

SISTEMA DE MONITOREO DE TEMBLOR EN PACIENTES CON TRASTORNOS DE MOVIMIENTO ANORMALES

ANDRÉS FELIPE CORREA ROJAS

Modalidad: Trabajo de investigación formativa
para aspirar al título de Ingeniero Electrónico

Director

Walter Serna, Ing. M.Sc

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
PEREIRA

2017

1. INTRODUCCIÓN

En la actualidad hay millones de personas que sufren algún tipo de trastornos de movimiento, los cuales en su gran mayoría están asociados con alteraciones patológicas de los ganglios basales (estriado, globo pálido, sustancia negra y núcleo subtalámico). Existen dos grandes grupos donde se pueden clasificar: los síndromes hipocinéticos, caracterizados por una pobre lentitud de movimiento y las discinecias (movimiento involuntarios), caracterizados por un exceso de movimiento, siendo la enfermedad de parkinson y los síndromes parkisonianos su principal exponente.[1]

Por ende la enfermedad de parkinson es de los trastornos de movimiento más frecuentes, afectando a unos 5.2 millones de hombres y mujeres en todo el mundo, y cada año se notifican entre 4 y 20 nuevos casos por cada 100.000 habitantes. Su afectación se centra en las personas mayores de 60 años, no obstante cerca del 5% de los casos reportados aparece en personas menores de 40 años;. Debido al hecho de que la enfermedad de parkinson se intensifica con la edad y el hecho que la expectativa de vida crece, significa que los casos de la enfermedad se ven en aumento.[2]

En este sentido el temblor es uno de los síntomas más comunes en los pacientes con enfermedad de parkinson y es ocasionado por la presencia de actividad eléctrica irregular en el tejido cerebral, este es un movimiento involuntario que se caracteriza por oscilaciones regulares o irregulares de una o varias partes del cuerpo; el temblor puede ser fisiológico ó patológico. El temblor patológico se presenta en cabeza, extremidades y en la voz, lo que da como resultado una degradación en la calidad de vida.[3]

Debido a que el diagnóstico depende de la experiencia y habilidades del profesional encargado de la evaluación, se ha visto en la necesidad de buscar métodos complementarios para un diagnóstico más certero, en este caso se realizo un sistema de monitoreo utilizando un acelerómetro, un arduino y la herramienta de programación Matlab con lo que se proceso la señal adquirida por el acelerómetro ADXL335. Los datos no necesitaron de circuitos adicionales de acoplamiento ya que el acelerómetro cuenta con un buen acondicionamiento interno que permite manejar directamente los datos del acelerómetro, estos datos fueron pasados a una señal discreta y enviados a un equipo de computo para ser filtrados, luego estos datos se transforman al dominio de la frecuencia por medio de la transformada rapida de fourier (FFT), donde se obtiene las características de frecuencia y amplitud entregados por el temblor. A los datos en el dominio de la frecuencia se les aplicó el método estadístico PCA el cual permite extraer los valores propios de los datos reduciendo significativamente la cantidad de características sin perder información y al finalizar se aplicó el método de agrupación K-Means para extraer los centroides característicos que nos permitan determinar si el sujeto de prueba presenta temblor o no.

2. MARCO CONCEPTUAL

2.1. TRASTORNOS DE MOVIMIENTOS

Los trastornos del movimiento incluyen un grupo de enfermedades caracterizadas bien por un exceso de movimientos (movimientos anormales involuntarios) o por una pobreza o lentitud de estos (síndromes rígido-acinéticos). Estos trastornos se deben a una disfunción de los ganglios basales los cuales están constituidos por un grupo de núcleos grises subcorticales y del tronco cerebral.

El temblor es un movimiento muscular involuntario, algo rítmico con componentes periódicas, que implica movimientos de balanceo (oscilaciones) de una o más partes del cuerpo que se produce por la contracción alternante de músculos agonistas y antagonistas, es el más común de los movimientos involuntarios y puede afectar las manos, los brazos, la cabeza, la cara, las cuerdas vocales, el tronco y las piernas. La mayoría de los temblores se produce en las manos; en algunas personas, el temblor es un síntoma de otro trastorno neurológico. La forma más común de temblor sin embargo se produce en personas sanas, expuestas a estrés, consumo de bebidas alcohólicas, consumo de drogas o cansancio físico entre otras.

Por lo tanto el temblor es un movimiento involuntario caracteriza por oscilaciones regulares o irregulares de uno o de varios segmentos corporales; se puede clasificar de acuerdo a su asignación anatómica, las circunstancias en que se produce, su frecuencia, amplitud, y si es fisiológico o patológico.[1, 3]

En este sentido la clasificación del temblor se da en tres grupos en función de sus características cinemáticas: temblor en reposo, temblor postural y temblor cinético o intencional es decir el temblor en reposo sucede cuando los brazos están totalmente relajados y hay ausencia de cualquier influencia gravitatoria, el temblor postural se manifiesta al mantener la extremidad en contra de la acción de la gravedad y el temblor cinético es aquel que ocurre durante la realización de un movimiento. Finalmente el temblor postural como el cinético se catalogan como temblores de acción.[4]

2.1.1. Clasificación del Temblor

- **Temblor en Reposo:** se asocia generalmente a dos condiciones, siendo la más frecuente el temblor de tipo parkinsoniano; la otra entidad es el temblor rúbrico o mesencefálico o tipo Holmes; presente cuando una extremidad está totalmente soportada a favor de la gravedad. Frecuencia 3-6 Hz .
- **Temblor Parkinsoniano:** Es un temblor que tiene una frecuencia entre 3-6 Hz, alternante entre músculos flexores y extensores, suele ser asimétrico y se asocia a otros síntomas como bradicinesia (lentitud de movimientos) y rigidez. Es un temblor que suele responder al tratamiento

con levodopa, agonistas dopaminérgicos o fármacos anticolinérgicos. Este temblor, aunque característico de la enfermedad de Parkinson, se puede asociar raramente a otros clínicos que permiten su diferenciación con la enfermedad de Parkinson idiopática como hipotensión ortostática, incontinencia urinaria, parálisis de la mirada vertical, deterioro cognitivo, piramidalismo, trastorno prominente y precoz de la marcha, entre otras.

