Módulo de Adquisición y Procesamiento de Señales Mioeléctricas

Javier De La O Ortiz [1*], Karla Pérez Anaya[1]

Daniel De La Cruz Muciño[2] , Miguel Gabriel Villarreal-Cervantes[2]

[1] UPIITA, Av. Instituto Politécnico Nacional 2580.

Barrio La Laguna Ticomán, Gustavo A. Madero, 07340, México .D.F.

[2]Departamento de Posgrado, Sección de Mecatrónica, Instituto Politécnico Nacional-CIDETEC.

Av. Juan de Dios Bátiz s/n, Esq. Miguel Othon de Mendizabal

Unidad Profesional Adolfo López Mateos, C.P. 07700.

Correo electrónico: javierdelaoupiita@hotmail.com, karly99@gmail.com

danieldlcm@msn.com, mvillarrealc@ipn.mx

Resumen

Los dispositivos diseñados para proporcionar rehabilitación funcional a los miembros tanto torásicos como pélvicos se basan en ejercicios de repetición que estimulan la neuroplasticidad cerebral, esto se debe a la capacidad del cerebro para crear redes neuronales en zonas en las que se ha perdido la conexión entre las neuronas contiguas. Este tipo de terapias hacen uso del análisis de la señal eléctrica producida por los grupo musculares que intervienen durante una contracción muscular.

El análisis de las señales mioeléctricas producidas por la contracción muscular del miembro pélvico, representa uno de los parámetros más importantes para determinar, dependiendo de la patología del paciente, el tipo de terapia más adecuada. Así mismo el análisis de dichas señales proporciona información precisa del estado en el que se encuentran los grupos musculares afectados de acuerdo con su actividad eléctrica a nivel celular. Para el análisis de estas señales se implementan circuitos especialmente diseñados para procesar la información contenida en ellas; de esta forma se obtiene un patrón característico con el cual se puede determinar el nivel de afección del miembro.

En este documento se describe el desarrollo de un módulo de adquisición y procesamiento de señales mioeléctricas que será implementado en un dispositivo de rehabilitación de miembro pélvico para pacientes que presentan hemiplejia. El módulo consiste en un circuito de preprocesamiento en donde se filtra la señal eléctrica obtenida por la polarización y despolarización de las membranas de las fibras musculares durante una contracción, se digitaliza utilizando un convertidor analógico digital y se transmite por medio de un microcontrolador a una computadora por puerto serial para su posterior procesamiento.

El circuito de procesamiento final que se diseñó y que se presenta en este documento ha atravezado

por una serie de modificaciones y etapas para conseguir una señal con las características suficientes y necesarias para su análisis posterior mediante el software especializado, el tratamiento de la señal se basa en un análisis en el dominio del tiempo en relación con una serie de muestras adquiridas en intervalos constantes.

*Los autores agradece el soporte económico recibido por la Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional (SIP-IPN), a través del proyecto 20131053, y de los programas EDI y COFAA del IPN, así como del Sistema Nacional de Investigación (SNI) del CONACYT.

1. Introducción

Las señales electromiográficas (EMG) superficiales de acuerdo con [1], son esencialmente un patrón unidimensional, por lo que cualquier técnica de procesamiento para la extracción de características y reconocimiento de patrones se puede aplicar a este tipo de señales. La necesidad de una rápida respuesta de los sistemas que implementan dichas señales limita la longitud de las muestras sobre las cuales se extraen las características produciendo una respuesta con un menor grado de precisión.

Las señales electromiográficas son colectadas típicamente mediante electrodos bipolares de superficie ubicados sobre la piel, estas señales proveen información sobre la actividad neuromuscular que la origina siendo ésta esencial en: diagnóstico clínico, rehabilitación, y como fuente de control para dispositivos activos y esquemas de estimulación eléctrica funcional. Estas señales son generadas por la contracción muscular, por lo que su adquisición requiere de una correcta identificación de las regiones musculares comprometidas en la ejecución de los movimientos a clasificar. Debido a la elevada resistencia eléctrica natural de la piel, se recomienda la aplicación de un gel que mejore la conductividad además de lograrse una buena superficie de contacto y adherencia con los electrodos. A pesar de estas disposiciones, las señales recogidas serán demasiado débiles, por lo que se hace necesario un procesamiento previo de amplificación y filtraje.

De acuerdo con [1], la amplitud típica de las señales electromiográficas es de 0-6mV por lo que se requiere de una etapa previa de amplificación que pueda ser persivida por instrumentos de medición como el osciloscopio y sistemas de procesamiento de señales. Esta etapa de preamplificación consiste en un amplificador diferencial de alta ganancia para evitar distorsiones de la infromación contenida en la señal. Una vez amplificada se debe considerar la eliminación de componentes de ruido de alta frecuencia y las provenientes de fuentes del entorno tales como la componente típica del ruido de baja frecuencia inducida por la red de distribución eléctrica de 60Hz, de igual forma se debe considerar el ancho de banda de la señal que caracteriza la contracción muscular, esté corresponde al intervalo entre los 50Hz y 150Hz aunque su canal de información típicamente se encuentra en el rango de 20Hz a 500Hz.

Estas señales al ser la superposición de una serie componentes a diferentes frecuencias en un ancho de banda específico pueden ser procesadas en una variedad de formas como se presenta en [2]. En trabajos como [3], se plantea el uso de un dispositivo de procesamiento de señales especializado para el control de un robot de dos grados de libertad, se determina la contracción muscular del miembro a partir de un método indirecto en el cual en base a parámetros estadísticos, se obtiene un valor umbral que identifica que se ha producido contracción muscular en secciones donde no es posible obtener la señal de forma directa.

En [4] se desarrolló un sistema similar al que se presenta en este trabajo, se muestra un circuito de preprocesamiento de la señal mioeléctrica de los músculos que consiste en una serie de filtros que limitan a un ancho de banda de 20 a 500Hz. La señal obtenida es digitalizada a partir del convertidor analógico digital de un microcontrolador de 16 bits, posteriormente es transmitida por medio de puerto serial a una computadora en donde es desplegada a través del software MATLAB.

En el presente trabajo se expone un procesamiento de la señal mioeléctrica en el dominio del tiempo, no se requiere de una transformación del espacio de estado debido a que el análisis temporal de la señal obtenida, proporciona información suficiente del estado de los grupos musculares de los cuales se obtiene

La poscición de los electrodos es muy importante en la recolección de registros EMG con propositos de control mioeléctrico ya que estos determinarán la eficiencia del dispositivo al cual se le asignan dichas

señales. Existe una gran variedad de posiciones en las cuales se puden colocar los electrodos superficiales para la adquisición de las señales mioeléctricas, sin embargo se deben considerar secciones musculares de las cuales se pueda extraer la mayor cantidad de información, dichas secciones musculares deben ser lo suficientemente extensas para la colocación de los electrodos sin que estos causen algún tipo de interferencia uno con el otro. Cabe señalar que para cada paciente la distribución de los electrodos correponderá a sus características tanto anatómicas como fisiológicas por lo que no se puede llegar a una

generalización en cuanto a la posición estandar para la adquisición de determinadas señales mioeléctricas.

El número de electrodos corresponderá al numero de canales a procesar y este es un parámetro escencial en el desempeño del clasificador. Se debe considerar el numero de canales con los que se trabaja ya que el procesamiento de las señal mioeléctrica de cada canal requerirá de cierto tiempo por lo que un mayor número de canales se traduce en un mayor tiempo de procesamiento y por lo tanto un mayor retardo en el despliegue de las señales, a su vez aumentará la interferencia producida por la interacción de canales contiguos debido a la distribución geométrica de los músculos y la propagación dispersa de la señal.

En la generación de señales EMG, durante la contracción muscular, se identifican dos fases o etapas:

1. El estado transitorio de la señal, que corresponde a los instantes de la rafaga de actividad mioeleléctrica que acompaña al esfuerzo muscular repentino en la ejecución del movimiento.

2. El estado estacionario, correspondiente al tiempo del esfuerzo muscular realizado durante un movimiento el cual es sostenido.

En la figura 1 se muestran los estados presentes durante la contracción muscular en ambos casos.

