

**UNIVERSIDAD RICARDO PALMA  
FACULTAD DE INGENIERÍA**



***Proyecto de Investigación 2016***

**Título del Proyecto:**

Red neuronal ART2 embebida en hardware para el reconocimiento de patrones de voz en tiempo real.

**Responsable:**

Dr. Ing. Pedro F. Huamaní Navarrete

**Lima - Perú  
Febrero - 2017**

# **RED NEURONAL ART2 EMBEBIDA EN HARDWARE PARA EL RECONOCIMIENTO DE PATRONES DE VOZ EN TIEMPO REAL**

**PEDRO FREDDY HUAMANÍ NAVARRETE**

## **RESUMEN**

En este trabajo de investigación se presenta, en primer lugar, la simulación de una técnica matemática de tratamiento digital de señales de audio para reconocer 10 mandos de voz, y en segundo lugar la implementación de tales técnicas en el hardware embebido NI myRIO 1900. Para alcanzar el reconocimiento de los mandos de voz se analizó dicha señal por segmentos, con la finalidad de recortar el silencio tanto al inicio como al final de la grabación. Luego, se utilizó un filtro de pre-énfasis seguido de un pasa banda, y posteriormente se obtuvieron los coeficientes de predicción lineal (LPC) que fueron utilizados para entrenar una red neuronal artificial adaptativa ART2. El reconocimiento de los mandos de voz se simuló a través de una combinación del encendido/apagado de los 4 leds del propio hardware embebido, logrando una efectividad del 88% en una base de datos de 600 grabaciones de voz digital.

Palabras claves: Red Neuronal ART2, hardware embebido myRIO, Coeficientes LPC.

## **ABSTRACT**

In this research, we first present the simulation of a mathematical technique of digital processing of audio signals to recognize 10 voice commands, and secondly the implementation of such techniques in the embedded hardware NI myRIO 1900. For Recognition of the voice commands was analyzed by segment, with the purpose of cutting the silence both at the beginning and at the end of the recording. Then, a pre-emphasis filter followed by a bandpass were used, and later the linear prediction coefficients (LPC) were used to train an adaptive artificial neural network ART2. The recognition of the voice commands was simulated through a combination of on/off the 4 leds of the embedded hardware itself, achieving a 88% effectiveness in a database of 600 digital voice recordings.

Keywords: Neural Network ART2, embedded myRIO hardware, LPC, LabVIEW.

## INTRODUCCIÓN

En los últimos años la tecnología digital viene gobernando todo el mundo, desde los electrodomésticos en nuestras casas, pasando por las grandes industrias hasta el sector militar y científico. Por tal razón, se cuenta con nuevas técnicas matemáticas que, al ser implementadas algorítmicamente en un lenguaje de computación de alto nivel, resuelven diversos problemas que la antigüedad resultaba muy difíciles de ser resueltos. Particularmente, el área de procesamiento digital de señales ha resultado ser una rama muy importante en cuanto a investigaciones se trata; pues, es muy aplicado para el sector de bioingeniería y asimismo para la biometría, apoyándose en software de computación científica como el Matlab y el Labview, dos herramientas de programación con enormes ventajas y facilidades para la implementación algorítmica con acceso al mundo real, a través de dispositivos de hardware externos. Tal es el caso del hardware embebido de National Instruments myRIO 1900<sup>1</sup>, con tecnología FPGA<sup>2</sup>.

De esta manera, la utilización de un hardware embebido es capaz de almacenar algoritmos de programación con el propósito de realizar una tarea particular, el reconocimiento de patrones de voz en tiempo real. Para ello, fue necesario complementar la investigación con la teoría de redes neuronales artificiales, con el fin de entrenar un grupo de patrones previamente procesados y asociarlos una salida particular. Sin embargo, como es de conocimiento, las voces pueden cambiar en tono o duración, por lo que se hizo necesario contar con un grupo elevado de grabaciones por cada uno de los participantes en el proceso de entrenamiento de la red neuronal. Específicamente, se empleó la red denominada ART2. Esta a su vez se encargó de formar nuevos conjuntos de patrones de entrada en base a las propias características de las señales de voz de un participante en particular.

Para el desarrollo de este proyecto de investigación no fue suficiente realizar la programación del hardware embebido con ayuda del software Labview, sino también fue necesario realizar las pruebas y simulaciones en el software de computación científica Matlab, por la facilidad de manejo de variables a través de vectores y matrices, así como por la ventaja de contar con una librería de procesamiento digital de señales y de redes neuronales. De esta forma, lograda la simulación, se procedió a embeber o incluir el algoritmo desarrollado en Matlab, en el hardware myRIO de National Instruments, para finalmente recibir los mandos de voz a través del micrófono conectado directamente a este dispositivo, y así lograr identificar dichos mandos para la realización de una tarea

---

<sup>1</sup> Dispositivo de diseño embebido.

<sup>2</sup> Dispositivo lógico constituido por un arreglo de compuertas programables.

en particular. Para esta última parte, solamente se definieron 10 tareas particulares correspondientes a los 10 mandos de voz, lo cual correspondió al encendido de las combinaciones de los 4 leds presentes en el hardware myRIO 1900. Pues, esto resultaba más accesible para trabajos futuros, porque facilitaría a personas con carencia de extremidades superiores, a permitir realizar tareas domésticas simples como encender o apagar artefactos electrodomésticos, girar botones de encendido, entre otros casos existentes, sólo con el mando de voz correspondiente.

### **Planteamiento del Problema**

Con el aumento de la población peruana, también incrementa la probabilidad de la existencia de un mayor número de personas con alguna discapacidad en sus extremidades, sea por una situación congénita o por algún accidente del que haya sido partícipe. Por lo tanto, para este grupo de personas se vuelve complicado encender o apagar algún artefacto electrodoméstico, cambiar las opciones de un control remoto, girar la pantalla de un televisor de un lado a otro, entre otras innumerables acciones propias del quehacer diario. Por tal razón, se propone a través de este proyecto de investigación, el desarrollo algorítmico de una técnica de procesamiento digital de señales complementada con una red neuronal artificial adaptativa, para embeberlo en un dispositivo de hardware de National Instruments denominado myRIO, para contar con una interacción en tiempo real, debido a que este dispositivo utiliza un procesador ARM® Cortex™-A9 dual-core en tiempo real así como Entradas/Salidas personalizadas propias de la tecnología FPGA Xilinx.

### **Importancia**

Este estudio resulta importante porque su implementación permitirá dar facilidades a las personas que carecen de extremidades superiores, a realizar diferentes acciones propias del día a día, y haciendo uso de un grupo de patrones de voz previamente almacenadas y procesadas en tiempo real, a través del dispositivo de hardware myRIO de National Instruments. Esto permitirá convertir el proyecto en un producto portátil, aislándose completamente de la computadora y facilitando la interacción hasta con 06 señales analógicas de salida al mismo tiempo. Por lo tanto, esta investigación se justifica por la factibilidad de utilizar técnicas matemáticas de procesamiento digital de señales de voz, por la variada bibliografía referente a redes neuronales adaptativas, así como por la facilidad de adquirir y procesar señales de voz en tiempo real a través del dispositivo de hardware myRIO, que posee un procesador ARM® Cortex™-A9 dual-

core, que cuenta con componentes internos, un acceso a software y una amplia biblioteca con recursos y tutoriales, tal como lo indica la página principal de National Instruments [2].

## **Antecedentes**

### *Nacionales*

- *Giovani Simón [13], en la tesis titulada Filtro adaptativo LMS y su aplicación en el reconocimiento de palabras aisladas para el control de un equipo de sonido por medio de la voz, de la Pontificia Universidad Católica del Perú, señala que el intentar controlar un equipo de sonido por medio de órdenes verbales es un gran desafío, debido a que los algoritmos de reconocimiento de voz son muy frágiles y tienen una buena eficiencia sólo en un ambiente sin ruido. Por lo cual, basta la presencia de un ligero sonido para que la voz no pueda ser reconocida fácilmente. De esta manera, la investigación desarrollada por Giovani intenta mejorar el porcentaje de aciertos de un sistema de reconocimiento ante la presencia de ruido, y por tal motivo se utilizó un equipo de sonido marca Panasonic y modelo SC-AK45 con control remoto modelo EUR644853, del cual se tomaron algunas de las funciones para el control mediante la voz. Por lo tanto, se empleó un filtro adaptativo LMS que es la interfase de entrada al sistema y que utiliza a la vez dos señales (la voz con ruido y la referencia) capturadas por dos micrófonos marca AKG D-230 implementado en la tarjeta de evaluación DSP56002EVM de Motorola. Este filtro cumple la función de reducir el nivel del sonido del equipo en la señal que contiene la voz, para que el sistema de reconocimiento pueda tener buena eficiencia y por lo tanto un bajo índice de error. La señal filtrada ingresó al sistema de reconocimiento implementado en la tarjeta de desarrollo TMS320C6711 que permitió el control en tiempo real, es decir en el instante en que la orden sea pronunciada, el sistema procesa la señal de voz y genera una respuesta enviando una señal infrarroja a través de un pequeño circuito IR que es la interfase entre la tarjeta y el equipo de sonido. Asimismo, el autor afirma que esta investigación es la base para controlar cualquier sistema que posea audio, como el caso de un TV, un DVD u otro.*

### *Internacionales*

- *Luis Cruz y Marco Acevedo [14], en el artículo titulado Reconocimiento de Voz usando Redes Neuronales Artificiales Backpropagation y Coeficientes LPC, utilizan en la captura 05 archivos de audio en formato WAV en 05 personas diferentes, bajo*

el formato de monofónico, 16 bits por muestra y una frecuencia de muestreo de 11 KHz. Luego, acotaron la señal de voz eliminando la parte inicial y final, para luego hacerlo pasar a través de un filtro pasa-bajo de la familia Wavelets Daubechies que dio como resultado un mayor porcentaje de compactación de energía de la señal acotada. Posteriormente, se continuó con la normalización de la nueva señal filtrada con la finalidad de ajustar los parámetros a una sola escala, y de esa forma al ser utilizados por la Red Neuronal no cause problema de estabilidad. Para la normalización, se procedió a obtener la media y la varianza de la señal de interés. Y se restó la media de la señal de voz dividida por la varianza, y también dividida por el mayor valor absoluto. Posteriormente, se continuó con la obtención de los coeficientes LPC, comenzando la prueba con 2 coeficientes y finalizando con 15 coeficientes. Finalmente, se procedió al entrenamiento de la red neuronal backpropagation conformada por 3 capas de neuronas: 25, 21 y 5, un error cuadrático medio igual a 0.005, y una tasa de aprendizaje de 0.05. Los resultados obtenidos de este proceso señalan que para los vectores de entrada que participaron en la etapa de entrenamiento, el grado de reconocimiento fue del 100%, mientras que para aquellos que no participaron en el entrenamiento, el reconocimiento disminuyó al 96%.

