ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA MECÁNICA

TÉCNICAS BIOMECÁNICAS EN ERGONOMÍA

IMPLEMENTACIÓN DE UN DISPOSITIVO ACTIVO SHIELD-EKG-EMG PARA LA ADQUISICIÓN DE BIOSEÑALES

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN MECÁNICA.

KLINSMANN DIEGO VACA MALDONADO

klinsmann.vaca@epn.edu.ec

DIRECTOR: WILLIAM RICARDO VENEGAS TORO

William.venegas@epn.edu.ec

Quito, 29 de septiembre del 2023

CERTIFICACIONES

Yo, KLINSMANN DIEGO VACA MALDONADO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

_____

KLINSMANN DIEGO VACA MALDONADO

Certifico que el presente trabajo de integración curricular fue desarrollado por KLINSMANN DIEGO VACA MALDONADO, bajo mi supervisión.

_____

WILLIAM RICARDO VENEGAS TORO
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

KLINSMANN DIEGO VACA MALDONADO

WILLIAM RICARDO VENEGAS TORO

DEDICATORIA

presente trabajo está dedicado a Dios, ya que gracias a sus bendiciones e logrado materializar un sueño, a mis padres ya que siempre han estado a mi lado y en base a su apoyo y consejos han hecho de mí una mejor persona , a mis profesores y compañeros quienes han sido las personas con las que he compartido todo este bonito tiempo de aprendizaje y que de una u otra manera han contribuido para el logro de este objetivo.

AGRADECIMIENTO

Primeramente, agradezco a la Ing. Yessenia Alejandra Barrera Jaramillo por todo su apoyo y motivación durante la carrera universitaria y la realización de este trabajo, gracias por su apoyo y compresión.

Agradezco a todas las personas que de una u otra manera se vieron involucrados en cada paso que doy, gracias por su preocupación y apoyo.

Agradezco a la carrera de Ingenieria Mecánica de la Escuela Politécnica Nacional por abrirme las puertas para desarrollar todos los conocimientos que hoy por hoy aplico y por permitirme cumplir un sueño.

ÍNDICE DE CONTENIDO

RESUMEN

El presente componente contiene la implementación de un Shield EKG-EMG para la adquisición de señales musculares, por medio de una interfaz desarrollada en APP Designer de Matlab, en interconexión con una placa Arduino, programada previamente, adicionalmente contiene un protocolo de pruebas, que consiste en la generación de cinco ciclos de movimientos musculares, para estimular el Biceps ubicado en el brazo, este protocolo de pruebas será usado para comparar la señal adquirida por medio del Shield, con una segunda señal obtenida por el equipo Delsys disponible en el Laboratorio de Bioingeniería de la Facultad de Ingeniería Mecánica de la Escuela Politécnica Nacional.
Las señales serán comparadas mediante un procesamiento de señal en función del tiempo, cruces por cero, previo a esta sección se muestra la digitalización de dichas señales, así como sus fases de suavizado, vectorizado e interpolado.
Finalmente se muestra la fase de normalización de las dos señales adquiridas, y verificadas por métodos estadísticos, a fin de obtener un error entre el 3% al 8% (rango ideal de confianza).
PALABRAS CLAVE: EMG, APP, Matlab, Musculares, Procesamiento, Bioingeniería.

ABSTRACT

This component comprises the implementation of an EKG-EMG Shield for the acquisition of muscle signals, through an interface developed in Matlab's APP Designer, interconnected with an Arduino board, pre-programmed. Additionally, it contains a testing protocol, consisting of the generation of five cycles of muscle movements to stimulate the Biceps located in the arm. This testing protocol will be used to compare the signal acquired through the Shield with a second signal obtained by the Delsys equipment available in the Bioengineering Laboratory of the Faculty of Mechanical Engineering at the National Polytechnic School.

The signals will be compared through signal processing based on time, zero crossings. Before this section, the digitization of these signals is shown, as well as their phases of smoothing, vectorization, and interpolation.

Finally, the normalization phase of the two acquired signals is shown, verified by statistical methods, in order to obtain an error between 3% and 8% (ideal confidence range).

KEYWORDS: EMG, APP, Matlab, Muscular, Processing, Bioengineering.

1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El estudio busca establecer nuevas técnicas de adquisición y procesamiento de bioseñales, basado en la investigación previa detallada en su estado del arte, en la cual se analizarán el procesamiento de señales musculares obtenidas de manera superficial, y los principales dispositivos EMG tanto activos como pasivos que permitan la adquisición de las mismas, se busca así también la implementación de dispositivos electrónicos (Shield-EKG-EMG) que faciliten el procesamiento de las señales adquiridas y permitan la calibración, validación y análisis de las mismas por medio de protocolos experimentales de pruebas, para ser evaluadas como patrones funcionales en una referencia de sujetos sanos, que servirán para caracterizar las tendencias de la capacidad de fuerza y ser estándar en contraste a variables funcionales en sujetos con perdida funcional en la activación muscular, ya que este estudio podrá ser validado por el equipo DelSys de registro pasivo de electrodos superficiales.

1.1 Objetivo general

Implementar un dispositivo Shield-EKG-EMG para la adquisición de bioseñales.

1.2 Objetivos específicos

1. Levantar un estudio del estado del arte referente a dispositivos EMG pasivos y activos, procesamiento de bioseñales superficiales musculares.
2. Implementar un dispositivo Shield-EKG-EMG.
3. Desarrollar un protocolo experimental de pruebas piloto de los movimientos de calibración para la validación de bioseñales Shield –EMG en contraste del sistema EMG delsys.
4. Analizar los bioseñales bajo un procesamiento de datos.
5. Normalizar y Validar la bioseñal.

1.3 Alcance

Se utilizará el Laboratorio de Bioingeniería de la Facultad de Ingeniería Mecánica, donde se establecerá un protocolo de pruebas piloto que asocie la movilidad cíclica con la actividad muscular, buscando obtener registros en bruto de EMG y fuerza obtenidos en los sistemas delsys de electriomiografía (EMG). Además, se implementará un dispositivo Shield-EKG-EMG de registro activo de electrodos superficial, el mismo que será validado con el equipo delsys de registro pasivo de electrodos superficial, mediante un pre-procesamiento de datos (rectificado y filtrado).

1.4 Marco teórico

¿Qué es la electromiografía?

Es el estudio dedicado a adquirir, registrar y analizar el funcionamiento del sistema nervioso periférico (conjunto músculo y nervio) por medio de la detección de actividad eléctrica muscular. Las fibras musculares al contraerse generan descargas que son captadas por medio de los electrodos (superficiales, de aguja, implantados) y permiten generar patrones de comportamiento del sistema neuromuscular, permitiendo así analizar patrones de activación muscular normales o anormales según sea requerido por quien las obtiene.

Esta técnica es empleada en su mayoría para el diagnóstico de enfermedades relacionadas con la pérdida de fuerza, masa o debilidad muscular, de un área en específico o en miembros inferiores y superiores. (Electromiografía y electroneurografía, 2020)

El estudio de la electromiografía a promovido el desarrollo de análisis con el objetivo de obtener un mayor conocimiento de las propiedades de la energía eléctrica, así como el desarrollo de nuevas tecnologías de la mano de la eléctrica y electrónica. A mitad del siglo XX, se comienza a introducir el primer equipo comercial electromiográfico para uso en el campo de la medicina, el mismo que estaba basado en lógica meramente analógica. La aparición de la tecnología digital permitió disponer de sistemas más efectivos y mucho más dedicados gracias a la aparición de los microcontroladores, los cuales cada vez eran más fiables, y mucho más potentes, lo que permitía no solo captar estas señales, sino que permitió un procesamiento y digitalización más avanzado (representación, almacenaje, análisis y clasificación). (R.N. Khushaba, 2012)

UMS o Unidades Motoras

Estas unidades, también conocidas como motoneuronas, se encargan de la transmisión de impulsos nerviosos hacia los músculos, a su vez estos músculos son controlados por un centro nervioso superior que regula una respuesta motriz, (movimiento muscular). Los Axones de las UMs inician su recorrido desde la médula espinal llegando hasta las fibras musculares a lo largo de todo el cuerpo, cada uno de estos axónes sufre de varias divisiones y ramificaciones, terminales nerviosos, con los cuales se conectan uno a uno a las fibras a través de la llamada “Placa motora”. En la Figura 1 se muestra el conjunto formado por una UMs y las fibras musculares.

Podemos definir a la electromigrafía como el registro de actividades eléctricas realizadas por un músculo estirado. Las contracciones de las fibras musculares de manera individual generan el conocido potencial de acción, sabiendo que el potencial de una unidad motoro o PAUM es la suma de estos potenciales o contracciones que

componen una UMs (Unidad motora). Las señales EMG están compuestas por estos potenciales de acción resultado de las UM superpuestas. Para medir estas señales se puede realizar la descomposición de los potenciales de acción de las UMs que la constituyen.

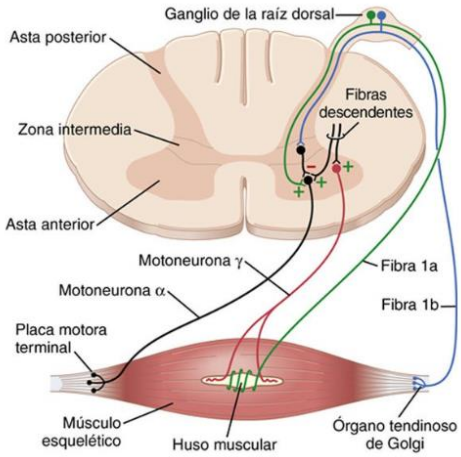


Figura 1 Unidad motora (UM) **(Martinez, 2021)**

¿Qué son las señales Electromiográficas o (EMG)?

Son las señales eléctricas obtenidas mediante el proceso de electromiografía; estas señales reflejan las fuerzas que los músculos generan, así como el tiempo que duran sus durante sus comandos motores.

Las señales EMG poseen amplitudes que varía desde los microvoltios (μV) hasta los milivoltios (mV) en un rango bajo, menor a los 10mV (DALCAME- Grupo de investigacion biomédica, 2005).

Las amplitudes, así como las propiedades de las señales EMG analizadas en el dominio del tiempo o de frecuencia, dependen de algunos factores:

- Tiempo de la contracción muscular
- Distancia en la zona muscular, entre electrodo y músculo.
- Espesor y presencia de tejido adiposo (propiedades de la piel)
- Amplificador (diseño)
- Piel + electrodo (calidad de contacto/ adherencia)
- Conversión
- Almacenamiento de datos (analógica – digital)

La calidad de la señal EMG puede describirse como la relación entre la señal EMG y el ruido generado por el ambiente, donde el objetivo es amplificar la señal mientras se minimiza el ruido, esta relación está determinada en una gran medida por los electrodos, siendo más específicos por las propiedades y calidad del electrodo durante el contacto con la piel.

Como hemos mencionado las contracciones en las fibras musculares, generan actividades eléctricas la misma que puede ser percibida y medida por los electrodos que se encuentran adheridos-fijados a la superficie de la piel próxima al grupo muscular en análisis.

La señal EMG obtenida de manera superficial usando electrodos puede ser modelada como si se tratase de un proceso estocástico con variación en el tiempo y con media cero. La desviación estándar de estas señales (sin procesar, datos en bruto) está monotónicamente relacionada con el número de UMs activas y con la velocidad de su propia activación; Esta desviación estándar usualmente es empleada en la aproximación de la magnitud de las actividades eléctricas musculares, conocida también como amplitud EMG. La amplitud EMG tiene una variedad de aplicaciones, tales como la señal de control para prótesis mioeléctricas, estimaciones ergonómicas, sistemas de realimentación (Biofeedback), y también ha sido usada para estimar el par asociado a una articulación. (DALCAME- Grupo de investigacion biomédica, 2005)

El análisis de las actividades EMG comprende tres estados funcionales durante el registro de la señal los cuales son:

- Reposo, Músculo completamente relajado (No registra actividad).
- Contracción débil, Activación escasa de UMs y descargas de PAUMs.
- Contracción voluntaria máxima, UMs funcionando.

Procesamiento de señales electromiográficas

Una vez que las señales eléctricas musculares han sido adquiridas es necesario realizar un análisis y procesamiento de la información, así como los datos obtenidos.

En la Figura 2 se muestran las principales etapas que se pueden encontrar durante la adquisición de la señal, así como el procesamiento y la obtención de las características de la señal EMG obtenida a partir de una fibra muscular.

Suavizado de la señal, remoción de artefactos y de Bias.

Conocemos que el rango de las señales EMG se encuentran entre los microvoltios, la señal puede verse afectada por ruidos externos o por algunos generados por el proceso de adquisición mismo (dispositivos electrónicos usados); Muchos de estos inconvenientes pueden ser solucionados con algunos tips previos a la adquisición, por ejemplo, realizar una buena preparación de la piel, es decir que la misma se encuentre limpia y sin presencia de lociones o cremas, la ubicación y adherencia de los electrodos, entre otros; Por otro lado, existen otras señales que pueden mezclarse con las señales EMG durante su adquisición y requieren ser removidas y/o filtradas a fin de evitar confusiones y datos no deseados; un ejemplo de estas señales son las ECG o

electrocardiográficas.

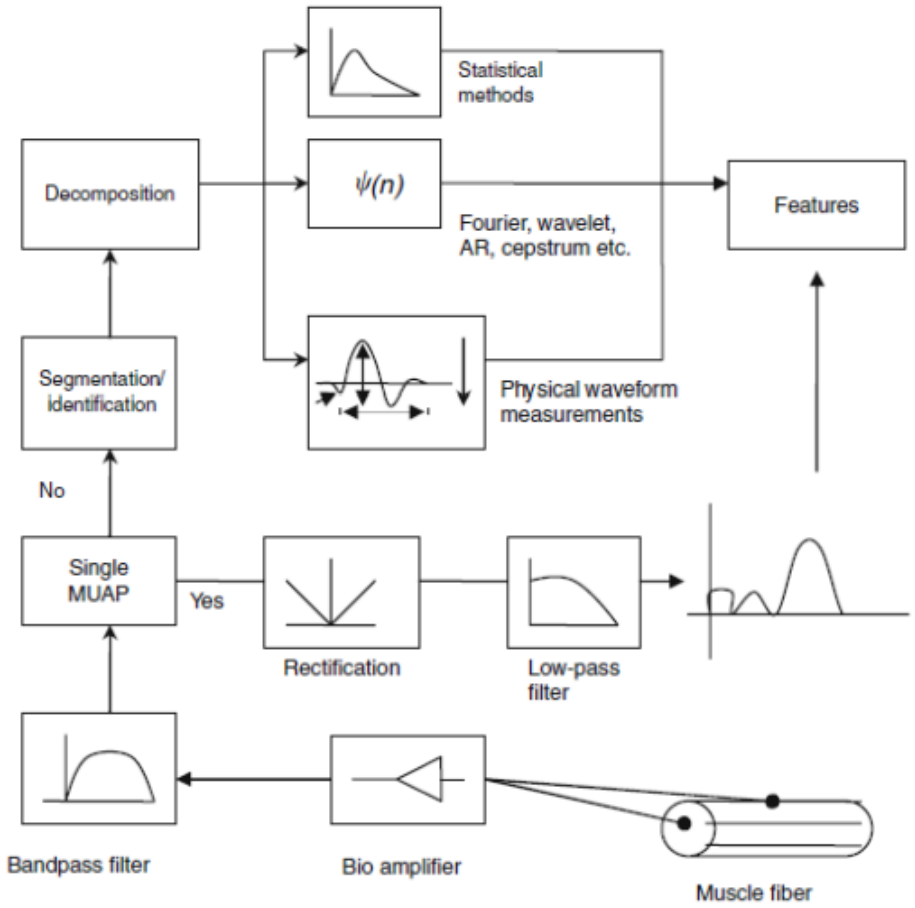


Figura 2 Diagrama de bloques, adquisición de señales EMG

Filtrado digital

Independientemente de haberse realizado una fase de filtrado de las señales por medio de componentes electrónicos y haberse realizado filtros que limitan el ancho de banda de la señal, muchos de los dispositivos requieren ser eliminados por medio de un filtrado digital. Al combinar algunos filtros adaptativos se puede llegar a reconocer patrones los cuales pueden eliminar o depurar las señales ECG mencionadas evitando que las adquisiciones de la señal EMG se vean alteradas, adicionalmente se puede también eliminar el ruido que pudiera producirse por el paso en los filtros convencionales que pueden no ser tan efectivos en niveles tan bajas de amplitud como lo son las señales EMG.

