

# Interfaz para análisis y procesamiento de señales EMG (Electromiográficas) utilizando MATLAB.

## Interface for analysis and processing of EMG (Electromyographic) signals using MATLAB.

Luis Manuel López Jiménez (1).

Estudiante de la Facultad de Sistemas Biológicos e Innovación Tecnológica / Universidad Autónoma del Estado de Oaxaca.

[lslopez.alu@uabjo.mx](mailto:lslopez.alu@uabjo.mx).

Ximena Denisse Pinedo Ramírez\* (2). Estudiante de la Facultad de Sistemas Biológicos e Innovación Tecnológica / Universidad Autónoma del Estado de Oaxaca. [denissepinedo21@gmail.com](mailto:denissepinedo21@gmail.com).

\*corresponding author.

Artículo recibido en septiembre 03, 2024; aceptado en octubre 25, 2024.

### Resumen.

*En los últimos años, los avances en bioingeniería han permitido resolver problemas que afectan al ser humano desde hace mucho tiempo, como la detección de señales corporales como EMG, EEG y ECG. Estas señales se utilizan en diversas áreas, como el análisis de emociones y el control de prótesis robóticas. El descubrimiento y aprovechamiento de estas señales ha mejorado la calidad de vida, permitiendo entender mejor el funcionamiento del cuerpo en distintas situaciones. En el caso de las señales EMG, se encargan de enviar información sobre la actividad muscular, registrando los impulsos eléctricos que genera el cuerpo al usar un músculo. Estas señales son de gran importancia en campos como el reconocimiento de patrones, control de dispositivos, rehabilitación de extremidades y prótesis biomédicas. Además, su obtención es sencilla mediante métodos no invasivos, lo que elimina la necesidad de perforar la piel. Actualmente, existen sensores que permiten obtener estas señales tanto por métodos no invasivos como invasivos. Los sensores de electrodos son ejemplos de métodos no invasivos. Este trabajo presenta una interfaz para el análisis y procesamiento de señales electromiográficas, las cuales son obtenidas mediante un sensor no invasivo conectado a Arduino. La interfaz, desarrollada en Matlab, permite analizar los datos obtenidos utilizando diversas fórmulas, facilitando el estudio y la interpretación de las señales EMG para aplicaciones técnicas.*

**Palabras claves:** Análisis, EMG, Matlab, sensores.

### Abstract.

*In recent years, advances in bioengineering have enabled the resolution of problems that have affected humans for a long time, such as the detection of bodily signals like EMG, EEG, and ECG. These signals are used in various fields, such as emotion analysis and the control of robotic prostheses. The discovery and utilization of these signals have improved the quality of life, allowing a better understanding of how the body functions. In the case of EMG signals, they are responsible for sending information about muscle activity, recording the electrical impulses that the body generates when using a muscle. These signals are of great importance in fields such as pattern recognition, device control, limb rehabilitation and biomedical prostheses. In addition, they are easily obtained using non-invasive methods, which eliminates the need to pierce the skin. Currently, there are sensors available to obtain these signals through both non-invasive and invasive methods. Electrode sensors are examples of non-invasive methods. This paper*

*presents an interface for analyzing and processing electromyographic signals, which are obtained using a non-invasive sensor connected to Arduino. The interface, developed in Matlab, allows for the analysis of the data using various formulas, facilitating the study and interpretation of EMG signals for technical applications.*

**Keywords:** Analysis, EMG, Matlab, sensors.

## 1. Introducción.

La interacción entre el humano y las máquinas, cada vez se va mejorando, en ciencias como la medicina, la electrónica ayuda a obtener mejores resultados respondiendo en forma oportuna a situaciones donde se requiere controles críticos de forma automática (Zalva, 2018). La electromiografía es un conjunto de procedimientos en los cuales el fin es determinar enfermedades neuromusculares en personas, para lo cual es necesario tomar, almacenar y procesar medidas en la forma de señales eléctricas de la actividad muscular del individuo (Zalva, 2017).

