presentan la desventaja de requerir un procesamiento más avanzado y más tardado [5].

Frecuencia Media Instantánea (MNIF)

Es una medida de la frecuencia media de la señal, que combina la frecuencia media ponderada (WMNF) con información de las amplitudes de la señal.

$$MNIF = \frac{\sum_{i=1}^n \|a_i\| WMIF(i)}{\sum_{i=1}^n \|a_i\|} \quad (10)$$

En donde:

$$WMNF(i) = \frac{\sum_{j=1}^N f_i(j) a_i^2(j)}{\sum_{j=1}^N a_i^2(j)} \quad (11)$$

6.4. Aprendizaje Automático

El aprendizaje automático (*Machine Learning*) es una rama de la inteligencia artificial que permite a las computadoras aprender y cambiar su comportamiento de manera autónoma basándose en su experiencia. Las técnicas de aprendizaje automatizado se emplean principalmente para clasificar y predecir datos, en base a sus características distintivas, en

diferentes tipos de aplicaciones. Entre los tipos de clasificadores más comunes se encuentran las máquinas de vectores de soporte (SVM) y las redes neuronales artificiales (RNA).

Para comparar entre distintos clasificadores es necesario llevar a cabo una evaluación de modelos, en donde se evalúe el desempeño y se estime como se comportará el clasificador al ponerlo en marcha. La evaluación final de un modelo nunca debe hacerse utilizando los datos que sirvieron para entrenamiento, por lo que se debe dividir el conjunto de datos disponibles en tres grupos:

- **Datos de entrenamiento:** Es la porción de datos que se utiliza para entrenar y construir el modelo.
- **Datos de validación:** Esta porción de datos se utiliza para optimizar parámetros y validar el modelo.
- **Datos de prueba:** Esta porción de datos se utiliza para evaluar el clasificador y medir su efectividad [11].



Figura 2: División de datos en conjuntos de entrenamiento, validación y prueba.

Existen varios métodos para evaluar modelos a parte del mencionado anteriormente en donde los datos se dividen en tres grupos principales. Uno de las técnicas más comunes es el de validación cruzada.

6.4.1. Validación Cruzada

La validación cruzada es una técnica de remuestreo que se utiliza para evaluar modelos de aprendizaje automático. Principalmente, se utiliza estimar cómo se espera que se comporte el modelo al momento de realizar predicciones sobre datos que no se utilizaron para entrenar el modelo [12].

Esta técnica considera múltiples muestras de aprendizaje ($k > 1$), que se obtiene a partir de la muestra original y se utilizan para desarrollar distintos modelos. A menudo esta técnica se denomina validación cruzada de k veces (*k fold cross validation*), en donde k hace referencia al número de grupos en los que se divide la muestra original de datos. El proceso consiste básicamente en tomar uno de estos grupos para realizar el proceso de validación del clasificador. Mientras que los grupos restantes se utilizan para entrenamiento. Este proceso se repite hasta que todos los grupos hayan sido utilizados para validar el modelo, dando origen a k modelos mutuamente excluyentes [13].

Esta técnica es de las comunes dado que generalmente da como resultado una estimación menos sesgada y más robusta en presencia de muestras de tamaño reducido.

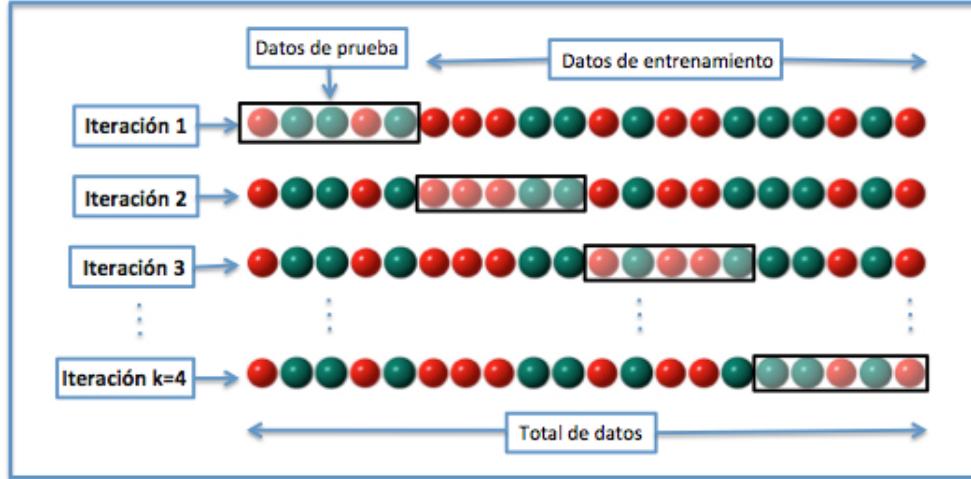


Figura 3: Proceso de validación cruzada [12].

6.4.2. Máquinas de Vectores de Soporte

Las máquinas de vectores de soporte o SVM (*Support Vector Machines*) son parte de un tipo de algoritmos de aprendizaje automatizado. Inicialmente, se crearon para resolver problemas de clasificación binaria en los que las clases eran linealmente separables. Sin embargo, es posible utilizar esta herramienta para resolver problemas de regresión y clasificación multiclas [14].

Uno de los principales objetivos de las máquinas de soporte vectorial es encontrar un hiperplano óptimo de separación en donde se maximiza el margen de separación entre las muestras de cada clase. Esto se logra mediante un entrenamiento previo con una serie de datos dispuestos especialmente para esta aplicación [14].

En los casos en donde los datos no sean linealmente separables, es decir, no se puedan separar por un hiperplano porque se encuentran mezclados en una región del espacio. Se realizan transformaciones a los datos, por medio de funciones kernel, que transforman el espacio a uno de mayor dimensión. De esta forma aumenta la posibilidad de que los datos entre clases sean linealmente separables [15].

Una de las ventajas de las máquinas de vectores de soporte es que proporcionan soluciones con un número reducido de datos, además de presentar una mejor generalización al fundamentarse en teorías de aprendizaje estadístico [15].

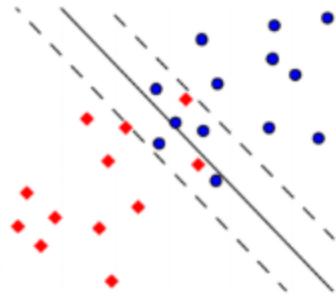


Figura 4: Datos no separables linealmente [15].

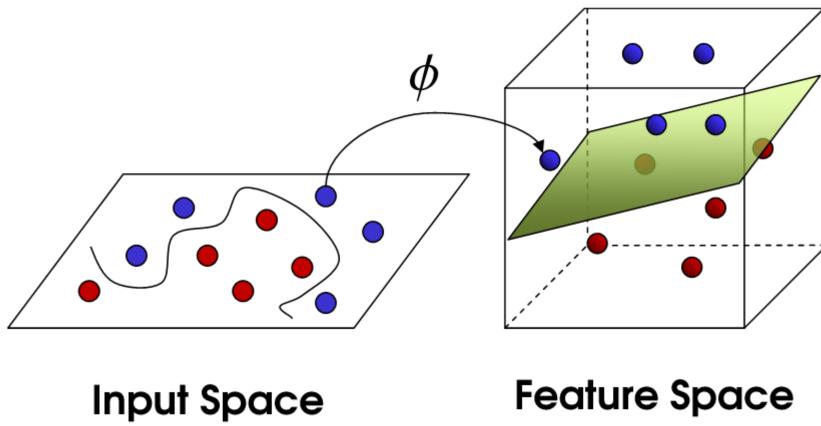


Figura 5: Separación de clases mediante la transformación de datos de entrada a un espacio de mayor dimensión [16].

6.4.3. Redes Neuronales

Las redes neuronales (RNAs) son una de las ramas más destacadas de la inteligencia artificial. Estas redes buscan modelar el comportamiento del cerebro humano, que se caracteriza por el aprendizaje a través de la experiencia. Por ello, estos modelos son capaces de resolver problemas de clasificación, identificación de patrones, diagnóstico, optimización o predicción de datos, por medio de algoritmos de aprendizaje supervisado o no supervisado [3].

Al ser inspiradas por comportamientos biológicos y el sistema nervioso, las redes neuronales se modelan como un sistema de interconexiones en capas. En donde las neuronas artificiales colaboran entre sí para generar salidas que ayuden a resolver problemas, por medio del procesamiento de ciertos datos de entrada [13].

Las redes neuronales presentan una gran ventaja dada su capacidad de adaptarse y modificar su arquitectura según su entorno. Además, de ser capaces de trabajar en paralelo, según la cantidad de neuronas artificiales que intervienen en el proceso. Por lo que pueden procesar gran cantidad de información rápidamente [13].

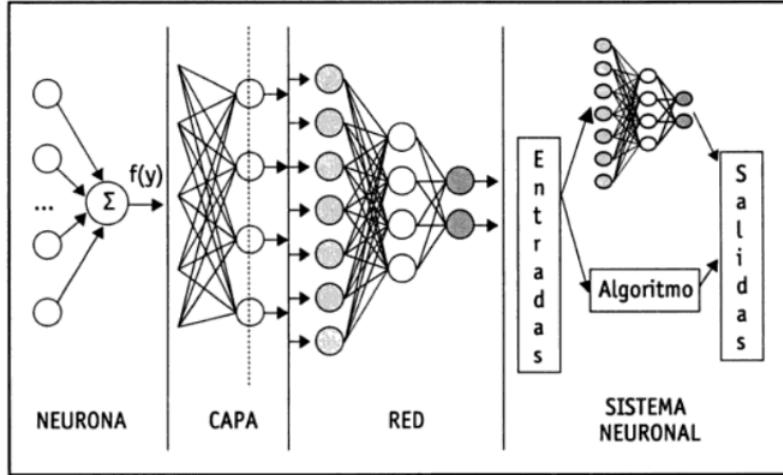


Figura 6: Estructura de un sistema de redes neuronales [13].

6.5. Dispositivos Robóticos

Según la Organización Internacional de Normalización ISO por sus siglas en inglés (*International Organization for Standardization*) se define a un robot como una máquina formada por un mecanismo con varios grados de libertad, generalmente con apariencia de uno o más brazos que terminan con una herramienta capaz de sujetar piezas de trabajo o un elemento de inspección. Estas máquinas son de propósito múltiple y se diseñan a menudo para llevar a cabo funciones repetitivas, sin embargo, son reprogramables y se pueden adaptar a otras funciones [17].

Generalmente, se habla de un robot cuando la máquina o brazo dispone de sensores que interactúan con el entorno. Y de manipuladores cuando el brazo o elemento articulado no está provisto de sensores que interactúen con el medio. Los dispositivos robóticos se componen principalmente de un sistema de control y de una interfaz para el sistema de control.

- **Sistema de Control:** Se refiere al conjunto de funciones de potencia y de control lógico que permiten el seguimiento y control de la estructura mecánica del robot y la comunicación con el entorno, incluyendo otros equipos y a los usuarios [18].
- **Interfaz:** Es la encargada de adaptar las señales y comandos entregados por el computador a los actuadores del robot, para que estos realicen un movimiento o secuencia de movimientos específicos. Por otra parte, los sensores del robot también entregan información al computador por medio de la interfaz, con el fin de que este reconozca el ambiente de trabajo según la tarea o proceso que se ejecute [18].

6.5.1. Manipulador Serial

Un manipulador serial, también conocido como manipulador de cadena cinemática abierta, se forma de varios elementos conectados consecutivamente por medio de juntas de tipo

prismático o esférico. El primer elemento de un manipulador serial se conoce como base. Mientras que el último elemento se conoce como efecto final, el cual consiste en una herramienta conectada al último eslabón del manipulador que se encarga de realizar las tareas de manipulación en el espacio de trabajo [19].

Una de las aplicaciones más comunes para los manipuladores seriales en la industria son las tareas de *pick and place*, estas tareas consisten básicamente en recoger productos para posteriormente colocarlos en algún otro lugar en el espacio de tarea.

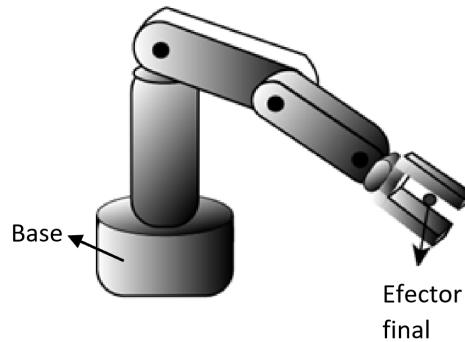


Figura 7: Manipulador serial [19].

CAPÍTULO 7

Dispositivos para Adquisición de Señales sEMG

Al seleccionar como línea de investigación el área de “Biomédica”, se inició con la investigación sobre señales bioeléctricas (electromiográficas, encefalográficas y electrocardiográficas), así como el procesamiento de las mismas y los instrumentos que se utilizan para su medición.

Después de esta investigación previa, se planteó como proyecto la creación de una interfaz biomédica para el control de dispositivos robóticos utilizando señales bioeléctricas. Dado que la universidad del Valle cuenta con el dispositivo Bitalino [20], un módulo basado en el microcontrolador de Arduino, que permite recoger datos biomédicos a través de una serie de módulos individuales. Entre ellos un módulo que permite medir la actividad eléctrica producida por los músculos del cuerpo. Se concluyó utilizar esta herramienta para medir señales electromiográficas, enfocando el proyecto en el desarrollo de una interfaz biomédica que mediante la adquisición, filtrado y un posterior proceso de clasificación automática de señales electromiográficas de superficie, sea capaz que controlar dispositivos robóticos con los que cuenta la universidad del Valle.

7.1. Familiarización con el Bitalino y *OpenSignals*

Dado que no existen fases anteriores de este proyecto en la universidad del Valle, se inició con la familiarización de las herramientas disponibles para la medición de señales EMG.

El Bitalino es un módulo que permite recoger datos biomédicos a través de una serie de módulos individuales. Cuenta con cuatro canales, de los cuales dos permiten llevar a cabo la medición de señales electromiográficas. Para una interacción directa con la información proveniente del Bitalino se utilizó la herramienta *OpenSignals* [21], un *software* que permite la visualización en tiempo real de los datos mediante una conexión *bluetooth* con el Bitalino. Además, permite generar un registro de los datos adquiridos, almacenándolos en un archivo de texto al finalizar la sesión.



Figura 8: Conexión Bitalino y *OpenSignals*.

Al hacer la conexión entre el software *OpenSignals* y el Bitalino, se procedió a realizar pruebas de adquisición de señales EMG en tiempo real, como se observa en las figuras 9 y 10. Para estas primeras pruebas se conectó un electrodo de referencia en el hueso pisiforme y dos electrodos en los músculos superiores del brazo derecho como se observa en la figura 11. Estas pruebas se realizaron únicamente para verificar la factibilidad del uso del Bitalino y para familiarizarse con las señales sEMG.

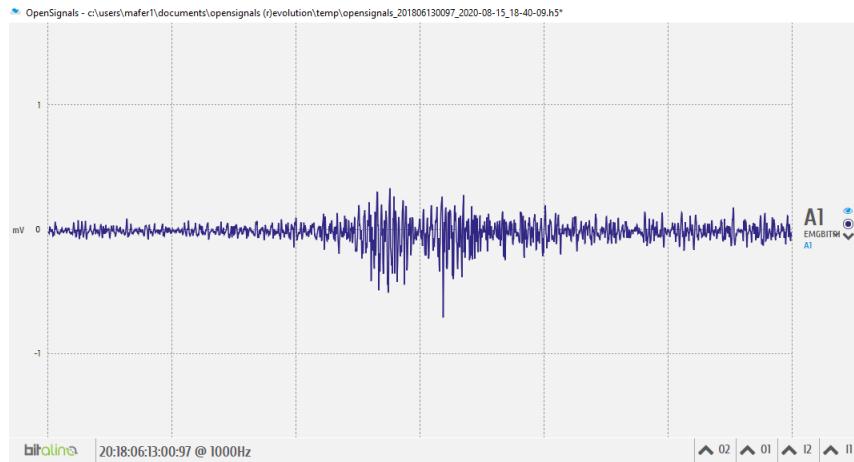


Figura 9: Señal electromiográfica de prueba visualizada con *OpenSignals*.

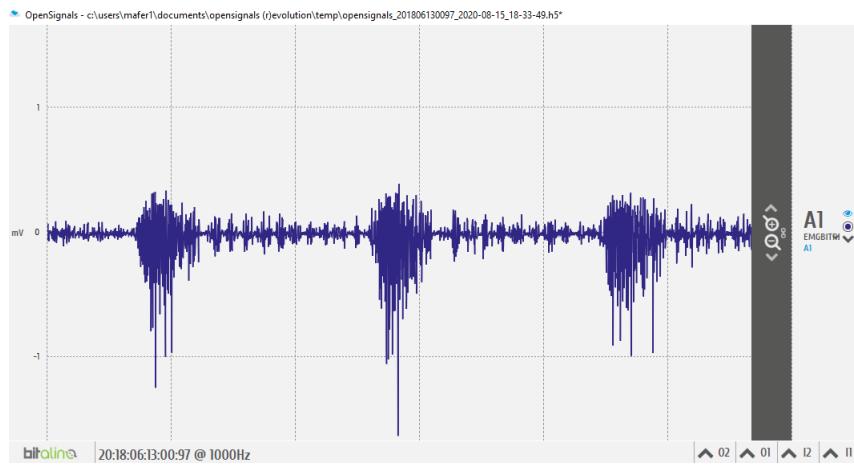


Figura 10: Señal electromiográfica de prueba visualizada con *OpenSignals*.

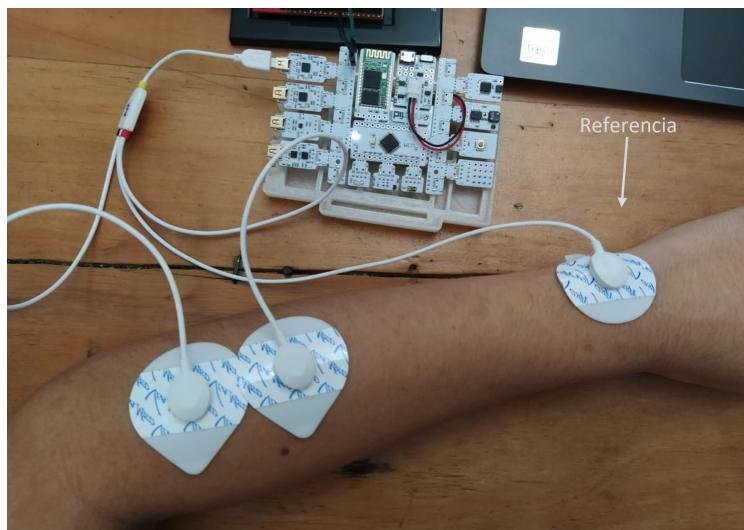


Figura 11: Ubicación de los electrodos para adquisición de señales sEMG.

La herramienta *OpenSignals* es de mucha utilidad para un primer acercamiento en el ámbito de la ingeniería biomédica y la visualización de señales bioeléctricas en tiempo real. Sin embargo, para el desarrollo de este proyecto se necesitaba tener acceso a los datos numéricos en tiempo real, para poder realizar un proceso de filtrado y análisis posterior a la adquisición. Por lo tanto, se requirió del uso de otro microcontrolador conectado al Bitalino, para extraer los datos directamente del Bitalino. Esto se describe en el próximo capítulo.

CAPÍTULO 8

Adquisición de Señales sEMG

8.1. Desarrollo de Algoritmo para Lectura de Datos

Como se mencionó anteriormente *OpenSignals* no permite interactuar con los datos numéricos en tiempo real, por lo que se utilizó un Arduino Uno para extraer estos datos directamente del Bitalino para un análisis posterior.

El enlace entre el Bitalino y Arduino se realizó mediante la conexión de las salidas analógicas del Bitalino con los puertos analógicos del Arduino. Sin embargo, esto solo constituyó un proceso intermedio para la adquisición de señales. Se requería de una herramienta de programación más avanzada para el procesamiento y la clasificación automática. Por lo que se evaluó trabajar con los lenguajes de programación Matlab y Python, para el desarrollo del algoritmo de adquisición de señales.

Luego de evaluar las alternativas se decidió utilizar Matlab por varias razones. Primero, la experiencia que ya se tenía en cuanto al desarrollo de filtros y procesamiento de señales. Segundo, esta opción cuenta con más documentación acerca de algoritmos de clasificación automática. Específicamente, Matlab posee distintas herramientas y librerías que facilitan la implementación de clasificadores como redes neuronales y máquinas de vectores de soporte. Y por último, esta plataforma cuenta con herramientas como la *Robotics Toolbox* de Peter Corke [22] que permite diseñar, simular y probar manipuladores seriales. Este *toolbox* fue de mucha utilidad para el desarrollo de la interfaz biomédica.

El algoritmo de adquisición de señales se inició estableciendo la conexión entre Arduino y Matlab mediante comunicación serial. El código de Matlab se comenzó estableciendo el puerto y el baudaje a utilizar para la comunicación serial, para este caso se utilizó 115200 baudios/s. Con la función `fwrite`, al ejecutar el programa se envía a través del puerto serial a arduino el comando “0”, el cual activa la lectura y transmisión de datos de arduino hacia Matlab. Luego de enviar este comando, se inició un bucle en donde utilizando la función `fscanf` se lee constantemente la información enviada por el arduino. Dado que las entradas analógicas del Arduino Uno disponen de 10 bits de resolución, los valores recibidos

corresponden a valores entre 0-1024, por lo que posteriormente se realizó en Matlab una conversión a estos para tener su valor correspondiente entre 0-5 voltios. Este proceso se describe en la figura 12.

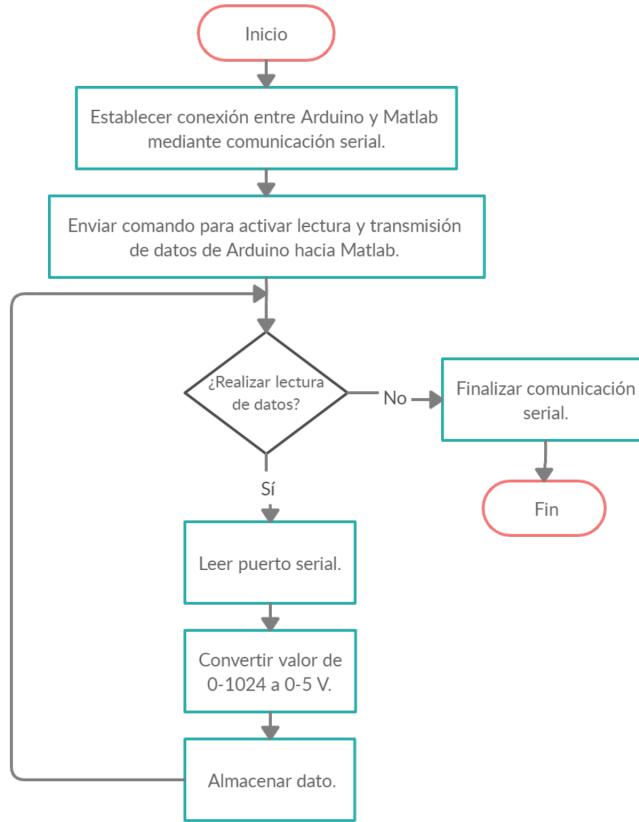


Figura 12: Algoritmo de adquisición de datos Matlab.

En el caso del código desarrollado en Arduino, descrito en la figura 13. Primero se importó la librería *TimerOne*, se prosiguió realizando la inicialización de variables, estableciendo las entradas analógicas y la inicialización de la comunicación serial. Al haber realizado la conexión, se utilizó la función `Serial.read` con la cual se espera a recibir el comando “0” para inicializar la lectura y envío de datos. Utilizando la librería *TimerOne* y las funciones `analogRead` y `Serial.println` se realiza la lectura y envío de datos cada 1 ms, a través del puerto serial hacia Matlab.

Se llegó a la conclusión de enviar los datos cada 1 ms después de haber realizado pruebas con tiempos menores y establecer que los datos se corrompián al momento de ser recibidos por Matlab. Impidiendo capturar fielmente los datos de las señales sEMG, lo cual podía comprometer el proceso de extracción de características y entrenamiento de clasificadores. Por otra parte, se concluyó también que realizar el envío cada 1 ms es suficiente considerando que las frecuencias de las señales EMG se encuentran principalmente entre 0 a 500 Hz.

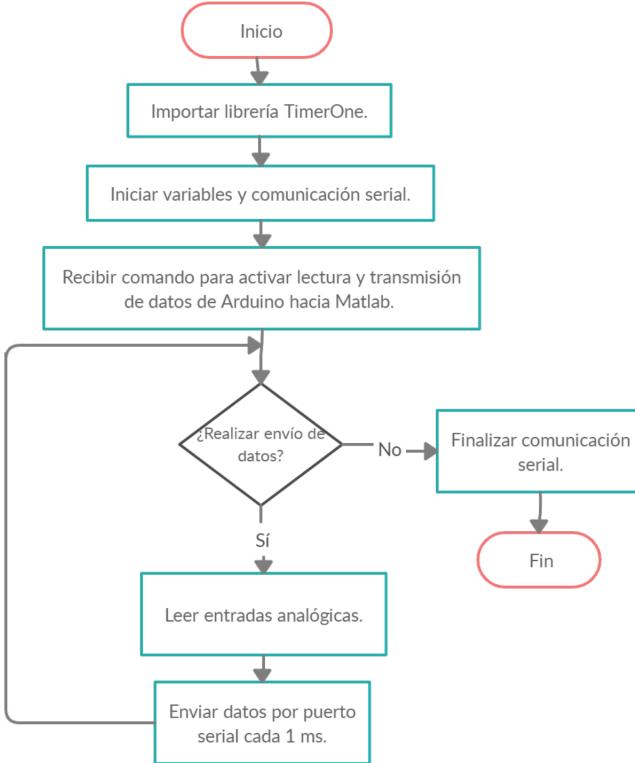


Figura 13: Algoritmo de adquisición de datos Arduino.

8.2. Procesamiento de Señales

Dada la naturaleza de las señales electromiográficas y que estas son obtenidas utilizando electrodos de superficie, los registros presentan un alto nivel de ruido. Por lo que, después de obtener los datos numéricos del Bitalino, se requirió un proceso de filtrado para obtener señales similares a las que se observan gráficamente con la herramienta *OpenSignals*.

Según la investigación realizada anteriormente sobre señales bioeléctricas, se determinó que las señales EMG presentan frecuencias que van desde 0 hasta 500 Hz, destacando mayor actividad en el rango de 50 a 150 Hz. Además, el ruido generado por el movimiento de los cables se encuentra entre 0 a 15 Hz. Por lo que se planteó el uso de un filtro pasa banda, para eliminar el efecto de frecuencias no deseadas y minimizar cualquier ruido proveniente de los artefactos. Por otra parte, se planteó el uso de un filtro rechaza banda para eliminar el ruido generado por la corriente alterna que circula a 60 Hz en los aparatos conectados a la red eléctrica.

Para llevar a cabo esta tarea se utilizó la herramienta *Filter Designer* disponible en la *Signal Processing Toolbox* de Matlab.

