

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**IMPLEMENTACIÓN DE UN ALGORITMO PARA EL
RECONOCIMIENTO DE GESTOS DE LA MANO EN TIEMPO REAL,
USANDO SEÑALES ELECTROMIOGRÁFICAS**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

JAVIER ALEJANDRO ORDÓÑEZ FLORES

DIRECTOR: Ph.D. ROBIN GERARDO ÁLVAREZ RUEDA

Quito, junio 2021

AVAL

Certifico que el presente trabajo fue desarrollado por Javier Alejandro Ordóñez Flores

Ph.D. Robin Álvarez Rueda

DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Javier Alejandro Ordóñez Flores, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Javier Alejandro Ordóñez Flores

DEDICATORIA

Dedico de manera especial este trabajo a mis padres Efrén y Tania, pues ellos son los mentores que han permitido la construcción de mi vida profesional.

AGRADECIMIENTO

A mis padres Efrén y Tania por su apoyo incondicional, que me ha permitido alcanzar cada una de mis metas.

A mi hermano Joel y abuelito Aníbal por ser parte de los mejores momentos de mi vida.

A mi abuelita Margarita, quien ahora es un ángel en mi vida y sé que se encuentra muy orgullosa de su nieto y desde donde está me bendice.

A la familia Ruque Chamba por su apoyo en la construcción de mi vida profesional.

A mis amigos y compañeros por cada uno de los momentos compartidos.

Al Dr. Robin Álvarez por su tiempo, paciencia y excelente guía en este trabajo de titulación.

A los profesores de la Escuela Politécnica Nacional, por todo el esfuerzo en formar profesionales de calidad

Javier Ordóñez

ÍNDICE DE CONTENIDO

AVAL.....	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
RESUMEN	VIII
ABSTRACT.....	IX
1. INTRODUCCIÓN	1
1.1 OBJETIVOS	2
1.2 ALCANCE.....	3
1.3 MARCO TEÓRICO	5
1.3.1 ESTADO DEL ARTE	5
1.3.2 ELECTROMIOGRAFÍA (EMG)	6
1.3.3 EL BRAZALETE MYO.....	8
1.3.4 TÉCNICAS DE PROCESAMIENTO DIGITAL DE SEÑALES	10
1.3.4.1 Espectro de potencias	10
1.3.4.2 Rectificación.....	10
1.3.4.3 Envolvente de una señal.....	11
1.3.4.4 Suavizado de curvas	13
1.3.4.5 Área bajo la curva	14
1.3.4.6 Dimensión fractal de Katz	15
1.3.5 ANÁLISIS DE COMPONENTES PRINCIPALES.....	18
1.3.6 CLASIFICACIÓN.....	20
1.3.6.1 Distancia euclidiana al centroide más cercano	21
1.3.6.2 Redes neuronales artificiales (MLP)	22
2. METODOLOGÍA	26
2.1 CONDICIONES DE FUNCIONAMIENTO	26
2.2 COMUNICACIÓN ENTRE MYO ARMBAND Y LA PC	26
2.2.1 SOFTWARE PARA LA COMUNICACIÓN ENTRE MYO ARMBAND Y LA PC	27
2.2.1.1 Matlab.....	28
2.2.1.2 Mingw	28

2.2.1.3	Myo Connect Installer	29
2.2.1.4	Myo Mex master y Myo SDK	31
2.2.2	INSTALACIÓN Y EJECUCIÓN DE LAS LIBRERÍAS	33
2.2.3	PROGRAMA INICIAL DE LECTURA	35
2.3	ALGORITMO DE RECONOCIMIENTO DE GESTOS	36
2.3.1	ETAPA DE EXTRACCIÓN DE SEÑALES EMG	37
2.3.1.1	Fase 1: Determinación adaptativa de nivel ruido	38
2.3.1.2	Fase 2: Detección de inicio de gesto	40
2.3.1.3	Fase 3: Ventana de análisis	41
2.3.1.4	Caso de gestos sostenidos	43
2.3.2	ETAPA DE PREPROCESAMIENTO EN EL DOMINIO TEMPORAL	46
2.3.2.1	Extracción de señal útil	46
	47
2.3.2.2	Rectificación	47
2.3.2.3	Envolvente	48
2.3.2.4	Suavizado de curvas	50
2.3.3	ETAPA EXTRACCIÓN DE CARACTERÍSTICAS	51
2.3.3.1	Área bajo la curva (dominio del tiempo)	51
2.3.3.2	Área de densidad espectral de potencia suavizada (dominio de la frecuencia)	52
2.3.3.3	Dimensión fractal (Dominio de la complejidad)	56
2.4	CONFORMACIÓN DE CLÚSTERS N-DIMENSIONALES	59
2.5	ETAPA DE CLASIFICACIÓN	61
2.5.1	MÉTODO 1: DISTANCIA EUCLIDIANA AL CENTROIDE MÁS CERCANO	61
2.5.2	MÉTODO 2: RED NEURONAL BASADA EN PERCEPTRÓN MULTICAPA (MLP)	62
2.6	VARIABILIDAD INTRAPERSONAL E INTERPERSONAL	68
2.6.1	VARIABILIDAD INTRAPERSONAL	68
2.6.2	VARIABILIDAD INTERPERSONAL	71
3.	RESULTADOS Y DISCUSIÓN	73
3.1	COMPARACIÓN DE LOS TRES DOMINIOS Y SELECCIÓN	73
3.1.1	DISTANCIA ENTRE CENTROIDES	74
3.1.2	DISPERSIÓN DE DATOS	75
3.1.3	CARGA COMPUTACIONAL	75

3.2	VARIABILIDAD INTRAPERSONAL	76
3.2.1	NIVEL DE BATERÍA	77
3.2.1.1	Distancia entre centroides	79
3.2.1.2	Ubicación de clústers	80
3.2.1.3	Dispersión de clústers	81
3.2.2	DISTANCIA CODO - MYO	81
3.2.2.1	Distancia entre centroides	82
3.2.2.2	Ubicación de clústers	82
3.2.2.3	Dispersión de clústers	83
3.2.3	GIRO DE BRAZALETE	83
3.2.3.1	Distancia entre centroides	84
3.2.3.2	Ubicación de clústers	84
3.2.3.3	Dispersión de clústers	85
3.3	VARIABILIDAD INTERPERSONAL	85
3.3.1	VARONES	88
3.3.1.1	Distancia entre centroides	91
3.3.1.2	Ubicación de clústers	91
3.3.1.3	Dispersión de clústers	92
3.3.2	MUJERES	93
3.3.2.1	Distancia entre centroides	95
3.3.2.2	Ubicación de clústers	95
3.3.2.3	Dispersión de clústers	96
3.4	CLASIFICACIÓN	96
3.4.1	MÉTODO 1: DISTANCIA EUCLIDIANA AL CENTROIDE MÁS CERCANO	96
3.4.2	METODO 2: RED NEURONAL BASADO EN PERCENTRÓN MULTICAPA (MLP)	97
3.5	SISTEMA EN TIEMPO REAL	98
4.	CONCLUSIONES Y RECOMENDACIONES	99
4.1	CONCLUSIONES	99
4.2	RECOMENDACIONES	101
5.	REFERENCIAS BIBLIOGRÁFICAS	102
	ANEXOS	104

RESUMEN

En este documento se propone un sistema de reconocimiento de 5 gestos de la mano (dedos separados, derecha, izquierda, puño y doble toque), basado en 8 canales de Electromiografía (EMG), utilizando el dispositivo Myo armband colocado en el antebrazo. Tradicionalmente, la etapa de preprocesamiento y extracción de características son realizados sin una metodología que las justifique o muestre su influencia en la exactitud y precisión de clasificación y la mayor importancia se asigna al método de clasificación. En nuestro caso, en general, si se seleccionan más de tres variables, sus valores experimentales conformarán nubes o clústers de puntos en el espacio n-dimensional. De este modo, ahora la selección de aquellos algoritmos está basada en la maximización de la distancia euclidiana entre clústers y en la minimización de la dispersión de dichas nubes. Por otro lado, un asunto muy descuidado es la comprensión del fenómeno en función de la variabilidad tanto intrapersonal como interpersonal, considerando en esta última el sexo, grado de entrenamiento del sujeto y su contextura física. Finalmente, luego de la comprensión de estos escenarios, se propone una metodología la cual, aplicada a un grupo de 12 sujetos, obtuvo una exactitud de clasificación de los 5 gestos del 97%.

ABSTRACT

In this document we propose a recognition system of 5 hand gestures (fist, wave in, wave out, open and pinch), based on 8 channels of Electromyography (EMG), using the Myo armband device placed on the forearm. Traditionally, the preprocessing and feature extraction stages are performed without a methodology that justifies them or shows their influence on the accuracy and precision of classification and the highest importance is assigned to the classification method. In our case, in general, if more than three variables are selected, their experimental values will form clouds or clusters of points in n-dimensional space. Thus, now the selection of those algorithms is based on maximizing the Euclidean distance between clusters and minimizing the dispersion of these clouds. On the other hand, a much neglected matter is the understanding of the phenomenon based on both intrapersonal and interpersonal variability, considering the sex, training degree of the subject and physical makeup. Finally, after understanding these scenarios, a methodology is proposed which, applied to a group of 12 subjects, obtained a classification accuracy of the 5 gestures of 97%.

1. INTRODUCCIÓN

La dificultad de comunicación oral de personas sordomudas, la dificultad para realizar movimientos con las manos de personas con discapacidad en sus extremidades, el desarrollo actual de nuevas interfaces de entrada entre humano-máquina para juegos, sistemas de realidad virtual, robots, equipos médicos, domótica, etc. Son problemáticas que enfrentan a diario muchas personas, las cuales podrían mejorar su estilo de vida, con mecanismos capaces de comandar prótesis de mano, sistemas de traducción de lenguaje de señas a texto o voz, inclusive para aumentar el realismo dentro de los videojuegos [1].

Para resolver estos inconvenientes, existen sensores de reconocimiento de gestos de la mano como: guantes instrumentados, cámaras de profundidad, cámaras de color, sensores ultrasónicos, guantes con marcadores de color, sensores de señales electromiográficas. Un ejemplo es la utilización de cámaras de profundidad, las cuales no requieren estar adheridas a las extremidades, no son afectadas por factores de iluminación, sombra o color, pero son de elevado costo y difícil uso en modelos de reconocimiento portables [2].

Otro sistema capaz de reconocer los gestos de la mano son los guantes instrumentales, estos proporcionan más información sobre cada uno de los dedos de la mano, pero su utilización es incómoda y no existen tallas para diferentes usuarios [3]. Las cámaras de color y los sensores ultrasónicos son otra opción para reconocer los gestos de la mano, estos son de bajo costo, pero difíciles de usar en modelos de reconocimiento portables [4]. También están los sensores de electromiografía, capaces de medir las contracciones musculares del antebrazo, los cuales no son afectados por variaciones de luz como sucede con las cámaras de color, pero tiene presente ruido de diferentes fuentes que podrían afectar el análisis de las señales y su correcta clasificación [5].

El presente proyecto técnico, propone el desarrollo de un algoritmo para el reconocimiento de 5 gestos de la mano, según la aplicación de la mejor variable a ser considerada en la etapa de extracción de características, después de un análisis en tres dominios (tiempo, frecuencia y complejidad), el método de clasificación más apropiado, la verificación de la estabilidad del sistema a lo largo del tiempo, para un mismo sujeto de experimentación (variabilidad intrapersonal) y los cambios a considerar en la aplicación con distintos usuarios (variabilidad interpersonal).

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es: implementar un algoritmo para el reconocimiento de gestos de la mano en tiempo real, usando señales electromiográficas.

Los objetivos específicos del Proyecto Técnico son:

- i) Determinar el estado del arte en cuanto a algoritmos y tecnologías de detección de gestos de la mano utilizando señales electromiográficas.
- ii) Implementar un algoritmo de reconocimiento de 5 gestos de la mano (dedos separados, derecha, izquierda, puño y doble toque), en base a señales EMG, obtenidas mediante el dispositivo Myo Armband en tiempo real.
- iii) Analizar la variabilidad intrapersonal de un sujeto de experimentación, observando su comportamiento EMG a lo largo del tiempo.
- iv) Analizar la variabilidad interpersonal, observando las diferencias en las señales EMG de distintas personas, para su utilización con cualquier usuario.

1.2 ALCANCE

Los sujetos de experimentación que se utilizarán en el proyecto serán los mayores posibles debido a la situación de la pandemia.

La adquisición de señales electromiográficas se realizará mediante el dispositivo comercial Myo Armband. En este trabajo no se realizará ningún diseño electrónico.

Se utilizarán las librerías existentes en Matlab para la comunicación con el dispositivo Myo Armband mediante Bluetooth.

La clasificación de los gestos de la mano se los realizará en base a los ocho canales de electromiografía, proporcionados por el dispositivo Myo Armband.

En este proyecto se trabajará con personas que no tengan problemas de movilidad en sus extremidades.

Los sujetos de experimentación pueden ser de diferente contextura física y cualquier sexo (varones y mujeres).

La cantidad de gestos de la mano a clasificar serán 5: dedos separados (open), derecha (wave out), izquierda (wave in), puño (fist) y doble toque (pinch) (Figura 1.1).



Figura 1.1. Gestos que serán realizados.

Las etapas para implementar serán las siguientes: adquisición de señal, extracción de señales, preprocesamiento, extracción de características y clasificación.

La adquisición de señal consistirá en la obtención de señales EMG por medio del dispositivo Myo Armband y Matlab.

La etapa de extracción de señales consistirá en obtener el nivel de ruido en cada canal del dispositivo Myo armband para la detección de inicio de un gesto y preparar las señales EMG para su análisis.

La etapa de preprocesamiento realizado para el caso del dominio del tiempo implementará las siguientes 3 fases: rectificación, envolvente y suavizado de curvas utilizadas para obtener una mayor separación de clústers.

En la extracción de características se la realiza 3 dominios (variables): tiempo, frecuencia y complejidad. De todas las variables analizadas se tomará la que mayor separación entre nubes conseguido, la que menor tiempo de procesamiento requerido y la de menor dispersión de clústers.

La etapa de clasificación utiliza dos métodos: distancia euclidiana al centroide más cercano y redes neuronales artificiales basado en perceptrón multicapa (MLP).

Por lo anterior, este trabajo sí consta de producto final demostrable.

El producto final demostrable consta de un programa realizado en Matlab que realiza el reconocimiento de los cinco gestos y que incluye las etapas antes mencionadas. Al final, se imprime, en tiempo real, el gesto realizado por el usuario por medio de una imagen que muestre la figura y el nombre del gesto en el command windows.

Para observar su influencia en la clasificación, las pruebas que se realizarán con el producto final serán las siguientes:

- Distintas velocidades en el movimiento de los gestos (rápido, normal, lento).
- Distintas posiciones de la mano (vertical, horizontal).
- Brazo estirado y sin estirar.
- Gestos por un tiempo prolongado (sosteniendo el gesto) y sin sostener el gesto.

Para estudiar la variabilidad intrapersonal se realizarán pruebas en diferentes horas y días, con un mismo sujeto de experimentación.

Para estudiar la variabilidad interpersonal se realizarán pruebas con diferentes personas (varones y mujeres) de diferente perímetro de antebrazo.

1.3 MARCO TEÓRICO

1.3.1 ESTADO DEL ARTE

Existen varias investigaciones relacionadas con el reconocimiento de gestos de la mano utilizando señales EMG. Cada proyecto presenta su propia metodología que los caracteriza según la exactitud de reconocimiento conseguido y métodos utilizados. Existen modelos de reconocimiento de gestos de la mano específicos, que requieren un entrenamiento previo a la utilización del sistema. Otros modelos se caracterizan por ser generales, es decir, no requieren entrenamiento alguno para su utilización. Estos modelos también se definen por presentar dos maneras diferentes de funcionamientos: en tiempo real y en tiempo diferido (no tiempo real). Por ejemplo en [6] se propone un sistema de reconocimiento de 3 posiciones de brazo (flexionado, semiflexionado y estándar), utilizando como método de preprocesamiento un filtro butterworth de orden 8; las variables utilizadas dentro de la etapa de extracción de características fueron: valor absoluto medio (MAV), varianza (VAR), raíz cuadrada media (RMS), desviación estándar (DS), entropía espectral de potencia (PSE); finalmente en la etapa de clasificación se empleó SVM (Support-Vector Machine). En [7] se plantea el control de una prótesis para el reconocimiento de dos gestos de mano (apertura y cierre) y de dos movimientos del brazo (flexión y extensión), utilizando en la etapa de extracción de características la estimación de torque articular (movimiento articular producido por la tensiones musculares) a partir de señales EMG y en la etapa de clasificación una red neuronal. En [8] se presenta el reconocimiento entre flexión y extensión de la muñeca basada en sensores de EMG colocados en el antebrazo: en la etapa de extracción características se utilizaron el valor absoluto medio (MAV), el número de cruces por cero (Nzc), el cambio de signo de la pendiente (NSSC) y la longitud de onda (WL); finalmente, en la etapa de clasificación se empleó LDA (análisis discriminante lineal). En [9], se propone un sistema de reconocimiento de 40 gestos de la mano, utilizando el dispositivo Myo Armband, en la etapa de extracción de características se usaron los siguientes métodos: a) en el dominio temporal se emplearon: RMS, MAV, relación de energía (ER), histograma, VAR, amplitud Willison (WAMP), Nzc, b) en el dominio de frecuencia se emplearon: espectro de amplitud, frecuencia media modificada (MMDF), frecuencia promedio modificada (MMNF); finalmente, en la etapa de clasificación se utilizó SVM. Como se puede ver en las investigaciones antes mencionadas, es común utilizar algunos algoritmos dentro de la etapa de preprocesamiento y algunas variables empleadas dentro de la etapa de extracción de características; sin embargo, no se realiza una

evaluación de su utilización y, peor aún, no se demuestra su grado de influencia en la exactitud de la clasificación.

Por otro lado, en la etapa de clasificación, las diferentes investigaciones se centran en la aplicación de diversos métodos y en la comparación de los resultados alcanzados por estos: MLP (Multilayer Perceptron), PNN (Probabilistic Neural Network), K-NN (K-Near Neighbor), NB (Naive Bayes), SVM (Support Vector Machines), DT (Decision Tree), etc. Por ejemplo, en [10] se hace una comparación de los métodos K-NN, NB y SVM para la detección de línea de corte en sistemas de protección eléctrica, siendo el clasificador NB el que mejores resultados obtuvo. En [11], se compara algunos métodos de clasificación supervisada para la detección de la enfermedad de Parkinson, siendo SVM el que mejores resultados obtuvo. Si bien en cada uno de estos trabajos, uno de estos métodos de clasificación obtiene el mejor resultado, no existe un estudio comparativo que permita determinar de forma general cuál de dichos métodos es el mejor y siempre queda pendiente la respuesta a este interrogante.

1.3.2 ELECTROMIOGRAFÍA (EMG)

La electromiografía es la medición, análisis y procesamiento de potenciales eléctricos o también llamadas señales electromiográficas o mioeléctricas producidas por las contracciones musculares. En el proceso de contracción de los músculos, estos generan tensiones de aproximadamente 100mV. Estas tensiones son aminoradas en el tejido interno de la piel, lo que afecta a las señales obtenidas en su superficie. Durante la actividad motora que se genera por contracción muscular voluntaria aparecen potenciales de acción de unidad motora. A estos potenciales se les estudia su morfología, amplitud, frecuencia y número [12]. Las señales analizadas se conocen como electromiográficas o EMG, las cuales se caracterizan por ser modeladas como un proceso estocástico no-estacionario es decir que son señales aleatorias no estacionarias (varían continuamente).

La figura 1.2, muestra un ejemplo de una señal EMG producida por los músculos

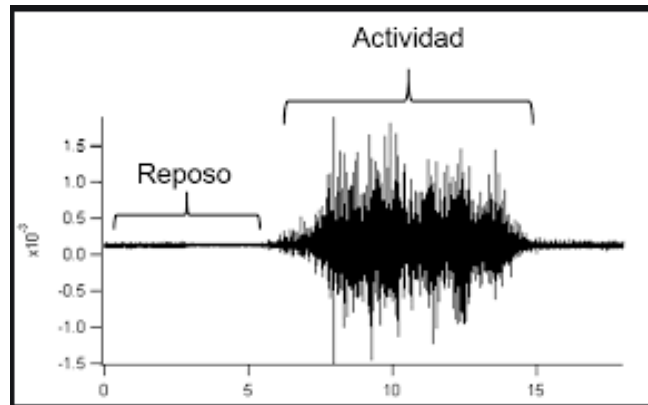


Figura 1.2. Ejemplo de señal de electromiografía

1.3.2.1 Miología del antebrazo

El estudio de los músculos involucrados en los movimientos de la mano, es muy importante para determinar la posición más idónea para la colocación de los sensores de electromiografía.

Los músculos del antebrazo son todos aquellos que están ubicados entre el codo y la muñeca, poseen dos huesos: el radio y el cubito y algunos tendones importantes para el movimiento de la mano [13]. La figura 1.3, muestra los músculos del antebrazo.

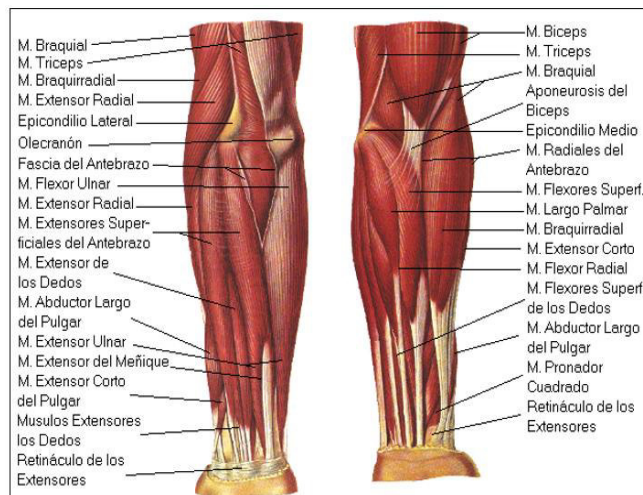


Figura 1.3. Músculos del antebrazo

Los músculos más utilizados en los movimientos básicos de la mano son:

- M. Flexores superficiales
- M. Extensores superficiales
- M. Braquirradial
- M. Flexor Ulnar

- M. Largo Palmar
- M. Pronador redondo
- M. Flexores superficiales de los dedos

Estos músculos, que en su gran mayoría se ubican en la parte superior del antebrazo, brindan la mayor información acerca de los movimientos que realiza la mano, por ello se ha considerado utilizar el dispositivo Myo Armband, que maneja 8 sensores que pueden ser colocados alrededor de estos músculos principales.

1.3.3 EL BRAZALETE MYO

El brazalete Myo Armband, es un dispositivo para el reconocimiento de gestos de la mano, fabricado por Thalmic Labs. Este se coloca en el antebrazo y permite diferentes niveles de ajuste gracias a sus bandas elásticas. Los perímetros de antebrazo que permite van de 19 a 34 cm.

El brazalete myo Armband contiene 8 sensores de electrodos secos, considerados no invasivos, de rápida colocación y de larga vida útil, pero proporcionan señales débiles (orden micro voltios) y pueden inducir ruido electrónico que es capaz de afectar a las señales adquiridas. Cada sensor distribuido alrededor del antebrazo, capta las señales electromiográficas producidas en las contracciones musculares y por medio de circuitos operaciones se amplifica las señales para su análisis. También envía la información de las señales EMG a un PC por medio de la tecnología Bluetooth. El brazalete mide las señales EMG a una frecuencia de muestreo de 200 Hz y representa cada medición en 8 bits [14].

El myo Armband posee una batería recargable de litio y cuenta con un software privado para el reconocimiento de gestos de la mano en tiempo real, capaz de reconocer los gestos: Fist (Puño), Wave in (Izquierda), Wave out (Derecha), Open (Abierto), Pinch (Doble toque). En la figura 1.4 se puede observar el dispositivo myo Armband utilizado para la implementación del algoritmo de reconocimiento de gestos de la mano.

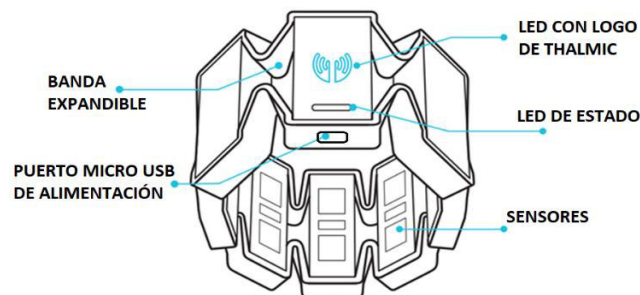




Figura 1.4. Brazalete Myo Armband

Myo Armband cuenta con un led indicador de estado y un led de logo Thalmic, estos leds brindan información sobre el estado del dispositivo Myo. La tabla 1.1, muestra a detalle la condición de cada led.

Tabla 1.1. Detalle de indicadores led del brazalete [15]

Led de Estado 	Led LogoThalmic 	Descripción
Apagado	Apagado	* Batería descargada *Myo en modo sleep, moverlo para despertarlo * Myo apagado
Constante azul y flash naranja	Constante azul	*Batería Baja o reposo
Constante verde	Apagado	*Se ha completado la carga
Destello naranja	Apagado	*Cargando
Constante azul	Constante azul	*Bluetooth conectado y se ha realizado la sincronización.
Constante azul	Destello azul	*Bluetooth conectado y no se ha realizado sincronización.
Apagado	Constante azul	*Bluetooth desconectado y se ha realizado la sincronización. Myo no entra en modo de suspensión.
Apagado	Destello azul	*Bluetooth desconectado y sin sincronización. Myo entrará en modo de suspensión después de un periodo de tiempo.
Apagado	Flash Azul	*Myo sincronizado y Bluetooth desconectado. Myo no entrará en modo de suspensión
Destello purpura	Apagado	*Myo está completando la actualización de su firmware
Constante azul	Flash Azul	*Myo está calentándose. Vibrará después de estar listo
Flash rojo	Apagado	*Error. Reinstalar firmware
Constante rojo	Apagado	*Error. Contactar con servicio al cliente

1.3.4 TÉCNICAS DE PROCESAMIENTO DIGITAL DE SEÑALES

1.3.4.1 Espectro de potencias

La transformada discreta de Fourier o DFT es una herramienta matemática, muy utilizada en el campo del procesamiento de señales digitales, ya que permite realizar un análisis en frecuencia de cualquier señal. En la Ecuación 1.1, se muestra la transformada discreta de Fourier, con la cual podemos encontrar el espectro de potencias de una señal EMG.

$$Y(k) = \sum_{n=0}^{N-1} y(n)W^{nk} \rightarrow DFT \quad (1.1)$$

donde:

$y(n)$: muestras de la serie temporal

N : número total de muestras de la serie temporal

$Y(k)$: números complejos obtenidos, uno por cada muestra temporal

$$W^{nk} = e^{\frac{-i2\pi}{N}}$$

n : número de muestra

k : frecuencia

i : unidad imaginaria

Aunque es posible utilizar la transformada discreta de Fourier DFT para el caso de pocas muestras, su carga computacional es prohibitiva y es mejor emplear la transformada rápida de Fourier o Fast Fourier Transform (FFT) [16].

1.3.4.2 Rectificación

La rectificación es el proceso de convertir una señal continua o discreta, para diferentes aplicaciones. Algunas de estas aplicaciones son: conversión de corriente alterna en continua en señales eléctricas, análisis de amplitud en señales discretas, detección de frecuencias bajas en espectros de potencia, entre otras. Existen dos tipos de rectificación de señal: de media onda y de onda completa.

La rectificación de media onda consiste en utilizar la parte positiva de la señal, descartando los valores negativos.

La rectificación de onda completa consiste en convertir los valores negativos de una señal en positivos, aplicando el valor absoluto a toda la señal.

La figura 1.5, muestra los dos tipos de rectificación existentes aplicado a una señal sinusoidal continua.

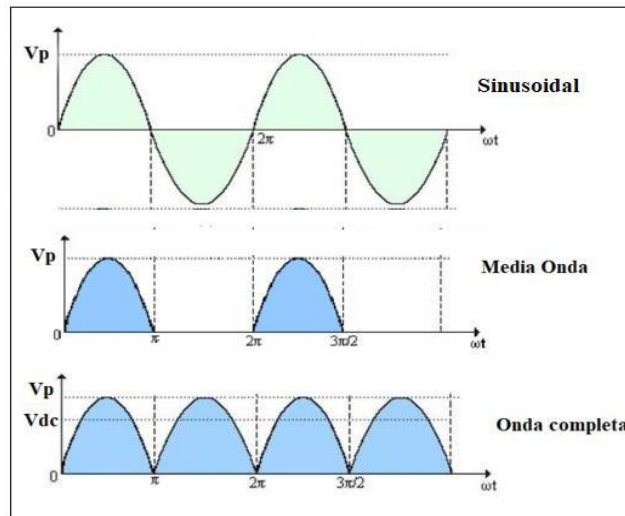


Figura 1.5. Rectificación de media onda y onda completa de una señal sinusoidal

1.3.4.3 Envoltente de una señal

La envoltente de una señal es la curva que enlaza tangencialmente a una señal, pudiendo envolver a una señal en la parte superior o inferior, incluso puede hallarse una envoltente que describa un valor RMS de la señal.

Un método utilizado para calcular la envoltente de una señal discreta que permite obtener una silueta superior o inferior de una señal con bastante precisión es mediante la interpolación por splines cúbicos que consiste en obtener una función polinómica a trozos $S(x)$ formada de splines cúbicos que son polinomios de tercer grado dados por la Ecuación 1.2.

$$Fi(x) = ax^3 + bx^2 + cx + d \quad (1.2)$$

La función polinómica a trozos (envoltente) es la unión de los polinomios de tercer grado, la cual se expresa según la Ecuación 1.3 [17].

$$F(x) = \left\{ \begin{array}{l} F1(x) \text{ si } x \in [t1, t2] \\ F2(x) \text{ si } x \in [t2, t3] \\ F3(x) \text{ si } x \in [t3, t4] \\ \vdots \\ Fn(x) \text{ si } x \in [tn, t(n+1)] \end{array} \right\} \quad (1.3)$$

donde:

$$F_i(x) = \frac{z_i}{6h_i} (t_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i} (x - t_i)^3 + \left(\frac{y_{i+1}}{h_i} + \frac{z_{i+1} \cdot h_i}{h_i} \right) (x - t_i) + \left(\frac{y_i}{h_i} + \frac{z_i \cdot h_i}{h_i} \right) (t_{i+1} - x) \quad (1.4)$$

Zi: es el resultado de la expresión matricial

$$\begin{bmatrix} u_1 & h_1 & & & & & \\ h_1 & u_2 & h_2 & & & & \\ & h_2 & u_3 & h_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & h_{n-3} & u_{n-2} & h_{n-2} & \\ & & & & h_{n-2} & u_{n-2} & \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{n-2} \\ z_{n-3} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{n-2} \\ v_{n-3} \end{bmatrix} \quad (1.5)$$

donde:

$$h_i = t_{i+1} - t_i$$

$$u_i = 2(h_i + h_{i-1}) - \frac{h_{i-1}^2}{u_i}$$

$$b_i = \frac{6}{h_i} (y_{i+1} - y_i)$$

$$v_i = b_i - b_{i-1} - \frac{h_{i-1} \cdot v_{i-1}}{u_{i-1}}$$

La figura 1.6, muestra el método de interpolación por splines cúbicos de valores tabulados, para describir una función aproximada.

