DISEÑO DE UNA HERRAMIENTA ROBÓTICA EDUCATIVA CONTROLADA POR VOZ UTILIZANDO REDES NEURONALES ARTIFICIALES (RNA)

DIANA CAROLINA CHAUX POLO
CÓD. 2002200614

DUVAN DARÍO QUINTERO CARDOZO
CÓD. 2003102894

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA
2008

DISEÑO DE UNA HERRAMIENTA ROBÓTICA EDUCATIVA CONTROLADA POR VOZ UTILIZANDO REDES NEURONALES ARTIFICIALES (RNA)

DIANA CAROLINA CHAUX POLO
CÓD. 2002200614

DUVAN DARÍO QUINTERO CARDOZO
CÓD. 2003102894

Proyecto de grado presentado para optar
al título de Ingeniero Electrónico

Director:
GERMÁN EDUARDO MARTÍNEZ BARRETO
Ingeniero Electrónico

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA
2008

Nota de aceptación

_____
_____
_____
_____
_____
_____

_____
Firma del presidente del jurado

_____
Firma del primer jurado

_____
Firma del segundo jurado

Neiva, 28 de Noviembre de 2008

DEDICATORIA

A Dios por ser mi guía, llenarme de bendiciones y haberme dado la salud para lograr mis objetivos.

A mis padres por los valores que siempre me han infundido y hacer de mi lo que soy ahora. A mis hermanos por su compañía y apoyo incondicional. A mis sobrinos María Camila, Jorge Andrés y Juan Sebastián por hacerme mi vida más alegre y ser mi motivación para ser mejor persona.

A mi mejor amiga Luisa Fernanda, por comprenderme y estar siempre ahí para apoyarme. A Ivan Mauricio, Oscar Fabián y Carlos Octavio por ser mis compañeros desde primer semestre, brindarme su amistad y hacer de esta carrera una de las mejores experiencias de mi vida.

Al Ing. Bollman de Jesús Blanco, por sus consejos, especialmente por el ultimo que me dió.

Al resto de mi familia, amigos y maestros por formar parte de mi vida, ya que sin su apoyo este proyecto no sería una realidad.

Diana Carolina

¿Que es importante en la vida?, algunas veces creo que la vida vale poco, a veces me pongo a pensar cómo será la vida de otras personas, no desde lo material sino desde la forma de ver el mundo… sé que si yo no existiera aquí, podría tener una familia y amigos en otro lugar y todo quizá sería igual que hoy, eso es triste ya que te hace pensar que en realidad las personas que mas quieres son solo fruto del azar de la vida, ahí es donde creo que la vida vale poco, sin embargo eso es solo lo que creo porque muy en el fondo sé que la vida definitivamente si vale mucho!.

A la idea universal de bien común “Dios”, por darme la fuerza para seguir adelante. A mis padres y mis hermanos, porque sé que sin ellos definitivamente la vida vale poco. A mis amigos, por todo lo que me han enseñado y los buenos momentos que hemos vivido. A Javier H. Rubio, por ser un excelente amigo. A Bollman de Jesús Blanco, por decirme las cosas como son.

Duván Darío

AGRADECIMIENTOS

Al Ing. Germán Eduardo Martínez Barreto, profesor de Ingeniería Electrónica de la Universidad Surcolombiana y director de este proyecto por su interés y colaboración.

Al Ing. José de Jesús Salgado Patrón por su valiosa colaboración.

A los docentes Institución Educativa Mauricio Sánchez García, especialmente a la profesora María Nelly Sánchez de Sterling por recibirnos amablemente en sus instalaciones y colaborarnos en el proceso de toma de las diferentes muestras de voz.

A todas aquellas personas que de una u otra forma colaboraron con la realización de este trabajo.

CONTENIDO

pág.

LISTA DE TABLAS

pág.

LISTA DE FIGURAS

pág.

LISTA DE ANEXOS

pág.

GLOSARIO

ACÚSTICA: es una ciencia interdisciplinaria que estudia el sonido, infrasonido y ultrasonido, es decir ondas mecánicas que se propagan a través de la materia más no del vacío.

BUCLE: es una sentencia que se realiza repetidas veces, hasta que la condición asignada a dicho bucle deje de cumplirse.

CEPSTRUM: es el resultado de calcular la transformada de Fourier (FT, del inglés Fourier Transform) del espectro de la señal estudiada en escala logarítmica (dB). Su nombre se deriva de invertir las cuatro primeras letras de *spectrum*.

COEFICIENTES CEPSTRALES: son los coeficientes resultantes del cálculo del cepstrum.

CORRELACIÓN: es una medida sobre el grado de relación entre dos variables, sin importar cuál es la causa y cuál es el efecto.

DISLALIAS: son alteraciones en la articulación de los fonemas. Son las alteraciones más conocidas y más fáciles de identificar. Se suele definir comúnmente como mala pronunciación.

DSP: es un sistema basado en un procesador o microprocesador que posee un juego de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad. Debido a esto es especialmente útil para el procesado y representación de señales analógicas en tiempo real.

ESPECTRO DE FRECUENCIAS: es una medida de la distribución de amplitudes de cada frecuencia.

FIR: *Finite Impulse Response* o respuesta finita al impulso, se trata de un tipo de filtros digitales en el que, como su nombre indica, si la entrada es una señal impulso, la salida tendrá un número finito de términos no nulos.

FORMANTES: frecuencias naturales de resonancia de todas las cavidades supraglóticas, en el momento de producir un sonido determinado. Ordenados en forma creciente, se denominan: F1, F2, F3, F4, etc. Los tres primeros formantes determinan el timbre particular de cada vocal. Los restantes aportan cualidades secundarias que dan los matices personales y contribuyen a dar la calidad vocal única y distintiva de cada individuo.

FRECUENCIA FUNDAMENTAL: frecuencia más baja de vibración de las cuerdas vocales.

GUI: (Interfaz Gráfica de Usuarios) o *Graphical User Interface* es un componente de una aplicación informática que visualiza el usuario y a través de la cual interactúa con un sistema. Está conformada por ventanas, botones, menús e iconos, entre otros elementos.

LIFTERING: es el proceso de filtrado en el dominio cepstral. Su nombre se deriva de la palabra *filtering*.

LPC: los coeficientes de predicción lineal son los valores de predicción de una señal periódica a partir de valores anteriores.

MATLAB: abreviatura de *MATrix LABoratory*, "laboratorio de matrices" es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).

MFCCs: los coeficientes cepstrales en las frecuencias de Mel (Mel Frequency Cepstral Coefficients) son coeficientes para la representación del habla basados en la percepción auditiva humana. Se derivan de la Transformada de Fourier (FT) o de la Transformada de coseno discreta (DCT).

ORIENTACIÓN ESPACIAL: capacidad de un individuo para percibir y orientarse en el espacio con respecto a puntos de referencia definidos por el entorno o de forma egocéntrica.

PSICOMOTRICIDAD: es la capacidad que tiene el individuo para coordinar movimientos e ideas desde su nacimiento hasta culminar su total desarrollo.

QUEFRENCY: es la variable independiente del cepstrum. Su nombre proviene de la variable *frequency* y tiene carácter temporal.

RUIDO PROPIO: es el ruido que se produce cuando no hay ninguna señal externa que excite el micrófono.

SIG: el grupo de interés especial (SIG) de Bluetooth es una organización privada y sin ánimo de lucro. Los miembros SIG impulsan el desarrollo de la tecnología inalámbrica *Bluetooth*, y la implantan y comercializan en sus productos.

SISTEMAS HOMOMÓRFICOS: son sistemas no lineales que obedecen a un principio de superposición. Son de gran utilidad para el procesado de voz porque ofrecen un método eficaz para separar la estructura fina y los formantes del espectro de la señal de voz.

STREAMING: es un término que se refiere a ver u oír un archivo directamente en una página web sin necesidad de descargarlo antes al ordenador.

UML: el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

VB.NET: Visual Basic .NET es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET.

RESUMEN

La robótica educativa está tomando un abrumante auge en la educación actual. Con el paso de los años se ha convertido en una herramienta pedagógica inmejorable para que el profesor implemente una enseñanza constructiva en niños desde temprana a avanzada edad; conjugando en un mismo proceso aspectos lúdicos y del aprendizaje.

Este proyecto es uno de los desarrollos del grupo de robótica educativa del programa de Ingeniería Electrónica de la Universidad Surcolombiana y tiene como finalidad implementar una herramienta robótica controlada por voz que permite a los niños en etapa preescolar adquirir y reforzar conceptos básicos para la construcción inicial de su desarrollo psicomotor como lo son la orientación espacial y la identificación de colores.

Este documento pretende que el lector conozca las diferentes etapas que se llevaron a cabo para el desarrollo de esta herramienta educativa, para tal fin se ha dividido en 7 capítulos, que se describen a continuación:

En el capítulo 1, se hace una breve introducción a los fundamentos e importancia de la robótica educativa como escenario perfecto para que los niños construyan su conocimiento.

En el capítulo 2 se estudia el proceso de producción de la voz, para posteriormente en el capítulo 3, describir cada una de las etapas del algoritmo de reconocimiento de palabras aisladas así como la función de cada una de ellas para efectuar finalmente el control del robot por medio de la voz.

En el capítulo 4, se estudia el hardware de los Lego Mindstorms NXT (herramienta robótica empleada para desarrollar este proyecto) y su modulo bluetooth que permitió la transmisión de datos del computador a este.

En el capítulo 5, se detalla la interfaz grafica de usuario realizada para finalmente en el capítulo 6 describir las diferentes pruebas que se realizaron para evaluar de las diferentes etapas del sistema de reconocimiento y en consecuencia de la herramienta robótica diseñada.

ABSTRACT

Educational robotics is booming in actual education. Through years it has became an unsurpassable pedagogical tool for teachers because they can implement a constructive teaching in children since early to advanced age it combines in the same process playful and apprenticeship aspects.

This project is a development of electronic engineering's educational robotics group at Surcolombiana University and it has as purpose to implement a robotics tool controlled through voice which allows to children of preschool stage to acquire and strengthen basics concepts for initial structure their psychomotor development such as spatial orientation and identification of colors.

This document pretends that reader knows the different stages carried out for development of this educational tool for that purpose it has been divided in seven chapters which are described below:

The first chapter makes a brief introduction of fundamentals and importance of educational robotics as perfect scenario for children build their own knowledge.

The second chapter studies the process of voice production and then in the third chapter describes each stage of isolated word recognition algorithm as well as the function of these for finally get to control of robot through voice.

The fourth chapter studies the hardware of Lego Mindstorms NXT (robotics tool used for this application) and its bluetooth module which allowed the data transmission from computer to it.

In the fifth chapter, a graphical user interface is detailed for finally in the sixth chapter describes the different tests carried out to evaluate the performance of recognition system and accordingly of robotics tool designed.

INTRODUCCIÓN

Actualmente uno de los campos de la ingeniería con mayor aplicación es la electrónica, involucrando cada vez más ramas del conocimiento, con el objetivo de aportar soluciones a los diferentes problemas y requerimientos de la sociedad actual. Por ejemplo, la electrónica hace uso de la informática al emplear las diferentes técnicas de inteligencia artificial como lo son las Redes Neuronales Artificiales (RNA) las cuales emulan algunas de las características principales del cerebro humano. Sus múltiples aplicaciones las encontramos en las áreas de control de procesos, medicina y economía, principalmente.

El reconocimiento del habla, es una de las aplicaciones de la inteligencia artificial. Básicamente este proceso hace referencia a la extracción de características de la señal de voz. Esta es una tarea fundamentalmente de reconocimiento de patrones. En la década de los 90's las investigaciones sobre el procesamiento de voz se incrementaron, debido a la aparición de herramientas de captura y análisis que permitieron a los investigadores buscar métodos para extraer características de la voz que permitieran discriminar entre palabras diferentes (reconocimiento de voz) o para discriminar entre personas (reconocimiento del locutor).

En la actualidad el reconocimiento de habla es uno de los campos de investigación que día a día incrementa su número de adeptos, dejando como resultado una gran variedad de aplicaciones especialmente aquellas donde se requiere una comunicación hombre – máquina, entre los usos más comunes encontramos: el dictado automático, el control por comandos, telefonistas automáticas, sistemas de asistencia a discapacitados, entre otros.

Otro de los campos que también se ha desarrollado a pasos agigantados en nuestro tiempo es el de la robótica, alcanzando metas y objetivos que años atrás se consideraban casi imposibles de lograr; todo esto gracias a los aportes de la informática y a las metodologías de inteligencia artificial. En 1998, la empresa danesa LEGO lanzó al mercado unos robots con fines educativos los cuales fueron la introducción a lo que hoy se conoce como robótica educativa. Uno de estos robots son los *Lego Mindstorms NXT* que fueron adquiridos en el 2007 por la Universidad Surcolombiana, con lo que se dio inicio al grupo de investigación de robótica educativa.

El presente proyecto empleó dicho robot para realizar una herramienta didáctica que engloba las ramas mencionadas anteriormente y que permite a los niños en la etapa preescolar adquirir nociones de orientación espacial, empleando para ello un robot el cual programaran por medio de su voz para ejecute una tarea específica, de esta manera los niños aprenderán jugando algunos conceptos que le son difíciles de adquirir.

1. ROBÓTICA EDUCATIVA

La robótica educativa es una tecnología relativamente nueva que surgió como tal aproximadamente hacia el año 1960. Desde entonces han transcurrido pocos años y el interés que ha despertado es superior a cualquier previsión que en su nacimiento se pudiera formular, debido a que esta tecnología se ha convertido en el escenario perfecto que permite a los niños, desde temprana a avanzada edad, construir su propio conocimiento llevándolos de la mano hacia el saber científico; permitiéndoles aprender en una forma más práctica, sencilla y movilizadora.

Robótica educativa significa poner al alcance de los alumnos las herramientas necesarias para que desarrollen dispositivos externos a la computadora, controlados por ésta, a través de una interfaz. Seymour Papert, uno de los mentores de esta tecnología, en su libro *MindStorms: Children, Computers, and Powerful Ideas*, describe sus ideas respecto al empleo de las computadoras como impulsoras del aprendizaje. Toma de su compañero Jean Piaget la concepción de niño como "constructor de sus propias estructuras mentales". Es partidario del construccionismo, tesis que sostiene que el niño crea su conocimiento de forma activa y que la educación debe de facilitarle herramientas para realizar actividades que impulsen esta actividad[1].

A fines de la década del 90 comienzan a aparecer en el mercado un conjunto de kits educativos de robótica. El más popular de estos kits es el de la firma Lego y fueron llamados *Mindstorms NXT* derivado del libro desarrollado por Papert. Estos kits rompieron las fronteras de las instituciones educativas y hoy gracias a la Internet, posee una comunidad inmensa de educadores, científicos y desarrolladores de software y hardware para extender sus funcionalidades.

El desarrollo de la robótica educativa en el salón de clases es una estrategia didáctica sustentada en las corrientes cognitivas del aprendizaje, el aprendizaje colaborativo y la resolución de problemas. La introducción de materiales y procesos en la construcción de un robot es una experiencia educativa con alto impacto en el aprendizaje escolar. Algunas de las finalidades alcanzadas a través de esta metodología, incluyen:

- Generación de entornos de aprendizaje basados en la actividad; al concebir, desarrollar e implementar la construcción de robot. Además, fomenta la imaginación, despierta inquietudes y ayuda a comprender mejor el mundo que los rodea, permitiendo el trabajo en equipo facilitando la comunicación, responsabilidad, toma de decisiones.

- Desarrollo e implementación de una nueva cultura tecnológica, favoreciendo la búsqueda de soluciones a problemas en el ámbito del análisis, la planeación y el diseño.

- Conjuga los aspectos lúdicos del juego y el aprendizaje en un solo proceso, no es justo distanciar el juego del aprendizaje como requisito escolar.

[1]SÁNCHEZ COLORADO, Mónica María. Ladrillos programables para robótica educativa Lego vs. Crickets. Eduteka. 2004.

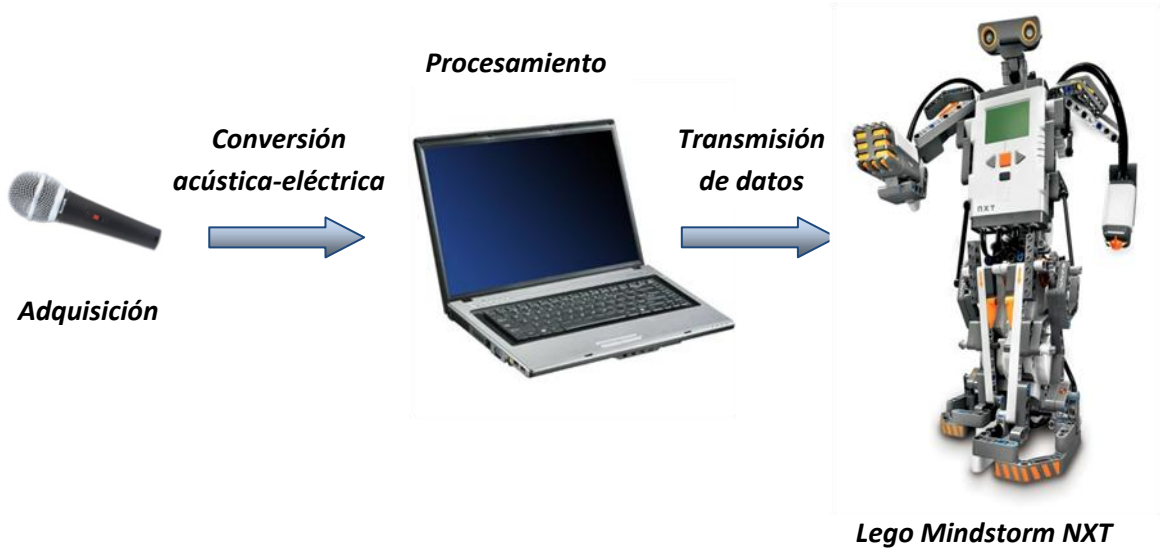
Una de las habilidades que más tardan en desarrollar y manejar correctamente los niños en su proceso de aprendizaje resulta ser el de orientación espacial. Esta habilidad no es única, depende de otros factores, como lo son el proceso de laterización y el desarrollo psicomotor.

Y es precisamente en la necesidad de corrección de estas falencias que se presentan en el proceso de aprendizaje de los niños, donde se origina la idea de desarrollar una herramienta robótica didáctica que empleó la ingeniería (reconocimiento de voz) para fines educativos. Cabe decir, que un bajo desarrollo de la orientación espacial puede incidir entre otros aspectos del desarrollo de básico de los niños como lo son los deportes e inclusive juega un papel sumamente importante en el desarrollo de la lectura y la escritura. Además, se decidió adicionar a esta herramienta la identificación de colores, que aunque no es una noción tan complicada de desarrollar como lo es la orientación espacial vale la pena reforzar.

Para que dicha herramienta de control por voz funcione correctamente, se ha desarrollado un sistema de reconocimiento de voz que responde adecuadamente a las palabras comandos que el robot debe ejecutar.

Finalmente se realizó una interfaz grafica de usuario (GUI), amigable y didáctica que permite al niño introducir los respectivos comandos sonoros de las acciones que el robot debe ejecutar. Una vez el sistema de reconocimiento detecte la palabra comando pronunciada, este enviará inalámbricamente una señal que el Lego Mindstorms NXT pueda interpretar para realizar el movimiento o reconocimiento especificado por el niño (Ver Figura 1).

Figura 1. Diagrama general de la herramienta robótica desarrollada



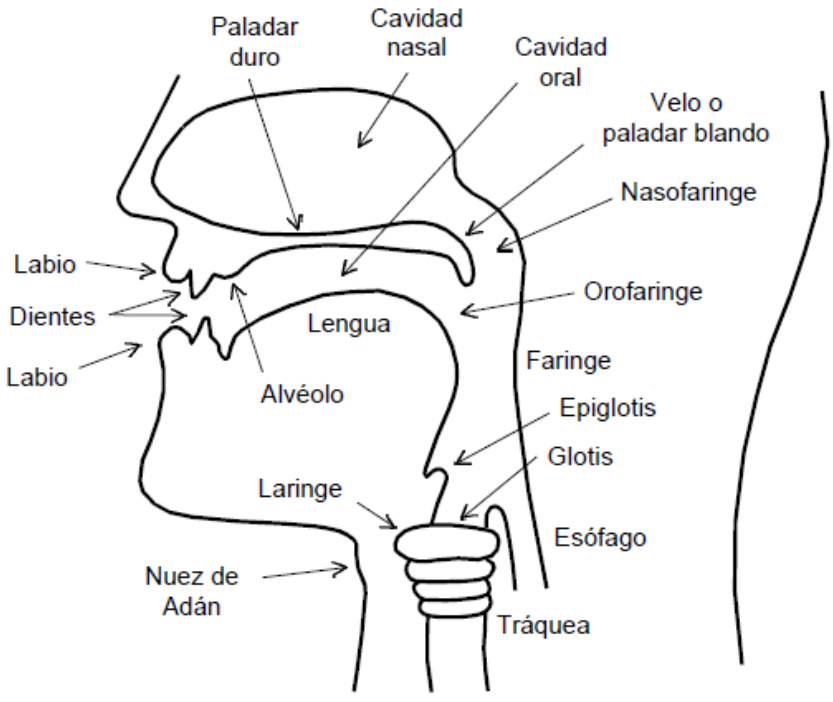
2. LA VOZ HUMANA

La voz es el sonido producido voluntariamente por el aparato fonatorio humano[2]. El investigador Sueco Johan Sundberg, especializado en la voz humana, la ha definido así: "Sonido complejo formado por una frecuencia fundamental (fijada por la frecuencia de vibración de los ligamentos vocales) y un gran número de armónicos o sobretonos".

Según las leyes de la acústica, hay tres elementos indispensables para la producción del sonido: un cuerpo vibrante, un medio elástico que propague las vibraciones y una caja de resonancia que las amplifique, con el fin de que puedan ser percibidas por el oído.

El aparato fonatorio humano cumple con las tres condiciones señaladas: el cuerpo que vibra son las cuerdas vocales, situadas en la laringe; el medio de propagación es el aire proveniente de los pulmones y la caja de resonancia está formada por la cavidad torácica, la faringe, las cavidades oral y nasal y una serie de elementos articulatorios, que son los labios, los dientes, el paladar, el velo paladar y la lengua, Ver Figura 2.

Figura 2. Corte esquemático del aparato fonatorio humano



La voz humana. MIYARA, Francisco.

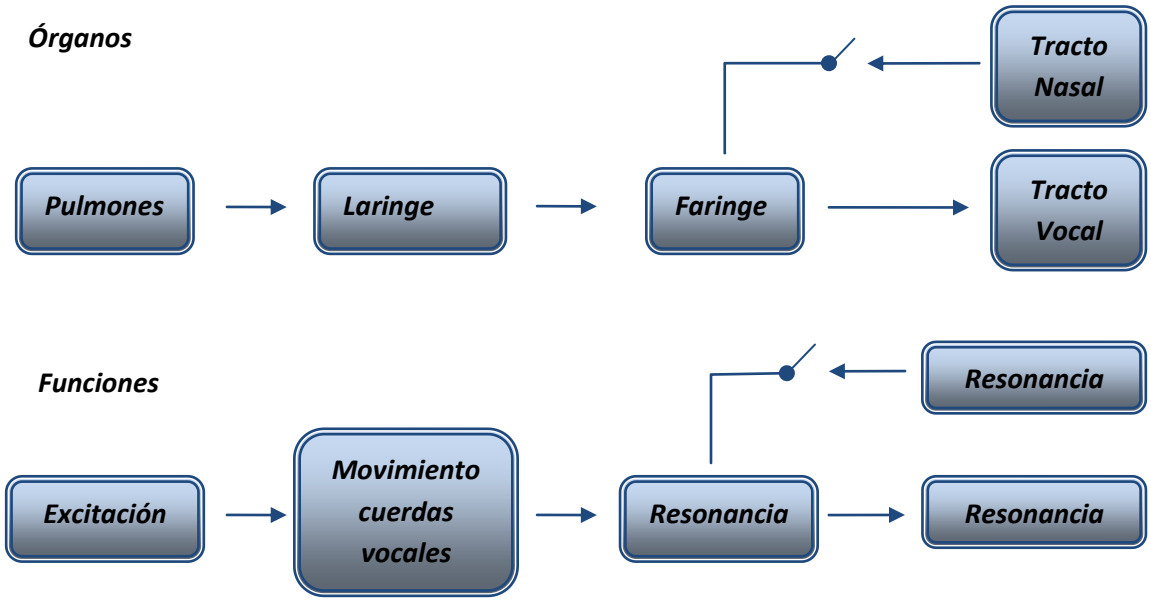
[2] Sundberg, Johan. The Science of Musical Sounds. San Diego, 1991. Academic Press.

2.1 DESCRIPCIÓN DEL PROCESO DE PRODUCCIÓN DE LA VOZ

El proceso básico de producción de la voz se inicia con la voluntad. En principio aparece el deseo de emitir un sonido, y éste desencadena en el sistema nervioso central un gran número de órdenes que pondrán en funcionamiento todos los elementos que producen la voz: mecanismos de la respiración, de la fonación, de la articulación, de la resonancia, de la expresión, etc.

Cuando se quiere emitir un sonido, ya sea para hablar o cantar, las cuerdas vocales se cierran. En esta situación el aire espirado no encuentra vía libre para salir y se crea una presión; cuando ésta alcanza un grado determinado, vence la resistencia que ofrecían las cuerdas vocales y al pasar a través del espacio que éstas le dejan las hace vibrar, produciendo un leve sonido que será más grave o más agudo según el grado de tensión a que sean sometidas (entre otras condiciones). En la Figura 3, se muestran los órganos que intervienen en la producción de la voz con sus respectivas funciones.