Normalización

La normalización en las señales EMG consiste en expresar de manera porcentual los valores obtenidos durante la captación de señales eléctricas de las actividades del músculo durante una fase de prueba calibrada (contracción) para de esta manera eliminar la variabilidad entre los valores medidos.

Una de las metodologías más utilizadas para este proceso de normalización, es por

medio de la toma de porcentajes, se tomará el porcentaje de la máxima conducción neuronal obtenida, mientras se realiza una contracción voluntaria máxima isométrica; esta metodología tiene como objetivo calibrar un valor en microvoltios obtenido de la señal EMG y llevarlo a una única unidad de calibración que pudiera ser usado desde el estudio fisiológico, como un porcentaje de la capacidad máxima de inervación. (Procesamiento de señales EMG en trastornos Neuromusculares, 2013) (véase Figura 3)

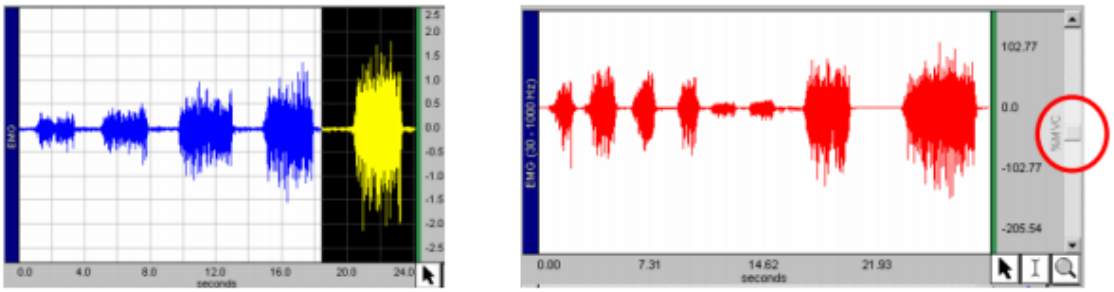


Figura 3 Normalización de la señal EMG

Rectificación y promediado

El proceso de rectificado consiste en la conversión de las amplitudes negativas a amplitudes positivas. Esta acción nos permite obtener parámetros relevantes como: media, valor pico, área bajo la curva, etc. (Konrad, 2005). Estos parámetros son elementales para el análisis matemático de la señal, ya que describe las características de la curva obtenida de las actividades de las UMs durante un tiempo o ciclo determinado. (véase Figura 4)

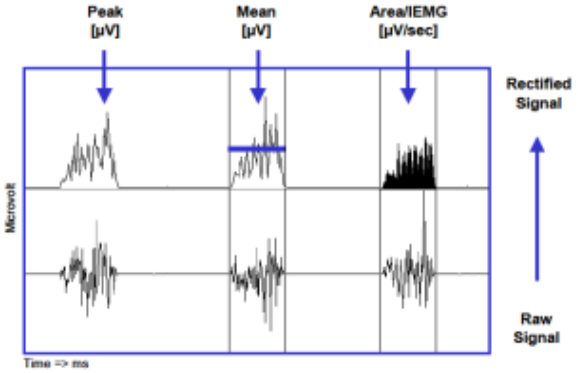


Figura 4 Rectificación de la señal EMG

Análisis temporal

Una vez que se ha procesado la información original y se ha logrado eliminar y depurar toda información inútil (ruido o señales ECG) se procede a realizar un análisis temporal (en función del tiempo) de los datos. Estos datos son evaluados y extraídos de cada uno de los segmentos o ciclos, esto con el objetivo de crear un patrón característico del conjunto que representen la señal EMG obtenida. Entre las más importantes tenemos:

Conteo de picos o valores superiores al valor RMS (TRMS)

Definido como el número de muestras, con amplitud mayor que la RMS de la señal obtenida, se usa la normalización en función al número total de muestras que contenga la señal, a fin de que el valor obtenido sea independiente de la duración de la medida. (véase Ecuación 1)

$$T_{RMS} = \frac{1}{T}\sum_{i=1}^{n} t_i \; con \begin{matrix} t = 1 & x_i > RMS \\ t = 0 & x_i < RMS \end{matrix}$$

$$(n = \#de\ muestras;\ T = Duración\ de\ la\ medida;\ t = tiempo)$$

Ecuación 1 Conteo de picos

Cruces por cero

La tasa de cruces por cero o (Zero-crossing Rate ZCR) consiste en obtener el número de veces que la señal llega a cruzar por el eje x o de las abscisas. Este cruce refleja las fluctuaciones aleatorias temporales, así como puede también ayudarnos en la detección de patrones por patologías. Podríamos decir que esta técnica nos permite medir la frecuencia de la señal de una manera más sencilla. Se define como:

$$ZCR = \frac{1}{2N}\left\{\sum_{i=1}^{N-1} |sgn[x(k)] - sgn[x(k-1)]\right\} \qquad sgn[x] = \begin{cases} 1x \geq 0 \\ -1x \leq 0 \end{cases}$$

Ecuación 2 Cruces por cero

Correlación y autocorrelación

Esta técnica es principalmente empleada en el reconocimiento de patrones dentro de la señal, identificación de pautas o la determinación de relaciones entre muestra sucesivas. Su primera aplicación es en la determinación del desfase existente entre dos señales, otra de sus aplicaciones es para la determinación de la periodicidad "escondida" de la señal, ya que esto refleja el grado de la similitud en distintas partes de una serie de datos obtenidos en función del tiempo. Para una secuencia de datos f(n) de longitud N, la función se la calcula como lo muestra la Ecuación 3:

$$r_f = \frac{1}{N}\sum_{n^{-0}}^{N\ 1|\tau|} \int(n) \int(n + |\tau|) \qquad siendo\ \tau\ el\ retardo\ de\ correlación$$

Ecuación 3 Correlación y autocorrelación

Cuando la señal se encuentre en la fase máxima este tipo de correlación podrá ser empleada como análisis, la velocidad de conducción de unidades motoras aisladas o MUCV son uno de los parámetros esenciales durante el uso de esta técnica y es determinado durante los desfases temporales en los picos máximos de lo canales adyacentes orientados a las fibras musculares paralelamente con su distancia espacial.

Análisis frecuencial

Transformada de Fourier

La Fast Fourier Transformations (FFT) o Transformada rápida de Fourier es usada en el análisis y estimación de frecuencias de las señales EMG. Usualmente se puede considerar a una EMG como la suma de ondas sinusoidales con frecuencias diferentes. La FFT puede ser definida como la descomposición de la señal EMG bajo su contenido sinusoidal puro; este tipo de análisis nos permite la obtención de graficas de distribución de energía de la señal, en un intervalo de tiempo definido, y mediante la distribución de energía por cada una de las frecuencias asociadas a la señal (véase Figura 5)

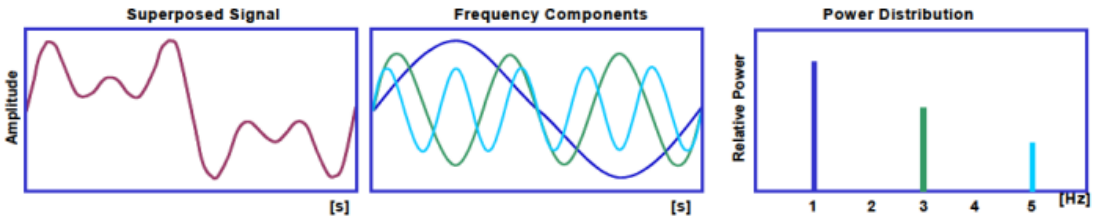


Figura 5 Transformada de Fourier

Transformada de Wavelet

Es una herramienta empleada en el análisis local de señales transitorias no estacionarias, una de sus ventajas es que puede ser empleado como un filtro en tiempo discreto, conocido como Filtro WT y podría ser un método bastante eficiente para la clasificación de las señales EMG; se usa también para la evaluación de la forma de onda del PAUM y su espectro a través del tiempo.

La WT está compuesta por la CWT (transformada contínua) y la DWT (transformada discreta). Estas transformadas permiten el análisis de señales de forma similar y comparable a la transformada de Fourier, la diferencia se encuentra en que la WT entrega información tanto temporal como en frecuencia de manera casi simultánea mientras que la TF solo nos permite una obtener información en función de la frecuencia. (Véase Figura 6)

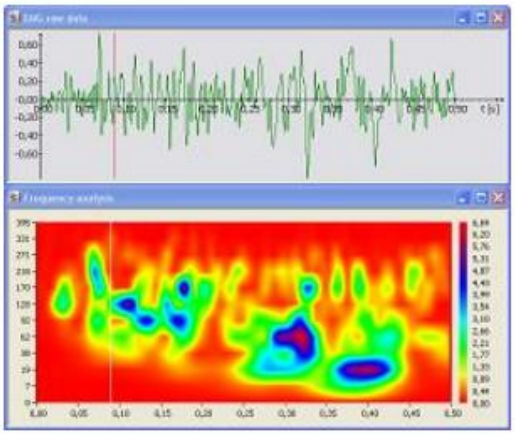


Figura 6 Transformada Wavelet

Distribución deWigner-Ville (WVD)

Consiste en una representación de la frecuencia en función del tiempo, ya que utiliza la totalidad de la información disponible de la señal obtenida, sin embargo, en algunas ocasiones se puede considerar y percibir la señal de manera casi estacionaria. Esta distribución nos permite también distinguir los rangos frecuenciales de las UM, es posible también visualizar patrones de reclutamiento durante la ejecución de movimientos complicados o de sobreesfuerzos del músculo.

$$W_x(t,\omega) = \int_{-\infty}^{\infty} x\left(t+\frac{\tau}{2}\right) x^*\left(t-\frac{\tau}{2}\right) e^{-j\omega t} d\tau \qquad \begin{matrix} x(t) = señal \\ x^*(t) = conjugada\ correspondiente \end{matrix}$$

Ecuación 4 Distribución de Wigner-Ville (WVD)

Análisis estadístico

Entropía de la señal (H)

Relación entre amplitud de cada muestra vs la señal medida. Así se cuantifica la capacidad de predicción del valor de amplitud valor a valor y sus características. En otras palabras, refleja la variabilidad estocástica de la señal.

Primer cruce por cero de la función autocorrelación (ACFZERO)

Al igual que la entropía de la señal, este análisis también cuantifica la variabilidad de la señal, estocásticamente, sin embargo, también toma la función ACF de autocorrelación la cual describe cuan similares son dos partes distintas de la misma señal.

Chi-value (x)

Basado en la distribución frecuencial de las muestras, esta frecuencia corresponde a la probabilidad de que se obtenga una muestra con un valor de amplitud en un intervalo determinado de tiempo. El Chi-Value corresponde al valor absoluto de la desviación, cabe mencionar que debe asumirse que se trata de una distribución Gaussiana.

Clasificación de parámetros

Una vez obtenidos los parámetros elementales de las señales EMG en función del tiempo, de la frecuencia y completado por un análisis estadístico, se puede realizar una clasificación de cada uno de estos parámetros y posteriormente se puede realizar una comparación con patrones de patologías o enfermedades ya definidos, también podría ser comparable entre individuos sanos y encontrar variaciones entre ellos para recalibrar los procedimientos.

Redes Neuronales artificiales

Las redes neuronales son sistemas computacionales que buscan el en sus características asemejarse a una red neuronal biológica por medio de aprendizaje. Estas redes neuronales artificiales cuentan ciertas características como: generalización, habilidad de aprendizaje bajo concepto de la experiencia y sin la necesidad de la

implementación de un modelo matemático inicial, cuenta con una adaptación a cambios ambientales y dispone de una habilidad para procesar datos en condiciones de degrado o incompletos, lo cual lo posiciona como un sistema con grandes ventajas para este tipo de aplicaciones. Por otro lado, su eficiencia depende de la gran cantidad de muestras disponibles, así como un tiempo prolongado de entrenamiento y aprendizaje. (Véase Figura 7)

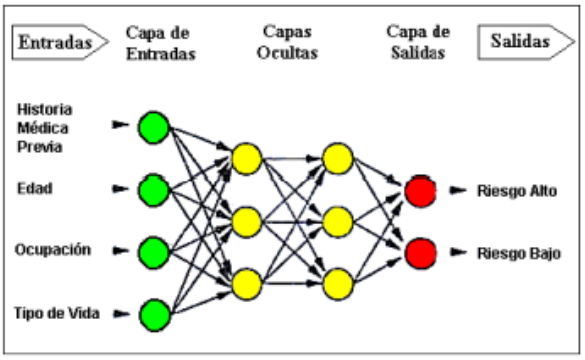


Figura 7 Redes neuronales artificiales

Lógica difusa

Al igual que las redes neuronales artificiales, la lógica difusa simula el comportamiento de la decisión humana, introduciendo experiencia de un experto durante la elaboración de las reglas de decisión. Este sistema consiste en una aproximación no lineal, obtenido por medio de la descomposición del espacio de varios sistemas lineales, de entrada, en espacios parciales y salida por medio de una ecuación lineal.

Cada una de estas entradas toman un valor distinto el mismo que luego es linealizado a fin de obtener una proporción estadística y no un valor absoluto sobre la medición. (véase Figura 8)

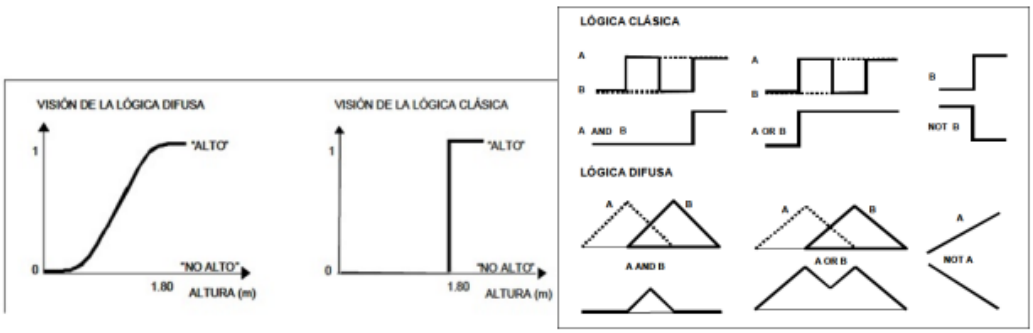


Figura 8 Visión de la lógica difusa

Aplicaciones de las señales EMG

Trastornos neuromusculares

Son aquellas patologías que pueden afectar el sistema nervioso periférico, este sistema se divide en varias secciones anatómicamente definidas a partir de:

- Células de la asta anterior
- Ganglios sensoriales
- Raíces nerviosas
- Plexos
- Nervios periféricos
- Unión neuromuscular
- Músculos distales

Estas divisiones serán puntos de partida para diagnosticar y buscar causas subyacentes de alguna enfermedad.

Las EMG son el procedimiento más eficaz para la localización y pronóstico de radiculopatía; en algunos casos existen activaciones espontáneas de los axones, los cuales ocurren en las fibras musculares denotando un daño axonal; esta actividad espontánea evidencia la existencia de una denervación aguda; En la Figura 9 se puede observar la clasificación de los trastornos musculares.

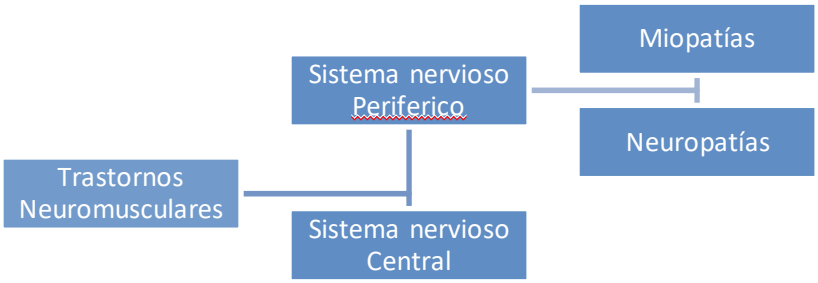


Figura 9 Clasificación de Trastornos Neuromusculares

Análisis musculares

Cuando se produce una actividad espontánea en las señales EMG durante la colocación del electrodo de aguja y sin haber ejercido ningún tipo de esfuerzo, podemos deducir que no es una acción normal y que se está presenciando algún tipo de patología.