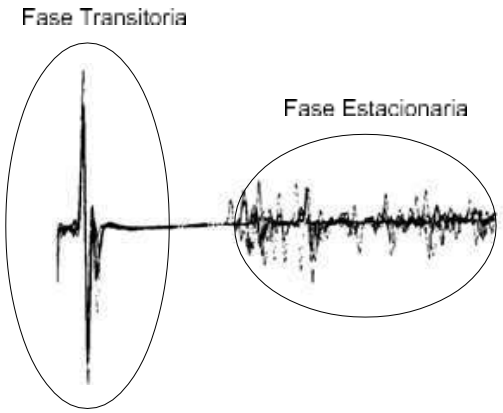


Figura 1: Fases durante la contracción del músculo

La fase inicial de contracción posee un gran contenido de información discriminante para determinado tipo de procesamiento; sin embargo debido a que representan un instante de la actividad mioeléctrica en un esfuerzo repentino no resulta útil cuando se requiere información bajo periodos en los cuales la contracción es sostenida. De este modo la fase estacionaria proporciona una mayor cantidad de información que puede ser procesada para dicho propósito.

La introducción del procesamiento digital en el análisis de las señales EMG sugiere su digitalización mediante un proceso de muestreo. La tasa de muestreo en la práctica puede ser de 1000Hz con el fin de registrar componentes de frecuencia en la señal de hasta 500Hz según el teorema de Nyquist, o muestrear a una frecuencia de 500Hz para registrar componentes de frecuencia de hasta 250Hz, Generalmente no se requieren de mayores tasas de muestreo debido a que la máxima concentración de energía de las señales electromiográficas se encuentra en el rango de 50 a 150Hz. De cualquier manera se debe considerar la

frecuencia de muestreo ya que incidirá directamente en la longitud de las muestras de la señal a procesar implicando mayor o menor tiempo de cómputo sobre cada registro de EMG y comprometiendo con ello el retardo de la respuesta del sistema.

La monitorización de las señales mioeléctricas para su estudio parte del hecho de que el valor instantáneo de las señales EMG no contiene información y según los modelos aceptados estas son

estocásticas, es este comportamiento aleatorio el que dificulta la extracción de patrones característicos dentro de la señal por lo que se recurren a métodos de análisis de señales más complejos para determinar características específicas que proporcionan más información al sistema. Por ser un dispositivo en el cual se obtiene información del estado de los músculos de secciones específicas la información más importante dentro de la señal que producen depende de la amplitud con respecto al tiempo.

En este artículo se presenta un método de caracterización de la señal mioeléctrica producto de la contracción de secciones musculares del miembro pélvico, este método consiste en un módulo de preprocesamiento con base en un conjunto de etapas acopladas, en la cuales, se filtra la señal, se digitaliza en forma discreta y se transmite a un sistema de visualización y procesamiento final. En esta última etapa de procesamiento se obtienen parámetros cuantificables de la señal original con los cuales se determina el nivel de contracción muscular asociado al movimiento del miembro pélvico.

El artículo se organiza de la siguiente forma: En la sección 2 se presenta la parte teórica-experimental del proyecto, se muestran los circuitos prototipo y final de la etapa de preprocesamiento de las señales mioeléctricas, se desarrollan los cálculos de los filtros involucrados y se presenta la simulación de su comportamiento bajo condiciones iniciales. Se muestran de igual forma imágenes de las señales obtenidas en cada una de las etapas que componen a dichos circuitos y su interpretación correspondiente. Se dan a conocer resultados parciales con respecto a los modelos planteados y se concluye con el desarrollo en PCB de un circuito de preprocesamiento final. De igual forma se realiza una descripción general del entorno de visualización programado haciendo referencia a un código desarrollado en Java.

En la sección 3 se describen los resultados experimentales finales del módulo de adquisición y su integración con el entorno de visualización, se da a conocer el método de procesamiento interno de la señal para la extracción de características con base en la contracción muscular.

En la sección 4 se presentan las conclusiones del trabajo, se señalan los alcances del proyecto y se describen aspectos generales de la realización del módulo.

2. Módulo de Adquisición y Procesamiento de Señales Mioeléctricas

Durante el diseño de la etapa de pre procesamiento de las señales electromiográficas se presentaron diversas dificultades en cuanto a la extracción de características que relacionaran la morfología de la señal con el nivel de contracción muscular, se llevó a cabo una serie de simulaciones y pruebas experimentales para determinar las etapas que conforman el circuito final. Una de las primeras aproximaciones que se diseñaron para el procesamiento se muestra en la figura 2.

Este circuito se compone de tres etapas bien definidas, una etapa de amplificación, una etapa de filtrado y una etapa de detección. La primera amplifica la señal proveniente de los electródos colocados en la sección muscular, puesto que el nivel de tensión inicial de estas señales no es lo suficientemente elevado para los requerimientos de los dispositivos electrónicos con los que se realiza el procesamiento. Inicialmente las señales electromiográficas presentan un efecto de montaje en una señal de DC generada por los amplificadores operacionales que se emplean, es preciso discriminar este efecto si se requiere encontrar una relación entre la contracción muscular y su interpretación. La etapa de filtrado se encarga de segregar esta señal de montaje a través de un filtro paso alto con una frecuencia de corte cercana a los

0Hz que corresponde a la frecuencia de oscilación de las señales de DC.

La naturaleza estocástica de las señales electromiográficas dificultan la identificación de un patrón característico en su morfología; sin embargo la euritmia de la amplitud de la señal con el nivel de contracción poseé un comportamiento con cierto grado de linealidad con el que fácilmente se puede trabajar para generar una señal de control. La última etapa de procesamiento utiliza este principio para

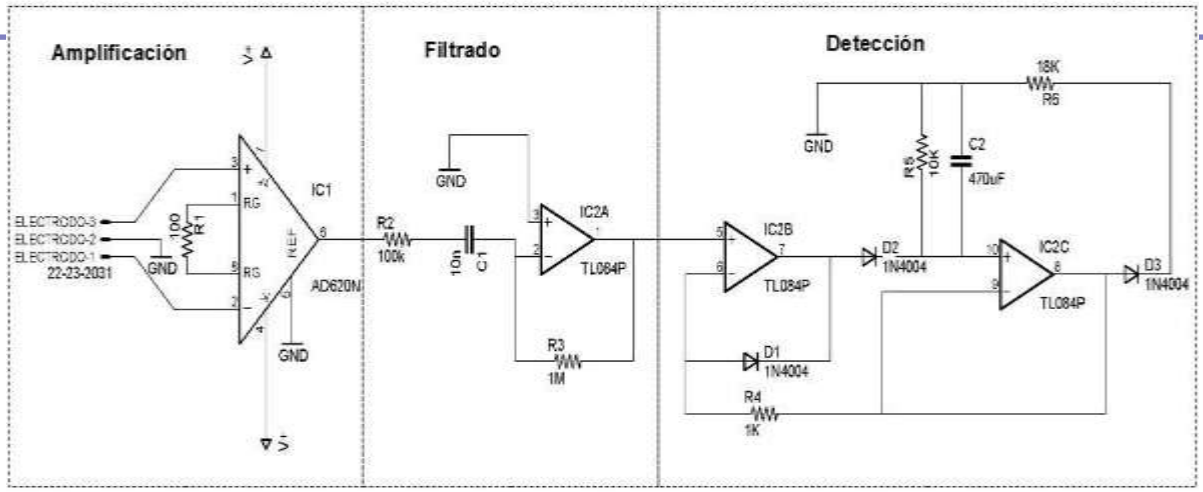


Figura 2: Primer circuito de preprocesamiento de la señal mioeléctrica

generar una señal de DC correspondiente, detecta los cambios instantáneos positivos de la señal y produce un nivel de tension a la misma amplitud; sin embargo puesto que el nivel de tensión proporcionado por este circuito depende de la carga de un capacitor, el comportamiento estocástico de la señal produce variaciones indeseables que pueden afectar el funcionamiento de la etapa de control del mecanismo.En las figuras 3, 4 y 5 se muestran las señales a la salida de cada etapa.

