



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍAS EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN MECATRÓNICA**

TEMA:

**“ESTIMACIÓN DE POSICIÓN DE LA MUÑECA A TRAVÉS
DE SEÑALES ELECTROMIOGRÁFICAS”**

AUTOR: Angelo Fernando Flores Taramuel

DIRECTORA: MSc. Gabriela Alexandra Verdezoto Carranco

IBARRA, 2019



**UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA
UNIVERSIDAD TECNICA DEL NORTE**

1. IDENTIFICACIÓN DE OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100394152-1		
APELLIDOS Y NOMBRES:	FLORES TARAMUEL ANGELO FERNANDO		
DIRECCIÓN:	QUETZAL 1-31 Y TUNGURAHUA (IBARRA)		
EMAIL:	afflorest@utn.edu.ec		
TELÉFONO FIJO:	062-643 638	TELÉFONO MÓVIL:	(+593) 982652192

DATOS DE LA OBRA	
TÍTULO:	"ESTIMACIÓN DE POSICIÓN DE LA MUÑECA A TRAVÉS DE SEÑALES ELECTROMIOGRÁFICAS"
AUTOR:	FLORES TARAMUEL ANGELO FERNANDO
FECHA:	19 - JULIO 2019
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
DIRECTOR:	MSc. Gabriela Alexandra Verdezoto Carranco

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 19 días del mes de Julio de 2019

EL AUTOR:

Flores Taramuel Angelo Fernando



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

En calidad de directora del trabajo de grado **“ESTIMACIÓN DE POSICIÓN DE LA MUÑECA A TRAVÉS DE SEÑALES ELECTROMIOGRÁFICAS”**, presentado por el egresado ANGELO FERNANDO FLORES TARAMUEL, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, Julio 2019

MSc. Gabriela Verdezoto

DIRECTORA DE TESIS

AGRADECIMIENTO

Agradezco a Dios por darme sabiduría para culminar una meta más en mi vida, a mis padres por el apoyo incondicional y al amor de mi vida por la motivación.

A mi Directora de tesis la MSc, Gabriela Verdezoto, a mis Asesores el MSc. Iván Iglesias y la MSc. Luz María Tobar quienes con su conocimiento me han ayudado a culminar con éxito este proyecto de investigación.

Al apoyo brindado por el Lic. Jorge Luis Zambrano y el Lic Jaime Muñoz quienes aportaron y asesoraron el proyecto en la parte médica.

Fernando Flores T.

DEDICATORIA

Este grandioso sueño hecho realidad, quiero dedicarlo a María Checa, mi abuelita, quien ha sido mi principal motivación en toda mi vida para lograr grandes cosas, y quien ha formado ahora un gran hombre, también a mis padres y hermanos quien han sido parte de esto y han estado apoyándome en todo momento.

Al amor de mi vida Sharon Guerra, mi fiel compañera, quien ha estado a mi lado en los momentos más difíciles de mi vida, cuidando de mí y alentándome a seguir luchando para salir adelante. Su optimismo, su comprensión, su cariño y su grandioso amor, son parte fundamental de este logro. Ella me ha motivado día a día para cumplir este sueño tan anhelado. Le doy gracias por compartir conmigo los mejores momentos de mi vida y por entregarme su corazón por completo, es de gran orgullo para mí poder ser su compañero de vida.

Fernando Flores T.

RESUMEN

El presente estudio, expone la estimación de posición de la muñeca a través de señales electromiográficas, para conseguir esto, se efectuó la selección de las posiciones angulares en los movimientos Flexión-Extensión y Abducción-Aducción, siendo la base principal para el desarrollo de esta investigación. Posteriormente, se eligieron los músculos de los que se extrajeron las señales mioeléctricas, de acuerdo con parámetros como el tamaño del músculo, la superficialidad y el aporte al movimiento en ejecución.

Una vez definido un músculo de estudio por movimiento, se extrajeron muestras electromiográficas con la ayuda de tarjetas electrónicas y electrodos superficiales de Plata/Cloruro de Plata (Ag/AgCl). Posteriormente, se conformó una base de datos de cada movimiento constituido por 34 muestras, mismas que estaban integradas por 3 activaciones musculares.

Conformada la base de datos, se realizó el procesamiento, acondicionamiento y digitalización de la señal; más adelante, se llevó a cabo el proceso de caracterización con análisis en el dominio del tiempo y frecuencia, de donde se obtuvo características particulares de la señal, que son de importancia para los clasificadores.

Ulteriormente de obtener las características principales, se aplicó técnicas de aprendizaje de máquina, analizando la clasificación con varios algoritmos de tipo supervisados como Redes Neuronales, Maquinas de Soporte Vectorial, Análisis Discriminante, Nearest Neighbor y de tipo no supervisados como Clustering con la técnica Mapas de auto-organización (SOM); esto, con la finalidad de encontrar un clasificador óptimo que permita la identificación acertada del movimiento.

ABSTRACT

The present research exposes the position estimation of the wrist through electromyographic signals, to achieve this, the selection of the angular positions in the movements Flexion-Extension and Abduction-Adduction was made, being the main basis for the development of this investigation. Subsequently, the muscles from which the myoelectric signals were extracted were chosen, according to parameters such as muscle size, superficiality and the contribution to the movement in execution.

A study muscle was defined by movement, electromyographic samples were extracted with the help of electronic cards and surface electrodes of Silver / Silver Chloride (Ag / AgCl). Then a database was formed of each movement consisting of 34 samples, which were integrated by 3 muscle activations. Once the database was formed, the processing, conditioning and digitalization of the signal was carried out; later on, the characterization process was carried out with analysis in the time and frequency domain, from which specific characteristics of the signal were obtained, which are of importance for the classifiers. Subsequently, to obtain the main characteristics, machine learning techniques were applied, analyzing the classification with several supervised type algorithms such as Neural Networks, Vector Support Machines, Discriminated Analysis, Nearest Neighbor and unsupervised types such as Clustering with the self-organization Maps (SOM); this, in order to find an optimal classifier that allows the successful identification of the movement.

TABLA DE CONTENIDOS

RESUMEN	vi
ABSTRACT	vii
TABLA DE CONTENIDOS	viii
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE TABLAS	xv
INTRODUCCIÓN	1
Planteamiento del problema.....	1
Objetivos.....	2
Objetivo general.....	2
Objetivos específicos	2
Alcance	2
Justificación	3
CAPÍTULO I: MARCO TEÓRICO	5
1.1 Electromiografía (EMG).....	5
1.2 Unidades EMG y composición	5
1.2.1 Unidad Motora (UM).....	5
1.2.2 Fibras Musculares.....	6
1.2.3 Dendrita de la UM.....	6
1.2.4 Axón de la UM.....	6
1.3 Composición de la señal electromiográfica	7
1.3.1 Disparos de la Unidad Motora (UM).....	7
1.3.2 Potencial de Unidad Motora (PAUM)	7
1.3.3 Tren de Potencial de Unidad Motora (TPAUM)	7

1.4 Anatomía de la muñeca.....	8
1.4.1 Huesos y articulaciones en el complejo articular de la muñeca.....	8
1.5 Biomecánica de la muñeca.....	9
1.6 Movilidad de la muñeca.....	10
1.6.1 Movilidad básica.....	10
1.6.2 Movimiento flexión-extensión.....	12
1.6.3 Movimiento abducción-aducción.....	13
1.7 Anatomía del antebrazo	14
1.8 Biomecánica del antebrazo	15
1.9 Instrumentos médicos	16
1.9.1 Goniómetro	16
1.10 Extracción de características EMG.....	17
1.10.1 Análisis en el dominio del tiempo	18
1.10.2 Análisis en el dominio de la frecuencia	18
1.10.3 Análisis en el dominio tiempo frecuencia.....	19
1.11 Métodos de clasificación.....	20
1.12 Aprendizaje Predictivo o Supervisado.....	21
1.12.1 Redes Neuronales Artificiales (RNA)	21
1.12.1.1 Redes Neuronales de Perceptrones de capas Múltiples	21
1.12.1.2 Redes Neuronales Monocapa.....	26
1.13 Aprendizaje Descriptivo o No Supervisado.....	26
CAPÍTULO II: METODOLOGÍA	27
2.1 Elección de músculos principales	27
2.2 Ubicación de los electrodos	28
2.2.1 Ubicación en Flexor radial del carpo.....	30

2.2.2 Ubicación en Extensor cubital del carpo	31
2.2.3 Ubicación en Flexor cubital del carpo	31
2.2.4 Ubicación en Extensor largo del pulgar.....	32
2.3 Selección de ángulos de estudio	32
2.4 Diseño Conceptual	33
2.4.1 Selección de equipos para el sistema	34
2.4.1.1 Selección de Tarjeta de Procesamiento.....	34
2.4.1.2 Selección de Tarjeta de Adquisición.....	37
2.4.1.3 Selección de Electrodo.....	40
2.4.1.4 Selección de Software.....	43
2.5 Conexión entre Discovery STM32F4 y Shield EKG-EMG.	45
2.5.1 Configuración y comunicación de la tarjeta Discovery STM32F4	46
2.6 Adquisición y procesamiento de la señal EMG	47
2.6.1 Filtrado digital de la señal EMG	49
2.6.2 Rectificado de la señal EMG	50
2.6.3 Detección de actividad de la señal EMG	51
2.7 Caracterización de señales EMG	52
2.8 Clasificación de señales EMG	61
CAPITULO III: PRUEBAS Y ANÁLISIS DE RESULTADOS	73
3.1 Pruebas de adquisición de señales EMG	73
3.2 Pruebas de clasificación con algoritmos supervisados	77
3.2.1 Clasificador Red Neuronal de Backpropagation.....	77
3.2.1.1 Análisis del Cambio de Función de Activación.....	77
3.2.1.2 Análisis del Cambio del Número de Neuronas.....	85
3.2.2 Clasificador Nearest Neighbor (K-NN).....	93

3.2.3 Clasificador Análisis Discriminante	95
3.2.4 Clasificador Máquinas de Soporte Vectorial (SVM).....	96
3.3 Pruebas de clasificación con algoritmos no supervisados	99
3.3.1 Algoritmo Clustering por Mapa auto organizado (SOM).....	99
3.4 Prueba de Funcionamiento OFF-LINE.....	103
CONCLUSIONES Y RECOMENDACIONES	109
Conclusiones	109
Recomendaciones	111
BIBLIOGRAFÍA	113
ANEXOS	115
ANEXO A: Supervisión y Certificación Médica.....	115
ANEXO A.1: Oficio de especialista asignado.....	115
ANEXO B: Esquema general de conexiones.....	116
ANEXO C: Código implementado en el sistema	117
ANEXO C.1: Código para Abducción de muñeca	117
ANEXO C.2: Código para Aducción de muñeca	121
ANEXO C.3: Código para Flexión de muñeca.....	126
ANEXO C.4: Código para Extensión de muñeca.....	130

ÍNDICE DE FIGURAS

Figura 1.1 Unidad motora [4].	6
Figura 1.2 (a)Tren de potenciales de acción de las UM y sumatoria de las TPAUM (b) Modelo de generación de las señales EMG [5]......	7
Figura 1.3 Complejo articular de la muñeca con sus huesos y articulaciones [7].	9
Figura 1.4 Relación de los músculos que desarrollan los movimientos del complejo articular de la muñeca [7]......	10
Figura 1.5 Movimiento extensión-flexión básico de la muñeca [8].	11
Figura 1.6 Movimiento abducción-aducción básico de la muñeca [8].	11
Figura 1.7 Representación del movimiento de muñeca flexión-extensión [6].	13
Figura 1.8 Representación del movimiento de la muñeca abducción-aducción [6].	14
Figura 1.9 Anatomía del antebrazo [11].	15
Figura 1.10 Goniómetro metálico que muestra un brazo móvil (A), un brazo fijo (B), un cuerpo transportador (C) y un eje (D) [12].	17
Figura 1.11 Técnicas de aprendizaje automático [19].	20
Figura 1.12 Red Neuronal MLP con metodología de aprendizaje de retro-propagación [21]. ...	22
Figura 1.13 Red Neuronal de retro-propagación [23]......	22
Figura 1.14 Función de activación Logística o sigmoideal [24].	24
Figura 1.15 Función de activación tangente hiperbólica [24]......	25
Figura 1.16 Función de activación Lineal [24]......	25
Figura 1.17 Red Neuronal Unicapa [18]......	26
Figura 2.18 Ubicación de los electrodos en los músculos (a) Flexor radial del carpo. (b) Extensor cubital del carpo. (c) Flexor cubital del carpo. (d) Extensor largo del pulgar.	29
Figura 2.19 Modelo eléctrico de conexión del electrodo [30].	30
Figura 2.20 Ubicación de electrodos al músculo Flexor radial del carpo. (a) Ubicación en simulador. (b) Ubicación real en el usuario.	31
Figura 2.21 Ubicación de electrodos al músculo Extensor cubital del carpo. (a) Ubicación en simulador. (b) Ubicación real en el usuario.	31
Figura 2.22 Ubicación de electrodos al músculo Flexor cubital del carpo. (a) Ubicación en simulador. (b) Ubicación real en el usuario.	32

Figura 2.23 Ubicación de electrodos al músculo Extensor largo del pulgar. (a) Ubicación en simulador. (b) Ubicación real en el usuario.	32
Figura 2.24 Tarjeta de procesamiento Discovery STM32F4 [32].	36
Figura 2.25 Tarjeta de adquisición de señales EKG-EMG Olimex [32].	39
Figura 2.26 Cable conector DLR electrodo con salida Jack 3.5 [32].	40
Figura 2.27. Electrodo de Plata/Cloruro de plata (Ag/AgCl).	42
Figura 2.28 Conexión entre Discovery STM32F4 y Shield EKG-EMG.	45
Figura 2.29 Bloque de configuración en Simulink de tarjeta Discovery STM32F4.	46
Figura 2.30 Bloque de configuración en simulink para la adquisición de datos EMG.	47
Figura 2.31 Ubicación de goniómetro para movimiento de: (a) Flexión. (b) Extensión. (c) Abducción. (d) Aducción.	48
Figura 2.32 Señales en estado puro del movimiento Abducción en las 4 posiciones.	49
Figura 2.33 Señales filtradas del movimiento Abducción.	50
Figura 2.34 Señales rectificadas del movimiento Abducción.	51
Figura 2.35 Señales procesadas con detección de actividad del movimiento Abducción.	52
Figura 2.36 Características en 2 dimensiones de movimientos: (a) Abducción (b) Aducción.	61
Figura 2.37 Características en 2 dimensiones de movimientos: (a) Flexión (b) Extensión.	62
Figura 2.38 Características en 3 dimensiones de movimientos: (a) Abducción (b) Aducción.	62
Figura 2.39 Características en 3 dimensiones de movimientos: (a) Flexión (b) Extensión.	63
Figura 2.40 Pseudocódigo del algoritmo Red Neuronal Backpropagation.	67
Figura 2.41 Pseudocódigo del Clasificador Nearest Neighbor.	69
Figura 2.42 Pseudocódigo del Clasificador Análisis Discriminante.	70
Figura 2.43 Pseudocódigo del Clasificador Máquina de Soporte Vectorial.	71
Figura 2.44 Pseudocódigo del Clasificador Clustering SOM.	72
Figura 3.45 Sistema de adquisición de señales EMG de 4 canales	73
Figura 3.46 Ubicación de electrodos: (a) Por un especialista médico. (b) Para sistema de adquisición de 4 canales.	74
Figura 3.47 Adquisición de 4 canales ejecutando el movimiento Flexión de muñeca.	74
Figura 3.48 Adquisición de 4 canales ejecutando el movimiento Extensión de muñeca.	75
Figura 3.49 Adquisición de 4 canales ejecutando el movimiento Aducción de muñeca.	75
Figura 3.50 Adquisición de 4 canales ejecutando el movimiento Abducción de muñeca.	76

Figura 3.51 Estructura de red con función de activación sigmoide y 11 características.	77
Figura 3.52 Estructura de red con función de activación sigmoide y 3 características.	79
Figura 3.53 Estructura de red con función de activación lineal y 11 características.	80
Figura 3.54 Estructura de red con función de activación lineal y 3 características.	81
Figura 3.55 Estructura de red con función de activación tangente y 11 características.	82
Figura 3.56 Estructura de red con función de activación tangente y 3 características.	83
Figura 3.57 Pruebas de entrenamiento del clasificador KNN con modelo Distancia Métrica del Coseno: (a) Datos sin entrenamiento. (b) Datos entrenados.	93
Figura 3.58 Pruebas de entrenamiento del clasificador KNN con modelo Distancia Métrica Cúbica: (a) Datos sin entrenamiento. (b) Datos entrenados.	94
Figura 3.59 Pruebas de entrenamiento del clasificador Análisis Discriminante: (a) Datos sin entrenamiento. (b) Datos entrenados.	95
Figura 3.60 Pruebas de entrenamiento del clasificador Máquinas de Soporte Vectorial: (a) Datos sin entrenamiento. (b) Datos entrenados.	96
Figura 3.61 Arquitectura del SOM [38].	100
Figura 3.62 (a) Diseño de la red SOM. (b) Topología de las neuronas SOM.	100
Figura 3.63 Distancia de pesos SOM.	101
Figura 3.64 Datos asociados a las neuronas.	102
Figura 3.65 Distribución de neuronas.	102
Figura 3.66 Prueba de adquisición en movimiento Abducción 10 grados.	103
Figura 3.67 Resultados de prueba offline en movimiento Abducción a 10 grados.	104
Figura 3.68 Prueba de adquisición en movimiento Aducción 30 grados.	104
Figura 3.69 Resultados de prueba offline en movimiento Aducción a 30 grados.	105
Figura 3.70 Prueba de adquisición en movimiento Flexión 60 grados.	106
Figura 3.71 Resultados de prueba offline en movimiento Flexión a 60 grados.	107
Figura 3.72 Prueba de adquisición en movimiento Extensión 30 grados.	107
Figura 3.73 Resultados de prueba offline en movimiento Extensión a 30 grados.	108

ÍNDICE DE TABLAS

Tabla 2.1 Músculos que intervienen en cada movimiento y rango de movilidad.....	27
Tabla 2 .2 Selección de los 4 músculos principales.....	28
Tabla 2.3 Ángulos de estudio para los 4 movimientos de la muñeca.	33
Tabla 2.4 Evaluación del peso específico de cada criterio para la selección de tarjeta de procesamiento.	34
Tabla 2.5 Evaluación del peso específico del criterio HARDWARE.	34
Tabla 2.6 Evaluación del peso específico del criterio VELOCIDAD.	35
Tabla 2.7 Evaluación del peso específico del criterio PRECIO (bajo).....	35
Tabla 2.8 Tabla de conclusiones para la selección de tarjeta de procesamiento.	35
Tabla 2.9 Evaluación del peso específico de cada criterio para la selección de la tarjeta de adquisición.	37
Tabla 2.10 Evaluación del peso específico del criterio PRECISIÓN.....	37
Tabla 2.11 Evaluación del peso específico del criterio COMPATIBILIDAD.	38
Tabla 2.12 Evaluación del peso específico del criterio PRECIO (bajo).....	38
Tabla 2.13 Tabla de conclusiones para la selección de la tarjeta de adquisición.	38
Tabla 2.14 Evaluación del peso específico de cada criterio para la selección de los electrodos.	41
Tabla 2.15 Evaluación del peso específico del criterio POTENCIAL.	41
Tabla 2.16 Evaluación del peso específico del criterio REDUCCIÓN DE RUIDO	41
Tabla 2.17 Evaluación del peso específico del criterio PRECIO (bajo).....	42
Tabla 2.18 Tabla de conclusiones para la selección de los electrodos.	42
Tabla 2.19 Evaluación del peso específico de cada criterio para la selección del software.	43
Tabla 2.20 Evaluación del peso específico del criterio UTILIDAD.	43
Tabla 2.21 Evaluación del peso específico del criterio COMPATIBILIDAD.	44
Tabla 2.22 Evaluación del peso específico del criterio RAPIDEZ.....	44
Tabla 2.23 Tabla de conclusiones para la selección del software.....	44
Tabla 2.24 Conexión de salidas y entradas entre la tarjeta de procesamiento y adquisición.	45
Tabla 2.25 Características en el movimiento Abducción.	57
Tabla 2.26 Características en el movimiento Aducción.	58

Tabla 2.27 Características en el movimiento Flexión.....	59
Tabla 2.28 Características en el movimiento Extensión.....	60
Tabla 2.29 Algoritmos de clasificación.	64
Tabla 2.30 Configuración de datos de entrada para entrenamiento.....	65
Tabla 2.31 Configuración de datos de salida para entrenamiento del clasificador Red Neuronal de Retropropagación.	66
Tabla 2.32 Configuración de datos de salida para entrenamiento del clasificador Nearest Neighbor.	68
Tabla 3.33 Resultados de la red neuronal con la función de activación SIGMOIDE y 11 características de entrada.	78
Tabla 3.34 Resultados de la red neuronal con la función de activación SIGMOIDE y 3 características de entrada.	79
Tabla 3.35 Resultados de la red neuronal con la función de activación LINEAL y 11 características de entrada.	80
Tabla 3.36 Resultados de la red neuronal con la función de activación LINEAL y 3 características de entrada.	81
Tabla 3.37 Resultados de la red neuronal con la función de activación TANGENTE y 11 características de entrada.	83
Tabla 3.38 Resultados de la red neuronal con la función de activación TANGENTE y 3 características de entrada.	84
Tabla 3.39 Resultados del Cambio de Función de Activación.	85
Tabla 3.40 Resultados variando el número de neuronas con función SIGMOIDE y 11 características.....	86
Tabla 3.41 Resultados variando el número de neuronas con función SIGMOIDE y 3 características.....	87
Tabla 3.42 Resultados variando el número de neuronas con función LINEAL y 11 características.....	88
Tabla 3.43 Resultados variando el número de neuronas con función LINEAL y 3 características.....	89
Tabla 3.44 Resultados variando el número de neuronas con función TANGENTE y 11 características.....	90

Tabla 3.45 Resultados variando el número de neuronas con función TANGENTE y 3 características.....	91
Tabla 3.46 Resultados de la Variación del Número de Neuronas.	92
Tabla 3.47 Resultados del clasificador K-NN con modelo Distancia métrica del coseno y cúbica.	94
Tabla 3.48 Resultados del clasificador Análisis Discriminante.....	96
Tabla 3.49 Resultados del clasificador Máquinas de Soporte Vectorial	97
Tabla 3.50 Resultados de desempeño de clasificadores supervisados.....	98
Tabla 3.51 Resultados de Red Neuronal Backpropagation en todos los movimientos.	99

INTRODUCCIÓN

Planteamiento del problema

Las fracturas y lesiones de muñeca son muy habituales en deportistas asociadas con deportes que emplean: raqueta, bates, palos como el tenis, béisbol o el golf; además de otros deportes de contacto como artes marciales y defensa personal; supone un estrés crónico desencadenante de un proceso inflamatorio (tenosinovitis) que desencadena un proceso degenerativo (tendinosis) y finalmente la rotura o fractura. [1]

En dichos deportes, tras las lesiones lumbares que son las más frecuentes, el segundo lugar en incidencia lo ocupan las lesiones de la muñeca derecha; entre ellas, las más frecuentes son la tendinitis, las fracturas del gancho ganchoso, o el síndrome de impacto radial. Además del micro-traumatismo repetido, suele achacarse a un mecanismo traumático puntual relacionado con un evento traumático agudo que supone la supinación, desviación cubital y flexión forzada de la muñeca. Todo ello facilita la rotura de los tendones, las poleas y la luxación del extensor carpi ulnaris (ECU). La fractura del escafoides es la fractura más frecuente de la muñeca, tras las fracturas de la extremidad distal del radio (EDR), que concentra del 75 al 80% de todas las fracturas del carpo, siendo más habitual en los jóvenes entre los 15 a los 30 años. [1]

Los rehabilitadores actualmente tienen un mecanismo de funcionamiento no amigable con el paciente debido a que cumple su objetivo de movimiento total, sin tener en cuenta la rehabilitación brusca y apresurada que el paciente puede tener; es por esto que, usuarios han visto la necesidad de optar por otros sistemas de rehabilitación los cuales se adapten a su necesidad, en la cual se tiene una rehabilitación pasiva, prolongada y porcentual, donde sus avances sean progresivos y no sientan fatiga ni molestias a la hora de su rehabilitación.

Debido a esto se propone el desarrollo de este proyecto; con la finalidad de aportar con información necesaria para la mejora en el diseño y construcción de rehabilitadores de muñeca activos, garantizando dispositivos fiables.

Objetivos

Objetivo general

Estimar la posición de la muñeca en los movimientos flexión-extensión y radial-cubital a través de señales electromiográficas.

Objetivos específicos

- Revisar bibliografía y documentación relacionada a la caracterización de señales electromiográficas en función del movimiento de la muñeca.
- Adquirir las señales EMG de los músculos asociados al movimiento de la muñeca.
- Extraer las características de las señales EMG.
- Establecer la relación entre la señal EMG y la posición utilizando técnicas de aprendizaje de máquina.
- Comparar los resultados a través del análisis de desempeño de los métodos propuestos.

Alcance

En el presente proyecto se estimará la posición de la muñeca en movimientos de flexión-extensión y radial-cubital, movimientos que se utilizan en la rehabilitación de lesiones y fracturas. Se realizará la adquisición y parametrización de las señales electromiográficas de los músculos del antebrazo; luego, se realizará la relación entre las señales electromiográficas con la posición angular

Justificación

La rehabilitación física es el proceso de entrenamiento físico que alguien utiliza para mejorar o recuperarse de sus funciones físicas perdidas debido a la lesión de la médula espinal (LME), lesión cerebral traumática (TBI) o accidente cerebrovascular (ACV). La pérdida de las funciones físicas normalmente limita el desempeño de las actividades de la vida diaria de una persona y conduce a una mala calidad de vida. Por lo tanto, numerosos investigadores han desarrollado sistemas de rehabilitación física para restaurar las funciones perdidas y promover la calidad de vida de los pacientes. [2]

En cuanto a los sistemas de rehabilitación física, han sido ampliamente desarrollados para restaurar las funciones perdidas de pacientes con deficiencias de servidor. Sin embargo, se ha demostrado que un sistema de rehabilitación normal no proporcionará una recuperación rápida o un sistema eficaz a menos que el sistema está integrado con señales biológicas humanas, especialmente con la señal de EMG. [2]

Para mejorar los sistemas de rehabilitación normales, se genera la idea de poder saber datos sobre la rehabilitación, en este caso saber la posición de los movimientos de la muñeca a través de señales EMG que son ampliamente utilizadas en el desarrollo de sistemas de rehabilitación física debido a su capacidad de proporcionar información sobre las actividades musculares tales como la contracción de los músculos, fatiga muscular, la frecuencia de la tensión y la acumulación de actividades reflejas.

CAPÍTULO I: MARCO TEÓRICO

1.1 Electromiografía (EMG)

La captación de las señales eléctricas producidas por los músculos durante una contracción muscular se conoce como electromiografía (EMG), consiste básicamente en la adquisición, registro y análisis de la actividad eléctrica generada en nervios y músculos a través de la utilización de los electrodos (superficiales, de aguja, implantados), estas señales son generadas por el intercambio de iones a través de las membranas de las fibras musculares debido a una contracción muscular. [3]

1.2 Unidades EMG y composición

1.2.1 Unidad Motora (UM).

Las neuronas motoras se originan en la corteza cerebral motora, de ahí se interconectan con otras neuronas motoras del tronco cerebral (Tálamo y Bulbo Raquídeo) y finalmente a la medula espinal, en el cuerno ventral terminando en su órgano receptor que son los músculos; en la Figura 1.1, se explican las partes que componen la UM, las fibras musculares que están unidas por las ramificaciones de la moto neurona, dendrita y axón que se extiende a las fibras individuales. [4]

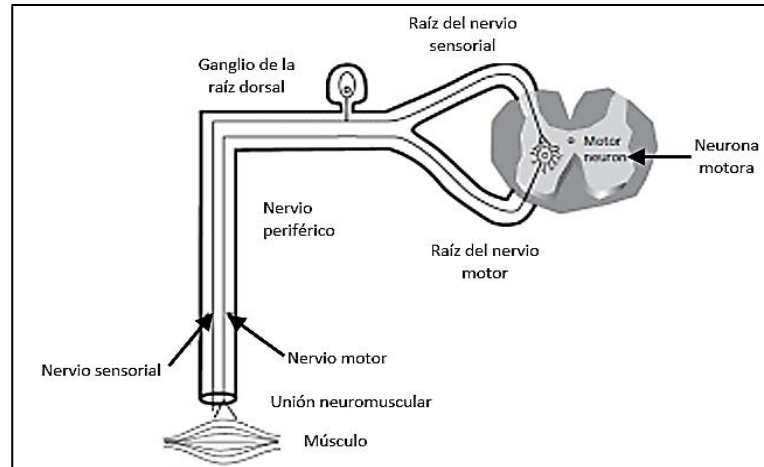


Figura 1.1 Unidad motora [4].

1.2.2 Fibras Musculares.

Son fibras individuales de un músculo los cuales están inervados por las ramificaciones de la motoneurona; cuando se excitan, estas fibras producen una fuerza de contracción las cuales en forma consecutiva genera la fuerza tetánica, aquella fuerza que permite interactuar con el medio; en la Figura 1.1, se puede observar estas fibras musculares las cuales conforman el músculo completo. [4]

1.2.3 Dentrita de la UM.

“Está unida al cuerno ventral o anterior de la espina dorsal y es la unión que tiene la UM al cuerno” [4].

1.2.4 Axón de la UM.

La motoneurona se ramifica en las fibras musculares por medio de su axón o cuerpo de la sinapsis quien permite el envío de señales eléctricas a los músculos, cuando se llega a conectar o unir a las fibras individuales, se llega a tomar el término de inervación. [4]

1.3 Composición de la señal electromiográfica

1.3.1 Disparos de la Unidad Motora (UM)

Es la tasa de excitación de las unidades motoras que producen la fuerza de contracción a las fibras musculares individuales, cuando la tasa de disparos aumenta esta produce más contracciones produciendo al final la fuerza tetánica o fuerza de salida. [5]

1.3.2 Potencial de Unidad Motora (PAUM)

“Es el potencial eléctrico producido por una unidad motora hacia sus fibras musculares individuales, estos potenciales se producen cada vez que haya un disparo de la UM” [5].

1.3.3 Tren de Potencial de Unidad Motora (TPAUM)

Es la sucesión de PAUM producido por una UM. El conjunto de trenes de potencial de unidades motoras son los que producen la señal EMG que son captados por el sensor. Esto también recibe el nombre de superposición de TPAUM; en la Figura 1.2 (a), se puede observar que la sumatoria de estos TPAUM produce la señal EMG vista por equipos electrónicos especializados y en la Figura 1.2 (b), se explica a manera de diagrama de bloques la superposición de los TPAUM para formar la señal EMG. [5]

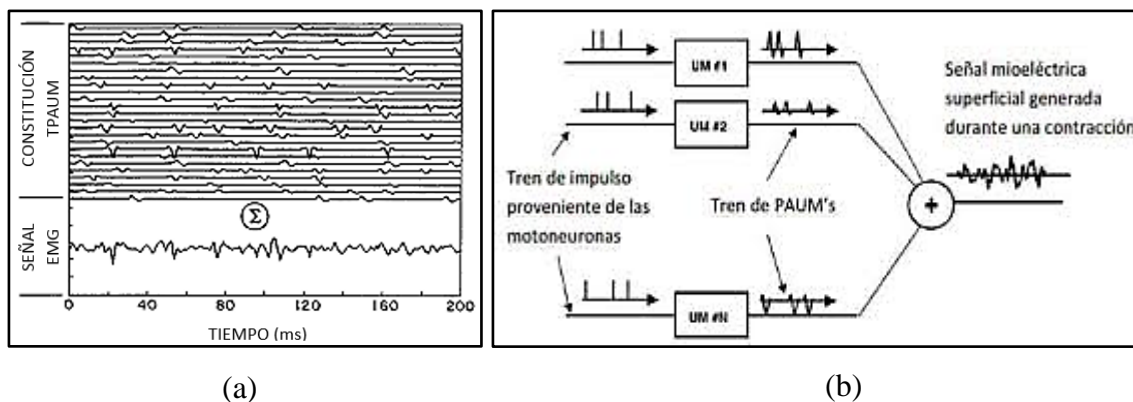


Figura 1.2 (a) Tren de potenciales de acción de las UM y sumatoria de las TPAUM (b) Modelo de generación de las señales EMG [5].

1.4 Anatomía de la muñeca

La muñeca, articulación distal del miembro superior, permite que la mano segmento efector adopte la posición óptima para la prensión; de hecho, el complejo articular de la muñeca posee dos grados de libertad, flexión-extensión y abducción-aducción; la pronosupinación, es la rotación del antebrazo sobre su eje longitudinal, que añade un tercer grado de libertad a la muñeca; la mano se puede orientar en cualquier ángulo para sujetar un objeto. [6]

La muñeca es el conjunto articular más complejo que existe en el organismo. Su área anatómica, establece la unión entre el antebrazo y la mano, incluye las extremidades metaepifisiarias distales de los huesos radio y cúbito, las dos hileras de huesos del carpo y las bases de los huesos metacarpiano; este concepto resulta importante, pues como refleja, no son solo los huesos del carpo, como aparece en varios textos y artículos científicos, sino que incluye porciones distales de los huesos del antebrazo (radio y cúbito), así como porciones proximales de los huesos metacarpianos. [7]

1.4.1 Huesos y articulaciones en el complejo articular de la muñeca.

La muñeca; observar Figura 1.3, que en su conjunto es una articulación condílea, es en realidad un conjunto complejo de articulaciones formado por la articulación radiocarpiana (elipsoidea), entre las carillas de la superficie articular del radio y los huesos escafoides y semilunar del carpo; radiocubital (trocoide), articulación trocoide entre cavidad sigmoidea de la cara interna del radio y la cabeza cubital. [7]

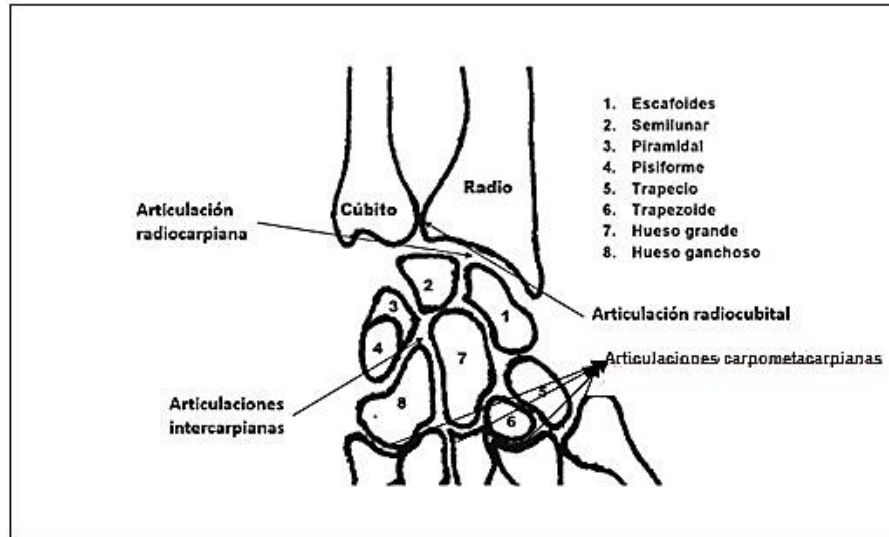


Figura 1.3 Complejo articular de la muñeca con sus huesos y articulaciones [7].

1.5 Biomecánica de la muñeca

La biomecánica es la disciplina que estudia los aspectos mecánicos que se producen en los organismos biológicos, que comprende a los segmentos óseos como palancas, las articulaciones como los puntos de apoyos, los músculos agonistas como la fuerza de potencia, y la sobrecarga como las fuerzas de resistencia. [7]

En relación con los elementos biomecánicos en el complejo articular de la muñeca, considerando el conjunto de huesos, articulaciones y músculos que lo forman, se observa que este sistema permite el desarrollo de distintos tipos de movimientos complejos en diferentes planos del espacio como se observa en la Figura 1.4 [7].

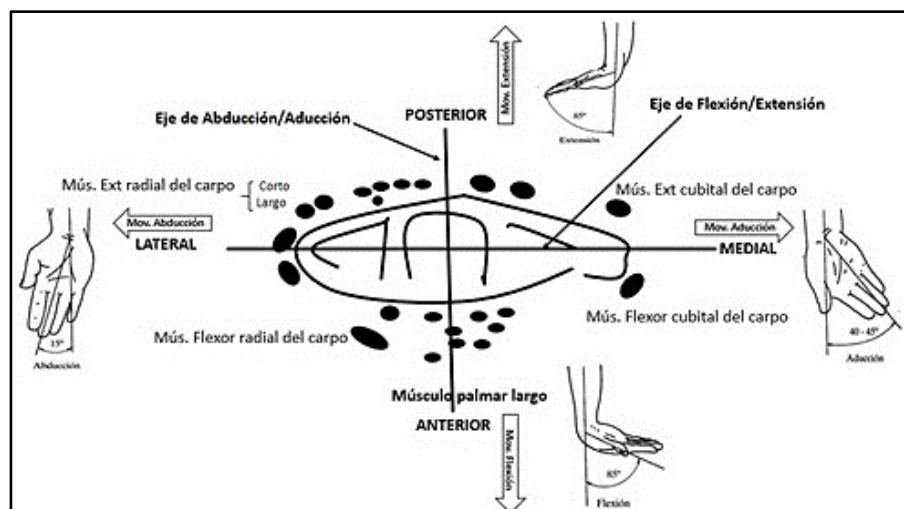


Figura 1.4 Relación de los músculos que desarrollan los movimientos del complejo articular de la muñeca [7].

Estos movimientos se han ido organizando como consecuencia del desarrollo filogenético, por lo que, con el tiempo y las necesidades funcionales de la mano, se han logrado movimientos en flexo-extensión, aducción-abducción. Estos movimientos se producen en el marco de una conveniente estabilidad de la articulación, lográndose con esto una adecuada funcionabilidad de la mano como órgano vital en el desarrollo evolutivo del homo sapiens. [7]

1.6 Movilidad de la muñeca

1.6.1 Movilidad básica

La muñeca es la articulación que une los huesos cubito y radio al carpo, considerada en conjunto, efectuara movimientos de flexión-extensión y abducción-aducción; aunque también, esta pudiese realizar movimientos compuestos, como la circunducción [8].

La flexión se muestra como la sección palmar de la mano, este movimiento tiene un ángulo de libertad de 0° hasta 80° ; la extensión se representa como el movimiento de la región dorsal de la mano cerca de la cara posterior del antebrazo, dicho movimiento tiene un ángulo de libertad de 0° hasta 70° [8]. Observar Figura 1.5.

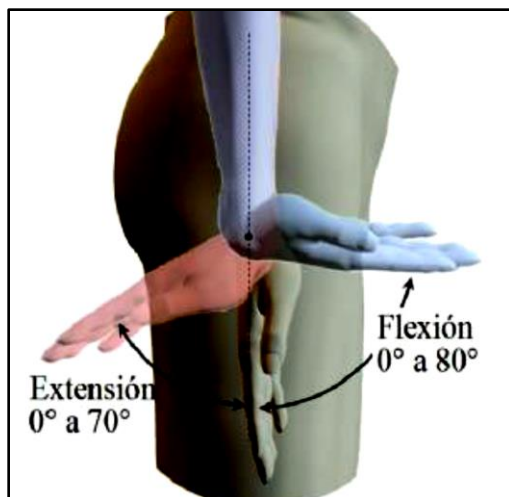


Figura 1.5 Movimiento extensión-flexión básico de la muñeca [8].

En el movimiento de aducción; observe la Figura 1.6, el dedo meñique se acerca al antebrazo con un ángulo de liberación de 0° a 30° , mientras que la abducción el pulgar se acerca al antebrazo con un ángulo de liberación de 0° hasta 20° . [8]

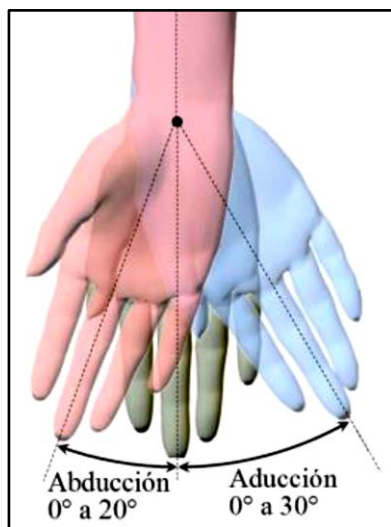


Figura 1.6 Movimiento abducción-aducción básico de la muñeca [8].

1.6.2 Movimiento flexión-extensión.

El movimiento de flexión, se inicia en la segunda hilera del carpo que provoca la tensión de los ligamentos de la articulación mediocarpiana (principalmente el ligamento piramidal-trapecio-trapezoide) para acabar moviendo el escafoides (que moverá el semilunar y piramidal); este movimiento de flexión varía entre unos 70° a 90° y en él las articulaciones intercarpianas intervienen en aproximadamente un 60 % y la articulación radiocarpiana en un 40 %. [9]

La flexión se consigue cuando se activan los músculos: palmar mayor, cubital anterior, abductor del pulgar, y los flexores de los dedos con estos en extensión, y se reduce de manera importante cuando se realiza de manera sinérgica a la flexión de los dedos o con la flexión previa de estos. [9]

El movimiento de extensión, aproximación de la cara dorsal de la mano hacia el dorso del antebrazo tiene un arco de movilidad entre 65° y 85° y, a diferencia del movimiento de flexión, este movimiento se debe fundamentalmente a la articulación radiocarpiana que aporta aproximadamente el 66 % del rango y menos a las articulaciones mediocarpianas que contribuyen con el 34 %; esto se debe, a que la cara articular del radio se extiende dorsalmente más que las caras articulares distales del semilunar y el escafoides; la extensión de la muñeca se reduce si se asocia a la extensión previa de los dedos. [9] En la Figura 1.7, se representa lo descrito sobre el movimiento de flexión y extensión.

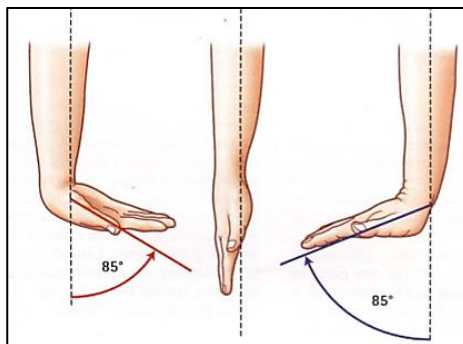


Figura 1.7 Representación del movimiento de muñeca flexión-extensión [6].

1.6.3 Movimiento abducción-aducción.

El movimiento de abducción es la inclinación radial de la muñeca, tiene una amplitud que oscila entre 15° y 25° con la muñeca en supinación, pero en general entre 20° y 60° , dado que la estiloides cubital es más corta que la radial; este movimiento se inicia en la segunda hilera del sistema de huesos del carpo que se mueve hacia el radio, mientras la primera hilera se mueve hacia cubital además de flexionarse, en este movimiento el escafoides y el semilunar se desplazan medialmente, llegando este último a articularse con el ligamento triangular interno; el 60 % de este movimiento es responsabilidad de la articulación mediocarpiana. [9]

El movimiento de aducción, inclinación cubital de la muñeca, se inicia en la segunda hilera que se mueve hacia cubital mientras que la primera se mueve a radial, generándose movimientos contrarios a la abducción en las dos hileras del hueso del carpo, al igual que el movimiento de abducción, el 66 % del movimiento es responsabilidad de la articulación mediocarpiana. [9] En la Figura 1.8, se observa la representación de lo antes mencionado.