8.2.1. Signal Processing Toolbox

Signal Processing Toolbox es una herramienta disponible en Matlab que proporciona una variedad de funciones y aplicaciones para analizar, procesar y extraer características de señales. Dentro de esta *toolbox* se encuentran herramientas como *Filter Designer*, que mediante una interfaz gráfica permite al usuario diseñar rápidamente filtros de respuesta finita al impulso (FIR) o filtros de respuesta infinita al impulso (IIR) digitales, ofreciendo además la opción de generar el código correspondiente para implementar el filtro digital diseñado.

Antes de iniciar con el proceso de diseño de filtros se determinó qué tipo de filtros se utilizaría. Se decidió trabajar con filtros IIR ya que usualmente presentan funciones de transferencia más simples que las de los filtros FIR para operaciones similares, lo que implica que demandan menos poder computacional y se ejecutan más rápido. Posteriormente, se determinó utilizar filtros tipo *Butterworth* dado que se caracterizan por presentar una respuesta en frecuencia suave, sin ondulaciones en la banda de paso o de rechazo [23].

El primer filtro que se diseñó fue un filtro pasa banda con frecuencias de corte en 20 y 450 Hz. Con el fin de obtener únicamente las frecuencias correspondientes a señales EMG y delimitar el ruido de los aparatos. El segundo filtro consistió en un filtro rechaza banda con frecuencias de corte en 58 y 62 Hz, para eliminar el ruido de la red eléctrica. En las figuras 14 y 15 se pueden observar las especificaciones de diseño que se establecieron para cada filtro mencionado anteriormente. Cabe destacar que para cada filtro se especificó la frecuencia de muestreo igual a 1 kHz, consistente con el período de 1 ms con el que se obtienen los datos.

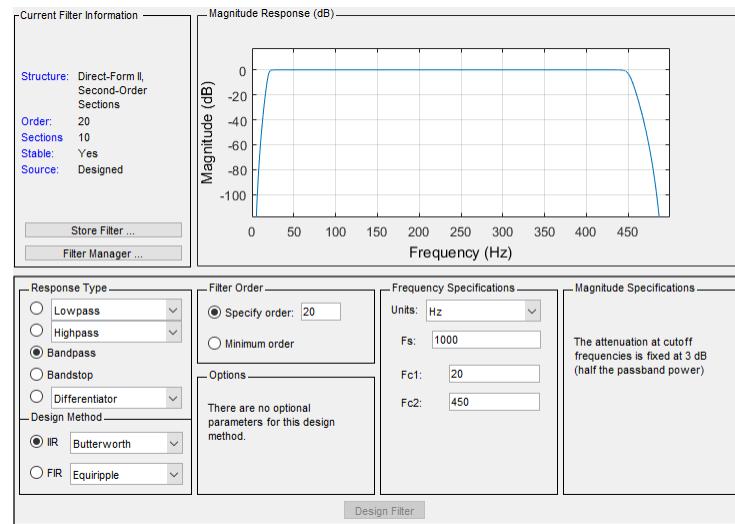


Figura 14: Diseño filtro pasa banda tipo *Butterworth*.

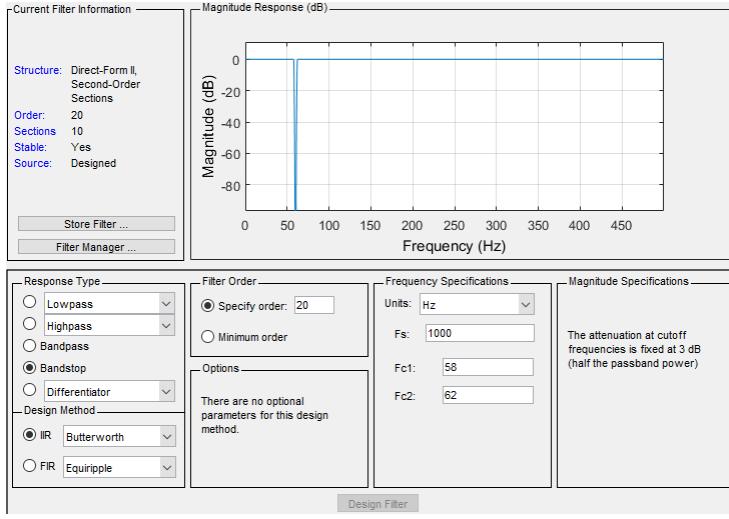


Figura 15: Diseño filtro rechaza banda tipo *Butterworth*.

Para verificar el funcionamiento de los filtros se realizó una prueba filtrando una señal conocida. En este caso se creó una señal conformada por un sinusoide de 60 Hz amplitud 1, un sinusoide de 100 Hz amplitud 1 y un sinusoide de 200 Hz amplitud 0.7. A continuación, se le aplicó a esta señal con la función `filter` un filtro pasa banda previamente diseñado con frecuencias de corte en 20 y 150 Hz. A la señal resultante de esta primera etapa de procesamiento se le aplicó un filtro rechaza banda con frecuencias de corte en 58 y 61 Hz. Los resultados de esta prueba se observan en la figura 16, en donde en la parte superior se observa la señal original de color rojo, en el medio la señal después de aplicar el filtro pasa banda de color verde y en la parte inferior el resultado de aplicar el filtro rechaza banda de color azul. Además, en esta figura se observa el espectro de frecuencias que conforma cada señal, demostrando que los filtros remueven efectivamente las frecuencias estipuladas.

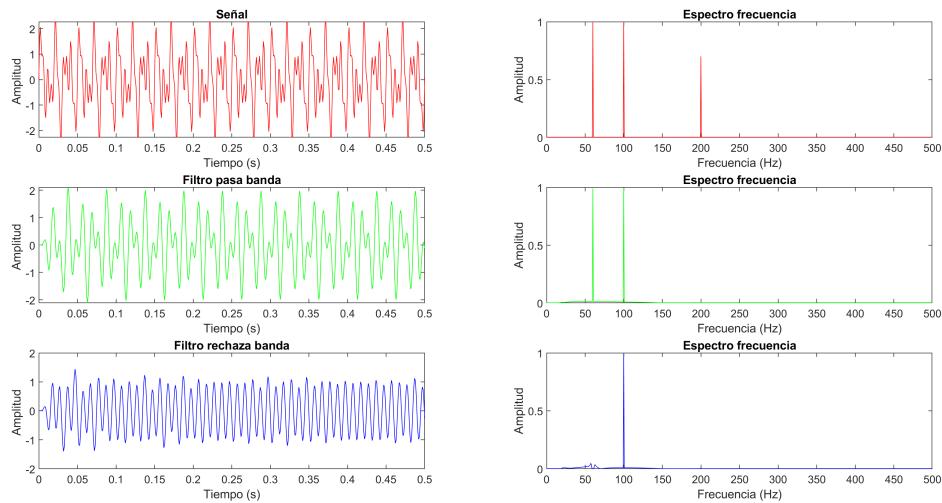


Figura 16: Filtrado señal de prueba.

8.3. Desarrollo de Algoritmo de Identificación de Actividad

Con el fin de hacer más eficiente el algoritmo de lectura de datos, se planteó el desarrollo de un algoritmo de identificación de actividad. Este algoritmo tiene como objetivo realizar una lectura continua de datos y al momento de detectar actividad eléctrica proveniente de los músculos iniciar el filtrado y almacenamiento de la señal. Al detectar actividad, dentro del algoritmo se especifica la ventana de tiempo que se quiere grabar, al concluir el tiempo especificado el algoritmo continúa con la lectura continua de datos, hasta detectar de nuevo actividad.

El primer paso para desarrollar este algoritmo consistió en la identificación de un umbral que determine el valor máximo de voltaje correspondiente al ruido de la señal. Para determinar este umbral, se requiere que el usuario permanezca dos segundos sin realizar ningún movimiento mientras se captura la señal. Luego, se rectifica la señal transformando los voltajes negativos a positivos y se obtiene el valor máximo de voltaje. Al obtener el valor del umbral se inicia el algoritmo de lectura de datos, donde continuamente se realiza la lectura del puerto serial. Al detectar un valor de voltaje mayor al umbral, más cierto nivel de tolerancia, se inicia el proceso de centrado y filtrado de la señal.

Para evitar la pérdida de información, se creó un buffer que almacena una ventana de tiempo con información previa a la actividad. Por lo que la señal resultante se compone de una ventana de información previa a la actividad y una ventana de información posterior a la detección de actividad, el tiempo de duración de cada ventana se declara previamente al inicio del código. Superado el tiempo de captura estipulado, la señal se almacena y se gráfica. Este proceso se encuentra dentro de un ciclo, por lo que al haber almacenado la señal final, se regresa a la lectura continua del puerto serial, para seguir en la espera de actividad muscular generada por la ejecución de un movimiento/gesto.

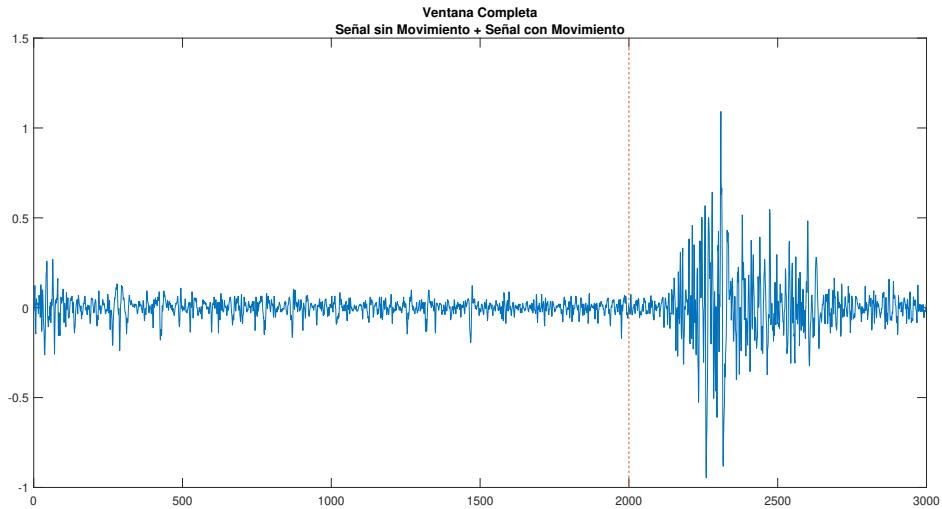


Figura 17: Ventana completa que incluye la señal sin movimiento más la señal con movimiento.

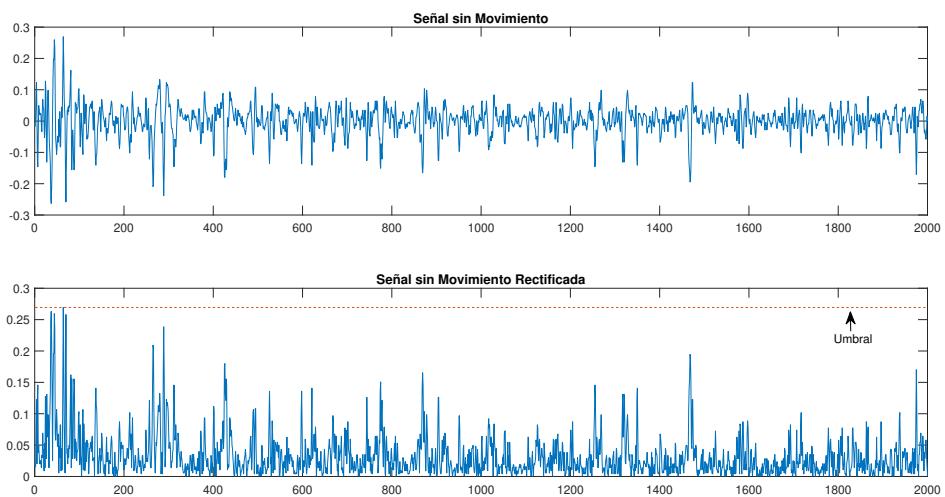


Figura 18: Señal sin movimiento y detección de umbral.

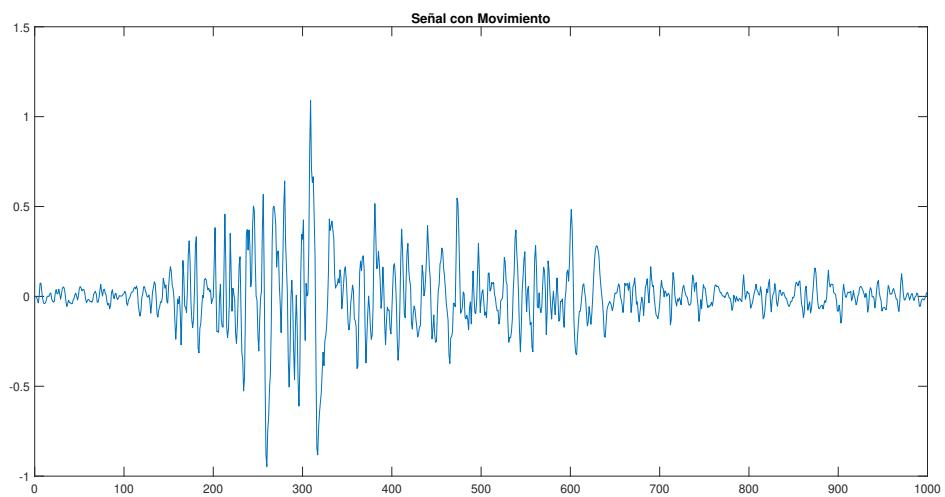


Figura 19: Señal con movimiento.

CAPÍTULO 9

Base de Datos de Señales sEMG

Para llevar a cabo el proceso de clasificación de señales es necesario definir las fronteras entre las diferentes clases. Estas fronteras se determinan mediante un proceso de entrenamiento en el que se usan las características de una serie de datos conocidos que ejemplifican las diferentes clases. Por esta razón, se requiere tener una colección de señales sEMG conocidas, que se puedan someter a un proceso de extracción de características que servirán para llevar a cabo el entrenamiento de clasificadores que determinará las fronteras entre clases. Para posteriormente poder aplicar el clasificador utilizando señales no conocidas.

Para iniciar con pruebas preliminares de entrenamiento de clasificadores se utilizó como colección de señales sEMG conocidas la información contenida en una base de datos pública. Mientras que, para llevar a cabo las pruebas finales de entrenamiento, se utilizaron señales que se recopilaron usando los dispositivos Bitalino y Arduino. Así como los algoritmos de detección de actividad y lectura de datos descritos en el capítulo 8. Esta señales se organizaron y almacenaron en una base de datos propia, que se detalla más adelante en este capítulo.

9.1. Base de Datos Pública

Con el fin de realizar preliminares de entrenamiento de clasificadores se utilizó una base de datos pública utilizada en investigaciones anteriores sobre clasificación de señales EMG [24]. En esta base de datos se recopila información de cinco sujetos de prueba que realizan repetidamente seis tipos de agarre, los cuales se muestran en la figura 20, se tiene un total de 30 repeticiones por cada movimiento en una ventana de tiempo de 6 segundos.

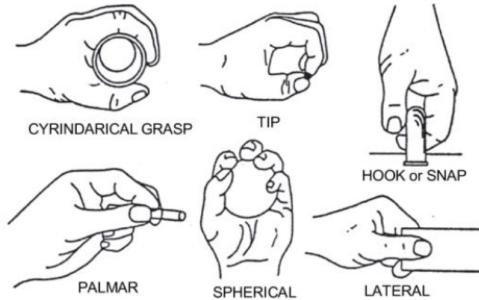


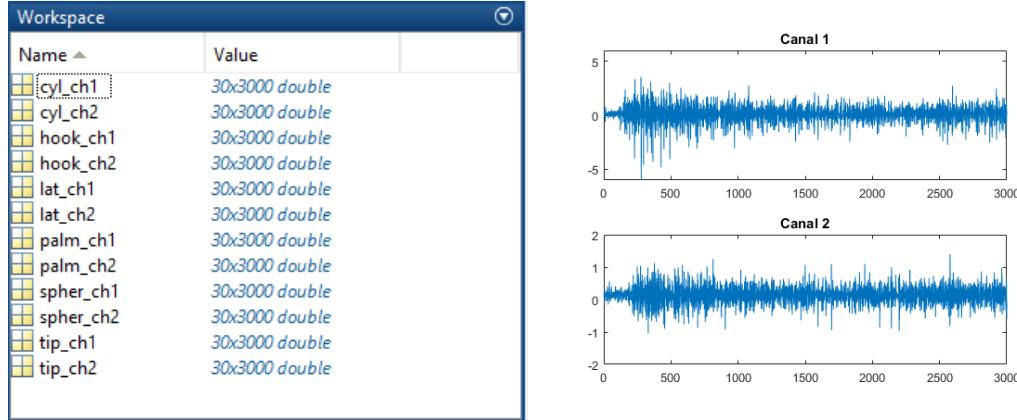
Figura 20: Tipos de agarre incluídos en la base de datos.

Las señales EMG presentes en esta base de datos fueron adquiridas utilizando electrodos de superficie y transmitidas por medio de dos canales, la información se adquirió a una frecuencia de muestreo de 500 Hz y posteriormente fue filtrada utilizando un filtro pasa bandas *Butterworth* con frecuencias de corte de 15 a 500 Hz y un filtro *Notch* a 50 Hz.

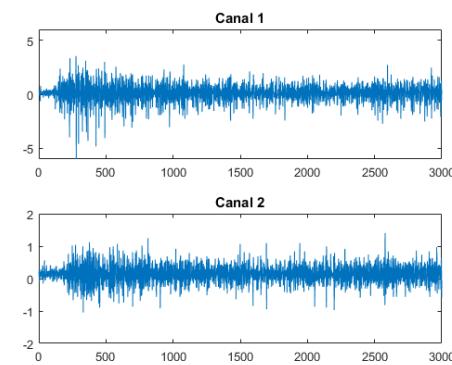
Sujeto	Grabación	No. muestras	Muestras por movimiento
1	female_1.mat	180	30
2	female_2.mat	180	30
3	female_3.mat	180	30
4	male_1.mat	180	30
5	male_2.mat	180	30

Cuadro 1: Descripción de las grabaciones almacenadas en la base de datos pública.

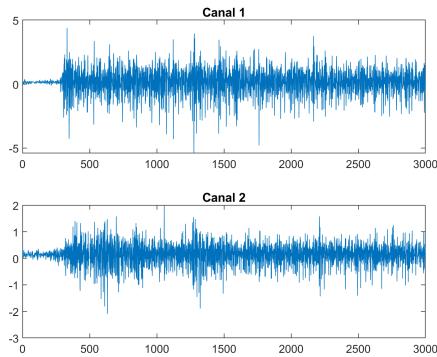
Se realizaron pruebas de entrenamiento con los datos recopilados de los cinco sujetos de prueba (cuadro 1). Cuyas señales se encuentran almacenadas en archivos .mat que incluyen cada uno 12 matrices, dos matrices por cada movimiento cada una correspondiente a un canal. Cada matriz se compone de 30 filas (número de repeticiones) y 3000 columnas (puntos de la señal). La organización de estos archivos .mat se detalla en la figura 21.



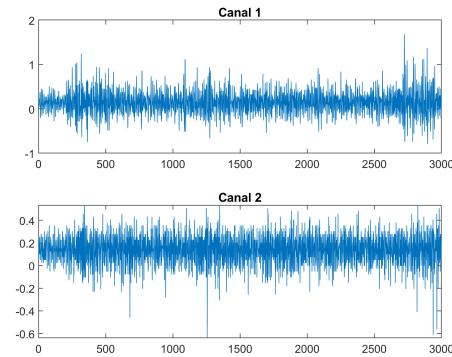
(a) Organización base de datos.



(b) Visualización señal, agarre cilíndrico.



(c) Visualización señal, agarre de gancho.



(d) Visualización señal, agarre esférico.

Figura 21: Información contenida en la base de datos pública.

9.2. Generación de Base de Datos Propia

Para las siguientes pruebas de entrenamiento se creó una base de datos propia. Para crear esta base de datos se obtuvieron las señales sEMG utilizando el algoritmo de adquisición de señales descrito en el capítulo 8.

Las señales captadas corresponden a 4 movimientos, descritos en la figura 22. Se realizaron repetidamente 40 grabaciones para cada movimiento, en donde cada grabación recopila información de un segundo de actividad.

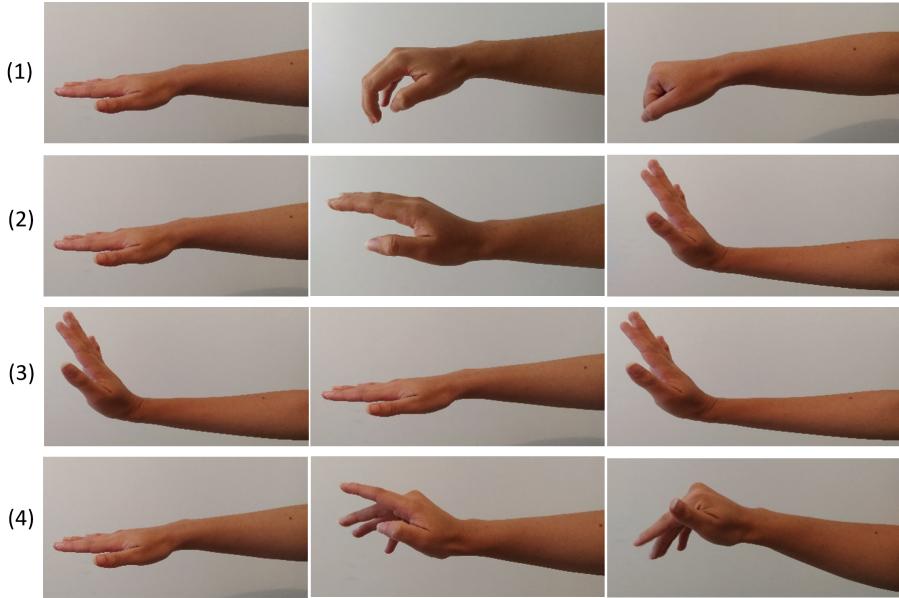


Figura 22: Movimientos incluídos en la base de datos: 1) empuñar, 2) arriba, 3) doble arriba y 4) movimiento de dedos.

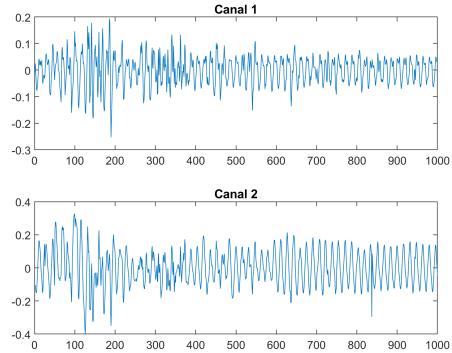
Las señales sEMG presentes en esta base de datos se adquirieron mediante dos canales del Bitalino, utilizando electrodos de superficie. La información se adquirió a una frecuencia de muestreo de 1 kHz y posteriormente fue filtrada utilizando un filtro pasa bandas *Butterworth* con frecuencias de corte de 20 a 450 Hz y un filtro rechaza bandas *Butterworth* con frecuencias de corte de 58 a 62 Hz.

Las señales captadas se almacenaron en un archivo .mat que reúne tanto la información de las señales, como los vectores de etiquetas necesarios para el entrenamiento de clasificadores. En las figuras 23 y 24 se observa la información contenida en esta base de datos, en donde la matriz `data` contiene las señales captadas. El número de filas corresponde al número de grabaciones, en este caso son 4 clases, 40 grabaciones por clase resultando en un total de 160 grabaciones. Y el número de columnas corresponde al número de puntos captados por ambos canales en una ventana de tiempo de 1 segundo, los primeros mil puntos corresponden al canal 1 y los siguientes mil puntos corresponden al canal 2.

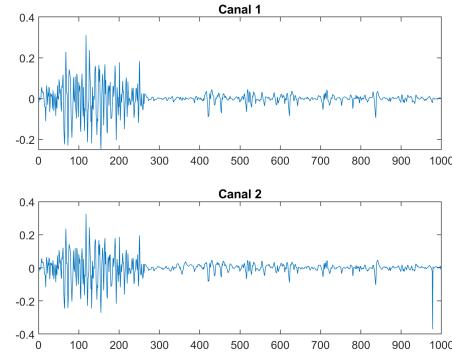
Se incluyen también los vectores de etiquetas `labels` y `labels_svm` que se utilizarán posteriormente para el entrenamiento de clasificadores con redes neuronales y SVM respectivamente. La estructura de los vectores de etiquetas se describe a detalle en la sección 10.2.

Workspace	
Name	Value
<code>data</code>	<code>140x2000 double</code>
<code>labels</code>	<code>4x140 double</code>
<code>labels_svm</code>	<code>1x140 double</code>

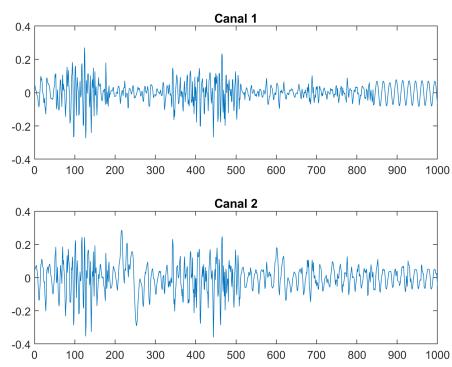
Figura 23: Información contenida en la base de datos propia correspondiente a 4 clases.



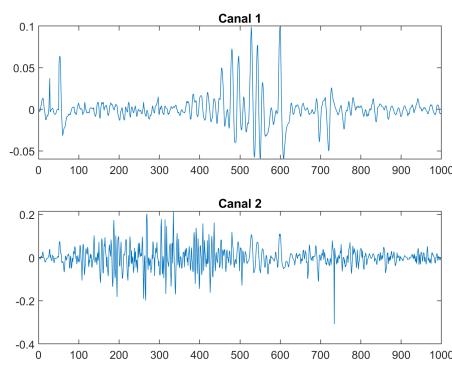
(a) Empuñar.



(b) Arriba.



(c) Doble arriba.



(d) Movimiento dedos.

Figura 24: Visualización de señales correspondientes a cada movimiento/clase.

CAPÍTULO 10

Clasificación de Movimientos/Gestos

Dado que no se tenía experiencia previa trabajando con clasificación automática, se inició con la lectura de documentos enfocados en como aplicar métodos basados en *Machine Learning* para realizar procesos de clasificación. Principalmente, se investigó acerca de máquinas de vectores de soporte y redes neuronales. Algunos de los documentos que sirvieron de apoyo se encuentran en [14], [13], [3]. Además, se trabajó con ejemplos que brinda Matlab sobre el uso de redes neuronales, lo que sirvió como base para entender su funcionamiento.

Para el caso de las máquinas de vectores de soporte, se utilizó como recurso la librería **libsvm** disponible para Matlab que se encuentra en [25], que incluye además un instructivo de uso y ejemplos de cómo llevar a cabo el proceso de clasificación. Estos recursos sirvieron de base para el desarrollo de un sistema de clasificación propio basado en SVM. En este capítulo se recopilan los resultados obtenidos al implementar diferentes técnicas de clasificación, así como variaciones en los parámetros del clasificador.

10.1. Extracción de Características

La etapa de extracción de características consiste principalmente en seleccionar una serie de valores representativos para un determinado conjunto de datos. En este caso el proceso resultó ser complejo y de alta duración, dada la variedad de características que se pueden obtener de una señal y las combinaciones que se pueden realizar entre ellas. Por ello, fue necesario llevar a cabo una serie de pruebas para determinar las características más adecuadas para entrenar al clasificador según el grupo de señales con las que se estaba trabajando. Es importante resaltar que los resultados de dichas pruebas se detallan en secciones posteriores, mientras que en esta sección se describe únicamente el proceso que se llevó a cabo para extraer dichas características.