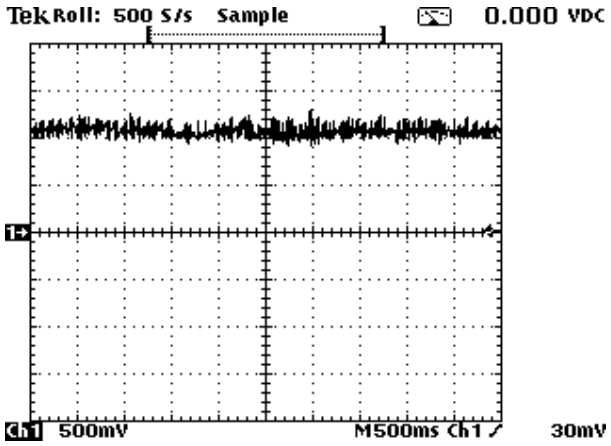


Figura 3: Señal obtenida de la salida de la etapa de amplificación

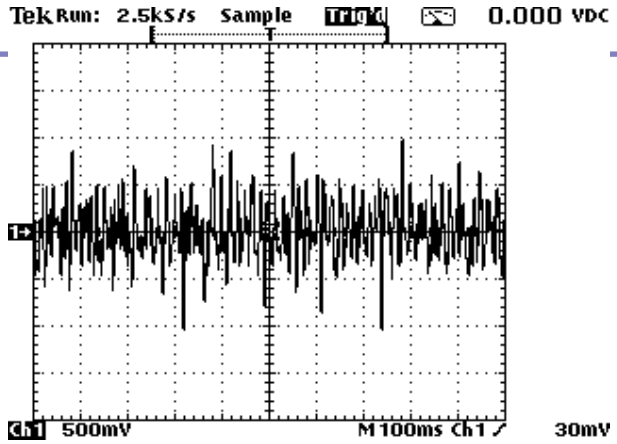


Figura 4: Señal obtenida de la etapa de filtraje

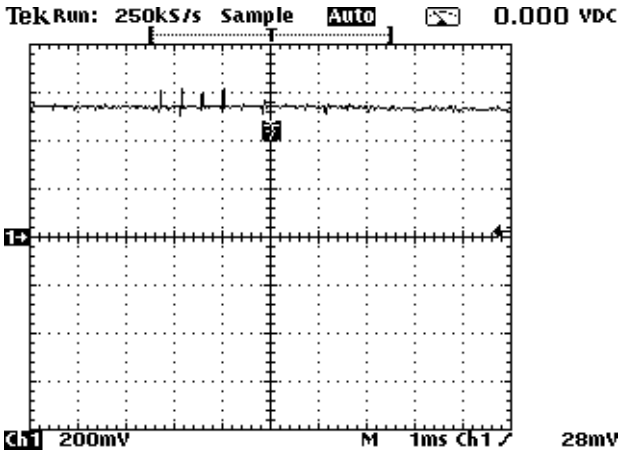


Figura 5: Señal obtenida de la salida de la etapa de detección

Con este primer circuito se consigió obtener una señal que nos permite relacionar la amplitud de la señal con el nivel de contracción muscular; sin embargo como se observa en la figura 5, existen variaciones que se acentúan en relación al tiempo de contracción produciendo efectos inesperados y variaciones en la amplitud de la señal de control producto del ruido inducido por fuentes externas. Por esta razón el circuito de la figura 2 fue modificado introduciendo dos etapas de filtrado más, asegurando la integridad de la señal obtenida, el circuito se muestra en la figura 6.

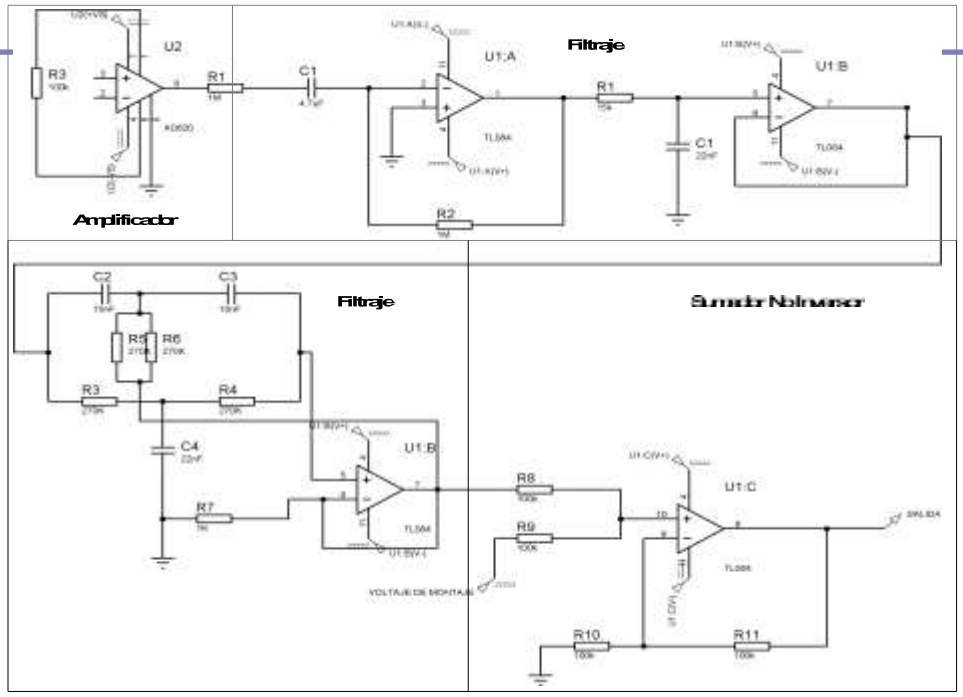


Figura 6: Circuito de Preprocesamiento de Señales Mioeléctricas Modificado

Este circuito implementa una etapa de amplifiación como en el caso anterior, tres etapas de filtrado para limitar el ancho de banda de la señal obtenida y una etapa de montaje final que proporciona las características necesarias para la digitalización de la señal para su manipulación en un sistema de cómputo

En la figura 6 se pueden observar etapas complementarias que mejoran en el desempeño del circuito de la figura 2. El filtro pasa altas y el filtro pasa bajas limitan la respuesta del circuito a un rango de frecuencias comprendida entre los $0Hz$ y $500Hz$ donde se concentra la mayor cantidad de energía de la señal y por lo tanto la mayor cantidad de información relacionada con la contracción muscular. Para el filtro pasa altas se calcula la frecuencia de corte de acuerdo con la siguiente ecuación:

$$f_c = \frac{1}{2\pi R_1 C_1}$$

en donde:

f_c: Frecuencia de Corte del Filtro
R_1: Resistencia de Entrada
C_1 : Capacitor de Entrada

Una aproximación a una frecuancia de corte cercana a $0Hz$ se puede conseguir utilizando los siguientes componentes:

$R_1 = 1M\,\Omega$
$R_2 = 1M\,\Omega$
$C_1 = 4{,}7\mu F$

La frecuencia de corte corresponde a:

$$f_c = \frac{1}{2\pi(1M\,\Omega)(4{,}7\mu F)} = 0{,}03Hz$$

Se realizó la simulación del circuito con una entrada de tipo senoidal con una amplitud unitaria y frecuencia de 1kHz, en el software Proteus antes de su implementación en el circuito físico y se obtuvieron los siguientes resultados (figura 7).

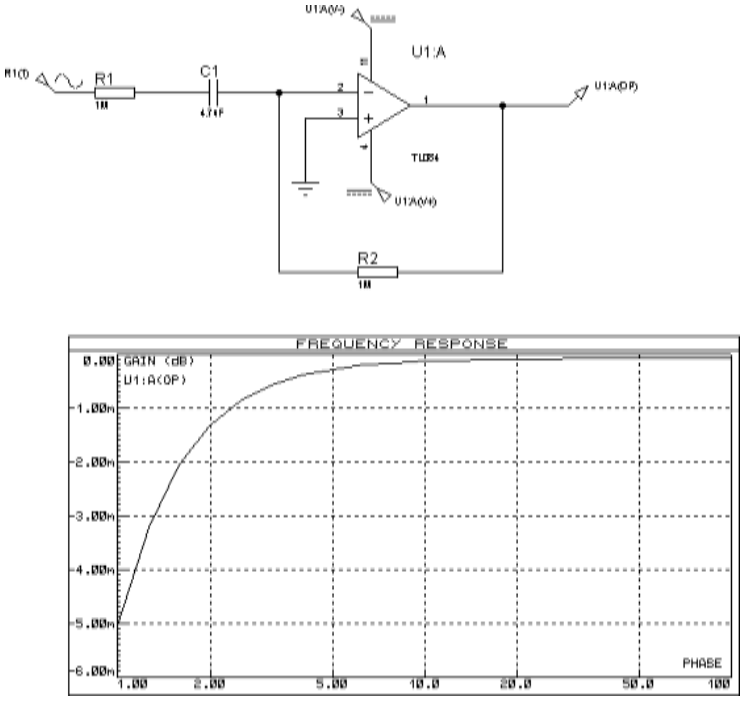


Figura 7: Diagrama esquemático y simulación del Filtro pasa Altas.

