

Control de señales EMG y ECG



Manuel Medina Parra

I.E.S Javier García Téllez



En colaboración con:



ÍNDICE

1. Objeto y Alcance	3
2. Alternativa Profesional (con aparato y programa diseñado para ello)	3
3. Descripción	6
3.1. EMG (Electromiografía)	6
3.2. ECG (Electrocardiografía)	9
4. Elementos utilizados	11
4.1. Hardware	11
(https://www.sparkfun.com/products/12650)	13
4.2. Software	13
5. Programación y Configuración de Equipos	14
5.1. (Arduino) - Señal EMG	14
5.2. (Arduino) - Señal ECG	15
5.3. (Matlab) - Señal EMG	16
5.4. (Matlab) - Señal ECG	26
6. Planificación	28
7. Presupuesto	28
7.1. Mediciones	28
7.2. Cuadro de Precios	29
7.3. Presupuesto	29
7.3. Resumen	30
8. Metodología	31
8.1. FlexDex	31
8.2. SensOR	32
9. Planos	35
10. Bibliografía y Webgrafía	35
10.1. Bibliografía	35
10.2. Webgrafía	36
11. Conclusiones	36
12. Anexos	37

1. Objeto y Alcance

Este proyecto se basa en realizar un electromiograma a través del sensor MyoWare Muscle Sensor conectado a un arduino. A partir de la información que nos vuelque la placa, se realizará el acondicionamiento de la señal, para eliminar las interferencias y ruidos producidos por diversos agentes exógenos al aparato como por ejemplo:

- Las frecuencias de 50Hz(Europa) y 60Hz(EEUU)
- las señales que provienen de músculos cercanos, al músculo monitorizado
- La Frecuencia cardiaca del sujeto de pruebas
- Halógenos y otros aparatos, que produzcan ondas electromagnéticas

La gráfica resultante nos servirá para saber cuánto esfuerzo realiza el músculo en un determinado ejercicio. Esto nos puede ayudar a saber si el cirujano está cómodo en la operación, si se le cargan los músculos, si su postura es la adecuada, etc. Estos detalles pueden favorecer al adecuado desempeño quirúrgico del cirujano.

Cuando el electromiograma esté funcionando correctamente, procederemos a la utilización de la señal ECG (electrocardiograma) de la placa AD8232 de MyoWare, que ya irá filtrada. Con esta señal, podremos ver su actividad cardiaca durante las pruebas, además la señal nos arroja información importante junto con la señal EMG, para conocer el estado fisiológico de la persona durante la acción de la prueba.

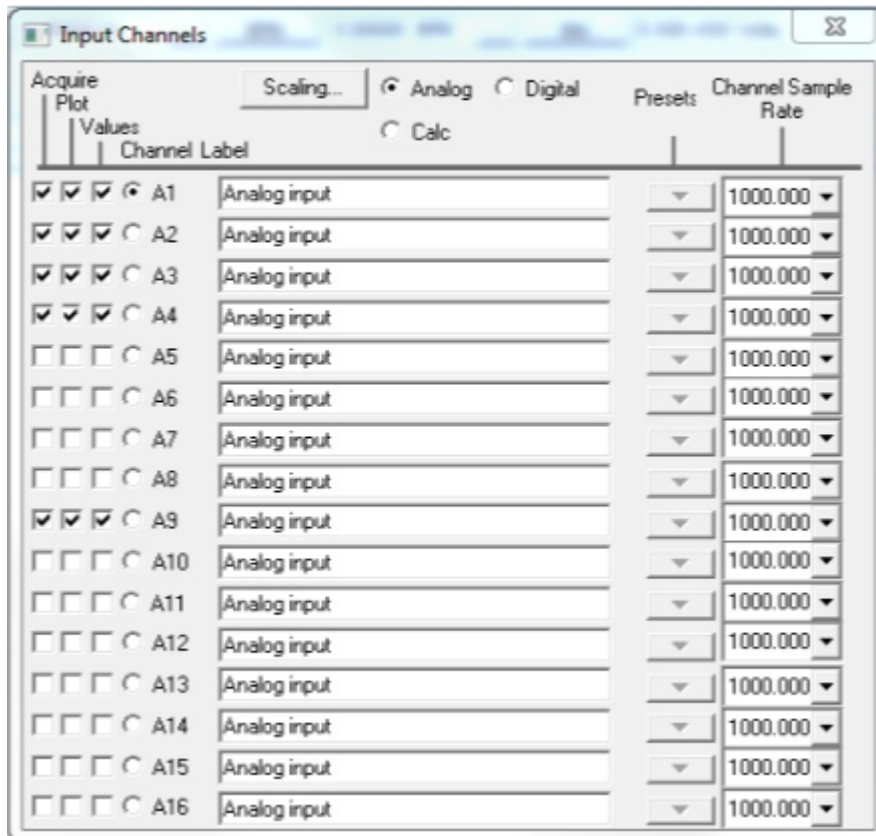
Este proyecto incluso se puede aplicar a la activación de una prótesis, para personas que carecen de alguna extremidad o incluso para la ayuda a personas con movilidad reducida.

2. Alternativa Profesional (con aparato y programa diseñado para ello)

El aparato que se necesita para esta alternativa es un Electromiógrafo y un software diseñado para captar y realizar los correspondientes filtros y las gráficas para ser fácilmente entendible por el usuario. Además el software AcqKnowledge 3.9.1, permite realizar múltiples configuraciones para la detección (con tarjetas de entradas) y el procesamiento de la

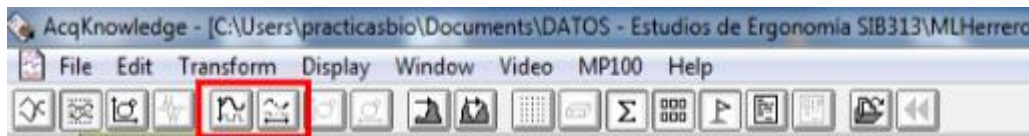
señal. Para utilizar el Electromiógrafo junto al software AcqKnowledge 3.9.1, explico el protocolo de actuación.

- Encender ordenador y abrir software AcqKnowledge 3.9.1
- Comprobar que los módulos (EMG100C) están bien configurados.
 - Cada uno de ellos deben estar en una canal diferente. Botón de la parte superior de cada módulo.
 - La ganancia debe estar en 2000
- Para hacer un nuevo registro dar a “new file” ”new data view”
- Configurar los canales de registro de la siguiente manera:
 - Hacer click en MP100 y elegir “Set Up Channels”
 - Elegir las tres casillas de la izquierda para activar los canales de EMG
 - Es aconsejable disponer los módulos en el siguiente orden
 - o Canal 1. EMG - Bíceps
 - o Canal 2. EMG - Tríceps
 - o Canal 3. EMG - Flexor
 - o Canal 4. EMG - Extensor
 - o Canal 5. Goniómetro
 - o Canal 6. Goniómetro
 - o Canal 7. Goniómetro
 - o Canal 8. Goniómetro
 - o Canal 9. EMG - Trapecio
- Si se va a utilizar los cinco canales de electromiografía, quedaría así:



- Configurar la escala pinchando en “Scaling” y ponemos las unidades en mV
- Tras esto, configurar la adquisición de la señal en MP100 “Set Up Acquisition”
 - En “Adquisition Sample rate” 1000 samples/second
 - “Total length” duración del registro (poner una duración mayor por si acaso)
- La colocación de los electrodos(visitar la Referencia)
- Comenzar el registro
 - Para comenzar el registro le damos a “Start” (abajo a la derecha)
 - Tras finalizar el registro podemos guardar la sesión en “File” “Save As” en .txt
- Visualización de la señal

Para comprobar que el registro es válido observamos la amplitud de la señal y su línea base. Para ello la línea base debe estar en el orden de los μV , y la amplitud en mV. Podemos observar la señal en su totalidad pulsando los botones de reajuste



-Bibliografía

-Criswell, E. (Ed.). (2010). Cram's introduction to surface electromyography. Jones & Bartlett Publishers.

-EMG100C – Electromyogram amplifier module. BioPac System.

-http://www.biopac.com/Product_Spec_PDF/EMG100C.pdf

-BioPac EMG tutorial. NCSU Human Factors and Ergonomics (HFE) Area . Nc State University.

-http://www.ise.ncsu.edu/ergolab/tutorial/BIOPAC_EMG/BIOPACEMG_Tutorial.php

3. Descripción

El primer paso de todos es recabar información sobre el proyecto, ya sea por otros proyectos parecidos o consultando a personas que sepan sobre ese tema y decidir en primera instancia como se va a realizar el proyecto, aunque se cambien partes del mismo posteriormente. Después de haber recabado información sobre el proyecto, sabremos lo que necesitaremos, y en este caso necesitaremos descargar e instalar varios softwares.

·Realizar la medición con los sensores

A la hora de conectar los electrodos y el sensor se tienen que tener en cuenta una serie de aspectos como el músculo donde se va a hacer la medición, la colocación de Electrodos y demás que explicará a continuación:

3.1. EMG (Electromiografía)

·LA LOCALIZACIÓN IDÓNEA PARA COLOCAR EL ELECTRODO

Las superficies de detección del electrodo deben estar orientadas de manera que se crucen normalmente la longitud de las fibras musculares. La longitud de las fibras que recorren la distancia entre las superficies de detección determina la cantidad de tiempo requerido para que los potenciales de acción atraviesen la distancia. Dado que la separación entre la

superficie de detección es fija, el valor de la velocidad de conducción aparente será una función del coseno del ángulo entre las fibras y las superficies de detección. Use estimulación eléctrica o mapeo eléctrico de superficie para ubicar las zonas de inervación.