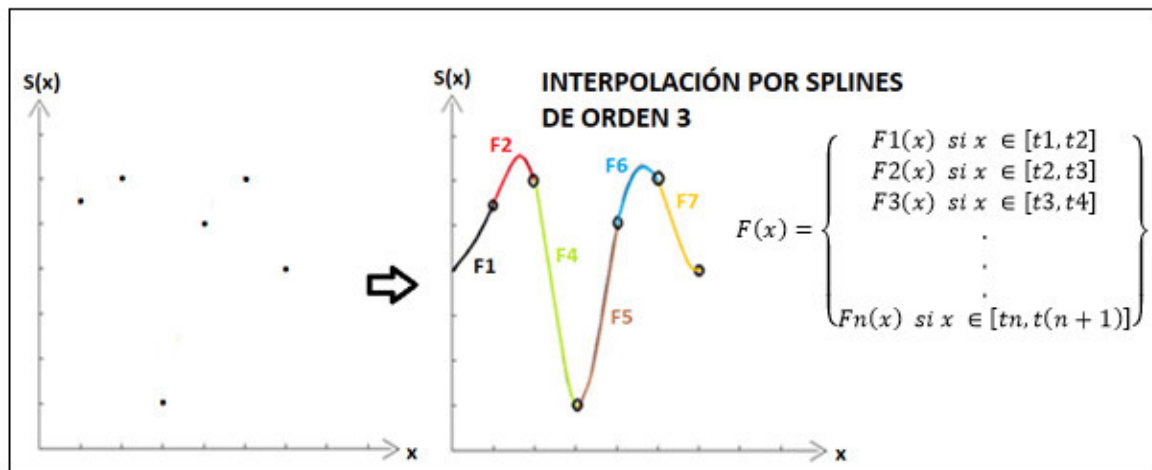


Figura 1.6. Interpolación segmentario por splines cúbicos

Para determinar la envolvente de una señal discreta, se utiliza puntos de referencia cada cierto número de muestras, de esta manera se encuentra los polinomios cúbicos entre los

puntos. La figura 1.7, muestra un ejemplo de determinación de envolvente de una señal, utilizando splines cúbicos.

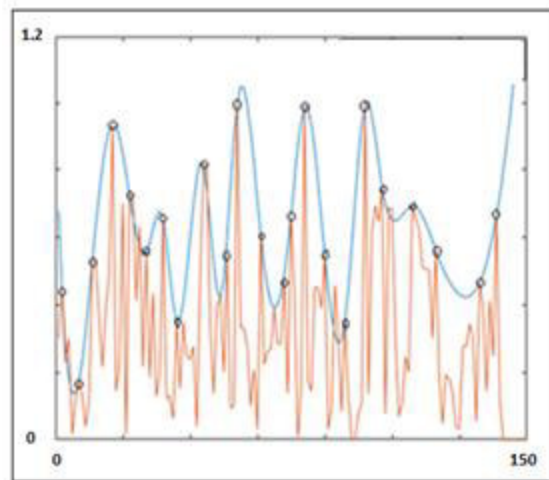


Figura 1.7. Envolvente con interpolación segmentaria por splines cúbicos

1.3.4.4 Suavizado de curvas

El suavizado de curvas es un método utilizado en procesamiento de señales para modificar estéticamente líneas digitalizadas. Si se aplica un suavizado de curvas a las envolventes de señales, estamos proporcionando información sobre la tendencia de estas envolventes, al eliminar picos indeseados, disminuyendo su grado de ajuste.

Se puede emplear algunos algoritmos para un suavizado como: filtro de media móvil, filtro savitzky golay o raíz media cuadrada. El método de menor complejidad que se utiliza es el filtro de media móvil, que consiste en aplicar un filtrado a una señal, utilizando una media aritmética a los valores de una ventana deslizante predeterminada de tamaño N, que recorre a toda la señal. La Ecuación 1.6 muestra la fórmula aplicada en la ventana deslizante para cada valor de la señal envolvente discreta.

$$z[n] = \frac{1}{M} \sum_{k=0}^{M-1} y[n - k] \quad (1.6)$$

donde:

$y[n]$: señal sin suavizado

M: número de muestras o tamaño de la ventana deslizante

$z[n]$: señal suavizada

La figura 1.8, muestra un ejemplo de suavizado de curvas, aplicado a una señal sinusoidal con ruido.

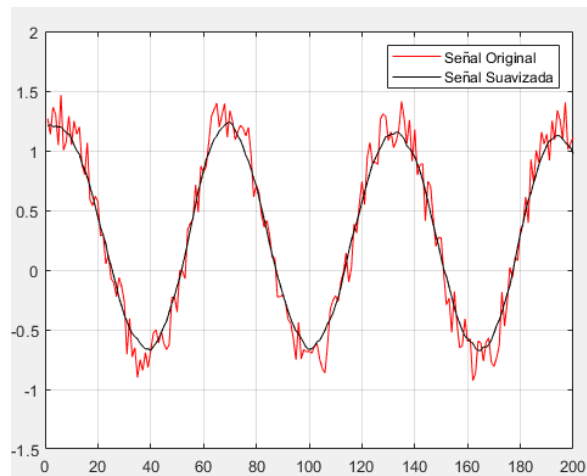


Figura 1.8. Filtro de media móvil para suavizado de datos

1.3.4.5 Área bajo la curva

El área bajo la curva de una señal es una medida que describe la superficie encerrada por una figura geométrica regular o irregular formada bajo la envolvente de una señal discreta o continua. El área bajo la curva de una señal continua puede ser calculada utilizando la integral de la función.

La figura 1.9, muestra el área bajo curva, calculado a partir de la integral de la función $y=f(x)$, de a hasta b .

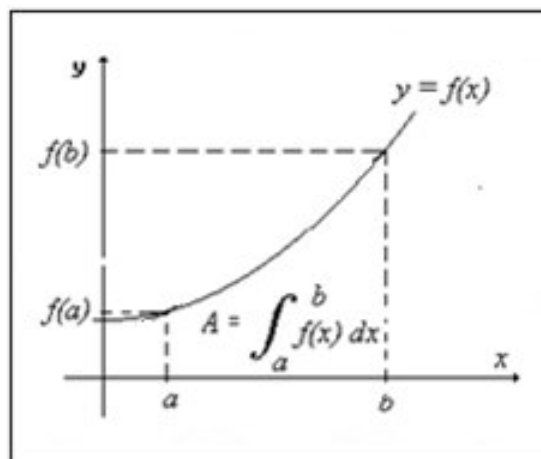


Figura 1.9. Área bajo la curva

Para señales discretas, el método utilizado para hallar el área bajo la curva, se conoce como el método del trapecio, este método consiste en encontrar el área de uno o muchos trapecio formado bajo la curva de una función.

La figura 1.10, muestra el área bajo la curva, calculado utilizando la sumatoria de áreas de trapecios formados bajo la de curva de una función.

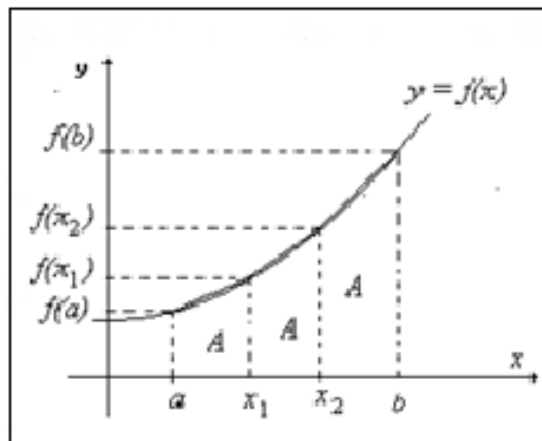


Figura 1.10. Área bajo la curva utilizando sumatoria de trapecios

La Ecuación 1.7, muestra la expresión utilizada para el cálculo del area bajo la curva de una señal utilizando el método del trapecio.

$$\int_a^b f(x)dx \approx \frac{1}{2} \sum_{n=1}^N (f(x_n) + f(x_{n+1})) \quad (1.7)$$

1.3.4.6 Dimensión fractal de Katz

Una de las herramientas utilizadas para el análisis de señales que presentan un comportamiento caótico (no predecible o aleatorio) es la dimensión fractal. Este término propuesto por el matemático Benoit Mandelbrot, se aplica a objetos geométricos o a señales autosimilares, cuya forma se repite a diferentes niveles de ampliación (diferentes escalas).

En la figura 1.11, se muestra un fractal ideal conocido como conjunto de Mandelbrot, creada por algoritmos iterativos que describen patrones irregulares y en donde se puede observar el concepto de autosimilaridad a diferentes escalas.

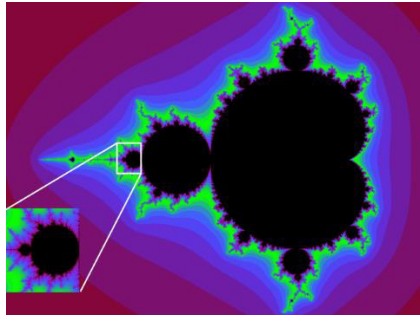


Figura 1.11. Conjunto de Mandelbrot

En la figura 1.12, se muestra un fractal producido por la naturaleza, que puede ser descrito mediante la geometría fractal y presenta autosimilaridad a distintas escalas.

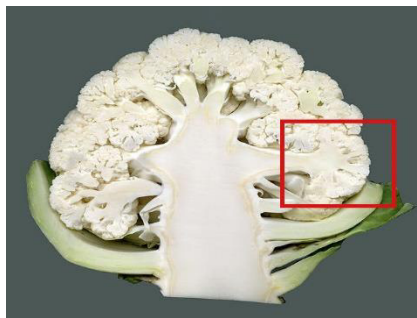


Figura 1.12. Coliflor (Verdura)

La figura 1.13, muestra a la izquierda una señal modelada como un proceso estocástico no-estacionario de autosimilaridad que puede ser descrita mediante la geometría fractal, ya que las señales se parecen entre sí a diferentes escalas de amplitud. Comparado con la señal a la derecha, que con cada ampliación resulta más irregular y de menor frecuencia.

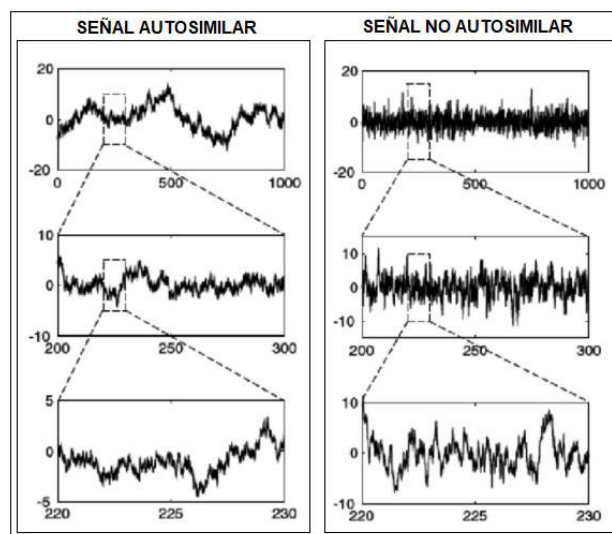


Figura 1.13. Señal autosimilar y no autosimilar

La dimensión fractal es la forma de medir un conjunto fractal, hallando un número fraccionario (dimensión fraccional) de cualquier serie temporal autosimilar.

Las señales EMG son series de tiempo irregulares representadas en forma de onda, que pueden ser descritas como una serie de tiempo discreta de la forma $S = \{S_1, S_2, \dots, S_n\}$

donde:

n : representa el total de muestras de la serie temporal

S : los valores sucesivos de las señales EMG.

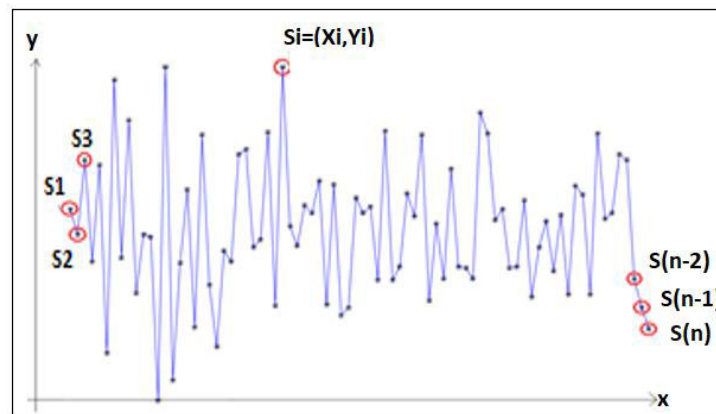


Figure 1.14. Señal EMG representada como serie de tiempo

Dentro del estudio de la dimensión fractal existen algunos métodos muy utilizados para el análisis de series temporales como son:

- Dimensión fractal de Higuchi
- Dimensión fractal de Katz
- Dimensión fractal multiresolución de conteo de cajas.

El método que se utiliza es dimensión fractal de Katz, debido a su baja carga computacional, comparado con los otros dos métodos y por la gran diferenciación conseguida entre los gestos estudiados.

La Ecuación 1.8, muestra la expresión para el cálculo de la dimensión fractal de Katz [18].

$$DFK = \frac{\log_{10} N}{\log_{10} \frac{d}{L} + \log_{10} N} \quad (1.8)$$

donde:

L : suma de las distancias euclidianas entre puntos sucesivos

$$L = \sum_{i=1}^N \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2} \quad i=1,2,3,n-1 \quad (1.9)$$

N: longitud o número de muestras de la serie de tiempo

d: distancia máxima entre el primer punto y el punto más lejano de la serie temporal
 $\max \{\text{dist}(s_1, s_i), i = 1, \dots, N\}$

El producto de aplicar el algoritmo de dimensión fractal de Katz a una serie temporal irregular con autosimilaridad es un número fraccionario que identifica la señal analizada. La figura 1.15, muestra un ejemplo de aplicación de dimensión fractal a una señal.

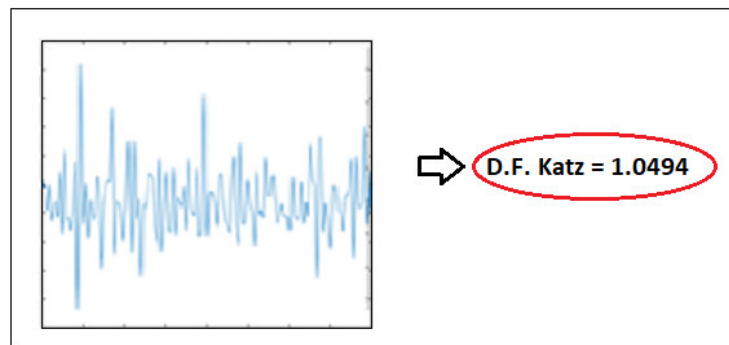


Figura 1.15. Dimensión fractal de Katz

El valor de 1.0494 es la dimensión fractal de la serie temporal, la cual identifica de manera única a la señal presentada.

1.3.5 ANÁLISIS DE COMPONENTES PRINCIPALES

El análisis de componentes principales es un método algebraico-estadístico de visualización de clases utilizado para reducción de dimensionalidad, es decir, describe un conjunto de datos en nuevas componentes que no están correlacionadas (linealmente independientes), de tal manera que se conserve la mayor cantidad de información del conjunto de datos original [19]. La figura 1.16, muestra un ejemplo de reducción de dimensionalidad de un conjunto de datos en 2 dimensiones a 1 dimensión.

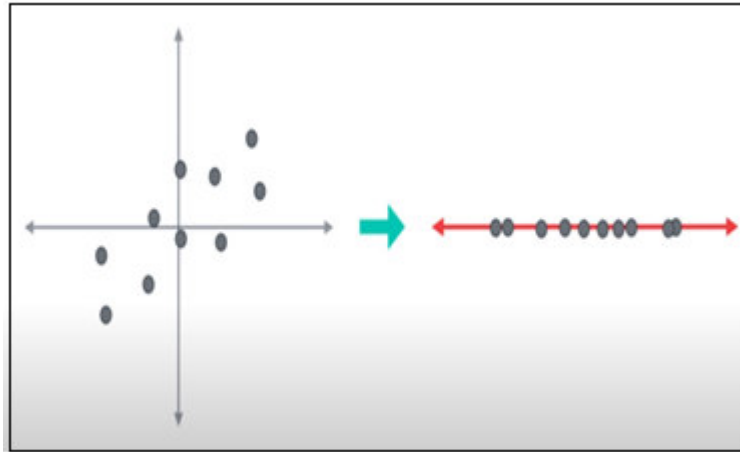


Figura 1.16. Reducción de dimensionalidad 2 a 1 utilizando PCA

El cálculo de las componentes principales parte de un conjunto de datos representado por una matriz $X_{n \times m}$ donde n representa los valores de información y m (columnas) representa las variables del conjunto de datos. Se pretende reducir el número de variables introducidas (dimensión) de m a k variables (columnas) de tal forma que mantengan la mayor cantidad de información, siendo $k < m$. Para calcular las componentes principales se sigue los siguientes pasos:

1. Cálculo la matriz de covarianza (Ecuación 1.10)

$$\Sigma_{m \times m} = \begin{bmatrix} \text{Var}(X_{nx1}) & \text{Cov}(X_{nx1}, X_{nx2}) & \dots & \text{Cov}(X_{nx1}, X_{nxm}) \\ \text{Cov}(X_{nx2}, X_{nx1}) & \text{Var}(X_{nx2}) & \dots & \text{Cov}(X_{nx2}, X_{nxm}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(X_{nxm}, X_{nx1}) & \text{Cov}(X_{nxm}, X_{nx2}) & \dots & \text{Var}(X_{nxm}) \end{bmatrix} \quad (1.10)$$

donde:

$$\text{Var}(X_{nxm}) = \frac{\sum_{i=1}^n (X_{ixm} - \bar{x})^2}{n-1} \rightarrow \text{Varianza}$$

$$\bar{x} = \frac{\sum_{i=1}^n X_{ixm}}{n-1} \rightarrow \text{Promedio}$$

2. Cálculo el polinomio característico de la matriz de covarianza (Ecuación 1.11)

$$\text{Polinomio}^m(\lambda) = \det(\Sigma - \lambda I) \text{ donde } I_{m \times m} = \begin{bmatrix} 1_{11} & \dots & 0_{1m} \\ \vdots & \ddots & \vdots \\ 0_{m1} & \dots & 1_{mm} \end{bmatrix} \quad (1.11)$$

3. Cálculo de los valores propios del polinomio característico λ_i con $i = 1, \dots, m$

4. Cálculo de los vectores propios a partir de la resolución un sistema de ecuación de m variables para cada valor propio. (Ecuación 1.12)

$$[\Sigma_{m \times m}][V_{m \times 1}] = \lambda_i [V_{m \times 1}] \text{ con } i = 1, \dots, m \quad (1.12)$$

Cada vector propio $V_{i_{m \times 1}}$, de magnitud λ_i , con $i = 1, \dots, m$, corresponde a i componentes principales, de las cuales se requiere utilizar los k vectores propios siendo $k < m$, que contengan la mayor cantidad de información sobre el conjunto de datos. Se usa los k vectores propios de mayor magnitud como componentes principales.

Mediante la proyección de los n vectores fila del conjunto original, sobre el plano que forman los k vectores propios, se encuentra una nueva matriz $Y_{n \times k}$, que corresponde a la matriz reducida en dimensionalidad que contiene la mayor cantidad de información.

1.3.6 CLASIFICACIÓN

La clasificación es el proceso de identificar la categoría a la cual pertenece una nueva observación dentro de un conjunto de clases conocidas. Las tareas propias de la clasificación pueden ser llevadas por diferentes métodos, desarrollas bien sea por métodos estadísticos o por inteligencia artificial. Existe gran variedad de métodos, que pueden ser utilizados dentro de la clasificación de gestos de la mano como:

- Método K vecinos más próximos (KNN)
- Métodos jerárquicos
- Método K vecinos más lejanos (KFN)
- Método de la media (UPGMA)
- Método del centroide
- Método de la mediana
- Método de Ward
- Método flexible de Lance y Williams
- Método basado en redes neuronales artificiales
- Máquinas de vectores de soporte (SVM)

En este proyecto se han puesto a prueba dos métodos de clasificación, que se diferencian ampliamente en la complejidad de su implementación. A continuación se describe a detalle los 2 métodos utilizados.

1.3.6.1 Distancia euclidiana al centroide más cercano

La distancia euclidiana es la distancia entre dos puntos deducida a partir del teorema de Pitágoras.

La distancia euclidiana entre dos puntos P1 y P2, de coordenadas cartesianas (x1, y1) y (x2, y2) viene dado por la Ecuación 1.13.

$$de = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1.13)$$

En un espacio multidimensional, la distancia euclidiana viene dada por la Ecuación 1.14.

$$de = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + \dots + (z_n - z_{n-1})^2} \quad (1.14)$$

La clasificación se realiza calculando la distancia euclidiana entre una nueva muestra ingresado, respecto a todos los centroides cálculos mediante un promedio que representa a cada una de las clases. La figura 1.17 muestra un ejemplo del método de clasificación centroide con distancia euclidiana para 2 dimensiones.

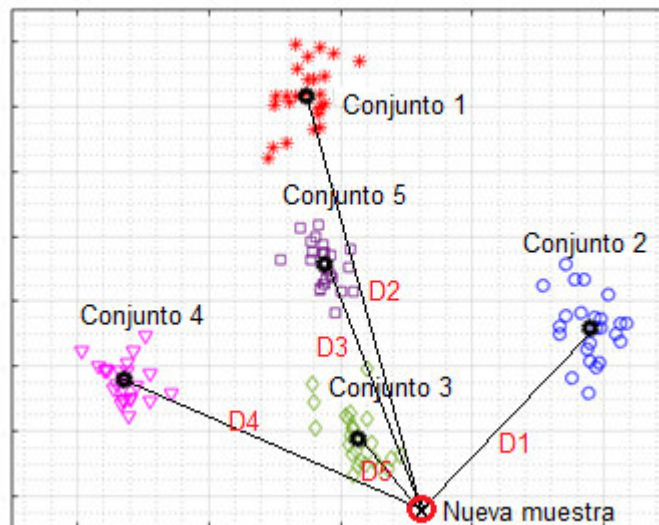


Figura 1.17. Método centroide con distancia euclidiana

Como se puede apreciar en la figura 1.17, cada una de las clases está ubicada en una zona específica del plano. Los puntos encerrados en rojo son los centroides, calculados al hallar el promedio de las muestras que conforman el conjunto. Si una nueva muestra aparece en el plano, se calcula la distancia euclidiana de dicha muestra con cada uno de los centroides y de esta manera al comparar las distancias obtenidas, la menor distancia conseguida, identifica el conjunto al cual pertenece. Para el ejemplo de la figura 1.17, existen 5 distancias euclidianas, calculadas entre la nueva muestra y cada uno de los

centroides (D1, D2, D3, D4, D5). De las cinco distancias, D5 es la menor y por lo tanto la nueva muestra pertenece al conjunto 3.

1.3.6.2 Redes neuronales artificiales (MLP)

Una red neuronal es un modelo aplicativo en sistemas computacionales basado en el funcionamiento de las redes neuronales cerebrales. Es un campo dentro de la inteligencia artificial con las siguientes características:

- Su unidad base es la neurona artificial
- Realiza aprendizaje supervisado
- Requiere datos de entrada y salida

La representación gráfica de una neuronal artificial según McCulloch y Pitts (pioneros en redes neuronales artificiales) es la mostrada en la figura 1.18.

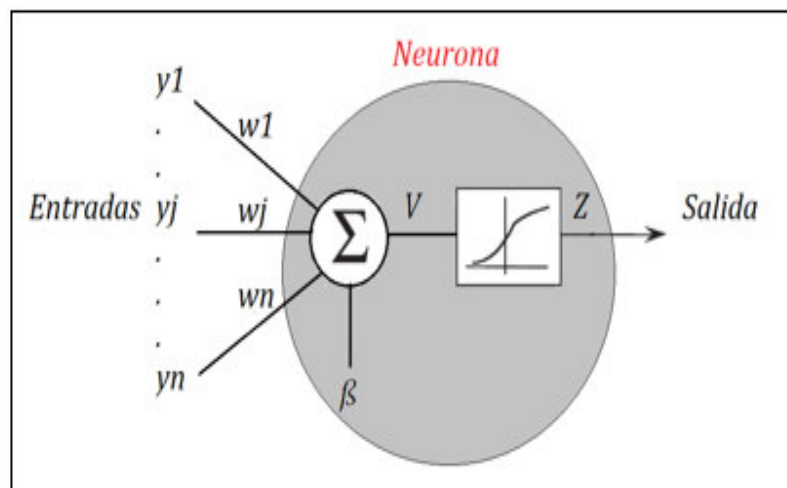


Figura 1.18. Neurona artificial

donde:

y_n : entradas

w_n : pesos

V : función de malla

Z : función de activación

β : bias

La función de malla determina la combinación de las entradas según algunos tipos de funciones, la tabla 1.2, muestra algunas funciones de malla utilizadas.

Tabla 1.2. Tipos de funciones de malla

Función de malla	Expresión
Lineal ponderada (weighted)	$V = \sum_{j=1}^n w_j y_j + \beta$
Lineal ponderada de segundo orden	$V = \sum_{j=1}^n \sum_{k=1}^n w_{jk} y_j y_k + \beta$
Delta	$V = \prod_{j=1}^n w_j y_j$

La función de activación es la encargada de devolver valores de salida a partir de los valores de entrada de la función de malla, normalmente los valores de salida están en un rango determinado, similar a las compuestas lógicas, AND, OR, NOT y sus combinaciones.

Algunos ejemplos de funciones de activación se especifican en la tabla 1.3.

Tabla 1.3. Tipos de funciones de activación

Función de activación	Expresión
Sigmoidal unipolar o logarítmica	$Z = f(V) = \frac{1}{1 + e^{-V}}$
Sigmoidal bipolar o hiperbólica	$Z = f(V) = \frac{1 - e^{-V}}{1 + e^{-V}}$
Tangencial hiperbólica	$Z = f(V) = \tanh(V)$
Tangencial inversa	$Z = f(V) = \frac{2}{\pi} \tan^{-1}\left(\frac{V}{T}\right)$
Threshold	$Z = f(V) = \begin{cases} 1 & V > 0 \\ -1 & V < 0 \end{cases}$
Gaussiana	$Z = f(V) = e^{-\frac{\ V-m\ ^2}{\sigma^2}}$
Lineal	$Z = f(V) = aV + b$
Softmax	$Z = f(V_i) = \frac{e^{V_i}}{\sum_{j=0}^J e^{V_j}}$

Algunas aplicaciones en donde se utiliza las redes neuronales son: el aprendizaje supervisado y no supervisado, la clasificación, la regresión, el reconocimiento de patrones y el clustering [20].

El modelo más utilizado para el análisis con redes neuronales artificiales es el modelo perceptrón multicapa (MLP) que utiliza un algoritmo de retropropagación (backpropagation) con minimización de función de costo para su entrenamiento.

Algunos algoritmos utilizados en la retropropagación son:

- Trainlm (Levenberg-Marquardt Backpropagation)
- trainbfg (BFGS Quasi-Newton Backpropagation)
- trainrp (Resilient Backpropagation)
- trainscg (Scaled Conjugate Gradient Backpropagation)
- traincgb (Conjugate Gradient with Powell/Beale Restarts)
- traincgf (Fletcher-Powell Conjugate Gradient)
- traincgp (Polak-Ribière Conjugate Gradient)
- trainoss (One Step Secant)
- traingdx (Variable Learning Rate Backpropagation)

El modelo perceptrón multicapa consiste en una red neuronal en capas, cada capa está formada por un conjunto de neuronas dentro de tres capas diferentes: La capa de entrada, que se encarga de propagar los patrones de entrada hacia todas las neuronas. La capa oculta, que realiza un procesamiento no lineal de los patrones recibidos y puede estar formada por una o algunas capas internas y la capa de salida, que proporciona una respuesta de acuerdo a los patrones de entrada. La figura 1.19, muestra la arquitectura del modelo perceptrón multicapa.

donde:

K= entradas

M= número de neuronas de la capa oculta 1

H= número de neuronas de la capa oculta 2

N= número de neuronas de la capa de salida

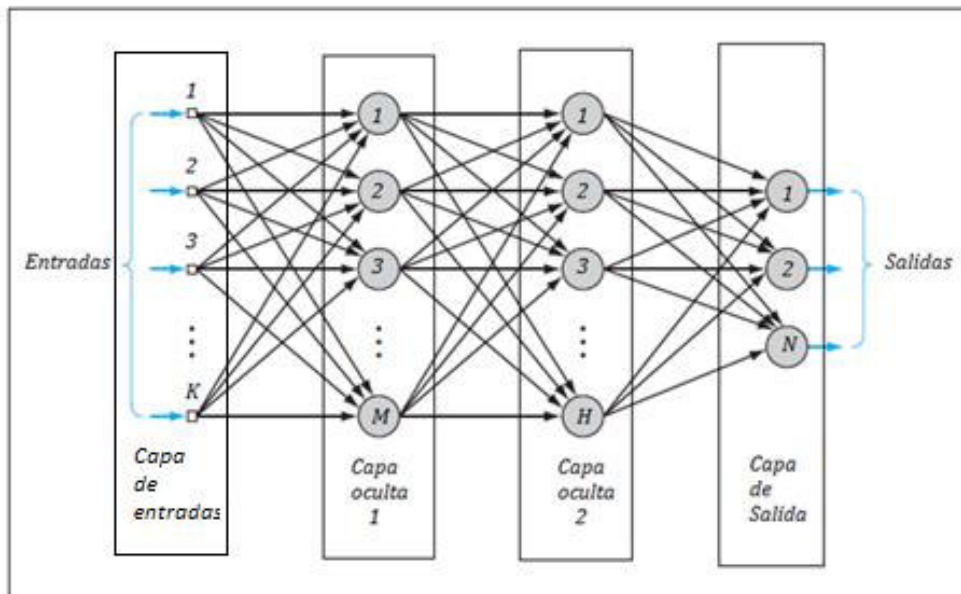


Figura 1.19. Modelo perceptrón multicapa

2. METODOLOGÍA

En este capítulo se presenta el diseño del algoritmo de reconocimiento de gestos de la mano. Este algoritmo está basado en el nivel de activación de las señales producidas por las contracciones musculares del antebrazo, medidas con 8 diferentes sensores distribuidos alrededor del antebrazo.

2.1 CONDICIONES DE FUNCIONAMIENTO

Para realizar el reconocimiento de gestos de la mano se debe cumplir los siguientes requisitos:

- Se recomienda colocar el brazalete a una distancia desde el codo al Myo Armband de aproximadamente 6 cm.
- El giro de colocación del brazalete puede estar en cualquier posición.
- Si es necesario se debe asegurar un buen ajuste del brazalete en el antebrazo del usuario.
- Antes de iniciar con las pruebas, se debe realizar una configuración rápida al inicio de cada sesión.
- No se debe realizar los gestos demasiado rápido, ni demasiado lento.
- Si se desea utilizar el sistema con el brazo estirado, se recomienda no aplicar fuerza en el estiramiento, ya que se detecta señales EMG no deseables que afectan el reconocimiento gestos de la mano.

Algunas características del sistema:

- Se puede mantener un gesto el tiempo necesario por el usuario y se seguirá mostrando el gesto realizado.
- No afecta la posición de la mano (horizontal o vertical).

2.2 COMUNICACIÓN ENTRE MYO ARMBAND Y LA PC

La comunicación entre Myo armband y la PC con Windows 10, se realiza de forma inalámbrica mediante el protocolo Bluetooth Low Energy (BLE). Este protocolo está diseñado para proporcionar un bajo consumo de energía y mantener un rango de alcance similar a Bluetooth tradicional (10 m). Las principales características de Bluetooth Low Energy son:

- Opera en la banda ISM de 2.4 GHz
- BLE permanece en suspensión constante, excepto cuando se inicia una conexión.
- Los tiempos de conexión son de pocos milisegundos
- Tiene una velocidad de emisión y transferencia de 32 Mbps

La figura 2.1, muestra el esquema de conexión utilizado para la adquisición de las señales EMG



Figura 2.1. Esquema de conexión

2.2.1 SOFTWARE PARA LA COMUNICACIÓN ENTRE MYO ARMBAND Y LA PC

Los programas y librerías necesarios para la comunicación entre Myo y la PC son los siguientes:

- Myo_connect
- Librería MyoMex-master
- Librería Myo-sdk
- Matlab 2020a
- Mingw

La figura 2.2, muestra el software necesario para la adquisición de señales EMG



Figura 2.2. Software para la adquisición de señales EMG

2.2.1.1 Matlab

Matlab es un sistema de cálculo numérico, con lenguaje de programación propio (lenguaje .m), disponible para Windows, GNU/Linux, MacOs y Unix.

Entre sus principales características se encuentran:

- Manipulación de matrices
- Representación de datos y funciones
- Procesamiento de señales
- Herramientas para inteligencia artificial
- Programación orientada a objetos
- etc.

Se ha decidido utilizar Matlab debido a la gran cantidad de herramientas para el procesamiento y clasificación de señales y su amplia documentación sobre el funcionamiento de sus instrucciones.