El estudio de la electromiografía se remonta a mediados del siglo XVII, cuando el doctor y científico Francisco Redi, demostró la existencia de un musculo capaz de generar electricidad en el pez raya. Investigación a partir de la cual permitió el posterior descubrimiento de señales eléctricas en el tejido humano (Ávila, 2019). En la actualidad la electromiografía se emplea con mayor frecuencia en procesos de rehabilitación y fisioterapia debido a la información que esta otorga sobre el funcionamiento del estado fisiológico de los músculos analizados (Quizhpe, 2016).

Se ha dado gran importancia al desarrollo de dispositivos controlados mediante señales EMG, diseñados especialmente para la ayuda a personas con alguna discapacidad física, entrenamiento muscular o para la ejecución de tareas sin contacto directo por parte del usuario [Caluguillin, 2019], la señal electromiográfica del músculo activo es una de las señales más eficaces para el control de las prótesis de las extremidades superiores con alimentación externa (García, 2014), las posibilidades de aplicación y el rendimiento de la electromiografía han evolucionado paralelamente al conocimiento de las propiedades de la energía eléctrica y al desarrollo de la tecnología eléctrica y electrónica (Gila, 2009), Actualmente la tecnología ha permitido el desarrollo de dispositivos y equipos inteligentes, mejorando en cierta forma el modelo de control, pasando por sistemas de control manual y electrónico, a implementar sistemas inteligentes mediante un controlador inmerso en las personas (Figueroa, 2016).

En el presente trabajo, se muestra un sistema para el adquisición, análisis y procesamiento de señales EMG, para identificar lo diferentes movimientos realizados por el brazo humano, como lo son flexión, extensión o relajación, realizado por medio de la aplicación MATLAB, para la configuración del programa encargado de realizar las ecuaciones necesarias para dicho análisis. El sistema lee las señales bioeléctricas del musculo del antebrazo con el fin de analizar los diferentes estados de este y posteriormente mostrarlos por medio de una gráfica en la interfaz, realizada con herramientas de MATLAB.

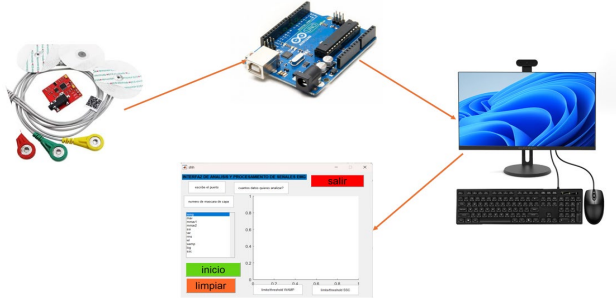
El presente trabajo se enfoca en la programación de la interfaz de Matlab, de las distintas fórmulas las cuales permiten el análisis y procesamiento de las señales electromiograficas obtenidas, esto con el fin de presentar una alternativa más intuitiva al observar por medio de graficas el comportamiento de los músculos al realizar distintas acciones, buscando una mayor facilidad para su aprovechamiento en aplicaciones que así lo requieran.

## 2. Métodos.

El análisis de señales electromiograficas ayuda a los investigadores a observar de qué manera se comporta el músculo en el cuerpo de una persona de una manera constante y no invasiva, al mismo tiempo que de acuerdo con el método de procesamiento dichas señales pueden ser aprovechadas para su uso en diversas aplicaciones.

La interfaz presentada realiza una obtención de datos eléctricos musculares, es decir, señales electromiograficas mediante electrodos conectados al sensor AD8232, los cuales detectan los impulsos eléctricos del musculo en sus distintos estados de relajación y contracción. Posteriormente las mediciones obtenidas, son enviadas al

microcontrolador Arduino UNO, el cual por medio del puerto COM6 de la computadora los envía a MATLAB, donde son analizadas, por medio de distintas ecuaciones, lo que permite observar mediante gráficas el comportamiento del músculo en el tiempo de recolección, siendo esta conexión entre las piezas el arreglo experimental utilizado, observable en la figura 1.



