DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO DE COMANDOS DE VOZ BASADO EN REDES NEURONALES PARA LA ACTIVACIÓN DE DISPOSITIVOS ELECTRÓNICOS

JOHN JAIRO MARTÍNEZ BENAVIDES
EDGAR ALBERTO UNIGARRO CALPA

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2010

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO DE COMANDOS DE VOZ BASADO EN REDES NEURONALES PARA LA ACTIVACIÓN DE DISPOSITIVOS ELECTRÓNICOS

JOHN JAIRO MARTÍNEZ BENAVIDES
EDGAR ALBERTO UNIGARRO CALPA

Trabajo de Grado presentado como requisito para obtener el título de Ingeniero Electrónico

Asesor: MSC. DARÍO FERNANDO FAJARDO FAJARDO

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2010

Nota de Responsabilidad

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Acuerdo 1. Articulo 324. Octubre 11 de 1966. Emanado del honorable Consejo Directivo de la Universidad de Nariño.

Nota de aceptación:

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

_____
Firma del jurado

_____
Firma del jurado

San Juan de Pasto, Mayo de 2010

DEDICATORIA

“A mis padres por darme todo el apoyo y colaboración a lo largo de mi vida, al Ingeniero Dario Fajardo por ayudarme a despertar el deseo de investigación, y a Nohora Erasso quien es esa persona que con su entendimiento, apoyo y compañía hace que mis sueños se conviertan en realidades…”

John Jairo Martínez

“A mis padres por cuidar siempre de mí y darme lo que siempre necesite, a Claribel Melendez por darme su apoyo y sus fuerzas para continuar y a Mis hijos que espero pueda darles la felicidad que serán para mi“

Edgar Unigarro

CONTENIDO

pág.

LISTA DE TABLAS

pág.

LISTA DE FIGURAS

pág.

GLOSARIO

ADC: Es un componente electrónico que se encarga de convertir la señal análoga a digital. La sigla ADC significa *Analog to Digital Converter* que en español traduce Conversor Análogo Digital.

ANÁLISIS ESPECTRAL: Se refiere a la acción de descomponer una señal difícil de interpretar en componentes más simples, generalmente componentes de frecuencia.

ANN: Son un paradigma de aprendizaje y procesamiento inspirado en el funcionamiento del sistema nervioso del ser humano. La sigla ANN significa *Artificial Neural Network* que en español traduce Red Neuronal Artificial.

ASR: Forma parte de la inteligencia artificial y busca permitir una comunicación entre los seres humanos y los componentes electrónicos. La sigla ASR significa *Automatic Speech Recognition* que en español traduce Reconocimiento Automático de Voz.

CODEC: También conocido como códec de audio, es un componente electrónico que implementa un conjunto de algoritmos que permiten la modificación de las señales de audio.

DSC: Es un componente electrónico que incorpora algunas ventajas de los DSP y cuenta con la facilidad de manejo y programación de los microcontroladores. La sigla DSC significa *Digital Signal Controller* y traduce en español Controlador Digital de Señales.

DSP: Es un componente electrónico basado en un procesador que se encuentra capacitado para realizar operaciones matemáticas a muy alta velocidad. La sigla DSP significa *Digital Signal Processor* y traduce en español Procesador Digital de Señales.

FFT: Es el término utilizado para referirse al cálculo de un análisis espectral mediante un algoritmo ejecutado de manera eficiente. La sigla FFT significa *Fast Fourier Transformation* y puede traducirse al español como Transformada Rápida de Fourier.

FILTRO ELECTRÓNICO: Es un elemento capaz de discriminar determinadas frecuencias para modificar tanto su amplitud como su fase.

FONEMA: Son las unidades teóricas básicas que cuenta con forma de sonidos que permiten reconocer las palabras de una lengua.

HMM: Es un modelo estadístico que cuenta con parámetros desconocidos. La sigla HMM significa *Hidden Markov Models* y traduce al español Modelos Ocultos de Markov.

KSPS: Sigla utilizada para referirse a la cantidad de muestras que un ADC puede adquirir durante un periodo de un segundo.

MIPS: Sigla utilizada para referirse a la cantidad máxima de instrucciones que un procesador puede realizar durante un periodo de un segundo.

PDF: Describe la probabilidad relativa para una variable aleatoria de ocurrir en un punto dado del espacio de observaciones. La sigla PDF significa *Probability Density Function* y traduce al español Función de Densidad de Probabilidad.

RESUMEN

La idea central en desarrollo del presente trabajo es comprobar la factibilidad de construir un reconocedor de palabras aisladas de pequeño vocabulario en un dispositivo embebido y al mismo tiempo mostrar que es posible el uso de redes neuronales artificiales (ANN) dentro del reconocimiento automático de voz (ASR) en dispositivos embebidos.

