



UNIVERSIDAD DE PAMPLONA
Una universidad incluyente y comprometida
Con el desarrollo integral



PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

Autor:

Alejandro Andrés Cantillo Sánchez

Director:

**MSc. Jesús Eduardo Ortiz Sandoval
MSc. (C). Luis Alberto Muñoz Bedoya**

**Ingeniería Electrónica
Departamento de Ingenierías Eléctrica, Electrónica, Sistemas y
Telecomunicaciones
Facultad de Ingenierías y Arquitectura
Universidad de Pamplona
Pamplona, 24 de junio de 2016**

**Universidad de Pamplona
Facultad de Ingenierías y Arquitectura
Departamento de Ingenierías Eléctrica, Electrónica,
Sistemas y Telecomunicaciones
Programa de Ingeniería Electrónica
Trabajo presentado para optar por el título de
Ingeniero Electrónico**

Tema:

**Procesamiento de Señales EMG en un Sistema Embebido para el
Control Neuronal de un Brazo Robótico**

**Fecha de inicio del trabajo: 2 de mayo 2015
Fecha de terminación del trabajo: 27 de mayo 2016**

Nombres y firmas de autorización para la sustentación:

**Alejandro Andrés Cantillo Sánchez
Autor**

**Jesús Eduardo Ortiz
Director**

**Judith Cristancho Pabón
Directora de programa**

Jurado calificador:

Oscar Gualdrón

Julio C. Ospino

Jesús Ortiz

**Pamplona - Norte de Santander
Colombia
Junio 24 de 2016**

Para Ustedes

*Marfelina Sánchez Urrea, Julio Cesar Cantillo Perdomo,
Andreina Cantillo Sánchez, Juliana Cantillo Sánchez*

Eterna Gratiud

INDICE

PREFACIO.....	IX
AGRADECIMIENTOS.....	XI
INTRODUCCIÓN.....	- 1 -
1.1 PROBLEMA.....	- 2 -
1.2 OBJETIVOS.....	- 3 -
1.3 RESUMEN	- 4 -
1.4 DISTRIBUCIÓN BIBLIOGRÁFICA.....	- 4 -
2. MARCO TEÓRICO.....	- 5 -
2.1 ANTECEDENTES.....	- 7 -
2.2 SISTEMA MUSCULAR: MUSCULATURA DE LAS EXTREMIDADES	- 10 -
2.2.1 MÚSCULOS QUE MUEVEN EL BRAZO	- 10 -
2.2.2 MÚSCULOS QUE MUEVEN EL ANTEBRAZO Y LA MANO	- 12 -
2.3 ELECTROMIOGRAFÍA.....	- 13 -
2.3.1 ELECTROMIOGRAFÍA DE SUPERFICIE.....	- 14 -
2.3.2 ELECTRODOS SUPERFICIALES.....	- 14 -
2.3.3 ASPECTOS IMPORTANTES.....	- 14 -
2.4 FUNDAMENTOS ELECTRÓNICOS.....	- 15 -
2.4.1 AMPLIFICADOR OPERACIONAL.....	- 15 -
2.4.2 SEGUIDOR DE VOLTAJE	- 16 -
2.4.3 AMPLIFICADOR INVERSOR.....	- 16 -
2.4.4 AMPLIFICADOR NO INVERSOR.....	- 17 -
2.4.5 AMPLIFICADOR DE INSTRUMENTACIÓN	- 17 -
2.4.6 FILTROS ACTIVOS	- 19 -
2.5 REDES NEURONALES.....	- 21 -
2.5.1 TOPOLOGÍAS DE LAS REDES NEURONALES.....	- 24 -
2.5.2 ELEMENTOS DE UNA RED NEURONAL MULTICAPA	- 24 -
2.6 TARJETAS DE DESARROLLO.....	- 26 -

2.7 LENGUAJE DE PROGRAMACIÓN EN PYTHON.....	- 29 -
2.8 BRAZO ROBÓTICO.....	- 31 -
3. DISEÑO DE COMPONENTES	- 33 -
3.1 SELECCIÓN DEL DISPOSITIVO.....	- 35 -
3.2 SISTEMA DE ADQUISICIÓN DE SEÑALES	- 37 -
3.2.1 SENSORES	- 37 -
3.2.2 AMPLIFICADOR DE INSTRUMENTACIÓN	- 38 -
3.2.3 ACONDICIONAMIENTO	- 40 -
3.2.3.1 FILTRO PASABANDA.....	- 40 -
3.2.3.2 SUMADOR DE VOLTAJE.....	- 43 -
3.2.3.3 LIMITADOR DE AMPLITUD.....	- 44 -
3.2.4 TARJETA DE ADQUISICIÓN IMPLEMENTADA.....	- 44 -
3.3 BRAZO ROBÓTICO.....	- 47 -
3.4 SISTEMA CON HARDWARE IMPLEMENTADO	- 51 -
4. PROGRAMACIÓN Y DESARROLLO	- 53 -
4.1 CARACTERÍSTICAS DEL SISTEMA	- 54 -
4.1.1 CONFIGURACIÓN DEL HARDWARE.....	- 55 -
4.1.2 CONFIGURACIÓN DEL SOFTWARE	- 56 -
4.2 CAPTURA DE DATOS Y EXTRACCIÓN DE PATRONES.....	- 56 -
4.3 RED MLP EN MATLAB.....	- 59 -
4.3.1 PESOS DE LAS NEURONAS Y UMBRALES	- 61 -
4.3.2 RESULTADOS DE ENTRENAMIENTO	- 63 -
4.4 ESTRUCTURA DEL PROGRAMA PRINCIPAL EN PYTHON	- 64 -
5. ANÁLISIS Y RESULTADOS	- 67 -
5.1 ANALISIS Y RESULTADOS	- 68 -
5.2 ADC DE LA BEAGLEBONE BLACK	- 68 -
5.3 RED NEURONAL EN PYTHON.....	- 69 -
5.3.1 LIMITANTES	- 70 -
5.3.2 RESULTADOS Y PORCENTAJE DE ERROR	- 70 -
5.4 SISTEMA IMPLEMENTADO.....	- 71 -
5.5 ANÁLISIS ECONÓMICO	- 72 -

6. CONCLUSIONES..... - 74 -

6.1 CONCLUSIONES FINALES - 75 -

BIBLIOGRAFÍA - 77 -

ÍNDICE DE FIGURAS

<i>Figura 1. Músculos que mueven el brazo vista anterior</i> [9]	- 11 -
<i>Figura 2. Músculos que mueven el brazo vista posterior</i> [9]	- 11 -
<i>Figura 3. Músculos que mueven el antebrazo y la mano</i> [9].....	- 13 -
<i>Figura 4. Electrodo superficial de gel adhesivo marca 3M. Ag/AgCl.</i>	- 14 -
<i>Figura 5. Esquema eléctrico amplificador operacional.</i>	- 15 -
<i>Figura 6. Seguidor de voltaje.</i> [12].....	- 16 -
<i>Figura 7. Amplificador inversor.</i> [12].....	- 16 -
<i>Figura 8. Amplificador no inversor.</i> [12]	- 17 -
<i>Figura 9. Amplificador de instrumentación.</i> [12]	- 18 -
<i>Figura 10. Filtro pasabajas</i> [11].....	- 20 -
<i>Figura 11. Filtro pasaaltas</i> [11].....	- 20 -
<i>Figura 12. Filtro pasabanda</i> [11].....	- 21 -
<i>Figura 13. Neurona biológica.</i> [13].....	- 22 -
<i>Figura 14. Neurona Artificial.</i> [14].....	- 23 -
<i>Figura 15. Estructura de una Red MLP.</i> [14].....	- 24 -
<i>Figura 16. Tarjeta de desarrollo Raspberry pi 2.</i> [15].....	- 27 -
<i>Figura 17. Tarjeta de desarrollo Beaglebone Black.</i> [16].....	- 28 -
<i>Figura 18. Tarjeta de desarrollo Tiva C TM4C1294.</i> [17].....	- 29 -
<i>Figura 19. Brazo robótico.</i> [19].....	- 31 -
<i>Figura 20. Diagrama en bloques sistema de adquisición.</i>	- 37 -
<i>Figura 21. Conectores para electrodos y electrodos 3M.</i>	- 38 -
<i>Figura 22. Circuito de adquisición etapa Amp. Instrumentación.</i>	- 39 -
<i>Figura 23. Circuito para tomar la data.</i>	- 41 -
<i>Figura 24. Regresión potencial.</i>	- 42 -
<i>Figura 25. Filtro pasaalta de 4 orden.</i>	- 43 -
<i>Figura 26. Sumador de Voltaje.</i>	- 44 -
<i>Figura 27. Tarjeta de adquisición canal 1 en ARES.</i>	- 45 -
<i>Figura 28. Tarjeta de adquisición canal 2 y 3 en ARES.</i>	- 45 -
<i>Figura 29. Tarjeta de adquisición canal 1.</i>	- 46 -
<i>Figura 30. Tarjeta de adquisición canal 2 y 3.</i>	- 46 -
<i>Figura 31. Diagrama básico de brazo robótico.</i>	- 47 -
<i>Figura 32. Partes cortadas en acrílico.</i>	- 48 -
<i>Figura 33. Pinza de aleación de aluminio.</i>	- 48 -
<i>Figura 34. Diagrama para calcular el torque.</i>	- 49 -

Figura 35. <i>Brazo robótico implementado</i>	- 50 -
Figura 36. <i>Imagen donde se muestran todo el sistema acoplado</i>	- 51 -
Figura 37. <i>Diagrama de bloques del hardware del sistema</i>	- 52 -
Figura 38. <i>Diagrama de flujo del sistema</i>	- 55 -
Figura 39. <i>Diagrama de flujo captura y extracción de patrón de la señal EMG</i>	- 57 -
Figura 40. <i>Señal EMG movimiento de la mano</i>	- 58 -
Figura 41. <i>Señal EMG movimiento del antebrazo</i>	- 58 -
Figura 42. <i>Señal EMG movimiento del brazo</i>	- 59 -
Figura 43. <i>Señal EMG posición de referencia</i>	- 59 -
Figura 44. <i>Algoritmo Red MLP por partes</i>	- 60 -
Figura 45. <i>Entrenamiento de la red neuronal</i>	- 61 -
Figura 46. <i>Función tangente hiperbólica sigmoidea [13]</i>	- 65 -
Figura 47. <i>Programa principal</i>	- 66 -
Figura 48. <i>Esquema de prueba para el ADC</i>	- 68 -
Figura 49. <i>Circuito de prueba para el ADC implementado</i>	- 69 -
Figura 50. <i>Sistema implementado realizando una validación</i>	- 72 -

INDICE DE TABLAS

Tabla 1. <i>Músculos que mueven el antebrazo y la mano [9]</i>	- 12 -
Tabla 2. <i>Filtros ideales [12]</i>	- 19 -
Tabla 3. <i>Características ADC (SAR) [21]</i>	- 37 -
Tabla 4. <i>Amplificadores de Instrumentación para aplicaciones médicas [7]</i>	- 38 -
Tabla 5. <i>Datos capacitancia vs frecuencia de corte</i>	- 41 -
Tabla 6. <i>Valores de los pesos y umbrales de las neuronas de la capa oculta</i>	- 62 -
Tabla 7. <i>Valores de los pesos y umbral de la capa de salida</i>	- 63 -
Tabla 8. <i>Resultados validación en Matlab</i>	- 63 -
Tabla 9. <i>Resultados prácticos ADC</i>	- 69 -
Tabla 10. <i>Recursos utilizados por la Beaglebone Black</i>	- 70 -
Tabla 11. <i>Resultados validación con datas</i>	- 71 -
Tabla 12. <i>Resultados validación con sistema implementado</i>	- 71 -
Tabla 13. <i>Inversión del proyecto</i>	- 73 -

PREFACIO

PREFACIO.....IX
AGRADECIMIENTOS.....XI

“¿Has visto hombre solícito en su trabajo? Delante de los reyes estará; No estará delante de los de baja condición”

Proverbios 22:29

AGRADECIMIENTOS

En primer lugar quiero dar gracias a Dios por darme la oportunidad de vivir, de tener una maravillosa familia, de estudiar y por permitirme realizar este proyecto, muchas gracias Dios no me alcanza la vida para agradecerte todo lo que has hecho.

Le agradezco a mi madre Marfelina Sánchez Urrea, la cual ha sido el apoyo más importante de mi vida; gracias por brindarme la oportunidad de estudiar, por ser ese apoyo incondicional, por ser esa voz de aliento, gracias por su amor incondicional. Además quiero dedicar este trabajo a mi padre Julio Cesar Cantillo Perdomo sé que siempre estas a mi lado, que mi Dios te tenga en su gloria.

También a tío Davinson Pedrozo por su apoyo, gracias por ser un padre, a mis dos hermanas Andreina Cantillo Sánchez y Juliana Cantillo Sánchez gracias por existir, gracias por cada momento vivido. Agradecimiento especial a mi abuela Idilia Urrea y a mis tías Socorro Urrea, Bleida Sánchez y Rut Urrea, gracias por su apoyo. Además agradecer a la colaboración de Edier Sánchez primo hermano.

Agradezco a los ingenieros Jesús Ortiz Sandoval y Luis Alberto Muñoz por su apoyo y colaboración en este proyecto como directores de trabajo de grado, gracias por creer en este proyecto, que honor tenerlos por directores; y también agradecer a todo el profesorado del programa de ingeniería electrónica el cual me formo académicamente como ingeniero electrónico.

También al ingeniero Julio Cesar Ospino y Doctor Oscar Eduardo Gualdrón gracias por su ayuda, son profesionales íntegros y talentosos lo cual fue un honor tenerlos como jurados de mi trabajo de grado.

INTRODUCCIÓN

INTRODUCCIÓN.....	- 1 -
1.1 PROBLEMA.....	- 2 -
1.2 OBJETIVOS.....	- 3 -
1.3 RESUMEN.....	- 4 -
1.4 DISTRIBUCIÓN BIBLIOGRÁFICA.....	- 4 -

1.1 PROBLEMA

En Colombia el número de personas discapacitadas es 2.018.078 DANE marzo 2010 – Dirección de Censos y Demografía [1], el 20,48%(413.269) de la población discapacitada son personas que presentan problemas con las extremidades superiores e inferiores, debido a que han sufrido accidentes que provocaron la pérdida de una de sus extremidades o nacieron sin una de ellas. Estas personas presentan dificultades para realizar algunas tareas cotidianas como lo es tomar un objeto con la mano, abrir y cerrar la mano, mover el brazo, etc.

En respuesta a esta problemática desde hace varios años se comenzó a trabajar en el desarrollo de prótesis que puedan ayudar a las personas a recuperar el miembro faltante de su cuerpo y algunas funciones de este. Con los avances tecnológicos de estos últimos 25 años, las investigaciones para el desarrollo de prótesis eléctricas o electrónicas han aumentado, jugando un papel fundamental las señales EMG y EEG para el desarrollo de estas. Porque son las señales de control de este tipo de prótesis, por lo cual se han desarrollado técnicas de adquisición, acondicionamiento y procesamiento de este tipo de señales en los últimos años.

La mayoría de universidades en Suramérica, han realizado muchos trabajos sobre este tipo de aplicaciones, pero hay un detalle el procesamiento de la información es realizado en un ordenador, por lo cual el sistema, está limitado al laboratorio, lo cual delimita el proceso de comercialización. Surge la necesidad de llevar el procesamiento de la información del ordenador a un sistema embebido (Beaglebone Black) como una posible solución a este inconveniente.

Este proyecto se desarrolla viendo la necesidad de procesar la información en un sistemas embebido, para que el sistema desarrollo no este limitado al laboratorio. Lo que se busca con este proyecto es procesar las señales EMG del brazo, en un sistema embebido en este caso una Beaglebone Black, y dispositivo por medio de una red

neuronal pueda clasificar los diferentes movimientos para controlar un brazo robótico, esto abre una brecha para desarrollo de prótesis electrónicas utilizando sistemas embebido y para otro aplicaciones.

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

- Procesar las señales EMG en un sistema embebido para el control neuronal de un brazo robótico.

1.2.1 OBJETIVOS ESPECÍFICOS

- Adquirir las señales de los músculos responsables del movimiento del brazo, antebrazo y mano.
- Procesar las señales EMG adquiridas.
- Entrenar una red neuronal con la capacidad de clasificar diferentes movimientos de por lo menos 3 grados de libertad.
- Validar la red neuronal entrenada.
- Implementar en un sistema electrónico embebido la red neuronal con la data de las señales EMG.
- Implementar un brazo robótico de al menos 3 grados de libertad.
- Validar el sistema desarrollado.

1.3 RESUMEN

El objetivo de este trabajo es el de adquirir las señales EMG del brazo humano, para entrenar una red neuronal que pueda clasificar diferentes movimientos, de por lo menos 3 grados de libertad para controlar un brazo robótico. Para ello se diseñó el sistema de adquisición y acondicionamiento de las señales EMG. Se identificaron los músculos responsables del movimiento de la mano, antebrazo y brazo para localizar los electrodos en el musculo indicado. Se partió de una posición de referencia para realizar los movimientos de la mano, antebrazo y brazo. Las señales provenientes del sistema de adquisición y acondicionamiento se enviaron al ADC de la tarjeta de desarrollo, en donde se implementó un algoritmo que analizó el registro de las señales EMG para generar la data con la que se entrenó la red neuronal. Seguidamente se implementó la red neuronal en el sistema embebido (Beaglebone Black Rev. C) y se acopló con el brazo robótico.

1.4 DISTRIBUCIÓN BIBLIOGRÁFICA

El libro se organiza mediante cinco capítulos concernientes a la Introducción, Marco Teórico, Diseño de Componentes, Programación, Resultados y Conclusiones.

Cada capítulo tiene los ítems correspondientes al tema de interés.

2. MARCO TEÓRICO

2.MARCO TEÓRICO.....	5 -
2.1 ANTECEDENTES.....	7 -
2.2 SISTEMA MUSCULAR: MUSCULATURA DE LAS EXXTREMIDADES.....	10 -
2.2.1 MÚSCULOS QUE MUEVEN EL BRAZO.....	10 -
2.2.2 MÚSCULOS QUE MUEVEN EL ANTEBRAZO Y LA MANO.....	12 -
2.3 ELECTROMIOGRAFÍA.....	13 -
2.3.1 ELECTROMIOGRAFÍA DE SUPERFICIE.....	14 -
2.3.2 ELECTRODOS SUPERFICIALES.....	14 -
2.3.3 ASPECTOS IMPORTANTES.....	15 -
2.4 FUNDAMENTOS ELECTRÓNICOS.....	15 -
2.4.1 AMPLIFICADOR OPERACIONAL.....	15 -
2.4.2 SEGUIDOR DE VOLTAJE.....	16 -
2.4.3 AMPLIFICADOR INVERSOR.....	17 -
2.4.4 AMPLIFICADOR NO INVERSOR.....	17 -

2.4.5 AMPLIFICADOR DE INSTRUMENTACIÓN.....	18 -
2.4.6 FILTROS ACTIVOS.....	19 -
2.5 REDES NEURONALES.....	22 -
2.5.1 TOPOLOGIAS REDES NEURONALES.....	25 -
2.5.2 ELEMENTOS DE UNA RED NEURONAL MULTICAPA.....	25 -
2.6 TERJETAS DE DESARROLLO.....	27 -
2.7 LENGUAJE DE PROGRAMCIÓN PYHTON.....	31 -
2.8 BRAZO ROBÓTICO.....	32 -

2.1 ANTECEDENTES

El estudio de las señales EMG es un campo de investigación que se ha venido explotando fuertemente desde hace más 20 años en las universidades del mundo, se encuentran muchos trabajos académicos relacionados con la adquisición y el procesamiento de las señales EMG, para aplicaciones médicas como prótesis electrónicas y manos mecánicas. También se encuentran trabajos académicos sobre redes neuronales, por lo cual el sistema que se desarrolla en este trabajo tiene fuertes antecedentes; se pueden apreciar varios trabajos académicos en esta área de los cuales podemos citar los siguientes:

SISTEMA DE CONTROL DE MOVIMIENTO PARA PRÓTESIS DE MANO.

El trabajo consiste en la adquisición de señales mioeléctricas (EMG), estas señales son los comando de apertura y cierre de la prótesis de mano, este sistema es implementado en un computador donde se construye una interfaz con entorno Windows para el manejo del prototipo de prótesis de mano. [2]

SISTEMA DE IDENTIFICACIÓN DE INTENCIÓN DE MOVIMIENTO PARA EL CONTROL MIOELÉCTRICO DE UNA PRÓTESIS DE MANO ROBÓTICA.