Figura 3. Sistema de producción de la voz



La porción que incluye las cavidades faríngea, oral y nasal se denomina genéricamente cavidad supraglótica, en tanto que los espacios por debajo de la laringe, es decir la tráquea, los bronquios y los pulmones, se denominan cavidades infraglóticas. Varios de los elementos de la cavidad supraglótica se controlan a voluntad, permitiendo modificar dentro de márgenes muy amplios los sonidos producidos por las cuerdas vocales o agregar partes distintivas a los mismos, e inclusive producir sonidos propios. Todo esto se efectúa por dos mecanismos principales: el filtrado y la articulación.

El filtrado actúa modificando el espectro del sonido. Tiene lugar en las cuatro cavidades supraglóticas principales: la faringe, la cavidad nasal, la cavidad oral y la cavidad labial. Las mismas constituyen resonadores acústicos que enfatizan determinadas bandas frecuenciales del espectro generado por las cuerdas vocales, conduciendo al concepto de formantes, es decir una serie de picos de resonancia ubicados en frecuencias o bandas de frecuencias que son características de cada tipo de sonido.

La articulación es una modificación principalmente a nivel temporal de los sonidos y está directamente relacionada con la emisión de los mismos y con los fenómenos transitorios que los acompañan. Está caracterizada por el lugar del tracto vocal en que tiene lugar, por los elementos que intervienen y por el modo en que se produce, factores que dan origen a una clasificación fonética de los sonidos.

2.2 MODELADO DE LA SEÑAL DE VOZ

El modelado de una señal es un tipo de representación que persigue principalmente conseguir una mayor eficiencia y flexibilidad al transmitir o almacenar señales. La naturaleza del modelo depende de su objetivo:

- Si es clasificar señales, se concentrará en eliminar detalles irrelevantes.
- Si es la codificación y transmisión, se concentrará en eliminar las partes de la señal que no son perceptibles.
- Si es modificar la señal, se concentrará en aislar parámetros de control dentro de ella. Aunque existen procesos comunes para todos los objetivos.

Las ecuaciones fundamentales que se aplican a la acústica son lineales, se pueden utilizar sistemas lineales en el modelado consiguiendo una precisión considerable. Estos modelos lineales siempre serán aproximaciones, pero utilizar modelos no lineales es extremadamente complejo.

Una aproximación razonable a la producción de voz desde el punto de vista de un modelo digital es considerar a las cuerdas vocales como un generador de impulsos cuasi-periódicos que produce los sonidos sonoros. Por otra parte los sonidos sordos tendrían su origen en un generador de ruido aleatorio. La salida de ambos generadores pasaría a través de una cavidad resonante (tracto vocal) considerada como un filtro digital variable en el tiempo tal como se muestra en la Figura 4.

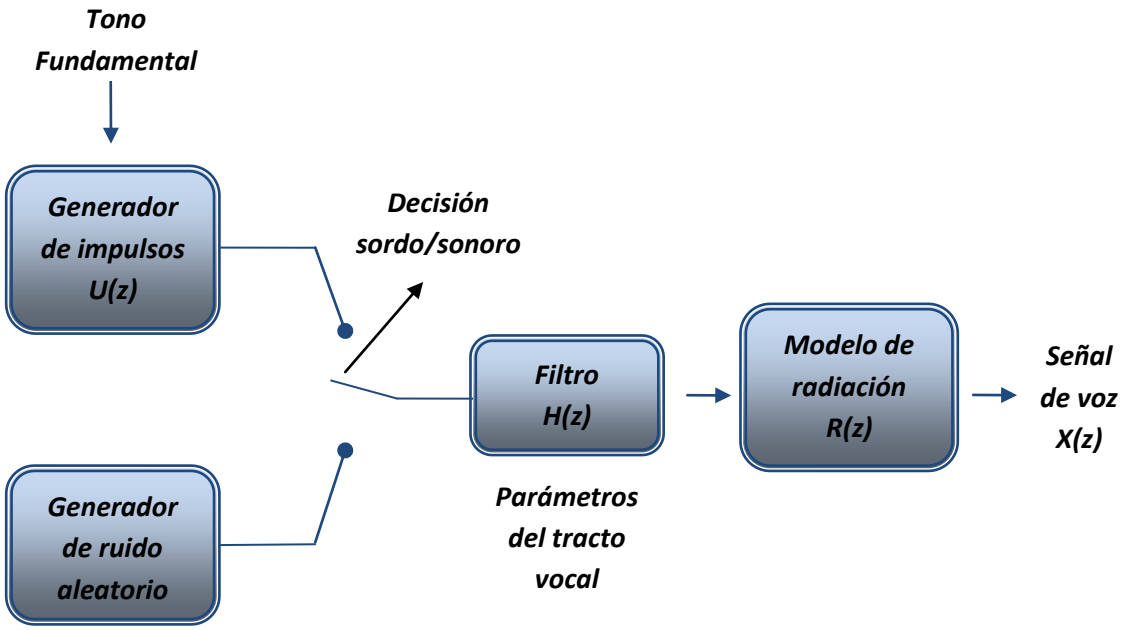
El modelo de radiación R(z) se encarga de reproducir el efecto de la impedancia de radiación que el medio opone a la salida del habla desde la boca. Cuando la señal abandona los labios se ocasiona una atenuación de altas frecuencias sonoras, por tal razón generalmente se aplica un filtro de pre-énfasis que busca amplificar las altas frecuencias que fueron atenuadas anteriormente.

Este modelo ha sido ampliamente aceptado en el campo del reconocimiento de voz gracias a los buenos resultados que ofrece. Sobre la base de este modelo se han diseñado numerosas

representaciones de la señal de voz que reducen la cantidad de datos a procesar en la fase de reconocimiento.

En este modelo digital se asume que las muestras de la onda de la voz son la salida de un filtro digital variable en el tiempo que aproxima las propiedades de transmisión del tracto vocal y las propiedades espectrales del pulso glotal. Para los sonidos sonoros un generador de tren de impulsos excita el filtro creando un tren de impulsos cuasi-periódicos en el cual el espacio entre impulsos corresponde al periodo fundamental (pitch) de la excitación glotal. Para los sonidos sordos, es un generador de ruido aleatorio el que excita al filtro por medio de un ruido de espectro plano.

Figura 4. Modelo digital de producción de voz



Sea X(z), U(z) y H(z), las transformadas Z de las señales de voz, excitación y filtro variable. El modelo de producción de la voz está dado por la expresión siguiente:

$$X(z) = U(z)\,H(z)$$

En la práctica, en la mayoría de las aplicaciones se modela el filtro H(z) como un filtro todo-polos:

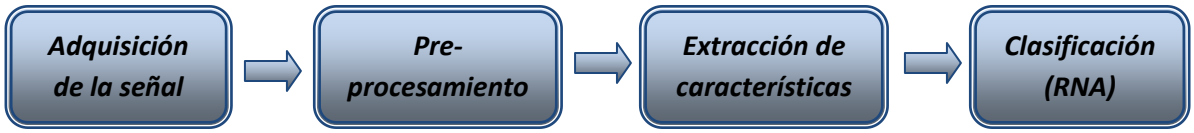
$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} \propto_k z^{-k}}$$

3. PROCESAMIENTO DIGITAL DE LA SEÑAL DE VOZ PARA EL RECONOCIMIENTO DE PALABRAS AISLADAS

Como ya se había mencionado, el objetivo principal de este proyecto es elaborar una herramienta robótica educativa controlada por voz. Esto implica que se debe desarrollar un algoritmo que permita identificar efectivamente las palabras comando pronunciadas por el usuario. En dicho algoritmo inicialmente se adquirirá la señal de voz, posteriormente se almacenará digitalmente, se pre-procesará empleando para ello las técnicas matemáticas necesarias para el análisis de señales no estacionarias; que permitirán la extracción de las características de la señal sonora y el posterior reconocimiento de patrones a través de una Red Neuronal Artificial (RNA).

Por tal razón se puede decir que el sistema de reconocimiento desarrollado consta de 4 etapas bien definidas como se muestra en la Figura 5: Adquisición, Pre-procesamiento, Extracción de características y Clasificación. En este capítulo se describirá con detalle cada una de estas etapas.

Figura 5. Etapas sistema de reconocimiento



Para realizar los algoritmos que permitieron realizar el proceso de reconocimiento, se eligió como software de programación MATLAB (*MATrix LABoratory*) de *MathWorks*; por ser un lenguaje que integra computación, visualización, y programación, en un entorno fácil de usar donde los problemas y las soluciones son expresados en la más familiar notación matemática[3]. Además, MATLAB presenta una familia de soluciones a aplicaciones específicas de acoplamiento rápido llamadas toolboxes y para esta aplicación resultaron muy útiles las toolboxes de adquisición de datos (*Data Acquisition*), procesamiento de señales (*Signal Processing*) y redes neuronales (*Neural Networks*)

3.1 ADQUISICIÓN DE LA SEÑAL DE VOZ

Una de las etapas más importantes en el proceso de reconocimiento es precisamente la relacionada con la adquisición de la señal de voz, ya que la calidad de las muestras tomadas incide directamente a la hora de evaluar el rendimiento del algoritmo de reconocimiento.

En la Figura 6 se muestra el diagrama de bloques de esta etapa. El hardware empleado para llevar a cabo la adquisición de la señal de voz es la tarjeta de sonido de la cual se realiza un breve descripción en uno de los documentos contenidos en el CD que se adjunta al presente documento.

[3] K. INGLE, Vinay y G. PROAKIS, Jhon. Digital. Signal Processing using MATLAB. PWS Publishing Company, 1997.

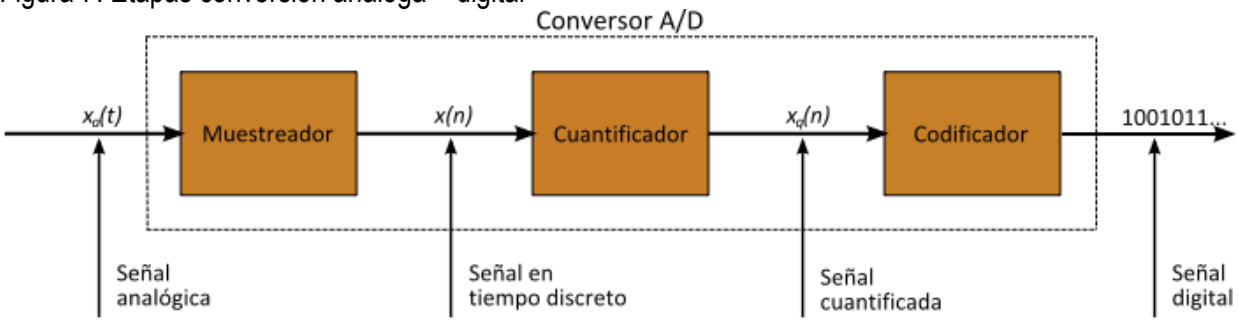
Figura 6. Diagrama de bloques etapa de adquisición

3.1.1 Conversión acústica – eléctrica. La adquisición de la señal de voz (comandos) pronunciado por el niño se inicia con la captura de la misma empleando para ello un transductor acústico (micrófono), que convierte la variación de la presión ejercida sobre su membrana en una señal eléctrica analógica que varía en el tiempo. Esta señal es introducida al computador mediante la conexión del micrófono a la entrada de la tarjeta de sonido.

Para la adquisición de las muestras empleadas para el entrenamiento de la Red Neuronal Artificial (RNA), se trató que el ambiente donde fueron tomadas no fuera tan ruidoso, con el objetivo de tener una relación señal a ruido S/N considerable; ya que el problema de discriminación del habla con respecto al ruido de fondo no es trivial, salvo cuando se cumple esta condición.

3.1.2 Conversión Análoga – Digital A/D. Una vez convertida la voz en una señal eléctrica analógica, se procede a la digitalización de la misma mediante un conversor análogo/digital. Para ello se hace uso del conversor A/D que contiene la tarjeta de sonido. Este proceso se lleva a cabo en 3 fases: muestreo, cuantificación y codificación, como se muestra en la Figura 7.

Figura 7. Etapas conversión análoga – digital



Wikipedia, enciclopedia libre.

3.1.2.1 Muestreo. El muestreo se encarga de convertir la señal analógica, continua en el tiempo, en una señal discreta en el tiempo. El ritmo de este muestreo, se denomina frecuencia o tasa de muestreo y determina el número de muestras que se toman en un intervalo de tiempo.

Según el teorema de muestreo de Nyquist-Shannon, para poder replicar con exactitud la forma de una onda es necesario que la frecuencia de muestreo sea superior al doble de la máxima frecuencia a muestrear[4].

Para el muestreo de la señal de voz, se empleó una frecuencia de muestreo de 8Khz. La elección de dicha frecuencia de muestreo se basó atendiendo a tres aspectos que se explican seguidamente.

El primero de ellos es que los estudios sobre las características de las señales de voz han demostrado que en el espectro de la señal de voz la mayoría de su energía se concentra entre los 20 y 4000 Hz y que la mayor parte de la información necesaria para la inteligibilidad del habla se encuentra por debajo de los 4000Hz. La segunda es que a pesar de que las frecuencias audibles por un ser humano se encuentran entre 20 y 20000 Hz, los canales telefónicos trabajan con un ancho de banda de 300 a 3400 Hz, sin que la pérdida de información (frecuencias menores a 300 y superiores a 3400Hz) suponga un déficit sustancial en la información del habla. El tercero es basándonos en las pruebas realizadas ya que se muestreo la señal a diferentes tasas de muestreo (11.025kHz, 22.050 kHz y 44.100 kHz). El incremento en la frecuencia de muestreo permitía obtener una fidelidad más alta de la señal, pero el posterior procesamiento de dichas señales por parte del PC era lento y complicado dada la cantidad de muestras tomadas.

3.1.2.2 Cuantificación. Una vez se tiene la señal discretizada en el tiempo queda discretizarla en magnitud para tener una señal digital. Durante el proceso de cuantificación se mide el nivel de tensión de cada una de las muestras, obtenidas en el proceso de muestreo y se les atribuye un valor finito (discreto) de amplitud, seleccionado por aproximación dentro de un margen de niveles previamente fijado (número de bits).

Para la adquisición de las señales se ha decidido cuantificar a 16 bits. Teniendo de esta manera una velocidad de transmisión de datos de 128 kbits/s.

Inicialmente, se tomaron muestras cuantificando a 8 bits, obteniendo una calidad aceptable. Luego se hicieron pruebas cuantificando a 16 bits y con la que se consiguió una reducción significativa de ruido. Esto se debe a que cada bit adicional que se agrega, contribuye a mejorar la relación señal a ruido S/N en aproximadamente 6 dB. La señal de voz exhibe un rango dinámico de unos 50 a 60 dB, por lo que una cuantificación a 8 bits resultaría apropiada, pero para mejorar aún más esta relación se optó por cuantificar a 16 bits.

3.1.2.3 Codificación. Consiste en la traducción de los valores de tensión eléctrica analógicos que ya han sido cuantificados (ponderados) al sistema binario, mediante códigos preestablecidos. La señal analógica va a quedar transformada en un tren de impulsos digital (sucesión de ceros y unos). En la Tabla 1 se resumen los valores empleados para la adquisición.

[4] K. INGLE, Vinay y G. PROAKIS, Jhon. Digital Signal Processing, Principles, Algorithms and Applications. Tercera edición. Prentice-Hall.

Tabla 1. Parámetros de adquisición de las muestras

Frecuencia de Muestreo	8000 Hz
Número de bits	16
Tiempo de grabación	3 s

3.1.3 Adquisición de la señal de voz por medio de MATLAB. Para la realización del código que permitió llevar a cabo la adquisición de la señal sonora por medio de la tarjeta de sonido, se empleó el *toolbox* de adquisición de datos de MATLAB *Data Acquisition*.

El adaptador asociado a la tarjeta de sonido en MATLAB es *winsound*. Se puede consultar la información relacionada con el adaptador con el comando *daqhwinfo('winsound').* Lo primero que se hizo fue crear un objeto de entrada análogo asociado a la tarjeta de sonido que permita la entrada de la señal de voz, esto lo conseguimos ejecutando la siguiente línea:

```
in = analoginput('winsound');
```

Seguidamente se procedió a agregar un canal (Mono estéreo) al objeto análogo creado anteriormente con la función *addchannel*. Solo se empleó un canal ya que con las pruebas realizadas se detectó que el hecho de adicionar más canales, no influía mucho en el momento de la adquisición pero si complicaba el posterior procesamiento de la señal. Con la función *get(in)*, se pueden ver las propiedades asociadas al objeto creado.

Todas estas propiedades se pueden modificar empleando para ello la función *set.* Cuando la propiedad *StandardSampleRates* está en "on", se toma el valor más bajo de las tasas de muestreo estándar que es 8000. Si se desea una tasa de muestreo diferente se debe desactivar dicha propiedad y colocar la frecuencia de muestreo deseada con la propiedad *SampleRate,* como se muestra a continuación:

```
set(in,'StandardSampleRates','off');
set(in,'SampleRate', 11025);
```

Otra de las propiedades que debemos configurar para que la adquisición se realice de acuerdo a los parámetros que deseamos es *BitsPerSample*, que nos permite introducir el número de bits con el cual se va a realizar la cuantificación, que como ya se había mencionado es 16 bits.

También se debe configurar la forma como se activará el inicio de la adquisición de la señal y el número de muestras que tomará por cada disparo con las propiedades *TriggerType* y *SamplesPerTrigger.* Los tipos de disparo son Inmediato, Manual o Software. Si es inmediato el disparo ocurre inmediatamente después de que se ejecuta la función *start.* Si es manual este ocurre inmediatamente después de que se ejecuta la función *trigger* y si es software el disparo se produce cuando la condición de disparo asociada es satisfecha. En cuanto a la propiedad

SamplesPerTrigger, en nuestro caso esta tomaría un valor de 24000, ya que se desea adquirir la señal por 3 s y en cada segundo se toman 8000 muestras.

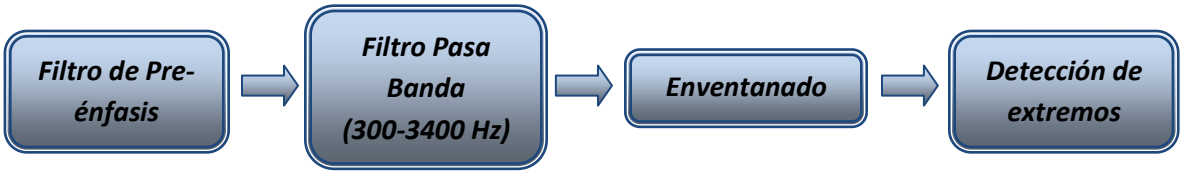
Finalmente la señal adquirida es almacenada en un archivo de sonido ".wav" o wave con la función *wavewrite,* y el cual se leerá con la función *wavread* para realizar su posterior procesamiento. Para más detalle sobre el código desarrollado para realizar la adquisición de la señal se puede consultar los M- Files en el CD que se anexa a este documento.

3.2 PRE-PROCESAMIENTO DE LA SEÑAL SONORA

Cuando hablamos de pre-procesamiento se hace referencia a una serie de operaciones matemáticas que se le deben realizar a la señal sonora en el dominio del tiempo, con el fin de atenuar o eliminar características indeseables y resaltar otras que serán bastante útiles a la hora de realizar la clasificación e identificación de la palabra.

Como se muestra en la Figura 8, el pre-procesamiento que se realizó a la señal adquirida incluye la aplicación de filtros de pre-énfasis y pasa banda, la posterior segmentación de la señal en tramas o enventanado y finalmente la supresión de silencios iníciales y finales de la señal adquirida. Al finalizar esta etapa se conseguirá eliminar una gran cantidad de datos que no aportan ninguna información relevante en el proceso de reconocimiento

Figura 8. Diagrama de bloques etapa de pre-procesamiento

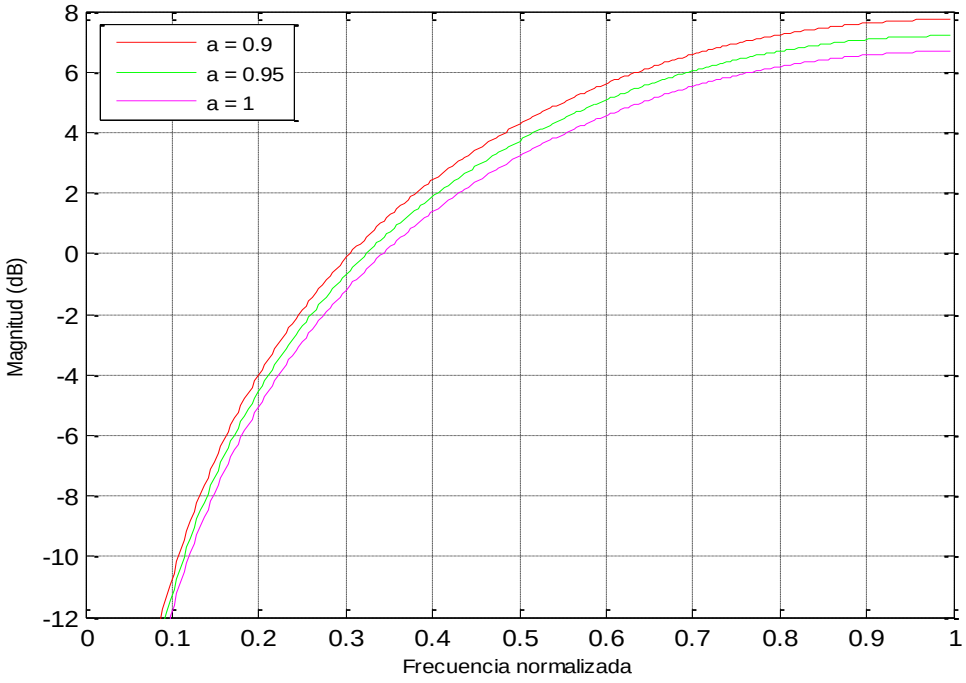


3.2.1 Filtro de pre-énfasis. Una de las razones por las cuales se aplica este filtro es con el objetivo de para aplanar la respuesta en frecuencia de la señal para hacerla menos susceptible a los efectos de cálculo digital (efectos del uso de precisión finita del computador). Este filtro está definido por la siguiente función de transferencia:

$$H(z) = 1 - az^{-1} \qquad 0.9 \leq a \leq 1$$

Pero el principal motivo de la aplicación de este filtro, se fundamenta en el modelo de radiación R(z), explicado…en la sección 2.2…Como se puede ver en la Figura 9, este es un filtro pasa alta que realza las frecuencias que fueron atenuadas cuando la señal abandonó los labios.

Figura 9. Respuesta en frecuencia del filtro de pre-énfasis para diferentes valores de *a*



Realizando pruebas con las diferentes muestras adquiridas, se optó por utilizar un *a = 0.95*. Con este valor se obtuvieron buenos resultados. En la Figura 10 se muestra como la aplicación del filtro de pre-énfasis permitió eliminar componentes de baja frecuencia que se encontraban en el silencio inicial de la señal adquirida.

3.2.2 Filtro pasa banda. Aunque con la aplicación del filtro de pre-énfasis se logró eliminar componentes de baja frecuencia de la señal adquirida; se hace aun necesaria la aplicación de otro filtro que nos permita eliminar las frecuencias correspondientes al ruido y obtener una mejor relación señal a ruido S/N.

Para conseguir dicho objetivo se ha sometido la señal resultante de la aplicación del filtro de pre-énfasis a un filtro con respuesta impulso finita FIR. Aunque los filtros FIR requieren un orden (número de coeficientes) mucho mayor que los filtros IIR, lo que implica un mayor gasto computacional; se ha seleccionado este tipo de filtro debido a su estabilidad y a su respuesta de fase lineal, lo que hace que la señal que pase a través de él no sea distorsionada (Ver Figura 11).

El filtro FIR que se empleó es un filtro pasa banda de aproximadamente 300 a 3400Hz (ancho de banda del canal telefónico tradicional), que se consideró apropiado dada la aplicación del proyecto y con base en el hecho de que una señal que represente la voz humana no suele tener información relevante mas allá de este ancho de banda.

Figura 10. Respuesta en tiempo antes y después de la aplicación del filtro de pre-énfasis para la palabra “abajo”

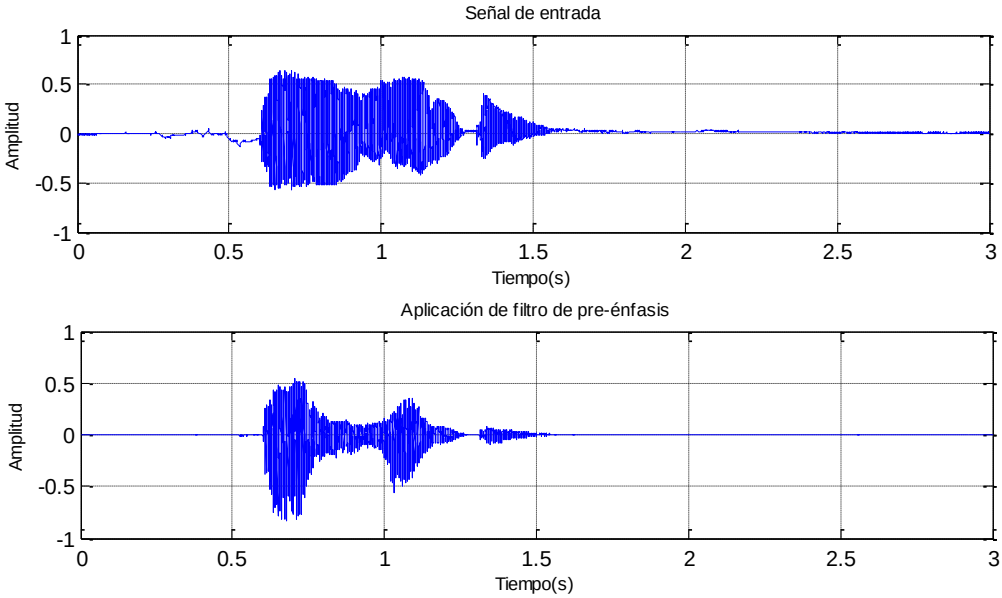
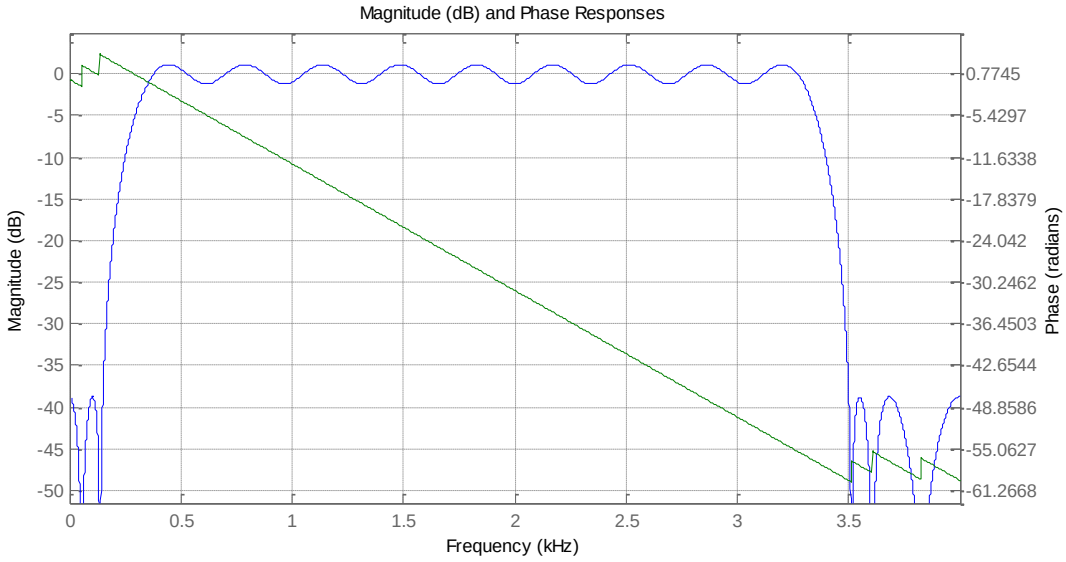


Figura 11. Respuesta en magnitud y fase del filtro pasa banda FIR diseñado



El diseño de este filtro se realizó con la herramienta *FDATool* (*Filter Design and Analysis Tool*) del *toolbox* de procesamiento de señales de MATLAB. En la Tabla 2 se especifican los parámetros característicos más relevantes del filtro FIR utilizado.