El número y el tipo de características a extraer varía según el problema, la naturaleza específica de la señal y la cantidad de datos disponibles. Por ello se aconseja iniciar con un conjunto de características que hayan sido utilizadas en investigaciones anteriores y prose-

uir con una serie de pruebas para determinar cuáles aportan más valor en el proceso de clasificación y qué combinación de características brinda un mejor porcentaje de rendimiento.

En este trabajo se tomó como punto de partida la investigación realizada en [5], en donde utilizan características en el domino del tiempo, frecuencia y tiempo-frecuencia. Destacando, que las características temporales presentan una ventaja sobre el resto al requerir de menos tiempo de procesamiento. Las características temporales propuestas en esta investigación son: valor medio absoluto (MAV), cruces por cero (ZC), raíz media cuadrada (RMS), desviación estándar (STD), varianza (VAR) y longitud de forma de onda (WL). Dado que investigaciones previas han reportado resultados satisfactorios utilizando características temporales, se tomó la decisión de emplear este tipo de características para el proceso de entrenamiento de clasificadores.

El primer paso para iniciar con el proceso de pruebas de entrenamiento de clasificadores consistió en la creación de funciones para extraer las siguientes características temporales: MAV, ZC, IEMG, RMS, VAR, STD y WL. Estas características se encuentran detalladas en la sección 6.3. Con las funciones ya definidas, se procedió a realizar las primeras pruebas de extracción de características y clasificación con la base de datos pública.

El proceso que se realizó para extraer características consistió en tomar los 3000 datos, que corresponden a una ventana de tiempo de 6 segundos en donde se ejecuta un movimiento, para posteriormente extraer la característica utilizando las funciones desarrolladas. Para cada movimiento o gesto se muestraron señales en dos canales. A los datos de cada canal se le calcularon las características. Finalmente, el conjunto de características de ambos canales conforma un vector de características que describe el gesto.

Ya extraídas todas las características, se procedió a almacenar estos vectores en una matriz, en donde cada fila corresponde al vector de características representativo de un movimiento diferente. En la figura 25 se pude observar un ejemplo de la matriz que contiene los vectores de características de las señales almacenadas en la base de datos pública. Para este ejemplo se trajeron dos características MAV y ZC, dado que la información proviene de dos canales se obtuvieron vectores de 4 dimensiones porque a cada gesto se le calculan 2 características por canal. Por lo que las primeras dos columnas corresponden al resultado de MAV para el primer canal y el segundo canal respectivamente. Y las últimas dos columnas corresponden al resultado de ZC para el primer canal y segundo canal respectivamente. La base de datos almacena 180 señales, por lo que la matriz de características cuenta con 180 filas una por cada movimiento.

	1	2	3	4
1	0.6088	0.2457	1136	1115
2	0.6011	0.2441	1176	1109
3	0.5761	0.2065	1222	1018
4	0.6471	0.2577	1161	1184
5	0.6780	0.2219	1230	1033
6	0.6626	0.2177	1208	960
7	0.5230	0.2104	1158	996
8	0.6977	0.2351	1193	1118
9	0.6306	0.2321	1208	1078
10	0.5977	0.2295	1114	912
11	0.6152	0.2244	1157	980
12	0.6118	0.2138	1158	990

Figura 25: Vectores de características para la base de datos pública utilizando MAV y ZC.

10.2. Entrenamiento de Clasificadores

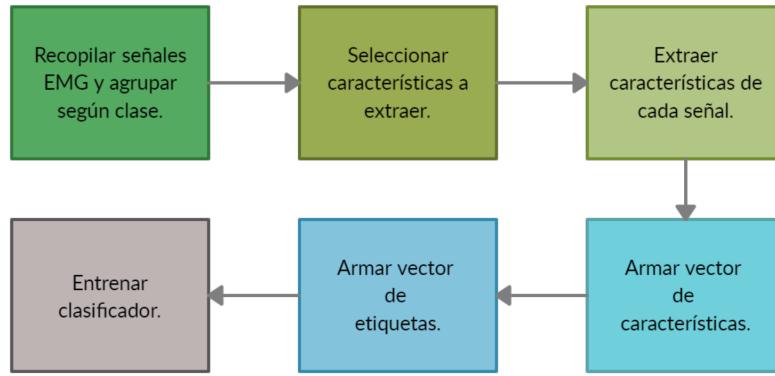


Figura 26: Proceso para entrenamiento de clasificadores.

Se trabajó con dos tipos de clasificadores principales, máquinas de vectores de soporte y redes neuronales. En la figura 26 se resume el proceso general que se llevó a cabo para entrenar los clasificadores, los detalles específicos del entrenamiento de cada tipo de clasificador se detallan a continuación.

Como se mencionó anteriormente para el desarrollo del clasificador basado en máquinas de vectores de soporte se utilizó como recurso la librería `libsvm`. Como paso inicial se descargó e instaló la librería para Matlab disponible en [25]. Posteriormente, se utilizó la función `svmtrain` para entrenar el clasificador SVM. Esta función recibe como argumentos la matriz de características, el vector de etiquetas y una serie de opciones que proporciona la librería para establecer el tipo de clasificador, tipo de kernel, validación cruzada, entre otras. Esta librería requiere un formato especial en los datos de entrada, por lo que previo al entrenamiento se utilizaron las funciones `libsvmwrite` y `libsvmread` para convertir tanto la matriz de características como el vector de etiquetas en el formato requerido.

La estructura de la matriz de características utilizada en SVM es igual a la que se observa

en la figura 25. Por otra parte, el vector de etiquetas es únicamente un vector unidimensional en donde se identifica cada clase con una etiqueta determinada. Por ejemplo, el vector de etiquetas para la base de datos pública se compone de números del 1 al 6, en donde cada número representa un tipo diferente de agarre.

Con el fin de garantizar que los resultados son independientes de la partición entre datos de entrenamiento y datos de validación, se implementó la técnica *five fold cross validation*, esto se logró colocando entre las opciones ‘-v 5’ al utilizar la función `svmtrain`. Cabe mencionar que el proceso de validación cruzada no devuelve el modelo final del clasificador, sino únicamente el porcentaje de rendimiento del mismo con los parámetros establecidos. Por lo que posteriormente, al determinar los parámetros finales, se requiere de un segundo entrenamiento deshabilitando la opción de validación cruzada. De esta manera es posible generar el modelo final que se utilizará para futuras predicciones.

En cuanto al desarrollo del clasificador basado en redes neuronales se utilizó la *Neural Network App* de Matlab. Esta aplicación cuenta con una interfaz que facilita la interacción, en la cual se introducen los vectores tanto de etiquetas como de características y se establece el porcentaje de datos para entrenamiento, validación y prueba.

La matriz de características representativa de la porción de datos de entrenamiento se estructura como se observa en la figura 27. En donde las columnas corresponden a los vectores de características, cada uno correspondiente a un movimiento. En este ejemplo se utilizaron 2 características MAV y ZC, como las señales provienen de dos canales se obtuvieron vectores de 4 dimensiones, las primeras dos filas corresponden al resultado de MAV para el canal 1 y 2 respectivamente. Y las últimas dos filas corresponden al resultado de ZC para el canal 1 y 2 respectivamente. Cabe mencionar que los datos de esta matriz de características están transpuestos en comparación a los datos de la matriz de características para SVM mostrada anteriormente. Esto debido a que la funciones utilizadas por cada tipo de clasificador requieren un formato específico que les permita manejar los datos de entrada.

	1	2	3	4	5	6	7	8	9	10
1	0.6088	0.6011	0.5761	0.6471	0.6780	0.6626	0.5230	0.6977	0.6306	0.5977
2	0.2457	0.2441	0.2065	0.2577	0.2219	0.2177	0.2104	0.2351	0.2321	0.2295
3	1136	1176	1222	1161	1230	1208	1158	1193	1208	1114
4	1115	1109	1018	1184	1033	960	996	1118	1078	912

Figura 27: Ejemplo de matriz de características utilizado para entrenamiento con redes neuronales.

Por otro lado, a diferencia de SVM, el entrenamiento de clasificadores con redes neuronales requiere que los vectores de etiquetas se compongan únicamente de unos y ceros. Por lo que se crea una matriz de etiquetas cuyo número de filas es igual al número de clases a clasificar y el número de columnas igual al total de muestras capturadas. Para cada muestra (columna) se coloca un uno en la fila que corresponda al número de clase a la que pertenece la muestra, en el resto de filas de colocan ceros. Un ejemplo de esta estructura se ve en la figura 28, en donde se observa que las primeras diez muestras corresponden a la clase número 1.

	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0

Figura 28: Ejemplo de matriz de etiquetas para entrenamiento con redes neuronales.

10.3. Resultados Base de Datos Pública

Como se mencionó en capítulos anteriores para realizar pruebas preliminares de entrenamiento de clasificadores se utilizó una base de datos pública que reúne señales sEMG correspondientes a 6 distintos tipos de agarre. Para realizar la clasificación de señales cada tipo de agarre corresponde a una clase. En total se tienen 180 muestras para clasificar entre las distintas 6 clases, 30 muestras por clase.

Se realizaron pruebas con dos tipos de clasificadores, máquinas de vectores de soporte y redes neuronales. Para cada clasificador, se realizaron entrenamientos con distintos grupos de características, con el fin de evaluar su rendimiento. Los resultados se muestran a continuación mediante matrices de confusión.

Una matriz de confusión es una tabla de contingencia que sirve como herramienta estadísticas para la visualización y análisis del desempeño de algoritmos basados en aprendizaje automático. A través de estas matrices se contabiliza el grado de semejanza entre un conjunto de datos bajo control y un conjunto de datos de referencia, para los que se ha establecido una clasificación [26]. Para este caso el conjunto de datos de referencia representa la clase real a la que pertenecen las señales/movimientos y el conjunto de datos bajo control representa la clase predicha por el clasificador.

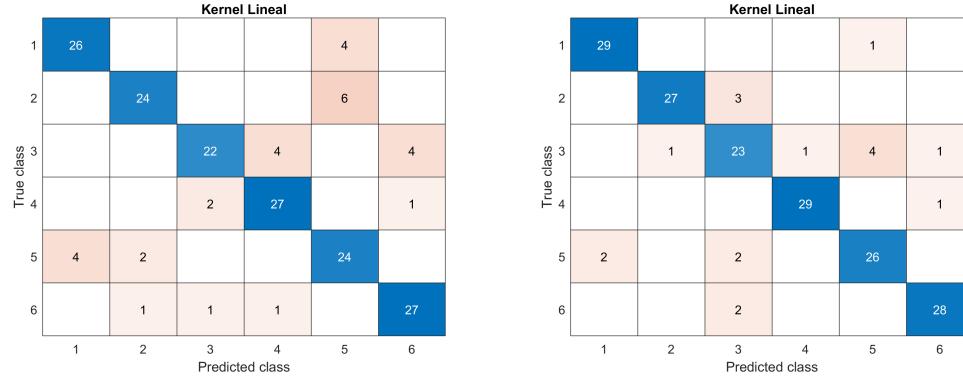
Las matrices de confusión son cuadradas de dimensión $M \times M$, en donde M representa el número de clases de la clasificación. Las clases del conjunto de datos de referencia se denominan clases referencia, mientras que las del conjunto de datos bajo control se conocen como clases producto. Las celdas de la diagonal de la matriz de confusión representan las señales bien clasificadas, en donde la clase referencia coincide con la clase producto. Por otra parte, las celdas fuera de la diagonal representan los errores resultantes de la clasificación. En resumen, las matrices de confusión brindan una visión completa de la distribución de coincidencias y errores entre clases, en procesos de clasificación automática [26].

Máquinas de vectores de soporte

Se utilizó la librería *libsvm* en Matlab, para realizar la clasificación de señales. Se implementó la técnica *five fold cross validation* para garantizar que los resultados son independientes de la partición entre datos de entrenamiento y de validación. Tomando en cada corrida el 80 % de datos para entrenamiento y el 20 % para validación. Además, se realizaron pruebas con dos tipos de Kernel, lineal y polinomial.

Kernel Lineal

Resultados con 2 características: MAV y ZC.

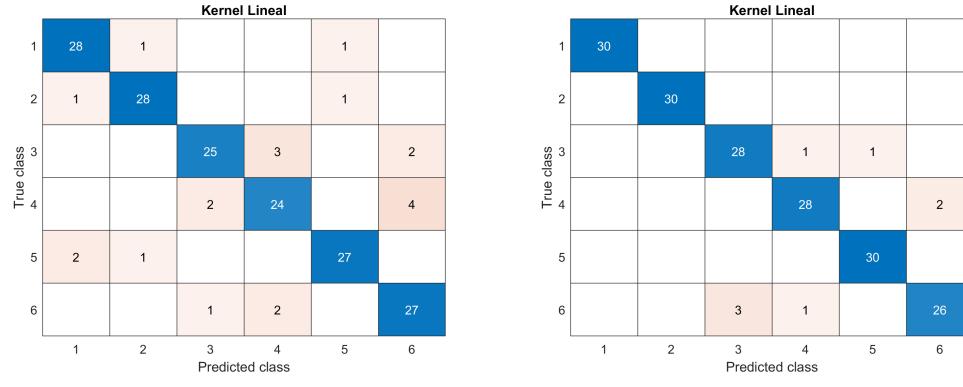


(a) Sujeto 1: *female_1.mat*.

(b) Sujeto 4: *male_1.mat*.

Figura 29: Resultados obtenidos con SVM lineal para clasificación de 6 clases, 2 características.

Resultados con 3 características: MAV, ZC y WL



(a) Sujeto 1: *female_1.mat*.

(b) Sujeto 4: *male_1.mat*.

Figura 30: Resultados obtenidos con SVM lineal para clasificación de 6 clases, 3 características.

Resultados con 5 características: MAV, ZC, IEMG, WL y RMS.

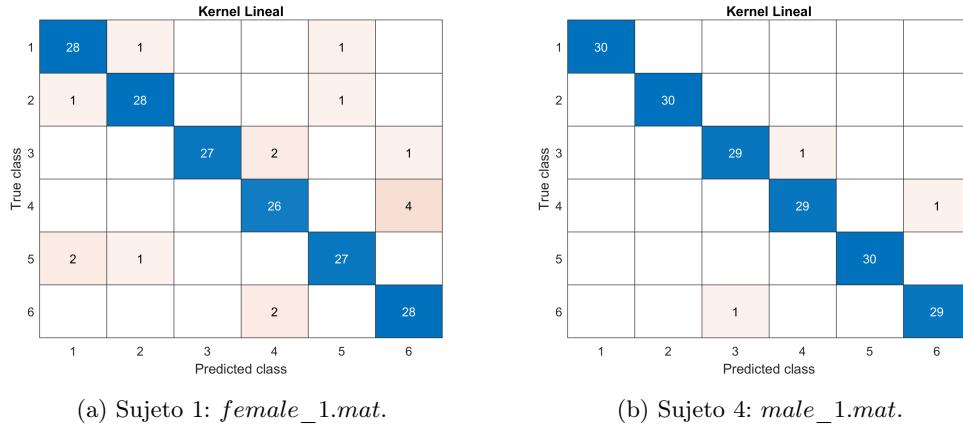


Figura 31: Resultados obtenidos con SVM lineal para clasificación de 6 clases, 5 características.

Resultados con 7 características: MAV, ZC, IEMG, WL, RMS, VAR y STD.

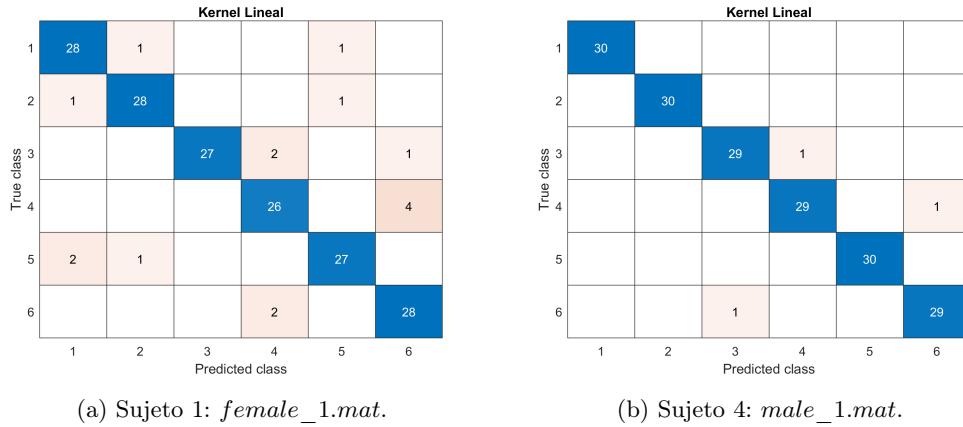
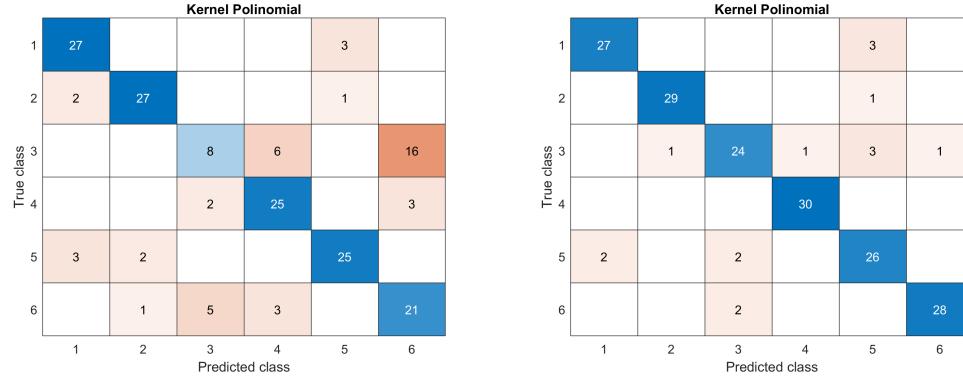


Figura 32: Resultados obtenidos con SVM lineal para clasificación de 6 clases, 7 características.

Kernel Polinomial

Resultados con 2 características: MAV y ZC.

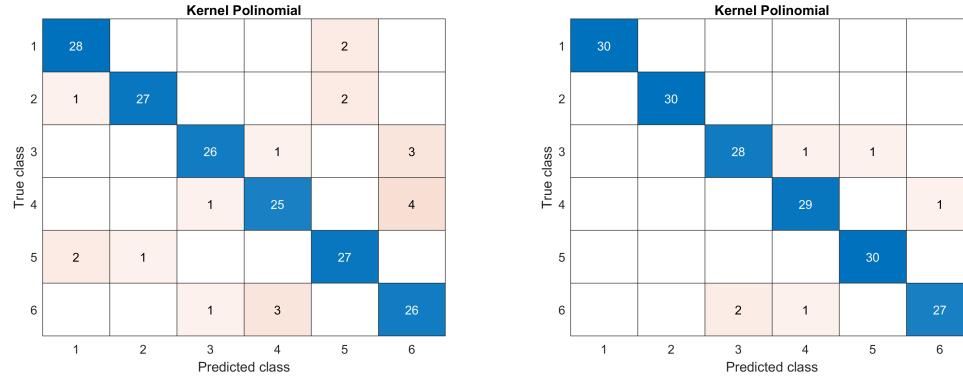


(a) Sujeto 1: *female_1.mat*.

(b) Sujeto 4: *male_1.mat*.

Figura 33: Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 2 características.

Resultados con 3 características: MAV, ZC y WL



(a) Sujeto 1: *female_1.mat*.

(b) Sujeto 4: *male_1.mat*.

Figura 34: Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 3 características.

Resultados con 5 características: MAV, ZC, IEMG, WL y RMS.

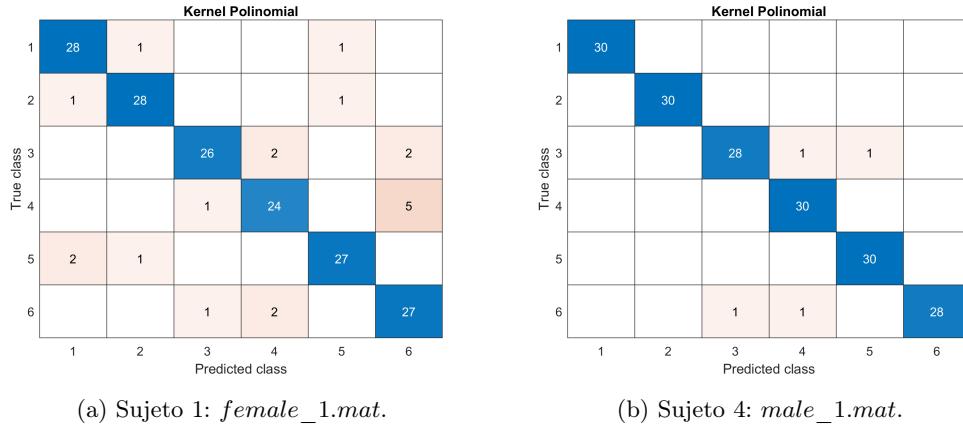


Figura 35: Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 5 características.

Resultados con 7 características: MAV, ZC, IEMG, WL, RMS, VAR y STD.

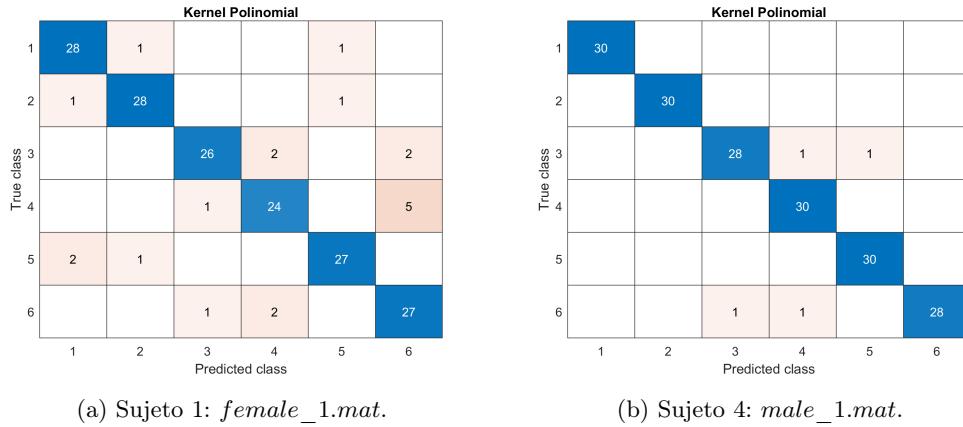


Figura 36: Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 7 características.

Redes neuronales

Se utilizó la *Neural Pattern Recognition App* de Matlab para implementar el clasificador, utilizando un 80 % de los datos para entrenamiento y un 20 % para validación y prueba. El rendimiento de la red generada se analizó mediante matrices de confusión, que indican los aciertos y los errores resultantes de la clasificación.

En las figuras 37-44 se observan los resultados de clasificación obtenidos con redes neuronales y distintos grupos de características para dos sujetos de prueba. En estas figuras se incluyen cuatro matrices de confusión, la primera recopila los resultados de la porción de datos utilizada para entrenamiento. Las siguientes dos representan los resultados de clasificación utilizando la porción de datos de validación y prueba. Y por último se resume el resultado de los aciertos y errores de las tres matrices anteriores, en una sola matriz. En estas

matrices las columnas representan la clase real a la que pertenecen las señales/movimientos y las filas el resultado predicho por el clasificador.

Resultados con 2 características: MAV y ZC.



Figura 37: Resultados obtenidos con RN para clasificación de 6 clases, 2 características. Sujeto 1: *female_1.mat*.

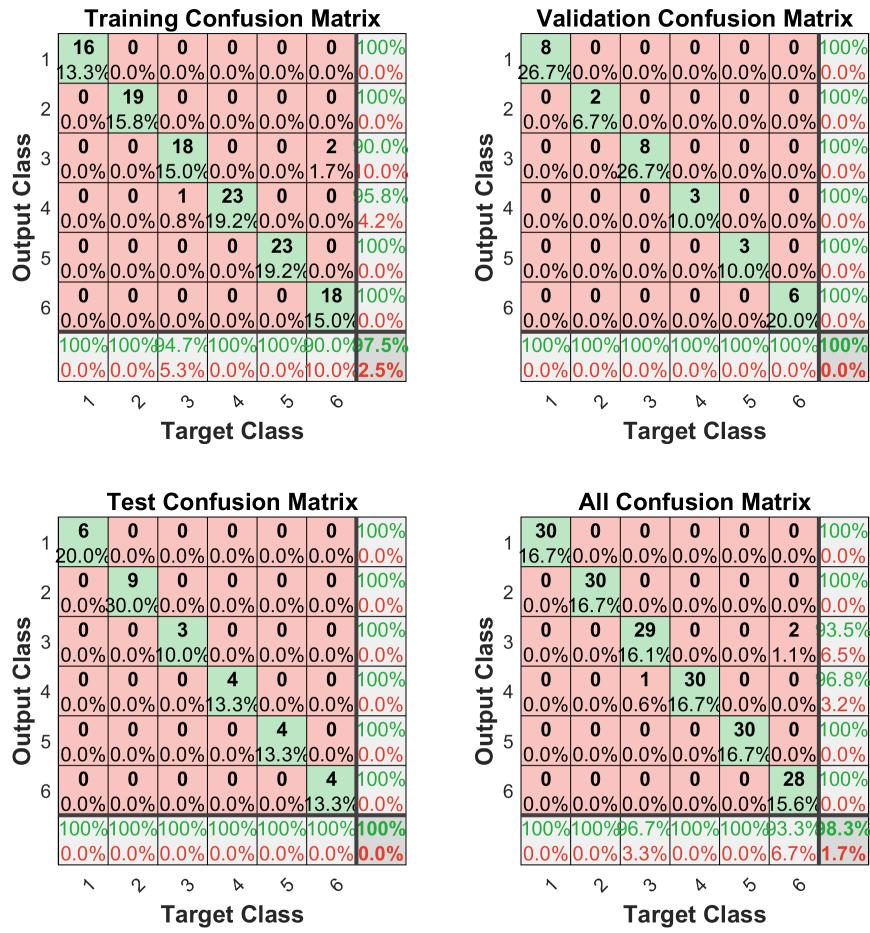


Figura 38: Resultados obtenidos con RN para clasificación de 6 clases, 2 características. Sujeto 4: *male_1.mat*.

Resultados con 3 características: MAV, ZC y WL.