El software Matlab recomendado para este proyecto, es cualquier versión superior a la 2015a. Para versiones anteriores las librerías para la comunicación entre el Myo y Matlab no están disponibles.

2.2.1.2 Mingw

MinGW-w64 es un conjunto de compiladores para Windows basado en la cadena de herramientas GNU. Incluye un compilador GCC y herramientas relacionadas para compilar aplicaciones C y C ++ para Windows. Las aplicaciones C y C ++ compiladas con MinGW-w64 GCC se pueden llamar desde MATLAB usando MEX (Archivos MEX se refieren a las secuencias de comandos y comandos necesarios para la automatización de las tareas comunes que se lleva a cabo en los sistemas operativos Windows y otros programas [21]).

La instalación de Mingw se realizó mediante Add/Ons, disponible dentro de la barra de herramientas Home de Matlab. Se debe seguir los siguientes pasos:

- 1) Ingresar a Add/Ons para la instalación del toolbox de Matlab (figura 2.3).

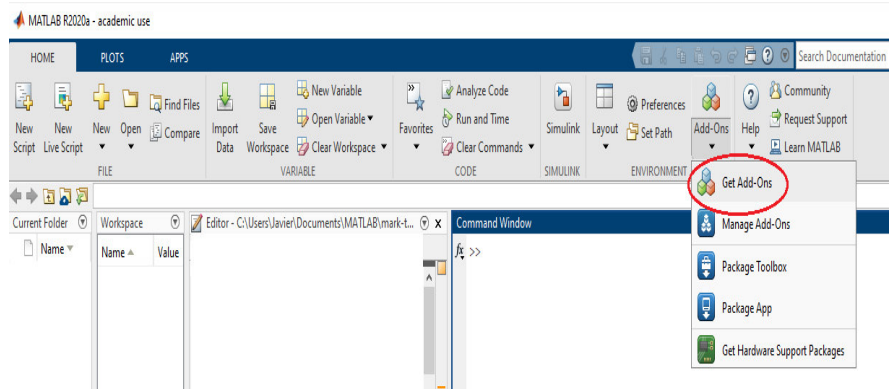


Figura 2.3. Menú para la instalación de Mingw

2) Buscar e instalar MinGW-w64 (figura 2.4).



Figura 2.4. Compilador Mingw

2.2.1.3 Myo Connect Installer

Myo connect Installer es un software ejecutable creado por Thalmic Labs, que permite la interacción entre el brazalete Myo y la PC. Este software ofrece las siguientes funcionalidades:

- Permite el reconocimiento de 5 gestos de la mano: Fist, Wave in, Wave out, Open y Pinch.
- Permite controlar múltiples tareas de navegación y acceso a youtube, netflix y control sobre el mouse.
- Permite la creación de perfiles para cada usuario que utilice el dispositivo.
- Permite visualizar el nivel de carga del Myo.
- Permite activar y desactivar el brazalete.
- Permite realizar ping desde la PC al Myo para verificar si existe comunicación.

Este software se puede descargar de la siguiente página web:

<https://myo-connect.software.informer.com/0.9b/>.

Para sincronizar el brazalete con la pc, se debe buscar y ejecutar el programa instalado Myo Connect, como se muestra en la figura 2.5. Con este paso el Myo se ha sincronizado con la PC.

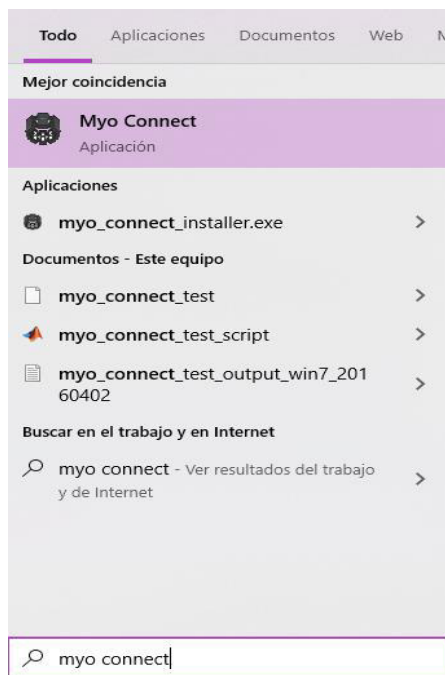


Figura 2.5. Sincronización de brazalete con PC

En los iconos ocultos de la barra de tareas de Windows, aparece un nuevo icono de myo como se muestra en la figura 2.6.

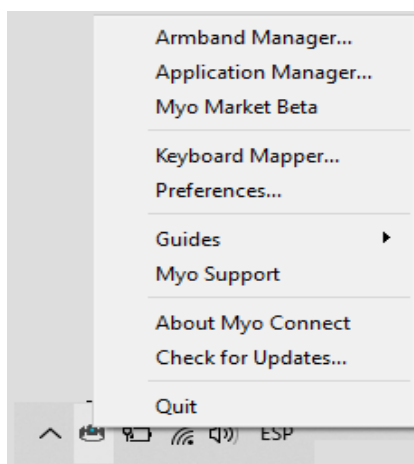


Figura 2.6. Opciones de myo connect

Para ingresar a las opciones del icono, se debe dar clic derecho y se podrá realizar las siguientes opciones:

- Quitar la conexión de Myo con la PC (Quit)
- Verificar la ventana de control del brazalete (Armband Manager). Figura 2.7.

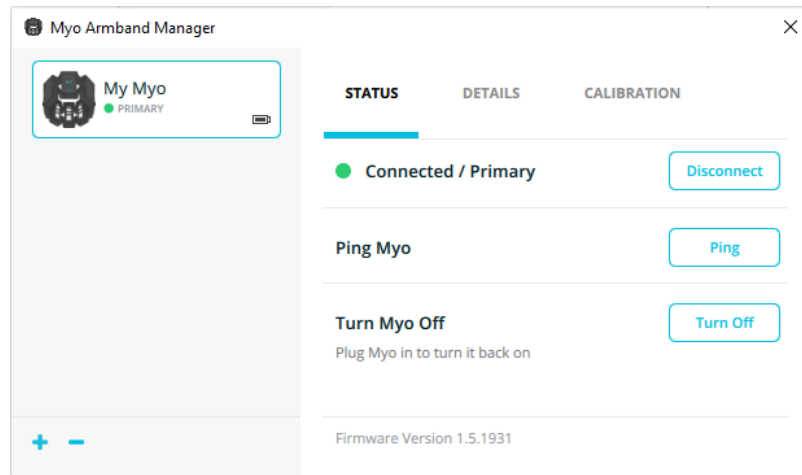


Figura 2.7. Ventana de control del brazalete

2.2.1.4 Myo Mex master y Myo SDK

El brazalete Myo cuenta con una unidad de medida inercial IMU (informa sobre la velocidad, orientación usando acelerómetros y giroscopios). Sensores EMG (para adquisición de señales) y un SDK (Software Development Kit), herramienta para crear aplicaciones específicas con Myo por medio de la modificación de sus librerías internas.

La carpeta Myo Mex master contiene una clase de código .m simplificada. Estas librerías permiten transmitir datos desde uno o dos dispositivos Myo a una PC con una frecuencia de muestreo de 50 Hz para IMU y 200 Hz para EMG.

Para descargar todas las librerías se lo puede hacer a través de la siguiente dirección electrónica:

<https://github.com/mark-toma/MyoMex>

Una vez descargado se procede a descomprimir, guardar en la unidad C del disco y añadir las librerías al path de Matlab como se muestra en las figuras 2.8 y 2.9.

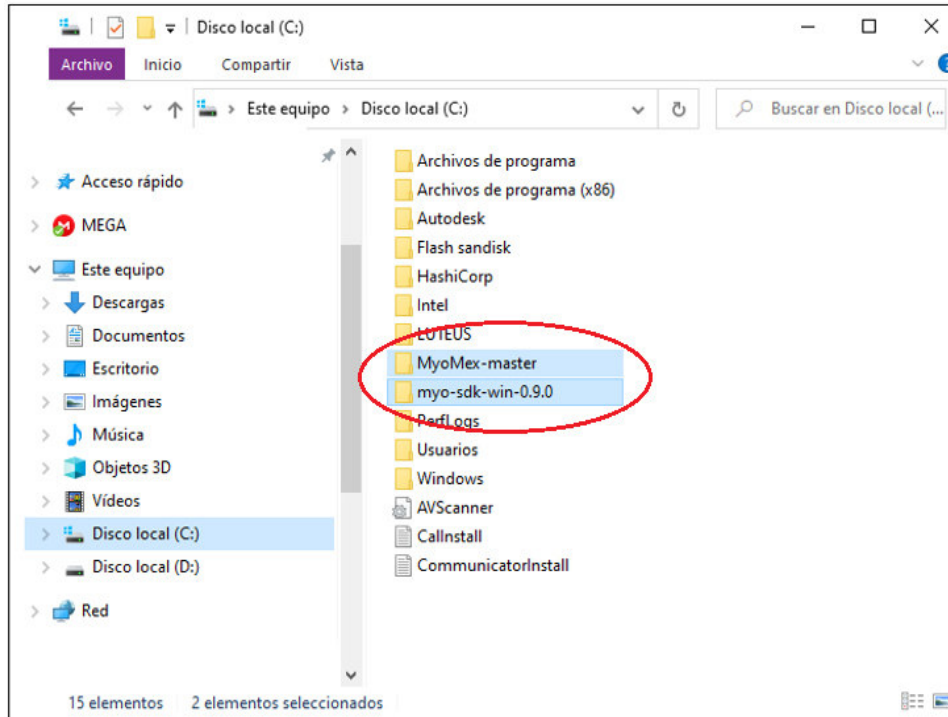


Figure 2.8. Librerías Myo Mex y SDK descargadas y guardadas en el disco C de la PC

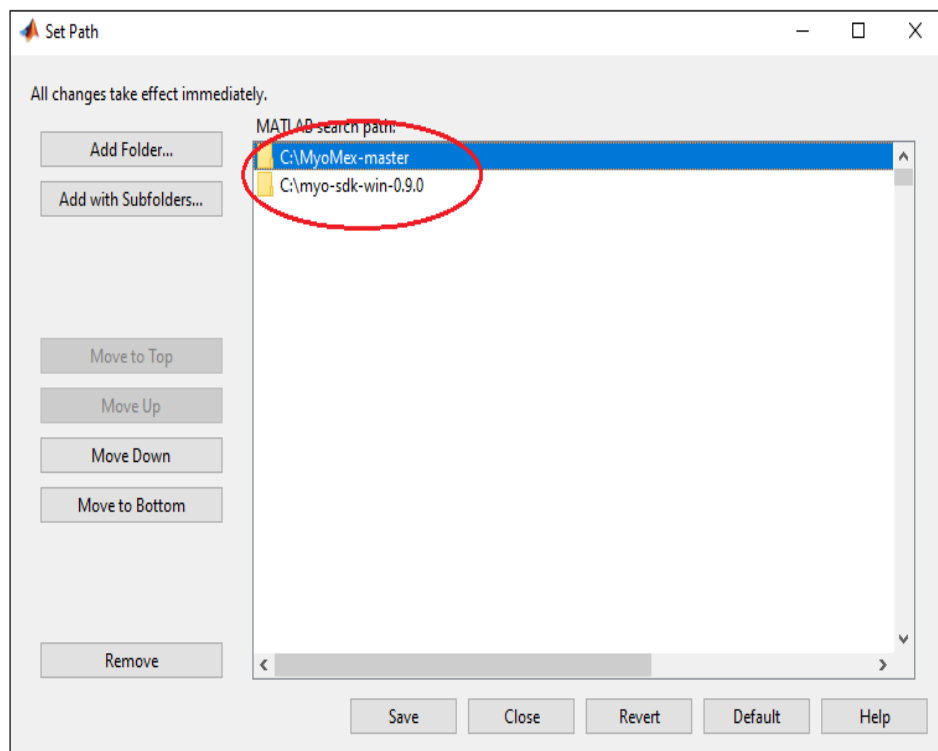


Figure 2.9. Librerías Myo Mex y SDK descargadas y guardadas en el path de Matlab

2.2.2 INSTALACIÓN Y EJECUCIÓN DE LAS LIBRERÍAS

Las librerías Myo Mex deben ser correctamente instaladas en el computador. Para ello se sigue los siguientes pasos.

1) Ir a sistema, por medio del buscador de Windows, como se muestra en la figura 2.10.

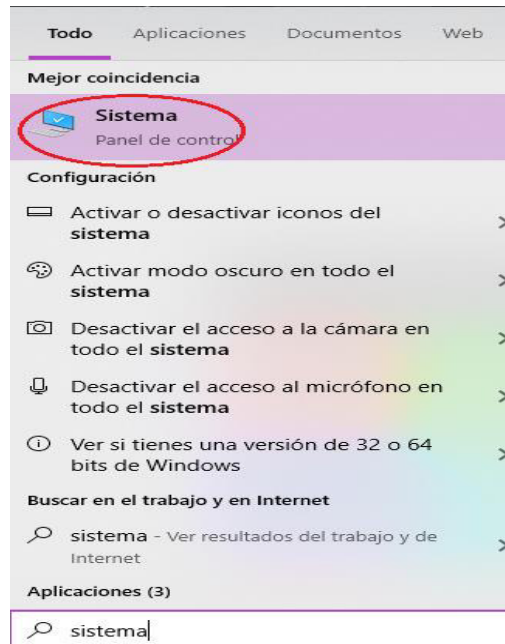


Figura 2.10. Buscador de sistema en Windows 10

2) En sistema y se procede a elegir configuración avanzada de sistema. Figura 2.11.

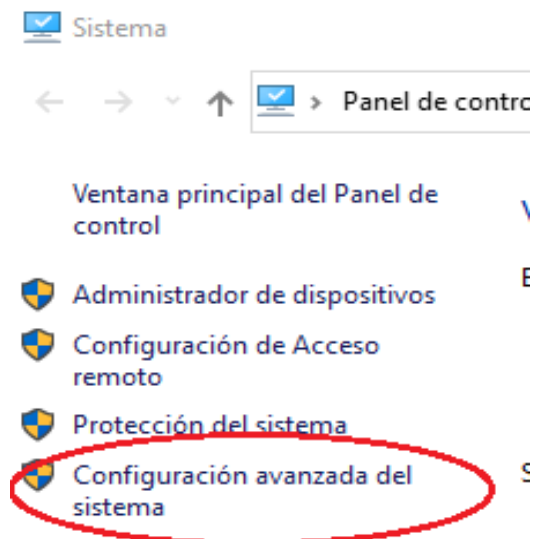


Figura 2.11. Selección de configuración avanzada del sistema

3) En propiedades del sistema y se procede a elegir variables de entorno

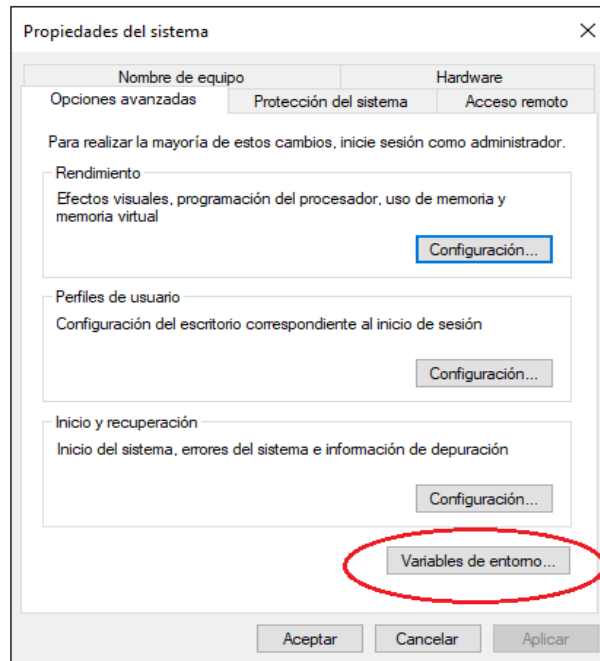


Figura 2.12. Ventana de propiedades del sistema

4) En variables de entorno se procede a seleccionar path e ingresar a editar (figura 2.13).

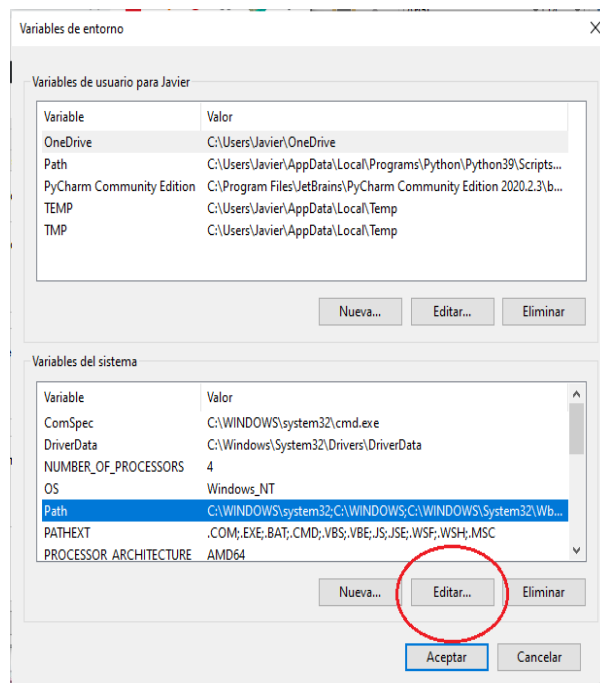


Figura 2.13. Opción para editar path en el sistema

5) En editar se va a crear un nuevo path con la ubicación donde se encuentra la carpeta myo-sdk y se procede a aceptar todas las ventanas (figura 2.14).

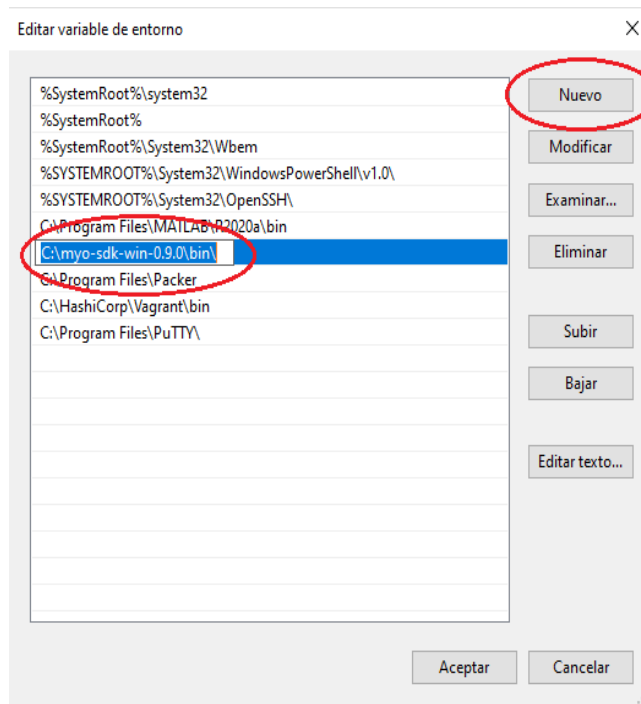


Figura 2.14. Agregación de Path de Myo-Sdk

El siguiente paso es colocarse el brazaletes y abrir la función `install_my_mex.m` en Matlab, ubicada en el disco local C, en la carpeta de MyoMex-master y ejecutar.

Una vez realizado el punto anterior, se procede a abrir el archivo `MyoMex_Quickstart.m` ubicado en la misma carpeta MyoMex-master y se procede a ejecutar,

Al finalizar la ejecución anterior, se debe crear un archivo `.m`, donde se desarrolla el algoritmo para el reconocimiento de gestos de las mano.

2.2.3 PROGRAMA INICIAL DE LECTURA

Al ingresar a Matlab, se crea un nuevo script con las instrucciones requeridas para establecer la comunicación entre Myo armband y Matlab y se crea un bucle infinito `While`, donde se implementa el algoritmo en tiempo real para el reconocimiento de gestos de la mano, tal como se muestra en el siguiente código de programación.

```

clc
clear
close all
% CONSTANTES PARA LA ADQUISIÓN DE SEÑALES (MATLAB - MYO ARMBAND)
countMyos=1;
m = MyoMex(countMyos); % propiedades de la instancia MyoMex
m1 = m.myoData(1); % objetos para cada uno de los Myos
m1.timeEMG;
m1.emg;
m1.stopStreaming();
m.myoData.clearLogs();
m1.startStreaming();
emg=zeros(40,8);
order=1;
runMYO=1;
ventana=2; % tamaño de ventana de 400 muestras
%INICIO DEL BUCLE INFINITO
while runMYO==1
    timeEMG = m1.timeEMG_log;
    T_emg=timeEMG(:,1)>=(timeEMG(end,1)-ventana);
    emg=m1.emg_log(T_emg,WM_X(order));
%[SIGNALS] ES LA MATRIZ QUE CONTIENE LAS SEÑALES EMG, su tamaño es 400x8
    assignin('base','S',emg(:,,:))
SECCIÓN DONDE SE IMPLEMENTA EL ALGORITMO DE RECONOCIMIENTO DE GESTOS DE LA MANO EN TIEMPO REAL
end

```

2.3 ALGORITMO DE RECONOCIMIENTO DE GESTOS

En este trabajo se realiza un análisis en tres dominios (tiempo, frecuencia y complejidad) para la etapa de extracción de características como puede verse en la figura 2.15. Los tres dominios utilizados emplean la misma etapa de extracción de señales; el dominio del tiempo emplea 3 algoritmos de preprocesamiento y como variable el área bajo las curvas de las envolventes suavizadas. El dominio de la frecuencia, emplea el área de densidad espectral de potencia suavizada de las señales y el dominio de la complejidad utiliza la dimensión fractal calculada con el algoritmo de Katz.

EXTRACCIÓN DE SEÑALES			PREPROCESAMIENTO			EXTRACCIÓN DE CARACTERÍSTICAS	
DETERMINACIÓN ADAPTATIVA DE RUIDO	DETECCIÓN DE INICIO DE GESTO	VENTANA DE ANÁLISIS	EXTRACCIÓN DE SEÑAL ÚTIL				DIMENSIÓN FRACTAL DE KATZ
				RECTIFICACIÓN DE ONDA COMPLETA	ENVOLVENTE	SUAVIZADO DE CURVAS	AREA
							AREA DE DENSIDAD ESPECTRAL DE POTENCIA SUAVIZADA

Figure 2.15. Algoritmo de reconocimiento de 5 gestos basado en señales de EMG

2.3.1 ETAPA DE EXTRACCIÓN DE SEÑALES EMG

El modelo propuesta se desarrolla a partir de la conexión creada entre Myo Armband y Matlab, en donde obtenemos 8 señales de EMG, modeladas como una serie temporal, representadas por la matriz $S_{N \times 8} = (S1, \dots, S8) \in \mathbb{R} [-1,1]$ donde 8 corresponde a las 8 señales obtenidas de los 8 canales del dispositivo (figura 2.16), con vectores columna $\vec{S}_i = (S_i(1), \dots, S_i(N)) \in \mathbb{R} [-1,1]$ con $i = 1, \dots, 8$.

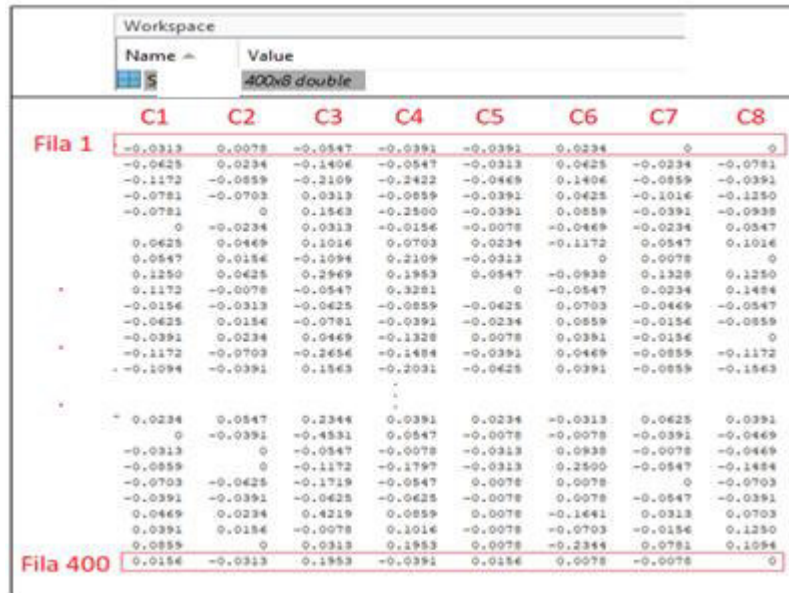


Figura 2.16. Matriz de adquisición de señales S

El valor N, corresponde al tamaño de los vectores columna que contienen los valores de amplitud de las señales, la cual se configura manualmente. La configuración de la ventana de análisis dentro del programa inicial se lo realiza en la constante VENTANA. La tabla 2.1, muestra el tamaño de ventana configurable y la cantidad de valores obtenidos en todas las columnas de la matriz $S_{N \times 8}$.

Tabla 2.1. Ventanas configurables

Ventana	N
1	200
2	400
3	600
4	800
5	1000

Para este trabajo, utilizamos una ventana de tamaño $N=400$ valores, en donde los 400 valores de la matriz $S_{400 \times 8}$ se actualizan en cada iteración del algoritmo. La actualización de la matriz se produce mediante una ventana deslizante de 40 valores aproximadamente de cada columna, partiendo desde la muestra N, en saltos de 40 valores hasta la muestra

0. Cabe mencionar que en cada actualización de la matriz S , se puede observar como inician, como se desplazan y como desaparecen las señales EMG, desde la muestra N hasta la muestra 0, como se observa en cada iteración del algoritmo en la figura 2.17.

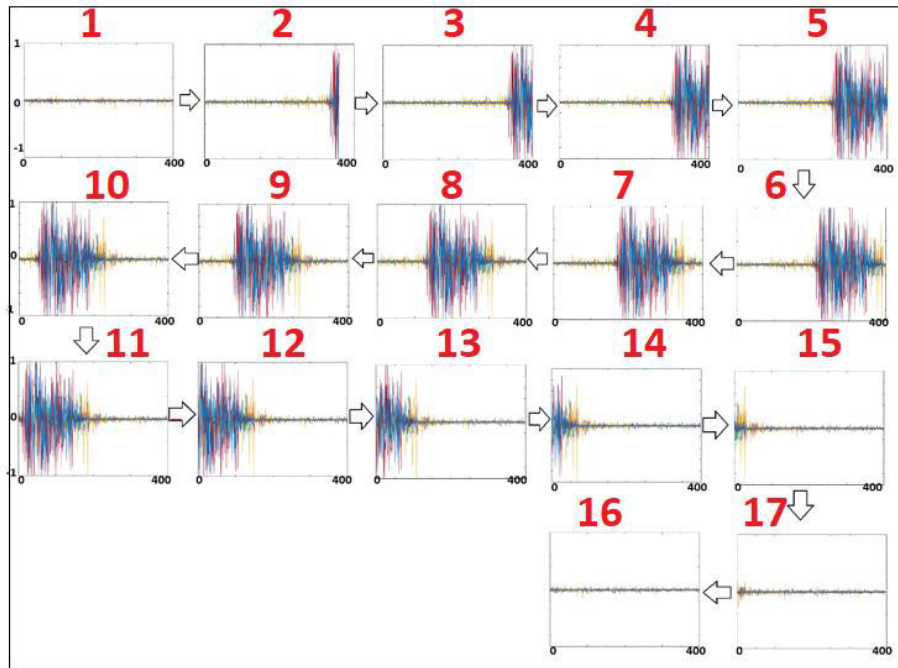


Figure 2.17. Ventanas de adquisición de señales

2.3.1.1 Fase 1: Determinación adaptativa de nivel ruido

En esta fase se realiza una separación de las columnas de la matriz $S_{400 \times 8}$, para aplicar un análisis individual a cada señal, debido a que cada canal presenta diferentes niveles de ruido que pueden afectar la detección de los gestos en el momento de su realización. La figura 2.18, muestra la separación de las señales y el distinto nivel de ruido para cada canal.

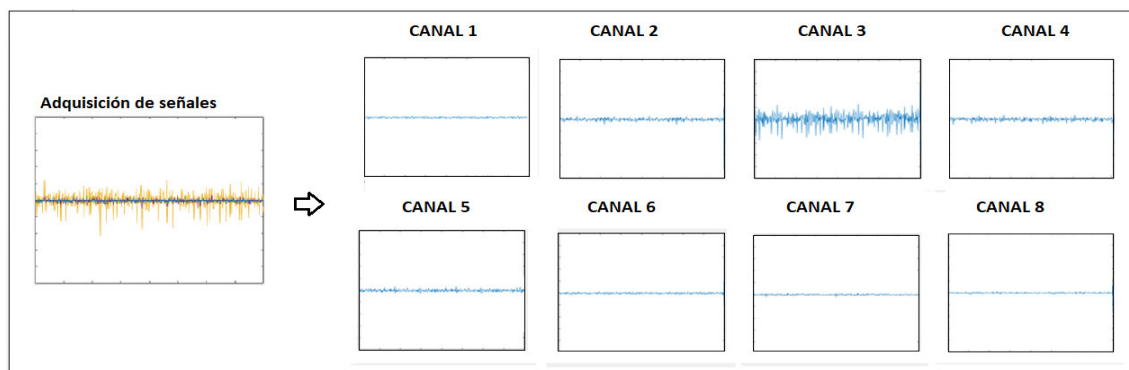


Figura 2.18. Separación de señales

A continuación se muestra la función creada en Matlab, para obtener cada columna de la matriz S para un análisis individual.

```
function [c1,c2,c3,c4,c5,c6,c7,c8] = separacion(S,i)
% obtención de la columna 1, que corresponde la señal EMG de cada canal
c1{i}=S(:,1);
c2{i}=S(:,2);
c3{i}=S(:,3);
c4{i}=S(:,4);
c5{i}=S(:,5);
c6{i}=S(:,6);
c7{i}=S(:,7);
c8{i}=S(:,8);
end
```

En la experimentación se encontró que el nivel de ruido varía en cada canal según el giro de colocación del brazalete y otras variables no controlables que afectan la detección de los gestos realizados, por ello se requiere el cálculo del valor máximo por columna en la matriz $S_{400 \times 8}$, cuando un usuario no realiza ningún movimiento, para obtener un vector fila $\vec{S}_{1 \times 8}$, con los valores de ruido máximo por canal. Esto permite establecer un umbral superior al doble del valor máximo de ruido en cada canal del dispositivo, que sirve de límite para determinar si un usuario ha realizado o no un movimiento. El cálculo del umbral se realiza al inicio de cada sesión en una configuración rápida donde se adquieren 3 muestras de cada gesto y se determina el máximo nivel de ruido por cada canal y de esta manera establecer los umbrales al doble del valor máximo nivel de ruido de cada señal, tal como se puede observar en la figura 2.19.