- Jaime Varela y Johan Loaiza [15], en el trabajo de grado titulado Reconocimiento de Palabras aisladas mediante redes neuronales sobre FPGA, señalan que el nivel de paralelismo de las Redes Neuronales es ampliamente significativo, tanto que no es explotado cuando su implementación se realiza en computadores de propósito general. Por lo cual, en este trabajo de investigación se buscó diseñar el sistema de reconocimiento de palabras aisladas en un computador de propósito general y diseñar en VHDL la red neuronal del sistema de clasificación apta para la implementación sobre una FPGA, para aprovechar el paralelismo y comparar los resultados del desempeño, eficiencia y confiabilidad. De esta manera, en esta tesis de grado se presentaron ambos desarrollos y se validaron según la velocidad de procesamiento de paquetes de palabras. El resultado final correspondió a diseños iguales, con aciertos similares, pero con velocidades de procesamiento realmente diferentes. Para el pre-procesamiento de la voz, se utilizó la energía y la magnitud para distinguir segmentos de voz sordos y sonoros. Asimismo, se empleó un filtro de pre-énfasis y una normalización con una ventana de Hamming. También se emplearon los coeficientes LPC para extraer las características de las señales de voz

utilizadas y posteriormente se complementó con los Coeficientes Cepstrum sobre la Escala de Frecuencias Mel (MFCC). En cuanto a la red neuronal, se valieron de tres estructuras diferentes, pero utilizando siempre 3 capas de neuronas, lo cual permitió que una de ellas alcance un mejor resultado respecto al error medio cuadrático.

- M. Yommi [16], en el artículo titulado Sistema automático de reconocimiento de voz para control de acceso, señala que esta investigación aborda el estudio de las técnicas de procesamiento de señales de voz que permiten extraer las características intrínsecas de la identificación del locutor. El objetivo de este trabajo fue delimitar el acceso a solo aquellos locutores identificados y habilitados, donde dicho procedimiento se realiza por medio de una interfaz máquina-usuario. El método propuesto en esta investigación posee una capa física (hardware) implementada en torno a Arduino Uno y una capa de aplicación (software) desarrollada sobre la plataforma de MatLab 7.10. Para evaluar las características de los locutores se utilizaron los coeficientes cepstrales de frecuencia Mel. Para la clasificación, una red neuronal artificial con topología perceptrón multicapa y un aprendizaje supervisado del tipo Backpropagation. El entrenamiento y prueba del sistema se llevó a cabo con una base total de 80 muestras diferentes de voz. Finalmente, el rendimiento arrojado refleja que el sistema posee un porcentaje de acierto promedio del 64,58 %.

## **Marco Conceptual**

El marco conceptual está comprendido por los siguientes subtítulos.

### a) Hardware embebido NI myRIO 1900

Es un dispositivo de diseño embebido que ofrece una herramienta accesible que ayuda a los estudiantes y profesores a convertir sus ideas de proyectos de ingeniería en sistemas reales. Es un desarrollo de la compañía National Instruments, como una evolución de las tarjetas de adquisición de datos NI USB 6008 y NI USB 6009. Este dispositivo permite la manipulación de hasta 10 entradas analógicas, 6 salidas analógicas, 40 líneas de entrada y salida digital. Además, cuenta con un acelerómetro interno, LEDs, tecnología inalámbrica por WiFi, un procesador dual-core ARM Cortex-A9 en tiempo real, un Xilinx FPGA (Field Programmable Gate Array) personalizado y es programable con el software Labview o lenguaje C, tal como lo indica la página oficial de National Instruments [2]. La figura 01 muestra una fotografía del dispositivo de hardware, y al lado derecho una representación de los principales puertos de acceso.

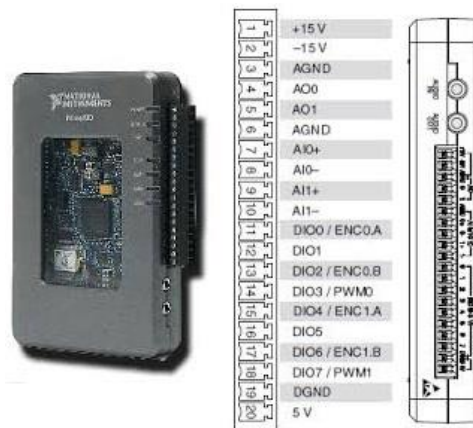


Figura 01. Fotografía del dispositivo myRIO 1900 y representación de los puertos de acceso [2].

*b) Red Neuronal Adaptativa ART2*

Una de las definiciones de red neuronal artificial es la de una agrupación de elementos procesadores interconectados masivamente, que imitan de manera somera el comportamiento de una neurona biológica. Con un campo de acción amplio y aplicaciones múltiples en reconocimiento de patrones, predicción de señales e identificación de sistemas. Por otro lado, una red neuronal artificial adaptativa ART (Adaptive Resonance Theory), es un modelo de red que cuenta con un aprendizaje del tipo competitivo, donde una neurona se identifica como ganadora dentro de un conjunto de neuronas definidas en la red. Esta neurona vencedora se caracterizará porque contará con un vector de pesos que representará a un cluster particular, tal como lo indica Martín Bonifacio en [1].

En cuanto a la diferencia de red neuronal adaptativa ART1 y ART2, se basa principalmente en el tipo de señal que se utilizará en el entrenamiento de dicha red. Es decir, si se utilizan señales binarias representadas por 0's y 1's, será la red ART1 la utilizada. Sin embargo, si se utilizan señales reales, la ART2 será la elegida. Para este trabajo de investigación, por tratarse del procesamiento y reconocimiento de señales de voz, se ha propuesto utilizar la red adaptiva ART2.

La representación matemática de la actualización de los pesos sinápticos, basado en el aprendizaje competitivo, se muestra a continuación, según S. Haykin [7]. Asimismo, la estructura de red neuronal se expresa en la figura 02.

$$w[n + 1] = w[n] + \alpha * (x[n] - w[n]) \quad (01)$$

Donde:



$w[n]$ :            representa el vector de pesos sináptico actual.  
 $w[n+1]$ :        representa el nuevo vector de pesos sináptico.  
 $x[n]$ :             representa el vector de entrada actual.  
 $\alpha$ :              representa al factor de aprendizaje.

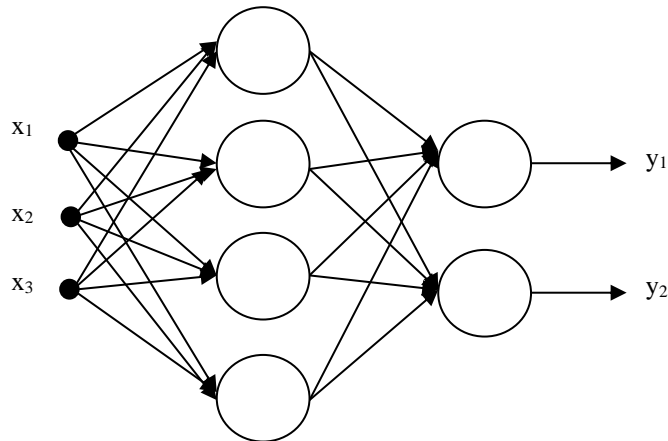


Figura 02. Ejemplo de estructura de red neuronal artificial adaptativa.

c) Filtrado y digitalización de la voz

Una señal de voz es un tipo de onda acústica que, a través de un transductor como el micrófono, se convierte a una señal eléctrica. Sin embargo, aún continúa siendo del tipo analógico. Por tal razón, es necesario realizar la digitalización de la misma, haciendo cumplir el Teorema de Muestreo.

De esta manera, una señal de voz, generalmente, es digitalizada a razón de 125 micro segundos. Lo cual representa a 8000 muestras por cada segundo de señal de voz.

Una vez realizada la operación de digitalización, es necesario aplicar un tipo de filtro digital que puede ser del recursivo o no recursivo. Según Antoniou [11], el filtro digital recursivo, todo polo, tiene la siguiente función de transferencia:

$$H(z) = \frac{1}{1 - \sum_{i=1}^p a_i z^{-i}} \quad (02)$$

Donde:

$a_i$ , para  $i=1, 2, 3, \dots, p$ , representa los coeficientes del filtro digital recursivo.

Y, el filtro no recursivo presenta una variante tal como se muestra a continuación:

$$H(z) = \sum_{i=1}^p a_i z^{-i} \quad (03)$$

Donde:

$a_i$ , para  $i=1, 2, 3, \dots, p$ , representa los coeficientes del filtro digital no recursivo.

#### d) Coeficientes Predictivos Lineales LPC

Representan a un conjunto de parámetros que caracterizan el aparato vocal, obtenidos a partir de un modelo de filtro-fuente de producción de voz, según Alcaim [5]. Asimismo, se indica, que el modelo LPC de producción de voz está basado en la observación de que cada muestra de voz puede ser estimada, a partir de combinaciones lineales de muestras anteriores. Es decir, se cumple la siguiente expresión matemática [5]:

$$\hat{s}[n] = \sum_{i=1}^P a_i * s[n-i] \quad (04)$$

Por lo tanto, considerando a  $e[n]$  como señal de excitación o error de una estimación, se tiene [5]:

$$s[n] = \left( \sum_{i=1}^P a_i * s[n-i] \right) + e[n] \quad (05)$$

Donde:

$a_i$  : es el coeficiente LPC.

$s[n]$ : señal de voz.

$e[n]$ : señal de excitación.

#### e) Software LabVIEW de National Instruments

Es un entorno de desarrollo diseñado específicamente para acelerar la productividad de ingenieros y científicos. Cuenta con una sintaxis de programación gráfica que facilita visualizar, crear y codificar sistemas de ingeniería. Asimismo, este software de programación gráfica, ayuda a ingenieros a convertir sus ideas en realidad, reducir tiempos de pruebas y ofrecer análisis de negocio basado en

datos recolectados, según como lo señala la página oficial de National Instruments [3]. Asimismo, cuenta con módulos adicionales que permiten ampliar sus prestaciones a diferentes campos de acción: módulo de Control Design and Simulation, módulo IMAQ Vision, módulo Real Time, módulo FPGA, entre otros. A esto se le suma la variedad de dispositivos de hardware, propias de National Instruments, desarrollados tanto para la enseñanza como para la industria, permitiendo de esta manera una mejor y mayor aplicación de la ingeniería.

## OBJETIVOS

### Objetivo General

Reconocer patrones de voz, en tiempo real, utilizando una red neuronal ART2 embebida en el dispositivo de hardware myRIO.