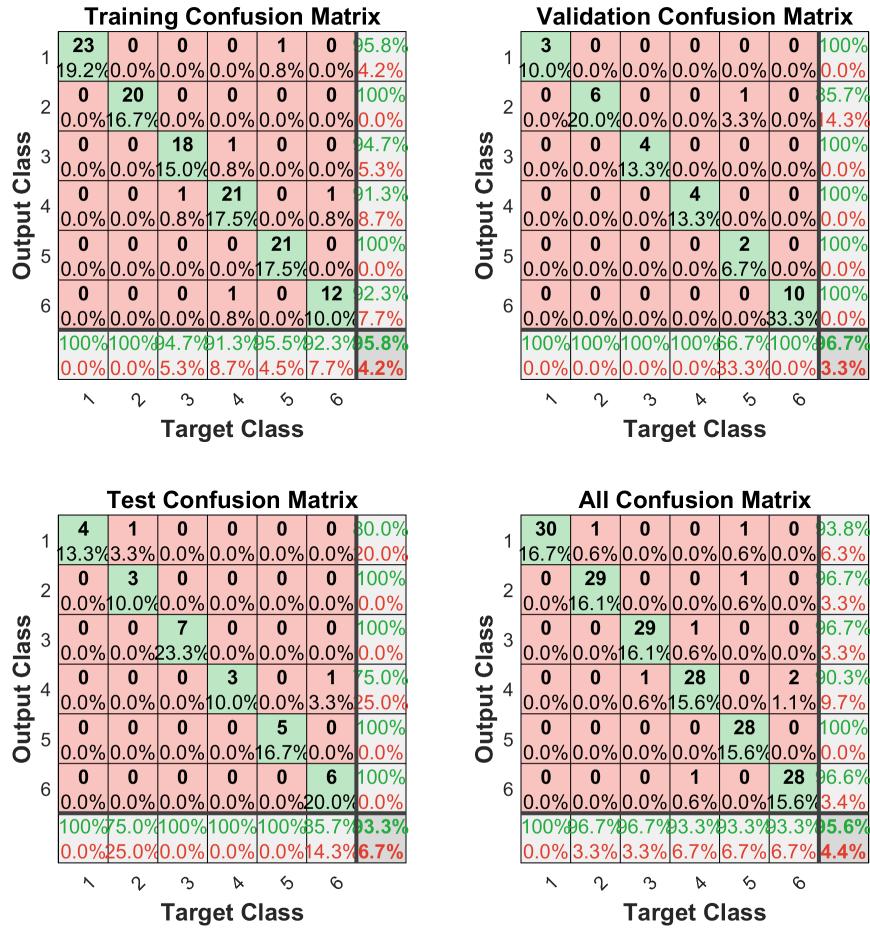


Figura 39: Resultados obtenidos con RN para clasificación de 6 clases, 3 características. Sujeto 1: *female_1.mat*.

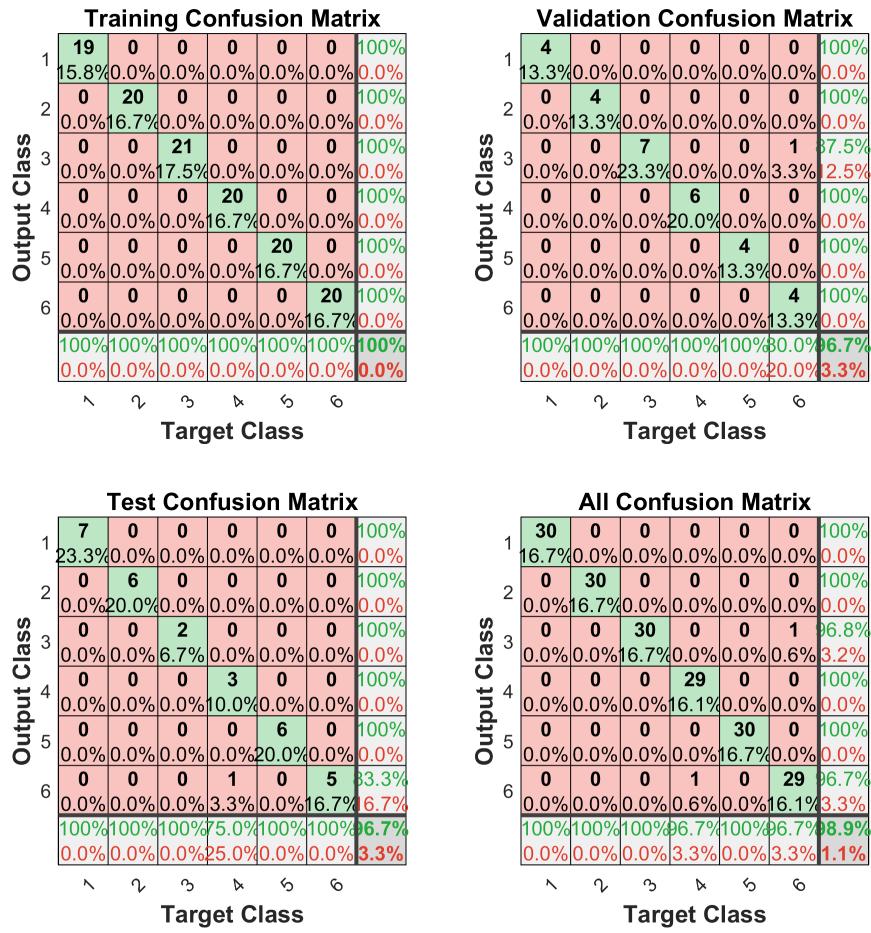


Figura 40: Resultados obtenidos con RN para clasificación de 6 clases, 3 características. Sujeto 4: *male_1.mat*.

Resultados con 5 características: MAV, ZC, IEMG, WL y RMS.

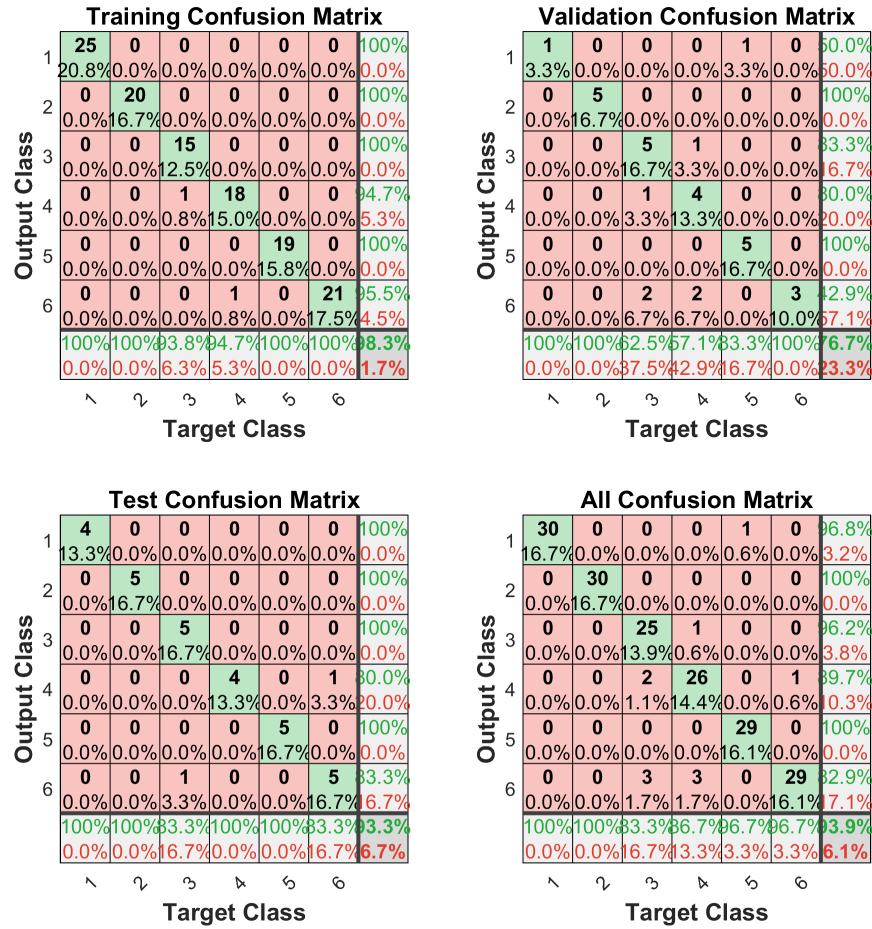


Figura 41: Resultados obtenidos con RN para clasificación de 6 clases, 5 características. Sujeto 1: *female_1.mat*.

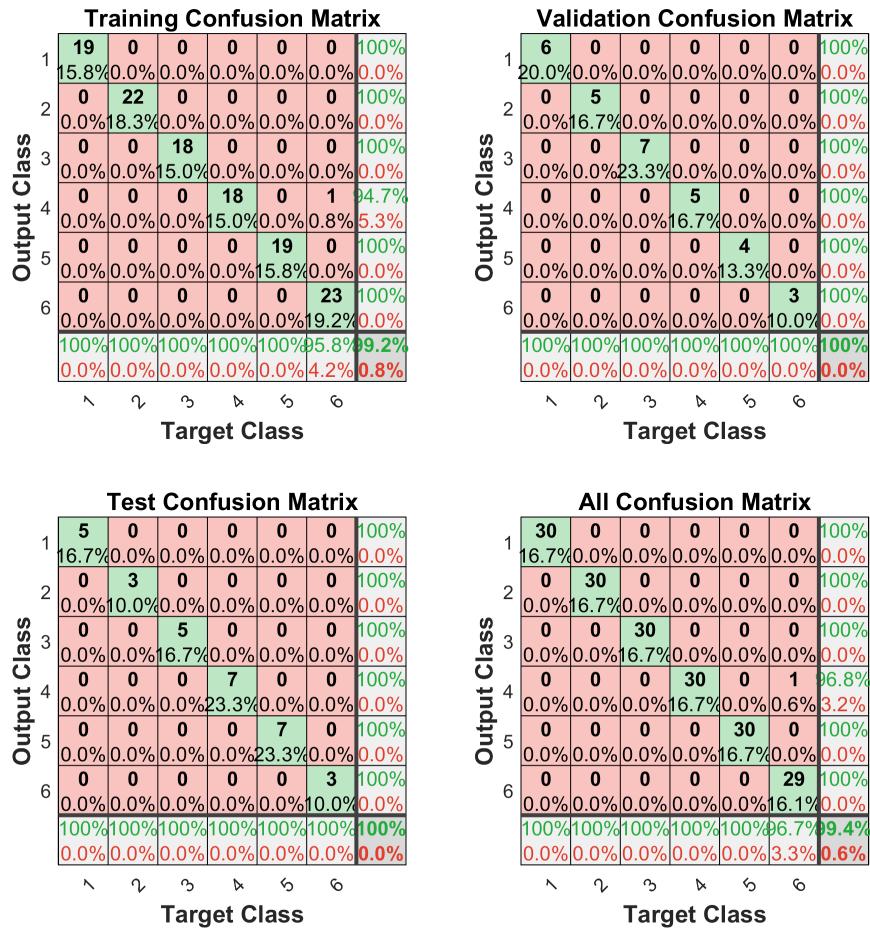


Figura 42: Resultados obtenidos con RN para clasificación de 6 clases, 5 características. Sujeto 4: *male_1.mat*.

Resultados con 7 características: MAV, ZC, IEMG, WL, RMS, VAR y STD.

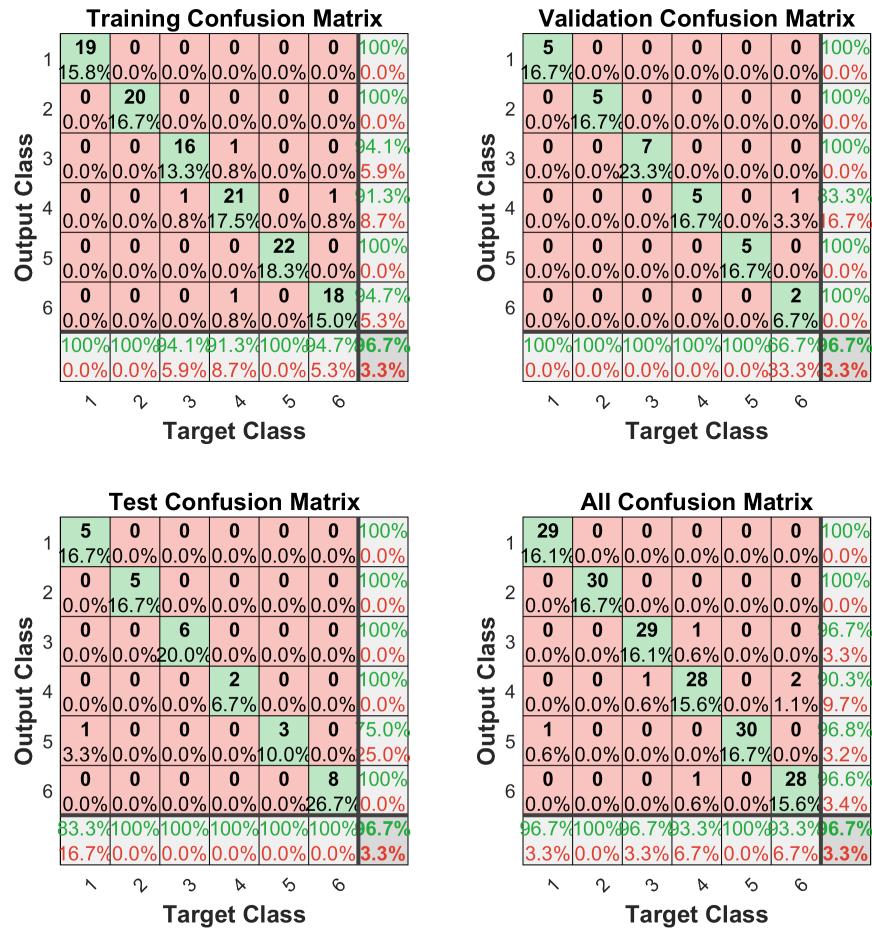


Figura 43: Resultados obtenidos con RN para clasificación de 6 clases, 7 características. Sujeto 1: *female_1.mat*.

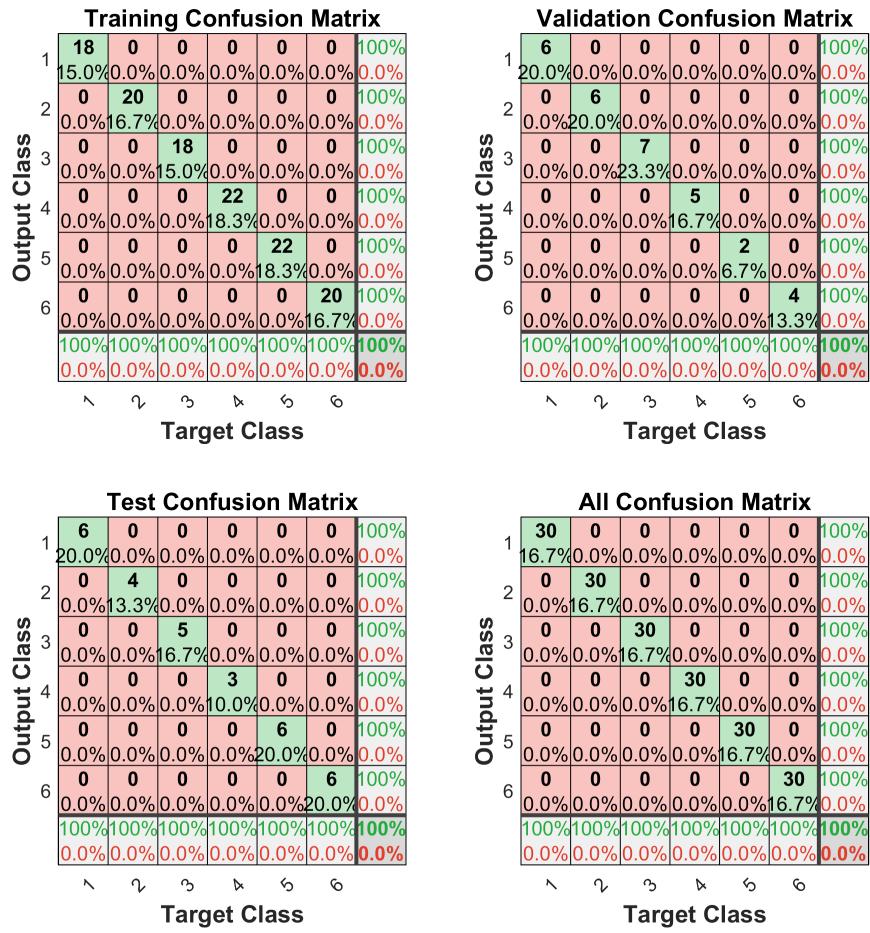


Figura 44: Resultados obtenidos con RN para clasificación de 6 clases, 7 características. Sujeto 4: *male_1.mat*.

En las figuras 45-48 se observa de manera más clara y comparativa el rendimiento de cada tipo de clasificador, según el grupo de características temporales empleadas, para cada sujeto de prueba disponible en la base de datos pública. En el capítulo 16 se recopilan los resultados de clasificación obtenidos tanto con SVM como RN, para los sujetos de prueba que no se detallan en esta sección.

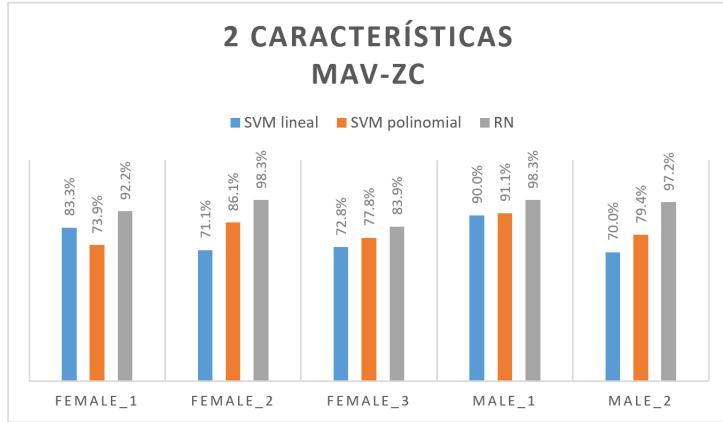


Figura 45: Comparación de resultados entre clasificadores con 2 características.

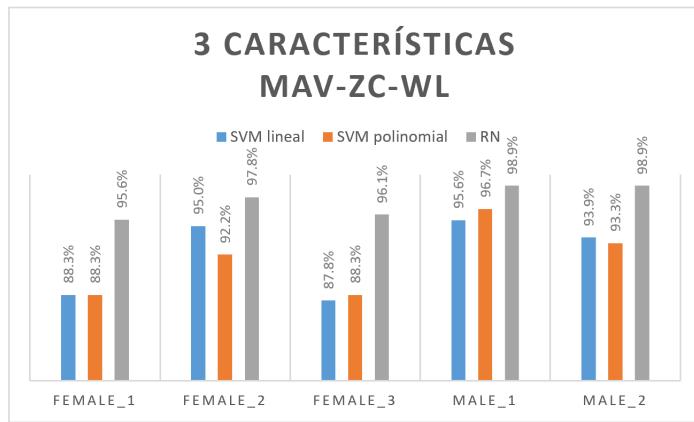


Figura 46: Comparación de resultados entre clasificadores con 3 características.

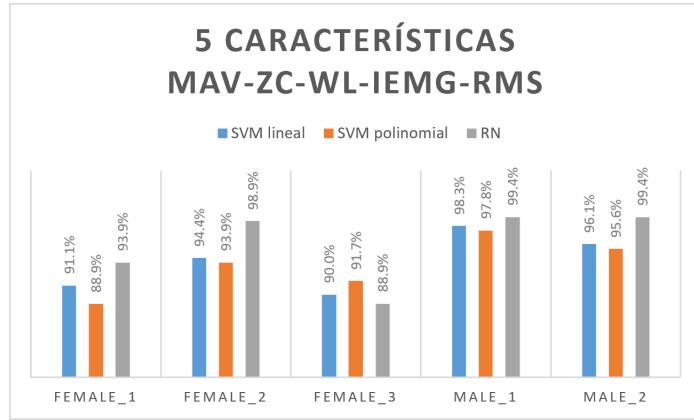


Figura 47: Comparación de resultados entre clasificadores con 5 características.

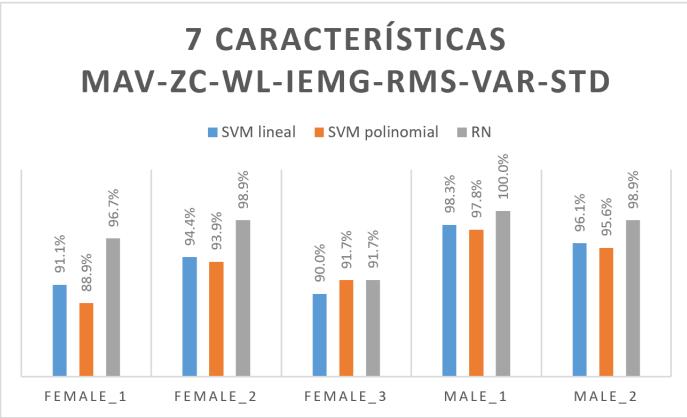


Figura 48: Comparación de resultados entre clasificadores con 7 características.

10.4. Resultados Base de Datos Propia

Se realizaron pruebas con dos tipos de clasificadores, máquinas de vectores de soporte y redes neuronales. Para cada clasificador, se realizaron entrenamientos con distintos sets de características, con el fin de evaluar su rendimiento. Además, para el caso de máquinas de vectores de soporte, se realizaron pruebas de entrenamiento con distintos tipos de kernel (lineal, polinomial y *radial basis function*).

En la figura 49 se recopilan los resultados obtenidos al entrenar cuatro tipos distintos de clasificadores. Para los clasificadores basados en SVM se utilizó la técnica *five fold cross validation*. Los resultados mostrados en 49 corresponden al rendimiento promedio obtenido de realizar diez entrenamientos con cada clasificador. Además, se realizaron pruebas de entrenamiento con diferentes combinaciones y cantidad de características en el dominio del tiempo, la especificación de estos grupos de características de detalla en el cuadro 2.

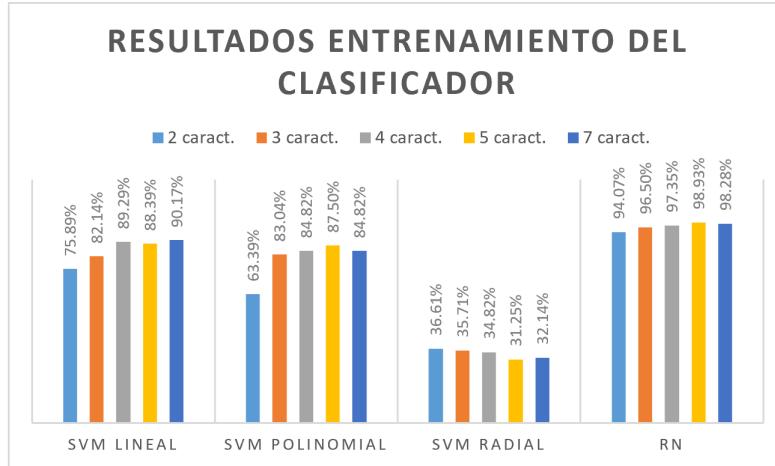


Figura 49: Comparación de resultados de entrenamiento entre distintos tipos de clasificadores para 4 clases.

No. características	Características
2	MAV,ZC
3	MAV,ZC,WL
4	MAV,ZC,WL,RMS
5	MAV,ZC,IEMG,WL,RMS
7	MAV,ZC,IEMG,WL,RMS,VAR,DESV

Cuadro 2: Combinación de características utilizadas para pruebas de entrenamiento de clasificadores.

Los resultados que se observan en la figura 49 muestran que los clasificadores basados en SVM utilizando kernel polinomial y radial, presentan porcentajes de rendimiento menores a los otros clasificadores, sin importar la cantidad de características que se utilice para su entrenamiento. Por lo que se decidió continuar las pruebas de entrenamiento de clasificadores utilizando únicamente los clasificadores basados en SVM con kernel lineal y redes neuronales.

Con el fin de evaluar el rendimiento de los clasificadores, utilizando señales diferentes a las empleadas para entrenamiento, se realizó una serie de grabaciones nuevas en donde se capturaron los movimiento descritos en la figura 22. Cada movimiento se realizó una cantidad de veces especificada, capturando un segundo de la señal. Para cada grabación se mantuvieron las mismas condiciones de filtrado, así como la duración de captura. En el cuadro 3 se detalla la cantidad de muestras totales capturadas en cada grabación, así como el número de muestras por movimiento/clase.

Grabación	No. muestras	Muestras por movimiento
datos_test_1.mat	20	5
datos_test_2.mat	20	5
datos_test_3.mat	20	5
datos_test_4.mat	28	7
datos_test_5.mat	28	7

Cuadro 3: Descripción de las grabaciones utilizadas para evaluar los clasificadores.

Dado que se tenía pleno conocimiento del orden en que se ejecutaron los movimientos, fue posible crear los vectores de etiquetas correspondientes a cada grabación. Con la información ya organizada, se procedió a realizar pruebas de entrenamiento y clasificación. Los resultados obtenidos en las figuras 50 y 53 representan el promedio de realizar diez entrenamientos con cada clasificador (SVM y RN) y evaluar el modelo obtenido utilizando los datos de prueba.

RESULTADOS CON DATOS DE PRUEBA SATISFACTORIOS

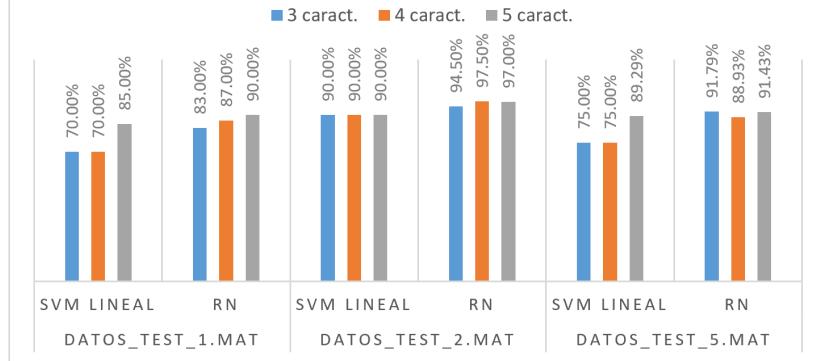


Figura 50: Resultados obtenidos con datos de prueba satisfactorios.

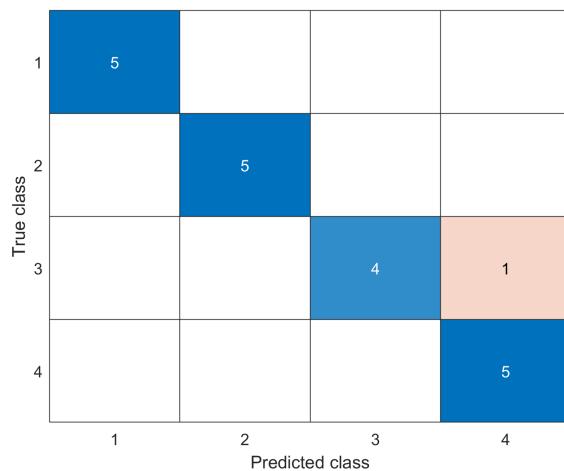


Figura 51: Resultados con RN para grabación ‘datos_test_1’ con 5 características.

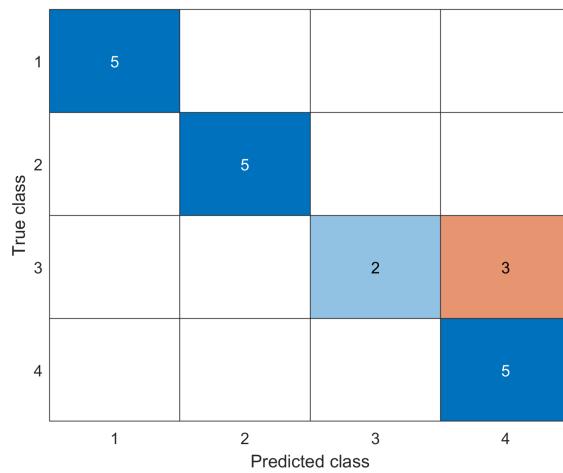


Figura 52: Resultados con SVM para grabación ‘datos_test_1’ con 5 características.