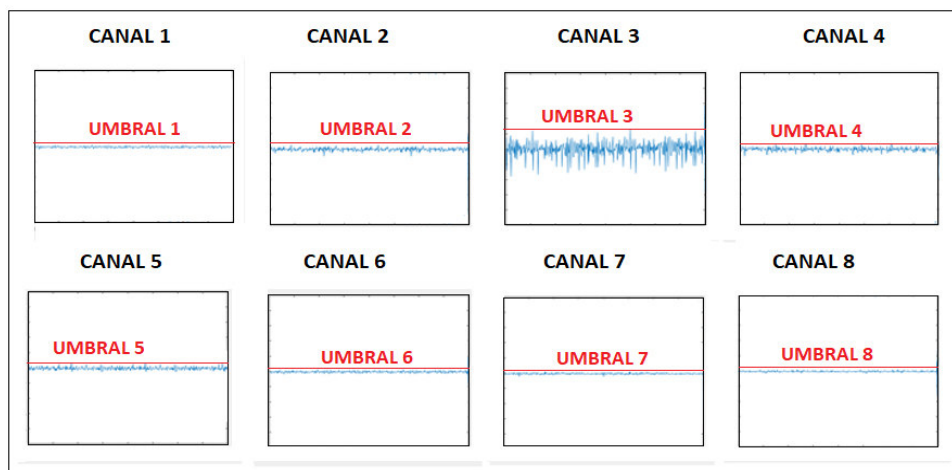


Figura 2.19. Umbrales para la detección de gesto

A continuación se muestra la función creada en Matlab para determinar el umbral en cada canal.

```

function
[um1,um2,um3,um4,um5,um6,um7,um8]=ruido_adaptativo(c1,c2,c3,c4,c5,c6,c7,c8,i)
% VENTANA DONDE SE ANALIZA EL MÁXIMO NIVEL DE RUIDO EN CADA CANAL
% Se obtiene una muestra de 200 donde se analiza el nivel máximo de ruido en
cada canal.
gp1=c1{i}(1:length(c1{i})-ve);
gp2=c2{i}(1:length(c2{i})-ve);
gp3=c3{i}(1:length(c3{i})-ve);
gp4=c4{i}(1:length(c4{i})-ve);
gp5=c5{i}(1:length(c5{i})-ve);
gp6=c6{i}(1:length(c6{i})-ve);
gp7=c7{i}(1:length(c7{i})-ve);
gp8=c8{i}(1:length(c8{i})-ve);
% HALLAR EL MÁXIMO VALOR DE RUIDO EN CADA CANAL
um_1=max(gp1);
um_2=max(gp2);
um_3=max(gp3);
um_4=max(gp4);
um_5=max(gp5);
um_6=max(gp6);
um_7=max(gp7);
um_8=max(gp8);
% OBTENER UN UMBRAL DE CADA CANAL SEGÚN SU NIVEL DE RUIDO QUE SEA
% EL DOBLE DEL MÁXIMO VALOR DE RUIDO
to=2;
um1=um_1*to;
um2=um_2*to;
um3=um_3*to;
um4=um_4*to;
um5=um_5*to;
um6=um_6*to;
um7=um_7*to;
um8=um_8*to;
end

```

2.3.1.2 Fase 2: Detección de inicio de gesto

Con el umbral establecido para cada canal, el programa identifica si un usuario ha realizado un gesto, cuando en cada iteración del algoritmo, se detectan valores de amplitud que superen el umbral calculado, al analizar cada columna de la matriz $S_{400 \times 8}$, lo que indica que se trata del inicio de las señales EMG. La figura 2.20, muestra un ejemplo de detección de gesto.

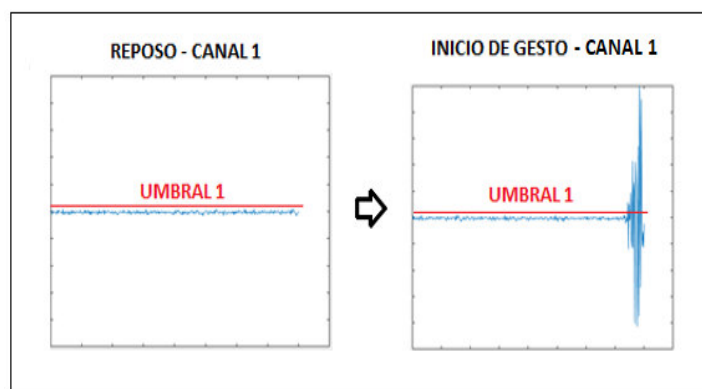


Figura 2.20. Detección de gesto

2.3.1.3 Fase 3: Ventana de análisis

La ventana de análisis corresponde al número de iteración del algoritmo, desde la detección de inicio de gesto, que contiene la cantidad información suficiente de las señales EMG exclusivamente de activación, para su análisis.

Ya que en cada iteración del algoritmo, la matriz $S_{400 \times 8}$ actualiza sus valores, se requiere que una de estas iteraciones contenga como mínimo 160 valores solo de EMG de activación, para su posterior análisis en el menor tiempo de procesamiento. Es decir cuando en la matriz $S_{400 \times 8}$, el inicio de las señales EMG de activación de la ventana deslizante se encuentre en la muestra 240, $\{\vec{S}_i = (S_i(240), \dots, S_i(400)) \in \mathbb{R} [-1,1]$ con $i = 1, \dots, 8\}$, significa que se obtuvo la información suficiente de las señales EMG para su análisis. Por ello desde la detección de inicio de gesto, se espera hasta la 4ta iteración, para implementar las etapas de preprocesamiento, extracción de características y clasificación.

La figura 2.21 muestra el proceso de selección de la 4ta iteración, desde la detección de inicio de gesto (1ra iteración).

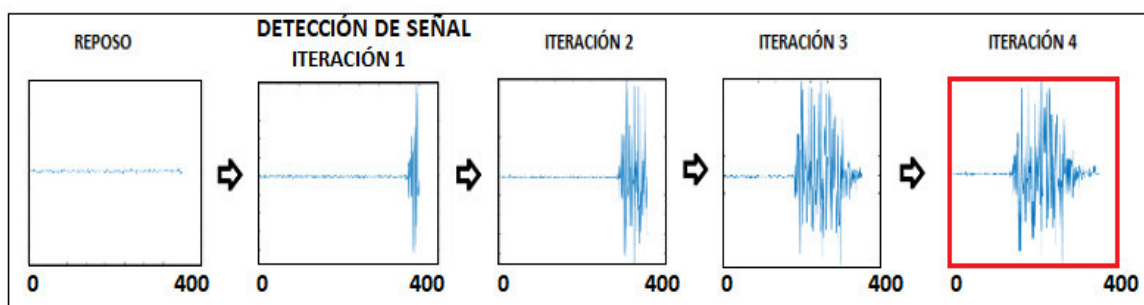


Figura 2.21. Utilización de la 4ta iteración

Es posible tomar las iteraciones siguientes, pero esto aumenta el tiempo de procesamiento, también es posible tomar iteraciones anteriores para disminuir el tiempo de procesamiento, pero se analizaría el gesto con menos información, lo que afecta en la etapa de clasificación.

La siguiente función creada en Matlab, realiza la detección de inicio de gesto y aplica un contador identificador de la 4ta iteración desde que se detecta la señal EMG.

```

function
[it]=umbral(c1,c2,c3,c4,c5,c6,c7,c8,i,it,Q,um1,um2,um3,um4,um5,um6,um7,um8)
% IGUALACIÓN DE MUESTRAS, AÑADIENDO CEROS PARA MANTENER VECTORES DE 400      % MUESTRAS EN
EL CASO QUE FALTARA EN CADA SEÑAL DE CADA CANAL
v=400;
t1=[c1{i};zeros(v-length(c1{i}),1)];
t2=[c2{i};zeros(v-length(c2{i}),1)];
t3=[c3{i};zeros(v-length(c3{i}),1)];
t4=[c4{i};zeros(v-length(c4{i}),1)];
t5=[c5{i};zeros(v-length(c5{i}),1)];
t6=[c6{i};zeros(v-length(c6{i}),1)];
t7=[c7{i};zeros(v-length(c7{i}),1)];
t8=[c8{i};zeros(v-length(c8{i}),1)];
% ENCUENTRA LOS VALORES MAYORES A CADA UMBRAL CALCULADO, EN CADA VENTANA DE 400 MUESTRAS Y
EN CADA ITERACION DEL BUCLE WHILE
p1=[0.01;t1(find(t1>um1))];
p2=[0.01;t2(find(t2>um2))];
p3=[0.01;t3(find(t3>um3))];
p4=[0.01;t4(find(t4>um4))];
p5=[0.01;t5(find(t5>um5))];
p6=[0.01;t6(find(t6>um6))];
p7=[0.01;t7(find(t7>um7))];
p8=[0.01;t8(find(t8>um8))];
% CÁLCULO UN PROMEDIO DE TODOS LOS VALORES MAYORES A SUS UMBRALES
E1=mean(p1);
E2=mean(p2);
E3=mean(p3);
E4=mean(p4);
E5=mean(p5);
E6=mean(p6);
E7=mean(p7);
E8=mean(p8);
% VERIFICA SI EL PROMEDIO DE VALORES DE LAS 400 MUESTRAS SUPERA A LOS      %
UMBRALES CALCULADOS, SI LA RESPUESTA ES POSITIVA, SIGNIFICA QUE SE TRATA      % DE UN
GESTO REALIZADO Y COMIENZA EL CONTEO PARA OBTENER LA VENTANA DE      % ANALISIS DE LA
CUARTA ITERACION
if( (E1>=um1) || (E2>=um2) || (E3>=um3) || (E4>=um4) || (E5>=um5) || (E6>=um6) || (E7>=um7) || (E8>=um8) )
    it=it+1;
else
    if Q==1 % condicional para solicitar realizar el gesto WAVE IN
        % pero únicamente en la configuración inicial
        imagen4=imread('WAVE_IN.jpg');
        imshow(imagen4);
        title('REALICE EL GESTO WAVE IN')
    elseif Q==2% condicional para solicitar realizar el gesto WAVE OUT
        % pero únicamente en la configuración rápida al inicio de sesión
        imagen5=imread('WAVE_OUT.jpg');
        imshow(imagen5);
        title('REALICE EL GESTO WAVE OUT')
    elseif Q==3% condicional para solicitar realizar el gesto FIST
        % pero únicamente en la configuración inicial
        imagen6=imread('FIST.jpg');
        imshow(imagen6);
        title('REALICE EL GESTO FIST')
    elseif Q==4% condicional para solicitar realizar el gesto OPEN
        % pero únicamente en la configuración inicial
        imagen7=imread('OPEN.jpg');
        imshow(imagen7);
        title('REALICE EL GESTO OPEN')

    elseif Q==5% condicional para solicitar realizar el gesto PINCH
        % pero únicamente en la configuración inicial
        imagen8=imread('PINCH.jpg');
        imshow(imagen8);
        title('REALICE EL GESTO PINCH')
    elseif Q==0
        imagen1=imread('NO_GESTURE.jpg');
        imshow(imagen1);
    end
end
end
end

```

2.3.1.4 Caso de gestos sostenidos

Una vez realizada la determinación de gesto, en donde se imprime en pantalla el gesto realizado, se implementa la fase de gestos sostenidos. Un gesto sostenido se produce cuando un usuario realiza un gesto y se mantiene en este un tiempo prolongado sin volver al reposo. La figura 2.22, muestra un ejemplo de una señal EMG cuando se sostiene un gesto, mostrando como se ve la señal en cada iteración del algoritmo desde su detección.

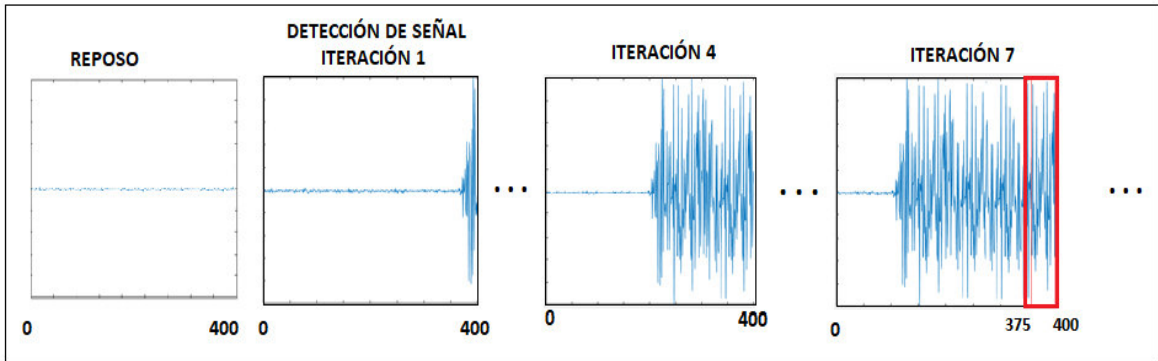


Figura 2.22. Movimientos sostenidos

La figura 2.23, muestra un ejemplo de una señal EMG cuando no se sostiene un gesto, es decir se vuelve al reposo.

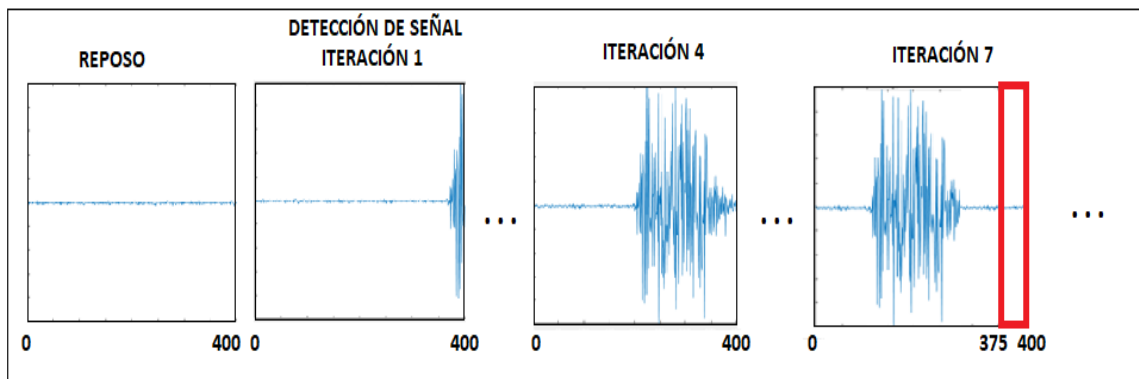


Figura 2.23. Movimientos sin sostenimiento

Cuando se sostiene un gesto, después de la 4ta iteración desde la detección de inicio de gesto, se muestran valores de amplitud de las señales que superan los umbrales calculados, como se observa en la figura 2.22. De esta manera, el programa verifica si en cada iteración del algoritmo, las 25 últimas muestras de los vectores $\vec{S}_i = (S_i(375), \dots, S_i(400))$ con $i = 1, \dots, 8$, superan los umbrales calculados y de esta manera imprime en pantalla el gesto clasificado en la 4ta iteración. El programa se mantiene imprimiendo el gesto clasificado hasta que el usuario vuelve al reposo donde se deja de

imprimir. La siguiente función creada en Matlab, realiza la verificación de gesto sostenido e imprime en pantalla el gesto clasificado en la 4ta iteración. Si las 25 últimas muestras no superan los umbrales calculados no se imprime ningún gesto.

```
function
[D1,D2,D3,D4,D5,D6,D7,D8,te,as]=sostenido(c1,c2,c3,c4,c5,c6,c7,c8,i,s,abc
d,fin,Q,FI,WI,WO,OP,PI)
% IMAGENES QUE SE MUESTRAN EN PANTALLA (ADICIONAL)
    imagen1=imread('NO_GESTURE.jpg');
    imagen2=imread('FIST.jpg');
    imagen3=imread('OPEN.jpg');
    imagen4=imread('WAVE_IN.jpg');
    imagen5=imread('WAVE_OUT.jpg');
    imagen6=imread('PINCH.jpg');
% IGUALACIÓN DE MUESTRAS, AÑADIENDO CEROS PARA MANTENER VECTORES DE 400
% MUESTRAS EN EL CASO QUE FALTARA EN CADA SEÑAL DE CADA CANAL
    val=400;
    tom1=[c1{i};zeros(val-length(c1{i}),1)];
    tom2=[c2{i};zeros(val-length(c2{i}),1)];
    tom3=[c3{i};zeros(val-length(c3{i}),1)];
    tom4=[c4{i};zeros(val-length(c4{i}),1)];
    tom5=[c5{i};zeros(val-length(c5{i}),1)];
    tom6=[c6{i};zeros(val-length(c6{i}),1)];
    tom7=[c7{i};zeros(val-length(c7{i}),1)];
    tom8=[c8{i};zeros(val-length(c8{i}),1)];
% OBTIENE LAS 25 ÚLTIMAS MUESTRAS DE LA VENTANA DE ANÁLISIS
% PARA VERIFICAR SI SE SIGUE MANTENIEDO EL GESTO
    valor=25;
    B1=tom1(val-valor:val);
    B2=tom2(val-valor:val);
    B3=tom3(val-valor:val);
    B4=tom4(val-valor:val);
    B5=tom5(val-valor:val);
    B6=tom6(val-valor:val);
    B7=tom7(val-valor:val);
    B8=tom8(val-valor:val);
% SE OBTIENE LOS VALORES DE LAS ÚLTIMAS 25 MUESTRAS QUE SUPERAN EL VALOR DE 0.3 ESTABLECIDO
% PARA DETERMINAR SI SE SIGUE REALIZANDO EL GESTO, PUDIERA SER LOS VALORES DE UMBRAL
% CALCULADOS PERO POR FACILIDAD SE ESTABLECIO ESTE VALOR
    num=0.3;
    C1=[0.1;B1(find(B1>num))];
    C2=[0.1;B2(find(B2>num))];
    C3=[0.1;B3(find(B3>num))];
    C4=[0.1;B4(find(B4>num))];
    C5=[0.1;B5(find(B5>num))];
    C6=[0.1;B6(find(B6>num))];
    C7=[0.1;B7(find(B7>num))];
    C8=[0.1;B8(find(B8>num))];
% SE CÁLCULA UN PROMEDIO DE TODAS LAS MUESTRAS MAYORES A 0.3
    D1=mean(C1);
    D2=mean(C2);
    D3=mean(C3);
    D4=mean(C4);
    D5=mean(C5);
    D6=mean(C6);
    D7=mean(C7);
    D8=mean(C8);
    um=0.3;
    te=0;
    as=0;
```

```

% SE VERIFICA SI EL PROMEDIO DE LAS 25 ÚLTIMAS MUESTRAS DE LA VENTANA
% DE ANÁLISIS SUPERAN EL VALOR DE 0.3 LO QUE DETERMINA SI EL USUARIO
% SIGUE MANTENIEDO EL GESTO O SE ENCUENTRA EN REPOSO, SI EL USUARIO SE
% ENCUENTRA EN REPOSO SE REESTABLECEN ALGUNAS VARIABLES
if
((D1>=um) || (D2>=um) || (D3>=um) || (D4>=um) || (D5>=um) || (D6>=um) || (D7>=um) || (D8>
=um))
    if s==1 && PI==1 % se verifica si el gesto PINCH se está sosteniendo
        elseif s==1 && FI==1 % se verifica si el gesto FIST se está
sosteniendo
            elseif s==1 && WI==1 % se verifica si el gesto WAVE IN se está
sosteniendo
                elseif s==1 && WO==1 % se verifica si el gesto WAVE OUT se está
sosteniendo
                    if fin==15 % condicional que verifica si se está sosteniendo el
gesto WAVE OUT para finalizar el programa
                        te=1;
                    end
                elseif s==1 && OP==1 % se verifica si el gesto OPEN se está
sosteniendo
                    end
                else % si la condición inicial no se cumple
                    as=1;
                    if abcd==1 % si el usuario esta en reposo se formatea algunas
variables necesarias
                        clc
                        clear BARRA
                        imshow(imagen1); % se muestra en pantalla la representación de
reposo
                    end
                end
                % Se pide realizar cada gesto pero solo para la configuración inicial
                if Q==1
                    imshow(imagen4);
                    title('REALICE EL GESTO WAVE IN')
                elseif Q==2
                    imshow(imagen5);
                    title('REALICE EL GESTO WAVE OUT')
                elseif Q==3
                    imshow(imagen2);
                    title('REALICE EL GESTO FIST')
                elseif Q==4
                    imshow(imagen3);
                    title('REALICE EL GESTO OPEN')
                elseif Q==5
                    imshow(imagen6);
                    title('REALICE EL GESTO PINCH')
                end
            end
        end
    end
end

```

2.3.2 ETAPA DE PREPROCESAMIENTO EN EL DOMINIO TEMPORAL

En esta etapa se implementa las herramientas descritas en el capítulo 1 (acondicionamiento de procesado de señales), donde se realiza una transformación de las señales EMG para su posterior análisis en la etapa de extracción de características.

2.3.2.1 Extracción de señal útil

Sabiendo que la matriz $S_{400 \times 8}$ en la 4ta iteración del algoritmo, desde la detección de inicio de gesto, contiene 160 valores exclusivamente de EMG de activación, en esta fase requerimos obtener únicamente los vectores $\{\vec{A}_i = (S_i(240), \dots, S_i(400))\}$ con $i = 1, \dots, 8$ que contienen a la señales EMG sin los valores de reposo para su análisis. La figura 2.24, muestra un ejemplo de extracción de señal útil aplicado a las señales EMG.

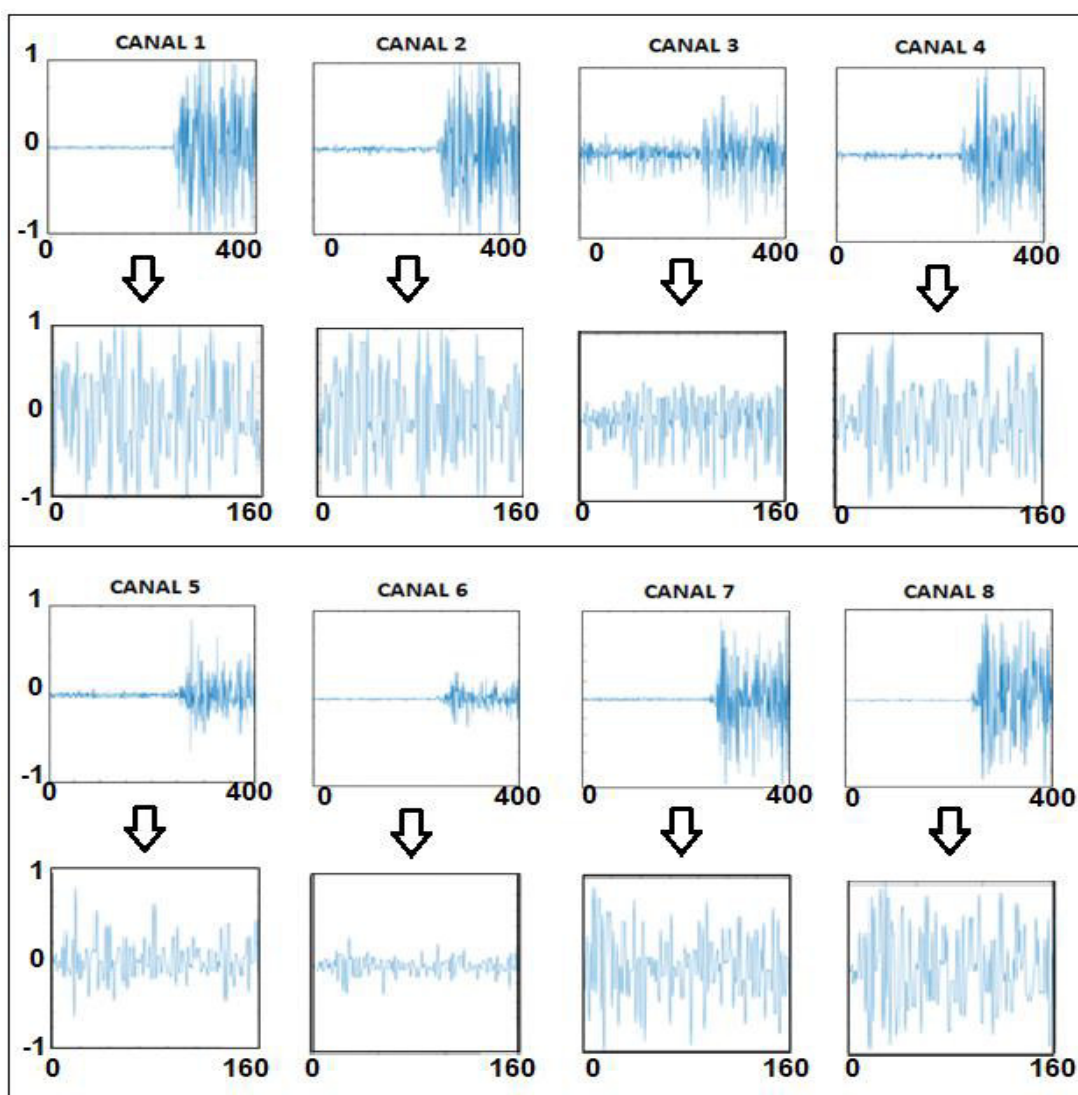


Figura 2.24. Extracción de señal útil de las 8 señales EMG

El código de Matlab utilizado para la extracción de señal útil de las 8 señales EMG, se describe por medio de la función creada llamada **senal_util**, en donde se obtiene el inicio de las señales EMG de activación desde la muestras 240 hasta la 400.

```
function [A1,A2,A3,A4,A5,A6,A7,A8]=senal_util (S1,S2,S3,S4,S5,S6,S7,S8,i)
%VENTANA DE 400 MUESTRAS DE LA ITERACIÓN 4 QUE CONTIENE LAS SEÑALES
EMG
%PARA RECORTAR Y EXTRAER UNICAMENTE LA SEÑAL EMG SIN LA PARTE DE
REPOSO
fin=400;
inicio=240;
A1=S1{i}(inicio:fin);
A2=S2{i}(inicio:fin);
A3=S3{i}(inicio:fin);
A4=S4{i}(inicio:fin);
A5=S5{i}(inicio:fin);
A6=S6{i}(inicio:fin);
A7=S7{i}(inicio:fin);
A8=S8{i}(inicio:fin);
end
```

2.3.2.2 Rectificación

En esta fase se utiliza la rectificación de una señal específicamente la rectificación de onda completa, ya que proporciona mejores resultados comparado con la rectificación de media onda, como se describe más adelante. En esta fase se convierte los valores negativos de una señal en positivos, aplicando el valor absoluto a los vectores que contienen las señales EMG de activación. La instrucción **abs** de Matlab es utilizada para hallar el valor absoluto de los vectores $\{\vec{B}_i = abs(\vec{A}_i)\}$.

$B=abs(A)$

Donde:

A: representa un vector con valores positivos y negativos.

B: representa el vector obtenido después de aplicar el valor absoluto a cada muestra del vector A

La figura 2.25, muestra un ejemplo de aplicación de rectificación de onda completa a las 8 señales EMG de activación, en donde observamos que no existen valores negativos de las señales EMG obtenidas.

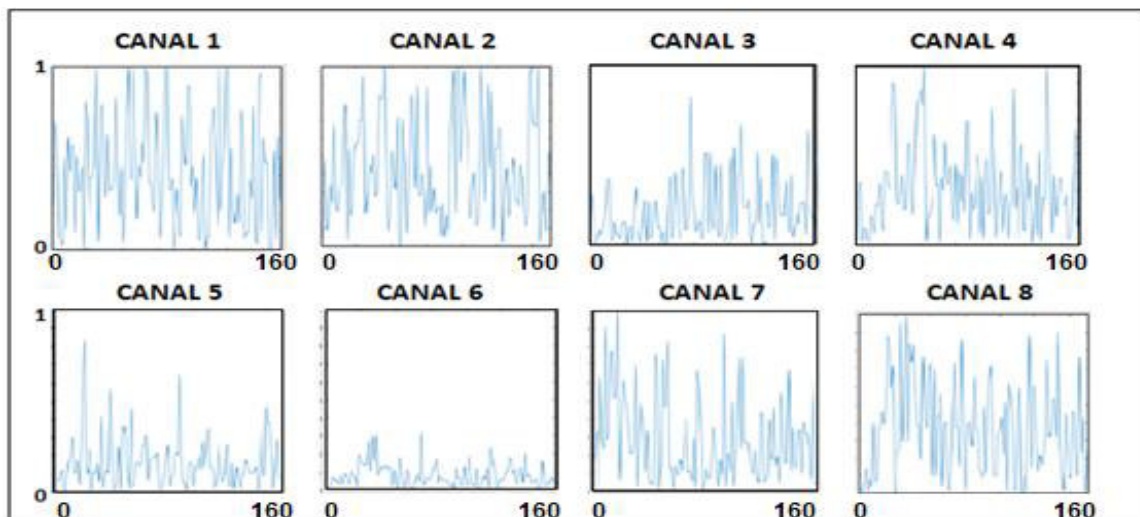


Figura 2.25. Rectificación de las 8 señales EMG

La función creada en Matlab llamada **rectificacion**, convierte los valores negativos en positivos de las 8 señales EMG, utilizando la instrucción `abs`, la cual queda de la siguiente manera:

```
function [B1,B2,B3,B4,B5,B6,B7,B8]=rectificacion(A1,A2,A3,A4,A5,A6,A7,A8)
    B1=abs(A1); % valor absoluto de la señal EMG del canal 1
    B2=abs(A2); % valor absoluto de la señal EMG del canal 2
    B3=abs(A3); % valor absoluto de la señal EMG del canal 3
    B4=abs(A4); % valor absoluto de la señal EMG del canal 4
    B5=abs(A5); % valor absoluto de la señal EMG del canal 5
    B6=abs(A6); % valor absoluto de la señal EMG del canal 6
    B7=abs(A7); % valor absoluto de la señal EMG del canal 7
    B8=abs(A8); % valor absoluto de la señal EMG del canal 8
end
```

2.3.2.3 Envolvente

En esta fase se utiliza la envolvente de una señal para calcular las envolventes de las 8 señales EMG rectificadas, para ello utilizamos interpolación segmentaria por splines cúbicos inmersa en la función **envelope** de Matlab, mediante su configuración **peak** con 3 muestras de separación entre puntos a interpolar $\{\vec{C}_i = envelope(\vec{B}_i)\}$.

```
[Superior,Inferior]=envelope(B,Np,'peak')
```

Donde:

- B*: señal
- Np*: número de muestras de separación entre puntos a interpolar
- peak*: método por interpolación segmentaria de “splines” sobre los valores máximos locales separados por al menos *np* muestras.

La figura 2.26, muestra un ejemplo de aplicación de envolvente de una señal a las 8 señales EMG de activación rectificadas, donde observamos la silueta superior de las señales.

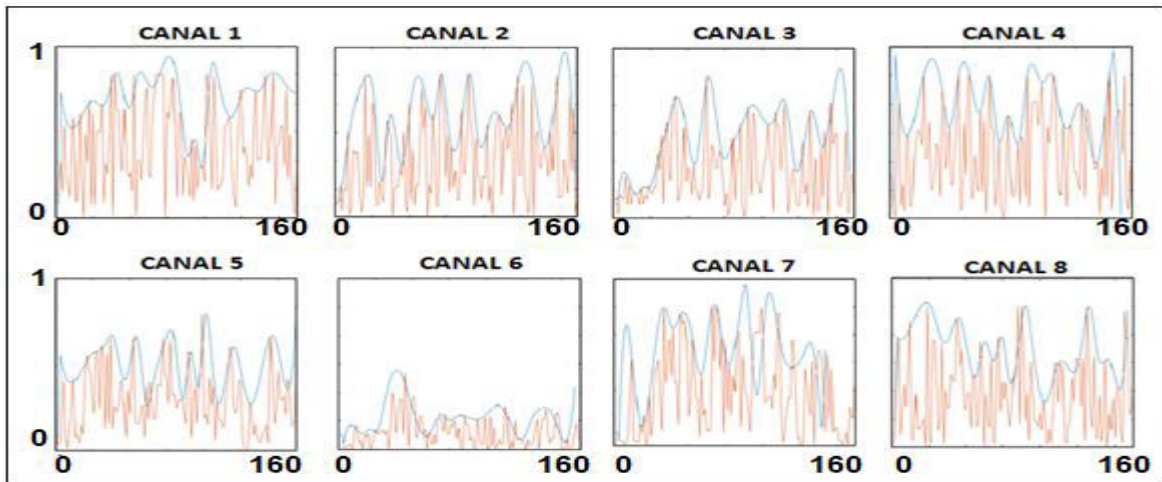


Figura 2.26. Envolvente de las 8 señales EMG

La función creada en Matlab llamada **envolvente**, la cual encuentra las envolventes de las 8 señales EMG queda de la siguiente manera:

```
function [C1,C2,C3,C4,C5,C6,C7,C8]=envolvente(B1,B2,B3,B4,B5,B6,B7,B8)
% ENVOLVENTE DE CANAL 1
df1=envelope(B1,3,'peak');
% eliminación de valores de la envolvente que sean menores a 0
ab1=[0.1;df1(find(df1>0))];
% eliminación de valores de la envolvente que sean mayores a 1.2
C1=[0.1;ab1(find(ab1<1.2))];
% ENVOLVENTE DE CANAL 2
df2=envelope(B2,3,'peak');
ab2=[0.1;df2(find(df2>0))];
C2=[0.1;ab2(find(ab2<1.2))];
% ENVOLVENTE DE CANAL 3
df3=envelope(B3,3,'peak');
ab3=[0.1;df3(find(df3>0))];
C3=[0.1;ab3(find(ab3<1.2))];
% ENVOLVENTE DE CANAL 4
df4=envelope(B4,3,'peak');
ab4=[0.1;df4(find(df4>0))];
C4=[0.1;ab4(find(ab4<1.2))];
% ENVOLVENTE DE CANAL 5
df5=envelope(B5,3,'peak');
ab5=[0.1;df5(find(df5>0))];
C5=[0.1;ab5(find(ab5<1.2))];
% ENVOLVENTE DE CANAL 6
df6=envelope(B6,3,'peak');
ab6=[0.1;df6(find(df6>0))];
C6=[0.1;ab6(find(ab6<1.2))];
% ENVOLVENTE DE CANAL 7
df7=envelope(B7,3,'peak');
ab7=[0.1;df7(find(df7>0))];
C7=[0.1;ab7(find(ab7<1.2))];
% ENVOLVENTE DE CANAL 8
df8=envelope(B8,3,'peak');
ab8=[0.1;df8(find(df8>0))];
C8=[0.1;ab8(find(ab8<1.2))];
end
```

2.3.2.4 Suavizado de curvas

En esta fase se utiliza un suavizado de curvas para reducir los picos indeseables y obtener la tendencia con menos ajuste de las envolventes de las 8 señales EMG, para ello utilizamos un filtro de media móvil, inmerso en la función **movmean** de Matlab, con un tamaño de ventana de 20 valores $\{\bar{D}_i = \text{movmean}(\bar{C}_i)\}$.

```
D= movmean(C,v)
```

Donde:

C: vector de entradas

v: tamaño de la ventana (número de valores para promedio)

La figura 2.27, muestra un ejemplo de aplicación de suavizado de curvas a las 8 envolventes de señales EMG de activación rectificadas, donde observamos la tendencia de las envolventes al reducir los picos indeseados.

