

DOI: <https://doi.org/10.24054/16927257.v32.n32.2018.3037>Recibido: 19 de abril de 2018
Aceptado: 20 de junio de 2018**PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA
EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO****SIGNAL PROCESSING EMG IN AN EMBEDDED SYSTEM FOR CONTROL
NEURONAL OF A ROBOTIC ARM****Ing. Alejandro Cantillo Maldonado ***, **PhD. Oscar Gualdron Guerrero ****
MSc. Jesús Ortiz Sandoval ***

* **Universidad de Pamplona**, Facultad de ingeniería y Arquitectura, Ingeniero Electrónico.
Km 1 vía Bucaramanga, Pamplona, Norte de Santander, Colombia.

** **Universidad de Pamplona**, Facultad de Ingeniería y Arquitectura, Programa de
Ingeniería Electrónica, Grupo de Investigación en Sistemas Multisensoriales y
Reconocimiento de Patrones.

Km 1 vía Bucaramanga, Pamplona, Norte de Santander, Colombia.
E-mail: oscar.gualdron@unipamplona.edu.co.

*** **Universidad Técnica Federico Santa María**, Facultad de Ingeniería, Doctorado en
Ingeniería Electrónica.

Avenida España 1680, Valparaíso, Provincia de Valparaíso, Chile.
E-mail: jesus.ortiz@sansano.usm.cl.

Resumen: El objetivo de este trabajo es adquirir las señales EMG del brazo humano, para entrenar una red neuronal que pueda clasificar diferentes movimientos, de al menos 3 grados de libertad para controlar un brazo robótico. Para esto, se diseñó la adquisición y el acondicionamiento del sistema de señales EMG. Los músculos responsables del movimiento de la mano, el antebrazo y el brazo se identificaron para ubicar los electrodos en el músculo indicado. Se empieza con una posición de referencia para el movimiento de la mano, el antebrazo y el brazo. Las señales del sistema de adquisición y acondicionamiento se enviaron a la tarjeta de desarrollo ADC, donde se implementó un algoritmo que analizó la grabación de señales EMG para generar los datos con los que se entrenó la red neuronal. Luego, la red neuronal se implementó en el sistema integrado (BEAGLEBONE Black Rev. C) y se combinó con el brazo robótico.

Palabras clave: Brazo robótico, Código Phyton, Electromiografía, red neuronal artificial.

Abstract: The objective of this work is to acquire the EMG signals of the human arm, to train a neural network that can classify different movements, of at least 3 degrees of freedom to control a robotic arm. For this, the acquisition and conditioning of the EMG signal system was designed. The muscles responsible for the movement of the hand, forearm and arm were identified to locate the electrodes in the indicated muscle. It starts with a reference position for the movement of the hand, the forearm and the arm. The signals of the acquisition and conditioning system were sent to the ADC development card, where an algorithm was implemented that analyzed the recording of EMG signals to generate the data with which the neural network was trained. Then, the neural network was implemented in the integrated system (BEAGLEBONE Black Rev. C) and combined with the robotic arm.

Keywords: Robotic arm, Phyton Code, Electromyography, artificial neural network.

1. INTRODUCCIÓN

El sistema de procesamiento de las señales de EMG que se han desarrollado en los últimos 5 años, hace que el procesamiento de la información en una computadora; por esa razón, el sistema se limita al laboratorio, que limita el sistema de comercialización (Durán-Acevedo y Jaimes-Mogollón, 2013, Sánchez-Presas, 2013, García-Pinzón, Mendoza y Flórez, 2015). De este modo, surge la necesidad de llevar a cabo el procesamiento informático de la información a un sistema integrado como una posible solución a ese problema (López-Castellanos, 2017; Niño-Vega, Martínez-Díaz y Fernández-Morales, 2016).

Esta investigación se lleva a cabo debido a la necesidad de no limitar el sistema de procesamiento al laboratorio. Lo que realmente pretende esta investigación es procesar las señales del brazo EMG en un Sistema Embebido, en este caso a Beaglebone Black, y a través de una Red Neural clasifica los diferentes movimientos para controlar el brazo robótico, esto será útil para el desarrollo de prótesis electrónicas usando un Sistema embebido y también para otras aplicaciones (Gualdrón-Guerrero, Durán-Acevedo, Ortiz-Sandoval y Araque-Gallardo, 2014, Marino-Vera, Mendoza y Gualdrón-Guerrero, 2017, Ortiz-Sandoval, Marín-Herrera y Gualdrón-Guerrero, 2016).