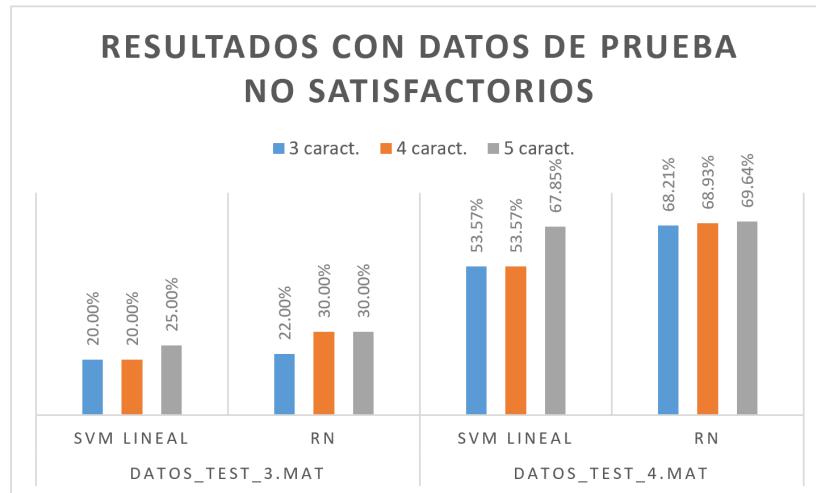


Figura 53: Resultados obtenidos con datos de prueba no satisfactorios.

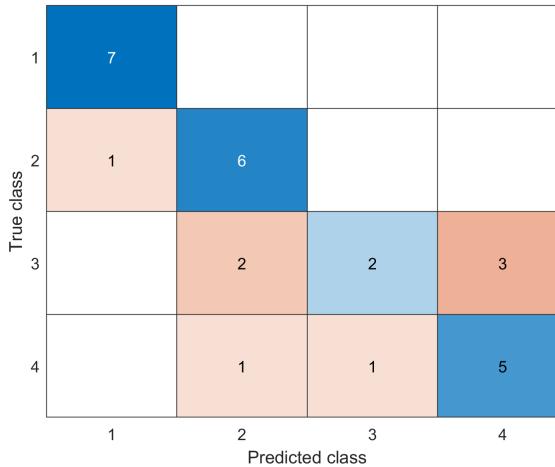


Figura 54: Resultados con RN para grabación ‘datos_test_4’ con 5 características.

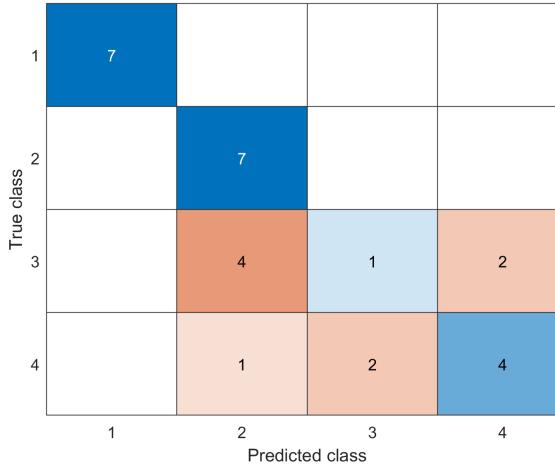


Figura 55: Resultados con SVM para grabación ‘datos_test_4’ con 5 características.

Se observó que con ciertas grabaciones los resultados fueron satisfactorios, presentando rendimientos mayores al 80 %, mientras que otras presentaron resultados poco satisfactorios con rendimientos menores al 70 %. Esta variación en resultados se atribuye principalmente a que las grabaciones no se realizaron en el mismo día, por lo que la posición de los electrodos así como la integridad de los mismos pudo ocasionar discrepancias en las señales capturadas generando que los resultados obtenidos no fueran los esperados.

Se propuso un nuevo método para realizar pruebas de entrenamiento de clasificadores, en este método la información de las señales provenientes de cada canal se dividió en dos secciones, al momento de realizar el proceso de extracción de características. Dando como resultado el doble de vectores de características a utilizar como entrada para el entrenamiento. Por ejemplo, si se utilizan dos características, anteriormente se tendrían 4 vectores de características (2 por cada canal), para este nuevo método el número se duplica resultando en un total de 8 vectores de características (4 por cada canal).

Al subdividir las señales se busca mejorar el rendimiento, ya que dada la naturaleza de este tipo de señales puede ocurrir el hecho de que sus características más distintivas se encuentren en una sección específica de la señal. Por lo que al aplicar este nuevo método se aprovecha mejor la información proveniente de las características para diferenciar entre distintas clases. En las figuras 56 y 59 se muestran los resultados del promedio de realizar diez entrenamientos con cada clasificador (SVM y RN) y evaluar el modelo obtenido utilizando los datos de prueba detallados en el cuadro 3.

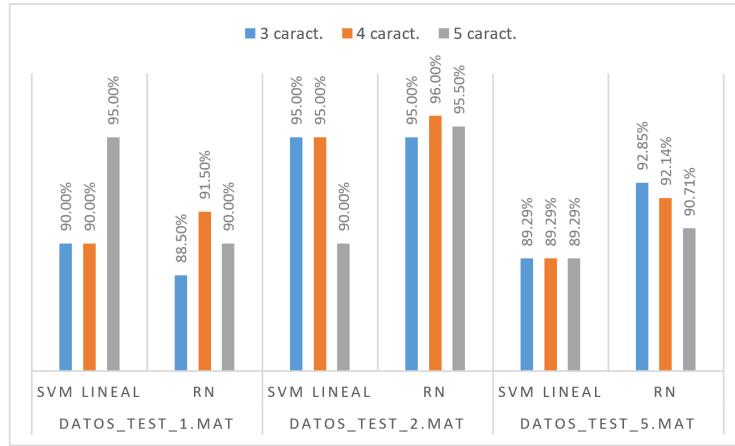


Figura 56: Resultados obtenidos con datos de prueba satisfactorios, dividiendo las señales en dos secciones.

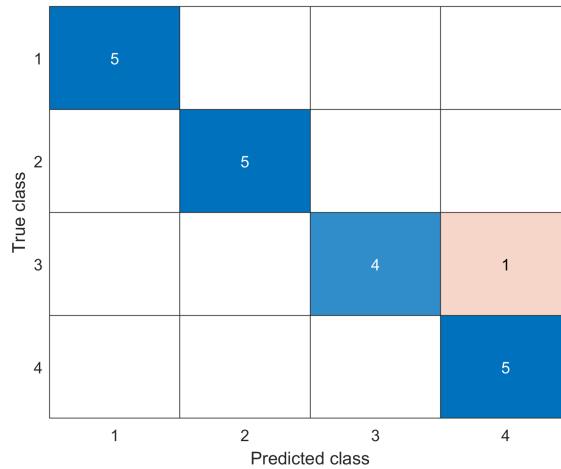


Figura 57: Resultados con RN para grabación ‘datos_test_1’ con 5 características dividiendo las señales.

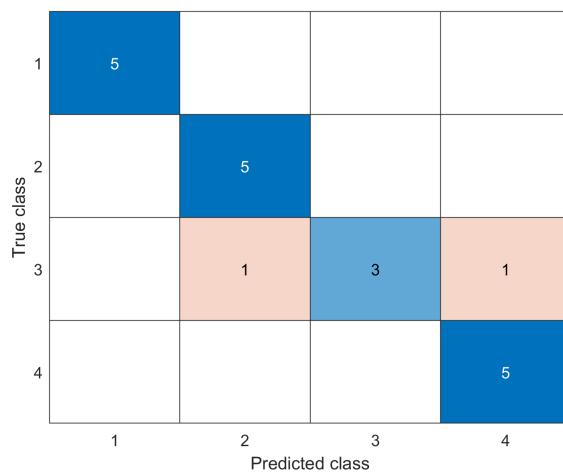


Figura 58: Resultados con SVM para grabación ‘datos_test_1’ con 5 características dividiendo las señales.

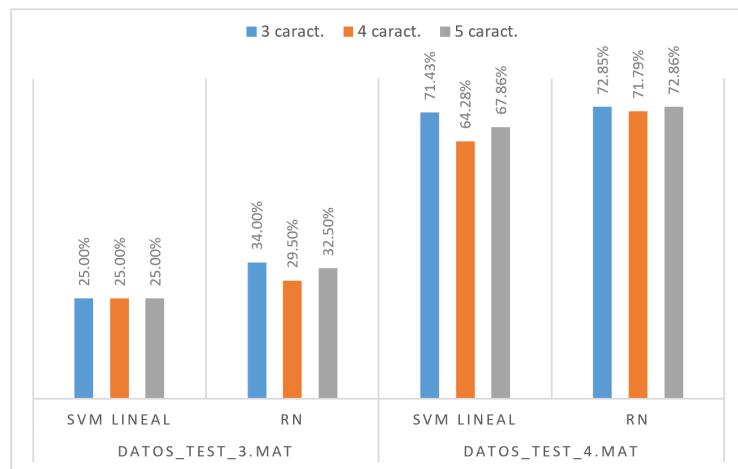


Figura 59: Resultados obtenidos con datos de prueba no satisfactorios, dividiendo las señales en dos secciones.

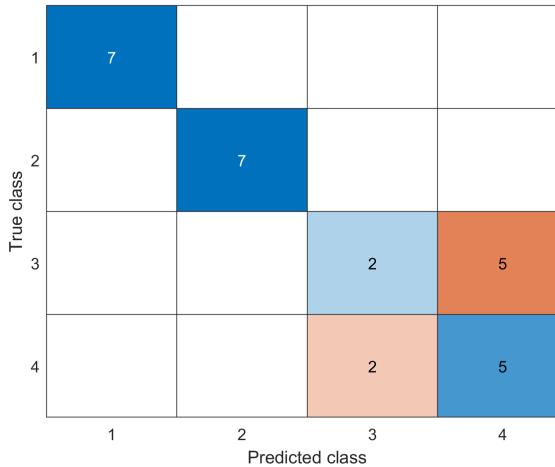


Figura 60: Resultados con RN para grabación ‘datos_test_4’ con 5 características dividiendo las señales.

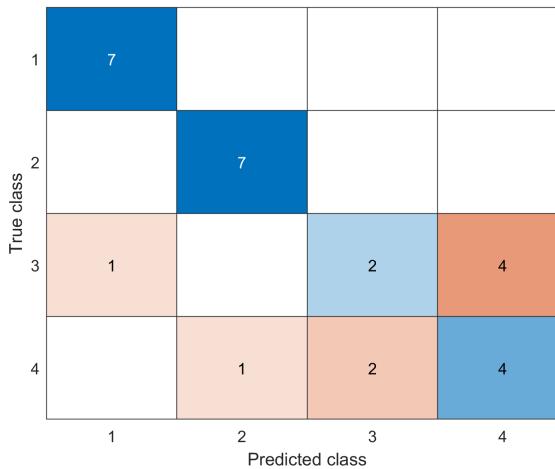
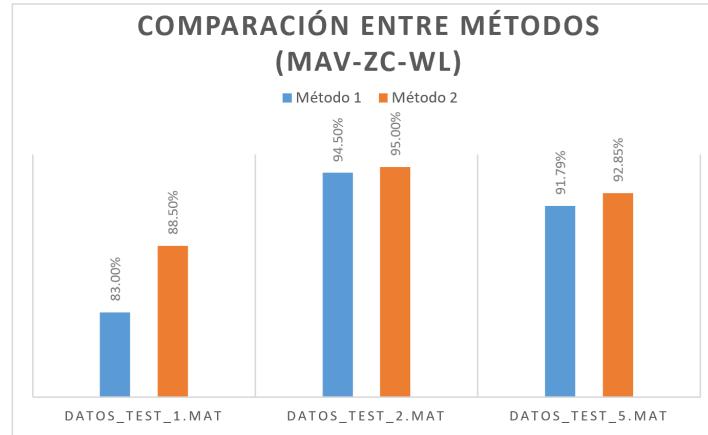


Figura 61: Resultados con SVM para grabación ‘datos_test_4’ con 5 características dividiendo las señales.

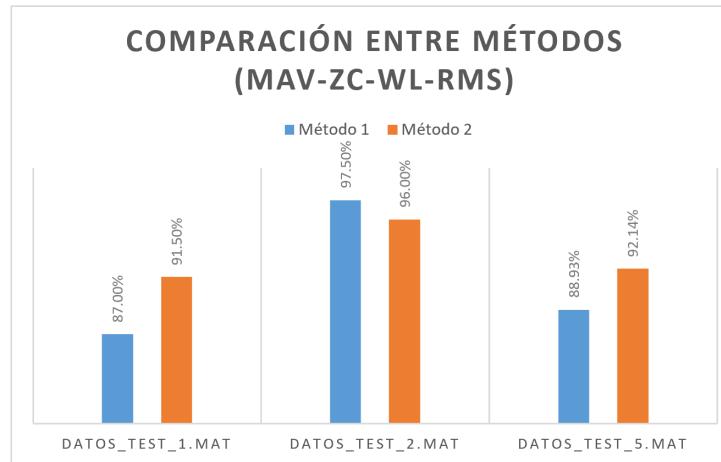
De los resultados presentados anteriormente en 49, 50 y 53 se determinó que los clasificadores basados en redes neuronales presenta un mejor rendimiento que los clasificadores basados en máquinas de vectores de soporte, sin importar el grupo de características empleado para realizar el entrenamiento. Por lo que se decidió utilizar redes neuronales para el modelo final del clasificador que se empleará posteriormente para el desarrollo de la interfaz biomédica. Además, al comparar los resultados presentados en las figuras 50 y 53 con los presentados en las figuras 56 y 59 se observa un claro aumento en el porcentaje de rendimiento al dividir las señales en dos secciones.

Esta comparación es más evidente en la figura 62 en donde se detallan los resultados únicamente del clasificador utilizando redes neuronales. El método 1 corresponde al entrenamiento sin divisiones, mientras el método 2 corresponde al entrenamiento dividiendo las

señales en dos secciones. Para estos resultados se tomaron solo las grabaciones de prueba que presentaron resultados satisfactorios. Concluyendo en que el uso del segundo método es más conveniente para este proyecto, al otorgar un mejor rendimiento al momento de distinguir entre clases.



(a) Tres características



(b) Cuatro características

Figura 62: Comparación entre métodos utilizando redes neuronales.

Por último, para determinar qué características emplear para encontrar el modelo final del clasificador, se recopilaron en el cuadro 4 los resultados obtenidos al utilizar el método 2 para entrenar el clasificador con redes neuronales un total de diez veces y evaluar el modelo obtenido utilizando los datos de prueba almacenados en cada grabación, para cada grupo de características especificado. Cabe mencionar que se muestran únicamente los resultados de las grabaciones que resultaron satisfactorias.

Grabación	MAV-ZC-WL	MAV-ZC-WL-RMS	MAV-ZC-WL-RMS-IEMG
datos_test_1.mat	88.50 %	91.50 %	90.00 %
datos_test_2.mat	95.00 %	96.00 %	95.50 %
datos_test_5.mat	92.85 %	92.14 %	90.71 %

Cuadro 4: Comparación de resultados de entrenamiento con diferentes combinaciones y cantidad de características temporales.

Basados en los resultados mostrados en el cuadro 4, se llegó a la conclusión de emplear 4 características (MAV-ZC-WL-RMS), para armar la matriz de características de entrada necesaria para entrenar el clasificador. Se optó por este grupo de características ya que presenta un mayor porcentaje de rendimiento en comparación a los otros grupos.

10.5. Resultados Finales

Analizando los resultados obtenidos con la base de datos propia que se discutieron en la sección anterior. Se concluyó utilizar un modelo basado en redes neuronales, implementar el segundo método de extracción de características en donde las señales se subdividen en dos secciones y extraer un total de 4 características temporales (MAV,ZC,WL y RMS).

Para que el clasificador presente un buen rendimiento, es necesario conseguir un equilibrio entre el número de muestras utilizadas para entrenamiento y el número de características a extraer. En este proyecto el número de señales que se disponía para realizar el entrenamiento de clasificadores fue limitado, por lo que para mantener este equilibrio se determinó utilizar únicamente las 4 características temporales mencionadas anteriormente.

El entrenamiento de la red final se realizó mediante la *Neuronal Network App* de Matlab, cuyo modelo resultante se almacenó en un archivo .mat para su posterior uso en la interfaz biomédica. Los resultados del entrenamiento de la red se detallan mediante matrices de confusión en la figura 63. Donde se puede observar que el modelo brinda un 98.1% de rendimiento, valor que cumple con las expectativas esperadas para este proyecto.

Es importante mencionar que los datos utilizados para entrenar el clasificador son los almacenados en la base de datos propia cuyo contenido se especifica en el capítulo 9.2.



Figura 63: Resultados con RN para clasificación de 4 clases con 4 características (MAV,ZC,WL y RMS).

CAPÍTULO 11

Desarrollo de Interfaz para el Control de Dispositivos Robóticos

Con el objetivo de desarrollar una interfaz que permita la traducción de los resultados del clasificador en comandos que puedan ser interpretados y realizados por un sistema robótico, se utilizó la herramienta GUIDE de Matlab junto a la *Robotic Toolbox* de Peter Corke, para simular el brazo robótico R17 situado en el departamento de Ingeniería Electrónica, Mecatrónica y Biomédica de la universidad del Valle de Guatemala.

11.1. *Robotics Toolbox*

La *Robotics Toolbox* es un herramienta desarrollada por Peter Corke que provee una variedad de funciones orientadas al estudio y simulación de sistemas robóticos, que requiere específicamente del entorno de Matlab para ejecutarse. Esta herramienta se utilizó para el desarrollo de la simulación del manipulador serial de 5 grados de libertad (GDL) más carrera R17.



Figura 64: Manipulador serial R17 de 5 GDL más carrera [27].

Como paso inicial se descargó e instaló el *toolbox* que se encuentra en [22]. Al finalizar este proceso se inició con la creación del manipulador R17, para ello se definió cada una de las juntas del robot bajo la convención de Denavit-Hartenberg empleando objetos de tipo **Link**. Para luego unirlos y obtener el modelo final del robot de tipo **SerialLink**. En este

caso específico el robot R17 cuenta con 5 juntas revolutas, definidas mediante la función Revolute. Y una junta prismática que representa la carrera y se define mediante la función Prismatic.

```
%> Creación Manipulador Serial R17
%Dimesiones robot
a1 = 0; a2 = 0; a3 = 375; a4 = 375; a5 = 0; a6 = 0;
d2 = -355; d3 = 0; d4 = 0; d5 = 0; d6 = 0;
alpha1 = -pi/2; alpha2 = -pi/2; alpha3 = 0; alpha4 = 0; alpha5 = -pi/2; alpha6 = 0;
theta1 = 0; theta2 = pi/2; theta3 = pi/2; theta4 = 0; theta5 = -pi/2; theta6 = pi/2;
q0 = zeros(1,6);

% Definición de links y creación del robot como objeto SerialLink
L1 = Prismatic('a', a1, 'alpha', alpha1, 'theta',0,'offset',theta1);
L1.qlim = [0,625/2]; %Topes de la primera junta
L2 = Revolute('d', d2, 'a', a2, 'alpha', alpha2, 'offset', theta2);
L3 = Revolute('d', d3, 'a', a3, 'alpha', alpha3, 'offset', theta3);
L4 = Revolute('d', d4, 'a', a4, 'alpha', alpha4, 'offset', theta4);
L5 = Revolute('d', d5, 'a', a5, 'alpha', alpha5, 'offset', theta5);
L6 = Revolute('d', d6, 'a', a6, 'alpha', alpha6, 'offset', theta6);
R17 = SerialLink([L1,L2,L3,L4,L5,L6], 'name', 'R17');

% Transformación de base (rotación -pi/2 en x)
R17.base = transl(0, 0, d2)*trotx(-pi/2);
```

Figura 65: Código para definir el manipulador R17 como objeto *SerialLink*.

Posteriormente, al tener ya definido el manipulador se utilizó la función **teach** que toma como argumentos el nombre del objeto tipo *SerialLink* y el vector de configuración **q**, el cual reúne los parámetros necesarios para describir la configuración del manipulador R17. Con esta función se generó una versión interactiva del robot que permite observar su comportamiento al cambiar los valores de la configuración de forma gráfica, como se observa en la figura 66, esto con el fin de visualizar los movimientos que se quiere que ejecute el robot R17 y posteriormente establecer los valores para el vector de configuración según cada clase.

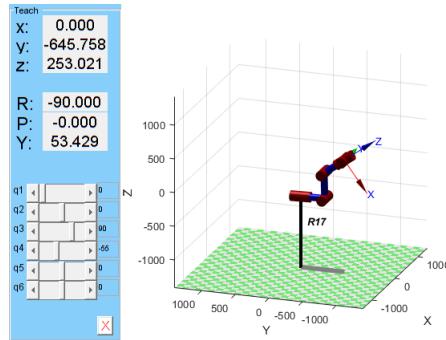


Figura 66: Versión interactiva del manipulador R17.

Adicionalmente, se definieron dos sistemas robóticos más utilizando la *Robotics Toolbox*. El primero fue el robot SCARA (por sus siglas en inglés *Selective Compliant Assembly Robot Arm*) que cuenta con 4 grados de libertad, dados por 3 juntas revolutas y una junta prismática que representa el posicionamiento horizontal del robot. El segundo fue un sistema robótico más simple compuesto únicamente por dos juntas revolutas. El objetivo de crear sistemas robóticos más simples radica en brindar más opciones al usuario en donde se observe con mayor claridad el comportamiento del robot.

CAPÍTULO 12

Simulaciones Finales

Se desarrollaron dos simulaciones que sirvieran como prueba de concepto y del funcionamiento del sistema final en donde se engloban todos los temas desarrollados en capítulos anteriores. En la figura 67 se resume el proceso que realiza el sistema final para controlar dispositivos robóticos usando señales EMG. Primero, el sistema es capaz de adquirir y filtrar señales electromiográficas en tiempo real. Posteriormente, se realiza el proceso de extracción de características, que se utilizan como entrada para el clasificador previamente entrenado. Por último, en base a los resultados de la clasificación el sistema es capaz de traducir estos resultados a comandos que son enviados mediante la interfaz a un sistema robótico que ejecuta el movimiento establecido según la clase identificada por el clasificador. La idea central de estas simulaciones radica en demostrar que el sistema es capaz de adquirir y clasificar señales electromiográficas en tiempo real, para controlar el movimiento de dispositivos robóticos según los resultados de la clasificación.

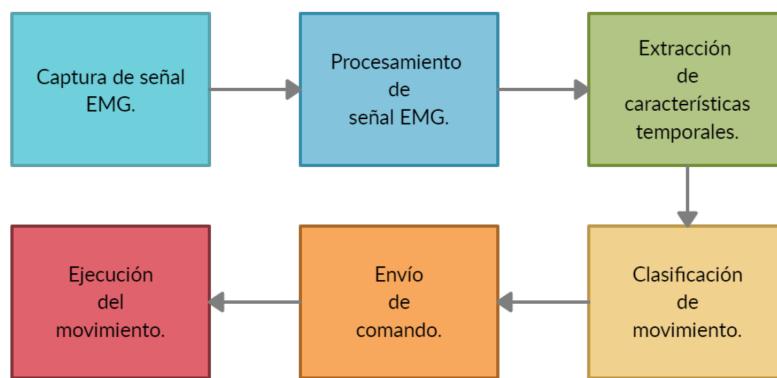


Figura 67: Funcionamiento del sistema final.

En este capítulo se detallan los resultados de las dos simulaciones. La primera utilizando la información contenida en la base de datos pública que se menciona en la sección 9.1 y la

segunda utilizando los datos de la base de datos propia que se detalla en la sección 9.2.

12.1. Simulación Usando la Base de Datos Pública

Se desarrolló una simulación en Matlab utilizando la información recopilada en la base de datos pública mencionada con anterioridad. En este programa se despliega una interfaz gráfica con dos ventanas, que permite colocar la simulación en modo manual o automático. En modo manual el usuario cuenta con la opción de escoger entre tres modelos de dispositivos robóticos y observar la acción que será ejecutada por el mismo según cada tipo de agarre almacenado en las señales de la base de datos pública. Además, se brinda la opción de desplegar o no en pantalla las características extraídas de las señales al activar el modo automático.

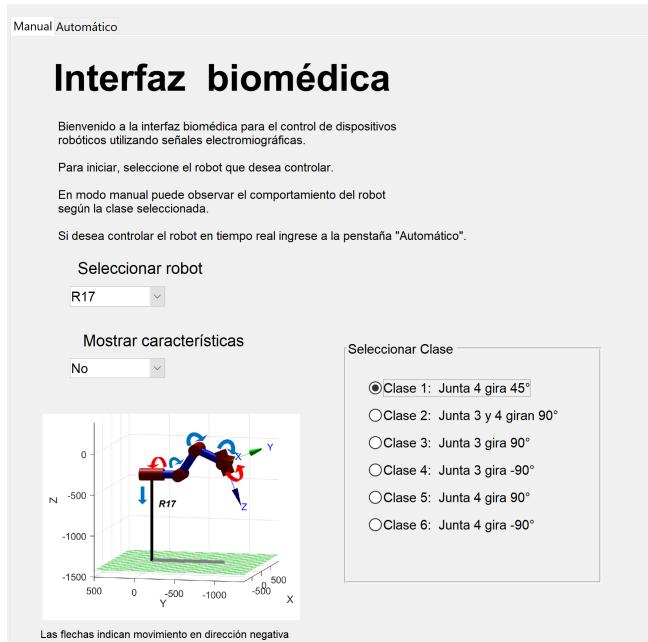


Figura 68: Interfaz en modo manual.

Por otra parte, en modo automático el programa despliega continuamente la información de las señales almacenadas en la base de datos, simulando que estas se están adquiriendo en tiempo real. Cabe mencionar que las señales de la base de datos ya están filtradas, por lo que no es necesario ejecutar la etapa de filtrado para esta simulación. Mientras se leen los datos, el programa se encarga de extraer el vector de características representativo de cada movimiento, el cual es utilizado como entrada en un clasificador previamente entrenado con señales de la base de datos. Según el resultado de dicha clasificación el dispositivo robótico ejecutará el movimiento establecido y se desplegará el resultado obtenido.