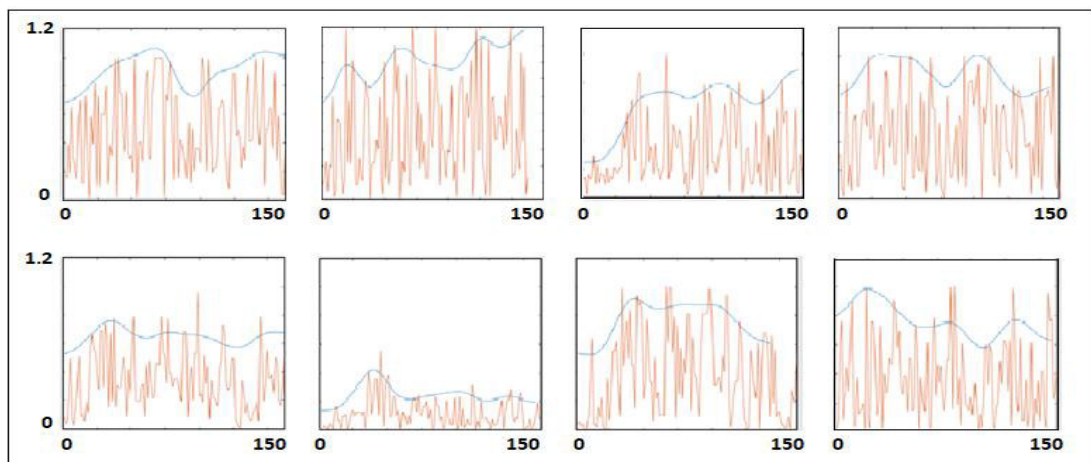


Figura 2.27. Suavizado de curvas de las 8 envolventes

La función creada en Matlab llamada **suavizado_curvas** para el suavizado de curvas de las 8 envolventes de las señales EMG queda de la siguiente manera:

```
function [D1,D2,D3,D4,D5,D6,D7,D8] = suavizado_curvas(C1,C2,C3,C4,C5,C6,C7,C8)
w=20;
D1=1.2*movmean(C1,w);
D2=1.2*movmean(C2,w);
D3=1.2*movmean(C3,w);
D4=1.2*movmean(C4,w);
D5=1.2*movmean(C5,w);
D6=1.2*movmean(C6,w);
D7=1.2*movmean(C7,w);
D8=1.2*movmean(C8,w);
end
```

2.3.3 ETAPA EXTRACCIÓN DE CARACTERÍSTICAS

La extracción de características consiste en encontrar patrones únicos que diferencien a cada gesto estudiado, algunas de estas características son: área bajo curva para el dominio temporal, dimensión fractal para el dominio de la complejidad, área de densidad espectral de potencia suavizada para el dominio de la frecuencia.

2.3.3.1 Área bajo la curva (dominio del tiempo)

Debido a la naturaleza aleatoria de las señales EMG, utilizar únicamente la envolvente de estas señales, no proporciona la suficiente información para diferenciar los gestos estudiados, por ello en esta fase se utiliza el área bajo la curva de las 8 envolventes suavizadas de las señales EMG de activación rectificadas, que mantienen prácticamente un mismo valor de área bajo curva en cada canal del dispositivo en distintas repeticiones de un mismo gesto. Para calcular el área bajo la curva utilizamos el método del trapecio, inmerso en la instrucción **trapz** de Matlab $\{\vec{E}_i = \text{trapz}(\vec{D}_i)\}$.

$$E = \text{trapz}(D)$$

Donde:

D: vector con datos numéricos

La tabla 2.2, muestra un ejemplo de aplicación de área bajo la curva a las 8 envolventes de las señales EMG, obteniendo un vector con 8 valores propios de área $\vec{E} = [Ar1, Ar2, Ar3, Ar4, Ar5, Ar6, Ar7, Ar8]$ que caracterizan a cada gesto estudiado, ya que según el gesto realizado, la activación de los canales es distinta.

Tabla 2.2. Área bajo la curva de las 8 envolventes temporales

	Canal 1	Canal 2	Canal 3	Canal 4	Canal 5	Canal 6	Canal 7	Canal 8
Fist	104	96	87	82	47	29	79	100
Wave in	76	35	55	23	21	82	103	86
Wave out	24	83	113	116	72	19	16	18
Open	36	61	106	66	18	39	92	65
Pinch	19	26	79	43	20	29	49	47

La función creada en Matlab llamada **area** para el cálculo de área bajo la curva de las 8 señales EMG queda de la siguiente manera:

```

function [E1,E2,E3,E4,E5,E6,E7,E8] = area(D1,D2,D3,D4,D5,D6,D7,D8)
% Instrucción de MATLAB para calcular el área bajo la curva utilizando
% el método del trapecio para las señales de cada canal
t1=trapz(Am1);
t2=trapz(Am2);
t3=trapz(Am3);
t4=trapz(Am4);
t5=trapz(Am5);
t6=trapz(Am6);
t7=trapz(Am7);
t8=trapz(Am8);
end

```

2.3.3.2 Área de densidad espectral de potencia suavizada (dominio de la frecuencia)

En esta fase se utiliza el periodograma de las 8 señales EMG, para ello utilizamos el valor absoluto del espectro de potencias DFT, inmerso en la instrucción **fft** y **abs** de Matlab $\{\bar{P}_i = abs(fft(\bar{S}_i))\}$.

```

Y=abs(fft(X,n)) % Cálculo de la transformada discreta de Fourier (DFT) usando
un algoritmo de transformada rápida de Fourier (FFT).

```

Donde:

X: vector de entrada (Señal temporal)

n: Longitud de transformada en potencias de 2 para aumentar rendimiento de la fft

La figura 2.28, muestra un ejemplo de cálculo de periodograma de las 8 señales EMG, para una frecuencia de muestreo al doble de la frecuencia máxima de la señal EMG, 100 Hz, según el teorema de muestro de Nyquist-Shannon.

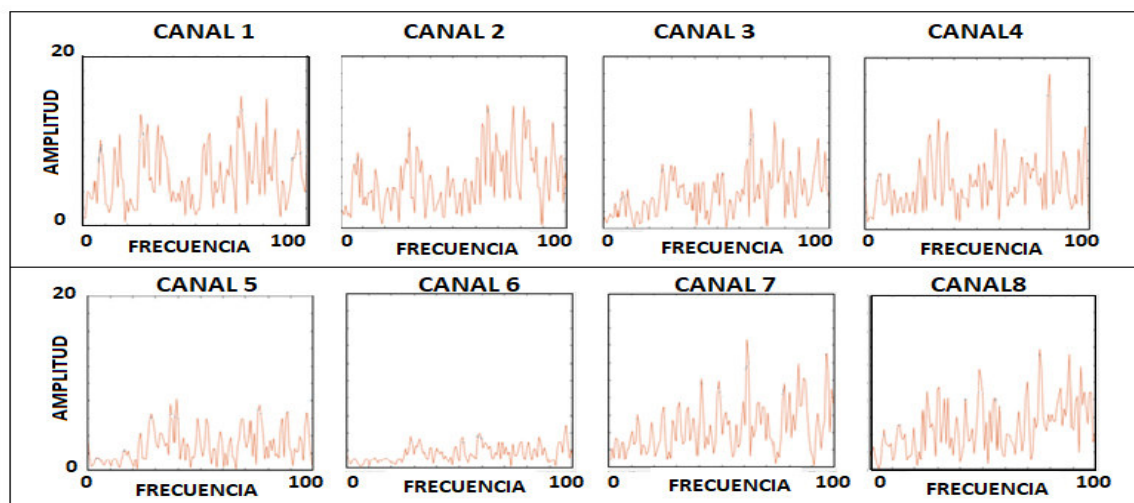


Figura 2.28. Periodograma de las 8 señales EMG

Debido a la naturaleza aleatoria de las señales EMG, el periodograma de las señales no presenta rangos de frecuencias específicas de trabajo, que puedan diferenciar a los gestos estudiados, siendo necesario calcular el área bajo la curva de las envolventes de los espectros de potencia para las 8 señales EMG.

La figura 2.29, muestra un ejemplo de aplicación de envolvente superior de una señal a los 8 periodogramas de las señales EMG.

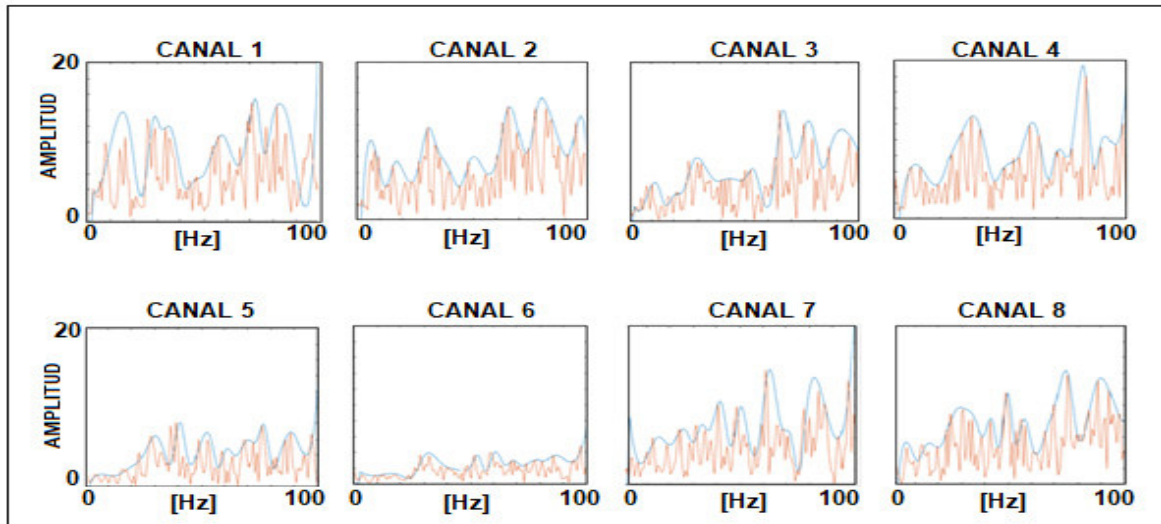


Figura 2.29. Envolventes de los 8 periodogramas

La figura 2.30, muestra un ejemplo de aplicación de suavizado de curvas a las 8 envolventes de los periodogramas de las señales EMG.

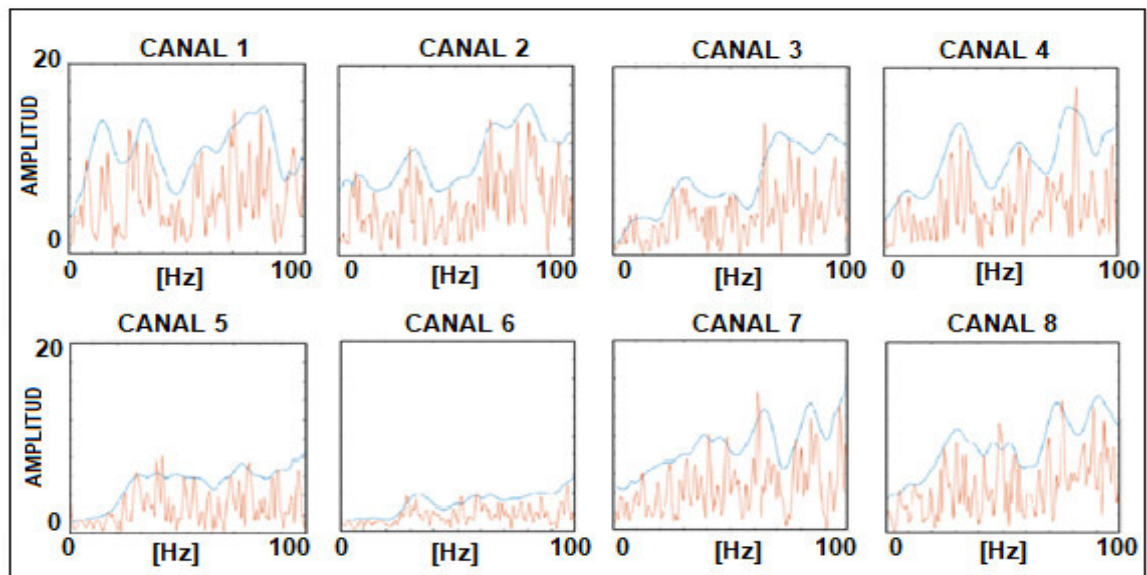


Figura 2.30. Suavizado de los 8 periodogramas

La tabla 2.3, muestra un ejemplo de aplicación de área bajo la curva a las 8 envolventes de los periodogramas, obteniendo un vector con 8 valores propios de área $\vec{E} = [Ar1, Ar2, Ar3, Ar4, Ar5, Ar6, Ar7, Ar8]$ que caracterizan a cada gesto estudiado.

Tabla 2.3. Área bajo las 8 envolventes espectrales

	Canal 1	Canal 2	Canal 3	Canal 4	Canal 5	Canal 6	Canal 7	Canal 8
Fist	632508	613842	538803	504140	294710	191670	415340	382700
Wave in	347720	148170	243510	176880	103630	196500	480230	342680
Wave out	195880	262800	545630	490070	415140	190290	190170	151450
Open	299710	161570	476050	431400	169960	183620	351370	315440
Pinch	169330	117710	281060	287560	180370	142890	235290	216140

La función creada en Matlab llamada **APSDS**, para el cálculo del área bajo las envolventes de los 8 periodogramas queda de la siguiente manera:

```
function [E1,E2,E3,E4,E5,E6,E7,E8]APSDS(S1,S2,S3,S4,S5,S6,S7,S8)
    Nn=length(S1);
    nFFTn=2;
    while nFFTn<Nn
        nFFTn=256*nFFTn;
    end
    P1=abs(fft(S1,nFFTn)); % CANAL 1
    P11=envelope(P1(1:length(P1)/2),3,'peak');
    P111=[0.1;P11(find(P11>0))];
    P1111=[0.1;P111(find(P111<15))];
    P11111=movmean(P1111,20);
    E1=trapz(P11111)
%-----
    Nn2=length(S2);
    nFFTn2=2;
    while nFFTn2<Nn2
        nFFTn2=256*nFFTn2;
    end
    P2=(abs(fft(S2,nFFTn2))); % CANAL 2
    P22=envelope(P2(1:length(P2)/2),3,'peak');
    P222=[0.1;P22(find(P22>0))];
    P2222=[0.1;P222(find(P222<15))];
    P22222=movmean(P2222,20);
    E2=trapz(P22222)
%-----
    Nn3=length(S3);
    nFFTn3=2;
    while nFFTn3<Nn3
        nFFTn3=256*nFFTn3;
    end
    P3=(abs(fft(S3,nFFTn3))); % CANAL 3
    P33=envelope(P3(1:length(P3)/2),3,'peak');
    P333=[0.1;P33(find(P33>0))];
    P3333=[0.1;P333(find(P333<15))];
    P33333=movmean(P3333,20);
    E3=trapz(P33333)
```



```

%-----
Nn4=length(S4);
nFFFn4=2;
while nFFFn4<Nn4
    nFFFn4=256*nFFFn4;
end
P4=(abs(fft(S4,nFFFn4))); % CANAL 4
P44=envelope(P4(1:length(P4)/2),3,'peak');
P444=[0.1;P44(find(P44>0))];
P4444=[0.1;P444(find(P444<15))];
P44444=movmean(P4444,20);
E4=trapz(P44444)

%-----
Nn5=length(S5);
nFFFn5=2;
while nFFFn5<Nn5
    nFFFn5=256*nFFFn5;
end
P5=(abs(fft(S5,nFFFn5))); % CANAL 5
P55=envelope(P5(1:length(P5)/2),3,'peak');
P555=[0.1;P55(find(P55>0))];
P5555=[0.1;P555(find(P555<15))];
P55555=movmean(P5555,20);
E5=trapz(P55555)

%-----
Nn6=length(g6);
nFFFn6=2;
while nFFFn6<Nn6
    nFFFn6=256*nFFFn6;
end
P6=(abs(fft(S6,nFFFn6))); % CANAL 6
P66=envelope(P6(1:length(P6)/2),3,'peak');
P666=[0.1;P66(find(P66>0))];
P6666=[0.1;P666(find(P666<15))];
P66666=movmean(P6666,20);
E6=trapz(P66666)

%-----
Nn7=length(g7);
nFFFn7=2;
while nFFFn7<Nn7
    nFFFn7=256*nFFFn7;
end
P7=(abs(fft(S7,nFFFn7))); % CANAL 7
P77=envelope(P7(1:length(P7)/2),3,'peak');
P777=[0.1;P77(find(P77>0))];
P7777=[0.1;P777(find(P777<15))];
P77777=movmean(P7777,20);
E7=trapz(P77777)

%-----
Nn8=length(g8);
nFFFn8=2;
while nFFFn8<Nn8
    nFFFn8=256*nFFFn8;
end
P8=(abs(fft(S8,nFFFn8))); % CANAL 8
P88=envelope(P8(1:length(P8)/2),3,'peak');
P888=[0.1;P88(find(P88>0))];
P8888=[0.1;P888(find(P888<15))];
P88888=movmean(P8888,20);
E8=trapz(P88888)
end

```

2.3.3.3 Dimensión fractal (Dominio de la complejidad)

En esta fase se aplica la dimensión fractal a las 8 señales EMG, mediante la implementación de las ecuaciones de Katz en Matlab como se muestra a continuación:

```
L = 0;
N = length(s); % tamaño de la serie temporal
% CALCULO DE (L)
for i = 1:(N - 1)
    aux = sqrt(1+((s(i)-s(i+1))^2)); % cálculo de la distancia
                                     % euclidiana entre dos puntos
    L=L+aux; % (L) sumatorio de las distancias euclidianas entre los
            % puntos sucesivos
    clear('aux');
end
% CÁLCULO DE (d)
for i=2:N
    dist(i)=sqrt(((1-i)^2)+(s(1)-s(i))^2);
end
d= max(dist); % valor máximo de todas las distancias obtenidas entre
              % el punto uno y cada punto
% ECUACION DE KATZ
KFD=log10(N)/(log10(d/L)+log10(N));
```

La dimensión fractal de Katz se aplica a las 8 señales EMG de activación \vec{A}_i , después de la extracción de señal útil, para obtener un vector con 8 valores decimales de dimensión fractal $\vec{E} = [KDF1, KDF2, KDF3, KDF4, KDF5, KDF6, KDF7, KDF8]$ que caracterizan a cada gesto estudiado.

La función creada en Matlab para el cálculo de la dimensión fractal de Katz de las 8 señales EMG, se presenta a continuación.

```

function [E1,E2,E3,E4,E5,E6,E7,E8]=KFD(A1,A2,A3,A4,A5,A6,A7,A8)
%CÁLCULO DE LA DIMENSIÓN FRACTAL DE KATZ PARA CADA CANAL
% CANAL 1 -----
L1=0;
N1=length(A1);
for i=1:(N1-1)
    aux=sqrt(1+((A1(i)-A1(i+1))^2));
    L1=L1+aux;
    clear('aux');
end
for i=2:N1
    dist1(i)=sqrt(((1-i)^2)+(A1(1)-A1(i))^2));
end
d1= max(dist1);
E1=log10(N1)/(log10(d1/L1)+log10(N1));
% CANAL 2 -----
L2=0;
N2=length(A2);
for i=1:(N2-1)
    aux=sqrt(1+((A2(i)-A2(i+1))^2));
    L2=L2+aux;
    clear('aux');
end
for i=2:N2
    dist2(i)=sqrt(((1-i)^2)+(A2(1)-A2(i))^2));
end
d2= max(dist2);
E2=log10(N2)/(log10(d2/L2)+log10(N2));
% CANAL 3 -----
L3=0;
N3=length(A3);
for i=1:(N3-1)
    aux=sqrt(1+((A3(i)-A3(i+1))^2));
    L3=L3+aux;
    clear('aux');
end
for i=2:N3
    dist3(i)=sqrt(((1-i)^2)+(A3(1)-A3(i))^2));
end
d3=max(dist3);
E3=log10(N3)/(log10(d3/L3)+log10(N3));
% CANAL 4 -----
L4=0;
N4=length(A4);
for i=1:(N4-1)
    aux=sqrt(1+((A4(i)-A4(i+1))^2));
    L4=L4+aux;
    clear('aux');
end
for i=2:N4
    dist4(i)=sqrt(((1-i)^2)+(A4(1)-A4(i))^2));
end
d4=max(dist4);
E4=log10(N4)/(log10(d4/L4)+log10(N4));

```

```

% CANAL 5 -----
L5=0;
N5=length(A5);
for i=1:(N5-1)
    aux=sqrt(1+((A5(i)-A5(i+1))^2));
    L5=L5+aux;
    clear('aux');
end
for i=2:N5
    dist5(i)=sqrt(((1-i)^2)+((A5(1)-A5(i))^2));
end
d5=max(dist5);
E5=log10(N5)/(log10(d5/L5)+log10(N5));
% CANAL 6 -----
L6=0;
N6=length(A6);
for i=1:(N6-1)
    aux=sqrt(1+((A6(i)-A6(i+1))^2));
    L6=L6+aux;
    clear('aux');
end
for i=2:N6
    dist6(i)=sqrt(((1-i)^2)+((A6(1)-A6(i))^2));
end
d6=max(dist6);
E6=log10(N6)/(log10(d6/L6)+log10(N6));
% CANAL 7 -----
L7=0;
N7=length(A7);
for i=1:(N7-1)
    aux=sqrt(1+((A7(i)-A7(i+1))^2));
    L7=L7+aux;
    clear('aux');
end
for i=2:N7
    dist7(i)=sqrt(((1-i)^2)+((A7(1)-A7(i))^2));
end
d7=max(dist7);
E7=log10(N7)/(log10(d7/L7)+log10(N7));
% CANAL 8 -----
L8=0;
N8=length(A8);
for i=1:(N8-1)
    aux=sqrt(1+((A8(i)-A8(i+1))^2));
    L8=L8+aux;
    clear('aux');
end
for i=2:N8
    dist8(i)=sqrt(((1-i)^2)+((A8(1)-A8(i))^2));
end
d8=max(dist8);
E8=log10(N8)/(log10(d8/L8)+log10(N8));

```

La tabla 2.2, muestra los valores de dimensión fractal obtenidos para las 8 señales EMG de activación para cada gesto estudiado.

Tabla 2.2. Dimensión fractal de Katz aplicado a las 8 señales EMG por gesto

	Canal 1	Canal 2	Canal 3	Canal 4	Canal 5	Canal 6	Canal 7	Canal 8
Fist	1,0494	1,0463	1,0417	1,0395	1,0091	1,0070	1,0391	1,0415
Wave in	1,0202	1,0065	1,0102	1,0070	1,0005	1,0196	1,0444	1,0155
Wave out	1,0017	1,0170	1,0468	1,0456	1,0133	1,0017	1,0004	1,0004
Open	1,0101	1,0101	1,0517	1,0314	1,0017	1,0039	1,0215	1,0155
Pinch	1,0053	1,0054	1,0288	1,0189	1,0012	1,0031	1,0092	1,0125

Como se puede observar en la tabla 2.2, los números decimales en los valores obtenidos por canal, contienen información que diferencia a cada gesto estudiado, si se analiza en conjunto los 8 valores obtenidos, por lo tanto es necesario realizar una extracción de decimales como se observa en la tabla 2.3.

Tabla 2.3. Extracción de decimales

	Canal 1	Canal 2	Canal 3	Canal 4	Canal 5	Canal 6	Canal 7	Canal 8
Fist	494	463	417	395	91	70	391	415
Wave in	202	65	102	70	5	196	444	155
Wave out	17	170	468	456	133	17	4	4
Open	101	101	517	314	17	39	215	155
Pinch	53	54	288	189	12	31	92	125

2.4 CONFORMACIÓN DE CLÚSTERS N-DIMENSIONALES

Ya que en este trabajo se emplean los 8 canales de EMG, para poder realizar una descripción gráfica de los vectores obtenidos en la etapa de extracción de características, se empleará el algoritmo PCA para reducción de dimensionalidad de 8 a 2, inmerso en la instrucción *pca* de Matlab para 2 componentes principales, como se muestra a continuación:

```
[X Y]=pca(Z, 'NumComponents',2); % Reducción de dimensionalidad en 2
                                % componentes principales
```

Donde:

Z: Matriz de 8 columnas con los valores propios obtenidos en la etapa de extracción de características

X: coeficientes del componente principal (vectores propios)

Y: Matriz de 2 columnas con las componentes principales

A continuación se muestra la distribución de los clústers obtenidos para cada fase en la etapa de extracción de características, al aplicar reducción de dimensionalidad (8 a 2) a 25 repeticiones de cada gesto (8 señales EMG por repetición) pregrabadas y procesadas según las etapas antes mencionadas.

La figura 2.31, muestra gráficamente la distribución de los clústers para 2 componentes principales, obtenidas a partir de la variable en el dominio temporal (área bajo las envolventes de las 8 señales EMG).

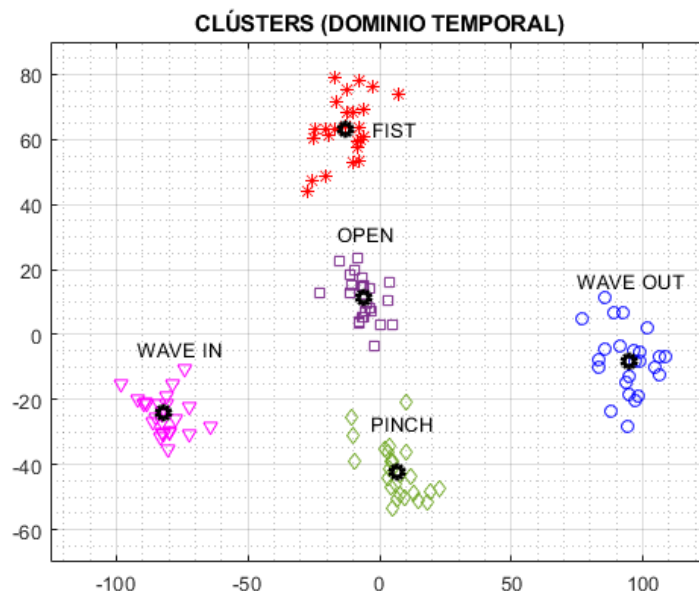


Figura 2.31. Clústers en el dominio temporal (2 componentes principales)

La figura 2.32, muestra gráficamente la distribución de los clústers para 2 componentes principales, obtenidas a partir de la variable en el dominio de la frecuencia (área de densidad espectral de potencia suavizada de las 8 señales EMG).

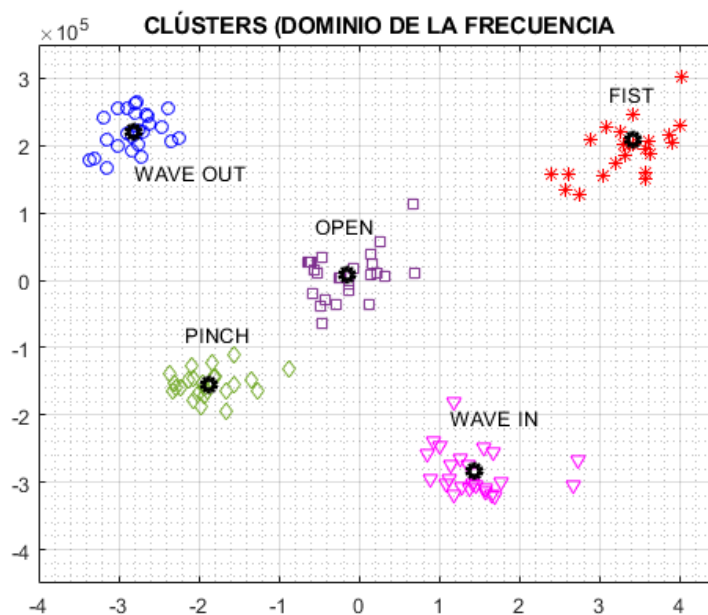


Figura 2.32. Clústers en el dominio de la frecuencia (2 componentes principales)

La figura 2.33, muestra gráficamente la distribución de los clústers para 2 componentes principales, obtenidas a partir de la variable en el dominio de la complejidad (dimensión fractal de Katz de las 8 señales EMG).

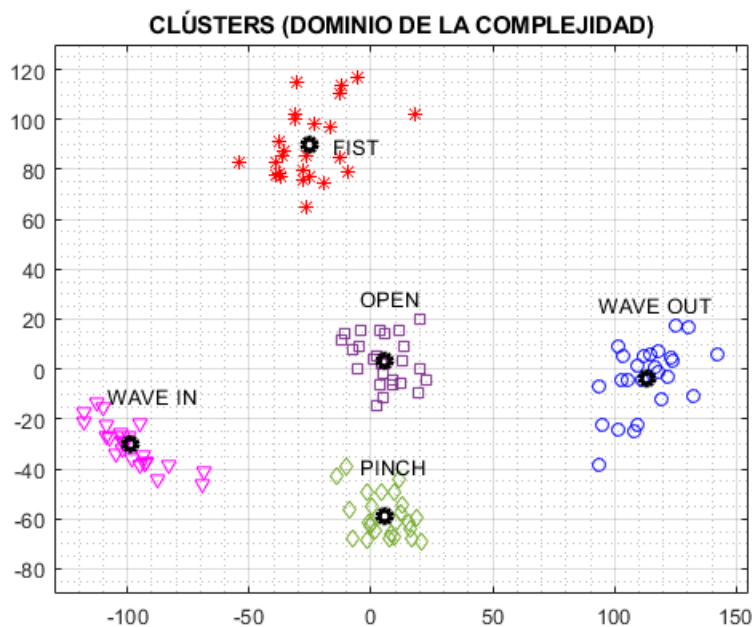


Figura 2.33. Clústers en el dominio de la complejidad (2 componentes principales)

2.5 ETAPA DE CLASIFICACIÓN

En la clasificación de gestos de la mano se utiliza dos métodos bastante conocidos, método del centroide y redes neuronales artificiales. El método del centroide por su fácil implementación y por la gran separación entre clústers de los gestos. El segundo método (redes neuronales artificiales), se utiliza por su gran capacidad de clasificación mediante un aprendizaje supervisado. Cabe mencionar que la etapa de clasificación se utiliza después de la selección de la mejor variable obtenida en la etapa de extracción de características como se explica más adelante.