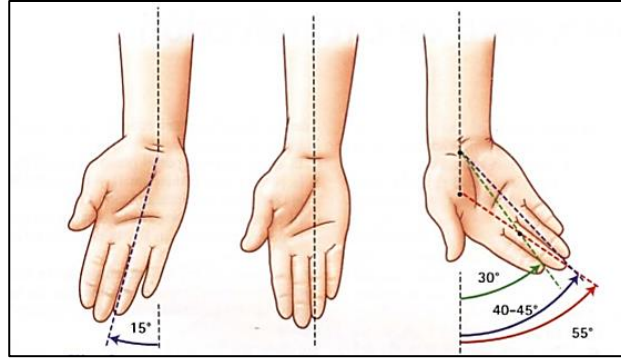


Figura 1.8 Representación del movimiento de la muñeca abducción-aducción [6].

1.7 Anatomía del antebrazo

El antebrazo está conformado por dos huesos: el cúbito y el radio, paralelos entre sí, en el sentido de la longitud del miembro, el uno por dentro y el otro por fuera, siendo el cúbito más largo que el radio; ambos huesos están unidos en sus extremos por articulaciones móviles, y en su parte media delimitan un espacio elíptico llamado espacio interóseo, ocupado por la membrana interósea que une transversalmente ambos huesos. [10] En la Figura 1.9 se distinguen tres regiones:

a). Anterior, conformada por ocho músculos, dispuestos en cuatro planos:

- Pronador redondo, palmar mayor, palmar menor y cubital anterior.
- Flexor común superficial de los dedos.
- Flexor propio del pulgar y flexor profundo común de los dedos.
- Pronador cuadrado.

b). Región externa, comprende cuatro músculos y son:

- Supinador largo, primer radial externo, segundo radial externo y supinador corto.

c). Región posterior, comprende ocho músculos dispuestos en dos capas:

- Capa superficial: extensor común de los dedos, extensor propio del dedo meñique, cubital posterior y el ancóneo.
- Capa profunda: abductor largo del pulgar, extensor corto del pulgar, extensor largo del pulgar y extensor propio del índice.

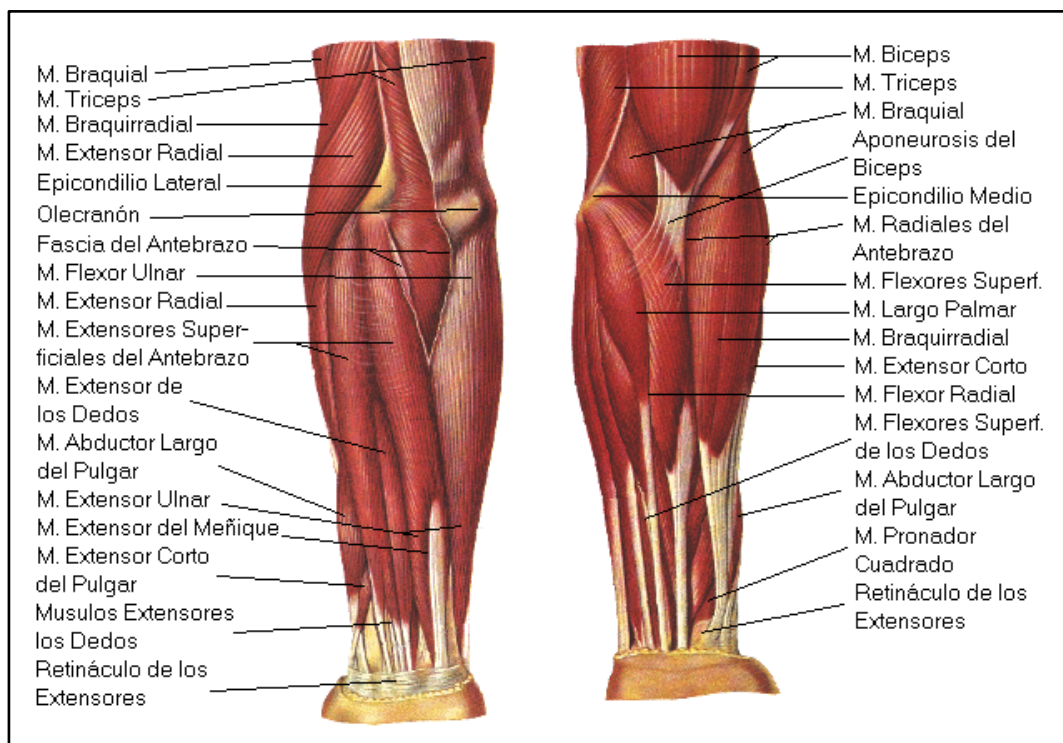


Figura 1.9 Anatomía del antebrazo [11].

Los músculos posteriores y externos están inervados por el nervio radial; de los anteriores, los dos fascículos internos del flexor común profundo y el cubital anterior, dependen del nervio cubital; los demás, del mediano. [10]

1.8 Biomecánica del antebrazo

La pronosupinación es el movimiento de rotación del antebrazo en torno a su eje longitudinal, para lo cual es necesaria la intervención de la asociación mecánica de dos articulaciones: la radio cubital superior y la radio cubital inferior. [10]

La pronosupinación es un movimiento completo, tan específico, que engloba articulaciones distantes y no es del codo ni del carpo, sino del antebrazo y se lleva a cabo a través de la integridad de la articulación húmero cubital, de la membrana interósea, de la articulación radio cubital distal, del fibrocartílago triangular y gracias a la especial forma del cúbito (recta) y el radio (curvada); En la supinación intervienen el supinador corto y el bíceps; en la pronación actúan el pronador cuadrado y el pronador redondo. [10]

1.9 Instrumentos médicos

“Los instrumentos médicos usados para medir la amplitud del movimiento articular se llaman goniómetros, o artrómetros. Las herramientas, a pesar de variar en tamaño forma y apariencia, poseen toda la capacidad de proporcionar información específica observando la movilidad articular” [12].

1.9.1 Goniómetro

El goniómetro universal es un instrumento terapéutico económico, práctico y sencillo de usar; observe Figura 1.10, que está formado por dos brazos y un cuerpo en el que se sitúa un transportador de ángulos que permite obtener la medida [13].

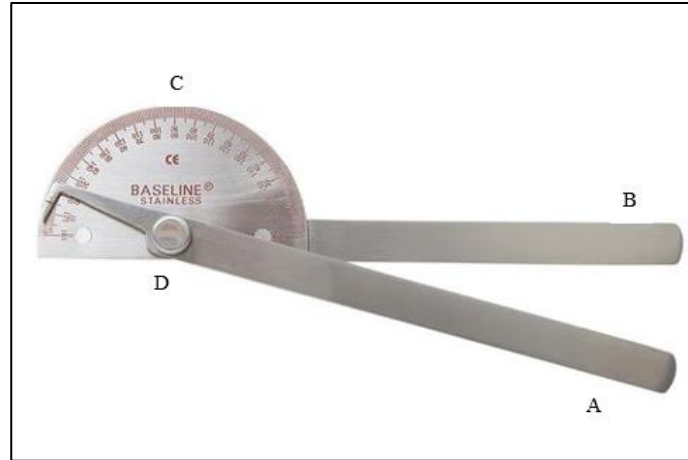


Figura 1.10 Goniómetro metálico que muestra un brazo móvil (A), un brazo fijo (B), un cuerpo transportador (C) y un eje (D) [12].

1.10 Extracción de características EMG

La extracción de características es un método útil para extraer información oculta en las EMG permitiendo eliminar información indeseada relacionada a interferencia y ruido [14]. Estas características se aprovechan para construir modelos capaces de predecir la clase de datos muestreados [15].

Las características más comunes de EMG son las características del dominio del tiempo (raíz media cuadrada, máximo valor absoluto, cruce de cero, cambio de signo de pendiente, etc.) y las características del dominio de la frecuencia (frecuencia promedio, frecuencia media, frecuencia máxima, potencia media, potencia total, etc.) [16]. La extracción de características posee tres categorías de análisis.

- En el dominio del tiempo
- En el dominio de la frecuencia
- En el dominio tiempo-frecuencia o tiempo-escala.

Los primeros dos grupos de categorías son los más utilizados en el control mioeléctrico de dispositivos asistidos, debido a que las características en el dominio tiempo-frecuencia no pueden ser usadas directamente por su alta dimensión [14].

1.10.1 Análisis en el dominio del tiempo

Las funciones en el dominio del tiempo se usan más comúnmente para el reconocimiento de patrones EMG, esto se debe a que son fáciles y rápidos de calcular, ya que no requieren ninguna transformación; las características del dominio del tiempo se calculan en función de la amplitud de las señales de entrada [17]. Entre las más conocidas están;

- Integral de la señal EMG (IEMG)
- Valor absoluto medio (MAV)
- Raíz Media Cuadrática (RMS)
- Varianza de la señal EMG (VAR)
- Longitud de Onda (WL)
- Simple Cuadrado Integral (SSI)
- Cambios en el Signo de la Señal (SSC)
- Amplitud de Willison (WAMP)
- Valor absoluto medio Modificado (MMAV1)
- Valor absoluto medio Modificado (MMAV2)

1.10.2 Análisis en el dominio de la frecuencia

Las características extraídas utilizando los dominios de frecuencia se basan normalmente en la densidad espectral de potencia (PSD) estimada de una señal [17]. El cual refleja la razón de activación de las unidades motoras por debajo de 40Hz y la morfología de los potenciales por encima de los 40Hz [14].

Las características de dominio de frecuencia en comparación con las características de dominio de tiempo tienden a requerir más recursos computacionales y tiempo [17]. Dichas características son:

- Coeficientes Autoregresivos (AR)
- Frecuencia Media (FMD)
- Frecuencia Promedio (FMN)
- Frecuencia Media Modificada (MMDF)
- Frecuencia Promedio Modificada (MMNF)
- Densidad de espectro de potencia (PSD)

1.10.3 Análisis en el dominio tiempo frecuencia

La transformada wavelet es uno de los métodos más utilizados en el dominio tiempo-escala, que a partir de los coeficientes en el dominio transformado permiten extraer información característica; Adicionalmente, el análisis wavelet puede comprimir y eliminar ruido en la señal sin una apreciable degradación, una generalización de la transformada wavelet es el paquete wavelet, que permite una mejor adaptación de la señal en el dominio tiempo-escala [17].

Generalmente, en el proceso de extracción de características, los parámetros temporales son mezclados con otros métodos, del mismo dominio temporal y frecuencial, con el objetivo de aumentar la separabilidad de las clases, exactitud y robustez en el proceso de reconocimiento de patrones; los resultados en lo relacionado al desempeño obtenido en la clasificación de patrones mioeléctricos, recomiendan la combinación de parámetros como el valor absoluto medio, la longitud de la señal, la amplitud de Willison y los modelos de predicción, basados en modelos auto-regresivos[17]. Algunos de estos métodos son:

- Transformada de Fourier (STFT)
- Transformada Wavelet (WT).
- Transformación de paquetes wavelet (WPT)

1.11 Métodos de clasificación

El aprendizaje es la clave de la plasticidad de una neurored y esencialmente es el proceso en el que se adaptan las sinapsis, para que la red responda de un modo distinto a los estímulos del medio; recordemos que, en una neurored, toda la información adquirida se guarda en el valor de cada peso sináptico; de hecho, las neuronas de la mayor parte de los seres vivos con sistema nervioso, desde un caracol hasta el hombre son esencialmente iguales [18].

El aprendizaje automático emplea dos tipos de técnicas; como se observa en la Figura 1.11, el aprendizaje supervisado, que entrena un modelo con datos de entrada y salida conocidos para que pueda predecir salidas futuras, y el aprendizaje no supervisado, que encuentra patrones ocultos o estructuras intrínsecas en los datos de entrada [19].

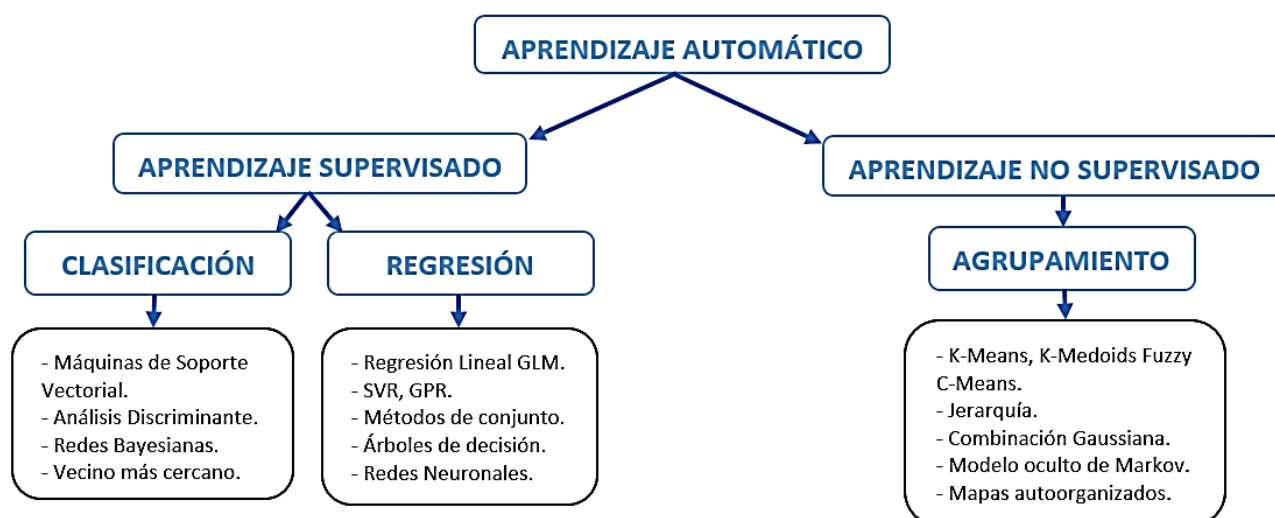


Figura 1.11 Técnicas de aprendizaje automático [19].

1.12 Aprendizaje Predictivo o Supervisado

En el enfoque de aprendizaje predictivo o supervisado, el objetivo es aprender un mapeo de las entradas x a las salidas y , dado un conjunto etiquetado de pares entrada-salida; los algoritmos trabajan con datos “etiquetados” (labeled data), intentado encontrar una función que dadas las variables de entrada (input data) les asigne la etiqueta de salida adecuada [20].

1.12.1 Redes Neuronales Artificiales (RNA)

Las Redes Neuronales Artificiales (RNA) o simplemente Neurored es un procesador de información, de distribución altamente paralela, constituido por muchas unidades sencillas de procesamiento llamadas neuronas [18]. Las neuroredes se caracterizan principalmente por:

- Tener una inclinación natural a adquirir el conocimiento a través de la experiencia, el cual es almacenado, al igual que en el cerebro, en el peso relativo de las conexiones interneuronales [18].
- Tienen una altísima plasticidad y gran adaptabilidad, son capaces de cambiar dinámicamente junto con el medio [18].
- Poseen un alto nivel de tolerancia a fallas, es decir, pueden sufrir un daño considerable y continuar teniendo un buen comportamiento, al igual como ocurre en los sistemas biológicos [18].

1.12.1.1 Redes Neuronales de Perceptrones de capas Múltiples

Las redes neuronales de Perceptrones de capas múltiples (MLP) son redes formadas por nodos o neuronas interconectadas en varios tipos de configuraciones y con una organización jerárquica, que tratan de interactuar con los objetos del mundo real de la misma manera que un sistema nervioso biológico [21].

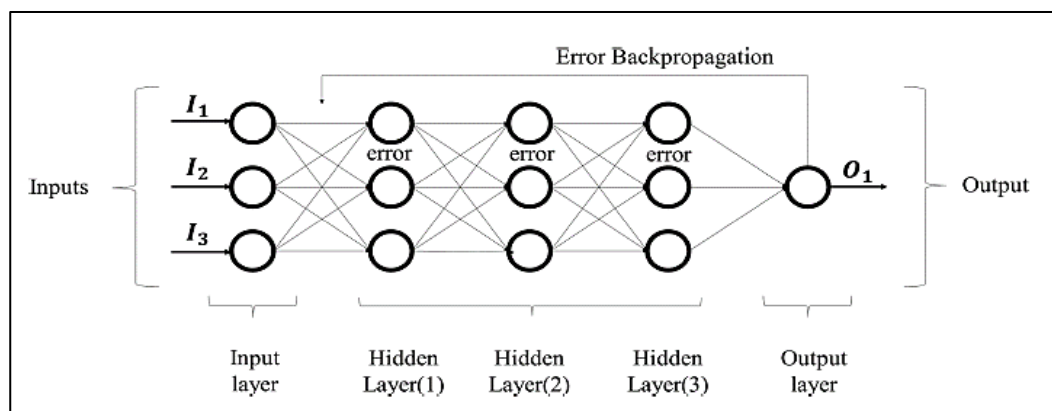


Figura 1.12 Red Neuronal MLP con metodología de aprendizaje de retro-propagación [21].

En la Figura 1.12, se muestra un esquema de la arquitectura y el proceso de aprendizaje de la MLP, las MLP están constituidas por neuronas interconectadas y dispuestas en tres o más capas, los datos entran a través de la capa de entrada, pasan a través de las capas ocultas y salen a través de la capa de salida; Cabe mencionar, que las capas ocultas pueden estar constituidas por varias capas [21].

Entre las más conocidas MLP está la Red Neuronal Backpropagation o retropropagación; observar Figura 1.13, es un algoritmo de descenso por gradiente, de aprendizaje de carácter supervisado que se emplea para redes multicapa [22].

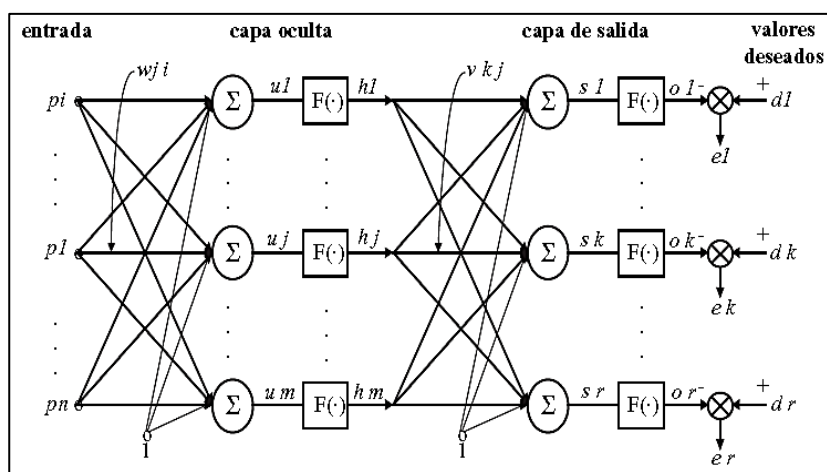


Figura 1.13 Red Neuronal de retro-propagación [23].

Las redes multicapas necesitan de una ley de aprendizaje adaptada para el cambio de los pesos sinápticos, ya que con la regla delta, sólo los pesos entre la penúltima y la última capa serían modificados; por ello, la regla delta generalizada o ley de retropropagación del gradiente fue propuesta, que para ser operacional, necesita que la función de transferencia en cada neurona sea continua; las neuronas de la capa de salida tienen estados 0 ó 1 [23]. Debido a esto la capa oculta de la red presenta varios tipos de función de activación.

Función de activación

La función de activación se encarga de devolver una salida a partir de un valor de entrada, normalmente el conjunto de valores de salida en un rango determinado como (0,1) o (-1,1); con lo cual, se buscan funciones que las derivadas sean simples, para minimizar con ello el coste computacional.

a.) Función de activación Sigmoide (logsig)

La función sigmoide transforma los valores introducidos a una escala (0,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0 [24].

$$f(n) = \frac{1}{1 - e^{-n}} \quad (1)$$

La función sigmoide; observe Figura 1.14, posee características como:

- Satura y mata el gradiente
- Lenta convergencia
- Está acotada entre 0 y 1
- Buen rendimiento en la última capa
- No está centrada en el cero

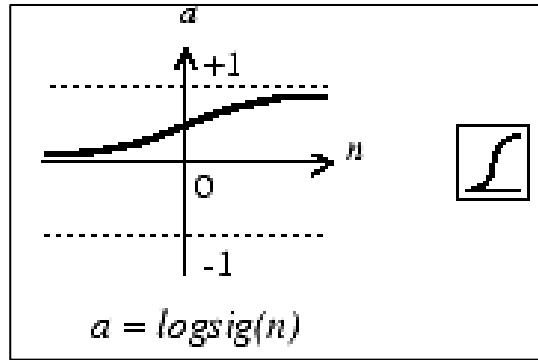


Figura 1.14 Función de activación Logística o sigmoideal [24].

b.) Función de activación Tangente Hiperbólica (tansig)

La función tangente hiperbólica transforma los valores introducidos a una escala (-1,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a -1 [24].

$$f(n) = \frac{2}{1 + e^{-2n}} - 1 \quad (2)$$

La función tangente; observar Figura 1.15, presenta características como:

- Muy similar a la signoide
- Satura y mata el gradiente.
- Lenta convergencia.
- Centrada en 0.
- Está acotada entre -1 y 1.
- Se utiliza para decidir entre una opción y la contraria.
- Buen desempeño en redes recurrentes.

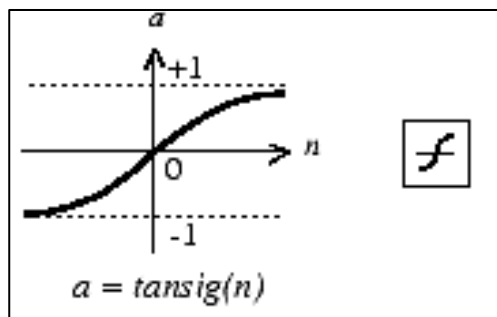


Figura 1.15 Función de activación tangente hiperbólica [24].

c.) Función de activación Lineal (purelin)

La función de activación Lineal rectificadora transforma los valores introducidos multiplicando los negativos por un coeficiente rectificativo y dejando los positivos según entran [24].

$$f(n) = n \quad (3)$$

La función Lineal; observar Figura 1.16, dispone de características como:

- Penaliza los negativos mediante un coeficiente rectificador.
- No está acotada.
- Se comporta bien con imágenes.
- Buen desempeño en redes convolucionales.

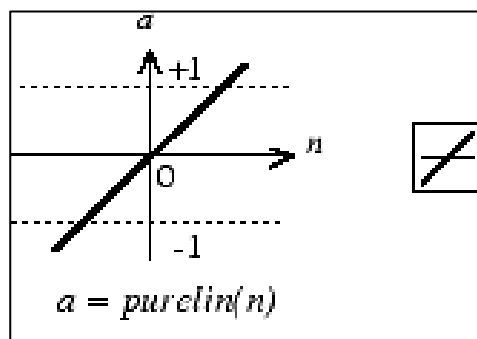


Figura 1.16 Función de activación Lineal [24].

1.12.1.2 Redes Neuronales Monocapa

En una red monocapa o unicapa, las neuronas de salida pueden ser lineales o no lineales, se acostumbra no contabilizar la capa de entrada, por lo tanto se dice que la red de la Figura 1.17 es unicapa, las sinapsis obviamente están ordenadas en una matriz [18].

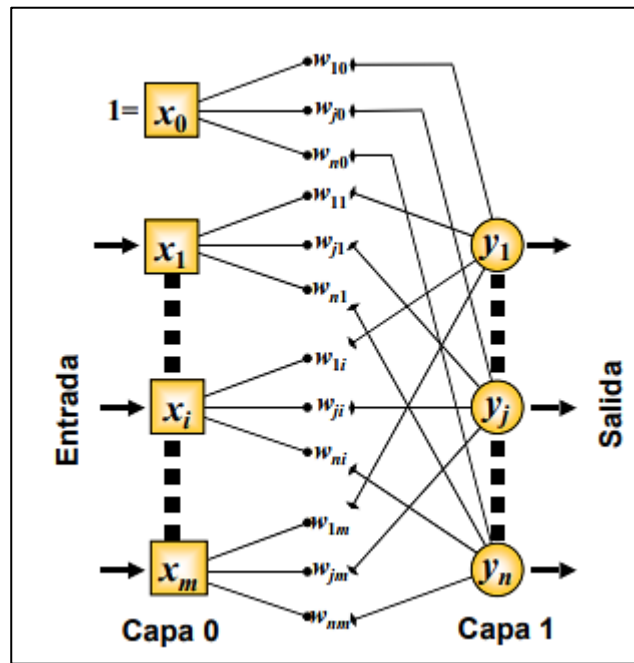


Figura 1.17 Red Neuronal Unicapa [18].

1.13 Aprendizaje Descriptivo o No Supervisado

En este tipo de aprendizaje solo se nos dan entradas, y el objetivo es encontrar “patrones interesantes” en los datos, esto a veces se llama descubrimiento del conocimiento; este es un problema mucho menos definido, ya que no se nos dice qué tipos de patrones buscar, y no hay una métrica de error obvia (a diferencia del aprendizaje supervisado, donde podemos comparar nuestra predicción de y para una x dada) [20].

CAPÍTULO II: METODOLOGÍA

2.1 Elección de músculos principales

Tras el análisis en la investigación se opta por establecer; ver Tabla 2.1, el rango de movilidad de la muñeca y el grupo de músculos superficiales e internos que intervienen en cada movimiento de estudio.

Tabla 2.1

Músculos que intervienen en cada movimiento y rango de movilidad

MOVIMIENTO	RANGO	MÚSCULOS SUPERFICIALES	MÚSCULOS INTERNOS
FLEXIÓN	0-80°	Palmar mayor. Flexor radial del carpo. Flexor cubital del carpo.	Flexor largo del pulgar. Flexor superficial de los dedos.
EXTENSIÓN	0-70°	Extensor cubital del carpo Extensor de los dedos. Extensor radial corto del carpo Extensor radial largo del carpo	Extensor del índice
ABDUCCIÓN (desviación radial)	0-20°	Flexor radial del carpo Extensor radial largo del carpo Extensor radial corto del carpo Extensor largo del pulgar	Abductor largo del pulgar. Extensor corto del pulgar.
ADUCCIÓN (desviación cubital)	0-30°	Flexor cubital del carpo Extensor cubital del carpo	

Una vez establecido correctamente los grupos musculares, se procede a realizar la selección del músculo de estudio por cada movimiento de la muñeca (flexión-extensión y abducción-aducción); ver Tabla 2.2, basándose en parámetros fundamentales como: La superficialidad, el mayor aporte a la movilidad, tamaño y masa del vientre muscular [25]. Además, de tomar en cuenta bibliografía y las recomendaciones y sugerencias de especialistas en Terapia física médica [26].

Tabla 2 .2
 Selección de los 4 músculos principales

MOVIMIENTO	MÚSCULO SELECCIONADO	ORIGEN	INSERCIÓN	INERVACIÓN	IRRIGACIÓN
FLEXIÓN	Flexor radial del carpo	Epicóndilo medial del húmero.	Base del 2do metacarpiano	Nervio cubital (C6,C7)	Arteria Cubital
EXTENSIÓN	Extensor cubital del carpo o cubital posterior	Epicóndilo lateral del húmero, por el tendón común.	Tubérculo del lado cubital de la base del 5to metacarpiano.	Rama profunda del nervio radial (C6, C8)	Arteria cubital.
ABDUCCIÓN (desviación radial)	Extensor largo del pulgar	Parte lateral del tercio medio de la cara dorsal del cuerpo del cúbito	Base de la última falange del pulgar	Nervio radial profundo (C6, C7)	Arteria interósea posterior
ADUCCIÓN (desviación cubital)	Flexor cubital del carpo	Epicóndilo medial del húmero y el olecranon del cúbito	Hueso pisiforme, hueso ganchoso, base del 5° metacarpiano	Nervio Cubital (C7, C8)	Arteria cubital

Como resultado se obtuvo que los músculos: Flexor radial del carpo, extensor cubital del carpo, extensor largo del pulgar y flexor cubital del carpo se usaran para el estudio de los movimientos: Flexión, Extensión, Abducción y Aducción respectivamente.

2.2 Ubicación de los electrodos

Según las recomendaciones producidas por la European Concerted Action Surface EMG para la evaluación no invasiva del músculo, indica que las medidas de un electrodo en su diámetro deben ser menor o igual a 10mm y la separación entre ambos electrodos debe ser menor o igual que 20 mm [27].

La ubicación de los electrodos; ver Figura 2.18, se realiza en base a los siguientes conectores:

- L (Negro): Punto de referencia como el hueso, o un músculo irrelevante
- L (Rojo): Junto al electrodo R.
- R (Blanco): Centro del músculo

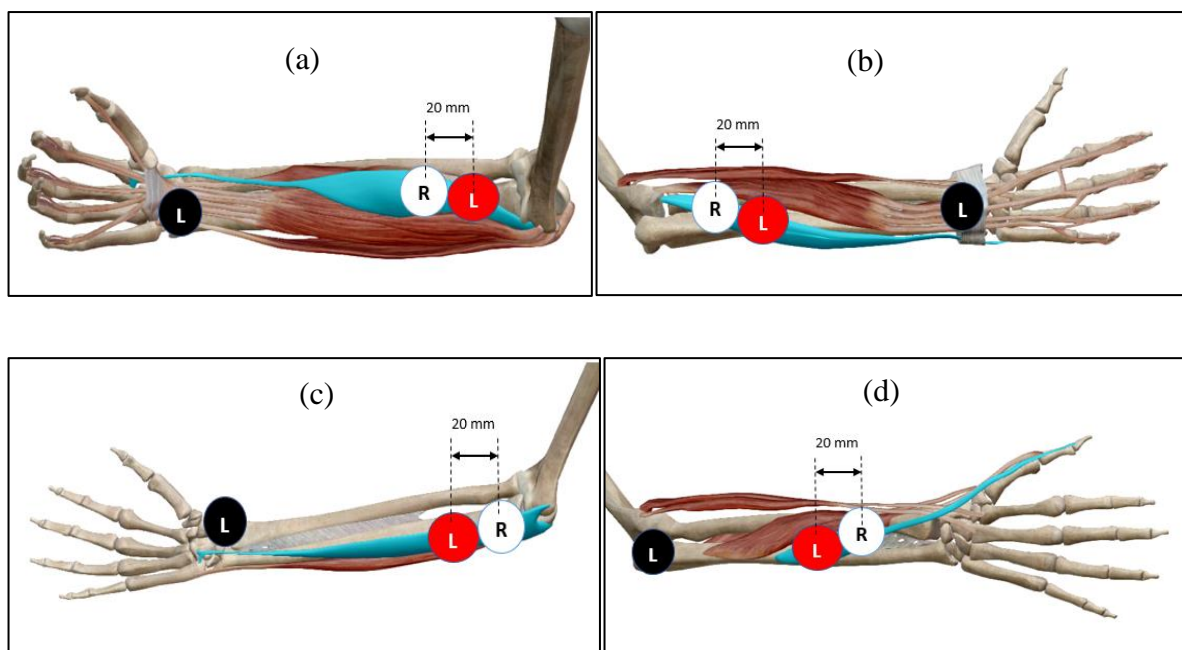


Figura 2.18 Ubicación de los electrodos en los músculos (a) Flexor radial del carpo. (b) Extensor cubital del carpo. (c) Flexor cubital del carpo. (d) Extensor largo del pulgar.

Los electrodos, son elementos primarios de un EMG que convierten las corrientes iónicas de los músculos del cuerpo humano en corrientes eléctricas, se utilizan electrodos de superficie, que están en contacto con la piel del paciente [28]. Para la ubicación del electrodo se sigue un protocolo de medición [29]. Tomando en consideración los siguientes aspectos:

- 1) No se debe aplicar lociones ni cremas sobre el área de análisis un día antes de realizar la prueba.
- 2) Se acomoda el antebrazo en una posición de descanso.
- 3) La zona donde se localiza los electrodos estaba debidamente depilada.
- 4) Se limpia la piel con alcohol antiséptico.
- 5) A continuación, se localizan los electrodos.
- 6) Se finalizó con el test Clínico.

Los electrodos tienen un modelo eléctrico de conexión con la superficie de la piel como se observa en la Figura 2.19.

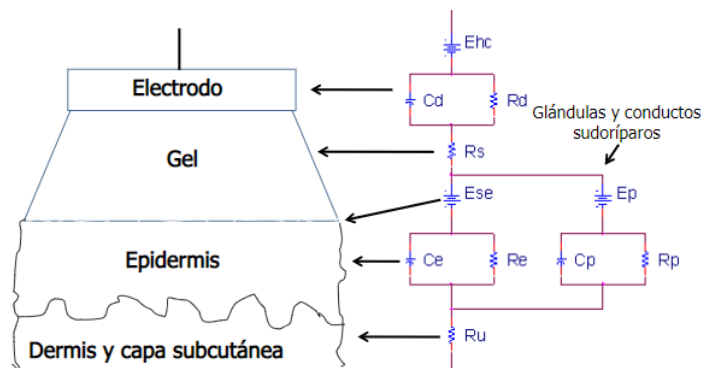


Figura 2.19 Modelo eléctrico de conexión del electrodo [30].

2.2.1 Ubicación en Flexor radial del carpo

La ubicación de los electrodos debe realizarse con las normativas respectivas y bajo la supervisión del médico especialista. En el caso del músculo Flexor radial del carpo primero se debe colocar la parte dorsal de la mano y la parte posterior del antebrazo sobre una superficie plana como se observa en la Figura 2.20 (b), posteriormente teniendo relajado los músculos se procede a colocar los electrodos como se muestran en la Figura 2.20 (a).

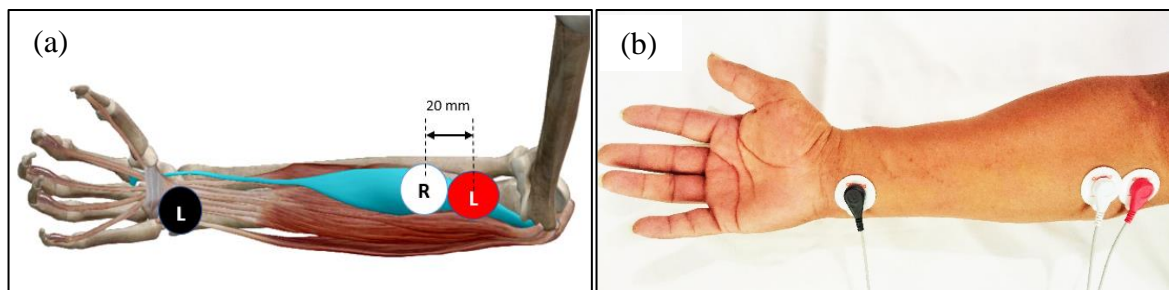


Figura 2.20 Ubicación de electrodos al músculo Flexor radial del carpo. (a) Ubicación en simulador. (b) Ubicación real en el usuario.

2.2.2 Ubicación en Extensor cubital del carpo

En el músculo Extensor cubital del carpo se coloca la palma de la mano y la parte anterior del antebrazo sobre una superficie plana; a continuación, se colocan los electrodos respectivos teniendo los músculos relajados como se ve en la Figura 2.21.

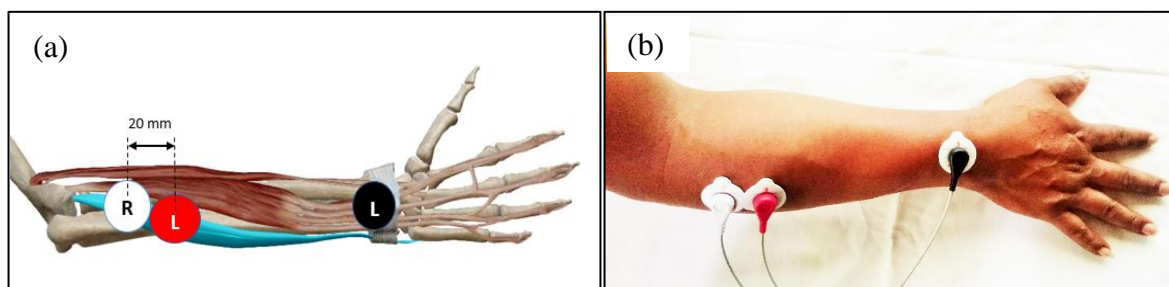


Figura 2.21 Ubicación de electrodos al músculo Extensor cubital del carpo. (a) Ubicación en simulador. (b) Ubicación real en el usuario.

2.2.3 Ubicación en Flexor cubital del carpo

Para el músculo Flexor cubital del carpo se debe ubicar la palma de la mano y la parte anterior del antebrazo sobre una superficie plana; a continuación, se detecta el músculo de estudio y procede a la colocación de los electrodos respectivamente teniendo los músculos relajados como se ve en la Figura 2.22.

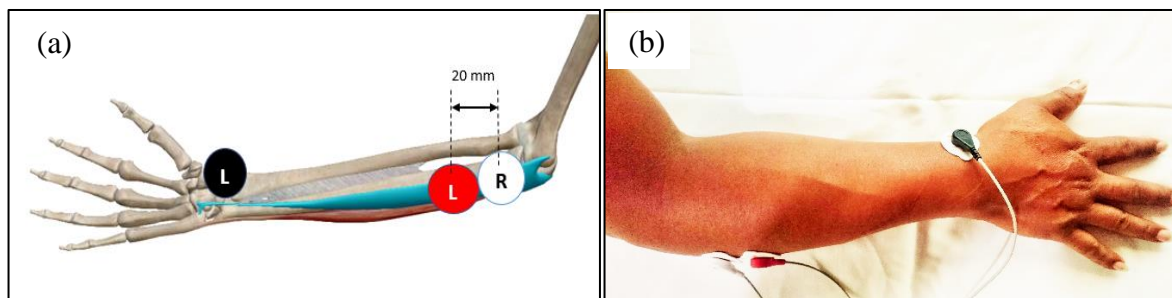


Figura 2.22 Ubicación de electrodos al músculo Flexor cubital del carpo. (a) Ubicación en simulador. (b) Ubicación real en el usuario.

2.2.4 Ubicación en Extensor largo del pulgar

En el músculo Extensor largo del pulgar se sitúa la palma de la mano y la parte anterior del antebrazo sobre una superficie plana; a continuación, se detecta el músculo de estudio y procede a la colocación de los electrodos según las normativas respectivas teniendo los músculos totalmente relajados como se ve en la Figura 2.23.

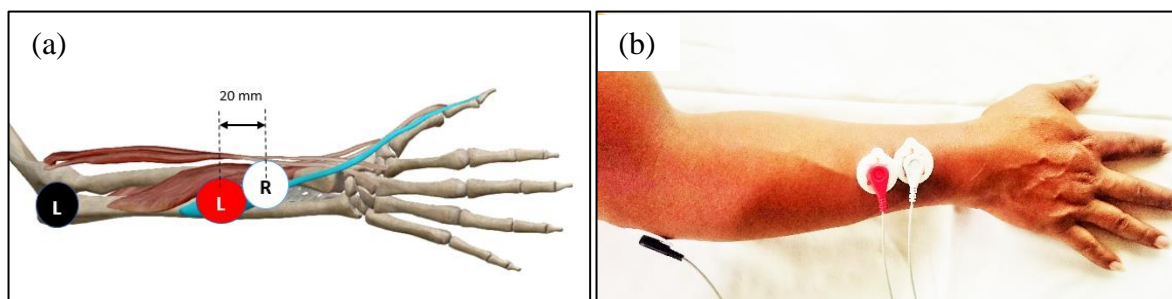


Figura 2.23 Ubicación de electrodos al músculo Extensor largo del pulgar. (a) Ubicación en simulador. (b) Ubicación real en el usuario.

2.3 Selección de ángulos de estudio

Para demostrar la estimación de la posición de la muñeca se plantea estudiar ángulos estandarizados; en este caso, 3 ángulos en el movimiento de Aducción y 4 ángulos en los movimientos de Flexión, Extensión y Abducción, ver Tabla 2.3.

Tabla 2.3
Ángulos de estudio para los 4 movimientos de la muñeca.

MOVIMIENTO	1ER ÁNGULO	2DO ÁNGULO	3ER ÁNGULO	4TO ÁNGULO
FLEXIÓN	0° – 30°	0° – 45°	0 – 60°	0° – 80°
EXTENSIÓN	0° – 30°	0° – 45°	0 – 60°	0° – 70°
ABDUCCIÓN	0° – 5°	0° – 10°	0 – 15°	0° – 20°
ADUCCIÓN	0° – 10°	0 – 20°	0° – 30°	

2.4 Diseño Conceptual

La mayor parte de las veces, para decidir entre diversas soluciones (especialmente en la etapa de diseño conceptual) basta conocer el orden de preferencia de su evaluación global. Es por ello que se recomienda el método ordinal corregido de criterios ponderados que, sin la necesidad de evaluar los parámetros de cada propiedad y sin tener que estimar numéricamente el peso de cada criterio, permite obtener resultados globales suficientemente significativos. [31]

Se basa en unas tablas donde cada criterio (o solución, para un determinado criterio) se confronta con los restantes criterios (o soluciones) y se asignan los valores siguientes: [31]

- 1 Si el criterio (o solución) de las filas es superior (o mejor; $>$) que el de las columnas.
- 0,5 Si el criterio (o solución) de las filas es equivalente ($=$) al de las columnas.
- 0 Si el criterio (o solución) de las filas es inferior (o peor; $<$) que el de las columnas.

Luego, para cada criterio (o solución), se suman los valores asignados en relación a los restantes criterios (o soluciones) al que se le añade una unidad (para evitar que el criterio o solución menos favorable tenga una valoración nula); después, en otra columna se calculan los valores ponderados para cada criterio (o solución). [31]

Finalmente, la evaluación total para cada solución resulta de la suma de productos de los pesos específicos de cada solución por el peso específico del respectivo criterio [31].

2.4.1 Selección de equipos para el sistema

2.4.1.1 Selección de Tarjeta de Procesamiento.

Para elegir la Tarjeta de procesamiento, se toma como opciones 4 alternativas que pasan por un proceso de selección basado en el diseño conceptual, iniciando con una evaluación donde se dará un peso a cada criterio según la necesidad.

- SOLUCIÓN A = Arduino UNO
- SOLUCIÓN B = Arduino MEGA
- SOLUCIÓN C = Raspberry PI
- SOLUCIÓN D = Discovery STM32F4

a. Evaluación del peso específico de cada criterio

Tabla 2.4

Evaluación del peso específico de cada criterio para la selección de tarjeta de procesamiento.

Hardware > Velocidad = Precio					
CRITERIO	Hardware	Velocidad	Precio	$\sum + 1$	Ponderación
Hardware		1	1	3	0,5
Velocidad	0		0,5	1,5	0,25
Precio	0	0,5		1,5	0,25
			SUMA	6	1

b. Evaluación del peso específico del criterio HARDWARE

Tabla 2.5

Evaluación del peso específico del criterio HARDWARE.