El sistema realiza tres procesos importantes; Cuenta con adquisición de datos, inteligencia artificial y ejecución de movimientos (Ponce-Cruz, 2010, Martínez, Jáquez, Rivera y Sandoval, 2008). El primer proceso, la adquisición de datos, captura 3000 muestras de la señal de EMG y las técnicas de tratamiento simples se aplican para extraer las características o patrones (Texas Instruments, 2016, Herrera-Baquero y Prieto-Ortiz, 2018).

Con respecto a la inteligencia artificial, se implementa una red neuronal que clasifica el patrón obtenido en tipos: el primer tipo representa la posición de referencia del brazo robótico, por lo que aquí no hay movimiento; El segundo representa el movimiento del brazo, el tercer tipo representa el movimiento de derecha y el cuarto representa el movimiento de la mano.

La ejecución del movimiento, una vez que la red neuronal identifica el tipo, procede a ejecutar el

movimiento que se representa en este tipo, para ello, el Beaglebone Black envía la señal PWM a los servomotores del brazo robótico para ejecutar el movimiento identificado.

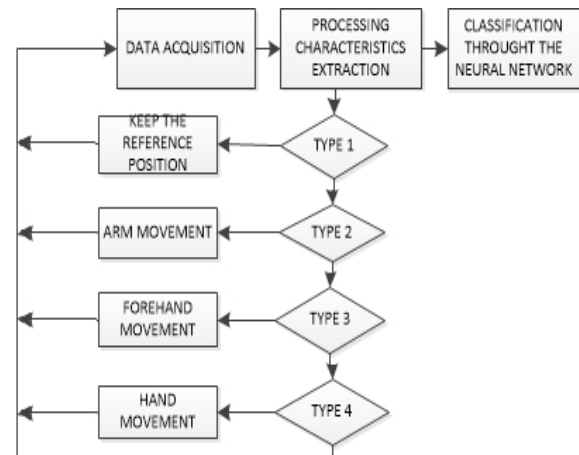


Fig. 1. Diagrama de flujo del sistema.

2. METODOLOGÍA

En la siguiente sección se describen cada uno de los métodos utilizados para el desarrollo del sistema integrado. Para el diseño de este sistema se tuvieron en cuenta las características ADC de Beaglebone Black. De manera general, el diagrama de bloques se muestra y describe a continuación.

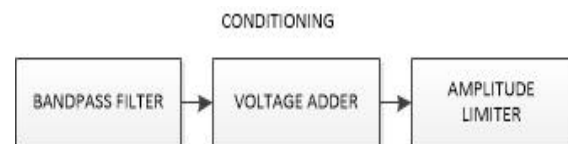


Fig. 2. Diagrama de bloques del sistema de adquisición.

Sensores: se usó una configuración de electrodos diferenciales de cloruro de plata (Ag / AgCl 3m), esos electrodos están conectados directamente a la piel; La conexión de los electrodos con el amplificador de instrumentación se realiza a través de conectores para electrodos.

El circuito integrado AD620 es el amplificador de instrumentación elegido para la captura y amplificación del sistema de señales EMG. Tiene las siguientes características: desviación de baja tensión máx. 500uV, un CMRR mín. 110dB, una ganancia ajustable de 1 a 10'000 y un rango de suministro de $\pm 2.3V$ a $\pm 18V$; esas

especificaciones permiten que los voltajes comunes sean removidos.

Acondicionamiento: Los electrodos en la superficie del brazo reciben información de algunas unidades motoras además de interferencias externas. Es necesario configurar la señal para atenuar todos los componentes de frecuencia que no forman parte de la señal EMG. Para ello, se implementó lo siguiente:

- **Filtro de paso de banda:** este filtro está formado por un filtro de paso de banda de octavo orden en cascada con un filtro de banda alta de cuarto orden; como la mayor concentración de información de la señal EMG es de 20 Hz y 500 Hz, la frecuencia de corte del paso bajo será de 500 Hz y la frecuencia de corte del paso alto será de 20 Hz.
- **Sumador de voltaje:** como el ADC del Beaglebone Black no admite voltajes negativos, es necesario agregar una desviación de voltaje dc a la señal de CA (EMG) para que esta desviación tome valores positivos y no destruya el ADC.
- **Limitador de amplitud:** el rango de voltajes que admite el ADC de Beagle Black es de 0-1.8V, lo que hace necesario limitar el voltaje máximo de la señal de EMG a 1.8V para no destruir el ADC de la placa.