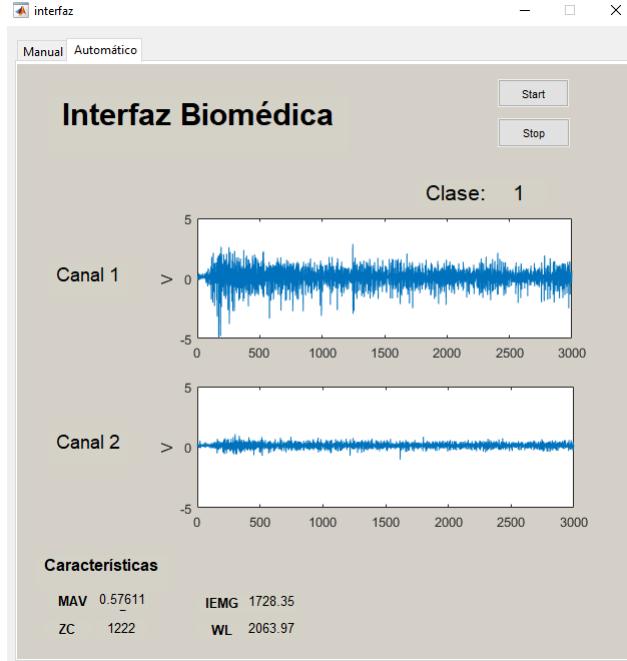
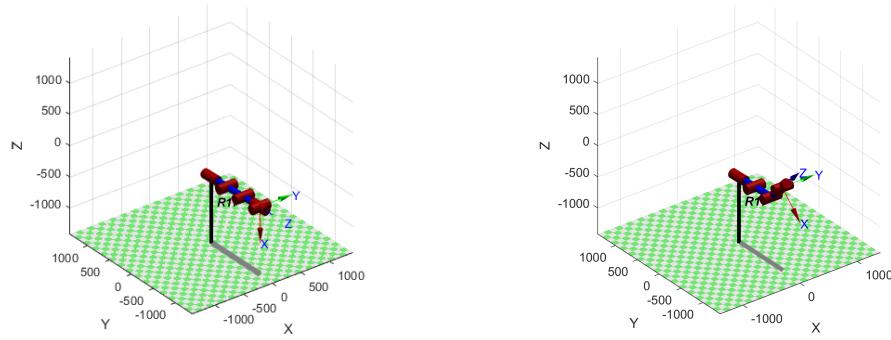


Figura 69: Interfaz en modo automático.

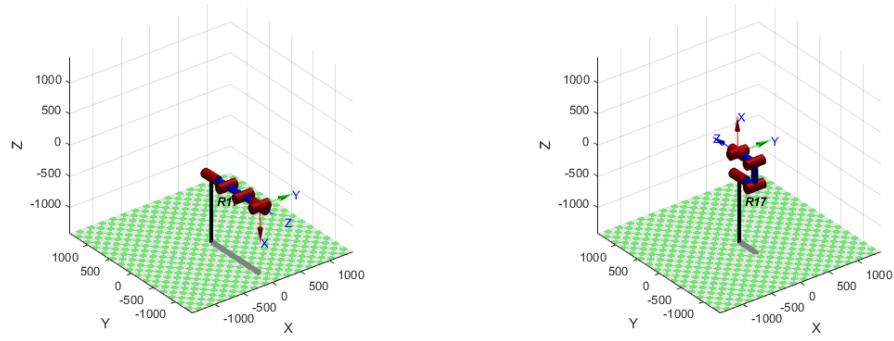
A continuación se muestran los resultados obtenidos al utilizar la interfaz para el control del manipulador R17. Como se mencionó anteriormente, la base de datos pública cuenta con seis clasificaciones correspondientes a seis diferentes tipos de agarre. Para cada clase se estableció un movimiento específico iniciando desde el origen, los movimientos realizados por el manipulador según el resultado de la clasificación se ilustran en las figuras 70 - 75.



(a) Posición inicial.

(b) Posición final.

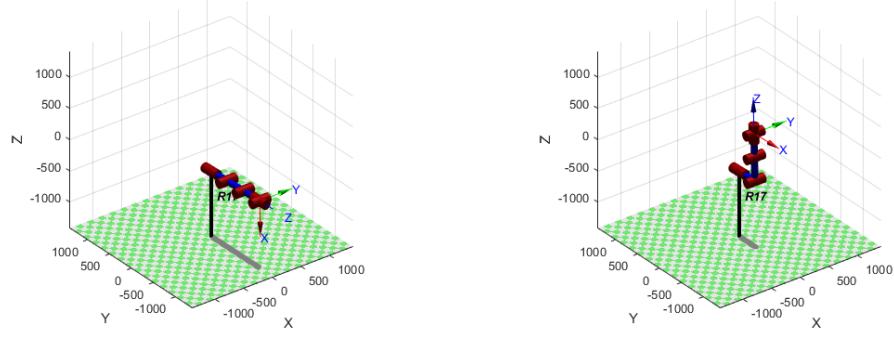
Figura 70: Clase 1: Junta 4 gira 45°.



(a) Posición inicial.

(b) Posición final.

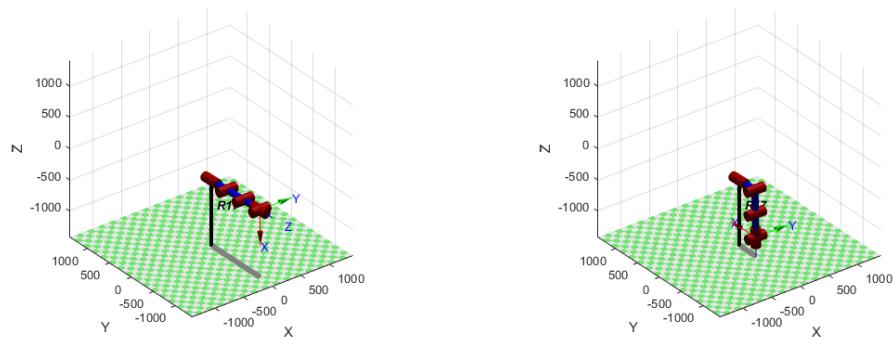
Figura 71: Clase 2: Junta 3 y junta 4 giran 90° .



(a) Posición inicial.

(b) Posición final.

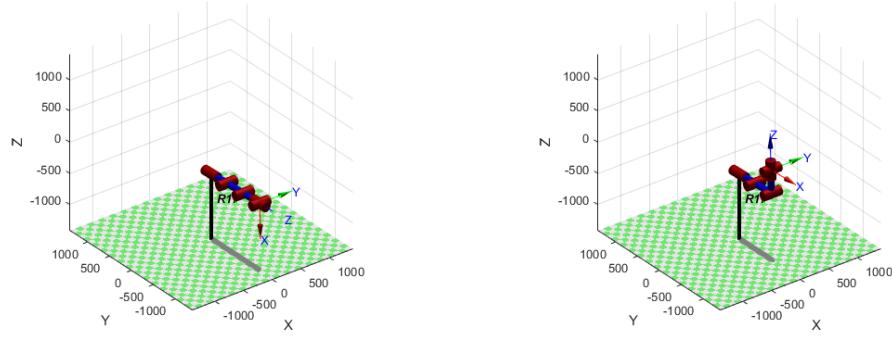
Figura 72: Clase 3: Junta 3 gira 90° .



(a) Posición inicial.

(b) Posición final.

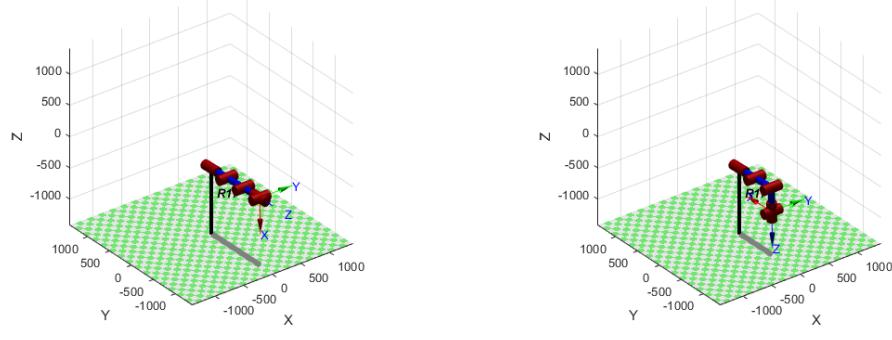
Figura 73: Clase 4: Junta 3 gira -90° .



(a) Posición inicial.

(b) Posición final.

Figura 74: Clase 5: Junta 4 gira 90° .



(a) Posición inicial.

(b) Posición final.

Figura 75: Clase 6: Junta 4 gira -90° .

12.2. Simulación con Datos Obtenidos en Tiempo Real

Esta simulación tuvo como fin englobar todos los temas desarrollados en los capítulos anteriores para demostrar el funcionamiento final del sistema. Inicialmente el sistema despliega una interfaz gráfica con dos ventanas, que permite colocar la simulación en modo manual o automático. En modo manual, figura 76, el usuario cuenta con la opción de escoger entre tres modelos de dispositivos robóticos y observar la acción que será ejecutada por el mismo según cada clase. Mientras en modo automático, figura 77, se despliega continuamente en pantalla la información que se está adquiriendo en tiempo real.

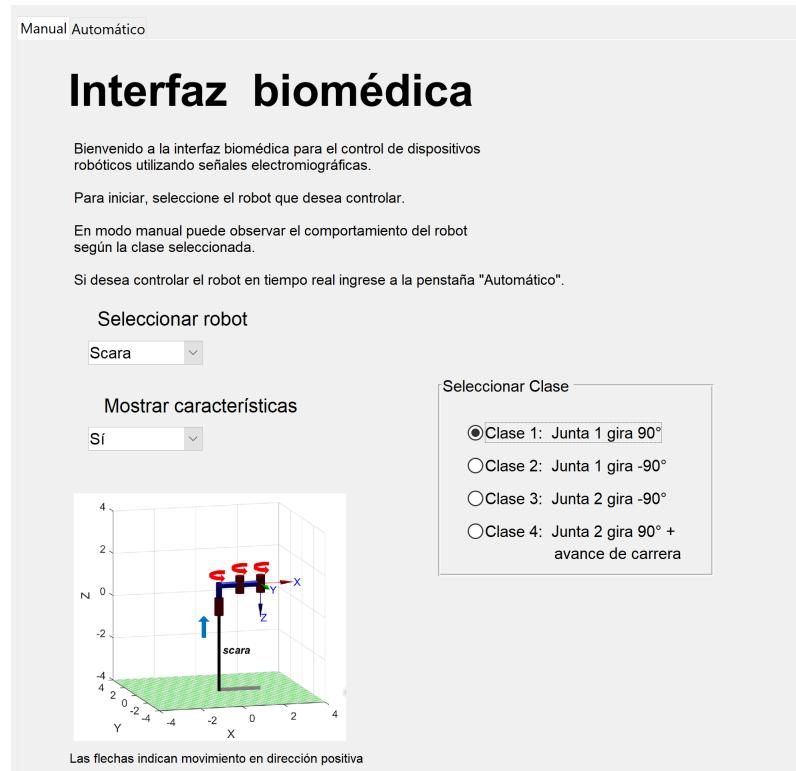


Figura 76: Interfaz en modo manual.

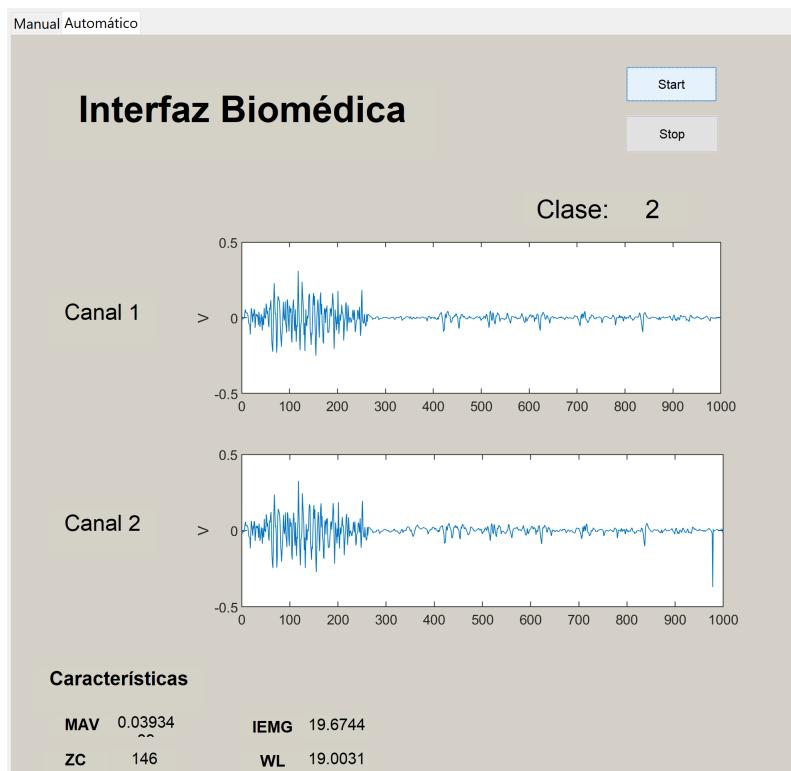


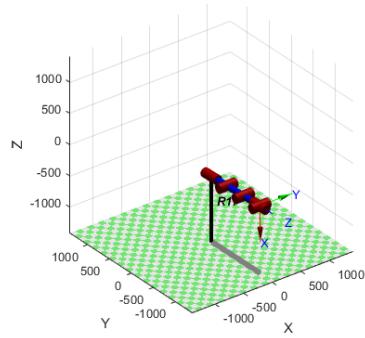
Figura 77: Interfaz en modo automático.

Mientras se adquieren los datos, el sistema es capaz filtrar las señales EMG y extraer las características representativas de cada movimiento, las cuales se utilizan como entrada para el clasificador previamente entrenado que se menciona en la sección 10.5. Según el resultado de dicha clasificación el dispositivo robótico ejecutará el movimiento establecido. Cabe mencionar que en esta simulación a las señales adquiridas en tiempo real se les aplica un filtro pasa banda tipo *Butterworth* con frecuencias de corte de 20 y 450 Hz, seguido por un filtro quita banda tipo *Butterworth* con frecuencias de corte de 58 y 61 Hz.

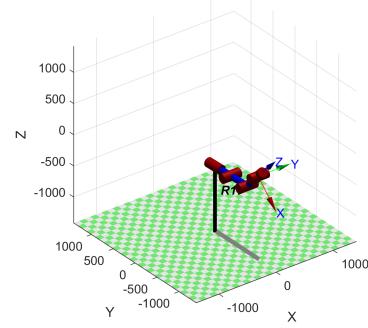
En las figuras que se presentan a continuación se muestra un ejemplo de los movimientos que ejecuta uno de los dispositivos robóticos disponibles en la simulación, al interpretar los resultados provenientes de la clasificación.



(a) Empuñar.



(b) Posición inicial.

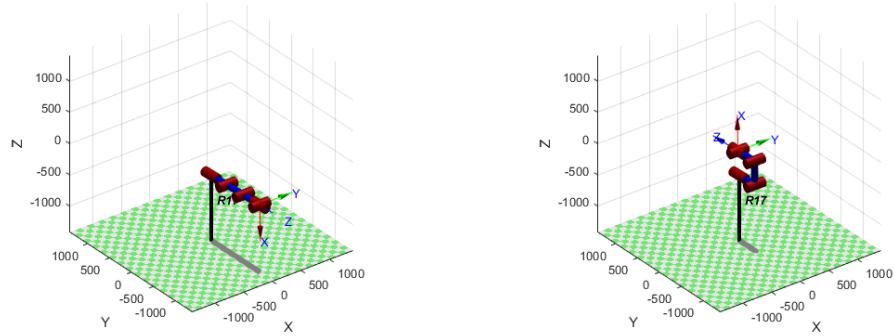


(c) Posición final.

Figura 78: Clase 1: Junta 4 gira 45° .



(a) Arriba.



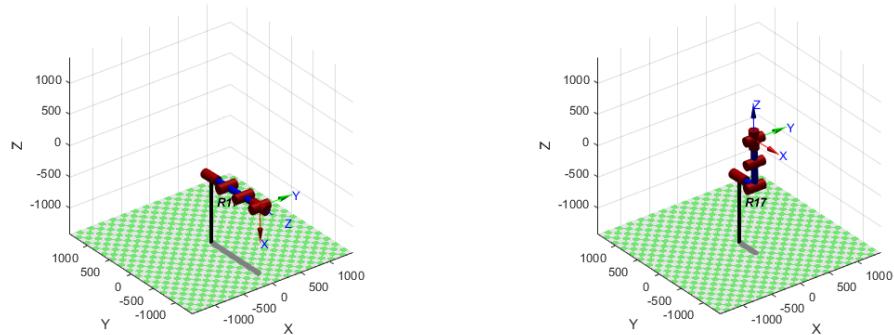
(b) Posición inicial.

(c) Posición final.

Figura 79: Clase 2: Junta 3 y junta 4 giran 90° .



(a) Doble arriba.



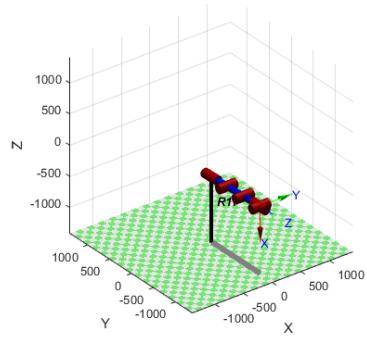
(b) Posición inicial.

(c) Posición final.

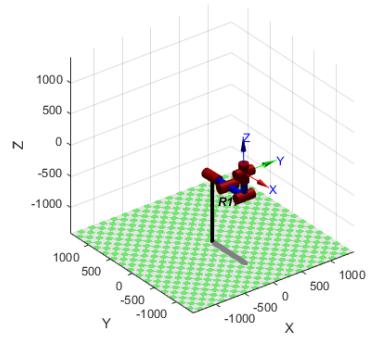
Figura 80: Clase 3: Junta 3 gira 90° .



(a) Movimiento dedos.



(b) Posición inicial.



(c) Posición final.

Figura 81: Clase 4: Junta 4 gira 90°.

CAPÍTULO 13

Conclusiones

- El algoritmo de adquisición de datos que se desarrolló comprobó ser una herramienta eficiente y accesible para la captura de señales electromiográficas de superficie.
- Se comprobó que los filtros IIR tipo *Butterwoth* representan una herramienta eficaz para el procesamiento de señales EMG, permitiendo extraer los espectros de la señal en donde se presenta mayor actividad.
- Las características en el dominio del tiempo utilizadas en este proyecto probaron ser suficientes para entrenar clasificadores con rendimientos mayores al 80 % e incluso alcanzando un 98.93 %.
- Se determinó que el uso de más características temporales no implica siempre un aumento en el rendimiento del clasificador. Para conseguir un buen rendimiento se debe lograr un equilibrio entre el número de muestras disponibles para entrenamiento y el número de características a extraer.
- Se comprobó que para las características temporales seleccionadas el clasificador con redes neuronales es más efectivo y presenta mejor rendimiento que las máquinas de vectores de soporte.
- Se determinó que es posible mejorar el rendimiento de la red neuronal al subdividir la información de ambos canales en dos secciones, duplicando el número de vectores de características utilizados para entrenamiento.
- La simulación final demuestra que el sistema es funcional y capaz de adquirir, filtrar y extraer características temporales de señales EMG, para controlar dispositivos robóticos en base a los resultados de un clasificador previamente entrenado.

CAPÍTULO 14

Recomendaciones

- Aunque el dispositivo Bitalino haya presentado facilidad y resultados satisfactorios al momento de capturar señales electromiográficas, es recomendable considerar el uso de otros dispositivos que permitan llevar a cabo esta tarea y además brinden más y mejores herramientas para el manejo de señales EMG.
- El uso de electrodos de superficie para capturar señales EMG representa una ventaja dada su accesibilidad, facilidad de uso y comodidad. Sin embargo, este tipo de electrodos se deterioran rápidamente con el uso y difícilmente se puede replicar la ubicación exacta de los mismos para cada grabación, afectando la calidad de la señal. Por lo que se recomienda considerar el desarrollo de algún sistema de sujeción o dispositivo que permita preservar lo mejor posible la ubicación de los electrodos y mantener su integridad.
- Como se mencionó en los resultados, el adquirir las señales mediante dos canales contribuye a mejorar el rendimiento del clasificador, por lo que se recomienda considerar el uso de más canales al momento de capturar las señales EMG. Esto, con el fin de determinar si el incorporar más información proveniente de diferentes canales aporta a mejorar aún más el rendimiento del clasificador.
- Para fases futuras se recomienda considerar nuevamente las plataformas de programación y dispositivos utilizados para establecer el intercambio de datos entre el Bitalino y la interfaz biomédica. Dispositivos como el arduino uno, utilizado en este proyecto, pueden presentar limitaciones en cuanto a la velocidad de envío de datos, lo que restringe la frecuencia de muestreo al adquirir la señal y limita el número de datos que se pueden adquirir en una ventana de tiempo razonable para esta aplicación.
- En futuras etapas se recomienda explorar el uso de clasificadores más complejos y considerar incorporar características en el dominio de la frecuencia y tiempo-frecuencia para entrenar el modelo del clasificador, con el fin de aumentar el porcentaje de rendimiento.

CAPÍTULO 15

Bibliografía

- [1] C. Alva, *Procesamiento de Señales de Electromiografía Superficial para la Detección de Movimientos de Dos Dedos de la Mano*. Perú: Universidad Ricardo Palma, 2012.
- [2] J. Brazeiro, S. Petracchia y M. Valdés, *Mano controlada por señales musculares*, Recuperado de: <https://iie.fing.edu.uy/publicaciones/2015/BPV15/BPV15.pdf>, 2015.
- [3] H. Romo, J. Realpe y P. Jojoa, “Análisis de Señales EMG Superficiales y su Aplicación en Control de Prótesis de Mano”, *Revista Avances en Sistemas e Informática*, vol. 4, 2007.
- [4] R. Morales, D. Morales y V. Grisales, *Caracterización de Señales Electromiográficas para la Discriminación de Seis Movimientos de la Mano*. Colombia: Universidad Tecnológica de Pereira, 2009.
- [5] I. Cifuentes, *Extracción de Características y Clasificación de Señales Electromiográficas Utilizando la Transformada Hilbert-Huang y Redes Neuronales*, Recuperado de: <https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/743/1/CifuentesGIA.pdf>, 2012.
- [6] L. Sarmiento, *Interfaces Cerebro-Computadora para el Reconocimiento Automático del Habla Silenciosa*. Colombia: Universidad Pedagógica Nacional, 2019.
- [7] R. Merletti y P. Parker, *Electromyography*. USA: A John Wiley Sons, INC., 2004.
- [8] J. Gárate, *Instrumentación en Electromiografía*, Recuperado de: <http://gecouusb.com.ve>.
- [9] S. Osuna, J. Gonzalez y A. Ilzarbe, *Procesado de Señales EMG en Trastornos Neuromusculares*. Universidad de Barcelona, 2013.
- [10] J. Figueiroa, E. Morales, J. Huerta, J. González y C. Cárdenas, “Sistema de Adquisición de Señales SEMG para la Detección de Fatiga Muscular”, *Revista Mexicana de Ingeniería Biomédica*, vol. 37, 2016.
- [11] E. Morales y H. Escalante, *Evaluación, validación y sobre-ajuste*. Instituto Nacional de Astrofísica, Óptica y Electrónica, 2015.

- [12] J. Brownlee, *A Gentle Introduction to k-fold Cross-Validation*, Recuperado de: <https://machinelearningmastery.com/k-fold-cross-validation/>, 2018.
- [13] R. Flórez y J. Fernández, *Las Redes Neuronales Artificiales*. España: Netbiblo, 2008.
- [14] F. Valle, *Implementación Eficiente de Clasificadores Prior-SVM para Matlab*. Universidad Carlos III de Madrid, 2010.
- [15] J. Muñoz, *Inteligencia Computacional Inspirada en la Vida*. Servicio Publicaciones UMA, 2010.
- [16] J. Jordan, *Support vector machines*, Recuperado de: <https://www.jeremyjordan.me/support-vector-machines/>, 2017.
- [17] G. Virk, S. Moon y R. Gelin, “ISO STANDARDS FOR SERVICE ROBOTS”, ago. de 2008, págs. 133-138, ISBN: 9789812835765. DOI: 10.1142/9789812835772_0016.
- [18] F. Watkins, *Dispositivos Robóticos y su Actuación*. Universidad de Santiago de Chile, 2018.
- [19] J. Medina, “Desarrollo de un Manipulador Didáctico con una Cadena Cinemática Abierta de 6 Grados de Libertad”, Tesis de mtría., Universidad Tecnológica de Pereira, 2016.
- [20] Bitalino, *Whar is BITalino? Is it for you?*, Recuperado de: <https://bitalino.com/en/>.
- [21] Biosignalsplus, *Software: OpenSignals*, Recuperado de: <https://biosignalsplus.com/products/software/opensignals.html>.
- [22] P. Corke, *Robotics, Vision Control*, ISBN 978-3-319-54413-7, 2017.
- [23] E. Pinargote, “Implementación de Filtros Digitales FIR e IIR Utilizando Matlab/Simulink”, Tesis de mtría., Universidad Católica de Santiago de Guayaquil, 2016.
- [24] C. Sapsanis, “Recognition of basic hand movements using Electromyography”, Tesis de mtría., University of Patras, 2013.
- [25] C.-C. Chang y C.-J. Lin, “LIBSVM: A library for support vector machines”, *ACM Transactions on Intelligent Systems and Technology*, vol. 2, 27:1-27:27, 3 2011, Recuperado de: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [26] A. López, J. Rodríguez y A. Fernández, “Control estricto de matrices de confusión por medio de distribuciones multinomiales”, *GeoFocus*, vol. 21, págs. 215-226, 2018, Recuperado de: http://www.miramon.uab.cat/geofocus_ojs/index.php/geofocus/article/view/591.
- [27] *Brazos robóticos de ST Robotics*, Recuperado de: <https://es.rs-online.com/web/generalDisplay.html?id=i%2Fst-robotics>.

CAPÍTULO 16

Anexos

16.1. Resultados Obtenidos con la Base de Datos Pública

Máquinas de Vectores de Soporte: Kernel Lineal

Resultados con 2 características: MAV y ZC.

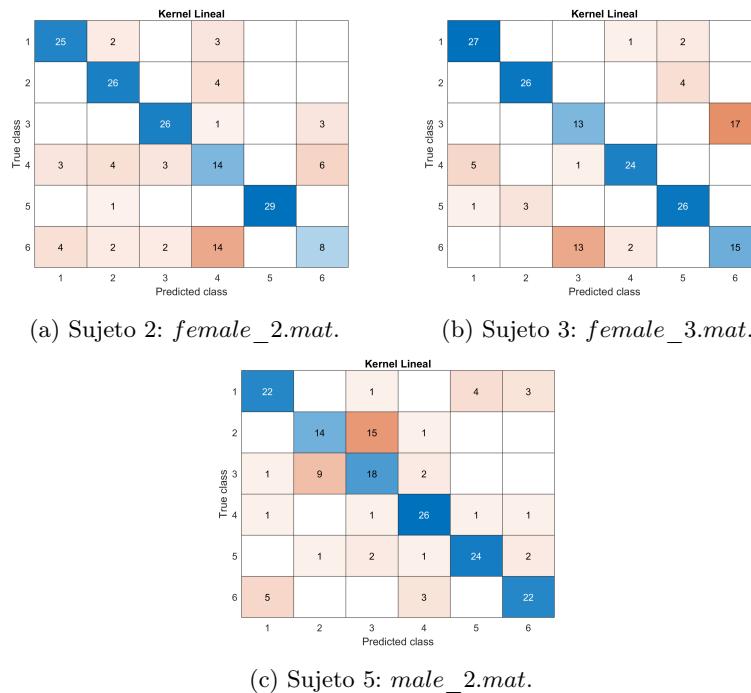


Figura 82: Resultados obtenidos con SVM lineal para clasificación de 6 clases, 2 características.

Resultados con 3 características: MAV, ZC y WL.