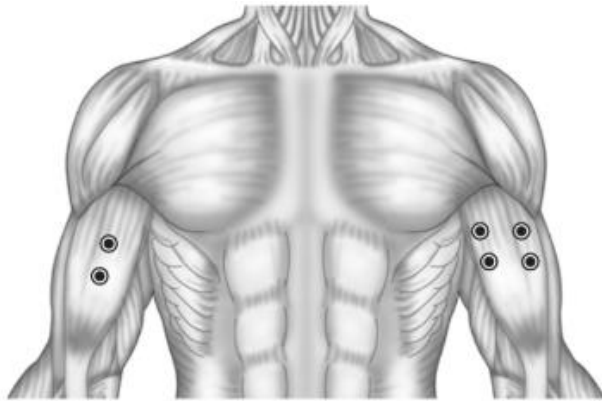
-Diámetro de la fibra: La velocidad de conducción de la fibra muscular es proporcional a (cierta potencia de) su diámetro. Por lo tanto, los músculos con fibras de mayor diámetro, como los que generalmente pertenecen a unidades motoras de umbrales más altos, tendrán mayores velocidades de conducción promedio que, a su vez, desplazarán el espectro de frecuencias hacia el rango de frecuencias altas y, en consecuencia, aumentarán el valor de la frecuencia media. Este efecto se puede notar cuando aumenta la salida de fuerza del músculo, pero la sensibilidad es baja (0.16 Hz /% MVC). No obstante, se esperarán distinciones en función del tamaño de la fibra como resultado del ejercicio, cuando la hipertrofia de las fibras musculares; como consecuencia del desuso, donde las fibras musculares se atrofian; y entre músculos de diferente género porque los machos generalmente tienen fibras de mayor diámetro.

Si se quiere saber la posición exacta, consulte: <http://www.seniam.org/>

·MÚSCULOS

Para realizar las pruebas y las mediciones para comprobar si el sensor mide bien utilizamos el bíceps, ya que la señal se puede medir fácilmente debido a lo cómodo que resulta poner los electrodos en el músculo. En la versión final del proyecto se utilizará otro músculo, pero por las razones citadas anteriormente, el software y parte del hardware se ha realizado tratando los valores del bíceps. En el apartado de abajo se cita donde se debe colocar los electrodos exactamente para la correcta captación de datos.

-BICEPS-BRACHIUM PLACEMENT



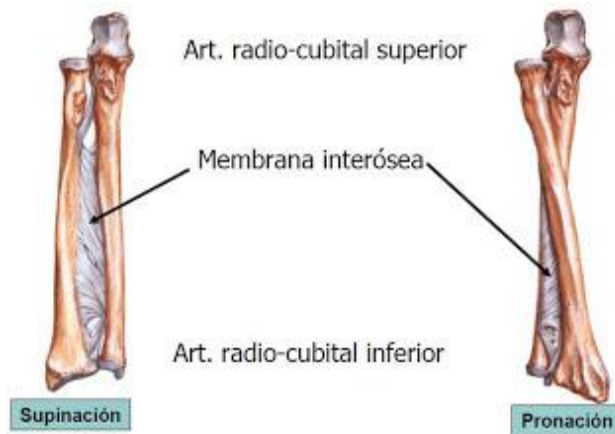
De acuerdo con los estándares europeos del sitio web de SENIAM (<http://seniam.org>), los sensores deben colocarse en la línea entre el acromion medial y la fosa cubital. Los electrodos deben orientarse en la dirección de la línea entre el acromion y fosa cubital. Se recomienda recordar limpiar la piel con alcohol propílico a 96°, sin abusar de la cantidad de alcohol para no cambiar la impedancia de la piel.

Para mantener el sensor conectado al brazo del paciente, tendremos que diseñar algo para mantenerlo conectado al paciente, esta parte es importante porque la torsión y el movimiento del brazo del paciente pueden llevar a medidas incorrectas.

El electrodo de referencia se puede colocar en la muñeca o en el codo, o en cualquier hueso, pero no en ningún otro lugar que no sea un hueso.

Para colocar correctamente el sensor sin ningún problema en la cabeza media del bíceps, debemos sentar al paciente en una silla con el codo flexionado en ángulo recto y la parte dorsal del antebrazo en una posición horizontal hacia abajo.

También hay que tener en cuenta que durante la supinación, el bíceps está activo mientras que el pronador está inactivo. Sin embargo, durante la pronación, el pronador teres está activo mientras el bíceps está inactivo.

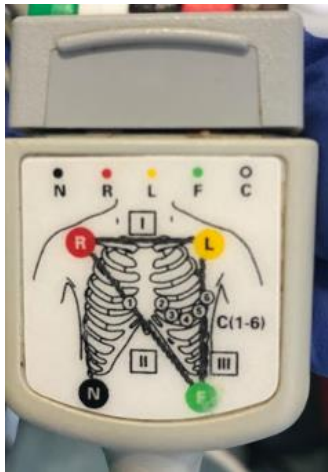


Una vez colocado los electrodos correctamente y haber tenido en cuenta los posibles ruidos y las interferencias, colocamos el jack al Myoware muscle sensor. Al Myoware muscle sensor le llegan tres cables desde el arduino, el +, - y analogico (Raw), el cual se encarga de la recogida de los valores dados por el sensor, para posteriormente pasarlos por el serial del arduino al ordenador donde corra el programa de matlab, mediante cable. Todas las anteriores conexiones están mostradas en los planos. El datasheet se encuentra en bibliografía y webgrafía.

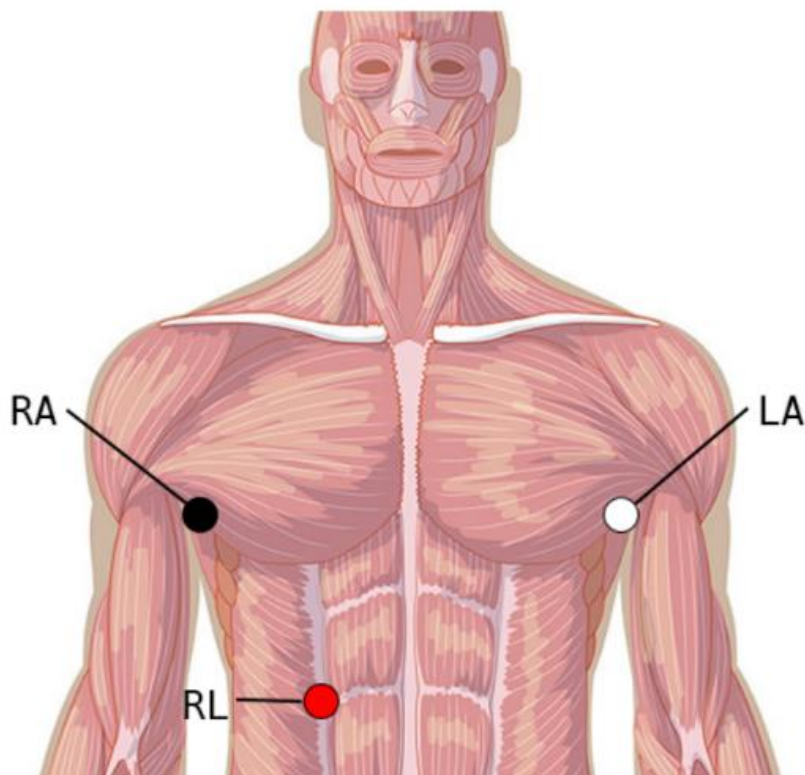
3.2. ECG (Electrocardiografía)

·LA LOCALIZACIÓN IDÓNEA PARA COLOCAR EL ELECTRODOS

La colocación de los electrodos para realizar la electrocardiografía se deben poner en triángulo (un triángulo que envuelva al corazón). Normalmente para detectar la actividad del corazón se utilizan las siguientes posiciones de los electrodos:



La imagen nos muestra tanto las posiciones de los electrodos como las formas en las que se pueden poner (derivaciones), este esquema es válido y fiable para una máquina diseñada para esta labor., Sin embargo para el sensor que nosotros utilizamos, la señal EMG nos puede ocasionar ruido en la señal ECG. Para evitar este problema, hemos empleado el siguiente esquema de posicionamiento para medir la señal ECG, que nos permite obtener la señal casi sin ruido.



Los colores de los electrodos están acordes al jack de conexión que hicimos para medir esta señal, aunque si se quiere utilizar el cable jack de la señal EMG, también se puede ya que solo se tiene que recordar que el electrodo de color blanco es el azul del jack del EMG.

Una vez colocado los electrodos correctamente, colocamos el jack al SparkFun AD8232. Al SparkFun AD8232 le llegan tres cables desde el arduino, el +, - y analogico(Sig), el cual se encarga de la recogida de los valores ya filtrados y amplificados por el sensor, para posteriormente pasarlos por el serial del arduino al ordenador donde corra el programa de matlab, mediante cable. Todas las anteriores conexiones están mostradas en los planos. El datasheet se encuentra en bibliografía y webgrafía.

4. Elementos utilizados

4.1. Hardware

·MyoWare Muscle Sensor



<https://www.sparkfun.com/products/13723>

Este dispositivo nos permite la medición de la actividad muscular mediante la detección de su potencial eléctrico, conocida como electromiografía (EMG).

El tablero MyoWare actúa midiendo la actividad eléctrica filtrada y rectificada de un músculo; emitiendo 0-Vs Volts dependiendo de la cantidad de actividad en el músculo seleccionado, donde Vs significa el voltaje de la fuente de energía.

Además de contar con una alimentación + 3.1V a + 5V, salida RAW EMG, clavijas de alimentación con protección contra polaridad. El sensor está diseñado para conectar los electrodos directamente al sensor sin necesidad de cables.

Con los accesorios siguientes:



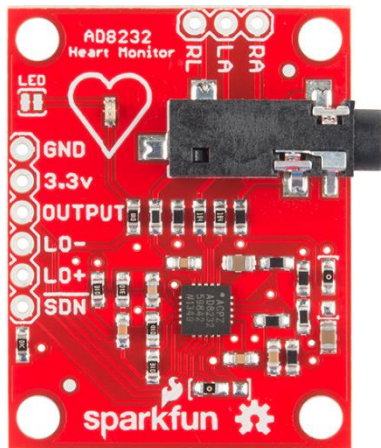
-Cable Shield: Utilizado por si se quiere colocar el hardware en otra localización, ya que viene con un jack de conexión para [los tres electrodos](#) .