Este artículo presenta el desarrollo de un sistema de adquisición de señales EMG que detecta la intención de movimiento para el control de un prototipo de prótesis mano robótica. El trabajo muestra que usando características temporales de cálculo simple es posible obtener un agrupamiento de vectores lo suficientemente discriminante como para que se puedan usar clasificadores de patrones muy simples. La técnica utilizada para la captación de las señales es no invasiva y el hardware utilizado para procesar las señales es un dsPIC a una frecuencia de 40MHz. Primero se captan las señales en el ADC del dsPIC y se envían a una computadora por medio del puerto USB en donde se registran los datos en Lab View, después del registro de los datos se aplican las técnicas de cálculo simple para llegar al algoritmo clasificador de los movimientos, y este es implementado en el dsPIC. [3]

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA PARA LA DETECCIÓN DE SEÑALES ELECTROMIOGRÁFICAS.

Este trabajo muestra el diseño de dos sistemas de adquisición de señales EMG específicamente las señales de los músculos del brazo, el primer sistema era con fuente bipolar y un solo canal, mientras que el segundo sistema era con fuente unipolar y un canal LVTTTL este sistema produce pulsos de sistemas LVTTTL (Low Voltage Transistor–Transistor Logic), en ambos sistemas la conversión analógica a digital se realiza mediante un microcontrolador el PIC16F687 y que envía los datos a una computadora donde se estudian las señales EMG en el software de Lab View y se comparan los dos sistemas para determinar el más eficiente. [4]

A REAL-TIME EMG PATTERN RECOGNITION METHOD FOR VIRTUAL MYOELECTRIC HAND CONTROL.

Este estudio propone un modelo de enfoque de reconocimiento en tiempo real electromiografía (EMG) para el control de multifunción manos mioeléctrica. En las técnicas, el tiempo y la información de frecuencia se extraen por paquetes transformada wavelet (WPT) y la energía de los coeficientes nodo WPT se selecciona como la función de las señales de EMG. A continuación, se desarrolla un nuevo método de selección de características en base a un algoritmo recursivo de búsqueda en profundidad para que las funciones de alta dimensión puedan reducir mediante un algoritmo de reducción característica supervisada. En consecuencia, la máquina de soporte vectorial (SVM) se adaptó para dar el resultado del reconocimiento. En el experimento, el sistema de reconocimiento de patrones EMG en tiempo real se ha desarrollado para controlar una mano virtual con señales de EMG del antebrazo. El procesamiento de la información se realiza en una computadora. [5]

ANÁLISIS DE SEÑALES EMG SUPERFICIALES Y SU APLICACIÓN EN CONTROL DE PRÓTESIS DE MANO.

En este artículo, se presenta una revisión de las técnicas más utilizadas en análisis y extracción de características discriminantes de las señales motoras EMGs con fines de aplicación en el control de prótesis virtuales de mano; haciendo énfasis en la evolución de las técnicas temporales clásicas a las técnicas espectrales más recientes

basadas en transformadas wavelets, y su aceptación para procesar este tipo de señales según su desempeño en la clasificación y ejecución de movimientos sobre la prótesis. Así mismo, se hace referencia de los trabajos más recientes de desarrollo e investigación en el campo del procesamiento de señales biomédicas EMGs y sus aplicaciones en control mioeléctrico. [6]

DESARROLLO DE UN SISTEMA DE COMUNICACIÓN SILENCIOSA DUAL BASADO EN HABLA SUBVOCAL.

El objetivo de este trabajo es el de mantener una conversación entre dos personas sin que ninguna de ellas mueva los labios, siendo esta una forma práctica de telepatía denominada Habla Subvocal. Para ello se diseñaron dos dispositivos de adquisición, acondicionamiento y codificación de las minúsculas señales eléctricas (EMG) provenientes de los nervios de las cuerdas bucales y los músculos de la garganta que controlan el habla. La adquisición y el estudio de estas señales electromiográficas es lo que se llama Habla Subvocal. Las señales se enviaron mediante un protocolo de comunicación serial a una tarjeta de desarrollo (Raspberry pi) en donde se implementó un algoritmo que analizó los registros de las señales EMG, procesándolas y extrayéndolas características o los patrones de palabras en el idioma español. Se utilizó la transformada de wavelet para extraer las características de las señales EMG y como técnica de clasificación las máquinas de soporte vectorial. [7]

ELECTROMYOGRAPHY ANALYSIS AND RECOGNITION FOR HUMAN DEVICE INTERFACE.

Este proyecto de ingeniería se enfoca en los movimientos de la mano para controlar el uso de un dispositivo de interfaz humana (HDI). Esto se hace a través del procesamiento, análisis y reconocimiento de las señales de los sensores de electromiografía de superficie (EMG) y un acelerómetro y giroscopios para controlar un HDI. El HDI controlado en este proyecto es un ratón de ordenador y las flechas de un teclado a través de la Makey-Makey (Tarjeta de comunicación por bluetooth). Se implementan el diseño de circuitos electrónicos para filtrar las señales de EMG, los métodos de procesamiento de señales digitales para caracterizar las señales EMG y técnicas de aprendizaje automático para clasificar las señales de EMG en gestos de las manos utilizando redes neuronales artificiales (ANN). Después, basado en el

gesto de la mano evaluado o movimiento, las señales se envían a un Makey-Makey para controlar el ratón y las teclas designadas en el teclado. El sistema, después de la aplicación, es capaz de reconocer cuatro gestos de las manos en aproximadamente ocho segundos. Este resultado permite el control exacto en tiempo real del HDI deseado. La red neuronal es implementada en una Beaglebone Black y se utiliza una tarjeta externa para la conversión análoga a digital una AD7689 con 8 CH esta se comunica con la Beaglebone Black por SPI. [8]

2.2 SISTEMA MUSCULAR: MUSCULATURA DE LAS EXTREMIDADES

Estos músculos son los responsables de mover las extremidades superiores e inferiores. Los músculos de las extremidades suman más o menos el 40% de todos los músculos estriados del organismo. En las extremidades hay dos grupos principales de músculos: 1) los músculos de la extremidad superior, y 2) los músculos de la extremidad inferior. Nos enfocaremos en los músculos de las extremidades superiores descartando los músculos de las extremidades inferiores debido a que el trabajo se enfoca en las señales EMG del brazo humano. Los músculos asociados a la extremidad superior pueden dividirse en tres grupos: 1) músculos que mueven el brazo; 2) músculos que mueven el antebrazo y la mano, y 3) músculos que mueven la mano y los dedos. [9]

2.2.1 MÚSCULOS QUE MUEVEN EL BRAZO

El **músculo deltoides** es el principal abductor del brazo, pero el **músculo supraespinoso** colabora al inicio de este movimiento. Los **músculos subescapular** y **redondo mayor** rotan el brazo en sentido medial, mientras que los **músculos infraespinoso** y **redondo menor** lo hacen en sentido lateral. El **músculo pectoral mayor** flexiona la articulación del hombro, y el **músculo dorsal ancho** la extiende. [9]

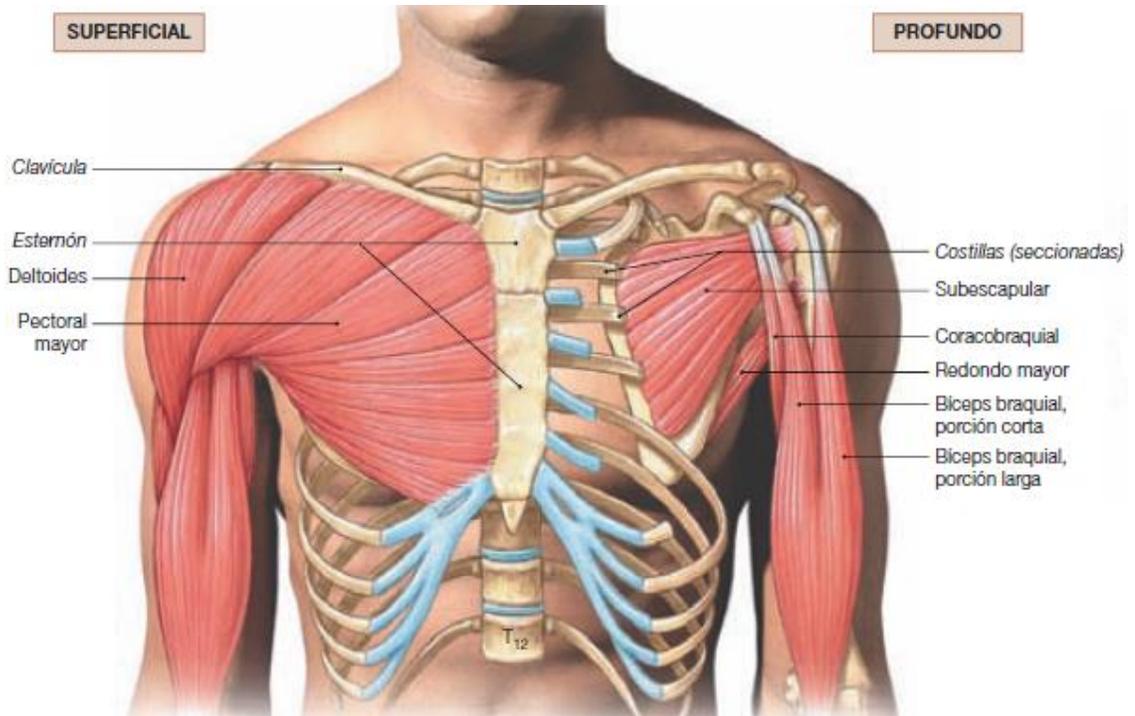


Figura 1. Músculos que mueven el brazo vista anterior[9]

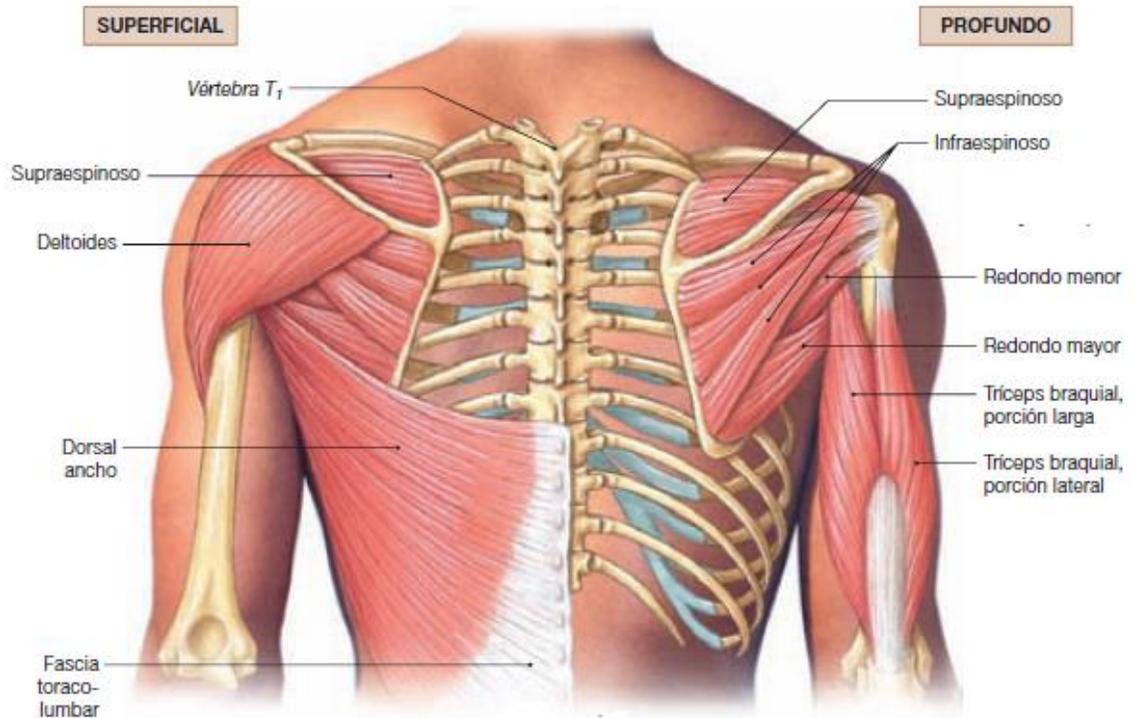


Figura 2. Músculos que mueven el brazo vista posterior[9]

2.2.2 MÚSCULOS QUE MUEVEN EL ANTEBRAZO Y LA MANO

Los músculos que mueven el antebrazo y la mano se detallan en la tabla 1.

Músculo	Acción
Bíceps braquial	Flexión del codo y del hombro; supinación
Braquial	Flexión del codo
Braquiorradial	Igual que arriba
Ancóneo	Extensión del codo
Tríceps braquial	Extensión del codo, extensión y aducción del hombro
Pronador cuadrado	Pronación del antebrazo y de la mano por rotación medial del radio en las articulaciones radiocubitales
Pronador redondo	Igual que arriba, más flexión del codo
Supinador	Supinación del antebrazo y de la mano por rotación lateral del radio en las articulaciones radiocubitales
Flexor radial del carpo	Flexión y abducción de la muñeca
Flexor cubital del carpo	Flexión y aducción de la muñeca
Palmar largo	Flexión de la muñeca, efectúa la flexión palmar de la mano. [10]

Tabla 1. Músculos que mueven el antebrazo y la mano [9]

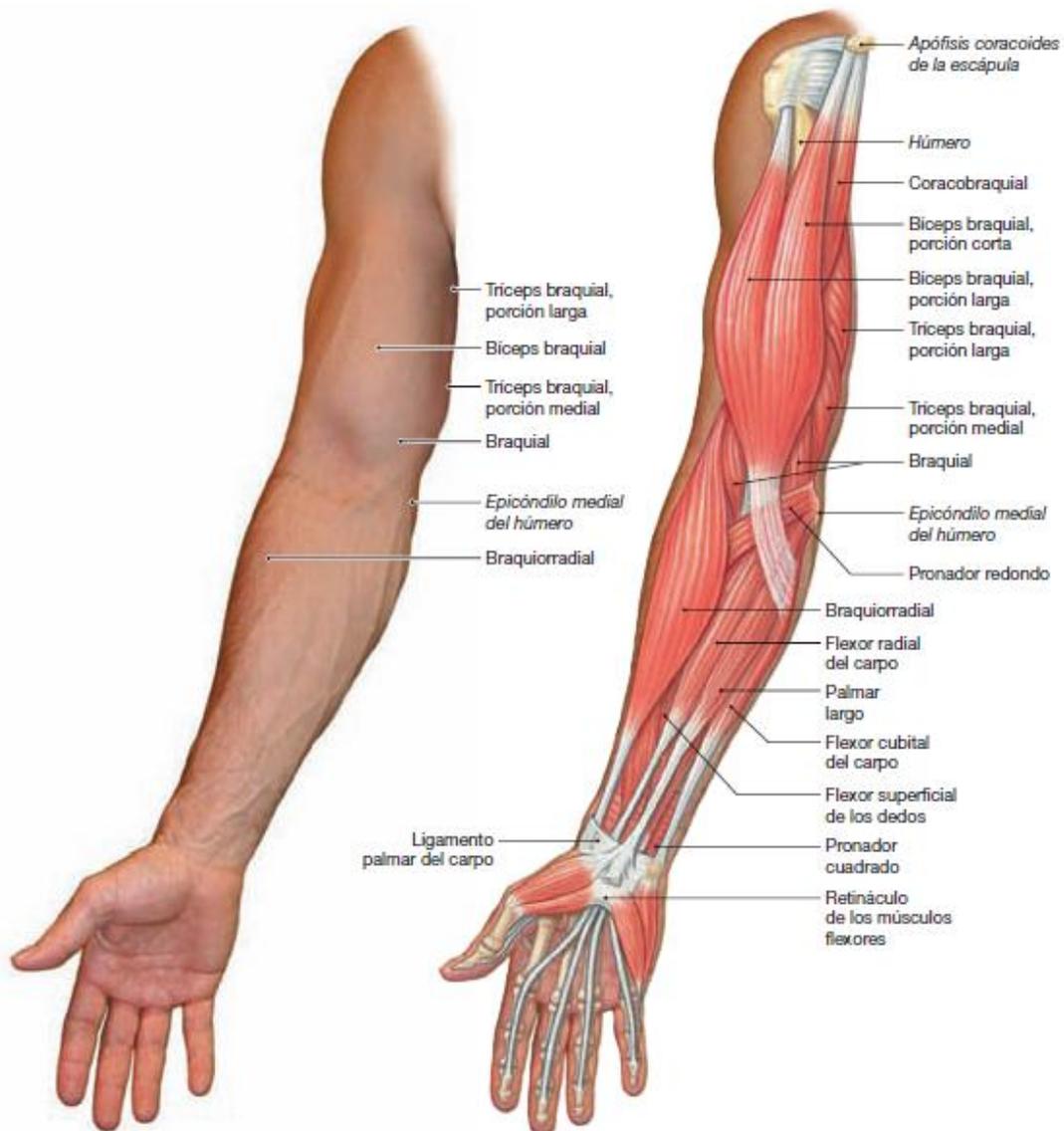


Figura 3. Músculos que mueven el antebrazo y la mano [9]

2.3 ELECTROMIOGRAFÍA

La electromiografía se encarga del estudio de la actividad eléctrica que proviene de los músculos en reposo y de aquellos que se contraen en forma activa haciendo uso de electrodos. El estudio de este tipo de señales puede diagnosticar patologías que afectan al Sistema Nervioso Periférico, alteraciones funcionales de las raíces nerviosas, estudio de los plexos y los troncos nerviosos periféricos, así como de patologías del músculo y de la unión neuromuscular. La mayor cantidad de actividad electromiográfica está presente alrededor de los

50Hz. En una contracción del musculo, los rangos de potenciales van desde los $50\mu\text{V}$ hasta los 5mV . Existen dos formas principales de registrar las señales electromiográficas; la electromiografía de superficie o no invasiva y la electromiografía invasiva. [7]

2.3.1 ELECTROMIOGRAFÍA DE SUPERFICIE

La electromiografía superficial SEMG, es una técnica no invasiva que se basa en el uso de electrodos superficiales; estos electrodos son colocados directamente sobre la piel del musculo del cual se quiere obtener información. [7]

2.3.2 ELECTRODOS SUPERFICIALES

Existen varios tipos de electrodos de superficie, estos se dividen principalmente en dos grupos: electrodos secos y electrodos húmedos. Los electrodos húmedos son aquellos en los que entre la placa de metal y la piel se encuentra una substancia electrolítica o gel conductor, esto se hace con el fin de minimizar el ruido intrínseco que se genera entre el contacto de la piel y el metal, este gel conductor mejora la conductividad y el flujo de la corriente. [7]



Figura 4. Electrodo superficial de gel adhesivo marca 3M. Ag/AgCl.

2.3.3 ASPECTOS IMPORTANTES

Antes de adquirir las señales EMG se deben tener en cuenta los siguientes aspectos:

- La remoción del vello de la superficie de la piel es esencial para mejora la adherencia y posterior limpieza de la piel, además de su retiro de los electrodos. [7]
- La preparación de la piel, existen cremas de limpieza con propiedades conductivas y abrasivas que sirven para remover las células muertas de la piel y la limpian de polvo y sudor. Otra alternativa es el uso de solo alcohol frotándolo con un trozo de tela. [7]
- La buena ubicación de los electrodos es indispensable para obtener una buena señal electromiográfica. La normatividad SENIAM define la distancia entre electrodos como “la distancia entre centros de las áreas de conductividad de los electrodos.” Con respecto a la distancia entre electrodos, la normatividad SENIAM recomienda que: “Los electrodos bipolares EMG de superficie tengan una distancia entre electrodos de entre 20mm y 30mm”. [7]

2.4 FUNDAMENTOS ELECTRÓNICOS

2.4.1 AMPLIFICADOR OPERACIONAL

Es un elemento de circuito diseñado para emplearse con otros elementos de circuito y efectuar una operación específica de procesamiento de señales. [11] A continuación tenemos el esquema eléctrico:

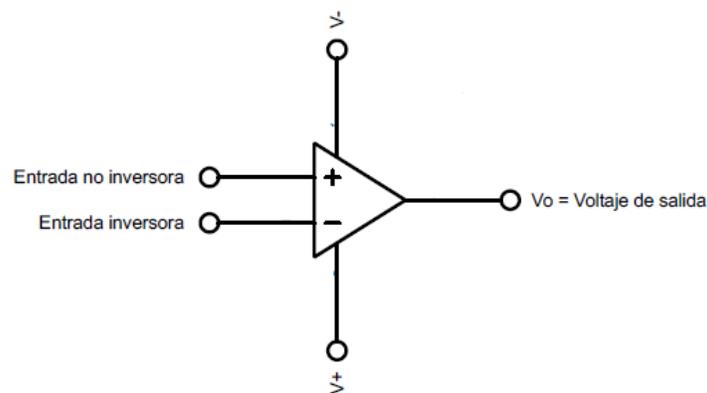


Figura 5. Esquema eléctrico amplificador operacional.