2.5.1 MÉTODO 1: DISTANCIA EUCLIDIANA AL CENTROIDE MÁS CERCANO

En esta fase se implementa el método del centroide con distancia euclidiana para 8 dimensiones, a partir de un centroide calculado y el vector de 8 muestras obtenido en la etapa de extracción de características, esto mediante una función creada en Matlab como se muestra a continuación:

```

function
[distancia]=metodo1(E1,E2,E3,E4,E5,E6,E7,E8,FI1,FI2,FI3,FI4,FI5,FI6,FI7,FI8,WI
1,WI2,WI3,WI4,WI5,WI6,WI7,WI8,WO1,WO2,WO3,WO4,WO5,WO6,WO7,WO8,OP1,OP2,OP3,OP4,
OP5,OP6,OP7,OP8,PI1,PI2,PI3,PI4,PI5,PI6,PI7,PI8)
%-----
% Fórmula para hallar la distancia euclidiana entre los centroides de cada
gesto y una nueva muestra.
D_FI=sqrt((FI1-E1)^2+(FI2-E2)^2+(FI3-E3)^2+(FI4-E4)^2+(FI5-E5)^2+(FI6-E6)^2+(FI7-
E7)^2+(FI8-E8)^2);

D_WI=sqrt((WI1-E1)^2+(WI2-E2)^2+(WI3-E3)^2+(WI4-E4)^2+(WI5-E5)^2+(WI6-E6)^2+(WI7-
E7)^2+(WI8-E8)^2);

D_WO=sqrt((WO1-E1)^2+(WO2-E2)^2+(WO3-E3)^2+(WO4-E4)^2+(WO5-E5)^2+(WO6-E6)^2+(WO7-
E7)^2+(WO8-E8)^2);

D_OP=sqrt((OP1-E1)^2+(OP2-E2)^2+(OP3-E3)^2+(OP4-E4)^2+(OP5-E5)^2+(OP6-E6)^2+(OP7-
E7)^2+(OP8-E8)^2);

D_PI=sqrt((PI1-E1)^2+(PI2-E2)^2+(PI3-E3)^2+(PI4-E4)^2+(PI5-E5)^2+(PI6-E6)^2+(PI7-
E7)^2+(PI8-E8)^2);
% Concatenación de las distancias calculadas entre un nuevo gesto y los
centroides de cada gesto
DI=[D_WI,D_WO,D_FI,D_OP,D_PI];
% Instrucción para hallar el índice del menor valor que identifica al gesto
realizado
[V,distancia]=min(DI);
end

```

2.5.2 MÉTODO 2: RED NEURONAL BASADA EN PERCEPTRÓN MULTICAPA (MLP).

Dentro del software Matlab se utiliza el toolbox Neural Pattern Recognition. Esta herramienta permite entrenar una red neuronal de tipo perceptrón multicapa.

A continuación se describen los pasos para instalar el toolbox Neural Pattern Recognition.

1) Ingresar al asistente de instalación de toolbox (figura 2.34)

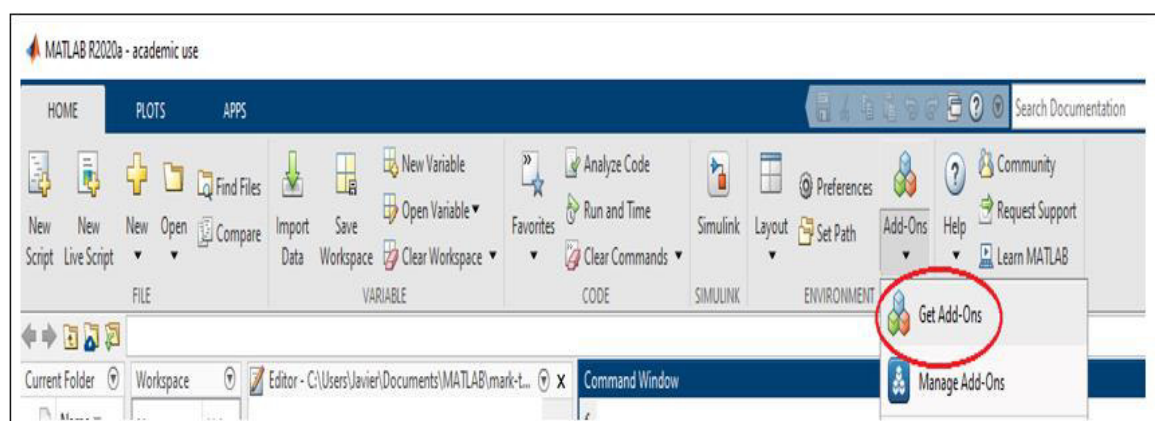


Figura 2.34. Menú para la instalación de Neural Pattern Recognition

2) Buscar e instalar el toolbox Deep Learning Toolbox (figura 2.35).

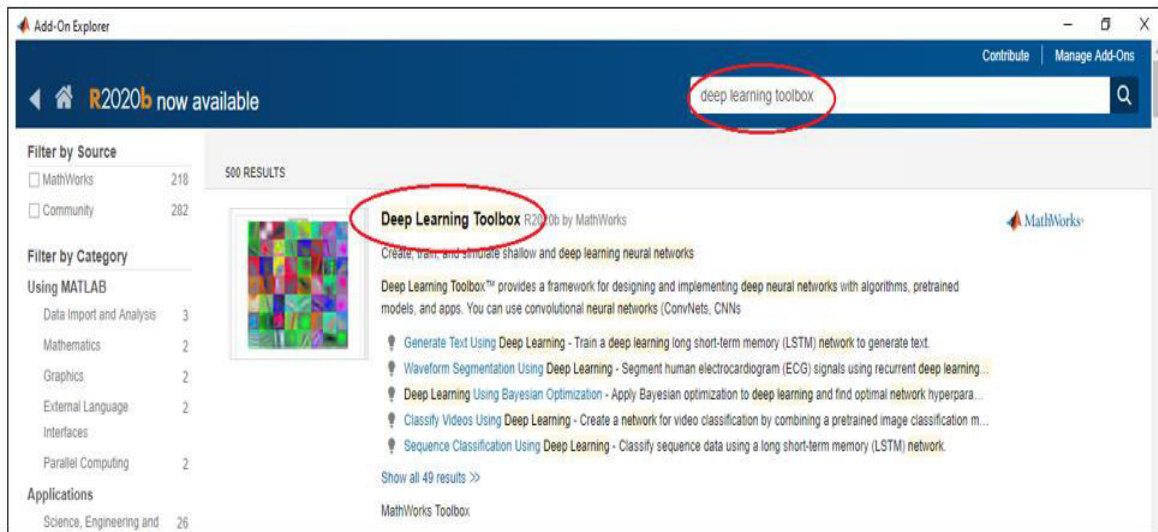


Figura 2.35. Instalación Deep Learning Toolbox

Para crear y configurar la red neuronal de clasificación en Matlab, se ejecuta el comando `nnstart` en el command windows de Matlab, donde se despliega una ventana para el diseño de aplicaciones de redes neuronales como: fitting, reconocimiento de patrones, clustering y series de tiempo. Figura 2.36.

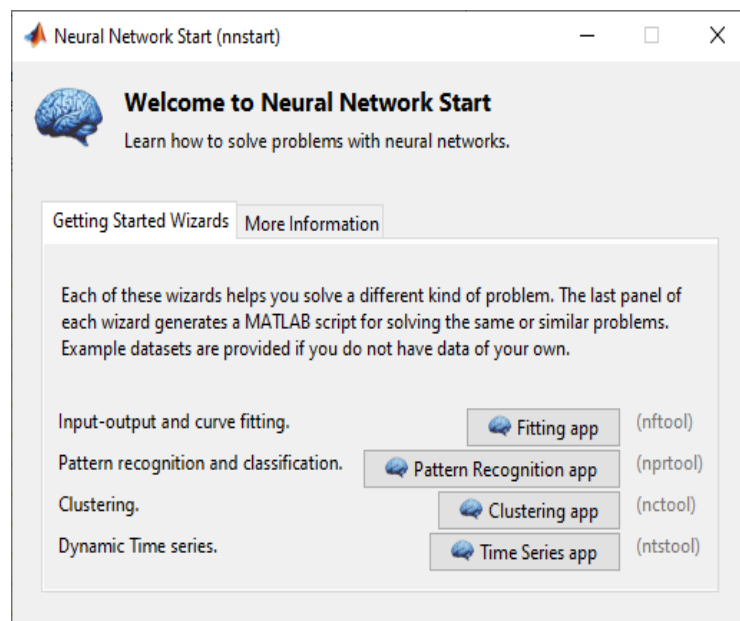


Figura 2.36. Menú de opciones para entrenamiento de redes neuronales artificiales

Para este trabajo se utiliza la opción de reconociendo de patrones en redes neuronales artificiales de Matlab como se muestra en la figura 2.37.

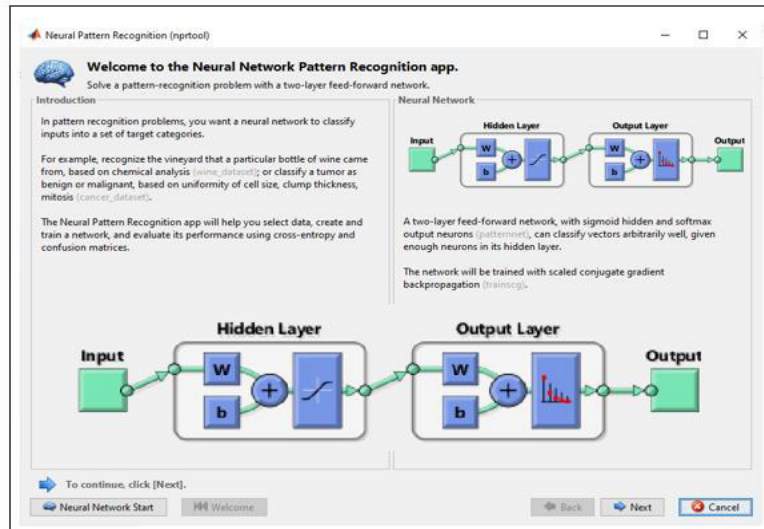


Figura 2.37. Aplicación de reconocimiento de patrones con redes neuronales

Las características de la red neuronal utilizada para el reconocimiento de patrones en Matlab se detallan a continuación:

Capa oculta

- ✓ **Función de malla:** Lineal Ponderada (Ecuación 2.1).

$$V = \sum_{j=1}^N W_j * Y_j + \beta \quad (2.1)$$

- ✓ **Función de activación:** Sigmoidal bipolar o hiperbólica (Ecuación 2.2).

$$Z = f(V) = \frac{1-e^{-V}}{1+e^{-V}} \quad (2.2)$$

Capa de salida

- ✓ **Función de malla:** Lineal Ponderada (Ecuación 2.3).

$$V = \sum_{j=1}^N W_j * Y_j + \beta \quad (2.3)$$

- ✓ **Función de activación:** Softmax para clasificación múltiples clases (Ecuación 2.4).

$$Z = f(V_i) = \frac{e^{V_i}}{\sum_{j=0}^J e^{V_j}} \quad (2.4)$$

La red neuronal para reconocimiento de patrones está formada de: una matriz de entradas, una matriz de targets o salidas, un número de neuronas especificado en la capa oculta, porcentaje de datos para entrenamiento, validación y testeo, el algoritmo de retropropagación utilizado y parámetros por defecto especificados en la tabla 2.4.

Tabla 2.4. Parámetros por defecto en una red neuronal

Parámetro	Valor	Descripción
net.trainParam.epochs	1000	Número máximo de épocas para entrenar
net.trainParam.goal	0	Objetivo de desempeño (error esperado)
Net.trainParam.time	Inf	Tiempo máximo para entrenar en segundos
Net.trainParam.min_grad	1e-6	Gradiente de rendimiento mínimo
Net.trainParam.max_fail	5	Máximo de fallos de validación

La matriz de entrada para la red neuronal tiene una dimensión de [8x250], donde 8 representa el vector obtenido en la etapa de extracción de características y 250 representa, 50 muestras de cada gesto estudiado por 5 gestos.

```
inputs=[8x250]; % área bajo las curvas de las señales de los 8 canales de EMG.
```

La matriz de salidas para la red neuronal o targets tiene una dimensión de [5x250], donde 5 representa los vectores de clase, que caracterizan a cada gesto estudiado.

```
wavein= [1;0;0;0;0]; % vector de clase WAVE IN
waveout= [0;1;0;0;0]; % vector de clase WAVE OUT
fist= [0;0;1;0;0]; % vector de clase WAVE FIST
open= [0;0;0;1;0]; % vector de clase WAVE OPEN
pinch= [0;0;0;0;1]; % vector de clase WAVE PINCH
targets=[repmat(wavein,1,50), repmat(waveout,1,50), repmat(fist,1,50),
, repmat(open,1,50), repmat(pinch,1,50)]; % matriz de clases para la red neuronal
```

La red neuronal se entrena con el método de gradiente conjugado escalado, el cual se encarga de actualizar los pesos, calculando la derivada parcial de la función de costo con respecto a cada uno de los pesos.

```
trainFcn = 'trainscg'; %algoritmo para entrenamiento
(backpropagation) Levenberg-Marquardt Backpropagation
```

El número de neuronas que se usan en la capa oculta son 15, ya que este número ha proporcionado el mayor porcentaje de clasificación correcta.

```
hiddenLayerSize=15; % número de neuronas para entrenamiento
net = patternnet(hiddenLayerSize); % inicialiación de la red neuronal
```

La división de datos para validación, entrenamiento, y testeo queda de la siguiente manera: 16% de los datos para validar o determinar cuándo detener el proceso de aprendizaje, 6%

de los datos para realizar el entrenamiento de la red neuronal y determinar los pesos y 78% para determinar el rendimiento de la red neuronal y mostrar su matriz de confusión.

```
% División de los datos para validación, entrenamiento y testeo
net.divideParam.trainRatio = 6/100;% porcentaje para entrenamiento
net.divideParam.valRatio = 16/100;% porcentaje para validación
net.divideParam.testRatio = 78/100;% porcentaje para testeo
```

A continuación se muestra el código completo para entrenar una red neuronal.

```
inputs[8x250]; % matriz de entrada
wavein= [1;0;0;0;0]; % vector de clase WAVE IN
waveout= [0;1;0;0;0]; % vector de clase WAVE OUT
fist= [0;0;1;0;0]; % vector de clase WAVE FIST
open= [0;0;0;1;0]; % vector de clase WAVE OPEN
pinch= [0;0;0;0;1]; % vector de clase WAVE PINCH
targets=[ repmat(wavein,1,50), repmat(waveout,1,50), repmat(fist,1,50)
), repmat(open,1,50), repmat(pinch,1,50)]; % matriz de clases para la
red neuronal

function [redes_neuronales] = redes_neuronales(inputs,targets)
    trainFcn = 'trainscg';%algoritmo para entrenamiento (backpropagation)
    hiddenLayerSize=15; % número de neuronas para entrenamiento
    net = patternnet(hiddenLayerSize); % inicialiación de la red neuronal
    % División de los datos para validación, entrenamiento y testeo
    net.divideParam.trainRatio = 6/100;% porcentaje para entrenamiento
    net.divideParam.valRatio = 16/100;% porcentaje para validación
    net.divideParam.testRatio = 78/100;% porcentaje para testeo
    [redes_neuronales,tr] = train(net,inputs,targets); % entrenamiento
    ntrainrtool close % cierre de la ventana de entrenamiento
end
```

El programa para evaluar la red neuronal queda de la siguiente manera

```
gesto_realizado=[E1;E2;E3;E4;E5;E6;E7;E8]; %vector donde se almacena el
área bajo la curva de los nuevos gestos realizados.
% DETERMINACION DE GESTO
y_5=redes_neuronales(gesto_realizado);%utilización de la red neuronal creada
[M, INDICE]=max(y_5);
[FI,WI,WO,OP,PI,abcd] = determinacion_gesto(INDICE);%función para mostrar
% el gesto realizado
```

La función determinación de gesto, muestra en pantalla la figura del gesto y en el comand windows de Matlab se muestra el nombre del gesto realizado.

El código completo para la determinación de los cinco gestos de la mano se muestra a continuación.

```

function [FI,WI,WO,OP,PI,abcd] = determinacion_gesto(LONGITUD)
    % imagenes a mostrar en pantalla de cada gesto
    imagen2=imread('FIST.jpg');
    imagen3=imread('OPEN.jpg');
    imagen4=imread('WAVE_IN.jpg');
    imagen5=imread('WAVE_OUT.jpg');
    imagen6=imread('PINCH.jpg');
    %DECISIÓN DE GESTO
    % condicional para mostrar en pantalla el gesto FIST
    if INDICE==3
        fprintf(1,'\n %s \n','FIST');
        imshow(imagen2);
        title('FIST')
        PI=0;
        FI=1;
        WI=0;
        OP=0;
        WO=0;
        abcd=1;
    % condicional para mostrar en pantalla el gesto WAVE IN
    elseif INDICE==1
        fprintf(1,'\n %s \n','WAVE IN');
        imshow(imagen4);
        title('WAVE IN')
        %toc
        PI=0;
        FI=0;
        WI=1;
        OP=0;
        WO=0;
        abcd=1;
    % condicional para mostrar en pantalla el gesto WAVE OUT
    elseif INDICE==2
        fprintf(1,'\n %s \n','WAVE OUT');
        imshow(imagen5);
        title('WAVE OUT')
        PI=0;
        FI=0;
        WI=0;
        OP=0;
        WO=1;
        abcd=1;
    % condicional para mostrar en pantalla el gesto OPEN
    elseif INDICE==4
        fprintf(1,'\n %s \n','OPEN');
        imshow(imagen3);
        title('OPEN')
        PI=0;
        FI=0;
        WI=0;
        OP=1;
        WO=0;
        abcd=1;
    % condicional para mostrar en pantalla el gesto PINCH
    elseif INDICE ==5
        fprintf(1,'\n %s \n','PINCH');
        imshow(imagen6);
        title('PINCH')
        PI=1;
        FI=0;
        WI=0;
        OP=0;
        WO=0;
        abcd=1;
    end
end

```

2.6 VARIABILIDAD INTRAPERSONAL E INTERPERSONAL

2.6.1 VARIABILIDAD INTRAPERSONAL

La variabilidad intrapersonal aplicado al reconocimiento de gestos de la mano, consiste en verificar cuan estable o inestable es el sistema al corto o largo plazo, bajo ciertas condiciones. Un análisis de variabilidad intrapersonal aplicado al sistema, consiste en verificar la estabilidad de los clústers. La evaluación de los centroides a lo largo del tiempo, se basa en 3 parámetros:

- **Ubicación de los centroides**

Se analiza si a lo largo del tiempo, los centroides mantienen la misma posición.

- **Distancia entre centroides**

Se analiza si a lo largo del tiempo, la distancia entre los centroides se mantiene relativamente estable.

- **Desviación estándar de los datos**

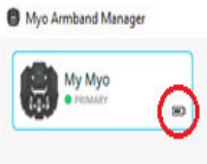
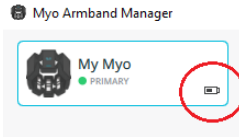
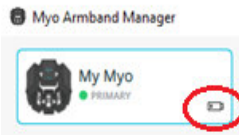




Se analiza si a lo largo del tiempo, la dispersión de los datos cambia de manera considerable.

Para realizar un análisis de variabilidad intrapersonal se ha considerado identificar las posibles variables que pueden influir directamente en el movimiento de los centroides. Estas variables se han clasificado según el tiempo: a largo plazo y a corto plazo y según el tipo: controlables y no controlables.

Los cambios que se realizó para el análisis de variables a corto plazo controlables fueron: cambiar el nivel de batería del dispositivo Myo Armband, colocar el dispositivo a diferente distancia desde el codo al Myo y colocar el dispositivo en diferente posición respecto al giro.

La tabla 2.5, muestra las posibles variables a corto plazo controlables, el análisis que se realiza y los cambios que se han aplicado para verificar su influencia.

Tabla 2.5. Variables intrapersonales a corto plazo controlables

Según el tiempo	Según el tipo	Posibles variables	Análisis	Cambios realizados	
A cortoplazo	Controlables	Nivel de batería	Se analiza la distancia entre centroides al cambiar el nivel de batería	100%	
			Se analiza la desviación estandar al cambiar el nivel de batería	50%	
			Se analiza la ubicación de centroides al cambiar el nivel de batería	<50%	
		Distancia codo-myo	Se analiza la distancia entre centroides al cambiar la distancia codo-myo	6 cm	
			Se analiza la dispersión de datos al cambiar la distancia codo-myo	11 cm	
		Se analiza la ubicación de centroides al cambiar la distancia codo-myo			
		Giro de brazaletes	Se analiza cómo influye la colocación del brazaletes en giro respecto a la ubicación de los centroides	0°	
				180°	

La tabla 2.6, muestra las posibles variables a corto plazo que no son controlables, para estas variables no se ha realizado ningún cambio y ningún análisis.

Tabla 2.6. Variables intrapersonales a corto plazo no controlables

Tiempo	Tipo	Posibles variables	Análisis	Cambios a realizar
A cortoplazo	No controlables	Temperatura	Se mide la temperatura ambiental en el momento de obtener las señales emg	X
		Grado de humedad	X	X
		Estado de ánimo	X	X
		Precisión en la colocación (giro)	X	X
		Precisión en la colocación (longitud codo-my)	X	X

La tabla 2.7, muestra las posibles variables a largo plazo que no son controlables, para estas variables no se ha realizado ningún cambio y ningún análisis.

Tabla 2.7. Variables intrapersonales a largo plazo no controlables

Tiempo	Tipo	Posibles variables	Análisis	Cambios a realizar
A largo plazo	No controlable	Cambio en la contextura corporal	X	X
		Nivel de práctica	X	X
		Otras	X	X

2.6.2 VARIABILIDAD INTERPERSONAL


Un análisis de variabilidad interpersonal aplicado al sistema de reconocimiento de gestos de la mano, consiste en verificar los cambios que experimentan los centroides al utilizar el sistema en diferentes usuarios.

Se considera los mismos parámetros que en la variabilidad intrapersonal.

- **Ubicación de los centroides**
Se analiza si a lo largo del tiempo, los centroides mantienen la misma posición.
- **Distancia entre centroides**
Se analiza si a lo largo del tiempo, la distancia entre los centroides se mantiene relativamente estable.
- **Desviación estándar de los datos**
Se analiza si a lo largo del tiempo, la dispersión de los datos cambia de manera considerable.


Para realizar un análisis de variabilidad interpersonal se ha considerado identificar las posibles variables que pueden influir directamente en un movimiento de los clústers. Las posibles variables se han clasificado según el tiempo: a largo y corto plazo y según el tipo: controlables y no controlables. A las posibles variables descritas en la variabilidad intrapersonal se añaden dos variables más que son características de cada persona. Estas variables son: dimensión de antebrazo y sexo. La tabla 2.8 muestra las posibles variables a corto plazo que son controlables y el análisis realizado.

Tabla 2.8. Variables interpersonales a corto plazo controlables

Tiempo	Tipo	Posibles variables	Análisis	Cambios a realizar
A corto plazo	Controlables	Distancia codo-myo	Para el análisis se toma en cuenta una misma distancia de colocación codo - myo de 7cm	X
		Giro de brazalete	Para el análisis se toma en cuenta un mismo giro de brazalete de 0° (centrado arriba)	X
		Sexo (varón o mujer)	Se analiza la ubicación de los centroides para hombres y mujeres	

El cambio que se realizó para el análisis de variables a corto plazo no controlables fue obtener diferentes usuarios con diferente perímetro de antebrazo. La tabla 2.9, muestra las posibles variables a corto plazo no controlables.

Tabla 2.9. Variables interpersonales a corto plazo no controlables

Tiempo	Tipo	Posibles variables	Análisis	Cambios a realizar
A corto plazo	No controlables	Dimensión de antebrazo	Se analizará la ubicación de los clústers con diferentes usuarios de similar y diferente textura de antebrazo	
		Temperatura	X	X
		Grado de humedad	X	X
		Estado de ánimo	X	X
		Precisión en la colocación (giro)	X	X
		Precisión en la colocación (longitud codo-myo)	X	X

La tabla 2.10, muestra las posibles variables a largo plazo no controlables. Para estas variables no se ha realizado ningún cambio.

Tabla 2.10. Variable interpersonal a largo plazo no controlable

Tiempo	Tipo	Posibles variables	Análisis	Cambios a realizar
A largo plazo	No controlables	Nivel de práctica	Se tomará en cuenta si los usuarios analizados han practicado antes de adquirir las señales emg para analizar clasificarlos como: Bajo (primera vez) medio (varias veces) alto (usuario permanente)	X

3. RESULTADOS Y DISCUSIÓN

3.1 COMPARACIÓN DE LOS TRES DOMINIOS Y SELECCIÓN

Con el fin de comprobar y escoger la variable (dominio) de la etapa de extracción de características que presente los mejores resultados se construyó una base de datos de un mismo usuario con las siguientes características:

Sujeto de experimentación

- Sexo: varón
- Edad: 25 años
- Dimensión de antebrazo: 28 cm

Colocación del brazalete

- Giro de brazalete: Logo centrado hacia arriba (0°)
- Distancia del codo al myo armband: 6 cm

Base de datos de las señales EMG

- 25 muestras del gesto fist
- 25 muestras del gesto wave in
- 25 muestras del gesto wave out
- 25 muestras del gesto open
- 25 muestras del gesto pinch

- Cada muestra posee las 8 señales EMG de cada canal

La selección de la mejor variable se define de acuerdo a los siguientes parámetros:

- La mayor separación entre clústers
- La menor dispersión de los datos (desviación estándar)
- La menor carga computacional.

Para poder comparar las tres variables de la etapa de extracción de características es necesario realizar una normalización de los valores obtenidos, ya que cada dominio presenta diferente escala y realizar una reducción de dimensionalidad 8 a 2. La figura 3.1, muestra la metodología para la selección de la mejor variable de los clústers normalizados utilizando: la distancia euclidiana entre centroides y la dispersión de datos.

Comparación entre Variables

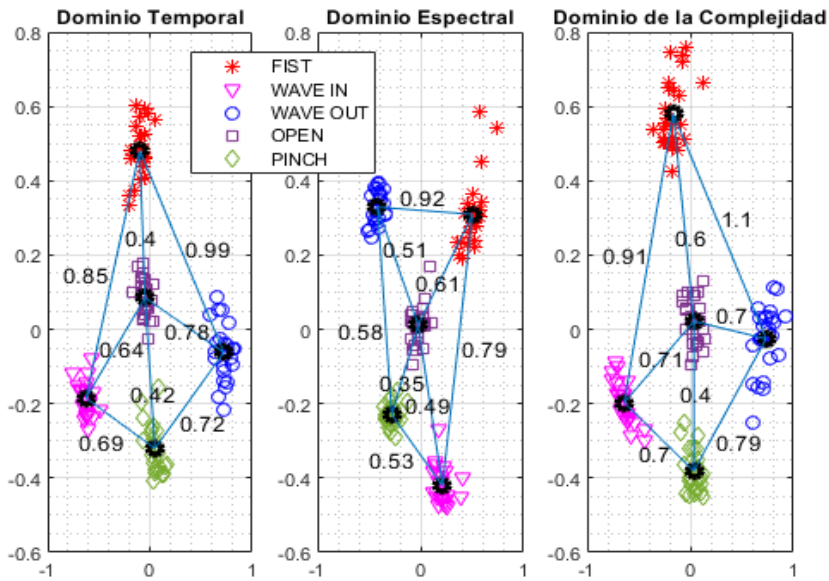


Figura 3.1. Clústers normalizados y distancia entre centroides

3.1.1 DISTANCIA ENTRE CENTROIDES

La distancia entre centroides consiste en verificar que tan separados se encuentran los cinco centroides de cada gesto.

La tabla 3.1, muestra el resultado de la distancia entre centroides para los pares de gestos. Los valores marcados en azul representan la mayor distancia conseguida.

Tabla 3.1. Distancia entre centroides

	Dominio Tiempo	Dominio Frecuencia	Dominio Complejidad
Fist-Wave in	0,85	0,79	0,91
Fist-Wave out	0,99	0,92	1,1
Fist-Open	0,4	0,61	0,6
Fist-Pinch	0,82	0,95	0,98
Wave in-Wave out	1,42	0,98	1,4
Wave in-Open	0,64	0,49	0,71
Wave in-Pinch	0,69	0,53	0,7
Wave out - Open	0,78	0,51	0,7
Wave out - Pinch	0,72	0,58	0,79
Open - Pinch	0,42	0,35	0,4

Los resultados muestran que el dominio de la complejidad posee la mayor separación entre pares de centroides para la mayoría de gestos, aunque los demás dominios igualmente consiguen gran separación.

3.1.2 DISPERSIÓN DE DATOS

La desviación estándar representa cuan dispersos se encuentran los datos respecto a su centroide. La tabla 3.2, muestra la desviación estándar de las 25 muestras de cada gesto. A la izquierda de la tabla se especifica en color azul, el dominio con la menor dispersión de los datos (desviación estándar), lo que implica un menor porcentaje de error al momento de clasificar.

Tabla 3.2. Desviación estándar de las 25 muestras de cada gesto

		Desviación Estándar	
		Componente principal 1 (eje x)	Componente principal 2 (eje y)
Dominio tiempo	Fist	0.0629	0.0722
Dominio frecuencia		0.0821	0.0955
Dominio complejidad		0.0948	0.0931
Dominio tiempo	Wave in	0.0530	0.0453
Dominio frecuencia		0.0686	0.0491
Dominio complejidad		0.0812	0.0574
Dominio tiempo	Wave out	0.0625	0.0743
Dominio frecuencia		0.0424	0.0425
Dominio complejidad		0.0808	0.0892
Dominio tiempo	Open	0.0450	0.0518
Dominio frecuencia		0.0529	0.0534
Dominio complejidad		0.0632	0.0625
Dominio tiempo	Pinch	0.0534	0.0288
Dominio frecuencia		0.0628	0.0645
Dominio complejidad		0.0617	0.0574

Si se analiza por gesto, se puede observar que el dominio temporal presenta la menor desviación estándar en cada eje de las 2 componentes principales analizadas.

3.1.3 CARGA COMPUTACIONAL

Analizar la carga computacional de cada dominio es clave al momento de implementar un sistema de reconocimiento de gestos de la mano en tiempo real. Para que el sistema de reconocimiento de gestos de mano funcione en tiempo real, se debe considerar un tiempo de respuesta menor a 300 ms. La tabla 3.3, muestra la carga computacional promedio desde la adquisición de señales, hasta la extracción de características para cada variable analizada (dominios), utilizando la función *tic – toc* proporcionada por Matlab.

Tabla 3.3. Carga computacional de los tres dominios

	Carga computacional
Dominio tiempo	27.4 [ms]
Dominio frecuencia	377.9 [ms]
Dominio complejidad	39.4 [ms]

El dominio del tiempo presenta la menor carga computacional, seguido del dominio de la complejidad, lo cual los hace idóneas para ser implementados en un sistema en tiempo real, al contrario el dominio de la frecuencia que presenta un tiempo de respuesta superior al valor mínimo considerado como tiempo real en reconocimiento de gestos de la mano.

En la implementación del algoritmo, la variable de área bajo las envolventes de las 8 señales EMG (dominio del tiempo) es utilizada debido a la simplicidad de los métodos utilizados (menor carga computacional en las funciones creadas en Matlab), la menor dispersión de los datos (desviación estándar) y su gran separación entre centroides.

3.2 VARIABILIDAD INTRAPERSONAL

Para la evaluación de variabilidad intrapersonal se determinó que el dominio temporal presenta las mejores características en cuanto a separación de gestos y carga computacional. Por ello, la variable que se analiza en este dominio es el área bajo la curva de las 8 señales EMG. De esta manera los resultados que se muestran a continuación, reflejan cuan estable es el área bajo las envolventes de las señales, en un periodo de tiempo establecido para un mismo sujeto de experimentación. Por lo tanto, con este análisis se determina si es factible crear un perfil único que sirva para todas las sesiones de uso del programa o crear un nuevo perfil en cada sesión.

Para la medición de las señales EMG a lo largo del tiempo, se especificaron las siguientes características de medición:

- Se adquirió las señales EMG de los 5 gestos durante 2 meses.
- Se adquirió las señales EMG de los 5 gestos durante 2 días a la semana: miércoles y domingo.
- Se adquirió las señales EMG de los 5 gestos durante 2 veces al día: mañana y noche.
- Se adquirió 25 muestras de cada gesto en cada tiempo de adquisición de señales.

El usuario para el análisis de la variabilidad intrapersonal presenta las siguientes características:

- Varón de 24 años
- Perímetro de antebrazo: 28 cm

Las variables controlables que se analizaron fueron:

1. Nivel de carga del dispositivo myo Armband
2. Colocación del brazalete en distancia (codo - Myo).
3. Colocación del brazalete en giro.