SOLUCIÓN B = SOLUCIÓN D > SOLUCIÓN C > SOLUCIÓN A						
HARDWARE	SOLUCIÓN A	SOLUCIÓN B	SOLUCIÓN C	SOLUCIÓN D	$\sum + 1$	Ponderación
SOLUCIÓN A		0	0	0	1	0,1
SOLUCIÓN B	1		1	0,5	3,5	0,35
SOLUCIÓN C	1	0		0	2	0,2
SOLUCIÓN D	1	0,5	1		3,5	0,35
				SUMA	10	1

c. Evaluación del peso específico del criterio VELOCIDAD

Tabla 2.6

Evaluación del peso específico del criterio VELOCIDAD.

SOLUCIÓN C > SOLUCIÓN D > SOLUCIÓN B > SOLUCIÓN A						
VELOCIDAD	SOLUCIÓN A	SOLUCIÓN B	SOLUCIÓN C	SOLUCIÓN D	$\Sigma + 1$	Ponderación
SOLUCIÓN A		0	0	0	1	0,1
SOLUCIÓN B	1		0	0	2	0,2
SOLUCIÓN C	1	1		1	4	0,4
SOLUCIÓN D	1	1	0		3	0,3
				SUMA	10	1

d. Evaluación del peso específico del criterio PRECIO (bajo)

Tabla 2.7

Evaluación del peso específico del criterio PRECIO (bajo).

SOLUCIÓN A > SOLUCIÓN D = SOLUCIÓN B > SOLUCIÓN C						
PRECIO	SOLUCIÓN A	SOLUCIÓN B	SOLUCIÓN C	SOLUCIÓN D	$\Sigma + 1$	Ponderación
SOLUCIÓN A		1	1	1	4	0,4
SOLUCIÓN B	0		1	0,5	2,5	0,25
SOLUCIÓN C	0	0		0	1	0,1
SOLUCIÓN D	0	0,5	1		2,5	0,25
				SUMA	10	1

e. Tabla de conclusiones

Tabla 2.8

Tabla de conclusiones para la selección de tarjeta de procesamiento.

CONCLUSIÓN	HARDWARE	VELOCIDAD	PRECIO	Σ	PRIORIDAD
SOLUCIÓN A	0,1	0,1	0,4	0,6	4
SOLUCIÓN B	0,35	0,2	0,25	0,8	2
SOLUCIÓN C	0,2	0,4	0,1	0,7	3
SOLUCIÓN D	0,35	0,3	0,25	0,9	1

Como resultado en la tabla de conclusiones; ver Tabla 2.8, se obtiene que la **SOLUCIÓN D** es óptima para este sistema, en este caso la tarjeta Discovery STM32F4 que posee las siguientes características:

- 16 convertidores ADC con una resolución 12 bit
- Salidas de alimentación de 3 y 5 V
- Oscilador interno PLL hasta 168MHz
- Conector Mini-USB para programación/depuración.
- Conector Micro-USB OTG “Host” o “Device”.

Como se ve en las características, la tarjeta de procesamiento; observe Figura 2.24, permite adquirir muestras con una resolución de adquisición de 12 bits es decir 4096 muestras por fracción de tiempo, reduciendo la pérdida de información durante el proceso de adquisición [32].

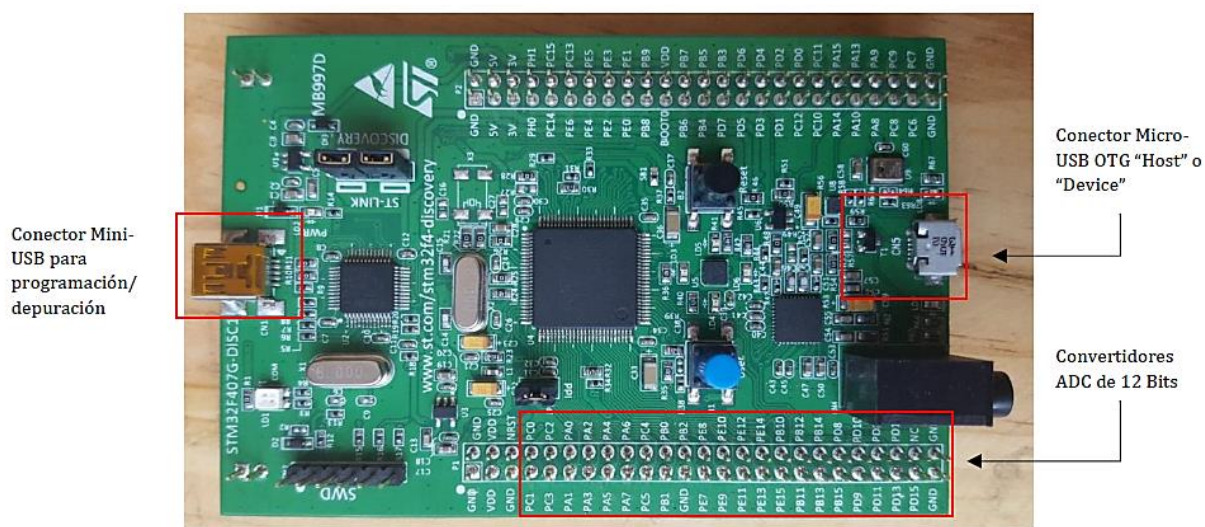


Figura 2.24 Tarjeta de procesamiento Discovery STM32F4 [32].

2.4.1.2 Selección de Tarjeta de Adquisición.

Para elegir la Tarjeta de adquisición de señales electromiográficas, se toma como opciones 2 alternativas que pasan por un proceso de selección basado en el diseño conceptual.

- SOLUCIÓN A = OLIMEX EMG/EKG
- SOLUCIÓN B = MYOWARE

a. Evaluación del peso específico de cada criterio.

Tabla 2.9

Evaluación del peso específico de cada criterio para la selección de la tarjeta de adquisición.

Precisión > Compatibilidad = Precio					
CRITERIO	Precisión	Compatibilidad	Precio	$\Sigma + 1$	Ponderación
Precisión		1	1	3	0,5
Compatibilidad	0		0,5	1,5	0,25
Precio	0	0,5		1,5	0,25
			SUMA	6	1

b. Evaluación del peso específico del criterio PRECISIÓN

Tabla 2.10

Evaluación del peso específico del criterio PRECISIÓN.

SOLUCIÓN A > SOLUCIÓN B				
PRECISIÓN	SOLUCIÓN A	SOLUCIÓN B	$\Sigma + 1$	Ponderación
SOLUCIÓN A		1	2	0,666666667
SOLUCIÓN B	0		1	0,333333333
		SUMA	3	1

c. Evaluación del peso específico del criterio COMPATIBILIDAD

Tabla 2.11

Evaluación del peso específico del criterio COMPATIBILIDAD.

SOLUCIÓN A > SOLUCIÓN B				
COMPATIBILIDAD	SOLUCIÓN A	SOLUCIÓN B	$\Sigma + 1$	Ponderación
SOLUCIÓN A		1	2	0,666666667
SOLUCIÓN B	0		1	0,333333333
		SUMA	3	1

d. Evaluación del peso específico del criterio PRECIO (bajo)

Tabla 2.12

Evaluación del peso específico del criterio PRECIO (bajo).

SOLUCIÓN B > SOLUCIÓN A				
PRECIO	SOLUCIÓN A	SOLUCIÓN B	$\Sigma + 1$	Ponderación
SOLUCIÓN A		0	1	0,333333333
SOLUCIÓN B	1		2	0,666666667
		SUMA	3	1

e. Tabla de conclusiones

Tabla 2.13

Tabla de conclusiones para la selección de la tarjeta de adquisición.

CONCLUSIÓN	PRECISIÓN	COMPATIBILIDAD	PRECIO	Σ	PRIORIDAD
SOLUCIÓN A	0,666666667	0,666666667	0,333333333	1,666666667	1
SOLUCIÓN B	0,333333333	0,333333333	0,666666667	1,333333333	2

Como resultado en la tabla de conclusiones; ver Tabla 2.13, se obtiene que la **SOLUCIÓN A** es óptima para este sistema, en este caso la tarjeta OLIMEX EMG/EKG; ver Figura 2.25, posee las siguientes características.

- Adecuado para electrocardiografía - monitoreo y recopilación de datos (bioseñales).
- Facilidad de ampliación de hasta 6 canales de adquisición EMG/EKG en cascada.
- Posee Cable de conectores, 2 electrodos (1 canal), 1 electrodo DLR, ver Figura 2.26.
- Conector de entrada para electrodos pasivos o activos.
- Funciona con alimentación de 3.3V y 5V.

Al ser una tarjeta de amplificación de señales de arquitectura abierta, no es necesario de extensas líneas de programación, pues es compatible con placas de comunicación como arduino, PIC y Discovery STM32F4 [32]. Al igual que trabaja con plataformas de programación como son:

- Arduino IDE
- Matlab, Simulink
- LabVIEW

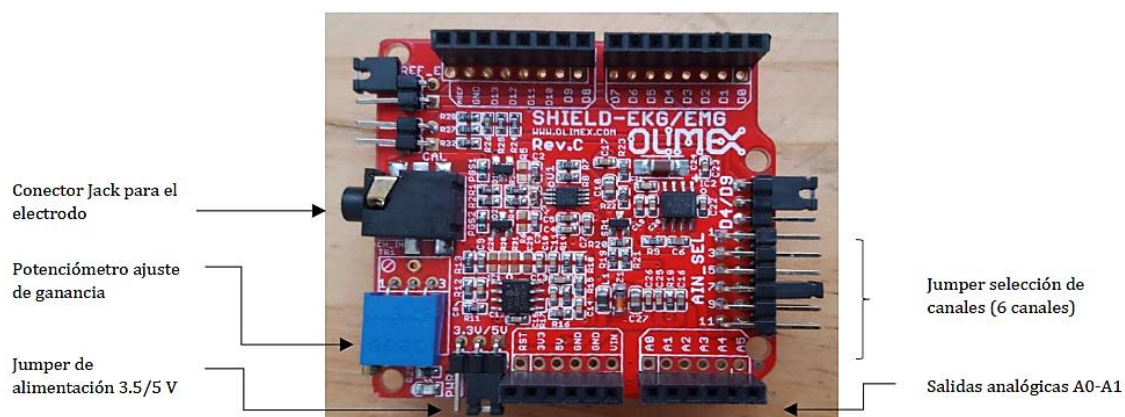


Figura 2.25 Tarjeta de adquisición de señales EKG-EMG Olimex [32].

Los cables de adaptación a los electrodos son de tres líneas de los cuales 2 conectores hacen 1 canal y 1 conector (DLR) es la retroalimentación del canal de adquisición, en el caso de que se trabaje con más de 1 canal de adquisición, no es necesario la utilización de los conectores (DLR) en los demás canales a diferencia del primero [32].

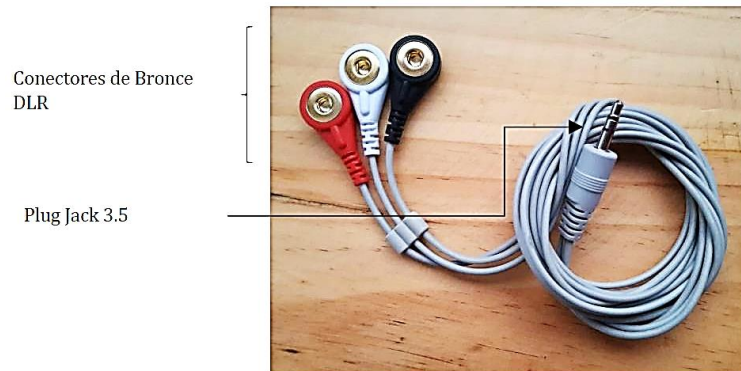


Figura 2.26 Cable conector DLR electrodo con salida Jack 3.5 [32].

2.4.1.3 Selección de Electrodo.

En la elección de los electrodos de superficie, se considera como alternativas 3 tipos de materiales, que pasan por un proceso de selección basado en el diseño conceptual.

- SOLUCIÓN A = Electrodo de plata/cloruro de plata
- SOLUCIÓN B = Electrodo de cobre/sulfato de cobre
- SOLUCIÓN C = Electrodo de Zinc

a. Evaluación del peso específico de cada criterio

Tabla 2.14

Evaluación del peso específico de cada criterio para la selección de los electrodos.

Potencial > Reducción de ruido > Precio					
CRITERIO	Potencial	Reducción de ruido	Precio	$\Sigma + 1$	Ponderación
Potencial		1	1	3	0,5
Reducción de ruido	0		1	2	0,33333
Precio	0	0		1	0,16667
			SUMA	6	1

b. Evaluación del peso específico del criterio POTENCIAL

Tabla 2.15

Evaluación del peso específico del criterio POTENCIAL.

SOLUCIÓN A > SOLUCIÓN B > SOLUCIÓN C					
POTENCIAL	SOLUCIÓN A	SOLUCIÓN B	SOLUCIÓN C	$\Sigma + 1$	Ponderación
SOLUCIÓN A		1	1	3	0,5
SOLUCIÓN B	0		1	2	0,333333333
SOLUCIÓN C	0	0		1	0,166666667
			SUMA	6	1

c. Evaluación del peso específico del criterio REDUCCIÓN DE RUIDO

Tabla 2.16

Evaluación del peso específico del criterio REDUCCIÓN DE RUIDO

SOLUCIÓN A > SOLUCIÓN B = SOLUCIÓN C					
REDUCCIÓN DE RUIDO	SOLUCIÓN A	SOLUCIÓN B	SOLUCIÓN C	$\Sigma + 1$	Ponderación
SOLUCIÓN A		1	1	3	0,5
SOLUCIÓN B	0		0,5	1,5	0,25
SOLUCIÓN C	0	0,5		1,5	0,25
			SUMA	6	1

d. Evaluación del peso específico del criterio PRECIO (bajo)

Tabla 2.17

Evaluación del peso específico del criterio PRECIO (bajo).

SOLUCIÓN B > SOLUCIÓN C > SOLUCIÓN A					
PRECIO	SOLUCIÓN A	SOLUCIÓN B	SOLUCIÓN C	$\Sigma + 1$	Ponderación
SOLUCIÓN A		0	0	1	0,166666667
SOLUCIÓN B	1		1	3	0,5
SOLUCIÓN C	1	0		2	0,333333333
			SUMA	6	1

e. Tabla de conclusiones

Tabla 2.18

Tabla de conclusiones para la selección de los electrodos.

CONCLUSIÓN	POTENCIAL	REDUCCIÓN DE RUIDO	PRECIO	Σ	PRIORIDAD
SOLUCIÓN A	0,5	0,5	0,1666667	1,1667	1
SOLUCIÓN B	0,3333333	0,25	0,5	1,0833	2
SOLUCIÓN C	0,1666667	0,25	0,3333333	0,75	3

Como resultado en la tabla de conclusiones; ver Tabla 2.18, se obtiene que la **SOLUCIÓN A** el electrodo de PLATA_CLORURO DE PLATA (Ag/AgCl) como se observa la Figura 2.27 que es un electrodo no polarizado y cumple con estándares adecuados, es óptima para el sistema.



Figura 2.27. Electrodo de Plata/Cloruro de plata (Ag/AgCl).

2.4.1.4 Selección de Software

En la elección de software, se considera como alternativas 2 programas de computadora para la adquisición y procesamiento de las señales, estas opciones pasan por un proceso de selección basado en el diseño conceptual.

- SOLUCIÓN A = LABVIEW

- SOLUCIÓN B = MATLAB

a. Evaluación del peso específico de cada criterio

Tabla 2.19

Evaluación del peso específico de cada criterio para la selección del software.

Utilidad > Compatibilidad > Rapidez					
CRITERIO	Utilidad	Compatibilidad	Rapidez	$\sum + 1$	Ponderación
Utilidad		1	1	3	0,5
Compatibilidad	0		1	2	0,33333
Rapidez	0	0		1	0,16667
			SUMA	6	1

b. Evaluación del peso específico del criterio UTILIDAD

Tabla 2.20

Evaluación del peso específico del criterio UTILIDAD.

SOLUCIÓN B > SOLUCIÓN A				
UTILIDAD	SOLUCIÓN A	SOLUCIÓN B	$\sum + 1$	Ponderación
SOLUCIÓN A		0	1	0,333333333
SOLUCIÓN B	1		2	0,666666667
		SUMA	3	1

c. Evaluación del peso específico del criterio COMPATIBILIDAD

Tabla 2.21

Evaluación del peso específico del criterio COMPATIBILIDAD.

SOLUCIÓN B > SOLUCIÓN A				
COMPATIBILIDAD	SOLUCIÓN A	SOLUCIÓN B	$\sum + 1$	Ponderación
SOLUCIÓN A		0	1	0,333333333
SOLUCIÓN B	1		2	0,666666667
		SUMA	3	1

d. Evaluación del peso específico del criterio RAPIDEZ

Tabla 2.22

Evaluación del peso específico del criterio RAPIDEZ.

SOLUCIÓN B = SOLUCIÓN A				
RAPIDEZ	SOLUCIÓN A	SOLUCIÓN B	$\sum + 1$	Ponderación
SOLUCIÓN A		0,5	1,5	0,5
SOLUCIÓN B	0,5		1,5	0,5
		SUMA	3	1

e. Tabla de conclusiones

Tabla 2.23

Tabla de conclusiones para la selección del software.

CONCLUSIÓN	UTILIDAD	COMPATIBILIDAD	RAPIDEZ	\sum	PRIORIDAD
SOLUCIÓN A	0,333333	0,333333	0,5	1,1666	2
SOLUCIÓN B	0,666667	0,666667	0,5	1,8334	1

Como resultado en la tabla de conclusiones; ver Tabla 2.23, se obtiene que la **SOLUCIÓN B** es óptima para este sistema, en este caso el software MATLAB es el que se usa en la investigación por todas las ventajas que presenta.

2.5 Conexión entre Discovery STM32F4 y Shield EKG-EMG.

Debido que para la adquisición de señales se utiliza un solo canal, la conexión entre las tarjetas de procesamiento y adquisición se realiza como se observa en la Tabla 2.24.

Tabla 2.24

Conexión de salidas y entradas entre la tarjeta de procesamiento y adquisición.

CARACTERÍSTICA	SALIDA	ENTRADA
TIPO DE TARJETA	Discovery STM32F4	Shield EKG-EMG
ALIMENTACIÓN	5V	5V
ALIMENTACIÓN	GND	GND
SEÑAL ANALÓGICA	PA0	A0

En la tarjeta de adquisición Shield EKG-EMG hay que tener en cuenta la posición del jumper referente al número del canal o salida analógica; observe la Figura 2.28, de esta forma se realiza las conexiones respectivas.

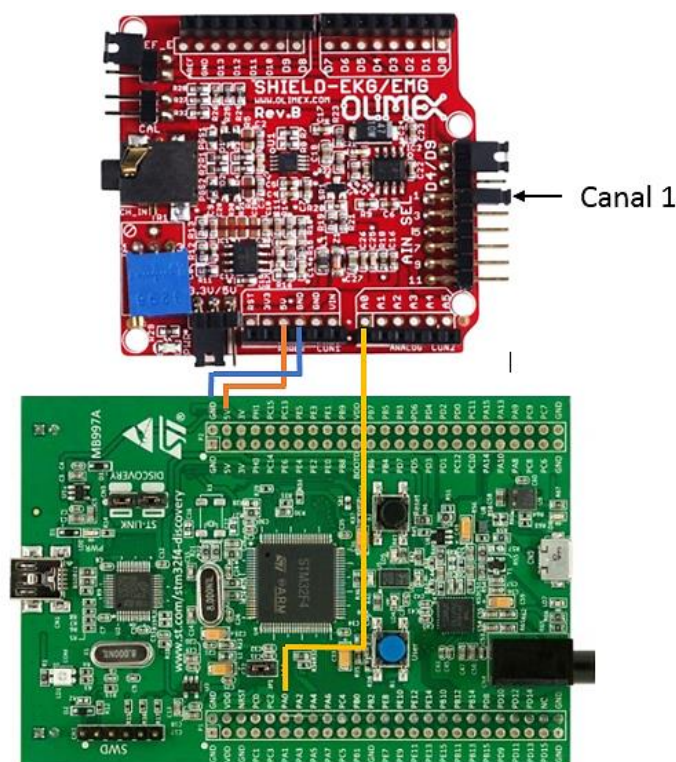


Figura 2.28 Conexión entre Discovery STM32F4 y Shield EKG-EMG.

2.5.1 Configuración y comunicación de la tarjeta Discovery STM32F4

Para la adquisición de señales EMG se utiliza bloques de programación en Simulink, permitiendo la configuración y comunicación host entre Matlab y el usuario, para esto es necesario la instalación de librerías, bloques y herramientas que permiten dicha acción [32]. entre los cuales están:

- Waijung Blockset de STM32F4 Target Blockset.
- Addons v2.6 Beyond Control.
- Compilador GNU ARM para compilar el código fuente generado.
- GNU Make utilities para controlar el proceso de compilación.
- Bibliotecas de software base para STM32F4 (USB COM port drivers).

Para cargar el archivo de configuración se debe conectar la tarjeta STM32F4 solo con el cable mini USB para programación y depuración de este, una vez que cargados los bloques de configuración se procede a realizar la comunicación host con la ejecución del archivo (EMG-read_1CH_HOST) en donde se puede observar las gráficas de las señales durante el proceso de adquisición [32]. Ver Figura 2,29 y Figura 2,30.

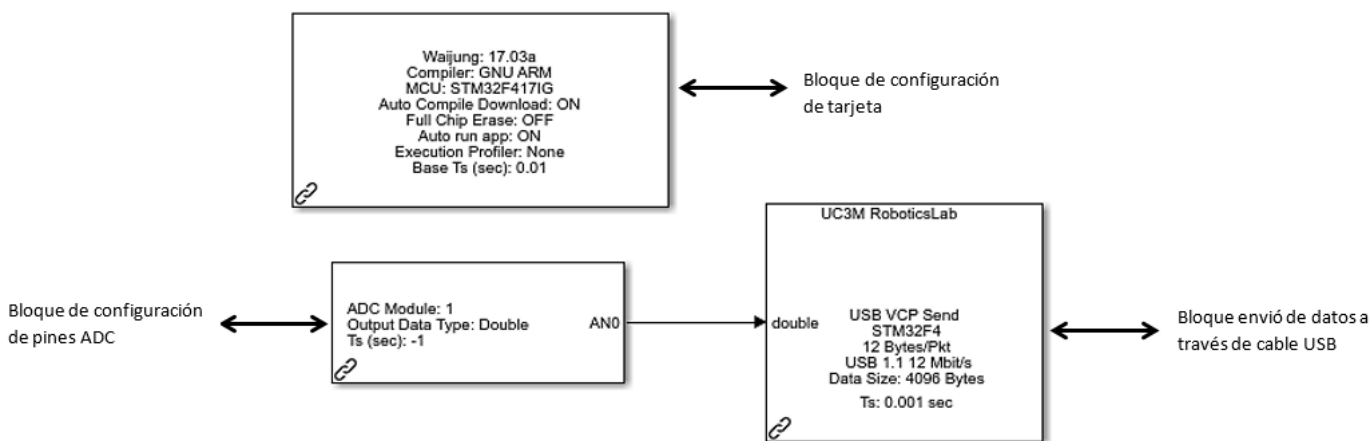


Figura 2.29 Bloque de configuración en Simulink de tarjeta Discovery STM32F4.

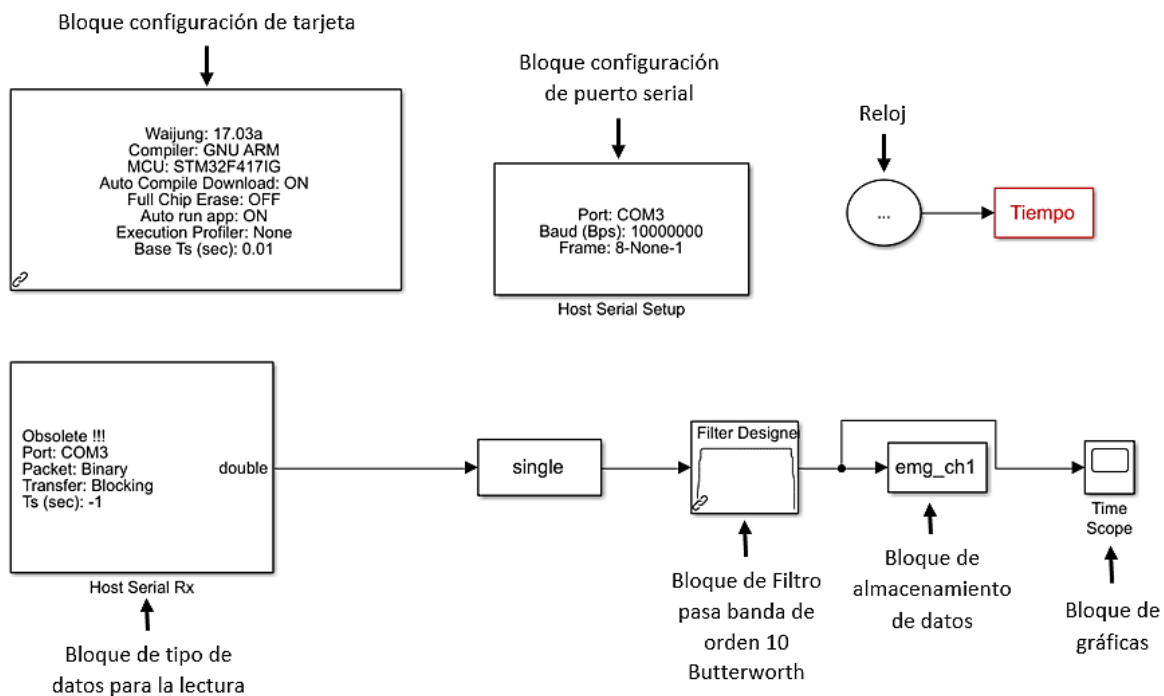


Figura 2.30 Bloque de configuración en simulink para la adquisición de datos EMG.

Cada bloque de adquisición se almacena en una matriz en función al tipo de movimiento realizado, los cuales se ubicarán en el espacio de trabajo de Matlab [32].

2.6 Adquisición y procesamiento de la señal EMG

Se debe tomar en cuenta ciertos aspectos al momento de adquirir las señales EMG [32]. Como:

- El punto de partida desde que se adquiere la señales EMG.
- Tipo de movimientos a ejecutar durante la adquisición.
- El tiempo establecido para la ejecución y reposo de una acción (movimiento de estudio).
- El número de acciones en una muestra de adquisición
- La ubicación de los electrodos supervisado y aprobado por un especialista.
- La ubicación correcta de los instrumentos médicos de medición.

Antes de realizar la adquisición de las señales EMG, es muy importante que la ubicación de los electrodos la realice un especialista, debido a su experiencia; también, la instrumentación médica en este caso el goniómetro debe estar debidamente colocado, ver Figura 2.31; Además, se puede hacer el uso de escuadras geométricas para ejecutar correctamente el movimiento.

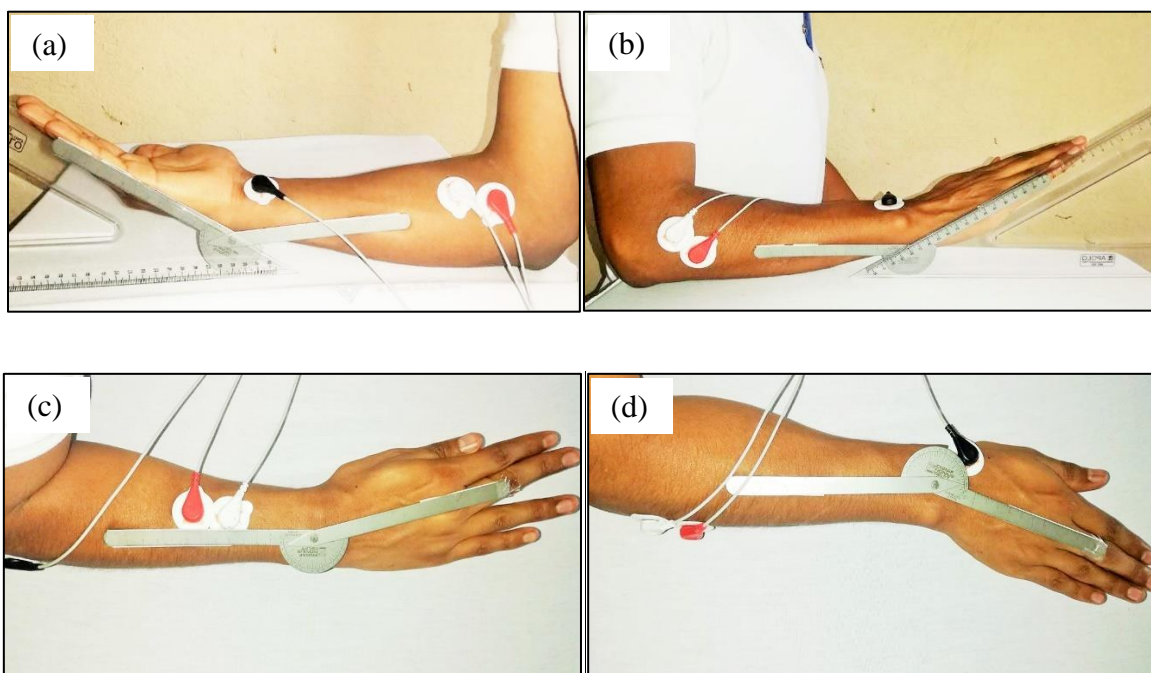


Figura 2.31 Ubicación de goniómetro para movimiento de: (a) Flexión. (b) Extensión. (c) Abducción. (d) Aducción.

Una vez que se ha ubicado correctamente los electrodos y la instrumentación médica se realiza la adquisición de la señal EMG en estado puro, en este caso se realiza el movimiento de abducción para el cual se obtendrá una base de datos del movimiento en la posición de 5° , 10° , 15° y 20° . Por cada posición se realiza 3 repeticiones del movimiento cada 5 segundos, ver Figura 2.32. Terminada la primera iteración se repite el mismo proceso por 34 veces, de igual forma en cada posición. Este proceso es igual para los movimientos de Flexión, Extensión y Aducción.

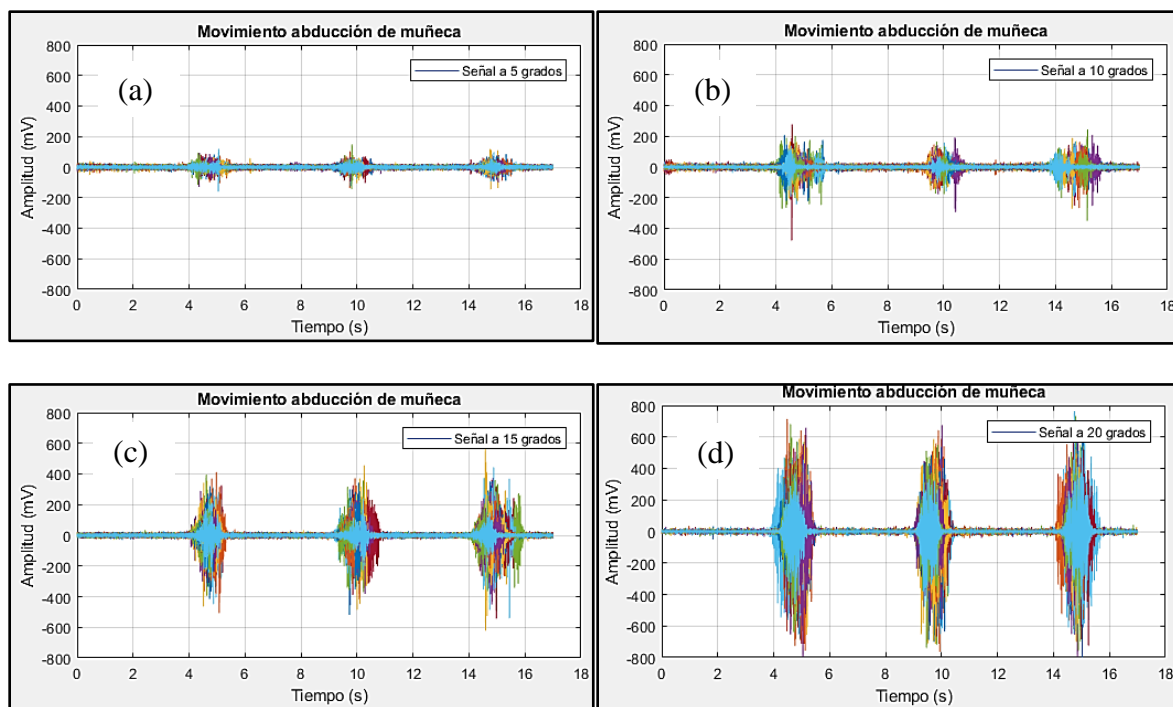


Figura 2.32 Señales en estado puro del movimiento Abducción en las 4 posiciones.

2.6.1 Filtrado digital de la señal EMG

El filtrado digital permite el acondicionamiento de la señal, eliminando información fuera de los parámetros de estudio, los cuales alteran en cierto grado la amplitud, frecuencia o fase de esta, para esto la aplicación de filtros digitales, permite la supresión de ruido generado por diferentes factores, como las líneas de potencia, ondas de radio, el contacto entre el electrodo y la piel, así como el dispositivo mismo de medición mioeléctrica [32].

Cabe recalcar que la tarjeta de adquisición Olimex EKG – EMG, dispone de un filtro de tipo Bessel, que ayuda a que las señales no se distorsionen durante el proceso de adquisición, limitando en cierto grado la afectación debido a ruidos en el entorno [32].

Es evidente que, tras la aplicación de filtros a diferentes rangos de frecuencia, se suprime información innecesaria, debido a la afectación de ruido inserto durante el proceso de adquisición, tal es el caso del filtro Notch que no tuvo gran aplicación a la eliminación de ruido en las señales debido al filtro instrumental que posee la tarjeta de adquisición, no obstante, la aplicación de filtros a rangos como 500Hz y 25Hz ayudaron a la obtención de una señal más pura [32]. ver Figura 2.33.

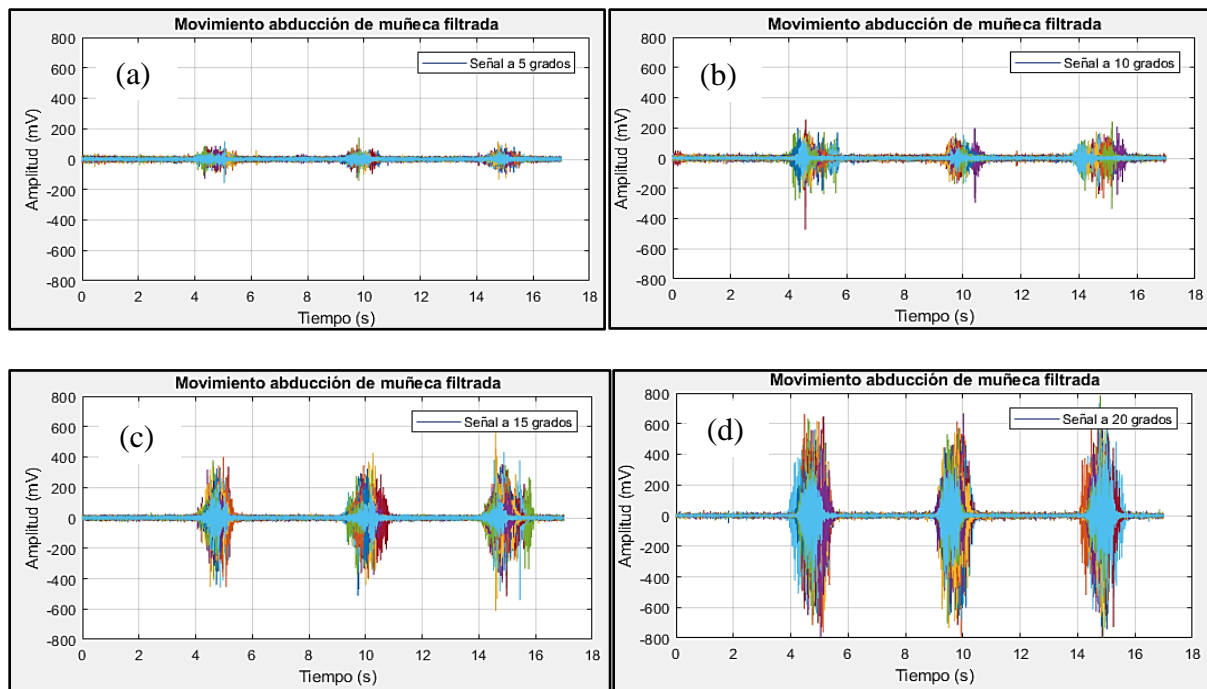


Figura 2.33 Señales filtradas del movimiento Abducción.

Esta etapa de filtrado se realiza también para los movimientos de Flexión, Extensión y Aducción.

2.6.2 Rectificado de la señal EMG

Tras el proceso de filtrado de la señal dentro de los parámetros de estudio se encuentra la parte de rectificado lo cual permite la limitación de datos de actividad igualitarios, discriminando valores negativos de la misma [32]. ver Figura 2.34.

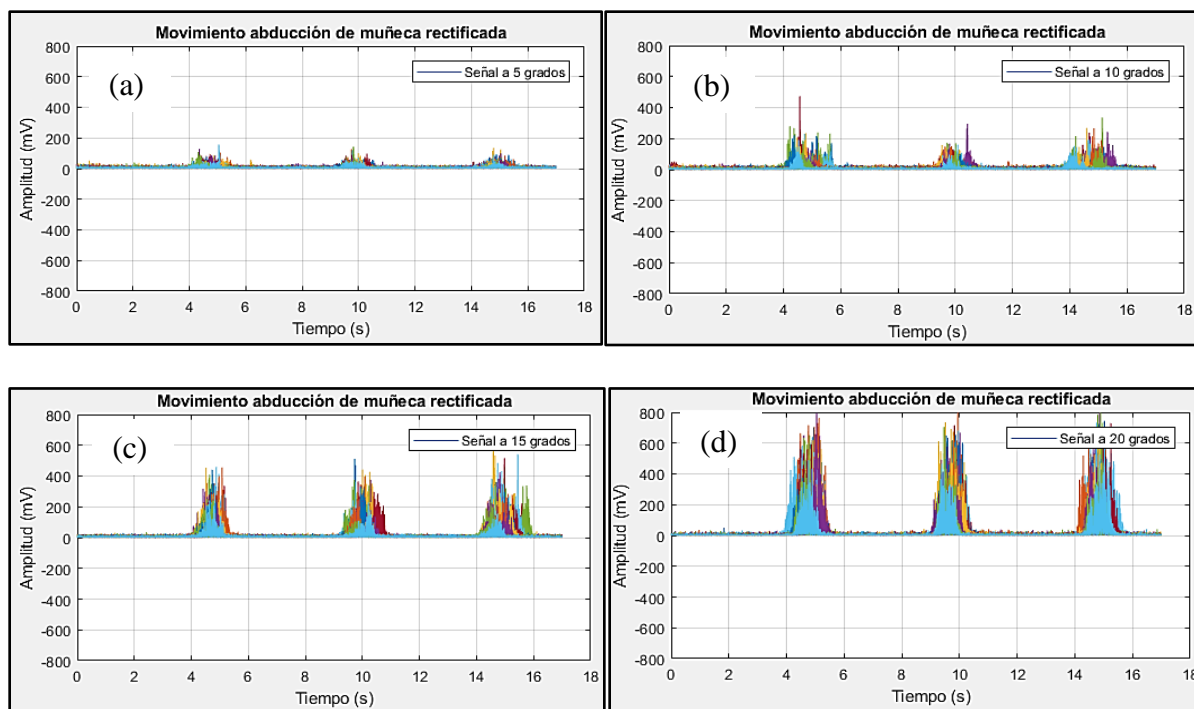


Figura 2.34 Señales rectificadas del movimiento Abducción.

Una vez terminado la etapa de rectificado de señal en el movimiento de Abducción, se repite para los movimientos de Flexión, Extensión y Aducción.

2.6.3 Detección de actividad de la señal EMG

Las señales EMG adquiridas, se someten al proceso de reconocimiento de actividad, ejecutándose en las interacciones o acciones de contracción muscular realizada tras el proceso de adquisición, mismos que tras el suavizado de las señales y un valor de umbral inferior determinado, se puede ubicar los campos de acción [32]. ver Figura 2.35.

Durante el tiempo de ejecución la mayor parte del tiempo la señal adquirida pasa en reposo, una vez que se realiza el movimiento en este caso cada 5 segundos nos indica una señal con amplitud, es ahí cuando el algoritmo de detección se activa delimitando la zona donde se encontró movimiento. Además, se estiman diferentes niveles de umbralización para determinar los coeficientes que se consideran no significativos [33].

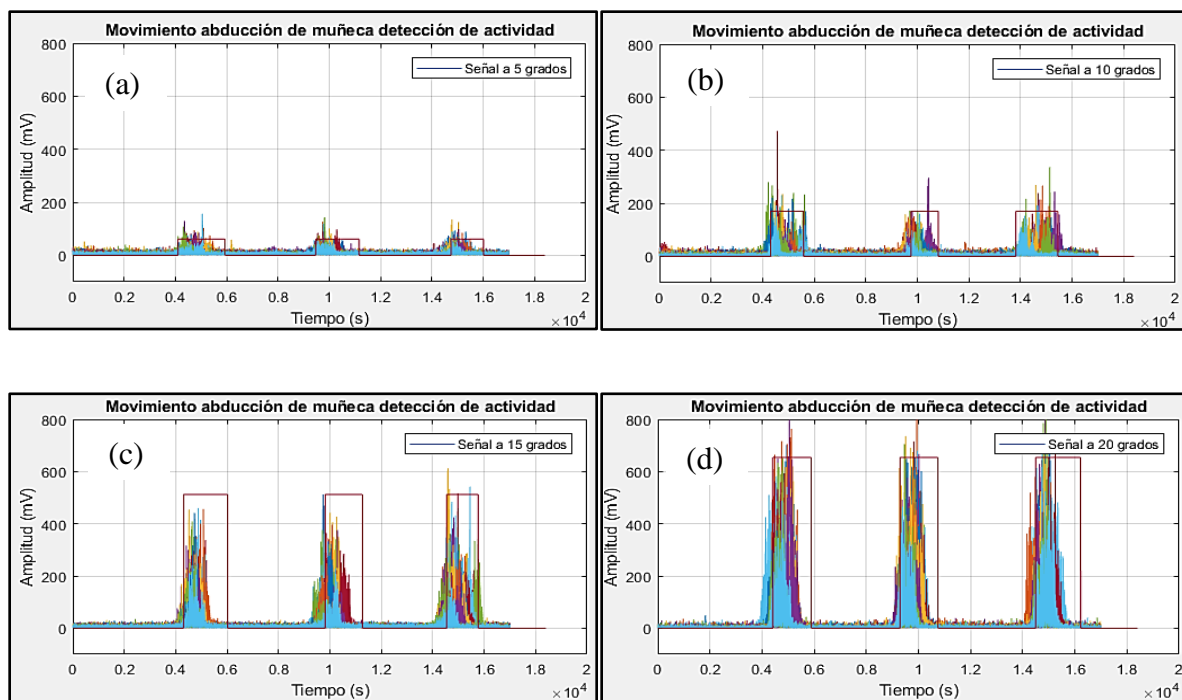


Figura 2.35 Señales procesadas con detección de actividad del movimiento Abducción.