### Objetivos Específicos

- Desarrollar algoritmos de procesamiento digital de señales para el reconocimiento, en tiempo real, de un conjunto de 10 mandos de voz, para luego embeberlos en el hardware myRIO 1900 de NI.
- Entrenar una red neuronal artificial adaptativa ART2, con la participación de un grupo de 10 personas, para luego embeber las matrices de pesos sinápticos en el hardware myRIO 1900 de NI.
- Utilizar el dispositivo de hardware myRIO 1900 de NI para realizar una acción final a partir del conjunto de mandos de voz reconocidos, en tiempo real.

## VARIABLES DE ESTUDIO

En cuanto a las variables de estudio empleadas en este trabajo de investigación, en función al tipo de investigación correlacional desarrollada, se tiene:

- Variable Independiente: Aplicación de técnicas matemáticas de procesamiento digital de voz.

Indicadores: Filtrado Digital de Voz, Segmentación de la Señal de Voz y Obtención de Coeficientes LPC.

- Variable Dependiente: Reconocimiento de 10 mandos de voz.

Indicadores: Encendido de una combinación de 4 leds del hardware embebido myRIO.

## METODOLOGÍA

### Tipo de investigación

El tipo de investigación empleado fue tecnológico y aplicativo.

### Método de investigación

El método de investigación fue del tipo empírico y experimental. Debido a que se tomaron datos a través de un medio digital y se procedió a aplicar técnicas matemáticas de procesamiento de imágenes de forma práctica.

### Diseño de investigación

El diseño específico de investigación es del tipo experimental y correlacional.

### Instrumento de recolección de datos: Captura y digitalización de mandos de voz

La captura y digitalización de las señales o mandos de voz se realizó directamente con el software Matlab, haciendo uso de un micrófono omnidireccional conectado al puerto multimedia de la Laptop, y apoyado de las librerías de Audio y Video del propio software utilizado. Seguidamente se muestra el código de programa utilizado para el proceso de captura y a la vez digitalización, teniendo presente las siguientes características para el procedimiento de digitalización.

- Frecuencia de muestreo igual a 8000 muestras/segundo.
- Número de bits por muestra igual a 8.
- Número de canales: monofónico.
- Número de segundos de grabación igual a 2.0 s.

```
>> Fs = 8000;
```

```
>> y = wavrecord( 2*Fs , Fs , 'double' );
```

Una vez digitalizado los mandos de voz, se continuó con el almacenamiento para crear la base de datos para este proyecto.

### Creación de Base de Datos

Como fueron 10 personas las que participaron en el proceso de captura y digitalización de los mandos de voz, se procedió a realizar 06 grabaciones de voz por cada una de

las personas participantes. Esto permitió que se alcance un total de 600 archivos de audio en formato WAV. Ver la tabla 01, así como la lista de los mandos de voz utilizados con los respectivos nombres de archivos de extensión WAV<sup>3</sup>.

- 1er mando de voz: IZQUIERDA → izquierda\_pp\_gg.wav
- 2do mando de voz: DERECHA → derecha\_pp\_gg.wav
- 3er mando de voz: DELANTE → delante\_pp\_gg.wav
- 4to mando de voz: ABAJO → abajo\_pp\_gg.wav
- 5to mando de voz: ABRIENDO → abriendo\_pp\_gg.wav
- 6to mando de voz: CERRANDO → cerrando\_pp\_gg.wav
- 7mo mando de voz: APAGAR → apagar\_pp\_gg.wav
- 8vo mando de voz: ENCENDER → encender\_pp\_gg.wav
- 9no mando de voz: PARADA → parada\_pp\_gg.wav
- 10mo mando de voz: AVANZA → avanza\_pp\_gg.wav

1ra persona	10 mandos de voz	6 grabaciones por mando de voz	60 archivos de audio
2da persona	10 mandos de voz	6 grabaciones por mando de voz	60 archivos de audio
3ra persona	10 mandos de voz	6 grabaciones por mando de voz	60 archivos de audio
...	...	...	...
10ma persona	10 mandos de voz	6 grabaciones por mando de voz	60 archivos de audio
<b>TOTAL DE ARCHIVOS O GRABACIONES</b>			600 archivos de audio

Tabla 01. Base de datos conformada por mandos de voz.

Como puede observarse, se eligieron 10 mandos de voz todos ellos conformados por palabras de tres sílabas, lo cual permitió la diferenciación entre ellos y la facilidad de reconocimiento por la técnica matemática empleada. Asimismo, por comodidad de lectura de los archivos de audio de mandos de voz, se utilizó una denominación particular.

A continuación, se explica con un ejemplo.

“avanza\_pp\_gg.wav”

<sup>3</sup> Formato de archivo de audio digital de Microsoft

Donde:

- avanza : es el nombre correspondiente al mando de voz utilizado.
- pp: conformado por dos caracteres que representan al número de persona que participó en la grabación del mando de voz.
- gg: conformado por dos caracteres que representan al grupo de grabación del mando de voz realizada por la persona “pp”.

Seguidamente se muestra el código de programa empleado para representar gráficamente 04 señales de audio digital de la segunda persona, y correspondientes a la primera grabación de los mandos de voz: IZQUIERDA, DERECHA, DELANTE y ABAJO. Ver la figura 03.

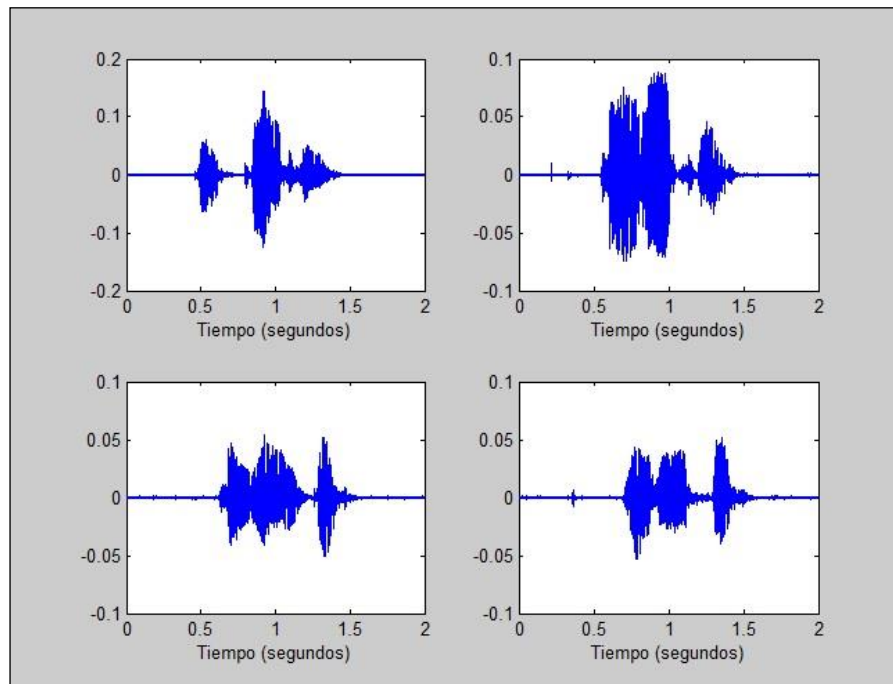


Figura 03. Representación temporal de 04 mandos de voz en función al tiempo.

```
>> Fs = 8000;  
>> t = linspace( 0 , 2 , 2*Fs );  
>> y1 = wavread('izquierda_02_01.wav');  
>> y2 = wavread('derecha_02_01.wav');  
>> y3 = wavread('delante_02_01.wav');  
>> y4 = wavread('abajo_02_01.wav');  
>> figure(1)  
>> subplot(2,2,1), plot( t , y1 ), xlabel('Tiempo (segundos)')
```

```

>> subplot(2,2,2), plot( t , y2 ), xlabel('Tiempo (segundos)')
>> subplot(2,2,3), plot( t , y3 ), xlabel('Tiempo (segundos)')
>> subplot(2,2,4), plot( t , y4 ), xlabel('Tiempo (segundos)')

```

## Técnicas de procesamiento de datos

Por tratarse de un tipo de investigación aplicativo y tecnológico, no se utilizaron técnicas estadísticas, sino únicamente matemáticas. Por lo cual, el procesamiento se dividió en diversas etapas comenzando con el desarrollo de los algoritmos de tratamiento digital de mandos de voz, siguiendo con la obtención de los coeficientes predictivos lineales (LPC), continuando con el entrenamiento de la Red Neuronal Adaptiva en el software Matlab, y finalizando con el almacenamiento de las técnicas matemáticas y los pesos sinápticos de la Red Neuronal, en el hardware embebido NI myRIO, con el fin de alcanzar el reconocimiento de los 10 mandos de voz. Seguidamente, se describen las etapas mencionadas, partiendo de la captura y digitalización de los mandos de voz. Ver el diagrama de bloques de la figura 04, como referencia.

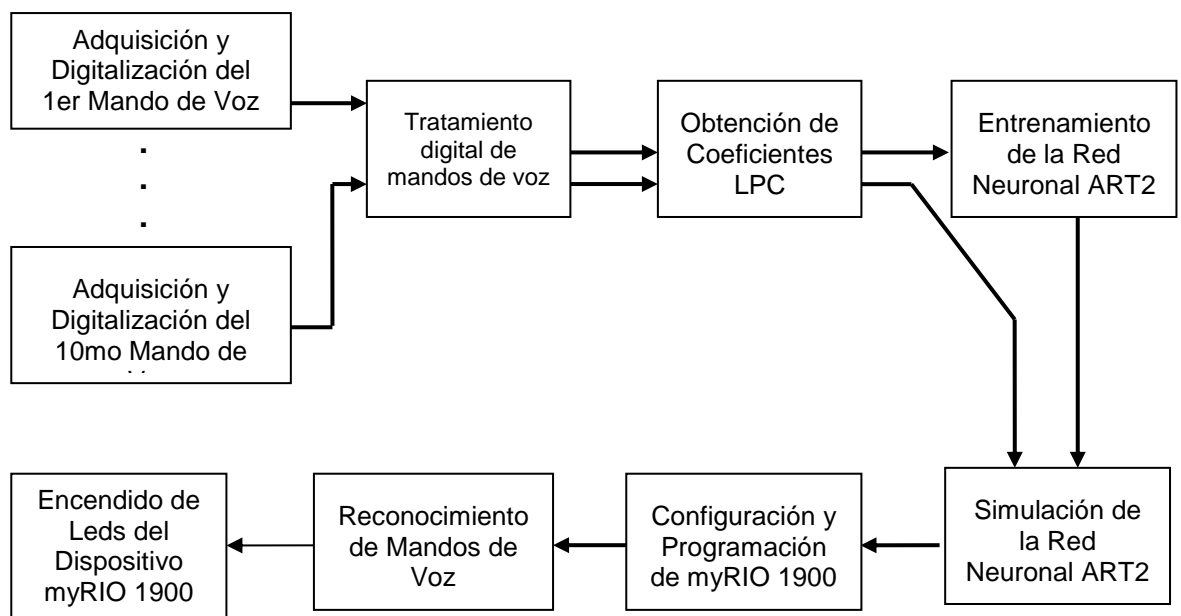


Figura 04. Diagrama de bloques de las etapas de la técnica de procesamiento de datos.