Las redes neuronales artificiales son aproximadores no lineales de cómo funciona el cerebro; por lo tanto, no debe compararse directamente con el cerebro, ni confundir los principios en los que se basa el funcionamiento de las redes neuronales artificiales y el cerebro, ni pensar que las redes neuronales solo se basan en redes biológicas; considerando que solo emulan de manera muy simple el funcionamiento del cerebro humano (Sandoval-Ruiz, 2014; Enciso, Zelaya & Quesada-Sarmiento, 2018; Cera-Martínez, Ortiz-Sandoval y Gualdrón-Guerrero, 2018).

Las ANN en el margen para ser "parecer" al cerebro, tienen características propias en relación con él. Por ejemplo, los ANN aprenden de la experiencia, generalizan de ejemplos anteriores a nuevos y toman las características principales de diferentes datos.

Aprendizaje: Adquirir el conocimiento de una cosa a través del estudio, ejercicio o experiencia. Las ANN pueden cambiar su comportamiento en función del entorno. Se les muestra un conjunto de entradas y se ajustan para producir salidas conscientes.

Generalizando: Extender o aumentar algo. Las ANN se generalizan automáticamente debido a su propia estructura y naturaleza. Estas redes pueden ofrecer, en un margen adecuado, respuestas a los insumos que presentan poca variación debido a los efectos del ruido o la distorsión (Ponce-Cruz, 2010; Cárdenas y Prieto-Ortiz, 2015).

Para el diseño del brazo robótico, tomamos en cuenta el movimiento que se va a realizar, son movimientos lineales como levantar y bajar el brazo, doblar la mano derecha y abrir y cerrar la mano; Vamos a tener tres grados de libertad, uno para el movimiento del brazo, otro para la flexión de la derecha y el último para el movimiento de la mano.

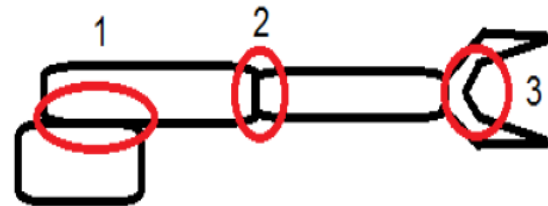


Fig. 3. Diagrama básico del brazo robótico.

Para el diseño de piezas se utilizó el software Solidwork 2015, estas piezas se cortaron en acrílico de 3 mm de espesor. Todo esto se muestra a continuación.

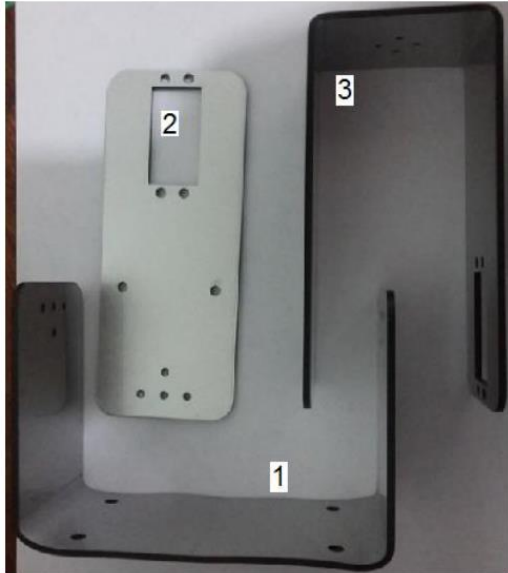


Fig. 4. Piezas recortadas en acrílico.



Fig. 5. Abrazadera de aleación de aluminio.

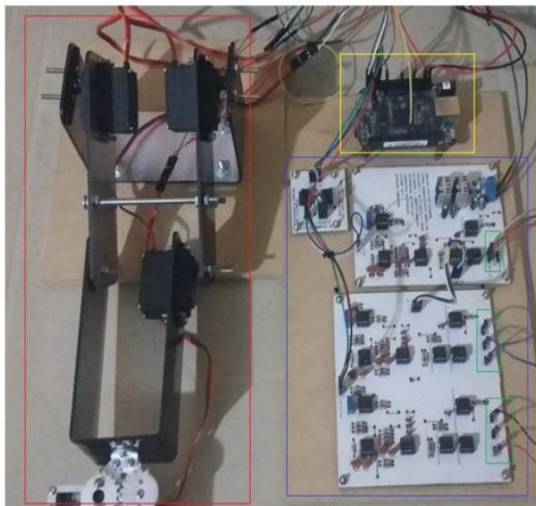


Fig. 6. Imagen de todo el sistema acoplado.