2.4.2 SEGUIDOR DE VOLTAJE

Es un amplificador de ganancia unitaria, que presenta las siguientes características una resistencia de entrada infinita ($R_i = \infty$) y una resistencia de salida nula ($R_o = 0$). [12]

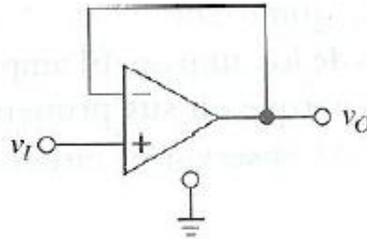


Figura 6. Seguidor de voltaje. [12]

El voltaje de salida es igual al voltaje de entrada tanto en magnitud como en signo.

$$V_i = V_o \quad (1)$$

2.4.3 AMPLIFICADOR INVERSOR

Es un amplificador con ganancia negativa; si la entrada es una onda senoidal, el circuito introducirá una inversión de fase, es decir, un cambio de fase de 180° . [12]

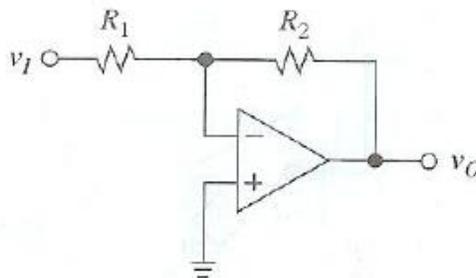


Figura 7. Amplificador inversor. [12]

La ganancia está dada por:

$$A = \frac{V_o}{V_i} = -\frac{R_2}{R_1} \quad (2)$$

2.4.4 AMPLIFICADOR NO INVERSOR

Consiste en un am op y dos resistores externos, la ganancia es positiva por lo cual la polaridad del voltaje de salida es la misma que la del voltaje de entrada. [12]

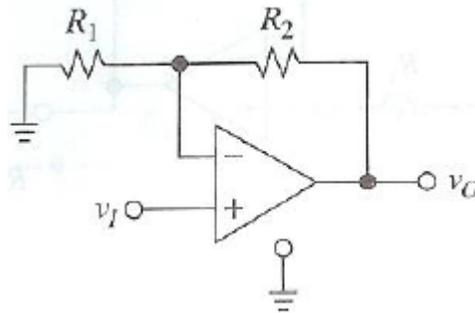


Figura 8. Amplificador no inversor. [12]

La ganancia viene dada por la siguiente ecuación:

$$A = \frac{V_o}{V_I} = 1 + \frac{R_2}{R_1} \quad (3)$$

2.4.5 AMPLIFICADOR DE INSTRUMENTACIÓN

Es un amplificador de diferencias que satisface las siguientes especificaciones: a) impedancias de entrada de modos diferencial y común; b) impedancia de salida muy baja (idealmente, cero); c) ganancia exacta y estable, por lo común en el rango de 1 V/V a 10^3 V/V; y d) razón de rechazo, por lo general, extremadamente elevada. Los amplificadores de instrumentación tienen muchas aplicaciones en el área de la medicina para el procesamiento de señales bioeléctricas. [12]

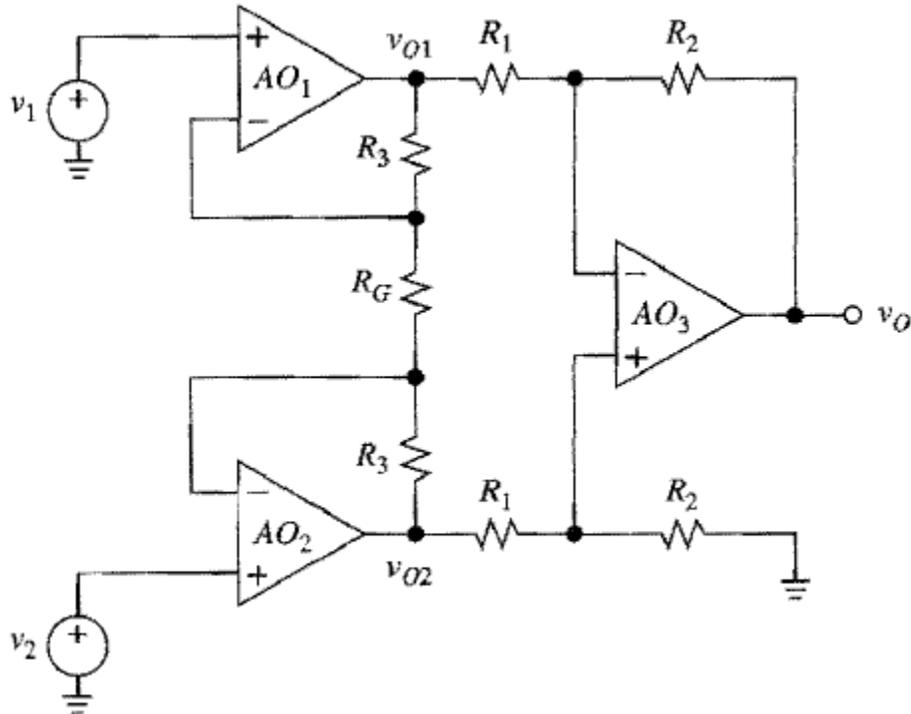


Figura 9. Amplificador de instrumentación. [12]

AO₁ y AO₂ forman la etapa de entrada y AO₃ forma la etapa de salida. Según la restricción del voltaje de entrada, el voltaje a través de R_G es $v_1 - v_2$. De acuerdo con la restricción de la corriente de entrada, las resistencias que se denotan como R_3 , llevan la misma corriente que R_G . Al aplicar la ley de Ohm se llega a que: [12]

$$v_{O1} - v_{O2} = \left(1 + \frac{2R_3}{R_G}\right)(v_1 - v_2) \quad (4)$$

La etapa de entrada también se conoce como amplificador de diferencias de entrada-diferencias de salida. El AO₃ es un amplificador de diferencia y por lo tanto

$$v_o = \frac{R_2}{R_1}(v_{O2} - v_{O1}) \quad (5)$$

Al combinar las dos últimas ecuaciones:

$$v_o = A(v_2 - v_1) \quad (6a)$$

$$A = A_I \times A_{II} = \left(1 + 2 \frac{R_3}{R_G}\right) \quad (6b)$$

Como AO_1 y AO_2 se operan en la configuración no inversora, sus resistencias de entrada de lazo cerrado son altas en extremo. Del mismo modo, la resistencia de salida de lazo cerrado de AO_3 es muy baja. [12]

2.4.6 FILTROS ACTIVOS

Un filtro es un circuito que procesa señales sobre una base que depende de la frecuencia. La manera en que su comportamiento varía con la frecuencia se llama respuesta a la frecuencia. [12] La respuesta en frecuencia de los filtros está caracterizada por tres parámetros del filtro: una ganancia K , la frecuencia de corte ω_0 , y el factor de calidad Q . [11] Los filtros activos involucran amplificadores operacionales y tenemos las siguientes clasificaciones que se muestran a continuación:

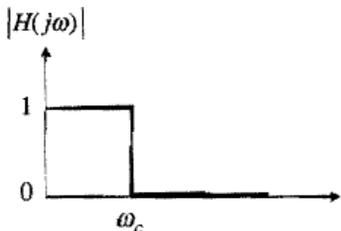
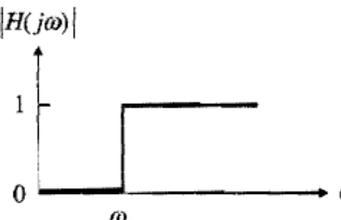
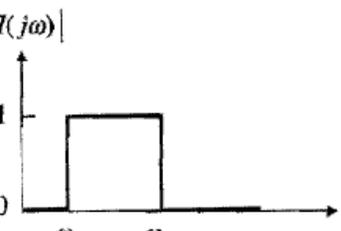
<i>Tipo</i>	<i>Respuesta en frecuencia</i>
<i>Pasabajas</i>	
<i>Pasaaltas</i>	
<i>Pasabanda</i>	

Tabla 2. Filtros ideales [12]

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

- **Pasabajas:** Atenúa las frecuencias que están por encima de la frecuencia de corte. Las ecuaciones de diseño son las siguientes: [11]

$$\omega_0 = \frac{1}{RC} \quad (7)$$

$$Q = \frac{1}{3 - A} \quad (8)$$

$$K = A \quad (9)$$

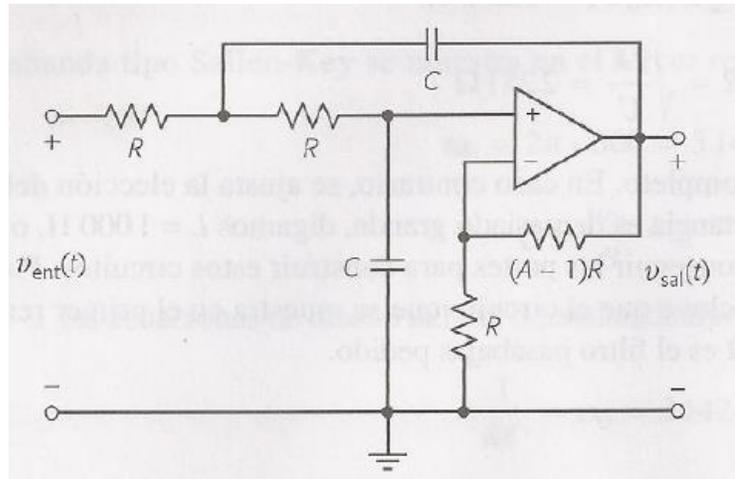


Figura 10. Filtro pasabajas [11]

- **Pasaaltas:** Atenúa las frecuencias que están por debajo de la frecuencia de corte. Las ecuaciones de diseño son las mismas del filtro pasabajas: [11]

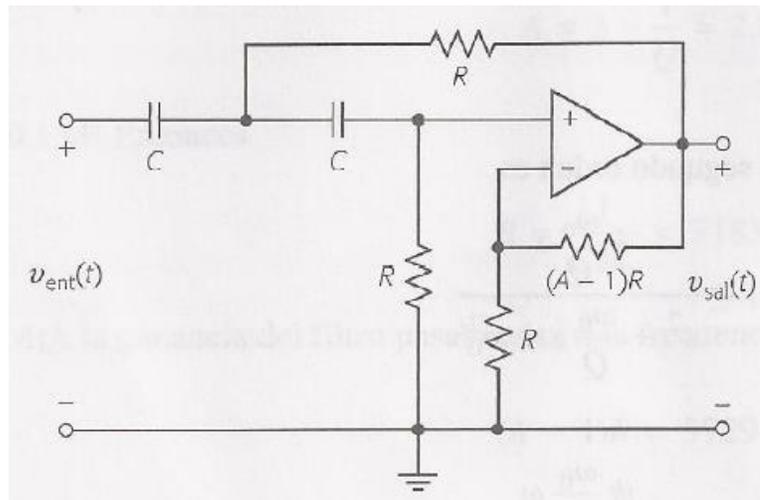


Figura 11. Filtro pasaaltas [11]

- **Pasabanda:** Atenúa las frecuencias que están por fuera de la banda de paso, las ecuaciones de diseños son casi iguales a la de los filtros pasabajas y pasaaltas solo difieren en la ecuación de la ganancia. Este tipo de filtro también se puede hacer con un filtro pasabajas unido a un filtro pasaaltas. [11]

$$K = QA \quad (10)$$

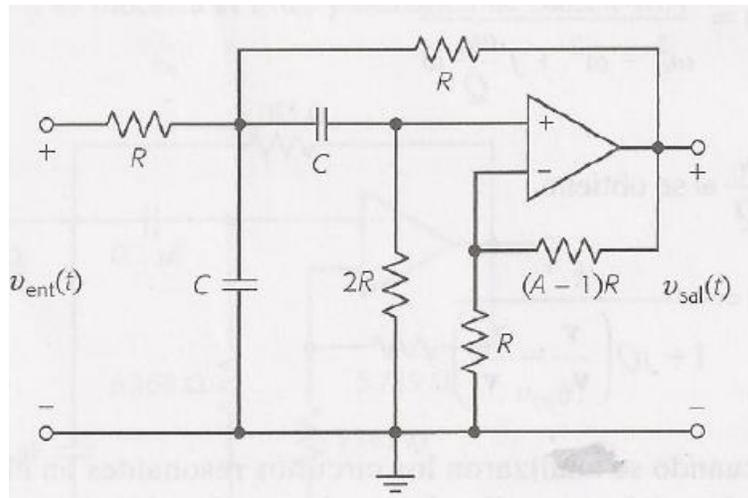


Figura 12. Filtro pasabanda [11]

2.5 REDES NEURONALES

Las redes neurales artificiales son aproximadores no lineales a la forma en que funciona el cerebro; por lo tanto no deben compararse directamente con el cerebro ni confundir los principios que fundamentan el funcionamiento de las redes neurales artificiales y el cerebro, ni pensar que las redes neurales se basan únicamente en las redes biológicas ya que sólo emulan en una parte muy simple el funcionamiento del cerebro humano. [13]

Las ANN al margen de "parecerse" al cerebro presentan una serie de características propias del cerebro. Por ejemplo las ANN aprenden de la experiencia, generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos. [14]

- **Aprender:** adquirir el conocimiento de una cosa por medio del estudio, ejercicio o experiencia. Las ANN pueden cambiar su comportamiento en función del entorno. Se les muestra un

conjunto de entradas y ellas mismas se ajustan para producir unas salidas consistentes. [14]

- **Generalizar:** extender o ampliar una cosa. Las ANN generalizan automáticamente debido a su propia estructura y naturaleza. Estas redes pueden ofrecer, dentro de un margen, respuestas correctas a entradas que presentan pequeñas variaciones debido a los efectos de ruido o distorsión. [14]
- **Abstraer:** aislar mentalmente o considerar por separado las cualidades de un objeto. Algunas ANN son capaces de abstraer la esencia de un conjunto de entradas que aparentemente no presentan aspectos comunes o relativos. [14]

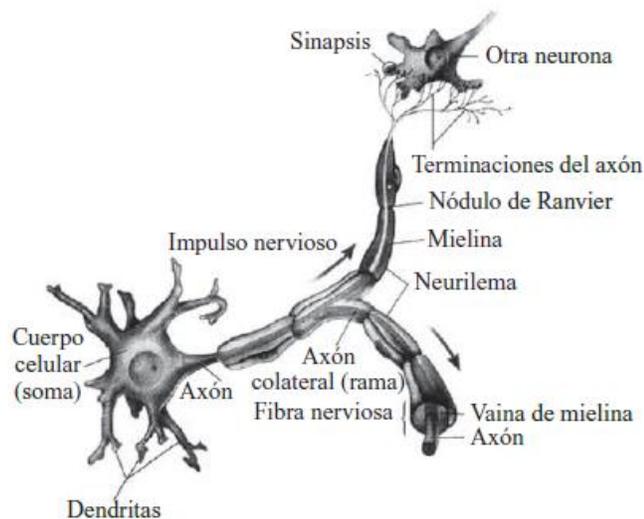


Figura 13. *Neurona biológica*. [13]

La neurona es la unidad fundamental del sistema nervioso y en particular del cerebro. Cada neurona es una simple unidad procesadora que recibe y combina señales desde y hacia otras neuronas. Si la combinación de entradas es suficientemente fuerte la salida de la neurona se activa. La Figura 13 muestra las partes que constituyen una neurona. El cerebro consiste en uno o varios billones de neuronas densamente interconectadas. El axón (salida) de la neurona se ramifica y está conectada a las dendritas (entradas) de otras neuronas a través de uniones llamadas sinapsis. La eficacia de

la sinapsis es modificable durante el proceso de aprendizaje de la red.
[14]

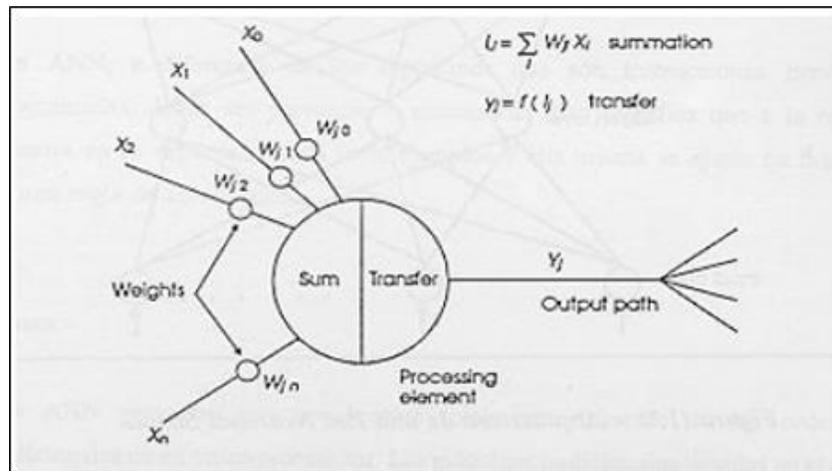


Figura 14. *Neurona Artificial.*[14]

En las Redes Neuronales Artificiales, ANN, la unidad análoga a la neurona biológica es el elemento procesador, PE (*process element*). Un elemento procesador tiene varias entradas y las combina, normalmente con una suma básica. La suma de las entradas es modificada por una función de transferencia y el valor de la salida de esta función de transferencia se pasa directamente a la salida del elemento procesador. La salida del PE se puede conectar a las entradas de otras neuronas artificiales (PE) mediante conexiones ponderadas correspondientes a la eficacia de la sinapsis de las conexiones neuronales. La Figura 14 representa un elemento procesador de una red neuronal artificial implementada en un computador. [14]

Una red neuronal consiste en un conjunto de unidades elementales PE conectadas de una forma concreta. El interés de las ANN no reside solamente en el modelo del elemento PE sino en las formas en que se conectan estos elementos procesadores. [14]

Generalmente los elementos PE están organizados en grupos llamados niveles o capas. Una red típica consiste en una secuencia de capas con conexiones entre capas adyacentes consecutivas. [14]

2.5.1 TOPOLOGÍAS DE LAS REDES NEURONALES

Dos de las topologías más usadas, de acuerdo con las diferencias en la manera de realizar las conexiones, son:

- **Redes de propagación hacia delante (feed-foward):** el flujo de información de las entradas a las salidas es exclusivamente hacia delante, extendiéndose por capas múltiples de unidades, pero no hay ninguna conexión de retroalimentación. [13]
- **Redes recurrentes:** contienen conexiones de retroalimentación, lo que puede derivarse en un proceso de evolución hacia un estado estable en el que no haya cambios en el estado de activación de las neuronas. [13]

2.5.2 ELEMENTOS DE UNA RED NEURONAL MULTICAPA

En el proyecto se implementó una red MLP, o una red multicapa, pero para el desarrollo del algoritmo es necesario comprender como es la estructura de este tipo de ANN, esta es una de las partes más importantes de la investigación. [14]

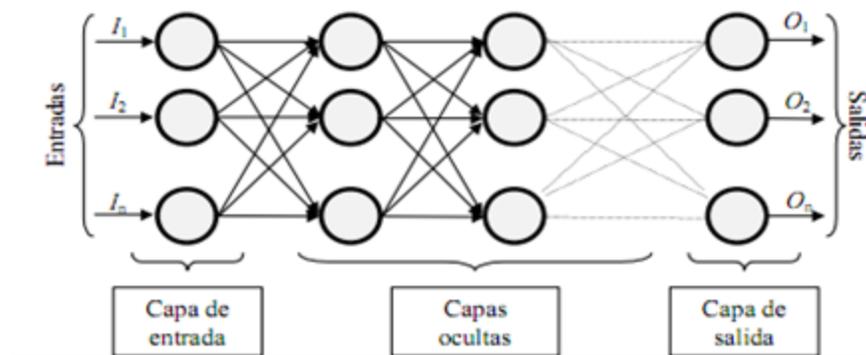


Figura 15. Estructura de una Red MLP.[14]

En la figura 8 está la representación pictográfica de una neuronal multicapa, la misma está constituida por neuronas interconectadas y arregladas en tres capas (esto último puede variar). Los datos ingresan por medio de la “capa de entrada”, pasan a través de la “capa oculta” y salen por la “capa de salida”. Cabe mencionar que la capa oculta puede estar constituida por varias capas. [14]

La neurona trata a muchos valores de entrada como si fueran uno solo; esto recibe el nombre de *entrada global*. Por lo tanto, ahora nos enfrentamos al problema de cómo se pueden combinar estas simples entradas (ini_1, ini_2, \dots) dentro de la entrada global, $gini$. Esto se logra a través de la *función de entrada*, la cual se calcula a partir del *vector entrada*. La función de entrada puede describirse como sigue: [14]

$$input_i = (in_{i1} * w_{i1}) * (in_{i2} * w_{i2}) * \dots (in_{in} * w_{in}) \quad (11)$$

En la ecuación 11 el símbolo * representa al operador apropiado (por ejemplo: máximo, sumatoria, producto, etc.), n al número de entradas a la neurona N_i y w_i al peso. Los valores de entrada se multiplican por los pesos anteriormente ingresados a la neurona. Por consiguiente, los pesos que generalmente no están restringidos cambian la medida de influencia que tienen los valores de entrada. Es decir, que permiten que un gran valor de entrada tenga solamente una pequeña influencia, si estos son lo suficientemente pequeños. [14]

Algunas de las funciones de entrada más comúnmente utilizadas y conocidas son:

- *Sumatoria de las entradas pesadas*: es la suma de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos. [14]

$$\sum_j (n_{ij} w_{ij}), \quad \text{con } j = 1, 2, \dots, n \quad (12)$$

- *Producto de las entradas pesadas*: es el producto de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos. [14]

$$\prod_j (n_{ij} w_{ij}), \quad \text{con } j = 1, 2, \dots, n \quad (13)$$

- *Máximo de las entradas pesadas*: solamente toma en consideración el valor de entrada más fuerte, previamente multiplicado por su peso correspondiente. [14]

$$\max_j(n_{ij} w_{ij}), \text{ con } j = 1, 2, \dots, n \quad (14)$$

Una neurona biológica puede estar activa (excitada) o inactiva (no excitada); es decir, que tiene un “estado de activación”. Las neuronas artificiales también tienen diferentes estados de activación; algunas de ellas solamente dos, al igual que las biológicas, pero otras pueden tomar cualquier valor dentro de un conjunto determinado. [14]

La *función activación* calcula el estado de actividad de una neurona; transformando la entrada global (menos el umbral, Θ_i) en un valor (estado) de activación, cuyo rango normalmente va de (0 a 1) o de (-1 a 1). Esto es así, porque una neurona puede estar totalmente inactiva (0 o -1) o activa (1). [14]

2.6 TARJETAS DE DESARROLLO

Actualmente podemos encontrar una gran variedad de tarjetas de desarrollo de diferentes fabricantes para diferentes tipos de aplicaciones, se pueden mencionar la Raspberry pi, Beaglebone Black, Tiva- c.