La variable no controlable que se pudo medir fue:

- La temperatura

La constante en la medición fue:

- El perímetro del antebrazo (El usuario analizado no ha experimentado grandes cambios corporales en los dos meses de prueba)

3.2.1 NIVEL DE BATERÍA

Con el fin de comprobar el funcionamiento del sistema, se ha realizado mediciones de las señales EMG de los 5 gestos con distintos niveles de carga del dispositivo, para verificar tres parámetros de estabilidad:

- Ubicación de los centroides
- Distancia entre los centroides
- Dispersión de los datos (desviación estándar)

Para verificar la influencia del nivel de carga del dispositivo y se ha definido tres niveles de batería: al 100%, al 50% y menor al 50%. Las figuras 3.2, 3.3 y 3.4, muestran el resultado de cambiar el nivel de batería del dispositivo Myo.

- Carga al 100% con colocación del brazalete a 6cm y 0°

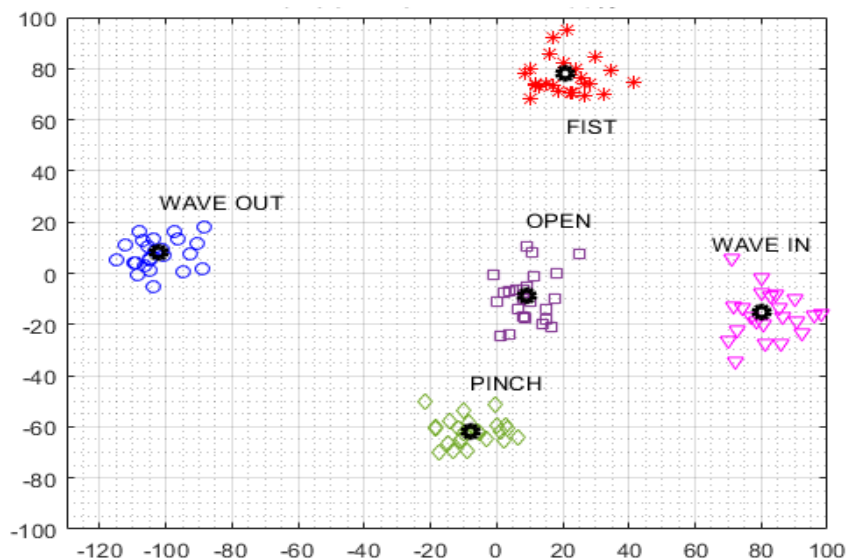


Figura 3.2. Clústers con un nivel de batería al 100% - 06/12/2020

- Carga al 50 % con colocación del brazalete a 6cm y 0°

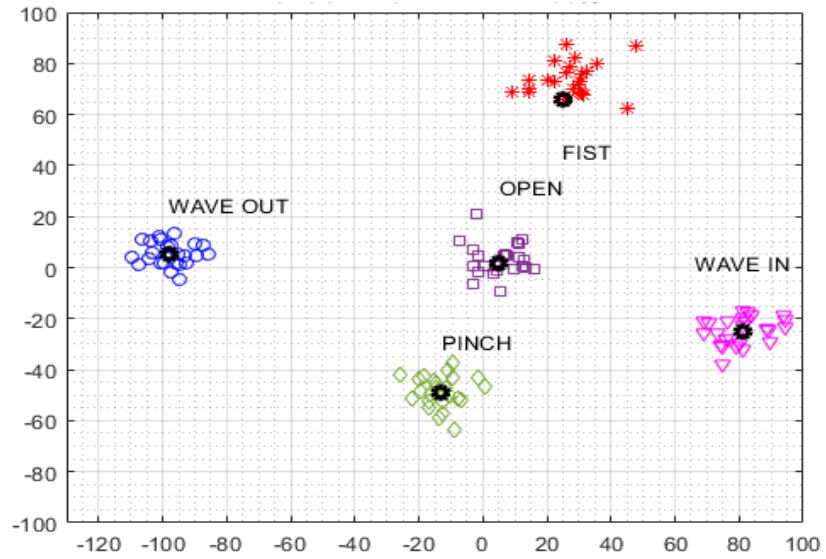


Figura 3.3. Clústers con un nivel de batería al 50% - 16/12/2020

- Carga menor al 50% con colocación del brazalete a 6cm y 0°

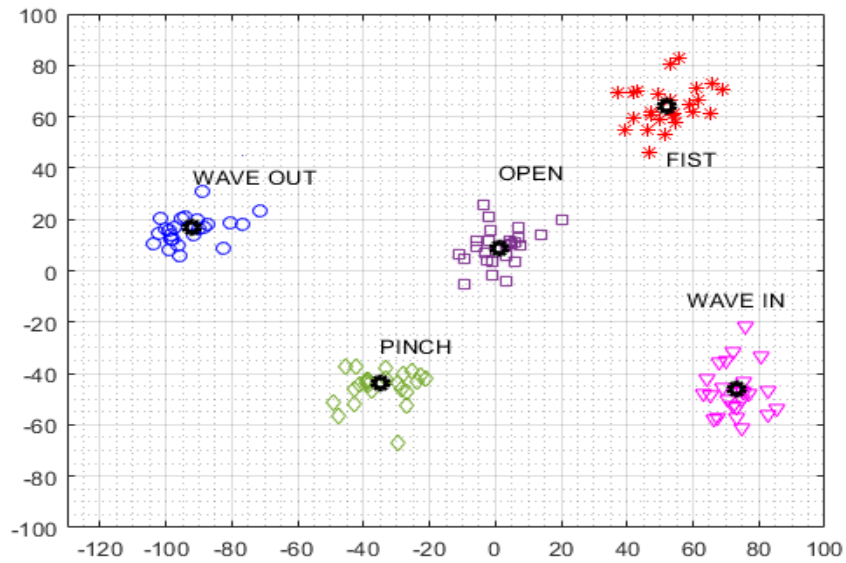


Figura 3.4. Clústers con un nivel de batería menor al 50% - 28/12/2020

3.2.1.1 Distancia entre centroides

La tabla 3.4, muestra la distancia entre pares de centroides conseguida en diferentes días, para distintos niveles de batería y la misma colocación del brazalete.

Tabla 3.4. Distancia entre centroides

Fecha	Tem	Bat [%]	Giro	Dist	Distancia									
					Fi-Wi	Fi-Wo	Fi-Op	Fi-Pi	Wi-Wo	Wi-Op	Wi-Pi	Wo-Op	Wo-Pi	Op-Pi
02/12/2020	17°	100	0°	6	107	145	94	136	172	92	104	117	104	69
02/12/2020	18°	100	0°	6	113	151	81	137	184	90	112	120	108	86
06/12/2020	17°	100	0°	6	113	143	96	141	184	86	104	120	120	71
06/12/2020	17°	100	0°	6	112	153	96	124	185	86	92	123	113	66
09/12/2020	17°	100	0°	6	106	138	80	125	181	79	96	133	118	75
09/12/2020	17°	100	0°	6	116	159	92	132	195	92	109	128	110	66
13/12/2020	17°	100	0°	6	116	154	90	131	186	95	105	120	103	70
13/12/2020	18°	50	0°	6	114	159	88	144	189	95	109	119	112	80
16/12/2020	17°	50	0°	6	108	139	78	121	181	94	102	112	104	66
16/12/2020	18°	50	0°	6	107	151	85	122	186	94	99	119	108	69
05/01/2021	14°	50	0°	6	123	161	90	140	184	104	106	111	104	73
20/12/2020	18°	100	0°	6	107	140	90	115	191	100	106	121	108	66
20/12/2020	18°	100	0°	6	116	153	79	142	188	98	107	121	113	84
23/12/2020	16°	100	0°	6	116	140	88	120	192	94	103	128	109	69
23/12/2020	17°	100	0°	6	108	152	79	118	180	93	93	112	102	62
27/12/2020	15°	100	0°	6	103	153	90	120	174	98	97	111	108	71
05/01/2021	18°	100	0°	6	117	143	77	118	185	94	87	123	117	70
28/12/2020	17°	<50	0°	6	115	154	86	139	178	102	114	105	92	75
10/01/2021	22°	<50	0°	6	118	143	75	130	180	108	99	113	109	87
13/01/2021	21°	<50	0°	6	124	149	81	138	184	105	103	113	109	80

Se observó que al cambiar el nivel de carga, las distancias entre centroides se mantienen en valores dentro de un mismo rango, por este motivo se demuestra que esta variable no está afectando a la distancia entre centroides.

3.2.1.2 Ubicación de clústers

La tabla 3.5, muestra el centroide de las 25 repeticiones de cada gesto para distintos niveles de batería y para una misma colocación del brazalete.

Tabla 3.5. Ubicación de centroides

Fecha	Tem	Bat [%]	Giro	Dist	Centroide		
					Componente principal 1 (eje x)	Componente principal 2 (eje y)	
02/12/2020	17°	100%	0°	6	36	62	Fist
02/12/2020	18°	100%	0°	6	34	78	
06/12/2020	17°	100%	0°	6	37	68	
13/12/2020	18°	50%	0°	6	47	67	
16/12/2020	17°	50%	0°	6	25	60	
16/12/2020	18°	50%	0°	6	42	64	
05/01/2021	14°	<50%	0°	6	34	87	
10/01/2021	22°	<50%	0°	6	26	66	
13/01/2021	21°	<50%	0°	6	29	76	
02/12/2020	17°	100%	0°	6	67	-37	Wave in
02/12/2020	18°	100%	0°	6	83	-17	
06/12/2020	17°	100%	0°	6	77	-33	
13/12/2020	18°	50%	0°	6	81	-36	
16/12/2020	17°	50%	0°	6	79	-31	
16/12/2020	18°	50%	0°	6	81	-30	
05/01/2021	14°	<50%	0°	6	81	-24	
10/01/2021	22°	<50%	0°	6	75	-40	
13/01/2021	21°	<50%	0°	6	78	-37	
02/12/2020	17°	100%	0°	6	-98	12	Wave out
02/12/2020	18°	100%	0°	6	-99	12	
06/12/2020	17°	100%	0°	6	-97	23	
13/12/2020	18°	50%	0°	6	-100	9	
16/12/2020	17°	50%	0°	6	-98	3	
16/12/2020	18°	50%	0°	6	-99	14	
05/01/2021	14°	<50%	0°	6	-101	3	
10/01/2021	22°	<50%	0°	6	-99	0	
13/01/2021	21°	<50%	0°	6	-100	5	
02/12/2020	17°	100%	0°	6	-13	-12	Open
02/12/2020	18°	100%	0°	6	12	14	
06/12/2020	17°	100%	0°	6	10	-8	
13/12/2020	18°	50%	0°	6	11	2	
16/12/2020	17°	50%	0°	6	3	-4	
16/12/2020	18°	50%	0°	6	10	6	
05/01/2021	14°	<50%	0°	6	0	19	
10/01/2021	22°	<50%	0°	6	0	18	
13/01/2021	21°	<50%	0°	6	-1	16	
02/12/2020	17°	100%	0°	6	-27	-58	Pinch
02/12/2020	18°	100%	0°	6	-17	-49	
06/12/2020	17°	100%	0°	6	-15	-61	
13/12/2020	18°	50%	0°	6	-16	-62	
16/12/2020	17°	50%	0°	6	-16	-52	
16/12/2020	18°	50%	0°	6	-12	-45	
05/01/2021	14°	<50%	0°	6	-16	-43	
10/01/2021	22°	<50%	0°	6	-16	-55	
13/01/2021	21°	<50%	0°	6	-20	-50	

Se observó que, al variar el nivel de carga del dispositivo myo armband, los centroides se encuentran en valores dentro de un mismo rango, por este motivo se demuestra que esta variable no está afectando a la ubicación de los clústers.

3.2.1.3 Dispersión de clústers

Respecto a la desviación estándar cuando se cambia el nivel de carga del dispositivo, no se ha observado cambios significativos.

3.2.2 DISTANCIA CODO - MYO

Con el fin de comprobar el funcionamiento del algoritmo, se ha realizado mediciones de las señales EMG de los cinco gestos, cambiando la distancia de colocación del brazalete y se obtuvo los clústers de cada gesto, para verificar tres parámetros de estabilidad:

- Distancia entre los centroides
- Ubicación de los centroides
- Dispersión de los datos

Para verificar cómo influye la distancia de colocación del brazalete, se ha definido dos distancias de medición: distancia codo – myo a 6 cm y distancia codo – myo a 11 cm. La figura 3.5 muestra el resultado de cambiar la distancia en la colocación del brazalete.

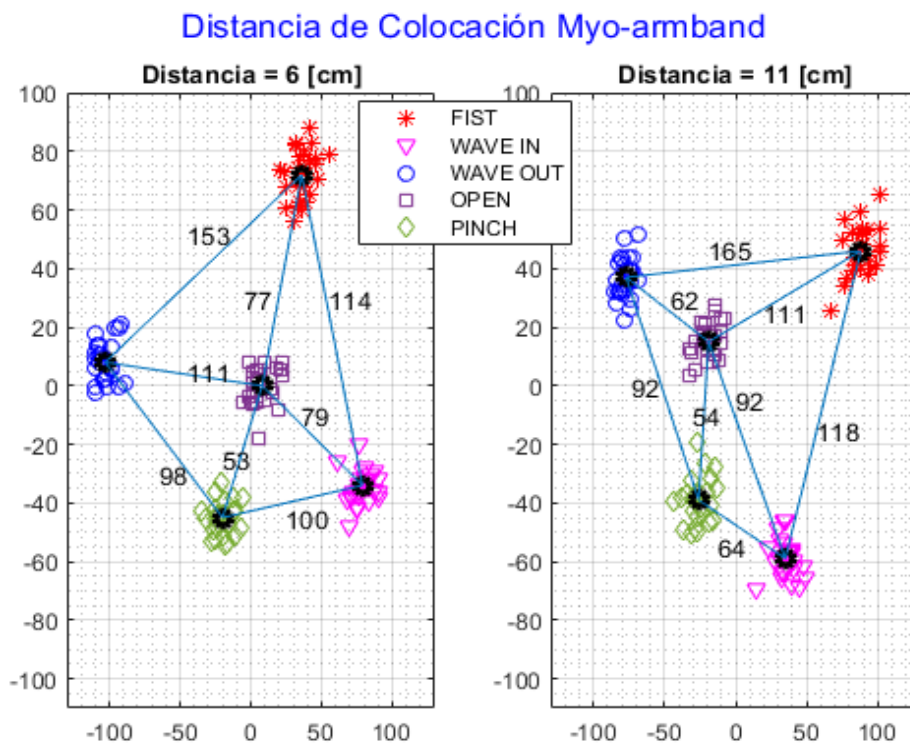


Figura 3.5. Clústers con una distancia codo-myo de 6 y 11 cm

3.2.2.1 Distancia entre centroides

La tabla 3.6, muestra la distancia entre centroides en diferentes días de medición, para las dos distancias de colocación del brazalete.

Tabla 3.6. Distancia entre centroides

Fecha	Tem	Bat [%]	Giro	Dist	Distancia									
					Fi-Wi	Fi-Wo	Fi-Op	Fi-Pi	Wi-Wo	Wi-Op	Wi-Pi	Wo-Op	Wo-Pi	Op-Pi
02/12/2020	17°	100	0°	6	107	145	94	136	172	92	104	117	104	69
02/12/2020	18°	100	0°	6	113	151	81	137	184	90	112	120	108	86
06/12/2020	17°	100	0°	6	113	143	96	141	184	86	104	120	120	71
06/12/2020	17°	100	0°	6	112	153	96	124	185	86	92	123	113	66
22/12/2020	17°	100	0°	11	108	164	108	140	159	95	85	97	89	74
23/12/2020	17°	100	0°	11	101	166	114	123	158	88	83	102	95	67
24/12/2020	16°	100	0°	11	102	155	104	120	161	99	75	101	87	64
13/01/2021	22°	100	0°	11	120	167	121	144	149	99	71	80	95	76

En todas las pruebas realizadas se puede observar que, a mayor distancia de colocación del brazalete, menor distancia entre los centroides de los gestos: Wave in – Wave out, Wave in – Pinch, Wave out – Open, Wave out – Pinch y mayor distancia Fist – Open. Esto implica una disminución en la exactitud de clasificación de los gestos.

3.2.2.2 Ubicación de clústers

La tabla 3.7, muestra la ubicación de los centroides en diferentes días de medición, para las dos distancias de colocación del brazalete.

Tabla 3.7. Ubicación de centroides

Fecha	Temp [°C]	Bat [%]	Giro	Dist	Centroide		
					Componente principal 1 (eje x)	Componente principal 2 (eje y)	
02/12/2020	18°	100%	0°	6	38	69	Fist
06/12/2020	17°	100%	0°	6	42	61	
06/12/2020	17°	100%	0°	6	31	57	
22/12/2020	17°	100%	0°	11	58	66	
23/12/2020	17°	100%	0°	11	55	57	
24/12/2020	16°	100%	0°	11	50	67	
02/12/2020	18°	100%	0°	6	83	-22	Wave in
06/12/2020	17°	100%	0°	6	75	-37	
06/12/2020	17°	100%	0°	6	77	-44	
22/12/2020	17°	100%	0°	11	55	-33	
23/12/2020	17°	100%	0°	11	53	-34	
24/12/2020	16°	100%	0°	11	64	-27	
02/12/2020	18°	100%	0°	6	-96	16	Wave out
06/12/2020	17°	100%	0°	6	-95	26	
06/12/2020	17°	100%	0°	6	-103	14	
22/12/2020	17°	100%	0°	11	-96	21	
23/12/2020	17°	100%	0°	11	-94	10	
24/12/2020	16°	100%	0°	11	-98	30	
02/12/2020	18°	100%	0°	6	14	-14	Open
06/12/2020	17°	100%	0°	6	7	-10	
06/12/2020	17°	100%	0°	6	5	-10	
22/12/2020	17°	100%	0°	11	-11	8	

23/12/2020	17°	100%	0°	11	-7	3	Pinch
24/12/2020	16°	100%	0°	11	-9	4	
02/12/2020	18°	100%	0°	6	-19	-53	
06/12/2020	17°	100%	0°	6	-17	-65	
06/12/2020	17°	100%	0°	6	-14	-58	
22/12/2020	17°	100%	0°	11	-23	-47	
23/12/2020	17°	100%	0°	11	-21	-46	
24/12/2020	16°	100%	0°	11	-17	-38	

Se observó que al cambiar la distancia de colocación del brazalete de 6 cm a 11 cm, existe un cambio de ubicación de los clústers, lo que implica que la distancia codo-myo de colocación influye negativamente en la exactitud de clasificación si se entrena un sistema con datos pregrabados.

3.2.2.3 Dispersión de clústers

Respecto a la desviación estándar cuando se cambia la colocación del brazalete, no se ha observado cambios significativos.

3.2.3 GIRO DE BRAZALETE

Con el fin de verificar lo que sucede con los clústers de gestos y sus centroides al colocarse el brazalete en diferente posición respecto al giro, se ha realizado mediciones de las señales EMG de los cinco gestos estudiados para verificar tres parámetros de estabilidad:

- Distancia entre los centroides
- Ubicación de los centroides
- Dispersión de los datos

Para verificar como influye la colocación del brazalete en giro, se ha definido dos posiciones de giro: a 0° y a 180°, con se muestra en la tabla 2.5 del capítulo anterior. Las figura 3.6 muestra el resultado de realizar un giro en la colocación del brazalete.

Giro de Colocación Myo-Amband

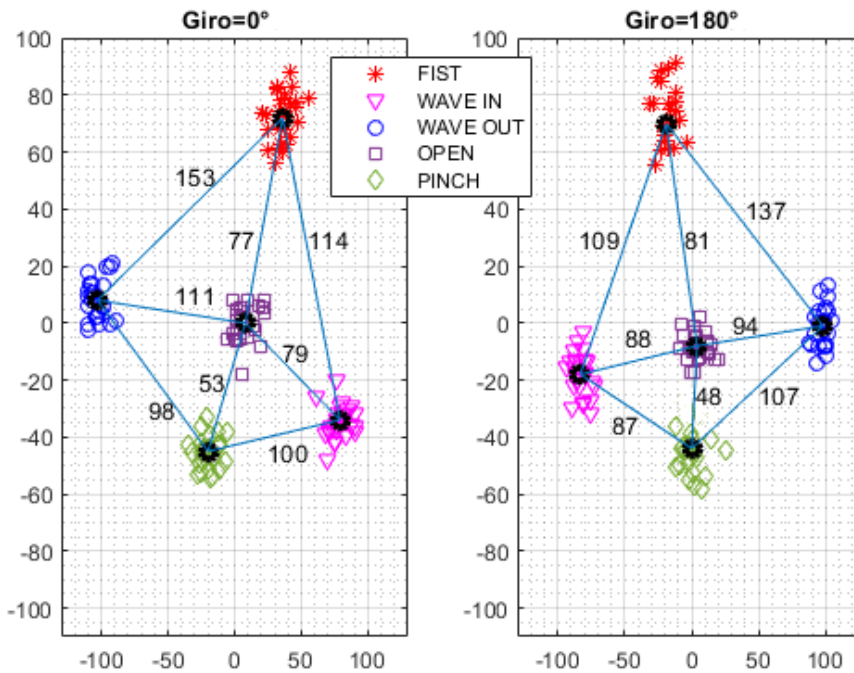


Figura 3.6. Clústers con giro de colocación de 0° y 180°

3.2.3.1 Distancia entre centroides

Al girar el brazalete en las posiciones indicadas, no se observó cambios significativos en la distancia entre centroides, lo que quiere decir que el giro de colocación no afecta la distancia entre centroides.

3.2.3.2 Ubicación de clústers

La tabla 3.8, muestra la ubicación de los centroides en diferentes días de medición, para los dos giros de colocación del brazalete.

Tabla 3.8. Ubicación de centroides

Fecha	Temp [°c]	Bat [%]	Giro	Dist	CENTROIDE		
					Componente principal 1 (eje x)	Componente principal 2 (eje y)	
20/12/2020	18°	100%	0°	6	-4	40	FIST
20/12/2020	18°	100%	0°	6	2	61	
23/12/2020	16°	100%	0°	6	3	43	
10/01/2021	22°	100%	180°	6	-55	23	
17/01/2021	18°	100%	180°	6	-82	12	
17/01/2021	17°	100%	180°	6	-68	12	
20/12/2020	18°	100%	0°	6	-66	4	WAVE IN
20/12/2020	18°	100%	0°	6	-54	16	
23/12/2020	16°	100%	0°	6	-58	9	
10/01/2021	22°	100%	180°	6	-28	-95	
17/01/2021	18°	100%	180°	6	-33	-84	
17/01/2021	17°	100%	180°	6	-28	-89	

20/12/2020	18°	100%	0°	6	110	-21	WAVE OUT
20/12/2020	18°	100%	0°	6	109	-35	
23/12/2020	16°	100%	0°	6	110	-35	
10/01/2021	22°	100%	180°	6	34	75	
17/01/2021	18°	100%	180°	6	22	61	
17/01/2021	17°	100%	180°	6	24	79	
20/12/2020	18°	100%	0°	6	22	30	OPEN
20/12/2020	18°	100%	0°	6	31	29	
23/12/2020	16°	100%	0°	6	24	28	
10/01/2021	22°	100%	180°	6	-37	-18	
17/01/2021	18°	100%	180°	6	-39	-3	
17/01/2021	17°	100%	180°	6	-43	-4	
20/12/2020	18°	100%	0°	6	35	-17	PINCH
20/12/2020	18°	100%	0°	6	40	-24	
23/12/2020	16°	100%	0°	6	37	-21	
10/01/2021	22°	100%	180°	6	1	-29	
17/01/2021	18°	100%	180°	6	-4	-22	
17/01/2021	17°	100%	180°	6	-4	-21	

Se observó que al cambiar el giro de colocación del brazalet de 0° a 180 ° existe un cambio de ubicación de los centroides, tal como era de esperarse, ya que simplemente los canales de medición cambian de posición.

3.2.3.3 Dispersión de clústers

No se observó cambios significativos en la desviación estándar de los datos al girar la posición del brazalet, se mantiene dentro de un rango estable que no afecta la clasificación.

3.3 VARIABILIDAD INTERPERSONAL

En el análisis de variabilidad interpersonal, los resultados que se muestran a continuación reflejan el comportamiento de los centroides y sus clústers, para diferentes sujetos de experimentación (hombres y mujeres) con distinto perímetro de antebrazo.

La medición de las señales EMG de los distintos sujetos de experimentación se ha definido de la siguiente manera:

- Se pudo obtener las señales EMG de 12 sujetos de experimentación.
- 8 sujetos de experimentación son varones.
- 4 sujetos de experimentación son mujeres
- La cantidad de muestras tomadas a cada sujeto de experimentación son 25 muestras de cada gesto.

Las variables controlables que se analizaron fueron:

- El perímetro del antebrazo (Todos los usuarios tienen perímetros de antebrazo distintos).
- Sexo (Varón o Mujer)
- Nivel de práctica de cada usuario antes de realizar los gestos
El nivel de práctica se ha considerado según los siguientes niveles:
 - Alto (usuario permanentemente realizando los gestos)
 - Medio (usuario a realizado los gestos algunas veces)
 - Bajo (usuario a realizado los gestos por primera vez)

La variable no controlable que ha sido posible medir es:

- La temperatura

Los valores constantes en la medición son:

- Nivel de carga del dispositivo myo Armband: 100 %
- Colocación del brazalete desde el codo al myo Armband: 6 cm
- Colocación del brazalete en giro: 0°

Para analizar el funcionamiento del sistema con diferentes usuarios, se estableció un rango de distancias entre los 5 centroides, que indica que tan cerca o lejos se encuentran unos de otros, para proporcionar una visión más clara de cuan efectivo es la separación para facilitar la clasificación.

Los valores establecidos se definieron al analizar visualmente los datos obtenidos como se muestra en las siguientes figuras 3.7, 3.8 y 3.9.

La figura 3.7, muestra los clústers de gestos bastante pegados, lo que significa que el sistema no clasificará correctamente los gestos.

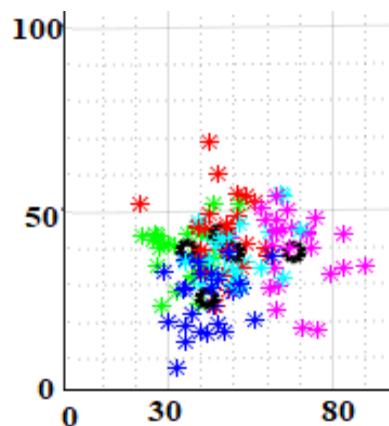


Figura 3.7. Distancias menores a 25 unidades

La figura 3.8, muestra los clústers de los gestos separados algunas unidades, lo que significa que el sistema clasifica correctamente la mayoría de gestos.

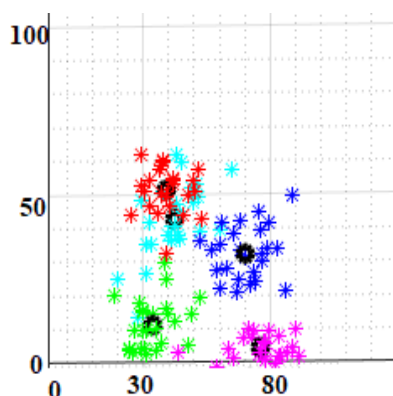


Figura 3.8. Distancias entre 25 y 50 unidades

La figura 3.9, muestra nubes de gestos bastante alejados, lo que significa que el sistema clasifica los gestos sin ningún error.

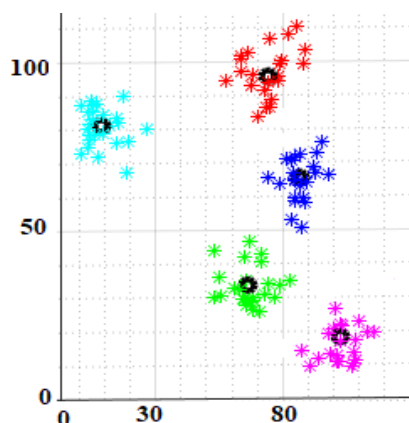


Figura 3.9. Distancias mayores a 50 unidades

En la tabla 3.9, se muestra los rangos y colores establecidos para identificar de mejor manera todo el conjunto de datos.

Tabla 3.9. Rango de distancias entre centroides

RANGO
DISTANCIA < 25
50 ≥ DISTANCIA ≥ 25
DISTANCIA > 50

Los valores en color verde indican que existen más de 50 unidades de separación (distancia euclidiana) entre los 5 gestos, siendo eficiente la separación entre centroides y la probabilidad de error es muy pequeña.

Los valores en color amarillo indican que existe entre 25 y 50 unidades de separación (distancia euclidiana) entre los 5 gestos, siendo regular la separación entre centroides y con probabilidad media de error.

Los valores en color rojo indican que existe menos de 25 unidades de separación entre los 5 gestos, siendo deficiente la separación entre centroides y con alta probabilidad de error.

3.3.1 VARONES

Se realizó pruebas con varones de diferente perímetro de antebrazo, para evidenciar que tan separados se encuentran los centroides de cada uno de los gestos, de esta manera se tiene una visión amplia de cuan eficiente es el sistema de reconocimiento de gestos de la mano en varones. Las siguientes figuras, muestran gráficamente, la ubicación de los centroides para cada uno de los usuarios.