El método de diseño seleccionado fue igual riple (*equiripple)*, que permitió tener una respuesta en frecuencia de la magnitud con rizo uniforme tanto en la banda de paso como en la banda de rechazo y minimizar el máximo error en la banda de paso como se muestra en la Figura 11. Se puede mejorar mucho más la respuesta de este filtro pasa banda, pero esto ocasiona un incremento en el orden del mismo y en consecuencia en el tiempo de procesamiento, efecto que no es deseable para esta aplicación.

Tabla 2. Parámetros de diseño de filtro FIR pasa banda empleado

Especificaciones FIR pasa banda	
Frecuencia de muestreo	8000 Hz
Frecuencia de corte de la banda de rechazo inferior	150 Hz
Frecuencia de corte de la banda de paso inferior	350 Hz
Frecuencia de corte de la banda de paso superior	3300 Hz
Frecuencia de corte de la banda de rechazo superior	3500 Hz
Atenuación en la banda eliminada inferior	40 dB Hz
Rizado en la banda de paso	2 dB Hz
Atenuación en la banda eliminada superior	40 dB Hz
Orden del filtro	48

3.2.3 Enventanado. Hasta el momento se ha logrado aplanar la respuesta en frecuencia y eliminar información no deseada de la señal adquirida (ruido). Se puede decir que con esto se ha realizado un pre-procesamiento inicial y que se puede iniciar el análisis de la señal de voz.

La señal de voz es claramente no estacionaria en sus características aunque se puede estudiar la estacionariedad de la señal si el tramo es lo suficiente breve (decenas de ms)[5]. Para aplicar técnicas de análisis y procesado se debe limitar el segmento a procesar en este orden de magnitud. Esto da origen al análisis localizado o análisis a corto plazo de la señal. La teoría afirma que la longitud de las tramas debe estar entre 10 y 45 ms, ya que con un segmento cuya longitud se encuentre en este rango hay bastantes posibilidades de conseguir una parte representativa de la señal.

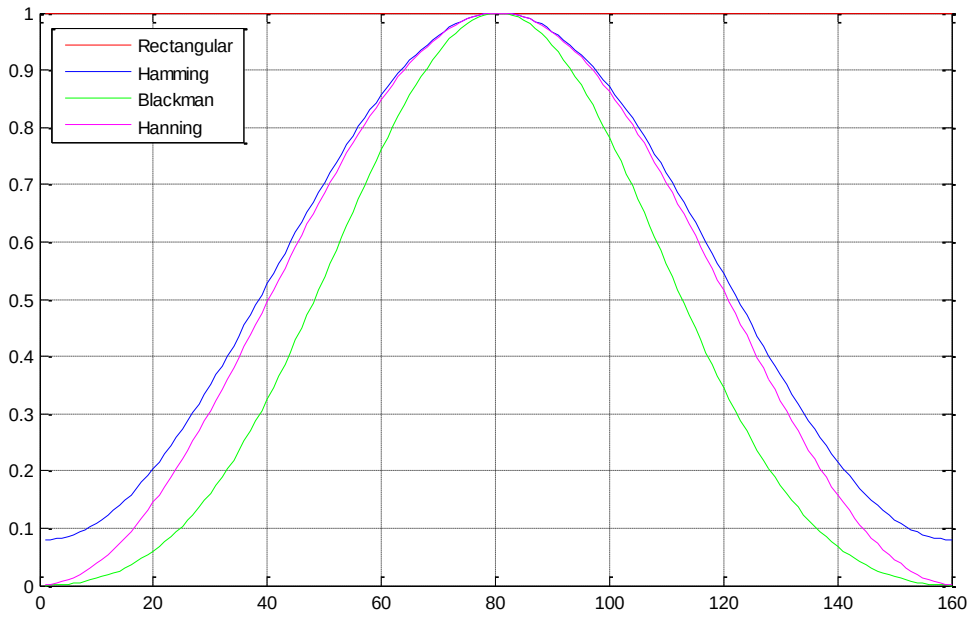
El proceso de segmentación o división de la señal en tramas, supone extraer una parte de la señal separándola de todo el conjunto, esto provoca un efecto negativo para el análisis de la evolución en el tiempo de las características de la señal. La solución a este problema reside en aplicar a cada segmento una función ventana, que suaviza los bordes del intervalo haciendo que estos tiendan a cero y resalta la parte central acentuando las propiedades características del segmento.

Entre los tipos de ventanas más conocidas encontramos: Rectangular, Hamming, Hanning y Blackman, las cuales se pueden apreciar en la Figura 12. La elección de la ventana incide en el

[5] PEREIRA RAMA, Antonio. Procesado digital de la señal sonora utilizando MATLAB.

espectro resultante de la señal ya que las características de inicio y finalización de las ventanas, permiten eliminar las discontinuidades generadas por la segmentación.

Figura 12. Diferentes tipos de ventanas generalmente utilizadas



Para la selección de la ventana más apropiada se analizó la respuesta espectral de las ventanas anteriormente mencionadas con base a este se decidió aplicar una ventana tipo Hamming:

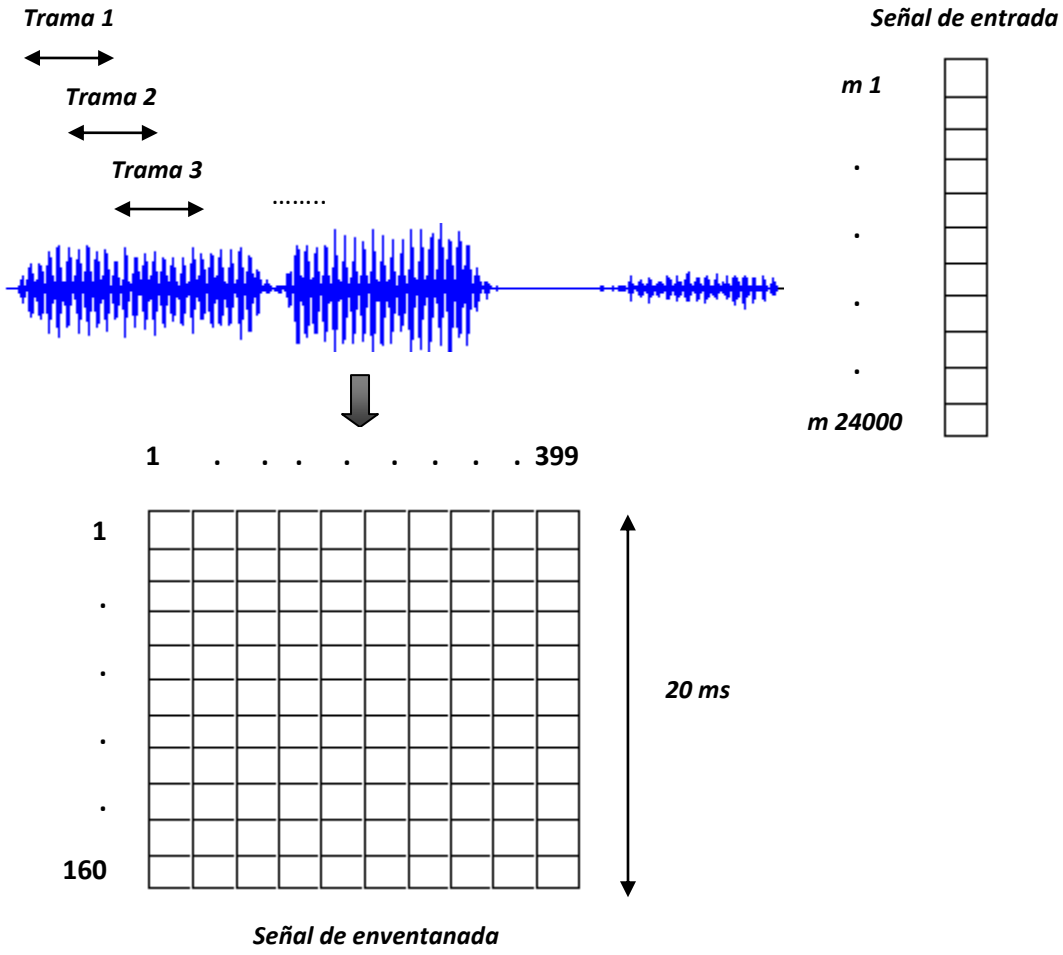
$$w(n) = 0.54 - 0.46\, cos\left(2\pi \frac{\mathrm{n}}{\mathrm{N}}\right), 0 \leq n \leq N$$

Este tipo de ventana presenta una buena resolución dado su estrecho lóbulo principal, que permite distinguir mejor las frecuencias cercanas (Ver ANEXO A). Además la ventana tipo Hamming presenta una atenuación considerable de los lóbulos secundarios con respecto al principal, lo que evita interacción de frecuencias y variaciones de amplitud.

Aunque el lóbulo principal de la ventana rectangular como se muestra en los ANEXO A es más estrecho que el de la ventana Hamming; la atenuación de los lóbulos secundarios con respecto al principal puede producir un efecto llamado *leakage* (pérdida), que origina valores erróneos de amplitud, dado que en los extremos de dicha ventana la función decae rápidamente.

Para llevar a cabo el proceso de enventanado, segmentamos la señal en tramas de 20 ms correspondientes a 160 muestras, solapadas 10 ms, tal como se ilustra en Figura 13, debido a que se busca que las ventanas consecutivas sean suficientemente próximas para que las transiciones rápidas de la señal de voz se puedan medir correctamente.

Figura 13. Diagrama esquemático del proceso de enventanado con tramas de 20 ms, solapamiento del 50%; para una señal muestreada a 8000Hz y con duración de 3 s



3.2.3 Detección de extremos. En el proceso de reconocimiento de palabras aisladas, se hace necesario determinar dónde empieza y finaliza la señal de voz, con el fin de eliminar los silencios iníciales y finales de la grabación que no contienen ninguna información que pueda caracterizar la señal.

La extracción de segmentos de voz y eliminación de silencios es lo que se conoce como detección de extremos (*Endpoint detection*) y se hace con el fin de separar la voz de otros eventos. Los algoritmos de detección de extremos son comúnmente basados en el uso de la energía como característica principal para la clasificación de los segmentos y posterior localización de los puntos de inicio y de fin debido a la sencillez para su cálculo.

3.2.3.1 Energía localizada. El uso del promedio de energía en periodos cortos es la forma más sencilla de clasificar la señal en segmentos de voz y silencio (ruido), dado que la señal presenta, a priori, una mayor energía que el ruido. El promedio temporal de la energía puede definirse como:

$$E = \sum [s(n) \cdot w(n-m)]^2$$

Donde s(n) es la señal de entrada de larga duración y w(n) es la ventana temporal aplicada. Este método de detección puede proporcionar un buen rendimiento cuando la energía de los periodos de voz es suficiente mayor que la del ruido del ambiente. Sin embargo, cuando la energía del ruido ambiente es tambien elevada, la característica energía de la señal proporcionará un rendimiento muy bajo.

El cálculo de la energía proporcionó información importante para realizar una efectiva detección de extremos, porque permitió diferenciar claramente segmentos de voz de ruido como se muestra en la Figura 14. En el proceso de detección de extremos que efectuamos partimos de la información proporcionada por la energia, a partir de esta se establece un umbral dinámico que determina si el segmento analizado corresponde o no a señal de voz.

Realizando pruebas con diferentes muestras definimos este umbral como la sexta parte del promedio de la energía de la señal. Cuando el segmento analizado supera este umbral se puede afirmar que contiene información referente a la palabra y en caso contrario se interpreta como un segmento de ruido.

Para identificar el inicio y final de la palabra simplemente se busca el primer y último segmento en que la señal superó el umbral establecido como se muestra en la Figura 14 Con la detección de estos segmentos, se puede seleccionar solo la parte correspondiente a la señal sonora, reduciendo de esta manera el tamaño del vector de entrada, lo que disminuye considerablemente el tiempo computacional de procesamiento.

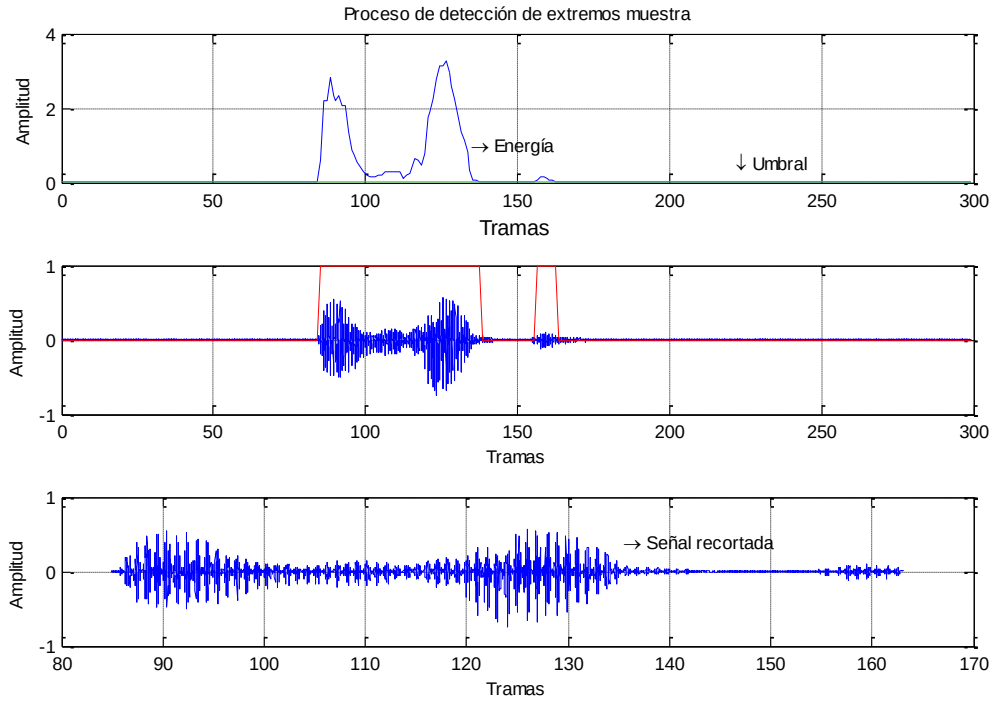
Cabe aclarar que este método de detección de extremos, arroja buenos resultados mientras el ruido ambiente no sea elevado porque cuando las condiciones de ruido son bastante adversas se puede realizar una detección incorrecta de la palabra.

3.3 EXTRACCIÓN DE CARACTERÍSTICAS

Con el pre-procesamiento aplicado anteriormente se ha logrado eliminar de la señal adquirida el ruido y segmentos que no aportan ninguna información valiosa y por el contrario dificultan el proceso de reconocimiento.

Lo que se busca con la extracción de características es que cada señal pueda ser representada inequívocamente como un conjunto de valores que se distinga de las demás. Pero la señal sonora presenta una gran variabilidad; es imposible que un locutor (y con más razón varios locutores) pronuncie dos veces exactamente igual una silaba, palabra o frase.

Figura 14. Proceso de detección de extremos para una muestra de la palabra “adelante”



Existen diferentes herramientas matemáticas útiles en el proceso de extracción de características; pero sea cual sea método escogido este deberá ser sencillo de obtener y significativo (suficiente para alcanzar los objetivos de reconocimiento propuestos). Con base en lo anterior, el método seleccionado en este proyecto para la extracción de características fue el cálculo de los coeficientes cepstrales en escala de Mel o MFCCs (*Mel-Frequency Cepstral Coefficients*).

Antes de explicar esta técnica se describirá brevemente la extracción de características por medio del cálculo de los coeficientes cepstrales, ya que esta técnica ayudará a entender el proceso que se debe llevar a cabo para la obtención de los MFCCs.

3.3.1 Coeficientes Cepstrales. Como se señaló…en la sección 2.2… la señal de voz se puede modelar como la convolución de la excitación de una señal cuasi-periódica con un filtro variante en el dominio tiempo que está determinado por la configuración del tracto vocal. Para caracterizar el modelo de producción de la voz en función de los anteriores parámetros, se debe realizar un proceso de desconvolución y el cepstrum es el procedimiento utilizado para realizar dicha operación.

El cepstrum de la señal de voz se define como la transformada inversa de Fourier del logaritmo de su espectro localizado[6]. El término cepstrum es indicativo de haber realizado una transformación inversa del *spectrum* (espectro). La variable independiente del cepstrum se denomina *quefrency*, término formado a partir de la palabra frecuencia.

[6] McCLELLAN, Stan, GIBSON, Jerry.D.y otros. Speech Signal Processing. *The Electrical Engineering Handbook, 2000.*

A partir del modelo simplificado de la producción de voz, estudiado… en la sección 2.2… se tiene:

$$S(w) = H(w) \cdot U(w)$$

$$\log(|S(w)|) = \log(|H(w)| \cdot |U(w)|)$$

La densidad espectral de potencia viene dada por:

$$\log(|S(w)|^2) = \log(|H(w)|^2 \cdot |U(w)|^2)$$

$$\log(|S(w)|^2) = \log(|H(w)|^2 + |U(w)|^2)$$

Aplicando la transformada inversa de Fourier obtenemos:

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \log(|S(w)|^2)\, e^{jwn}\, dw$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \log(|H(w)|^2)\, e^{jwn}\, dw + \frac{1}{2\pi}\int_{-\pi}^{\pi} \log(|U(w)|^2)\, e^{jwn}\, dw$$

$$c_s[n] = c_H[n] + c_U[n]$$

Como se puede observar en la ecuación anterior con este procedimiento se logra pasar de una convolución de difícil resolución a una suma de logaritmos, en definitiva una suma de dos señales de muy distintas características.

El valor de $c_s[n]$ es lo que se conoce como coeficientes cepstrales. El término $c_H[n]$ corresponde a la parte de baja de la cuefrencia de la señal representa la envolvente del espectro del tracto vocal y los valores altos de cuefrencia representan el periodo de la señal de excitación $c_U[n]$.

El proceso de separar las componentes cepstrales en estos dos factores se denomina *liftering* (derivado de la palabra *filtering*, filtrado). Dado que la densidad espectral de potencia es una función real y par, se puede demostrar que aplicar la transformada inversa de Fourier es equivalente a aplicarle su transformada directa.

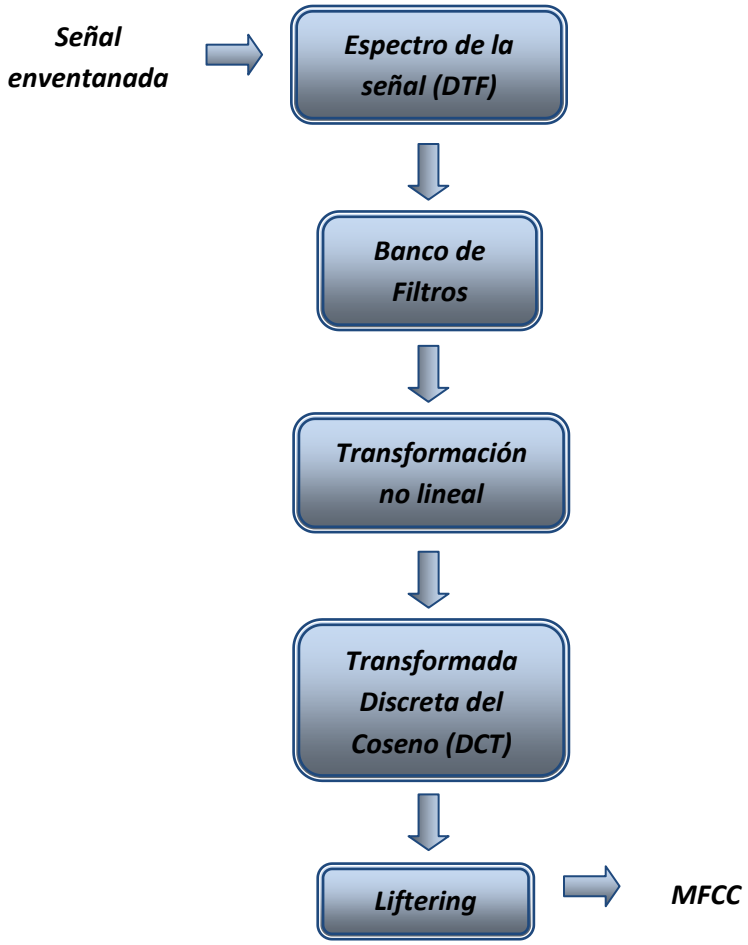
3.3.2 Coeficientes cepstrales en escala Mel. El cálculo de los coeficientes cepstrales en escala Mel o MFCCs es una técnica homomórfica muy utilizada en la actualidad, con la cual se obtienen buenos resultados en el proceso de reconocimiento automático de habla.

Este es un método mucho más robusto que al igual que los coeficientes cepstrales hace uso de la transformada de Fourier para obtener las características frecuenciales de la señal. El objetivo es desarrollar un conjunto de valores de características basados en el comportamiento del sistema auditivo humano. Para ello aplica un banco de filtros los cuales están en escala Mel.

Los parámetros MFCCs estiman la envolvente espectral desechando los parámetros cepstrales que provienen de la excitación de la señal de voz. En la Figura 15 se muestra el diagrama de bloques para la obtención de los coeficientes cepstrales en escala Mel.

Cada uno de los pasos involucrados en la obtención de los parámetros MFCC tiene un objetivo que se relaciona o bien con características perceptuales de nuestro sistema auditivo o con la adecuación de los parámetros al sistema de reconocimiento que seguidamente se explican.

Figura 15. Diagrama de bloques para la obtención de los MFCCs



3.3.2.1 Análisis Espectral. El análisis espectral hace referencia al proceso de descomponer frecuencialmente la señal. La herramienta matemática empleada para tal fin es la transformada discreta de Fourier designada con frecuencia DFT (Discrete Fourier Transform) y a la que en ocasiones se denomina transformada de Fourier finita. Esta es ampliamente empleada en tratamiento de señales y en campos afines para analizar las frecuencias presentes en una señal

muestreada, resolver ecuaciones diferenciales parciales y realizar otras operaciones, como convoluciones.

La definición de la transformada de Fourier supone un conocimiento de la señal para todo instante de tiempo y que cualquier propiedad o característica que se esté buscando con la aplicación de esta se mantenga invariante para todo instante de tiempo. Como se estudió…en la sección 3.2.3…, la señal de voz es una señal variante en el tiempo. Efectivamente, cada pocos milisegundos existe un cambio en esta; razón por la cual se segmenta la señal en tramas de 20 ms dando origen a un análisis localizado y de esta manera observar sus características frecuenciales.

La aplicación de la transformada de Fourier localizada proporciona información sobre las contribuciones que se tienen en la señal de voz que como se estudio… en la sección 2.2… corresponde al filtro variante que representa el tracto vocal (responsable de la estructura de formantes) y la excitación. Esta viene definida por la siguiente expresión:

$$S_n(e^{jw}) = \sum_{m=-\infty}^{\infty} s(m) \cdot w(n-m)\, e^{-jwm}$$

Para realizar el cálculo de la transformada de Fourier se ha empleado el algoritmo FFT (*Fast Fourier Transform),* el cual es una versión rápida de la transformada discreta de Fourier (DFT) que efectúa las mismas operaciones que la DTF pero en menos tiempo. No se tuvo en cuenta la fase de la FFT, debido a que no aporta información relevante para la discriminación de los sonidos, ya que el oído es insensible a las variaciones de fase. En la Figura 16 se puede observar el espectro para una de las palabras analizadas en este proyecto.

3.3.2.2 Banco de filtros en escala Mel. La cantidad de información obtenida de la aplicación de la transformada localizada de Fourier es excesiva y puede complicar el proceso de clasificación.

Para remediar este efecto se suele realizar grupos de bandas frecuenciales. Esto se logra mediante la aplicación de un filtro (generalmente triangular) a cada grupo lo que da origen al concepto de banco de filtros. La cantidad de grupos de bandas frecuenciales determina el número de canales del banco de filtros. El oído humano presenta una respuesta no lineal para las diferentes bandas de frecuencias, por lo tanto un banco de filtros en escala lineal sería inapropiado. Una solución a este problema es la transformación de frecuencias a escala Mel.