- **Temblor Postural:** Se presenta con mayor frecuencia en el temblor esencial. Sin embargo, hay componentes posturales presentes en otra clase de temblores, entre ellos el temblor fisiológico aumentado, la enfermedad de Parkinson y otros parkinsonismos, el temblor mesencefálicos, el temblor ortostático, el temblor postónico, el inducido por fármacos, el temblor cerebeloso y el temblor psicógeno.
- **Temblor Esencial:** Se caracteriza por tener rango de frecuencias que oscila entre 4 y 12 Hz, aunque la frecuencia varía con la intensidad, la localización y la edad, de forma que el temblor tiende a ser más lento pero de mayor amplitud, conforme ésta avanza aparece con el mantenimiento de la postura y durante el reposo. Es una condición que afecta principalmente a las extremidades superiores como resultado de la contradicción alternante o sincrónica de músculos agonistas y antagonistas. Generalmente, el temblor esencial se presenta de forma bilateral, pero puede tener un inicio unilateral donde a menudo empeora con el estrés y puede mejorar con la ingesta de alcohol, sin que se conozca el mecanismo exacto por el cual esto ocurre.
- **Temblor Fisiológico:** No se considera un temblor patológico ya que se encuentra presente en sujetos sanos y en todas aquellas articulaciones o músculos colocados en una posición en la que sea posible oscilar. Su frecuencia característica oscila entre 8 y 12 Hz y puede hacerse patente al exacerbarse (síndrome del temblor fisiológico acentuado) variante normal resultante de la contractura muscular por el estrés, la fatiga, la ansiedad, el hipertiroidismo o el consumo de fármacos excitadores del sistema nervioso central o de beta-adrenérgicos.
- **Temblor Ortostático:** Este tipo de temblor aparece frecuentemente en las extremidades inferiores y en el tronco a los pocos segundos de que el paciente adopta la postura de bipedestación. Ocasionalmente puede aparecer a nivel de las extremidades superiores. Es por ello que los estudios electrofisiológicos han mostrado un patrón de contracción alternante o de contracción en músculos antagonistas de las extremidades. Su frecuencia característica es de 16 Hz y el temblor desaparece con la sedestación y también puede hacerlo durante la marcha. A veces el temblor es tan fino que no se puede apreciar mediante la inspección visual pero sí mediante la palpación del cuádriceps. Sin embargo, otras veces se puede observar una contracción más vigorosa del cuádriceps, incluso con oscilación visible de la rótula. En estos

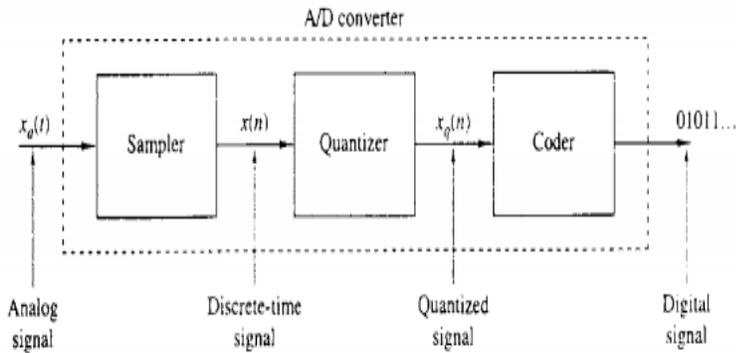


Figura 1. Partes básicas de un convertidor analógico a digital (A / D).[7]

casos la frecuencia del temblor suele ser un armónico de 8 Hz.

El temblor puede ser ocasionado por otro tipo de enfermedades como el alcoholismo, el consumo excesivo de alcohol, o la abstinencia del alcohol pueden matar ciertas células nerviosas, dando como resultado el temblor, especialmente en la mano. [5, 6]

2.2. Conversión Análogo a Digital

La mayoría de las señales de interés práctico, son de naturaleza señales analógicas como son las señales biológicas, Señales acústicas, señales de sonar y varias señales de comunicación como audio y Las señales de vídeo. Para procesar señales analógicas por medios digitales, es Necesario convertirlas en forma digital, es decir, convertirlos en una secuencia De números con precisión finita. Este procedimiento se denomina conversión analógico-digital (A / D), y los dispositivos correspondientes se denominan convertidores A / D (ADC). Conceptualmente, vemos la conversión A / D como un proceso de tres pasos [7]. Este proceso se ilustra en la Figura 1

1. Muestreo: Esta es la conversión de una señal de tiempo continuo en una señal de tiempo discreto obtenida tomando "muestras" de la señal de tiempo continuo en instantes de instante discretos, por lo que si $X_a(t)$ es entrada al muestreo, La salida es $X_a(nT) = x(n)$, donde Esto se denomina el intervalo de muestreo.
2. Cuantización: Ésta es la conversión de una señal de valor continuo de tiempo discreto en una señal discreta (digital) de tiempo discreto. El valor de cada muestra de la señal está represen-

tado por un valor seleccionado de un conjunto finito de valores posibles. La diferencia entre la muestra no cuantificada $x(n)$ y la salida cuantizada $x_q(n)$ se denomina error de cuantificación.

3. Codificación: En el proceso de codificación, cada valor discreto $x_q(n)$ está representado por una secuencia binaria de k bits, donde k es un valor característico de cada convertidor ADC.

2.3. Filtros Digitales

Un filtro es un sistema que, dependiendo de algunos parámetros, realiza un proceso de discriminación de una señal de entrada, obteniendo variaciones en su salida. Los filtros digitales tienen como entrada una señal digital y a su salida tienen otra señal digital, pudiendo haber cambiado en amplitud, frecuencia o fase dependiendo de las características del filtro. El filtrado digital es parte del procesamiento de señal digital.

Por lo tanto el filtro digital consiste en un procesamiento interno de datos de entrada y en ocasiones datos de salida pasados. El valor actual y los valores de entrada y salida de instantes anteriores previamente almacenados, son multiplicados por unos coeficientes definidos para las entradas y salidas. Por último todos los resultados de las multiplicaciones son sumados, para obtener una salida del instante actual.[8]

2.3.1. Clasificación de filtros según su respuesta en frecuencia :

- Filtro pasa bajos: Es aquel que bloquea el paso de las altas frecuencias y permiten el paso de frecuencias bajas, desde frecuencia 0 o continua hasta una determinada. Presentan ceros a alta frecuencia y polos a bajas frecuencia. También pueden definirse como aquellos filtros que no atenúan o atenúan un poco las señales de frecuencias comprendidas entre $f=0$ y $f=f_c$, siendo f_c la frecuencia de corte especificada según aplicación, mientras que producen una gran atenuación de las señales de frecuencias exteriores a aquellas.
- Filtro pasa altos: Es el que bloquea el paso de bajas de frecuencia y permite el paso de frecuencias desde una frecuencia de corte determinada hacia arriba, sin que exista un límite superior especificado. Presentan ceros a bajas frecuencias y polos a altas frecuencias.
- Filtro pasa banda: Son aquellos que permiten el paso de componentes frecuenciales contenidos en un determinado rango de frecuencias, comprendido entre una frecuencia de corte superior y otra inferior. También pueden definirse como aquellos filtros que no atenúan o atenúan poco las señales de frecuencias comprendidas entre $f_{c1} < f < f_{c2}$ y producen una gran atenuación de las señales de frecuencias exteriores a aquellas.

$$y[n] = \sum_{k=0}^{N-1} b_k x[n-k] = \sum_{k=0}^{N-1} h[k] x[n-k] \text{ con } h[k] = \{h_0, h_1, \dots, h_{N-1}\}$$

Figura 2. Ecuación filtro FIR[10]

- Filtro elimina banda: Es el que dificulta el paso de componentes frecuenciales contenidos en un determinado rango de frecuencias, comprendido entre una frecuencia de corte superior y otra inferior.

2.3.2. FILTROS FIR (Finite Impulse Response)

FIR es un acrónimo en inglés para Finite Impulse Response o Respuesta finita al impulso. Se trata de un tipo de filtros digitales en el que, como su nombre indica, si la entrada es una señal impulso, la salida tendrá un número finito de términos no nulos, son filtros del tipo no recursivo, por tal motivo estos filtros al no tener retroalimentación no tienen polos, y por lo tanto siempre son estables. Para obtener la salida solo se emplean valores de las entradas actuales y anteriores.