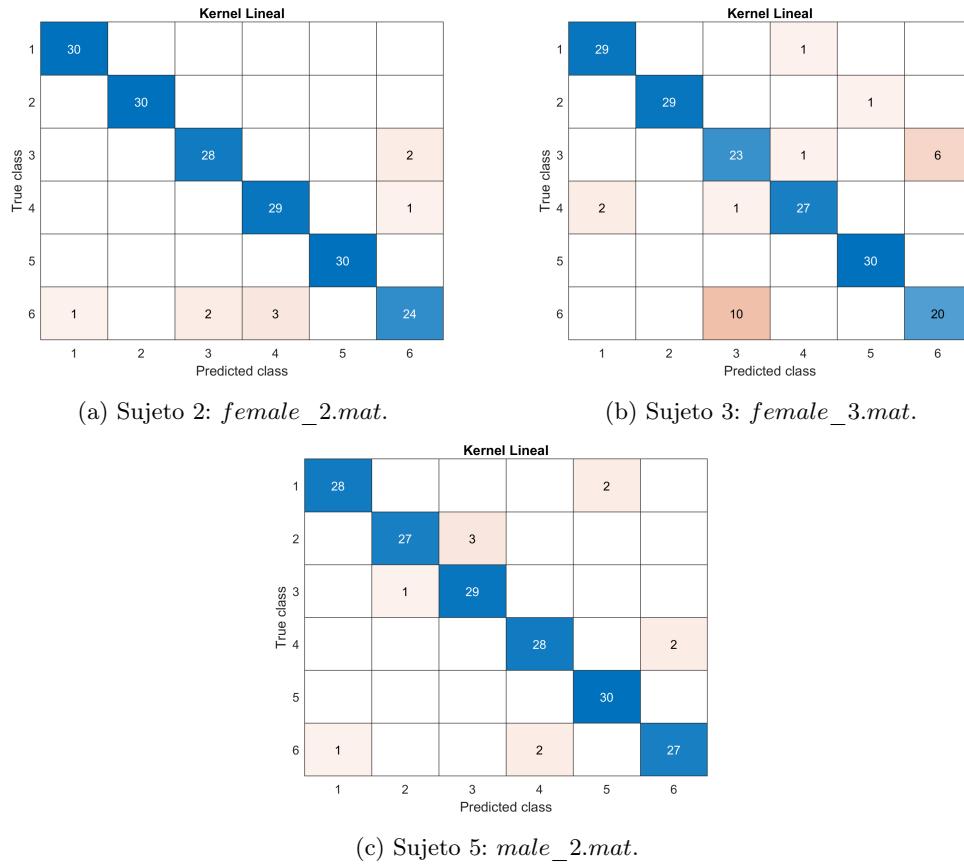


Figura 83: Resultados obtenidos con SVM lineal para clasificación de 6 clases, 3 características.

Resultados con 5 características: MAV, ZC, WL, IEMG y RMS.

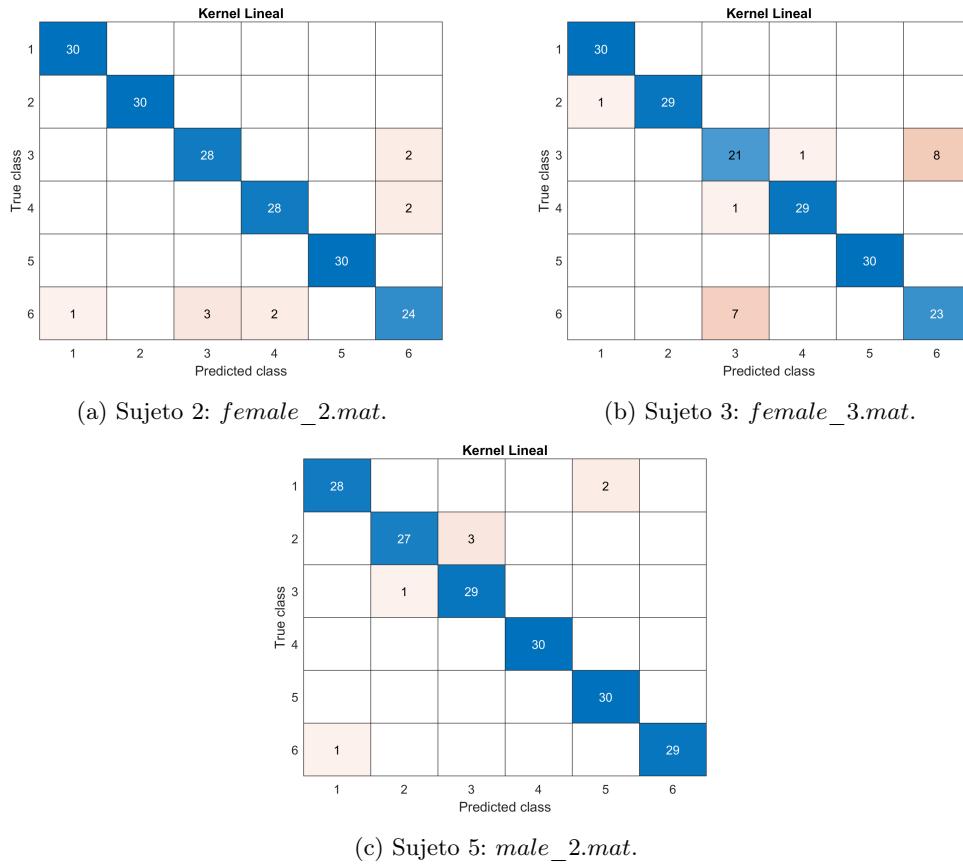


Figura 84: Resultados obtenidos con SVM lineal para clasificación de 6 clases, 5 características.

Resultados con 7 características: MAV, ZC, WL, IEMG, RMS, VAR y STD.

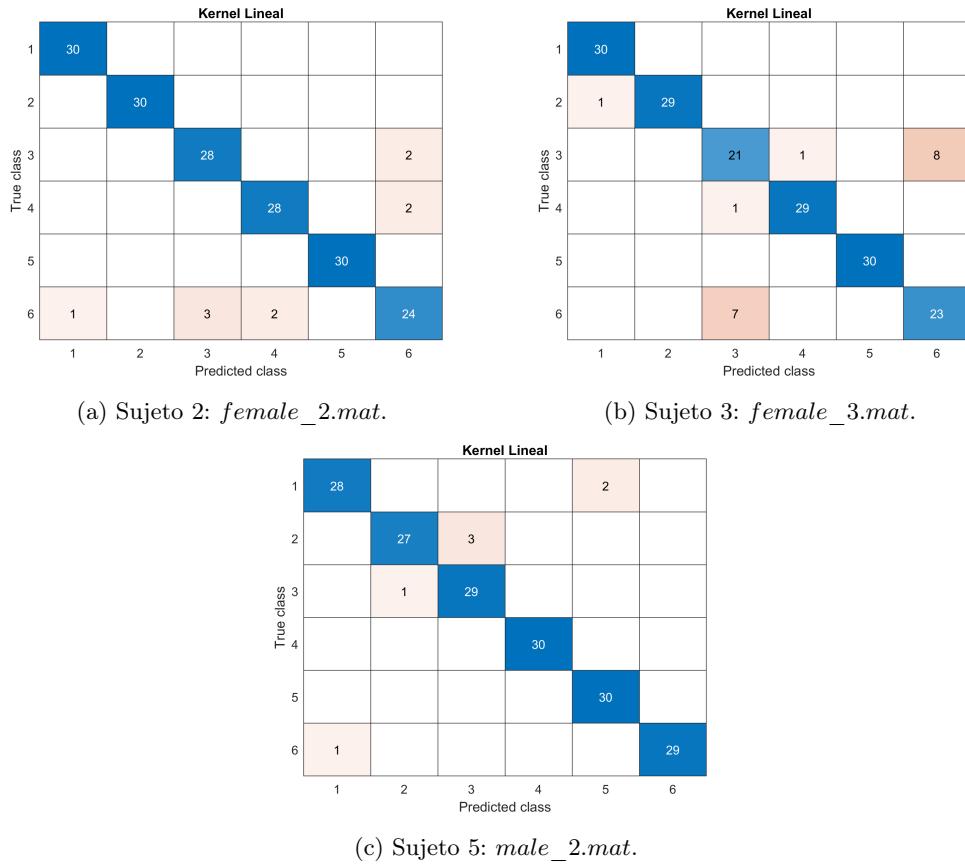


Figura 85: Resultados obtenidos con SVM lineal para clasificación de 6 clases, 7 características.

Máquinas de Vectores de Soporte: Kernel Polinomial

Resultados con 2 características: MAV y ZC.

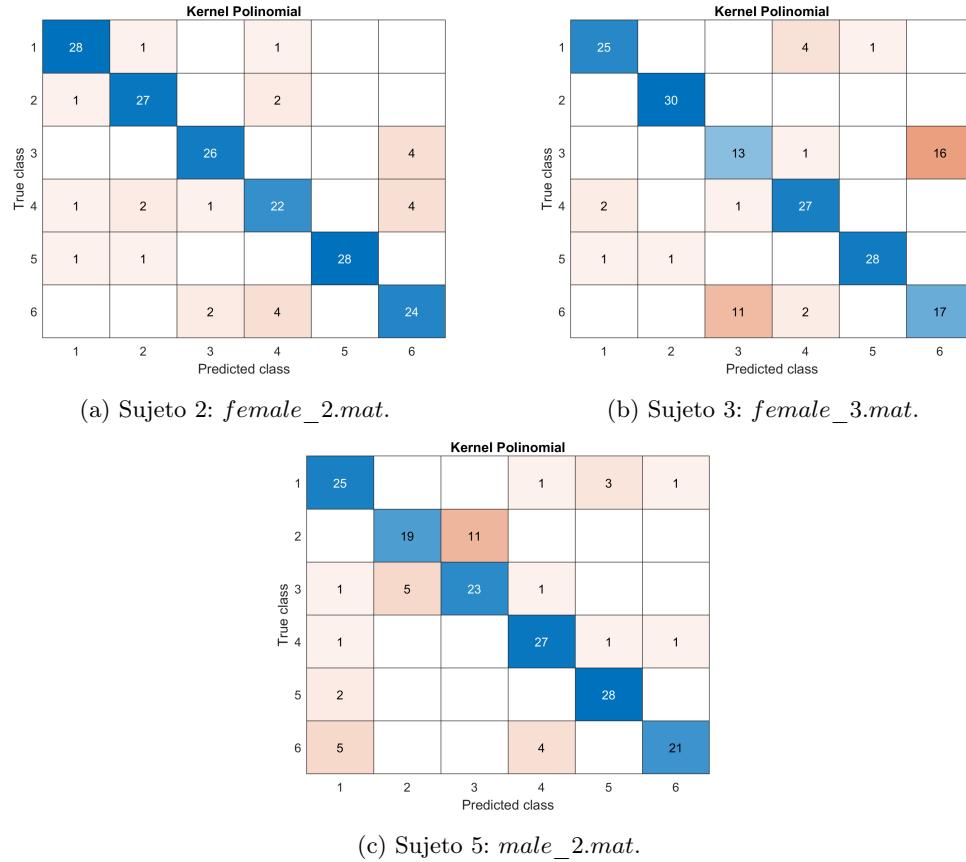


Figura 86: Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 2 características.

Resultados con 3 características: MAV, ZC y WL.

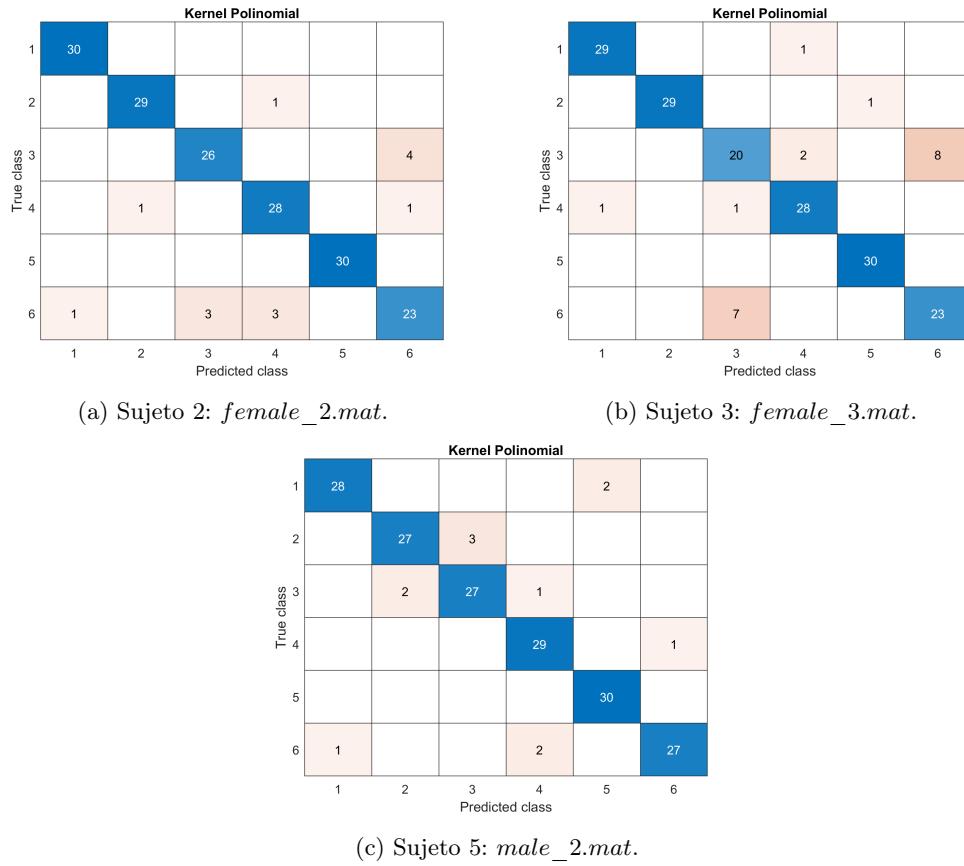


Figura 87: Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 3 características.

Resultados con 5 características: MAV, ZC, WL, IEMG y RMS.

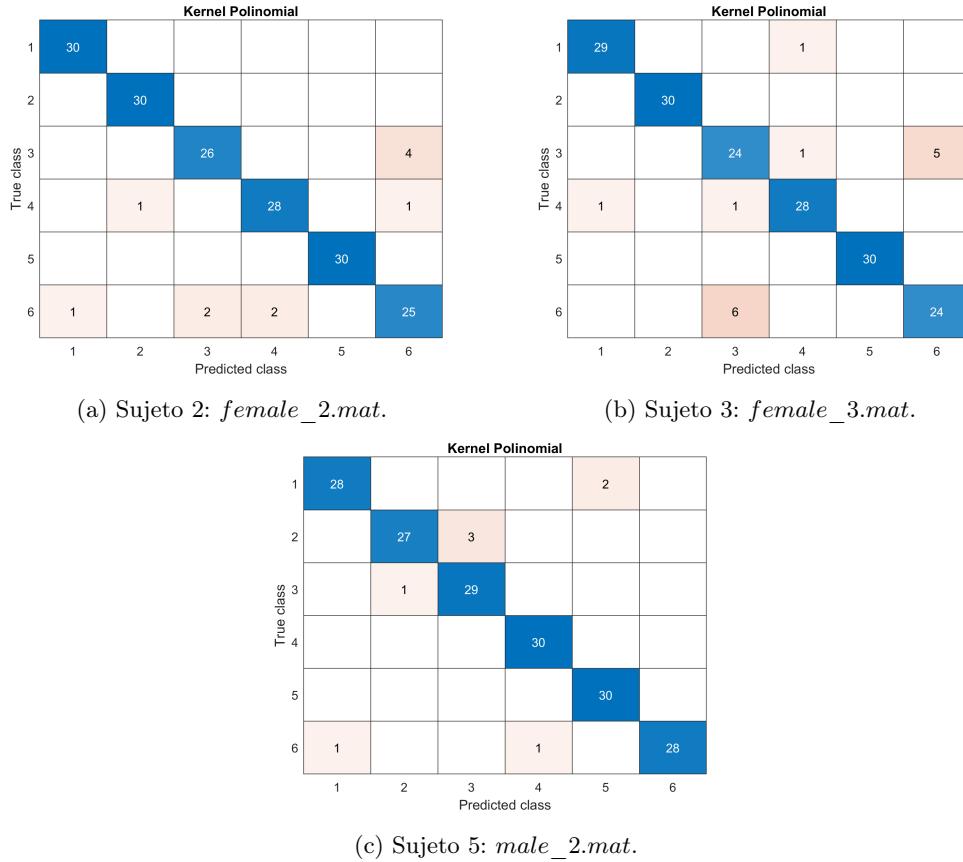


Figura 88: Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 5 características.

Resultados con 7 características: MAV, ZC, WL, IEMG, RMS, VAR y STD.

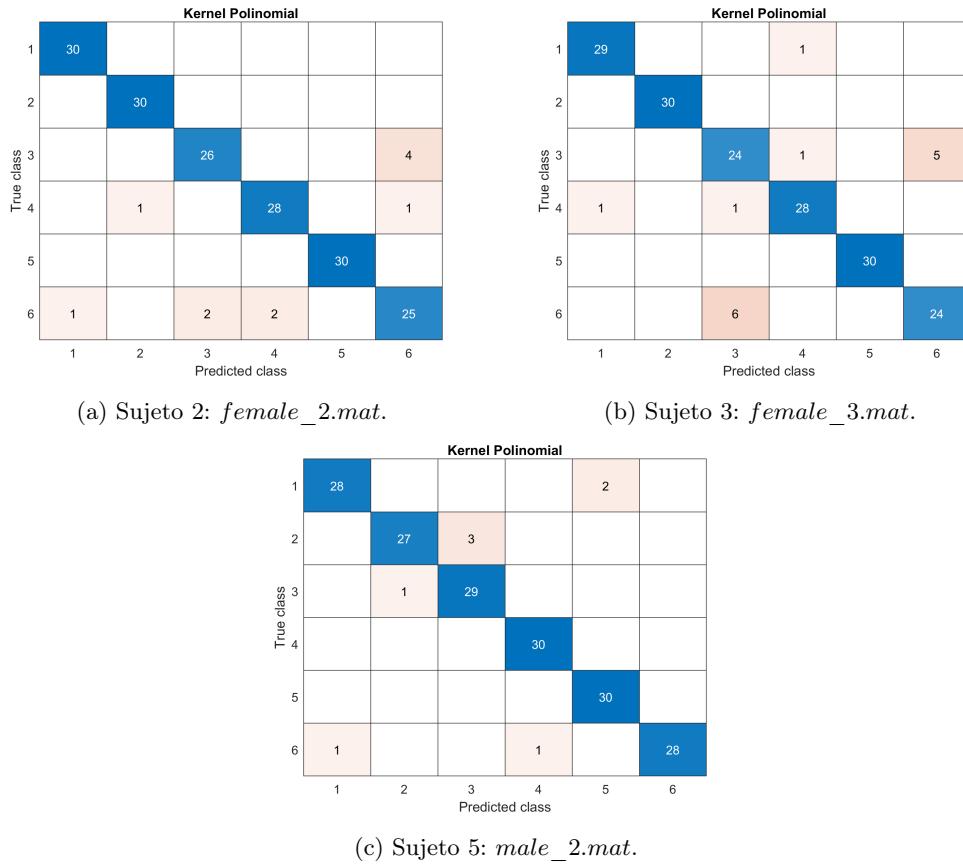


Figura 89: Resultados obtenidos con SVM polinomial para clasificación de 6 clases, 7 características.

Redes Neuronales

Resultados con 2 características: MAV y ZC.

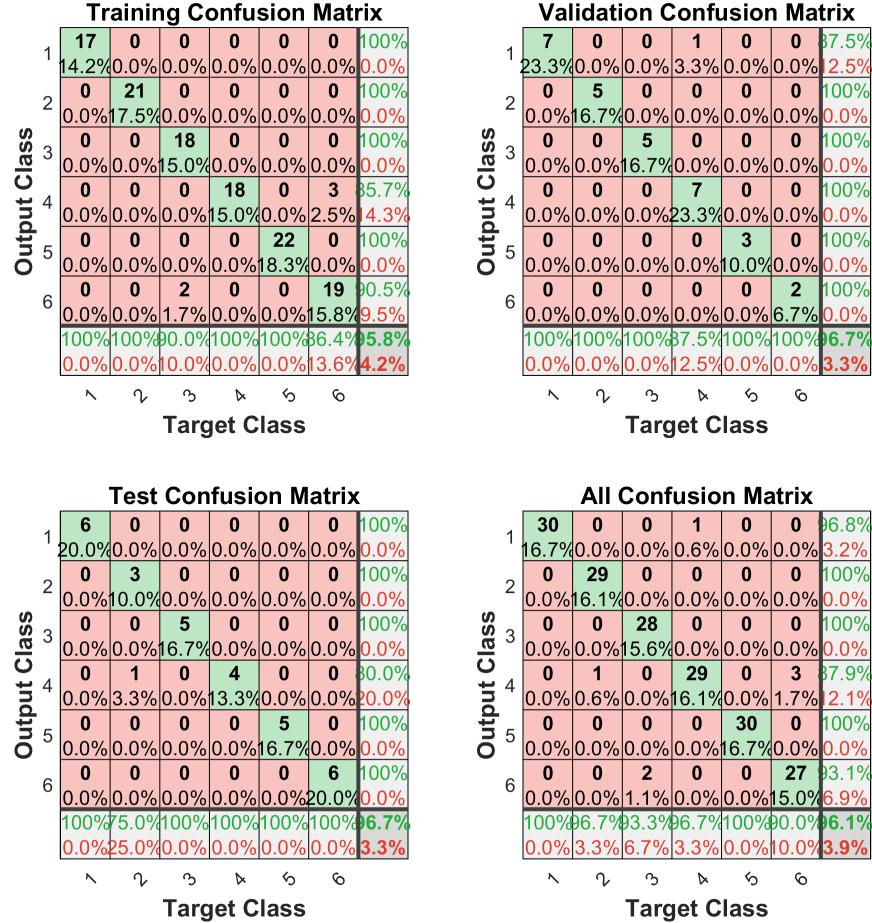


Figura 90: Resultados obtenidos con RN para clasificación de 6 clases, 2 características. Sujeto 2: *female_2.mat*.

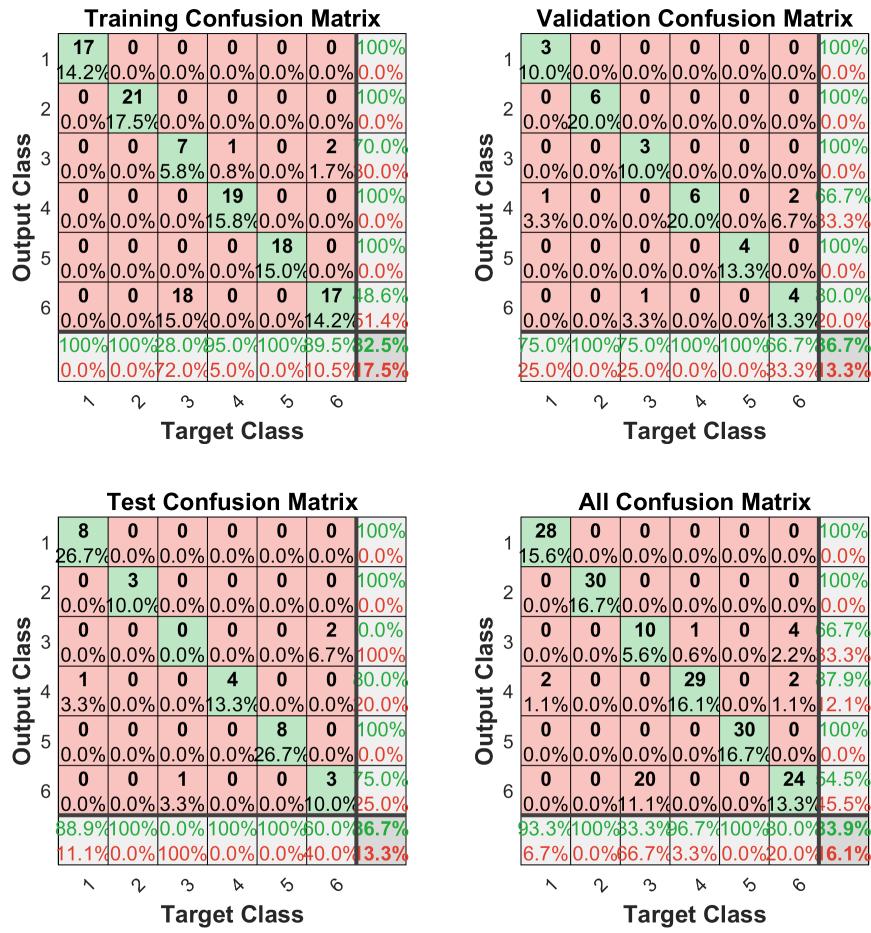


Figura 91: Resultados obtenidos con RN para clasificación de 6 clases, 2 características. Sujeto 3: *female_3.mat*.

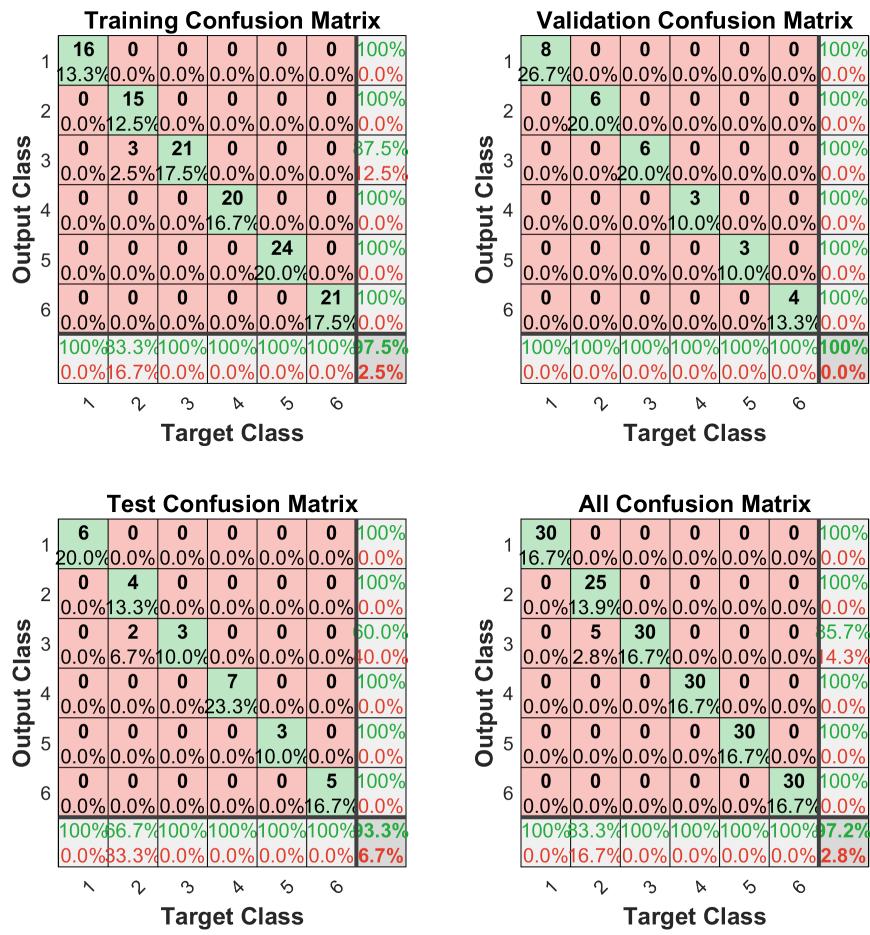


Figura 92: Resultados obtenidos con RN para clasificación de 6 clases, 2 características. Sujeto 5: *male_2.mat*.