En la Figura 6 podemos observar el sistema con todos los implementos acoplados. En el cuadro rojo se puede observar el brazo robótico que recibe la señal de control PWM de Beaglebone Black que ejecuta el movimiento respectivo.

En la Figura 7 se encuentra el sistema de señales de adquisición que se encarga de capturar las señales de EMG del brazo humano y realiza la renovación de la señal para enviarla al ADC del Beaglebone Black, que se encarga del procesamiento de la información y la clasificación de las señales para determinar el movimiento para ejecutar. En la figura 7 podemos encontrar los pines de entrada del sistema de adquisición donde están conectados los electrodos.

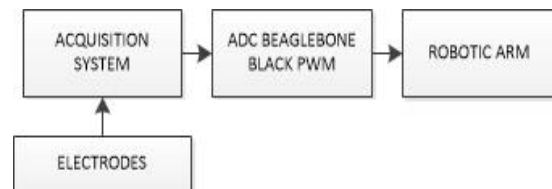


Fig. 7. Diagrama de bloques del hardware sistema.

MATLAB tiene la caja de herramientas ANN que permite trabajar con redes neuronales artificiales de múltiples capas (Ramírez-Corzo y Mendoza, 2016; Vera-Dávila, Delgado-Ariza y Sepúlveda-Mora, 2018). En un algoritmo se asignan los dígitos, vectores o matrices con los que se entrenará la red y aquellos con los que será válida. La cantidad de neuronas a utilizar, la función de activación de cada una de las capas, la interacción entre neuronas, etc., dependerán de la configuración del usuario. Una vez que la caja de herramientas finaliza el proceso, entrega toda la información de la red en una estructura llamada Net; a partir de ahí, es donde se obtienen los pesos y umbrales para llevarlos a la fuente.

```

clc
clear all, close all;
load('datos2.mat')
input('Carga de Data completada');
a=datos([1:160],:);%Entradas a la red, Vector de
error adsoluto.
b=salida(1:160);
dd=min(a);% siempre que exista una matriz es
necesario realizar este paso
ee=max(a);
ff=[dd; ee];
ff=ff';
efectividad=0;
while(efectividad<95)
net=newff(ff,[10 1],{'tansig' 'purelin'}); % crea
la red con 10 neuronas
net=trainlm(net,a',b');% entrenar
aaa=sim(net,a'); %valida con las int y la red creada
aaa=round(aaa);
error=0;
for(i=1:160)
if(aaa(i)==b(i))
error=error+1;
end
end
efectividad=((error/160)*100)
end

```

Fig. 8. Algoritmo red neuronal MLP por partes.

En la figura ocho se muestra el código implementado en MATLAB para entrenar la red neuronal, y los comentarios a continuación explicarán qué sucede en cada parte del código. Una vez encontrados los pesos que garantizan al menos el 95% de efectividad, la caja de herramientas de MATLAB se detiene y envía el resultado a la red; Este tenía el 100% de efectividad.

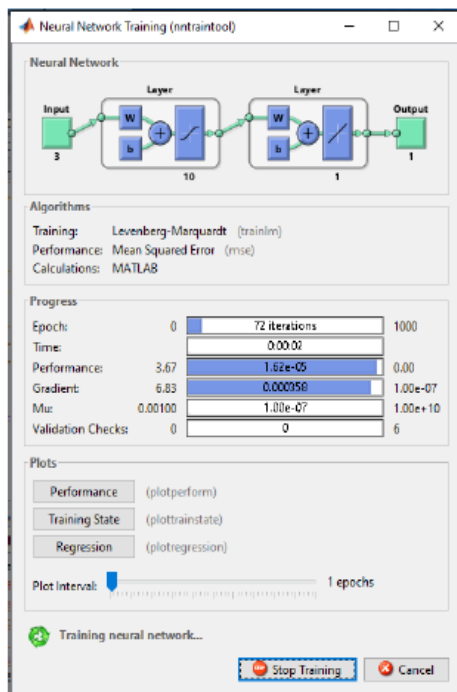


Fig. 9. Entrenamiento de la red neuronal.