Por ende, la estructura de señal a la salida del filtro se basa solamente en la combinación lineal de las entradas actuales y anteriores. La expresión de la Figura2 corresponde a la convolución de la señal de entrada $x(n)$ con la respuesta impulso del filtro FIR $h(n)$, en donde N es el orden del filtro (el número de términos no nulos coincide con el número de coeficientes b_k del filtro).[9]

Para diseñar un filtro FIR es necesario tener en cuenta que la cantidad de coeficientes o duración de la respuesta impulso del filtro es siempre finita, a diferencia de la respuesta impulso de su respectivo filtro ideal. Debido a esto, la respuesta impulso del filtro FIR exhibirá cierto truncamiento implícito manifestado en la respuesta en frecuencia como un fenómeno de Gibbs, el cual produce ondulaciones antes y después de cualquier discontinuidad. En consecuencia, se realiza el enventanado a través de un número finito de secuencias de $w(n)$ para reducir los rizos o lóbulos de la banda de rechazo en la respuesta en frecuencia del filtro.

Hay tres métodos básicos para diseñar los filtros FIR:

- Método de las ventanas.
- Muestreo en frecuencia
- Rizado constante (aproximación de Chebyshev y algoritmo de intercambio de Remez).

Los filtros FIR por lo general son de fase lineal y se caracterizan por no contar con retroalimentación, por lo cual cuenta con mayores localidades de memoria. Entre sus ventajas, se destaca su estabilidad y una mayor facilidad de implementación respecto a los filtros IIR. Durante el proceso de diseño de los filtros, se busca obtener un comportamiento determinado en el dominio del tiempo o de la frecuencia. Esto se traduce en la estabilidad, distorsión y ubicación espacial de los ceros del filtro.[10]

2.4. Transformada Discreta de Fourier

La transformada Discreta de Fourier, conocida con frecuencia por la abreviatura DFT (del inglés Discrete Fourier Transform), es un tipo de transformada discreta utilizada en el análisis de Fourier. Transforma una función matemática en dominio del tiempo a otra función en el dominio de la frecuencia. Pero la DFT requiere que la función de entrada sea una secuencia discreta y de duración finita. Dichas secuencias se suelen generar a partir del muestreo de una función continua, como puede ser el temblor o la voz. Por tanto la DFT es una operación ampliamente empleada en tratamiento de señales y en campos afines para analizar las frecuencias presentes en una señal muestreada. La Transformada discreta de Fourier puede calcularse de modo muy eficiente mediante el algoritmo FFT.

2.4.1. Transformada Rápida de Fourier

La transformada rápida de Fourier (FFT) es la abreviatura usual (del inglés Fast Fourier Transform) de un algoritmo eficiente que permite calcular la Transformada Discreta de Fourier (DFT) y su inversa. Una secuencia de N números complejos $X = x_0, x_1, \dots, x_{N-1}$ los cuales son muestras de una señal analógica que debe transformarse en una secuencia de la forma $F = f_0, f_1, \dots, f_{N-1}$ según la ecuación descrita en la Figura 3. La evaluación directa de esta fórmula requiere $O(N^2)$ operaciones aritméticas; mediante un algoritmo FFT se puede obtener el mismo resultado con solo $O(N \log N)$ operaciones.

En términos generales la idea que permite la optimización es la descomposición de la transformada en otras transformadas más simples repitiendo el procedimiento hasta llegar a dos elementos ($N=2$). Una vez resueltas las transformadas más simples, se reagrupan, reordenan y resuelven nuevamente en el nivel superior hasta llegar al nivel más alto. Al final de este proceso, los resultados deben ser reordenados.[11][12]

2.5. Análisis de Componentes Principales (PCA)

El análisis de componentes principales de una matriz de datos extrae los patrones dominantes en

$$f_k = \sum_{n=0}^{N-1} x_n e^{-jk \frac{2\pi}{N} n}; \quad k = 0, \dots, N-1$$

Siendo:

- e la base de los logaritmos naturales,
- j la unidad imaginaria ($j^2 = -1$)
- N es el número de muestras.
- $f_k \in F$

Figura 3. Ecuación de la Transformada Discreta de Fourier[11]

la matriz en términos de un conjunto complementario de parcelas de puntuación y carga. Este procedimiento estadístico utiliza una transformación ortogonal convirtiendo un conjunto de observaciones de variables posiblemente correlacionadas, en un conjunto de valores de forma lineal no correlacionadas variables llamadas componentes principales .

El número de componentes principales es menor o igual que el número de variables originales. Esta transformación se define de tal manera que el primer componente principal tiene la mayor posible varianza (es decir, representa como gran parte de la variabilidad en los datos como sea posible), y cada componente subsiguiente a su vez tiene la mayor varianza posible bajo la restricción que es ortogonal a los componentes anteriores. Los vectores resultantes son un conjunto base ortogonal no correlacionados. PCA es sensible a la escala relativa de las variables originales.[13, 14]

Los objetivos del PCA son

- Extraer la información más importante de la tabla de datos.
- Comprimir el tamaño del conjunto de datos manteniendo sólo Esta información importante.
- Simplificar la descripción del conjunto de datos.
- Analizar la estructura de las observaciones y la Variables.

2.6. Método K-Means

El método de agrupación K-Means, tiene como objetivo la partición de un conjunto de n puntos que se encuentran en el espacio d dimensional en conjunto k centros, cada uno de los cuales está situado

$$KM(X,C) = \sum_{i=1}^n \min_{j \in \{1 \dots k\}} ||x_i - c_j||^2$$

Figura 4. Función del Algoritmo K-Means[15].

en el centroide de los datos para los que es el centro más cercano. Para la función de pertenencia, cada punto de datos pertenece a su centro más cercano, formando una partición Voronoi de los datos.[15, 16] Su función se puede apreciar en la figura 4.

3. DESARROLLO METODOLÓGICO

Este estudio de monitoreo de trastornos de movimiento enfocado al temblor tuvo finalidad práctica, por lo que esta investigación se puede clasificar como aplicada, con la cual se pretendió ayudar por medio de un sistema de monitoreo a los especialistas del Instituto de Epilepsia y Parkinson del Eje Cafetero Neurocentro con una nueva herramienta para brindar un crecimiento en la efectividad de la valoración de los pacientes.



Figura 5. Diagrama de bloques de metodología.

El desarrollo metodológico implementado para el **sistema de monitoreo de temblor en pacientes con trastornos de movimiento anormales** a partir de las siguientes líneas de acción:

3.1. SUJETOS

Veinte sujetos saludables y sin ningún signo de presentar síntomas asociados a trastornos de movimiento o diagnóstico previo de enfermedad, en un rango de edad de 18 a 81 años fueron estudiados en este trabajo. De los veinte sujetos de estudio catorce de ellos son de sexo femenino y seis de sexo

masculino. Todos los sujetos dieron verbalmente su consentimiento ya que habían sido informados del procedimiento y el uso de los datos.

Los criterios de exclusión fueron: fracturas o lesión de tejidos blandos en los miembros superiores o la columna vertebral en los dos últimos años, lo suficientemente graves como para afectar las actividades normales; Enfermedades neurológicas o temblores en el sujeto o familiares cercanos; Drogas tomadas que podrían causar temblor; Alcohol tomado en las 12 horas anteriores.