- **Raspberry pi 2**

Esta tarjeta posee un el Broadcom BCM2836, un sistema en un chip que contiene quad-core ARM Cortex-A7 con punto flotante, corre a 900MHz, y un procesador de Video 4 GPU. El GPU provee de un GL ES 2.0 abierto, acelerador de hardware OpenVG, y decodificador de alto perfil 1080p30 H.264 y capaz de 1Gpíxeles, 1.5Gtexteles/s o 24 GFLOPs de propósito general. Esto significa que la Raspberry Pi puede ser conectada a una TVHD, se pueden ver videos de calidad BlueRay, usando H.264 a 40Mbits/s. [15]

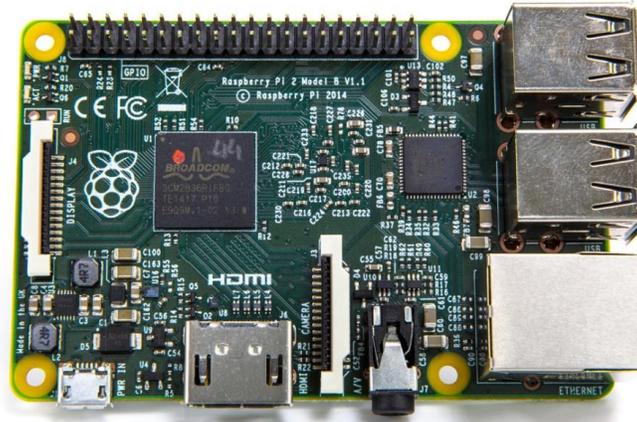


Figura 16. Tarjeta de desarrollo Raspberry pi 2. [15]

Posee 40 pines header GPIO con espaciamiento de 0.1" en la Pi te proveen de 26 GPIO, UART, I2C, SPI así como fuentes de 3.3V y 5V.

Dimensiones: 85mm x 56mm x 17mm

Características:

- 900 MHz BCM2836 ARMv7 Quad Core Processor SoC
- VideoCore IV GPU
- 1 GB RAM
- 4 x Puertos USB2.0 de hasta 1.2A de salida
- 40-pin GPIO Header
- Video/Audio Out a través de 4-pole 3.5mm conector, HDMI, o Raw LCD (DSI)
- Almacenamiento: microSD
- 10/100 Ethernet (RJ45)
- Periféricos de bajo nivel: 27 x GPIO
- UART
- I2C bus
- Bus SPI con 2 selectors de chip
- +3.3V
- +5V
- Tierra
- Requerimientos de energía: 5V @ 600 mA por MicroUSB o GPIO Header
- Soporte Windows 10, Debian GNU/Linux, Fedora, Arch Linux, RISC OS y Otros.

- **Beaglebone Black**

La Beaglebone Black cuenta con un procesador ARM Cortex A8 de Sitara XAM3359AZCZ100 con una frecuencia de trabajo a 1GHz de Texas Instruments, proporciona un puerto micro HDMI, 512 MB de DRAM DDR3L, 4 GB de memoria interna flash y JTAG opcional. Más de 3 millones de operaciones de operaciones aritméticas por segundo y el vector de punto flotante. [16]

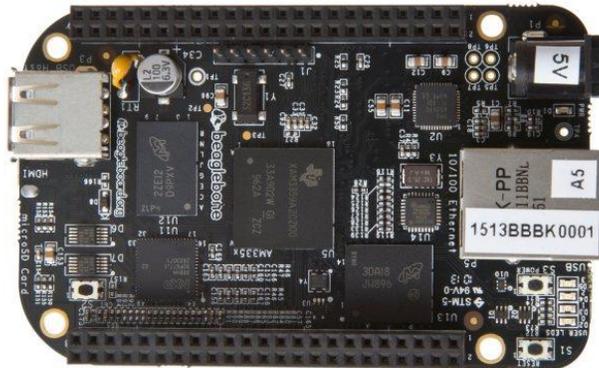


Figura 17. Tarjeta de desarrollo Beaglebone Black. [16]

Características:

- **Procesador:** Sitara AM335x 1GHz ARM® Cortex-A8
- 512MB DDR3L RAM 606 Mhz
- 4GB 8-bit eMMC on-board flash storage
- Acelerador de graficas 3D
- NEON acelerador punto flotante
- 2x PRU 32-bit microcontrolador

Conectividad:

- USB client, para alimentación y comunicaciones
- USB host
- Ethernet
- HDMI resolución 1280x1024
- 2x 46 pin headers

Software Compatible:

Debian, Android, Ubuntu, Cloud9 IDE on Node.js w/ BoneScript library, entre otros. [16]

- **Tiva C Series TM4C1294**

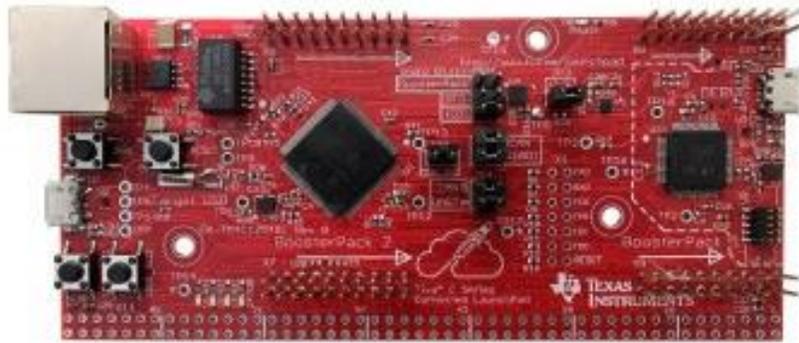


Figura 18. Tarjeta de desarrollo Tiva C TM4C1294. [17]

Características:

- **Microcontrolador:** TM4C1294NCPDTI ARM® Cortex-M4 MCU con punto flotante
- **Velocidad:** 120 MHz
- **Memoria Flash:** 1 MB
- **Memoria RAM:** 256 KB
- **Timers:** 8 de 32 bit
- **Comunicación serial :** 10 PC, 8 UARTS, 4 QSPI, 2 CAN, EPI, USB
- 8 PWMs
- 2 ADC de 12 bit
- 2 x 40 pines para boosterpacks
- Puerto de conexión Ethernet
- 4 Leds de usuario
- 2 Botones de usuario

2.7 LENGUAJE DE PROGRAMACIÓN EN PYTHON

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible; Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. [18]

- **Lenguaje interpretado o de script**

Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados). La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo los lenguajes interpretados son más flexibles y más portables. Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. [18]

- **Tipado dinámico**

La característica de tipado dinámico se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo del valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo. [18]

- **Fuertemente tipado**

No se permite tratar a una variable como si fuera de un tipo distinto al que tiene, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente. Por ejemplo, si tenemos una variable que contiene un texto (variable de tipo cadena o string) no podremos tratarla como un número (sumar la cadena "9" y el número 8). En otros lenguajes el tipo de la variable cambiaría para adaptarse al comportamiento esperado, aunque esto es más propenso a errores. [18]

- **Multiplataforma**

El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.) por lo que si no utilizamos librerías específicas de cada plataforma nuestro programa podrá correr en todos estos sistemas sin grandes cambios. [18]

- **Orientado a objetos**

La orientación a objetos es un paradigma de programación en el que los conceptos del mundo real relevantes para nuestro problema se trasladan a clases y objetos en nuestro programa. La ejecución del programa consiste en una serie de interacciones entre los objetos. Python también permite la programación imperativa, programación funcional y programación orientada a aspectos. [18]

2.8 BRAZO ROBÓTICO

Un brazo manipulador o brazo robótico se puede definir como el conjunto de elementos electromecánicos que propician el movimiento de un elemento terminal. La constitución física de la mayor parte de estos manipuladores guarda cierta similitud con la anatomía de las extremidades superiores del cuerpo humano. Una especificación general de un brazo robótico comprende: sus grados de libertad, su configuración y su cinemática directa e inversa. [19]

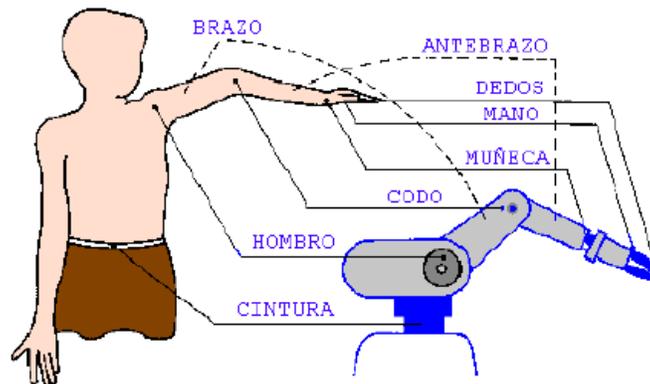


Figura 19. *Brazo robótico.* [19]

- **Cinemática**

Se interesa por la descripción analítica del desplazamiento espacial del robot como una función del tiempo, en particular de las relaciones entre la posición de las variables de articulación, y la posición y orientación del efector final del brazo robot. [20]

- **Dinámica**

Trata la formulación matemática de las ecuaciones del movimiento del brazo. Las ecuaciones dinámicas de movimiento de un manipulador son un conjunto de ecuaciones matemáticas que describen la conducta dinámica del mismo. [20]

- **Efectores**

Son las herramientas que le permiten al robot interactuar con su entorno, y realizar alguna acción particular. La interconexión del robot con su efector final es un aspecto importante en la

ingeniería del mismo, ya que proporciona un soporte físico al efector durante el ciclo de trabajo.

- **Sensores**

Son instrumentos capaces de percibir algún tipo de señal y transfórmala, para que pueda ser utilizado por algún sistema de control. La utilización de mecanismos sensores externos permite a un robot interactuar con su entorno de manera flexible. [20]

- **Articulaciones**

Permiten realizar los movimientos del robot. Están diseñadas para permitir que el efector final pueda llegar a una posición deseada dentro de los límites de tamaño del robot y de los movimientos de las articulaciones. [20]

- **Tipos de movimientos**

Cada uno de los movimientos independientes capaz de realizar una articulación, se denomina grado de libertad. Un grado de libertad puede indicar un movimiento longitudinal o de rotación.

Las articulaciones lineales implican un movimiento deslizante o de traslación de uniones de conexión. [20]

En las articulaciones giratorias se pueden distinguir 3 tipos: 1) rotacional, el eje de rotación es perpendicular a los ejes de las dos uniones, 2) implica un movimiento de torsión entre las uniones de entrada y salida. El eje de rotación de la articulación de torsión es paralelo a los ejes de ambas uniones y 3) es una articulación de revolución en la que la unión de entrada es paralela al eje de rotación y la salida es perpendicular a dicho eje. [20]

- **Limitaciones**

Son producidas por las características propias de los materiales y dispositivos con los cuales se construye, su anatomía, y el tiempo disponible para su construcción. Estos factores pueden provocar limitaciones en la estructura física del robot; en la construcción de un brazo robótico, definen el tamaño y alcance del brazo, la precisión de los movimientos, la fuerza y potencia que el mecanismo pueda aplicar, etc. [20]

3. DISEÑO DE COMPONENTES

3. DISEÑO DE COMPONENTES.....	33 -
3.1 SELECCIÓN DEL DISPOSITIVO.....	35 -
3.2 SISTEMA DE ADQUISICIÓN DE SEÑALES.....	36 -
3.2.1 SENSORES.....	37 -
3.2.2 AMPLIFICADOR DE INSTRUMENTACIÓN.....	38 -
3.2.3 ACONDICIONAMIENTO.....	40 -
3.2.3.1 FILTRO PASABANDA.....	40 -
3.2.3.2 SUMADOR DE VOLATJE.....	43 -
3.2.3.3 LIMITADOR DE AMPLITUD.....	44 -
3.2.4 TARJETA DE ADQUISICIÓN IMPLEMENTADA.....	44 -
3.3 BRAZO ROBÓTICO.....	47 -
3.4 SISTEMA CON HARDWARE IMPLEMENTADO.....	51 -

3.1 SELECCIÓN DEL DISPOSITIVO

Se tiene en cuenta los siguientes criterios para seleccionar el dispositivo hardware:

- Lenguaje de programación de alto nivel y vector de punto flotante para la implementación de la red neuronal.
- Que cuente con módulo ADC de mínimo 12 bits de 3 canales para la adquisición de las señales.
- Que cuente con sistema operativo como Android, Debian, Ubuntu, etc.
- Características de las tarjetas como tipo de procesador y velocidades de trabajo

Tiva C Series TM4C1294

Desventajas:

- No cuenta con sistema operativo.

Ventajas:

- Cuenta con vector de punto flotante, que es necesario para la implementación de la red neuronal.
- Beneficio/costo la tarjeta tiene un precio de 50,000 pesos el cual es relativamente bajo para sus prestaciones.
- Tiene un ARM Cortex M4 con una frecuencia de 120Mhz, y además posee 2 núcleos ADC de 12 bits con 20 canales.

Conclusión: Al no contar un sistema operativo no sirve para la implementación del proyecto a trabajar.

Beaglebone Black

Ventajas:

- Cuenta con un sistema operativo en tiempo Real sobre un procesador a 32 bit base Linux.

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

- Cuenta con una plataforma versátil para implementación de código de alto nivel como lenguaje de programación Python y tiene vector de punto flotante de doble precisión, que es necesario para la implementación de la red neuronal.
- Posee un ARM Cortex A8 y con una frecuencia de un 1 GHz, con una memoria RAM de 512MB, con 2 DMIPS/MHz (2000 DMIPS) y un ADC de 12 bits con 7 canales.
- Beneficio/costo no están bueno comparado con la Tiva C su precio está en unos 190'000 pesos.

Conclusión: Con esta tarjeta se puede implementar el sistema y tiene un procesador que superar el de la raspberry pi 2 con 100 DMIPS más.

Raspberry pi 2

Ventajas

- Cuenta con un sistema operativo en tiempo Real sobre un procesador a 32 bit base Linux.
- Permite la programación en alto nivel y tiene vector de punto flotante de doble precisión, que es necesario para la implementación de la red neuronal.
- Posee un ARM Cortex-A7 con una Frecuencia de 900 MHz con una memoria RAM de 1 GB, con 1,9DMIPS/MHz (1900 DMIPS)
- Beneficio/Costo es bueno en comparación con la Beaglebone Black con unos 180'000 pesos

Desventaja

- No posee un módulo ADC.

Conclusión: Al no tener un módulo ADC no sirve para la implementación de este proyecto.

3.2 SISTEMA DE ADQUISICIÓN DE SEÑALES

Una de las partes más importantes del proyecto es la adquisición de las señales EMG, ya que estas son las señales de control del sistema a implementar.

Para el diseño del sistema se tienen en cuenta las características del ADC de la Beaglebone Black que se describen en la siguiente tabla:

VDD_ADC	1.8V
Rango voltaje canal ADC	0-1.8V
Error de Ganancia	± 2 LSB
Error de Offset	± 2 LSB
SNR	70 dB
THD	75 dB
SNR más Distorsión	69 dB
Canal a canal de aislamiento	100 dB
Tiempo de conversión	15 ciclos de reloj ADC
Tiempo de adquisición	2 ciclos de reloj ADC
Frecuencia de muestro ADC clock = 3 MHz	200kSPS

Tabla 3. Características ADC (SAR) [21]

A nivel general se muestra el diagrama de bloques y seguidamente se describe cada uno de los bloques:

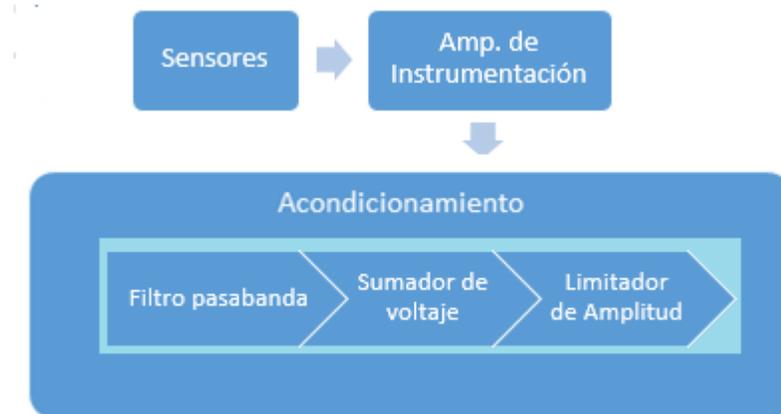


Figura 20. Diagrama en bloques sistema de adquisición.

3.2.1 SENSORES

Se utilizó una configuración diferencial de electrodos de cloruro de plata (Ag/AgCl 3M), estos se conectan directamente a la piel; la

conexión de los electrodos con el amplificador de instrumentación se realiza a través conectores para electrodos.



Figura 21. Conectores para electrodos y electrodos 3M.

3.2.2 AMPLIFICADOR DE INSTRUMENTACIÓN

En la siguiente tabla se muestran algunos amplificadores de instrumentación para aplicaciones electro médicas:

Referencia	Compañía	Especificaciones
INA118	Texas Instrument	Alimentación dual, 110dB CMRR, 50 μ V max Offset.
INA128	Texas Instrument	Alimentación dual, 120dB CMRR, 50 μ V max Offset.
AD620	Analog Devices	Alimentación dual, 90 dB CMRR con G=1 y CMRR=130dB con G=1000, 50 μ V max Offset
AD621	Analog Devices	Alimentación dual, 90 dB CMRR con G=1 y CMRR=130dB con G=1000, 50 μ V max Offset.

Tabla 4. Amplificadores de Instrumentación para aplicaciones médicas [7]

El circuito integrado AD620 es el amplificador de instrumentación escogido para la captura y amplificación de las señales EMG del sistema. Posee las siguientes características baja tensión de offset max 50 μ V, un CMRR min de 110 dB, una ganancia ajustable 1 a 10'000 y un rango de alimentación de \pm 2.3V a \pm 18V; estas

especificaciones permiten que los voltaje de modo común sean eliminados.