3. PROGRAMACIÓN Y DESARROLLO

La estructura principal del programa, como se mencionó anteriormente, tiene tres procesos principales; todos ellos ejecutados en un único Script implementado en el Beaglebone Black escrito en el lenguaje de programación Python (Challenger-Pérez, Díaz-Ricardo y Becerra-García, 2014).

Dejemos que sigan los procesos: 1- adquisición de datos, 2- inteligencia artificial, 3- ejecución de movimientos.

El sistema de adquisición entrega la señal EMG condicionada al ADC de la placa de desarrollo, el programa principal toma 750 muestras por segundo de la señal EMG, se toman 3000 muestras en total, luego el canal 1 toma 1000 muestras de la señal EMG del largo Músculo del palmeral, este se estimula cuando la mano se abre y se cierra. El canal 2 toma 1000 muestras de la señal de EMG de los bíceps y tríceps que se estimulan cuando se dobla la derecha y el canal 3 toma 1000 muestras del músculo deltoides que se estimula con el movimiento del brazo hacia arriba y hacia abajo.

Cada una de 1000 muestras se mantiene en un vector que mantiene la señal de EMG de diferentes músculos; para cada uno de los vectores de señal de EMG, los valores máximo y mínimo se calculan para establecer la diferencia entre esos valores y cada resultado se mantiene en un vector llamado Error relativo; para este último vector, el promedio de sus valores se calcula y terminará en un valor llamado Error absoluto.

Hay 3 errores absolutos: el error absoluto del canal 1, el error absoluto del canal 2 y el error absoluto del canal 3; esos valores son entradas de la red neuronal para la clasificación del movimiento.

La red neuronal tiene 3 entradas y la salida tiene una, la capa oculta tiene diez neuronas y la capa de salida tiene una, la función de activación para la primera es Tansig y para la salida es Purein. Una vez que el valor extraído viene en el proceso anterior, este se pasa por las primeras diez neuronas donde se multiplican por sus pesos, se agrega el umbral y se calcula el Tansig, la salida de cada neurona de la capa oculta es una entrada del neurona de salida. El Beaglebone Black posee un punto flotante y Python tiene bibliotecas matemáticas que permiten el uso directo de la función exponencial, una función que permite calcular la función sigmoide hiperbólica tangente

tangente, que permite implementar la red neuronal en esta placa de desarrollo. Lo que la red hace en el proceso es la clasificación del movimiento a través del patrón obtenido en los cuatro tipos; tipo 1 significa que se mantiene la posición de referencia, tipo 2 significa la posición del brazo, tipo 3 significa el movimiento de derecha y tipo 4 significa el movimiento de la mano.

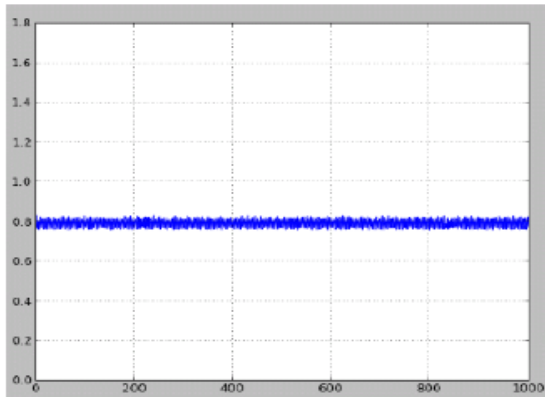


Fig. 10. Posición de referencia.

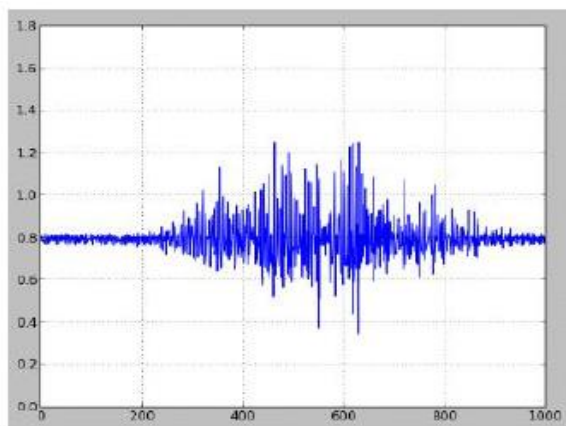


Fig. 11. Movimiento del brazo.