2.7 Caracterización de señales EMG

Una vez finalizado la adquisición y procesamiento de las señales EMG, los resultados obtenidos pasan a este procedimiento en el que se utiliza 11 métodos de caracterización en el dominio del tiempo, de la frecuencia y tiempo-frecuencia. Dichos métodos son:

- Integral de la señal EMG (IEMG)

La IEMG se usa generalmente como un índice de pre-activación para la actividad muscular. Es el área bajo la curva de la señal EMG rectificadas. IEMG se puede simplificar y expresar como la suma de los valores absolutos de la amplitud de EMG [17].

$$IEMG = \sum_{i=1}^N |x_i| \quad (4)$$

Donde N es la longitud del segmento, i es el incremento del segmento y xi es el valor de la amplitud de la señal.

- **Simple Cuadrado Integral (SSI)**

El Simple Cuadrado Integral (SSI) expresa la energía de la señal EMG como una característica utilizable [17].

$$SSI = \sum_{i=1}^N |x_i|^2 \quad (5)$$

- **Valor absoluto medio (MAV)**

El Valor Absoluto Medio (MAV) es un método para detectar y medir los niveles de contracción muscular. Se expresa como el promedio móvil de la señal EMG rectificadas de onda completa [17].

$$MAV = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (6)$$

- **Raíz Media Cuadrática (RMS)**

La raíz media cuadrática (RMS) se modela como el proceso aleatorio gaussiano de amplitud modulada en el que el RMS se relaciona con la fuerza constante y las contracciones no fatigantes de los músculos [17].

El método RMS de extracción de características es muy comúnmente utilizado. Como es computacionalmente eficiente y rápido, aún contiene datos precipitados [17].

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N |x_i|^2} \quad (7)$$

- **Varianza de la señal EMG (VAR)**

La variación de EMG (VAR) expresa la potencia de la señal de EMG como una característica utilizable [17].

$$VAR = \frac{1}{N-1} \sum_{i=1}^N x_i^2 \quad (8)$$

- **Longitud de Onda (WL)**

La Longitud de la forma de onda (WL) es intuitivamente la longitud acumulada de la forma de onda sobre el segmento. Los valores resultantes del cálculo de WL indican una medida de la amplitud, frecuencia y duración de la forma de onda [17].

$$WL = \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (9)$$

- **Cambios en el Signo de la Señal (SSC)**

Este es otro parámetro que representa la información en frecuencia de la señal mioeléctrica, su cálculo cuantifica el número de veces que la pendiente de la señal SME cambia de signo, el número de cambios entre las pendientes positivas y negativas, entre tres segmentos secuenciales es realizado con un nivel umbral para evitar el ruido inducido [14]. Su expresión matemática viene definida por:

$$SSC = \sum_{i=2}^{N-1} [f[(x_i - x_{i-1}) * (x_i - x_{i+1})]] \quad (10)$$

$$f(x) = \begin{cases} 1, & \text{si } x \geq \text{nivel umbral} \\ 0, & \text{en caso contrario} \end{cases}$$

El valor de umbral disponible es normalmente escogido entre 50 μV y 100 mV, dependiendo de la configuración del valor de ganancia del sistema de instrumentación y del nivel de ruido inducido [14].

- **Amplitud de Willison (WAMP)**

La amplitud de Willison (WAMP) es otra medida de la información en frecuencia de la señal mioeléctrica, este parámetro calcula el número de veces en que la diferencia resultante en amplitud de la señal mioeléctrica, entre dos segmentos consecutivos, exceden un umbral predefinido, el parámetro es un indicador de la activación de los potenciales de acción de las unidades motoras, y por consiguiente de los niveles de contracción muscular [14]. El cálculo es formulado por:

$$WAMP = \sum_{i=1}^{N-1} [f(|x_n - x_{n+1}|)] \quad (11)$$

$$f(x) = \begin{cases} 1, & \text{si } x \geq \text{nivel umbral} \\ 0, & \text{en caso contrario} \end{cases}$$

- **Energía de la señal (E)**

Los valores de la energía de la señal se encuentran entre 50Hz y 60Hz [34]. Para hallar estos datos, se utiliza la siguiente ecuación:

$$E = \max_i \left(\frac{\sum X^2}{N} \right) \quad (12)$$

Donde N es la longitud del segmento, i es el incremento.

- **Transformada de Fourier (STFT)**

La transformada de Fourier divide la señal de entrada en segmentos, al hacer esto, la señal en cada ventana es estacionaria [35]. La STFT para una señal dada $x(t)$ se puede expresar como:

$$STFT_x(t, w) = \int W^*(\tau - t)x(\tau)e^{-jw\tau}d\tau \quad (13)$$

Donde $W(t)$ es la función, $(*)$ es el conjugado complejo, τ representa el tiempo y w representa la frecuencia [35].

- **Densidad de espectro de potencia (PSD)**

Muestra en qué punto las variaciones de frecuencia son fuertes y débiles [35]. La densidad del espectro de potencia (PSD) se calcula por:

$$PSD = \left| \sum_{i=0}^{N-1} x_i e^{\frac{-j2\pi ki}{N}} \right|^2 \quad (14)$$

Donde $k = 0, 1, \dots, N - 1$, N es la longitud de los datos del EEG, y x_i representa las muestras discretas de los datos del EEG [35].

Los valores obtenidos tras el proceso de caracterización, nos indican los niveles de contracción muscular, así como, la potencia y energía promedio durante la ejecución del movimiento de estudio [32].

En el estudio se analiza los 4 movimientos es por esto que se muestran los datos de la caracterización para cada uno; dichos valores son resultado de las 11 características propuestas, de las cuales la característica Simple Cuadrado Integral, Raíz Media Cuadrática y Longitud de Onda son las más principales y las que posteriormente se usaran para el entrenamiento de clasificación.

Una vez obtenida la información del proceso de caracterización se expresa los resultados del primer movimiento en la Tabla 2.25.

Tabla 2.25
Características en el movimiento Abducción.

MOVIMIENTO ABDUCCIÓN	Características	Integral de la Señal	Simple Cuadrado Integral	Valor Medio Absoluto	Raíz Media Cuadrática	Varianza	Longitud de Onda	Transformada de Fourier	Energía	Densidad de espectro de potencia	Cambios en el signo de la señal	Amplitud de Willison	Porcentaje de incremento (%)
5 GRADOS	max	28,50	2,19	28,50	14,81	1,36	42,39	20,67	2,19	2,10	5,29	11,91	
	min	21,38	1,15	21,37	10,74	0,61	35,12	15,51	1,15	1,12	1,06	4,75	47,2
	medio	25,47	1,63	25,47	12,72	0,95	38,98	18,48	1,62	1,59	2,55	7,91	
10 GRADOS	max	37,78	6,29	37,78	25,08	5,19	56,04	27,40	6,29	6,45	32,80	40,67	
	min	27,94	2,51	27,94	15,86	1,51	42,62	20,27	2,51	2,38	7,14	11,95	59
	medio	32,69	3,89	32,69	19,57	2,92	48,68	23,71	3,89	3,86	16,83	26,46	
15 GRADOS	max	66,86	40,71	66,85	63,82	39,70	75,07	53,59	40,71	38,64	55,56	70,46	
	min	36,38	6,29	36,38	25,08	5,27	51,29	26,39	6,29	5,83	22,75	35,82	70
	medio	46,46	15,51	46,47	38,68	14,47	57,78	34,11	15,51	15,31	32,99	48,62	
20 GRADOS	max	100	100	100	100	100	100	100	100	100	100	100	
	min	69,83	48,05	69,81	69,32	46,79	68,72	65,61	48,05	50,92	53,17	63,89	100
	medio	82,79	69,61	82,79	83,10	68,88	82,57	79,94	69,61	73,22	77,35	82,17	

Los resultados mostrados en la Tabla 2.25 son datos expresados en un rango de 0 a 100, con la finalidad de apreciar fácilmente la diferencia que existe entre los ángulos de estudio; en la última columna se observa un crecimiento porcentual de las características, con lo cual se puede decir que, entre el movimiento de 5° y 10° hay un incremento del 11.8%, entre el de 10° y 15° un 11% y finalmente entre el de 15° y 20° el 30% de incremento en los datos. Con esto se logra evidenciar que, al aumentar el ángulo de la posición, también aumenta el valor de todos los datos resultantes de las características. En cuanto a las 3 características principales, también es evidente que el valor medio de cada posición analizada aumenta, esto confirma que existe una diferencia, la cual permite identificar si el movimiento realizado es de 5, 10, 15 o 20 grados.

A continuación, se realiza la caracterización para el movimiento de Aducción, estos resultados se muestran en la Tabla 2.26.

Tabla 2.26
Características en el movimiento Aducción.

MOVIMIENTO ADUCCIÓN	Características	Integral de la Señal	Simple Cuadrado Integral	Valor Medio Absoluto	Raíz Media Cuadrática	Varianza	Longitud de Onda	Transformada de Fourier	Energía	Densidad de espectro de potencia	Cambios en el signo de la señal	Amplitud de Willison	Porcentaje de incremento (%)
10 GRADOS	max	68,43	36,03	68,45	60,02	30,03	82,70	68,45	36,03	35,48	81,15	71,96	
	min	36,07	11,16	36,08	33,41	9,51	46,17	36,08	11,16	10,61	10,38	11,49	67,82
	medio	54,03	23,52	54,04	48,10	19,11	60,20	54,04	23,52	23,21	26,77	27,27	
20 GRADOS	max	82,65	58,61	82,67	76,57	54,31	87,52	82,67	58,61	61,55	75,00	80,49	
	min	58,75	29,58	58,74	54,40	23,81	60,12	58,74	29,58	29,33	16,54	25,93	85,36
	medio	70,89	43,02	70,90	65,30	37,55	75,76	70,90	43,02	42,92	46,23	55,53	
30 GRADOS	max	100	100	100	100	100	100	100	100	100	100	100	
	min	77,43	53,36	77,45	73,05	45,69	75,50	77,45	53,36	51,60	41,54	53,76	100
	medio	87,00	74,96	87,03	86,32	74,20	88,76	87,03	74,96	75,64	72,15	78,13	

Al analizar la tabla de resultados de este movimiento, se comprueba al igual que el movimiento anterior que, al aumentar el ángulo de la posición, aumenta el valor de todos los datos resultantes de las características, esto se evidencia a través de la última columna de la tabla en la que indica el incremento de los valores característicos en porcentaje, obteniendo un crecimiento del 17.54% entre el movimiento de 10° y 20° y el 14.64% entre el movimiento de 20° y 30° esto quiere decir que se puede identificar claramente la posición que se esté realizando.

En cuanto a las 3 características principales, también es evidente que el valor medio de cada posición analizada aumenta, esto confirma que existe una diferencia, la cual permite identificar si el movimiento realizado es de 10, 20 o 30 grados.

Una vez finalizada la caracterización en el movimiento de Aducción, se realiza para el movimiento de Flexión, estos resultados se muestran en la Tabla 2.27.

Tabla 2.27
Características en el movimiento Flexión.

MOVIMIENTO FLEXIÓN	Características	Integral de la Señal	Simple Cuadrado Integral	Valor Medio Absoluto	Raíz Media Cuadrática	Varianza	Longitud de Onda	Transformada de Fourier	Energía	Densidad de espectro de potencia	Cambios en el signo de la señal	Amplitud de Willison	Porcentaje de incremento (%)
30 GRADOS	max	24,82	2,30	24,83	15,18	1,51	41,81	19,98	23,04	21,17	35,09	6,86	
	min	20,43	1,30	20,44	11,40	0,67	36,67	16,45	12,99	11,44	6,14	1,30	25,07
	medio	22,60	1,73	22,61	13,11	0,97	39,24	18,19	17,29	15,92	19,71	3,74	
45 GRADOS	max	29,19	3,71	29,20	19,26	3,10	53,22	23,49	37,08	41,96	62,50	12,37	
	min	19,15	1,34	19,15	11,57	0,76	31,90	15,41	13,39	12,61	1,75	0,26	28,22
	medio	23,10	1,95	23,11	13,86	1,17	38,63	18,59	19,47	18,25	12,43	2,55	
60 GRADOS	max	39,78	8,79	39,79	29,65	7,29	53,78	32,01	87,87	82,62	45,39	26,20	
	min	26,17	2,55	26,18	15,97	1,58	42,83	21,06	25,51	24,97	13,38	3,59	65,47
	medio	31,24	4,52	31,26	20,96	3,25	46,87	25,15	45,16	42,82	22,15	10,86	
80 GRADOS	max	100	100	100	100	100	100	100	100	100	100	100	
	min	64,13	35,88	64,11	59,90	33,00	72,14	60,05	35,88	37,36	66,23	61,64	100
	medio	80,67	68,96	80,66	82,37	66,46	84,83	77,26	68,97	70,42	79,17	80,72	

Al finalizar la caracterización del movimiento de flexión, se comprueba en la tabla de resultados que, al aumentar el ángulo de la posición, aumenta el valor de las características correspondientes a cada ángulo; en la última columna de la tabla se da a conocer el incremento porcentual de los datos característicos, obteniendo un incremento del 3.15% entre el movimiento de 30° y 45°, un 37.25% entre el movimiento de 45° y 60° y un 34.53% entre el movimiento de 60° y 80°, con este análisis se identifica claramente la posición que se esté realizando.

Por consiguiente, en las 3 características principales, se evidencia que el valor medio de cada posición analizada aumenta, esto confirma que existe una diferencia, la cual permite identificar si el movimiento realizado es de 30, 45, 60 u 80 grados.

Por último, se realiza la caracterización para el movimiento de Extensión, estos resultados se muestran en la Tabla 2,28.

Tabla 2.28
Características en el movimiento Extensión.

MOVIMIENTO EXTENSIÓN	Características	Integral de la Señal	Simple Cuadrado Integral	Valor Medio Absoluto	Raíz Media Cuadrática	Varianza	Longitud de Onda	Transformada de Fourier	Energía	Densidad de espectro de potencia	Cambios en el signo de la señal	Amplitud de Willison	Porcentaje de incremento (%)
30 GRADOS	max	67,54	31,25	67,55	55,90	23,58	60,55	55,90	31,25	27,46	35,72	63,09	
	min	40,81	18,70	40,81	43,23	17,51	43,61	43,23	18,70	17,21	18,79	35,34	56,57
	medio	52,27	23,01	52,27	47,87	20,25	50,55	47,87	23,01	21,69	25,54	45,15	
45 GRADOS	max	65,03	37,33	82,78	75,43	55,75	77,55	75,43	56,92	56,09	62,70	80,96	
	min	45,56	21,44	48,49	49,49	24,75	52,52	49,49	24,50	23,73	35,29	50,15	67,69
	medio	52,49	28,07	56,59	57,77	34,11	60,49	57,77	33,78	31,97	47,07	59,76	
60 GRADOS	max	86,26	96,28	86,26	98,12	97,90	89,53	98,12	96,28	95,51	96,41	73,77	
	min	54,37	59,69	54,35	77,26	75,13	66,76	77,26	59,69	59,87	63,13	47,00	88,65
	medio	70,02	77,61	70,03	87,91	91,75	79,22	87,91	77,61	77,46	77,98	61,49	
70 GRADOS	max	100	100	100	100	100	100	100	100	100	100	100	
	min	72,44	52,56	72,43	72,46	52,08	71,63	72,46	52,56	49,65	55,67	69,01	100
	medio	89,67	81,77	89,67	90,20	81,46	89,36	90,20	81,77	79,47	81,69	87,77	

En la tabla de resultados del movimiento de extensión, se muestran los datos obtenidos en un rango de 0 a 100 como en los movimientos anteriores; en la última columna se indica el crecimiento porcentual que existe entre los datos correspondientes a cada movimiento, obteniendo un incremento del 11.12% entre el movimiento de 30° y 45°, un 20.96% entre el movimiento de 45° y 60° y por último el 11.35% entre el movimiento de 60° y 70°; con este análisis se puede decir que, al aumentar el ángulo de la posición, aumenta el valor de las características correspondientes a cada ángulo. En consecuencia, en las 3 características principales, se demuestra que el valor medio de cada posición analizada aumenta, esto confirma que existe una diferencia, la cual permite identificar si el movimiento realizado es de 30, 45, 60 o 70 grados.

2.8 Clasificación de señales EMG

Antes de iniciar el proceso de clasificación, se representa un esquema gráfico, para observar con claridad el comportamiento de las características en cada movimiento [32].

La gráfica de características se representa en función de los métodos de caracterización, en este caso, se muestran la raíz media cuadrática (RMS) y simple cuadrado integral (SSI); ver Figura 2.36, esta representación en 2 dimensiones permite observar si los datos obtenidos son linealmente separables para proponer un método de clasificación adecuado.

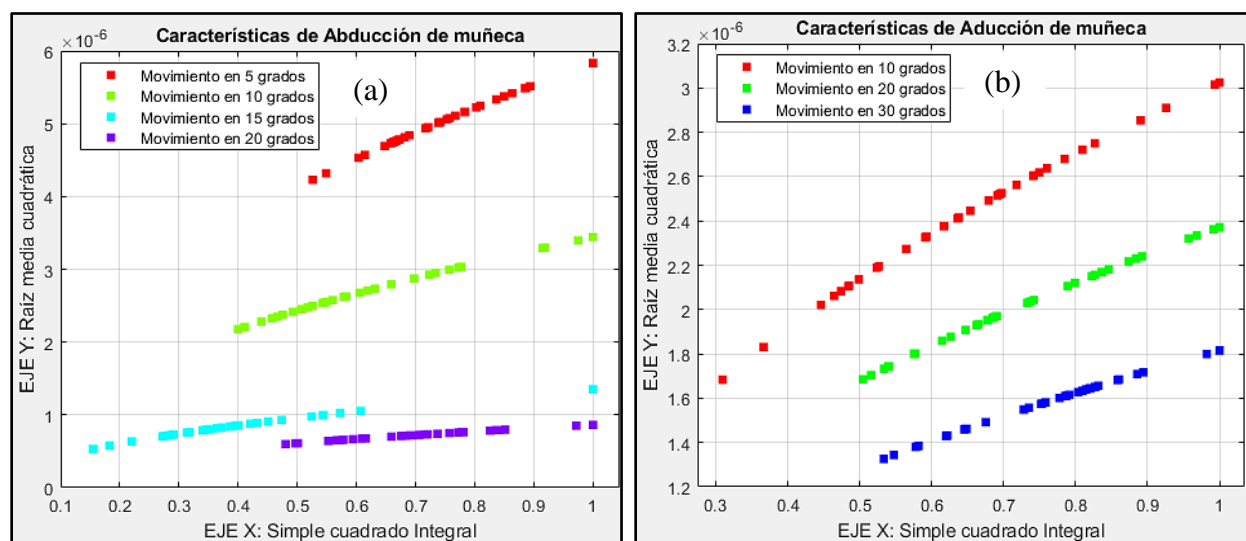


Figura 2.36 Características en 2 dimensiones de movimientos: (a) Abducción (b) Aducción.

Las figuras (a) y (b) corresponden a los movimientos de Abducción y Aducción; en dichas gráficas se observa las características de cada posición angular, es evidente que existe una gran diferencia entre ellas, ya que se puede diferenciar entre las características de una posición con otra. Con esto, se concluye que las características son linealmente separables.

De la misma forma, en la Figura 2.37 se observa las características correspondientes a las posiciones angulares de los movimientos Flexión y Extensión, en cada una de las gráficas se evidencia claramente que las características son linealmente separables.

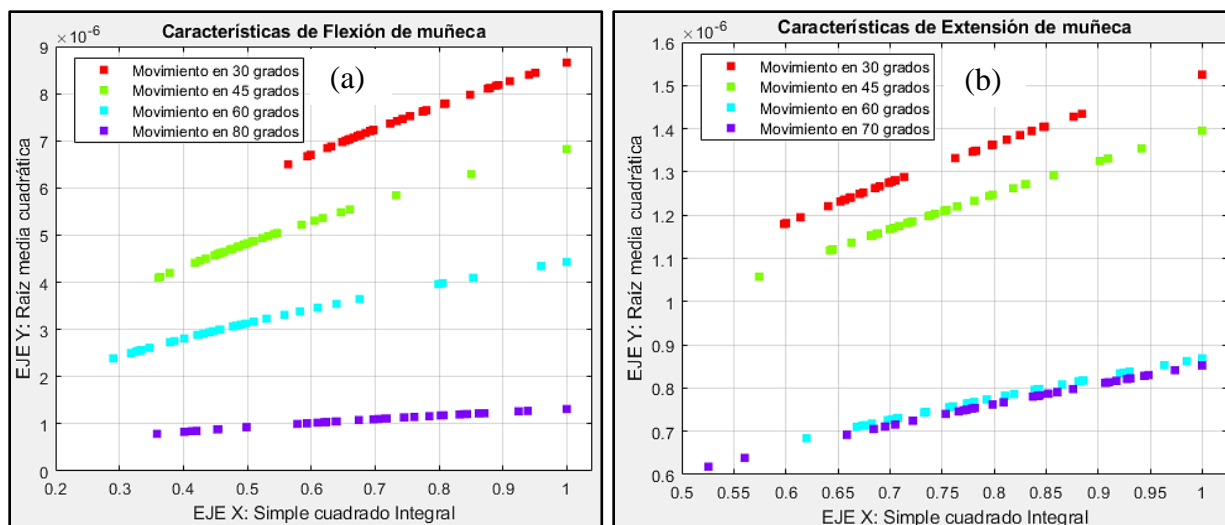


Figura 2.37 Características en 2 dimensiones de movimientos: (a) Flexión (b) Extensión.

Para tener una mejor visualización de los datos característicos se muestra una representación gráfica en 3 dimensiones de los datos más principales en este caso, raíz media cuadrática (RMS), simple cuadrado integral (SSI) y Longitud de forma de onda (WL); ver Figura 2.38 y Figura 2.39.

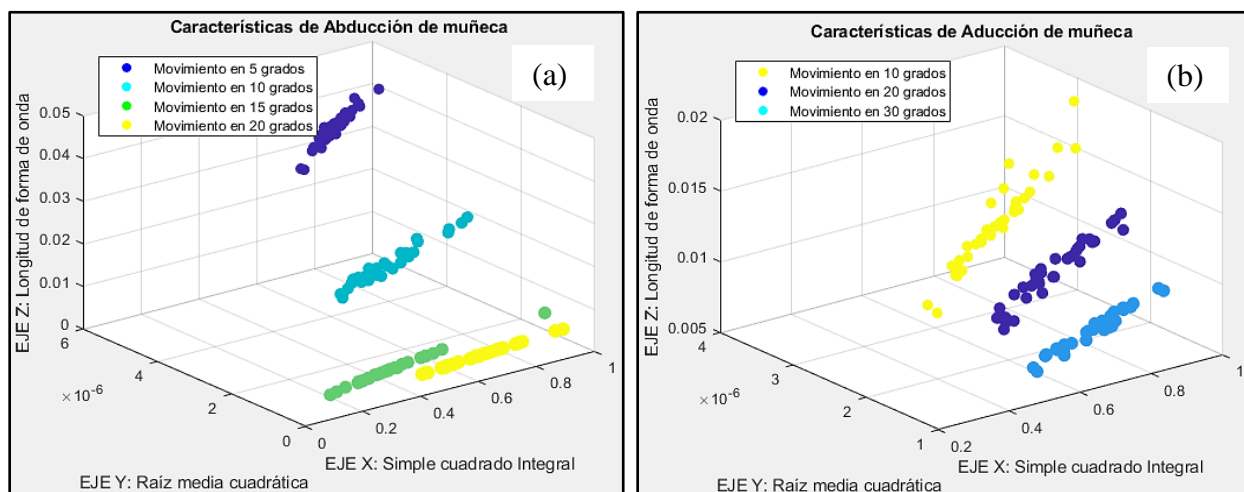


Figura 2.38 Características en 3 dimensiones de movimientos: (a) Abducción (b) Aducción.

En la Figura 2.38 se observa los datos correspondientes a los movimientos de Abducción y Aducción, para las posiciones de 5° , 10° , 15° , 20° y para las posiciones de 10° , 20° , 30° , respectivamente; En esta gráfica es más fácil demostrar que los datos son linealmente separables.

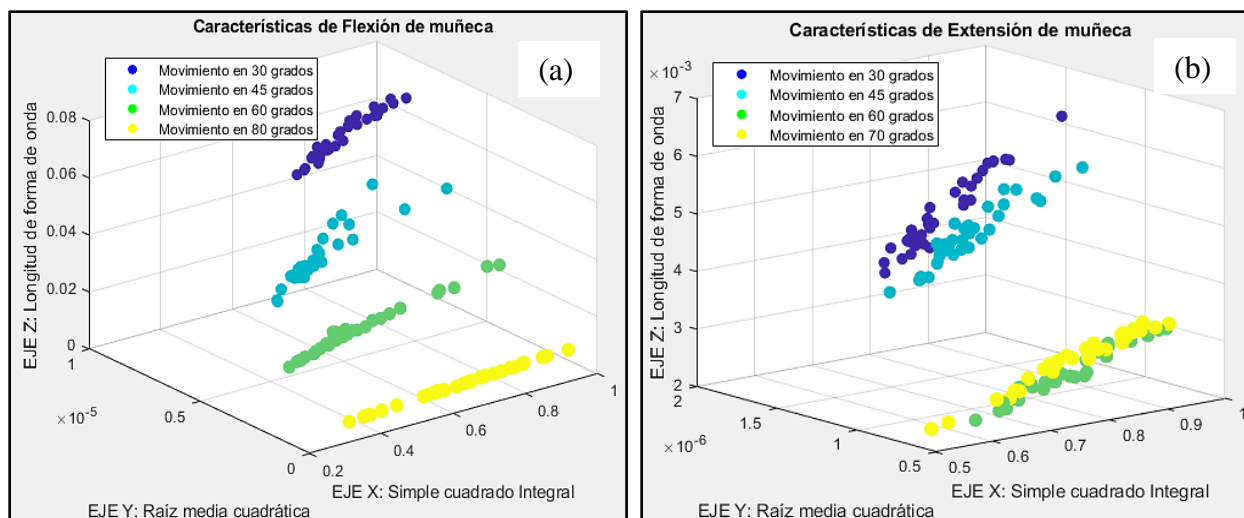


Figura 2.39 Características en 3 dimensiones de movimientos: (a) Flexión (b) Extensión.

Los datos mostrados en la Figura 2.39, son particularmente de los movimientos Flexión y Extensión, para las posiciones de 30°, 45°, 60°, 80° y para las posiciones de 30°, 45°, 60°, 70°, respectivamente; como se observa los datos no tienen ninguna relación entre ellos, por lo que se puede identificar claramente 4 grupos, debido a estos resultados se plantea un método de clasificación adecuado, el cual permita realizar una separación eficiente de los grupos y posteriormente los identifique correctamente.

Una vez que se ha evidenciado que los datos característicos son linealmente separables, se propone diferentes métodos de clasificación que satisfagan una identificación acertada del movimiento que se ejecuta, los cuales son:

- Flexión a 30°, 45°, 60° y 80°.
- Extensión a 30°, 45°, 60° y 70°.
- Abducción a 5°, 10°, 15° y 20°
- Aducción a 10°, 20° y 30°.

Los métodos de clasificación que se toman a consideración son tanto supervisados como no supervisados, esto permitirá tener una mejor idea de los algoritmos de clasificación, de tal forma que se opte por el algoritmo más eficiente. Observar Tabla 2,29.

Tabla 2.29
Algoritmos de clasificación.

SUPERVISADO	NO SUPERVISADO
Backpropagation	Clustering SOM
Nearest Neighbor	
Análisis discriminantes	
Máquinas de soporte vectorial	

Cada clasificador mostrado en la tabla anterior, tiene un uso particular de la información para el entrenamiento, no todos estos algoritmos requieren de los mismos datos, por lo que a continuación se indica los parámetros que utiliza cada clasificador.

Previamente los datos obtenidos en la caracterización deben ser normalizados, debido a que la adquisición de señales no tiene una amplitud de normalización referente a cada una de las características de análisis, es necesario la normalización de los datos dentro de un rango definido, permitiendo representar de mejor manera los datos característicos, es recomendable aplicar la técnica de normalización de mínimos y máximos [32]. Ver la ecuación (15).

$$Datos_{normalizados} = \frac{(x - x_{min}) * (d_{max} - d_{min})}{x_{max} - x_{min}} + d_{min} \quad (15)$$

Dónde: x es la matriz de característica a normalizar, xmin es el valor mínimo de la matriz, dmax y dmin son los limites definidos para el análisis y xmax es el valor máximo de la matriz

- Red Neuronal Backpropagation

El clasificador Red Neuronal de Retropropagación necesita de información para el entrenamiento de la red, estos parámetros son la matriz de entrada y salida, la configuración de la matriz de entrada está conformada por el tipo de movimiento, la cantidad de muestras y las características normalizadas como se observa en la Tabla 2.30.

Tabla 2.30
Configuración de datos de entrada para entrenamiento.

MOVIMIENTO ABDUCCIÓN	N° DE MUESTRA	Simple Cuadrado Integral	Raíz Media Cuadrática	Longitud de Onda
5 GRADOS	1	0,014	0,12	0,41
	2	0,014	0,11	0,39
	3	0,017	0,13	0,40
	:	:	:	:
	34	0,018	0,13	0,40
10 GRADOS	1	0,038	0,19	0,50
	2	0,034	0,18	0,51
	3	0,033	0,18	0,48
	:	:	:	:
	34	0,04	0,19	0,48
15 GRADOS	1	0,17	0,41	0,53
	2	0,11	0,34	0,55
	3	0,11	0,33	0,60
	:	:	:	:
	34	0,15	0,38	0,56
20 GRADOS	1	0,69	0,83	0,76
	2	1	1	0,93
	3	0,57	0,76	0,73
	:	:	:	:
	34	0,57	0,75	0,80

Los datos mostrados en la tabla, son valores normalizados en un rango de 0 a 1.

En este caso en particular la representación de los datos de entrada es del movimiento de Abducción; sin embargo, la misma configuración se utiliza para los demás movimientos.

Al igual que la matriz de entrada, la matriz de salida tiene una configuración para el entrenamiento de la red neuronal, dicha configuración se constituye por 4 salidas de forma binaria, la que representa a cada una de las posiciones del movimiento, observar la Tabla 2.31.

Tabla 2.31
Configuración de datos de salida para entrenamiento del clasificador Red Neuronal de Retropropagación.

MOVIMIENTO ABDUCCIÓN	N° DE MUESTRA	Y1	Y2	Y3	Y4
5 GRADOS	1	1	0	0	0
	2	1	0	0	0
	3	1	0	0	0
	:	:	:	:	:
	34	1	0	0	0
10 GRADOS	1	0	1	0	0
	2	0	1	0	0
	3	0	1	0	0
	:	:	:	:	:
	34	0	1	0	0
15 GRADOS	1	0	0	1	0
	2	0	0	1	0
	3	0	0	1	0
	:	:	:	:	:
	34	0	0	1	0
20 GRADOS	1	0	0	0	1
	2	0	0	0	1
	3	0	0	0	1
	:	:	:	:	:
	34	0	0	0	1

La red Neuronal al ser un algoritmo supervisado es un requerimiento indispensable que conozca lo que se quiere obtener como resultado, es por esto que se ingresa una matriz de salida. En este caso la configuración es del movimiento de Abducción; no obstante, es la misma para todos los movimientos.

Una vez obtenido la información de entrada y salida para el entrenamiento, se prosigue a ingresar estos parámetros en el código de entrenamiento; para tener una mejor idea de cómo ingresar los datos y de conocer cómo está trabajando el entrenador, se muestra una representación en forma de pseudocódigo, ver Figura 2.40.

```

Proceso Red_Neural_Backpropagation
|
Escribir 'Datos de entrada'; Leer Entrada;
Escribir 'Datos de salida'; Leer Salida;
Escribir 'Datos de prueba'; Leer Prueba;
Escribir 'Número de neuronas'; Leer Nneuronas;
Leer error;

Repetir
    Escribir 'Entrenar Red con datos de', Entrada, Salida;
    Si error=elevado Entonces
        Escribir 'Propagar hacia atras, (de fin a principio)'
    Sino
        Escribir 'Asigna los pesos a:', Nneuronas;
    Fin Si
Hasta Que Prueba=Salida;

FinProceso

```

Figura 2.40 Pseudocódigo del algoritmo Red Neuronal Backpropagation.

Esta representación permite conocer la estructura de creación del código y todos los parámetros fundamentales que necesita para que crear la red de entrenamiento.

- **Nearest Neighbor**

El clasificador Nearest Neighbor al ser un algoritmo supervisado al igual que el anterior necesita datos de entrada y datos de salida.

Los datos de entrada están representados en una matriz con valores resultantes de la caracterización, esta configuración es la misma que utiliza la red neuronal Backpropagation, conformada por el tipo de movimiento, número de muestras y las características, ver Tabla 2,30.

Para los datos salida la configuración cambia, ya que la salida no está representada de forma binaria; sino, que se identifica el tipo de movimiento con un valor numérico, observar Tabla 2.32.

Tabla 2.32

Configuración de datos de salida para entrenamiento del clasificador Nearest Neighbor.

MOVIMIENTO ABDUCCIÓN	Nº DE MUESTRA	SALIDA (Y1)
5 GRADOS	1	1
	2	1
	3	1
	:	:
	34	1
10 GRADOS	1	2
	2	2
	3	2
	:	:
	34	2
15 GRADOS	1	3
	2	3
	3	3
	:	:
	34	3
20 GRADOS	1	4
	2	4
	3	4
	:	:
	34	4

En este caso la configuración de salida es del movimiento de Abducción; no obstante, es la misma para todos los movimientos.

Se debe mencionar que, para comprender el funcionamiento del algoritmo es necesario tener una representación en forma de pseudocódigo previo a la programación de este; observar Figura 2.41, donde se especifica el ingreso correcto de la información, los parámetros necesarios del clasificador como el método de entrenamiento en número de vecinos y la estructura del código.

```

Proceso Nearest_Neighbor

  Escribir 'Datos de entrada'; Leer Entrada;
  Escribir 'Datos de salida'; Leer Salida;
  Escribir 'Datos de prueba'; Leer Prueba;
  Leer K; Leer CaracteristicaCercana; Leer Grupos;

  Repetir
    Escribir 'Entrenamiento con datos', Entrada, Salida;
    Repetir
      Mientras K = CaracteristicaCercana Hacer
        Escribir 'Asignar en:', Grupos;
      Fin Mientras
    Hasta Que Entrada=Grupos
  Hasta Que Prueba=Salida;

FinProceso

```

Figura 2.41 Pseudocódigo del Clasificador Nearest Neighbor.

- **Análisis Discriminantes**

El clasificador Análisis discriminante es un algoritmo de aprendizaje supervisado, esto quiero decir que al igual que los anteriores, requiere para su entrenamiento datos de entrada y datos de salida.

La configuración de los datos de entrada es la misma para estos clasificadores supervisados, por lo que se utiliza la misma que se observa en la Tabla 2.30, conformada por el tipo de movimiento, el número de muestras y las características.

Los datos de salida en este caso es el mismo que utiliza el clasificador Nearest Neighbor, ver Tabla 2,32, esta configuración es un identificador numérico para cada tipo de movimiento.

Además; se debe conocer la información que necesita el algoritmo y el uso correcta de esta, por lo que se representa en forma de pseudocódigo; observar Figura 2.42, el clasificador tiene una estructura propia, en la que se debe especificar parámetros como el método de entrenamiento, los identificadores de cada grupo, el tipo de entrenamiento la validación y las pruebas.

```

Proceso Análisis_Discriminante

    Escribir 'Datos de entrada'; Leer Entrada;
    Escribir 'Datos de salida'; Leer Salida;
    Escribir 'Datos de prueba'; Leer Prueba;
    Leer Grupo; Leer Caracteristica;

    Repetir
        Escribir 'Entrenamiento con datos', Entrada, Salida;
        Repetir
            Mientras Caracteristica != Grupo Hacer
                Escribir 'Discrimina o separa a otro grupo';
            Fin Mientras
        Hasta Que Entrada=Grupos
    Hasta Que Prueba=Salida;

FinProceso

```

Figura 2.42 Pseudocódigo del Clasificador Análisis Discriminante.

- Máquinas de soporte vectorial

Por último, está el clasificador supervisado Máquinas de Soporte Vectorial que como ya se sobre entiende, al ser supervisado es necesario de datos de entrada y datos de salida.

Los datos de entrada para el entrenamiento de este clasificador están conformados por el tipo de movimiento, el número de muestras y las características, mismos que utilizan todos los algoritmos para el entrenamiento, ver Tabla 2,30.

Este algoritmo también utiliza la misma configuración para los datos de salida, que el de Nearest Neighbor, ver Tabla 2,32, en la cual se asigna solo un valor numérico para identificar el tipo de movimiento.

Ya obtenido la información de entrada, se conforma la estructura del clasificador; previamente se realiza una representación en forma de pseudocódigo para comprender el comportamiento del algoritmo; observar Figura 2.43, cada clasificador tiene una configuración propia como es el caso de Máquinas de Soporte Vectorial que necesita de parámetros como el método de entrenamiento, funciones de Kernel, codificaciones; entre otros, además, es necesario de la validación y la puesta a prueba del algoritmo.

```

Proceso Maquinas_de_Soporte_Vectorial

Escribir 'Datos de entrada'; Leer Entrada;
Escribir 'Datos de salida'; Leer Salida;
Escribir 'Datos de prueba'; Leer Prueba;
Leer Hiperplano; Leer Caracteristica; Leer Clases;

Repetir
  Escribir 'Entrenamiento con datos', Entrada, Salida;
  Repetir
    Mientras Caracteristica = Hiperplano Hacer
      Escribir 'Asignar en:', Clases;
    Fin Mientras
  Hasta Que Entrada=Clases
Hasta Que Prueba=Salida;

FinProceso

```

Figura 2.43 Pseudocódigo del Clasificador Máquina de Soporte Vectorial.

- Clustering SOM

El algoritmo Clustering SOM es un clasificador de aprendizaje no supervisado, esto quiere decir, que tan solo necesita conocer los datos de entrada, debido a su método de clasificación que se basa en identificar grupos a través de centroides. La configuración de estos datos de entrada no varía para este clasificador por lo que usa la misma que el de Redes Neuronales, ver Tabla 2,30.

Es necesario conocer la forma en la que trabaja el clasificador, es por esto que, se realiza una representación en forma de pseudocódigo previo a la programación del algoritmo, observar Figura 2.44, el clasificador necesita de parámetros como el número de dimensiones que es para formar la topología de la red, debido a esto el valor de la dimensión debe ser el mismo; además, es necesario de la validación y la puesta a prueba del algoritmo.

```
Proceso Clustering_SOM

  Escribir 'Datos de entrada'; Leer Entrada;
  Escribir 'Datos de prueba'; Leer Prueba;
  Leer Caracteristicas; Leer Separables;

  Repetir
  .....
  Escribir 'Entrenamiento solo con datos de:', Entrada;
  Mientras Caracteristicas = Separables Hacer
  .....
    Asignar_a_Grupo<-Caracteristicas;
  Fin Mientras
  Hasta Que Prueba=Entrenamiento_con_Entrada;

FinProceso
```

Figura 2.44 Pseudocódigo del Clasificador Clustering SOM

CAPITULO III: PRUEBAS Y ANÁLISIS DE RESULTADOS

3.1 Pruebas de adquisición de señales EMG

Una vez establecido los músculos de estudio, del mismo modo que, la ubicación de los electrodos, se lleva a cabo la preparación de las tarjetas de adquisición y procesamiento, utilizando los 4 canales como primera prueba de adquisición, para esto, se lleva a la práctica la conexión de 4 tarjetas Olimex EKG-EMG y la tarjeta Discovery STM32F4, como se observa en la Figura 3.45.

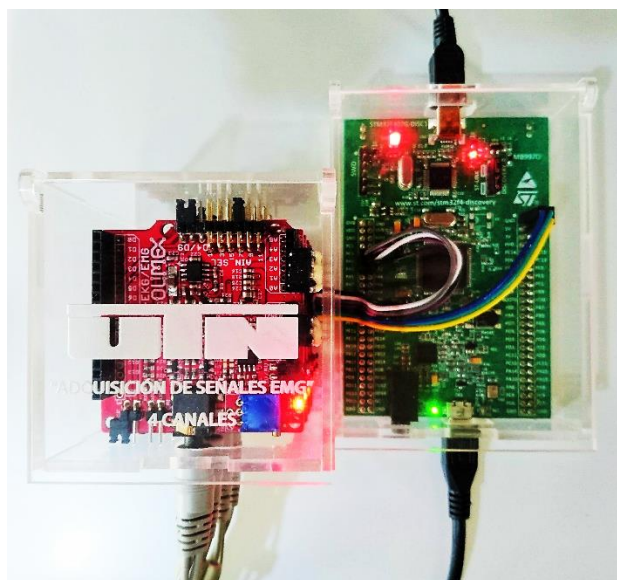


Figura 3.45 Sistema de adquisición de señales EMG de 4 canales

Por consiguiente, se procede a realizar la ubicación de los electrodos en los 4 músculos seleccionados anteriormente, este procedimiento debe ser evaluado por un especialista que certifique la ubicación correcta en la zona de estudio; una vez colocados los electrodos tomar en consideración que el conector de referencia L(negro) debe situarse una solo vez por el primer canal de la tarjeta de adquisición, observar Figura 3.46.

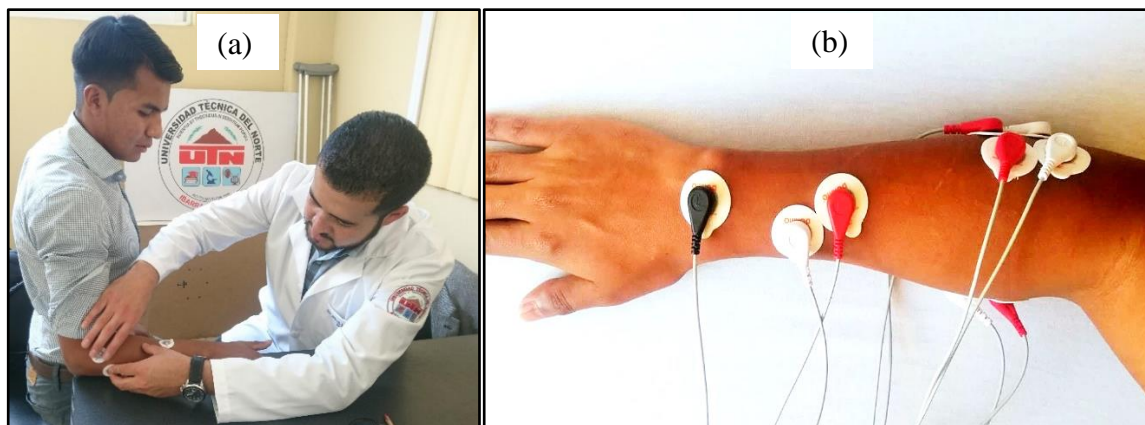


Figura 3.46 Ubicación de electrodos: (a) Por un especialista médico. (b) Para sistema de adquisición de 4 canales.

Ubicado los electrodos en los 4 músculos de estudio se realiza las primeras pruebas de los movimientos Flexión, Extensión, Abducción y Aducción, con el objetivo de conocer los canales que se activan al ejecutar un movimiento, observar Figura 3.47.