### a) Tratamiento digital de mandos de voz

Según lo indicado en la sección de Captura, Digitalización y Almacenamiento de Mandos de Voz, se utilizaron 2 segundos de grabación por cada mando de voz. Por lo tanto, si la frecuencia de muestreo utilizada fue igual a 8000 Hz, o también

entendido como 8000 muestras por segundo, entonces el mando de voz digitalizado ascendió a un total de 16000 muestras que fueron representadas a través de una variable del tipo vector en el software Matlab.

Por lo tanto, a continuación, se muestran cada uno de los pasos realizados para sobre el vector de 16000 muestras.

Primero, se procedió a recortar la señal digital de 16000 muestras. Pues, a pesar que dichas señales están constituidas por 3 sílabas, se presentaron partes de silencios al inicio y al final de la grabación. Para realizar el recorte de la señal de audio digital, se procedió a substraer cada vector representativo de la señal digital de mando de voz por su valor promedio. Luego, se normalizó dividiéndolo por el elemento máximo del nuevo vector obtenido. Este proceso fue realizado en cada segmento de 10 mili segundos, lo que equivale a 80 muestras, obteniendo el módulo seguido del promedio ponderado para después compararlo con un umbral, tanto en la búsqueda de izquierda a derecha, como de derecha a izquierda. A continuación, se representa el diagrama de flujo correspondiente a este primer procesamiento. Ver la figura 05.

Por otro lado, en la figura 06, se representa dos de los mandos de voz de la figura 03 complementados con el proceso de recorte del inicio y fin.

```
>> y1r = recorta( y1 );  
>> y2r = recorta( y2 );  
>> subplot(2,2,1); plot( t, y1 ), title('Señal IZQUIERDA ORIGINAL')  
>> subplot(2,2,2); plot( t, y2 ), title('Señal DERECHA ORIGINAL')  
>> subplot(2,2,3); plot( y1r ), title('Señal IZQUIERDA Recortada')  
>> subplot(2,2,4); plot( y2r ), title('Señal DERECHA Recortada')
```

Segundo, se continuó con el uso de un filtro de pre-énfasis, con la finalidad de hacer menos sensible la causa de los efectos de cuantización por longitud finita de palabra. Para esto, se empleó el filtro no recursivo de primer orden pasa alto mostrado en la ecuación 06.

$$H(z) = 1 - 0.95z^{-1} \quad (06)$$

Como tercer paso, se aplicó un filtro pasa banda del tipo recursivo, para limitar el ancho de banda en frecuencia de los mandos de voz digitalizados, evitando así tonos de frecuencia ajenos al de la de voz humana. En la figura 07 se muestra la



representación en frecuencia para dicho filtro con una banda limitada entre 500 Hz. y 3900 Hz. Y, en la figura 08 se representa el resultado de la operación de filtrado sobre las señales recortadas de la figura 03.

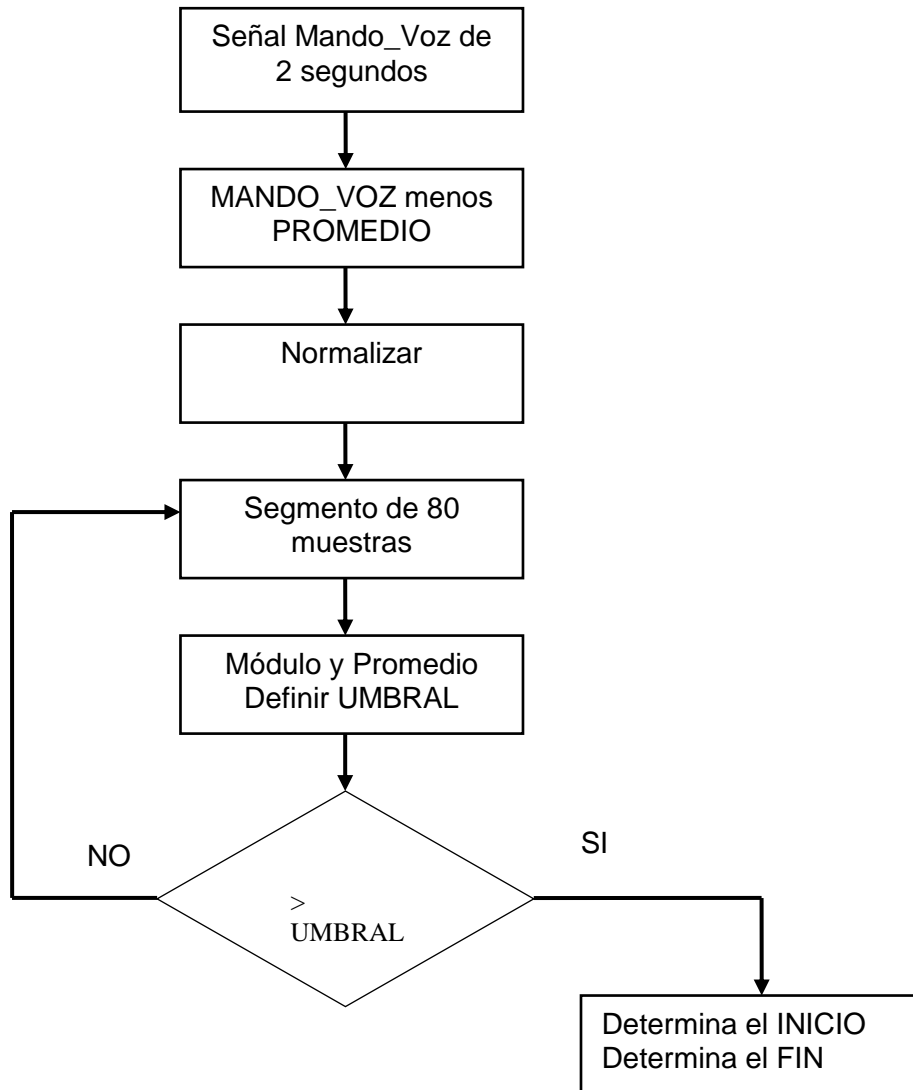


Figura 05. Diagrama de flujo del proceso de recorte de la señal digital.

```

>> Fs = 8000; Orden = 4; Ripple = 40;
>> Frecuencia_corte = [ 500 3900 ];
>> [ B , A ] = cheby2( Orden ,Ripple, Frecuencia_corte/(Fs/2) );
>> figure(1), freqz( B , A , Fs , 'whole' , Fs )
>> y1rf = filter( B , A , y1r );
>> figure(2), subplot(2,1,1), plot( y1r ), xlabel('Tiempo (segundos)')
>> figure(2), subplot(2,1,2), plot( y1rf ), xlabel('Tiempo (segundos)')
  
```

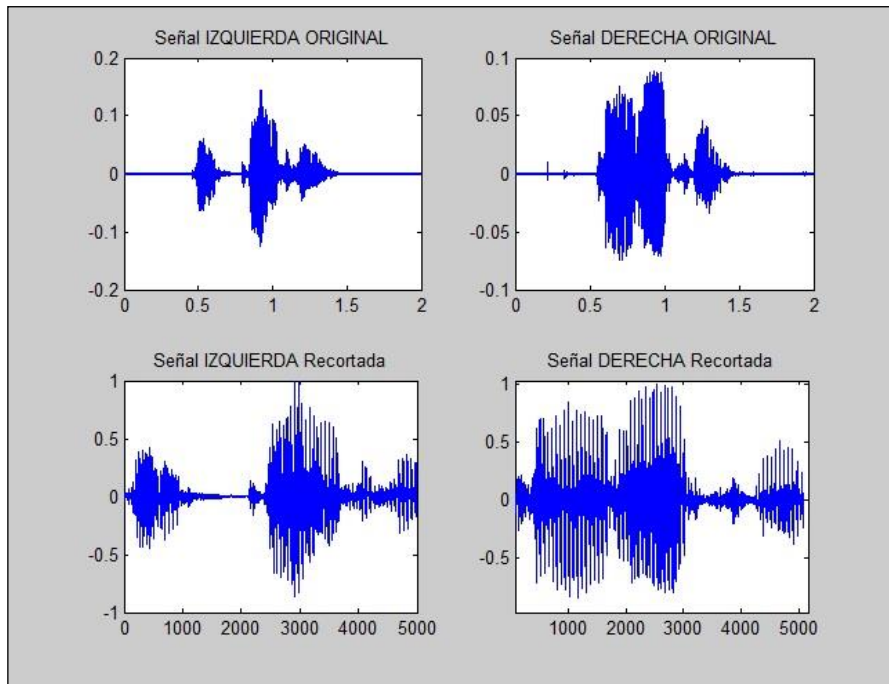


Figura 06. Representación temporal de 02 mandos de voz y sus respectivos recortes.

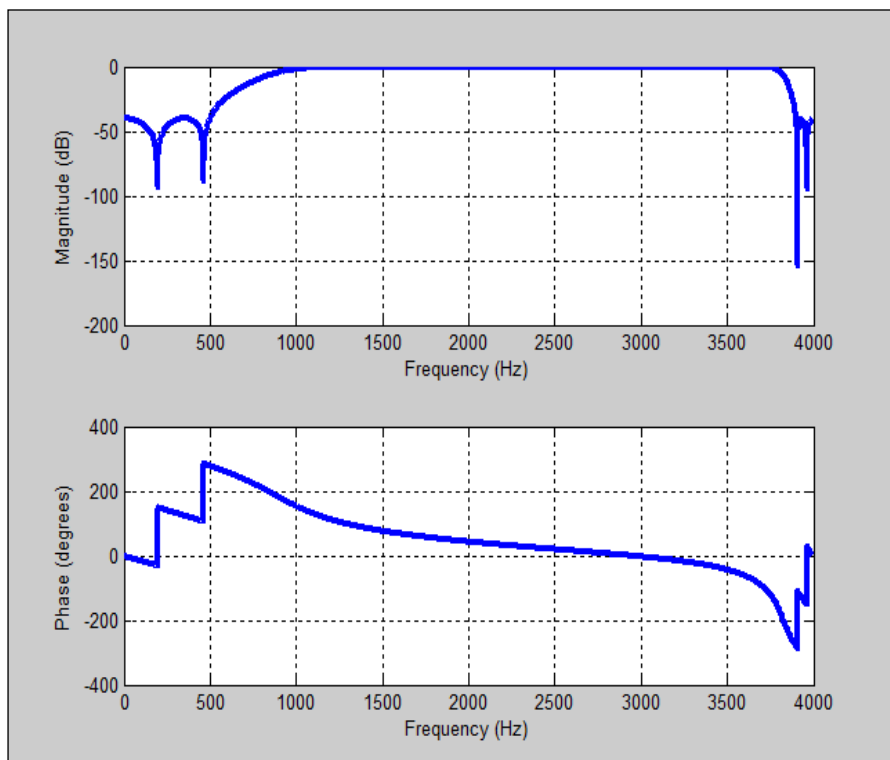


Figura 07. Filtro pasa banda Recursivo del tipo Chebyshev.