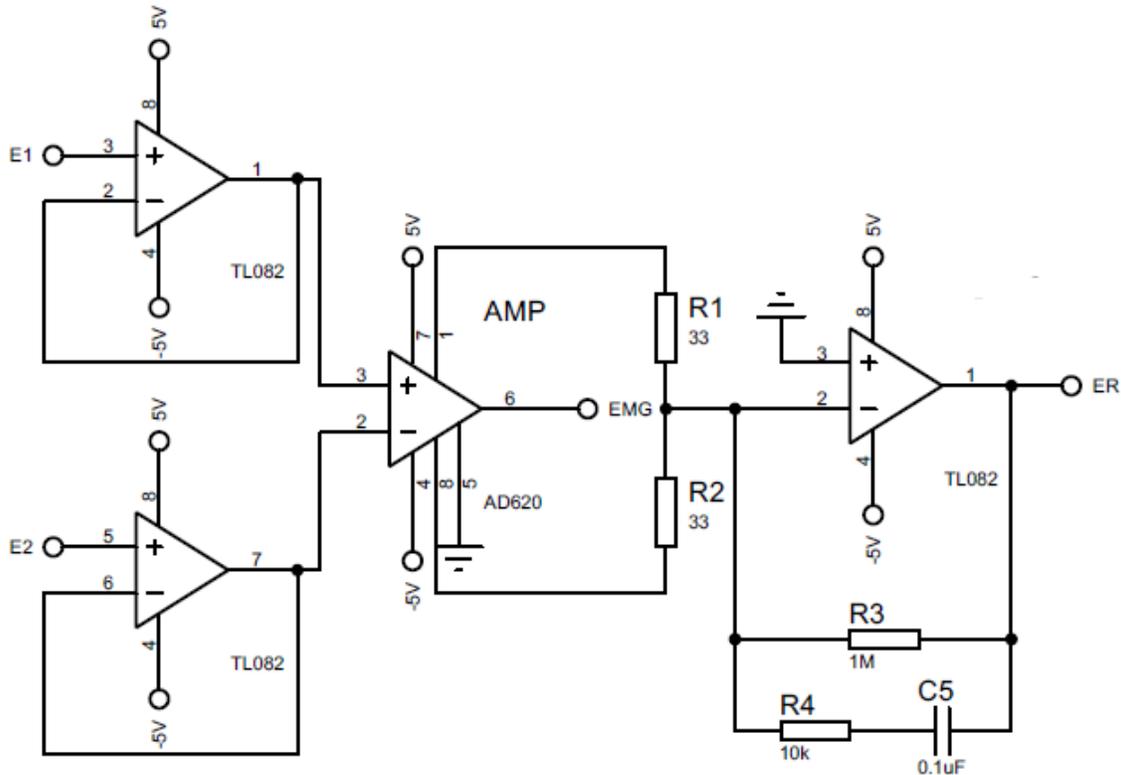


Figura 22. Circuito de adquisición etapa Amp. Instrumentación.

Los dos seguidores de tensión en los terminales 2 y 3 del AD620 están para garantizar una alta impedancia de entrada, para proporcionar un efecto de aislamiento de la salida con respecto a la señal de entrada proveniente de los electrodos, y además tiene la función de circuito de protección para proteger a la persona de posibles daños. La ganancia del amplificador de instrumentación (AD620) se calcula mediante la siguiente ecuación:

$$G = \frac{49.4K\Omega}{R_G} + 1 \quad (15)$$

$R_G = R1 + R2$, según el fabricante $R1 = R2 = \frac{R_G}{2}$, como $R1=R2=33\Omega$ entonces $R_G = 66\Omega$, por tanto nuestra ganancia es:

$$G = \frac{49.4K\Omega}{66\Omega} + 1 = 748,48$$

La señal de referencia del circuito de la figura 22. Se toma el valor medio de R_G , esta es una especificación dada por el fabricante para mantener la estabilidad.

3.2.3 ACONDICIONAMIENTO

Los electrodos ubicados en la superficie del brazo, captan información de varias unidades motoras, además de interferencias externas. Se hace necesario acondicionar la señal para atenuar todas las componentes de frecuencia que no hacen parte de la señal EMG. Para ello se implementó lo siguiente:

3.2.3.1 FILTRO PASABANDA

El filtro pasabanda está conformado por un filtro pasabaja de octavo orden en cascada con un filtro pasa altas de cuarto orden; como la mayor concentración de información de la señal EMG se encuentran entre los 20Hz y 500Hz, la frecuencia de corte del pasabaja será de 500Hz y la frecuencia de corte del pasaalta será de 20Hz. El filtro pasabaja se diseñó con el circuito integrado MAX295 que es un filtro Butterworth pasabaja de octavo orden, que tiene las siguientes características:

- Rango de frecuencia de corte 0.1Hz a 50KHz
- Alimentación dual $\pm 2.375V$ a $\pm 5.5V$
- Bajo nivel de ruido: -70dB THD + Ruido (Tipo)

Como el fabricante en la hoja de características no provee una ecuación para determinar la frecuencia de corte, se tomó una data para determinar la frecuencia de corte deseada, el circuito implementado para la toma de datos es el mostrado en la figura 23.

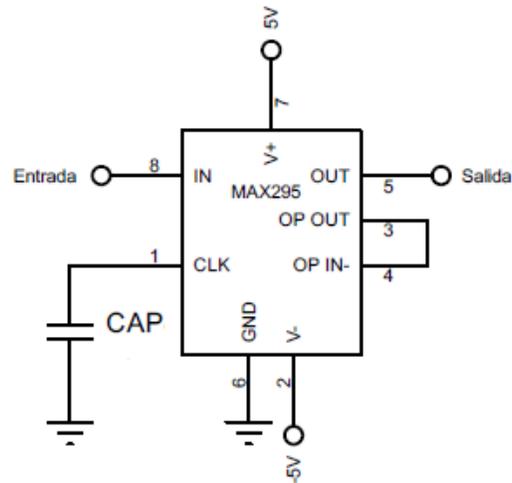


Figura 23. Circuito para tomar la data.

La metodología es la siguiente, en la terminal 8 entra una señal senoidal a una determinada frecuencia, el valor del capacitor determina la frecuencia de corte que tendrá el filtro; se tomaron valores de capacitores comerciales y para cada valor se registró la frecuencia de corte del filtro, los datos obtenidos se muestran en la siguiente tabla.

CAP(pF)	Fc(KHz)
10	24,24
22	16,34
30	14,1
47	10,4
56	8,86
100	5,68
120	4,77
330	2,3
470	1,54
1800	0,421
2200	0,317
5600	0,136
10000	0,0707
22000	0,0278
33000	0,02227
47000	0,015

Tabla 5. Datos capacitancia vs frecuencia de corte.

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

Utilizando Excel se aplicó una regresión con la cual se determinó una ecuación que nos permite calcular la frecuencia de corte; el tipo de regresión que se realizó fue una regresión potencial.

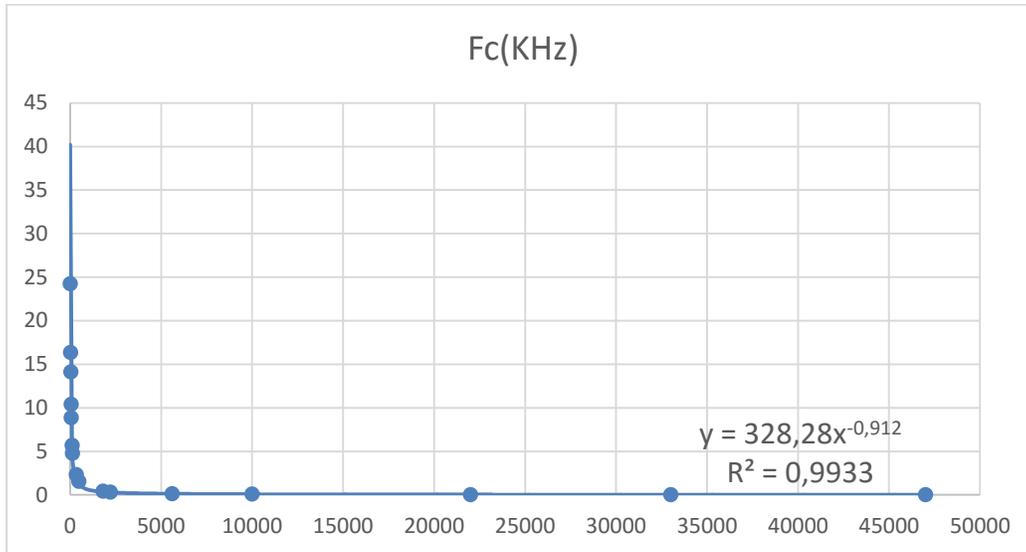


Figura 24. Regresión potencial.

La ecuación obtenida es la siguiente $y = 328,28x^{-0,912}$ donde y representa la frecuencia de corte y x la capacitancia, reescribiendo la ecuación tenemos:

$$f_c(KHz) = 328,28cap^{-0,912} \quad (16)$$

Como nuestra frecuencia de corte debe ser de 500Hz despejamos la capacitancia y tenemos:

$$cap(pF) = \left(\frac{f_c}{328.28}\right)^{-\frac{1}{0,912}} = \left(\frac{0.500}{328.28}\right)^{-\frac{1}{0,912}} = 1227.77pF$$

Para el diseño del filtro pasaalta se empleó la configuración Sallen Key y las ecuaciones descritas en el marco teórico. Como nuestra frecuencia de corte es 20Hz, suponemos un valor de condensador comercial y utilizando la ecuación 7 vamos a calcular los valores de resistencia que se necesitan.

$$\omega_0 = 2\pi f_c = \frac{1}{RC}$$

Despejamos la R para calcular su valor y tenemos:

$$R = \frac{1}{2\pi f_c C} = \frac{1}{2\pi(20)(0.22\mu F)} = 36,2K\Omega$$

Como el filtro es de 4 orden necesitamos conectar en cascadas dos etapas de 2 orden, la primera etapa tendrá una ganancia unitaria y la segunda etapa una ganancia de 1.2, para tener una ganancia total de 1.2 y el factor de calidad por medio de la ecuación 8 es de 0.555. El diagrama del filtro implementado es el siguiente.

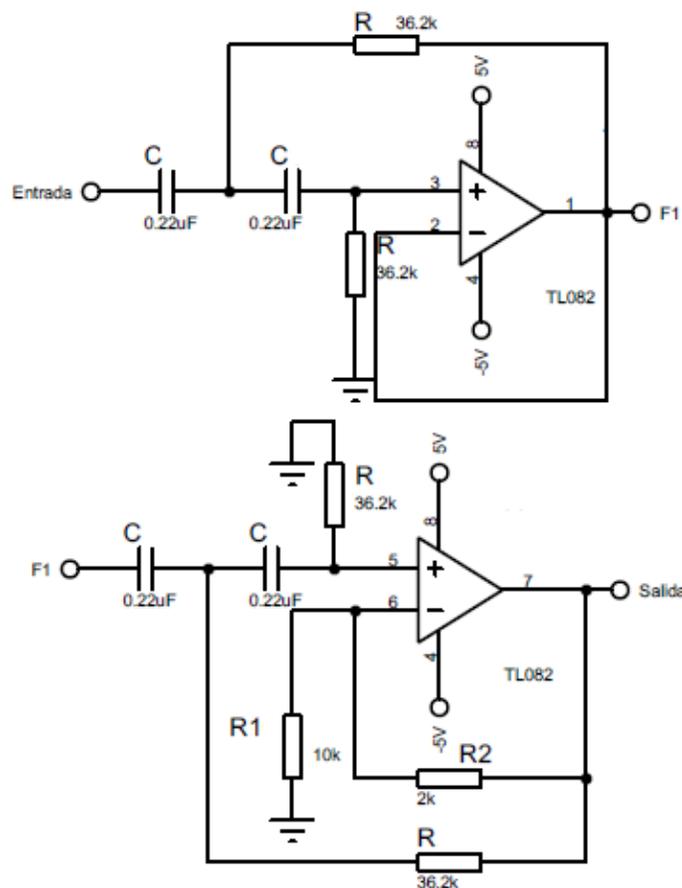


Figura 25. Filtro pasaalta de 4 orden.

3.2.3.2 SUMADOR DE VOLTAJE

Como el ADC de la Beaglebone Black no admite voltajes negativos necesitamos sumar una desviación de voltaje dc a la señal ac (EMG) para que esta solo tome valores positivos y no destruya el ADC.

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

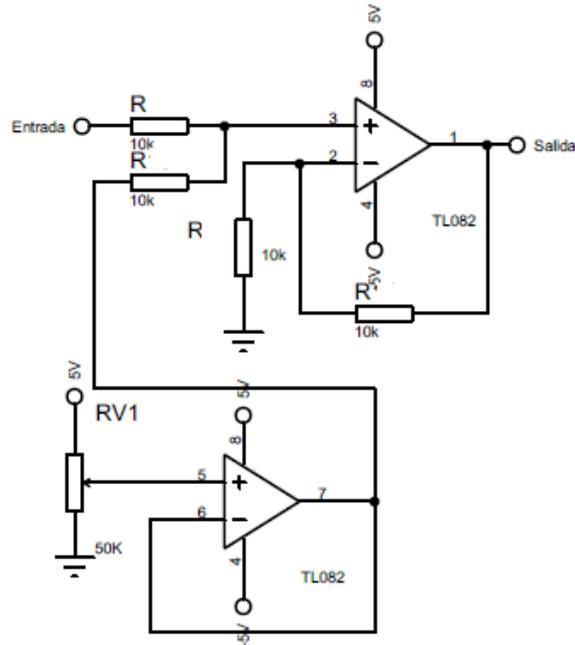


Figura 26. Sumador de Voltaje.

3.2.3.3 LIMITADOR DE AMPLITUD

El rango de voltajes que admite el ADC de la Beaglebone Black es de 0-1.8V por lo cual se hace necesario limitar el voltaje máximo de la señal EMG a 1.8V para no destruir el ADC de la tarjeta.

3.2.4 TARJETA DE ADQUISICIÓN IMPLEMENTADA.

El PCB de la tarjeta de adquisición se diseñó en el software Proteus en su aplicación ARES, la tarjeta tiene una sola capa y se respeta la distancia mínima entre los caminos de comunicación. Además como se estudian los músculos responsables del movimiento de la mano, antebrazo y brazo el sistema cuenta con 3 canales para la adquisición de las señales EMG.

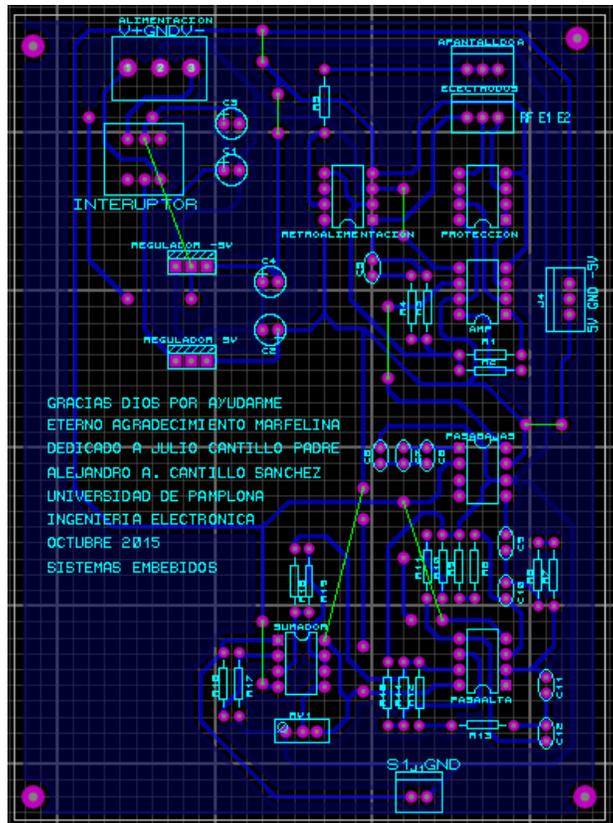


Figura 27. Tarjeta de adquisición canal 1 en ARES.

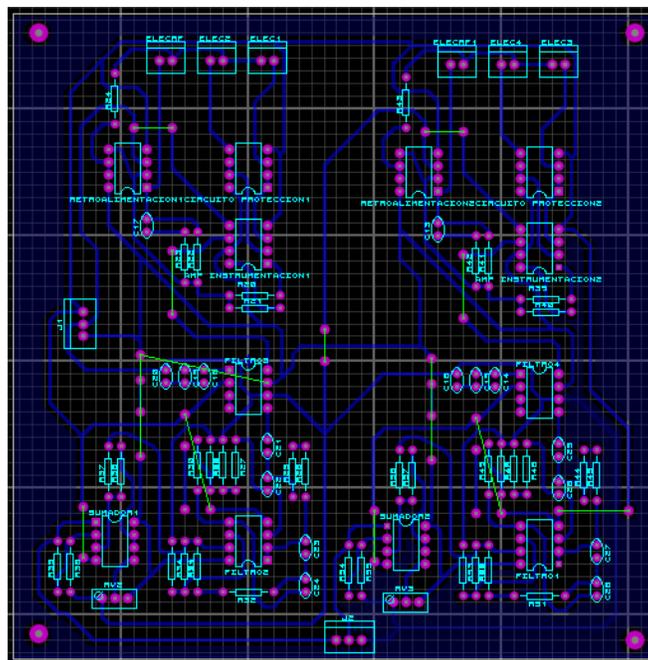


Figura 28. Tarjeta de adquisición canal 2 y 3 en ARES.

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

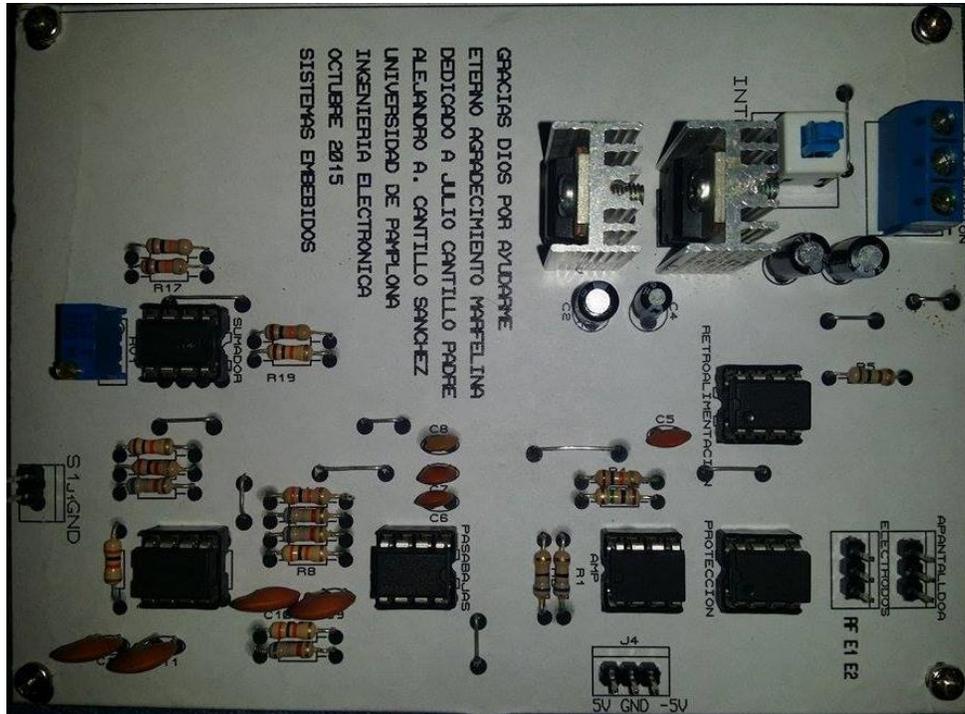


Figura 29. Tarjeta de adquisición canal 1.