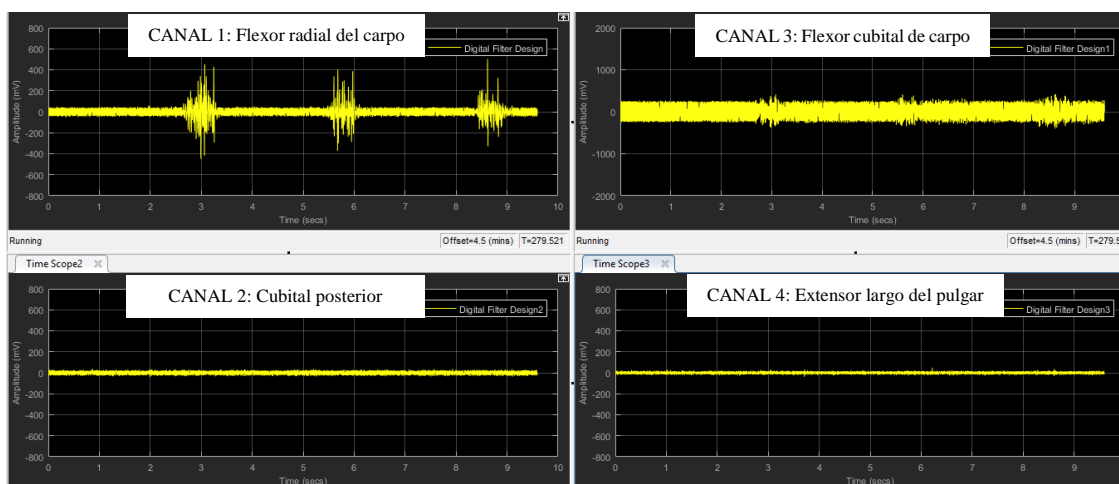


Figura 3.47 Adquisición de 4 canales ejecutando el movimiento Flexión de muñeca.

Está claro en la figura que, al momento de realizar el movimiento de flexión se activa el canal 1; el canal 2 que representa al movimiento extensión, como el 4 al movimiento de Abducción no presentan ninguna activación; no obstante, el canal 3 muestra unos pequeños indicios de actividad; sin embargo, no alcanza la misma amplitud que la del movimiento ejecutado.

A continuación, se ejecuta el movimiento Extensión de muñeca que representa el canal 2, ver Figura 3.48.

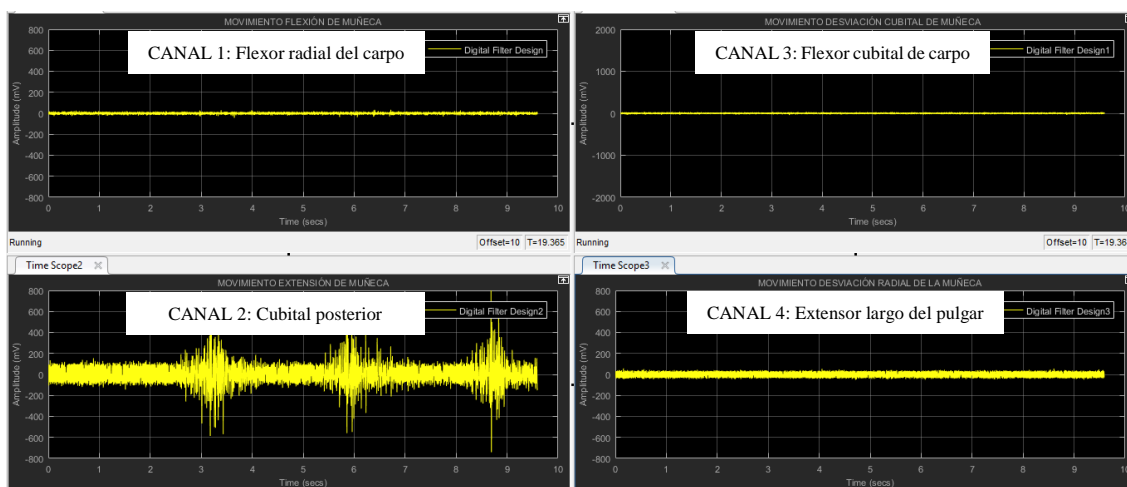


Figura 3.48 Adquisición de 4 canales ejecutando el movimiento Extensión de muñeca.

En la figura se muestra lucidamente que el canal que presenta activación es el número 2; o sea, el músculo Cubital posterior, a diferencia de los demás canales que no manifiestan ningún tipo de actividad; esto quiere decir que no hay nada en común con los otros músculos.

Posteriormente, se realiza el movimiento Aducción de muñeca, que se observa reflejado en el canal número 3 que indica la figura 3.49.

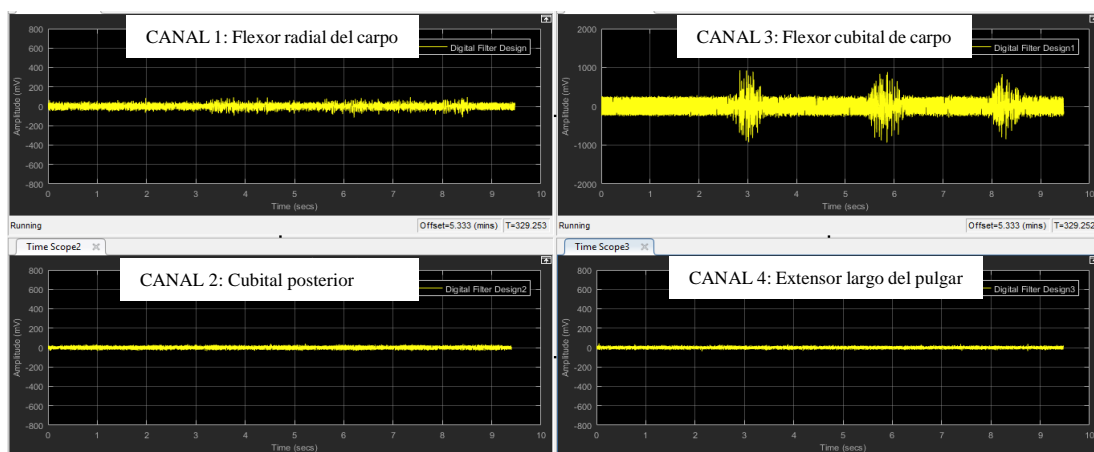


Figura 3.49 Adquisición de 4 canales ejecutando el movimiento Aducción de muñeca.

Es esta ocasión, se prueba la actividad que se presenta en los canales al ejecutar el movimiento de Abducción, como es evidente la activación se da únicamente en el canal 3; o sea, el músculo Flexor cubital del carpo; no obstante, los demás canales no presentan ningún tipo de actividad.

Por último, se lleva a cabo el movimiento Abducción de muñeca representando al canal número 4, como se observa en la Figura 3.50.

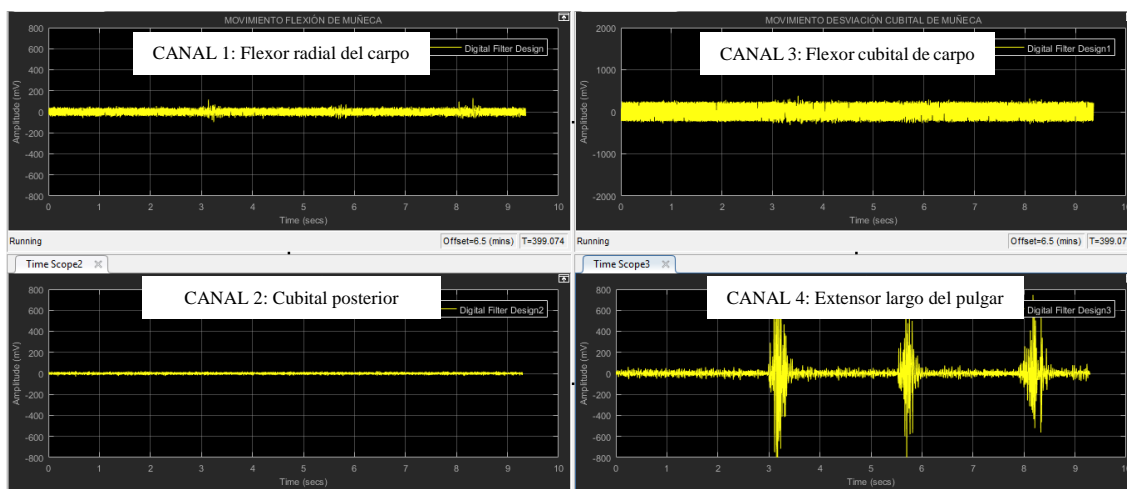


Figura 3.50 Adquisición de 4 canales ejecutando el movimiento Abducción de muñeca.

En esta oportunidad, se observa que al realizar el movimiento de Abducción se activa el canal 4, que representa al músculo Extensor largo del pulgar; de la misma manera, el canal 1 presenta unos pequeños indicios de activación; sin embargo, estos no alcanzan la misma amplitud que la del canal 4; respecto al canal 2 y 3 en todo momento están en estado de reposo.

Después haber realizado las pruebas con el sistema de adquisición de 4 canales, se puede decir como análisis, que permite ver con claridad que al ejecutar cada movimiento únicamente se activa su canal respectivo; con esto se llega a concluir, que no es necesario tener conectado las 4 tarjetas de adquisición, tan solo con una, se puede realizar el estudio, debido a que se obtendrá información exclusivamente del movimiento que se está ejecutando.

3.2 Pruebas de clasificación con algoritmos supervisados

3.2.1 Clasificador Red Neuronal de Backpropagation

El primer algoritmo supervisado puesto a prueba, es la red neuronal de tipo Feed Forward Backpropagation, que utiliza como función de entrenamiento el Gradiente Conjugado Escalado. Para tener un análisis de resultados amplio y seleccionar el algoritmo más eficiente, se realiza varias pruebas variando principalmente el número de neuronas y la función de activación en la capa oculta. Además, en el entrenamiento se cuenta con 2 bases de datos distintas, pero del mismo movimiento, con el objetivo de usar el 100% de una base de datos para el entrenamiento de la red y la otra base de datos especialmente para la validación del entrenamiento.

3.2.1.1 Análisis del Cambio de Función de Activación

Una vez definido los parámetros de la red, se realiza la primera prueba con los datos de entrada, en este caso se realiza la prueba con 3 y 11 características y se procede a varias la función de activación en la capa oculta teniendo como dato fijo el número de neuronas.

- Función de Activación Sigmoide (logsig)

Se establece 11 características como datos de entrada, 10 neuronas y la función de activación Sigmoide en la capa oculta, observe Figura 3.51.

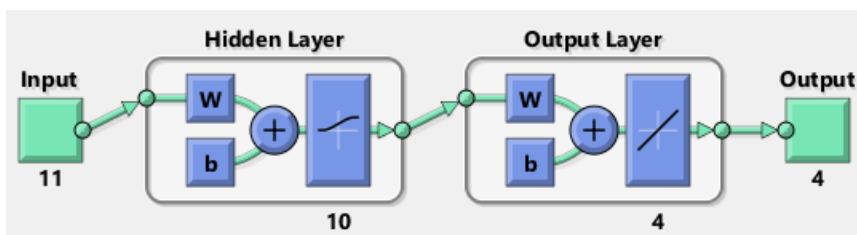


Figura 3.51 Estructura de red con función de activación sigmoide y 11 características.

Los resultados obtenidos después del entrenamiento se muestran en la Tabla 3.33.

Tabla 3.33
Resultados de la red neuronal con la función de activación SIGMOIDE y 11 características de entrada.

FUNCIÓN DE ACTIVACIÓN	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																																																
SIGMOIDE (logsig)	93 iteraciones	1 segundo	<table border="1"> <caption>Confusion Matrix</caption> <tr> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> <tr> <td>1</td> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <td>2</td> <td>0 0.0%</td> <td>34 25.0%</td> <td>25 18.4%</td> <td>0 0.0%</td> <td>57.6% 42.4%</td> </tr> <tr> <td>3</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>9 6.6%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <td>4</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>34 25.0%</td> <td>100% 0.0%</td> </tr> <tr> <td></td> <td>100% 0.0%</td> <td>100% 0.0%</td> <td>26.5% 73.5%</td> <td>100% 0.0%</td> <td>81.6% 18.4%</td> </tr> <tr> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> <tr> <td></td> <td colspan="4" style="text-align: center;">Target Class</td> <td></td> </tr> </table>		1	2	3	4		1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	34 25.0%	25 18.4%	0 0.0%	57.6% 42.4%	3	0 0.0%	0 0.0%	9 6.6%	0 0.0%	100% 0.0%	4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%		100% 0.0%	100% 0.0%	26.5% 73.5%	100% 0.0%	81.6% 18.4%		1	2	3	4			Target Class					81.6%
	1	2	3	4																																																
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																															
2	0 0.0%	34 25.0%	25 18.4%	0 0.0%	57.6% 42.4%																																															
3	0 0.0%	0 0.0%	9 6.6%	0 0.0%	100% 0.0%																																															
4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%																																															
	100% 0.0%	100% 0.0%	26.5% 73.5%	100% 0.0%	81.6% 18.4%																																															
	1	2	3	4																																																
	Target Class																																																			

Al Analizar los resultados, se obtiene que la red no es lo suficientemente eficaz, ya que alcanza un 81.6% de aciertos en la clasificación teniendo una confusión entre el segundo y tercer parámetro, para este resultado el entrenamiento de la red duró un tiempo de 1 segundo en 93 iteraciones.

Como segunda prueba de la red neuronal solo se utiliza los 3 datos característicos más importantes; en este caso, la raíz media cuadrática (RMS), simple cuadrado integral (SSI) y Longitud de forma de onda (WL), que son los mismos que se observó en la gráfica de características en el capítulo anterior.

De igual forma se establece 10 neuronas y la función de activación Sigmoides en la capa oculta; observe Figura 3.52, para el análisis de desempeño de la red en la clasificación.

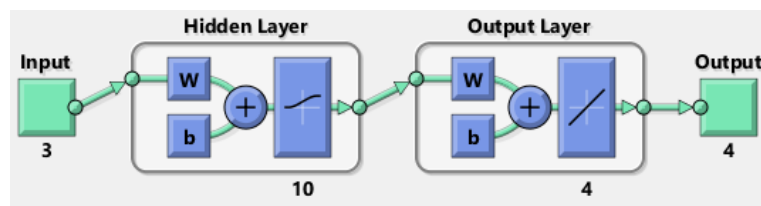


Figura 3.52 Estructura de red con función de activación sigmoide y 3 características.

Una vez configurado estos parámetros en la red, se realiza el entrenamiento para observar el comportamiento de esta; observar Tabla 3.34.

Tabla 3.34

Resultados de la red neuronal con la función de activación SIGMOIDE y 3 características de entrada.

FUNCIÓN DE ACTIVACIÓN	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																																																
SIGMOIDE (logsig)	42 iteraciones	0,76 segundos	<table border="1"> <caption>Confusion Matrix</caption> <tr> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> <tr> <td>1</td> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <td>2</td> <td>0 0.0%</td> <td>34 25.0%</td> <td>26 19.1%</td> <td>0 0.0%</td> <td>56.7% 43.3%</td> </tr> <tr> <td>3</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>8 5.9%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <td>4</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>34 25.0%</td> <td>100% 0.0%</td> </tr> <tr> <td></td> <td>100% 0.0%</td> <td>100% 0.0%</td> <td>23.5% 76.5%</td> <td>100% 0.0%</td> <td>80.9% 19.1%</td> </tr> <tr> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> <tr> <td></td> <td colspan="4" style="text-align: center;">Target Class</td> <td></td> </tr> </table>		1	2	3	4		1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	34 25.0%	26 19.1%	0 0.0%	56.7% 43.3%	3	0 0.0%	0 0.0%	8 5.9%	0 0.0%	100% 0.0%	4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%		100% 0.0%	100% 0.0%	23.5% 76.5%	100% 0.0%	80.9% 19.1%		1	2	3	4			Target Class					80.9%
	1	2	3	4																																																
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																															
2	0 0.0%	34 25.0%	26 19.1%	0 0.0%	56.7% 43.3%																																															
3	0 0.0%	0 0.0%	8 5.9%	0 0.0%	100% 0.0%																																															
4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%																																															
	100% 0.0%	100% 0.0%	23.5% 76.5%	100% 0.0%	80.9% 19.1%																																															
	1	2	3	4																																																
	Target Class																																																			

Una vez finalizado la simulación del entrenamiento, se obtiene que la red aun no cumple con una clasificación adecuada, ya que alcanza un 80.9% de aciertos en la clasificación teniendo una confusión entre el segundo y tercer parámetro, para este resultado el entrenamiento de la red duró un tiempo de 0 segundo en 42 iteraciones. Con esto se puede deducir que la función de activación sigmoide no es eficiente en este estudio.

- Función de Activación Lineal (purelin)

Como primera prueba con esta función de activación se establece 11 características de entrada, 10 neuronas y la función lineal en la capa oculta; observe Figura 3.53, para analizar el desempeño de la red neuronal.

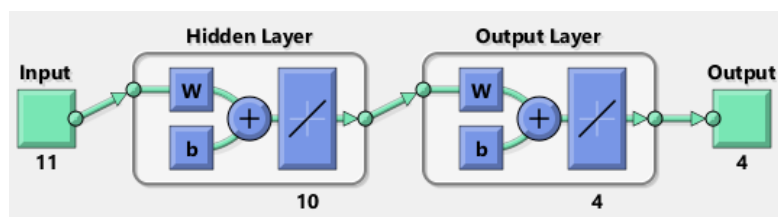


Figura 3.53 Estructura de red con función de activación lineal y 11 características.

Posteriormente, se realiza el entrenamiento de esta nueva estructura de red para obtener los resultados que se observan en la Tabla 3.35.

Tabla 3.35

Resultados de la red neuronal con la función de activación LINEAL y 11 características de entrada.

FUNCIÓN DE ACTIVACIÓN	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																																																
LINEAL (purelin)	56 iteraciones	1 segundo	<table border="1"> <caption>Confusion Matrix</caption> <tr> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> <tr> <td>1</td> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <td>2</td> <td>0 0.0%</td> <td>24 17.6%</td> <td>11 8.1%</td> <td>0 0.0%</td> <td>68.6% 31.4%</td> </tr> <tr> <td>3</td> <td>0 0.0%</td> <td>10 7.4%</td> <td>22 16.2%</td> <td>0 0.0%</td> <td>68.8% 31.3%</td> </tr> <tr> <td>4</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>1 0.7%</td> <td>34 25.0%</td> <td>97.1% 2.9%</td> </tr> <tr> <td></td> <td>100% 0.0%</td> <td>70.6% 29.4%</td> <td>64.7% 35.3%</td> <td>100% 0.0%</td> <td>83.8% 16.2%</td> </tr> <tr> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> <tr> <td></td> <td colspan="4" style="text-align: center;">Target Class</td> <td></td> </tr> </table>		1	2	3	4		1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	24 17.6%	11 8.1%	0 0.0%	68.6% 31.4%	3	0 0.0%	10 7.4%	22 16.2%	0 0.0%	68.8% 31.3%	4	0 0.0%	0 0.0%	1 0.7%	34 25.0%	97.1% 2.9%		100% 0.0%	70.6% 29.4%	64.7% 35.3%	100% 0.0%	83.8% 16.2%		1	2	3	4			Target Class					83.8%
	1	2	3	4																																																
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																															
2	0 0.0%	24 17.6%	11 8.1%	0 0.0%	68.6% 31.4%																																															
3	0 0.0%	10 7.4%	22 16.2%	0 0.0%	68.8% 31.3%																																															
4	0 0.0%	0 0.0%	1 0.7%	34 25.0%	97.1% 2.9%																																															
	100% 0.0%	70.6% 29.4%	64.7% 35.3%	100% 0.0%	83.8% 16.2%																																															
	1	2	3	4																																																
	Target Class																																																			

Finalizado el entrenamiento, se obtiene que esta estructura de red tampoco satisface la necesidad, ya que alcanza un 83.8% de aciertos en la clasificación teniendo una confusión entre el segundo y tercer parámetro, para este resultado el entrenamiento de la red tardó un tiempo de 1 segundo en 56 iteraciones.

Como segunda prueba de esta estructura de red neuronal, solo se utiliza las 3 características definidas anteriormente; para este caso se establece 10 neuronas y la función de activación lineal en la capa oculta; observe Figura 3.54, para analizar el desempeño de esta red.

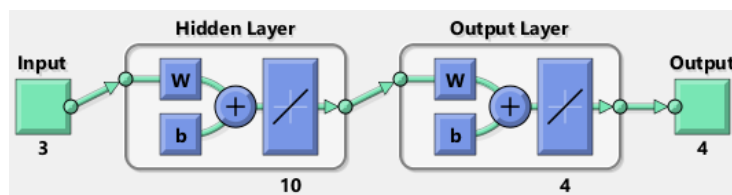


Figura 3.54 Estructura de red con función de activación lineal y 3 características.

Una vez creada la estructura de la red, se realiza el entrenamiento, para analizar los resultados que se muestran en la tabla 3.36.

Tabla 3.36

Resultados de la red neuronal con la función de activación LINEAL y 3 características de entrada.

FUNCIÓN DE ACTIVACIÓN	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																																															
LINEAL (purelin)	24 iteraciones	0,62 segundos	<table border="1"> <thead> <tr> <th colspan="5">Confusion Matrix</th> </tr> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th></th> </tr> </thead> <tbody> <tr> <th>1</th> <td>34 25.0%</td> <td>2 1.5%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>94.4% 5.6%</td> </tr> <tr> <th>2</th> <td>0 0.0%</td> <td>19 14.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <th>3</th> <td>0 0.0%</td> <td>13 9.6%</td> <td>32 23.5%</td> <td>1 0.7%</td> <td>69.6% 30.4%</td> </tr> <tr> <th>4</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>2 1.5%</td> <td>33 24.3%</td> <td>94.3% 5.7%</td> </tr> <tr> <th></th> <td>100% 0.0%</td> <td>55.9% 44.1%</td> <td>94.1% 5.9%</td> <td>97.1% 2.9%</td> <td>86.8% 13.2%</td> </tr> <tr> <th></th> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> </tbody> </table>	Confusion Matrix						1	2	3	4		1	34 25.0%	2 1.5%	0 0.0%	0 0.0%	94.4% 5.6%	2	0 0.0%	19 14.0%	0 0.0%	0 0.0%	100% 0.0%	3	0 0.0%	13 9.6%	32 23.5%	1 0.7%	69.6% 30.4%	4	0 0.0%	0 0.0%	2 1.5%	33 24.3%	94.3% 5.7%		100% 0.0%	55.9% 44.1%	94.1% 5.9%	97.1% 2.9%	86.8% 13.2%		1	2	3	4		86.8%
Confusion Matrix																																																			
	1	2	3	4																																															
1	34 25.0%	2 1.5%	0 0.0%	0 0.0%	94.4% 5.6%																																														
2	0 0.0%	19 14.0%	0 0.0%	0 0.0%	100% 0.0%																																														
3	0 0.0%	13 9.6%	32 23.5%	1 0.7%	69.6% 30.4%																																														
4	0 0.0%	0 0.0%	2 1.5%	33 24.3%	94.3% 5.7%																																														
	100% 0.0%	55.9% 44.1%	94.1% 5.9%	97.1% 2.9%	86.8% 13.2%																																														
	1	2	3	4																																															

Terminado el entrenamiento, se obtiene que esta estructura de red mejoró, pero aún no satisface la necesidad, ya que alcanza un 86.8% de aciertos en la clasificación teniendo una confusión en el segundo, tercer y cuarto parámetro, para este resultado el entrenamiento de la red duró un tiempo de 0 segundos en 24 iteraciones. Con esto se puede deducir que la función de activación lineal tampoco satisface con las necesidades para este estudio.

- **Tangente Hiperbólica (tansig)**

En la tercera prueba variando la función de activación se propone a la Tangente hiperbólica; con dicha función se conforma una red con 11 características como datos de entrada, 10 neuronas y la función mencionada en la capa oculta; observe Figura 3.55, para analizar el desempeño de la red neuronal creada.

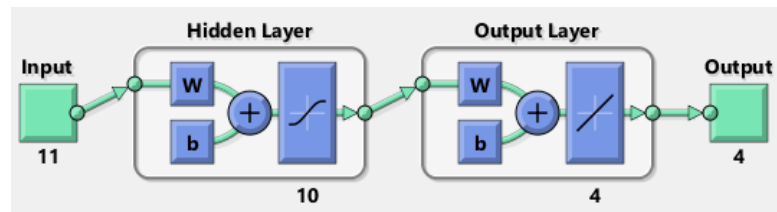
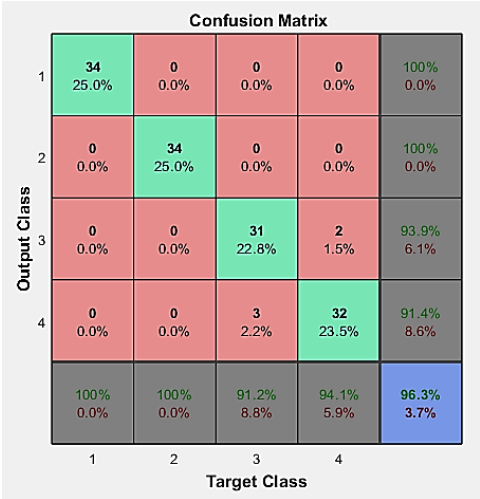


Figura 3.55 Estructura de red con función de activación tangente y 11 características.

Posteriormente, se realiza el entrenamiento de la red con esta nueva función, para analizar los resultados que se indican en la Tabla 3.37

Tabla 3.37
Resultados de la red neuronal con la función de activación TANGENTE y 11 características de entrada.

FUNCIÓN DE ACTIVACIÓN	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO
TANGENTE HIPERBÓLICA (tansig)	197 iteraciones	2 segundos		96.3%

Concluido el entrenamiento, se obtiene que esta estructura de red si satisface la necesidad, ya que alcanza un 96.3% de aciertos en la clasificación teniendo una mínima confusión entre el tercer y cuarto parámetro, para este resultado el entrenamiento de la red duró un tiempo de 2 segundos en 197 iteraciones.

Como segunda prueba con esta función se conforma una red con 3 características de entrada, 10 neuronas y la función de activación Tangente hiperbólica en la capa oculta; observe Figura 3.56, para analizar el desempeño de la red.

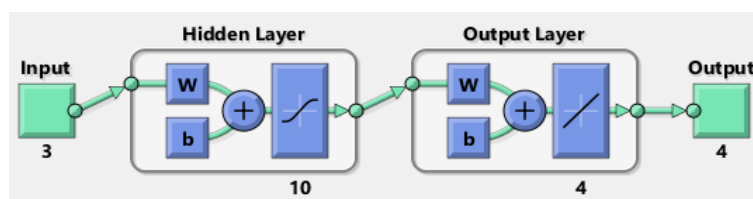


Figura 3.56 Estructura de red con función de activación tangente y 3 características.

A continuación, se realiza el entrenamiento con esta nueva estructura de red neuronal, para analizar los resultados que se indican en la Tabla 3.38

Tabla 3.38
Resultados de la red neuronal con la función de activación TANGENTE y 3 características de entrada.

FUNCIÓN DE ACTIVACIÓN	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																																				
TANGENTE HIPERBÓLICA (tansig)	116 iteraciones	0,97 segundos	<p>Confusion Matrix</p> <table border="1"> <tr> <td>1</td> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <td>2</td> <td>0 0.0%</td> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <td>3</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>34 25.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <td>4</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>34 25.0%</td> <td>100% 0.0%</td> </tr> <tr> <td></td> <td>100% 0.0%</td> <td>100% 0.0%</td> <td>100% 0.0%</td> <td>100% 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td></td> </tr> </table>	1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	34 25.0%	0 0.0%	0 0.0%	100% 0.0%	3	0 0.0%	0 0.0%	34 25.0%	0 0.0%	100% 0.0%	4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%		100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%		1	2	3	4		100%
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																			
2	0 0.0%	34 25.0%	0 0.0%	0 0.0%	100% 0.0%																																			
3	0 0.0%	0 0.0%	34 25.0%	0 0.0%	100% 0.0%																																			
4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%																																			
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%																																			
	1	2	3	4																																				

Ya concluido el entrenamiento, se obtiene que esta estructura de red es la más eficiente, ya que alcanza el 100% de aciertos en la clasificación, para este resultado el entrenamiento de la red duró un tiempo de 0 segundos en 116 iteraciones. Con esto se llega a deducir que la función de activación tangente satisface para este estudio y a comparación con la de 11 características esta red mejoró su desempeño al hacer una clasificación excelente.

Una vez finalizado todas las pruebas se lleva a cabo una comparación entre todas ellas, para seleccionar la más eficiente; observar Tabla 3.39.

Tabla 3.39
Resultados del Cambio de Función de Activación.

Función de Activación	Cantidad de Características	Número de neuronas	Desempeño de la Red (%)
Sigmoide	11	10	81,60%
	3	10	80,90%
Lineal	11	10	83,80%
	3	10	86,80%
Tangente	11	10	96,30%
	3	10	100%

Como es evidente, el desempeño de todas las redes indica la eficiencia en la clasificación, de lo cual se puede decir que con la función de activación Sigmoide y Lineal, no presentan una garantía suficiente; sin embargo, la función Tangente indica un alto rendimiento, siendo la más efectiva de las funciones, llegando al 100% en desempeño con una estructura conformado por 3 características como datos de entrada, 10 neuronas y dicha función en la capa oculta. Cabe recalcar que los resultados son particularmente del movimiento de Abducción.

3.2.1.2 Análisis del Cambio del Número de Neuronas

Una vez finaliza las pruebas cambiando la función de activación. Se prosigue a realizar pruebas variando el número de neuronas de la capa oculta en cada función y teniendo como datos de entrada 3 y 11 características.

- Función de Activación Sigmoide (logsig)

En el primer entrenamiento se comprueba el desempeño de la red con la Función de Activación sigmoide en la capa oculta y 11 características como datos de entrada, variando únicamente el número de neuronas, como se observa en la Tabla 3.40

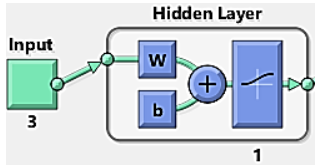
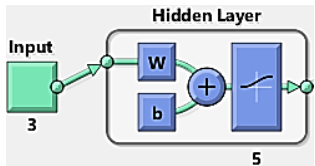
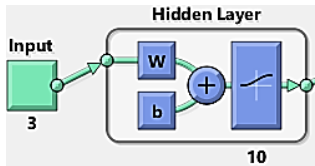
Tabla 3.40
Resultados variando el número de neuronas con función SIGMOIDE y 11 características.

FUNCIÓN DE ACTIVACIÓN SIGMOIDE																																									
Nº DE NEURONAS	ESTRUCTURA DE RED	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																																				
1		9 iteraciones	0,39 segundos	<table border="1"> <tr><td>1</td><td>34 25.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>2</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>NaN% NaN%</td></tr> <tr><td>3</td><td>0 0.0%</td><td>34 25.0%</td><td>34 25.0%</td><td>34 25.0%</td><td>53.3% 40.0%</td></tr> <tr><td>4</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>NaN% NaN%</td></tr> <tr><td></td><td>100% 0.0%</td><td>0.0% 100%</td><td>100% 0.0%</td><td>0.0% 100%</td><td>50.0% 50.0%</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr> </table>	1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	3	0 0.0%	34 25.0%	34 25.0%	34 25.0%	53.3% 40.0%	4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%		100% 0.0%	0.0% 100%	100% 0.0%	0.0% 100%	50.0% 50.0%		1	2	3	4		50 %
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																				
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%																																				
3	0 0.0%	34 25.0%	34 25.0%	34 25.0%	53.3% 40.0%																																				
4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%																																				
	100% 0.0%	0.0% 100%	100% 0.0%	0.0% 100%	50.0% 50.0%																																				
	1	2	3	4																																					
5		45 iteraciones	0,42 segundos	<table border="1"> <tr><td>1</td><td>34 25.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>2</td><td>0 0.0%</td><td>34 25.0%</td><td>1 0.7%</td><td>0 0.0%</td><td>57.1% 2.9%</td></tr> <tr><td>3</td><td>0 0.0%</td><td>0 0.0%</td><td>13 9.6%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>4</td><td>0 0.0%</td><td>0 0.0%</td><td>20 14.7%</td><td>34 25.0%</td><td>63.0% 37.4%</td></tr> <tr><td></td><td>100% 0.0%</td><td>100% 0.0%</td><td>35.2% 91.8%</td><td>100% 0.0%</td><td>84.6% 15.4%</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr> </table>	1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	34 25.0%	1 0.7%	0 0.0%	57.1% 2.9%	3	0 0.0%	0 0.0%	13 9.6%	0 0.0%	100% 0.0%	4	0 0.0%	0 0.0%	20 14.7%	34 25.0%	63.0% 37.4%		100% 0.0%	100% 0.0%	35.2% 91.8%	100% 0.0%	84.6% 15.4%		1	2	3	4		84.6%
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																				
2	0 0.0%	34 25.0%	1 0.7%	0 0.0%	57.1% 2.9%																																				
3	0 0.0%	0 0.0%	13 9.6%	0 0.0%	100% 0.0%																																				
4	0 0.0%	0 0.0%	20 14.7%	34 25.0%	63.0% 37.4%																																				
	100% 0.0%	100% 0.0%	35.2% 91.8%	100% 0.0%	84.6% 15.4%																																				
	1	2	3	4																																					
10		93 iteraciones	1 segundo	<table border="1"> <tr><td>1</td><td>34 25.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>2</td><td>0 0.0%</td><td>34 25.0%</td><td>25 18.4%</td><td>0 0.0%</td><td>57.6% 42.4%</td></tr> <tr><td>3</td><td>0 0.0%</td><td>0 0.0%</td><td>9 6.6%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>4</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>34 25.0%</td><td>100% 0.0%</td></tr> <tr><td></td><td>100% 0.0%</td><td>100% 0.0%</td><td>25.2% 72.8%</td><td>100% 0.0%</td><td>81.6% 18.4%</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr> </table>	1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	34 25.0%	25 18.4%	0 0.0%	57.6% 42.4%	3	0 0.0%	0 0.0%	9 6.6%	0 0.0%	100% 0.0%	4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%		100% 0.0%	100% 0.0%	25.2% 72.8%	100% 0.0%	81.6% 18.4%		1	2	3	4		81.6%
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																				
2	0 0.0%	34 25.0%	25 18.4%	0 0.0%	57.6% 42.4%																																				
3	0 0.0%	0 0.0%	9 6.6%	0 0.0%	100% 0.0%																																				
4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%																																				
	100% 0.0%	100% 0.0%	25.2% 72.8%	100% 0.0%	81.6% 18.4%																																				
	1	2	3	4																																					

Se demuestra claramente en la tabla de resultados, que el número de neuronas es un parámetro muy fundamental para que la red neuronal mejore su rendimiento, como se observa a mayor número de neuronas mayor efectividad en la clasificación, sin embargo, esta estructura de red no es óptima para implementar en el sistema debido a la función de activación que utiliza.

Para esta ocasión, se decide analizar el desempeño de las distintas redes neuronales creadas con la función de activación sigmoide, pero con las 3 características principales de entrada y variando el número de neuronas, ver Tabla 3.41.

Tabla 3.41
Resultados variando el número de neuronas con función SIGMOIDE y 3 características.

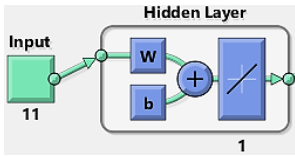
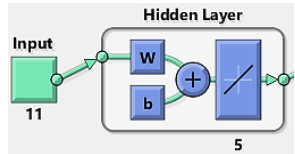
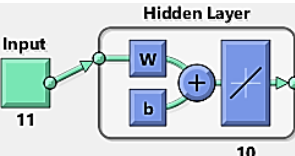
FUNCIÓN DE ACTIVACIÓN SIGMOIDE																																									
Nº DE NEURONAS	ESTRUCTURA DE RED	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																																				
1		16 iteraciones	0,25 segundos	<table border="1"> <thead> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th></th> </tr> </thead> <tbody> <tr> <th>1</th> <td>34 25.0%</td> <td>30 22.1%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>66.1% 46.6%</td> </tr> <tr> <th>2</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>NAN% NAN%</td> </tr> <tr> <th>3</th> <td>0 0.0%</td> <td>4 2.9%</td> <td>34 25.0%</td> <td>34 25.0%</td> <td>67.5% 52.8%</td> </tr> <tr> <th>4</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>NAN% NAN%</td> </tr> <tr> <th></th> <td>100% 0.0%</td> <td>0.0% 100%</td> <td>100% 0.0%</td> <td>100% 100%</td> <td>80.6% 50.6%</td> </tr> </tbody> </table>		1	2	3	4		1	34 25.0%	30 22.1%	0 0.0%	0 0.0%	66.1% 46.6%	2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NAN% NAN%	3	0 0.0%	4 2.9%	34 25.0%	34 25.0%	67.5% 52.8%	4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NAN% NAN%		100% 0.0%	0.0% 100%	100% 0.0%	100% 100%	80.6% 50.6%	50%
	1	2	3	4																																					
1	34 25.0%	30 22.1%	0 0.0%	0 0.0%	66.1% 46.6%																																				
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NAN% NAN%																																				
3	0 0.0%	4 2.9%	34 25.0%	34 25.0%	67.5% 52.8%																																				
4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NAN% NAN%																																				
	100% 0.0%	0.0% 100%	100% 0.0%	100% 100%	80.6% 50.6%																																				
5		21 iteraciones	0,37 segundos	<table border="1"> <thead> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th></th> </tr> </thead> <tbody> <tr> <th>1</th> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <th>2</th> <td>0 0.0%</td> <td>34 25.0%</td> <td>26 19.1%</td> <td>0 0.0%</td> <td>57.1% 43.3%</td> </tr> <tr> <th>3</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>NAN% NAN%</td> </tr> <tr> <th>4</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>8 5.9%</td> <td>34 25.0%</td> <td>81.0% 19.0%</td> </tr> <tr> <th></th> <td>100% 0.0%</td> <td>100% 0.0%</td> <td>0.0% 100%</td> <td>100% 0.0%</td> <td>78.6% 25.0%</td> </tr> </tbody> </table>		1	2	3	4		1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	34 25.0%	26 19.1%	0 0.0%	57.1% 43.3%	3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NAN% NAN%	4	0 0.0%	0 0.0%	8 5.9%	34 25.0%	81.0% 19.0%		100% 0.0%	100% 0.0%	0.0% 100%	100% 0.0%	78.6% 25.0%	75%
	1	2	3	4																																					
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																				
2	0 0.0%	34 25.0%	26 19.1%	0 0.0%	57.1% 43.3%																																				
3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NAN% NAN%																																				
4	0 0.0%	0 0.0%	8 5.9%	34 25.0%	81.0% 19.0%																																				
	100% 0.0%	100% 0.0%	0.0% 100%	100% 0.0%	78.6% 25.0%																																				
10		42 iteraciones	0,76 segundos	<table border="1"> <thead> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th></th> </tr> </thead> <tbody> <tr> <th>1</th> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <th>2</th> <td>0 0.0%</td> <td>34 25.0%</td> <td>26 19.1%</td> <td>0 0.0%</td> <td>56.7% 43.3%</td> </tr> <tr> <th>3</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>8 5.9%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <th>4</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>34 25.0%</td> <td>100% 0.0%</td> </tr> <tr> <th></th> <td>100% 0.0%</td> <td>100% 0.0%</td> <td>23.5% 76.5%</td> <td>100% 0.0%</td> <td>80.9% 19.1%</td> </tr> </tbody> </table>		1	2	3	4		1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	34 25.0%	26 19.1%	0 0.0%	56.7% 43.3%	3	0 0.0%	0 0.0%	8 5.9%	0 0.0%	100% 0.0%	4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%		100% 0.0%	100% 0.0%	23.5% 76.5%	100% 0.0%	80.9% 19.1%	80.9%
	1	2	3	4																																					
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																				
2	0 0.0%	34 25.0%	26 19.1%	0 0.0%	56.7% 43.3%																																				
3	0 0.0%	0 0.0%	8 5.9%	0 0.0%	100% 0.0%																																				
4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%																																				
	100% 0.0%	100% 0.0%	23.5% 76.5%	100% 0.0%	80.9% 19.1%																																				

Una vez finalizado el entrenamiento, se obtiene como resultados que la eficiencia de clasificación con 1 y 5 neuronas es muy bajo ya que se alcanza el 50% y 75% de acierto respectivamente; no obstante, el resultado con 10 neuronas mejora significativamente, sin embargo, no satisface las necesidades para la implementación en el sistema.

- Función de Activación Lineal (purelin)

La siguiente prueba de entrenamiento se realiza con la función de activación Lineal y las 11 características de entrada, ver Tabla 3.42, en este caso se crea 3 estructuras de red distintas variando únicamente el número de neuronas

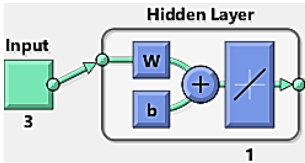
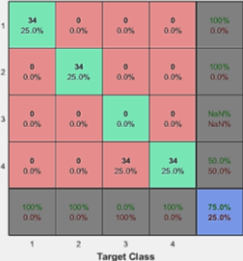
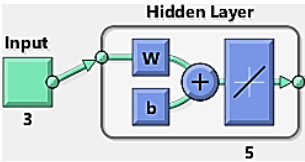
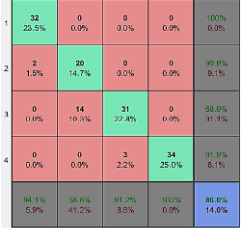
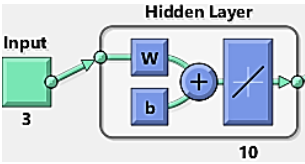
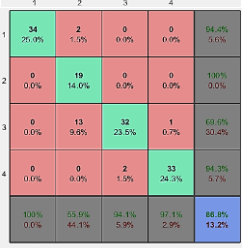
Tabla 3.42
Resultados variando el número de neuronas con función LINEAL y 11 características.