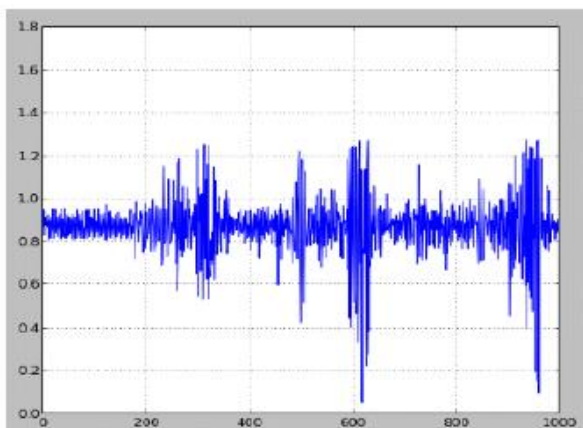


Fig. 12. movimiento a la derecha.

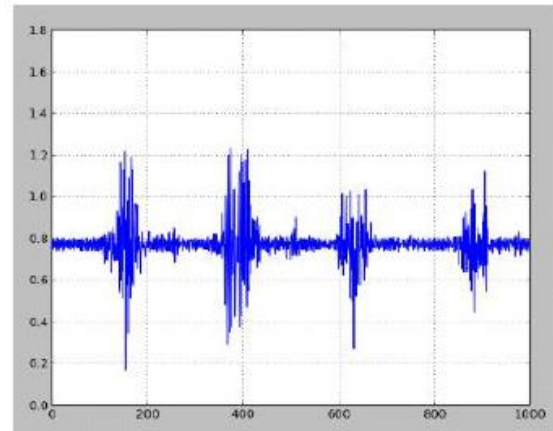


Fig. 13. Movimiento de la mano.

Ejecución del movimiento: Tan pronto como la red neuronal clasifica el movimiento, el Beaglebone Black envía una señal PWM a los servomotores del brazo robótico para realizar el movimiento respectivo. Por ejemplo, si la clasificación de red era de tipo 1, se realiza cualquier movimiento porque se mantiene la posición de referencia; y si la clasificación de red es de tipo 4, el brazo robótico realiza un movimiento de la mano (apertura y cierre) y luego vuelve a la posición de referencia para la siguiente clasificación de red.

4. RESULTADOS

A continuación, los resultados de la clasificación de la red neuronal implementada en Python.

Tabla 1: Validación de resultados con datos.

Movimiento	Tipo de muestra	Resultado
Posición de referencia	Tipo 1	Tipo 1
Brazo	Tipo 2	Tipo 2
Mano	Tipo 4	Tipo 4
Derechazo	Tipo 3	Tipo 3
Mano	Tipo 4	Tipo 4
Posición de referencia	Tipo 1	Tipo 1
Brazo	Tipo 2	Tipo 2
Mano	Tipo 4	Tipo 4
Brazo	Tipo 2	Tipo 2
Derechazo	Tipo 3	Tipo 3

El porcentaje de error obtenido fue del 0%; Al reunir todos los algoritmos con este resultado, los resultados obtenidos en una prueba en tiempo real serán los mismos. Las pruebas se realizan en tiempo real, trabajando con el código final y los resultados se muestran a continuación.

Tabla 2: Resultados validados con el sistema implementado.

Movimiento	Tipo de muestra	Resultado
Mano	Tipo 4	Tipo 4
Brazo	Tipo 2	Tipo 2
Posición de referencia	Tipo 1	Tipo 1
Derechazo	Tipo 3	Tipo 3
Mano	Tipo 4	Tipo 4
Posición de referencia	Tipo 1	Tipo 1
Brazo	Tipo 2	Tipo 2
Mano	Tipo 4	Tipo 4
Derechazo	Tipo 3	Tipo 3
Brazo	Tipo 2	Tipo 2
Derechazo	Tipo 3	Tipo 3

Los resultados fueron favorables con un porcentaje de error del 0%, lo que indica una operación ideal del sistema.