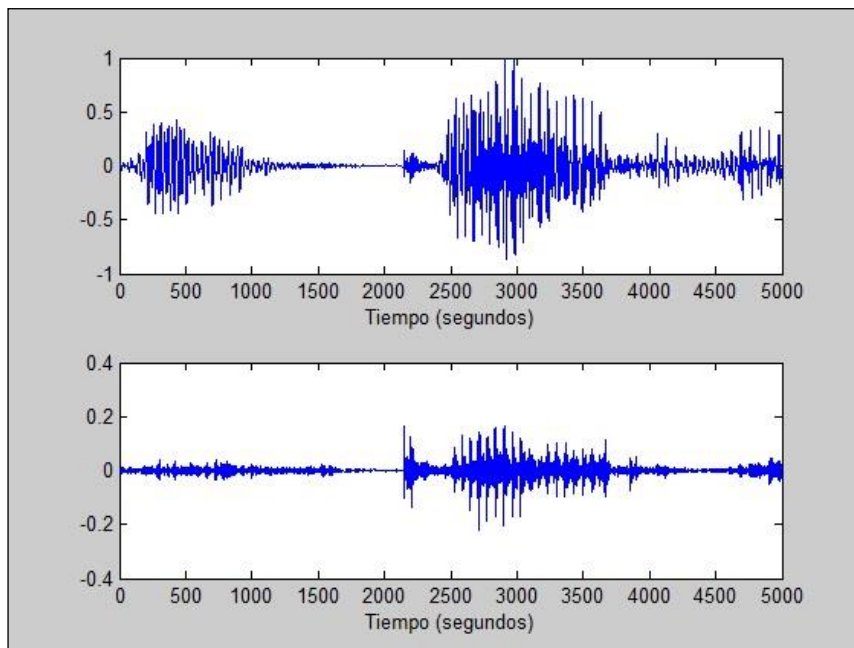


Figura 08. Mando de voz filtrado con el filtro Pasa Banda.

*b) Obtención de Coeficientes LPC*

La obtención de los Coeficientes de Predicción Lineal (LPC), fue el siguiente paso al tratamiento de los mandos de voz. Por lo cual, teniendo presente que las grabaciones de 2 segundos no siempre fueron recortadas a un tamaño uniforme de muestras, aun perteneciendo a una misma persona y grabación de mando de voz, se procedió a obtener un número fijo de coeficientes LPC para cada una de las señales recortadas.

De esta manera, después de realizar varias pruebas de efectividad en cuanto al número apropiado de coeficientes LPC, se determinó el valor de 84. Para ello, se recurrió al toolbox Signal Processing del Matlab y se utilizó la función denominada LPC sobre cada una de las señales digitales de mando de voz filtrada, recortada y ventaneada. El procedimiento de ventaneo tiene como significado, multiplicar el mando de voz pre procesado, por una ventana de hamming con el mismo tamaño de muestras. A continuación, se muestra el código de programa utilizado.

```
>> coef = 85;
>> h = hamming( coef );
>> y1rfv = y1rf .* h';
>> coef_LPC = lpc( y1rfv , coef );
>> coef_LPC_01 = coef_LPC( 2:end );
```

En vista que el conjunto de coeficientes LPC de cada mando de voz presentó oscilaciones continuas, se optó por aplicar un filtro pasa bajo de orden 30 y con frecuencia de corte igual a 2 KHz. a cada grupo de mandos de voz con la finalidad de suprimir tales oscilaciones. Por lo cual, antes de filtrar, se normalizó el conjunto de coeficientes LPC. Luego, se incrementó el tamaño del vector de coeficientes LPC con 30 muestras al inicio, y en forma invertida para así eliminar el transitorio resultante después de aplicar la operación de filtrado con un filtro no recursivo, de orden relativamente reducido. A continuación, se representa el código de programa empleado en el Matlab para la normalización y filtrado de los coeficientes LPC, así como una representación gráfica del resultado de la operación de filtrado sobre los coeficientes LPC. Ver la figura 09.

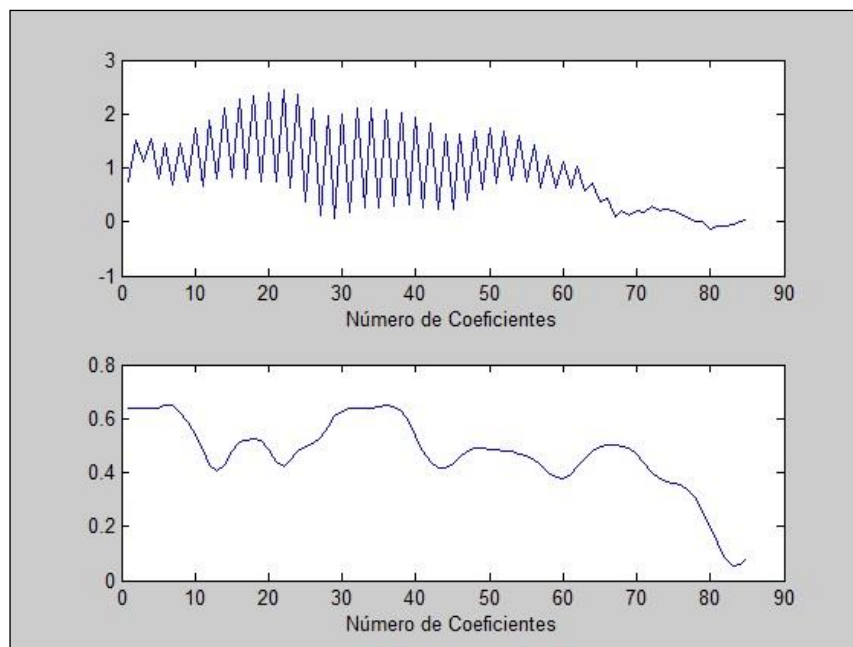


Figura 09. Filtrado pasa bajo de los coeficientes LPC de un mando de voz.

```
>> coef_LPC_01_N = coef_LPC_01 / max( coef_LPC_01 );
>> a = [ flipud( coef_LPC_01_N(1:30)) coef_LPC_01_N ];
>> hpb = fir1( 30 , 2000 / 4000 );
>> b = filter( hpb , 1, a );
>> coef_LPC_01_N_F = b( 31:end );
>> figure(2),
>> subplot(2,1,1 ),plot( coef_LPC_01 ), xlabel('Número de Coeficientes')
>> subplot(2,1,2 ),plot( coef_LPC_01_N_F ), xlabel('Número de Coeficientes')
```

c) Entrenamiento de la Red Neuronal ART2

La Red Neuronal ART2<sup>4</sup> es un tipo de estructura de red que utiliza el principio de hacer resonar los vectores de entrada con las clases reconocidas por la propia red neuronal. Es decir, si entra en resonancia con una clase la red considera que pertenece a ella y se realiza una adaptación que incluye algunas características de los nuevos datos a la categoría existente. Y, por el contrario, cuando no resuena, la red se encarga de crear una nueva clase con el vector de entrada como ejemplo de la misma. Todo esto lo señala Jorge Rodríguez en [04, pág. 23].

Consecutivamente, se muestra un diagrama de flujo del funcionamiento simplificado de la Red Neuronal ART2, el cual fue implementado a través de un código de programa del software Matlab. Ver la figura 10.

Para la realización de la etapa de entrenamiento de la Red Neuronal ART2, se utilizó el 80% del total de 600 vectores de audio almacenados inicialmente. Y el 20% para la etapa de prueba. Esto quiere decir, que se contó con dos grupos de vectores, uno para la etapa de entrenamiento y un segundo para la de prueba. De esta manera, haciendo uso de 480 vectores, el proceso de entrenamiento hizo que se creara una matriz de pesos sinápticos de tamaño 46 x 84, esto significa que se crearon 46 clases para reconocer los 10 tipos de mandos de voz, donde algunas de las clases pertenecieron a un mismo mando de voz. Por lo tanto, se procedió a almacenar dicha matriz de pesos en un archivo de extensión TXT, para posteriormente ser leído a través del software Labview.

El tamaño de la matriz de pesos sinápticos resultante dependió mucho del radio elegido en la superficie de la hiper-esfera. Es decir, se tomó un valor de radio igual a 0.15 y no mayor a este, con la finalidad de evitar crear sub clases mayores que podían encerrar a un conjunto de mandos de voz diferentes. Asimismo, la constante de entrenamiento elegida fue de un valor pequeño e igual a 0.05, para evitar inestabilidad de la red neuronal durante el proceso de entrenamiento. De esta manera, se hicieron tres pruebas de valores de radio diferentes, llegando a la conclusión de tomar una cantidad igual a 0.15. Es decir, fue una elección a partir de un procedimiento de ensayo y error.