En la primera fase del desarrollo se implementó un reconocedor de dígitos mediante lenguaje de alto nivel para demostrar la validez de la aplicación de las ANN en el reconocimiento de voz, para tal fin se montaron tres algoritmos de reconocimiento diferentes, la aproximación clásica mediante modelos ocultos de Markov que afronta los problemas de modelado acústico y temporal en un simple algoritmo, un híbrido de ANN/HMM que divide los modelados en los dos algoritmos y finalmente un híbrido mediante redes neuronales y alineamiento de tiempo dinámico (DTW) que modela acústicamente mediante las ANN y temporalmente mediante el DTW. Se evidenció que este último superó en porcentaje de reconocimiento a los dos otros algoritmos.

La implementación en dispositivo embebido se realizó en el eZDSP VC5505 USB Stick de Texas Instruments, una tarjeta que permite el procesamiento de señales de audio con buenas prestaciones y un costo razonable. La implementación reveló que la cantidad de operatoria requerida en la etapa de extracción de características acústicas de las señales de voz hace que sea muy complicado un procesamiento en tiempo real, pero a pesar de ello es posible si se trabaja con un dispositivo con muy alto desempeño. Al final del documento se expone la manera en que el porcentaje de reconocimiento se ve afectado por varios factores de implementación, por otra parte también se logra exhibir que la propiedad de generalización de la red neuronal bajo las condiciones del proyecto no es suficiente para afrontar el problema de independencia del locutor, aunque a pesar de ello, los resultados obtenidos fueron satisfactorios, ya que se demostró que el modelado permite la construcción de sistemas multiusuario.

ABSTRACT

The main idea in this research was to demonstrate the feasibility to build an embedded isolated words recognizer with a small vocabulary and at the same time showing that the use of Artificial Neural Networks (ANN) in Embedded Automatic Speech Recognition is possible.

The first stage implemented a digit recognizer in high level language programming in order to demonstrate the validity of the ANN application in speech recognition, for that purpose three recognition algorithms were tried, the Hidden Markov Models (HMM) classic approach which manages the acoustic and temporal modeling, an ANN/HMM hybrid which divides the modeling in both stages and finally a Neural Network / Dynamic Time Warping (DTW) hybrid which manages acoustic modeling with ANN and temporal with DTW. The stage showed that the ANN/DTW hybrid overcomes in recognition percentage to the other two algorithms.

The embedded implementation was done in eZDSP VC5505 USB Stick developed by Texas Instruments, a board that allows audio signal processing with good performance and reasonable price. The implementation revealed the process required for acoustic features extraction is a high operatory process which leads to a very difficult real time processing, but it's still possible if it's done with a high performance processor. Last part of the document exposes the key implementation factors which caused the recognized accuracy reduction and also exhibits that the neural network generalization property under the research conditions is not enough to handle the speaker independent issue, but despite that, the results were satisfying because it was shown that the designed model allows multi speaker implementations.

INTRODUCCIÓN

El reconocimiento automático de habla (ASR por sus siglas en inglés, *Automatic Speech Recognition*) es un área que ha sido objeto de investigación durante las últimas cuatro décadas y busca desarrollar sistemas capaces de procesar y entender las señales de voz emitidas por un ser humano, brindando numerosas posibilidades de interacción en diferentes campos. Los avances más significativos en esta área se han logrado durante los últimos años, puesto que el desarrollo de la inteligencia artificial y la aplicación de nuevos y más eficientes métodos estadísticos han permitido grandes mejoras en los sistemas de reconocimiento de voz, además la evolución de materiales semiconductores para la creación de dispositivos de procesamiento tales como DSPs (Procesadores Digitales de Señales) y DSCs (Controladores Digitales de Señales) han permitido que estos se vayan integrando en equipos tales como PDAs, celulares y automóviles; teniendo en cuenta el buen desempeño que se ha logrado a través de técnicas de inteligencia artificial se propone este proyecto encaminado a desarrollar un sistema de reconocimiento de comandos de voz a través de redes neuronales artificiales implementado como sistema embebido para el control de dispositivos electrónicos, con el fin de brindar nuevas posibilidades en cuanto a interfaces de usuario.

El presente documento muestra los resultados de la investigación desarrollada sobre el reconocimiento de voz mediante ANN en un dispositivo embebido. La investigación demuestra que las redes neuronales artificiales pueden formar parte de un reconocedor de palabras y que las ANN ofrecen ciertas ventajas sobre otras técnicas de reconocimiento, además se constata que la implementación de sistemas de reconocimiento de voz en dispositivos embebidos es posible.

DESCRIPCIÓN DEL PROBLEMA

PLANTEAMIENTO DEL PROBLEMA

El actual desarrollo en el campo de la electrónica ha permitido la inclusión de numerosos dispositivos en la vida diaria del ser humano con el fin de facilitar tareas, mejorar la eficiencia de procesos, proporcionar facilidades de comunicación y conectividad o simplemente suministrar entretenimiento, sin embargo, la manipulación e interacción de estos dispositivos puede llegar a ser compleja y en muchos casos inadecuada; esto ha ocasionado que la comunicación con el usuario sea uno de los aspectos que cobren mayor importancia en el desarrollo de equipos electrónicos. Generalmente, la interacción entre hombre y máquina se ha basado en el lenguaje escrito a través de la utilización de teclados. Durante las últimas décadas se han desarrollado equipos para facilitar las interfaces como pantallas táctiles, controles remotos, sistemas de reconocimiento de habla, entre otros.