-Arduino UNO



- Incorpora una autoselección del voltaje de alimentación (DC/USB).
- Dispone del nuevo bootloader OptiBoot que permite cargar programas a 115Kbps.
- El bootloader ha sido reducido en tamaño y ocupa 512bytes, tendremos más espacio para programar.

·SparkFun Single Lead Heart Rate Monitor - AD8232



(<https://www.sparkfun.com/products/12650>)

El monitor de frecuencia cardíaca de un solo cable SparkFun AD8232 es una placa rentable que se utiliza para medir la actividad eléctrica del corazón. Esta actividad eléctrica se puede graficar como un ECG o un electrocardiograma y emitirse como una lectura analógica. Los ECG pueden ser extremadamente ruidosos, el Monitor de Ritmo Cardíaco de Cable Único AD8232 actúa como un amplificador operacional para ayudar a obtener una señal clara.

El sensor trabaja a 3,3v y cuenta con una salida analógica "OUTPUT", la cual da valores a una entrada analógica de cualquier microcontrolador. Los demás pines son solo para comprobar si está conectado el sensor(Lo+ y Lo-) y la entrada "SDN" que mediante su activación el sensor pasa a un estado de bajo consumo.

4.2. Software

·Arduino

Nos iremos a la página oficial <https://www.arduino.cc/en/main/software> y entraremos en el apartado downloads, donde seleccionaremos Windows, Linux o Mac, dependiendo en que sistema operativo queramos correr el programa. Una vez descargado el instalador, le daremos a ejecutar como administrador y a seguir las indicaciones que nos vayan diciendo, hasta que nos muestre el botón de finalizar.

·Matlab

MATLAB es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Entre sus prestaciones básicas

se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. Este programa es un programa en el cual necesitas licencias, las cuales puedes adquirir en la página web <https://es.mathworks.com/store>.

5. Programación y Configuración de Equipos

En la parte de programación de este proyecto, se encuentran dos software.

El primero de todos es el Arduino, el cual se encarga de la captación y envío de datos y el segundo es el Matlab. El Matlab es un software matemático para tratamiento de señales.

5.1. (Arduino) - Señal EMG



Lo primero que hacemos es crear una variable donde se va a guardar la entrada que queremos. Esto se hace para tener el código más ordenado y también por si en algún momento se decide cambiar la entrada, para que solo se tenga que cambiar en esa variable y no en todas las líneas en las que se encuentre esa entrada. Además también especificamos la variable donde se van a guardar los datos (en este programa "value").

```
void setup() {  
  // put your setup code here, to run once:  
  
  pinMode(entrada, INPUT);  
  pinMode(enviar, INPUT);  
  pinMode(salida, OUTPUT);  
  //Serial.begin(115200);  
  Serial.begin(2000000);  
}
```

En el Setup debemos indicar las entradas y salidas que se necesiten en el proyecto. En este setup se encuentran dos entradas y una salida, la entrada que es imprescindible es la llamada “entrada”, las demás entradas y salidas son prescindibles ya que fueron para comprobar la comunicación. En la última línea encontramos los baudios a los que vamos a iniciar el puerto serial, utilizado más adelante.

```
void loop() {  
  // put your main code here, to run repeatedly:  
  
  value= analogRead(entrada);  
  //contador=contador +1;  
  Serial.println(value);  
  delay(1);  
}
```

En la última parte del proyecto encontramos el loop, donde en la primera línea leemos analógicamente los valores que recibimos del sensor. En la segunda lo imprimimos en el serial para que el Matlab pueda coger los datos y en la última realizamos una parada de 1ms, la cual significa que muestre esos valores cada milisegundo.

5.2. (Arduino) - Señal ECG



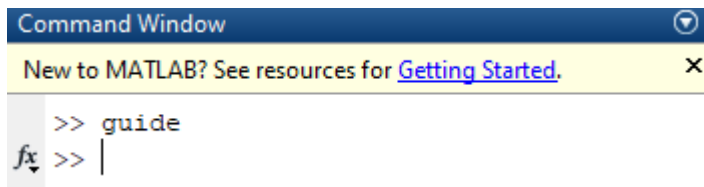
```
void loop() {
  // put your main code here, to run repeatedly:

  value= analogRead(entrada);
  //contador=contador +1;
  Serial.println(value);
  delay(1);
}
```

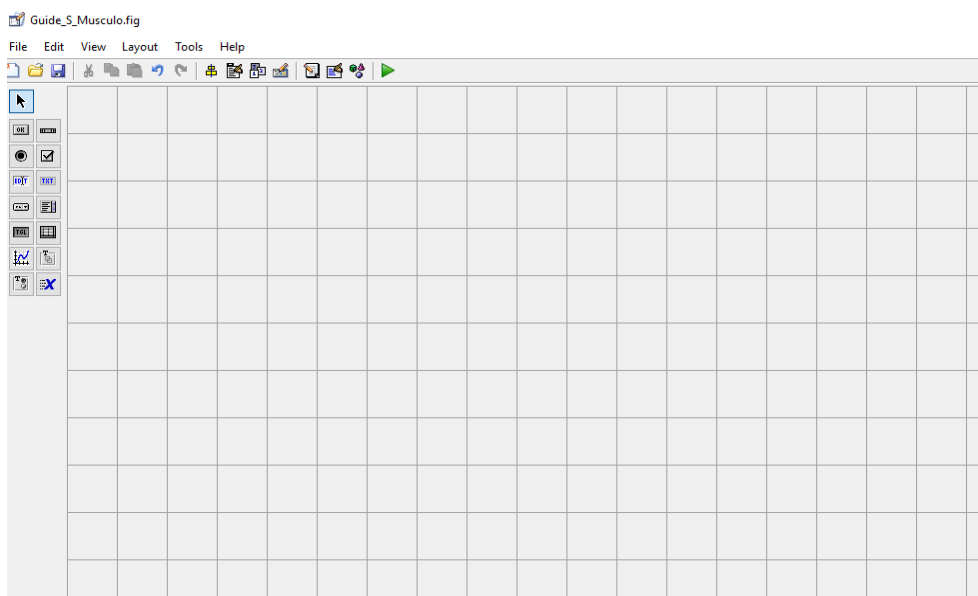
Como se observa, el código del arduino en donde se recoge la señal del sensor SparkFun AD8232 es el mismo que el del Muscule sensor, pero con la diferencia que cuando iniciamos el serial, lo hacemos a 9600 en vez de 2000000. Esta diferencia es debida a que el fabricante de este sensor recomienda estos baudios para obtener una correcta medición.

5.3. (Matlab) - Señal EMG

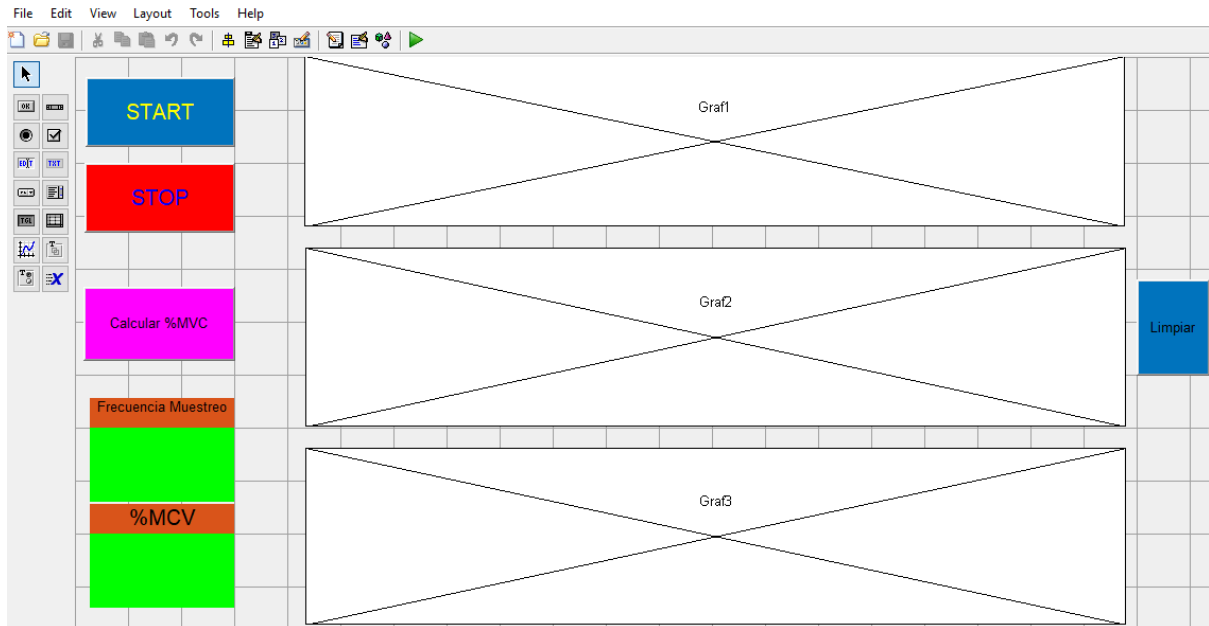
Lo primero que haremos es crear una interfaz interactiva compuesta de botones, cajas de texto y gráficas, con la aplicación “Guide”, que se encuentra dentro del paquete matlab. Escribimos en el command table la palabra “Guide”,



después de hacerlo nos saldrá una ventana donde nos dará la opción de abrir un proyecto ya existente o crear uno. Le daremos a crear uno nuevo y nos saldrá la aplicación sin ningún elemento.



Una vez aquí, procederemos a introducir elementos como botones y textos y gráficas, las cuales se encuentran en la parte izquierda, como se muestra en la imagen anterior. En este programa se ha diseñado la siguiente disposición, juntos con los elementos, ya que todos los elementos que se ven son utilizados en el programa tanto para su funcionamiento como para la visualización de la información volcada por el mismo.