3.2. EQUIPOS Y SISTEMA DE ADQUISICIÓN

El sistema utilizado para la recolección de todos los datos en este trabajo fue construido utilizando un sensor de aceleración o acelerómetro de referencia ADXL335, un Arduino Uno utilizado para realizar la conversión análogo a digital y comunicación con un equipo de cómputo necesaria para el procesamiento de la señal y un equipo de cómputo donde se utilizó la herramienta de software Matlab R2014a donde se implementó el procesamiento de la señal.

3.2.1. ACELERÓMETRO

El ADXL335 es un sistema completo de medición de aceleración de 3 ejes. El ADXL335 tiene un rango de medición de ± 3 g mínimo. Contiene un sensor micromecanizado de superficie de polisilicio y circuitos de acondicionamiento de señales para implementar una arquitectura de medición de aceleración de bucle abierto. Las señales de salida son tensiones analógicas que son proporcionales a la aceleración. El acelerómetro puede medir la aceleración estática de la gravedad en aplicaciones de detección de inclinación, así como la aceleración dinámica resultante del movimiento, choque o vibración. El sensor es una estructura micromecanizada de superficie de polisilicio construida sobre una oblea de silicio. Los resortes de polisilicio suspenden la estructura sobre la superficie de la oblea y proporcionan una resistencia contra las fuerzas de aceleración. La deflexión de la estructura se mide utilizando un condensador diferencial que consta de placas y placas fijas independientes unidas a la masa móvil. Las placas fijas son accionadas por ondas cuadradas fuera de fase de 180° . La aceleración desvía la masa móvil y desbalancea el condensador diferencial dando como resultado una salida del sensor cuya amplitud es proporcional a la aceleración. Las técnicas de demodulación sensible a la fase se utilizan entonces para determinar la magnitud y la dirección de la aceleración. El ADXL335 está disponible en un pequeño paquete de escala de chips de marco de plomo de $4\text{ mm} \times 4\text{ mm} \times 1,45\text{ mm}$, 16-plomo, de bajo perfil (LFCSP_LQ). En la figura 6 se puede ver el diagrama de bloques funcionales del ADXL335.

La salida del desmodulador es amplificada y sacada del chip a través de una resistencia de $32\text{ k}\Omega$. El usuario entonces establece el ancho de banda de señal del dispositivo añadiendo un condensador. Este filtrado mejora la resolución de la medición y ayuda a prevenir el aliasing. El ADXL335 utiliza

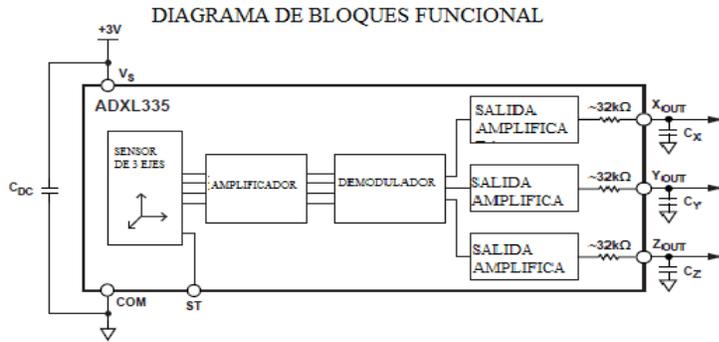


Figura 6. Diaframa de bloques funcionales [17]

una sola estructura para detectar los ejes X, Y y Z. Como resultado, las direcciones de sentido de los tres ejes son altamente ortogonales y tienen poca sensibilidad entre ejes. La desalineación mecánica de la matriz del sensor al paquete es la principal fuente de sensibilidad entre ejes. Por supuesto, la desalineación mecánica puede calibrarse a nivel del sistema. En lugar de utilizar circuitos de compensación de temperatura adicionales, las técnicas de diseño innovadoras garantizan que el alto rendimiento esté incorporado al ADXL335. Como resultado, no hay error de cuantificación o comportamiento no monótono, y la histéresis de temperatura es muy baja (típicamente menor de 3 mg en el intervalo de temperaturas de 0,25 ° C a +70 ° C).[17].

Debido a las buenas cualidades del ADXL335 mencionadas anteriormente se utilizó el sensor sin ningún circuito adicional de acoplamiento, y simplemente se maneja un acoplamiento en los pines de alimentación con un condensador de 0.1 uF. Adicional a esto, se realizó un algoritmo de calibración donde se colocó el acelerómetro en una superficie plana de modo que el eje a calibrar apunte hacia arriba y se elaboró la medición de la señal generando un valor de ganancia y compensación.

3.2.2. ARDUINO UNO

Arduino / Genuino Uno es una placa electrónica basada en el ATmega328P (ficha técnica). Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 se podrán utilizar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o la corriente con un adaptador de CA a CC o una batería.

El Arduino Uno tiene 6 entradas analógicas, A0 a A5 marcado, cada uno de los cuales proporcionan 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto se miden desde el suelo a 5 voltios, aunque es posible cambiar el extremo superior de su rango de uso de la función de la analogReference () pin AREF y. Hay un par de patas de la placa:

- AREF. Voltaje de referencia para las entradas analógicas. Se utiliza con `analogReference()`.
- Reiniciar. Llevar esta línea baja para reiniciar el microcontrolador. Normalmente se utiliza para añadir un botón de reinicio para escudos que bloquean la una en la mesa.

Arduino / Genuino Uno tiene una serie de instalaciones para la comunicación con un ordenador, otra placa Arduino / Genuino, u otros microcontroladores. El ATmega328 ofrece UART TTL (5V) de comunicación en serie, que está disponible en los pines digitales 0 (RX) y 1 (TX). Un ATmega16U2 en los canales de mesa esta comunicación en serie a través de USB y aparece como un puerto COM virtual para el software en el ordenador. El firmware 16U2 utiliza los controladores USB estándar COM, y no se necesita ningún controlador externo. Sin embargo, en Windows, es necesario un archivo `.inf`. El software de Arduino (IDE) incluye un monitor de serie que permite a los datos de texto simples para ser enviados hacia y desde el tablero. Los LEDs RX y TX de la placa parpadean cuando se están transmitiendo datos a través de la conexión USB chip y de USB a serie al ordenador (pero no para la comunicación en serie en los pines 0 y 1) [18].

Para este fin en particular, el Arduino Uno fue usado para la adquisición de la señal analógica proveniente del acelerometro ADXL335, los datos entregados por los tres ejes X, Y, y Z del sensor fueron ingresados a través de las entradas analógicas A0, A1 y A2 respectivamente para posteriormente ser convertidas a una señal discreta y enviada al equipo de computo utilizando la comunicación serial del Arduino Uno proporcionada por el puerto COM virtual a 9600 Baudios. El programa Utilizado se puede encontrar en los anexos de este trabajo.

3.2.3. PROGRAMA DE ADQUISICIÓN

En el equipo de computo y mediante el software Matlab R2014a, se diseño un algoritmo programa de manejar la comunicación entre el Arduino Uno y el equipo de computo, y principalmente manejar los tiempos de adquisición, frecuencia de muestreo, cantidad de muestras y filtrado de la señal; Para este proceso se utilizaron funciones para facilidad en el momento de la programación y entendimiento del programa de la siguiente manera:

- Función `SetupSerial`: En esta función se crea el objeto para la lectura del puerto COM, así como la configuración básica de la comunicación serial que incluye `DataBits` de 8, un `Stop-`

Bits en 1 y BaudRate en 9600. Finalmente al terminar la función se desplegaba un mensaje en la pantalla indicando que la comunicación serial esta configurada.