---

<sup>4</sup> Teoría de la Resonancia Adaptativa Tipo 2

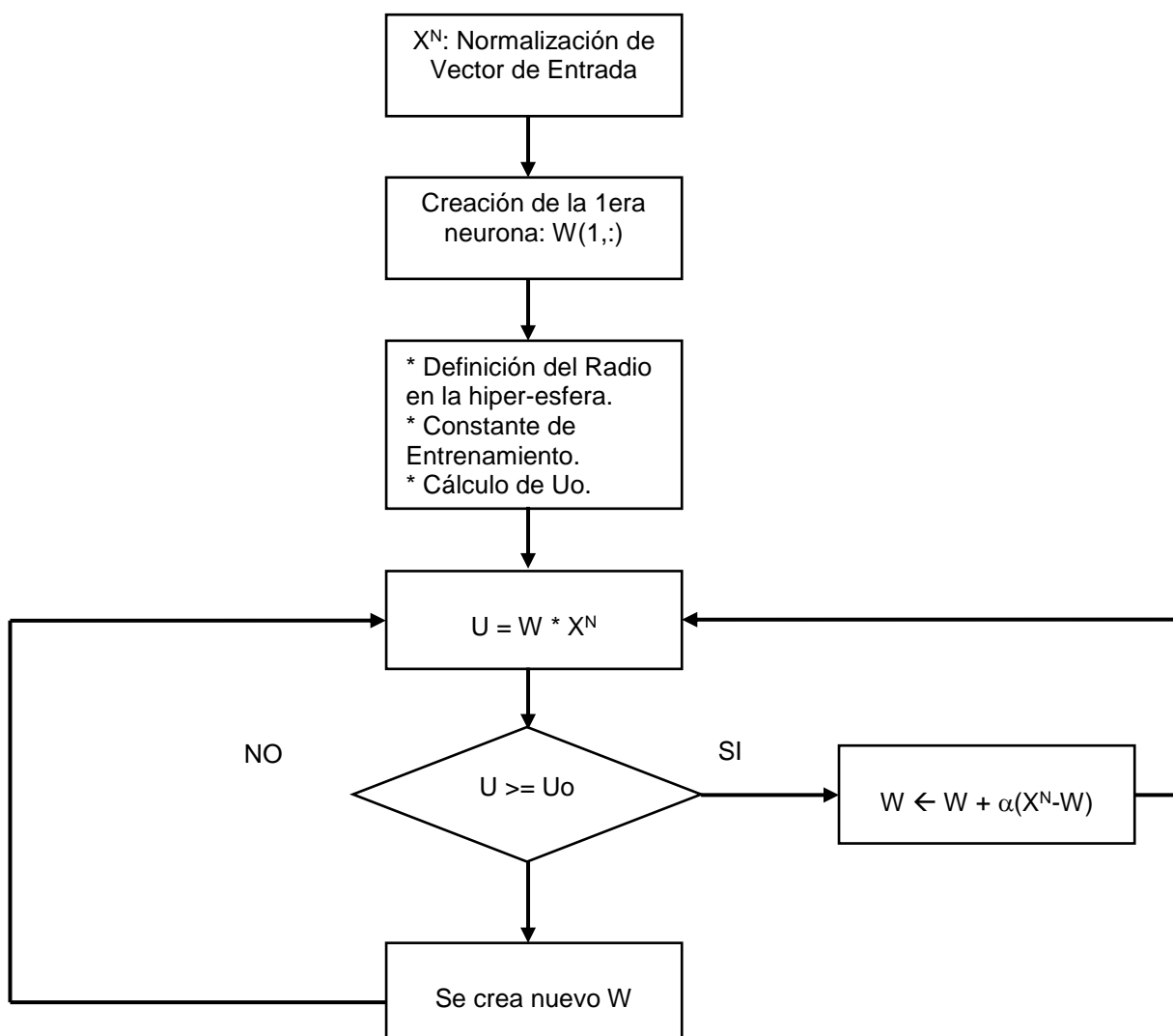


Figura 10. Diagrama de flujo de la Red Neuronal Artificial ART2.

Posteriormente, para mantener la precisión de cada uno de los elementos de la matriz de pesos final, se optó por almacenarlos con un total de hasta 8 dígitos decimales en formato de punto fijo.

```

>> fid=fopen('pesos.txt', 'w');
>> fprintf(fid, '%4.8f \t\n', w);
>> fclose(fid);
  
```

d) Simulación de la Red Neuronal ART2

Una vez entrenada la Red Neuronal ART2, se procedió a simularla utilizando los 600 vectores de entrada de mandos de voz por vez. Es decir, primero con el grupo de entrenamiento en donde los resultados fueron mejores, y luego con el grupo

de prueba. A continuación, se muestra el código de programa utilizado para el proceso de simulación con la red neuronal.

```
>> load Pesos  
  
>> clases = zeros( 1 , 600 );  
  
>> for i=1:600  
  
    d = W * coef_LPC_N_F( : , i );  
  
    [ x , y ] = max( d );  
  
    clases( i ) = y;  
  
end
```

#### e) Configuración y Programación de myRIO 1900

Para la configuración del dispositivo de hardware embebido myRIO 1900, se utilizó el software LabVIEW y las librerías correspondientes al hardware en mención. Por lo cual, consistió en conectar dicho hardware a la toma de 220 VAC y a la laptop por medio del cable USB. Esto permitió el reconocimiento automático del dispositivo con un número de serie y dirección IP, propio. Luego, una vez realizada la configuración automática, se procedió a hacer click en la opción Launch the Getting Started Wizard de la primera ventana, obteniendo una nueva presentación de ventana. Ver la figura 11.

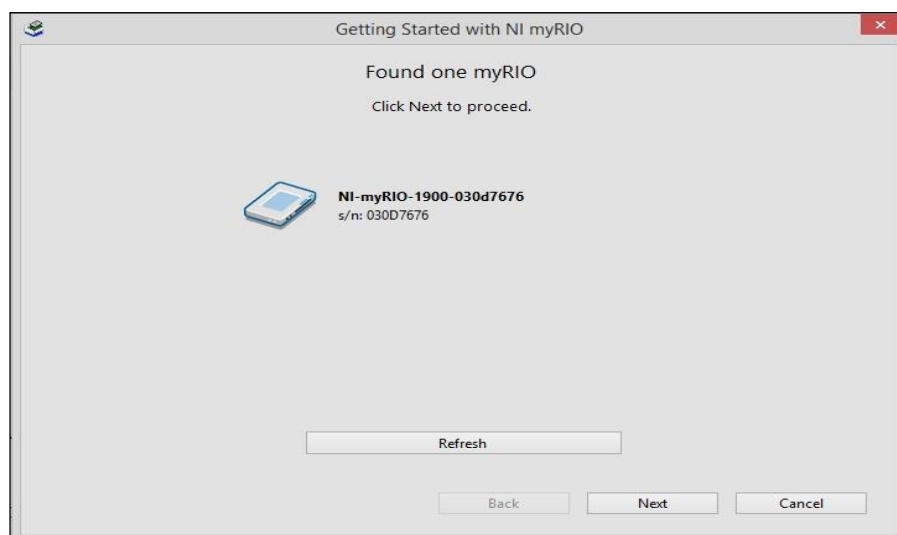


Figura 11. Ventana de Reconocimiento de la Configuración del hardware embebido myRIO.

Posteriormente, al hacer click en el botón NEXT, se llegar a una nueva ventana que sirve de prueba para verificar el funcionamiento del acelerómetro incluido, el encendido de un led a través de un botón, y el encendido directo de los 4 leds. Todo esto puede apreciarse en la figura 12. En este proyecto de investigación, solamente se utilizó la opción de encender los 4 leds del propio hardware embebido, pero a partir del reconocimiento de cada uno de los 10 mandos de voz.

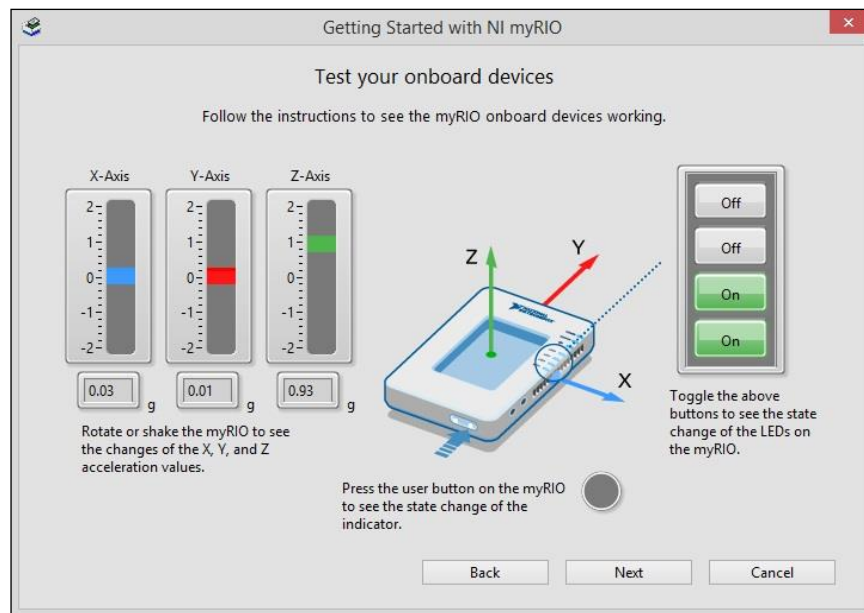


Figura 12. Ventana de prueba para el hardware embebido myRIO.

Realizada la configuración del myRIO, se continuó con la implementación de un código de programa gráfico para la adquisición de la señal de voz en tiempo real, a partir de un micrófono conectado directamente al puerto de audio de la laptop. Esto puede observarse en la figura 13, la cual representa la Ventana Diagrama de Bloques del Labview.

Luego, se procedió a trasladar todo el código de programa desarrollado en el software Matlab al software Labview. Para esto, se utilizaron librerías propias del Labview que permitieron que interactúe en tiempo real con el Matlab, tal es el caso del MATHSCRIPT NODE. Esta opción de programación solo es posible de utilizarlo si tanto el Matlab y el Labview se encuentran instalados correctamente en la misma PC o Laptop. Asimismo, dicha librería sólo necesitó de una entrada y una salida. Por un lado, la entrada es la señal de voz digitalizada en formato de un vector, con 16000 muestras. Mientras que la señal de salida es el filtrado pasa bajo del conjunto de 84 coeficientes LPC listos para ser ingresados a la red



neuronal artificial adaptativa, previamente entrenada en el software Matlab. Ver la figura 14.

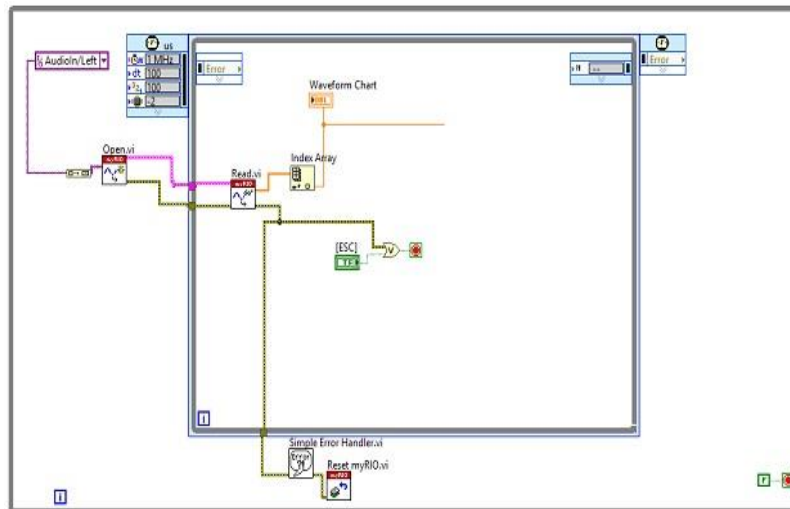


Figura 13. Programación de la lectura del audio en tiempo real.

```

1  %QUITAR SILENCIO
2  %%%%%%%%%%%
3  s = s - mean(s);
4  s = s/max(s);
5  s = s';
6  len = length(s); % length del vector
7  THRES = 0.1;
8  for i = 1:80:len-80 % cada 10ms
9      seg = abs( s(i:i+79,1) ); % segment
10     e = sum(seg)/80; % promedio de c
11     if( e > THRES ) % si el promedio er
12         inicio = i;break;
13     end;
14 end;
15 inicio = inicio - 2*80;
16 s1 = flipud(s);
17 for i = 1:80:len-80 % cada 10ms
18     seg = abs(s1(i:i+79,1)); % segment
19     e = sum(seg)/80; % promedio de c
20     if( e > THRES ) % si el promedio er
21         fin = i; break;
22     end;
23 end;
24 fin = fin - 2*80;
25 v1 = s(inicio:len-fin);

```

Figura 14. Programación del procesamiento de la señal de audio en el MathScript Node del Labview.