Con todos los elementos ya creados nos dará un programa lleno de funciones, las cuales corresponden a cada uno de los elementos interactivos(botones).

En el programa tenemos los botones llamados START y STOP, que se utilizan para la recogida de datos y para la parada. En sus funciones se encuentra este código.

```
% --- Executes on button press in START.
function START_Callback(hObject, eventdata, handles)
% hObject      handle to START (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of START
set(hObject, 'UserData', false);
% LLAMADA A LA FUNCIÓN
VecV=ADC_Serial2(handles);

% --- Executes on button press in STOP.
function STOP_Callback(hObject, eventdata, handles)
% hObject      handle to STOP (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of STOP
%PARAR EL BUCLE WHILE QUE SE ENCUENTRA EN LA FUNCIÓN LLAMADA ANTERIORMENTE POR EL BOTON START
set(handles.START, 'UserData', true);
```

En la función de START llamamos a ADC_Serial2 (handles), que se encarga de la recogida de datos y los correspondientes filtros de esos datos, además de poner a False la función para que solo la llame una sola vez. En la función de STOP solo se pone a TRUE, la función que en el START se puso a FALSE, ya que en la función ADC_Serial2 (handles) es necesario para parar un bucle, que se encuentra en esa función.

```
function VecV=ADC_Serial2(handles) %la funcion recibe el # de muestras que debe tomar

voltage=0;%Declara variable en la cual se van a guardar los valores
delete(instrfind({'port'},{'COM12'})); %borrar cualquier puerto serial abierto
puerto=serial('COM12'); %declaro variable llamada puerto y se crea el com4
puerto.BaudRate=2000000; %Establecer velocidad de transmisión
fopen(puerto);%abre el puerto a utilizar
contador=1;
%M= zeros();
VecV=0;
tiempo_inicio = cputime;
```

Lo primero que hacemos en la función es declarar la variable donde vamos a guardar los datos recogidos por el puerto (línea 3), poner los baudios en los que va el puerto serial , abrir el puerto de comunicación y coger el tiempo de la cpu.

En la siguiente función realizamos un bucle while para poder coger los datos que nos envía el sensor mediante el arduino por el puerto, pero antes de meterse en el bucle, abrimos un txt donde vamos a escribir tanto los datos, como el tiempo en el que se cogio ese valor.

Dentro del while realizamos un if, que nos comprueba si el valor es erróneo o no, permitiendo así coger solo los valores correctos, para no dar fallos en el tratamiento posterior de los datos.

Para salir del bucle infinito, hacemos una comprobación de la variable need_to_stop. Esta variable queda activa cuando se llama a la función del STOP (pulsando el botón STOP) y por lo tanto realiza un break para terminar con el bucle, además de cerrar el txt para que los datos se guarden correctamente.

Después de guardar los datos en el txt y haber parado el bucle, pasamos al filtrado de la señal que consta de dos filtros, uno que coge los valores de 0 a la frecuencia de muestreo y otro que filtra el ruido de 50Hz producido por la corriente. Primero de realizar los filtros tenemos que recoger y guardar esos datos en una matriz para poder filtrarlos en el matlab.

```

texto=fopen('C:\Users\Usuario\Desktop\FCT\PARA_MATLAB\MATLAB\EMG_real.txt','wt');
]while true
    drawnow();
    grid off;
    valorADC=fscanf(puerto,'%d%d');%Toma el valor recibido por el puerto y lo guarda en la variable
    if ~isnan(valorADC)
        VecV= valorADC*5/1023;%Hace la conversión a Voltios|
        %xlswrite('MatLab_Sensor.xlsx',VecV,3,'A(contador)');
        fprintf(texto,'%f',VecV);%\n para saltar de parrafo
        total = cputime - tiempo_inicio;
        %xlswrite('MatLab_Sensor.xlsx',total,3,'B(contador)');
        fprintf(texto,','');
        fprintf(texto,'%f',total);
        fprintf(texto,'\n');
        contador=contador+1;
        fprintf('%f seg. - %f v.\n',total, VecV);
        need_to_stop=get(handles.START, 'UserData');%COMPROBACIÓN SI SE A PULSADO EL STOP
    end
    if need_to_stop %SALIR DEL BUCLE WHILE
        fclose(texto);
        break;
    end
end
end

%LEEMOS LOS VALORES DEL TXT
cd('C:\Users\Usuario\Desktop\FCT\PARA_MATLAB\MATLAB')
datos=importdata('EMG_real.txt');
M=datos;
plot(handles.Graf1,M(:,2),M(:,1))
delete(puerto);

```

Abrimos el mismo txt que hemos utilizado para guardar los datos, solo que esta vez es para leer los valores y para ello primero le ponemos la ruta donde se encuentra el archivo en la variable "cd", así como el nombre del archivo terminado siempre con su extensión, que en este caso será ".txt". Los valores que ha leído el matlab al abrir el archivo, los guardamos en una matriz llamada M, para representar en la primera gráfica del guide los valores y el tiempo de la señal aun sin filtrar.

```

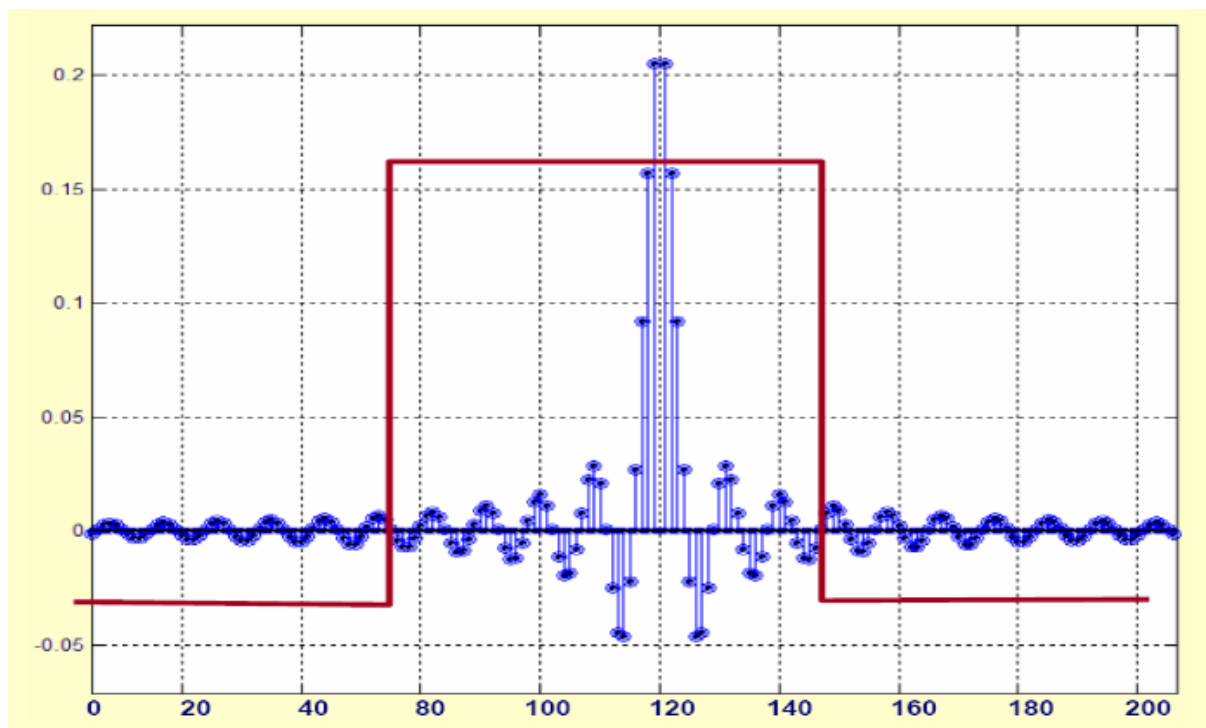
%FILTRADO

EMG_data= M(:,1);
% HIGHPASS FILTER (0-Fm Hz)
% =====
n = 4;
m=length(M(:,1))
m1=M(m,2)
m2= M(1,2)
l=(m1-m2)
Fm = (m/l) % ==> CALCULAR EL REAL (TiempoFinal - TiempoInicial)/NumeroMuestras
set(handles.text6, 'String', Fm);
Fcorteinf = 10;
highpassFilter = fdesign.highpass('n,f3db', n , 2*Fcorteinf*(1/Fm)); %high-pass filter
highpassFilterDesign = design(highpassFilter,'butter');
EMG_highpass = filter(highpassFilterDesign,EMG_data);

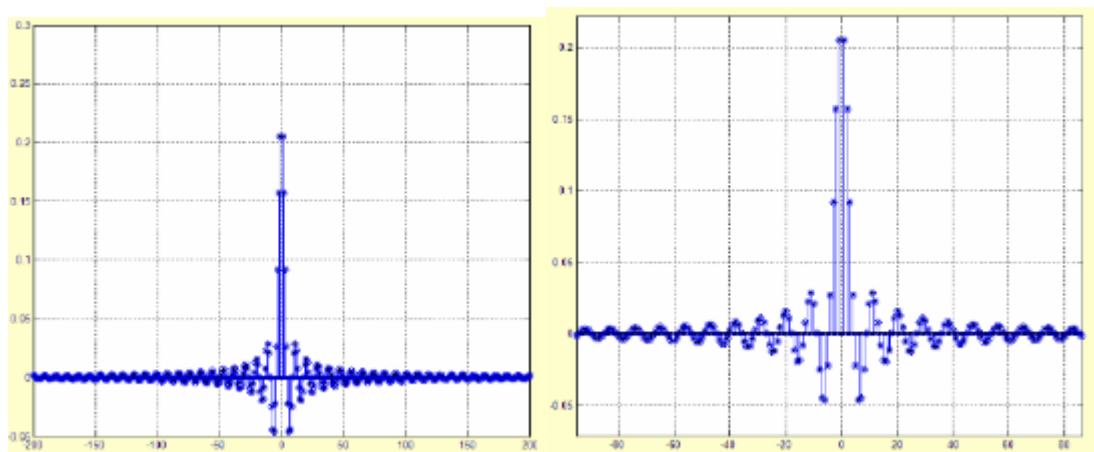
```