La escala Mel es una escala construida en base a la percepción humana del habla y sus valores han sido dados tras experimentos fisiológicos de muchos investigadores quienes han construido escalas de frecuencias basadas en la respuesta natural del sistema de audición humano. En la Figura 17 se muestra la equivalencia de frecuencias lineales (Hertz) en la escala Mel. Con base en pruebas realizadas se determinó que el número de canales del banco de filtros que proporciona mejores resultados es 33.

Figura 16. Espectro de la palabra “parar”

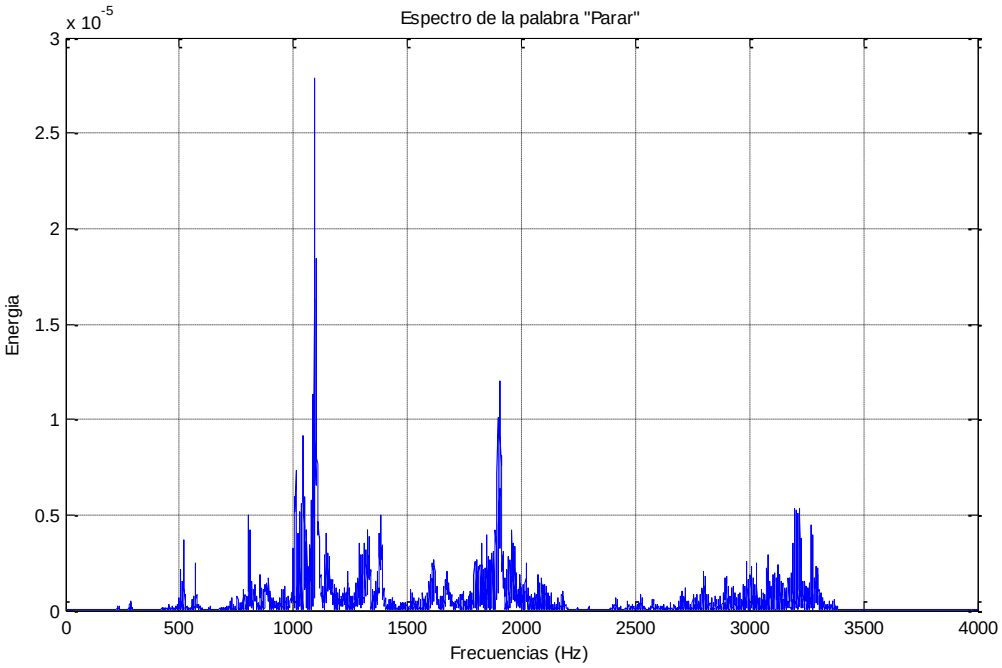
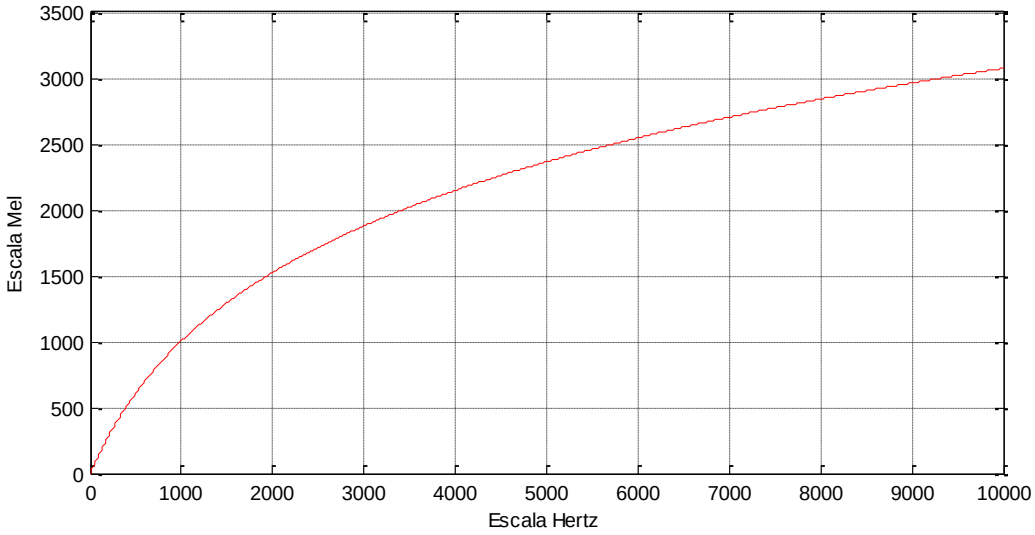


Figura 17. Grafica Hertz – Mels

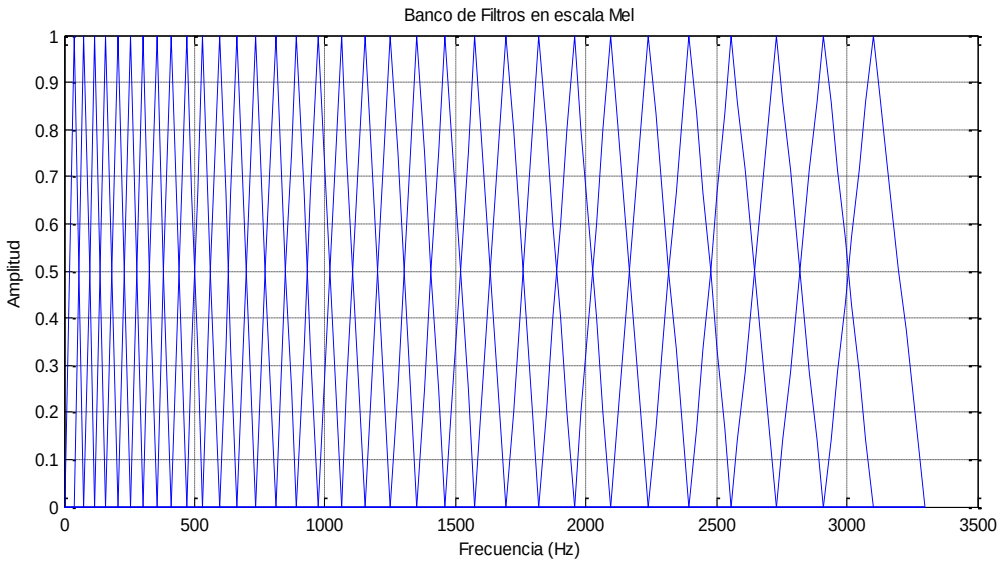


Como se puede observar en la Figura 18 el ancho de banda para los canales de bajas frecuencias es mucho más reducido, esto se debe a que la señal de voz tiene concentrada su energía en dichas frecuencias.

También se puede ver que la frecuencia máxima del banco de filtros es 3300 Hz, debido a que en etapas previas se aplico un filtro FIR pasa banda que limita el contenido frecuencial a este valor. Nótese que la base de cada triángulo está comprendida entre las frecuencias centrales de sus filtros adyacentes.

Al aplicar el banco de filtros a cada ventana del espectro de la señal, el vector resultante queda reducido al número de canales del banco de filtros aplicado (en este caso 33), con lo que se reduce considerablemente el tiempo de procesamiento. Cada valor de dicho vector representa el promedio de las componentes espectrales presentes por canal.

Figura 18. Banco de filtros en escala de Mel aplicado



3.3.2.3 Transformación no lineal. Una vez aplicado el banco de filtros en escala Mel a la primera trama de la señal enventanada, el paso a seguir es separar las dos componentes de la señal de voz (excitación y filtro variable).

Según lo estudiado… en la sección 4.3.1… el dominio espectral logarítmico es el óptimo para hacerlo ya que en dicho dominio ambos componentes son aditivos. Por tal razón, se calculó el logaritmo a la señal resultante de la aplicación del banco de filtros. Además, el logaritmo permite simular el comportamiento del oído y su sensibilidad ante distintas intensidades de presión sonora.

3.3.2.4 Transformada Discreta del Coseno (DCT). Los coeficientes cepstrales en escala de Mel son una variante de los coeficientes cepstrales dado a que a diferencia de lo señalado… en la sección 3.3.1… para calcular los MFCCs no se emplea la transformada discreta del Fourier sino que hace uso de la transformada discreta del coseno.

La aplicación de la DCT en lugar de la DTF se debe principalmente a que la transformación no lineal de la señal aplicada anteriormente entrega datos altamente correlacionados y con la aplicación de la transformada discreta del coseno debido a sus características se consigue los coeficientes más incorrelacionados. Además la transformada discreta del coseno se concentra en los coeficientes de más bajas *quefrencies* es decir en los componentes que provienen de la envolvente espectral.

3.3.2.5. Liftering. Con la aplicación de la transformada discreta de coseno se ha logrado separar las componentes de la señal de voz, solo queda seleccionar la información correspondiente al filtro que modela el tracto vocal y eliminar los representativos de la excitación.

Como se mencionó… en la sección 3.3.1…, el *liftering* es la técnica que permite conseguir dicho objetivo mediante la aplicación de la siguiente ventana:

$$w_{lift}(m) = 1 + \frac{N_c}{2} seno\left(\frac{\pi m}{N_c}\right)$$

Donde:

N_c es el número de coeficientes deseados por ventana y m = 1,2,3,... N_c. El resultado de la aplicación del *lifter* a cada ventana entrega como resultado un vector de coeficientes cepstrales de tamaño N_c.

Es común el empleo de 20 coeficientes MFCCs en ASR, aunque el rango de 10-12 coeficientes es a menudo considerado suficiente para codificar el habla[7]. El número de coeficientes cepstrales que se tomaron para esta aplicación es 12, este valor fue seleccionado con base a diferentes pruebas realizadas.

Inicialmente se calcularon los coeficientes cepstrales en escala Mel para cada una de las tramas de la señal enventanada como se describe en esta sección.

Debido a que el número de ventanas de la señal no es igual en todas las muestras ya que existen palabras de más duración que otras o que se pronuncian más lentamente lo que ocasiona que el tamaño de la información de la señal de voz (eliminando silencios iníciales y finales) sea variable.

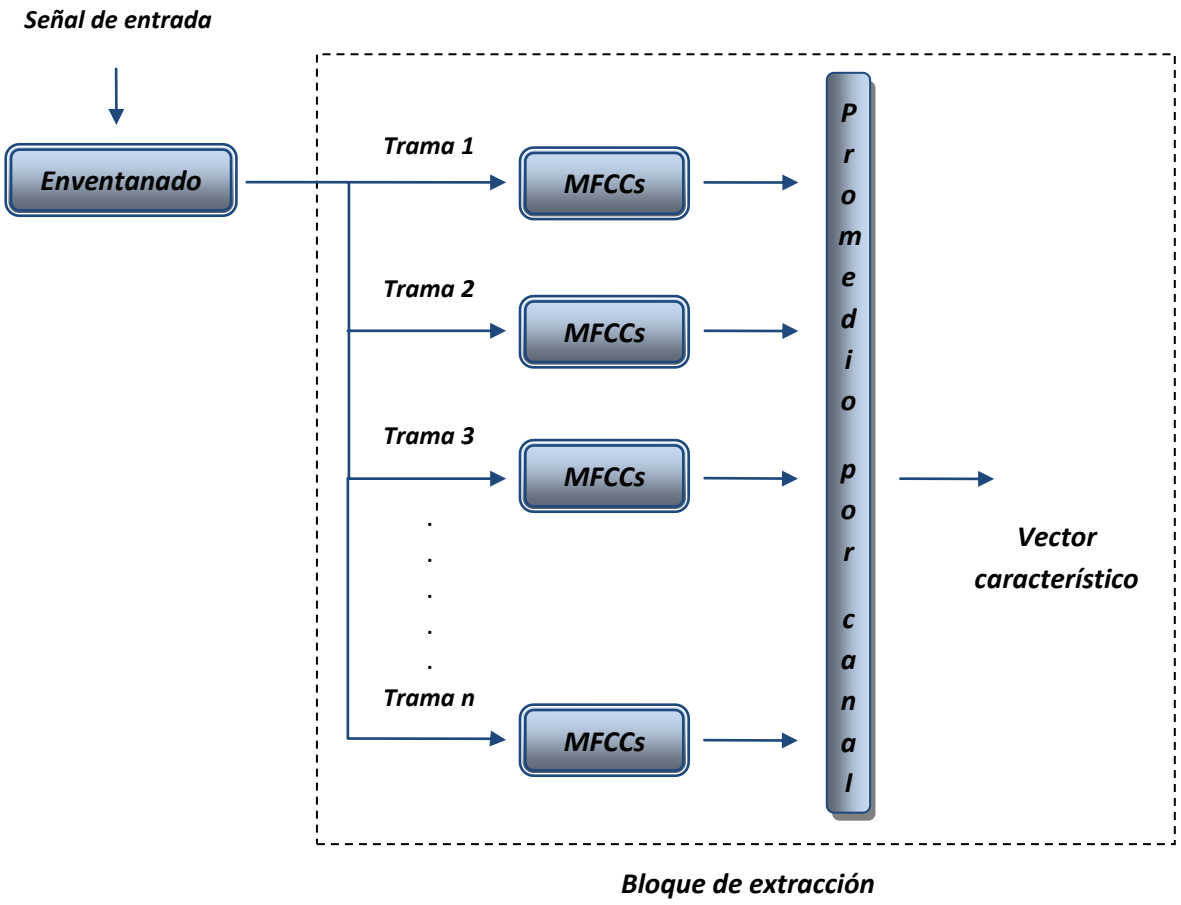
Como se desea obtener un vector que represente de la forma más precisa y reducida la señal; lo que se hizo fue calcular la media para cada uno de los coeficientes cepstrales en escala Mel de los diferentes segmentos de la señal enventanada como se esquematiza en la Figura 19., obteniendo de esta manera el vector característico de la muestra analizada de tamaño N_c.

En el análisis de resultados se verá como esta forma de extracción del vector característico permitió conseguir un resultado aceptable en cuanto al rendimiento del algoritmo de reconocimiento, pero en

[7] RÄSÄNEN, Okko. Speech Segmentation and Clustering Methods for a New Speech Recognition Architecture. Master's thesis of Science in Technology. 2007.

búsqueda de mejores resultados se optó por dividir la señal enventanada en 5 partes antes de calcular los MFCCs, con el objetivo de realizar una mejor representación de la señal analizada.

Figura 19. Diagrama esquemático del bloque de extracción de características

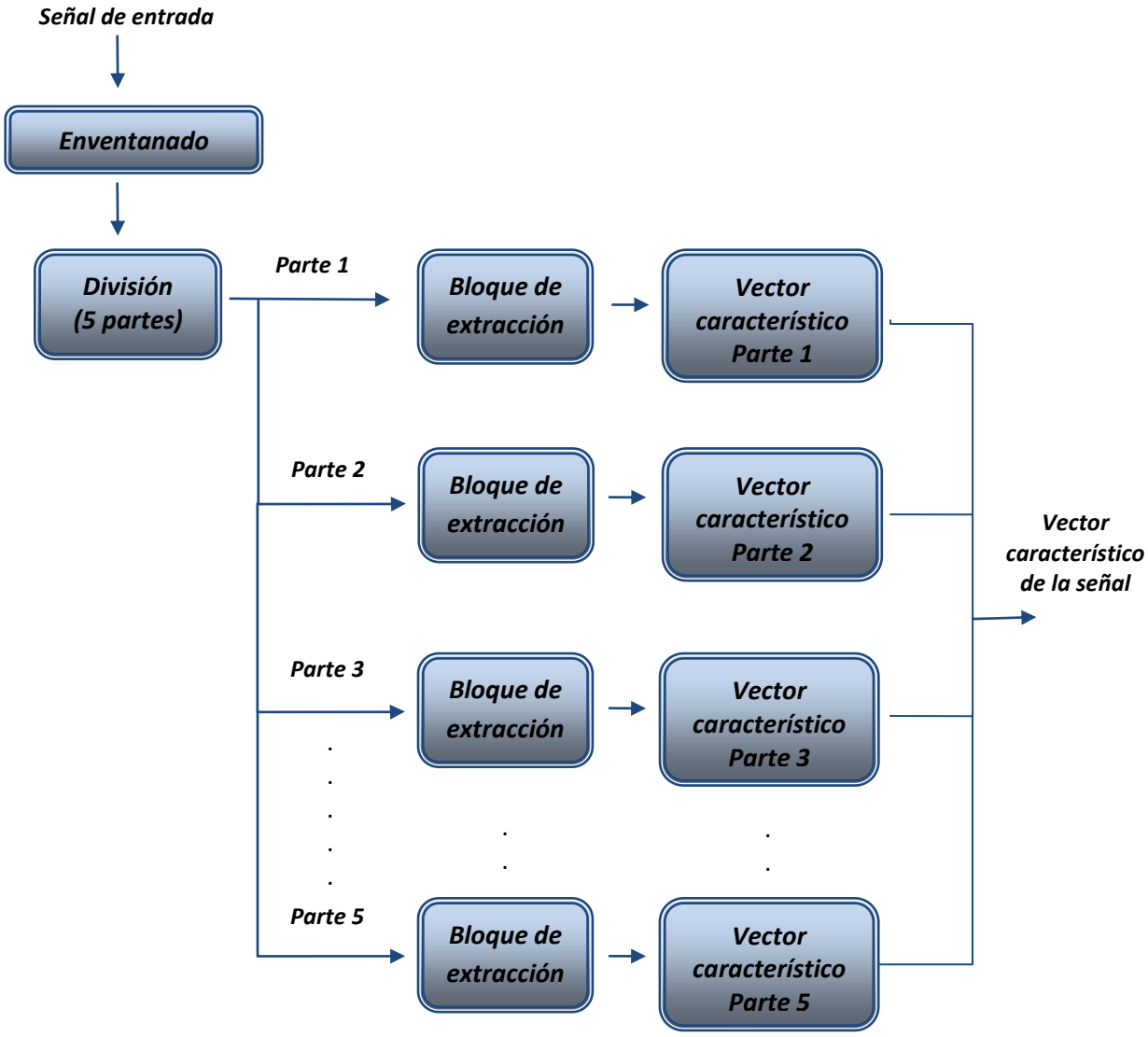


De esta manera para cada una de las partes se obtendrá un vector característico de la parte de tamaño N_c como se ilustra en la Figura 20.

Pero el vector característico que representa finalmente la muestra es la reunión de los vectores característicos de cada una de las partes, es decir un vector de tamaño $5N_c$.

Aunque con este procedimiento se incrementa el tamaño del vector característico representativo de la muestra analizada y con ello el tiempo de procesamiento, se eligió sacrificar lo anterior a cambio de mejorar la tasa de rendimiento del algoritmo de reconocimiento.

Figura 20. Diagrama esquemático del proceso de extracción de características realizado

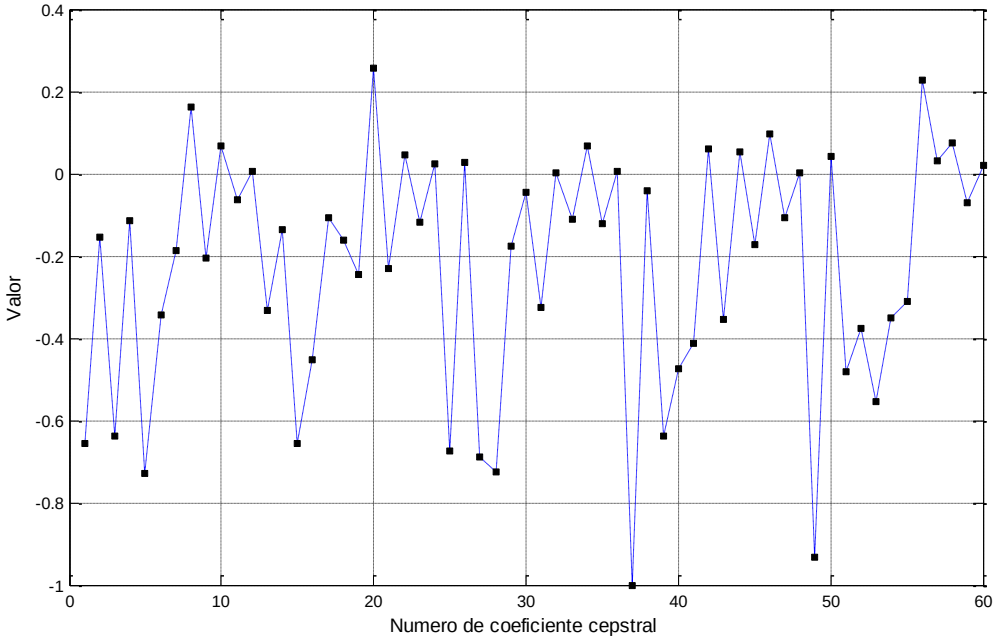


En la Figura 21 se puede observar el vector característico para una de las palabras empleadas en esta aplicación. Una vez extraído el vector que representa la palabra se procede a realizar la etapa de clasificación la cual se explica seguidamente.

3.4 CLASIFICACIÓN

Esta es la etapa final del algoritmo de reconocimiento que permite determinar la efectividad del mismo. Hasta el momento se ha realizado un proceso de extracción de características que buscó seleccionar los patrones más representativos de la señal para que estos puedan ser clasificados adecuadamente. Existen diversos métodos de clasificación de señales de voz entre los cuales se encuentran los HMMs, la transformada wavelet y las técnicas de inteligencia artificial.

Figura 21. MFCCs para una muestra de la palabra “rojo”



Para la categorización de los vectores característicos se optó por emplear una Red Neuronal Artificial (RNA), la cual es una de las metodologías más utilizadas para la clasificación y reconocimiento de patrones debido principalmente a su gran rapidez y efectividad de reconocimiento.

La gran ventaja de las redes neuronales es su capacidad de aprender a partir de variables que identifican el problema, extrayendo los datos necesarios para generar un modelo y una red capaz de resolverlo y, sobre todo, partiendo de un conocimiento mínimo de la esencia del problema[8]. Antes de entrar a detallar la estructura de la RNA empleada se hará una breve descripción de las características y funcionamiento de este método de clasificación que ayudaran a comprender posteriormente el diseño de la misma.

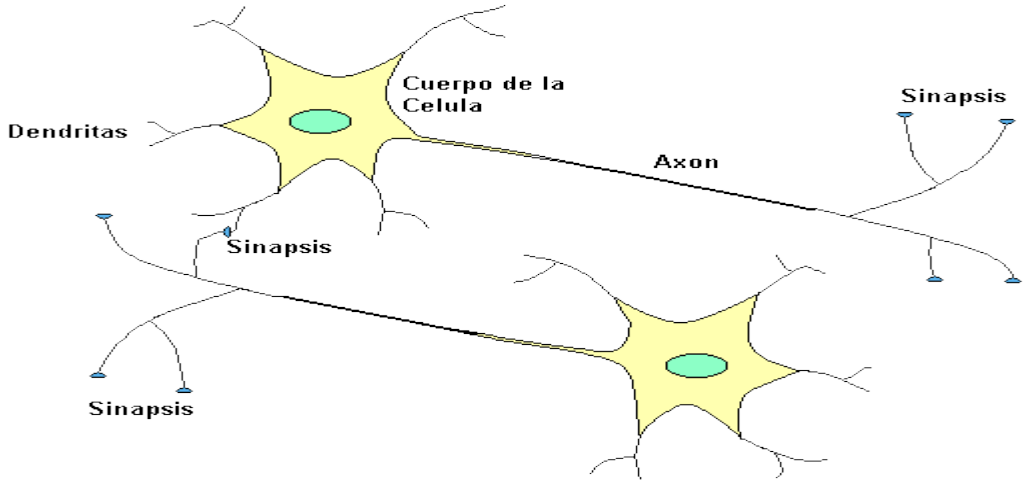
3.4.1 Redes Neuronales Artificiales (RNAs). Mediante esta técnica se intenta imitar el proceso de aprendizaje del cerebro humano. El cerebro esta formado por miles de millones de neuronas conectadas entre si. Utiliza información que es percibida, transmitida hasta las neuronas y alli procesada por ellas para dar respuesta a cada uno de los diferentes estimulos.

Cada neurona tiene 3 partes: un cuerpo celular, una estructura de entrada (dendrita) y una de salida (axón), como se muestra en la Figura 22. La mayoría de las terminales de los axones se conectan con las otras dendritas de otras neuronas lo que se conoce como sinapsis. El comportamiento de una neurona es el siguiente: recibe una señal de entrada con una fuerza determinada, dependiendo

[8] SHAVLIK, J.W., MOONEY, R.J., TOWELL, G.G. Symbolic and neural learning algorithm: An experimental comparison. Machine Learning. 1991. Vol. 6, p. 111-143.

de ellas la neurona emite una señal de respuesta, las sinapsis pueden variar de fuerza, algunas pueden dar una señal débil y otras una fuerte. A una neurona pueden llegar miles de señales de entrada, cada una con una fuerza o peso determinado. En consecuencia una red neuronal es la asociación de dos o más neuronas para desempeñar una tarea específica.

Figura 22. Diagrama de interconexión de dos neuronas



Tutorial redes neuronales. Universidad tecnológica de Pereira.

3.4.1.1 Modelo básico de la neurona artificial. En esta sección se estudiará el modelo matemático que describe el comportamiento de una neurona artificial también denominada unidad de proceso que es el elemento fundamental de una RNA; cuya función, simple y única, consiste en recibir las entradas de células vecinas y calcular un valor de salida.