Para el filtro pasa bajas se sigue el mismo procedimiento; sin embargo en este caso se considera una frecuencia de corte cercana a 500Hz, de acuerdo a la ecuación:

$$f_c = \frac{1}{2\pi R_1 C_1}$$

$$f_c = \frac{1}{2\pi(15k\Omega)(22nF)} = 482{,}28Hz$$

Como en el caso del filtro pasa altas, se realizo la simulación del filtro pasa bajas antes de ser implementado de forma física y se obtuvieron los siguientes resultados (figura 8).

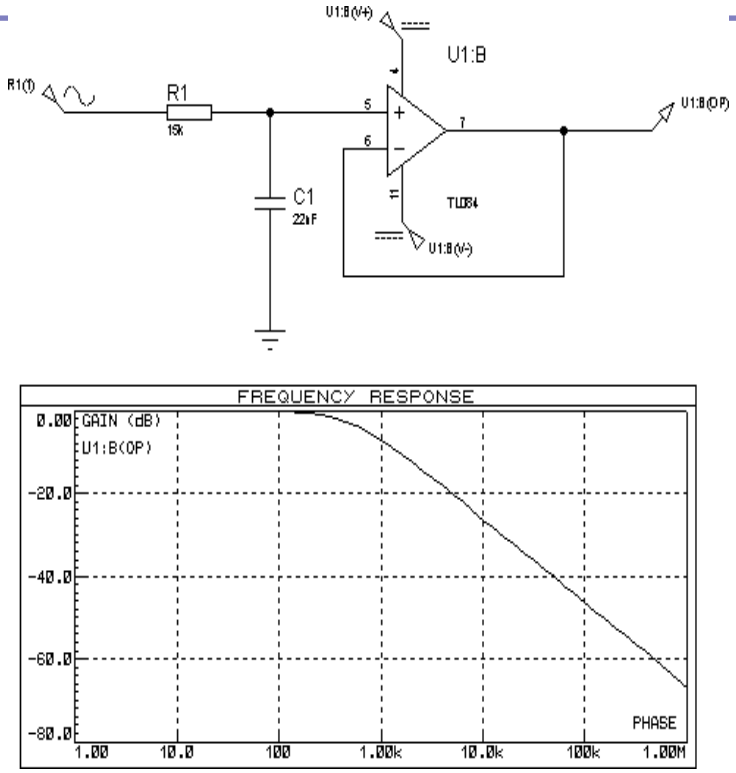


Figura 8: Simulación del filtro Pasa Bajas

Con la aplicación de estos dos filtros se realiza un proceso de discriminación de la información contenida dentro de la señal, haciendo más sencilla la identificación de características. Por otro lado aún dentro de este ancho de banda se presentan componentes de ruido de fuentes externas que pueden llegar a producir variaciones inesperadas una vez que se introduzca la señal en el proceso de control, por ello es indispensable minimizar las posibles perturbaciones que produzcan estos efectos. La ultima etapa de filtraje de la señal comprende un filtro rechaza bandas de $60Hz$, esta alteración es de tipo electromagnético y es producto de la oscilación de la señal de las fuentes de energía eléctrica. Al igual que en los filtros anteriores se realizó la simulación del circuito y se obtuvieron los siguientes resultados (figura 9).

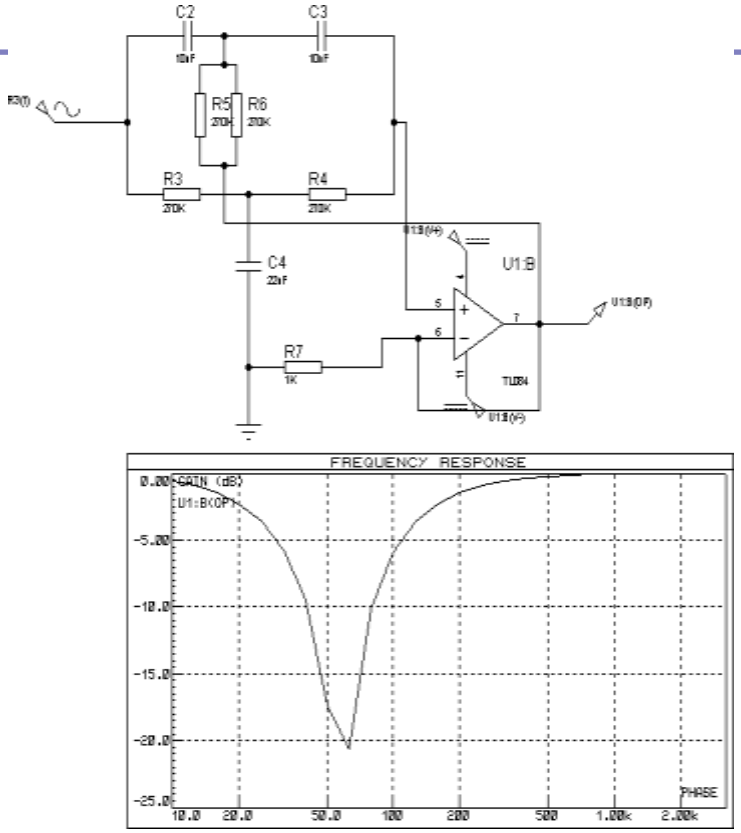


Figura 9: Simulación del Filtro Rechaza Bandas de 60Hz

La última sección del circuito representa la etapa de acondicionamiento de la señal para su digitalización, se manipula con la finalidad de que cumpla con los requerimiento necesarios para su procesamiento por medio de un microcontrolador. La señal es montada una vez más en una señal de DC, con la única diferencia de que este valor permanece constante independientemente del intervalo de tiempo y amplitud de la señal, de este modo las variaciones producidas por la contracción muscular se concentran en un rango de valores predeteminado, en este caso de 0 a 5v. Esta señal se introduce en un canal de conversión Analógico - Digital del microcontrolador PIC16F877A para su digitalización y transmisión por medio del puerto serial hacia una computadora, donde posteriormente se procesarán los datos y se determinará el nivel de contracción muscular con base en un algoritmo en donde se calcula el valor promedio de la señal en intervalos de tiempo constantes. En la figura 10 se muestra el circuito final de adquisición, procesamiento y transmisión.

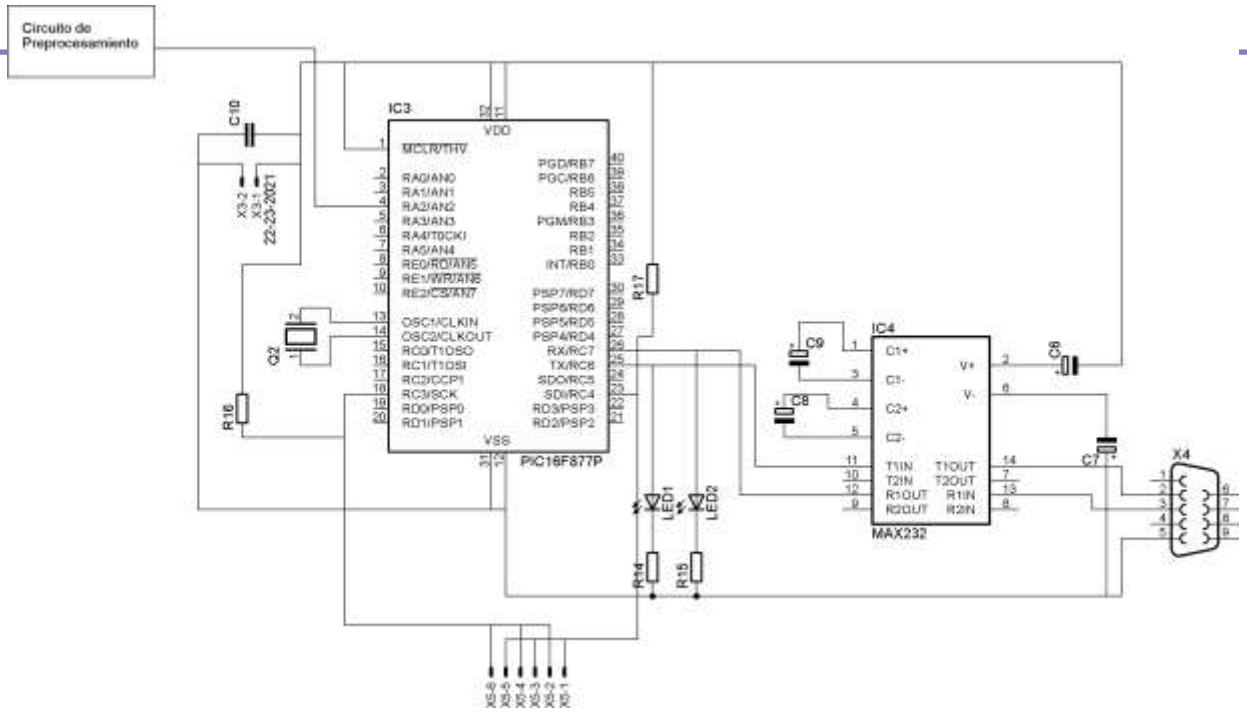


Figura 10: Circuito de Digitalización y Transmisión de la señal Mioeléctrica

Se realizó un circuito de experimentación inicial, en donde se obtuvieron los primeros resultados acerca de la calidad de la señal filtrada y el proceso de digitalización (figura 11)