Para el funcionamiento del filtro paso alto que utilizamos, solo necesitamos los valores del sensor, puesto que son los datos sobre los que el filtro va a actuar y para esto en la variable “EMG_data” guardaremos los valores de la primera columna de la matriz. Si se quiere realizar el filtro, primero hay que crearlo, en la creación del filtro es necesario saber en qué rango de frecuencias actuará, el rango es de 10 Hz a la Frecuencia de muestreo (F_m) que obtengamos. La F_m se obtiene al restar del tiempo final el tiempo inicial, para luego dividir la cantidad de valores entre ese tiempo. Los valores anteriores se encuentran entre las líneas 4-9, de tal modo que en la línea 5 calculamos el número de elementos, en las líneas 6-7 cogemos el tiempo final y el inicial para en la siguiente línea calcular el tiempo que se utilizara para la fórmula de la F_m (línea 9). Una vez calculada la F_m , la mostramos en la primera caja de texto del guide con la función “set” del matlab (línea 10), donde primero ponemos el objeto donde lo vamos a mostrar, segundo le decimos en qué formato lo vamos a escribir y por último el dato que queremos mostrar. En las líneas 12-13 creamos el filtro con la fórmula y las funciones que aparecen, la fórmula está compuesta de dos constantes que son “n y $F_{corteinf}$ ”. En la última línea aplicamos el filtro a los valores y nos da una matriz de (número de elementos x 1).

Para entender cómo funciona este este filtro puedes investigar sobre el filtro Fir (Finite Impulse Response o Respuesta finita al impulso), pero una forma fácil de enterlo, es que al aplicar el filtro nosotros solo cogemos los datos que se encuentren en el rango del filtro, puesto que los demás se toman como valores falsos (ruido)



A continuación te muestro una señal antes y después del filtro para visualizar mejor su efecto de eliminación de datos y suavizado de la señal.



El siguiente filtro es necesario para quitar el ruido electromagnético presente en todos los aparatos eléctricos (50Hz). En este filtro realizamos el mismo sistema que en el anterior, primero lo diseñamos y luego lo aplicamos, con una sola diferencia, que en este caso solo escogemos una banda para aplicar el filtro. Los datos que filtraremos los cogemos de la matriz anterior donde se guardo los valores filtrados por el filtro precedente a este (EMG_highpass) y los valores filtrados por los dos filtros se almacenarán en la matriz EMG_Filtered. Una vez hecho el filtro procederemos a guardar esos valores en un txt, para ello realizamos un bucle for de 1 a la longitud de los valores, es decir, la cantidad de valores que existen (línea 8).

Se guardará también el tiempo en el que se captó los valores de la medición, como se muestra en la imagen, donde se ve el bucle for y las acciones que se realizan dentro de él. Cuando se finaliza el bucle for, los valores ya se han escrito, y por lo tanto cerraremos el txt para que los valores se guarden correctamente, además de mostrar en la gráfica número 2 la representación gráfica de los valores filtrados respecto a su tiempo de captación.

```

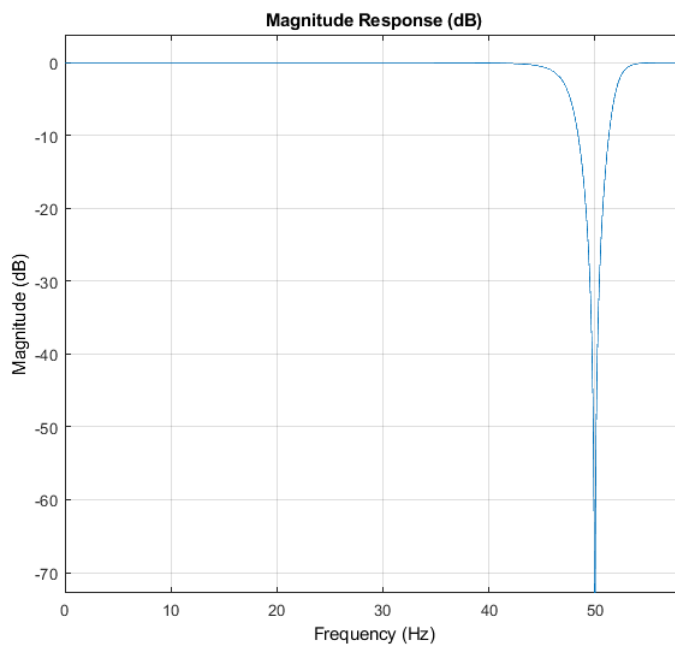
% NOTCH FILTER (50 HZ)
% =====
notchFilter = fdesign.notch('N,F0,Q',n,50,10,Fm); %notch filter (50Hz)
notchFilterDesign = design(notchFilter);
EMG_Filtered = filter(notchFilterDesign, EMG_highpass);

%ESCRIBIMOS EN EL TXT DE FILTRADO
texto=fopen('C:\Users\Usuario\Desktop\FCT\PARA_MATLAB\MATLAB\EMG_filtrado.txt','wt')
con=length(EMG_Filtered);
for i=1:con-1
    B=EMG_Filtered(i,1);
    Y=M(i,2);
    fprintf(texto,'%f',B);
    fprintf(texto,',');
    fprintf(texto,'%f',Y);
    fprintf(texto,'\n');
end
plot(handles.Graf2,M(:,2), EMG_Filtered);

fclose(texto);

```

La mejor forma de entender el filtro de 50Hz es visualizando la siguiente gráfica, la cual hace más visible su efecto.



A la hora de realizar los cálculos pertinentes, para la obtención del rms necesario para el cálculo del %MVC (Maximum voluntary contraction), pulsamos el botón “Calcular %MVC”, llamando así a su función. En esta función primero de todo elegimos el archivo txt, de donde sacaremos el rms de la máxima contracción voluntaria, a través de una ventana emergente en la cual nos da la opción de dar a aceptar y buscar el archivo de muestras MVC que nosotros queremos medir.

```

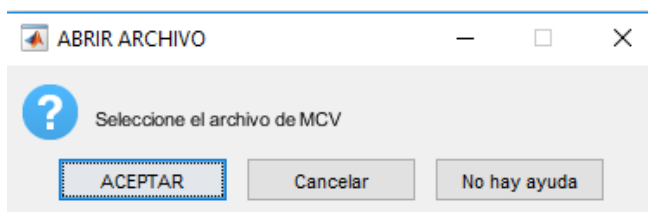
button=questdlg('Seleccione el archivo de MCV','ABRIR ARCHIVO','ACEPTAR','Cancelar','No hay ayuda','ACEPTAR')
if strcmp(button,'Cancelar')||strcmp(button,'No hay ayuda')
    return;
end;

[filename,pathsubj]=uigetfile('*.txt','Seleccione directorio con las carpetas de los SUJETOS');

if path==0
    return;
end;

cd(pathsubj)
datos=importdata(filename);

```



Como vemos en las últimas líneas, el cd y el filename se sacan al elegir el txt. Los datos abiertos se guardan en la matriz datos. Luego se muestra en la tercera gráfica los datos, para coger una ventana de dos puntos, los cuales elegimos para así seleccionar la máxima actividad de los datos recogidos del MVC.

```

muscle=datos(:,1);
T=datos(:,2);

plot(handles.Graf3,muscle);
aux=ginput(2);
aux = fix(aux);
vectorMuscle=muscle(aux(1):aux(2));
vectorTime=T(aux(1):aux(2));
rms =EMG_rms(vectorMuscle,100); % Ventana temporal de 100 muestras (100 ms)

```

Ya elegida la ventana, cogemos ese vector (vectorMuscle) y le realizamos el rms, a través de un algoritmo ya hecho, llamado “EMG_rms”.

Este software trabaja cogiendo ventanas de 100 muestras de los datos del vector, compuestos únicamente por los valores de la máxima actividad de los datos recogidos del MVC.

Los valores de rms se guardan en una matriz llamada “rms”. Para guardar estos datos los escribimos nuevamente en un txt.

```

%ESCRIBIMOS EN EL TXT DE MCV_EMG_RMS
texto=fopen('C:\Users\Usuario\Desktop\FCT\PARA_MATLAB\MATLAB\MCV_EMG_RMS.txt','wt');
Con=length(rms);
for i=1:1:Con-1
    C=rms(i,1);
    X=vectorTime(i,1);
    fprintf(texto,'%f',C);
    fprintf(texto,',' );
    fprintf(texto,'%f',X);
    fprintf(texto,'\n');
end
fclose(texto);
plot(handles.Graf3,vectorTime, rms);
i=1;
MCV(i)=mean(rms)

```

Para hacer visible el rms, lo representamos en la gráfica 3 y por último le damos a la variable MCV el valor del rms calculado.