**Figura 1.** Arreglo experimental.

Las ecuaciones utilizadas son: Integrated EMG (IEMG) (ecuación 1), Mean Absolute Value (MAV) (ecuación 2), Modified Mean Absolute Value 1 (MMAV1) (ecuación 3), Modified Mean Absolute Value 2 (MMAV2) (ecuación 4), Simple Square Integral (SSI) (ecuación 5), Variance of EMG (VAR) (ecuación 6), Root Mean Square (RMS) (ecuación 7), Waveform Length (WL) (ecuación 8), Wilson Amplitude (WAMP) (ecuación 9) Log Detector (LOG) (ecuación 10) y Slope Sign Change (SSC) (ecuación 11).

$$IEMG = \sum_{n=1}^N |x_n| \quad \dots\dots\dots \text{Ecuación 1.}$$

Donde  $N$  denota la longitud de la señal y  $X$  representa la señal EMG en un segmento  $n$ .

$$MAV = \frac{1}{N} \sum_{n=1}^N |x_n| \quad \dots\dots\dots \text{Ecuación 2.}$$

$$MMAV1 = \frac{1}{N} \sum_{n=1}^N W_n |x_n| \quad \dots\dots\dots \text{Ecuación 3.}$$

Donde  $W_n$  tiene un valor de 1 si  $0.25N < n < 0.75N$ , con un valor de 0.5 en caso contrario.

$$MMAV2 = \frac{1}{N} \sum_{n=1}^N W_n |x_n| \quad \dots\dots\dots \text{Ecuación 4.}$$

Donde  $W_n$  tiene un valor de 1 si  $0.25N < n < 0.75N$ , un valor de  $\frac{4n}{n}$  si  $n < 0.25N$  y  $\frac{4(n-N)}{n}$  si  $0.75N < n$

$$SSI = \sum_{n=1}^N |x_n|^2 \quad \dots\dots\dots \text{Ecuación 5.}$$

$$VAR = \frac{1}{N-1} \sum_{n=1}^N x_n^2 \quad \dots\dots\dots \text{Ecuación 6.}$$

$$RMS = \sqrt{\frac{1}{N} \sum_{n=1}^N x_n^2} \quad \dots\dots\dots \text{Ecuación 7.}$$

$$WL = \sum_{n=1}^N |x_{n+1} - x_n| \quad \dots\dots\dots \text{Ecuación 8.}$$

$$WAMP = \sum_{n=1}^N f |x_{n+1} - x_n| \quad \dots\dots\dots \text{Ecuación 9.}$$

Donde  $f(x)=1$  si  $x >$  el límite establecido, siendo 0 de lo contrario.

$$LOG = e^{\frac{1}{N} \sum_{n=1}^N \log|x_n|} \dots\dots\dots \text{Ecuación 10.}$$

$$SSC = \sum_{n=2}^N f|(x_n - x_{n-1})(x_n - x_{n+1})| \dots\dots\dots \text{Ecuación 9.}$$

Donde  $f(x)=1$  si  $x >$  el límite establecido, siendo 0 de lo contrario.

### 3. Desarrollo.

#### Análisis del hardware.

Para armar el receptor de señales se utilizó el microcontrolador Arduino UNO el cual es un microcontrolador basado en el ATmega328P, el cual dispone de 14 pines de entradas/salidas digitales, 6 entradas analógicas, conexión USB, conector de alimentación, así como un botón de reinicio; con facilidad para su alimentación, ya sea por medio de conexión a una computadora por medio de USB, o bien utilizando baterías.

Por parte del sensor, se utilizó el sensor AD8232 el cual esta diseñado para extraer, amplificar y filtrar pequeñas señales biopotenciales en presencia de condiciones ruidosas, como las creadas por el movimiento o la colocación remota de electrodos. Este diseño permite que un convertidor de analógico a digital (ADC) de ultra baja potencia o un microcontrolador integrado adquieran la señal de salida fácilmente, en este caso Arduino.