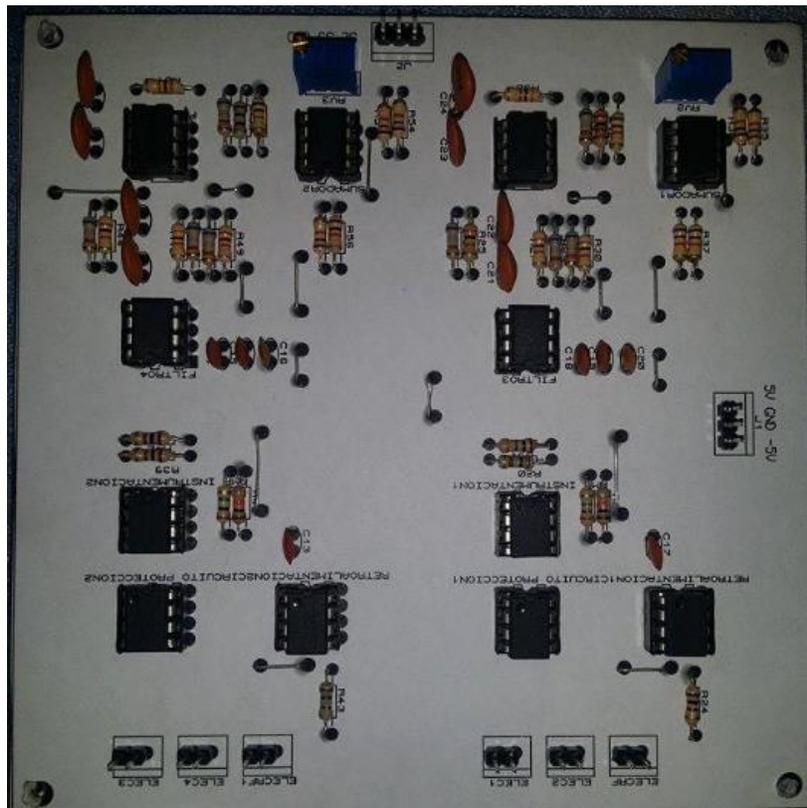


Figura 30. Tarjeta de adquisición canal 2 y 3.

3.3 BRAZO ROBÓTICO.

Para el diseño del brazo robótico tenemos en cuenta que los movimientos que va a realizar este, son movimientos lineales como lo es subir y bajar el brazo, la flexión del antebrazo y el abrir y cerrar la mano; vamos a tener 3 grados de libertad, uno para el movimiento del brazo, otro para la flexión del antebrazo y el último para el movimiento de la mano.

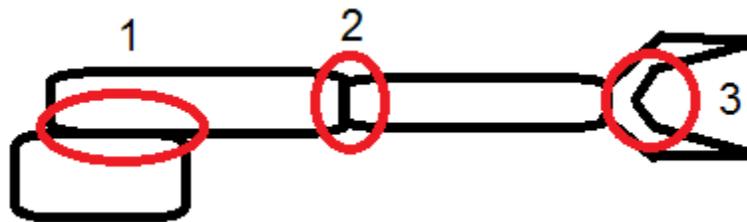


Figura 31. Diagrama básico de brazo robótico.

En la figura 31 se muestra el diagrama básico de nuestro brazo, en cada área marcada por la elipse roja van ubicados los motores (Servomotores) los cuales permiten realizar los movimientos del brazo. En el área 1 soportara todo el peso del brazo por lo cual es el área con mayor torque y los servomotores deben proveer el torque necesario para que el brazo robótico pueda realizar el movimiento sin complicaciones. El diseño de la mano será una pinza.

Para el diseño de las partes se utilizó el software Solidworks 2015, estas partes fueron cortadas en acrílico de 3mm de grosor, a continuación se muestran las partes en la siguiente imagen:

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

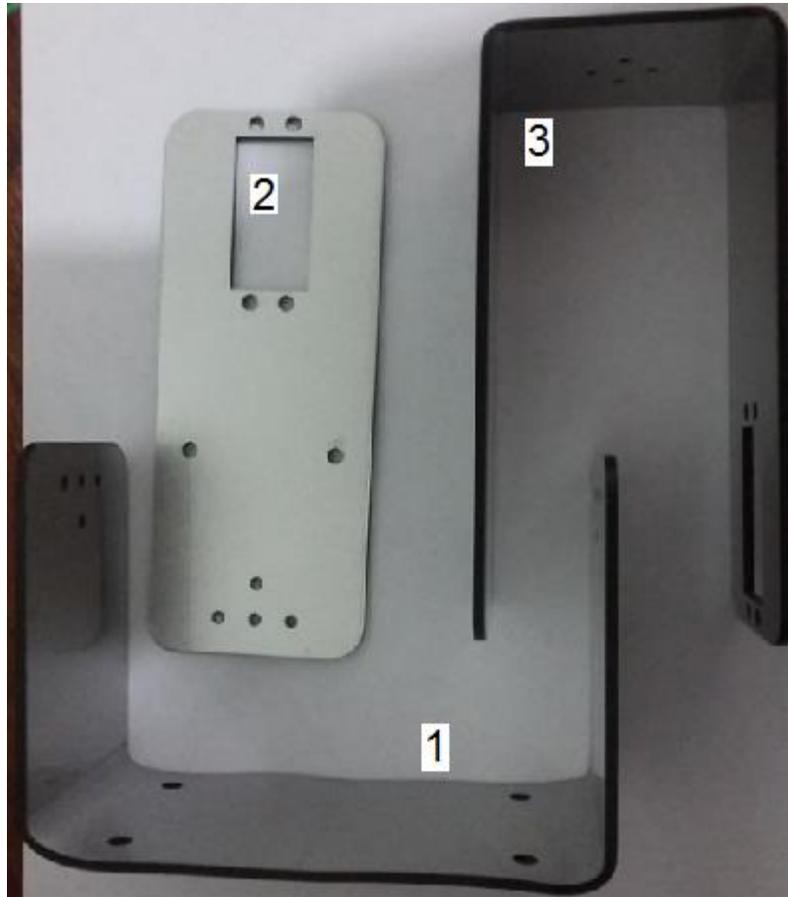


Figura 32. Partes cortadas en acrílico.

Para la parte la mano se compró una pinza de aleación de aluminio la cual se muestra a continuación:



Figura 33. Pinza de aleación de aluminio.

Para el cálculo del torque de los servomotores se debe tener en cuenta la densidad del acrílico y el volumen para hallar los kilogramos fuerza (Kgf) de cada pieza debido a que el torque de los servomotores viene dado en Kgf x cm, para eso se realizan los siguientes cálculos:

Peso fuerza = $w = \text{densidad del acrílico} \times \text{volumen del acrílico}$

$$v_2 = \text{volumen pieza 2} = 0,3 \times 6 \times 34 = 56,1 \text{ cm}^3$$

$$v_3 = \text{volumen pieza 3} = 0,3 \times 6 \times 15 = 27 \text{ cm}^3$$

La densidad del acrílico es de $1,18 \text{ g/cm}^3$, el peso fuerza de cada servomotor es de 55gf y la pieza 4 tiene un peso fuerza de 68gf, solo resta calcular el peso fuerza de las piezas 2 y 3, como la pieza 1 es una base no entra en el proceso para calcular el torque de los servomotores.

$$w_2 = (1,18 \text{ g/cm}^3) \times 56,1 \text{ cm}^3 = 66,198 \text{ gf}$$

$$w_3 = 2((1,18 \text{ g/cm}^3) \times 27 \text{ cm}^3) = 63,72 \text{ gf}$$

Procedemos a calcular el torque del servomotor 1 que es el que soportara todo el peso del brazo:

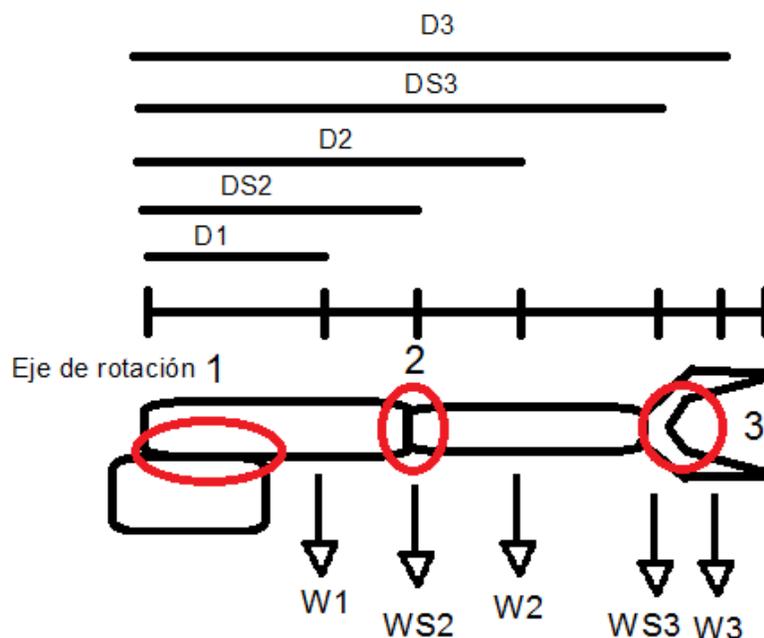


Figura 34. Diagrama para calcular el torque.

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

$$\text{Torque} = \text{distacia} \times \text{fuerza} = dx f$$

$$\text{Torque1} = (D1)(W1) + (DS2)(WS2) + (D2)(W2) + (DS3)(WS3) + (D3)(W3)$$

Donde:

W1=Peso fuerza pieza 2 W2=Peso fuerza pieza 3

W3=Peso fuerza pieza 4 WS2=Peso fuerza servo 2

WS3=Peso fuerza pieza 3

$$\begin{aligned} \text{Torque1} &= (8\text{cm})(66,198\text{gf}) + (13\text{cm})(55\text{gf}) + (22\text{cm})(63.72\text{gf}) \\ &+ (30\text{cm})(55\text{gf}) + (37\text{cm})(68\text{gf}) = 6812,424\text{gfcm} \\ &= 6,812 \text{ Kgfc} \end{aligned}$$

Los servomotores que se compraron son de la marca MG996R y tienen un torque de 9,4kgfcm por lo cual cumplen con los requisitos exigidos para esta aplicación.

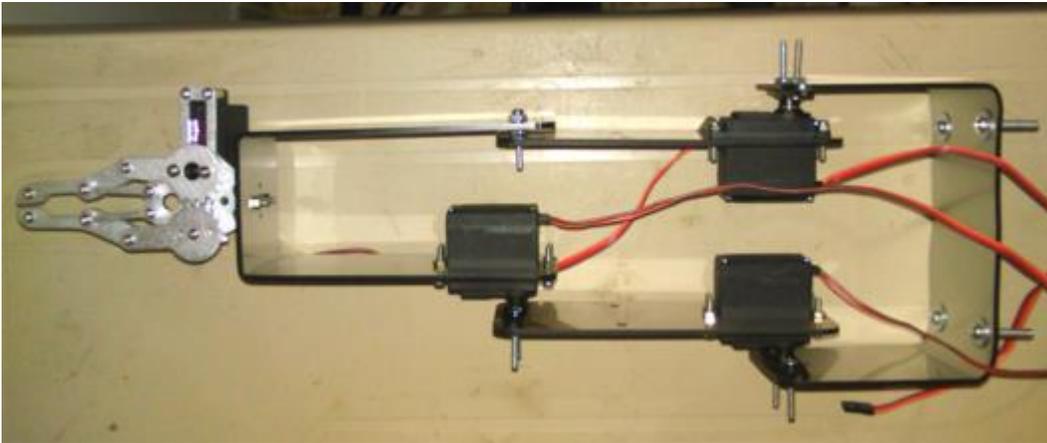


Figura 35. Brazo robótico implementado.

3.4 SISTEMA CON HARDWARE IMPLEMENTADO

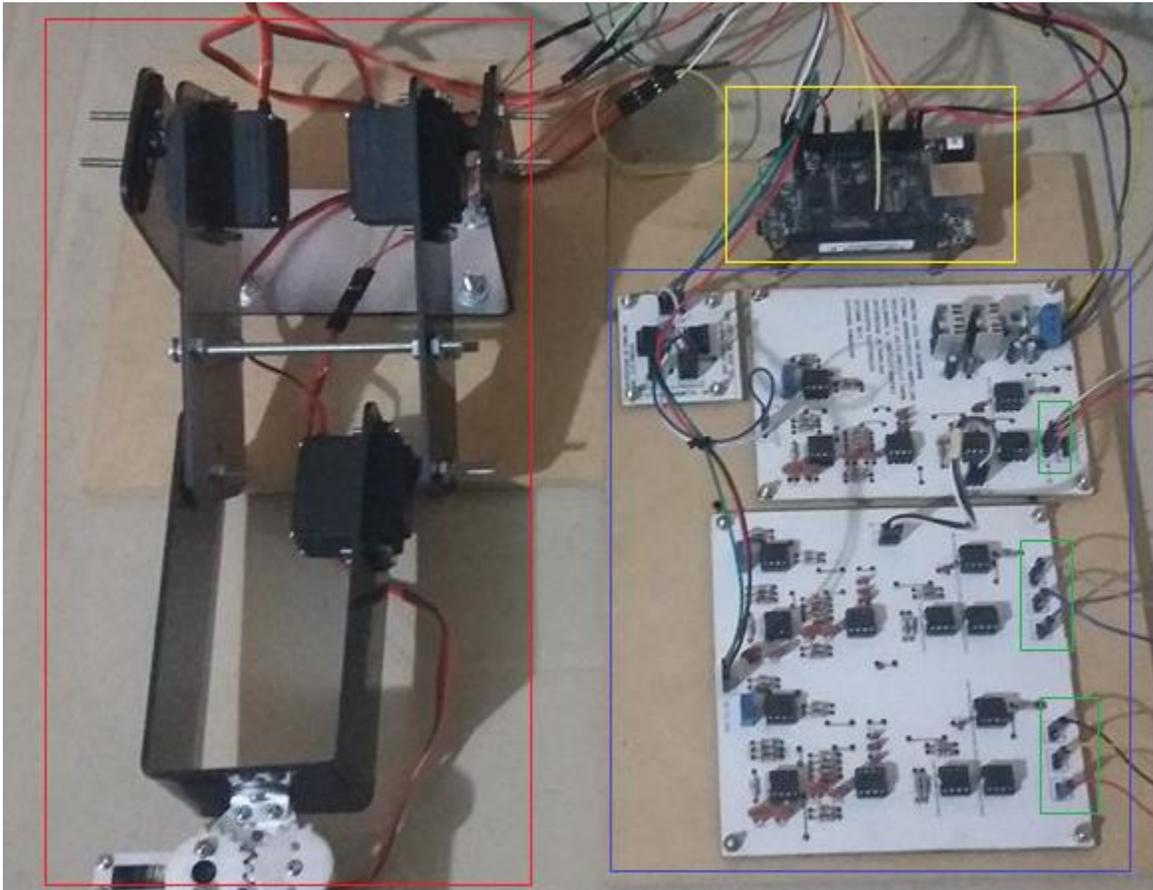


Figura 36. Imagen donde se muestran todo el sistema acoplado

En la imagen anterior podemos observar todo el sistema con sus componentes acoplados.

En el recuadro rojo se puede observar el brazo robótico el cual recibe la señal de control PWM proveniente de la Beaglebone Black ejecuta los movimientos respectivos.

En el recuadro azul se encuentra el sistema de adquisición de señales este es el encargado de captar las señales EMG del brazo humano y realiza el acondicionamiento de la señal para enviarla al ADC de Beaglebone Black.

El recuadro amarillo muestra la tarjeta de desarrollo Beaglebone Black, que es la encargada de hacer el procesamiento de la

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

información y la clasificación de las señales, para determinar el movimiento a ejecutar.

En los recuadros verdes encontramos los pines de entrada del sistema de adquisición donde van conectados los electrodos.

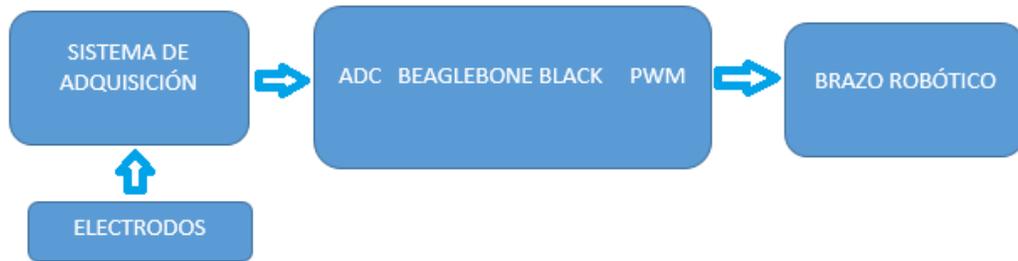


Figura 37. Diagrama de bloques del hardware del sistema

4. PROGRAMACIÓN Y DESARROLLO

4. PROGRAMACIÓN Y DESARROLLO.....	53 -
4.1 CARACTERISTICAS DEL SISTEMA.....	54 -
4.1.1 CONFIGURACIÓN DEL HARDWARE.....	55 -
4.1.2 CONFIGURACIÓN DEL SOFTWARE.....	56 -
4.2 CAPTURA DE DATOS Y EXTRACCIÓN DE PATRONES.....	56 -
4.3 RED MLP EN MATLAB.....	59 -
4.3.1 PESOS DE LAS NEURONAS Y UMBRALES.....	61 -
4.3.2 RESULTADOS DE ENTRANAMIENTO.....	63 -
4.4 ESTRUCTURA DEL PROGRAMA PRINCIPAL EN PYTHON.-	64 -

4.1 CARACTERÍSTICAS DEL SISTEMA

El sistema realiza tres procesos muy importantes, consta de adquisición de datos, inteligencia artificial y ejecución de movimiento, el primer proceso adquisición de datos realiza la captura de la señal EMG de 3000 muestras y se aplican técnicas de tratamiento de muy sencillas en el lenguaje Python para así extraer características o patrones.

En la parte de la inteligencia artificial se implementa una red neuronal la cual va clasificar en 4 clases los patrones obtenidos, la clase 1 representa la posición de referencia del brazo robótico por lo cual no hay movimiento en esta clase, la clase 2 representa el movimiento del brazo, la clase 3 representa el movimiento del antebrazo y la clase 4 representa el movimiento de la mano.

Ejecución del movimiento una vez la red neuronal identifica la clase, se procede a ejecutar el movimiento que representa dicha clase, para eso la Beaglebone Black envía la señal PWM a los servomotores del brazo robótico para ejecutar el movimiento identificado.

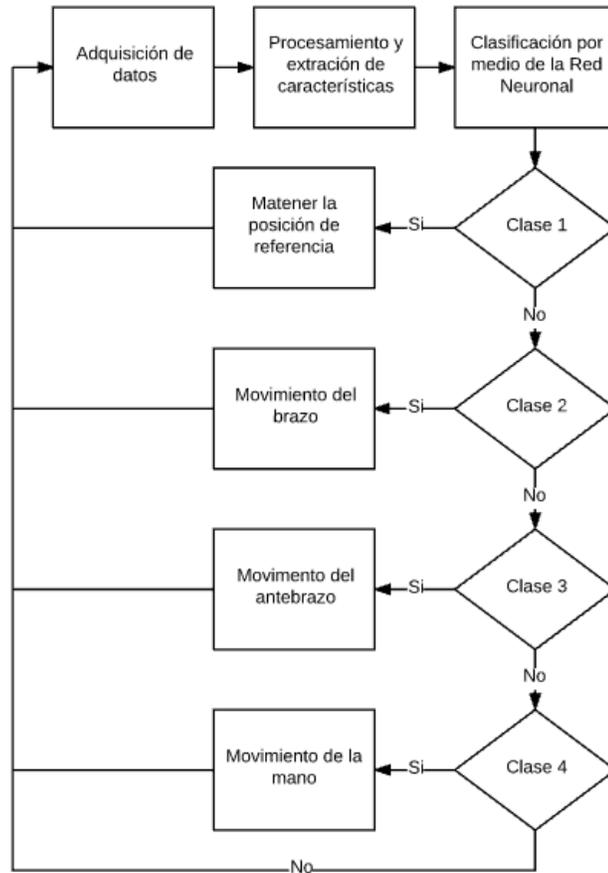


Figura 38. Diagrama de flujo del sistema.

La figura 37 muestra el diagrama de flujo del sistema. En él se pueden distinguir los diferentes procesos que se realizan.

4.1.1 CONFIGURACIÓN DEL HARDWARE

El sistema de adquisición de datos es el encargado de entregar las 3000 muestras de la señal EMG para el posterior procesamiento y extracción de características o patrones.

La Beaglebone Black tiene el sistema operativo Debian, en el cual viene instalado por defecto el idle de Python, donde se desarrolló el programa que implementa todo el sistema.

El brazo robótico recibe la señal PWM de la Beaglebone Black y ejecuta el respectivo movimiento, después vuelve a su posición de referencia, y el ciclo se vuelve a repetir.