- **Función Calibrate:** En esta función se realiza un proceso de calibración del sensor antes de la toma de muestras desplegando un mensaje en la pantalla para cada eje a calibrar indicando que se debe colocar sobre una superficie plana y el eje en dirección a la superficie para posterior a esto llamar la función de lectura y calcular un valor de compensación para cada eje.
- **Función ReadAcc:** En esta función se envía la solicitud de lectura de valores del sensor por medio del Arduino Uno y se determina los valores de compensación a utilizar para cada eje.
- **Función CloseSerial:** En esta función se cierra y borra el puerto COM liberando el puerto y se despliega un mensaje de puerto cerrada al usuario.
- **Función PasaBanda2:** Esta función fue diseñada a través de la herramienta de Filters Builder encontrada en Matlab R2014a y usando los conceptos anteriormente mencionados sobre filtros FIR en configuración pasa banda. por lo tanto se diseño un filtro con frecuencias de paso en 2 Hz y 18 Hz y frecuencias de atenuación o parada de 1Hz y 20 Hz, una ondulación de banda de paso de 1db y frecuencia de muestreo de 40Hz.

En el programa principal se realizo la adquisición con un tiempo de muestreo de 25ms aproximadamente y se recogieron 400 muestras por eje, seguidamente se hallo la magnitud de la aceleración y se filtro usando el filtro previamente diseñado para finalmente guardar la magnitud de la aceleración filtrada ya que con esta se llevara acabo toda la caracterización de la señal de temblor. Estas Funciones y el programa principal puede encontrarse en los anexos de este trabajo.

3.3. PROTOCOLO DE MEDIDA

Numerosas escalas de calificación y criterios de diagnóstico se utilizan para evaluar la gravedad de la EP, una de las mas caracterizadas en presentar síntomas de temblor en extremidades. Aunque en la actualidad la escala mas común es el UPDRS, muchos de los estudios en la base de datos informo otras escalas [19]. Basado en esto y enfocado únicamente en la sección de temblor en extremidades superiores, se uso el siguiente protocolo para la toma de datos:

Los sujetos estaban sentados en una silla de respaldo recto con los pies en el suelo. El sensor se aseguró En el dorso (espalda) de su mano dominante en un bolsillo sobre un soporte de muñeca

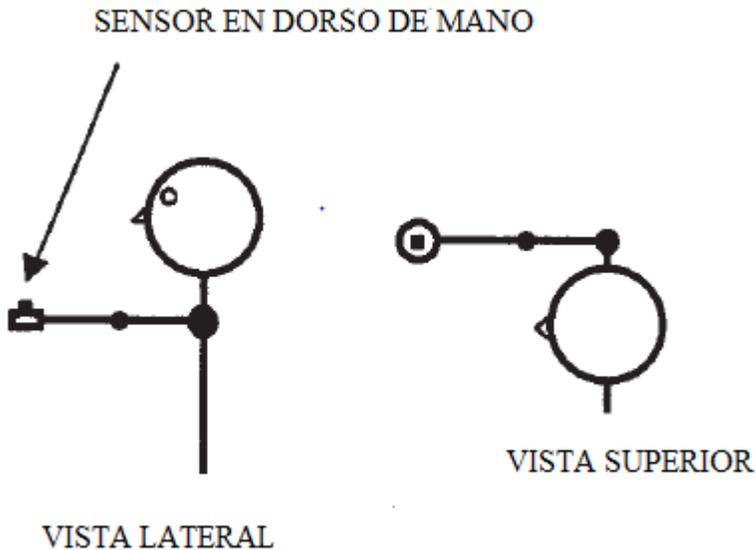


Figura 7. Ilustración de posturas del miembro superior Adoptada durante la recopilación de datos.[20]

mano ajustable. Los sujetos tenían que mantener la siguiente postura de los miembros superiores mientras se recogían los datos:

brazo dominante extendido derecho, Paralelo al suelo en ángulo recto con el hombro y el codo, con la mano extendida, adicional a esto la mano hacia abajo con el puño suelto y dedos extendidos [20]. ver figura7

El transmisor fue colocado sobre una mesa en frente Del sujeto. La distancia entre el transmisor Y el sensor era de aproximadamente 60 cm. Se pidió al sujeto que levantara el brazo hacia la postura indicada anteriormente y no hiciera movimientos voluntarios mientras se recogían los datos Durante 10 segundos. Este proceso se repitió para los mismos veinte sujetos, pero esta vez se les indico que hicieran movimientos simulando temblor en sus extremidades superiores. Finalmente se obtuvieron cuarenta muestras, donde veinte de ellas fueron tomadas sin ningún movimiento voluntario y veinte con movimientos voluntarios y en los cuales cada sujeto participo con una muestra de cada tipo.

3.4. DESCRIPCIÓN DE LA SEÑAL

La señal usada para este trabajo fue la señal obtenida del sensor ADXL335 el cual nos permite obtener la variación de la aceleración en los tres ejes (x, y, z). esta señal es adquirida por 10 se-

gundos a una frecuencia de muestreo de 40 Hz y resolución de 1024 bits, lo que nos genera 400 datos para cada señal. Durante el proceso de los datos y como el temblor de Parkinson no tiene una dirección fija, el primer paso es el de hallar la magnitud del vector aceleración como se muestra en la siguiente ecuación 1:

$$Mi = \sqrt{Xi^2 + Yi^2 + Zi^2} \quad (1)$$

Donde i es la secuencia de los datos del vector de punto.

3.5. EXTRACCIÓN DE CARACTERÍSTICAS DE LA SEÑAL

Para la extracción de características de la señal se usó la Transformada Rápida de Fourier (FFT) encontrada en la herramienta de Matlab, ya que esta nos permite realizar de manera fácil y efectiva la transformación de la señal del dominio del tiempo al dominio de la frecuencia, encontrando las características de la señal en el espectro de frecuencias que la caracterizan y los valores máximos de amplitud donde se puede ver la importancia de dicha frecuencia. En segundo lugar y luego de crear una base de datos de las señales transformadas al dominio de la frecuencia se utilizó la función de PCA en la herramienta de Matlab, la cual permitió obtener los valores propios que caracterizan de mejor manera la señal, la matriz de proyección W , la cual nos permite pasar de una cantidad de características P a una cantidad de características L donde la matriz $P \times L$ forman una base ortogonal para las características de L (Componentes de representación). Finalmente y para finalizar con el proceso de entrenamiento del sistema y caracterización de las señales tomadas, se usó la función de Kmeans de Matlab, esta función divide los puntos en la matriz de datos X $n \times p$ en k clusters. Esta partición iterativa minimiza la suma, sobre todos los conglomerados, de las sumas dentro del grupo de las distancias punto a clúster centroide. Las filas de X corresponden a puntos, las columnas corresponden a variables. Kmeans devuelve un vector $n \times 1$ que contiene los índices de clúster de cada punto. De forma predeterminada, kmeans utiliza distancias euclidianas cuadradas.