El sensor AD8232 fue el encargado de obtener las señales provenientes del músculo, en este caso del antebrazo, por medio de electrodos, y luego enviar estas señales a Arduino.

Como se observa en la figura 2 se utilizan 5 pines tanto del sensor, como de Arduino para la conexión entre estos; dichos pines se conectan entre si de la siguiente manera: para la alimentación, el pin de 3.3 v se conectó a la salida de 3.3v del Arduino, en segundo lugar, los GND de ambos (sensor y Arduino), el pin OUTPUT es la salida analógica del sensor por lo que se conectó a una entrada analógica de Arduino, y finalmente los pines LO- y LO+ que se conectan a dos pines digitales del Arduino Uno.

El sensor detecta las señales EMG por medio de electrodos, los cuales dependiendo de la utilidad que se le desee dar se pueden colocar en distintas partes de cuerpo, principalmente en antebrazos, piernas y pecho.

En el presente trabajo, los electrodos encargados del monitoreo de las señales EMG fueron colocados en la zona del antebrazo (figura 2), con el fin de observar la actividad muscular continuamente, para posteriormente analizar cómo cambian las señales recibidas del músculo ante un estado de reposo, movimiento o tensión.



Figura 2. Colocación de electrodos.

### Análisis del software.

Para el análisis y procesamiento de las señales obtenidas, se utilizó programación en Matlab y Arduino, esto con el fin de obtener un programa diseñado desde cero, en el cual se contengan diversas fórmulas, en el caso de Arduino, el programa se cargo en el microcontrolador y su única función era la de continuamente recibir la señal enviada desde el sensor, por otra parte el programa de Matlab era la parte principal de la interfaz, pues era el encargado de recibir la señal enviada desde Arduino, preguntar la cantidad total datos se deseaban analizar y cada cuantos realizar dicho análisis, posteriormente realizar los cálculos necesarios y mostrarlos en forma de gráfica.

La interfaz visual, es decir sin funciones agregadas, fue realizada utilizando la herramienta “guide” disponible en Matlab, donde se colocaron botones, grafica, espacios de entrada, lista de métodos de análisis y texto estático con el fin de obtener un funcionamiento más intuitivo al momento de utilizar dicha interfaz para obtener y analizar los datos, una vez terminada, viéndose como en la figura 3, Matlab genera su respectivo código, sobre el cual en cada botón, entrada de texto y grafica se escribió la programación necesaria para su funcionamiento.

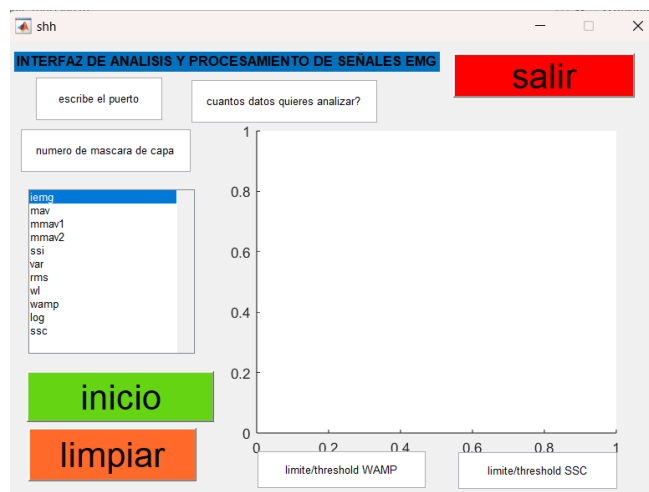


Figura 3. Interfaz realizada utilizando *guide*.

En las entradas de texto se escribe el puerto COM al cual está conectado Arduino UNO, en el siguiente el número de datos total a analizar y finalmente cuantos datos utilizara para realizar el análisis, es decir la máscara de capa, en el programa esto se realizó mediante una obtención de lo escrito mediante una variable de tipo *String*, posteriormente se transformó en una de tipo numérico. esto en el caso del número de datos, así como en la máscara de capa mediante el código 1.