En el cálculo del rms del test o de la prueba que se realiza para medir la actividad del músculo, se realiza la misma operación que la anterior, solo que con la diferencia de que en este cálculo cogemos todos los valores de la matriz. El programa también cuenta con un filtro anti picos (de 1,5 a -1,5) en este apartado (línea 16 - 21), para que el cálculo del rms se haga con una gráfica lo más normalizada posible y así evitar un mal cálculo.

```

%MVC Testttttttttt

button=questdlg('Seleccione el archivo del TEST','ABRIR ARCHIVO','ACEPTAR','Cancelar','No hay ayuda','ACEPTA
    if strcmp(button,'Cancelar')||strcmp(button,'No hay ayuda')
        return;
    end;
    [filename,pathsubj]=uigetfile('*.txt','Seleccione directorio con las carpetas de los SUJETOS');
    if path==0
        return;
    end;

    cd(pathsubj)
    datos=importdata(filename);
    display('*')
    display('*USTED SELECCIONO:')
    filename

    F=length(datos);
    for i=1:1:F-1
        if datos(i,1)<=1 && datos(i,1)>=(-1)
            Tes(i,1)=datos(i,1);
            Tes(i,2)=datos(i,2);
        end
    end
    musculo=Tes(:,1)
    Tiempo=Tes(:,2);
    rms2 =EMG_rms(musculo,100); % Ventana temporal de 100 muestras (100 ms)

```



```

%ESCRIBIMOS EN EL TXT DE EMG_RMS
texto=fopen('C:\Users\Usuario\Desktop\FCT\PARA_MATLAB\MATLAB\EMG_RMS.txt','wt');
S=length(rms2);
for i=1:1:S-1
    D=rms2(i,1);
    W=Tiempo(i,1);
    fprintf(texto,'%f',D);
    fprintf(texto,',' );
    fprintf(texto,'%f',W);
    fprintf(texto,'\n');
end
fclose(texto);
plot(handles.Graf3, rms2);
i=1;
MCV2(i)=mean(rms2)

```

Una vez obtenido el rms tanto del MVC como del test, realizamos una sencilla regla de tres para obtener el porcentaje, en la que el rms de la MVC es el 100% y por lo tanto el rms del test corresponderá a X% sobre su máxima. Para terminar y que se haga más visual el cálculo, el porcentaje resultante lo ponemos en la última caja de texto.

```

%REALIZAMOS EL CALCULO DEL %MCV DE LOS DOS RMS CALCULADOS ANTERIORMENTE

```

```

MCV

```

```

MCV_Porcentaje=( (MCV2*100)/MCV);

```

```

set(handles.text4,'String',MCV_Porcentaje);

```

El último botón “Limpiar” sirve para limpiar todas las variables del programa, las gráficas del guide y el command windows, ya que entre medición y medición pueden existir errores, pero si se pulsa este botón cada vez que queramos empezar una nueva medición, no habrá errores puesto que en algún sentido ponemos el programa desde cero. Las acciones que se realizan dentro de esta función son las siguientes:

```

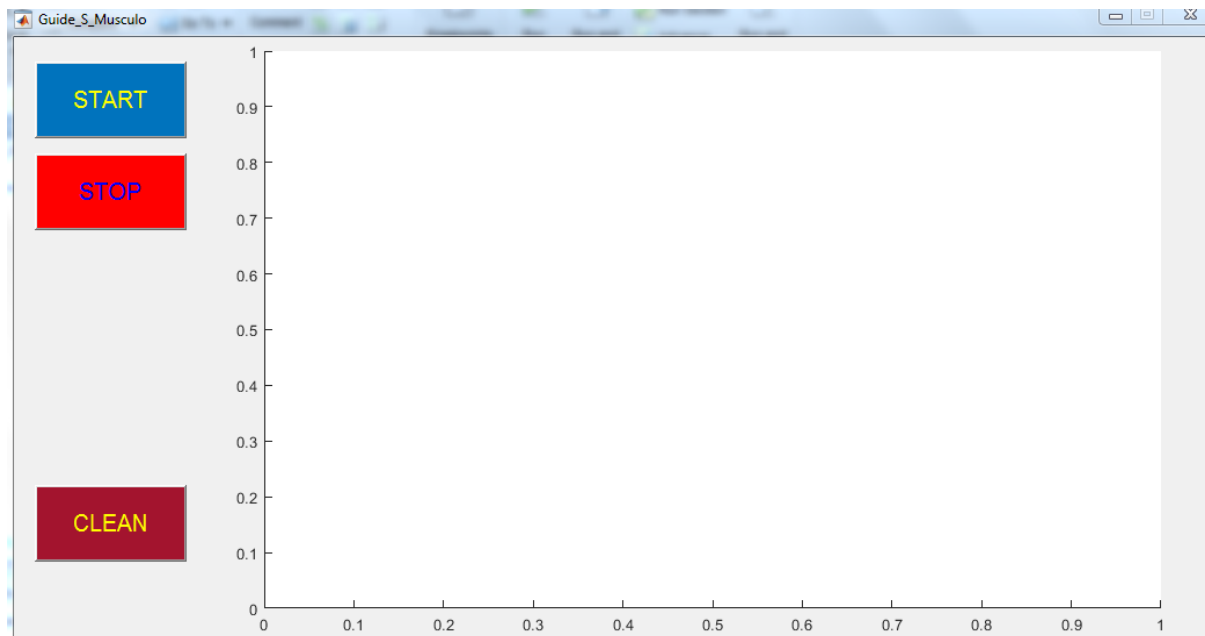
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
plot(handles.Graf1,0, 0);
plot(handles.Graf2,0, 0);
plot(handles.Graf3,0, 0);
set(handles.text4,'String',0.0);
set(handles.text6,'String',0.0);
clc
clear

```

5.4. (Matlab) - Señal ECG

En el programa de medición de señales ECG (Electrocardiograma) solo se recogen los valores dados por el sensor. Esto es debido a que el sensor nos da una señal amplificada y filtrada, lo cual hace que el programa sea más sencillo que el de la señal EMG.

En el Guide se encuentran los siguientes elementos:



Como en el anterior programa el botón de START y el de STOP, tienen la misma función, además de que a la hora de recoger y mostrar los datos utilizamos el mismo código de escritura en el txt y lectura del mismo para mostrarla en la gráfica del guide.

```
function VecV=ADC_Serial2(handles) %la funcion recibe el # de muestras que debe tomar

voltage=0;%Declara variable en la cual se van a guardar los valores
delete(instrfind({'port'},{'COM19'})); %borrar cualquier puerto serial abierto
puerto=serial('COM19'); %declaro variable llamada puerto y se crea el com4
puerto.BaudRate=9600; %Establecer velocidad de transmisión
fopen(puerto);%abre el puerto a utilizar
contador=1;
%M= zeros();
VecV=0;
tiempo_inicio = cputime;
```

```

texto=fopen('C:\Users\jasanchez\Documents\Sensor_M.txt','wt');
while true
    drawnow();
    grid off;
    valorADC=fscanf(puerto,'%d%d');%Toma el valor recibido por el puerto y lo guarda en la variable
    VecV= valorADC*3.3/1023;%Hace la conversión a milivoltios(PERO PARA NO PERDER DATOS SE MASA A MICROVLTIOS)
    fprintf(texto,'%f',VecV);%n para saltar de parrafo
    total = cputime - tiempo_inicio;
    fprintf(texto,'% ');
    fprintf(texto,'%f',total);
    fprintf(texto,'\n');|
    contador=contador+1;
    fprintf('%f seg. - %f v.\n', total, VecV);
    need_to_stop=get(handles.START, 'UserData');%COMPROBACIÓN SI SE A PULSADO EL STOP
    if need_to_stop %SALIR DEL BUCLE WHILE
        fclose(texto);
        break;
    end
end
-end

%LEEMOS LOS VALORES DEL TXT
cd('C:\Users\jasanchez\Documents')
datos=importdata('Sensor_M.txt');
M=datos;
%fclose(texto);
plot(handles.Graf1,M(:,2),M(:,1))
%winopen('C:\Users\jasanchez\Documents\Sensor_M.txt');
%VecV;
%contador;
%plot(M(1:1:contador-1));
%VecV(1:1:contador-1)
%fclose(puerto);
delete(puerto);

```

Como en el anterior programa, tenemos un botón llamado “Clean” que tiene la misma función que el botón “Limpiar” del anterior programa. Las acciones son las siguientes :

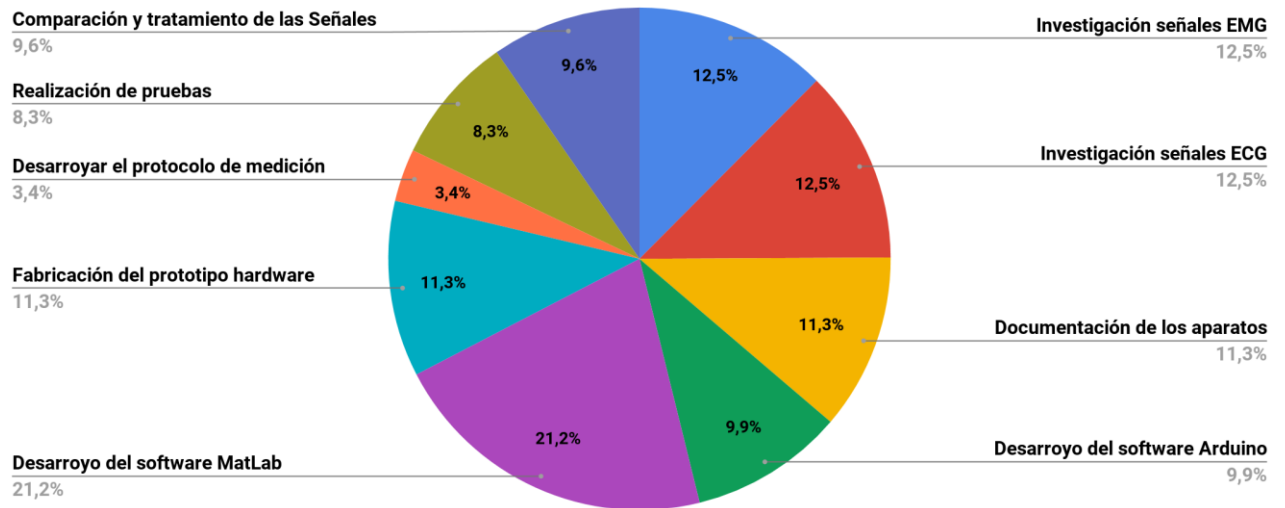
```

function pushbutton6_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
plot(handles.Graf1,0, 0);
plot(handles.Graf2,0, 0);
plot(handles.Graf3,0, 0);
set(handles.text4,'String',0.0);
set(handles.text6,'String',0.0);
clc
clear

```

6. Planificación

Planificación del Proyecto



7. Presupuesto

7.1. Mediciones

Concepto	Cantidad
Myoware Muscule Sensor	1
MyoWare Cable Shield	1
Single Lead Heart Rate Monitor	1
Sensor Cable - Electrode Pads	2
Placa Uno R3 ATmega328P	2
Cables (Conexión y transmisión)	1

7.2. Cuadro de Precios

Concepto	Precio
Myoware Muscule Sensor	26,75 €
MyoWare Cable Shield	3,50 €
Single Lead Heart Rate Monitor	14,06 €
Sensor Cable - Electrode Pads	3,49 €
Placa Uno R3 ATmega328P	11,84 €
Cables (Conexión y transmisión)	5,66 €

7.3. Presupuesto

Concepto	Cantidad	Precio Unitario	Precio Subtotal
Myoware Muscule Sensor	1	26,75 €	26,75 €
MyoWare Cable Shield	1	3,50 €	3,50 €
Single Lead Heart Rate Monitor	1	14,06 €	14,06 €
Sensor Cable - Electrode Pads	2	3,49 €	6,99 €
Placa Uno R3 ATmega328P	2	11,84 €	23,69 €
Cables (Conexión y transmisión)	1	5,66 €	5,66 €

Presupuesto de materiales 80,65€

7.3. Resumen

El resumen del presupuesto es:

Presupuesto de ejecución de materiales	80,65 €
6% de gastos generales	4,84 €
10% de beneficio industrial	8,07 €
Suma	93,55 €
IVA (21%)	19,65 €
TOTAL (€)	113,20 €

El coste total del proyecto es de ciento trece euros con veinte céntimos.