FUNCIÓN DE ACTIVACIÓN LINEAL																																									
Nº DE NEURONAS	ESTRUCTURA DE RED	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																																				
1		20 iteraciones	0,53 segundos	<table border="1"> <tr><td>1</td><td>34 25.0%</td><td>12 8.9%</td><td>0 0.0%</td><td>0 0.0%</td><td>73.9% 26.1%</td></tr> <tr><td>2</td><td>0 0.0%</td><td>22 16.2%</td><td>0 0.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>3</td><td>0 0.0%</td><td>0 0.0%</td><td>1 0.7%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>4</td><td>0 0.0%</td><td>0 0.0%</td><td>33 24.3%</td><td>34 25.0%</td><td>50.7% 49.3%</td></tr> <tr><td></td><td>100% 0.0%</td><td>56.1% 41.2%</td><td>2.0% 1.5%</td><td>100% 0.0%</td><td>88.9% 11.1%</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr> </table>	1	34 25.0%	12 8.9%	0 0.0%	0 0.0%	73.9% 26.1%	2	0 0.0%	22 16.2%	0 0.0%	0 0.0%	100% 0.0%	3	0 0.0%	0 0.0%	1 0.7%	0 0.0%	100% 0.0%	4	0 0.0%	0 0.0%	33 24.3%	34 25.0%	50.7% 49.3%		100% 0.0%	56.1% 41.2%	2.0% 1.5%	100% 0.0%	88.9% 11.1%		1	2	3	4		66.9%
1	34 25.0%	12 8.9%	0 0.0%	0 0.0%	73.9% 26.1%																																				
2	0 0.0%	22 16.2%	0 0.0%	0 0.0%	100% 0.0%																																				
3	0 0.0%	0 0.0%	1 0.7%	0 0.0%	100% 0.0%																																				
4	0 0.0%	0 0.0%	33 24.3%	34 25.0%	50.7% 49.3%																																				
	100% 0.0%	56.1% 41.2%	2.0% 1.5%	100% 0.0%	88.9% 11.1%																																				
	1	2	3	4																																					
5		37 iteraciones	0,71 segundos	<table border="1"> <tr><td>1</td><td>34 25.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>2</td><td>0 0.0%</td><td>20 14.7%</td><td>9 6.6%</td><td>0 0.0%</td><td>99.0% 0.1%</td></tr> <tr><td>3</td><td>0 0.0%</td><td>14 10.3%</td><td>22 16.2%</td><td>1 0.7%</td><td>59.5% 40.5%</td></tr> <tr><td>4</td><td>0 0.0%</td><td>0 0.0%</td><td>3 2.2%</td><td>33 24.3%</td><td>31.7% 68.3%</td></tr> <tr><td></td><td>100% 0.0%</td><td>68.9% 51.2%</td><td>64.7% 47.3%</td><td>97.1% 71.3%</td><td>80.1% 19.9%</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr> </table>	1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	20 14.7%	9 6.6%	0 0.0%	99.0% 0.1%	3	0 0.0%	14 10.3%	22 16.2%	1 0.7%	59.5% 40.5%	4	0 0.0%	0 0.0%	3 2.2%	33 24.3%	31.7% 68.3%		100% 0.0%	68.9% 51.2%	64.7% 47.3%	97.1% 71.3%	80.1% 19.9%		1	2	3	4		80.1%
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																				
2	0 0.0%	20 14.7%	9 6.6%	0 0.0%	99.0% 0.1%																																				
3	0 0.0%	14 10.3%	22 16.2%	1 0.7%	59.5% 40.5%																																				
4	0 0.0%	0 0.0%	3 2.2%	33 24.3%	31.7% 68.3%																																				
	100% 0.0%	68.9% 51.2%	64.7% 47.3%	97.1% 71.3%	80.1% 19.9%																																				
	1	2	3	4																																					
10		56 iteraciones	1 segundo	<table border="1"> <tr><td>1</td><td>34 25.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>2</td><td>0 0.0%</td><td>24 17.6%</td><td>11 8.1%</td><td>0 0.0%</td><td>88.9% 11.1%</td></tr> <tr><td>3</td><td>0 0.0%</td><td>19 14.1%</td><td>22 16.2%</td><td>0 0.0%</td><td>88.8% 11.2%</td></tr> <tr><td>4</td><td>0 0.0%</td><td>0 0.0%</td><td>1 0.7%</td><td>34 25.0%</td><td>97.1% 2.9%</td></tr> <tr><td></td><td>100% 0.0%</td><td>70.6% 52.4%</td><td>64.7% 47.3%</td><td>93% 68%</td><td>83.8% 16.2%</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr> </table>	1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	24 17.6%	11 8.1%	0 0.0%	88.9% 11.1%	3	0 0.0%	19 14.1%	22 16.2%	0 0.0%	88.8% 11.2%	4	0 0.0%	0 0.0%	1 0.7%	34 25.0%	97.1% 2.9%		100% 0.0%	70.6% 52.4%	64.7% 47.3%	93% 68%	83.8% 16.2%		1	2	3	4		83.8%
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																				
2	0 0.0%	24 17.6%	11 8.1%	0 0.0%	88.9% 11.1%																																				
3	0 0.0%	19 14.1%	22 16.2%	0 0.0%	88.8% 11.2%																																				
4	0 0.0%	0 0.0%	1 0.7%	34 25.0%	97.1% 2.9%																																				
	100% 0.0%	70.6% 52.4%	64.7% 47.3%	93% 68%	83.8% 16.2%																																				
	1	2	3	4																																					

Después de haber finalizado las pruebas, se evidencia que la cantidad de neuronas es un parámetro principal para mejorar el rendimiento del sistema de clasificación; sin embargo, ninguna de estas estructuras es eficiente para la implementación en el sistema ya que todas están por debajo del 90% de desempeño.

En este entrenamiento al igual que el anterior se tiene como parámetros la misma función de activación en la capa oculta y se varía el número de neuronas, pero ahora con 3 características principales como datos de entrada, observar Tabla 3.43.

Tabla 3.43
Resultados variando el número de neuronas con función LINEAL y 3 características.

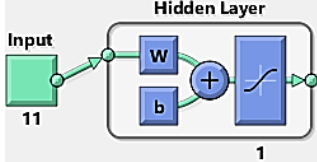
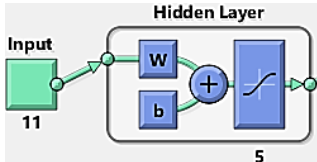
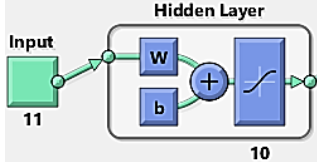
FUNCIÓN DE ACTIVACIÓN LINEAL					
Nº DE NEURONAS	ESTRUCTURA DE RED	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO
1		34 iteraciones	0,31 segundos		75%
5		28 iteraciones	0.45 segundos		86%
10		24 iteraciones	0,62 segundos		86.8%

Al igual que en los resultados anteriores se comprueba que con 1 neurona la eficiencia de la red es muy baja; no obstante, con 5 y 10 neuronas la red mejora significativamente alcanzando el 86 y 86.8% de aciertos en la clasificación respectivamente; sin embargo, esto demuestra que la función y el número de neuronas nos son los correctos, es por esto que no son óptimas para la implementación en el sistema.

- Tangente Hiperbólica (tansig)

Por último, se realiza las pruebas de entrenamiento con la función de activación Tangente Hiperbólica, variando la cantidad de neuronas y las 11 características como datos de entrada, Observar la Tabla 3.44.

Tabla 3.44
Resultados variando el número de neuronas con función TANGENTE y 11 características.

FUNCIÓN DE ACTIVACIÓN TANGENTE HIPERBÓLICA																																															
N° DE NEURONAS	ESTRUCTURA DE RED	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																																										
1		45 iteraciones	0,66 segundos	<table border="1"> <thead> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th></th> </tr> </thead> <tbody> <tr> <th>1</th> <td>34 25.0%</td> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>50.0% 50.0%</td> </tr> <tr> <th>2</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>NaN% NaN%</td> </tr> <tr> <th>3</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>34 25.0%</td> <td>34 25.0%</td> <td>50.0% 50.0%</td> </tr> <tr> <th>4</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>NaN% NaN%</td> </tr> <tr> <th></th> <td>100% 0.0%</td> <td>0.0% 0.0%</td> <td>100% 0.0%</td> <td>0.0% 100%</td> <td>50.0% 50.0%</td> </tr> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th></th> </tr> </tbody> </table>		1	2	3	4		1	34 25.0%	34 25.0%	0 0.0%	0 0.0%	50.0% 50.0%	2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	3	0 0.0%	0 0.0%	34 25.0%	34 25.0%	50.0% 50.0%	4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%		100% 0.0%	0.0% 0.0%	100% 0.0%	0.0% 100%	50.0% 50.0%		1	2	3	4		50%
	1	2	3	4																																											
1	34 25.0%	34 25.0%	0 0.0%	0 0.0%	50.0% 50.0%																																										
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%																																										
3	0 0.0%	0 0.0%	34 25.0%	34 25.0%	50.0% 50.0%																																										
4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%																																										
	100% 0.0%	0.0% 0.0%	100% 0.0%	0.0% 100%	50.0% 50.0%																																										
	1	2	3	4																																											
5		112 iteraciones	1 segundo	<table border="1"> <thead> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th></th> </tr> </thead> <tbody> <tr> <th>1</th> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <th>2</th> <td>0 0.0%</td> <td>31 22.8%</td> <td>24 17.6%</td> <td>0 0.0%</td> <td>55.4% 43.6%</td> </tr> <tr> <th>3</th> <td>0 0.0%</td> <td>3 2.2%</td> <td>9 6.6%</td> <td>0 0.0%</td> <td>75.0% 25.0%</td> </tr> <tr> <th>4</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>1 0.7%</td> <td>34 25.0%</td> <td>97.1% 2.9%</td> </tr> <tr> <th></th> <td>100% 0.0%</td> <td>91.2% 8.8%</td> <td>25.5% 75.5%</td> <td>100% 0.0%</td> <td>79.4% 20.6%</td> </tr> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th></th> </tr> </tbody> </table>		1	2	3	4		1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	31 22.8%	24 17.6%	0 0.0%	55.4% 43.6%	3	0 0.0%	3 2.2%	9 6.6%	0 0.0%	75.0% 25.0%	4	0 0.0%	0 0.0%	1 0.7%	34 25.0%	97.1% 2.9%		100% 0.0%	91.2% 8.8%	25.5% 75.5%	100% 0.0%	79.4% 20.6%		1	2	3	4		79.4%
	1	2	3	4																																											
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																										
2	0 0.0%	31 22.8%	24 17.6%	0 0.0%	55.4% 43.6%																																										
3	0 0.0%	3 2.2%	9 6.6%	0 0.0%	75.0% 25.0%																																										
4	0 0.0%	0 0.0%	1 0.7%	34 25.0%	97.1% 2.9%																																										
	100% 0.0%	91.2% 8.8%	25.5% 75.5%	100% 0.0%	79.4% 20.6%																																										
	1	2	3	4																																											
10		197 iteraciones	2 segundos	<table border="1"> <thead> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th></th> </tr> </thead> <tbody> <tr> <th>1</th> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <th>2</th> <td>0 0.0%</td> <td>34 25.0%</td> <td>0 0.0%</td> <td>0 0.0%</td> <td>100% 0.0%</td> </tr> <tr> <th>3</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>31 22.8%</td> <td>2 1.5%</td> <td>99.0% 0.1%</td> </tr> <tr> <th>4</th> <td>0 0.0%</td> <td>0 0.0%</td> <td>3 2.2%</td> <td>32 25.5%</td> <td>91.4% 8.6%</td> </tr> <tr> <th></th> <td>100% 0.0%</td> <td>0.0% 0.0%</td> <td>91.2% 8.8%</td> <td>94.1% 5.9%</td> <td>96.2% 3.7%</td> </tr> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th></th> </tr> </tbody> </table>		1	2	3	4		1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	34 25.0%	0 0.0%	0 0.0%	100% 0.0%	3	0 0.0%	0 0.0%	31 22.8%	2 1.5%	99.0% 0.1%	4	0 0.0%	0 0.0%	3 2.2%	32 25.5%	91.4% 8.6%		100% 0.0%	0.0% 0.0%	91.2% 8.8%	94.1% 5.9%	96.2% 3.7%		1	2	3	4		96.3%
	1	2	3	4																																											
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																										
2	0 0.0%	34 25.0%	0 0.0%	0 0.0%	100% 0.0%																																										
3	0 0.0%	0 0.0%	31 22.8%	2 1.5%	99.0% 0.1%																																										
4	0 0.0%	0 0.0%	3 2.2%	32 25.5%	91.4% 8.6%																																										
	100% 0.0%	0.0% 0.0%	91.2% 8.8%	94.1% 5.9%	96.2% 3.7%																																										
	1	2	3	4																																											

Después de haber finalizado el entrenamiento y obtenido los resultados mostrados en las tablas anteriores se comprueba nuevamente la importancia de la cantidad de neuronas, evidenciando que a mayor neuronas mayor eficiencia en la clasificación, en este caso en particular la estructura con 10 neuronas alcanzó un alto desempeño con un porcentaje de acierto del 96%, siendo este óptimo para la implementación en el sistema.

Para finalizar los entrenamientos con redes neuronales, se analiza el comportamiento de estas, con la misma función de activación, variando el número de neuronas y con 3 características principales como datos de entrada, ver Tabla 3.45.

Tabla 3.45
Resultados variando el número de neuronas con función TANGENTE y 3 características.

FUNCIÓN DE ACTIVACIÓN TANGENTE HIPERBÓLICA																																									
Nº DE NEURONAS	ESTRUCTURA DE RED	EPOCAS	TIEMPO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																																				
1		11 iteraciones	0,76 segundos	<table border="1"> <tr><td>1</td><td>34 25.0%</td><td>11 8.1%</td><td>1 0.7%</td><td>0 0.0%</td><td>73.0% 26.1%</td></tr> <tr><td>2</td><td>0 0.0%</td><td>4 2.9%</td><td>0 0.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>3</td><td>0 0.0%</td><td>13 9.6%</td><td>33 24.3%</td><td>34 25.0%</td><td>41.3% 56.6%</td></tr> <tr><td>4</td><td>0 0.0%</td><td>6 4.4%</td><td>0 0.0%</td><td>0 0.0%</td><td>0.0% 100%</td></tr> <tr><td></td><td>100% 0.0%</td><td>11.8% 8.2%</td><td>37.1% 2.9%</td><td>0.0% 1.0%</td><td>32.2% 47.8%</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr> </table>	1	34 25.0%	11 8.1%	1 0.7%	0 0.0%	73.0% 26.1%	2	0 0.0%	4 2.9%	0 0.0%	0 0.0%	100% 0.0%	3	0 0.0%	13 9.6%	33 24.3%	34 25.0%	41.3% 56.6%	4	0 0.0%	6 4.4%	0 0.0%	0 0.0%	0.0% 100%		100% 0.0%	11.8% 8.2%	37.1% 2.9%	0.0% 1.0%	32.2% 47.8%		1	2	3	4		52.2%
1	34 25.0%	11 8.1%	1 0.7%	0 0.0%	73.0% 26.1%																																				
2	0 0.0%	4 2.9%	0 0.0%	0 0.0%	100% 0.0%																																				
3	0 0.0%	13 9.6%	33 24.3%	34 25.0%	41.3% 56.6%																																				
4	0 0.0%	6 4.4%	0 0.0%	0 0.0%	0.0% 100%																																				
	100% 0.0%	11.8% 8.2%	37.1% 2.9%	0.0% 1.0%	32.2% 47.8%																																				
	1	2	3	4																																					
5		21 iteraciones	0,88 segundos	<table border="1"> <tr><td>1</td><td>34 25.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>2</td><td>0 0.0%</td><td>34 25.0%</td><td>12 8.8%</td><td>0 0.0%</td><td>73.0% 26.1%</td></tr> <tr><td>3</td><td>0 0.0%</td><td>0 0.0%</td><td>8 5.9%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>4</td><td>0 0.0%</td><td>0 0.0%</td><td>14 10.3%</td><td>34 25.0%</td><td>70.8% 29.2%</td></tr> <tr><td></td><td>100% 0.0%</td><td>100% 0.0%</td><td>23.5% 7.5%</td><td>100% 0.0%</td><td>80.9% 19.1%</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr> </table>	1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	34 25.0%	12 8.8%	0 0.0%	73.0% 26.1%	3	0 0.0%	0 0.0%	8 5.9%	0 0.0%	100% 0.0%	4	0 0.0%	0 0.0%	14 10.3%	34 25.0%	70.8% 29.2%		100% 0.0%	100% 0.0%	23.5% 7.5%	100% 0.0%	80.9% 19.1%		1	2	3	4		80.9%
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																				
2	0 0.0%	34 25.0%	12 8.8%	0 0.0%	73.0% 26.1%																																				
3	0 0.0%	0 0.0%	8 5.9%	0 0.0%	100% 0.0%																																				
4	0 0.0%	0 0.0%	14 10.3%	34 25.0%	70.8% 29.2%																																				
	100% 0.0%	100% 0.0%	23.5% 7.5%	100% 0.0%	80.9% 19.1%																																				
	1	2	3	4																																					
10		116 iteraciones	0,97 segundos	<table border="1"> <tr><td>1</td><td>34 25.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>2</td><td>0 0.0%</td><td>34 25.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>3</td><td>0 0.0%</td><td>0 0.0%</td><td>34 25.0%</td><td>0 0.0%</td><td>100% 0.0%</td></tr> <tr><td>4</td><td>0 0.0%</td><td>0 0.0%</td><td>0 0.0%</td><td>34 25.0%</td><td>100% 0.0%</td></tr> <tr><td></td><td>100% 0.0%</td><td>100% 0.0%</td><td>100% 0.0%</td><td>100% 0.0%</td><td>100% 0.0%</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td></td></tr> </table>	1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	2	0 0.0%	34 25.0%	0 0.0%	0 0.0%	100% 0.0%	3	0 0.0%	0 0.0%	34 25.0%	0 0.0%	100% 0.0%	4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%		100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%		1	2	3	4		100%
1	34 25.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%																																				
2	0 0.0%	34 25.0%	0 0.0%	0 0.0%	100% 0.0%																																				
3	0 0.0%	0 0.0%	34 25.0%	0 0.0%	100% 0.0%																																				
4	0 0.0%	0 0.0%	0 0.0%	34 25.0%	100% 0.0%																																				
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%																																				
	1	2	3	4																																					

Al analizar la tabla de resultados obtenida, deja claro nuevamente que la cantidad de neuronas es un factor muy importante para que la red alcance un desempeño alto, debido a que las redes con 1 y 5 neuronas presentan un rendimiento bajo; no obstante, la estructura creada con 10 neuronas alcanza el 100% de clasificación, siendo la mejor red de todos los análisis realizados anteriormente.

Una vez finalizado todos los entrenamientos respectivos con redes neuronales, se realiza una comparación de desempeño entre todas las pruebas; ver Tabla 3.46, con la finalidad de encontrar la estructura con mayor rendimiento

Tabla 3.46
Resultados de la Variación del Número de Neuronas.

Función de Activación	Cantidad de Características como datos de Entrada	Número de neuronas	Desempeño de la Red (%)
SIGMOIDE	11	1	50%
		5	84,60%
		10	81,60%
	3	1	50%
		5	75%
		10	80,90%
LINEAL	11	1	66,90%
		5	80,10%
		10	83,80%
	3	1	75%
		5	86%
		10	86,80%
TANGENTE HIPERBÓLICA	11	1	50%
		5	79,40%
		10	96,30%
	3	1	52,20%
		5	80,90%
		10	100%

Es evidente que al aumentar la cantidad de neuronas el desempeño de la red mejora; Sin embargo, la eficiencia de la red depende mucho de la cantidad de características como datos de entrada y de la función de activación en la capa oculta; por todas las pruebas realizadas se concluye que estos 3 parámetros son indispensables en la creación de una red neuronal, como se observa la que mejor rendimiento alcanza es la red creada con 3 características, 10 neuronas y con función tangente hiperbólica, logrando el 100% de aciertos al clasificar.

3.2.2 Clasificador Nearest Neighbor (K-NN)

Una vez probado el entrenamiento con redes neuronales, se realiza pruebas con otros clasificadores con la finalidad de descubrir un método eficiente. En este caso se utiliza el clasificador Nearest Neighbor o Vecino más Cercano (KNN) con dos métodos distintos.

- K-NN con Distancia Métrica del Coseno

El primer clasificador KNN puesto a prueba utiliza el método de Distancia Métrica del Coseno, observar Figura 3.57, en este método se evidencia a través de la gráfica sección b, que las características clasificadas incorrectamente están marcadas con una (x), por lo que se llega a concluir que, existe una confusión entre el segundo, tercer y cuarto movimiento.

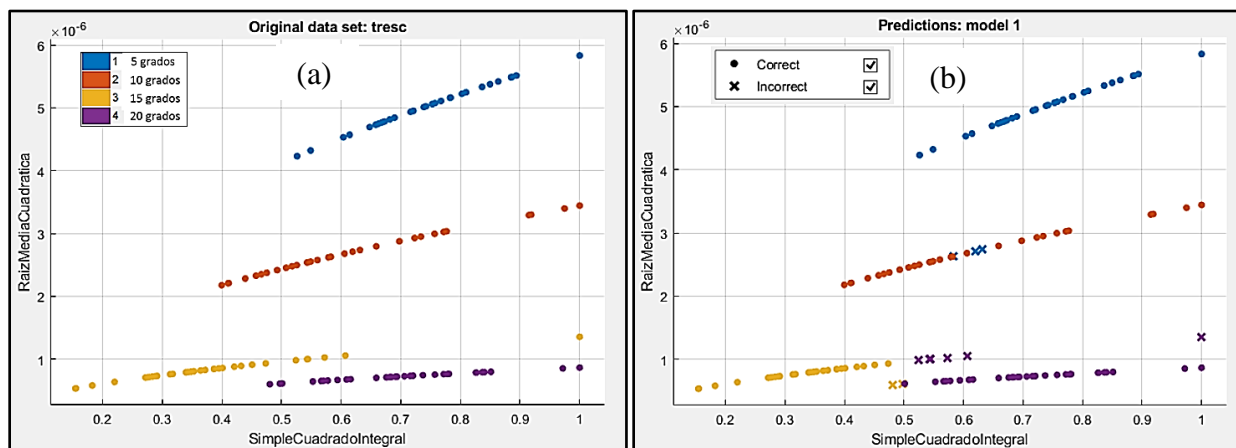


Figura 3.57 Pruebas de entrenamiento del clasificador KNN con modelo Distancia Métrica del Coseno: (a) Datos sin entrenamiento. (b) Datos entrenados.

- K-NN con Distancia Métrica Cúbica

En el segundo entrenamiento del clasificador K-NN, se utiliza el método de la Distancia Métrica Cúbica, observar Figura 3.58, es evidente que este método mejoró a comparación del anterior ya que como indica la gráfica no existe confusión entre los dos primeros movimientos.

Sin embargo, aun presenta una pequeña confusión entre el tercer y cuarto movimiento, estas confusiones están marcadas con una (x) en la gráfica sección b.

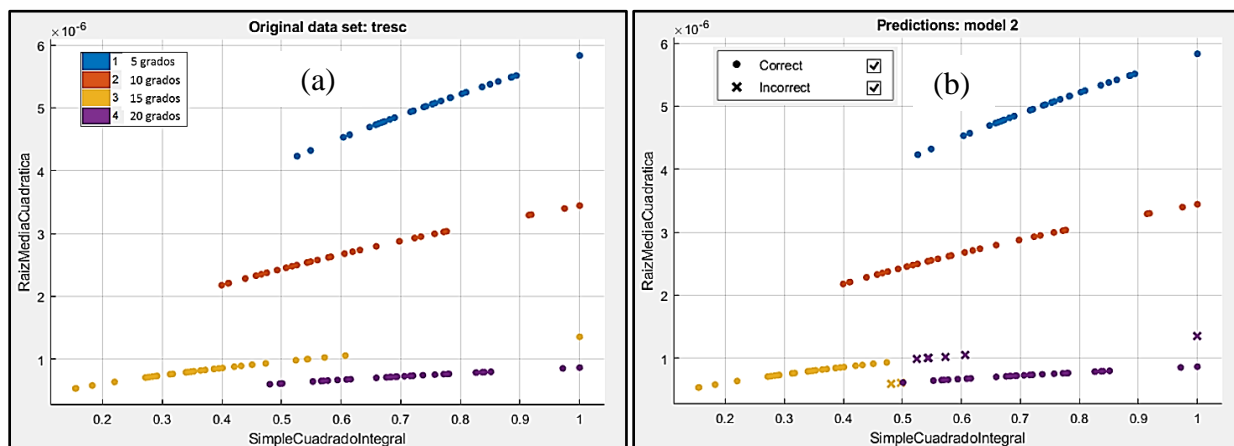


Figura 3.58 Pruebas de entrenamiento del clasificador KNN con modelo Distancia Métrica Cúbica: (a) Datos sin entrenamiento. (b) Datos entrenados.

Una vez terminado los entrenamientos, se muestran los resultados en la Tabla 3.47 para verificar si los errores mencionados anteriormente son ciertos.

Tabla 3.47 Resultados del clasificador K-NN con modelo Distancia métrica del coseno y cúbica.

CLASIFICADOR NEAREST NEIGHBOR																				
TIPO DE MODELO	NÚMERO DE VECINOS	TIEMPO DE ENTRENAMIENTO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																
Distancia métrica del coseno	10	4,451 segundos	<table border="1"> <caption>Model 1</caption> <tr><td>34</td><td></td><td></td><td></td></tr> <tr><td>3</td><td>31</td><td></td><td></td></tr> <tr><td></td><td></td><td>28</td><td>6</td></tr> <tr><td></td><td></td><td>2</td><td>32</td></tr> </table>	34				3	31					28	6			2	32	91.9%
34																				
3	31																			
		28	6																	
		2	32																	
Distancia métrica cúbica	10	1,2003 segundos	<table border="1"> <caption>Model 2</caption> <tr><td>34</td><td></td><td></td><td></td></tr> <tr><td></td><td>34</td><td></td><td></td></tr> <tr><td></td><td></td><td>28</td><td>6</td></tr> <tr><td></td><td></td><td>2</td><td>32</td></tr> </table>	34					34					28	6			2	32	94.1%
34																				
	34																			
		28	6																	
		2	32																	

Los resultados obtenidos indican que el clasificador K-NN con Distancia métrica del coseno, tiene un alto rendimiento; sin embargo, aún se evidencia que este modelo tiene problemas en el momento de clasificar el segundo, tercer y cuarto movimiento, existiendo una confusión entre ellos, está claro que, a comparación de los anteriores, este presenta una mejor garantía.

Además, en la tabla se observa los resultados alcanzados con el método Distancia métrica cúbica, confirmando que efectivamente el método mejoró, ya que logró un alto desempeño en la clasificación, obteniendo un 94.1% de aciertos, presentando una garantía lo suficientemente buena para la implementación en el sistema. Está claro en la matriz de confusión que aún existe un pequeño error entre el tercer y cuarto movimiento; sin embargo, puede ser considerable; además, este método tiene un menor tiempo de entrenamiento y un mayor rendimiento.

3.2.3 Clasificador Análisis Discriminante

Ahora se analiza el entrenamiento con el clasificador Análisis Discriminante; ver Figura 3.59, comparando entre las dos gráficas, a y b, es evidente que ya no existen errores en la clasificación a excepción del cuarto movimiento que presenta aún una pequeña confusión con el tercero.

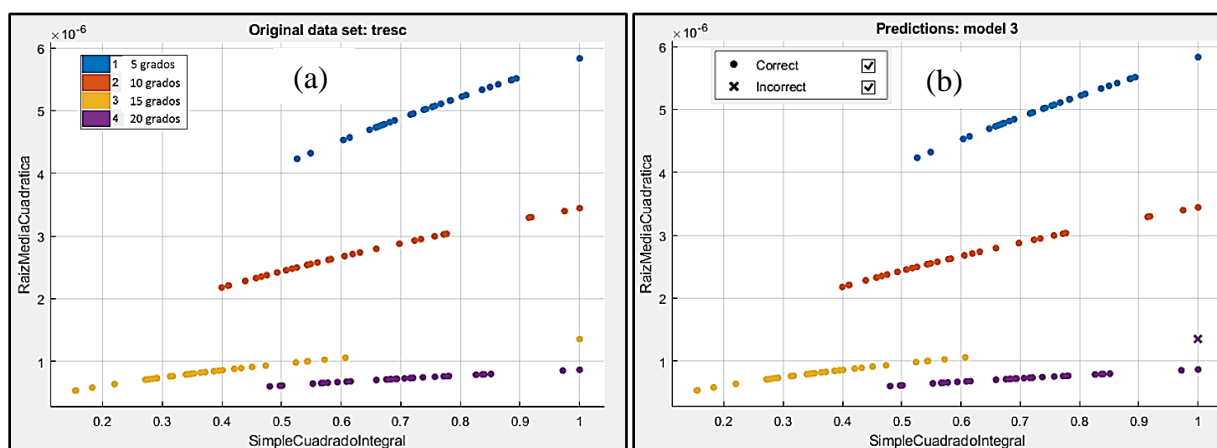
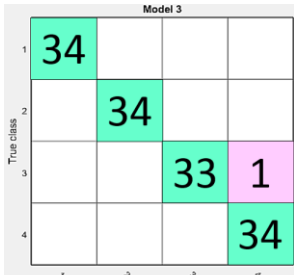


Figura 3.59 Pruebas de entrenamiento del clasificador Análisis Discriminante: (a) Datos sin entrenamiento. (b) Datos entrenados.

Luego de entrenar el clasificador se representa los resultados en la Tabla 3.48

Tabla 3.48
Resultados del clasificador Análisis Discriminante.

CLASIFICADOR ANÁLISIS DISCRIMINANTE				
TIPO DE MODELO	ESTRUCTURA DE COVARIANZA	TIEMPO DE ENTRENAMIENTO	MATRIZ DE CONFUSIÓN	% DE ACIERTO
Discriminante Lineal	Completa	1,9005 segundos		99.3%

Es perceptible en los resultados del clasificador Análisis Discriminante, observar Tabla 3.48, que alcanzo un excelente desempeño, clasificando casi todos los elementos correctamente en un tiempo muy corto, llegando a un 99.3% de aciertos en la clasificación.

3.2.4 Clasificador Máquinas de Soporte Vectorial (SVM)

Por último, se realiza el entrenamiento con el Clasificador Máquinas de Soporte Vectorial con el que se demuestra que es el mejor clasificador al no presentar ningún error en la clasificación.

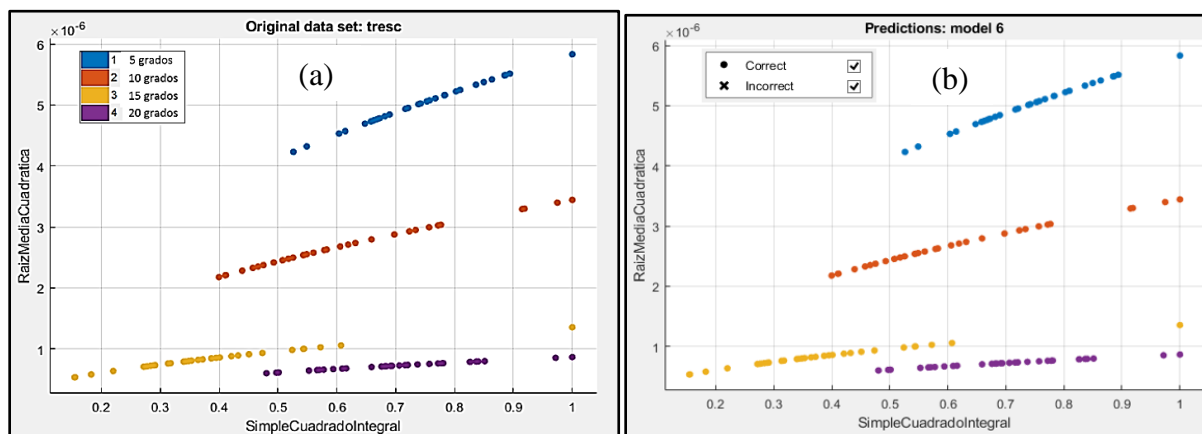
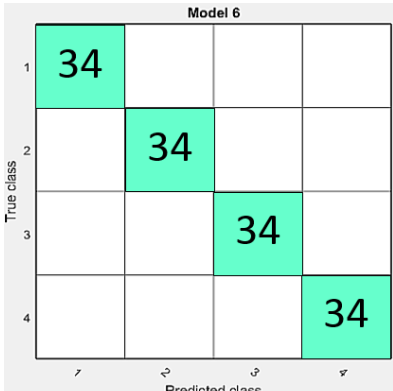


Figura 3.60 Pruebas de entrenamiento del clasificador Máquinas de Soporte Vectorial: (a) Datos sin entrenamiento. (b) Datos entrenados.

Posteriormente se indican los resultados del entrenamiento del clasificador Máquinas de Soporte Vectorial en la Tabla 3.49

Tabla 3.49
Resultados del clasificador Máquinas de Soporte Vectorial

CLASIFICADOR MÁQUINAS DE SOPORTE VECTORIAL																													
TIPO DE MODELO	FUNCIÓN DEL KERNEL	TIEMPO DE ENTRENAMIENTO	MATRIZ DE CONFUSIÓN	% DE ACIERTO																									
SVM Lineal	Lineal	3,0998 segundos	 <p>Model 6</p> <table border="1"> <tr> <td>1</td> <td>34</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td></td> <td>34</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td></td> <td></td> <td>34</td> <td></td> </tr> <tr> <td>4</td> <td></td> <td></td> <td></td> <td>34</td> </tr> <tr> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> </tr> </table> <p>True class</p> <p>Predicted class</p>	1	34				2		34			3			34		4				34		1	2	3	4	100%
1	34																												
2		34																											
3			34																										
4				34																									
	1	2	3	4																									

Concluido el entrenamiento con este clasificador, observar Tabla 3.50, se comprueba que es uno de los mejores clasificadores al presentar un desempeño excelente en la clasificación, alcanzando el 100% de aciertos en un tiempo considerable, esto quiere decir que es óptimo para la implementación en el sistema.

Luego de haber finalizado todos los entrenamientos con los algoritmos de aprendizaje supervisado, se lleva a cabo una comparación entre ellos para conocer el clasificador que mejores garantías presente, observar Tabla 3.50.

Tabla 3.50
Resultados de desempeño de clasificadores supervisados.

Clasificador	Datos de entrada	Tiempo de entrenamiento	Desempeño alcanzado (%)
Red Neuronal Backpropagation	3 Características	0,97 segundos	100%
K-NN con Distancia métrica del coseno	3 Características	4,451 segundos	91,90%
K-NN con Distancia métrica cúbica	3 Características	1,200 segundos	94,10%
Análisis Discriminante	3 Características	1, 900 segundos	99,30%
Máquinas de Soporte Vectorial	3 Características	3,099 segundos	100%

En la tabla comparativa 3.50 se puede evidenciar claramente que todos los algoritmos alcanzan un alto desempeño de clasificación, esto se debe a que son algoritmos que se utilizan en problemas en la que su información es linealmente separable, como en este caso. Analizando el porcentaje de desempeño de los clasificadores, se puede decir que la Red Neuronal Backpropagation y Máquinas de Soporte vectorial son los más eficientes en este estudio; Sin embargo, el tiempo de entrenamiento que tarda la Red Neuronal es mínimo, por lo que se considera como el mejor clasificador de aprendizaje supervisado.

Todas las pruebas realizadas con algoritmos de aprendizaje supervisados utilizaron la información del movimiento Abducción, con la finalidad de primero encontrar el algoritmo adecuado para luego implementarlo en los demás movimientos.

Inmediatamente después de seleccionar al clasificador Red Neuronal de Retropropagación, se realiza la prueba de entrenamiento con este algoritmo en los demás movimientos, con el objetivo de conocer la eficiencia en cada uno de ellos; observar Tabla 3.51.

Tabla 3.51
Resultados de Red Neuronal Backpropagation en todos los movimientos.

ALGORITMO RED NEURONAL BACKPROPAGATION		
MOVIMIENTO	TIEMPO DE ENTRENAMIENTO	DESEMPEÑO ALCANZADO
Abducción	0,97 segundos	100%
Aducción	0,66 segundos	99,30%
Flexión	0,82 segundos	100%
Extensión	1 segundo	99,30%

Es ostensible que los resultados que indica este clasificador son sobresalientes, al alcanzar un desempeño de más del 99% de aciertos en la clasificación de todos los movimientos, en el menor tiempo de entrenamiento posible. Por esta razón dicho algoritmo se implementa en el sistema.

3.3 Pruebas de clasificación con algoritmos no supervisados

Además de haber puesto a prueba métodos y algoritmos de aprendizaje de máquina supervisados, se plantea analizar el comportamiento de la clasificación con métodos de aprendizaje no supervisado, como es el caso del método Clustering.

3.3.1 Algoritmo Clustering por Mapa auto organizado (SOM)

El Mapa de autoorganización o el Mapa de Kohonen (SOM) es un algoritmo de aprendizaje competitivo basado en redes neuronales [36]. La capa de entrada representa un conjunto de vectores de entrada y la capa de salida contiene una serie de neuronas dispuestas en una cuadrícula hexagonal o rectangular; las neuronas en la capa de salida están completamente conectadas a las unidades en la capa de entrada con diferentes vectores de peso [37]. Observar Figura 3.61.

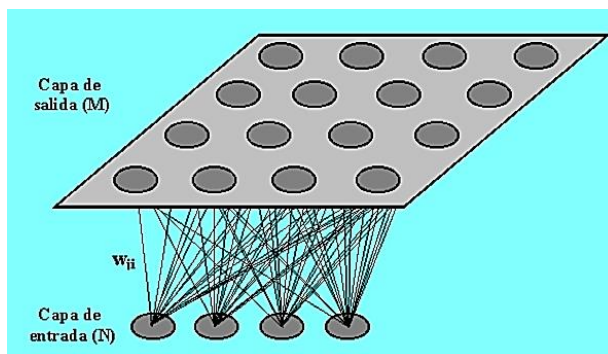


Figura 3.61 Arquitectura del SOM [38].

Donde N representa la capa de entrada; en este caso las características obtenidas de la señal, que son las encargadas de transmitir la información. M significa la capa de salida, se encarga de procesar la información y formar el mapa de rasgos, la capa de salida siempre se representada bidimensionalmente como se indica en la figura.

Teniendo los conceptos claros, se realiza el ingreso de las 3 características seleccionadas anteriormente para el estudio, a continuación, se elige el número de neuronas con el que se va a trabajar; en este caso 6, al ser bidimensional la representación será cuadrada de 6 por 6 obteniendo finalmente 36 neuronas. Observar Figura 3.62.

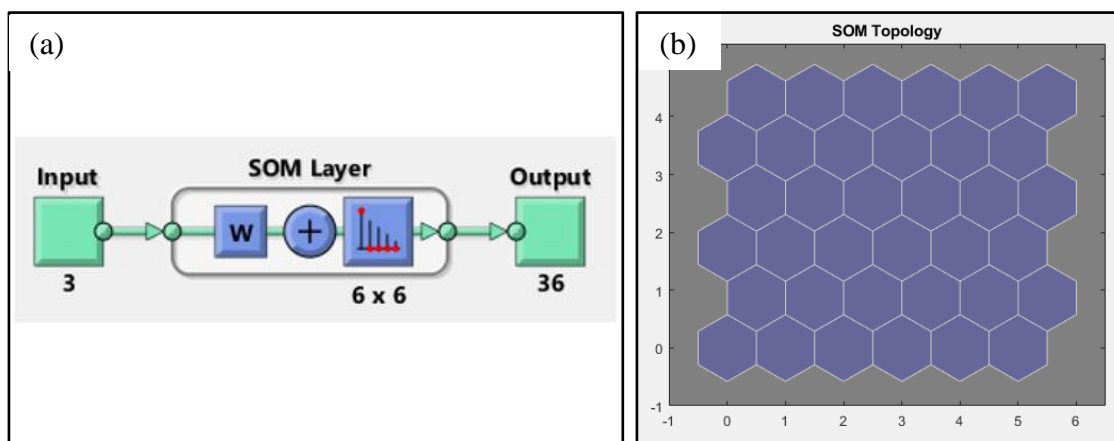


Figura 3.62 (a) Diseño de la red SOM. (b) Topología de las neuronas SOM.

Después de haber obtenido la matriz de neuronas se adquiere la distancia de pesos que existe entre neuronas; Al observar la Figura 3.63, se evidencia la codificación de colores, donde, los hexágonos azules representan las neuronas, las líneas rojas conectan las neuronas vecinas, los colores en las regiones que contienen las líneas rojas indican las distancias entre las neuronas, los colores más oscuros representan distancias más grandes, los colores más claros representan distancias más pequeñas.

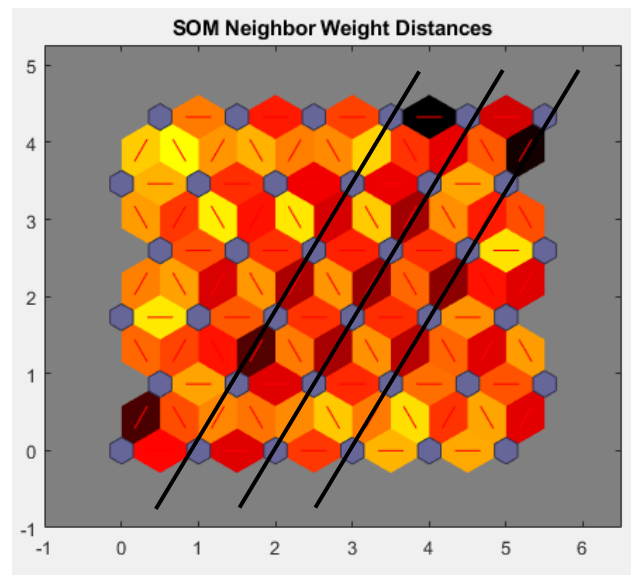


Figura 3.63 Distancia de pesos SOM.

Otro resultado importante, son las distancias asociados a las neuronas, ver Figura 3.64, este resultado es mejor si los datos se distribuyen bastante uniformemente a través de las neuronas; como se observa, los datos asignados a cada neurona, en su mayoría se distribuyen de forma equitativa. No obstante, la división que producen las neuronas de colores oscuros son los que permiten identificar regiones distintas, es así que en la Figura 3.63 se puede evidenciar 3 franjas oscuras, dando como resultado la identificación de 4 zonas.

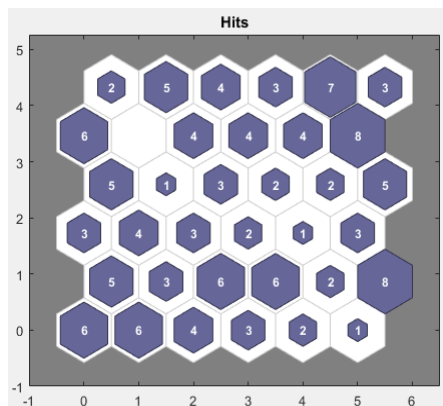


Figura 3.64 Datos asociados a las neuronas.

Por último, La ubicación de las neuronas en los datos característicos, es el resultado en el que se evidencia que la distribución de las neuronas es uniforme en todos los datos, ver Figura 3.65.

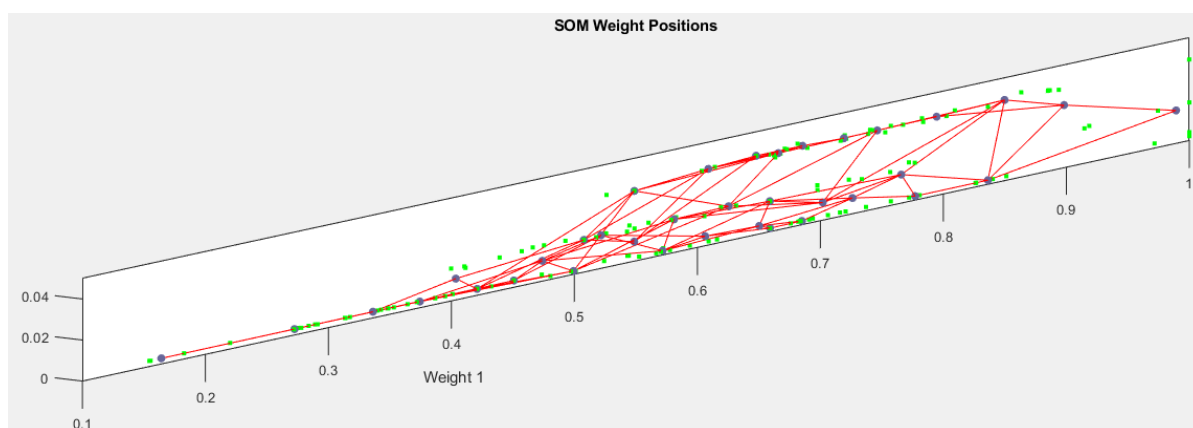


Figura 3.65 Distribución de neuronas.