Generalmente un músculo normal mantiene un potencial de membrana en reposo de -80mV respecto a los fluidos extracelulares. Si se presentara una lesión o denervación, este potencial de membrana es positivo cada vez más debido a iones de Na+ en una membrana celular dañada. (Procesamiento de señales EMG en trastornos Neuromusculares, 2013)

En la Tabla 1 se puede observar las diferentes características de los patrones patológicos en las señales EMG.

Tabla 1 Características de los patrones patológicos de la actividad espontánea en las señales EMG y las posibles enfermedades asociadas.

Señal	Patrón Patológico en el EMG	Característica de la señal	Miopatías asociadas	Neuropatías asociadas
	Ondas agudas positivas	Frecuencia: 0,5-1,5 Hz Amplitud: 20-1000uV Duración: 10-30 ms	Miopatías inflamatorias Distrofia muscular Miositis por cuerpo de inclusión Miopatía congénita Rabdomiólisis Trauma muscular Triquinosis	Radiculopatía Neuropatía periférica axonal Plexopatías Neuropatías por atrapamiento Enfermedad de la motoneurona Mononeuropatías
	Fibrilaciones	Frecuencia: 0,5-1,5 Hz Amplitud: 20-1000uV		
	Descarga mioquímica	Forma contínua: Frecuencia: 5-10 Hz Forma discontínua: Frecuencia: 0,1-10 Hz	Parálisis de Bell Esclerosis múltiple Polirradiculopatía	Cuadro de radiación plexopática Lesiones nerviosas crónicas
	Descarga miotónica	Frecuencia: 20-100 Hz	Distrofia miotónica Miotonía congénica Paramiotonía Parálisis periódica hiperpotasémica Polomiositis Deficiencia de maltasa ácida	Radiculopatía crónica Polineuropatía axonal crónica
	Descarga repetitiva compleja	Frecuencia: 10-100 Hz Amplitud: 50-500 uV De inicio y pausa rápida	Miopatías inflamatorias Distrofia muscular del anillo óseo Mixedema Síndrome de Schwartz-Jampel	Neuropatía crónica o radiculopatía Poliomielitis Atrofia muscular espinal Enfermedad de la motoneurona Neuropatía hereditaria

Reconocimiento de patrones de movimiento

A continuación, se muestra un pequeño resumen en la metodología de reconocimiento de patrones de movimiento a partir de señales EMG del sistema mostrado Brazo-antebrazo, este modelo parte desde el diseño y la construcción de un sistema de instrumentación que capta la señal tomando en consideración la normatividad SENIAM para electromiografías superficiales y a su vez obtener las características de las señales EMG (véase Figura 10). Adicionalmente se evidencia la aplicación de técnicas de procesamiento basado en:

- Aproximaciones temporales
- Modelamientos de parámetros
- FFT, STFT, WT

Lo cual busca reconocer patrones de movimiento por medio de redes neuronales. (BETANCOURT O. & FRANCO B., 2004)

Figura 10 Reconocimiento de patrones de movimiento en sistema Brazo-antebrazo

Control de prótesis

Las EMGS o señales electromiográficas superficiales son ideales para controlar dispositivos motores como que pueden homologar el movimiento de las extremidades, por ejemplo, prótesis para mano.

Existen muchas ventajas para emplear las señales EMGS como lo es:

- Facilidad en recolección de datos (muestras)
- Proceso no invasivo, uso cómodo para la persona que emplea un dispositivo prostético
- intercambiabilidad entre usuarios
- facilidad en mantenimiento
- facilidad en calibración

las técnicas en función del tiempo; mencionadas anteriormente como la transformada de Wavelets son de las más empleadas para el control de prótesis, se han empleado estas técnicas sobre el análisis de movimiento para fabricación de equipos prostéticos para manos. (Harold A. Romo, 2007) (véase Figura 11)

Un componente fundamental de muchas prótesis modernas es el sistema de control mioeléctrico, que utiliza las señales del electromiograma (EMG) de los músculos de un individuo para controlar los movimientos de la prótesis. A pesar de la extensa investigación centrada en el control mioeléctrico del brazo y los movimientos gruesos de la mano, el control individual y combinado de los dedos más diestros no ha recibido la misma atención. Discriminar con precisión entre los movimientos de los dedos individuales y combinados utilizando señales EMG de superficie, de modo que las diferentes posturas de los dedos de una mano protésica puedan controlarse en respuesta. Para ello, se utilizan dos electrodos EMG ubicados en el antebrazo humano para recopilar los datos EMG. Varios conjuntos de características se extraen y

proyectan de una manera que garantiza la máxima separación entre los movimientos de los dedos y luego se alimentan a dos clasificadores diferentes. La segunda contribución es el uso de un enfoque de posprocesamiento de fusión de datos Bayesiano para maximizar la probabilidad de clasificación correcta de los datos EMG pertenecientes a diferentes movimientos. (Rami N.Khushaba, 2012)

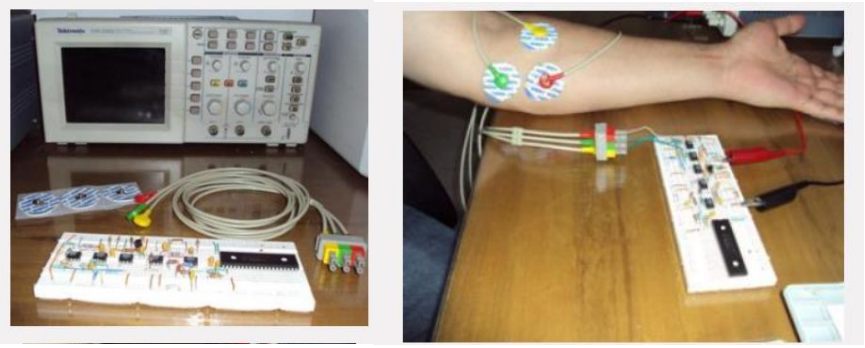


Figura 11 Adquisición de señales EMG

2 METODOLOGÍA

En esta sección se detallará el paso a paso de la implementación de este componente, así como el enfoque elegido para posteriormente iniciar las fases de prueba y verificación de resultados.

2.1 Arduino

Para este componente se usará una placa Arduino UNO; Arduino es una placa basada en un microcontrolador ATMEL, cuenta con su propio entorno de programación (véase **Figura 12**) , ARDUINO IDE, al igual que su propio lenguaje el cual es muy similar a C++. Este tipo de placas permiten realizar una gran variedad de programaciones, ya que se pueden usar para automatización, conectar varios dispositivos, interactuar entre varios programas, adquisición de señales, entre otros.

Para este componente la placa Arduino nos permitirá adquirir los datos referentes a la lectura de los electrodos, y será el nexo con MATLAB para el procesamiento y visualización de los mismo.

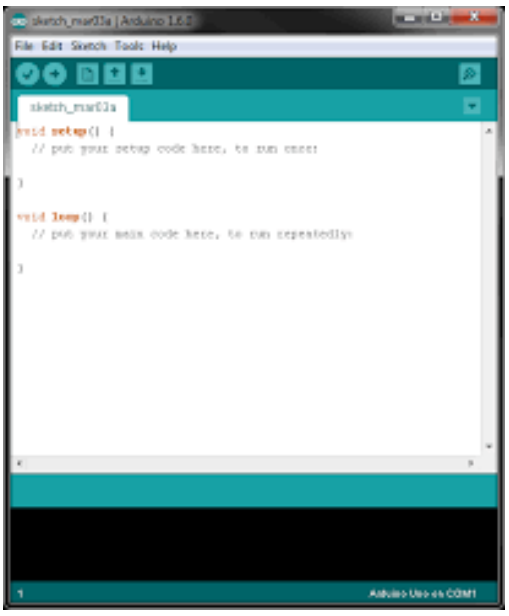


Figura 12 Interfaz Arduino IDE.

Configuración y adquisición de datos Arduino IDE

Para la adquisición de datos se usará una de las entradas analógicas de Arduino en este caso A0, como primera validación usaremos el monitor serial de Arduino el cual nos permitirá verificar que la placa se encuentra trabajando, se puede usar también la

opción de serial plotter, la cual nos permite visualizar una gráfica en tiempo real de los datos que están siendo adquiridos. (véase Figura 13)

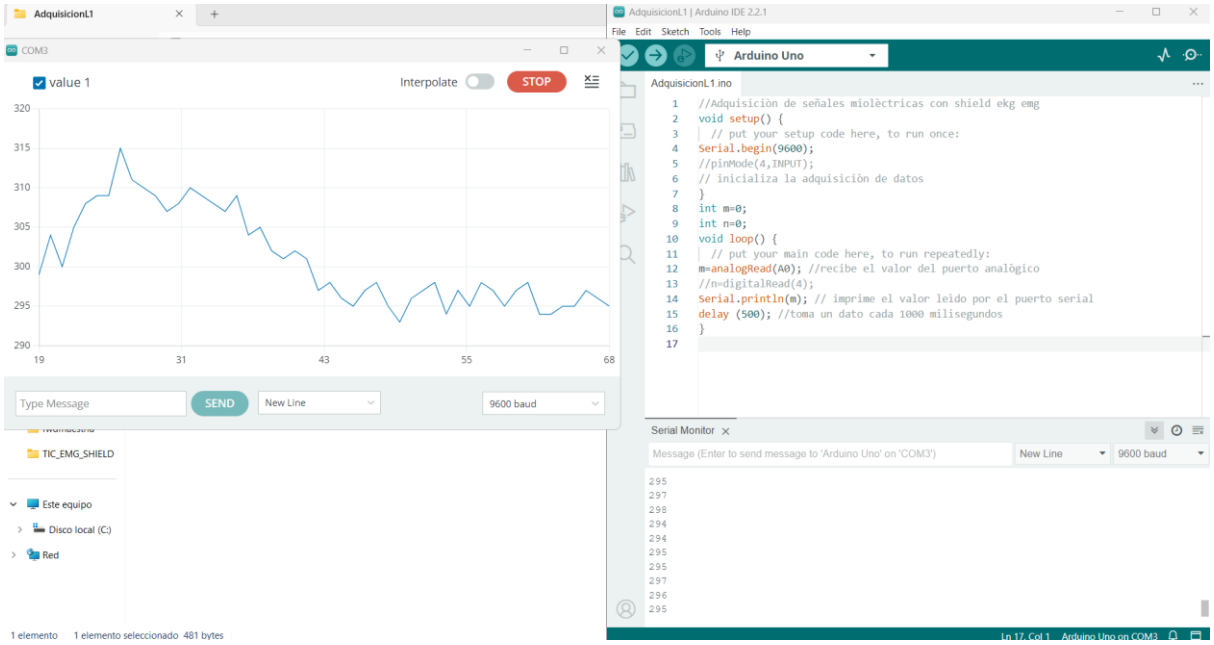


Figura 13 Programación Arduino IDE & Monitor Serial & Serial Plotter

Es importante considerar que para que la placa trabaje de manera adecuada se debe elegir correctamente el tipo de placa a usar, así como habilitar el puerto COM en el cual se conectará la entrada USB de Arduino, en la Figura 14 se muestra la alimentación y conexión a la placa y el pin usado para la adquisición de datos.

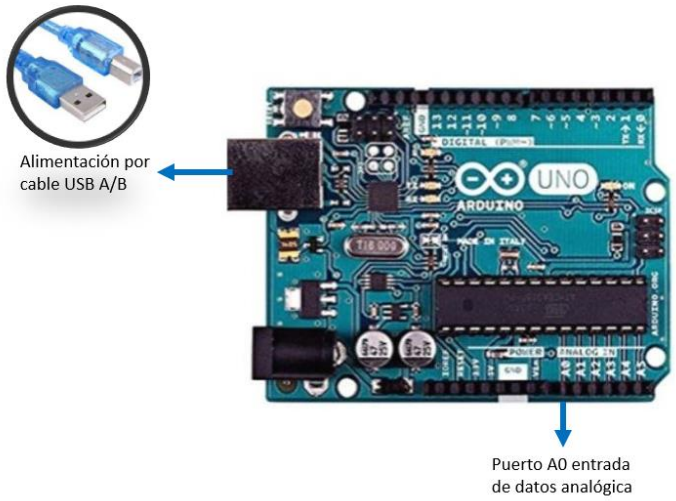


Figura 14 Alimentación y conexiones placa Arduino.

2.2 Shield-EKG-EMG

El Shield EKG-EMG es una placa compatible con Arduino la cual permite capturar señales electromiográficas (EMG) y electrocardiográficas (ECG/EKG), generalmente es

empleado en proyectos biomédicos o biomecánicos que analiza los movimientos musculares, cardiografías entre otros.

Este shield se conecta de manera directa a la tarjeta de Arduino, a fin de recibir la alimentación que genera la placa principal (Arduino) y el pin de salida del Shield que para nuestro ejercicio será el pin de entrada del Arduino. (véase Figura 15)

Es importante considerar la colocación de los jumpers en el shield los cuales habilitarán una u otra salida, y lo propio para la configuración de entrada de los electrodos.

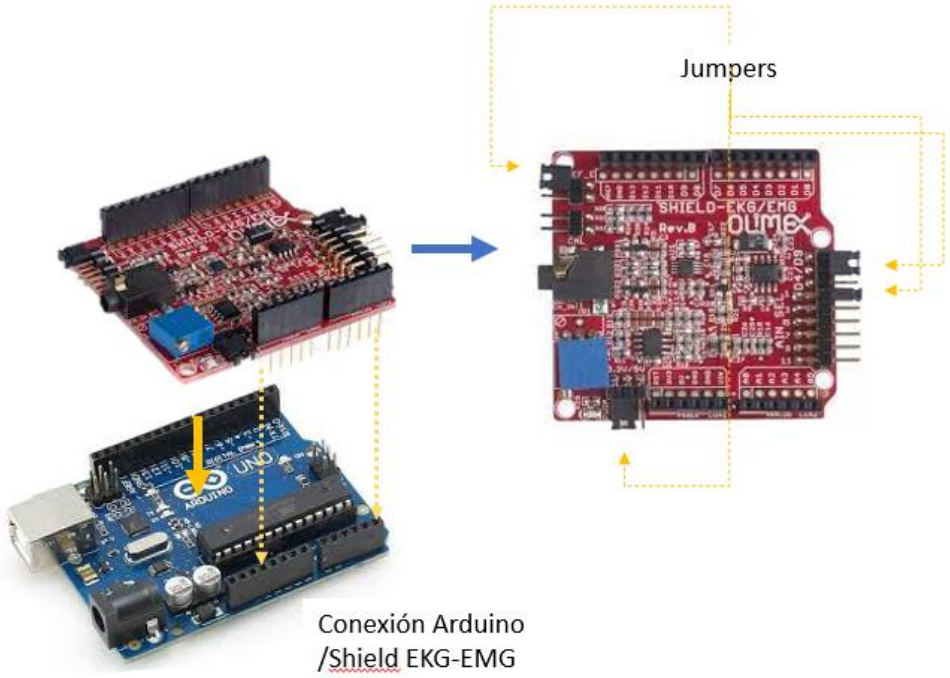


Figura 15 Conexión Arduino & Shield EKG-EMG

2.3 Software Matlab

Matlab es una plataforma de programación que nos permite analizar datos, generar algoritmos, crear modelos matemáticos, simulaciones complementarias, entre otros. Una de las principales ventajas de esa plataforma es su compatibilidad con otros programas u otros dispositivos lo que permite que se pueda analizar la información en tiempo real. Cuenta con su propio entorno y lenguaje de programación (lenguaje M), (véase Figura 16) y cuenta con una interfaz gráfica que permite realizar el procesamiento y adquisición de datos de manera más dinámica (véase Figura 17).

Para este componente, inicialmente se configurará una interfaz gráfica en APP Designer la cual nos permitirá visualizar los datos adquiridos por el Shield en conexión con Arduino.

Es importante considerar que para poder realizar la conexión entre Matlab y Arduino se requiere instalar las librerías adicionales de Arduino disponibles en la biblioteca de

Matlab.