En conclusión el proceso de extracción de características nos dio la opción de utilizar dos clúster centroides los cuales representan los componentes principales de las muestra que fueron tomadas de veinte de los cuarenta datos totales y donde diez eran datos de sujetos sin movimientos voluntarios y diez de sujetos con movimientos voluntarios simulando temblor en sus extremidades superiores; Por otra parte estos datos fueron usados para el entrenamiento del sistema y así lograr clasificar

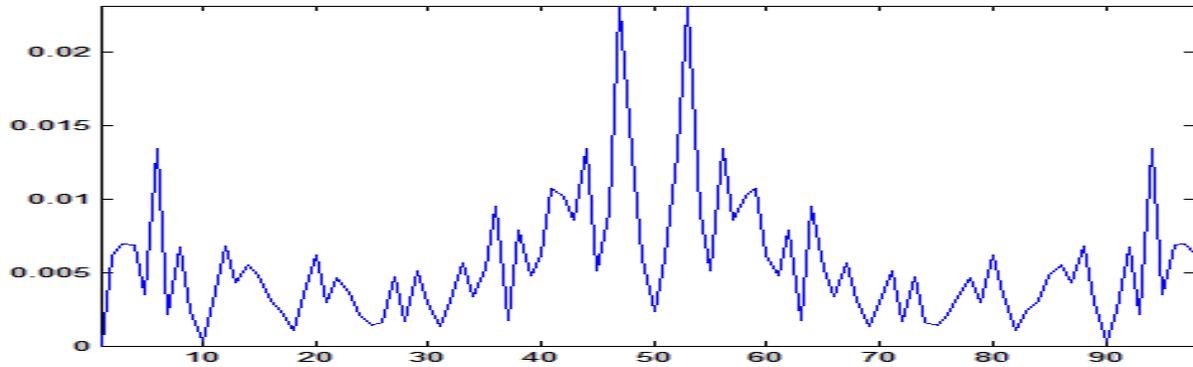


Figura 8. Espectro de la señal en sujetos sin movimientos voluntarios

nuevas muestras ingresadas al sistema como sujetos sin ningún movimiento voluntario y sujetos con movimientos voluntarios simulando temblor en sus extremidades superiores

4. ANÁLISIS DE RESULTADOS

A partir de los hallazgos encontrados en la extracción de características de la señal se pudo determinar:

Los datos tomados de los sujetos sin ningún movimiento voluntario no se logra evidenciar en el espectro ninguna frecuencias dominante o de características relevantes que puedan ser utilizadas como criterio de diagnostico, como se logra evidenciar en la figura 8 ya que sus amplitudes son fácilmente despreciables por su bajo valor.

Por lo contrario y como era de esperarse, los datos tomados de los sujetos con movimientos voluntarios simulando temblor en sus extremidades superiores se evidencia en el espectro una frecuencia característica en el rango de 8Hz a 16Hz como se muestra en la figura 9 las cuales eventualmente pueden ser objeto de estudio para diagnosticar tipos de temblores con respecto a las frecuencias típicas de cada tipo de temblor.

En otro orden de cosas, durante la aplicación del método PCA para obtener los valores propios se generaba el interrogante, de ¿cuantos valores propios representan de mejor manera la muestra?, este interrogante fue resuelto al identificar la gráfica de valores propios que se muestra en la figura 10, donde se evidencia que en los dos primeros valores propios se alcanza aproximadamente el 79% de las características de la muestra.

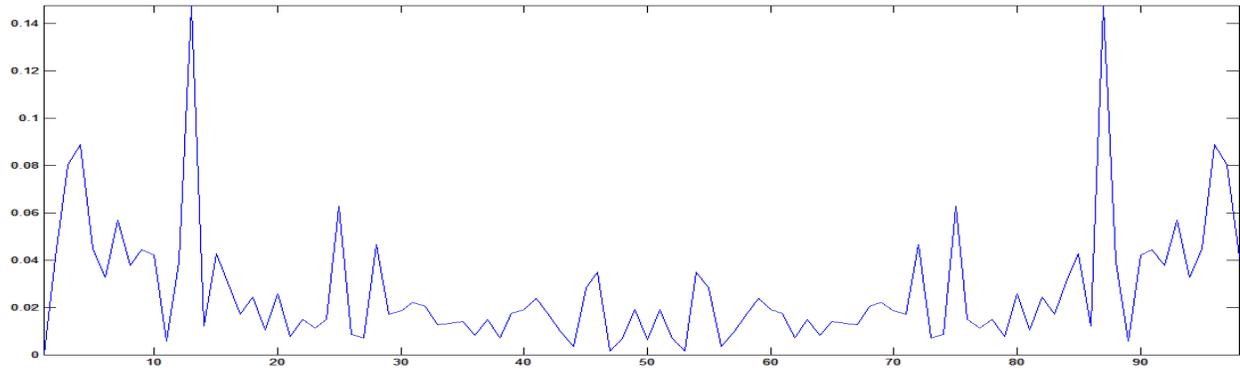


Figura 9. Espectro de la señal en sujetos con movimientos voluntarios simulando temblor en extremidades superiores

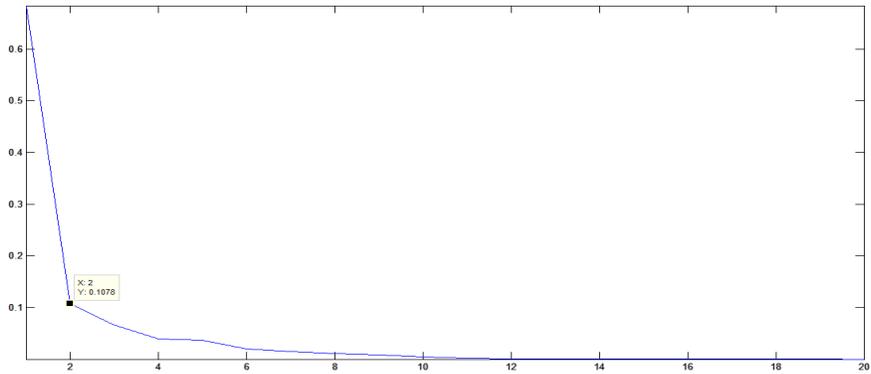


Figura 10. Gráfica de valores propios de la muestra

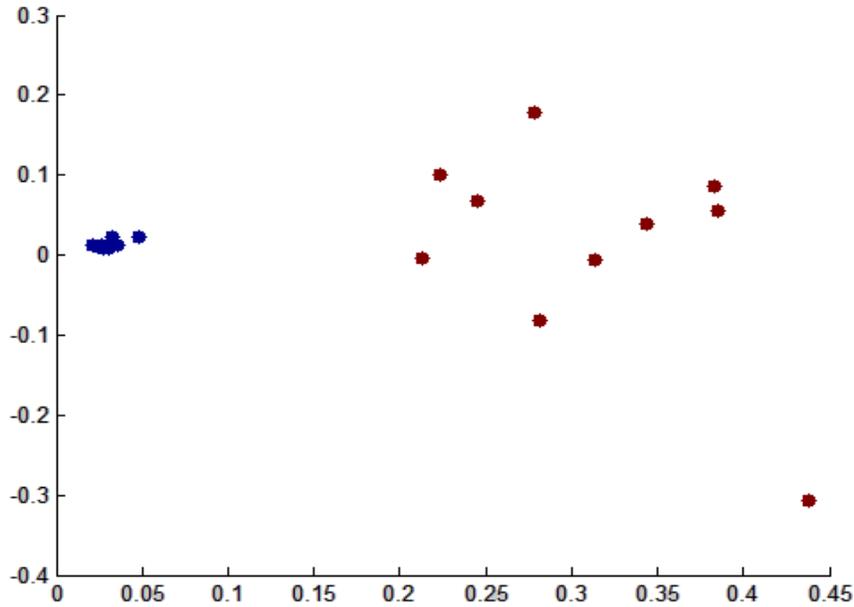


Figura 11. Gráfica de dispersión de datos

	sin movimiento voluntarios (Real)	con simulación de temblor (Real)
sin movimiento voluntarios (Encontrado)	10	0
con simulación de temblor (Encontrado)	3	7

Tabla 1. Matriz de confusión

Por ende, los datos obtenidos de la utilización del método PCA para obtener los valores propios de cada grupo de datos que para su mayor facilidad le llamaremos temblores simulados y sin movimiento voluntarios fue de gran aporte ya que estos muestran unas claras diferencias en sus valores propios como se puede apreciar en la figura 11 donde el grupo identificado con el color azul hace referencia a los valores sin movimientos simulados y el grupo identificado con el color rojo hace referencia a los valores de temblores simulados.