*En el presupuesto no se ha introducido la mano de obra realizada por mi parte ya que al haber necesitado el trabajo de más o menos tres meses, el presupuesto se hubiera encarecido considerablemente y por lo tanto hubiera opacado el presupuesto real del material necesario. Lo que pretendo es dar importancia al presupuesto del prototipo físico únicamente. En este se muestra un prototipo barato y funcional, capaz de hacer frente económicamente a un aparato tanto de electromiografía como de electrocardiografía, los cuales rondan desde los mil euros en adelante (el precio varia dependiendo de qué aparato se elija). Si se quisiera calcular el precio del trabajo realizado por el profesional, se podría ver los honorarios que le corresponderían por hora trabajada.

Yo personalmente no he introducido en el presupuesto la mano de obra, ya que para crear este prototipo en concreto se necesitan bastantes horas de trabajo, debido a que se debe investigar y realizar demás acciones detalladas previamente en la planificación, mientras

que cuando el prototipo ya ha sido creado y es completamente funcional, fabricar otro prototipo igual no conlleva más trabajo que montar las conexiones y comprar el material, puesto que el verdadero trabajo solo se realizó al crearlo y hacerlo funcional por primera vez. Por esta razón me parece que poner la mano de obra en este presupuesto no sería real, ya que la mano de obra sería cara la primera vez y las demás veces no. Entonces no tiene sentido hacer un presupuesto que solo valiese únicamente para la creación del primer prototipo.

8. Metodología

En este apartado pretendo exponer los resultados de las pruebas realizadas con cirujanos laparoscópicos, para así verificar la utilidad del proyecto y su aplicación en el entorno quirúrgico y en la evaluación y validación de nuevo equipamiento médico-quirúrgico.

8.1. FlexDex

8.1.1. Objetivos

Con este estudio buscamos evaluar objetivamente el uso del nuevo instrumento FlexDex® durante la práctica laparoscópica. Para ello, se evaluaron el rendimiento quirúrgico y la ergonomía del cirujano durante diferentes tareas en un simulador laparoscópico, comparándolas con el uso de un portaagujas laparoscópico convencional.

8.1.2. Material y métodos

Tres cirujanos laparoscópicos con diferentes niveles de experiencia realizaron una tarea de coordinación ojo-mano y una tarea de sutura en un simulador laparoscópico. La tarea de coordinación ojo-mano consistía en transferir, con ambas manos, una aguja quirúrgica a través de un circuito de anillas. La tarea de sutura consistió en realizar una sutura intracorpórea sobre un tejido orgánico con un nudo doble y dos nudos simples en direcciones opuestas. Cada tarea se repitió cinco veces y se llevaron a cabo utilizando el instrumento FlexDex® y un portaagujas convencional de forma aleatoria. Antes del estudio, los participantes realizaron una sesión de entrenamiento de 15 minutos con el nuevo instrumento. Durante el estudio se registró el tiempo de ejecución de cada tarea. Además, se analizó la flexión y la desviación radio-ulnar de la muñeca mediante un electrogoniómetro (Biopac systems, Inc.) conectado a la mano y al antebrazo del cirujano, así como la activación de los músculos del antebrazo por medio del brazalete MYO (Thalmic Labs).

8.1.3. Resultados

Los cirujanos necesitaron más tiempo para realizar tanto la tarea de coordinación ojo-mano ($183,8 \pm 53,2$ s frente a $295,8 \pm 55,7$ s; $p < 0,05$) como la sutura intracorpórea ($69,0 \pm 12,5$ s. frente a $73,2 \pm 13,8$ s; $p < 0,05$) utilizando el nuevo portaagujas laparoscópico. El uso del instrumento flexible condujo a una mayor activación muscular del músculo extensor (18.978 ± 1.816 RMS vs 26.165 ± 4.085 RMS; $p < 0.01$) y del flexor (12.373 ± 1.378 RMS vs 16.734 ± 3.043 RMS; $p < 0.01$) del antebrazo. Aunque la desviación cubital (23.047 ± 5.318 grados vs 22.878 ± 3.277 grados) y la flexión (12.030 ± 3.740 grados vs 10.730 ± 5.316 grados) de la muñeca fueron menores con el nuevo instrumento, no se mostraron diferencias estadísticamente significativas.

8.1.4. Conclusiones

Creemos que se requiere un período de entrenamiento más largo con el instrumento FlexDex® para lograr un rendimiento quirúrgico similar al de los instrumentos laparoscópicos convencionales. Los resultados preliminares de este estudio muestran que el nuevo instrumento permite posturas ergonómicamente más adecuadas para el cirujano durante la práctica laparoscópica, aunque es físicamente más exigente.

8.2. SensOR

8.2.1. Objetivos del estudio

Evaluar el efecto de la retroalimentación visual de la fuerza aplicada sobre el rendimiento quirúrgico, la fuerza aplicada y la ergonomía del cirujano durante el entrenamiento de la sutura laparoscópica mediante la tecnología SensOR (SensOR Medical Laboratories Ltd.).

8.2.2. Material y métodos

Cuatro cirujanos laparoscópicos novatos participaron en este estudio. Los participantes realizaron una sutura laparoscópica en un estómago de cerdo ex-vivo. Los cirujanos fueron asignados, de manera aleatoria, al grupo que recibía retroalimentación visual de la fuerza aplicada (VFF) o al grupo de control (Non-VFF) sin retroalimentación visual. En total, se realizaron once ensayos de entrenamiento (T0-T10). Se realizaron tres pruebas de evaluación durante T0, T5 y T10 para analizar la curva de aprendizaje de los participantes. La fuerza aplicada sobre el tejido por el disector fue registrada por medio de la tecnología SensOR y proporcionando una retroalimentación visual de la fuerza aplicada por medio de un dispositivo móvil adicional. Durante cada prueba, se analizó el tiempo de ejecución. Además, se empleó tecnología portátil (wearable) para estudiar el porcentaje de la máxima contracción voluntaria de la actividad del músculo trapecio alto por medio de un sistema EMG portable y personalizado, la activación de los músculos flexores y extensores del

cirujano mediante el brazalete MYO (Thalmic Labs) y la fuerza aplicada por los dedos pulgar, índice y medio por medio de la tecnología Finger TPS (Pressure Profile Systems).