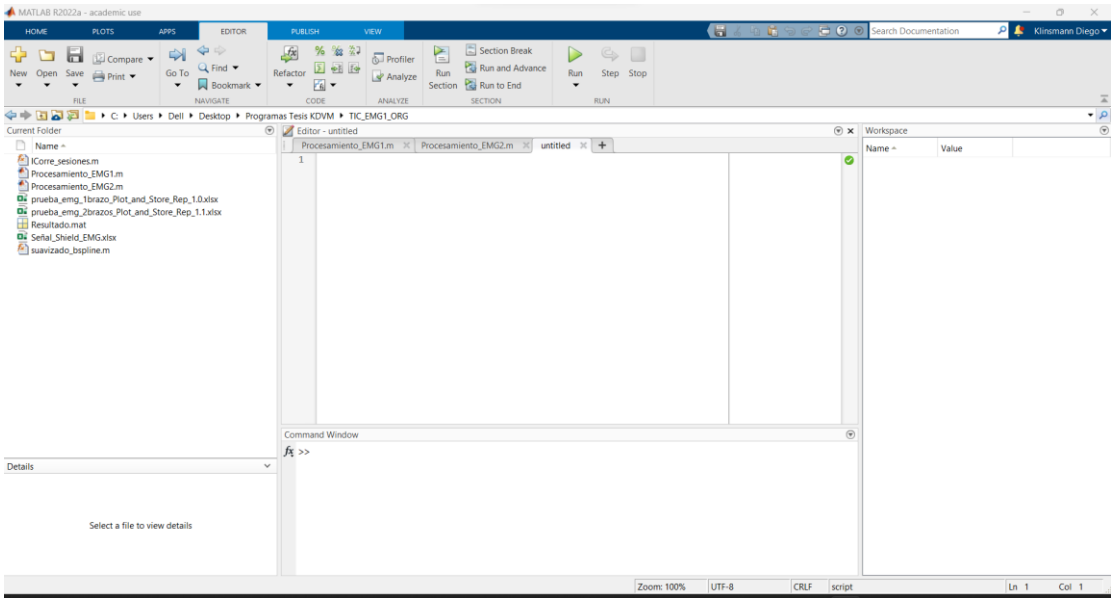


Figura 16 Interfaz Home Matlab

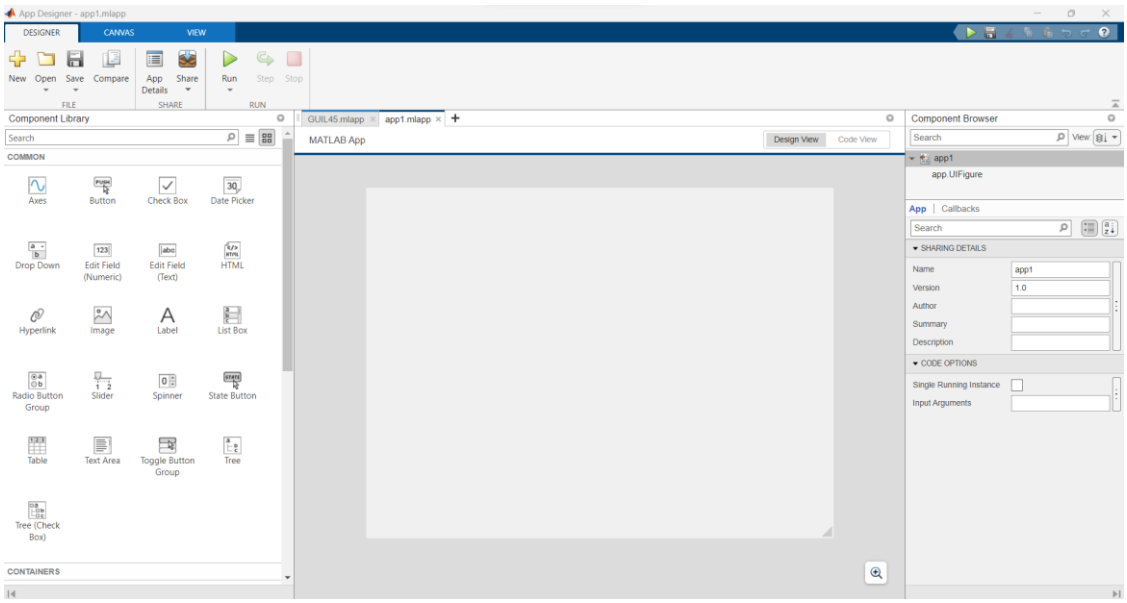


Figura 17 Interfaz APP Designer Matlab

Configuración y diseño de interfaz Matlab

Como hemos mencionado para lograr un análisis de datos más dinámico es necesario la interacción entre diferentes programas y la implementación de interfaces gráficas que nos permitan evidenciar la adquisición.

En este caso se realiza una interfaz en APP Designer la cual contendrá una gráfica, una tabla de datos y los respectivos botones de interacción. (véase Figura 18)

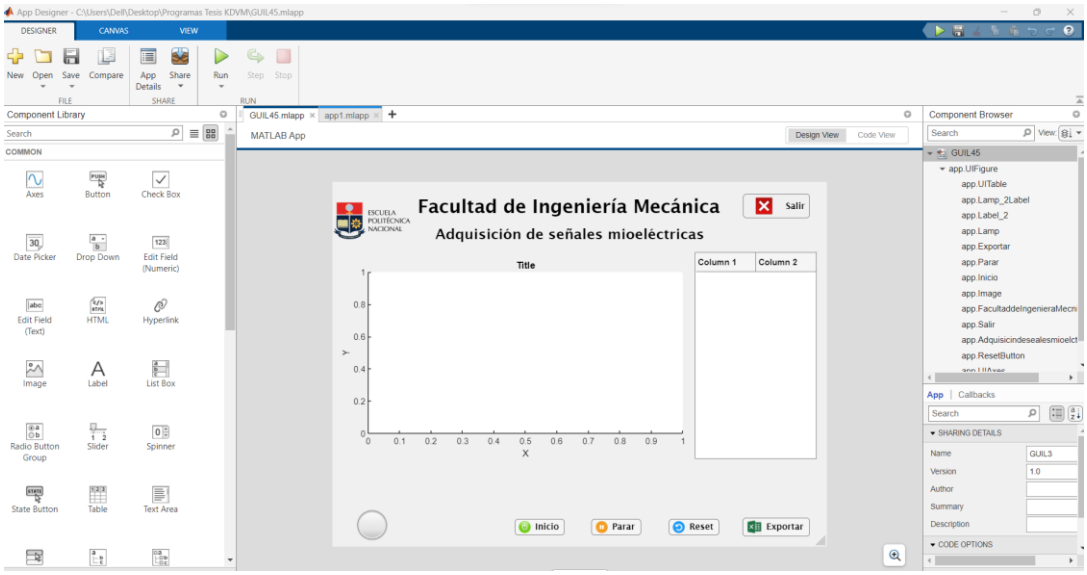


Figura 18 Diseño de interfaz gráfica APP Designer (Design View)

Una vez que se tenga el diseño de interfaz se procederá a configurar cada uno de los elementos, lo cual podremos hacer desde la vista de Código (Code View) cada uno de los elementos que se incluyan en la interfaz tendrá un nombre en específico y es con el cual se podrán programar según nuestra necesidad.

Para la conexión Arduino- Matlab se empleará la siguiente programación, Figura 19, se hará el llamado al Puerto COM configurado previamente para que por medio del Puerto serial (conectado a Arduino) los datos puedan ser transferidos e interpretados en la interfaz de Matlab.

```
properties (Access = private)
    S1 % Propiedad para el switch de adquisición
end

methods (Access = private)
     function startupFcn(app)
    % app.S1=serialport("COM3",9600,"Timeout",5);
      end

end
```

```
% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn2(app)
        app.S1=serialport("COM3",9600,"Timeout",1);
    end
```

Figura 19 Programación & Conexión Arduino-Matlab

En la Figura 20 se indica la estructura de la interfaz, en la cual se tienen los siguientes componentes:

- Axes: Gráfica en tiempo real de datos adquiridos.

- Table: Tabla de valores en tiempo real
- Botoneras:
    - Inicio: Inicia la adquisición.
    - Parar: Coloca en pausa la adquisición de datos
    - Reset: Resetea el Arduino
    - Exportar: Exporta los datos adquiridos a una hoja excel para su análisis.
    - Salir: Cierra todos los programas. (Interfaz, APP designer, y matlab)

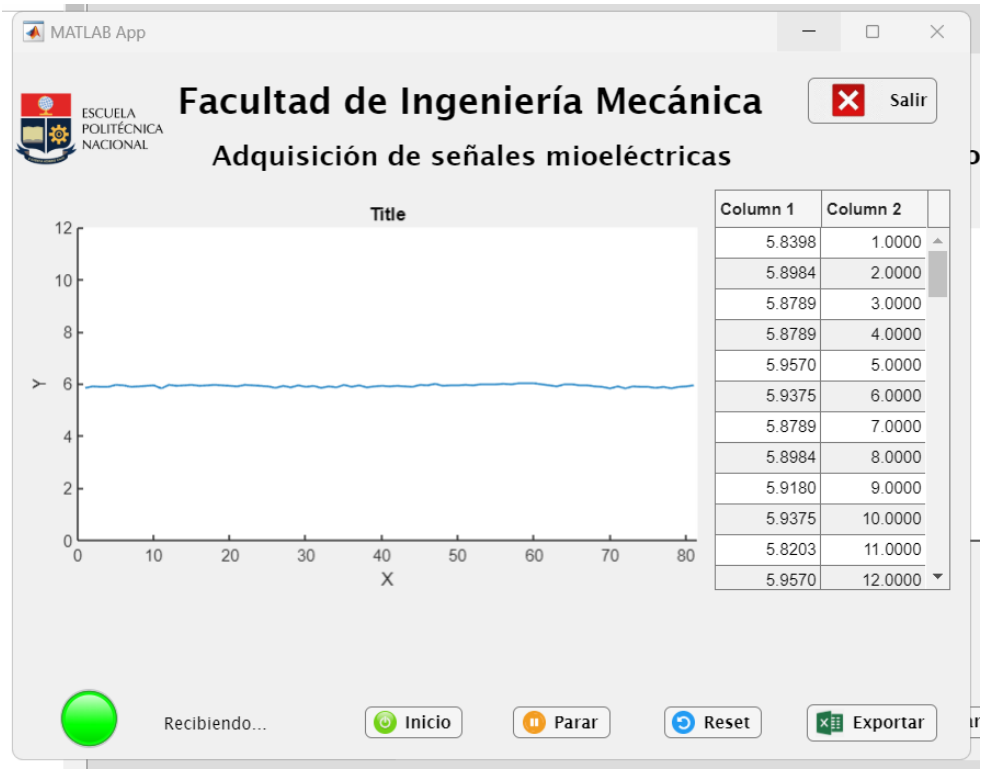


Figura 20 Interfaz gràfica para adquisiciòn de señales mioeléctricas

2.4 Electrodos (Ubicación y consideraciones)

Los electrodos que se usarán son parches adhesivos con cables que se conectan a la tarjeta shield y permitirán realizar la adquisición de señales en la interfaz desarrollada. Es importante considerar que la ubicación de estos electrodos es fundamental para obtener una lectura adecuada.

Para este componente se usará cable estéreo para electrodos como se muestra en la Figura 21 este terminal se conectará en el puerto estéreo del shield, esto nos permitirá recibir las señales de 3 electrodos, los mismos que serán ubicados de manera estratégica para garantizar la calidad de la señal adquirida.

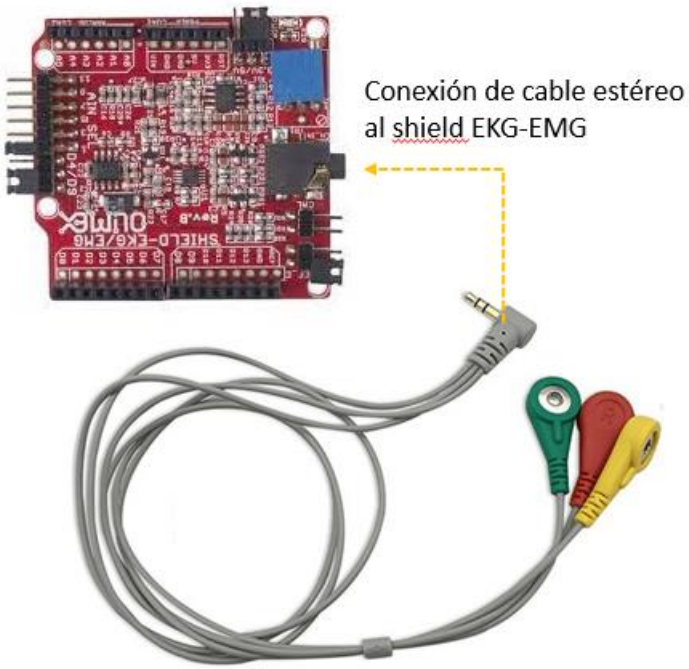


Figura 21 Conexión cable electrodos & Shield

Ubicación de electrodos sobre brazos y hombros

Los electrodos nos permitirán medir la actividad eléctrica de músculos y nervios, los nervios emiten señales eléctricas las cuales son captadas por los músculos reaccionando al tipo de estímulo emitido, durante este proceso se emiten señales a manera de pulso eléctrico las mismas que pueden ser medidas.

Para lograr la lectura de los músculos de los brazos, del hombro al codo, y de antebrazos, desde el codo hasta la muñeca, la manera ideal de ubicar los electrodos es colocándolos a una distancia mínima entre ellos, y cerca del punto de elongación de las fibras musculares, tal como se muestra en la Figura 22, el tercer electrodo podrá ser colocado como referencia en una zona ósea.

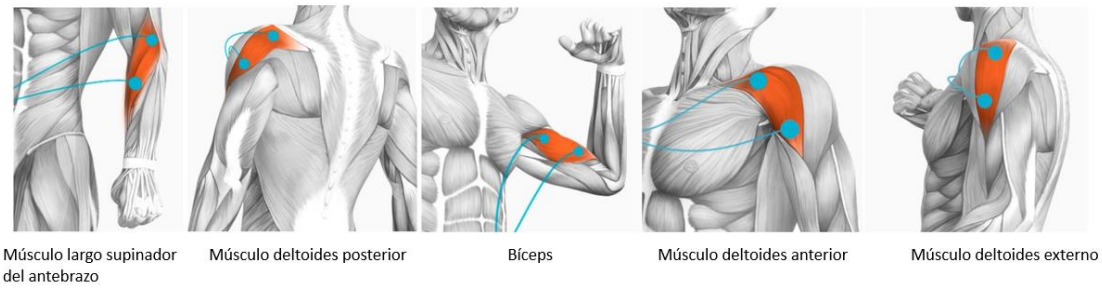


Figura 22 Ubicación de electrodos en músculos de brazo y hombre

Consideraciones previo al uso de electrodos

Es importante considerar los siguientes puntos antes de realizar la adquisición de

señales:

- Posición de los electrodos.
    - En este caso tenemos 3 cables de electrodos, rojo, verde y amarillo; donde los cables rojo y verde son la señal de referencia (R) los cuales irán en el músculo que se desea medir, y el electrodo amarillo (L) irá conectado a una sección ósea alejada del músculo analizado.
- La piel debe estar limpia y sin ningún tipo de loción o crema, para que la adherencia del electrodo sea adecuada.
- Los electrodos deben estar en buenas condiciones de preferencia nuevos, en cada fase de prueba.

2.5 Adquisición de señales

Luego de realizar la programación de cada elemento de la interfaz, y validar la comunicación entre Arduino y Matlab, se procede a las pruebas de adquisición, para ello es necesario considerar lo siguiente:

- El rango de frecuencia en la que trabajan estás señales musculares que es de 20Hz-500Hz con una amplitud de 50 μV a 5mV
- Los electrodos deben encontrarse en buen estado, de esto dependerá una buena adquisición.
- En ocasiones es requerido una etapa de amplificación de la señal debido al rango de amplitud en el que trabaja las señales musculares
- La etapa de filtrado es esencial para obtener una señal más nítida y que permita realizar un análisis adecuado.
- Placa Arduino conectada al shield EKG-EMG como se indica en la sección 2.2
- Cable de alimentación USB A/B
- Matlab, debe contar con las librerías de Arduino para garantizar la conexión.

En la Figura 23 , se muestra el diagrama de bloques empleado para este componente, la mayoría de las fases de pre-amplificación, amplificación y filtrado se manejará de manera digital, y con ayuda del shield EKG-EMG, la digitalización será generada por medio de la programación y configuración en ARDUINO IDE Y MATLAB, al igual que el procesamiento de la señal.

La fase de pre-amplificación es ejecutada dentro del IDE inicial y con ayuda del Shield, al ser un componente dedicado a este tipo de adquisiciones, incluye en su configuración amplificadores operacionales, por lo que su señal en bruto es visible y diferenciable desde su adquisición.