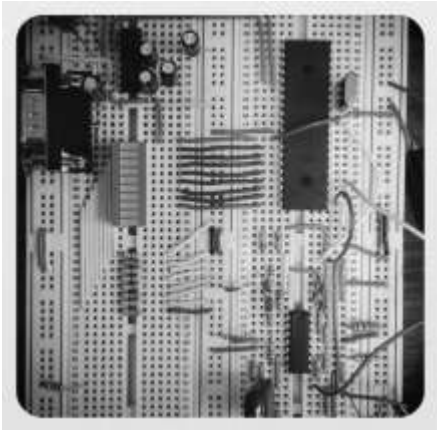


Figura 11: Circuito de experimentación

Posterior al diseño experimental, una vez que se obtuvieron las señales correctas se desarrolló la placa PCB en el software Eagle Cadsoft (figura 12) obteniendo el circuito impreso para realizar pruebas de funcionamiento (figura 13).

Para la digitalización de la señal mioeléctrica procesada se desarrolló un programa en el software MicroCode Studio Plus para el PIC16F877A en donde se implementa el Convertidor A/D y el protocolo de comunicación Serial (vease apendice A). En este código se configuran los registros del convertidor A/D y de la comunicación Serial, posteriormente se realizan lecturas simultáneas y se transmite cada dato

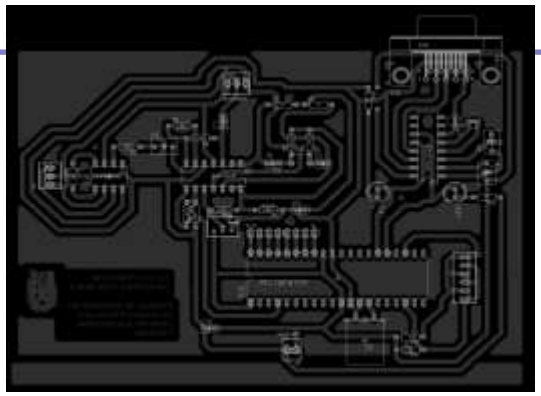


Figura 12: Diseño PCB del circuito de experimentación en Eagle

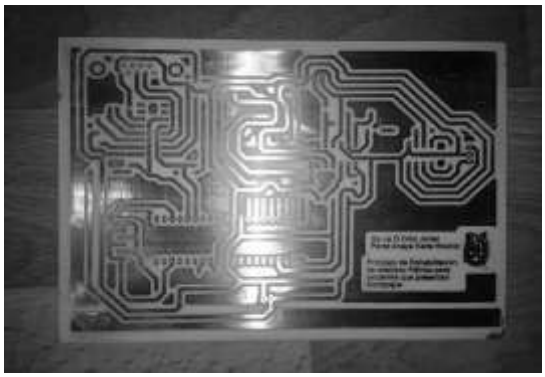


Figura 13: Fotografía del Modulo de Preprocesamiento

a la computadora para su posterior procesamiento. Se toman en cuenta los parámetros de configuración para la comuniación serial ya que estos mismos parámetros deben ser configurados en la interfaz gráfica de usuario para una correcta sincronización de hardware y software.

La interfaz gráfica de usuario se diseñó en el software de prueba Netbeans IDE basado en código Java. El programa consta de un formulario de registro en donde se introduce datos principales del paciente que se almacenan dentro del programa para generar un panel de información visible que ayude al terapeuta a llevar un control de las sesiones de rehabilitación que se están realizando, de igual forma la interfaz cuenta con un panel de visualización de la actividad muscular del miembro pélvico que se encuentra bajo terapia, alimentado por los datos obtenidos del módulo de procesamiento, de esta forma el terapeuta puede observar en todo momento la actividad eléctrica de los músculos involucrados en la contracción y detectar posibles anomalias producto de la patología que se está tratando y a su vez observar la respuesta del paciente frente a diversas terapias para seleccionar la más adecuada. La interfaz gráfica de usuario esta confromada por una panel de configuración y un panel de visualización, el primer panel transmite los parámetros configurados al panel secundario generado una ventana de visualización y control.

La ventana primaria o ventana transmisora que se muestra en la figura 14 se constituye de tres páneles, el primer panel está dedicado a la recepción de los datos personales del paciente tales como: nombre, apellidos, sexo, edad, patología, número de terapias programadas, observaciones importantes que el médico o terapeuta deben considerar para la ejecución de los ejercicios de rehabilitación y además

le permite introducir una fotografía del paciente para asegurar que la próxima vez que se someta a una terapia se seleccione la ficha de registro correcta evitando posibles confusiones. El segundo panel contiene los campos de configuración del protocolo de comuniación serial, permite seleccionar el puerto y la velocidad de transmisión de los datos obtenidos del módulo de procesamiento de las señales

mioeléctricas. El tercer panel constituye la configuración del tiempo de ejecución de las sesiones de rehabilitación, el médico o terapeuta introduce el tiempo en minuto de la duración de los ejercicios en cada terapia. Para que todos los parámetros configurados tengan un efecto sobre el panel secundario todos los campos de texto y de selección se relacionan con el estado de tres botones: un botón de configuración, un botón de limpieza y un botón de consulta. El botón de configuración almacena los datos capturados en los campos de texto y en los campos de selección y los transmite al panel secundario produciendo la ventana de visualización. El botón de limpieza borra toda la información contenida en los campos y los prepara para recibir la nueva información. El botón de consulta compara la información contenida en el campo del nombre del paciente con la información previamente almacenda, si encuentra alguna coincidencia rellena automáticamente los campos en blanco con la información correspondiente a dicho paciente. En la figura 14 se muestra la ventana primaria desarrollado en Netbeans y el código se presentan en el apéndice B.

Figura 14: Interfaz Gráfica de Usuario Primaria desarrollada en Netbeans.

La ventana secundaria que se muestra en la figura 15, se encarga de recibir los parámetros de configuración y la información de los campos para generar dos paneles, un panel de visualización personal (Panel IV)y un panel de visualización de la actividad eléctrica del músculo (Panel V). El panel IV personal consulta la información almacenada del paciente y la muestra en un formato en donde tambien

se puede visualizar su fotografía, debajo de este panel se encuentra una plantilla de graficación con ejes coordenados X y Y, el eje de las abscisas (eje X) contiene la información del tiempo transcurrido en milisegundos y el eje de las ordenadas (eje Y) contiene los datos recibidos desde el módulo de procesamiento de las señales mioeléctricas. Ambos ejes son autoescalables los que permite observar en forma dinámica los datos que son recibidos por la contracción muscular del miembro. En la figura 15

se muestra la ventana secundaria desarrollada en Netbeans y el código se muestra en el apéndice C.