Finalmente, se logro realizar un entrenamiento del sistema hallando los centroides que contienen las características principales de los datos obtenidos a través de el hallazgo de los valores propios y su debida agrupación mediante el método de Kmeans, con esto y usando los veinte datos datos restantes se procedió a usar una matriz de confusión y así visualizar el funcionamiento del algoritmo y su porcentaje de error.

En el cuadro 1 se puede visualizar la Matriz de confusión encontrada para el algoritmo realizado

una precisión del 85 %

5. CONCLUSIONES, APORTES Y RECOMENDACIONES

Mediante la implementación del método PCA se logró evidenciar los valores propios que caracterizan de mejor manera la muestra, adicionalmente se identificó que con tan solo los dos primeros valores propios obtenidos se logra alcanzar aproximadamente el 79 % de la caracterización de la muestra.

La Gráfica de dispersión obtenida de los valores propios de la muestra claramente dos grupos con similares características con lo cual se puede concluir que mediante un entrenamiento adecuado el sistema puede responder de manera optima a la búsqueda de grupos y diagnostico adecuado de las características entrenadas en el sistema.

Durante el análisis de resultados de la matriz de confusión se logró evidenciar que el método de extracción de características usado en este trabajo logró alcanzar una precisión de 85 %, pero se evidencia un 30 % de probabilidad de encontrar falsos positivos en el hallazgo de sujetos sin movimientos voluntarios, aunque su porcentaje sigue siendo bueno.

BIBLIOGRAFÍA

- [1] López del Val, L. and Linazasoro. (2002). Trastornos del movimiento. Barcelona (España): MASSON, s.A.
- [2] Espanol.ninds.nih.gov, (2016). Enfermedad de Parkinson: Esperanza en la investigación : National Institute of Neurological Disorders and Stroke (NINDS). [online] Available at: http://espanol.ninds.nih.gov/trastornos/parkinson_disease_spanish.htm [Accessed 10 Feb. 2016].
- [3] Ai, L., Wang, J., & Yao, R. (2011). Classification of parkinsonian and essential tremor using empirical mode decomposition and support vector machine. *Digital Signal Processing*, 21(4), 543-550.
- [4] Rosen, M., & Adelstein, B. (1981, March). The effect of mechanical impedance on abnormal intention tremor. In *Proceedings of the 9th Annual Northeast Bioengineering Conference* (pp. 205-209).
- [5] Rosen, M., & Adelstein, B. (1981, March). The effect of mechanical impedance on abnormal intention tremor. In *Proceedings of the 9th Annual Northeast Bioengineering Conference* (pp. 205-209).
- [6] Temblor de reposo, Neurowikia El portal de contenido en neurología, 2010. [online] Disponible en: <http://www.neurowikia.es/content/temblor-de-reposo>
- [7] Proakis, J. G. M., Dimitris, S. G., Santalla del Río, V., Castro, A., & Luis, J. (1998). Tratamiento digital de señales.
- [8] Álvarez Cedillo, J. A., Bos, L., Michael, K., & Martínez Romero, G. (2008). Implementación de Filtros Digitales Tipo FIR en FPGA. *Polibits*, (37), 83-87.
- [9] Tompkins, W. J. (1993). *Biomedical digital signal processing*. Editorial Prentice Hall.

- [10] FCEF y N Universidad Nacional de Cordoba, Procesamiento Digital de Señales, Introducción a los Filtros digitales- filtros FIR, 2010. [online] Disponible en:http://www.dsp.efn.unc.edu.ar/documentos/Filtro_FIR.pdf
- [11] I. Mario Alfredo Ibarra Carrillo, Sistemas de comunicaciones, Capítulo 7 Análisis espectral de señales periódicas con FFT., 2009 [online] Disponible en: https://maixx.files.wordpress.com/2011/11/cap10_anc3a1lisis-espectral_con_fft_v_12_01_01.pdf
- [12] Tomasi, W. (2003). Sistemas de comunicaciones electrónicas. Pearson educación.
- [13] Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3), 37-52.
- [14] Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4), 433-459.
- [15] Hamerly, G., & Elkan, C. (2002, November). Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management* (pp. 600-607). ACM.
- [16] MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, No. 14, pp. 281-297).
- [17] ANALOG DEVICES, (2016). ADXL335: Small, Low Power, 3-Axis ± 3 g Accelerometer Data Sheet. [online] Available at: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL335.pdf> [Accessed 11 Feb. 2016].
- [18] Arduino.cc. (2017). Arduino - ArduinoBoardUno. [online] Available at: <https://www.arduino.cc/en/Main/ArduinoBoardUno> [Accessed 19 Jan. 2017].

- [19] Movement Disorder Society Task Force on Rating Scales for Parkinson's Disease. (2003). The Unified Parkinson's Disease Rating Scale (UPDRS): status and recommendations. *Movement disorders: official journal of the Movement Disorder Society*, 18(7), 738.
- [20] Spyers-Ashby, J. M., & Stokes, M. J. (2000). Reliability of tremor measurements using a multidimensional electromagnetic sensor system. *Clinical rehabilitation*, 14(4), 425-432.

ANEXOS

Algoritmo 1 Programa de Arduino para conversor ADC y comunicación serial

```
const int Az = A0; const int Ay = A1; const int Ax = A2; const int G0 = A3;
int mode = -1;
unsigned int sensorValue = 0;
void setup() { // the setup routine runs once when you press reset:
Serial.begin(9600); // initialize serial communication at 9600 bits per second:
Serial.println('a'); //Comprobar la comunicación serial - rutina de reconocimiento
char a = 'b';
while(a!='a')
{
a=Serial.read(); //Espera por un caracter específico desde el PC
}
}
void loop() { // the loop routine runs over and over again forever:
if(Serial.available(>0) //verifica si algun dato ha sido enviado por el PC
{
mode = Serial.read(); //Verifica si hay requerimientos para los valores del acelerómetro
if(mode == 'R') //Usado para configurar diferentes modos read the input on analog pin 0:
{
sensorValue = analogRead(Ax);
Serial.println(sensorValue);
sensorValue = analogRead(Ay);
Serial.println(sensorValue);
sensorValue = analogRead(Az);
Serial.println(sensorValue); // print out the value you read:
}
}
delay(1); // delay in between reads for stability
}
}
```