Este modelo fue establecido por el grupo de investigación de procesamiento paralelo distribuido de la universidad de California en San Diego[9] y se describe de la siguiente manera: las entradas de la neurona, son ponderadas (atenuadas o amplificadas) a través de un peso positivo (excitatorio) o negativo (inhibitorio). Si la suma de estas entradas ponderadas es mayor o igual que el umbral de la neurona, entonces la neurona se activa. Una vez se ha calculado la activación del nodo, el valor de salida se obtiene al aplicar una función de activación dependiente de las características específicas de cada red. La Figura 23 se muestra el esquema básico de una neurona artificial, también conocida como perceptrón, que se puede expresar matemáticamente como:

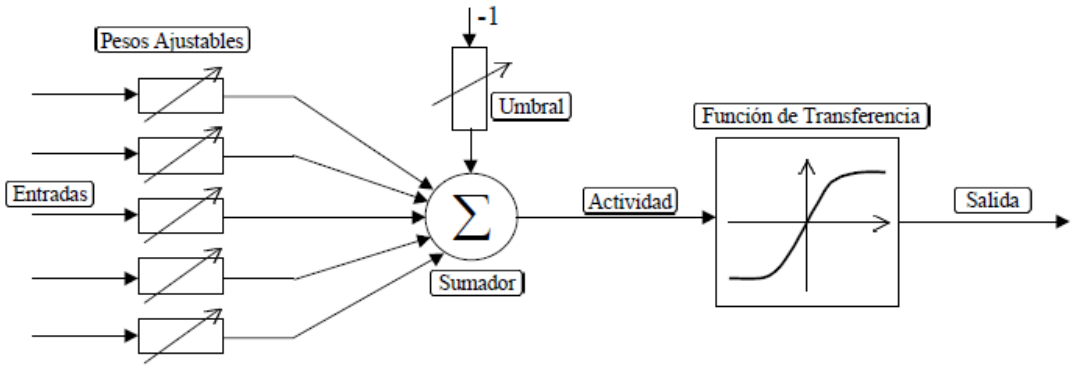
$$y = f\left(\sum_{i=1}^{n} w_i\, x_i - \theta\right)$$

[9] SANZ MOLINA, Alfredo y MARTIN DEL BRÍO, Bonifacio. Redes Neuronales y Sistemas Difusos. Segunda edición. Universidad de Zaragoza. p. 12.

Donde:

i es el número de entradas de la red neuronal.
f es la función de transferencia que determina el nivel de salida de la red neuronal.
w son los pesos para cada una de las entradas.
x son las entradas de la red neuronal.
θ es el umbral que la neurona debe sobrepasar para activarse.

Figura 23. Esquema de una neurona artificial



Introducción a las redes neuronales. ISA – Ingeniería de sistemas y automática.

3.4.1.2 Características generales. Dentro de las principales características de las redes neuronales artificiales encontramos las siguientes:

- Pesos: Las redes neuronales pueden tener factores de peso adaptable o fijo. Las que tienen pesos adaptables emplean leyes de aprendizaje para ajustar el valor de la fuerza de interconexión con otras neuronas. Si utilizan pesos fijos, su tarea debe estar previamente definida. Los pesos son determinados a partir de una descripción completa del problema a tratar.

- Topología: Cuando se efectúa un estudio de las RNAs en términos topológicos, se habla sobre la disposición de las neuronas de la red y la manera en que se encuentran distribuidas e interconectadas. Se suele distinguir entre las redes de una sola capa o redes monocapa y las redes con múltiples capas o redes multicapa.

  Las redes monocapa establecen conexiones laterales, cruzadas o auto recurrentes entre las neuronas que pertenecen a la única capa que constituye la red. Se utilizan en tareas relacionadas con lo que se conoce como auto asociación; por ejemplo, para generar informaciones de entrada que se presentan a la red incompletas o distorsionadas. Mientras que las redes multicapa disponen de conjuntos de neuronas agrupadas en varios niveles o capas.

- Tipo de conexión: Se distinguen las redes *feedforward*, en las cuales todas las señales se propagan hacia adelante a través de las capas de la red y las redes *feedback* donde las salidas

de las neuronas de capas posteriores se conectan a las entradas de capas anteriores. Este tipo de conexiones dan origen a redes *feedforward/feedback* en las cuales la información circula tanto hacia adelante como hacia atrás.

- Aprendizaje: Se utilizan dos tipos de aprendizaje: supervisado y no supervisado. En el primero se le proporciona a la red tanto la salida como la entrada correcta, y la red ajusta sus pesos para disminuir el error en la salida que ella calcula. Este tipo es utilizado principalmente en el reconocimiento de patrones. En el aprendizaje no supervisado a la red se le proporcionan únicamente los estímulos y la salida es calculada por la red. La forma de aprendizaje empleada depende del tipo de problema que se intenta resolver.

- Fases de operación: Se presentan dos fases en la operación de la red neuronal artificial: entrenamiento y recuperación o validación de lo aprendido. En la primera fase se le proporcionan estímulos de entrada y salida (según el caso), para que la red ajuste sus pesos de interconexión y minimice el error en la salida que calcula. En la segunda fase la red solamente calcula la respectiva salida.

- Necesitan un patrón: Las redes neuronales no son capaces de reconocer nada que no tenga algún tipo de patrón. Son muy buenas resolviendo problemas de asociación, evaluación y reconocimiento de patrones.

3.4.1.3 Backpropagation. Aunque se han elaborado hasta el momento una gran cantidad de modelos de RNAs, algunos más especializados que otros, solo se va a estudiar la red backpropagation que fue la red empleada para realizar la clasificación de los vectores característicos.

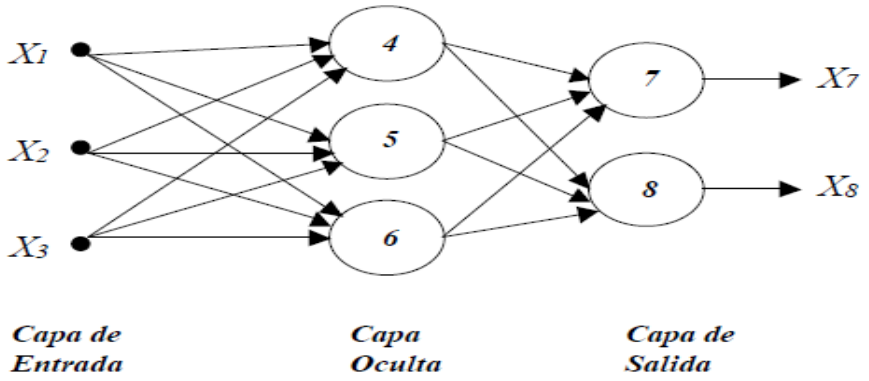
Cuando se combinan varios perceptrones en una capa y los estímulos de entrada después son sumados, se tiene ya una red neuronal. La falta de métodos de entrenamiento apropiados para los perceptrones multicapa (MLP) hizo que declinara el interés en la redes neuronales en los años 60 y 70. Esto no cambio hasta la reformulación del método de entrenamiento para la MLP backpropagation a mediados de los 80 por Rumelhart en 1986. En esta red, se interconectan varias unidades de procesamiento en capas, las neuronas de cada capa se conectan entre sí. Cada neurona de una capa proporciona una entrada a cada una de las neuronas de la siguiente capa como se muestra en la Figura 24. El termino backpropagation se refiere al método para calcular el gradiente de error en una red de aprendizaje supervisado, que es la aplicación de la regla de la cadena de cálculo fundamental. Básicamente el entrenamiento de este tipo de red consiste en lo siguiente:

- Pasada hacia adelante: Las salidas son calculadas y el error en las unidades de salida es calculado.

- Pasada hacia atrás: El error de las salidas es utilizado para alterar los pesos de las unidades de salida. Luego el error en las neuronas de las capas ocultas es calculado mediante propagación

hacia atrás del error en las unidades de salida y los pesos en las capas ocultas son alterados usando esos valores.

Figura 24. Modelo de una red backpropagation



Apéndice D. CARDONA ARBOLEDA, Omar Darío

3.4.2 Estructura de la red neuronal para el reconocimiento de palabras. La elección del tipo red, el número de neuronas de cada una de las capas que la componen, así como las funciones de activación para cada una de ellas; fue un proceso extenso en el que se efectuaron diferentes pruebas con el objetivo de elegir la estructura de la red que permitiera conseguir los mejores resultados. Dicho proceso se describirá con detalle en el análisis de resultados. En esta sección solo se mencionará la estructura final elegida con base en las pruebas realizadas. Por razones que se explicarán en el análisis de resultados se decidió dividir las palabras en dos grupos incluyendo en cada uno ellos las palabras correspondientes a los comandos de movimiento y color (Ver Tabla 3), teniéndose de esta manera 2 redes neuronales, una para cada grupo.

Aunque el objetivo inicial de este proyecto era realizar un sistema de control dependiente del hablante (entrenamiento para cada niño), con la adquisición de muestras de las diferentes palabras se optó por diseñar un sistema independiente del hablante; en el que los niños simplemente pronunciaran las acciones que desea que el robot ejecute, sin tener que introducir muestras para efectuar el entrenamiento de la red. Pero como se estudiará en el análisis de resultados este sistema no presentó la tasa de rendimiento más favorable debido a las dislalias que presentan algunos niños, el cual fue uno de los factores que más incidió en el rendimiento final de algoritmo de reconocimiento. Para solucionar este inconveniente se eligió emplear los dos sistemas dependiente e independiente, de tal manera que los niños que presentan dislalias utilicen el sistema dependiente para evitar errores con el sistema independiente.

Dado que el tamaño del vector característico que se obtuvo finalmente es de tamaño $5N_c$ como se mencionó…en la sección 3.3.2.5…la red neuronal que llevará a cabo el proceso de clasificación tendrá 60 neuronas en la capa de entrada. El número de neuronas de la capa de salida depende del número de palabras que la RNA debe reconocer, es decir, variará dependiendo si se va a reconocer un comando de movimiento (7 neuronas) o color (5 neuronas).

Tabla 3. División de las diferentes palabras comando

Tipo de comando	***Movimiento***	***Color***
Palabras	"abajo", "adelante", "arriba", "atrás", "derecha", "izquierda", "parar".	"azul", "blanco", "negro", "rojo" "verde".
Número total de palabras	7	5

La red que se empleó para realizar la clasificación de los diferentes vectores característicos es una *feedforward backpropagation* y algoritmo de entrenamiento empleado fue *resilient backpropagation,* dado que este es un algoritmo diseñado para redes de gran tamaño, con convergencia rápida y mínimo requerimiento de almacenamiento que opera en el modo batch y por estas razones es útil para el reconocimiento de patrones o funciones de clasificación[10]. En cuanto al número de capas ocultas y número de neuronas correspondientes a cada una de ellas, después de múltiples pruebas se determinó que los mejores rendimientos se obtenían con una sola capa oculta compuesta por 200 neuronas. Las funciones de activación empleadas para la capa oculta y de salida fueron *radbas* y *tansig*, respectivamente. En la Tabla 4, se muestran las estructuras de las redes de movimiento y color.

Tabla 4. Características de la RNAs empleadas

Red	***Capa***	***Número de neuronas***	***Función de transferencia***
Movimiento	*Entrada*	60	Lineal
	Oculta	200	Radbas
	Salida	7	Tansig
Color	*Entrada*	60	Lineal
	Oculta	200	Radbas
	Salida	5	Tansig

Esta estructura es igual tanto para el sistema dependiente como el independiente del hablante. Cuando el usuario seleccione en la interfaz gráfica la opción nuevo usuario, le pedirá al mismo que introduzca un número determinado de muestras por palabras (mínimo tres), con el cual se entrenará la RNA y posteriormente se podrá controlar el robot. En caso de que se elija la opción anónimo, la aplicación desarrollada le solicitará al niño que pronuncie la palabra de la acción que desea que ejecute el robot, sin realizar ningún entrenamiento previo; ya que con esta opción el programa cargará a la red neuronal los pesos obtenidos en el entrenamiento del sistema independiente.

[10] OTERO BARREIRO, Adriana Sofía y TRUJILLO LEMUS, Gustavo Adolfo. Sistema para el reconocimiento óptico de caracteres alfanuméricos en placas de automóviles particulares. 2005. p. 100-104. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

4. LEGO MINDSTORMS NXT

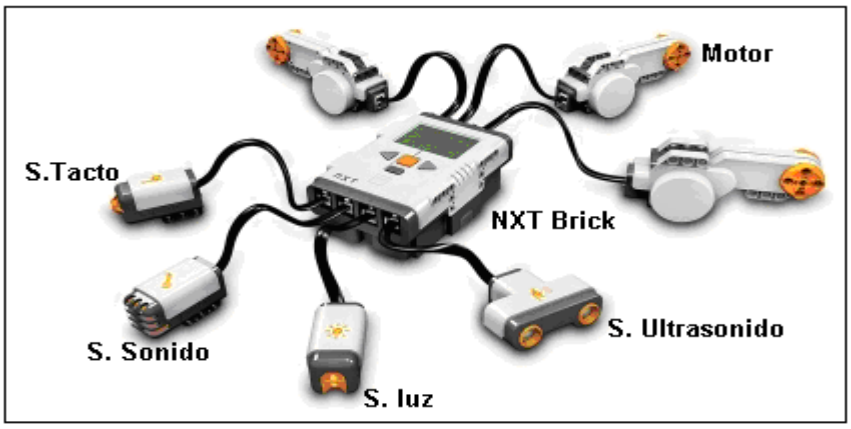
En este capítulo inicialmente se estudiará el hardware del LEGO Mindstorms NXT, que fue la herramienta robótica empleada para la realización de este proyecto, luego el estudio se centrará en el módulo bluetooth que presenta este robot y que permitió la transmisión de los datos del computador al robot; para finalmente describir el diseño del sensor que se implementó para realizar la detección de colores.

Como se mencionó… en el capítulo 1…la empresa danesa LEGO lanzó al mercado unos robots con fines educativos. A través del tiempo estos han evolucionado hasta convertirse en poderosas maquinas con múltiples sensores e interfaces de comunicación. Entre estos encontramos el LEGO Mindstorms NXT.

El Mindstorms NXT de Lego incluye tres servomotores, sensor de sonido, sensor de luz, sensor de tacto, sensor de luz, Bluetooth, USB, pantalla de 100×64 píxeles y un parlante (Ver Figura 25). Adicionalmente el Mindstorms incluye 519 piezas didácticas para construir humanoides, insectos y autos entre otros. La pieza principal del Mindstorms NXT es el Brick NXT (Bloque principal) es un procesador ARM7 de 32 bit, en el cual corre un sistema operativo de tiempo real (RTOS) llamado LegOS, este se encarga de administrar la ejecución de los programas y las comunicaciones USB y Bluetooth a través de las cuales se puede controlar el Mindstorms en su totalidad el desde un computador o celular.

El Mindstorms NXT resulta sumamente interesante tanto para novatos como programadores experimentados debido a que LEGO ha pensado en cada persona, es decir así como ha desarrollado un magnífico entorno de programación visual, ha liberado el hardware-firmare y los protocolos de comunicación entre estos. De esta manera se puede controlar desde muchos lenguajes de programación como VS.NET, MATLAB, LABVIEW.

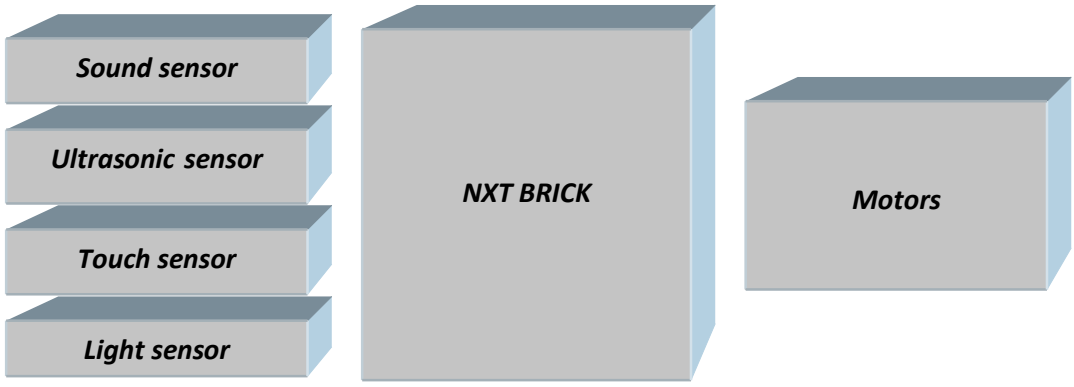
Figura 25. LEGO Mindstorms NXT



4.1 HARDWARE

En la Figura 26 se muestra un diagrama de bloques de alto nivel del Mindstorms, en el se pueden apreciar los sensores, actuadores y el bloque de control. Para conocer la arquitectura del LEGO Mindstorms NXT se estudiará cada uno de los bloques, empezando con los sensores, luego los actuadores y finalmente se analizará el bloque de control (NXT Brick). La fuente de información principal para estudiar el hardware del Mindstorms es el paquete de documentos llamado "*LEGO MINDSTORMS NXT Hardware Developer Kit*", este paquete se puede descargar desde la página oficial de *mindstorms* http://www.lego.com.

Figura 26. Diagrama de bloques de alto nivel

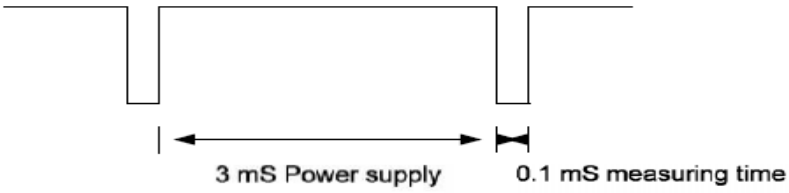


4.1.1 Sensores. El LEGO Mindstorms NXT incluye sensores de sonido, ultrasonido, luz y toque, los cuales se comunican de forma diferente con el Brick NXT dependiendo que clase de sensor sea. Se analizarán las clases de sensores, prestando especial atención a la interface (Sensor - NXT BRICK) más que al principio de funcionamiento del sensor en particular. Este análisis es de gran importancia ya que permitirá desarrollar nuevos sensores y así ampliar las capacidades del robot. Los sensores se clasifican en tres grupos activos, pasivos y digitales; esta clasificación se hace de acuerdo a la forma en que el sensor se comunica con el NXT Brick y los cuales se explican seguidamente.

4.1.1.1 Sensores activos. El LEGO Mindstorms NXT no incluye este tipo de sensores, sin embargo se puede manejar por compatibilidad con versiones anteriores, se llaman activos porque requieren una señal de alimentación especial, y tienen un diagrama de tiempos especial para la medición. El NXT BRICK tiene un generador que controla la energía entregada al sensor activo, este alimenta por 3 ms el sensor y luego mide el valor análogo durante los siguientes 0.1 ms como se muestra en la Figura 27.

4.1.1.2 Sensores pasivos. Todos los sensores que no necesitan un diagrama de tiempos especial como los sensores activos son llamados sensores pasivos, el valor entregado por estos sensores es muestreado cada 3 ms, entre los sensores pasivos encontramos los siguientes: sensor de toque, sensor de luz, sensor de sonido y el sensor de temperatura.

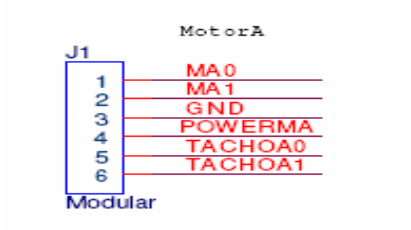
Figura 27. Diagrama de tiempos para un sensor activo



4.1.1.3 Sensores digitales. Son sensores que usan comunicación I2C para comunicarse con el NXT BRICK. Se llaman digitales porque tienen un microcontrolador que se encarga de la comunicación I2C y la medición de la variable deseada. El sensor ultrasónico del LEGO Mindstorms NXT es un sensor digital.

4.1.2 Actuadores. El LEGO Mindstorms NXT tiene tres salidas para controlar tres motores conectados al NXT BRICK, cada motor tiene una interfaz digital de 6 pines tal como se muestra en la Figura 28.

Figura 28. Diagrama del conector del motor A



MA0 : Entrada, PWM para controlar el motor. Hasta 700mA.
MA1 : Entrada, PWM para controlar el motor. Hasta 700mA.
GND : Tierra del circuito.
POWERMA : Voltaje de alimentación 4.3V
TACHOA0 : Salida, Señal de encoder.
TACHOA1 : Salida, Señal de encoder.

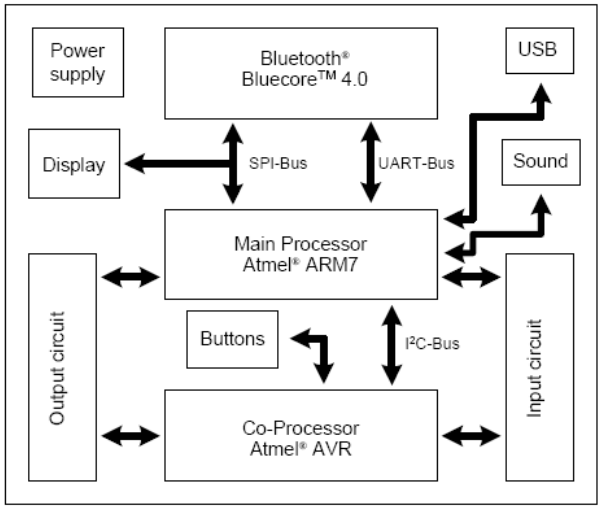
En resumen NXT BRICK proporciona dos señales de PWM: MA0 y MA1 las cuales pueden manejar una corriente de hasta 700mA, y dos entradas digitales TACHOA0 y TACHOA1. Cada actuador cuenta con: dos entradas que permiten controlar la velocidad y sentido de giro del motor y dos señales de salida de un encoder en cuadratura con las cuales se puede medir la velocidad y sentido de giro del motor.

4.1.3 NXT brick. Este es el circuito de control central del Mindstorms, cuenta con las características que se muestran en la Tabla 5. En el diagrama de bloques de la Figura 29 se puede ver en forma general como están conectados y controlados los diferentes módulos dentro del NXT BRICK. A continuación se estudiaran las principales características de cada módulo.

Tabla 5. Características NXT brick

Procesador principal	ATMEL 32-Bits ARM - 48Mhz AT91SAM7S256, 256K Flash, 64KB RAM
Co- procesador	ATMEL 8-Bits - 8Mhz ATMega48, 4K Flash, 512B RAM
Modulo Bluetooth	CSR BlueCore V2.0 + EDR System - Soporta SPP "*Serial Port Profile*"
Modulo USB	Full Speed port 12Mbit/s
4 Puertos de entrada	Conector RJ-12, soporta entrada análoga y digital
3 Puertos de salida	Conector RJ-12, soporta entrada de encoders
Pantalla	100x64 píxeles monocromático
Parlante	Resolución 8Bit, Tasa de muestreo de 2-16 KHz
Botones	Cuatro botones
Baterías	Seis baterías AA

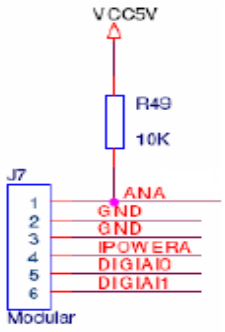
Figura 29. Diagrama de bloques del NXT BRICK



4.1.3.1 Puertos de salida. El NXT Brick tiene tres puertos de salida para controlar actuadores conectados a ellos. Estos puertos tienen una interfaz digital de seis líneas como se mencionó anteriormente.

4.1.3.2 Puertos de entrada. El LEGO Mindstorms NXT tiene 4 puertos de entrada que le permiten medir diferentes parámetros del mundo físico dependiendo de qué sensor esté conectado (luz, sonido o distancia). Los puertos de entrada tienen una interfaz de 6 líneas (Ver Figura 30), y permite entradas análogas y digitales lo cual da la posibilidad de construir sensores análogos y digitales para el Mindstorms NXT.

Figura 30. Puerto de entrada



ANA: Entrada análoga conectada a ADC de 10bits y una tasa de muestreo de 333Hz.
GND: Tierra del circuito.
IPOWERA: Fuente de 4.3V, puede suministrar un máximo de 180mA.
DIGITAIO, DIGITAI1: Son usados para comunicación I2C con sensores digitales.

4.2 TECNOLOGÍA BLUETOOTH

Antes de estudiar el modulo bluetooth con el que cuenta el NXT se hará una breve descripción de la tecnología bluetooth y su clasificación. La iniciativa Bluetooth se comenzó a principios de 1998 con un SIG promovido por Ericsson, IBM, Intel, Nokia y Toshiba bajo el nombre "Bluetooth" pero se hizo público hasta el 20 de Mayo del mismo año. Al transcurso de 2 meses, la primera especificación Bluetooth 1.0 fue liberada y en la actualidad se está trabajando en Bluetooth 2.0 con la colaboración de 3Com, Ericsson, IBM, Intel, Lucent Technologies, Microsoft, Motorola, Nokia y Toshiba. Como se puede apreciar, el número de compañías interesadas en el proyecto casi se ha duplicado y actualmente el SIG cuenta con 1883 miembros.

Bluetooth trabaja en dos capas del modelo OSI que son la de enlace y aplicación, incluye un transceiver que trasmite y recibe a una frecuencia de 2.4 Ghz. Las conexiones que se realizan son de uno a uno con un rango máximo que va desde 1 a 100 metros.

Los dispositivos de radio que soportan la tecnología no requieren de licencia y deben tener un espectro de 2.4 GHz para asegurar la compatibilidad en todo el mundo. Estos dispositivos usan *spread spectrum*, *frequency hopping* y *full-duplex signal* a más de 1600 hops/s. Además se pueden establecer y mantener más de seis conexiones simultáneas.

Bluetooth por cuestiones de seguridad cuenta con mecanismos de encriptación de 64 bits y autentificación para controlar la conexión y evitar que dispositivos puedan acceder a los datos o realizar su modificación.

Durante la transferencia de datos el canal de comunicaciones permanece abierto y no requiere la intervención directa del usuario cada vez que se desea transferir voz o datos de un dispositivo a otro. La velocidad máxima que se alcanza durante la transferencia es de 700 kb/s.

4.2.1 Clasificación de los dispositivos bluetooth. Los dispositivos bluetooth se pueden clasificar según diferentes parámetros, entre ellos el alcance y el ancho de banda. Como se puede ver en la Tabla 6 la clasificación según el alcance toma como referencia la potencia de transmisión del dispositivo.