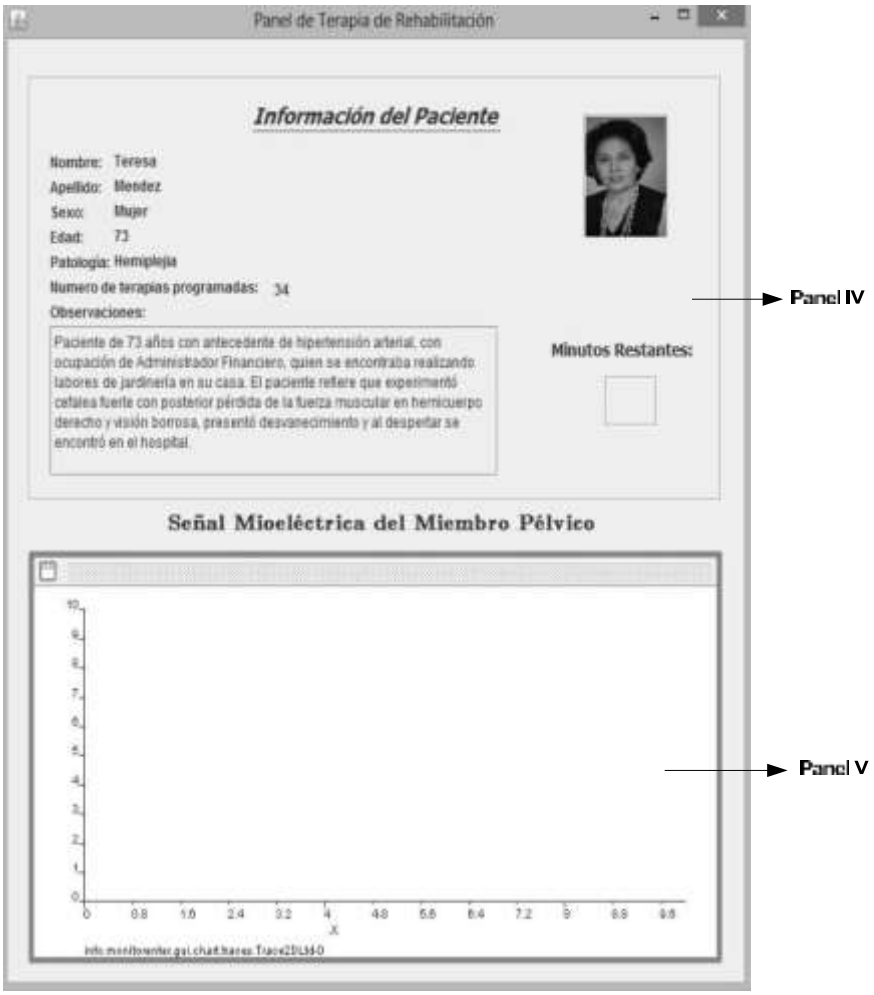


Figura 15: Interfaz Gráfica de Visualización desarrollada en Netbeans

La ventana primaria y la ventana secundaria se relacionan con un programa principal que se muestra en el apéndice en donde se encuentran funciones que se utilizan a lo largo del programa, éste se encarga de coordinar las acciones de cada panel recibiendo y transmitiendo información de forma continua y controlando el flujo del programa. En el programa principal se declaran las librerias que los páneles utilizan para establecer los parámetros de configuración de la comuniación serial, la reproducción de la plantilla de graficación y las plantillas de los páneles, estas directivas son de suma importancia ya que representan secciones de código especializadas que facilitan la implementación de acciones en el entorno de desarrollo.

Dentro del programa principal se realiza un procesamiento con base en una muestra finita de la señal para determianar el nivel de contracción que se ejerce durante un ejercicio. A la muestra de la señal se le aplica un algoritmo que calcula su valor promedio basado en el numero de datos procesados y el valor

que representa cada uno de estos de acuerdo con la siguiente ecuación.

$$RMS = \sqrt{\frac{1}{N}\sum_{i=0}^{N}[x(i)^2]}$$

En donde:

$x(i)$: Es la i-esima muestra de la señal de entrada

N : Es el número de muestras.

RMS : Es el valor promedio de la señal

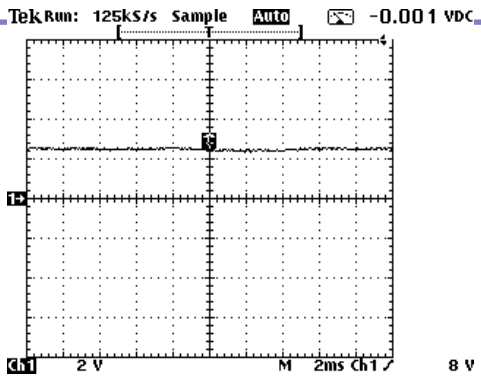
Este algoritmo se basa en el procesamiento de la señal mioeléctrica en el dominio del tiempo y produce una respuesta lineal en relación al nivel de contracción muscular. De esta forma podemos segmentar esta respuesta para fijar rangos específicos relacionados con la amplitud de la señal original, con base en ésto se crea una señal de control que es transmitida al módulo central de procesamiento y direccionada al modulo de control para el movimiento del prototipo de rehabilitación.

3. Resultados

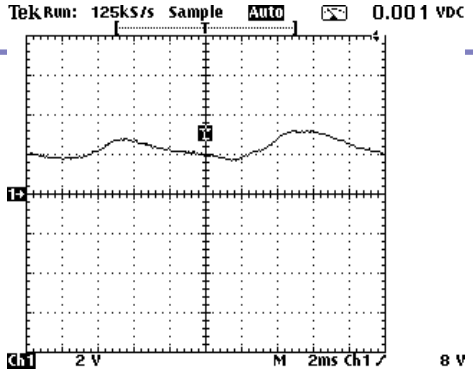
Con el circuito de procesamiento final de la figura 6 se logró conseguir una señal con la mínima cantidad de ruido inducido por fuentes externas, lo que facilita su posterior implementación en la etapa de control del dispositivo mecánico que proporcionará las terapias de rehabilitación. En la figura 16 se muestra la señal pre procesada por el módulo que se desarrolló. Se observa que las caracterrísticas morfológicas de la señal nos permiten interpretarla para su posterior digitalización ya que cumple con los requerimientos del convertidor A/D del PIC, como lo son: la amplitud mínima y máxima de la señal.

Como se observa en las imágenes de la figura 16 la señal mioeléctrica no presenta componentes de ruido inducido y su morfología correspone a la naturaleza estocástica producto del potencial de acción de los paquetes de células musculares que se relacionan con la contracción muscular durante la contracción, los valores que proporciona se encuentran en una rango de 0 a $5v$ por lo que la señal puede ser introducida directamente al microcontrolador para su digitalización.

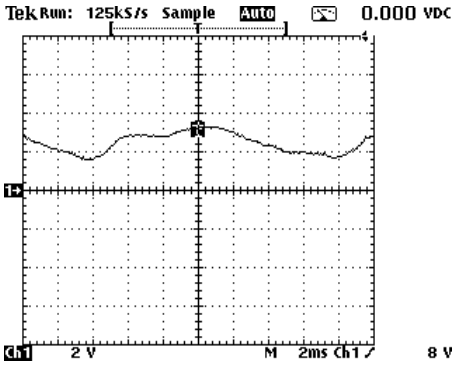
El cuadro superior izquierdo de la figura 16 muestra un nivel de tensión constante aproximadamente a $2{,}5v$ que representa la señal eléctrica del músculo en la etapa de relajación, este efecto se produce ya que la señal previamente fue montada sobre una señal de DC, en este caso las variaciones por debajo de este nivel se consideren positivas y pueden ser procesadas por el microcontrolador sin ningún problema a diferencia de que la señal no presentara este montaje las variaciones por debajo de un nivel de tension a $0v$ se considerarían negativas produciendo resultados inesperados durante la conversión producto de estos valores fuera del rango de conversion (0 a $5v$). En los cuadros superio derecho e inferior izquierdo de la figura se muestra la señal eléctrica del músculo en la etapa de contracción sostenida, se puede observar una mayor calidad en cuanto a su morfología y su caracterización. Finalmente en el cuadro inferior derecho de la figura se muestra la señal electrica del músculo en la etapa de contracción sostenida pero a una escala de visualización diferente con el objetivo de remarcar los valores máximos que puede tomar la señal tanto en el limite superior como en el inferior.



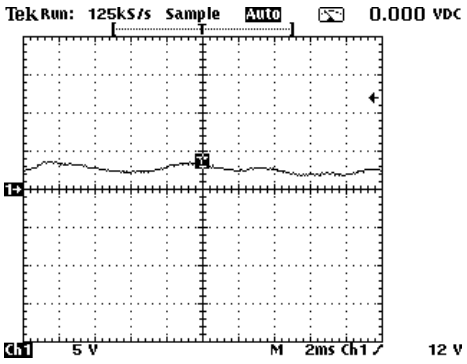
Señal del Musculo sin contracción.