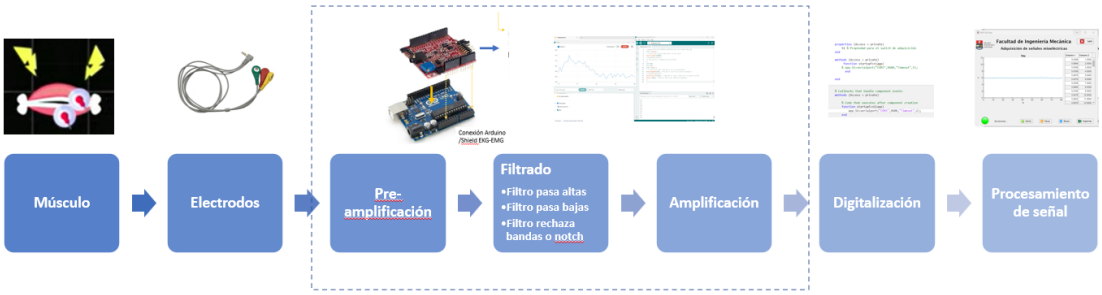


Figura 23 Diagrama de bloques del componente

Por otro lado, la fase de filtrado la misma que nos permite eliminar ruidos o cualquier tipo de alteraciones en nuestra señal también se ve beneficiada por el Shield, sin embargo, dentro de la configuración de la interfaz se emplean una serie de filtros digitales que nos permitan mejorar la calidad de la señal, como se indica en la Figura 24.

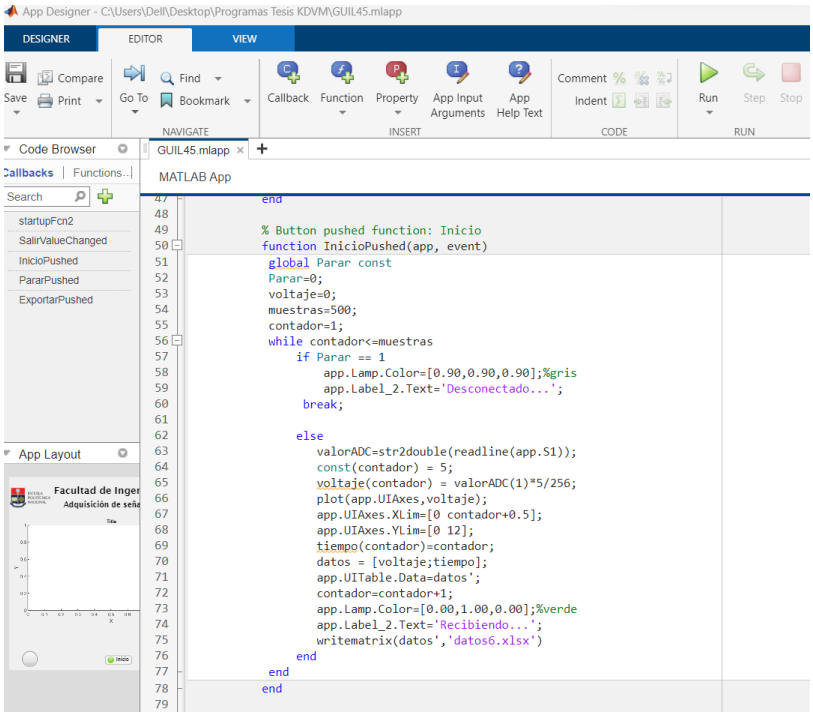


Figura 24 Fase de filtrado & amplificado final APP Designer

Finalmente se iniciará la adquisición de señales por medio de un protocolo de prueba que se detallará más adelante junto con la sección de resultados, donde se analizará y desarrollará también el procesamiento de la señal.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En esta sección se detallará los resultados de la adquisición de datos y el procesamiento de estos, luego del procesamiento se emitirán conclusiones y recomendaciones.

3.1 Resultados

Protocolo de pruebas

Para la adquisición de datos se ejecutará el siguiente protocolo de pruebas, como se muestra en la Tabla 2.

Tabla 2 Protocolo de pruebas

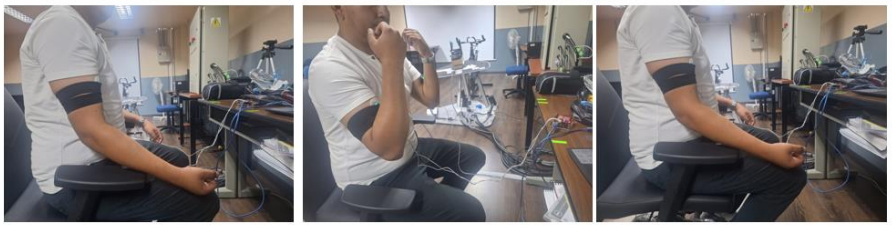
Protocolo de pruebas		
	Shield EKG-EMG	**Sistema Delsys**
Cantidad de ciclos	5	5
Extremidad por analizar	Brazo	Brazo
Músculo	Bíceps	Bíceps
Tipo de adquisición	muestreo	muestreo
Delay microcontrolador (ms)	500	>100
Descripción de la prueba	Colocar 2 electrodos (verde & rojo) a lo largo del músculo a medir. Conectar los cables estéreo de los electrodos al plug del shield. Conectar el USB A/B al computador y la placa Arduino respectivamente. Abrir APP Designer en Matlab. Correr APP, e iniciar la interfaz. Pausar y exportar datos.	Colocar dispositivo lector en el músculo a medir. Conexión entre dispositivo e interfaz del equipo Delsys Correr prueba en interfaz. Detener y exportar datos.
Observaciones	Es importante que la piel se encuentre limpia, seca y libre de lociones o cremas, de esto dependerá la buena calidad de la señal	

Se realizaron 5 ciclos de movimiento, y se tomaron datos de los dos dispositivos, tanto desde la interfaz desarrollada en Matlab como en el equipo Delsys disponible en el Laboratorio de Bioingeniería de la Facultad de Ingeniería Mecánica.

Cada ciclo consiste en el movimiento de abajo hacia arriba del antebrazo, lo cual estimula el músculo Bíceps en el brazo. En la Figura 25 , se muestra cada una de las fases del ciclo.

En la Figura 26 se puede observar los datos adquiridos por la interfaz Matlab los cuales serán comparados más adelante con los valores obtenidos del equipo Delsys.

Prueba interfaz Matlab



Prueba equipo Delsys

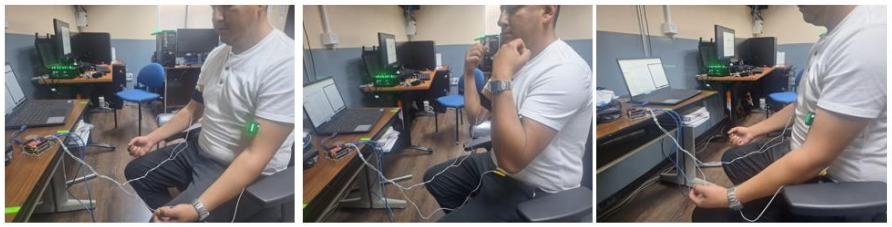


Figura 25 Ciclo de prueba interfaz & Delsys.

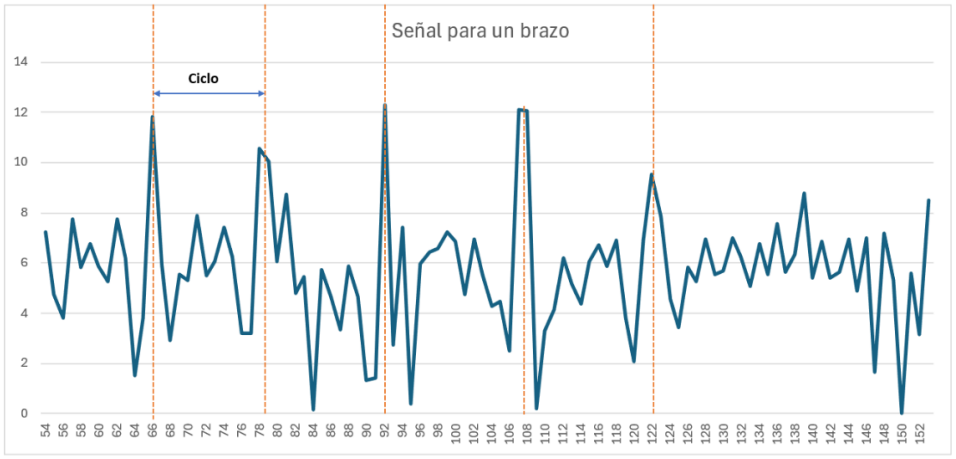


Figura 26 Datos Shield + Arduino + Matlab (sin suavizar)

Procesamiento de datos

Para el procesamiento de datos es necesario tenerlos en una hoja de cálculo, en nuestro caso Excel. El procesamiento se realizará cargando los datos filtrados y rectificados de manera digital previamente en la programación. (véase Figura 27)

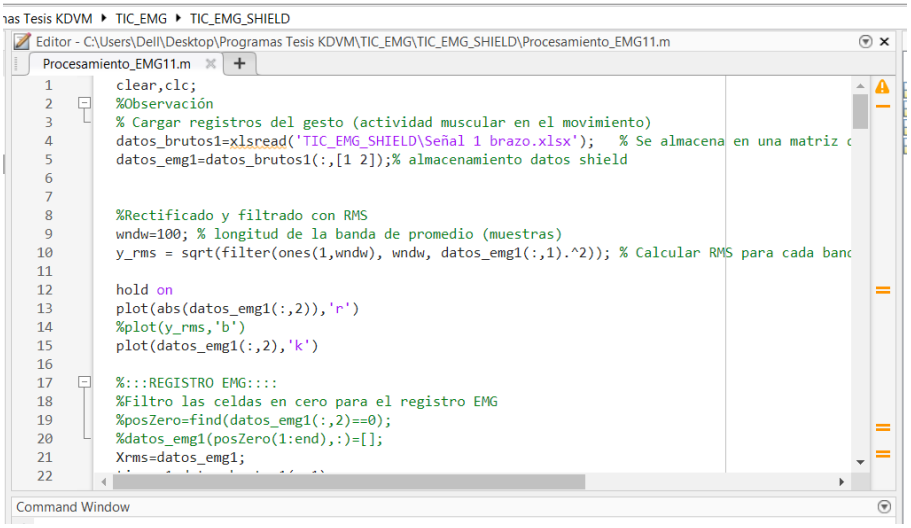


Figura 27 Carga de registros, rectificado y filtrado Shield

En la Figura 28, se muestran los datos adquiridos por medio de la interfaz Shield EKG-EMG en esta grafica se pueden notar los diferentes picos de esfuerzo que responden a los ciclos generados.

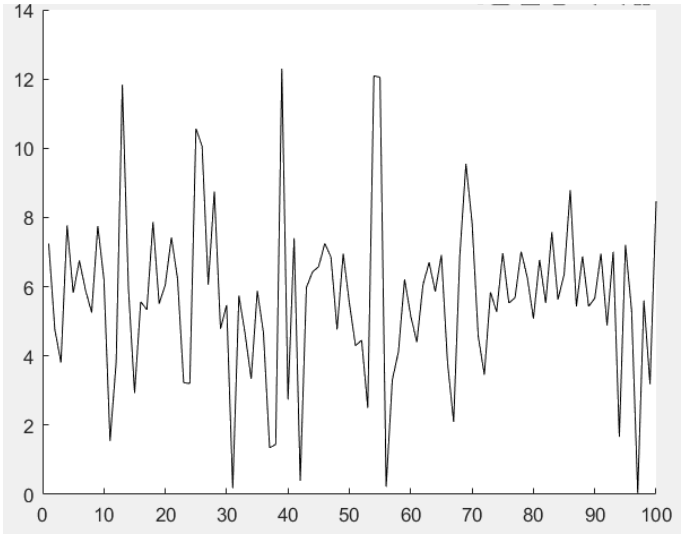


Figura 28 Señal datos brutos Shield EKG-EMG

Se realiza un suavizado de la gráfica obtenida a fin de obtener un modelo con menos ruido y un poco más lineal, en la Figura 29 se muestra tanto la gráfica inicial como la suavizada (negra y azul respectivamente.

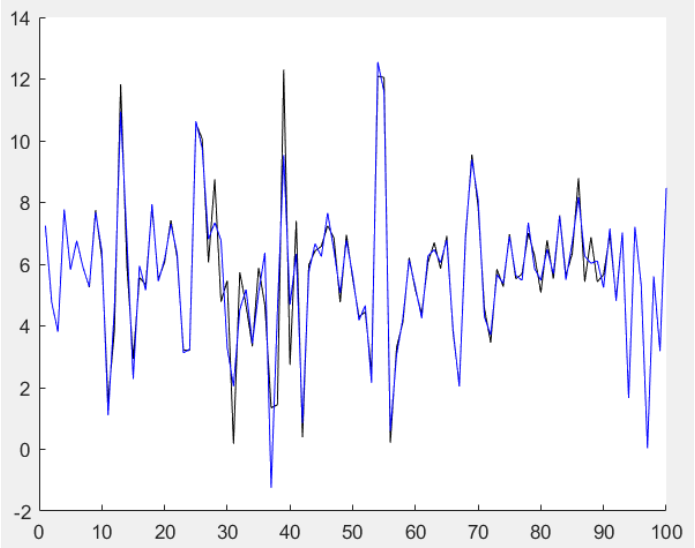


Figura 29 Valores iniciales & Valores suavisados

Se selecciona uno de los ciclos obtenidos para ser comparado con el equipo Delsys del Laboratorio de Bioingeniería de la Facultad de Ingeniería Mecánica, como se muestra en la Figura 30.

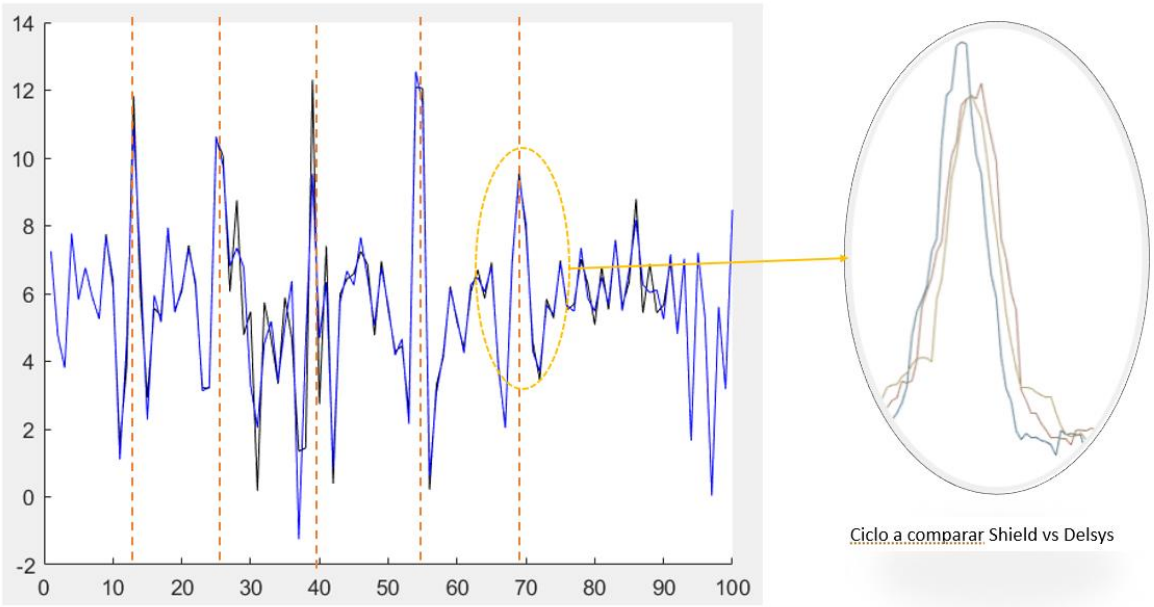


Figura 30 Selecciòn de ciclo de trabajo

Una vez que se dispone de la señal suavizada, realizamos una vectorizaciòn de la misma, que nos permita realizar una ampliación lógica del procesamiento masivo de los arreglos de datos obtenidos previamente en la fase de adquisición.

Para ello emplearemos la programación como se muestra en la Figura 31.