Tabla 6. Clasificación de dispositivos bluetooth según la potencia de transmisión

Clase	Potencia máxima permitida(mW)	Potencia máxima permitida(dBm)	Rango (aproximado)
I	100 mW	20 dBm	~100 metros
II	2.5 mW	4 dBm	~20 metros
III	1 mW	0 dBm	~1 metro

Tabla 7. Clasificación de dispositivos bluetooth según el ancho de banda

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 + EDR	3 Mbit/s
UWB Bluetooth (propuesto)	53 - 480 Mbit/s

4.3 MODULO BLUETOOTH. Un perfil Bluetooth es la especificación de una interfaz de alto nivel para su uso entre dispositivos Bluetooth. Para utilizar una cierta tecnología Bluetooth un dispositivo deberá soportar ciertos perfiles.

Los perfiles son descripciones de comportamientos generales que los dispositivos pueden utilizar para comunicarse, formalizados para favorecer un uso unificado. La forma de utilizar las capacidades de Bluetooth se basa, por tanto, en los perfiles que soporta cada dispositivo. El perfil que maneja el Lego Mindstorms NXT es el SPP (Serial Port Profile) el cual emula una línea serie y provee una interfaz de reemplazo de comunicaciones basadas en RS-232, con las señales de control típicas.

El NXT Brick soporta comunicación inalámbrica usando bluetooth, puede conectarse inalámbricamente a tres dispositivos pero solo puede comunicarse con uno a la vez. Igualmente el LEGO Mindstorms NXT puede comunicarse con otros dispositivos (palms, celulares, computadores) que estén programados para comunicarse utilizando el protocolo "*LEGO MINDSTORMS NXT Comunication Protocol*" y que soporten el modo SPP.

Para reducir el consumo de potencia del Bluetooth, este ha sido implementado usando un módulo Bluetooth Clase II, lo cual quiere decir que este puede comunicarse hasta una distancia máxima de 10 metros.

4.3.1 Protocolo de comunicación Lego Mindstorms NXT. El protocolo de comunicación LEGO Mindstorms NXT da la posibilidad de controlar el Mindstorms desde cualquier dispositivo que implemente el perfil SPP y este programado para utilizar el protocolo.

La descripción detallada del funcionamiento de este protocolo se encuentra en una serie de documentos llamada "LEGO MINDSTORMS NXT Bluetooth Developer Kit", la cual se puede descargar de la página oficial de LEGO. Este protocolo proporciona diferentes tipos de comandos, los cuales deben ser enviados sobre una conexión SPP en el orden en que se muestra en la Figura 31.

Figura 31. Estructura de un comando

Length, LSB	Length, MSB	Command Type	Command	Byte 5	Byte 6	Etc.

Se debe tener en cuenta que este protocolo es muy complejo y su estudio está fuera del alcance de este proyecto, por lo cual para controlar el Mindstorms vía bluetooth se utilizará alguna librería que lo implemente.

4.3.2 Librerías para controlar el NXT. Durante la etapa de documentación de este proyecto se encontró que existen muchas librerías que implementan el protocolo de comunicación Lego Mindstorms NXT y que brindan funciones de alto nivel para controlar el robot. Como la aplicación se realizó en MATLAB y VB.NET se explicarán las librerías disponibles para estos programas.

4.3.2.1 VS.NET. La librería NXT# es un proyecto de software libre desarrollado para controlar el Mindstorms NXT desde VS.NET, da un grupo de objetos que permiten de forma fácil manejar actuadores y sensores, este paquete puede ser descargado desde http://nxtsharp.fokke.net. Algunos de los objetos que proporciona esta librería son:

NxtBrick: Permite conectarse a un NXT BRICK.
NxtMotor: Brinda las funciones de control de un motor.
NxtSensor: útil para leer un sensor pasivo.
NxtSoundSensor: Especialmente diseñado para trabajar con el sensor de sonido.
NxtligthSensor: Diseñado para leer el sensor de luz.

4.3.2.2 MATLAB. El toolbox RWTH Mindstorms NXT, es un proyecto de software libre desarrollado por *Aachen University* .The RWTH - Mindstorms NXT Toolbox para MATLAB proporciona comunicación con el robot LEGO Mindstorms NXT a través de Bluetooth. El toolbox incluye rutinas que soportan la interacción entre el robot y MATLAB. Algunas de las funcionalidades del toolbox RWTH son: Abrir y cerrar conexiones bluetooth, enviar y recibir datos entre MATLAB y el robot, control de alto nivel para los motores y lectura de alto nivel para sensores entre otros. Este proyecto puede descargarse desde la página http://www.mindstorms.rwth-aachen.de.

4.4 REQUISITOS DE SOFTWARE/HARDWARE PARA CONTROLAR LEGO NXT MINDSTORMS.

En la Tabla 8 se resumen los requerimientos hardware y software necesarios para controlar el robot tanto para la librería NXT y RWTH. Se realizaron pruebas para verificar el funcionamiento de las librerías y se obtuvieron resultados satisfactorios, el hardware/software utilizado fue Microsoft Windows XP Profesional SP2, MATLAB 7.4.0.287 (R2007a), Visual Studio 2008 Express, el firmware del robot actualizado a la versión "LEGO Mindstorms NXT firmware v1.05" y se utilizó un módulo Bluetooth BlueSoleil de la empresa IVT Corporation que soporta los siguientes perfiles: PAN, Headset, AV,DUN, FTP, HID, Printer y SPP.

Tabla 8. Requisitos de software/hardware para controlar LEGO NXT Mindstorms

Sistema operativo	NXT# - Windows RWTH – Windows o Linux
Lenguaje de programación	NXT# - VS.NET RWTH – MATLAB Version 7.4 o superior
Firmware del LEGO	LEGO Mindstorms NXT firmware v1.05
Adaptador Bluetooth	Adaptador Bluetooth 2.0 con soporte SPP

4.5 SENSOR COLOR

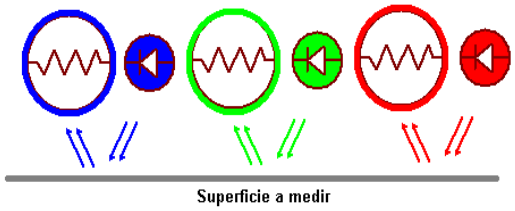
En un comienzo se pensó en llevar a cabo el reconocimiento de color utilizando para ello el sensor de luz que incorpora el LEGO Mindstorms NXT, pero luego de algunas pruebas se concluyó que este era incapaz de distinguir entre colores (rojo, verde, azul, blanco y negro), así que se decidió diseñar un sensor de color (R-G-B) en base al estudio realizado sobre la arquitectura de los sensores.

Se diseñó el sensor de modo que funcione como un sensor pasivo, es decir el sensor producirá una señal análoga la cual será medida por el NXT Brick y a partir de ella se podrá determinar el color a medir.

El principio de funcionamiento del sensor consiste en que, al incidir la luz blanca sobre un cuerpo este refleja solo algunas longitudes de onda, las cuales corresponden al color del objeto iluminado. Así por ejemplo el blanco refleja todas las longitudes de onda ya que este es una combinación de todos los colores, y el negro no refleja ninguna longitud de onda ya que el negro es la ausencia de color.

El transductor consta de tres diodos LEDs como se muestra en la Figura 32, los cuales emiten cada uno un color diferente (RGB) sobre la superficie a medir, igualmente hay tres fotorresistencias cada una con un filtro de color que mide la cantidad de luz reflejada para cada componente (RGB), en base a estas medidas resulta muy sencillo estimar el color o los colores que componen un determinado objeto.

Figura 32. Esquema básico del sensor de color



En la Figura 33 se muestra el diagrama de bloques del sensor diseñado y a continuación se detalla brevemente la función de cada uno de los bloques que lo componen:

- LEDs R G B: Se encargan de iluminar la superficie a medir con cada uno de los colores aditivos primarios (Rojo, Verde, Azul).
- Fotoresistores y filtro RGB: Su función es medir el nivel de cada componente RGB reflejado por la superficie.
- Microcontrolador: Se encarga de encender los LEDs, y medir los valores análogos entregados por cada fotocelda, luego los analiza y entrega un valor binario que representa el color de la superficie iluminada.
- Interfaz Lego NXT Mindstorms: Es un DAC que convierte el valor binario entregado por el microcontrolador a una tensión análoga la cual es leída por el ADC del NXT BRICK.

Para la implementación del sensor se empleo un microcontrolador PIC16F877A, el cual cuenta con un ADC de 10Bit. Se diseñaron dos circuitos el primero con los diodos RGB y las fotoceldas y el otro con el circuito de control que incluye un microcontrolador PIC16F877A y un DAC discreto compuesto por tres resistores de 1k, 2.2k y 4.7k respectivamente; como se muestra en el ANEXO B.

Figura 33. Diagrama de bloques del sensor de color

5. INTERFAZ GRÁFICA DE USUARIO

La interfaz gráfica de usuario diseñada va a ser la encargada de indicarle al niño cada una de las tareas que debe realizar para lograr controlar el robot por medio de su voz. Dicha interfaz debe ser amigable y sencilla dado que los usuarios de ella se encuentran en etapa preescolar. Esta interfaz fue diseñada en Visual Basic.NET. Antes de describir la GUI realizada, se especificará el diseño UML de la aplicación ya que este es un lenguaje gráfico que ayudará a comprender fácilmente la interfaz gráfica desarrollada.

5.1 DISEÑO DE LA APLICACIÓN EN UML

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. El Lenguaje Unificado de Modelado o UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

Aunque UML tiene gran variedad de diagramas para representar diferentes situaciones y puntos de vista, en el proceso de desarrollo de software tiene especial importancia: los diagramas de casos de uso y el diagrama de componentes.

5.1.1 Diagrama de casos de uso. Este diagrama muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Los elementos que pueden aparecer en un diagrama de casos de uso son: actores, casos de uso y relaciones entre casos de uso. Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palotes. Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica.

Para la aplicación que se desarrolló se han determinado tres actores: persona, usuario y administrador, igualmente se han identificado los casos de uso para cada actor, como se muestra en la Figura 34.

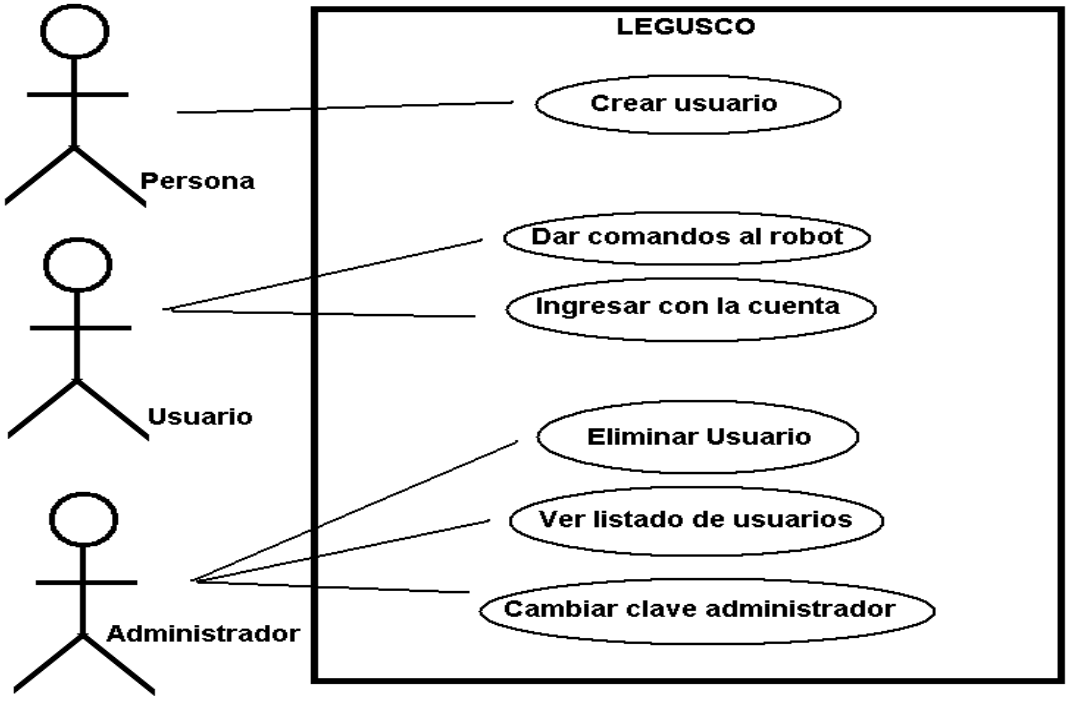
5.1.2 Diagrama de componentes. Este diagrama describe los elementos físicos del sistema y sus relaciones, muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, ejecutables, etc.

UML define cinco estereotipos estándar que se aplican a los componentes:

- Executable: Especifica un componente que se puede ejecutar en un nodo.

- Library: Especifica una biblioteca de objetos estática o dinámica.
- Table: Especifica un componente que representa una tabla de una base de datos.
- File: Especifica un componente que representa un documento que contiene código fuente o datos.
- Document: Especifica un componente que representa un documento.

Figura 34. Diagrama casos de uso LEGUSCO



Es muy común que algunos componentes requieran de otros para funcionar, por ejemplo cuando se declara una librería de enlace dinámico en una aplicación para usar sus funciones. En la Figura 35 se muestra el diagrama de componentes de la aplicación, teniendo en cuenta las librerías que se involucran en el proceso de implementación de la aplicación. Como se puede ver el diagrama de componentes no brinda ninguna descripción de que funciones ejecuta cada librería, esto se hará en la siguiente sección donde se describirá la interfaz grafica de usuario.

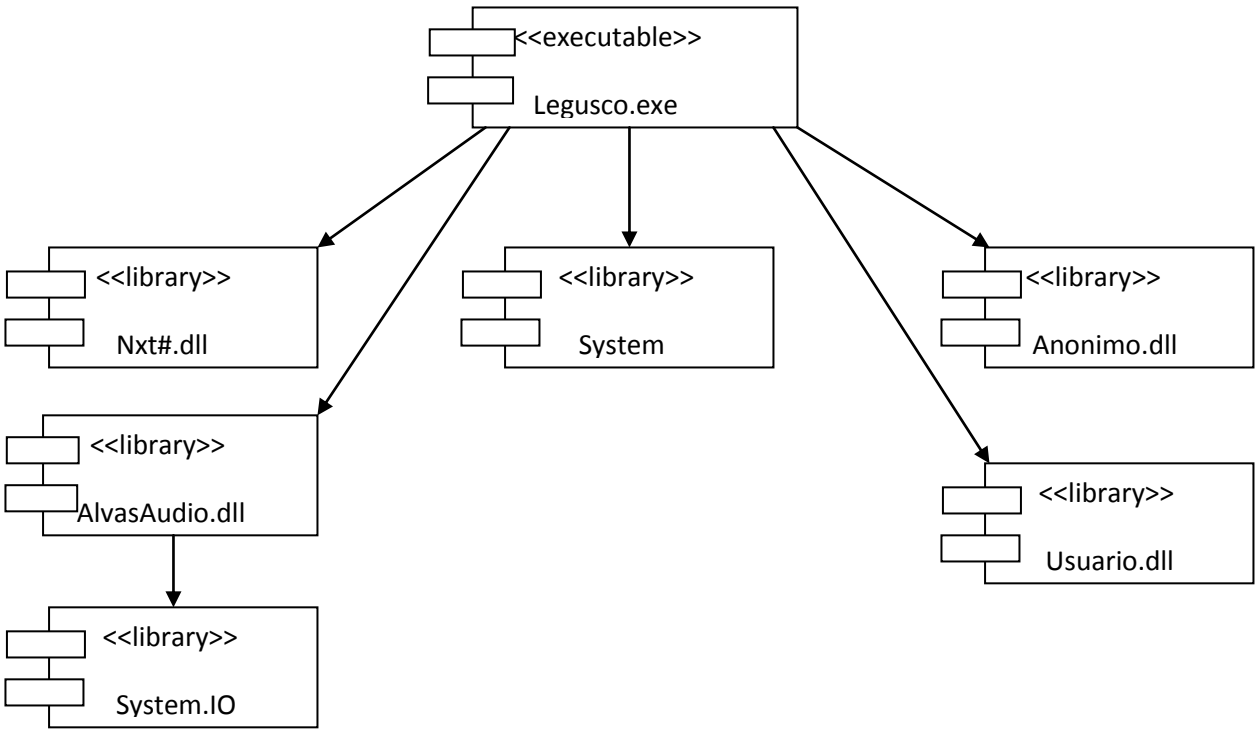
5.1 INTERFAZ GRAFICA EN VB.NET

Luego de realizar el modelado de la aplicación, el siguiente paso consiste en implementar la aplicación en algún lenguaje de programación, el lenguaje empleado debe permitir crear una interfaz grafica de usuario (GUI) muy amigable ya que los usuarios finales serán niños en la etapa preescolar (6 a 7 años), algunas de las características que debe cumplir la interfaz son las siguientes:

- Debe brindar un entorno de trabajo agradable para el niño.
- Debe ser altamente intuitiva ya que los niños a esta edad aun no han aprendido a leer.
- Debe ser altamente confiable, es decir debe ser un sistema estable y robusto frente a posibles errores.
- Debe ser altamente audio visual, para facilitar el entendimiento de esta por parte del niño.
- Debe diseñarse en un lenguaje de programación gratis/libre, para evitar el pago de licencias.

Sin embargo desarrollar la GUI altamente intuitiva (audio - visual) y en un lenguaje diferente al que se desarrollo el sistema de reconocimiento (MATLAB), inevitablemente trae algunos retos interesantes que se describen en las siguientes secciones.

Figura 35. Diagrama de componentes LEGUSCO



5.1.1 Formas de enlazar MATLAB y VB.NET. Para usar funciones de MATLAB desde un programa externo hay cuatro posibles soluciones. API de C de bajo nivel, DDE, COM y DLL.

5.1.1.1 COM. MATLAB implementa la funcionalidad de servidor COM, gracias a esta característica se puede usar la interoperabilidad COM de .NET para comunicar estos dos programas. Para crear el servidor COM de MATLAB, se debe crear un objeto de tipo "Matlab.Application", en VB.NET esto se hace de la siguiente manera:

```
m = CreateObject("Matlab.Application")
m.Execute("cmd");
```

Luego de crear el servidor COM, se puede enviar comandos y datos a MATLAB e igualmente leer los resultados, como se muestra en el siguiente ejemplo, en la cuarta línea el segundo parámetro "base" es el espacio de trabajo donde se encuentra la variable *a*. De acuerdo a la documentación, hay dos *workspaces*: base y global.

```
double [,] MReal;
double [] MImag;
Result = Matlab.Execute("a = [ 1 2 3 ; 4 5 6 ; 7 8 9];");
call m.GetFullMatrix("a", "base", MReal, MImag);
call m.PutFullMatrix("b", "....)
```

5.1.1.2 DDE. *Dynamic Data Exchange* es un servicio viejo pero poderoso de windows que permite que aplicaciones intercambien datos, como en COM, se tiene un servidor "MATLAB DDE Server" y el cliente que es el programa en .NET.

5.1.1.3 C API. El acceso directo a las API de MATLAB es la mayor solución en términos de rendimiento y características. Las librerías en C internamente usan Unix pipes o COM para comunicarse con la instancia principal de MATLAB, pero en este caso la transferencia de datos se realiza usando bloques de memoria y punteros, esto da la posibilidad de transferir eficientemente matrices de y hacia MATLAB.

5.1.1.4 DLL. Una de las características interesantes de MATLAB es la traducción de M-Files en DLL que puede ser usada para distribuir algoritmos de una forma eficiente, esta traducción se lleva a cabo por medio del comando *mcc* y permite crear librerías compatibles con C, C++ y .NET entre otros.

En base al estudio realizado se llego a la conclusión, que la mejor manera de enlazar .NET y MATLAB es creando una DLL ya que esta no requiere que MATLAB esté instalado, mientras que las otras opciones COM, DDE y C API si lo necesitan.

5.1.2 DLL en MATLAB. Como ya se había mencionado el comando *mcc* permite crear una DLL "*Dinamic Link Library*", sin embargo el compilador *mcc* tiene una limitación grave, no soporta objetos, esto quiere decir que solo se podrá compilar el código correspondiente a la extracción de

parámetros, los comandos *sim*, *train* y *newff* como trabajan y crean objetos no están soportados por el *mcc* (versión 7.4.0.287 (R2007a)).

Se explicará con un ejemplo sencillo de una función que nos devuelva el cuadrado mágico de n siempre y cuando n sea impar y mayor a cero, si no que devuelva una matriz de ceros; la forma de crear una DLL en MATLAB, como se muestra a continuación:

1. Crear el algoritmo

```
n=input('Valor de n: ');
if (n>=0)
   if (mod(n,2)==1); Respuesta=magic(n); else; Respuesta=zeros(n,n);end;
else
   n=abs(n); Respuesta=zeros(n,n);
end
```

2. Encapsular en una función el algoritmo

```
function [Respuesta] = Magico(n)
if (n>=0)
   if (mod(n,2)==1); Respuesta=magic(n); else; Respuesta=zeros(n,n);end;
else
   n=abs(n); Respuesta=zeros(n,n);
end
```

3. Compilar y empaquetar: Se puede usar *mcc* o *deploytool*, la diferencia es que *mcc* es el compilador y *deploytool* es un asistente para usar el *mcc*. Se escribe en la línea de comandos *deploytool*, se selecciona nuevo y en la ventana que aparece se elige la opción deseada, para esta aplicación *Matlab Builder for .NET* y *NET Component,* Ahora se agrega el archivo con Add File y se agregan las funciones deseadas en este caso el M-File, luego se da clic en el botón compilar y después en el botón empaquetar.

3. Usar la DLL desde VB.NET: Para usarla DLL en VB.Net lo primero que hay que hacer es agregar la una referencia al archivo que se muestra en la Figura 36. Luego se debe importar algunas librerías:

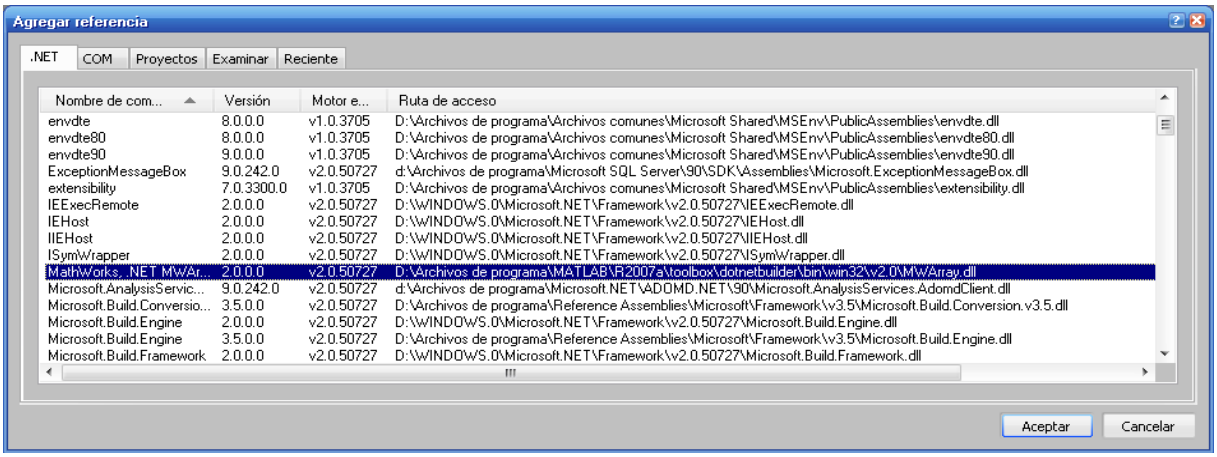
```
Imports System
Imports System.Reflection
Imports MathWorks.MATLAB.NET.Utility
Imports MathWorks.MATLAB.NET.Arrays
```

Ahora se hace el enlace con el siguiente codigo:

```
Dim arraySize As MWNumericArray = Nothing
Dim magicSquare As MWNumericArray = Nothing
```

```
Dim Tammagic As String = "1"
arraySize = New MWNumericArray(System.Int32.Parse(Tammagic), False)
Dim magic As MagicDemoComp.MagicDemoCompclass = New _
MagicDemoComp.MagicDemoCompclass
magicSquare = magic.makesquare(arraySize)
Dim nativeArray(,) As Double = _ CType(magicSquare.ToArray(MWArrayComponent.Real),
Double(,))
Dim index As Integer = arraySize.ToScalarInteger()
For i As Integer = 0 To index - 1
        For j As Integer = 0 To index - 1
        MsgBox(nativeArray(i, j).ToString)
        Next j
Next i
```

Figura 36. Ventana de referencias VB.NET Express 2008



5.1.3 Manejo del NXT desde .NET. Para manejar el LEGO desde .NET, se usó la librería NXT#, algunos de los objetos que brinda esta son: NXTBrick, NXTmotor y NXTSensor.

- NXTBrick: Permite enlazar con el bloque principal, esto se lleva a cabo por medio de la propiedad **COMPortName** y el método **Connect.** Donde **COMPortName** es el número del puerto serial virtual "SPP" al que está conectado el Brick NXT.

- NXTMotor: Permite controlar un motor a la vez, requiere conectarse a un control NXTBrick, las propiedades y métodos más importantes son *Brick*, *Port*, *Turn* y *brake*. *Brick* permite especificar a qué objeto NXTBrick se conectará el NXTMotor, *Port* sirve para especificar en qué puerto de LEGO está conectado el motor a controlar; mientras *Turn* permite hacer girar el motor y *brake* permite parar el motor.

- NXTSensor: Al igual que motor tiene las propiedades NxtBrick y Port, permite leer el valor con la función *Poll.*

5.1.4 Grabar y reproducir sonidos desde .NET. Para grabar y reproducir sonidos en .NET se utilizó la librería *Alvas.Audio.Net*, de esta librería se usaron los objetos: *recorder* y *player*.

- Recorder: Permite grabar sonidos especificando la resolución (8 o 16 Bits), tasa de muestreo (8000, 11025, 22050 y 44100) y los canales (monofónico o estéreo).

- Player: Permite reproducir archivos de diferentes formatos, el archivo se indica en la propiedad *FileName* y con los métodos *play*, *pause* y *stop* se controla la reproducción.