---

#### Código 1. Obtención de datos escritos en la interfaz.

---

```
i=get (hObject, 'String');
global data
data=i;
disp(i);
```

---

De la misma manera, los límites WAMP y SSC, se obtuvieron utilizando el mismo código, estos fueron utilizados para el análisis de los datos, por medio de las ecuaciones con el mismo nombre/abreviatura.

Para los botones salir, inicio y limpiar se buscó una función de salir de la interfaz, iniciar la recopilación de los datos y eliminar los datos obtenidos hasta el momento para iniciar un nuevo análisis, esto se logró mediante los códigos 2, 3 y 4 respectivamente.

---

**Código 2. Botón de salir.**

---

```
% --- Executes on button press in salir.
function salir_Callback(hObject, eventdata, handles)

close(gcf);
```

---



---

**Código 3. Botón de inicio.**

---

```
function inicio_Callback(hObject, eventdata, handles)
    global x
    %global puert
    %puerto=puert;
    global data
    global masc
    global dat
    global nummasc
    s = serial('COM6', 'BaudRate', 9600); % Establecer el puerto serial
    fopen(s); % Abrir el puerto serial
    % Leer y mostrar los datos de la señal EMG en la ventana de comandos
    dat=str2double(data);
    disp(dat);
    nummasc=str2double(masc);
    for i = 1:dat % Suponiendo que quieres mostrar n muestras
        emgData = str2double(fgetl(s)); % Leer dato de la señal EMG desde Arduino
        %disp(emgData); % Mostrar el dato en la ventana de comandos
        x(i)=emgData;
    end
    fclose(s); % Cerrar el puerto serial
    delete(s); % Eliminar el objeto del puerto serial

    fprintf (' %i ', x);
    ii = 1:dat
    plot(handles.axes1, ii,x);
```

---



---

**Código 4. Botón de limpiar.**

---

```
% --- Executes on button press in limpiar.
function limpiar_Callback(hObject, eventdata, handles)

plot(0,0);
global x
x=0;
```

---

Para el análisis de las señales obtenidas, se utilizaron una serie de ecuaciones, presentes en la lista del lado izquierdo de la interfaz, las ecuaciones anteriormente presentadas, fueron integradas al programa utilizando el código 5.

---

**Código 5. Programación de la primera ecuación.**

---

```
function listbox1_Callback(hObject, eventdata, handles)
I=get (hObject, 'value'); %recuperar el valor seleccionado en la listbox
```

```

disp(I);%mostrar en command window
    global nummasc
    global dat
    global x
    global lwamp
    global lssc
if I==1
    n=dat;
    num=x;
    fprintf (' %i ',num);
    %calcular las máscaras completas
    mit=(nummasc+1)/2;
    ii=0;
    e=nummasc-1;
    for i=mit:(n-(mit-1))
        ii=ii+1;
        e=e+1;
        sum=0;
        for g=ii:e
            sum=sum+num(g);
        end
        vab(i)= abs (sum);
    end
    %calcular las máscaras inferiores
    fin=mit-1;
    suma=0;
    for y=1:mit-1
        fin=fin+1;
        for z=1:fin
            suma=suma+num(z);
        end
        vab(y)= abs(suma);
        suma=0;
    end
    %calcular las máscaras superiores
    if nummasc==3
        fina=n-2;
    else
        rest=nummasc-2;
        fina=n-rest;
    end
    sumaa=0;

    for y=n-(mit-2):n
        fina=fina+1;
        for t=fina:n
            sumaa=sumaa+num(t);
        end
        vab(y)= abs(sumaa);
        sumaa=0;
    end
    fprintf ('\n');
    fprintf (' %i ',vab);
    i=1:n;
    plot(handles.axes1, i,vab);
end

```

El código 5, presenta la programación necesaria para realizar la ecuación 1, el cual fue tomado como base, para que por medio de modificaciones fuera utilizado en el resto de las ecuaciones, por ejemplo, para la ecuación 2 se agrego

una multiplicación por  $\frac{1}{N}$ , como se puede observar en el código 6 o en la ecuación 5, donde se multiplico al cuadrado observable en el código 7.