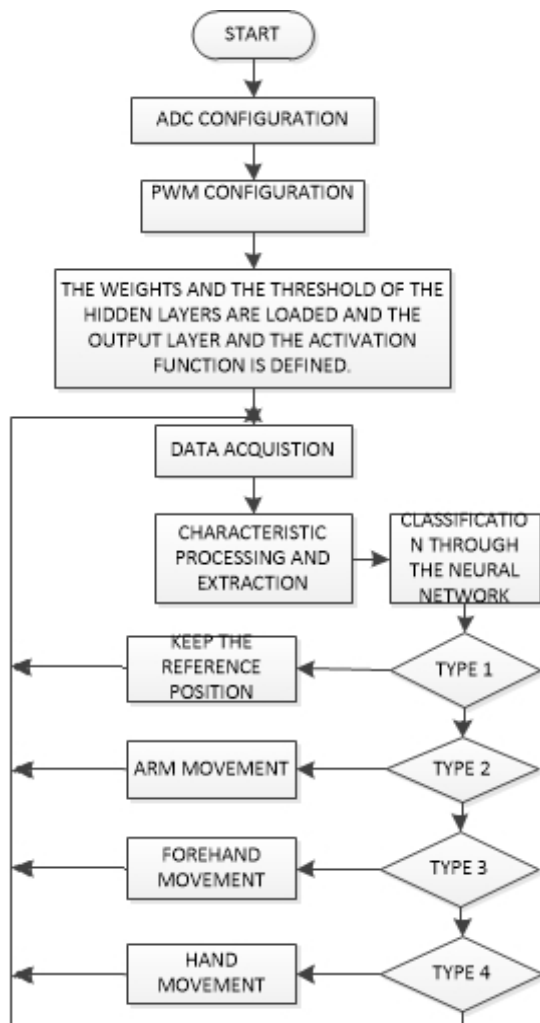


Fig. 14. Programa principal.

5. CONCLUSIONES

Se eligió el tablero Beaglebone Black debido a que es una herramienta poderosa para aplicaciones en instrumentos electrónicos, por lo tanto, permite trabajar en diferentes entornos de programación,

especialmente en Python, que es uno de los lenguajes más utilizados para la inteligencia artificial en sistemas integrados. Se diseñó e implementó un sistema de adquisición de señales EMG, que permite realizar la adquisición y el acondicionamiento de la señal, también permite la transmisión a través del panel de desarrollo Beaglebone Black de señales electrográficas del brazo humano para procesar y extraer patrones y generar datos que Sirve como soporte para la formación en red.

Fue capaz de identificar que, para el movimiento del brazo, el músculo deltoides es el más estimulado, para la flexión hacia la derecha, el bíceps y el tríceps son los músculos más estimulados, y para la mano, el palmar largo es el más estimulado, por eso La ubicación de los electrodos es vital en la adquisición de la señal EMG.

Para procesar las señales de EMG del movimiento del brazo, la flexión de derecha, abrir y cerrar la mano; No es necesario aplicar técnicas complejas de procesamiento como la transformada de Fourier, la transformada Discret Coseno, la transformada Wavelet, etc. La aplicación de esas técnicas desperdicia recursos útiles de la placa de desarrollo, porque con técnicas simples como la resta, máximos y mínimos, promedio, etc. Se pueden extraer patrones para clasificar esas señales.

La red neuronal se implementó en el sistema integrado con la ayuda de Beaglebone Black, también con la versatilidad y optimización de las bibliotecas de Python para el mismo propósito. La capacitación de la red se realizó con los datos obtenidos de las señales de EMG del brazo a través de la MLP, que obtuvo el 100% de efectividad en la validación del software MATLAB. Una de las limitaciones fue que el sistema solo puede realizar dos movimientos simultáneamente, como flexionar el brazo y abrir y cerrar la mano, la red puede dar resultados erróneos o clasificar solo uno de los dos movimientos. Se implementó un brazo robótico de tres grados de libertad, se acopló todo el sistema, luego se probó todo el sistema y se obtuvo un porcentaje de error del 0%; El sistema clasifica con éxito el movimiento para controlar el brazo robótico.

REFERENCIAS

Cárdenas, J. A., & Prieto-Ortíz, F. A. (2015). Diseño de un algoritmo de corrección automática

de posición para el proceso de perforado PCB, empleando técnicas de visión artificial. *Revista de Investigación, Desarrollo e Innovación*, 5 (2), 107-118. doi: 10.19053/20278306.3720

Cera-Martínez, D., Ortiz-Sandoval, J. E., & Gualdrón-Guerrero, O. E. (2018). Sintonización de un controlador de temperatura a través de un autómata programable. *Revista de Investigación, Desarrollo e Innovación*, 9 (1).

Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, 20 (2). Recuperado de: http://www.ciencias.holguin.cu/index.php/ciencias_holguin/article/view/826

Durán-Acevedo, C., & Jaimes-Mogollón, A. (2013). Optimización y clasificación de señales EMG a través de métodos de reconocimiento de patrones. *ITECKNE*, 10 (1), 67-76. doi: <http://dx.doi.org/10.15332/iteckne.v10i1.181>

Enciso, L., Zelaya, E., & Quezada-sarmiento, P. A. (2018). Análisis y simulación de herramientas gramaticales: L-System y Vlab para determinar el nivel de crecimiento de una planta. *Espacios*, 39 (23), 28. Recuperado de: <http://www.revistaespacios.com/a18v39n23/18392328.html>

García-Pinzón, J. A., Mendoza, L. E., & Flórez, E. G. (2015). Control de brazo electrónico usando señales electromiográficas. *Facultad de Ingeniería*, 24 (39), 71-83.

Gualdrón-Guerrero, O. E., Durán-Acevedo, C. M., Ortiz-Sandoval, J. E., & Araque-Gallardo, J. A. (2014). Implementación de un modelo neuronal en un dispositivo hardware (FPGA) para la clasificación de compuestos químicos en un sistema multisensorial (nariz electrónica). *Revista Colombiana de Tecnologías de Avanzada*, 2 (24), 127-133. doi: <https://doi.org/10.24054/16927257.v24.n24.2014.2347>

Herrera-Baquero, L. P., & Prieto-Ortiz, F. A. (2018). Metodología para la inspección de la herramienta en el taladrado de piezas. *Revista de Investigación, Desarrollo e Innovación*, 9 (1).

López-Castellanos, J. M. (2017). Implementación y evaluación de un esquema de control mioeléctrico ON/OFF utilizando hardware de bajo coste (Tesis de Maestría). Alicante, España: Universidad de

Alicante. Recuperado de: <http://hdl.handle.net/10045/6835>

Marino-Vera, H., Mendoza, L., & Gualdrón-Guerrero, O. (2017). Medición automática de variables antropométricas para la evaluación de la respiración usando visión artificial. *Revista de Investigación, Desarrollo e Innovación*, 8 (1), 161-169. doi: <https://doi.org/10.19053/20278306.v8.n1.2017.7407>

Martínez, G. M., Jáquez, S. A., Rivera, J., & Sandoval, R. (2008). Diseño propio y Construcción de un Brazo Robótico de 5 GDL. *RIEE&C: Revista de Ingeniería Eléctrica, Electrónica y Computación*, 4 (1), 1-7.

Niño-Vega, J. A., Martínez-Díaz, L. Y., & Fernández-Morales, F. H. (2016). Mano robótica como alternativa para la enseñanza de conceptos de programación en Arduino. *Revista Colombiana de Tecnologías de Avanzada*, 2 (28), 132-139. doi: <https://doi.org/10.24054/16927257.v28.n28.2016.2476>

Ortiz-Sandoval, J. E., Marín-Herrera, A., & Gualdrón-Guerrero, O. E. (2016). Implementation of a banking system security in embedded systems using artificial intelligence. *Advances In Natural And Applied Sciences*, 10 (17), 95-102.

Ponce-Cruz, P. (2010). *Inteligencia Artificial con Aplicaciones a la Ingeniería*. México: Alfaomega.

Ramírez-Corzo, J., & Mendoza, L. (2016). Desarrollo de un sistema de comunicación silenciosa dual basado en habla subvocal y Raspberry Pi. *Facultad de Ingeniería*, 25 (43), 111-121. doi: <https://doi.org/10.19053/01211129.v25.n43.2016.5304>

Sánchez-Dams, R. D. (2013). Estado del arte del desarrollo de sistemas embebidos desde una perspectiva integrada entre el hardware y software. *Revista Colombiana de Tecnologías de Avanzada*, 2 (22), 98-105. doi: <https://doi.org/10.24054/16927257.v22.n22.2013.416>

Sandoval-Ruiz, C. (2014). Adaptive Control in VHDL Applied to a Solar Oven. *Revista Colombiana de Tecnologías de Avanzada*, 2 (20), 80-85. doi:

<https://doi.org/10.24054/16927257.v23.n23.2014.2323>

Texas Instruments (2016). ARM Cortex-M4F based MCU TM4C1294 Connected LaunchPad. Recuperado de: <http://www.ti.com/tool/ek-tm4c1294xl>. [Citado el 30 de Octubre de 2015].

Vera-Dávila, A., Delgado-Ariza, J., & Sepúlveda-Mora, S. (2018). Validación del modelo matemático de un panel solar empleando la herramienta Simulink de Matlab. *Revista de Investigación, Desarrollo e Innovación*, 8 (2), 343-356. doi: <https://doi.org/10.19053/20278306.v8.n2.2018.7972>