5.1.5. Entrenamiento y simulación de la RNA desde MATLAB. Para llevar a cabo el proceso de entrenamiento y simulación de RNA desde MATLAB se encontraron tres soluciones:

- Como se mencionó… en la sección 6.1.2…*mcc* no es compatible con *newff*, *train* y *sim*, así que no se pueden compilar, sin embargo COM y DDE permite usar cualquier comando de MATLAB, ya que lo que se hace con estos métodos es iniciar una instancia de MATLAB y comunicarlos. Sin embargo la desventaja de este método es que requiere que MATLAB esté instalado.

- La segunda forma consiste en implementar el algoritmo backpropagation, utilizando una programación estructurada en lugar de orientada a objetos, ya que como se sabe *mcc* es totalmente compatible con la programación estructurada.

- Utilizar una librería para .NET que permita llevar a cabo el proceso de entrenamiento de la red; la librería más indicada para esta tarea es *FANN* "*Fast Artificial Neural Network Library Version 2.0*", la cual está escrita en C# el lenguaje más rápido, potente, eficiente y por excelencia para programar en VS.NET. *FANN* tiene innumerables ventajas frente a otros paquetes comerciales como "*Neuro Solutions*", algunas de ellas son: Licencia GPL, distribuirla sin pagar licencias, se puede ver y modificar el código, ejecución más rápida, ya que está escrita en C# el lenguaje por excelencia del paquete VS.NET.

Para solucionar el problema de la simulación y entrenamiento de RNA en .NET se optó por escoger la tercera opción; ya que con la librería *FANN* se puede realizar el entrenamiento de la red neuronal de una manera sencilla, sin establecer comunicación alguna con MATLAB. Además, el tiempo que tarda la rede neuronal en su entrenamiento empleando dicha librería es mucho menor que el que emplea el algoritmo backpropagation implementado con programación estructurada.

Desafortunadamente la librería *FANN* no cuenta con la función de transferencia *radbas*; por tal razón hubo la necesidad de buscar una nueva estructura de una red neuronal que permitiera obtener buenos resultados. En la Tabla 9 se especifican las funciones de transferencia empleadas, así como el algoritmo de aprendizaje que finalmente se empleo. Cabe aclara que dicha estructura es exactamente igual para las redes de movimiento y color.

Tabla 9. Estructura de la RNA con la librería *FANN*

Capa	*Entrada*	*Oculta*			*Salida*
Numero de neuronas	60	300	200	100	7/5
Función de transferencia	Lineal	Tansig	Tansig	Tansig	Tansig
Algoritmo de entrenamiento	RPROP (*Resilent Backpropagation*)				

En la Figura 37 se puede observar finalmente la interfaz gráfica de usuario, en ella se encuentra un botón *anónimo* que permite ingresar en modo independiente del hablante. Cuando un niño ingrese en este modo e introduzca una palabra mal pronunciada, el robot le dirá que no entiende el comando y motivara al niño a que pronuncie adecuadamente la palabra. A continuación de este se encuentra el botón *agregar* que crea un nuevo usuario, seguidamente aparece el botón *usuario* con el cual se puede ingresar luego de crear un usuario, luego se tiene el botón *administrador* el cual permite eliminar y ver las cuentas de usuario. Al dar clic en cada botón aparecen nuevas ventanas las cuales se explican claramente en la ayuda de la aplicación.

La aplicación, junto con el código fuente se encuentra en el CD que acompaña este documento, en la carpeta instalador.

Figura 37. Interfaz grafica de usuario GUI desarrollada

6. ANÁLISIS DE RESULTADOS

En este capítulo se mostrarán los resultados obtenidos en las diferentes pruebas realizadas a la herramienta robótica diseñada controlada por voz. Inicialmente se describirán las pruebas y modificaciones que se le realizaron a las diferentes etapas del algoritmo de reconocimiento descrito de en el capítulo 4; con el objetivo de encontrar su mejor rendimiento, determinando de esta manera la influencia de factores varios en el proceso de reconocimiento.

6.1 ANÁLISIS DE LA ETAPA DE ADQUISICIÓN DE LA SEÑAL DE VOZ

La etapa de adquisición es una de las que más incidió en el rendimiento del algoritmo de reconocimiento debido principalmente a la variabilidad de la señal de voz. Como ya se había mencionado… en la sección 3.3… la pronunciación de las palabras de un mismo hablante (y con más razón varios hablantes) nunca es exactamente igual, lo que complica el proceso de clasificación. Esta variabilidad depende de varios factores que se explican a continuación.

6.1.1 Factores intrínsecos. Los factores intrínsecos son aquellos asociados al fenómeno de producción de voz, es decir son dependientes del hablante más no del medio. Al ser este un proyecto para niños en etapa pre-escolar estos factores fueron bastante determinantes, dado a que los algunos niños que se encuentran en esta etapa presentan dislalias que no favorecen el proceso de reconocimiento y clasificación de palabras.

Estas dislalias se presentaron principalmente en la pronunciación de las palabras “adelante”, “atrás”, “rojo” y “verde”. Este problema fue el que ocasionó que la GUI desarrollada tuviera la opción de entrenamiento para un nuevo usuario ya que aunque se consiguió que algunas de estas dislalias fueran reconocidas por la RNA en otros casos resultan imposibles de reconocer.

Otro factor intrínseco que se presentó en el proceso de adquisición fue el relacionado con el acento. La población Huilense tiene un acento marcado en la mayoría de su población y los niños no son la excepción. A diferencia de las dislalias este factor no fue determinante al evaluar la tasa de rendimiento del sistema.

6.1.2. Factores extrínsecos. Estos factores están relacionados con el medio, es decir, son externos al hablante. En esta aplicación los factores intrínsecos que se involucran son el ruido y el medio utilizado para la conversión acústica – eléctrica, el micrófono. A continuación se describirá la influencia de estos dos factores en el proceso de adquisición.

El ruido es totalmente indeseable en el proceso de reconocimiento, por lo que se recomienda que la adquisición se haga en un ambiente totalmente libre de este. Pero dada la aplicación de este proyecto lo anterior no resulta tan conveniente. Este es un proyecto educativo para niños, lo que hace suponer que sea utilizado en las aulas de clase de las instituciones educativas donde los niveles de ruido son bastantes elevados. Por tal razón se optó por tomar las muestras en un ambiente real donde será utilizado, una escuela.

Aunque los filtros empleados permitieron eliminar el ruido en la mayoría de las muestras, existen otras en las que este efecto no deseado no pudo ser eliminado lo que ocasiona problemas en la detección de extremos como se detallara más adelante.

El micrófono también es decisivo en el proceso de reconocimiento el cual se realiza adecuadamente si se cuenta con muestras de calidad. En búsqueda de dichas muestras se realizaron varias pruebas que se comentan a continuación.

Inicialmente se adquirieron las muestras con un micrófono para PC sencillo, pero las señales adquiridas con este presentaban gran cantidad de ruido de fondo que para la etapa de filtrado fue imposible eliminar. Para solucionar este inconveniente se colocó una espuma alrededor de la rejilla del micrófono lo que permitió reducir un poco el ruido.

Pero el mejor resultado se obtuvo cuando se empleó un micrófono de más nivel con el cual se minimizó considerablemente los efectos del ruido, dado a que tiene una menor medida de ruido propio, lo que favorece la relación señal a ruido S/N.

Además, este micrófono presenta una respuesta más plana, es decir tiene aproximadamente la misma relación de transformación de presión acústica en tensión eléctrica en todo el rango de frecuencias audibles (20 y 20000 Hz). En la Figura 38 se muestra como el micrófono puede agregar ruido propio a la señal adquirida.

Otro factor que puede añadir ruido y distorsionar significativamente el espectro de la señal son la colocación del micrófono y distancia entre el micrófono y el hablante.

Por tal razón se recomienda que para la adquisición de las muestras el micrófono este ubicado de frente al hablante a aproximadamente 3 cm de la boca, con el fin de obtener muestras de buena calidad. Además, se recomienda que el usuario que va a controlar el robot, pronuncie con buen tono para que el proceso de clasificación y extracción se pueda desarrollar adecuadamente.

6.2 ANÁLISIS DE LA ETAPA DE PRE-PROCESAMIENTO

Cuando se tiene una relación señal a ruido S/N considerable los procesos involucrados en la etapa de pre-procesamiento no presentan inconveniente alguno. Los problemas aparecen cuando el nivel de ruido de la señal es elevado como se muestra en la Figura 39.

Aunque para eliminar el ruido de la señal de voz se efectúa un proceso de filtraje tanto de pre-énfasis como pasa banda, que busca eliminar las frecuencias correspondientes al ruido, como se demostró… en la sección 3.2…, existen muestras en las que resulta imposible filtrar por completo el ruido de la señal, lo que da origen en algunos casos a una detección incorrecta de extremos que complica etapas posteriores del proceso de reconocimiento.

Figura 38. Ruido propio en la adquisición de la señal de voz

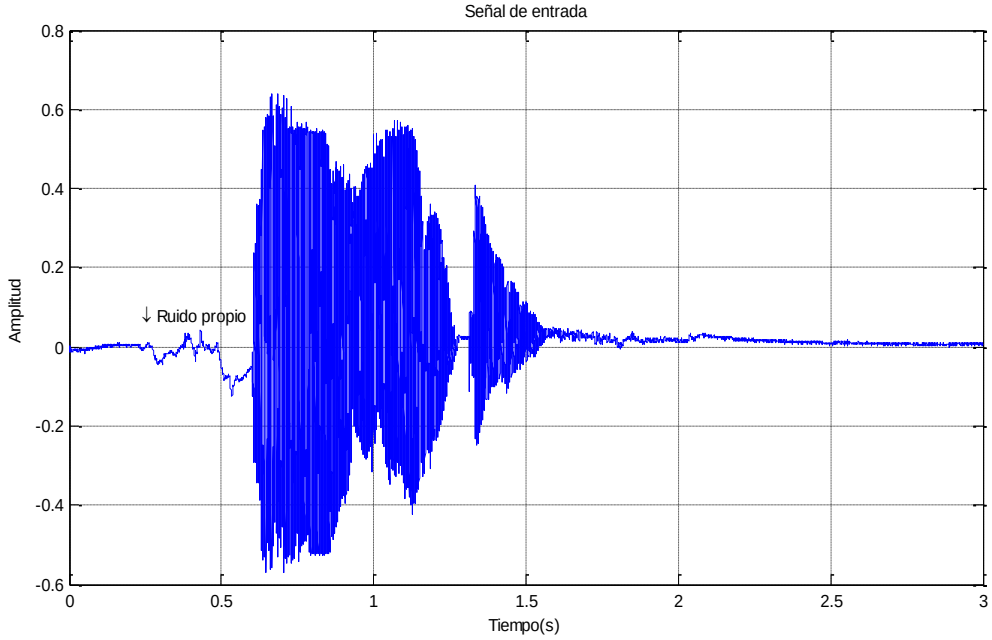
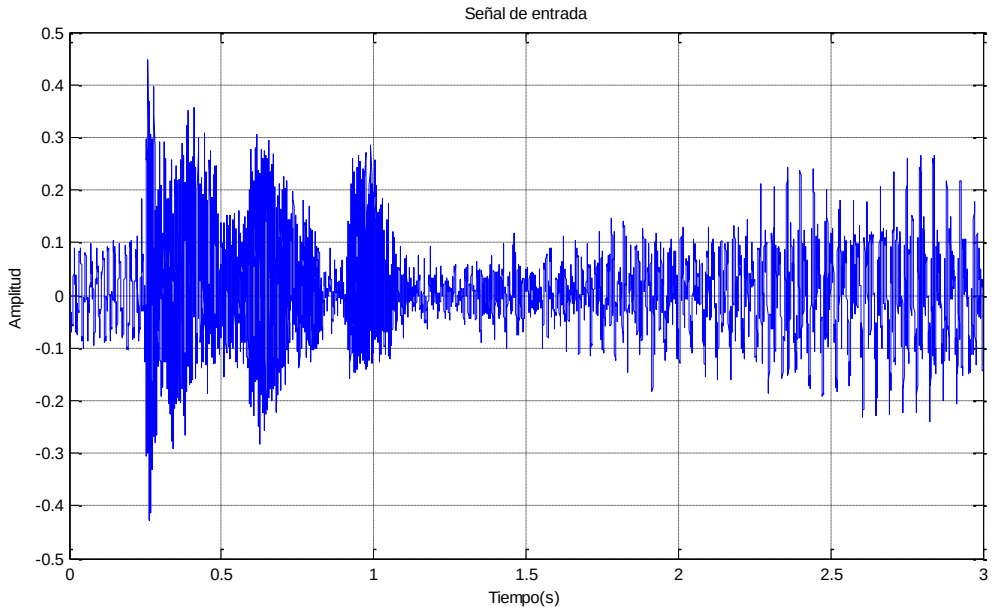
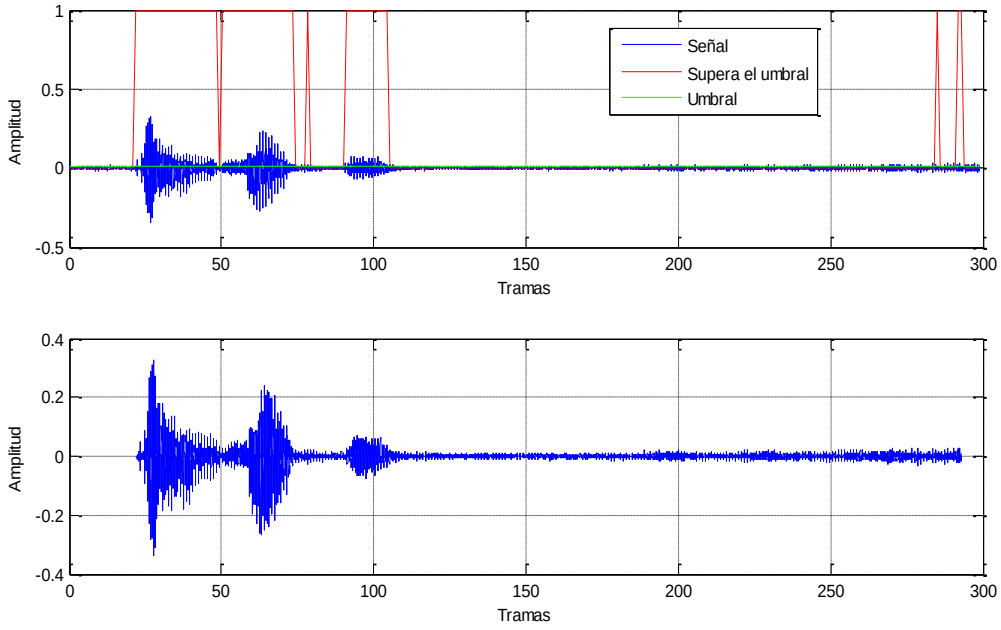


Figura 39. Muestra de la palabra “Adelante” con elevado nivel de ruido



Un ejemplo de lo anterior se puede observar en la Figura 40 en donde el nivel de energía de la parte final de la muestra es superior al umbral dinámico establecido, debido a la alta presencia de ruido, lo que ocasiona que el algoritmo detecte esta parte como señal de voz; realizándose de esta manera una detección incorrecta de la palabra.

Figura 40. Extracción incorrecta de extremos debido a los altos niveles de ruido



6.3 ANÁLISIS ETAPAS DE EXTRACCIÓN DE CARACTERÍSTICAS

El reconocimiento de palabras aisladas es un proceso dependiente de cada una de las etapas que se realizan en él; por ejemplo, si se tienen problemas en el pre-procesamiento de la señal estos conllevaran a que el vector característico que se extraiga posteriormente no represente de la mejor manera la señal de voz y en consecuencia no se clasifique correctamente.

Evaluar que tan representativo es el vector que se obtiene en la extracción de características no es algo que se pueda determinar en esta etapa. Es realmente en la etapa de clasificación donde se verificará si el método empleado para extraer dicho vector fue o no correcto. Por tal razón en análisis de resultados de la etapa de extracción de características se realizará en la siguiente sección, cuando se halla efectuado y definido el método de clasificación.

6.4 ANÁLISIS ETAPA DE CLASIFICACIÓN

Inicialmente se describirá el proceso que se realizó para determinar la estructura de la red neuronal artificial empleada para realizar la clasificación de las diferentes palabras que proporcionó un mejor rendimiento para posteriormente evaluar la representatividad del vector característico.

La selección de los parámetros apropiados para que la RNA realice el proceso de clasificación de las diferentes palabras fue un proceso tedioso, dado que este es básicamente un proceso de prueba y error. Antes de iniciar el entrenamiento de la red se normalizaron los valores de entrada, es decir los vectores característicos. Inicialmente se buscó que estuvieran en el rango de 0 a 1, pero con esta normalización no se obtuvieron buenos resultados debido a que al cero que se maneja como valor

mínimo genera valores nulos lo que ocasiona que los cambios de peso en la red sean bruscos. Para evitar esto finalmente se normalizó empleando valores bipolares definidos en el rango entre -1 y 1. Las salidas también se encuentran expresadas en dicho rango de valores, donde 1 representa un nivel activo o detección y -1 inactividad.

Cuando la red no asocia la palabra pronunciada con ningún patrón se produce un estado que se ha llamado no reconocimiento en el cual coloca todas sus salidas en -1. Lo ideal sería que este estado solo se presentara cuando se pronuncia una palabra que no maneje la red, pero debido al ruido excesivo que presenta algunas muestras, en algunas ocasiones estas son categorizadas en este estado.

Con el fin de hacer el entrenamiento de la red neuronal más robusto los vectores característicos de entrada fueron contaminados con ruido aleatorio. A cada vector se le sumo y resto dicho ruido, generándose de esta manera dos vectores característicos adicionales; que son de gran utilidad cuando el sistema es dependiente del hablante ya que el número de vectores con los que se entrena la red es bastante reducido y la adición de estos ayuda a que la RNA tenga más patrones representativos y efectué un mejor reconocimiento.

Como se mencionó… en la sección 4.4… la red que se empleó para realizar la clasificación de los diferentes vectores característicos fue una *backpropagation* o de retro propagación. Se seleccionó este tipo de red porque cuando se realiza un entrenamiento apropiado de ellas estas tienden a dar respuestas razonables cuando se presentan entradas que nunca se han considerado. Además, el objetivo del algoritmo *backpropagation* hace que los pesos de los niveles escondidos generen una representación interna adecuada del problema a resolverse. Estas características y su porcentaje de éxito lo han convertido en uno de los algoritmos de aprendizaje más populares[11].

En cuanto al algoritmo de entrenamiento se realizaron pruebas iníciales con el *Levenberg Marquardt (trainlm* en MATLAB*)*, ajustando el parámetro de memoria debido a los requerimientos de almacenamiento que presenta este algoritmo. Pero finalmente se identificó, que el funcionamiento del *trainlm* es poco eficiente en problemas de clasificación de patrones.

El algoritmo que finalmente se empleó para el entrenamiento de la red neuronal fue el *Resilent Backpropagation* representado en MATLAB con la función *trainrp*. Este algoritmo fue el más rápido de todos los probados y con el que mejores resultados se obtuvieron.

Una red multicapas consiste de al menos tres capas: Una capa de entrada, una o más capas ocultas y una capa de salida. Como se mencionó…en la sección 3.4.2…el número de neuronas de las capas de entrada y salida depende de cada aplicación en particular. Sin embargo, aunque el funcionamiento de la red depende en forma importante del número de nodos en las capas ocultas, no existe aún un método confiable que permita determinar con precisión el número óptimo de estos.

[11] OTERO BARREIRO, Adriana Sofía y TRUJILLO LEMUS, Gustavo Adolfo. Sistema para el reconocimiento óptico de caracteres alfanuméricos en placas de automóviles particulares. 2005. p. 100-104. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

Una manera de estimar el número óptimo de nodos en la capa oculta es detener el entrenamiento después de un cierto número de iteraciones y determinar cuántos patrones fueron propiamente reconocidos con el número actual de neuronas usadas en la capa oculta. Si el resultado de esta prueba no es satisfactorio se agregarán una o más neuronas en la capa oculta para mejorar el desempeño de la red. Sin embargo, desafortunadamente en estos casos la red tiene que ser entrenada completamente[12].

El proceso mencionado anteriormente fue el que finalmente se empleó para determinar el número de capas ocultas y neuronas de cada una de ellas más apropiados. Para ello se realizó un bucle en el cual inicialmente se implementó una red con una sola capa oculta a la cual se le fueron incrementando el número de neuronas de dicha capa a razón de 50. De acuerdo a los resultados obtenidos se efectuó una variación a una menor escala, para evaluar si el número de neuronas seleccionado es el más óptimo o existe un número de estas que proporcione mejores resultados.

Después de obtener el número de neuronas óptimo con una sola capa oculta, se procedió a incrementar el número de capas ocultas, pero con esto no se mejoraron los rendimientos obtenidos anteriormente. El mejor resultado se obtiene con una sola capa oculta compuesta por 200 neuronas. En cuanto a las funciones de transferencia de la capa oculta y de salida, se hicieron diferentes pruebas para estimar cuál de ellas proporcionaba la mejor respuesta siendo finalmente seleccionadas las funciones *radbas* y *tansig*. Dada los valores de salida manejados la función *tansig* resulta apropiada para la capa de salida. En cuanto a la función *radbas* que es la función de transferencia que se emplea en redes de base radial y que permitió mejorar considerablemente el rendimiento del algoritmo de reconocimiento dado que este tipo de función resulta apropiada en problemas de clasificación.

Una vez definida la estructura de la RNA se puede entrar a evaluar la representatividad del vector característico obtenido en la etapa de extracción a través de la tasa de rendimiento del algoritmo de reconocimiento T_R, como se muestra en la siguiente expresión:

$$T_R = \left(\frac{nnv - (nerrores + nnoreconocidas)}{nnv} \right) \times 100$$

Donde,

nnv es el número de muestras con los que validó la RNA.
nerrores es el número de muestras que la RNA detectó erróneamente.
nnoreconocidas es el número de muestras que la RNA no clasificó en ninguna de las categorías.

[12] SÁNCHEZ, G., PÉREZ, H. * y NAKANO, M. Red Neuronal Creciente Usando Perturbación Simultánea. México. Instituto Politécnico Nacional.

Para efectuar un análisis que permita evaluar la efectividad del algoritmo de reconocimiento de la manera más adecuada, se hicieron pruebas con niños de diferentes sexos, edades (entre 6 y 8 años) y bajo condiciones variables de ruido. Inicialmente se analizará el rendimiento del algoritmo dependiente del hablante para luego analizar los resultados del algoritmo independiente del mismo.

6.4.1 Dependiente del hablante. Como se mencionó…en la sección 3.4.2…este algoritmo fue diseñado para niños que presentan fuertes dislalias que ocasionan detecciones incorrectas de palabras empleando el algoritmo independiente del hablante.

Para las pruebas que se describen a continuación se tomaron dos muestras por palabra para el entrenamiento de la red y no se tomó un número fijo de muestras para validar el rendimiento del algoritmo; con el objetivo de realizar una evaluación global del mismo. Se estudiará el rendimiento de este algoritmo realizando 5 entrenamientos para 20 hablantes (10 niños y 10 niñas).

Inicialmente se realizaron pruebas clasificando las palabras comando pronunciadas con el niño, en dos categorías movimiento y color. Como se mencionó…en la sección 3.3.2…se probaron básicamente dos formas de extracción de características basadas en el cálculo de los coeficientes cepstrales en escala Mel. La primera de ellas que se ha llamado método inicial, consistió en calcular el promedio por canal de los coeficientes cepstrales de cada una de las tramas que compone la señal enventanada.

Con esta forma de extracción de características, se alcanzó un rendimiento promedio de 78.0813% que aunque es en términos generales aceptable, no es el óptimo. El promedio realizado en la etapa de extracción, debido al desigual tamaño de las muestras, hace que se pierdan algunos patrones característicos importantes de la señal que junto con las dislalias que presentan algunos niños, se convierten en los principales responsables de esta tasa de rendimiento.

Dado este rendimiento, se buscó una mejor forma de representar la información contenida de la señal de voz como se comentó…en la sección 3.3.2…obteniéndose los rendimientos que se muestran en la Figura 41. Como se puede ver en la Figura 42, con esta forma de extracción de características se mejoró considerablemente el rendimiento promedio obtenido anteriormente. Por tal razón, esta forma fue finalmente la seleccionada para extraer las diferentes características de la señal de voz. Los rendimientos finales para las redes de movimiento y color se muestran en la Tabla 9.

Tabla 10. Rendimientos promedios finales reconocimiento dependiente del hablante.

Rendimiento promedio red movimiento	88.0754%
Rendimiento promedio red color	88.4921%

También se realizaron pruebas uniendo en una sola red los comandos de movimiento y color alcanzándose un rendimiento promedio aceptable del 73.5955%. Pero finalmente se optó por

emplear 2 redes neuronales (una para movimiento y otra para color), para que los niños puedan distinguir y asociar cada comando en la respectiva categoría.

Figura 41. Rendimiento final de la red movimiento dependiente del hablante.