Señal del músculo en contracción N^{o} 1



Señal del músculo en contracción Nº2



Señal del músculo en contracción escala.

Figura 16: SeñalesEléctricas del Músculo en diferentes Estados

Una vez que la señal ha sido digitalizada se introduce a la computadora que recibe los datos a través del puerto serie, estos datos son graficados uno a uno en el panel de visualización relacionando el tiempo de ejecución del programa con el valor de la señal digitalizada en dicho momento. Como se observa en la figura 17 la señal permanece en un pequeño rango de valores cercanos a cero, esta etapa representa la señal mioeléctrica del paciente con el músculo relajado, posteriormente se observa un aumento en la amplitud de la señal en determinados intervalos de tiempo, esta etapa representa la señal mioeléctrica del paciente en contracción muscular sostenida. La señal es graficada en un panel con ejes autoescalables lo que permite visualizar el cambio en la respuesta del paciente sin importar la duración de la terapia, los ejes se ajustan automáticamente desplazandose conforme el programa siga en ejecución.

Figura 17: Señal mioeléctrica del Paciente durante una sesión de terapia

4. Conclusión

El modulo de adquisición y procesamiento de señales mioeléctricas es un dispositivo de monitoreo que permite a los usuarios observar la actividad eléctrica de las secciones musculares específicas, así mismo proporciona la información necesaria para que el médico o terapeuta mantega un control preciso durante las sesiones de terapia.

La etapa de preprocesamiento de la señal mioeléctrica se implementa por medio de hardware, es filtrada y acondicionada para su posterior digitalización por medio del convertido analógico digital del microcontrolador. La señal procesada es tratada internamente via software, en donde se toma una muestra de la señal y se genera una patrón lineal relacionado con el nivel de contracción muscular. Este tipo de procesamiento se centra en la señal original sin ninguna transformación, esto quiere decir que la señal es procesada en el dominio del tiempo ya que para la aplicación que se está desarrollando la señal característica resultado de movimientos específicos de cualquiera de los miembros pélvicos es irrelevante. El valor RMS que se calcula para las diversas muestras de la señal en el tiempo, es un parámetro que nos proporciona información suficiente para conocer el nivel de contracción muscular, independientemente de que para su implementación sea necesario una cantidad de muestras lo suficientemente amplia para su procesamiento, comprometiendo en cierta forma el tiempo de respuesta del mecanismo con respecto al

tiempo de cálculo del algoritmo; sin embargo este factor no influye en el flujo de la señal de control.

Referencias

[1] H. A. Romo and J. C. Realpe, "Surface emg signals analysis and its applications in hand prosthesis control," *Universidad de Cauca*, vol. 125, no. 1, pp. 127–136, 2007.

[2] D. Stashuk, "Emg signal decomposition: how can it be accomplished and used," *Journal of Electromyography and Kinesiology*, vol. 11, no. 1, pp. 151–173, 2001.

[3] O. Eugenio, N. López, and C. Soria, "Procesamiento de señales mioeléctricas implementado en procesador digital de señales."

[4] M. N. López, V. Toranzos, and O. G. Lombardero, "Sistema de adquisición y visualización de señales mioeléctricas," *Ingeniería Clínica Mar del Plata*, vol. 1, no. 1, pp. 1–6, Septiembre 2011.

A. Código de Programación del Microcontrolador PIC16F877A

```
'*************************************************************************
'* Name : CONVERTIDOR ANALOGICO DIGITAL PARA ADQUISICION DE *
'* SEÑALES MIOELECTRICA *
'* Author : JAVIER DE LA O ORTIZ *
'* Date : 14/01/2013 *
'* Version : 1.0 *
'* Notes : ADQUISICION DE SEÑALES MIOELECTRICAS A PARTIR *
'* DEL CIRCUITO DE LA CARPETA DIGITALIZADOR DENTRO *
'* DE LA CARPETA MIS DOCUMENTOS CON EL TITULO ETAPA *
'* DE ACONDICIONAMIENTO DE LA SEÑAL *
'* *
'*************************************************************************
'Decalración de los defines que seran utilizados a lo largo del programa
'*************************************************************************
INCLUDE ̈MODEDEFS.BAS'¨Incluyen los modos de comunicación
DEFINE CHAR_PACING 1000;
'*************************************************************************
'Configuración del los puertos del microcontrolador. Se habilita el puerto AN0
'como entrada analógica para la conversión.
'*************************************************************************
'* ADCON0: ADCS1(SR) / ADCS0(SR) / CHS2(SC) / CHS1(SC) / CHS0(SC) / GO-DONE(BS) / RESERVADO / ADON(CE)/
'* ADCON1: ADFM(SF) / ADCS2(SR) / RESERVADO / RESERVADO / PCFG3(CP) / PCFG2(CP) / PCFG1(CP) / PCFG0(CP)/
'* SR: Selección del la frecuencia de reloj del ADC.
'* SC: Seleeción del canal con el que va a trabajar el ADC.
'* BS: Bit de Status del ADC.
'* CE: Bit de encendido del convertidor.
```

```
'* SF: Selección de justificacion del formato de la conversion (1=derecha,0=izquierda)
'* CP: Control de configuración de los puertos para el ADC
TRISA = %00000100; Bit 2 del puerto PORTA como salida.
TRISB = %00000000; Puerto PORTB como salida.
PORTB = %00000000; Puerto PORTB en estado bajo inicial.
```

```
ADCON0 = %01010101; Frecuencia del reloj interno, canal AN0, convertidor encendido
ADCON1 = %10000010; Justificacion derecha, AN0 como entrada analogica.
PAUSE 200;
RESULT VAR BYTE 'Variable de almacenamiento de la conversion A/D
'*****************************************************************************
'*Programa principal
'*****************************************************************************
LOOP:
    ADCON0 = %01010101; 'Configuración inicial del A/D
    RESULT = ADRESL; 'Se carga el valor del convertidor a la variable RESULT
    SEROUT PORTC.6, T9600, [#RESULT]; 'Se envía por puerto serie el dato almacenado
    PAUSE 10; Se esperan 10 milisegundo
    GOTO LOOP; Se cicla la conversion para que se realice continuamente
END 'Fin del programa
```