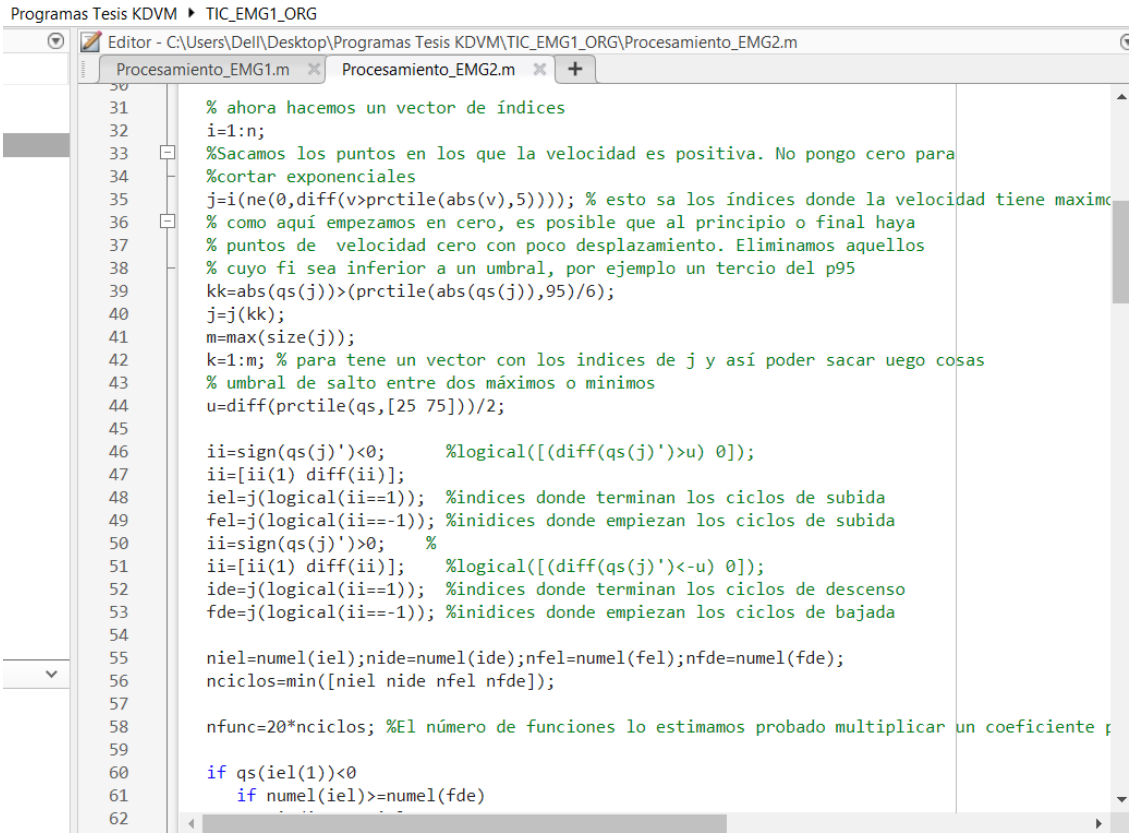


Figura 31 Programación vectorización Matlab-Datos Shield.

Es importante considerar que esta vertorización incluirá los índices de subida y bajada por cada uno de los ciclos, así como su ajuste en polos y ceros, en cada sección como se detalla en la Figura 32 .

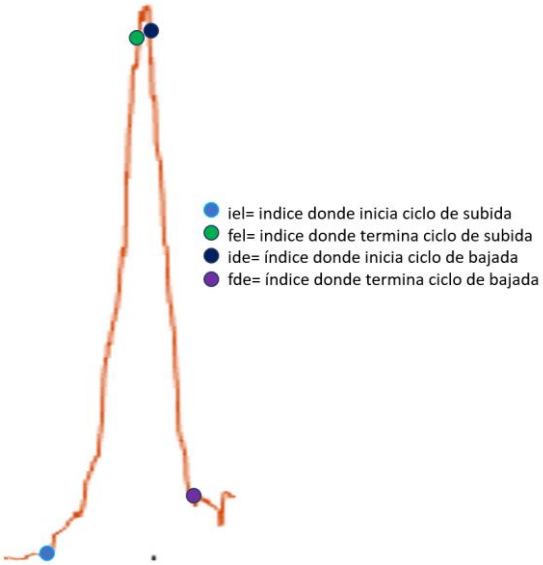


Figura 32 Indices de ciclos subida-bajada (vectorización)

Luego de la fase de vectorización, realizamos una fase de interpolación para obtener una señal más nítida y comparable con el equipo Delsys; la gráfica mostrada en la Figura 33 , denota los 5 ciclos ejecutados, sus amplitudes, longitud entre ciclos.

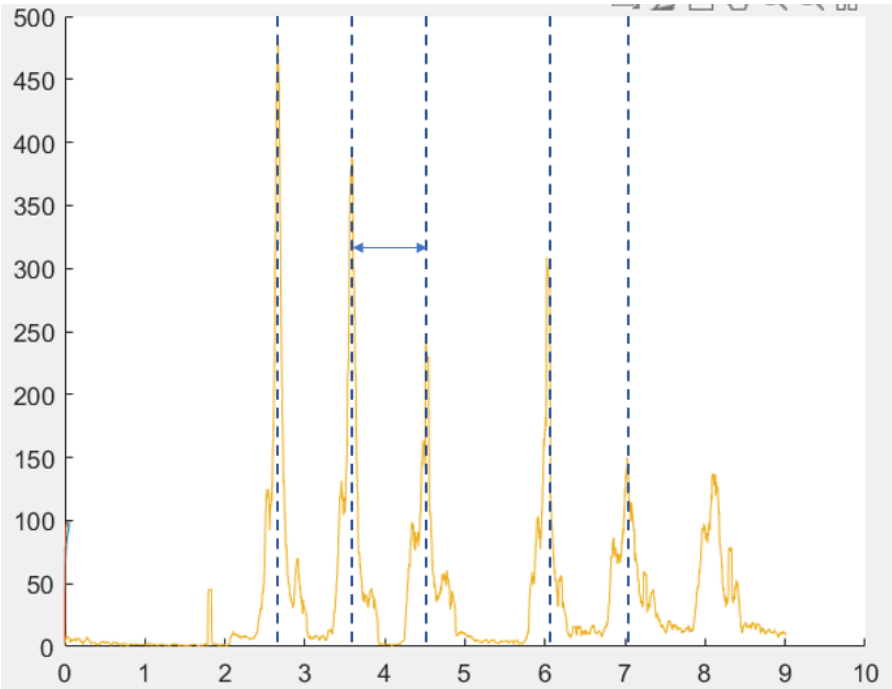
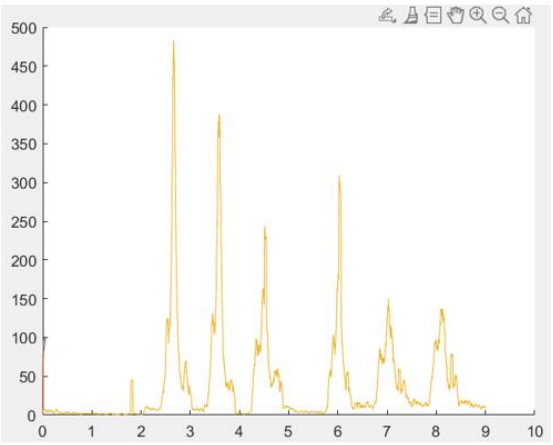
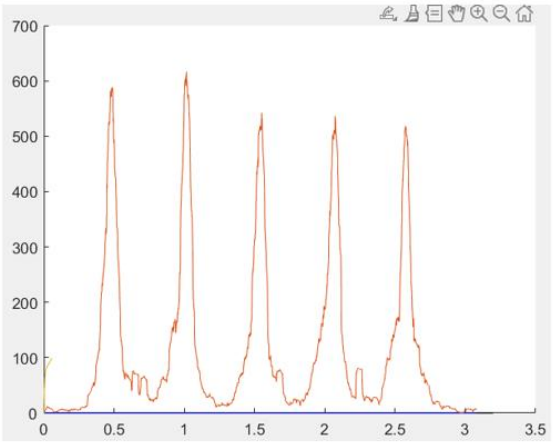


Figura 33 Curva final procesamiento Shield EKG-EMG

Luego de un procesamiento similar de vectorización e interpolación obtendremos la gráfica final, del equipo Delsys; en la Figura 34 se muestran las dos graficas a comparar.



Curva Shield EKG-EMG



Curva Equipo Delsys

Figura 34 Gráficas finales Shield vs Delsys

Finalmente realizamos una normalización de las dos gráficas obtenidas, a fin de comparar la similitud de la una versus la otra, en la Figura 35 , se puede evidenciar la diferencia entre los dos tipos de adquisición, siendo la curva azul la obtenida por nuestra interfaz (Shield EKG-EMG).

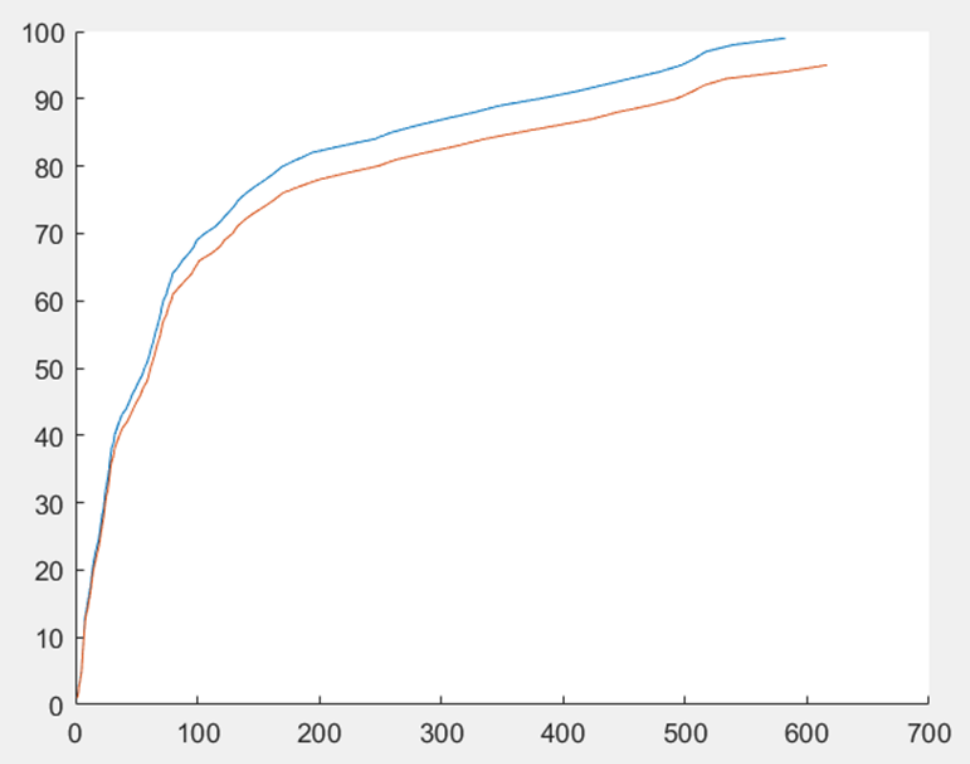


Figura 35 Comparación Shield vs Delsys (normalizadas)

Analizamos adicionalmente los datos estadísticos de las dos curvas, como Máx, Min, media, mediana, desviación, entre otros, como se muestra en la Figura 36.

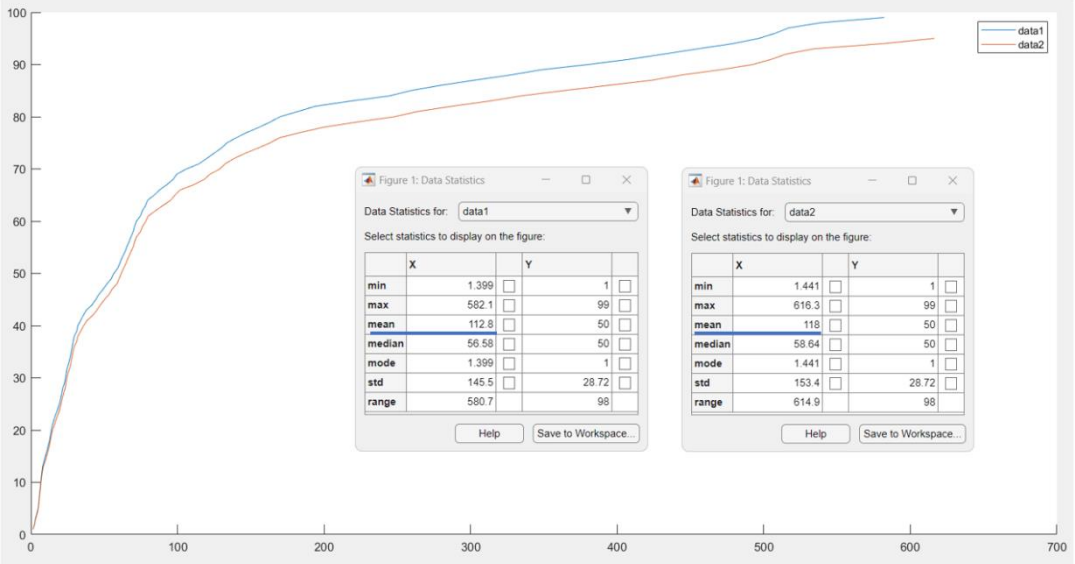


Figura 36 Datos estadísticos curvas obtenidas.

Usuaremos el valor de la media de las curvas obtenidas, para calcular el error entre los dos tipos de adquisición, donde el valor teórico o referencial, será el obtenidos por el equipo Delsys, como se muestra en la Ecuación 5 donde:

$\overline{x_2} = media\ Delsys$

$\bar{x}_1 = media\ Shield\ EKG - EMG$

$$error = \frac{\bar{x}_2 - \bar{x}_1}{\bar{x}_2} * (100\%)$$

Ecuación 5 Cálculo del error

$$\overline{x_2} = 118$$

$$\bar{x}_1 = 112.8$$

$$= \frac{118 - 112.8}{118} * 100\% = \mathbf{4.6}\%$$

De este modo Podemos evidenciar que los datos obtenidos por medio de la implementación del Shield-EKG-EMG más la interfaz de Matlab tienen un error del 4.6% respecto a los datos obtenidos en el equipo Delsys, lo que refleja un margen de error acceptable ya que nos da un nivel de confianza por encima del 95%.

3.2 Conclusiones

- Se realizó la implementación de un dispositivo Shield EKG-EMG para la adquisición de bioseñales, para lograr esta implementación se empleó una programación en Arduino y una interfaz en APP Designer de Matlab, es importante mencionar que para lograr esta adquisición se requiere de una fase de pre-amplificación, filtrado, amplificación, digitalización y procesamiento, la mayor cantidad de estas fases se lograron digitalmente por medio de programación en los diferentes softwares.

- Por medio de la investigación realizada para este componente se pudo evaluar que las señales musculares trabajan en un rango de 20Hz-500Hz con una amplitud de 50 mV a 5mV, por lo que la adquisición de las mismas por medio de Arduino es bastante conveniente por los rangos en los que esta placa trabaja. (rango de los milivoltios)

- La amplitud y las propiedades de las señales EMG dependerán siempre de varios factores, como son:
    - El tiempo de la contracción o estimulación muscular, así como de la intensidad de la misma.
    - La distancia entre los electrodos y el musculo a ser medido.
    - La zona muscular que desea analizarse.
    - El estado de la piel (espesor, tejidos, limpieza, etc.)
    - El buen estado de los electrodos así como la calidad del mismo.
    - El buen contacto entre la piel y el electrodo (se sugiere que se limpie la zona y que la misma no cuente con lociones o cremas que pudieran desprender o reducir la adherencia del electrodo).
    - El procesamiento de los datos obtenidos.

- Existen varios tipos de procesamientos para las bioseñales musculares, uno de los más comunes son el suavizado de señales, filtrado digital, normalizado, rectificado; si estas señales son analizadas en el dominio del tiempo se puede emplear: el conteo de picos o valores superiores al RMS, cruces por cero, correlación y autocorrelación; si lo son en el dominio de la frecuencia, la transformada de Fourier, la distribución de Wigner-Ville, y el método de Choi-Williams, son de los más empleados;

- Existen varios patrones de las señales EMG, estos patrones permiten reconocer

patologías durante su análisis, para poder realizarlo es importante reconocer también las características de frecuencia y amplitud que cada uno de estos tipos de señal pueden presentar.