---

**Código 6.** Programación de la segunda ecuación.

---

```

if I==2
    n=dat;
    num=x;
%calcular las máscaras completas
mit=(nummasc+1)/2;
ii=0;
e=nummasc-1;
for i=mit:(n-(mit-1))
    ii=ii+1;
    e=e+1;
    sum=0;
    for g=ii:e
        sum=sum+num(g);
    end
    vab(i)= abs (sum);

end

fin=mit-1;
suma=0;
for y=1:mit-1
    fin=fin+1;
    for z=1:fin
        suma=suma+num(z);
    end
    vab(y)= abs (suma);
    suma=0;
end
if nummasc==3
    fina=n-2;
else
    rest=nummasc-2;
    fina=n-rest;
end
sumaa=0;

for y=n-(mit-2):n
    fina=fina+1;
    for t=fina:n
        sumaa=sumaa+num(t);
    end
    vab(y)= abs (sumaa);
    sumaa=0;
end
vabb=vab/n;
fprintf ('\n');
fprintf (' %f ',vabb);
i=1:n;
plot(handles.axes1, i,vabb);
end

```

---



**Código 5.** Programación de la quinta ecuación.

---

```

if I==5
    n=dat;
    rec=x;
    fprintf (' %i ',num);
    num=rec.^2;
    %calcular las máscaras completas
    mit=(nummasc+1)/2;
    ii=0;
    e=nummasc-1;
    for i=mit:(n-(mit-1))
        ii=ii+1;
        e=e+1;
        sum=0;
        for g=ii:e
            sum=sum+num(g);
        end
        vab(i)= abs (sum);
    end

    %calcular las máscaras inferiores
    fin=mit-1;
    suma=0;
    for y=1:mit-1
        fin=fin+1;
        for z=1:fin
            suma=suma+num(z);
        end
        vab(y)= abs (suma);
        suma=0;
    end

    %calcular las máscaras superiores
    if nummasc==3
        fina=n-2;
    else
        rest=nummasc-2;
        fina=n-rest;
    end
    end
    sumaa=0;

    for y=n-(mit-2):n
        fina=fina+1;
        for t=fina:n
            sumaa=sumaa+num(t);
        end
        vab(y)= abs (sumaa);
        sumaa=0;
    end
    fprintf ('\n');
    fprintf (' %i ',vab);

    i=1:n;

    plot(handles.axes1, i,vab);
end

```

---

## Resultados.

Una vez terminada la programación de las ecuaciones, así como todas las partes de la interfaz, se procedió con las pruebas en distintas personas, así como en distintas cantidades de datos, esto con el fin de observar una correcta adquisición y recepción de las señales.

Se llevaron a cabo pruebas en distintas personas, así como un distinto numero de datos, en los cuales la recepción de estos por medio del sensor, y su posterior análisis por medio del programa, fue satisfactoria en el 90% de los casos, resultados los cuales se presentan por medio de la tabla 1.

**Tabla 1.** Resultados obtenidos.

Número	Edad	No. De datos	Recepción
1	20	50	Satisfactoria
2	22	100	Satisfactoria
3	21	100	Aceptable
4	22	150	Satisfactoria
5	18	50	Satisfactoria
6	45	200	Satisfactoria
7	35	200	Satisfactoria
8	19	200	Satisfactoria
9	89	80	Deficiente
10	65	100	Aceptable

Se analizaron hasta 200 datos diferentes, donde fue posible observar cómo después de un tiempo realizando cierta acción, en el caso de la figura 4 tensar el musculo, se presenta un estado de fatiga, al mismo tiempo que fue posible observar el cambio de un estado de reposo, a un estado de tensión en el músculo.