Algoritmo 2 Programa principal para la adquisición de datos desde Matlab.

```
clc
clear all
close all
% -----
comport = 'COM10';
[acelerometro.s,flag] = setupSerial(comport);
calCo = calibrate(acelerometro.s);
% -----
f=figure; i=0;
muestra=0;
while 1 i=i+1;
[gx(i), gy(i), gz(i)] = readAcc(acelerometro, calCo);
muestra=muestra+1;
if (get(f,'currentkey')== 'a') break;
end
if muestra==400 break;
end
end
% -----
closeSerial;
% -----
for i=1:length(gx)
ME2(i)=sqrt((gx(i)^2)+(gy(i)^2)+(gz(i)^2));
end
figure ();plot(ME2);title('LA SEÑAL MAGNITUD');
MF2 = pasabanda2(ME2);
%filtro 2 Hz a 18Hz
figure ();plot(MF2);title('LA SEÑAL FILTRADA DE MAGNITUD');
% guardar señal de acelerometro adquiridad
save Sujeto1.mat MF2
```

Algoritmo 3 Funcion SetupSerial para configuracion de comunicació.

```
function [s,flag] = setupSerial(comPort)
flag = 1;
s = serial(comPort);
set(s,'DataBits', 8);
set(s,'StopBits', 1);
set(s,'BaudRate', 9600);
set(s,'Parity', 'none');
fopen(s);
%Conecta el puerto serial "s" al arduino
a='b';
while(a~='a') a=fread(s,1,'uchar');
end if(a=='a') disp('serial read');
end fprintf(s,'%c',a);
%Escribe en el puerto serial el caracter a
mbox = msgbox('Comunicación serial configurada');
uiwait(mbox); fscanf(s,'%u')
end
```

Algoritmo 4 Funcion Calibrate para calibrar sensor.

```
function calCo = calibrate(s)
out.s=s; calCo.offset = 0; calCo.g = 1;
%read the raw accelerometer output at three different orientations
%gZ = 1, gX = gY = 0 orientation
mbox = msgbox('Coloque el acelerómetro en una superficie plana de modo que el eje Z apunte
arriba');
uiwait(mbox);
[ gx_z gy_z gz_z ] = readAcc(out, calCo);
%gX = 1, gZ = gY = 0 orientation
mbox = msgbox('Coloque el acelerómetro en una superficie plana de modo que el eje X apunte
arriba');
uiwait(mbox);
[ gx_x gy_x gz_x ] = readAcc(out, calCo);
%gY = 1, gZ = gX = 0 orientation
mbox = msgbox('Coloque el acelerómetro en una superficie plana de modo que el eje Y apunte
arriba');
uiwait(mbox);
[ gx_y gy_y gz_y ] = readAcc(out, calCo);
offsetX = (gx_z + gx_x)/2;
offsetY = (gy_z + gy_x)/2;
offsetZ = (gz_x + gz_y)/2;
%calculate scaling factors
gainX = gx_x - offsetX;
gainY = gy_y - offsetY;
gainZ = gz_z - offsetZ;
calCo.offset = [offsetX offsetY offsetZ];
calCo.g = [gainX gainY gainZ];
mbox = msgbox('Fin de la calibración');
uiwait(mbox);
end
```

Algoritmo 5 Funcion ReadAcc para leer sensor.

```
function [gx, gy, gz] = readAcc(out,calCo)
fprintf(out.s, 'R');
%read values from accelerometer
readings(1)= fscanf(out.s, '%u');
readings(2)= fscanf(out.s, '%u');
readings(3)= fscanf(out.s, '%u');
%determine what offset and gain values to use
offset = calCo.offset;
gain = calCo.g; accel = (readings - offset)./gain;
gx = accel(1);
gy = accel(2);
gz = accel(3);
end
```

Algoritmo 6 Funcion CloseSerial para cerrar el COM.

```
% clc
% clear all
if ~isempty(instrfind) fclose(instrfind);
delete(instrfind);
end
close all
disp('Puerto serial cerrado')
```

Algoritmo 7 Funcion PasaBanda2para filtrar la señal

```
function y = pasabanda2(x)
%PASABANDA2 Filters input x and returns output y.
% MATLAB Code % Generated by MATLAB(R) 8.3 and the DSP System Toolbox 8.6. % Gene-
rated on: 05-Jan-2017 22:11:43
%#codegen
% To generate C/C++ code from this function use the codegen command
.% Type 'help codegen' for more information.
persistent Hd;
if isempty(Hd)
% The following code was used to design the filter coefficients:
%% Fstop1 = 1; % First Stopband Frequency % Fpass1 = 2;
% First Passband Frequency % Fpass2 = 18; % Second Passband Frequency
% Fstop2 = 20; % Second Stopband Frequency % Astop1 = 60;
% First Stopband Attenuation (dB) % Apass = 1;
% Passband Ripple (dB) % Astop2 = 60; % Second Stopband Attenuation (dB)
% Fs = 40; % Sampling Frequency %
% h = fdesign.bandpass('fst1,fp1,fp2,fst2,ast1,ap,ast2', Fstop1, Fpass1, ...
% Fpass2, Fstop2, Astop1, Apass, Astop2, Fs); %
% Hd = design(h, 'equiripple', ... % 'MinOrder', 'any', ... % 'SystemObject', true);
Hd = dsp.FIRFilter( ... 'Numerator', [-0.0118199987408043 0.0121628797113327 ...
0.0187653420961211 -0.00388595669167525 0.00617302703660451 ... -0.00799330781446442
0.000649387236100319 -0.00853505115886611 ... -0.00350492002495766 -
0.00711106285974248 -0.00734644359199591 ... -0.00366446356246432 -
0.010147490726142 0.00166119112740602 ... -0.0106818164382491 0.0079811801677634
-0.00794958057228637 ... 0.0136739609401164 -0.00171043663393992 0.0167318505505062
... 0.00724764206845578 0.0153031617213245 0.0169724046181995 ... 0.00823820894327906
0.0246104546693057 -0.00429902872200197 ... 0.0270032365664332 -0.0205818565716777
0.0211767014494165 ... -0.037411327634736 0.00493228214819368 -0.0501802038611561
... -0.0229725070924578 -0.0530003924446708 -0.0641682213069025 ... -
0.0369265964862386 -0.126487814949273 0.024480349293953 ... -0.276321101997861
0.548935421457221 0.548935421457221 ... -0.276321101997861 0.024480349293953 -
0.126487814949273 ... -0.0369265964862386 -0.0641682213069025 -0.0530003924446708 ...
-0.0229725070924578 -0.0501802038611561 0.00493228214819368 ... -0.037411327634736
0.0211767014494165 -0.0205818565716777 ... 0.0270032365664332 -0.00429902872200197
0.0246104546693057 ... 0.00823820894327906 0.0169724046181995 0.0153031617213245 ...
0.00724764206845578 0.0167318505505062 -0.00171043663393992 ... 0.0136739609401164
-0.00794958057228637 0.0079811801677634 ... -0.0106818164382491 0.00166119112740602
-0.010147490726142 ... -0.00366446356246432 -0.00734644359199591 -0.00711106285974248
... -0.00350492002495766 -0.00853505115886611 0.000649387236100319 ... -
0.00799330781446442 0.00617302703660451 -0.00388595669167525 ... 0.0187653420961211
0.0121628797113327 -0.0118199987408043]);
end
y = step(Hd,x);
```