4.1.2 CONFIGURACIÓN DEL SOFTWARE

Primero se instala el sistema operativo Debian en la micro sd, para eso debe bajar la imagen iso del sistema operativo Debian de la página web del fabricante, que es la siguiente <https://beagleboard.org>. Después de instalar el sistema en el micro sd, se conecta el micro sd a la Beaglebone Black y se inicia el sistema operativo. Debian trae por defecto instalado el idle de Python.

Para el proceso de adquisición de las datos que servirá como soporte para el entrenamiento de la red neuronal se utilizó un script, además se instalaron los paquetes de numpy que es un paquete para la computación científica con Python, y matplotlib que es una biblioteca de trazado 2D que produce figuras de calidad al estilo de MATLAB.

El entrenamiento de la red se hace por medio del software Matlab el cual cuenta con una toolbox ANN que permite trabajar con redes neuronales artificiales multicapa; luego su implementación se realiza en la tarjeta de desarrollo Beaglebone Black en lenguaje Python.

4.2 CAPTURA DE DATOS Y EXTRACCIÓN DE PATRONES

El sistema de adquisición transmite la señal EMG al ADC de Beaglebone Black, donde se utilizó un script que permitió capturar las señales EMG y extraer los patrones, con los cuales se realizó la data que sirvió como soporte para entrenar la red neuronal. Se tomaron 3000 muestras a intervalos de tiempo iguales, realizando los diferentes movimientos como mover la mano, flexionar el antebrazo y levantar el brazo; para cada 3000 muestras se calculaban los valores máximos y mínimos de la señal, se realizaba una diferencia entre los valores máximos y mínimos, estos valores se guardaban en un vector de error al cual se le calculaba el promedio y ese promedio se guardaba en un vector llamado data, este vector es guardado en la memoria de la Beaglebone Black para su posterior tratamiento en el software MATLAB para entrenar la red neuronal.

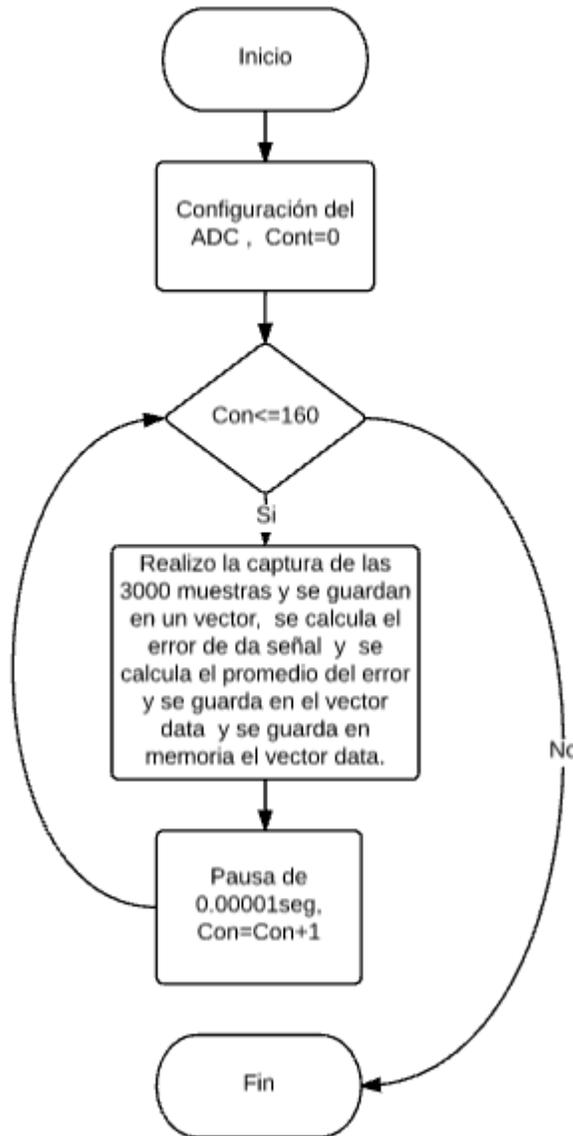


Figura 39. Diagrama de flujo captura y extracción de patrón de la señal EMG.

El tratamiento utilizado es simple se utilizan cálculos de valores máximos y mínimos de la señal, para calcular errores variables y finalmente se calcula un error absoluto de la señal, que es el promedio de los errores variables, cada error absoluto va conformando la data. El patrón que recibe la red neuronal para su entrenamiento es el error absoluto.

Se toman 160 datos que conformar la data para entrenar a red neuronal, para la posición de referencia se toman 40 datos, para el

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

movimiento de la mano 40 datos, para el movimiento del antebrazo 40 datos y para el movimiento del brazo 40 datos.

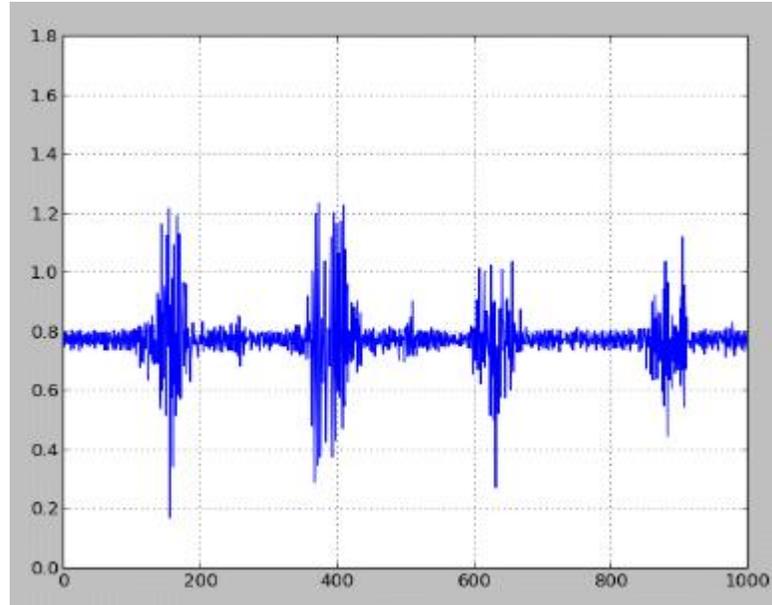


Figura 40. Señal EMG movimiento de la mano.

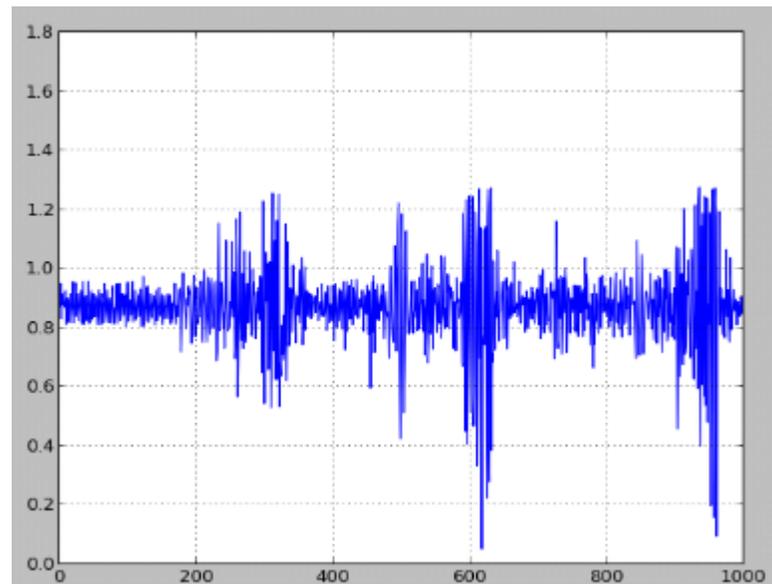


Figura 41. Señal EMG movimiento del antebrazo.

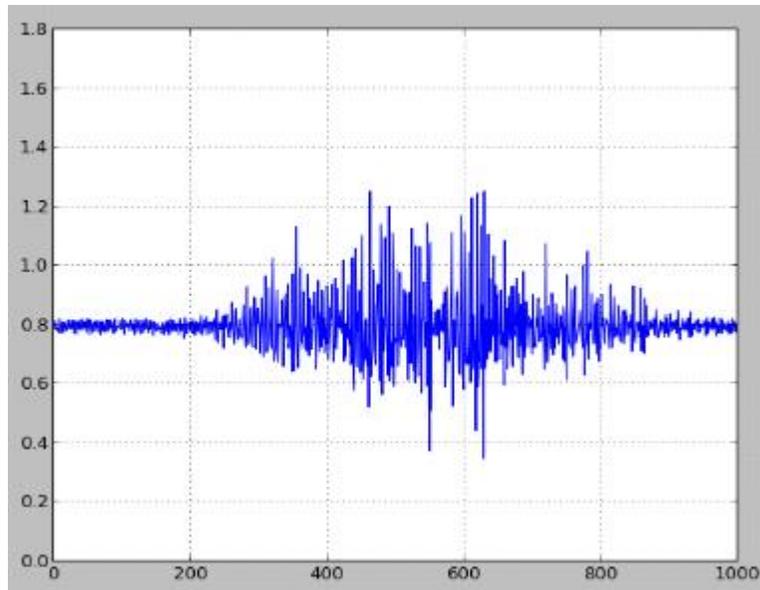


Figura 42. Señal EMG movimiento del brazo.

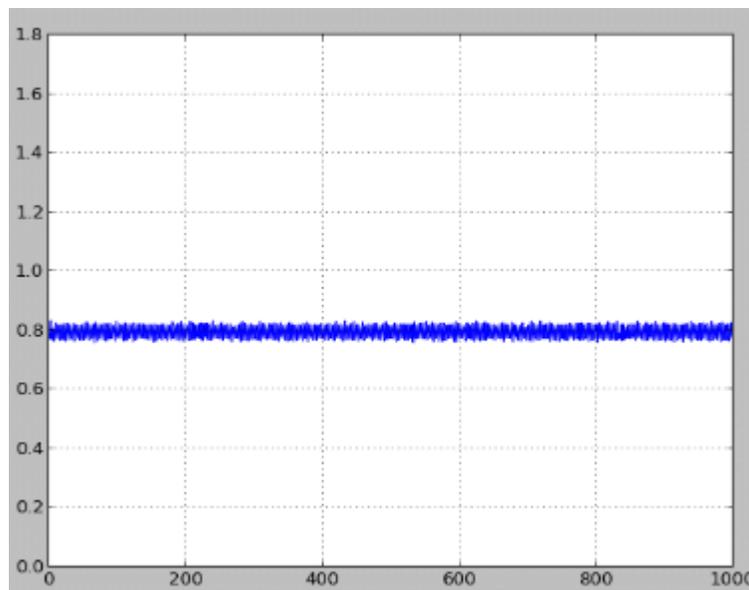


Figura 43. Señal EMG posición de referencia.

4.3 RED MLP EN MATLAB

MATLAB cuenta con la toolbox ANN que permite trabajar con redes neuronales artificiales multicapa. En un algoritmo sencillamente se asigna con que dígitos, vectores o matrices la red va a entrenar y con cuales va a validar. Queda a disposición del usuario configurar cuantas neuronas va a utilizar, función de activación de cada una de las capas, iteraciones entre neuronas, etc. Una vez la toolbox ha