Resultados con 3 características: MAV, ZC y WL.

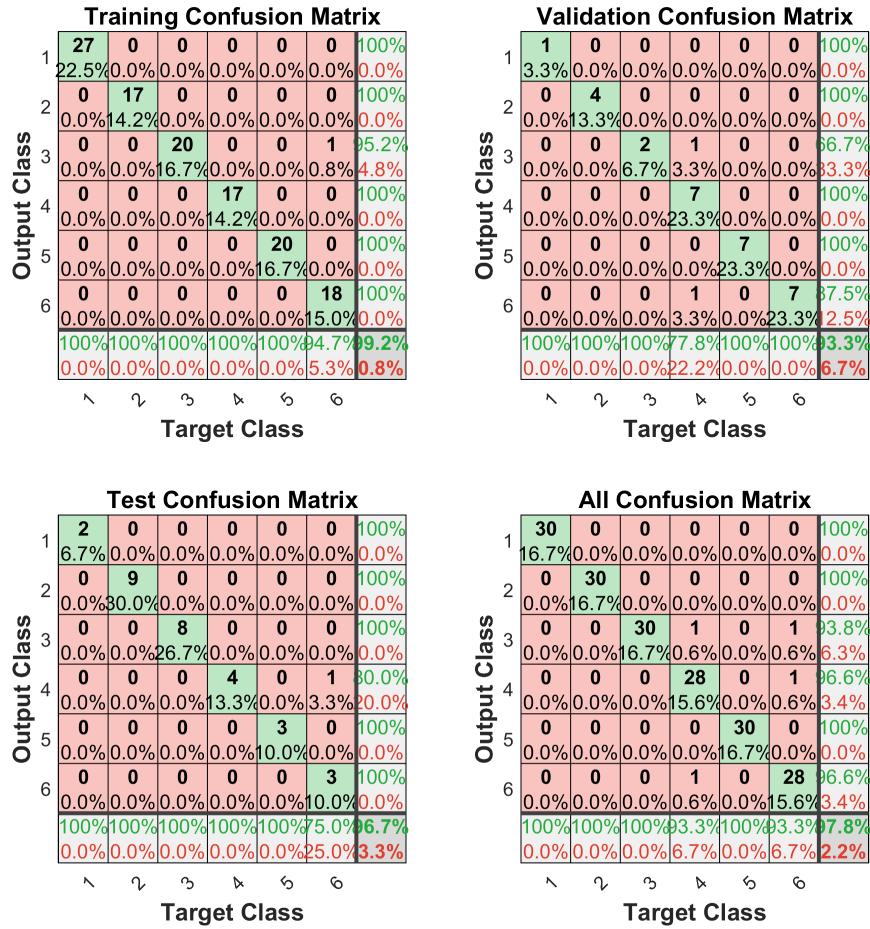


Figura 93: Resultados obtenidos con RN para clasificación de 6 clases, 3 características. Sujeto 2: *female_2.mat*.

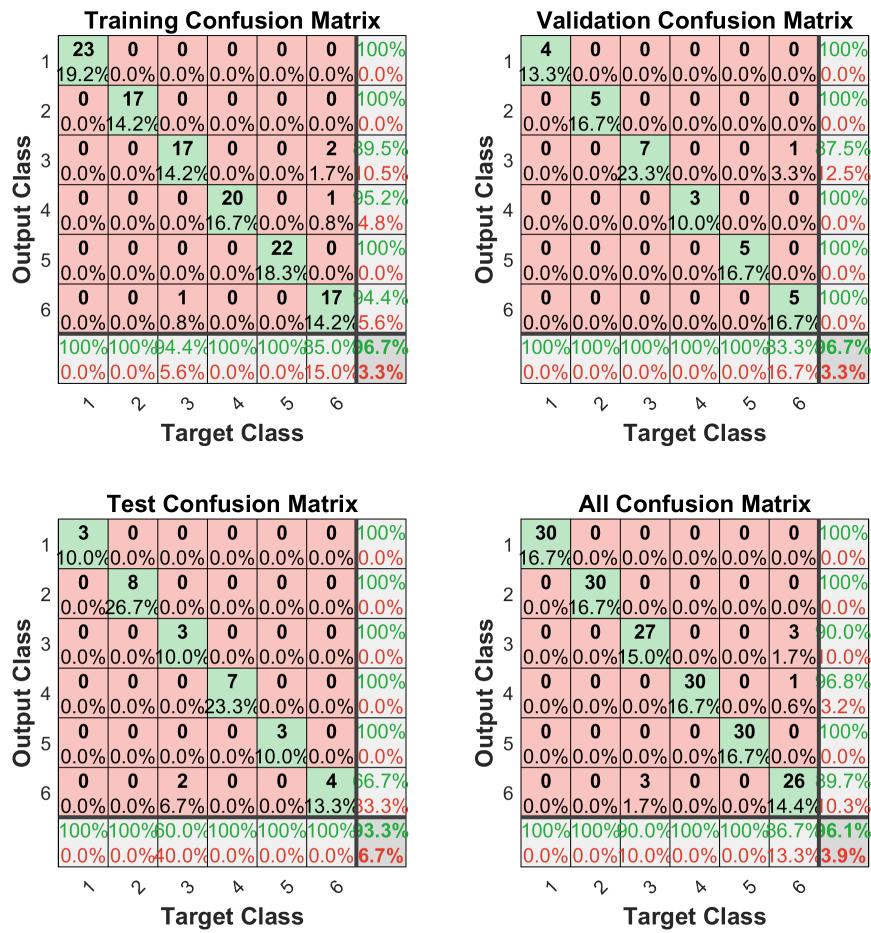


Figura 94: Resultados obtenidos con RN para clasificación de 6 clases, 3 características. Sujeto 3: *female_3.mat*.

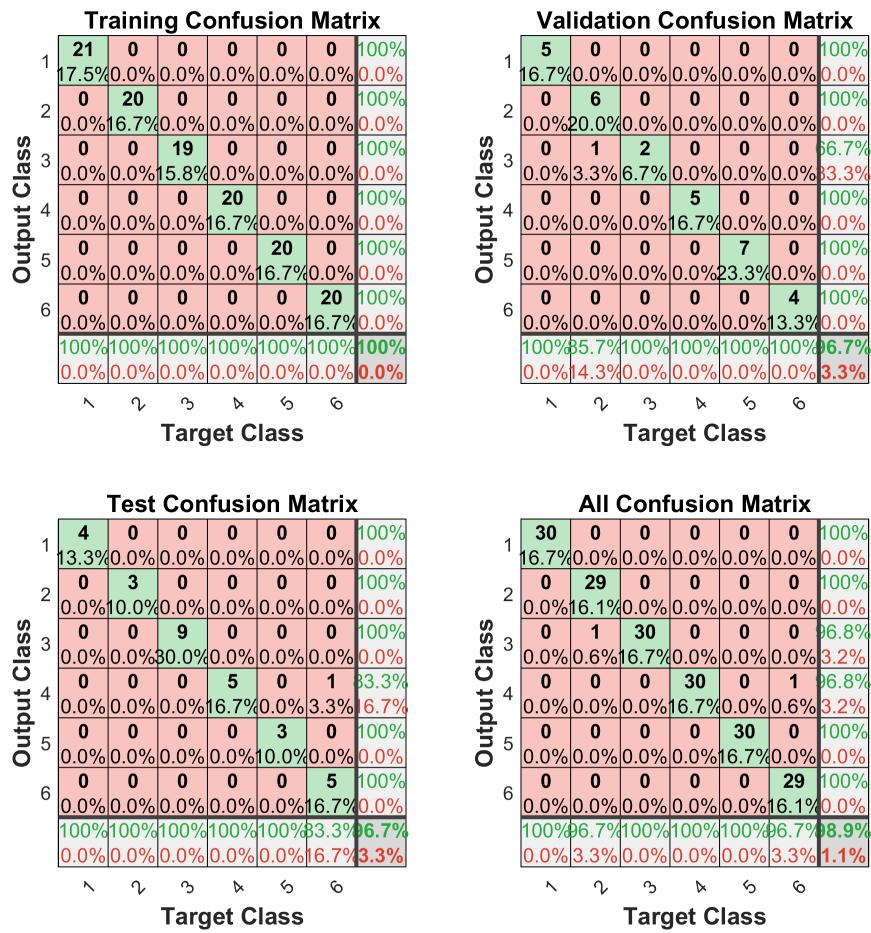


Figura 95: Resultados obtenidos con RN para clasificación de 6 clases, 3 características. Sujeto 5: *male_2.mat*.

Resultados con 5 características: MAV, ZC, WL, IEMG y RMS.

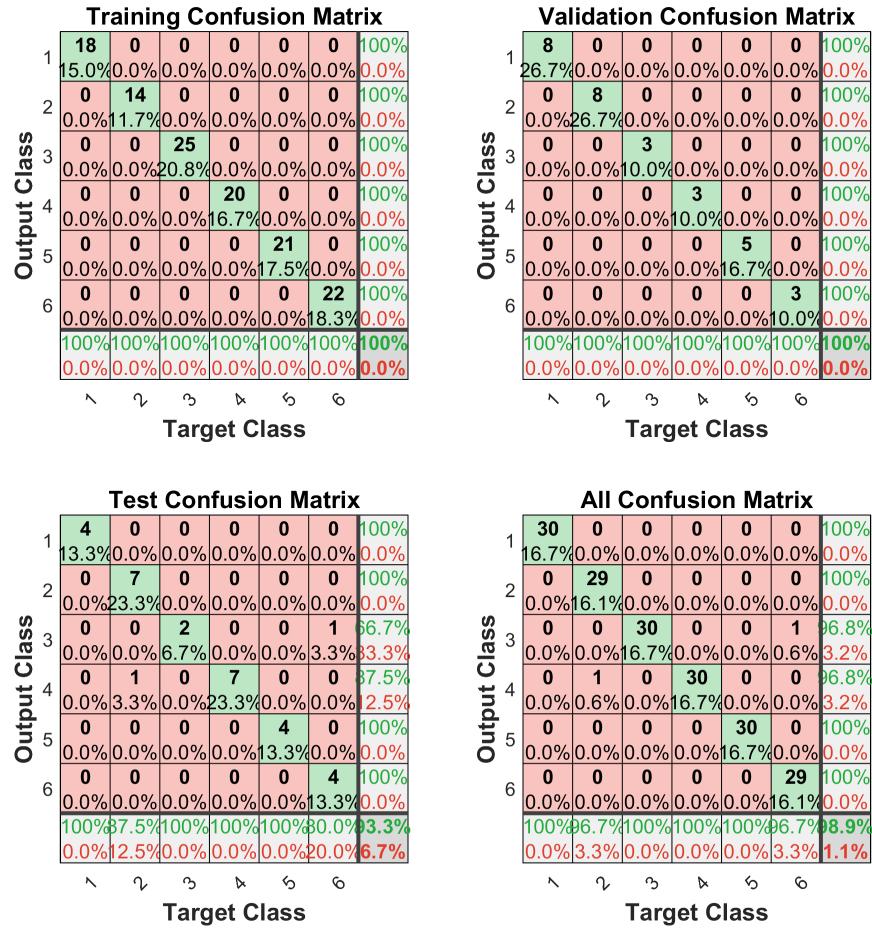


Figura 96: Resultados obtenidos con RN para clasificación de 6 clases, 5 características. Sujeto 2: *female_2.mat*.

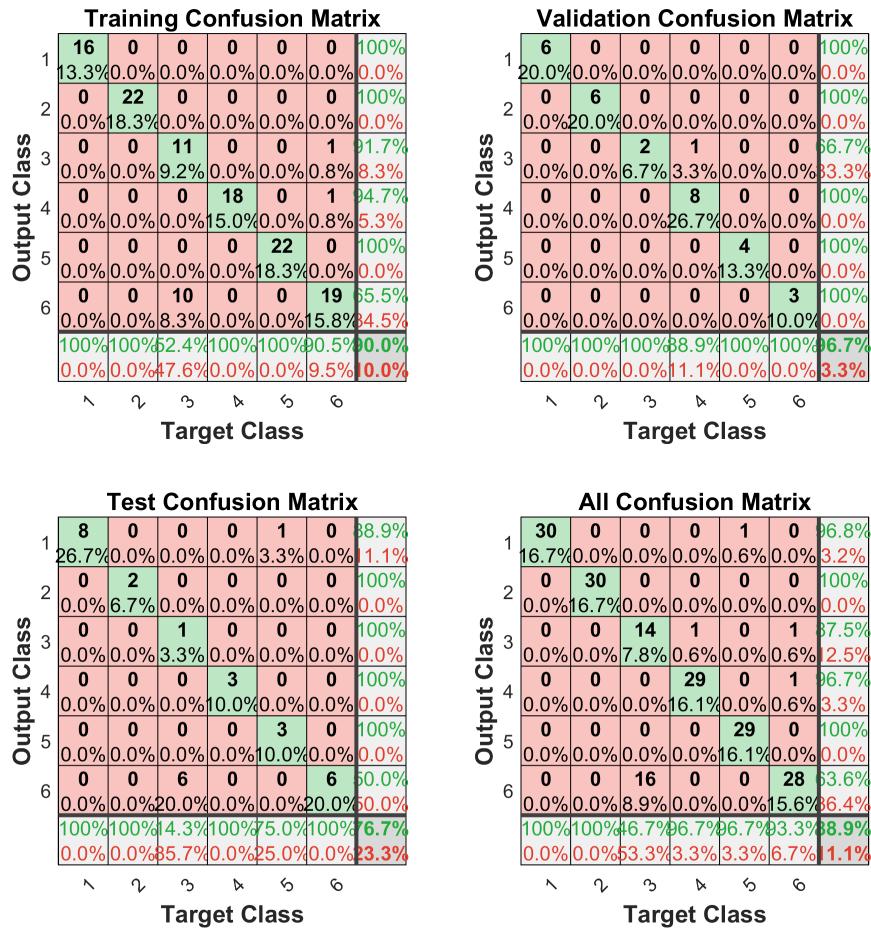


Figura 97: Resultados obtenidos con RN para clasificación de 6 clases, 5 características. Sujeto 3: *female_3.mat*.

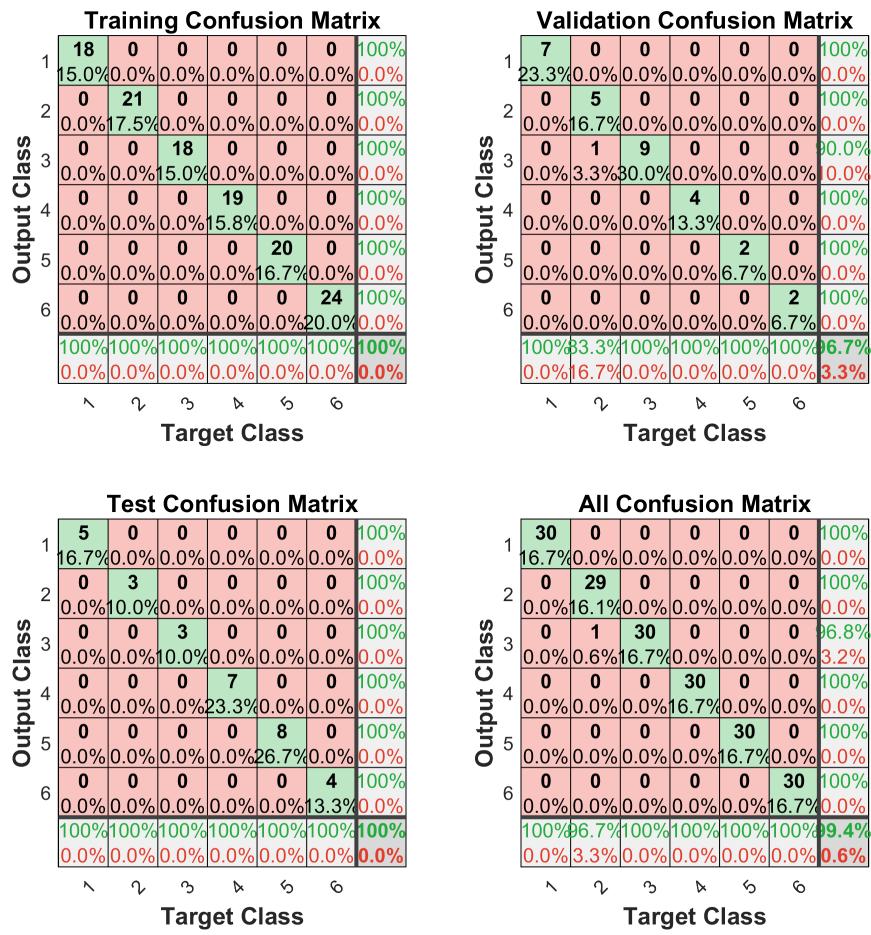


Figura 98: Resultados obtenidos con RN para clasificación de 6 clases, 5 características. Sujeto 5: *male_2.mat*.

Resultados con 7 características: MAV, ZC, WL, IEMG, RMS, VAR y STD.

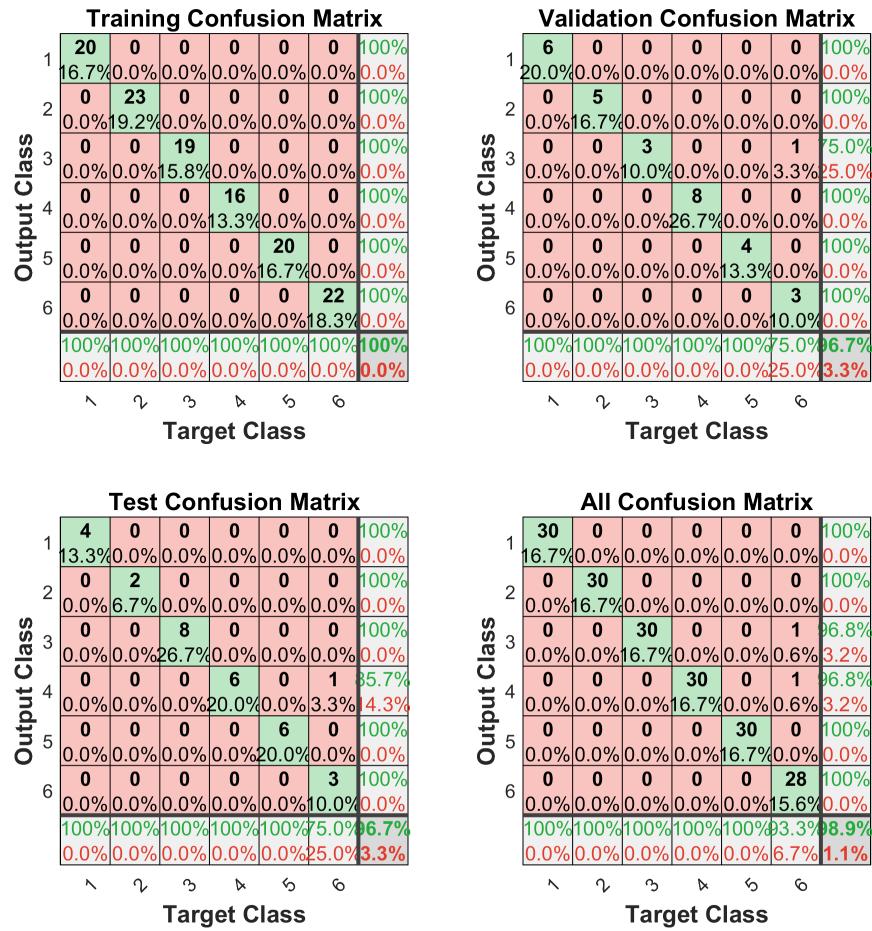


Figura 99: Resultados obtenidos con RN para clasificación de 6 clases, 7 características. Sujeto 2: *female_2.mat*.

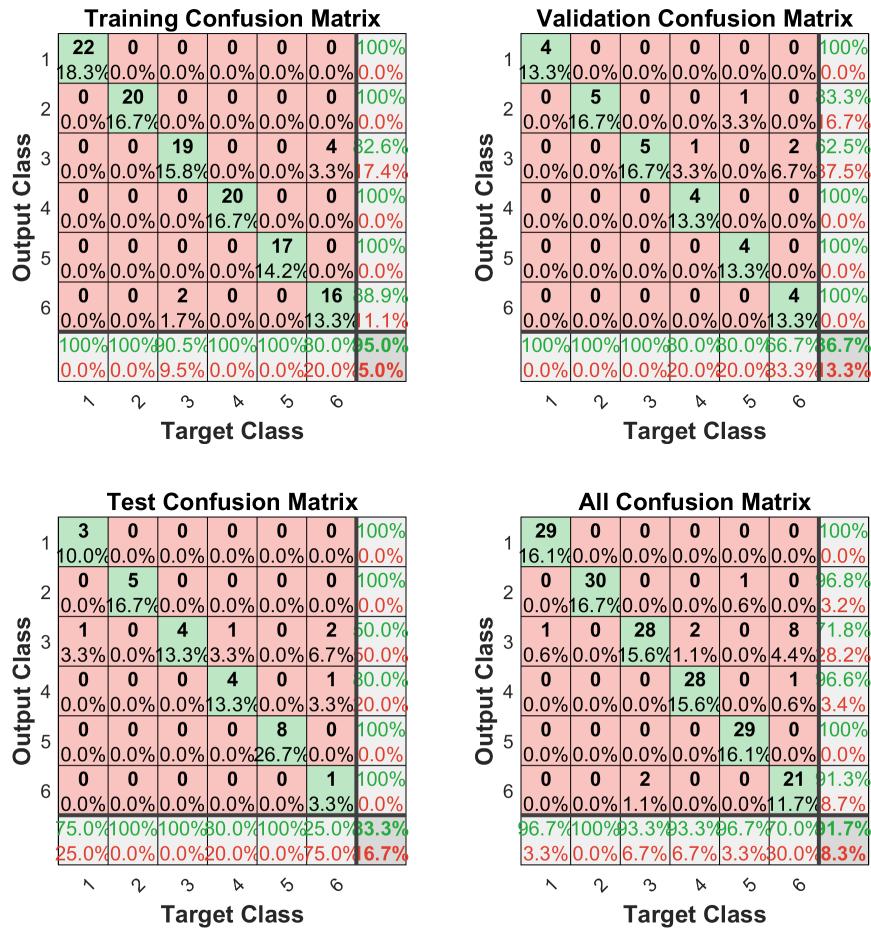


Figura 100: Resultados obtenidos con RN para clasificación de 6 clases, 7 características. Sujeto 3: *female_3.mat*.

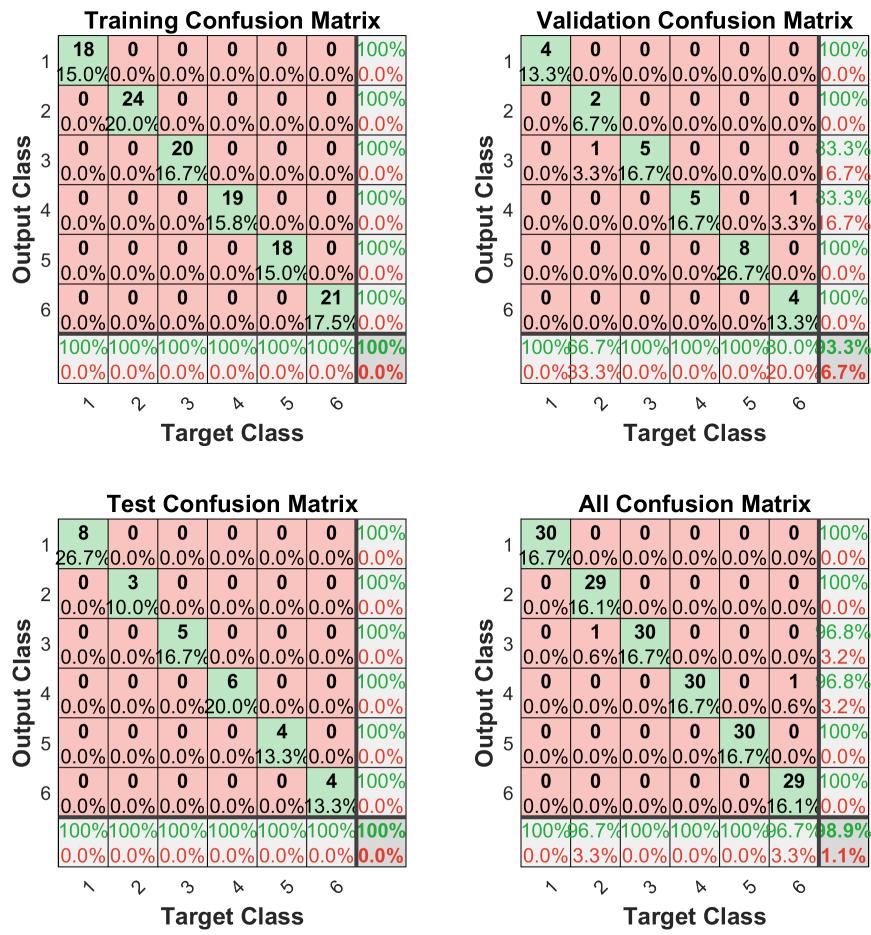


Figura 101: Resultados obtenidos con RN para clasificación de 6 clases, 7 características. Sujeto 5: *male_2.mat*.

CAPÍTULO 17

Glosario

Arduino Plataforma de desarrollo basada en una placa de *hardware* libre que incorpora un microcontrolador re-programable.. 19, 22

Bitalino Módulo que permite recoger datos biomédicos a través de una serie de módulos individuales.. 19

EMG Son un tipo de señales bioeléctricas que se generan debido a la actividad eléctrica que se produce en la fibra muscular durante la contracción o relajación del músculo.. 8

filtro digital Filtro de se aplican a señales discretas con el fin de atenuar o resaltar características específicas de la señal.. 25

hueso pisiforme Hueso medial y anterior de la primera fila del carpo, localizado en la muñeca con un tamaño y forma parecida a un guisante.. 20

IEMG Sumatoria de cada elemento en un segmento de tiempo específico.. 11

kernel Las funciones kernel son funciones matemáticas empleadas en máquinas de vectores de soporte, para convertir un problema de clasificación no lineal en un problema de clasificación lineal transformando el espacio dimensional.. 15

Machine Learning Rama de la inteligencia artificial que permite desarrollar sistemas que aprenden automáticamente basados en su experiencia.. 13

Matlab Programa computacional interactivo orientado a la ejecución de proyectos que impliquen una variedad de operaciones y tareas matemáticas y la visualización gráfica de los mismos.. 22

MAV Promedio del valor absoluto en un conjunto de datos obtenidos en un tiempo específico.. 11

redes neuronales Modelo de aprendizaje y procesamiento automático inspirado en el funcionamiento del sistema nervioso humano.. 14, 16

RMS Fuerza aplicada bajo condiciones de no fatiga.. 11

SVM Conjunto de métodos de aprendizaje supervisado utilizados para resolver problemas de clasificación, regresión y detección de valores atípicos.. 14, 15

WL Longitud de onda de una señal acumulada en un segmento de tiempo específico.. 12

ZC Número de veces que una señal pasa por cero en un segmento especificado.. 11