B. Programa en Netbeans del Formulario

```
package formulariopaciente;
import java.awt.Image;
import java.io.FileReader;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
public class Formulario extends javax.swing.JFrame {

 private static int ciclo;
 private static int nv = 0;
 private static String camponombre;
 public static String ruta = ;

 public Formulario() {
 initComponents();
 }
public void grupoboton1(){
 grupoboton1.add(hombre);
 grupoboton1.add(mujer);
}

 private void hombreActionPerformed(java.awt.event.ActionEvent evt) {
 hombre.setEnabled(true);
 mujer.setRolloverEnabled(false);
 }
```

```
private void caja1ActionPerformed(java.awt.event.ActionEvent evt) {
}
private void BorrarActionPerformed(java.awt.event.ActionEvent evt) {
jTextField1.setText();
```

```
jTextField2.setText();
jTextField3.setText();
jTextField4.setText();
jTextField5.setText();
grupoboton1.clearSelection();
}
private void consultarActionPerformed(java.awt.event.ActionEvent evt) {
String r = jTextField1.getText();
Principal.Consulta(r);
}
private void InicioActionPerformed(java.awt.event.ActionEvent evt) {
String a = jTextField1.getText();
String b = jTextField2.getText();
String c = jTextField3.getText();
String l = jTextField4.getText();
String t = jTextField5.getText();
String text = observaciones.getText();
Icon ic = fotopaciente.getIcon();
String tmt = tiempo.getText();
long ns = 2;
if(hombre.isSelected()){
ns = 0;
}
if(mujer.isSelected()){
ns = 1;
}
long d = Long.parseLong(c);
long h = Long.parseLong(t);
Principal.Guardar(a, b, ns, d, l, h);
jTextField1.setText();
jTextField2.setText();
jTextField3.setText();
jTextField4.setText();
jTextField5.setText();
grupoboton1.clearSelection();
String ps = (String) caja1.getSelectedItem();
String bdr = baudrate.getText();
try {
Principal.Configuracion(ps, bdr,a, b, c, l, t, ns, text, ic, tmt);
} catch (Exception ex) {
}
Formulario.this.dispose();
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
JFileChooser foto = new JFileChooser();
int opcion = foto.showOpenDialog(this);
ruta = foto.getSelectedFile().getPath();
try {
FileReader fotoselec = new FileReader(ruta);
```

```
ImageIcon bft = new ImageIcon(ruta);
Icon icono = new ImageIcon(bft.getImage().getScaledInstance(fotopaciente.getWidth(),
fotopaciente.getHeight(), Image.SCALE_DEFAULT));
this.fotopaciente.setIcon(icono);
this.repaint();
} catch (Exception ex) {
}
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
ruta = ;
this.fotopaciente.setIcon(null);
}
private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}
public void completardatos(Informacion e){
this.jTextField2.setText(e.getApellido());
this.jTextField3.setText(Long.toString(e.getNidentificacion()));
this.jTextField4.setText(e.getPatologia());
this.jTextField5.setText(Long.toString(e.getNterapias()));
if(e.getSexo()==0){
this.hombre.setSelected(true);
}
if(e.getSexo()==1){
this.mujer.setSelected(true);
}
}

public void nocompletados(String g){
this.jTextField1.setText(g);
this.jTextField2.setText(g);
this.jTextField3.setText(g);
this.jTextField4.setText(g);
this.jTextField5.setText(g);
grupoboton1.clearSelection();
}

public void enviodedatos(String dt){
}

public static void main(String args[]) {
try {
for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
if ("Nimbus".equals(info.getName())) {
```

```
javax.swing.UIManager.setLookAndFeel(info.getClassName());
break;
}
}
} catch (ClassNotFoundException ex) {
```

```
    }
    public void run() {
    new Formulario().setVisible(true);
    }
    });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton Borrar;
    private javax.swing.JButton Inicio;
    private javax.swing.JLabel apellido;
    private javax.swing.JTextField baudrate;
    private javax.swing.JComboBox caja1;
    private javax.swing.JButton consultar;
    public javax.swing.JLabel fotopaciente;
    private javax.swing.ButtonGroup grupoboton1;
    private javax.swing.JRadioButton hombre;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JRadioButton mujer;
    private javax.swing.JLabel nombre;
    private javax.swing.JLabel numeroi;
    public javax.swing.JTextPane observaciones;
    private javax.swing.JLabel puerto;
    private javax.swing.JLabel tasa;
    public javax.swing.JTextField tiempo;
```

```
// End of variables declaration
}
```

C. Programa en Netbeans del Panel de Visualización.

```
package formulariopaciente;
public class Progreso extends javax.swing.JFrame {
 public Progreso() {
 initComponents();
 }
 @SuppressWarnings(ünchecked")
 private void initComponents() {
 }
 public static void main(String args[]) {
 java.awt.EventQueue.invokeLater(new Runnable() {
 public void run() {
 new Progreso().setVisible(true);
 }
 });
 }
 // Variables declaration - do not modify
 public javax.swing.JLabel espacio1;
 public javax.swing.JLabel espacio2;
 public javax.swing.JLabel espacio3;
 public javax.swing.JLabel espacio4;
 public javax.swing.JLabel espacio5;
 public javax.swing.JLabel espacios;
 public javax.swing.JLabel fotopas;
 private javax.swing.JDesktopPane jDesktopPane1;
 private javax.swing.JLabel jLabel1;
 private javax.swing.JLabel jLabel10;
 private javax.swing.JLabel jLabel11;
 private javax.swing.JLabel jLabel2;
 private javax.swing.JLabel jLabel3;
 private javax.swing.JLabel jLabel4;
 private javax.swing.JLabel jLabel5;
 private javax.swing.JLabel jLabel6;
 private javax.swing.JLabel jLabel7;
 private javax.swing.JLabel jLabel8;
 private javax.swing.JLabel jLabel9;
 private javax.swing.JPanel jPanel1;
 private javax.swing.JScrollPane jScrollPane1;
 public javax.swing.JTextPane obs;
 public javax.swing.JPanel panelchart;
 public javax.swing.JLabel restantes;
 // End of variables declaration
}
```

D. Programa Principal de la Interfaz Gráfica.

```
package formulariopaciente;
```

```
import giovynet.serial.Baud;
import giovynet.serial.Com;
import giovynet.serial.Parameters;
import info.monitorenter.gui.chart.Chart2D;
import info.monitorenter.gui.chart.ITrace2D;
import info.monitorenter.gui.chart.traces.Trace2DLtd;
import java.awt.Color;
import java.util.Timer;
import java.util.TimerTask;
import javax.swing.Icon;
import javax.swing.JInternalFrame;
/**
 *
 * @author Javier
 */
public class Principal {

 private static Formulario miFormulario;
 public static Progreso ventana2;
 private static Informacion Info[];
 private static datolectura matriz[];
 private static int duracion;
 private static int contador;
 private static int fallo;
 private static double cnt = 5;
 private static long m_starttime = System.currentTimeMillis();

 public static void main(String[] args)throws Exception{
 miFormulario = new Formulario();
 Info = new Informacion[10];
 matriz = new datolectura[1000];
 for (int i = 0; i<=9; i++){
 Info[i] = new Informacion();
 }
 for (int j = 0; j<=99; j++){
 matriz[j] = new datolectura();
 }
 duracion = 0;
 contador = 0;
 fallo = 0;
 miFormulario.setVisible(true);
 }
 public static void Guardar(String n, String s, long sx, long x, String pt, long nt){
 Info[contador].setNombre(n);
```

```
Info[contador].setApellido(s);
Info[contador].setNidentificacion(x);
Info[contador].setPatologia(pt);
Info[contador].setNterapias(nt);
Info[contador].setSexo(sx);
```

```
contador++;
}

public static void Consulta(String q){
for(int u = 0; u<Info.length; u++){
if (Info[u].getNombre().equals(q)){
miFormulario.completardatos(Info[u]);
fallo++;
}
}
if(fallo == 0){
miFormulario.nocompletados("PACIENTE NO REGISTRADO");
}
else{
fallo = 0;
}
}

public static void Configuracion(String npt, String bdr, String nmb, String apl, String id, String pt, String ndt,
long sxo, String desc, Icon ftp, final String tdt)
throws Exception{

Chart2D chart = new Chart2D();
final ITrace2D trace = new Trace2DLtd(5000);
trace.setColor(Color.blue);
chart.addTrace(trace);
final Progreso ventananueva = new Progreso();
ventananueva.setVisible(true);
if(sxo == 0){
ventananueva.espacios.setText("Hombre");
}
if(sxo == 1){
ventananueva.espacios.setText("Mujer");
}
ventananueva.espacio1.setText(nmb);
ventananueva.espacio2.setText(apl);
ventananueva.espacio3.setText(id);
ventananueva.espacio4.setText(pt);
ventananueva.espacio5.setText(ndt);
ventananueva.obs.setText(desc);
ventananueva.fotopas.setIcon(ftp);
JInternalFrame finterno = new JInternalFrame();
finterno.setVisible(true);
finterno.add(chart);
```

```
finterno.setSize(650,330);
ventananueva.panelchart.add(finterno);
Parameters param = new Parameters();
param.setPort(npt);
param.setBaudRate(Baud.valueOf(bdr));
```

```
param.setParity("N");
final Com mySP = new Com(param);
TimerTask timerTask = new TimerTask(){
String dato = ;
double numero = 0;
double suma = 0;
double array [] = new double [50];
int muestra = 0;
double aux = 0;
private long m_starttime = System.currentTimeMillis();
@Override
public void run() {
try {
dato = mySP.receiveToString(3);
numero = Double.parseDouble(dato);
numero = numero - 512;
array[muestra] = Math.pow(numero,2);
//System.out.println(.El dato es:-dato);
trace.addPoint((((double) System.currentTimeMillis() - this.m_starttime)*1000),numero);
suma = suma + array[muestra];
muestra++;
if(muestra == 49){
aux = suma / 50;
aux = Math.sqrt(aux);
System.out.println(.El dato es:-dato);
muestra = 0;
aux = 0;
suma = 0;
/*if(aux <= 32){
mySP.sendSingleData(.A");
}
if(aux > 32 & aux <= 55 ){
mySP.sendSingleData("B");
}
if(aux > 55){
mySP.sendSingleData(Ç");
}*/
}
} catch (Exception ex) {

}
}
};
Timer timer1 = new Timer();
```

```
        timer1.scheduleAtFixedRate(timerTask, 0,2);
    }
}
```