**Figura 4.** Fatiga en el músculo.

En la gráfica se pueden observar 3 picos, momento en el cual se llevo a cabo la tensión del músculo, sin embargo, el segundo y tercer pico son significativamente mas pequeños que el primero, esto debido al tiempo que se mantuvo tenso el músculo lo que causo fatiga y por lo tanto una menor tensión al repetir el movimiento.

En la figura 5, se puede observar el análisis realizado a 200 datos mediante la ecuación 8, mientras que en la figura 6, se puede observar los mismos 200 datos, sin embargo, esta vez analizados mediante la ecuación 6. Esta diferencia presentada en ambas ecuaciones con los mismos datos nos permite utilizarlos para funciones diferentes entre sí.

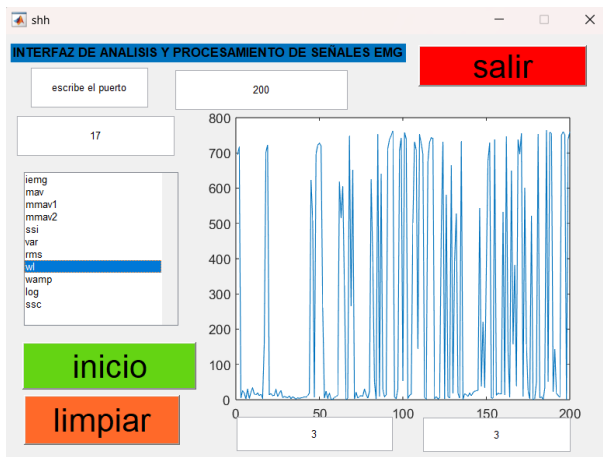


Figura 5. Análisis con WL.



Figura 6. Análisis con VAR.

En el caso del análisis de la figura 6, por medio de los picos tan pronunciados, es posible utilizarlos para la realización de un controlador de tipo todo o nada, para una pinza, pues al colocar un límite superior e inferior, los picos y valles donde se presenta tensión y relajación permitirían abrir y cerrar dicha pinza respectivamente.

Por otro lado, el análisis de la figura 7, permitiría una aplicación en prótesis, pues al ser un movimiento más limpio, y estable, brinda una mayor variedad en los datos obtenidos, lo cual, a su vez, permite una configuración de prótesis más avanzada, dando pie a una diferencia en las tensiones aplicadas por esta mismas, permitiendo una aplicación de fuerza variable de acuerdo con la situación requerida.

## Conclusiones.

Con la interfaz de análisis y procesamiento de señales EMG se permite observar el comportamiento de los músculos en sus distintos estados y limpiar la señal de acuerdo con las necesidades presentadas al momento de la obtención, facilita el análisis de estas señales para su uso en diversos dispositivos, como lo son brazos robóticos ya que muestra de una forma más interactiva y fácil de comprender las diversas fórmulas utilizadas en el análisis de este tipo de señales.

Con el trabajo presentado es posible facilitar el entendimiento de las señales recibidas, mediante la graficación de las mismas, y de esta forma hacer más intuitivo su uso. Los resultados obtenidos, demuestran la posibilidad de un análisis de varios datos a la vez, así como su análisis mediante las ecuaciones aquí presentadas sin la necesidad de tomar datos nuevos, permitiendo un análisis múltiple de un mismo conjunto de datos, al mismo tiempo que de ser necesario permite la obtención de un nuevo conjunto de datos sin la necesidad de reiniciar el programa.

Las pruebas realizadas en distintas personas, así como en un distinto número de datos, fue satisfactoria en el 90% de los casos, siendo la excepción de esto, en una persona de edad avanzada, donde el sensor no fue capaz de detectar el 100% de los datos esperados.

Se está trabajando en el control de dirección de un robot omnidireccional por medio de las señales recibidas, lo cual permite un control a distancia, se espera la mejora de la recepción de los datos y su aplicación final como controlador.

## Créditos.

Los autores agradecen a la Facultad de Sistemas Biológicos e Innovación Tecnológica de la Universidad Autónoma Benito Juárez de Oaxaca por el apoyo al momento de la adquisición de conocimientos necesarios para la realización del presente proyecto.