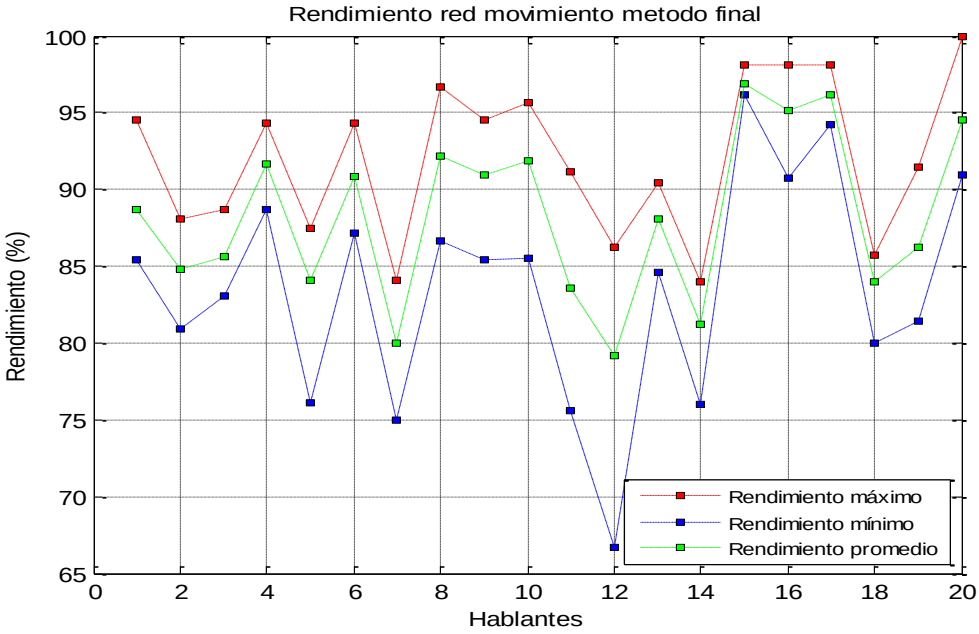
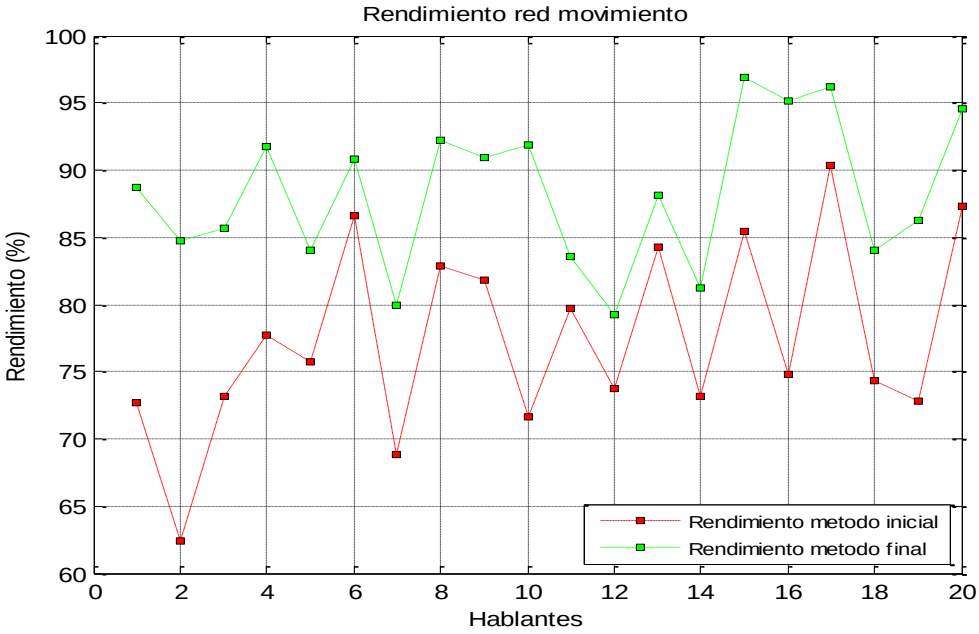


Figura 42. Comparativa de rendimientos promedios de los métodos empleados para la extracción de características.



6.4.2 Independiente del hablante. Realizar el entrenamiento de la RNA que permitió obtener un rendimiento aceptable del algoritmo de reconocimiento para que fuera independiente del hablante, fue un proceso complicado y tedioso; ya que la selección de las muestras que conforman el conjunto de entrenamiento se convierte en el punto más determinante para tener un buen rendimiento.

Inicialmente se entrenó la red neuronal con un número distinto de muestras por palabras que generó una tasa de rendimiento indeseada, debido a que la red presenta un buen rendimiento en aquellas palabras donde se incluyeron más muestras y deficiente para las palabras con pocas muestras. Por tal razón las siguientes pruebas se realizaron con el mismo número de muestras por palabras.

También se agregaron al entrenamiento muestras con dislalias con el objetivo de hacer el algoritmo más robusto, pero este objetivo no se cumplió ya que la tasa de rendimiento decayó empleando este tipo de muestras. Por lo que se optó por entrenar la red con las mejores muestras con las que se contaban.

Para encontrar el mejor conjunto de muestras para el entrenamiento se hicieron diversas pruebas, variando el número de hablantes y el número de muestras por hablante. Después de estas se determinó el mejor conjunto de entrenamiento el cual está conformado por 10 hablantes (5 niños y 5 niñas) y 8 muestras de cada palabra por hablante, para un total de 560 y 400 muestras para las redes de movimiento y color respectivamente.

La validación del entrenamiento se realizó con 20 hablantes diferentes a los que se emplearon en el entrenamiento. El número total de muestras empleadas para la validación de las redes de movimiento y color fue 1520 y 1150, respectivamente. En la Tabla 10 se especifican el rendimiento mínimo, máximo y promedio obtenidos para cada red.

Tabla 11. Rendimientos promedios finales reconocimiento independiente del hablante

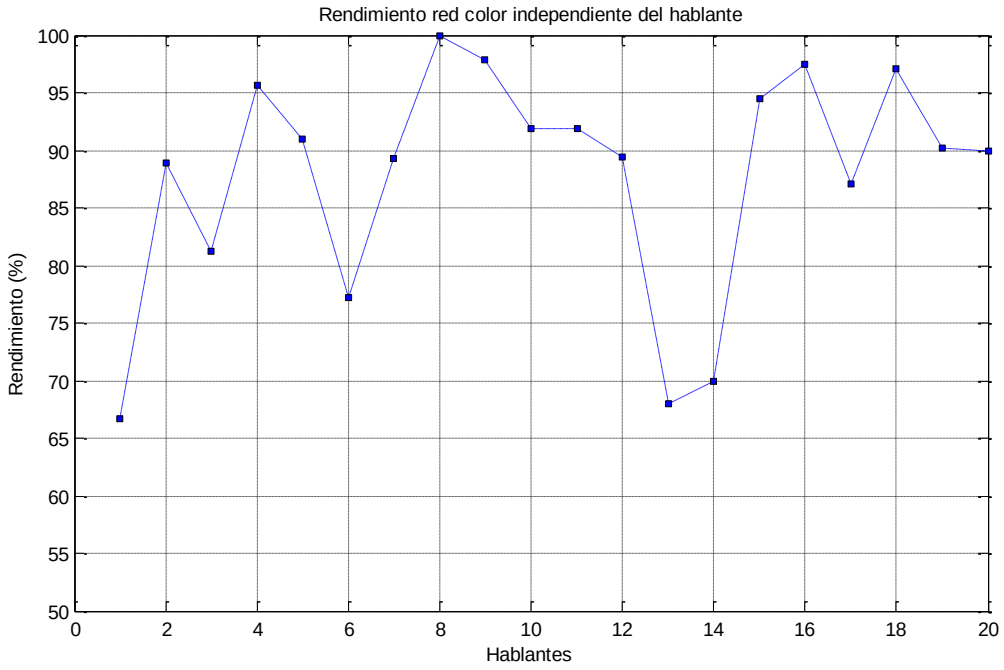
Red movimiento	
Rendimiento mínimo	65.9574%
Rendimiento máximo	97.7778%
Rendimiento promedio	86.7170%
Red color	
Rendimiento mínimo	66.6667%
Rendimiento máximo	100%
Rendimiento promedio	87.7533%

En la Figura 43 se muestra el rendimiento de la red color. Como se puede observar en ella para algunos hablantes el rendimiento se cae hasta aproximadamente el 65% que no es la tasa de rendimiento más deseada. Así que analizamos que factores influyeron en el bajo rendimiento que se presenta siendo estos los siguientes:

- El niño empieza a hablar antes de que se inicie la grabación.
- El ruido no se eliminó por completo en la etapa de filtrado, en consecuencia se presentó una detección de extremos errónea de la palabra.
- Dislalias.
- El tono de la voz del hablante es demasiado bajo.

La mayoría de los factores mencionados anteriormente están relacionados con la etapa de adquisición de la señal de voz y…en la sección 6.1…se realizaron las respectivas recomendaciones que permiten que algunos de los factores mencionados anteriormente se presenten.

Figura 43. Rendimiento de la red color independiente del hablante.



7. CONCLUSIONES

- Con la realización de este proyecto se demostró que la robótica educativa se constituye en una fuerte herramienta pedagógica, ya que esta es altamente atractiva para los estudiantes dado que proporciona un ambiente en el cual ellos pueden desarrollar y poner en práctica ciertos aprendizajes facilitándole de esta manera la construcción de su propio conocimiento de forma lúdica e integrando diferentes áreas del conocimiento. Específicamente con esta aplicación se logró no solo que los niños por medio de la robótica educativa adquirieran nociones de orientación espacial sino que también se les motivó a que realizaran una pronunciación adecuada de las diferentes palabras comandos afianzándolos también en esta área.

- En los sistemas de reconocimiento de habla las menores tasas de rendimiento se obtienen cuando los hablantes corresponden a la población infantil, debido a que los niños apenas empiezan su conocimiento del lenguaje y en algunas ocasiones pronuncian inadecuadamente algunas palabras. La herramienta robótica diseñada está dirigida a niños de etapa preescolar donde los problemas anteriores se ponen en manifiesto y este fue uno de los mayores inconvenientes que se presentó en la realización de este proyecto.

- El reconocimiento de palabras aisladas es un proceso básicamente de extracción de características en el que se busca que cada palabra analizada pueda ser representada inequívocamente como un conjunto de valores que se distinga de las demás. Para tal fin se realizaron una serie de operaciones tanto en el dominio tiempo como en el dominio de la frecuencia, con el objetivo de atenuar o eliminar características indeseables y resaltar otras que fueron bastante útiles a la hora de realizar la clasificación e identificación de la palabra.

- Se determinó que una de las etapas más importantes en el proceso de reconocimiento es precisamente la relacionada con la adquisición de la señal de voz, ya que la calidad de las muestras tomadas incide directamente a la hora de evaluar el rendimiento del algoritmo de reconocimiento. Existen factores tanto intrínsecos como extrínsecos asociados a esta etapa, los factores extrínsecos se intentan corregir con la etapa de pre-procesamiento pero desafortunadamente con los factores intrínsecos no hay nada que se pueda hacer ya que estos están relacionados directamente con el proceso de producción de la voz.

- Se determinaron que los factores que más inciden en el rendimiento final del sistema de reconocimiento son los niveles de ruido y las dislalias. Para el primero de estos factores el algoritmo de reconocimiento incorpora una etapa de filtrado, pero desafortunadamente en algunas ocasiones cuando la relación señal a ruido S/N no es significativa; esta etapa no es capaz de eliminar las componentes frecuenciales del ruido lo que ocasiona problemas en las etapas posteriores. En cuanto a las dislalias, eliminarlas no está dentro del alcance del sistema de reconocimiento, ya que este es un factor intrínseco de la producción de la señal de voz, por

tal razón la única salida que se le encontró a este problema fue la elaboración de un algoritmo dependiente del hablante, para evitar los bajos rendimientos que se producen debido a este factor.

- Los algoritmos diseñados fueron sometidos a diferentes pruebas con el fin de evaluar la efectividad del algoritmo de reconocimiento bajo diversas condiciones. Para ello se hicieron pruebas con niños de diferentes sexos, edades (entre 6 y 8 años) y bajo condiciones variables de ruido. Pero aun así se hace necesario realizar pruebas del desempeño del sistema en el lugar en el cual se va a implementar, ya que tomando las características del lugar se pueden configurar adecuadamente algunos parámetros de reconocimiento con el objetivo de alcanzar mejores resultados en el proceso de reconocimiento.

- Se demostró que el reconocimiento de palabras aisladas es un proceso dependiente de cada una de las etapas que se realizan en él; los errores y problemas que no puedan ser solucionados en la fase anterior se reflejaran en la fase siguiente ocasionando que el rendimiento final del sistema no sea el más optimo.

- Se comprobó que los coeficientes cepstrales en escala de Mel (MFCCs) son una técnica que proporcionan buenos resultados en los sistemas de reconocimiento debido a que realizan la discriminación frecuencial de la misma manera que la realiza el oído empleando para ello la escala de Mel. Aunque se suelen emplear las primeras y segundas derivadas de estos coeficientes para formar el vector característico de la señal de voz, en esta aplicación no se tuvieron en cuenta dado a que no se generaron buenos resultados cuando se agregaron dichos coeficientes.

- Inicialmente se realizó el reconocimiento de las 12 palabras que maneja esta aplicación en una sola red con la que se alcanzó un rendimiento promedio del 73.5955%. Pero finalmente se optó por emplear 2 redes neuronales (una para movimiento y otra para color), para que los niños puedan distinguir y asociar cada comando en la respectiva categoría.

- Dadas las dislalias que presentan algunos niños se desarrollaron dos algoritmos de reconocimiento uno dependiente del hablante y otro independiente del mismo. Con el algoritmo dependiente se alcanzaron rendimientos promedios de 88.0754% y 88.4921% para las redes de movimiento y color respectivamente. Empleando el algoritmo independiente del hablante se obtuvieron resultados promedios del 86.7170% y 87.7533%. Pero desafortunadamente las tasas de rendimiento de ambos algoritmos no son estables ya que para algunos hablantes esta decae hasta el 65% debido principalmente a factores relacionados con la adquisición de la señal de voz.

8. RECOMENDACIONES

- Para mejorar el algoritmo de reconocimiento se puede aplicar algún método de reducción de ruido y con esto mejorar la detección de extremos de la palabra a analizar, ya que este fue uno de los factores que más incidió a la hora de determinar el rendimiento del mismo.

- Una realización futura inmediata podría ser la implementación en tiempo real del algoritmo de reconocimiento de voz empleando para esto un DSP, para tal fin sería necesario realizar los algoritmos que se han utilizado en la elaboración de este proyecto en lenguaje de programación C.

- Con el presente proyecto se pretende impulsar a nuevos tesistas a que exploren campos que no habían sido explorados en los proyectos de grado de la universidad: la robótica educativa y el reconocimiento de voz. Hasta la elaboración de este proyecto no se había realizado proyecto alguno de grado que utilizara los *Lego Mindstorms NXT* y son muchas las aplicaciones que se pueden realizar con dicho robot no solo en el campo de la robótica educativa.

  En cuanto al reconocimiento de voz son múltiples las aplicaciones que se pueden realizar como lo son: los sistemas diseñados para discapacitados, el control por comandos de sistemas de telefonía o el acceso de acceso por identificación de voz. Para la realización de dichas aplicaciones el presente proyecto serviría de base, pero se considera necesario explorar otros métodos de clasificación de señales de voz, como por ejemplo los HMMs (Hidden Markov Models) que es considerada una técnica mucho más eficiente que los MFCCs.

BIBLIOGRAFÍA

LIBROS

G. PROAKIS, Jhon y G. MANOLAKIS, Dimitris. Digital Signal Processing. Principles, Algorithms and Applications. Tercera edición. Prentice Hall International.

G. PROAKIS, Jhon y K. INGLE, Vinay. Digital Signal Processing using MATLAB. PWS Publishing Company. 1997.

TESIS

ALMARIO ARIZA, Gustavo Felipe y SANTANDER MONTANO, Liliana Isabel. Equipo de electrocardiografía portátil inalámbrico. 2006. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

CARDONA, Enrique y REYES, Jonnattan. Sistema inteligente de predicción del precio del KWh en la bolsa y demanda de energía eléctrica. 2005. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

OTERO BARREIRO, Adriana Sofía y TRUJILLO LEMUS, Gustavo Adolfo. Sistema para el reconocimiento óptico de caracteres alfanuméricos en placas de automóviles particulares. 2005. Trabajo de grado (Ingeniero Electrónico). Universidad Surcolombiana. Facultad de Ingeniería.

ENLACES

http://www.mathworks.com

http://www.lego.com

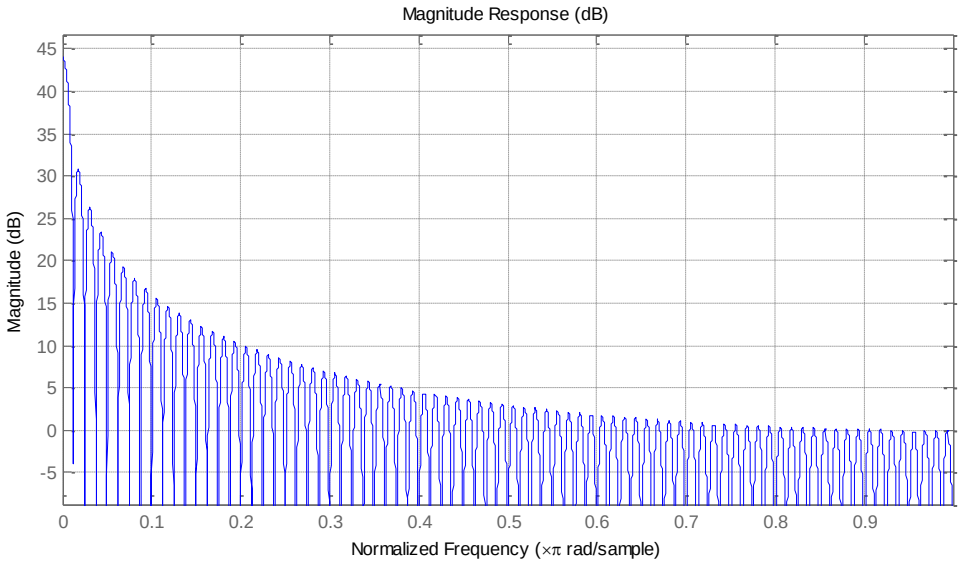
http://www.vozprofesional.cl/

http://www.from.okay.pl/mgalczer/

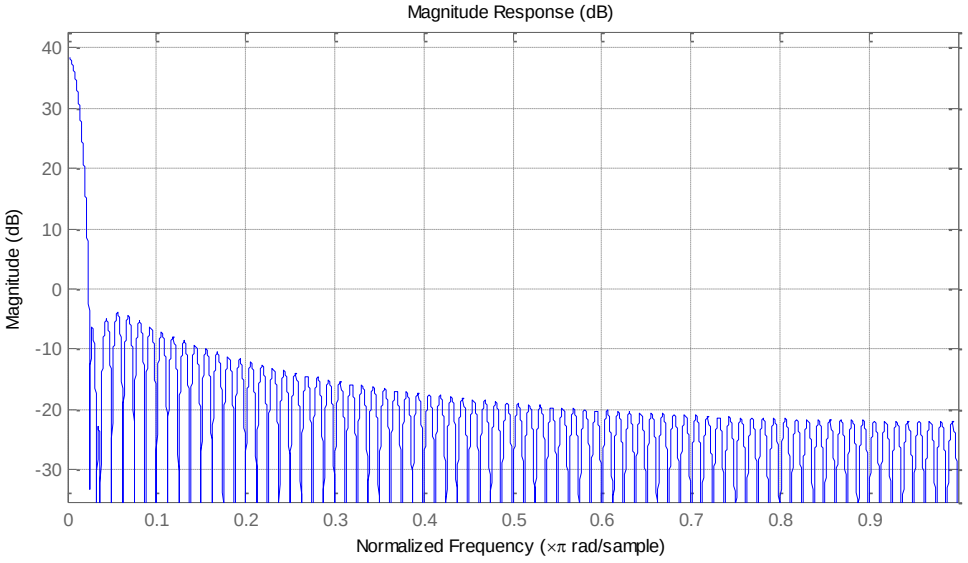
http://www.lfb.rwth-aachen.de/en/education/ws07/mindstorms.html

http://homepages.udayton.edu/~hardierc/ECE203/sound.htm

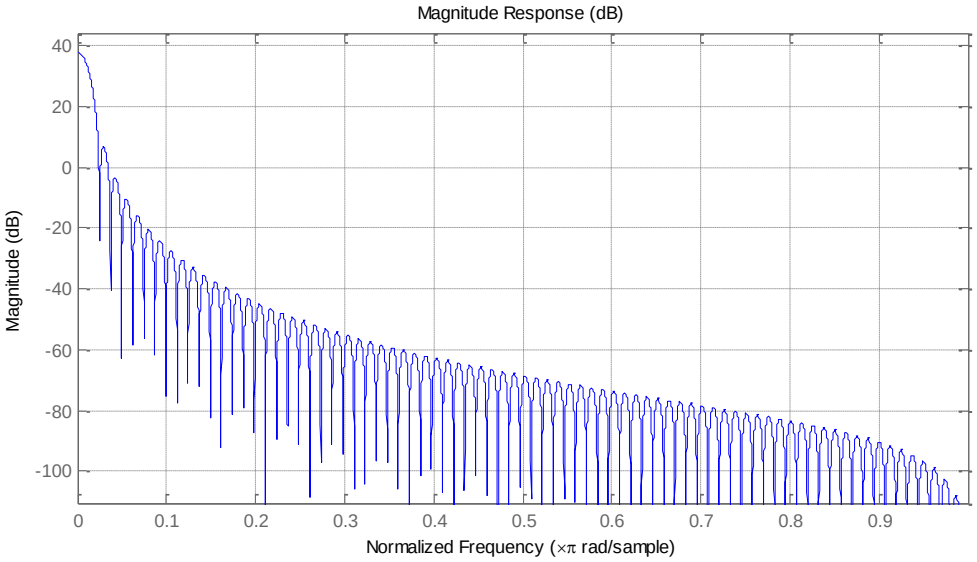
ANEXO A. Respuesta espectral de diferentes ventanas



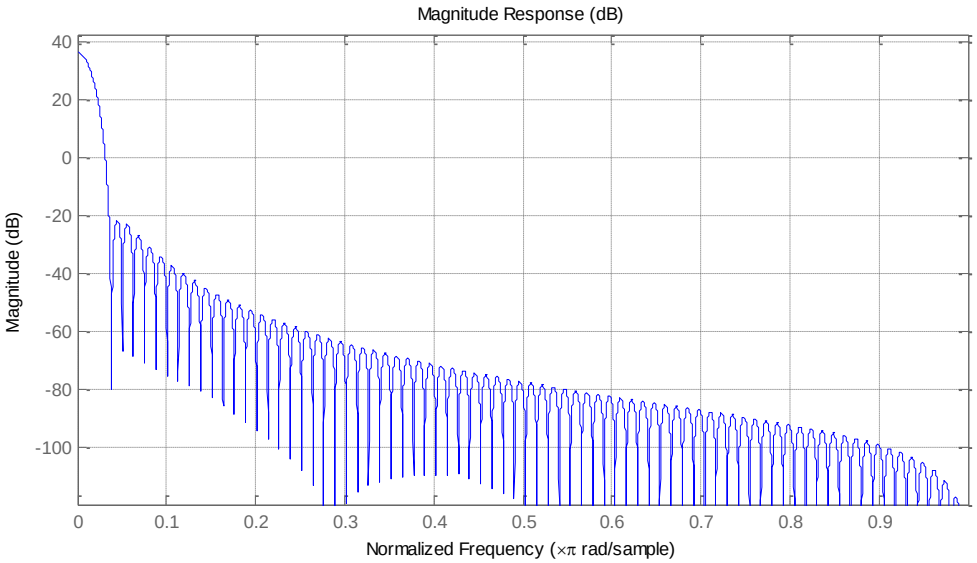
Espectro ventana Rectangular



Espectro ventana Hamming

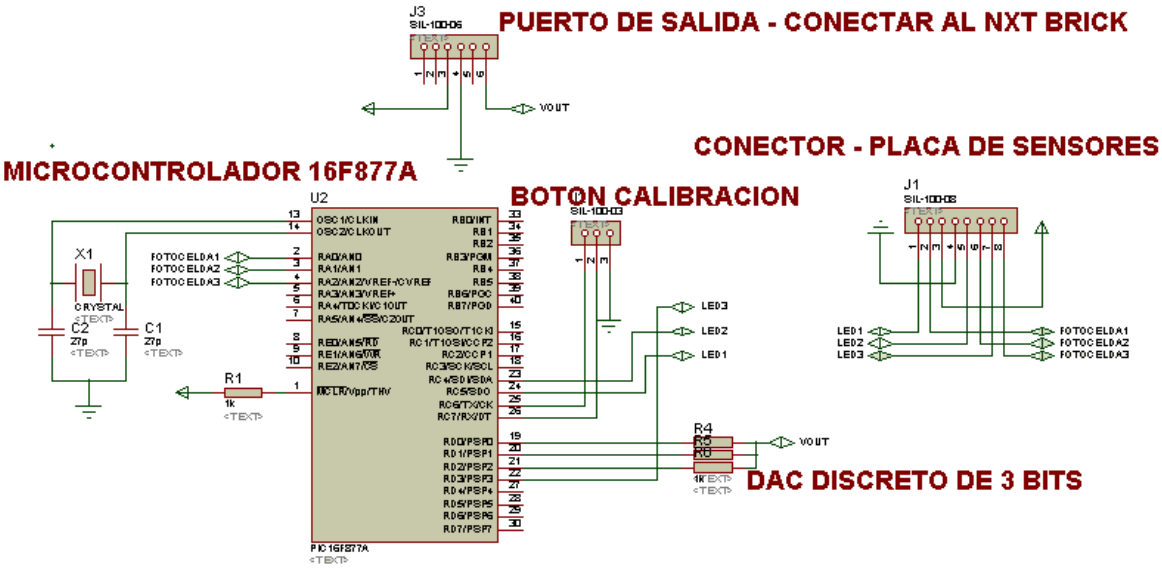


Espectro ventana Hanning

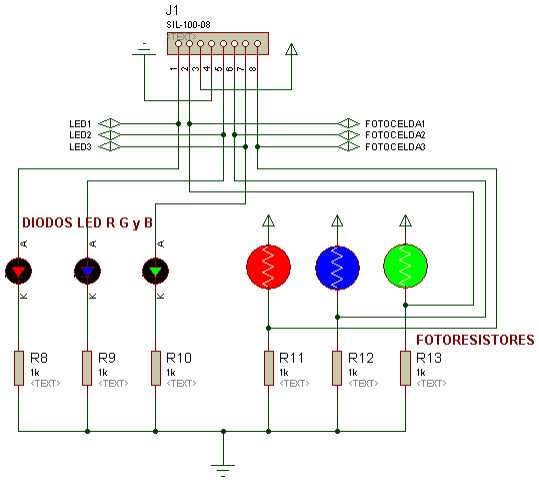


Espectro ventana Blackman

ANEXO B. Circuitos del sensor color diseñado



Circuito de control del sensor



Transductores del sensor