Es evidente que el mapa de auto organización ayuda a identificar los grupos existentes; Sin embargo, no se puede obtener un resultado que permita evaluar el desempeño de clasificación alcanzado, como es el caso de los métodos supervisados, este método de clasificación faculta el análisis a través de gráficas.

Los resultados mostrados son particularmente de la información del movimiento de Abducción, No obstante, este análisis se puede realizar para los demás movimientos con la finalidad de comprobar el comportamiento en dichos movimientos.

3.4 Prueba de Funcionamiento OFF-LINE

Para la prueba de funcionamiento Off-line, se realiza la adquisición de las muestras con un usuario diferente al que se le utilizó para realizar el entrenamiento de la red; esto, con la finalidad de demostrar que el sistema está funcionando correctamente, independientemente del usuario que se utilice. Se lleva a la práctica la adquisición de señales con una muestra por movimiento.

En el movimiento de Abducción se adquiere una muestra a 10 grados con el nuevo usuario como se observa en la Figura 3.66, esta muestra, se guarda en una nueva base de datos del mismo tamaño de la base de datos de entrenamiento.



Figura 3.66 Prueba de adquisición en movimiento Abducción 10 grados.

Los datos de entrenamiento de la red neuronal son de 136 elementos, 34 por cada movimiento de estudio; por este motivo, los datos de prueba off-line deben ser igualmente de 136 elementos, pero en este caso todos los datos corresponderán solo al movimiento de Abducción en 10 grados. Los resultados alcanzados con esta prueba se pueden evidenciar a través de la matriz de confusión; como se observa en la Figura 3.67, donde se comprueba efectivamente que los 136 elementos que corresponden a la matriz de prueba offline son acertados al 100 % por la Red Neuronal del sistema.

Output Class	1	2	3	4		
1	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	5 GRADOS
2	0 0.0%	136 100%	0 0.0%	0 0.0%	100% 0.0%	10 GRADOS
3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	15 GRADOS
4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	20 GRADOS
	NaN% NaN%	100% 0.0%	NaN% NaN%	NaN% NaN%	100% 0.0%	DESEMPEÑO
	1	2	3	4		
	Target Class					

Figura 3.67 Resultados de prueba offline en movimiento Abducción a 10 grados.

En el movimiento de Aducción se adquiere una muestra a 30 grados con el nuevo usuario como se observa en la Figura 3.68; de la misma forma que el movimiento anterior, se guarda en una nueva base de datos del mismo tamaño de la base de datos de entrenamiento.



Figura 3.68 Prueba de adquisición en movimiento Aducción 30 grados.

En esta ocasión, los datos de entrenamiento de la red neuronal para el movimiento de Abducción son de 136 elementos; como para este movimiento los ángulos de estudio son 3, se agrega un ángulo que representa 0 grados; esto, con el objetivo de tener los datos de entrada del mismo tamaño; por este motivo, los datos de prueba off-line son igualmente de 136 elementos, pero en este caso todos los datos corresponden solo al movimiento de Aducción en 30 grados. Los resultados alcanzados con esta prueba se pueden comprobar a través de la matriz de confusión; como se observa en la Figura 3.69, donde se confirma efectivamente que los 136 elementos de la matriz de prueba off-line son acertados al 100 % por la Red Neuronal del sistema, ya que todos los datos se ubican en el movimiento que corresponden.

PRUEBA OFFLINE ADUCCIÓN A 30 GRADOS						
1	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	0 GRADOS
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	10 GRADOS
3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	20 GRADOS
4	0 0.0%	0 0.0%	0 0.0%	136 100%	100% 0.0%	30 GRADOS
	NaN% NaN%	NaN% NaN%	NaN% NaN%	100% 0.0%	100% 0.0%	DESEMPEÑO
	1	2	3	4		
	Target Class					

Figura 3.69 Resultados de prueba offline en movimiento Aducción a 30 grados.

En el movimiento de Flexión se adquiere una muestra a 60 grados con el nuevo usuario como se observa en la Figura 3.70; de la misma forma que en los movimientos anteriores, se guarda en una nueva base de datos del mismo tamaño de la base de datos de entrenamiento.



Figura 3.70 Prueba de adquisición en movimiento Flexión 60 grados.

A continuación, los datos de entrenamiento de la red neuronal en el movimiento de Flexión son de 136 elementos, 34 para la posición de 30, 45, 60 y 80 grados; por este motivo, los datos que conforman la matriz de la prueba de funcionamiento off-line deben ser igualmente de 136 elementos, pero en este caso todos los datos corresponden solo al movimiento de Flexión en 60 grados.

Los resultados alcanzados con esta prueba se pueden comprobar a través de la matriz de confusión; como se observa en la Figura 3.71, donde se verifica efectivamente que los 136 elementos que corresponden a la matriz de prueba off-line son acertados al 100 % por la Red Neuronal del sistema.

PRUEBA OFFLINE FLEXIÓN A 60 GRADOS						
1	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	30 GRADOS
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	45 GRADOS
3	0 0.0%	0 0.0%	136 100%	0 0.0%	100% 0.0%	60 GRADOS
4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	80 GRADOS
	NaN% NaN%	NaN% NaN%	100% 0.0%	NaN% NaN%	100% 0.0%	DESEMPEÑO
	1	2	3	4		
	Target Class					

Figura 3.71 Resultados de prueba offline en movimiento Flexión a 60 grados.

Por último, en el movimiento de Extensión se adquiere una muestra a 30 grados con el nuevo usuario como se observa en la Figura 3.72; de la misma forma que en los movimientos anteriores, se guarda en una nueva base de datos del mismo tamaño de la base de datos de entrenamiento.

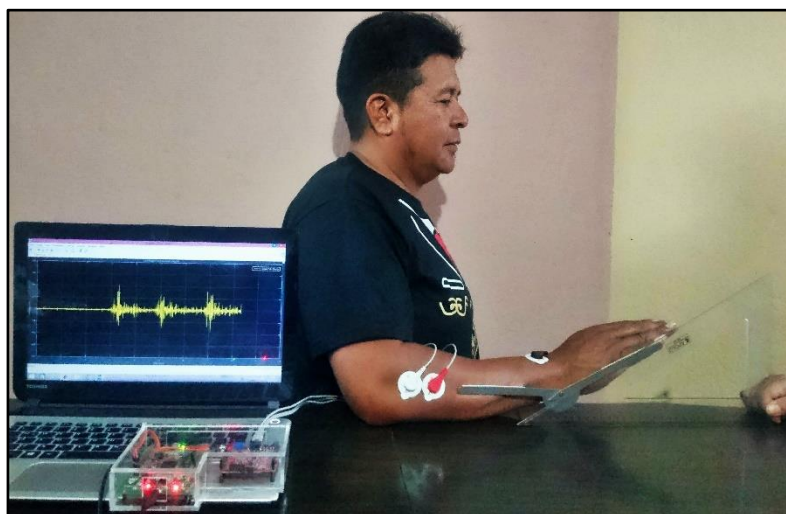


Figura 3.72 Prueba de adquisición en movimiento Extensión 30 grados.

Para finalizar, los datos de entrenamiento de la red neuronal en el movimiento de Extensión son de 136 elementos, 34 para la posición de 30, 45, 60 y 70 grados; por este motivo, los datos que conforman la matriz de la prueba de funcionamiento off-line deben ser igualmente de 136 elementos, pero en este caso todos los datos corresponden solo al movimiento de Extensión en 30 grados.

Los resultados conseguidos con esta prueba, se puede comprobar a través de la matriz de confusión; como se observa en la Figura 3.73, donde se verifica efectivamente que los 136 elementos que corresponden a la matriz de prueba off-line son acertados al 100 % por la Red Neuronal del sistema.

PRUEBA OFFLINE EXTENSIÓN A 30 GRADOS						
1	136 100%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	30 GRADOS
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	45 GRADOS
3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	60 GRADOS
4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%	70 GRADOS
	100% 0.0%	NaN% NaN%	NaN% NaN%	NaN% NaN%	100% 0.0%	DESEMPEÑO
	1	2	3	4		
	Target Class					

Figura 3.73 Resultados de prueba offline en movimiento Extensión a 30 grados.

La prueba de funcionamiento off-line permite verificar que la Red Neuronal está trabajando correctamente, al demostrar que los movimientos ejecutados los clasifica efectivamente.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- En la adquisición de las señales EMG, es evidente que al aumentar la posición angular también aumenta la amplitud de la señal, independientemente del movimiento que se esté ejecutando; esto se comprueba a través del valor de la Raíz Media Cuadrática (RMS); para el movimiento de Abducción en 5° su valor RMS es 12.72, en 10° es 19.57, en 15° es 38.68 y en 20° es 83.10; para el movimiento Aducción en 10° su valor RMS es 48.10, en 20° es 65.30 y en 30° es 86.32; para el movimiento de Flexión en 30° su valor RMS es 13.11, en 45° es 13.86, en 60° es 20.96 y en 80° es 82.37; por último, para el movimiento Extensión en 30° su valor RMS es 47.87, en 45° es 57.77, en 60° es 87.91, y en 70° es 90.20; esto quiere decir que, a mayor posición angular existe una mayor contracción muscular, por esta razón, la amplitud incrementa proporcionalmente a la posición.
- La representación gráfica de las características correspondientes a cada movimiento de estudio, ayudan a comprender el tipo de datos que se está obteniendo, ya que permite identificar si las características son linealmente separables y no existe alguna relación entre ellas; además, facilita posteriormente a seleccionar correctamente el algoritmo de clasificación que se desea implementar en el sistema.

- Los valores obtenidos en las 11 características son directamente proporcionales a la posición angular, dicho de otro modo, los datos propios a cada posición incrementan, al aumentar el grado angular; para el movimiento de Abducción entre 5° y 10° hay un incremento del 11.8%, entre el de 10° y 15° un 11% y entre el de 15° y 20° el 30% de incremento en los datos; para el movimiento Aducción hay un crecimiento del 17.54% entre 10° y 20° y el 14.64% entre 20° y 30° ; para el movimiento Flexión hay un incremento del 3.15% entre 30° y 45° , un 37.25% entre 45° y 60° y un 34.53% entre 60° y 80° ; por último, para el movimiento Extensión hay un incremento del 11.12% entre 30° y 45° , un 20.96% entre 45° y 60° y el 11.35% entre 60° y 70° ; esto permite identificar notoriamente la relación de los datos con el movimiento.
- Los clasificadores de aprendizaje supervisado, son bastante óptimos en problemas donde sus características son linealmente separables; El clasificador Red Neuronal de Retropropagación tiene un desempeño de clasificación del 100 %, el clasificador K-NN con distancia métrica del coseno un desempeño del 91.9%, el clasificador K-NN con distancia métrica cúbica alcanza el 94.10%, el clasificador Análisis Discriminante alcanza el 99.30% y el clasificador Máquinas de Soporte Vectorial alcanza un desempeño del 100%; sin embargo, no todos pueden garantizar la exactitud de la estimación del movimiento, es por esto que se realizan varias pruebas con distintos algoritmos con la finalidad de conocer el método más eficiente.

Recomendaciones

- Los procesos médicos en este caso: la selección de los músculos, ubicación de los electrodos y la instrumentación se recomienda a que sean asesores por un especialista en medicina o terapia física médica; debido a que los profesionales cuentan con mucha más experiencia en este campo; tomar en cuenta este aspecto, evitaría que exista algún tipo de errores en los datos obtenidos.
- La ubicación de los electrodos se debe realizar bajo normas y estándares, debido a que esto podría afectar en la adquisición de datos, generando una especie de ruido en las señales; tal es el caso que se debe cumplir una serie de instrucciones al momento de poner en contacto el electrodo con la piel.
- Durante la adquisición de las señales EMG, se recomienda que, una vez iniciado el proceso de adquirir muestras para conformar un base de datos, no se retire los electrodos del músculo hasta terminar de obtener todas las muestras, esto podría ocasionar alteraciones en los datos; Además, de tomar descansos durante la adquisición, para evitar fatigas musculares.
- Se recomienda analizar los resultados de los clasificadores con otras características de las 11 disponibles, con el objetivo de conocer el comportamiento de las demás características en la clasificación.

- Para el entrenamiento de la red neuronal se recomienda tener dos bases de datos distintas por cada movimiento, esto con la finalidad de utilizar una base de datos únicamente para el entrenamiento de la red y la otra para las pruebas y validación de la misma.
- Para tener una mejor idea del comportamiento de clasificación con técnicas de aprendizaje no supervisado se recomienda experimentar con otros métodos de clustering.

BIBLIOGRAFÍA


- [1] L. G. Santos, J. J. A. Beltran, y C. B. Pitarque, «Lesiones deportivas de la muñeca y mano», *Archivos de medicina del deporte: revista de la Federación Española de Medicina del Deporte y de la Confederación Iberoamericana de Medicina del Deporte*, 2014.
- [2] Y. M. Aung y A. Al-Jumaily, «Estimation of Upper Limb Joint Angle Using Surface EMG Signal», *Int. J. Adv. Robot. Syst.*, vol. 10, n.º 10, p. 369, oct. 2013.
- [3] E. C. Casallas, J. D. R. Toro, y I. F. T. Castrillón, «Sistema de entrenamiento virtual para prótesis mioeléctrica transradial a partir del tratamiento de señales bioeléctricas», *Rev. POLITÉCNICA*, vol. 11, n.º 21, pp. 97-106, 2015.
- [4] A. Coras y C. Arturo, «Procesamiento de señales de electromiografía superficial para la detección de movimiento de dos dedos de la mano», *Univ. Ricardo Palma*, 2012.
- [5] P. Moromenacho y J. Luis, «Diseño e implementación de un sistema de adquisición de señales de electromiograma (emg) basado en electrodos secos y su utilización para el control de una mano robótica», oct. 2017.
- [6] A. I. Kapandji, R. Tubiana, y M. Torres Lacomba, *Fisiología articular: esquemas comentados de mecánica humana. Vol. 1, Vol. 1.* Madrid: Editorial Médica Panamericana, 2006.
- [7] S. T. Rios, «Anatomía de la Muñeca y Mano: ATLAS RADIOLÓGICO», 24-may-2012.
- [8] D. Silva-Vaca, E. González-Galván, y R. D. León-Lomelí, «Diseño y desarrollo de herramienta para terapia de muñeca, implementado en Raspberry», *Mem. Congr. Nac. Ing. Bioméd.*, vol. 2, n.º 1, pp. 361-364, ago. 2017.
- [9] C. Medina Gonzalez, M. Benet Rodríguez, y F. Marco Martínez, «El complejo articular de la muñeca: aspectos anatofisiológicos y biomecánicos, características, clasificación y tratamiento de la fractura distal del radio», *MediSur*, vol. 14, n.º 4, pp. 430-446, ago. 2016.
- [10] R. T. Broome, «Técnica quiropráctica de las articulaciones periféricas». Editorial Paidotribo, 2005.
- [11] F. Paulsen y J. Waschke, *Sobotta. Atlas de anatomía humana vol 1: Anatomía general y aparato locomotor*. Elsevier Health Sciences, 2018.
- [12] M. L. Palmer y M. E. Epler, *FUNDAMENTOS DE LAS TÉCNICAS DE EVALUACIÓN MUSCULOESQUELÉTICA (Bicolor)*. Editorial Paidotribo, 2002.
- [13] M. Gil Fernández y J. C. Zuñil Escobar, «Fiabilidad y correlación en la evaluación de la movilidad de rodilla mediante goniómetro e inclinómetro», *Fisioterapia*, vol. 34, n.º 2, pp. 73-78, mar. 2012.
- [14] A. Delis y A. F. Ruiz Olaya, «Métodos Computacionales para el Reconocimiento de Patrones Mioeléctricos.», *Rev. IngeUAN*, vol. 3, pp. 42-59, oct. 2012.
- [15] E. Campbell, A. Phinyomark, y E. Scheme, «Feature Extraction and Selection for Pain Recognition Using Peripheral Physiological Signals», *Front. Neurosci.*, vol. 13, p. 437, 2019.
- [16] H.-J. Yoo, H.-J. Park, y B. Lee, «Myoelectric Signal Classification of Targeted Muscles Using Dictionary Learning», *Sensors*, vol. 19, n.º 10, may 2019.
- [17] C. Spiewak, M. R. Islam, M. Assad-Uz-Zaman, y M. Rahman, «A Comprehensive Study on EMG Feature Extraction and Classifiers», *Open Access J. Biomed. Eng. Its Appl.*, vol. 1, feb. 2018.
- [18] F. Izaurieta y C. Saavedra, «Redes Neuronales Artificiales», jun. 2019.
- [19] W. L. Martinez, A. R. Martinez, y A. R. Martinez, *Computational Statistics Handbook with MATLAB*. Chapman and Hall/CRC, 2015.
- [20] K. P. Murphy, *Machine learning: a probabilistic perspective*. Cambridge, MA, 2012.

- [21] I. D. Herrera-Granda *et al.*, «Artificial Neural Networks for Bottled Water Demand Forecasting: A Small Business Case Study», en *Advances in Computational Intelligence*, 2019, pp. 362-373.
- [22] Y. LeCun, Y. Bengio, y G. Hinton, «Deep learning», *Nature*, vol. 521, n.º 7553, pp. 436-444, may 2015.
- [23] «Redes Neuronales y Análisis de Datos - PDF». 11-jun-2019.
- [24] «Función de activación - Redes neuronales», *Diego Calvo*, 07-dic-2018. .
- [25] M. A. Cabello, J. M. Martín, y J. V. T. Marhuenda, *Electroterapia práctica + StudentConsult en español: Avances en investigación clínica*. Elsevier España, 2016.
- [26] José María Rodríguez Martín, *ELECTROTERAPIA EN FISIOTERAPIA. : 3ª EDICION : Agapea Libros Urgentes*, Editorial Médica Panamericana S.A. Médica Panamericana S.A., 2014.
- [27] A. Merlo y I. Campanini, «Technical Aspects of Surface Electromyography for Clinicians», *Open Rehabil. J.*, vol. 3, n.º 1, nov. 2010.
- [28] K. E. Tutillo Taipe y N. R. Sinchiguano Chilinguina, «Diseño y construcción de un electroestimulador y electromiograma con comunicación inalámbrica para la detección y tratamiento de enfermedades neuromusculares.», ago. 2015.
- [29] C. Duran y L. Jaimes, «Optimización y clasificación de señales EMG a través de métodos de reconocimiento de patrones», *ITECKNE*, vol. 10, jun. 2013.
- [30] P. Moromenacho y J. Luis, «Diseño e implementación de un sistema de adquisición de señales de electromiograma (emg) basado en electrodos secos y su utilización para el control de una mano robótica», oct. 2017.
- [31] C. R. Romeva, *Diseño concurrente*. Univ. Politèc. de Catalunya, 2002.
- [32] C. Pujota y H. David, «Análisis de características de señales electromiográficas para la determinación de movimientos de una mano», may 2019.
- [33] D. M. Ballesteros y A. E. Gaona, «Detección de actividad muscular en registros EMG superficiales en aplicaciones de compresión de datos», en *IV Latin American Congress on Biomedical Engineering 2007, Bioengineering Solutions for Latin America Health*, 2008, pp. 25-29.
- [34] J. A. G. Pinzon y L. E. Mendoza, «Adquisición y procesamiento de señales emg para controlar movimiento de un brazo hidráulico», *Mundo FESC*, vol. 4, n.º 7, pp. 49-60, jun. 2014.
- [35] E. J. Rechy-Ramirez y H. Hu, «Stages for Developing Control Systems using EMG and EEG Signals: A survey», 2011.
- [36] D. K. Roy y H. M. Pandey, «A New Clustering Method Using an Augmentation to the Self Organizing Maps», en *2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, 2018, pp. 739-742.
- [37] H. Mo, B. Xu, W. Ouyang, y J. Wang, «Color segmentation of multi-colored fabrics using self-organizing-map based clustering algorithm», *Text. Res. J.*, vol. 87, n.º 3, pp. 369-380, feb. 2017.
- [38] Poojitha V, M. Bhadauria, S. Jain, y A. Garg, «A collocation of IRIS flower using neural network clustering tool in MATLAB», en *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, 2016, pp. 53-58.

ANEXOS

ANEXO A: Supervisión y Certificación Médica

ANEXO A.1: Oficio de especialista asignado



UNIVERSIDAD TÉCNICA DEL NORTE
UNIVERSIDAD ACREDITADA RESOLUCIÓN Nro. 001 – 073 – CEAACES – 2013 – 13
Ibarra – Ecuador

CARRERA TERAPIA FÍSICA MÉDICA

Ibarra, 27 de enero del 2019.
Oficio 259-TFM-UTN

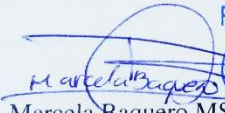
Ingeniero
Diego Ortiz
COORDINADOR CIME-FICA-UTN
Presente

De mi consideración:

En atención al oficio UTN-FICA-CIME-2019-008-M, comedidamente comunico a usted que se autoriza al Licenciado Jorge Luis Zambrano – Técnico Docente de la Carrera de Terapia Física Médica, para que asesore la Tesis Titulada “ESTIMACION DE LA POSICION DE LA MUÑECA A TRAVES DE SEÑALES ELECTROMIOGRAFICAS”, y se solicita al señor Angelo Fernando Flores Taramuel, coordinar con el mencionado funcionario el horario para proceder con el asesoramiento.

Particular que comunico para los fines pertinentes.

Atentamente,
“CIENCIA Y TÉCNICA AL SERVICIO DEL PUEBLO”


Lic. Marcela Baquero MSc.
COORDINADORA CARRERA TERAPIA FÍSICA

FACULTAD CIENCIAS DE LA SALUD
COORDINACIÓN
CARRERA
TERAPIA FÍSICA MÉDICA

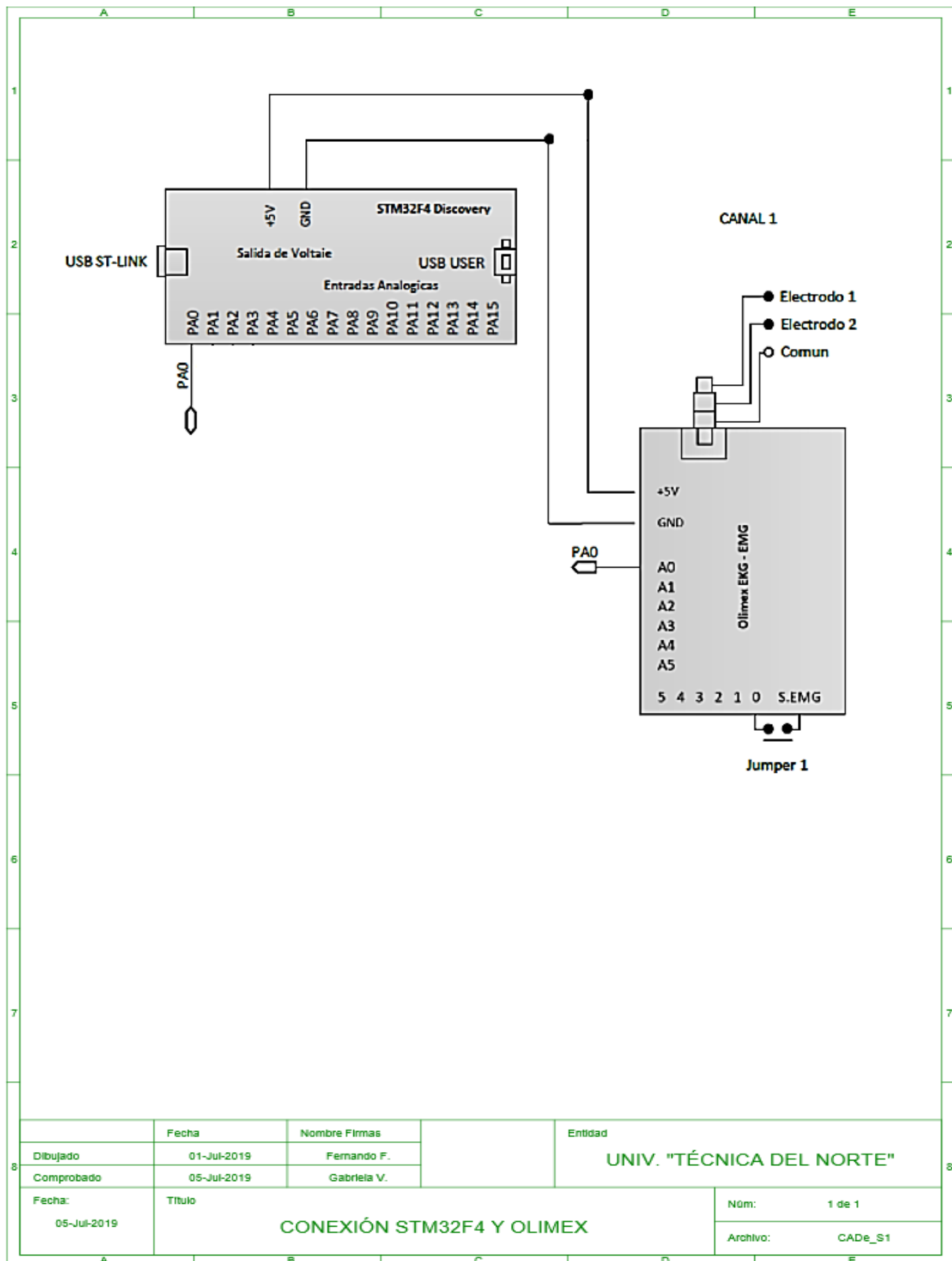
Anabel R.

Copia: Licenciado Jorge Luis Zambrano.

MISIÓN INSTITUCIONAL
“Contribuir al desarrollo educativo, científico, tecnológico, socioeconómico y cultural de la región norte del país.
Formar profesionales comprometidos con el cambio social y con la preservación del medio ambiente”.

Ciudadela Universitaria Barrio El Olivo
Telefax: 2609-420 Ext. 7407 Casilla 199

ANEXO B: Esquema general de conexiones



	Fecha	Nombre Firmas	Entidad
Dibujado	01-Jul-2019	Fernando F.	UNIV. "TÉCNICA DEL NORTE"
Comprobado	05-Jul-2019	Gabriela V.	
Fecha: 05-Jul-2019	Título CONEXIÓN STM32F4 Y OLIMEX		Núm: 1 de 1 Archivo: CADe_S1

ANEXO C: Código implementado en el sistema

ANEXO C.1: Código para Abducción de muñeca

```

% ===== CÓDIGO PARA MOVIMIENTO DE ABDUCCIÓN DE MUÑECA =====
clc; clear all; close all;
% ===== Cargar las señales del movimiento en cada posición =====
load abduccion5grados.mat;           % Base de datos para entrenamiento de la red
load abduccion10grados.mat;
load abduccion15grados.mat;
load abduccion20grados.mat;
load abduccion5gradosprueba.mat;    % Base de datos para validación y prueba de la red
load abduccion10gradosprueba.mat;
load abduccion15gradosprueba.mat;
load abduccion20gradosprueba.mat;

% ===== Asignar la base de datos a una variable de tipo double =====
signal1 = double(AB5G);
signal2 = double(AB10G);
signal3 = double(AB15G);
signal4 = double(AB20G);
signal5 = double(AB5GP);
signal6 = double(AB10GP);
signal7 = double(AB15GP);
signal8 = double(AB20GP);

Fs = 1000;           %Frecuencia de muestreo
T = 1/Fs;           %Periodo de Muestro
L = length(signal1); %Tamaño de la señal
t = (0:L-1)*T;      %Tiempo de la señal

% ===== Filtrado de Señal usando función Filter_signal=====
[fil1,fil2,fil3,fil4] = Filter_signal(signal1,signal2,signal3,signal4,Fs);
[fil5,fil6,fil7,fil8] = Filter_signal(signal5,signal6,signal7,signal8,Fs);

% ===== Rectificado de la señal =====
signalrec1= abs(fil1); signalrec5= abs(fil5); % Se aplica valor absoluto
signalrec2= abs(fil2); signalrec6= abs(fil6);
signalrec3= abs(fil3); signalrec7= abs(fil7);
signalrec4= abs(fil4); signalrec8= abs(fil8);

% ===== Detección de actividad usando función Activ =====
for i = 1:34
    sigrec1 = signalrec1(:,i);
    sigrec2 = signalrec2(:,i);
    sigrec3 = signalrec3(:,i);
    sigrec4 = signalrec4(:,i);

    [ACTSNL1] = Activ(sigrec1);
    [ACTSNL2] = Activ(sigrec2);
    [ACTSNL3] = Activ(sigrec3);
    [ACTSNL4] = Activ(sigrec4);
end

% ===== CARACTERIZACIÓN DE LA SEÑAL =====

% *** Para la caracterización se utiliza las funciones correspondientes a *
% *** cada característica, todas las funciones inician con el nombre Feat **
% *** Se realiza para la base de datos de entrenamiento y de validación ***
% *** y en algunas características se usa el número de muestras (34) y la
% *** frecuencia de muestreo (Fs)

```

```

% ===== Característica Integral de la señal EMG (IEMG) =====
[SNLIEMG1] = Featiemg(signalrec1); [SNLIEMG5] = Featiemg(signalrec5);
[SNLIEMG2] = Featiemg(signalrec2); [SNLIEMG6] = Featiemg(signalrec6);
[SNLIEMG3] = Featiemg(signalrec3); [SNLIEMG7] = Featiemg(signalrec7);
[SNLIEMG4] = Featiemg(signalrec4); [SNLIEMG8] = Featiemg(signalrec8);

% ===== Característica Simple cuadrado integral (SSI) =====
[SNLSSI1] = Featssi(signalrec1); [SNLSSI5] = Featssi(signalrec5);
[SNLSSI2] = Featssi(signalrec2); [SNLSSI6] = Featssi(signalrec6);
[SNLSSI3] = Featssi(signalrec3); [SNLSSI7] = Featssi(signalrec7);
[SNLSSI4] = Featssi(signalrec4); [SNLSSI8] = Featssi(signalrec8);

% ===== Característica Valor medio absoluto (MAV) =====
[SNLMAV1] = Featmav(signalrec1,34); [SNLMAV5] = Featmav(signalrec5,34);
[SNLMAV2] = Featmav(signalrec2,34); [SNLMAV6] = Featmav(signalrec6,34);
[SNLMAV3] = Featmav(signalrec3,34); [SNLMAV7] = Featmav(signalrec7,34);
[SNLMAV4] = Featmav(signalrec4,34); [SNLMAV8] = Featmav(signalrec8,34);

% ===== Característica raíz media cuadrática (RMS) =====
[SNLRootMS1] = Featrootms(signalrec1,34); [SNLRootMS5] = Featrootms(signalrec5,34);
[SNLRootMS2] = Featrootms(signalrec2,34); [SNLRootMS6] = Featrootms(signalrec6,34);
[SNLRootMS3] = Featrootms(signalrec3,34); [SNLRootMS7] = Featrootms(signalrec7,34);
[SNLRootMS4] = Featrootms(signalrec4,34); [SNLRootMS8] = Featrootms(signalrec8,34);

% ===== Característica Varianza (VAR) =====
[SNLVAR1] = Featvar(signalrec1,34); [SNLVAR5] = Featvar(signalrec5,34);
[SNLVAR2] = Featvar(signalrec2,34); [SNLVAR6] = Featvar(signalrec6,34);
[SNLVAR3] = Featvar(signalrec3,34); [SNLVAR7] = Featvar(signalrec7,34);
[SNLVAR4] = Featvar(signalrec4,34); [SNLVAR8] = Featvar(signalrec8,34);

% ===== Característica Longitud de Onda (WL) =====
[SNLWL1] = Featwl(signalrec1,34); [SNLWL5] = Featwl(signalrec5,34);
[SNLWL2] = Featwl(signalrec2,34); [SNLWL6] = Featwl(signalrec6,34);
[SNLWL3] = Featwl(signalrec3,34); [SNLWL7] = Featwl(signalrec7,34);
[SNLWL4] = Featwl(signalrec4,34); [SNLWL8] = Featwl(signalrec8,34);

% ===== Característica Transformada de Fourier rápida (FFT) =====
[SNLFFT1] = Feat_fft(signalrec1,Fs,34); [SNLFFT5] = Feat_fft(signalrec5,Fs,34);
[SNLFFT2] = Feat_fft(signalrec2,Fs,34); [SNLFFT6] = Feat_fft(signalrec6,Fs,34);
[SNLFFT3] = Feat_fft(signalrec3,Fs,34); [SNLFFT7] = Feat_fft(signalrec7,Fs,34);
[SNLFFT4] = Feat_fft(signalrec4,Fs,34); [SNLFFT8] = Feat_fft(signalrec8,Fs,34);

% ===== Característica Energía (E) =====
[SNLENERG1] = Feat_energ(signalrec1,Fs); [SNLENERG5] = Feat_energ(signalrec5,Fs);
[SNLENERG2] = Feat_energ(signalrec2,Fs); [SNLENERG6] = Feat_energ(signalrec6,Fs);
[SNLENERG3] = Feat_energ(signalrec3,Fs); [SNLENERG7] = Feat_energ(signalrec7,Fs);
[SNLENERG4] = Feat_energ(signalrec4,Fs); [SNLENERG8] = Feat_energ(signalrec8,Fs);

% ===== Característica Densidad de espectro de potencia (PSD) =====
[SNLPSD1] = Feat_PSD(signalrec1,Fs,34); [SNLPSD5] = Feat_PSD(signalrec5,Fs,34);
[SNLPSD2] = Feat_PSD(signalrec2,Fs,34); [SNLPSD6] = Feat_PSD(signalrec6,Fs,34);
[SNLPSD3] = Feat_PSD(signalrec3,Fs,34); [SNLPSD7] = Feat_PSD(signalrec7,Fs,34);
[SNLPSD4] = Feat_PSD(signalrec4,Fs,34); [SNLPSD8] = Feat_PSD(signalrec8,Fs,34);

% ===== Característica Amplitud de Willison (WAMP) =====
[SNLWAMP1] = Featwamp(signalrec1,34); [SNLWAMP5] = Featwamp(signalrec5,34);
[SNLWAMP2] = Featwamp(signalrec2,34); [SNLWAMP6] = Featwamp(signalrec6,34);
[SNLWAMP3] = Featwamp(signalrec3,34); [SNLWAMP7] = Featwamp(signalrec7,34);
[SNLWAMP4] = Featwamp(signalrec4,34); [SNLWAMP8] = Featwamp(signalrec8,34);

% ===== Característica Cambio de signo de la señal (SSC) =====
[SNLSSC1] = Featssc(signalrec1,34); [SNLSSC5] = Featssc(signalrec5,34);
[SNLSSC2] = Featssc(signalrec2,34); [SNLSSC6] = Featssc(signalrec6,34);

```



```

    0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0
0 1 0; 0 0 1 0; 0 0 1 0; ...
    0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0
0 1 0; ...
    0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1; 0 0 0 1; 0 0 0 1; ...
    0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1; 0 0 0 1; 0 0 0 1; ...
    0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1];

% ===== EXTRAER CARACTERÍSTICAS PRINCIPALES DE MATRIZ GENERAL =====
xc = Mat_general(:,2); %2 Simple Cuadrado Integral
yc = Mat_general(:,4); %4 Raíz media cuadrática
zc = Mat_general(:,6); %6 Longitud de forma de onda
% ===== EXTRAER CARACTERÍSTICAS PRINCIPALES DE LA MATRIZ DE PRUEBA =====
xcprueba = Mat_gprueba(:,2); %2 Simple Cuadrado Integral
ycprueba = Mat_gprueba(:,4); %4 Raíz media cuadrática
zcprueba = Mat_gprueba(:,6); %6 Longitud de forma de onda

% =====
% ===== GRAFICAMOS CARACTERÍSTICAS =====
S = repmat([50,60,70,80],34,1); % Asigna tamaño acuerdo al movimiento
C = repmat([1,2,3,4],34,1); % Asigna color acuerdo al movimiento
s = S(:);
c = C(:);
figure (1)
scatter3(xc,yc,zc,s,c, 'filled') % Gráfica en 3 dimensiones
figure (2)
gscatter(xc,yc,c) % Gráfica en 2 dimensiones

% =====
% ===== CONFORMAR MATRICES PARA LA RED NEURONAL =====
Mat_trescarac = [xc,yc,zc]; % Para el entrenamiento de la red
Mat_tresprueba = [xcprueba,ycprueba,zcprueba]; % Para la validación de la red

% -----
% ----- RED NEURONAL BACKPROPAGATION-----

% ===== Transponer matrices para ingresar al entrenamiento de la red =====
x = Mat_trescarac'; % Matriz de entrada de 136 x 3
xp = Mat_tresprueba'; % Matriz de validación de 136 x 3
ts = Mat_salida'; % Matriz de salida de 136 x 4
% Escoger la Función de entrenamiento
trainFcn = 'trainscg'; % Gradiente conjugado escalado backpropagation.

% Crear la red de reconocimiento de patrones
hiddenLayerSize = 10; % Establecer número de neuronas
net = patternnet(hiddenLayerSize, trainFcn); % Se crea la red

% Elija las funciones de entrada y salida de pre / post procesamiento
net.input.processFcns = {'removeconstantrows','mapminmax'};

% Escoger la división de los datos
net.divideFcn = 'dividetrain'; % Divide la información solo para el entrenamiento

% Escoger la función de desempeño
net.performFcn = 'crossentropy'; % Cross-Entropy

% Escoger las gráficas de las funciones
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
'plotconfusion', 'plotroc'};

```

```

% Entrenamiento de la red neuronal
[net,tr] = train(net,x,ts); % Con la matriz de entrada y salida

% Prueba y validación de la red neuronal
y = net(xp); % Con la matriz de prueba
e = gsubtract(ts,y);
performance = perform(net,ts,y) % Calcula el desempeño
tind = vec2ind(ts);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind); % Calcula el error

% Recalcula el entrenamiento, la validación y la prueba
trainTargets = ts .* tr.trainMask{1};
valTargets = ts .* tr.valMask{1};
testTargets = ts .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% Observar la red neuronal
view(net)

% Gráficas de interés
figure, plotperform(tr)
figure, plottrainstate(tr)
figure, ploterrhist(e)
figure, plotconfusion(ts,y)
figure, plotroc(ts,y)

```

ANEXO C.2: Código para Aducción de muñeca

```

% ===== CÓDIGO PARA MOVIMIENTO DE ADUCCIÓN DE MUÑECA =====
clc; clear all; close all;
% ===== Cargar las señales del movimiento en cada posición =====
load aduccion10grados.mat; % Base de datos para entrenamiento de la red
load aduccion20grados.mat;
load aduccion30grados.mat;
load aduccion0grados.mat;
load aduccion0gradosprueba.mat; % Base de datos para validación y prueba de la red
load aduccion10gradosprueba.mat;
load aduccion20gradosprueba.mat;
load aduccion30gradosprueba.mat;

% ===== Asignar la base de datos a una variable de tipo double =====
signal1 = double(AD10G);
signal2 = double(AD20G);
signal3 = double(AD30G);
signal4 = double(AD0G);
signal5 = double(AD10GP);
signal6 = double(AD20GP);
signal7 = double(AD30GP);
signal8 = double(AD0GP);

Fs = 1000; %Frecuencia de muestreo
T = 1/Fs; %Periodo de Muestro
L = length(signal1); %Tamaño de la señal
t = (0:L-1)*T; %Tiempo de la señal

```

```

% ===== Filtrado de Señal usando función Filter_signal=====
[fil1,fil2,fil3,fil4] = Filter_signal(signal1,signal2,signal3,signal4,Fs);
[fil5,fil6,fil7,fil8] = Filter_signal(signal5,signal6,signal7,signal8,Fs);

% ===== Rectificado de la señal =====
signalrec1= abs(fil1); signalrec5= abs(fil5); % Se aplica valor absoluto
signalrec2= abs(fil2); signalrec6= abs(fil6);
signalrec3= abs(fil3); signalrec7= abs(fil7);
signalrec4= abs(fil4); signalrec8= abs(fil8);

% ===== Detección de actividad usando función Activ =====
for i = 1:34
    sigrec1 = signalrec1(:,i);
    sigrec2 = signalrec2(:,i);
    sigrec3 = signalrec3(:,i);
    sigrec4 = signalrec4(:,i);

    [ACTSNL1] = Activ(sigrec1);
    [ACTSNL2] = Activ(sigrec2);
    [ACTSNL3] = Activ(sigrec3);
    [ACTSNL4] = Activ(sigrec4);
end

% ===== CARACTERIZACIÓN DE LA SEÑAL =====

% *** Para la caracterización se utiliza las funciones correspondientes a *
% *** cada característica, todas las funciones inician con el nombre Feat **
% *** Se realiza para la base de datos de entrenamiento y de validación ***
% *** y en algunas características se usa el número de muestras (34) y la
% *** frecuencia de muestreo (Fs)

% ===== Característica Integral de la señal EMG (IEMG) =====
[SNLIEMG1] = Featiemg(signalrec1); [SNLIEMG5] = Featiemg(signalrec5);
[SNLIEMG2] = Featiemg(signalrec2); [SNLIEMG6] = Featiemg(signalrec6);
[SNLIEMG3] = Featiemg(signalrec3); [SNLIEMG7] = Featiemg(signalrec7);
[SNLIEMG4] = Featiemg(signalrec4); [SNLIEMG8] = Featiemg(signalrec8);

% ===== Característica Simple cuadrado integral (SSI) =====
[SNLSSI1] = Featssi(signalrec1); [SNLSSI5] = Featssi(signalrec5);
[SNLSSI2] = Featssi(signalrec2); [SNLSSI6] = Featssi(signalrec6);
[SNLSSI3] = Featssi(signalrec3); [SNLSSI7] = Featssi(signalrec7);
[SNLSSI4] = Featssi(signalrec4); [SNLSSI8] = Featssi(signalrec8);

% ===== Característica Valor medio absoluto (MAV) =====
[SNLMAV1] = Featmav(signalrec1,34); [SNLMAV5] = Featmav(signalrec5,34);
[SNLMAV2] = Featmav(signalrec2,34); [SNLMAV6] = Featmav(signalrec6,34);
[SNLMAV3] = Featmav(signalrec3,34); [SNLMAV7] = Featmav(signalrec7,34);
[SNLMAV4] = Featmav(signalrec4,34); [SNLMAV8] = Featmav(signalrec8,34);

% ===== Característica raíz media cuadrática (RMS) =====
[SNLRootMS1] = Featrootms(signalrec1,34); [SNLRootMS5] = Featrootms(signalrec5,34);
[SNLRootMS2] = Featrootms(signalrec2,34); [SNLRootMS6] = Featrootms(signalrec6,34);
[SNLRootMS3] = Featrootms(signalrec3,34); [SNLRootMS7] = Featrootms(signalrec7,34);
[SNLRootMS4] = Featrootms(signalrec4,34); [SNLRootMS8] = Featrootms(signalrec8,34);

% ===== Característica Varianza (VAR) =====
[SNLVAR1] = Featvar(signalrec1,34); [SNLVAR5] = Featvar(signalrec5,34);
[SNLVAR2] = Featvar(signalrec2,34); [SNLVAR6] = Featvar(signalrec6,34);
[SNLVAR3] = Featvar(signalrec3,34); [SNLVAR7] = Featvar(signalrec7,34);
[SNLVAR4] = Featvar(signalrec4,34); [SNLVAR8] = Featvar(signalrec8,34);

% ===== Característica Longitud de Onda (WL) =====
[SNLWL1] = Featwl(signalrec1,34); [SNLWL5] = Featwl(signalrec5,34);