finalizado el proceso en una estructura llamada net entrega toda la información de la red, y de ahí es precisamente donde se obtienen los pesos y los umbrales para llevarlos al código 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 absoluto.
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
```

Figura 44. Algoritmo Red MLP por partes.

En la figura 43 muestra el código implementado en MATLAB para entrenar la red neuronal, con los comentarios se va explicando que sucede en cada parte del código. Una vez que se ha encontrado los pesos que garantizan por lo menos el 95% de efectividad, la toolbox de MATLAB se detiene y envía los resultados de la red, la red tuvo una efectividad del 100%.

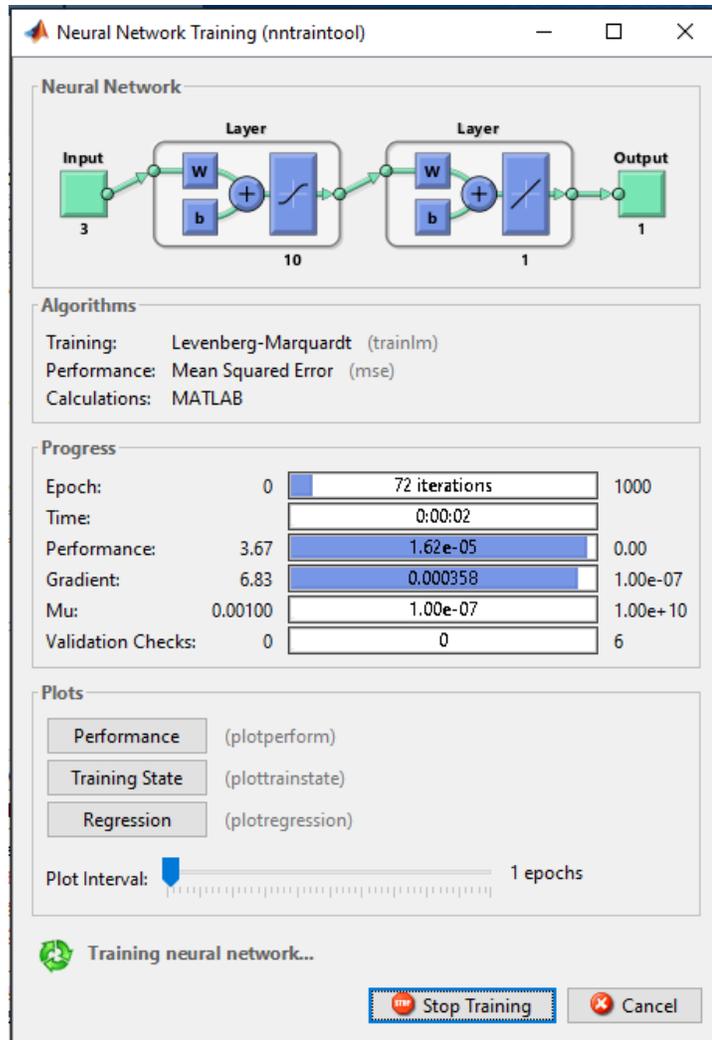


Figura 45. Entrenamiento de la red neuronal

4.3.1 PESOS DE LAS NEURONAS Y UMBRALES

Una vez entrenada la red, el toolbox envía una estructura donde se encuentran los valores de los pesos en la capa oculta, capa de salida y los umbrales respectivos. Esos valores son los que son importantes pues en base a ellos es que se diseña el programa en Python para la clasificación de las señales EMG.

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

Capa oculta	Pesos	Umbrales
Neurona 1	-1.350638643380458	5.1098958436778128
	7.8828374146046896	
	-7.4218459404932338	
Neurona 2	35.037685202385312	-19.907727506630454
	43.552266042322167	
	43.501057342743465	
Neurona 3	4.6840553294371965	-6.9040252142713587
	-13.692422975175973	
	6.4769291418071093	
Neurona 4	-7.6374185783332944	6.3733659017289321
	-9.2126831556408479	
	10.622175441571009	
Neurona 5	-68.982838566851555	2.2906106402495077
	-48.141134066069235	
	95.19570269114692	
Neurona 6	-12.541154788966319	-13.192720310711744
	39.967953901403071	
	28.254419055460403	
Neurona 7	73.932363950917747	-2.3905877385851957
	50.781332314404878	
	-100.80688806759301	
Neurona 8	35.062081839777761	-20.078977134998734
	43.575174175818532	
	43.54660422625949	
Neurona 9	12.088229652650444	12.581136620709051
	-38.074725143421382	
	-27.320729002161002	
Neurona 10	-10.654254286195938	-10.587644406408645
	31.933758407167048	
	24.276426624899674	

Tabla 6. Valores de los pesos y umbrales de las neuronas de la capa oculta

Capa de salida	Pesos	Umbral
Neurona 1	0.069285010088561361	1.8095898875555849
	5.6089347674519257	
	0.37886049441148978	
	2.3600826106295042e-06	
	0.98583465705445472	
	-3.7771516132781935	
	1.4858310824463998	
	-4.1089485556089649	
	-3.5734959921299474	
	-0.29634743930490587	

Tabla 7. Valores de los pesos y umbral de la capa de salida.

Estos son los valores que se usaran en código Python para la implementación de la red neuronal en la Beaglebone Black la cual será la encargada de realizar la clasificación de las señales EMG.

4.3.2 RESULTADOS DE ENTRENAMIENTO

En el entrenamiento de la red se puede simular para saber la efectividad de esta misma, con la función sim de Matlab se hacen una comparación de la salida de la red con la data de salida correspondiente para verificar la efectividad de la red.

Se tienen 10 muestras de las 4 clases, estas son aleatorias y se pasan por la red clasificadora. Ella entrega un resultado y de ahí se evalúa su efectividad.

Nombre del Movimiento	Resultado RN	Resultado Teórico
Posición referencia	Clase 1	Clase 1
Brazo	Clase 2	Clase 2
Antebrazo	Clase 3	Clase 3
Mano	Clase 4	Clase 4
Antebrazo	Clase 3	Clase 3
Brazo	Clase 2	Clase 2
Mano	Clase 4	Clase 4
Posición referencia	Clase 1	Clase 1
Antebrazo	Clase 3	Clase 3
Mano	Clase 4	Clase 4

Tabla 8. Resultados validación en Matlab

Este proceso se puede hacer con las demás muestras y va a dar el mismo porcentaje de efectividad del 100% presente en la tabla 8, lo que confirma que la red neuronal quedo bien estructurada y hace un excelente proceso clasificador.

4.4 ESTRUCTURA DEL PROGRAMA PRINCIPAL EN PYTHON

La estructura del programa principal como se había mencionado con anterioridad consta de tres procesos importantes, todos ejecutados en uno solo script implementado en la Beaglebone Black escrito en el lenguaje de programación Python.

Para recordar lo tres procesos consisten: 1. Adquisición de datos, 2. Inteligencia artificial, 3. Ejecución de movimiento.

Adquisición de datos

El sistema de adquisición entrega la señal EMG acondicionada al ADC de la tarjeta de desarrollo, el programa principal toma 750 muestras por segundo de la señal EMG, se toman en total 3000 muestras, el canal 1 toma 1000 muestras de la señal EMG proveniente del músculo palmar largo, este se estimula cuando se abre y cierra la mano; el canal 2 toma 1000 muestras de la señal EMG proveniente de los músculos bíceps y tríceps, estos se estimulan cuando se flexiona el antebrazo; y el canal 3 toma otras 1000 muestras provenientes del músculo deltoides, este se estimula cuando se mueve el brazo como subir y bajar el brazo. Cada una de las 1000 muestras se guarda en un vector que guarda las señales EMG provenientes de los diferentes músculos, a cada vector de señal EMG se le calcula los valores mínimo y máximos, para hacer una diferencia entre estos valores y cada resultado se guarda en un vector llamado error relativo, a este último vector se le calcula el promedio de sus valores y se llega a un valor llamada error absoluto. En total se tienen 3 errores absolutos que son el error absoluto del canal 1, el error absoluto del canal 2 y el error absoluto del canal 3, estos valores son las entradas de la red neuronal para la clasificación de los movimientos.

Red Neuronal

La red neuronal tiene 3 entradas y una salida, la capa oculta tiene 10 neuronas y la capa de salida tiene 1 neurona, la función de activación para las primeras es Tansig y para la de salida es Purelin.

Una vez que llega el valor del patrón extraído en el anterior proceso este se pasa por las 10 primeras neuronas donde se multiplican por sus pesos, se le suma el umbral y se calcula la función Tansig, la salida de cada neurona de la capa oculta es una entrada de la neurona de salida. En la neurona de salida las señales provenientes de la capa oculta se multiplican por los pesos, se suma el umbral y la salida al ser Purelin, la entrada es la misma salida.

La Beaglebone Black cuenta con punto flotante y Python tiene librerías matemáticas que permiten el uso directo de la función exponencial, se realiza una función la cual permita calcular el valor de la función tangente hiperbólica sigmoidea, lo cual permite implementar la red neuronal en esta tarjeta de desarrollo.

La función tangente hiperbólica sigmoidea (Tansig) se muestra en la figura 45.

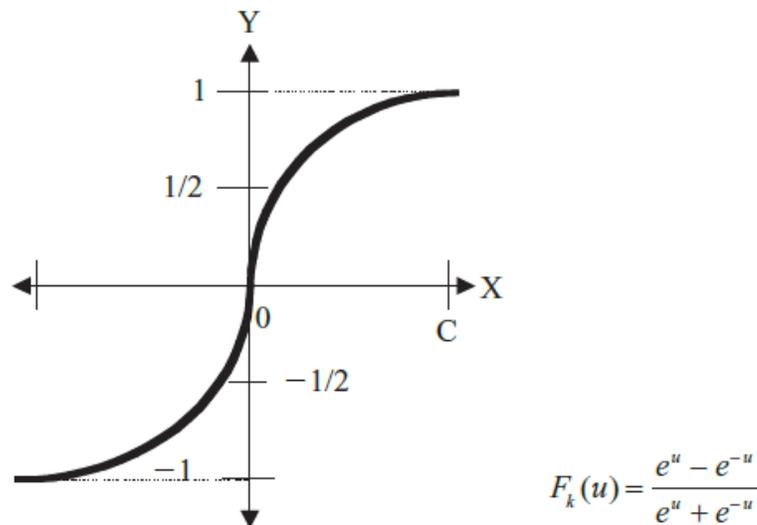


Figura 46. Función tangente hiperbólica sigmoidea [13]

Lo que realiza la red en este proceso es la clasificación del movimiento por medio del patrón obtenido en 4 clases, **clase 1** significa que se mantiene la posición de referencia, la **clase 2** significa el movimiento del brazo, la **clase 3** significa el movimiento del antebrazo, y la **clase 4** significa el movimiento de la mano.

Ejecución de movimiento

Después que la red neuronal clasifica el movimiento, la Beaglebone Black envía una señal PWM a los servomotores de brazo robótico para

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

realizar el movimiento respectivo, por ejemplo si la clasificación de la red fue clase 1 no se realiza ningún movimiento por que se mantiene la posición de referencia, y si la clasificación de la red es clase 4 el brazo robótico realiza el movimiento de la mano abriéndola y cerrándola y después vuelve a su posición de referencia para la siguiente clasificación de la red.

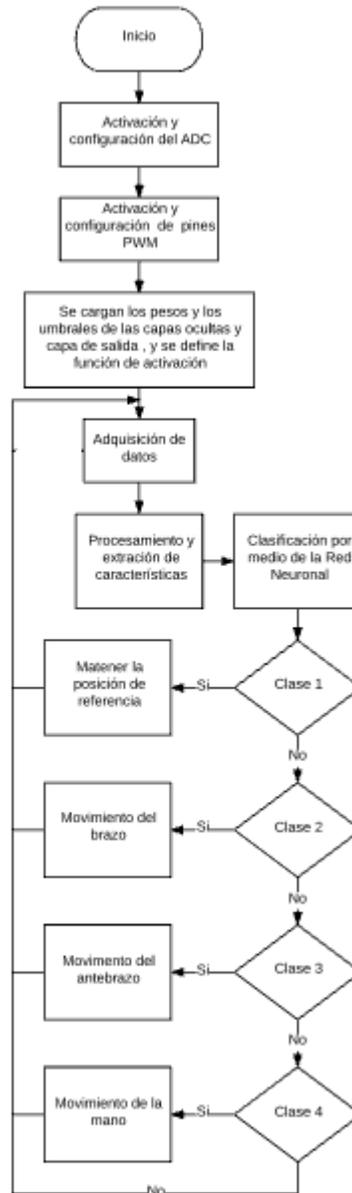


Figura 47. Programa principal.

5. ANÁLISIS Y RESULTADOS

5. ANÁLISIS Y RESULTADOS.....	- 67 -
5.1 ANÁLISIS Y RESULTADOS.....	- 68 -
5.2 ADC DE LA BEAGLEBONE BLACK.....	- 68 -
5.3 RED NEURONAL EN PYTHON.....	- 69 -
5.3.1 LIMITANTES.....	- 70 -
5.3.2 RESULTADOS Y PORCENTAJE DE ERROR.....	- 70 -
5.4 SISTEMA IMPLEMENTADO.....	- 71 -
5.5 ANALISIS ECONÓMICO.....	- 72 -

5.1 ANALISIS Y RESULTADOS

En esta parte del libro se presentan los resultados del proyecto en su parte final, los inconvenientes, porcentajes de error, validaciones y alternativas de desarrollo que se tuvieron en cuenta para implementar el proyecto completamente.

5.2 ADC DE LA BEAGLEBONE BLACK

El ADC es parte vital para el desarrollo a buen término del proyecto, ya que este es el encargado de convertir la señal EMG análoga en su equivalente digital, para facilitar la tarea del procesamiento de la información; por lo cual se debe verificar el correcto funcionamiento de este. El ADC de la tarjeta de desarrollo tiene las siguientes características: una resolución de 12 bits (0-4095), rango de voltaje de 0-1.8V, 7 canales análogos, ADC clock 3MHz, etc.

Para determinar si el dispositivo es apto para el proyecto se implementó una serie de pruebas con diferentes voltajes para verificar con la medida de entrada que tan preciso es el dispositivo.

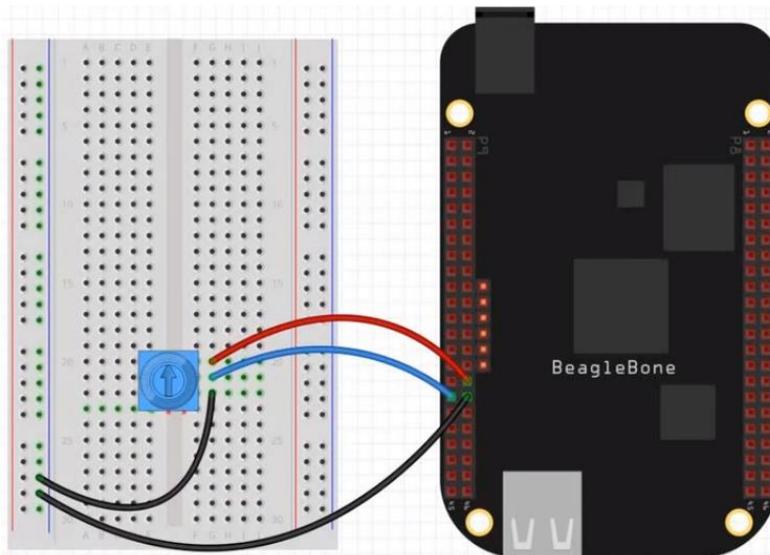


Figura 48. Esquema de prueba para el ADC.

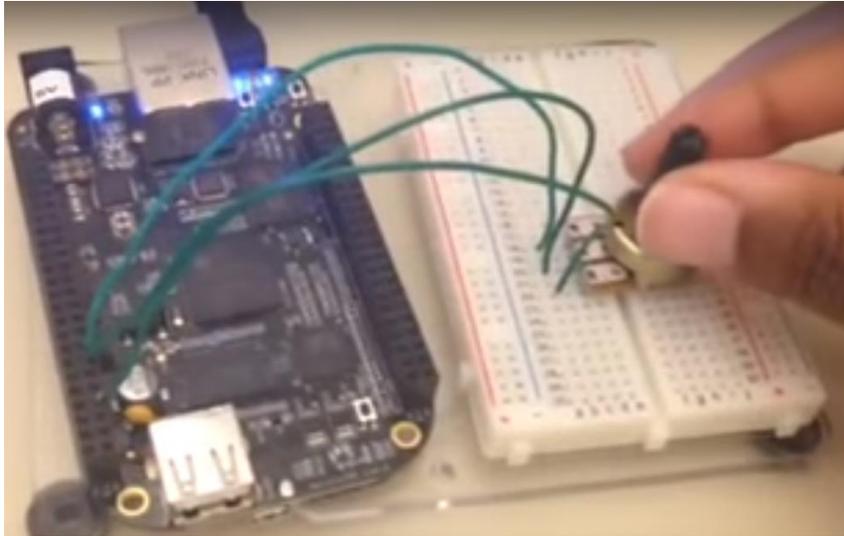


Figura 49. Circuito de prueba para el ADC implementado.

Voltaje ingresado Medido multímetro	Voltaje correspondiente ADC	Porcentaje de error
1V	0.9999999V	0.00001%
1.1V	1.0999999V	0.000009%
0.5V	0.4999997V	0.00006%
0.7V	0.6999998V	0.000028%
0.9V	0.8999997V	0.000033%
1.5V	1.4999998V	0.000013%
1.8V	1.7999996V	0.000022%

Tabla 9. Resultados prácticos ADC.

Los porcentajes de error arrojados por las pruebas del ADC son relativamente bajos, por la cual es idóneo el uso del ADC de la Beaglebone Black para el proyecto.

5.3 RED NEURONAL EN PYTHON

Antes de unir todos los programas y hacer una sola validación es necesario probar uno de los componentes principales del proyecto que es la red clasificadora para asegurar que esté funcionando correctamente.

La prueba para poder verificar que el algoritmo está funcionando correctamente es relativamente simple. Se trabaja con la data con la

cual se entrenó la red en MATLAB. Se introducen los datos de una respectiva clase y se verifica que la red arroje la clase indicada, por ejemplo los datos que entran a la red corresponden a la clase 2 y la red arroja que en la clasificación que el resultado es clase 2.

Si la red neuronal tuvo un entrenamiento óptimo, se encontrara en la capacidad de clasificar los nuevos datos que se capturen, a pesar de que estos sean totalmente nuevos.

El script como tal alcanza las 360 líneas de programación y el tiempo de ejecución de una validación es un promedio de 9s y 20%

Beaglebone Black	Consumo
RAM	20%(de 512Mb)
Líneas de código	360
Tiempo de ejecución	9s

Tabla 10. Recursos utilizados por la Beaglebone Black.

En la tabla 10 se encuentran los resultados de tiempo de ejecución, líneas de códigos del programa, recursos de memoria de la Beaglebone Black en la validación del script del sistema.

5.3.1 LIMITANTES

Una de las mayores limitante del sistema es que solo se puede clasificar un movimiento a la vez, debido a que el sistema se está ejecutando en código secuencial, por lo cual se ejecuta un movimiento y se esperar que el sistema realice la clasificación, para poder realizar otro movimiento. Para realizar procesos paralelos se necesita ejecutar el sistema en un dispositivo que trabaje bajo código concurrente, como los dispositivos lógicos programables como PLDs y FPGAs.

5.3.2 RESULTADOS Y PORCENTAJE DE ERROR

Los resultados de la clasificación de la red neuronal implementada en Python se muestran en la siguiente tabla.

Movimiento	Clases de las Muestras	Resultado obtenido
Posición referencia	Clase 1	Clase 1
Brazo	Clase 2	Clase 2
Mano	Clase 4	Clase 4
Antebrazo	Clase 3	Clase 3
Mano	Clase 4	Clase 4
Posición referencia	Clase 1	Clase 1
Brazo	Clase 2	Clase 2
Mano	Clase 4	Clase 4
Brazo	Clase 2	Clase 2
Antebrazo	Clase 3	Clase 3

Tabla 11. Resultados validación con datas.

El porcentaje de error obtenido fue del 0%; con este resultado es de esperar que al unir todos los algoritmos el resultado sea igual de efectivo que al hacerlo capturando los datos en tiempo real.

5.4 SISTEMA IMPLEMENTADO

Las pruebas se hacen en tiempo real, trabajando con el código final, y los resultados se muestran en la siguiente tabla.

Movimiento	Clases de las Muestras	Resultado obtenido
Mano	Clase 4	Clase 4
Brazo	Clase 2	Clase 2
Posición referencia	Clase 1	Clase 1
Antebrazo	Clase 3	Clase 3
Mano	Clase 4	Clase 4
Posición referencia	Clase 1	Clase 1
Brazo	Clase 2	Clase 2
Mano	Clase 4	Clase 4
Antebrazo	Clase 3	Clase 3
Brazo	Clase 2	Clase 2
Antebrazo	Clase 3	Clase 3

Tabla 12. Resultados validación con sistema implementado.



Figura 50. Sistema implementado realizando una validación

Los resultados fueron satisfactorios con un porcentaje de error de un 0%, lo cual indica que el sistema tiene un funcionamiento óptimo.

5.5 ANÁLISIS ECONÓMICO

A continuación se detalla los costos del proyecto, en la siguiente tabla se muestra la información.

COMPONENTES	CANTIDAD	PRECIO
Beaglebone Black	1	190.000 + 10.000 de envío
AD620	3	45.000 + 10.000 de envío
Conectores para electrodos	3	67.800 + 10.000 de envío
Memoria sd 8Gb	1	20.000
Servomotores	4	144.000 + 10.000 de envío
Pinza de aluminio	1	46.000 + 8.500 de envío
Tornillo y tuercas	60	8.000
Corte de partes en acrílico	4	18.000
Cables jumper 40 pines	1	8.000
Bolsa electrodos	3	75.000
Baquela fibra de vidrio 30x30	1	10.000
Regletas 40 pines	3	3.000
LF412	4	Muestras gratis
MAX295	3	Muestra gratis
TL082	8	Muestras gratis
TLV2254	3	Muestra gratis
Condensadores y Resistencias	40	3.000
Madera MDF	6	8.000
Conectores Jack de audio	3	4.500
TOTAL		\$698.800

Tabla 13. *Inversión del proyecto*

6. CONCLUSIONES

6. CONCLUSIONES.....	- 74 -
6.1 CONCLUSIONES FINALES.....	- 75 -

6.1 CONCLUSIONES FINALES

Se seleccionó la tarjeta de Beaglebone Black debido a que es una herramienta poderosa, dedicada a aplicaciones de instrumentación electrónica, y además permite trabajar con diferentes entornos de programación, especialmente el entorno de Python, que es uno de los lenguajes más utilizados para aplicaciones de inteligencia artificial en sistemas embebidos.

Se diseñó e implementó un sistema de adquisición de señales EMG que permite realizar satisfactoriamente la adquisición y el acondicionamiento de la señal. Además permite la transmisión hacia la tarjeta de desarrollo Beaglebone Black de las señales electromiográficas provenientes del brazo humano para procesarlas y extraer patrones, y generar una data que sirve como soporte de entrenamiento de la red.

Se logró identificar que para el movimiento del brazo el músculo que más se estimula es el deltoides, para la flexión del antebrazo los músculos que más se estimulan son el bíceps y el tríceps, y para el movimiento de la mano el músculo que más se estimula es el palmar largo; por lo cual la ubicación de los electrodos juega un papel muy importante en la adquisición de las señales EMG.

Para procesar señales EMG correspondientes a movimientos del brazo, la flexión del antebrazo, abrir y cerrar la mano, no es necesario aplicar técnicas de procesamiento complejas como Transformada de Fourier, Transformada Discretas del Coseno, Transformada Wavelet, etc., aplicar estas técnicas sería gastar recursos innecesarios de la tarjeta de desarrollo, porque con técnicas de procesamiento sencillas como restas, valores máximos y mínimos, promedios, etc., se pueden extraer patrones para clasificar estas señales.

Se implementó la red neuronal en el sistema embebido gracias a las prestaciones de la Beaglebone Black, y a la versatilidad y optimización de las librerías de Python para este tipo de aplicaciones.

PROCESAMIENTO DE SEÑALES EMG EN UN SISTEMA EMBEBIDO PARA EL CONTROL NEURONAL DE UN BRAZO ROBÓTICO

El entrenamiento de la red se realizó con la data obtenida de las señales EMG del brazo y por medio del (MLP), la cual obtuvo un 100% de efectividad en la validación con el software MATLAB.

Una de las limitantes más relevantes del sistema es que solo puede clasificar un movimiento a la vez, porque estamos trabajó bajo código secuencial, no se pueden realizar 2 movimientos a la vez como flexional el brazo y abrir y cerrar la mano, la red puede arrojar un resultado erróneo o puede clasificar alguno de los dos movimientos.

Se implementó un brazo robótico de 3 grados de libertad, se acopló todo el sistema, posteriormente se validó todo el sistema desarrollado presentando un porcentaje de error de un 0%; el sistema clasifica exitosamente los movimientos para controlar el brazo robótico.

BIBLIOGRAFÍA

- [1] DANE. ESTADÍSTICAS POR TEMA. DEMOGRAFÍA Y POBLACIÓN. DISCAPACIDAD. INFORMACIÓN ESTADÍSTICA. TOTAL NACIONAL. [En línea].
<http://www.dane.gov.co/index.php/poblacion-y-demografia/discapacidad>. [Citado el 1 de Abril de 2016].
- [2] W. J. ARCOS LEGARDA, "SISTEMA DE CONTROL DE MOVIMIENTO PARA PRÓTESIS DE MANO," pp. 7–9, 2005.
- [3] C. A. Quinayás-Burgos y C. A. Gaviria-López, "SISTEMA DE IDENTIFICACIÓN DE INTENCIÓN DE MOVIMIENTO PARA EL CONTROL MIOELÉCTRICO DE UNA PRÓTESIS DE MANO ROBÓTICA", Ing. Univ., vol. 19, no. 1, pp. 27-50, Ene.-Jun., 2015.
- [4] I. A. Cifuentes González, "DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA PARA LA DETECCIÓN DE SEÑALES ELECTROMIOGRÁFICAS", pp. 36–91, 2010.
- [5] K. Xing, P. Yang, J. Huang, Y. Wang, Q. Zhu, "A REAL-TIME EMG PATTERN RECOGNITION METHOD FOR VIRTUAL MYOELECTRIC HAND CONTROL", ScienceDirect, pp. 1-11, 2014.
- [6] Esp. H. Romo, Ing. J. C. Realpe, PhD. P. E. Jojoa, "ANÁLISIS DE SEÑALES EMG SUPERFICIALES Y SU APLICACIÓN EN CONTROL DE PRÓTESIS DE MANO", Revista Avances en Sistemas e Informática, Vol. 4 No. 1, pp. 1-10, 2007,
- [7] J. D. Ramírez Corzo, "DESARROLLO DE UN SISTEMA DE COMUNICACIÓN SILENCIOSA DUAL BASADO EN HABLA SUBVOCAL", pp. 7-81, 2015.

- [8] D. Nahmias, "ELECTROMYOGRAPHY ANALYSIS AND RECOGNITION FOR HUMAN DEVICE INTERFACE", Engineering 090: Senior Design Project, pp. 1-72, 2014.
- [9] M.J. Timmons y R.B. Tallitsch, "SISTEMA MUSCULAR: MUSCULATURA DE LAS EXTREMIDADES", en Anatomía Humana, Sexta edición, Ed. PEARSON EDUCACIÓN S. A., 2009, pp 284-296.
- [10] M. Prives, N. Lisenkov y V. Bushkovich, Anatomía Humana, Tomo 1, Quinta Edición, Ed. MIR Moscú, 1984, pp 341-350.
- [11] R.C. Dorf y J.A. Svoboda, "Amplificador Operacional ", en Circuitos Eléctricos, Sexta Edición, Ed. Alfaomega, 2006, pp 1-762.
- [12] S. Franco, "Fundamentos de los Amplificador Operacionales ", en Diseño con Amplificadores Operacionales y Circuitos Integrados Analógicos, Tercera Edición, Ed. McGraw-Hill, 2005, pp 1-408.
- [13] P. Ponce Cruz, "Redes Neuronales Artificiales", en Inteligencia Artificial con Aplicaciones a la Ingeniería, Primera Edición, Ed. Alfaomega, 2010, pp. 193-282.
- [14] J. E. Ortiz Sandoval, "IMPLEMENTACIÓN DE UN MODELO NEURONAL EN UN DISPOSITIVO HARDWARE (FPGA) PARA LA CLASIFICACION DE COMPUESTOS QUIMICOS EN UN SISTEMA MULTISENSORIAL (OLFATO ELECTRÓNICO)," pp. 1–34, 2013.
- [15] RASPBERRY PI. Raspberry pi 2. [En línea]. <https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/>. [Citado el 30 de Octubre de 2015].
- [16] TEXAS INSTRUMENTS. BeagleBone Black Developmental. [En línea]. <http://www.ti.com/tool/beaglebk>. [Citado el 30 de Octubre de 2015].

- [17] TEXAS INSTRUMENTS. ARM Cortex-M4F based MCU TM4C1294 Connected LaunchPad. [En línea]. <http://www.ti.com/tool/ek-tm4c1294xl>. [Citado el 30 de Octubre de 2015].
- [18] A. E. Marín Herrera, "IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD BANCARIO EN SISTEMAS EMBEDIDO UTILIZANDO INTELIGENCIA ARTIFICIAL", pp. 39–42, 2015.
- [19] G. M. Martínez, S. A. Jáquez, J. Rivera y R. Sandoval, "Diseño propio y Construcción de un Brazo Robótico de 5 GDL", RIEE&C, Vol. 4, pp. 1-7, 2008.
- [20] C.R. Batz Saquimux, "Diseño y construcción de un brazo robótico", Universidad de San Carlos Guatemala, pp. 17-131, 2005.
- [21] TEXAS INSTRUMENTS. Sitara Processor AM3358. [En línea]. <http://www.ti.com/lit/ds/symlink/am3358.pdf>. [Citado el 30 de Octubre de 2015].