- El dispositivo Shield EKG-EMG es un dispositivo bastante amigable al usuario ya que es de fácil conexión, puede ser usado con su propia tarjeta de programación (Olimexino) o a su vez con las tarjetas Arduino UNO y Mega ya que su configuración de pines lo permite, por otro lado, cuenta ya con varios elementos electrónicos que facilitan la adquisición de este tipo de señales ya que cuenta con amplificadores operaciones y rectificadores que mejoran la calidad de la señal adquirida.

- Durante el desarrollo del protocolo de pruebas, es importante seleccionar adecuadamente el músculo a analizar, recordemos que se debe analizar el mismo músculo, de preferencia exactamente en el mismo punto, a fin de garantizar señales comparables, cabe mencionar que para lograr una calibración ideal es importante mantener la repetibilidad al ejercicio, y al ser señales en músculos en movimiento se puede complicar en cierta medida mantener esta repetibilidad debido a factores adicionales, como son: fatiga del músculo, variaciones en el esfuerzo generado, posición de los electrodos, desgaste de los mismos, entre otros.

- Para este componente el procesamiento de datos se realizó mediante la adquisición de señales por medio de la interfaz, estos datos fueron suavizados, interpolados y desarrollamos una vectorización para obtener una ampliación lógica de los valores obtenidos, al ser un análisis en el dominio del tiempo trabajamos con cruces por cero lo que nos permitió obtener una señal mucho más nítida y comparable. Las señales obtenidas por medio del equipo Delsys tuvieron un procesamiento similar a fin de que las dos curvas fueran comparables. Luego de una fase de normalización para las dos señales, pudimos realizar un análisis estadístico que permitió calcular el error entre ellas. Como media referencial tomamos el del equipo Delsys, ya que se deseaba comprobar la veracidad del sistema implementado por medio del Shield, el error obtenido es de 4,46% lo que refleja un margen de error acceptable ya que nos da un nivel de confianza por encima del 95%, demostrando que implementar un Shield para la adquisición de este tipo de señales es bastante viable.

3.3 Recomendaciones

- Es importante considerar las siguientes recomendaciones previo a la adquisición de las señales, la posición de los electrodos es muy importante al igual que el estado y

la calidad de los mismos, se recomienda usar electrodos nuevos cada vez que se realicen estas pruebas; la piel debe estar limpia y sin ningún tipo de loción o crema, para que la adherencia del electrodo sea adecuada.

- Los softwares deben ser compatibles, es importante que previo al desarrollo de la interfaz o al inicio de su programación en Matlab, las librerías referentes a Arduino sean descargadas e instaladas.

- Es importante validar los puertos COM habilitados en sus dispositivos, ya que si estos nos se encuentran habilitados y/o activos la interconexión entre Arduino y Matlab no podría llevarse a cabo.

- Se recomienda no sobrecargar el músculo analizado con pruebas prolongadas, ya que se puede fatigar y entregar señales erróneas.

- Es importante manejar el mismo tipo de formato en las hojas de cálculo entre dispositivos, de los datos adquiridos, ya que si son diferentes, los procesamientos podrían complicarse y no lograr la normalización y comparación deseada.

4 REFERENCIAS BIBLIOGRÁFICAS

BETANCOURT O., &. G., & FRANCO B., &. F. (26 de Diciembre de 2004). RECONOCIMIENTO DE PATRONES DE MOVIMIENTO A PARTIR DE SEÑALES. *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal*, págs. 53-58.

DALCAME- Grupo de investigacion biomédica. (2005). *DALCAME*. Obtenido de DALCAME- Grupo de investigacion biomédica: https://www.dalcame.com/emg.html#.YtXwXnbMI2w

Electromiografía y electroneurografía. (2020). *Electromiografía y electroneurografía*. Obtenido de Electromiografía y electroneurografía: https://www.cun.es/enfermedades-tratamientos/pruebas-diagnosticas/electromiografia-electroneurografia#:~:text=La%20electromiograf%C3%ADa%20y%20electroneurograf%C3%ADa%20son,la%20cuant%C3%ADa%20de%20la%20conducci%C3%B3n.

Harold A. Romo, &. J. (26 de Marzo de 2007). Análisis de Señales EMG Superficiales y su . págs. 18-25.

Konrad, P. (2005). ABC of EMG. *A Practical Introduction to Kinesiological Electromyography*. Obtenido de ABC of EMG.

Martinez, A. (11 de Oct. de 2021). *Temas de Fisiología.* Obtenido de Temas de Fisiología: https://kevindotmd.wordpress.com/2021/10/11/medula-espinal/

Procesamiento de señales EMG en trastornos Neuromusculares. (2013). *Procesamiento de señales EMG en trastornos Neuromusculares.* Barcelona: Universidad Politécnica de Catalunya.

R.N. Khushaba, S. K. (15 de Sep. de 2012). Hacia un mejor control de los dedos protésicos utilizando señales de electromiograma de superficie (EMG). *Sistemas Expertos con Aplicaciones*, págs. 10731-10738. Obtenido de https://www.sciencedirect.com/science/article/pii/S0957417412004654

Rami N.Khushaba, &. S. (15 de Sep de 2012). Toward improved control of prosthetic fingers using surface electromyogram (EMG) signals. *Expert Systems with Applications*, págs. 731-738.

5 ANEXOS

ANEXO I. Programación Arduino IDE, adquisición de datos.

```
//Adquisiciòn de señales miolèctricas con shield ekg emg
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
//pinMode(4,INPUT);
// inicializa la adquisiciòn de datos
}
int m=0;
int n=0;
void loop() {
  // put your main code here, to run repeatedly:
m=analogRead(A0); //recibe el valor del puerto analògico
//n=digitalRead(4);
Serial.println(m); // imprime el valor leìdo por el puerto serial
delay (500); //toma un dato cada 1000 milisegundos
}
```

ANEXO II. Programación interfaz APP Designer.

```
classdef GUIL45 < matlab.apps.AppBase


% Properties that correspond to app components
properties (Access = public)
UIFigure matlab.ui.Figure
UITable matlab.ui.control.Table
Lamp_2Label matlab.ui.control.Label
Label_2 matlab.ui.control.Label
Lamp matlab.ui.control.Lamp
Label matlab.ui.control.Label
Exportar matlab.ui.control.Button
Parar matlab.ui.control.Button
Inicio matlab.ui.control.Button
Image matlab.ui.control.Image
FacultaddeIngenieraMecnicaLabel matlab.ui.control.Label
Salir matlab.ui.control.StateButton
AdquisicindesealesmioelctricasLabel matlab.ui.control.Label
ResetButton matlab.ui.control.Button
UIAxes matlab.ui.control.UIAxes
ContextMenu matlab.ui.container.ContextMenu
end


properties (Access = private)
S1 % Propiedad para el switch de adquisición
end
methods (Access = private)
function startupFcn(app)
% app.S1=serialport("COM3",9600,"Timeout",5);
```

```
end
end


% Callbacks that handle component events
methods (Access = private)


% Code that executes after component creation
function startupFcn2(app)
app.S1=serialport("COM3",9600,"Timeout",1);
end


% Value changed function: Salir
function SalirValueChanged(app, event)
quit
end


% Button pushed function: Inicio
function InicioPushed(app, event)
global Parar const
Parar=0;
voltaje=0;
muestras=500;
contador=1;
while contador<=muestras
if Parar == 1
app.Lamp.Color=[0.90,0.90,0.90];%gris
app.Label_2.Text='Desconectado...';
break;
else
valorADC=str2double(readline(app.S1));
const(contador) = 5;
voltaje(contador) = valorADC(1)*5/256;
plot(app.UIAxes,voltaje);
app.UIAxes.XLim=[0 contador+0.5];
app.UIAxes.YLim=[0 12];
tiempo(contador)=contador;
datos = [voltaje;tiempo];
app.UITable.Data=datos';
contador=contador+1;
app.Lamp.Color=[0.00,1.00,0.00];%verde
app.Label_2.Text='Recibiendo...';
writematrix(datos','datos6.xlsx')
end
end
end


% Button pushed function: Parar
function PararPushed(app, event)
global Parar
```

```
Parar=1;
end


% Button pushed function: Exportar
function ExportarPushed(app, event)
global datos
writematrix(datos','datos5.xlsx')
end
end


% Component initialization
methods (Access = private)


% Create UIFigure and components
function createComponents(app)


% Get the file path for locating images
pathToMLAPP = fileparts(mfilename('fullpath'));


% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 698 515];
app.UIFigure.Name = 'MATLAB App';


% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, 'Title')
xlabel(app.UIAxes, 'X')
ylabel(app.UIAxes, 'Y')
zlabel(app.UIAxes, 'Z')
app.UIAxes.Position = [12 125 492 283];


% Create ResetButton
app.ResetButton = uibutton(app.UIFigure, 'push');
app.ResetButton.Icon = fullfile(pathToMLAPP, 'Flat_restart_icon.svg.png');
app.ResetButton.FontName = 'Lucida Sans';
app.ResetButton.FontWeight = 'bold';
app.ResetButton.Position = [476 17 71 23];
app.ResetButton.Text = 'Reset';


% Create AdquisicindesealesmioelctricasLabel
app.AdquisicindesealesmioelctricasLabel = uilabel(app.UIFigure);
app.AdquisicindesealesmioelctricasLabel.FontName = 'Lucida Sans';
app.AdquisicindesealesmioelctricasLabel.FontSize = 20;
app.AdquisicindesealesmioelctricasLabel.FontWeight = 'bold';
app.AdquisicindesealesmioelctricasLabel.Position = [146 430 389 26];
```

```
app.AdquisicindesealesmioelctricasLabel.Text = 'Adquisición de señales
mioeléctricas';


% Create Salir
app.Salir = uibutton(app.UIFigure, 'state');
app.Salir.ValueChangedFcn = createCallbackFcn(app, @SalirValueChanged, true);
app.Salir.Icon = fullfile(pathToMLAPP, 'iconsalir.jpg');
app.Salir.Text = 'Salir';
app.Salir.BackgroundColor = [0.9412 0.9412 0.9412];
app.Salir.FontName = 'Lucida Sans';
app.Salir.FontWeight = 'bold';
app.Salir.Position = [581 465 94 34];


% Create FacultaddeIngenieraMecnicaLabel
app.FacultaddeIngenieraMecnicaLabel = uilabel(app.UIFigure);
app.FacultaddeIngenieraMecnicaLabel.FontName = 'Lucida Sans';
app.FacultaddeIngenieraMecnicaLabel.FontSize = 26;
app.FacultaddeIngenieraMecnicaLabel.FontWeight = 'bold';
app.FacultaddeIngenieraMecnicaLabel.Position = [121 465 440 34];
app.FacultaddeIngenieraMecnicaLabel.Text = 'Facultad de Ingeniería Mecánica';


% Create Image
app.Image = uiimage(app.UIFigure);
app.Image.Position = [1 407 111 109];
app.Image.ImageSource = fullfile(pathToMLAPP, 'Logo_EPN.png');


% Create Inicio
app.Inicio = uibutton(app.UIFigure, 'push');
app.Inicio.ButtonPushedFcn = createCallbackFcn(app, @InicioPushed, true);
app.Inicio.Icon = fullfile(pathToMLAPP, 'Green-computer-switch-icon-symbol-with-
glossy-design.png');
app.Inicio.FontName = 'Lucida Sans';
app.Inicio.FontWeight = 'bold';
app.Inicio.Position = [258 17 71 23];
app.Inicio.Text = 'Inicio';


% Create Parar
app.Parar = uibutton(app.UIFigure, 'push');
app.Parar.ButtonPushedFcn = createCallbackFcn(app, @PararPushed, true);
app.Parar.Icon = fullfile(pathToMLAPP, 'pause icon.png');
app.Parar.FontName = 'Lucida Sans';
app.Parar.FontWeight = 'bold';
app.Parar.Position = [366 17 71 23];
app.Parar.Text = 'Parar';


% Create Exportar
            app.Exportar = uibutton(app.UIFigure, 'push');
```

```
            app.Exportar.ButtonPushedFcn = createCallbackFcn(app, @ExportarPushed,
true);
            app.Exportar.Icon = fullfile(pathToMLAPP, 'png-transparent-excel-logo-
logos-logos-and-brands-icon.png');
            app.Exportar.FontName = 'Lucida Sans';
            app.Exportar.FontWeight = 'bold';
            app.Exportar.Position = [580 15 95 27];
            app.Exportar.Text = 'Exportar';


            % Create Label
            app.Label = uilabel(app.UIFigure);
            app.Label.HorizontalAlignment = 'right';
            app.Label.Position = [12 21 25 22];
            app.Label.Text = '';


            % Create Lamp
            app.Lamp = uilamp(app.UIFigure);
            app.Lamp.Position = [35 10 43 43];
            app.Lamp.Color = [0.902 0.902 0.902];


            % Create Label_2
            app.Label_2 = uilabel(app.UIFigure);
            app.Label_2.FontName = 'Lucida Sans';
            app.Label_2.Position = [111 17 98 22];
            app.Label_2.Text = '';


            % Create Lamp_2Label
            app.Lamp_2Label = uilabel(app.UIFigure);
            app.Lamp_2Label.HorizontalAlignment = 'right';
            app.Lamp_2Label.Position = [-23 61 25 22];
            app.Lamp_2Label.Text = '';


            % Create UITable
            app.UITable = uitable(app.UIFigure);
            app.UITable.ColumnName = {'Column 1'; 'Column 2'};
            app.UITable.RowName = {};
            app.UITable.Position = [513 125 172 292];


            % Create ContextMenu
            app.ContextMenu = uicontextmenu(app.UIFigure);


            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end
```

```
    % App creation and deletion
    methods (Access = public)


        % Construct app
        function app = GUIL45


            % Create UIFigure and components
            createComponents(app)


            % Register the app with App Designer
            registerApp(app, app.UIFigure)


            % Execute the startup function
            runStartupFcn(app, @startupFcn2)


            if nargout == 0
                clear app
            end
        end


        % Code that executes before app deletion
        function delete(app)


            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

ANEXO III. Manual Shield-EKG-EMG.

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/resources/SHIELD-EKG-EMG.pdf

ANEXO IV. Programación Procesamiento Datos Delsys.