Por otro lado, el MathScript Node de la Red Neuronal Adaptativa recibió como primera entrada el filtrado pasa bajo de los coeficientes LPC, previamente pre procesados, y como segunda entrada el nombre del archivo donde se almacenaron los pesos sinápticos de la red neuronal. Y otorgó como salida un

valor correspondiente a la clase de mando de voz reconocida. Asimismo, como parte de esta programación, se incluyó la lectura del archivo PESOS.TXT que contiene a la matriz de pesos sinápticos entrenados previamente en el software Matlab. Ver la figura 15 para identificar la programación realizada en Labview, correspondiente a la etapa de reconocimiento por parte de la red neuronal.

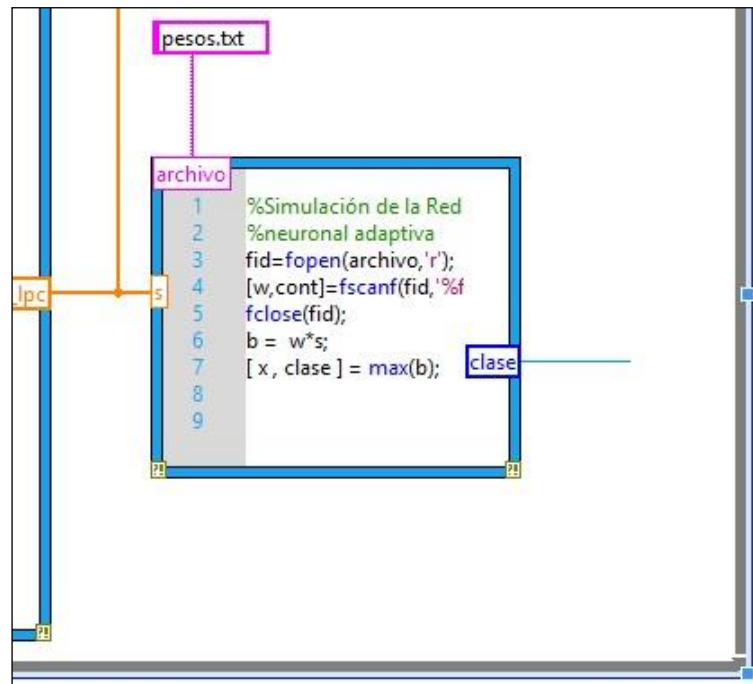


Figura 15. Programación del reconocimiento de los mandos de voz por medio de la Red Neuronal ART2, implementada en el MathScript Node del Labview.

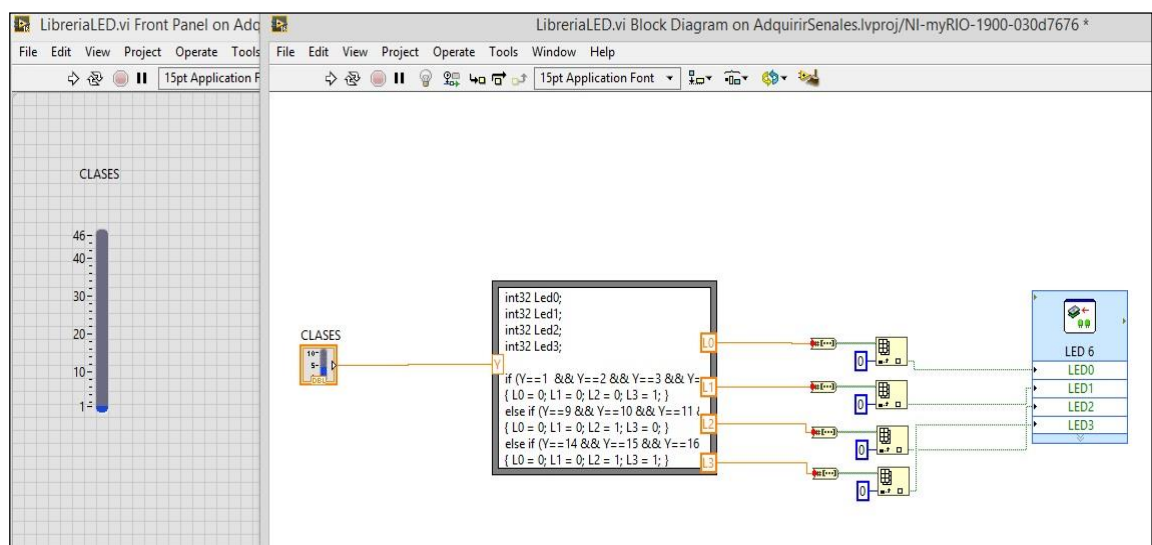


Figura 16. Librería para encendido/apagado de leds del myRIO 1900.

También se puede apreciar, de la figura 16, una combinación de encendido y apagado de leds, con una conversión previa de un formato numérico a booleano según al conjunto de clase a la cual pertenece un mando de voz específico.

La lista de mandos de voz en función al encendido y apagado de los leds, puede observarse en la tabla 02, lo que equivale a 10 combinaciones utilizando los 4 leds del myRIO 1900.

	<b>MANDOS DE VOZ</b>	<b>LED 01</b>	<b>LED 02</b>	<b>LED 03</b>	<b>LED 04</b>
1	IZQUIERDA	OFF	OFF	OFF	ON
2	DERECHA	OFF	OFF	ON	OFF
3	DELANTE	OFF	OFF	ON	ON
4	ABAJO	OFF	ON	OFF	OFF
5	ABRIENDO	OFF	ON	OFF	ON
6	CERRANDO	OFF	ON	ON	OFF
7	APAGAR	OFF	ON	ON	ON
8	ENCENDER	ON	OFF	OFF	OFF
9	PARADA	ON	OFF	OFF	ON
10	AVANZA	ON	OFF	ON	OFF

Tabla 02. Situación de encendido de 4 leds del hardware embebido myRIO 1900.

Además, es preciso añadir que el reconocimiento no fue totalmente efectivo debido a algunas razones que serán explicadas en la sección de resultados. Por lo cual, hubo clases compartidas por mandos de voz diferentes, tal es el caso del mando de voz DERECHA y DELANTE.

## **RESULTADOS**

Para mostrar los resultados de este proyecto, se hicieron pruebas con la matriz de pesos sinápticos obtenidos después del entrenamiento de la red neuronal, y tomando como entrada las 600 grabaciones capturadas inicialmente con el software Matlab. Por lo cual, en la tabla 03 se muestra un resumen del grado de reconocimiento que se tuvo al identificar los 10 mandos de voz tomados como referencia.

<b>MANDOS DE VOZ</b>	<b>RECONOCE</b>	<b>NO RECONOCE</b>
IZQUIERDA	53 señales	07 señales
DERECHA	52 señales	08 señales
DELANTE	52 señales	08 señales
ABAJO	55 señales	05 señales
ABRIENDO	54 señales	06 señales
CERRANDO	54 señales	06 señales
APAGAR	53 señales	07 señales
ENCENDER	52 señales	08 señales
PARADA	50 señales	10 señales
AVANZA	53 señales	07 señales
<b>SUB TOTAL</b>	<b>528 señales</b>	<b>72 señales</b>
<b>TOTAL</b>	600 señales	

Tabla 03. Resumen del grado de reconocimiento con los mandos de voz capturados y almacenados.

Del resultado anterior, se obtiene el porcentaje de Reconocimiento y No Reconocimiento del total de señales de audio utilizadas.

$$\%No\_Reconocimiento = \frac{72 \cdot 100}{600} = 12 \%$$

$$\%Reconocimiento = \frac{528 \cdot 100}{600} = 88 \%$$

También, se procedió a realizar pruebas reales con un nuevo número de mandos de voz, dictados en tiempo real a través del micrófono conectado a la Laptop. Para esto, se eligió a una de las personas participantes para que pronunciara los 10 mandos de voz, bajo la misma característica de los mandos de voz inicialmente

capturados. Por lo cual, los resultados alcanzados, se muestran a continuación en la tabla 04.

<b>MANDOS DE VOZ</b>	<b>RECONOCE</b>
IZQUIERDA	SI
DERECHA	SI
DELANTE	SI
ABAJO	SI
ABRIENDO	NO
CERRANDO	NO
APAGAR	SI
ENCENDER	SI
PARADA	SI
AVANZA	NO
<b>RECONOCE</b>	<b>07 MANDOS</b>
<b>NO RECONOCE</b>	<b>03 MANDOS</b>
<b>TOTAL</b>	<b>10 MANDOS</b>

Tabla 04. Resumen del grado de reconocimiento con los mandos de voz capturados en tiempo real.

De esta manera, el proceso de reconocimiento de los 10 mandos de voz tuvo como principal objetivo capturar efectivamente la voz humana a través del micrófono omnidireccional, para luego ser procesado por el hardware embebido y mostrar la combinación de encendido/apagado de leds en forma correcta. A continuación, en la figura 17 se muestra una fotografía de la manera como se conectó el hardware

myRIO a la Laptop y a su vez el encendido y apagado de los leds, según el mando de voz reconocido. Para este caso: cerrando.