```

```

[SNLWL2] = Featwl(signalrec2,34); [SNLWL6] = Featwl(signalrec6,34);
[SNLWL3] = Featwl(signalrec3,34); [SNLWL7] = Featwl(signalrec7,34);
[SNLWL4] = Featwl(signalrec4,34); [SNLWL8] = Featwl(signalrec8,34);

% ===== Característica Transformada de Fourier rapida (FFT) =====
[SNLFFT1] = Feat_fft(signalrec1,Fs,34); [SNLFFT5] = Feat_fft(signalrec5,Fs,34);
[SNLFFT2] = Feat_fft(signalrec2,Fs,34); [SNLFFT6] = Feat_fft(signalrec6,Fs,34);
[SNLFFT3] = Feat_fft(signalrec3,Fs,34); [SNLFFT7] = Feat_fft(signalrec7,Fs,34);
[SNLFFT4] = Feat_fft(signalrec4,Fs,34); [SNLFFT8] = Feat_fft(signalrec8,Fs,34);

% ===== Característica Energia (E) =====
[SNLENERG1] = Feat_energy(signalrec1,Fs); [SNLENERG5] = Feat_energy(signalrec5,Fs);
[SNLENERG2] = Feat_energy(signalrec2,Fs); [SNLENERG6] = Feat_energy(signalrec6,Fs);
[SNLENERG3] = Feat_energy(signalrec3,Fs); [SNLENERG7] = Feat_energy(signalrec7,Fs);
[SNLENERG4] = Feat_energy(signalrec4,Fs); [SNLENERG8] = Feat_energy(signalrec8,Fs);

% ===== Característica Densidad de espectro de potencia (PSD) =====
[SNLPSD1] = Feat_PSD(signalrec1,Fs,34); [SNLPSD5] = Feat_PSD(signalrec5,Fs,34);
[SNLPSD2] = Feat_PSD(signalrec2,Fs,34); [SNLPSD6] = Feat_PSD(signalrec6,Fs,34);
[SNLPSD3] = Feat_PSD(signalrec3,Fs,34); [SNLPSD7] = Feat_PSD(signalrec7,Fs,34);
[SNLPSD4] = Feat_PSD(signalrec4,Fs,34); [SNLPSD8] = Feat_PSD(signalrec8,Fs,34);

%% ===== Característica Amplitud de Willison (WAMP) =====
[SNLWAMP1] = Featwamp(signalrec1,34); [SNLWAMP5] = Featwamp(signalrec5,34);
[SNLWAMP2] = Featwamp(signalrec2,34); [SNLWAMP6] = Featwamp(signalrec6,34);
[SNLWAMP3] = Featwamp(signalrec3,34); [SNLWAMP7] = Featwamp(signalrec7,34);
[SNLWAMP4] = Featwamp(signalrec4,34); [SNLWAMP8] = Featwamp(signalrec8,34);

% ===== Característica Cambio de signo de la señal (SSC) =====
[SNLSSC1] = Featssc(signalrec1,34); [SNLSSC5] = Featssc(signalrec5,34);
[SNLSSC2] = Featssc(signalrec2,34); [SNLSSC6] = Featssc(signalrec6,34);
[SNLSSC3] = Featssc(signalrec3,34); [SNLSSC7] = Featssc(signalrec7,34);
[SNLSSC4] = Featssc(signalrec4,34); [SNLSSC8] = Featssc(signalrec8,34);

% ===== FORMAR MATRIZ CON CARACTERÍSTICAS PARA CADA ÁNGULO =====

matriz_ch1 = [SNLIEMG1; SNLSSI1; SNLMAV1; SNLRootMS1; SNLVAR1; SNLWL1; SNLFFT1;
SNLENERG1; SNLPSD1; SNLSSC1; SNLWAMP1];
matriz_ch2 = [SNLIEMG2; SNLSSI2; SNLMAV2; SNLRootMS2; SNLVAR2; SNLWL2; SNLFFT2;
SNLENERG2; SNLPSD2; SNLSSC2; SNLWAMP2];
matriz_ch3 = [SNLIEMG3; SNLSSI3; SNLMAV3; SNLRootMS3; SNLVAR3; SNLWL3; SNLFFT3;
SNLENERG3; SNLPSD3; SNLSSC3; SNLWAMP3];
matriz_ch4 = [SNLIEMG4; SNLSSI4; SNLMAV4; SNLRootMS4; SNLVAR4; SNLWL4; SNLFFT4;
SNLENERG4; SNLPSD4; SNLSSC4; SNLWAMP4];

matriz_ch5 = [SNLIEMG5; SNLSSI5; SNLMAV5; SNLRootMS5; SNLVAR5; SNLWL5; SNLFFT5;
SNLENERG5; SNLPSD5; SNLSSC5; SNLWAMP5];
matriz_ch6 = [SNLIEMG6; SNLSSI6; SNLMAV6; SNLRootMS6; SNLVAR6; SNLWL6; SNLFFT6;
SNLENERG6; SNLPSD6; SNLSSC6; SNLWAMP6];
matriz_ch7 = [SNLIEMG7; SNLSSI7; SNLMAV7; SNLRootMS7; SNLVAR7; SNLWL7; SNLFFT7;
SNLENERG7; SNLPSD7; SNLSSC7; SNLWAMP7];
matriz_ch8 = [SNLIEMG8; SNLSSI8; SNLMAV8; SNLRootMS8; SNLVAR8; SNLWL8; SNLFFT8;
SNLENERG8; SNLPSD8; SNLSSC8; SNLWAMP8];

% ===== Normalización de matrices conformadas =====
Norm_MAT1 = matriz_ch1/max(abs(matriz_ch1(:))); Norm_MAT5 =
matriz_ch5/max(abs(matriz_ch5(:)));
Norm_MAT2 = matriz_ch2/max(abs(matriz_ch2(:))); Norm_MAT6 =
matriz_ch6/max(abs(matriz_ch6(:)));
Norm_MAT3 = matriz_ch3/max(abs(matriz_ch3(:))); Norm_MAT7 =
matriz_ch7/max(abs(matriz_ch7(:)));

```

```

Norm_MAT4 = matriz_ch4/max(abs(matriz_ch4(:))); Norm_MAT8 =
matriz_ch8/max(abs(matriz_ch8(:)));

% ===== Transpuesta de la matriz de características para cada ángulo =====
Caract10g = Norm_MAT1';      Caract10gp = Norm_MAT5';
Caract20g = Norm_MAT2';      Caract20gp = Norm_MAT6';
Caract30g = Norm_MAT3';      Caract30gp = Norm_MAT7';
Caract0g = Norm_MAT4';      Caract0gp = Norm_MAT8';

% =====
% ===== CONFORMAR MATRICES GENERALES =====
Mat_general = [Caract10g; Caract20g; Caract30g; Caract0g]; % MATRIZ GENERAL PARA
ENTRENAMIENTO
Mat_gprueba = [Caract10gp; Caract20gp; Caract30gp; Caract0gp]; % MATRIZ GENERAL PARA
VALIDACIÓN

% MATRIZ GENERAL DE SALIDA EN FORMA BINARIA
Mat_salida = [1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0;
1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; ...
1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1
0 0 0; 1 0 0 0; 1 0 0 0; ...
1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1
0 0 0; ...
0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0
1 0 0; 0 1 0 0; 0 1 0 0; ...
0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0
1 0 0; 0 1 0 0; 0 1 0 0; ...
0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0
1 0 0; ...
0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0
0 1 0; 0 0 1 0; 0 0 1 0; ...
0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0
0 1 0; ...
0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1; 0 0 0 1; 0 0 0 1; ...
0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1; 0 0 0 1; 0 0 0 1; ...
0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1];

% ===== EXTRAER CARACTERÍSTICAS PRINCIPALES DE MATRIZ GENERAL =====
xc = Mat_general(:,2); %2 Simple Cuadrado Integral
yc = Mat_general(:,4); %4 Raíz media cuadrática
zc = Mat_general(:,6); %6 Longitud de forma de onda
% ===== EXTRAER CARACTERÍSTICAS PRINCIPALES DE LA MATRIZ DE PRUEBA =====
xcprueba = Mat_gprueba(:,2); %2 Simple Cuadrado Integral
ycprueba = Mat_gprueba(:,4); %4 Raíz media cuadrática
zcprueba = Mat_gprueba(:,6); %6 Longitud de forma de onda

% =====
% ===== GRAFICAMOS CARACTERÍSTICAS =====
S = repmat([50,60,70,80],34,1); % Asigna tamaño acuerdo al movimiento
C = repmat([1,2,3,4],34,1); % Asigna color acuerdo al movimiento
s = S(:);
c = C(:);
figure (1)
scatter3(xc,yc,zc,s,c, 'filled') % Gráfica en 3 dimensiones
figure (2)
gscatter(xc,yc,c) % Gráfica en 2 dimensiones

% =====

```

```

% ===== CONFORMAR MATRICES PARA LA RED NEURONAL =====
Mat_trescarac = [xc,yc,zc]; % Para el entrenamiento de la red
Mat_tresprueba = [xcprueba,ycprueba,zcprueba]; % Para la validación de la red

% -----
% ----- RED NEURONAL BACKPROPAGATION-----

% ===== Transponer matrices para ingresar al entrenamiento de la red =====
x = Mat_trescarac'; % Matriz de entrada de 136 x 3
xp = Mat_tresprueba'; % Matriz de validación de 136 x 3
ts = Mat_salida'; % Matriz de salida de 136 x 4
% Escoger la Función de entrenamiento
trainFcn = 'trainscg'; % Gradiente conjugado escalado backpropagation.

% Crear la red de reconocimiento de patrones
hiddenLayerSize = 10; % Establecer número de neuronas
net = patternnet(hiddenLayerSize, trainFcn); % Se crea la red

% Elija las funciones de entrada y salida de pre / post procesamiento
net.input.processFcns = {'removeconstantrows','mapminmax'};

% Escoger la division de los datos
net.divideFcn = 'dividetrain'; % Divide la información solo para el entrenamiento

% Escoger la función de desempeño
net.performFcn = 'crossentropy'; % Cross-Entropy

% Escoger las gráficas de las funciones
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotconfusion', 'plotroc'};

% Entrenamiento de la red neuronal
[net,tr] = train(net,x,ts); % Con la matriz de entrada y salida

% Prueba y validación de la red neuronal
y = net(xp); % Con la matriz de prueba
e = gsubtract(ts,y);
performance = perform(net,ts,y) % Calcula el desempeño
tind = vec2ind(ts);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind); % Calcula el error

% Recalcula el entrenamiento, la validación y la prueba
trainTargets = ts .* tr.trainMask{1};
valTargets = ts .* tr.valMask{1};
testTargets = ts .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% Observar la red neuronal
view(net)

% Gráficas de interes
figure, plotperform(tr)
figure, plottrainstate(tr)
figure, ploterrhist(e)
figure, plotconfusion(ts,y)
figure, plotroc(ts,y)

```

ANEXO C.3: Código para Flexión de muñeca

```

% ===== CÓDIGO PARA MOVIMIENTO DE FLEXIÓN DE MUÑECA =====
clc; clear all; close all;
% ===== Cargar las señales del movimiento en cada posición =====
load flexion30grados.mat; % Base de datos para entrenamiento de la red
load flexion45grados.mat;
load flexion60grados.mat;
load flexion80grados.mat;
load flexion30gradosprueba.mat; % Base de datos para validación y prueba de la red
load flexion45gradosprueba.mat;
load flexion60gradosprueba.mat;
load flexion80gradosprueba.mat;

% ===== Asignar la base de datos a una variable de tipo double =====
signal1 = double(F30G);
signal2 = double(F45G);
signal3 = double(F60G);
signal4 = double(F80G);
signal5 = double(F30GP);
signal6 = double(F45GP);
signal7 = double(F60GP);
signal8 = double(F80GP);

Fs = 1000;           %Frecuencia de muestreo
T = 1/Fs;           %Periodo de Muestro
L = length(signal1); %Tamaño de la señal
t = (0:L-1)*T;      %Tiempo de la señal

% ===== Filtrado de Señal usando función Filter_signal=====
[fil1,fil2,fil3,fil4] = Filter_signal(signal1,signal2,signal3,signal4,Fs);
[fil5,fil6,fil7,fil8] = Filter_signal(signal5,signal6,signal7,signal8,Fs);

% ===== Rectificado de la señal =====
signalrec1= abs(fil1); signalrec5= abs(fil5); % Se aplica valor absoluto
signalrec2= abs(fil2); signalrec6= abs(fil6);
signalrec3= abs(fil3); signalrec7= abs(fil7);
signalrec4= abs(fil4); signalrec8= abs(fil8);

% ===== Detección de actividad usando función Activ =====
for i = 1:34
    sigrec1 = signalrec1(:,i);
    sigrec2 = signalrec2(:,i);
    sigrec3 = signalrec3(:,i);
    sigrec4 = signalrec4(:,i);

    [ACTSNL1] = Activ(sigrec1);
    [ACTSNL2] = Activ(sigrec2);
    [ACTSNL3] = Activ(sigrec3);
    [ACTSNL4] = Activ(sigrec4);
end

% ===== CARACTERIZACIÓN DE LA SEÑAL =====

% *** Para la caracterización se utiliza las funciones correspondientes a *
% *** cada característica, todas las funciones inician con el nombre Feat **
% *** Se realiza para la base de datos de entrenamiento y de validación ***
% *** y en algunas características se usa el número de muestras (34) y la
% *** frecuencia de muestreo (Fs)

% ===== Característica Integral de la señal EMG (IEMG) =====

```

```

[SNLIEMG1] = Featiemg(signalrec1); [SNLIEMG5] = Featiemg(signalrec5);
[SNLIEMG2] = Featiemg(signalrec2); [SNLIEMG6] = Featiemg(signalrec6);
[SNLIEMG3] = Featiemg(signalrec3); [SNLIEMG7] = Featiemg(signalrec7);
[SNLIEMG4] = Featiemg(signalrec4); [SNLIEMG8] = Featiemg(signalrec8);

% ===== Característica Simple cuadrado integral (SSI) =====
[SNLSSI1] = Featssi(signalrec1); [SNLSSI5] = Featssi(signalrec5);
[SNLSSI2] = Featssi(signalrec2); [SNLSSI6] = Featssi(signalrec6);
[SNLSSI3] = Featssi(signalrec3); [SNLSSI7] = Featssi(signalrec7);
[SNLSSI4] = Featssi(signalrec4); [SNLSSI8] = Featssi(signalrec8);

% ===== Característica Valor medio absoluto (MAV) =====
[SNLMAV1] = Featmav(signalrec1,34); [SNLMAV5] = Featmav(signalrec5,34);
[SNLMAV2] = Featmav(signalrec2,34); [SNLMAV6] = Featmav(signalrec6,34);
[SNLMAV3] = Featmav(signalrec3,34); [SNLMAV7] = Featmav(signalrec7,34);
[SNLMAV4] = Featmav(signalrec4,34); [SNLMAV8] = Featmav(signalrec8,34);

% ===== Característica raiz media cuadrática (RMS) =====
[SNLRootMS1] = Featrootms(signalrec1,34); [SNLRootMS5] = Featrootms(signalrec5,34);
[SNLRootMS2] = Featrootms(signalrec2,34); [SNLRootMS6] = Featrootms(signalrec6,34);
[SNLRootMS3] = Featrootms(signalrec3,34); [SNLRootMS7] = Featrootms(signalrec7,34);
[SNLRootMS4] = Featrootms(signalrec4,34); [SNLRootMS8] = Featrootms(signalrec8,34);

% ===== Característica Varianza (VAR) =====
[SNLVAR1] = Featvar(signalrec1,34); [SNLVAR5] = Featvar(signalrec5,34);
[SNLVAR2] = Featvar(signalrec2,34); [SNLVAR6] = Featvar(signalrec6,34);
[SNLVAR3] = Featvar(signalrec3,34); [SNLVAR7] = Featvar(signalrec7,34);
[SNLVAR4] = Featvar(signalrec4,34); [SNLVAR8] = Featvar(signalrec8,34);

% ===== Característica Longitud de Onda (WL) =====
[SNLWL1] = Featwl(signalrec1,34); [SNLWL5] = Featwl(signalrec5,34);
[SNLWL2] = Featwl(signalrec2,34); [SNLWL6] = Featwl(signalrec6,34);
[SNLWL3] = Featwl(signalrec3,34); [SNLWL7] = Featwl(signalrec7,34);
[SNLWL4] = Featwl(signalrec4,34); [SNLWL8] = Featwl(signalrec8,34);

% ===== Característica Transformada de Fourier rapida (FFT) =====
[SNLFFT1] = Feat_fft(signalrec1,Fs,34); [SNLFFT5] = Feat_fft(signalrec5,Fs,34);
[SNLFFT2] = Feat_fft(signalrec2,Fs,34); [SNLFFT6] = Feat_fft(signalrec6,Fs,34);
[SNLFFT3] = Feat_fft(signalrec3,Fs,34); [SNLFFT7] = Feat_fft(signalrec7,Fs,34);
[SNLFFT4] = Feat_fft(signalrec4,Fs,34); [SNLFFT8] = Feat_fft(signalrec8,Fs,34);

% ===== Característica Energía (E) =====
[SNLENERG1] = Feat_energy(signalrec1,Fs); [SNLENERG5] = Feat_energy(signalrec5,Fs);
[SNLENERG2] = Feat_energy(signalrec2,Fs); [SNLENERG6] = Feat_energy(signalrec6,Fs);
[SNLENERG3] = Feat_energy(signalrec3,Fs); [SNLENERG7] = Feat_energy(signalrec7,Fs);
[SNLENERG4] = Feat_energy(signalrec4,Fs); [SNLENERG8] = Feat_energy(signalrec8,Fs);

% ===== Característica Densidad de espectro de potencia (PSD) =====
[SNLPSD1] = Feat_PSD(signalrec1,Fs,34); [SNLPSD5] = Feat_PSD(signalrec5,Fs,34);
[SNLPSD2] = Feat_PSD(signalrec2,Fs,34); [SNLPSD6] = Feat_PSD(signalrec6,Fs,34);
[SNLPSD3] = Feat_PSD(signalrec3,Fs,34); [SNLPSD7] = Feat_PSD(signalrec7,Fs,34);
[SNLPSD4] = Feat_PSD(signalrec4,Fs,34); [SNLPSD8] = Feat_PSD(signalrec8,Fs,34);

%% ===== Característica Amplitud de Willison (WAMP) =====
[SNLWAMP1] = Featwamp(signalrec1,34); [SNLWAMP5] = Featwamp(signalrec5,34);
[SNLWAMP2] = Featwamp(signalrec2,34); [SNLWAMP6] = Featwamp(signalrec6,34);
[SNLWAMP3] = Featwamp(signalrec3,34); [SNLWAMP7] = Featwamp(signalrec7,34);
[SNLWAMP4] = Featwamp(signalrec4,34); [SNLWAMP8] = Featwamp(signalrec8,34);

% ===== Característica Cambio de signo de la señal (SSC) =====
[SNLSSC1] = Featssc(signalrec1,34); [SNLSSC5] = Featssc(signalrec5,34);
[SNLSSC2] = Featssc(signalrec2,34); [SNLSSC6] = Featssc(signalrec6,34);
[SNLSSC3] = Featssc(signalrec3,34); [SNLSSC7] = Featssc(signalrec7,34);

```



```

[SNLSSC4] = Featssc(signalrec4,34); [SNLSSC8] = Featssc(signalrec8,34);

% =====
% ===== FORMAR MATRIZ CON CARACTERÍSTICAS PARA CADA ÁNGULO =====

matriz_ch1 = [SNLIEMG1; SNLSSI1; SNLMAV1; SNLRootMS1; SNLVAR1; SNLWL1; SNLFFT1;
SNLENERG1; SNLPSD1; SNLSSC1; SNLWAMP1];
matriz_ch2 = [SNLIEMG2; SNLSSI2; SNLMAV2; SNLRootMS2; SNLVAR2; SNLWL2; SNLFFT2;
SNLENERG2; SNLPSD2; SNLSSC2; SNLWAMP2];
matriz_ch3 = [SNLIEMG3; SNLSSI3; SNLMAV3; SNLRootMS3; SNLVAR3; SNLWL3; SNLFFT3;
SNLENERG3; SNLPSD3; SNLSSC3; SNLWAMP3];
matriz_ch4 = [SNLIEMG4; SNLSSI4; SNLMAV4; SNLRootMS4; SNLVAR4; SNLWL4; SNLFFT4;
SNLENERG4; SNLPSD4; SNLSSC4; SNLWAMP4];

matriz_ch5 = [SNLIEMG5; SNLSSI5; SNLMAV5; SNLRootMS5; SNLVAR5; SNLWL5; SNLFFT5;
SNLENERG5; SNLPSD5; SNLSSC5; SNLWAMP5];
matriz_ch6 = [SNLIEMG6; SNLSSI6; SNLMAV6; SNLRootMS6; SNLVAR6; SNLWL6; SNLFFT6;
SNLENERG6; SNLPSD6; SNLSSC6; SNLWAMP6];
matriz_ch7 = [SNLIEMG7; SNLSSI7; SNLMAV7; SNLRootMS7; SNLVAR7; SNLWL7; SNLFFT7;
SNLENERG7; SNLPSD7; SNLSSC7; SNLWAMP7];
matriz_ch8 = [SNLIEMG8; SNLSSI8; SNLMAV8; SNLRootMS8; SNLVAR8; SNLWL8; SNLFFT8;
SNLENERG8; SNLPSD8; SNLSSC8; SNLWAMP8];

% ===== Normalización de matrices conformadas =====
Norm_MAT1 = matriz_ch1/max(abs(matriz_ch1(:))); Norm_MAT5 =
matriz_ch5/max(abs(matriz_ch5(:)));
Norm_MAT2 = matriz_ch2/max(abs(matriz_ch2(:))); Norm_MAT6 =
matriz_ch6/max(abs(matriz_ch6(:)));
Norm_MAT3 = matriz_ch3/max(abs(matriz_ch3(:))); Norm_MAT7 =
matriz_ch7/max(abs(matriz_ch7(:)));
Norm_MAT4 = matriz_ch4/max(abs(matriz_ch4(:))); Norm_MAT8 =
matriz_ch8/max(abs(matriz_ch8(:)));

% ===== Transpuesta de la matriz de características para cada ángulo =====
Caract30g = Norm_MAT1'; Caract30gp = Norm_MAT5';
Caract45g = Norm_MAT2'; Caract45gp = Norm_MAT6';
Caract60g = Norm_MAT3'; Caract60gp = Norm_MAT7';
Caract80g = Norm_MAT4'; Caract80gp = Norm_MAT8';

% =====
% ===== CONFORMAR MATRICES GENERALES =====
Mat_general = [Caract30g; Caract45g; Caract60g; Caract80g]; % MATRIZ GENERAL PARA
ENTRENAMIENTO
Mat_gprueba = [Caract30gp; Caract45gp; Caract60gp; Caract80gp]; % MATRIZ GENERAL PARA
VALIDACIÓN

% MATRIZ GENERAL DE SALIDA EN FORMA BINARIA
Mat_salida = [1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0;
1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; ...
1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1
0 0 0; 1 0 0 0; 1 0 0 0; ...
1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1
0 0 0; ...
0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0
1 0 0; 0 1 0 0; 0 1 0 0; ...
0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0
1 0 0; 0 1 0 0; 0 1 0 0; ...
0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0
1 0 0; ...
0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0
0 1 0; 0 0 1 0; 0 0 1 0; ...
0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0
0 1 0; 0 0 1 0; 0 0 1 0; ...
0 1 0; 0 0 1 0; 0 0 1 0; ...

```

```

    0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0
0 1 0;...
    0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1; 0 0 0 1; 0 0 0 1;...
    0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1; 0 0 0 1; 0 0 0 1;...
    0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1];

% ===== EXTRAER CARACTERÍSTICAS PRINCIPALES DE MATRIZ GENERAL =====
xc = Mat_general(:,2); %2 Simple Cuadrado Integral
yc = Mat_general(:,4); %4 Raíz media cuadrática
zc = Mat_general(:,6); %6 Longitud de forma de onda
% ===== EXTRAER CARACTERÍSTICAS PRINCIPALES DE LA MATRIZ DE PRUEBA =====
xcprueba = Mat_gprueba(:,2); %2 Simple Cuadrado Integral
ycprueba = Mat_gprueba(:,4); %4 Raíz media cuadrática
zcprueba = Mat_gprueba(:,6); %6 Longitud de forma de onda

% =====
% ===== GRAFICAMOS CARACTERÍSTICAS =====
S = repmat([50,60,70,80],34,1); % Asigna tamaño acuerdo al movimiento
C = repmat([1,2,3,4],34,1); % Asigna color acuerdo al movimiento
s = S(:);
c = C(:);
figure (1)
scatter3(xc,yc,zc,s,c, 'filled') % Gráfica en 3 dimensiones
figure (2)
gscatter(xc,yc,c) % Gráfica en 2 dimensiones

% =====
% ===== CONFORMAR MATRICES PARA LA RED NEURONAL =====
Mat_trescarac = [xc,yc,zc]; % Para el entrenamiento de la red
Mat_tresprueba = [xcprueba,ycprueba,zcprueba]; % Para la validación de la red

% -----
% ----- RED NEURONAL BACKPROPAGATION-----

% ===== Transponer matrices para ingresar al entrenamiento de la red =====
x = Mat_trescarac'; % Matriz de entrada de 136 x 3
xp = Mat_tresprueba'; % Matriz de validación de 136 x 3
ts = Mat_salida'; % Matriz de salida de 136 x 4
% Escoger la Función de entrenamiento
trainFcn = 'trainscg'; % Gradiente conjugado escalado backpropagation.

% Crear la red de reconocimiento de patrones
hiddenLayerSize = 10; % Establecer número de neuronas
net = patternnet(hiddenLayerSize, trainFcn); % Se crea la red

% Elija las funciones de entrada y salida de pre / post procesamiento
net.input.processFcns = {'removeconstantrows','mapminmax'};

% Escoger la división de los datos
net.divideFcn = 'dividetrain'; % Divide la información solo para el entrenamiento

% Escoger la función de desempeño
net.performFcn = 'crossentropy'; % Cross-Entropy

% Escoger las gráficas de las funciones
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
'plotconfusion', 'plotroc'};

% Entrenamiento de la red neuronal
[net,tr] = train(net,x,ts); % Con la matriz de entrada y salida

```

```

% Prueba y validación de la red neuronal
y = net(xp); % Con la matriz de prueba
e = gsubtract(ts,y);
performance = perform(net,ts,y) % Calcula el desempeño
tind = vec2ind(ts);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind); % Calcula el error

% Recalcula el entrenamiento, la validación y la prueba
trainTargets = ts .* tr.trainMask{1};
valTargets = ts .* tr.valMask{1};
testTargets = ts .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% Observar la red neuronal
view(net)

% Gráficas de interés
figure, plotperform(tr)
figure, plottrainstate(tr)
figure, ploterrhist(e)
figure, plotconfusion(ts,y)
figure, plotroc(ts,y)

```

ANEXO C.4: Código para Extensión de muñeca

```

% ===== CÓDIGO PARA MOVIMIENTO DE EXTENSIÓN DE MUÑECA =====
clc; clear all; close all;
% ===== Cargar las señales del movimiento en cada posición =====
load extension30grados.mat; % Base de datos para entrenamiento de la red
load extension45grados.mat;
load extension60grados.mat;
load extension70grados.mat;
load extension30gradosprueba.mat; % Base de datos para validación y prueba de la red
load extension45gradosprueba.mat;
load extension60gradosprueba.mat;
load extension70gradosprueba.mat;

% ===== Asignar la base de datos a una variable de tipo double =====
signal1 = double(EX30G);
signal2 = double(EX45G);
signal3 = double(EX60G);
signal4 = double(EX70G);
signal5 = double(EX30GP);
signal6 = double(EX45GP);
signal7 = double(EX60GP);
signal8 = double(EX70GP);

Fs = 1000; %Frecuencia de muestreo
T = 1/Fs; %Periodo de Muestro
L = length(signal1); %Tamaño de la señal
t = (0:L-1)*T; %Tiempo de la señal

% ===== Filtrado de Señal usando función Filter_signal=====
[fil1,fil2,fil3,fil4] = Filter_signal(signal1,signal2,signal3,signal4,Fs);
[fil5,fil6,fil7,fil8] = Filter_signal(signal5,signal6,signal7,signal8,Fs);

```

```

% ===== Rectificado de la señal =====
signalrec1= abs(fil1); signalrec5= abs(fil5); % Se aplica valor absoluto
signalrec2= abs(fil2); signalrec6= abs(fil6);
signalrec3= abs(fil3); signalrec7= abs(fil7);
signalrec4= abs(fil4); signalrec8= abs(fil8);

% ===== Detección de actividad usando función Activ =====
for i = 1:34
    sigrec1 = signalrec1(:,i);
    sigrec2 = signalrec2(:,i);
    sigrec3 = signalrec3(:,i);
    sigrec4 = signalrec4(:,i);

    [ACTSNL1] = Activ(sigrec1);
    [ACTSNL2] = Activ(sigrec2);
    [ACTSNL3] = Activ(sigrec3);
    [ACTSNL4] = Activ(sigrec4);
end

% ===== CARACTERIZACIÓN DE LA SEÑAL =====

% *** Para la caracterización se utiliza las funciones correspondientes a *
% *** cada característica, todas las funciones inician con el nombre Feat **
% *** Se realiza para la base de datos de entrenamiento y de validación ***
% *** y en algunas características se usa el número de muestras (34) y la
% *** frecuencia de muestreo (Fs)

% ===== Característica Integral de la señal EMG (IEMG) =====
[SNLIEMG1] = Featiemg(signalrec1); [SNLIEMG5] = Featiemg(signalrec5);
[SNLIEMG2] = Featiemg(signalrec2); [SNLIEMG6] = Featiemg(signalrec6);
[SNLIEMG3] = Featiemg(signalrec3); [SNLIEMG7] = Featiemg(signalrec7);
[SNLIEMG4] = Featiemg(signalrec4); [SNLIEMG8] = Featiemg(signalrec8);

% ===== Característica Simple cuadrado integral (SSI) =====
[SNLSSI1] = Featssi(signalrec1); [SNLSSI5] = Featssi(signalrec5);
[SNLSSI2] = Featssi(signalrec2); [SNLSSI6] = Featssi(signalrec6);
[SNLSSI3] = Featssi(signalrec3); [SNLSSI7] = Featssi(signalrec7);
[SNLSSI4] = Featssi(signalrec4); [SNLSSI8] = Featssi(signalrec8);

% ===== Característica Valor medio absoluto (MAV) =====
[SNLMAV1] = Featmav(signalrec1,34); [SNLMAV5] = Featmav(signalrec5,34);
[SNLMAV2] = Featmav(signalrec2,34); [SNLMAV6] = Featmav(signalrec6,34);
[SNLMAV3] = Featmav(signalrec3,34); [SNLMAV7] = Featmav(signalrec7,34);
[SNLMAV4] = Featmav(signalrec4,34); [SNLMAV8] = Featmav(signalrec8,34);

% ===== Característica raíz media cuadrática (RMS) =====
[SNLRootMS1] = Featrootms(signalrec1,34); [SNLRootMS5] = Featrootms(signalrec5,34);
[SNLRootMS2] = Featrootms(signalrec2,34); [SNLRootMS6] = Featrootms(signalrec6,34);
[SNLRootMS3] = Featrootms(signalrec3,34); [SNLRootMS7] = Featrootms(signalrec7,34);
[SNLRootMS4] = Featrootms(signalrec4,34); [SNLRootMS8] = Featrootms(signalrec8,34);

% ===== Característica Varianza (VAR) =====
[SNLVAR1] = Featvar(signalrec1,34); [SNLVAR5] = Featvar(signalrec5,34);
[SNLVAR2] = Featvar(signalrec2,34); [SNLVAR6] = Featvar(signalrec6,34);
[SNLVAR3] = Featvar(signalrec3,34); [SNLVAR7] = Featvar(signalrec7,34);
[SNLVAR4] = Featvar(signalrec4,34); [SNLVAR8] = Featvar(signalrec8,34);

% ===== Característica Longitud de Onda (WL) =====
[SNLWL1] = Featwl(signalrec1,34); [SNLWL5] = Featwl(signalrec5,34);
[SNLWL2] = Featwl(signalrec2,34); [SNLWL6] = Featwl(signalrec6,34);
[SNLWL3] = Featwl(signalrec3,34); [SNLWL7] = Featwl(signalrec7,34);
[SNLWL4] = Featwl(signalrec4,34); [SNLWL8] = Featwl(signalrec8,34);

```

```

% ===== Característica Transformada de Fourier rapida (FFT) =====
[SNLFFT1] = Feat_fft(signalrec1,Fs,34); [SNLFFT5] = Feat_fft(signalrec5,Fs,34);
[SNLFFT2] = Feat_fft(signalrec2,Fs,34); [SNLFFT6] = Feat_fft(signalrec6,Fs,34);
[SNLFFT3] = Feat_fft(signalrec3,Fs,34); [SNLFFT7] = Feat_fft(signalrec7,Fs,34);
[SNLFFT4] = Feat_fft(signalrec4,Fs,34); [SNLFFT8] = Feat_fft(signalrec8,Fs,34);

% ===== Característica Energía (E) =====
[SNLENERG1] = Feat_energy(signalrec1,Fs); [SNLENERG5] = Feat_energy(signalrec5,Fs);
[SNLENERG2] = Feat_energy(signalrec2,Fs); [SNLENERG6] = Feat_energy(signalrec6,Fs);
[SNLENERG3] = Feat_energy(signalrec3,Fs); [SNLENERG7] = Feat_energy(signalrec7,Fs);
[SNLENERG4] = Feat_energy(signalrec4,Fs); [SNLENERG8] = Feat_energy(signalrec8,Fs);

% ===== Característica Densidad de espectro de potencia (PSD) =====
[SNLPSD1] = Feat_PSD(signalrec1,Fs,34); [SNLPSD5] = Feat_PSD(signalrec5,Fs,34);
[SNLPSD2] = Feat_PSD(signalrec2,Fs,34); [SNLPSD6] = Feat_PSD(signalrec6,Fs,34);
[SNLPSD3] = Feat_PSD(signalrec3,Fs,34); [SNLPSD7] = Feat_PSD(signalrec7,Fs,34);
[SNLPSD4] = Feat_PSD(signalrec4,Fs,34); [SNLPSD8] = Feat_PSD(signalrec8,Fs,34);

%% ===== Característica Amplitud de Willison (WAMP) =====
[SNLWAMP1] = Featwamp(signalrec1,34); [SNLWAMP5] = Featwamp(signalrec5,34);
[SNLWAMP2] = Featwamp(signalrec2,34); [SNLWAMP6] = Featwamp(signalrec6,34);
[SNLWAMP3] = Featwamp(signalrec3,34); [SNLWAMP7] = Featwamp(signalrec7,34);
[SNLWAMP4] = Featwamp(signalrec4,34); [SNLWAMP8] = Featwamp(signalrec8,34);

% ===== Característica Cambio de signo de la señal (SSC) =====
[SNLSSC1] = Featssc(signalrec1,34); [SNLSSC5] = Featssc(signalrec5,34);
[SNLSSC2] = Featssc(signalrec2,34); [SNLSSC6] = Featssc(signalrec6,34);
[SNLSSC3] = Featssc(signalrec3,34); [SNLSSC7] = Featssc(signalrec7,34);
[SNLSSC4] = Featssc(signalrec4,34); [SNLSSC8] = Featssc(signalrec8,34);

% =====
% ===== FORMAR MATRIZ CON CARACTERÍSTICAS PARA CADA ÁNGULO =====

matriz_ch1 = [SNLIEMG1; SNLSSI1; SNLMAV1; SNLRootMS1; SNLVAR1; SNLWL1; SNLFFT1;
SNLENERG1; SNLPSD1; SNLSSC1; SNLWAMP1];
matriz_ch2 = [SNLIEMG2; SNLSSI2; SNLMAV2; SNLRootMS2; SNLVAR2; SNLWL2; SNLFFT2;
SNLENERG2; SNLPSD2; SNLSSC2; SNLWAMP2];
matriz_ch3 = [SNLIEMG3; SNLSSI3; SNLMAV3; SNLRootMS3; SNLVAR3; SNLWL3; SNLFFT3;
SNLENERG3; SNLPSD3; SNLSSC3; SNLWAMP3];
matriz_ch4 = [SNLIEMG4; SNLSSI4; SNLMAV4; SNLRootMS4; SNLVAR4; SNLWL4; SNLFFT4;
SNLENERG4; SNLPSD4; SNLSSC4; SNLWAMP4];

matriz_ch5 = [SNLIEMG5; SNLSSI5; SNLMAV5; SNLRootMS5; SNLVAR5; SNLWL5; SNLFFT5;
SNLENERG5; SNLPSD5; SNLSSC5; SNLWAMP5];
matriz_ch6 = [SNLIEMG6; SNLSSI6; SNLMAV6; SNLRootMS6; SNLVAR6; SNLWL6; SNLFFT6;
SNLENERG6; SNLPSD6; SNLSSC6; SNLWAMP6];
matriz_ch7 = [SNLIEMG7; SNLSSI7; SNLMAV7; SNLRootMS7; SNLVAR7; SNLWL7; SNLFFT7;
SNLENERG7; SNLPSD7; SNLSSC7; SNLWAMP7];
matriz_ch8 = [SNLIEMG8; SNLSSI8; SNLMAV8; SNLRootMS8; SNLVAR8; SNLWL8; SNLFFT8;
SNLENERG8; SNLPSD8; SNLSSC8; SNLWAMP8];

% ===== Normalización de matrices conformadas =====
Norm_MAT1 = matriz_ch1/max(abs(matriz_ch1(:))); Norm_MAT5 =
matriz_ch5/max(abs(matriz_ch5(:)));
Norm_MAT2 = matriz_ch2/max(abs(matriz_ch2(:))); Norm_MAT6 =
matriz_ch6/max(abs(matriz_ch6(:)));
Norm_MAT3 = matriz_ch3/max(abs(matriz_ch3(:))); Norm_MAT7 =
matriz_ch7/max(abs(matriz_ch7(:)));
Norm_MAT4 = matriz_ch4/max(abs(matriz_ch4(:))); Norm_MAT8 =
matriz_ch8/max(abs(matriz_ch8(:)));

% ===== Transpuesta de la matriz de características para cada ángulo =====
Caract30g = Norm_MAT1';      Caract30gp = Norm_MAT5';

```

```

Caract45g = Norm_MAT2';    Caract45gp = Norm_MAT6';
Caract60g = Norm_MAT3';    Caract60gp = Norm_MAT7';
Caract70g = Norm_MAT4';    Caract70gp = Norm_MAT8';

% =====
% ===== CONFORMAR MATRICES GENERALES =====
Mat_general = [Caract30g; Caract45g; Caract60g; Caract70g]; % MATRIZ GENERAL PARA
ENTRENAMIENTO
Mat_gprueba = [Caract30gp; Caract45gp; Caract60gp; Caract70gp]; % MATRIZ GENERAL PARA
VALIDACIÓN

% MATRIZ GENERAL DE SALIDA EN FORMA BINARIA
Mat_salida = [1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0;
1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0;...
    1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1
0 0 0; 1 0 0 0; 1 0 0 0;...
    1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1 0 0 0; 1
0 0 0;...
    0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0
1 0 0; 0 1 0 0; 0 1 0 0;...
    0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0
1 0 0; 0 1 0 0; 0 1 0 0;...
    0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0 1 0 0; 0
1 0 0;...
    0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0
0 1 0; 0 0 1 0; 0 0 1 0;...
    0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0 0 1 0; 0
0 1 0;...
    0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1; 0 0 0 1; 0 0 0 1;...
    0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1; 0 0 0 1; 0 0 0 1;...
    0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0 0 0 1; 0
0 0 1];

% ===== EXTRAER CARACTERÍSTICAS PRINCIPALES DE MATRIZ GENERAL =====
xc = Mat_general(:,2); %2 Simple Cuadrado Integral
yc = Mat_general(:,4); %4 Raíz media cuadrática
zc = Mat_general(:,6); %6 Longitud de forma de onda
% ===== EXTRAER CARACTERÍSTICAS PRINCIPALES DE LA MATRIZ DE PRUEBA =====
xcprueba = Mat_gprueba(:,2); %2 Simple Cuadrado Integral
ycprueba = Mat_gprueba(:,4); %4 Raíz media cuadrática
zcprueba = Mat_gprueba(:,6); %6 Longitud de forma de onda

% =====
% ===== GRAFICAMOS CARACTERÍSTICAS =====
S = repmat([50,60,70,80],34,1); % Asigna tamaño acuerdo al movimiento
C = repmat([1,2,3,4],34,1); % Asigna color acuerdo al movimiento
s = S(:);
c = C(:);
figure (1)
scatter3(xc,yc,zc,s,c, 'filled') % Gráfica en 3 dimensiones
figure (2)
gscatter(xc,yc,c) % Gráfica en 2 dimensiones

% =====
% ===== CONFORMAR MATRICES PARA LA RED NEURONAL =====
Mat_trescarac = [xc,yc,zc]; % Para el entrenamiento de la red
Mat_tresprueba = [xcprueba,ycprueba,zcprueba]; % Para la validación de la red

% -----

```

```

% ----- RED NEURONAL BACKPROPAGATION-----

% ===== Transponer matrices para ingresar al entrenamiento de la red =====
x = Mat_trescarac';    % Matriz de entrada de 136 x 3
xp = Mat_trespueba';  % Matriz de validación de 136 x 3
ts = Mat_salida';     % Matriz de salida de 136 x 4
% Escoger la Función de entrenamiento
trainFcn = 'trainscg'; % Gradiente conjugado escalado backpropagation.

% Crear la red de reconocimiento de patrones
hiddenLayerSize = 10; % Establecer número de neuronas
net = patternnet(hiddenLayerSize, trainFcn); % Se crea la red

% Elija las funciones de entrada y salida de pre / post procesamiento
net.input.processFcns = {'removeconstantrows','mapminmax'};

% Escoger la división de los datos
net.divideFcn = 'dividetrain'; % Divide la información solo para el entrenamiento

% Escoger la función de desempeño
net.performFcn = 'crossentropy'; % Cross-Entropy

% Escoger las gráficas de las funciones
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
               'plotconfusion', 'plotroc'};

% Entrenamiento de la red neuronal
[net,tr] = train(net,x,ts); % Con la matriz de entrada y salida

% Prueba y validación de la red neuronal
y = net(xp);    % Con la matriz de prueba
e = gsubtract(ts,y);
performance = perform(net,ts,y) % Calcula el desempeño
tind = vec2ind(ts);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind); % Calcula el error

% Recalcula el entrenamiento, la validación y la prueba
trainTargets = ts .* tr.trainMask{1};
valTargets = ts .* tr.valMask{1};
testTargets = ts .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% Observar la red neuronal
view(net)

% Gráficas de interés
figure, plotperform(tr)
figure, plottrainstate(tr)
figure, ploterrhist(e)
figure, plotconfusion(ts,y)
figure, plotroc(ts,y)

```