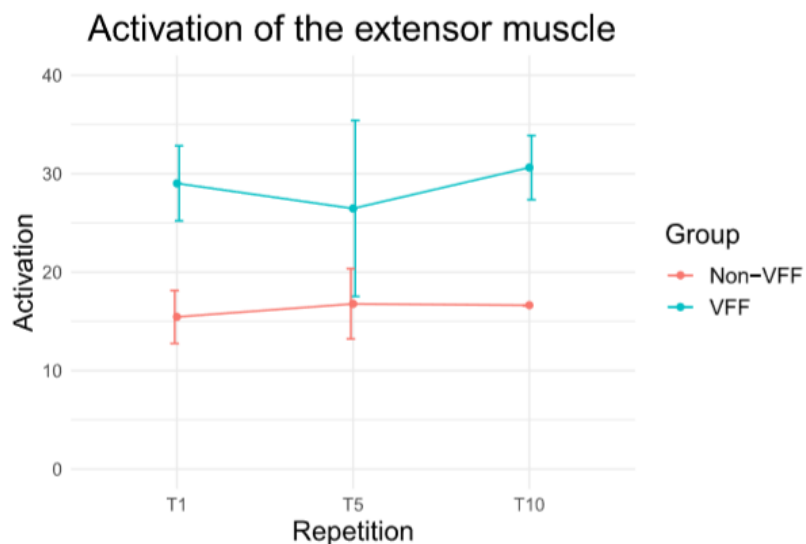
8.2.3. Resultados preliminares

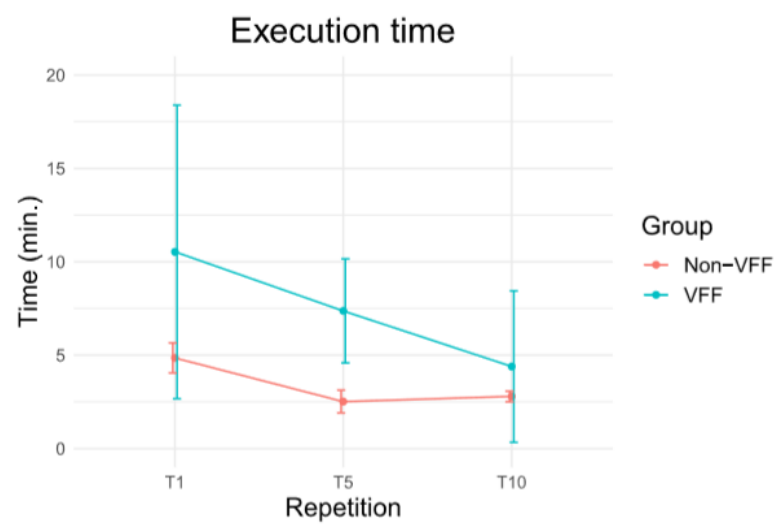
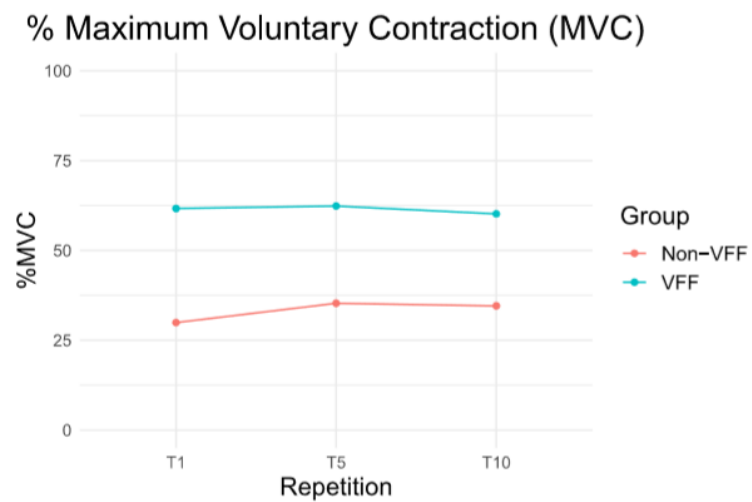
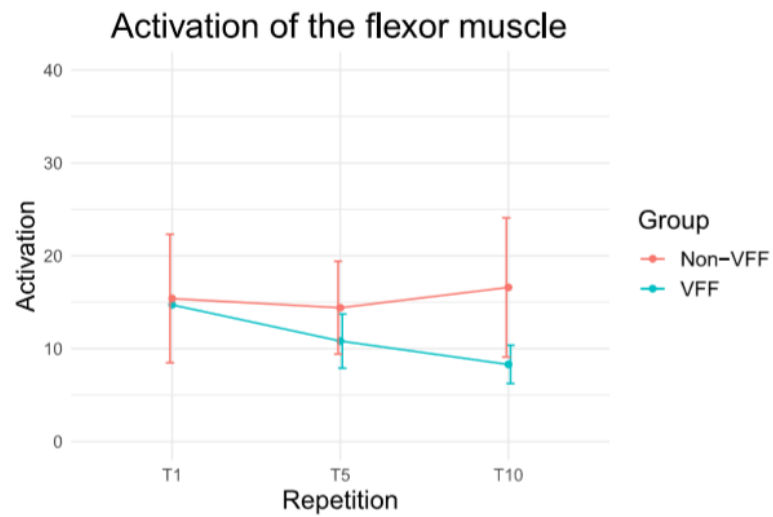
El uso de la retroalimentación visual de la fuerza aplicada con el instrumental quirúrgico requirió más tiempo para realizar la sutura laparoscópica, pero con una curva de aprendizaje positiva, e incrementó la activación del músculo extensor y trapecio alto de los cirujanos. Sin embargo, esta tecnología llevó a una activación más baja del músculo flexor y permitió a los cirujanos aplicar menos fuerza con la punta del instrumento en el tejido, así como con el dedo índice durante el manejo del instrumental quirúrgico.

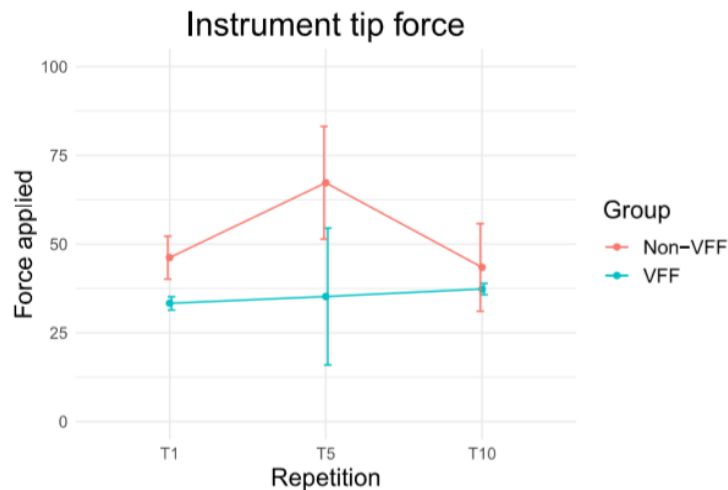
8.2.4. Conclusiones

Los resultados preliminares de este estudio muestran que del uso de la tecnología SensOR para ayudar al entrenamiento laparoscópico conduce a una mejora en la fuerza aplicada al tejido, así como en el manejo de los instrumentos laparoscópicos. Se requieren más estudios con un número mayor de participantes para reforzar estas conclusiones.

Para visualizar mejor estos resultados, se facilita una representación gráfica de las medias:







9. Planos

- Plano N° 1→ Conexionado del sensor MyoWare muscle sensor con su respectivo arduino.
- Plano N° 2→ Conexionado del sensor **Single Lead Heart Rate Monitor - AD8232.** con su respectivo arduino.
- Plano N° 3→ Mediciones y representación de la caja que contiene los sensores y los microcontroladores.
- Plano N° 4→ Conexionado entre los microcontroladores y los ordenadores, mediante cable.

10. Bibliografía y Webgrafía

10.1. Bibliografía

- Day - 2002 - Important Factors in Surface EMG Measurement.
- Criswell - 2010 - Cram's introduction to surface electromyography.
- Florimond - Basics of Surface Electromyography Applied to Physical Rehabilitation.(2010).
- De Luca, Luca - 1997 - The use of surface electromyography in biomechanics.
- mangukiya.(2017).
- Barbero [Book, Springer] - Atlas of Muscle Innervation Zones, Understanding Surface Electromyography and Its Applications.(2012).
- Rose - EMG analysis.(2012).

10.2. Webgrafía

-https://es.mathworks.com/support.html?s_tid=gn_supp

-<http://www.seniam.org/>

11. Conclusiones

Al finalizar este proyecto, se consiguió alcanzar la finalidad para la cual fue propuesto, realizar mediante un arduino y un conjunto de sensores, la captación tanto de señales EMG como de ECG.

Esta finalidad fue comprobada mediante un conjunto de pruebas en un entorno quirúrgico experimental para evaluar y validar nuevo equipamiento quirúrgico, para demostrar que el prototipo cumplía con la función, en todas las pruebas y que su funcionamiento no tuviese ningún problema si la cantidad de datos y de tiempo fuese demasiado grande. Después de todas las comprobaciones, se llegó a la conclusión de que este prototipo es útil y efectivo, para la captación de las señales mencionadas anteriormente, aunque no se compara con la calidad de frecuencia de muestreo de un aparato EMG o ECG, si resulta más portable y económico. Este sensor permite reducir el ruido en la señal debido a que la distancia entre los electrodos y el prototipo es relativamente pequeña, además de que el sensor aísla un poco las ondas electromagnéticas. Cabe mencionar que este prototipo se puede utilizar en pruebas clínicas, por la fiabilidad en la transmisión de datos obtenida en los estudios realizados, pero necesitaría una mejora en el aspecto de la frecuencia de muestreo y también realizar una placa para que las conexiones fuesen por soldadura en vez de por pines de conexión. Para poder conseguir más frecuencia de muestreo, se podría utilizar otro microcontrolador con más precisión y más potencia de cálculo que el arduino, como por ejemplo una Raspberry.

Para llegar al prototipo final, hizo falta un estudio exhaustivo sobre la captación y el filtrado de las señales. La captación de la señal EMG es la más compleja debido a que tiene demasiadas variantes, como la piel, las fibras musculares, etc, pero para solucionar este problema, se hizo un protocolo de medición, en el cual se detalla la posición adecuada de los electrodos, que hay que limpiar previamente la zona con alcohol, utilizar electrodos nuevos y cambiarlos para cada medición, además de tener en cuenta el músculo y las fibras adyacentes a la zona donde se sitúan los electrodos, para recoger una señal lo más limpia posible. A la hora de realizar el filtrado, se tuvo problemas con la frecuencia de muestreos ya que muchos filtros nos pedían un mínimo de 10000 Hz, al ser diseñados para equipos profesionales, pero finalmente conseguimos realizar los filtros con la frecuencia de muestreo que nos arrojaba el dispositivo. Los problemas que aparecieron durante la construcción del prototipo fueron bastantes, pero gracias al trabajo y a la investigación, se consiguió realizar un prototipo sin fallos.

Este prototipo no solo es útil para conocer la ergonomía o la activación de los músculos y el ritmo cardíaco de los sujetos, sino que también puede ser utilizado en la fabricación y

puesta en marcha de prótesis para personas con amputaciones. En otras líneas más lejanas, se podría utilizar para evaluar la salud de los cirujanos durante el desempeño quirúrgico y, de este modo, contribuir a la mejora de la calidad del servicio prestado a los pacientes. Del mismo modo, este proyecto se podría aplicar con el objetivo de ayudar a personas con movilidad reducida mediante un traje especial que capte la señal y la envíe a actuadores para ayudar a las personas a realizar tareas.

Este proyecto a supuesto trabajo, dedicación y un ímpetu enorme para realizar un prototipo que no solo funcionase para la tarea que fue pensada, sino que también pudiese servir para otros campos y líneas de investigación.

En conclusión, este prototipo ha logrado realizar la función de un aparato de EMG y ECG, con arduino y un conjunto de sensores. Todo ello, con un coste ínfimo comparado con un aparato profesional.

12. Anexos

Anexo 1 : MyowareUserManualAT-04-001-1223951

Anexo 2 : AD8232 Heart Rate Monitor Hookup Guide