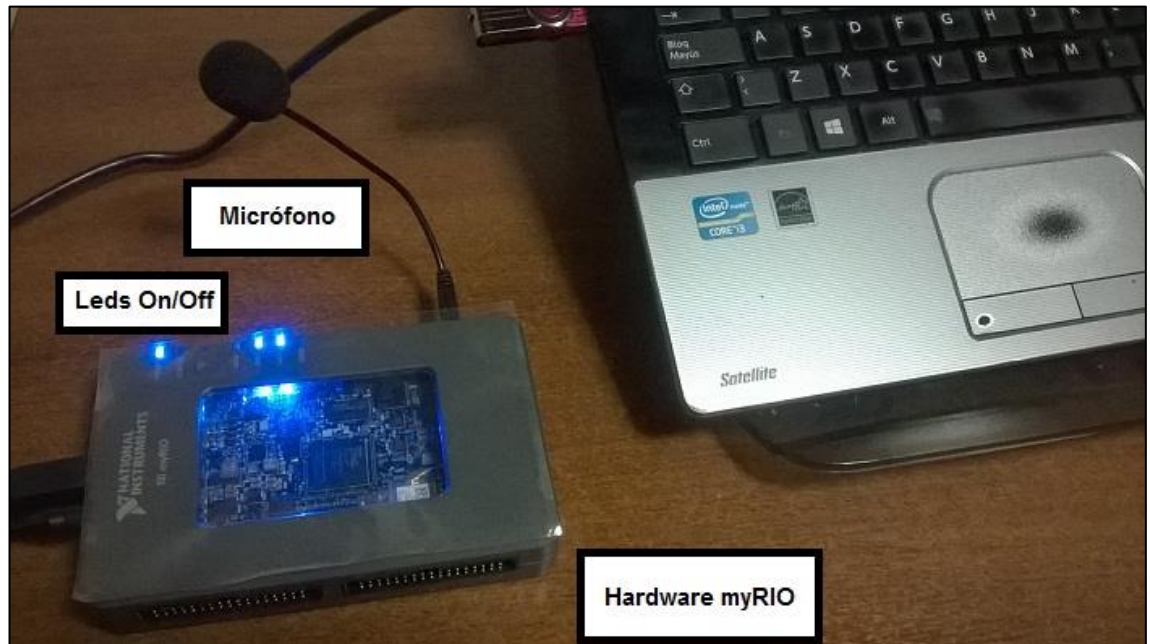


Figura 17. Fotografía del hardware embebido después de reconocer un mando de voz ingresado por el micrófono.

## DISCUSIÓN

Para mejorar más aún esta investigación, es necesario contar con un micrófono de mejores características técnicas para capturar la señal de voz, y así contar con una información con menos ruido y con el uso de técnicas matemáticas de procesamiento digital menos complejas, para lograr por completo la etapa de pre procesamiento.

Como una posterior investigación, se podría optar por utilizar un procesamiento frecuencial adicional para extraer mejor las características de la señal de audio capturada. Como, por ejemplo, los Coeficientes Cepstrales de frecuencia MEL. O como también podría ser necesario utilizar los Modelos Ocultos de Markov para obtener una mejor extracción de las principales características de una señal digital de voz. De esta manera, el entrenamiento de la red neuronal mejoraría al contar con un total de padrones de entrada menos similares entre ellos. Por lo cual, mejoraría la definición de las clases y muy probable simplificaría el número de neuronas en la red neuronal.

Además, para convertir este proyecto en una aplicación más práctica y útil para personas discapacitadas, se haría necesario invertir en una etapa de diseño de hardware para manipular señales de voltaje o corriente en continua o alterna. Para ello sería necesario circuitos electrónicos para potencia, aislamiento y amplificación. Debido a que la tarjeta myRIO cuenta con puertos de salida analógica, pero para voltaje continuo y a una baja corriente. Por tal razón, se puede extender el desarrollo de este proyecto al uso de otro tipo de hardware denominado CompactRIO, el cual si se acomoda para situaciones reales e industriales.

## CONCLUSIONES

Las conclusiones correspondientes al desarrollo de esta investigación, son:

- a) Se logró desarrollar las técnicas de procesamiento digital de señales para el reconocimiento de los mandos de voz. Inicialmente, las técnicas matemáticas fueron implementadas algorítmicamente en el software Matlab para verificar la efectividad de las mismas. Posteriormente, se trasladaron hacia el software Labview, haciendo uso de la librería MathScript Node que facilitó tal migración conservando el lenguaje de código propio del Matlab. No obstante, se hace necesario que tanto el software Matlab como el Labview, se encuentren correctamente instalados en la misma PC o Laptop utilizada. Entre las técnicas matemáticas utilizadas se tuvo al del recorte de los segmentos no sonoros basada en una comparación de energía frente a un umbral, por cada segmento de 10 mili segundos de duración. De igual manera, se aplicaron operaciones de filtrado digital de pre-énfasis de orden igual a 1, y un pasa banda de orden igual a 4. Y finalmente, se determinaron los coeficientes LPC, los cuales fueron filtrados por un pasa bajo. Todo ello fue señalado en la sección de Metodología de este informe de proyecto. En conclusión, se afirma que existe una variedad de técnicas matemáticas para pre procesar y extraer las características principales de una señal de voz, y por lo cual debe comprobarse las distintas combinaciones de técnicas existentes.
- b) El entrenamiento de la red neuronal artificial ART2 fue lograda para un total de 480 vectores, que a su vez correspondieron a 10 personas distintas al momento de capturar los 10 mandos o patrones de voz digital. Este grupo de vectores se denomina conjunto de entrenamiento, y se hace necesario su uso

para verificar luego la efectividad de la red neuronal con otro conjunto de vectores que no participaron. Denominados conjunto de prueba. Cabe señalar, que dicho entrenamiento se basó en implementar algorítmicamente el proceso de actualización de pesos sinápticos de la red neuronal ART2 en el software Matlab, debido a que este proceso de entrenamiento no se requiere que sea en tiempo real. Una vez realizada dicha tarea, se obtuvo como resultado la matriz de pesos sinápticos con 84 columnas debido a la cantidad de coeficientes LPC utilizados. Por lo cual, fue necesario almacenar dicha matriz de pesos en un archivo de texto, para luego ser leído por el software Labview al momento de utilizar la red neuronal en el proceso de validación de los vectores de mandos de voz. Si bien es cierto, que, en este proyecto de investigación, el hardware embebido no almacenó la matriz de pesos; existe la posibilidad de realizar rutinas de programación que, si lo permitan, haciendo que el myRIO 1900 tome el control total del reconocimiento de patrones a través de la red neuronal. En conclusión, la red neuronal adaptativa fue entrenada en off-line en el software Matlab, y luego se almacenó la matriz de pesos sinápticos obtenido en un archivo de texto del propio disco duro de la laptop utilizada, para finalmente leer tal archivo y utilizar los pesos en el reconocimiento de los mandos de voz. Tal proceso se realizó a través de rutinas de programación en el software Labview que fueron insertadas o embebidas en el myRIO 1900.

- c) Con la inserción de las técnicas matemáticas de procesamiento de señal, y la rutina de reconocimiento de patrones por medio de la red neuronal adaptativa en el dispositivo de hardware myRIO 1900 de NI, se permitió simular una acción final según el mando de voz reconocido. Dicha simulación correspondió a la combinación de encendido/apagado de 4 leds, propios del myRIO. No obstante, pudo ser posible utilizar los puertos de salida de este dispositivo para poder conectarlos externamente. Pero, asimismo, hubiera sido necesario contar con una circuitería electrónica adicional para facilitar el tema de potencia o mayor voltaje/corriente, según sea el caso. En conclusión, se logró combinar el encendido o apagado de los 4 leds del dispositivo de hardware myRIO, en función al tipo de mando de voz reconocido en tiempo real, cuando fue capturado a través de un micrófono omnidireccional que también fue conectado al mismo hardware en mención.



## REFERENCIAS BIBLIOGRÁFICAS

Para el desarrollo de este proyecto se recurrió al siguiente material bibliográfico:

- [1] M. Bonifacio y A. Sanz, *Redes Neuronales y Sistemas Borrosos*, 2da edición. México D.F.: Alfaomega y Ra-Ma, 2001.
- [2] National Instruments. (2016) NI myRIO Project Essentials Guide. Visitado en noviembre del 2016. [Online]. Disponible: <http://www.ni.com/academic/myrio/project-guide.pdf>
- [3] National Instruments. (2016) Software de Desarrollo de Sistemas NI LabVIEW. Visitado en diciembre del 2016. [Online]. Disponible: <http://www.ni.com/labview/esa/>
- [4] J. Rodríguez, "Redes Neuronales Artificiales para la Clasificación de Imágenes Satelitales," *AVANCES Investigación e Ingeniería*, No 9, p. 20-31, 2008. [Online]. Disponible: [http://www.unilibre.edu.co/revistaavances/avances\\_9/r9\\_art2.pdf](http://www.unilibre.edu.co/revistaavances/avances_9/r9_art2.pdf)
- [5] A. Alcaim e C. Dos Santos, *Fundamentos do Processamento de Sinais de Voz e Imagem*. Rio de Janeiro, Brasil: Interciência, 2011.
- [6] P. Diniz, E. Barros e S. Lima, *Processamento Digital de Sinais. Projecto e Análise de Sistemas*. Porto Alegre, Brasil: Editorial Bookman, 2004.
- [7] S. Haykin, *Neural Networks A Comprehensive Foundation*. New Jersey, USA: Editorial Prentice Hall, 1994.
- [8] S. Haykin, *Adaptive Filter Theory*, Third Edition. New Jersey, USA: Editorial Prentice Hall, 1996.
- [9] J. Lajara y J. Pelegrí, *LabVIEW Entorno gráfico de programación*, 2da edición. México D.F.: Alfaomega Grupo Editor, 2011.
- [10] P. Porcuna, *Robótica y domótica básica con Arduino*. Bogotá, Colombia: Editorial Ra-Ma, 2016.
- [11] A. Antoniou, *Digital Filters Analysis, Design, and Applications*, Second Edition. Singapore: McGraw-Hill, 1993.
- [12] M. Hagan, H. Demuth, M. Hudson, O. De Jesús, "Neural Networks Design", Second Edition. eBooks. [Online]. Disponible: [hagan.okstate.edu/nnd.html](http://hagan.okstate.edu/nnd.html)
- [13] G. Simón, "Filtro Adaptativo LMS y su aplicación en el reconocimiento de palabras aisladas para el control de un equipo de sonido por medio de la voz", Tesis de Ingeniero Electrónico, Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería, Perú, 2004. [Online]. Disponible: <http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/280/SIMON>

**[\\_BENDEZU\\_GIOVANI\\_FILTRO\\_ADAPTIVO\\_LMS\\_APLICACION.pdf?sequence=1](#)**

- [14] L. Cruz y M. Acevedo, "Reconocimiento de voz usando Redes Neuronales Artificiales Backpropagation y Coeficientes LPC", CICos, p. 89-99, 2008. [Online]. Disponible:  
**<http://campusv.uaem.mx/cicos/cicos2008/imagenes/memorias/6tocicos2008/Articulos/Ponencia%207.pdf>**
- [15] J. Varela y J. Loaiza, "Reconocimiento de palabras aisladas mediante redes neuronales con FPGA", Tesis de Ingeniero Electricista, Universidad Tecnológica de Pereira, Facultad de Ingeniería, Colombia, 2008. [Online]. Disponible:  
**<http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/1712/6213822V293.pdf?sequence=1>**
- [16] M. Yommi, "Sistema automático de reconocimiento de voz para control de acceso", XIII Congreso Argentino de Acústica, CADA E, 2015. [Online]. Disponible:  
**<http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/1712/6213822V293.pdf?sequence=1>**