```
clear,clc;
%Observación
% Cargar registros del gesto (actividad muscular en el movimiento)
datos_brutos1=xlsread(['prueba_emg_1brazo_Plot_and_Store_Rep_1.0.xlsx']);   % Se
almacena en una matriz datos_emg
datos_emg1=datos_brutos1(:,[2 4]);


%Rectificado y filtrado con RMS
wndw=100; % longitud de la banda de promedio (muestras)
y_rms = sqrt(filter(ones(1,wndw), wndw, datos_emg1(:,1).^2)); % Calcular RMS para
cada banda de muestra
```

```
hold on
plot(abs(datos_emg1(:,1)),'r')
%plot(y_rms,'b')
plot(datos_emg1(:,2),'k')

%:::REGISTRO EMG::::
%Filtro las celdas en cero para el registro EMG
posZero=find(datos_emg1(:,2)==0);
datos_emg1(posZero(1:end),:)=[];
Xrms=datos_emg1;
tiempo1=datos_brutos1(:,1);
tiempo1(posZero(1:end),:)=[];

t1=tiempo1(:,1)-tiempo1(1,1); %Desplazo el tiempo tomando en cuenta los las celdas
EMG en cero,
                                        % la aproximación de pasos de tiempo
difieren, por lo que linealizo el tiempo
n=max(size(t1)); %Halla el numero de filas
time_EMG1=t1(end)*(0:n-1)'/n; %Vuelvo a linealizar el tiempo con pasos iguales

fdamat = suavizado_bspline(datos_emg1, time_EMG1, 100, 6);
        dat=fdamat(:,:,1);
        Ddat=fdamat(:,:,2);
        D2dat=fdamat(:,:,3);


hold on
plot(abs(datos_emg1(:,1)),'r')
plot(datos_emg1(:,2),'k')
plot(dat(:,2),'b')

%Deltoides
for kn=1:2
qs=dat(:,kn);
qs=qs-mean(qs);
v=Ddat(:,kn);

% ahora hacemos un vector de índices
i=1:n;
%Sacamos los puntos en los que la velocidad es positiva. No pongo cero para
%cortar exponenciales
j=i(ne(0,diff(v>prctile(abs(v),5)))); % esto sa los índices donde la velocidad
tiene maximo o minimo
% como aquí empezamos en cero, es posible que al principio o final haya
% puntos de  velocidad cero con poco desplazamiento. Eliminamos aquellos
% cuyo fi sea inferior a un umbral, por ejemplo un tercio del p95
kk=abs(qs(j))>(prctile(abs(qs(j)),95)/6);
j=j(kk);
m=max(size(j));
k=1:m; % para tene un vector con los indices de j y así poder sacar uego cosas
% umbral de salto entre dos máximos o minimos
u=diff(prctile(qs,[25 75]))/2;

ii=sign(qs(j)')<0;      %logical([(diff(qs(j)')>u) 0]);
ii=[ii(1) diff(ii)];
iel=j(logical(ii==1));  %indices donde terminan los ciclos de subida
fel=j(logical(ii==-1)); %inidices donde empiezan los ciclos de subida
ii=sign(qs(j)')>0;    %
```

```
ii=[ii(1) diff(ii)];    %logical([(diff(qs(j)')<-u) 0]);
ide=j(logical(ii==1));  %indices donde terminan los ciclos de descenso
fde=j(logical(ii==-1)); %inidices donde empiezan los ciclos de bajada

niel=numel(iel);nide=numel(ide);nfel=numel(fel);nfde=numel(fde);
nciclos=min([niel nide nfel nfde]);

nfunc=20*nciclos; %El número de funciones lo estimamos probado multiplicar un
coeficiente por el número de ciclos

if qs(iel(1))<0
   if numel(iel)>=numel(fde)
       indicesLR=iel;
   else
       indicesLR=fde;
   end

   if numel(fel)>=numel(ide)
       indicesRL=fel;
   else
       indicesRL=ide;
   end
else
   if numel(fel)>=numel(ide)
       indicesLR=fel;
   else
       indicesLR=ide;
   end

   if numel(iel)>=numel(fde)
       indicesRL=iel;
   else
       indicesRL=fde;
   end
end
Criterio=D2dat(:,kn);
vars={'Xrms','dat','Criterio'};


clear Xcyclos_LR tt
Tel=zeros(numel(indicesLR)-1,1);
Pc=101;
nvar=numel(vars);
%Se interpola las variables segmentadas por ciclos en escalas de tiempos
%iguales para los movimientos en el orden de elevación y descenso o derecha
%a izquierda
for r=1:numel(indicesLR)-1
    tt=time_EMG1(indicesLR(r):indicesLR(r+1))-time_EMG1(indicesLR(r));
    TLR(r)=max(tt)-eps;%el eps se pone para que nunca salga por encima del valor
máximo, si no da errores en la última medida
    tn=TLR(r)*(0:(Pc-1))'/(Pc-1);
    tLR(:,:,r)=tn;
    nf=floor(max(size(tt))/5);
    for i=1:nvar
            kk=eval([vars{i},'(indicesLR(r):indicesLR(r+1),:,:)']);
            fdamat = suavizado_bspline(kk, tt, nf, 4,tn);
                if fdamat(end,1,1)==0
                   fdamat(end,:,1)=fdamat(end-1,:,1);
                   fdamat(end,:,2)=fdamat(end-1,:,2);
```

```
                fdamat(end,:,3)=fdamat(end-1,:,3);
            else
            end
        eval([vars{i}, '_LR(:,:,r) = ','fdamat(:,:,1)']);
    end
end

if numel(indicesLR)-1>2
% RL movement from right to left
% Creamoas una matriz de ceros
A=zeros(numel(indicesLR)-1,numel(indicesLR)-1);

% Calculamos la distancia entre todas las curvas
in=1:numel(indicesLR)-1;
for i=1:numel(indicesLR)-1
    for j=i+1:numel(indicesLR)-1
          A(i,j)= sum(sum((Criterio_LR(:,:,i)-Criterio_LR(:,:,j)).^2,2));%%
          A(j,i)=A(i,j);
    end
end
% Ahora elegimos la que está más alejada del resto
[~,jj]=sort(sum(A));
%La primera que hay que quitar es jj(nciclos)=j1
ja1=jj(numel(indicesLR)-1);
%Asignamos cero a la fila y la columan de A y volvemos a calcula la
%siguiente que está mas alejada de las que quedan
A(ja1,:)=0;
A(:,ja1)=0;
[~,jj]=sort(sum(A));
ja2=jj(numel(indicesLR)-1);
A(ja2,:)=0;
A(:,ja2)=0;
[~,jj]=sort(sum(A));
ja3=jj(numel(indicesLR)-1);
A(ja3,:)=0;
A(:,ja3)=0;
ja4=jj(numel(indicesLR)-1);

%estos son los indices que hay que mantener
inLR=in(((in==ja1)|(in==ja2))==0);
%inLR=in(((in==ja1)|(in==ja2)|(in==ja3))==0);

else
    inLR=1:size(Criterio_LR,3);
end

corte=0;
eval(['ciclosLR_',num2str(kn),'.indices_',' =inLR']);
eval(['ciclosLR_',num2str(kn),'.t_',' =tLR(:,:,inLR)']);


for i = 1:nvar
      eval(['ciclosLR_',num2str(kn),'.',vars{i},      ' = ',vars{i},'_LR(1:Pc-
corte,:,inLR)']);
end

for i = 1:nvar
      eval(['mediaLR_',num2str(kn),'.',vars{i}, ' =
mean(',vars{i},'_LR(1+corte:Pc,:,inLR),3)']);
```

```
      eval(['medianLR_',num2str(kn),'.',vars{i}, ' =
median(',vars{i},'_LR(1+corte:Pc,:,:),3)']);
end

eval(['medianLR_',num2str(kn),'.Reposo_ref','=min(medianLR_',num2str(kn),'.Xrms(:,
:))']);
eval(['medianLR_',num2str(kn),'.Referencia_ref','=max(medianLR_',num2str(kn),'.Xrm
s(:,:))']);
end

hold on
for i=1:3
plot(ciclosLR_2.Xrms(:,2,i))
end

%Actividad
      %Señal calibrada
       %plot(Xrms(:,1))
       EMGP(:,1)=100*(Xrms(:,1)-
medianLR_1.Reposo_ref(1))/(medianLR_1.Referencia_ref(1)-medianLR_1.Reposo_ref(1))
       EMGP(:,2)=100*(Xrms(:,2)-
medianLR_2.Reposo_ref(2))/(medianLR_2.Referencia_ref(2)-medianLR_2.Reposo_ref(2))



Xmin=min(EMGP);
EMGP=EMGP-Xmin;

 plot(EMGP(:,2))

%Normalizacion de frecuencias acumuladas APDF
P=1:99;
for i=1:2
[fi,xi]= ecdf(EMGP(:,i));% esto ya da los valores de frecuenciaacumulada fi, para
cada valor de x, pero no está en los puntos que nos interesa
p=fi*100; %se pasa a porcentajes
%ahora calculamos los valores de X en las frecuencias P
X1(:,i)=interp1(p,xi,P);
% y ahora los niveles de fondo (X(10)), mediano X(50) y máximo X(90)
PER1(i,:)=[X1(10,i); X1(50,i);X1(90,i)];
end

plot(X1(:,2),P)
save Resultado.mat X1 PER1
```

ANEXO V. Programación Procesamiento de datos Shield EKG-EMG.

```
clear,clc;
load('Resultado.mat')
%Observación
% Cargar registros del gesto (actividad muscular en el movimiento)
datos_brutos1=xlsread(['Señal_Shield_EMG_1.xlsx']);   % Se almacena en una matriz
datos_emg
datos_emg1=datos_brutos1(:,[2 4]);
%:::REGISTRO EMG::::
%Filtro las celdas en cero para el registro EMG
posZero=find(datos_emg1(:,2)==0);
datos_emg1(posZero(1:end),:)=[];
Xrms=datos_emg1;
tiempo1=datos_brutos1(:,1);
```

```
tiempo1(posZero(1:end),:)=[];

t1=tiempo1(:,1)-tiempo1(1,1); %Desplazo el tiempo tomando en cuenta los las celdas
EMG en cero,
                                          % la aproximación de pasos de tiempo
difieren, por lo que linealizo el tiempo
n=max(size(t1)); %Halla el numero de filas
time_EMG1=t1(end)*(0:n-1)'/n; %Vuelvo a linealizar el tiempo con pasos iguales

fdamat = suavizado_bspline(datos_emg1, time_EMG1, 100, 6);
        dat=fdamat(:,:,1);
        Ddat=fdamat(:,:,2);
        D2dat=fdamat(:,:,3);

%Deltoides
for kn=1:2
qs=dat(:,kn);
qs=qs-mean(qs);
v=Ddat(:,kn);

% ahora hacemos un vector de índices
i=1:n;
%Sacamos los puntos en los que la velocidad es positiva. No pongo cero para
%cortar exponenciales
j=i(ne(0,diff(v>prctile(abs(v),5)))); % esto sa los índices donde la velocidad
tiene maximo o minimo
% como aquí empezamos en cero, es posible que al principio o final haya
% puntos de  velocidad cero con poco desplazamiento. Eliminamos aquellos
% cuyo fi sea inferior a un umbral, por ejemplo un tercio del p95
kk=abs(qs(j))>(prctile(abs(qs(j)),95)/6);
j=j(kk);
m=max(size(j));
k=1:m; % para tene un vector con los indices de j y así poder sacar uego cosas
% umbral de salto entre dos máximos o minimos
u=diff(prctile(qs,[25 75]))/2;

ii=sign(qs(j)')<0;      %logical([(diff(qs(j)')>u) 0]);
ii=[ii(1) diff(ii)];
iel=j(logical(ii==1));  %indices donde terminan los ciclos de subida
fel=j(logical(ii==-1)); %inidices donde empiezan los ciclos de subida
ii=sign(qs(j)')>0;    %
ii=[ii(1) diff(ii)];    %logical([(diff(qs(j)')<-u) 0]);
ide=j(logical(ii==1));  %indices donde terminan los ciclos de descenso
fde=j(logical(ii==-1)); %inidices donde empiezan los ciclos de bajada

niel=numel(iel);nide=numel(ide);nfel=numel(fel);nfde=numel(fde);
nciclos=min([niel nide nfel nfde]);

nfunc=20*nciclos; %El número de funciones lo estimamos probado multiplicar un
coeficiente por el número de ciclos

if qs(iel(1))<0
   if numel(iel)>=numel(fde)
       indicesLR=iel;
   else
       indicesLR=fde;
   end

   if numel(fel)>=numel(ide)
```

```
        indicesRL=fel;
    else
        indicesRL=ide;
    end
else
    if numel(fel)>=numel(ide)
        indicesLR=fel;
    else
        indicesLR=ide;
    end

    if numel(iel)>=numel(fde)
        indicesRL=iel;
    else
        indicesRL=fde;
    end
end
Criterio=D2dat(:,kn);
vars={'Xrms','dat','Criterio'};


clear Xcyclos_LR tt
Tel=zeros(numel(indicesLR)-1,1);
Pc=101;
nvar=numel(vars);
%Se interpola las variables segmentadas por ciclos en escalas de tiempos
%iguales para los movimientos en el orden de elevación y descenso o derecha
%a izquierda
for r=1:numel(indicesLR)-1
    tt=time_EMG1(indicesLR(r):indicesLR(r+1))-time_EMG1(indicesLR(r));
    TLR(r)=max(tt)-eps;%el eps se pone para que nunca salga por encima del valor
máximo, si no da errores en la última medida
    tn=TLR(r)*(0:(Pc-1))'/(Pc-1);
    tLR(:,:,r)=tn;
    nf=floor(max(size(tt))/5);
    for i=1:nvar
            kk=eval([vars{i},'(indicesLR(r):indicesLR(r+1),:,:)']);
            fdamat = suavizado_bspline(kk, tt, nf, 4,tn);
                if fdamat(end,1,1)==0
                   fdamat(end,:,1)=fdamat(end-1,:,1);
                   fdamat(end,:,2)=fdamat(end-1,:,2);
                   fdamat(end,:,3)=fdamat(end-1,:,3);
                else
                end
            eval([vars{i}, '_LR(:,:,r) = ','fdamat(:,:,1)']);
    end
end

if numel(indicesLR)-1>2
% RL movement from right to left
% Creamoas una matriz de ceros
A=zeros(numel(indicesLR)-1,numel(indicesLR)-1);

% Calculamos la distancia entre todas las curvas
in=1:numel(indicesLR)-1;
for i=1:numel(indicesLR)-1
    for j=i+1:numel(indicesLR)-1
          A(i,j)= nansum(nansum((Criterio_LR(:,:,i)-Criterio_LR(:,:,j)).^2,2));
          A(j,i)=A(i,j);
```

```
    end
end
% Ahora elegimos la que está más alejada del resto
[~,jj]=sort(sum(A));
%La primera que hay que quitar es jj(nciclos)=j1
ja1=jj(numel(indicesLR)-1);
%Asignamos cero a la fila y la columan de A y volvemos a calcula la
%siguiente que está mas alejada de las que quedan
A(ja1,:)=0;
A(:,ja1)=0;
[~,jj]=sort(sum(A));
ja2=jj(numel(indicesLR)-1);
A(ja2,:)=0;
A(:,ja2)=0;
[~,jj]=sort(sum(A));
ja3=jj(numel(indicesLR)-1);
A(ja3,:)=0;
A(:,ja3)=0;
ja4=jj(numel(indicesLR)-1);

%estos son los indices que hay que mantener
inLR=in(((in==ja1)|(in==ja2))==0);
%inLR=in(((in==ja1)|(in==ja2)|(in==ja3))==0);

else
    inLR=1:size(Criterio_LR,3);
end

corte=0;
eval(['ciclosLR_',num2str(kn),'.indices_',' =inLR']);
eval(['ciclosLR_',num2str(kn),'.t_',' =tLR(:,:,inLR)']);


for i = 1:nvar
      eval(['ciclosLR_',num2str(kn),'.',vars{i},      ' = ',vars{i},'_LR(1:Pc-
corte,:,inLR)']);
end

for i = 1:nvar
      eval(['mediaLR_',num2str(kn),'.',vars{i}, ' =
mean(',vars{i},'_LR(1+corte:Pc,:,inLR),3)']);
      eval(['medianLR_',num2str(kn),'.',vars{i}, ' =
median(',vars{i},'_LR(1+corte:Pc,:,:),3)']);
end

eval(['medianLR_',num2str(kn),'.Reposo_ref','=min(medianLR_',num2str(kn),'.Xrms(:,
:))']);
eval(['medianLR_',num2str(kn),'.Referencia_ref','=max(medianLR_',num2str(kn),'.Xrm
s(:,:))']);
end


%Actividad
      %Señal calibrada
       %plot(Xrms(:,1))
       EMGP(:,1)=100*(Xrms(:,1)-
medianLR_1.Reposo_ref(1))/(medianLR_1.Referencia_ref(1)-medianLR_1.Reposo_ref(1))
       EMGP(:,2)=100*(Xrms(:,2)-
medianLR_2.Reposo_ref(2))/(medianLR_2.Referencia_ref(2)-medianLR_2.Reposo_ref(2))
```

```
Xmin=nanmin(EMGP);
EMGP=EMGP-Xmin;



%Normalizacion de frecuencias acumuladas APDF
P=1:99;
for i=1:2
[fi,xi]= ecdf(EMGP(:,i));% esto ya da los valores de frecuenciaacumulada fi, para
cada valor de x, pero no está en los puntos que nos interesa
p=fi*95; %se pasa a porcentajes
%ahora calculamos los valores de X en las frecuencias P
X2(:,i)=interp1(p,xi,P);
% y ahora los niveles de fondo (X(10)), mediano X(50) y máximo X(90)
PER2(i,:)=[X2(10,i); X2(50,i);X2(90,i)];
end

save Resultado.mat X1 PER1 X2 PER2

P=1:99;
hold on
plot(X1(:,2),P)
plot(X2(:,2),P)

Xc(:,:,1)=X1(:,2);
Xc(:,:,2)=X2(:,2);
%Fiabilidad-reproducibilidad
%Posicion angular-variabilidad
[Max Min Rango Media Mediana SEM DesvStan Varianza Sig_F F_test Sig_Q Q_test
Percentil IQR fxi xi Icc CMC CMCm cmc Pearson]=ICorre_sesiones(3,'single',Xc)

%Media(DesvStan)
%SEM
%Mediana(IQR)

%Icc,cmc,Pearson
```