### Referencias bibliográficas.

- Ávila, J., Fajardo F. (2019).** *Análisis ente señales electromiográficas en los músculos cervicales y el ángulo de inclinación de la cabeza en niños con PCI entre 5 y 10 años aplicada a terapia cervical* (Tesis de licenciatura). escuela superior politécnica de chimborazo, Ecuador.
- Calaguillín, H. (2019).** *análisis de características de señales electromiográficas para la determinación de movimientos de una mano.* (Tesis de Licenciatura). universidad técnica del Norte, Ecuador
- Figuroa, C. (2016).** *Sistema de adquisición de señales EMG para la detección de fatiga muscular,* iB Ingeniería Biomédica, vol. 37, pp. 17-27
- García, J., Enrique, L. (2014).** *adquisición y procesamiento de señales emg para controlar movimiento de un brazo hidraulico.* Mundo FESC, 4(7), 49–60.
- Gila, L., Malanda, A., Rodriguez, I., Rodriguez, J., & Navallas, J. (2009).** *Métodos de procesamiento y análisis de señales electromiográficas.* Suplemento, Vol. 32, 27–43.
- Quizhpe, C., Ortiz, F. (2016).** *Valoración del movimiento en el área cervical mediante el análisis de señales electromiográficas en pacientes con parálisis.* Índice. Revista de estadística y sociedad, 68 (9–11). ISSN: 1697-2325. *parálisis cerebral infantil.* (PCI), Cuenca.
- Zalva, L. (2017).** *Desarrollo de un sistema de adquisición y procesamiento de señales electromiográficas con Matlab para la identificación de los movimientos extensión, flexión, pronación y supinación de un brazo humano.* (Tesis de maestría). escuela superior politécnica de Chimborazo, Ecuador
- Zalva, L., Vera, J., Nuñez, C., & Guaño, A. (2018).** *Análisis e identificación de los movimientos de un brazo humano utilizando un sistema de adquisición y procesamiento de señales electromiográficas.* Espirales revista multidisciplinaria de investigación, ISSN: 2550-6862.

### Información de los autores.



**Luis Manuel López Jiménez,** Luis nació el 24 de noviembre del 2002 en Oaxaca de Juárez, Oaxaca, Mex.. Estudiante de octavo semestre de la licenciatura en ingeniería en innovación tecnológica por la Universidad Autónoma Benito Juárez de Oaxaca. Apasionado por la investigación y la obtención de nuevos conocimientos, especialmente en el área de usar la tecnología en conjunto con el cuerpo humano. Anteriormente ha trabajado en proyectos en el laboratorio de Robótica de la Universidad de Colima, trabajo que fue presentado en el congreso internacional del programa Delfin en Puerto Vallarta y publicado en un capítulo del libro “Kaizen y mecatrónica”; y participado en una estancia de investigación en el laboratorio de microelectrónica del Instituto Nacional de Astrofísica, Óptica y Electrónica (CONAHCYT) donde trabajo en el uso de memristores. Le emociona compartir experiencias y habilidades que adquirió a lo largo de su carrera.



**Ximena Denisse Pinelo Ramírez**, Ximena nació el 21 de noviembre del 2005 en Oaxaca de Juárez, Oaxaca, Mex., Es estudiante del tercer semestre de la Licenciatura en Ingeniería e Innovación Tecnológica por la Universidad Autónoma Benito Juárez de Oaxaca. A raíz de su participación en el programa Mujeres líderes en STEAM, donde colaboró como embajadora, se dedicó a la divulgación científica, Presento ponencias acerca de la biotecnología, tras lo cual decidió formalizar su interés estudiando la licenciatura antes mencionada. Durante su época estudiando ha presentado platicas de robótica e inteligencia artificial, así como talleres de ciencia y tecnología a estudiantes de educación básica y media superior, demostrando entusiasmo por compartir lo importante que es el desarrollo tecnológico para el futuro del mundo.