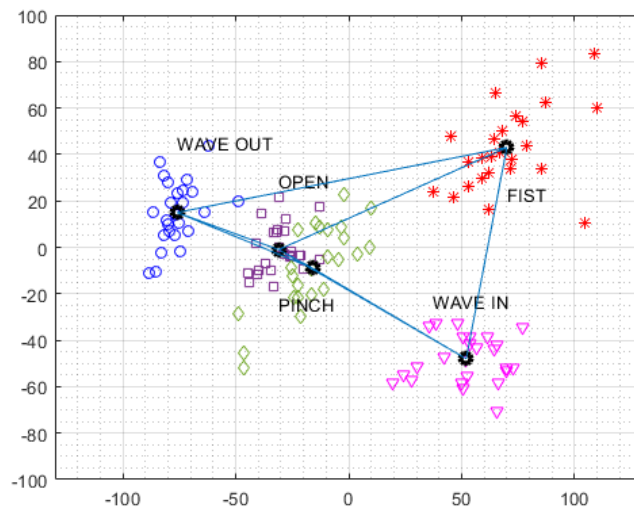


Figura 3.10 Clústers usuario 1

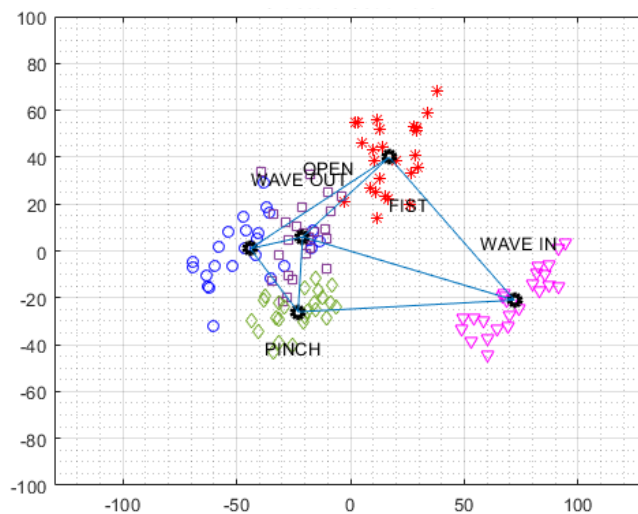


Figura 3.11 Clústers usuario 2

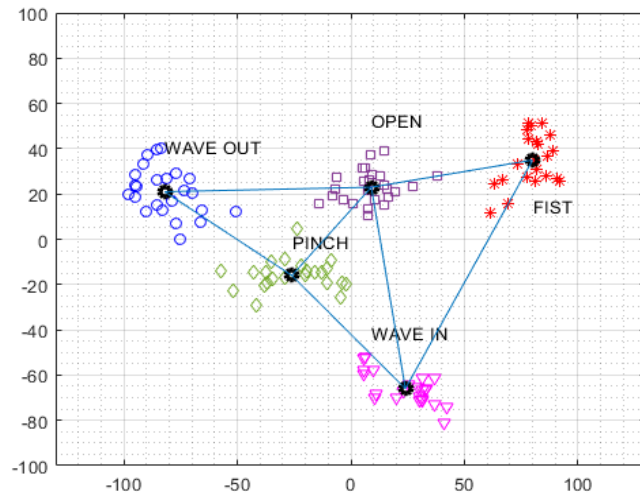


Figura 3.12 Clústers usuario 3

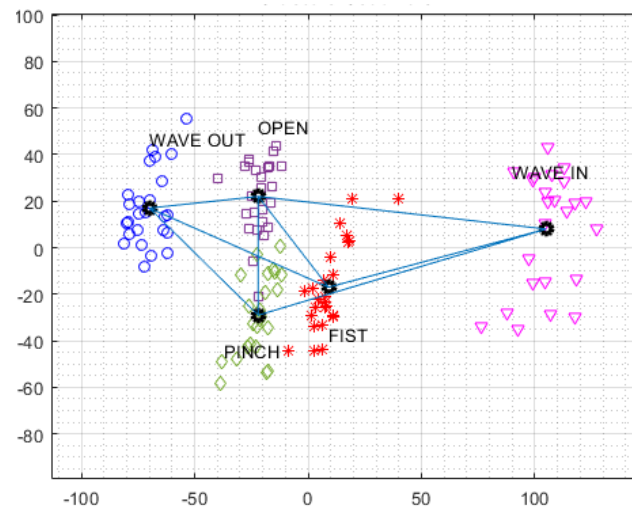


Figura 3.13 Clústers usuario 4

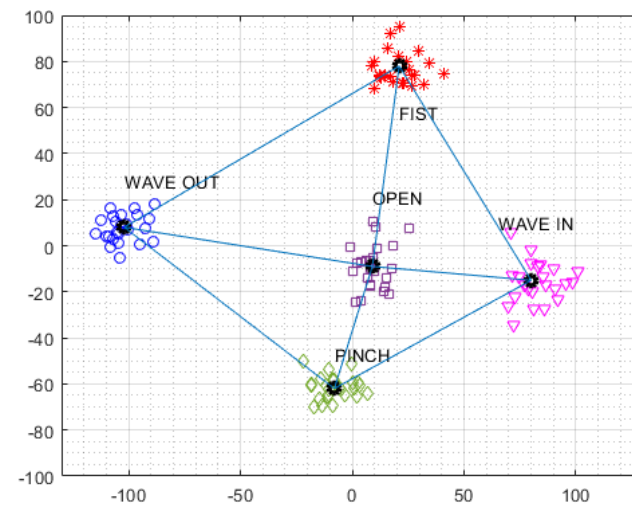


Figura 3.14 Clústers usuario 5

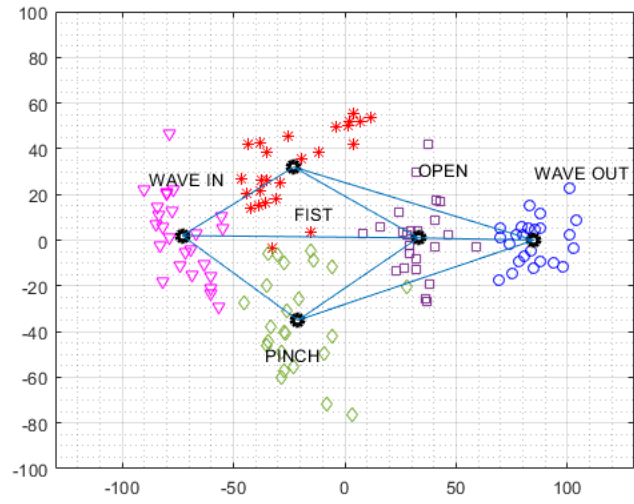


Figura 3.15 Clústers usuario 6

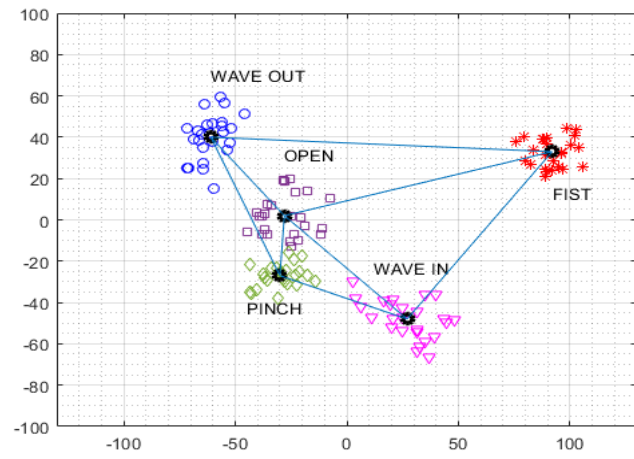


Figura 3.16 Clústers usuario 7

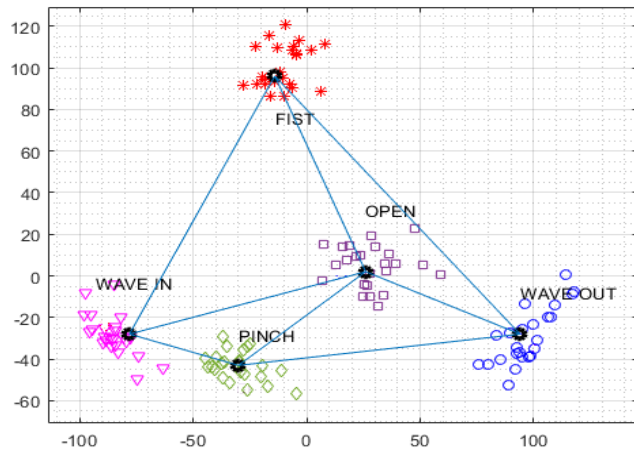


Figura 3.17 Clústers usuario 8

3.3.1.1 Distancia entre centroides

La tabla 3.10, muestra la distancia entre centroides de los sujetos de experimentación, junto al valor de perímetro de antebrazo de cada usuario.

Tabla 3.10. Distancia entre centroides de varones

	Perímetro Antebrazo [cm]	Distancia									
		Fi-Wi	Fi-Wo	Fi-Op	Fi-Pi	Wi-Wo	Wi-Op	Wi-Pi	Wo-Op	Wo-Pi	Op-Pi
User 1	21,5	93	150	112	104	143	97	84	54	73	40
User 2	24	84	80	53	78	119	99	97	44	47	34
User 3	26,5	116	158	80	118	131	91	78	94	74	69
User 4	27,5	109	96	65	64	177	131	135	62	75	56
User 5	28	113	131	78	109	178	103	97	117	99	73
User 6	28,5	78	118	77	72	159	111	73	64	114	68
User 7	29	104	154	126	136	124	82	65	64	77	55
User 8	29	151	168	121	145	184	128	74	103	132	102

En todas las pruebas realizadas se pudo observar que, los usuarios presentan distancias mayores a 50 unidades en su mayoría de gestos, lo que significa que hay baja probabilidad de error en el reconocimiento de gestos de la mano. También se puede evidenciar que los dos varones con menor perímetro de antebrazo presentan menor separación de centroides entre los gestos Wo-Op, Wo-Pi y Op-Pi (amarillo).

3.3.1.2 Ubicación de clústers

La tabla 3.11, muestra la ubicación de los clústers para cada usuario analizado

Tabla 3.11. Ubicación de los centroides de los usuarios varones

	Centroide		
	Componente principal 1 (eje x)	Componente principal 2 (eje y)	
User 1	70	43	Fist
User 2	17	40	
User 3	74	28	
User 4	9	-17	
User 5	-13	63	
User 6	-23	32	
User 7	92	33	
User 8	-14	96	
User 1	52	-48	Wave in
User 2	72	-21	
User 3	7	-52	
User 4	105	8	
User 5	-82	-25	
User 6	-73	2	
User 7	27	-48	
User 8	-78	-28	
User 1	-76	15	Wave out
User 2	-44	1	
User 3	-60	21	
User 4	-70	17	
User 5	95	-8	

User 6	85	0	
User 7	-61	40	
User 8	94	-28	
User 1	-31	-1	Open
User 2	-21	6	
User 3	0	16	
User 4	-22	22	
User 5	-6	12	
User 6	33	1	
User 7	-28	2	
User 8	26	2	
User 1	-16	-9	Pinch
User 2	-23	-26	
User 3	-26	-16	
User 4	-22	-33	
User 5	5	-41	
User 6	-26	-31	
User 7	-29	-27	
User 8	-31	-43	

En todas las pruebas realizadas se pudo observar que cada usuario presenta diferente ubicación de clústers, a pesar de que ha sido cuidadosamente colocado el brazalete, en la misma posición para todos los usuarios.

3.3.1.3 Dispersión de clústers

La tabla 3.12, muestra los resultados de desviación estándar para cada sujeto de experimentación, junto al nivel de práctica definido en la tabla 2.10 del capítulo anterior. Los valores en azul muestran los usuarios que consiguen la menor desviación estándar.

Tabla 3.12. Desviación estándar de usuarios varones

	Nivel de práctica	Desviación estándar		
		Componente principal 1 (eje x)	Componente principal 2 (eje y)	
User 1	BAJO	10,940	17,897	Fist
User 2	MEDIO	6,560	10,587	
User 3	MEDIO	10,267	12,801	
User 4	BAJO	9,616	11,853	
User 5	ALTO	5.598	5.236	
User 6	BAJO	10,131	10,858	
User 7	BAJO	7,886	9,592	
User 8	MEDIO	8,551	7,876	
User 1	BAJO	16,281	14,739	Wave in
User 2	MEDIO	9,808	5,870	
User 3	MEDIO	9,049	11,357	
User 4	BAJO	14,631	16,834	
User 5	ALTO	5,424	3,846	
User 6	BAJO	11,075	15,829	
User 7	BAJO	14,184	11,312	
User 8	MEDIO	7,531	7,493	
User 1	BAJO	7,411	9,172	Wave out
User 2	MEDIO	3,893	4,977	
User 3	MEDIO	5,403	13,800	
User 4	BAJO	13,457	8,255	

User 5	ALTO	5,785	8,983	Open
User 6	BAJO	12,776	12,174	
User 7	BAJO	14,733	8,939	
User 8	MEDIO	6,696	8,024	
User 1	BAJO	15,202	8,815	
User 2	MEDIO	8,243	5,905	
User 3	MEDIO	5,571	9,735	
User 4	BAJO	16,844	10,143	
User 5	ALTO	3,996	4,606	Pinch
User 6	BAJO	14,753	10,066	
User 7	BAJO	13,297	8,303	
User 8	MEDIO	6,194	5,897	
User 1	BAJO	11,264	10,557	
User 2	MEDIO	7,088	3,279	
User 3	MEDIO	5,311	9,479	
User 4	BAJO	9,615	6,009	
User 5	ALTO	5,205	3,392	Pinch
User 6	BAJO	11,456	12,468	
User 7	BAJO	12,761	9,637	
User 8	MEDIO	4,406	2,109	

En todas las pruebas realizadas se puede observar que, a mayor nivel de práctica, menor desviación estándar, es decir que los usuarios que han practicado previo a la adquisición de las muestras, consiguen realizar los gestos lo más parecido posible en cada repetición y de esta manera logran que las nuevas señales EMG se mantengan cerca de su centroide.

3.3.2 MUJERES

Se realizó pruebas con mujeres de diferente perímetro de antebrazo, para evidenciar que tan separados se encuentran los centroides de cada uno de los gestos, de esta manera se tiene una visión amplia de cuan eficiente es el sistema de reconocimiento de gestos de la mano en mujeres.

A continuación, se muestra gráficamente, los clústers de cada uno de los gestos.

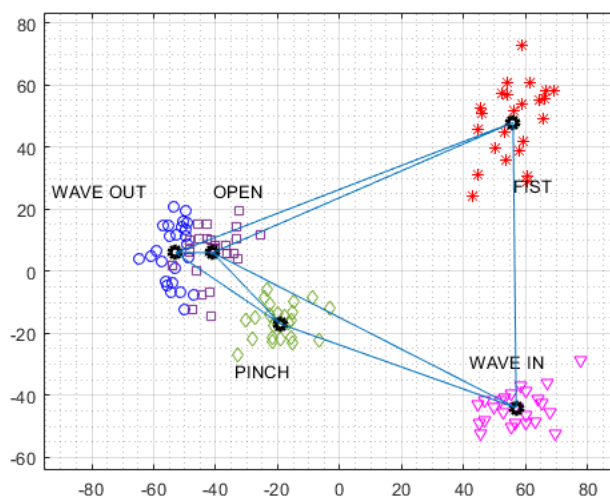


Figura 3.18. Nubes usuario 9

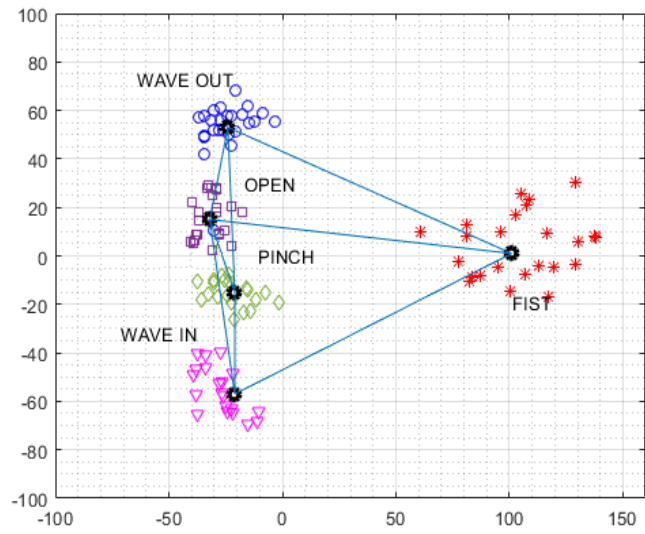


Figura 3.19. Nubes usuario 10

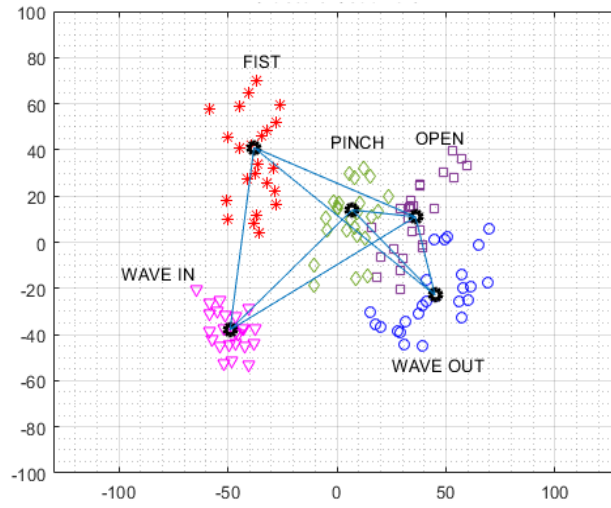


Figura 3.20. Nubes usuario 11

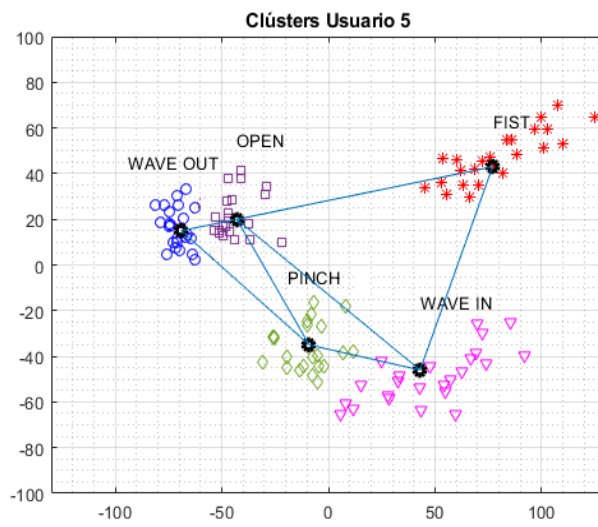


Figura 3.21. Nubes usuario 12

3.3.2.1 Distancia entre centroides

La tabla 3.13, muestra la distancia entre centroides conseguida para cada uno de los gestos para los usuarios mujeres analizadas y su perímetro de antebrazo.

Tabla 3.13. Distancia entre centroides

	Perímetro antebrazo [cm]	Distancia									
		Fi-wi	Fi-wo	Fi-op	Fi-pi	Wi-wo	Wi-op	Wi-pi	Wo-op	Wo-pi	Op-pi
User 9	25	139	135	136	131	117	79	69	46	76	63
User 10	25	101	151	129	114	134	116	57	42	79	69
User 11	26	93	118	107	100	121	111	84	53	50	39
User 12	29	85	105	86	64	98	99	73	46	56	35

Para los usuarios analizados se puede observar que, existe gran separación de centroides, lo que implica baja probabilidad de error, excepto para el gesto Wave out-Open y Open-Pinch, en donde pueden existir algunos errores de reconocimiento.

3.3.2.2 Ubicación de clústers

La tabla 3.14, muestra los valores de área bajo la curva de las 25 muestras de cada gesto.

Tabla 3.14. Ubicación de los centroides

	Centroide		
	Componente principal 1 (eje x)	Componente principal 2 (eje y)	
User 9	77	43	Fist
User 10	101	1	
User 11	-38	41	
User 12	77	43	
User 9	43	-46	Wave in
User 10	-21	-57	
User 11	-49	-38	
User 12	43	-46	
User 9	-69	15	Wave out
User 10	-24	53	
User 11	45	-23	
User 12	-69	15	
User 9	-43	6	Open
User 10	-32	15	
User 11	36	11	
User 12	-43	20	
User 9	-19	-17	Pinch
User 10	-23	-14	
User 11	6	11	
User 12	-9	-35	

En todas las pruebas realizadas se puede observar que cada usuario presenta valores diferentes de centroide, a pesar de que ha sido cuidadosamente colocado el brazalete, en la misma posición.

3.3.2.3 Dispersión de clústers

La tabla 3.15, muestra los valores de desviación estándar para cada sujeto de experimentación, junto al nivel de práctica que presenta cada uno para las 25 muestras de cada gesto adquiridas.

Tabla 3.15. Desviación estándar de usuarios mujeres

	Nivel de práctica	Desviación estándar		
		Componente principal 1 (eje x)	Componente principal 2 (eje y)	
User 9	Medio	9,953	9,510	Fist
User 10	Medio	14,760	15,661	
User 11	Bajo	15,312	13,034	
User 12	Bajo	11,129	13,512	
User 9	Medio	4,866	3,776	Wave in
User 10	Medio	11,374	8,535	
User 11	Bajo	16,060	15,480	
User 12	Bajo	7,980	4,103	
User 9	Medio	4,010	9,150	Wave out
User 10	Medio	1,515	4,367	
User 11	Bajo	4,785	9,274	
User 12	Bajo	3,765	7,323	
User 9	Medio	5,423	7,341	Open
User 10	Medio	3,205	6,985	
User 11	Bajo	6,847	7,192	
User 12	Bajo	6,178	8,005	
User 9	Medio	3,473	4,265	Pinch
User 10	Medio	3,823	8,601	
User 11	Bajo	12,557	15,214	
User 12	Bajo	6,985	4,928	

En todas las pruebas realizadas se puede observar que, a mayor nivel de práctica, menor desviación estándar, es decir que los usuarios que han practicado los gestos algunas veces consiguen realizar los gestos lo más parecido posible en cada repetición y de esta manera logran que las nuevas señales EMG se mantengan cercano a su centroide.

3.4 CLASIFICACIÓN

3.4.1 MÉTODO 1: DISTANCIA EUCLIDIANA AL CENTROIDE MÁS CERCANO

Para la evaluación del método del centroide con distancia euclidiana, se obtuvieron 50 muestras de cada gesto (Fist, Wave in, Wave out, Open y Pinch), en total 250 muestras, de 12 usuarios (8 varones y 4 mujeres). Los gestos fueron almacenados en una base de datos y se procedió a ejecutar el algoritmo para tiempo diferido. Se utilizó 3 muestras de cada gesto como referencia para hallar el centroide como configuración inicial rápida y de esta manera determinar a qué clase pertenecen las siguientes 47 muestras de cada gesto y para cada usuario.

La tabla 3.16, muestra el porcentaje de clasificación correcto por sujeto, utilizando el método de centroides con distancias euclidianas.

Tabla 3.16. Resultados de clasificación usando método 1

	Usuario	Total de muestras tomadas	Número de muestras por gesto	Aciertos	Errores	Precisión [%]	Error [%]
Varones	1	250	50	248	2	99,2	0,8
	2	250	50	247	3	98,8	1,2
	3	250	50	249	1	99,6	0,4
	4	250	50	246	4	98,4	1,6
	5	250	50	244	6	97,6	2,4
	6	250	50	246	4	98,4	1,6
	7	250	50	245	5	98	2
	8	250	50	242	8	96,8	3,2
Mujeres	9	250	50	241	9	96,4	3,6
	10	250	50	244	6	97,6	2,4
	11	250	50	239	11	95,6	4,4
	12	250	50	238	12	95,2	4,8
Promedio						97,63	2,37

3.4.2 METODO 2: RED NEURONAL BASADO EN PERCENTRÓN MULTICAPA (MLP)

Se utilizó la misma base de datos del método anterior, para evaluar el método basado en perceptrón multicapa. Se utilizó 3 muestras de cada gesto, como referencia para entrenar la red neuronal y 47 muestras de cada gesto para evaluar la red neuronal.

La tabla 3.17, muestra el porcentaje de clasificación correcto por sujeto, utilizando redes neuronales artificiales.

Tabla 3.17. Resultados de clasificación usando método 2

	Usuario	Total de muestras tomadas	Número de muestras por gesto	Aciertos	Errores	Precisión [%]	Error [%]
Varones	1	250	50	249	1	99,6	0,4
	2	250	50	246	4	98,4	1,6
	3	250	50	246	4	98,4	1,6
	4	250	50	248	2	99,2	0,8
	5	250	50	247	3	98,8	1,2
	6	250	50	245	5	98	2
	7	250	50	247	3	98,8	1,2
	8	250	50	241	9	96,4	3,6
Mujeres	9	250	50	245	5	98	2
	10	250	50	247	3	98,8	1,2
	11	250	50	240	10	96	4
	12	250	50	237	13	94,8	5,2
Promedio						97,93	2,07

3.5 SISTEMA EN TIEMPO REAL

De los análisis realizado tanto en la variabilidad intrapersonal e interpersonal (diferentes distancias entre centroides, diferente ubicación entre centroides y diferente desviación estándar de datos), con la variable utilizada en la etapa de extracción de características (área bajo la envolvente en el dominio del tiempo de las señales EMG), se puede confirmar el problema que involucra utilizar el sistema con un mismo usuario y con diferentes usuarios, lo que significa que sería imposible tener un perfil único que pueda representar a todos los usuarios siendo esta alternativa totalmente inviable.

Para solucionar este inconveniente se ha planteado realizar los siguientes pasos para aumentar la precisión en el reconocimiento de gestos de la mano:

1. Colocar el brazalete a una distancia codo – myo de aproximadamente 6 cm (el giro no importa)
2. Realizar una configuración rápida al inicio de cada sesión (tomar unas pocas muestras)
3. Extraer un nivel de ruido adaptativo para la etapa de detección de inicio de gesto.

La tabla 3.18, muestra la carga computacional que se requiere para mostrar el gesto realizado por el usuario, desde que se adquieren las señales EMG hasta la determinación del gesto, para cada uno de los métodos de clasificación.

Tabla 3.18. Carga computacional en tiempo real

Métodos de clasificación	Número de muestras	Tiempos de respuesta
Método 1	250	57,12 [ms]
Método 2	250	67,79 [ms]

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

1. En este trabajo se ha expuesto una metodología para encontrar las mejores variables que hacen que los clústers n-dimensionales se alejen lo máximo posible. Aplicando la misma idea, ahora ya es posible seleccionar algoritmos de preprocesamiento de una manera justificada y observando sus efectos en el grado de alejamiento de dichas nubes n-dimensionales.
2. Los 3 dominios analizados (tiempo, frecuencia y complejidad), presentan gran separación de centroides entre los cinco gestos. Lo cual los hace idóneas para lograr diferenciar los 5 gestos.
3. El dominio del tiempo y complejidad presentan una carga computacional de 27.4 ms y 39.4 ms respectivamente, siendo estos valores adecuados para implementar en sistemas de reconocimiento de gestos de la mano en tiempo real y óptimo para dispositivos de pocos recursos computacionales.
4. Un aporte importante es la comprensión del fenómeno basada en el entendimiento de la influencia de algunos parámetros tanto en la variabilidad intrapersonal como interpersonal. Esto puede constituirse en un nuevo paradigma en lugar de tener que recopilar una gran cantidad de casos de experimentación tal como lo hacen los métodos basados en inteligencia artificial.
5. La variabilidad intrapersonal, refleja un cambio de ubicación de los centroides de un mismo sujeto de experimentación a lo largo del tiempo, aun cuando se mantienen controladas las variables: nivel de batería, distancia de colocación codo – myo, giro del brazalete y perímetro del antebrazo. Lo que significa que otras variables no controlables, provocan un cambio en el área bajo la curva de las señales EMG.
6. En la variabilidad intrapersonal, cuando se coloca el brazalete a diferente distancia desde el codo al dispositivo, se produce un cambio de ubicación de los clústers de los gestos de un mismo sujeto de experimentación a lo largo del tiempo. Esto implica que para la variable de área bajo la curva de las 8 señales EMG, no es posible mantener un perfil único si se pretende colocarse el brazalete en cualquier posición.
7. El nivel de práctica influye inversamente en la desviación estándar, es decir a mayor nivel de práctica, las repeticiones realizadas tendrán valores muy parecidos de área bajo la curva. Lo cual disminuye la probabilidad de error en el algoritmo.

8. Hombres obtienen aproximadamente 15% más de separación de nubes que las mujeres para todos los gestos realizados, aun cuando, el perímetro del antebrazo, la distancia codo-myoe y el giro del brazalete son los mismos. Lo que haría pensar que un mayor grado de desarrollo muscular característico de los sujetos de experimentación varones, genera mayor activación en los sensores del Myoe Armband.
9. En la variabilidad interpersonal, la dimensión del antebrazo, la contextura física y el sexo de las personas influye directamente en la morfología y amplitud de las señales EMG y por consiguiente en la ubicación de los clústers de los gestos estudiados. Lo cual impide mantener un perfil general para diferentes personas.
10. El algoritmo presenta una precisión de clasificación del 97.6% para el método 1 y una precisión de clasificación del 97.9% para el método 2. Esta precisión de clasificación de gestos es un indicador indirecto para evaluar los gestos realizados con la mano.
11. Existen varias investigaciones para el reconocimiento de gestos de la mano con altos porcentajes de clasificación en tiempo discreto, pero debido a la variabilidad intrapersonal e interpersonal, estos sistemas disminuyen aquel porcentaje de clasificación.
12. Si bien aquí se ha demostrado la metodología expuesta para clasificar 5 gestos basados en sus señales de EMG, esta metodología puede ser aplicada en cualquier campo de investigación.
13. Un aporte de este proyecto es la construcción de una base de datos con las señales EMG de los 5 gestos expuestos de 12 sujetos de experimentación, que puede servir para posteriores investigaciones.

4.2 RECOMENDACIONES

- Si se trata de una persona con contextura de antebrazo delgado, se recomienda realizar un ajuste mecánico del dispositivo para obtener mayor detección de las señales EMG y conseguir mayor separación de gestos.
- Para un trabajo futuro, se recomienda probar el algoritmo con un grupo de personas más grande, para fortalecer el análisis de variabilidad intrapersonal e interpersonal.
- Para un trabajo futuro, se recomienda utilizar el dominio de la complejidad como método para la extracción de características para el análisis de variabilidad intrapersonal e interpersonal.
- Para un trabajo futuro, se recomienda realizar un análisis de variabilidad intrapersonal e interpersonal de las señales EMG para la implementación de un sistema que no requiera entrenamiento previo (perfil personalizado), ya que estos factores son determinantes al momento de crear una base de datos y utilizarlos como referencia con distintos usuarios.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] “Un sistema analiza los gestos de la mano; la herramienta puede servir a personas que están en rehabilitación | El Comercio.” <https://www.elcomercio.com/tendencias/sistema-gestos-mano-inteligencia-artificial.html> (accedido Nov. 03, 2020).
- [2] C. Motoche, “Reconocimiento de gestos de la mano en tiempo real usando señales electromiográficas y redes neuronales artificiales”, Repositorio Escuela Politécnica Nacional, Quito, 2018. [Online]. Disponible: <https://bibdigital.epn.edu.ec/handle/15000/19598>
- [3] L. Chen, F. Wang, H. Deng y K. Ji, “A Survey on Hand Gesture Recognition”, 2013 International Conference on Computer Sciences and Applications, 2013.
- [4] R. Z. Khan y N. A. Ibraheem, “Surver on Various Gesture Recognition Technologies”, International Journal of Computer Applications, vol. 50, nº 7, pp. 38 – 44, 2012.
- [5] R. Chowdhury, M. Reaz, A. Mohd, A. Bakar, C. Kalaivani y T. Chang, “Surface Electromyography Signal Processing and Classification Techniques”, Sensors, vol. 13, nº 12, pp. 12431 – 12466, 2013.
- [6] S. Kavya, M. Dhatri, R. Sushma y B. N. Krupa, Controlling the hand and forearm movements of an, Bangalore, 2015.
- [7] M. Satoshi, K. Toshiyuki y I. koji, Estimation of Forearm Movement from EMG Signal, Seoul, 2001.
- [8] J. B. Bence y S. Péter, Estimating the instantaneous wrist flexion angle, Budapest, 2013.
- [9] K. Frederic, P. Michael y K. Antonio, User-Independent Real-Time Hand Gesture Recognition, Saarbrücken, 2017.
- [10] A. M. Ibrahim, M. Ezzat and A. Y. Abdelaziz, "Performance comparison of classification methods for line outage detection", Cairo, 2016, pp. 26-32, doi: 10.1109/MEPCON.2016.7836867.
- [11] A. Bourouhou, A. Jilbab, C. Nacir and A. Hammouch, "Comparison of classification methods to detect the Parkinson disease", 2016, pp. 421-424, doi: 10.1109/EITech.2016.7519634.
- [12] A. Ysunza y E. Perusquía, “Electrodiagnóstico”, *medigraphic*, nº 74, pp. 73-80, 2007.
- [13] L. Pérez y A. Rosas, “eprints”, 2018. [En línea]. Available: <https://eprints.ucm.es/id/eprint/49214/1/T40214.pdf>. [Último acceso: 6 Abril 2021].

- [14] T. L. Inc, "Myo Thakmic Labs Inc", Thalmic Labs Inc, 2013. [En línea]. Available: <https://developerblog.myo.com/>. Último acceso: 27 Enero 2018].
- [15] G. Romero, "Diseño e implementación de un sistema de control basado en señales de electromiograma para el control de una mano robótica", Repositorio Escuela Politécnica Nacional, Quito, 2017. [Online]. Disponible: <https://bibdigital.epn.edu.ec/handle/15000/18831>.
- [16] Frigo, M., and S. G. Johnson. "FFTW: An Adaptive Software Architecture for the FFT." *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. Vol. 3, 1998, pp. 1381-1384.
- [17] I. Naula y L. Alcides, "Análisis de datos funcionales aplicado a la distribución a la población ecuatoriana", Quito, EPN, 2016, pp. 28-33.
- [18] S. Fernández Fraga y J. Rangel Mondragón, Comparison of Higuchi, Katz and Multiresolution Box Counting Fractal Dimension Algorithms in EEG Signals Based on Event-Related Potentials, *magazine*, vol. 14, n.º 27, pp. 73–83, sep. 2017.
- [19] Ilin, A., and T. Raiko. "Practical Approaches to Principal Component Analysis in the Presence of Missing Values." *J. Mach. Learn. Res.*. Vol. 11, August 2010, pp. 1957–2000.
- [20] "¿Qué es una red neuronal? - MATLAB & Simulink", *La.mathworks.com*, 2021. [Online]. Available: <https://la.mathworks.com/discovery/neural-network.html>. [Accessed: 29- Jan- 2021].
- [21] "MATLAB Support for MinGW-w64 C/C++ Compiler", *La.mathworks.com*, 2021. [Online]. Available: <https://la.mathworks.com/matlabcentral/fileexchange/52848-matlab-support-for-mingw-w64-c-c-compiler>. [Accessed: 27- Jan- 2021].

ANEXOS

ANEXO A. Paper “Una nueva metodología para el reconocimiento de patrones aplicada al reconocimiento de gestos con las manos mediante EMG. Análisis de variabilidad intrapersonal e interpersonal”.

El paper está disponible en el anexo digital.

ORDEN DE EMPASTADO