El reconocimiento de habla brinda ciertas ventajas frente a otros tipos de interfaces, por ejemplo la realización de actividades multitarea, ya que una persona puede continuar realizando actividades manuales y visuales mientras habla. Otro aspecto ventajoso es la velocidad, una persona puede pronunciar alrededor de 200 palabras por minuto mientras que existen muy pocas que pueden digitar más de 60 en el mismo tiempo. El reconocimiento de habla también posibilita la comunicación con dispositivos electrónicos a personas con cierto tipo de discapacidades brindándoles comodidad y disminuyendo su dificultad de interacción con las máquinas lo que conlleva a un aumento en su calidad de vida.

El reconocimiento automático del habla es una rama de la inteligencia artificial que se ha venido estudiando durante mucho tiempo, pero a pesar de ello sigue siendo un área de investigación abierta, ya que se enfrenta a un conjunto de dificultades que pueden abordarse de diferentes maneras. Existen sistemas dedicados al reconocimiento de voz de un único locutor que limitan la utilización a los nuevos usuarios, pues, cada vez que un nuevo hablante desee hacer uso del sistema se requiere una nueva secuencia de entrenamiento. Debido a que el objetivo de los sistemas es brindar agilidad de comunicación se debe realizar un análisis continuo de la voz y dividir esta secuencia en partes (palabras o fonemas) para posibilitar su identificación, esto a su vez conlleva a que se esté guardando constantemente información generando la necesidad de tener una gran memoria para el almacenamiento de datos. Este último inconveniente ha llevado a los diseñadores de sistemas de reconocimiento de habla a dedicarse principalmente a la elaboración de sistemas que funcionen bajo computadores y no a extenderse a sistemas embebidos.

En los últimos años la evolución del manejo de materiales semiconductores ha permitido crear nuevos y mejores elementos de procesamiento con una capacidad de memoria significativa, lo que ha impulsado a los diseñadores a desarrollar tecnologías de reconocimiento de habla mediante sistemas embebidos para equipos como grabadoras, cierta clase de juguetes, celulares, automóviles, entre otros, utilizando principalmente el método de los modelos ocultos de Markov (HMM por sus siglas en inglés, *Hidden Markov Models*). Es así como surge la idea de crear un sistema embebido de reconocimiento de comandos de voz basado en redes neuronales artificiales (ANN por sus siglas en inglés, Artificial Neural Networks) para la manipulación de dispositivos electrónicos, con la finalidad de explorar un poco más en el uso de las redes neuronales como método para reconocimiento de patrones en las señales de voz así como la implementación de estas dentro de sistemas embebidos.

El sistema diseñado es capaz de reconocer palabras aisladas para múltiples locutores. Se seleccionaron tres dispositivos electrónicos y un conjunto de comandos de voz por cada uno de ellos, el sistema de reconocimiento de voz permite el manejo de los dispositivos a través de dichos comandos. El sistema de reconocimiento de voz es de aplicación específica y posee un vocabulario limitado, además el ambiente de trabajo está limitado a condiciones de bajo ruido como laboratorios o ambientes domésticos.

FORMULACIÓN DEL PROBLEMA

¿Es posible implementar un sistema embebido de reconocimiento de voz para activación de dispositivos electrónicos mediante redes neuronales artificiales?

OBJETIVOS

OBJETIVO GENERAL

Diseñar e implementar un sistema embebido para activación de dispositivos electrónicos mediante comandos de voz a través de redes neuronales artificiales.

OBJETIVOS ESPECÍFICOS

Seleccionar los dispositivos a operar a través del sistema de reconocimiento de comandos de voz.

Elaborar un sistema de reconocimiento de comandos de voz mediante software especializado implementado sobre una computadora.

Implementar un sistema embebido de reconocimiento de comandos de voz a través de la codificación en lenguaje de bajo nivel del sistema desarrollado mediante software especializado.

Evaluar el funcionamiento del sistema.

JUSTIFICACIÓN

El reconocimiento automático de voz es un área que ha sido objeto de investigación durante casi cuatro décadas. Inicialmente a pesar de los múltiples esfuerzos realizados no fueron muchos los resultados obtenidos, pero durante las últimas 2 décadas los crecientes avances en cuanto a computación y modelado estadístico han permitido abordar el problema desde otras perspectivas, mostrando muy buenos resultados, aunque no completamente satisfactorios puesto que aún son muchos los inconvenientes por resolver antes de lograr desarrollar un sistema de reconocimiento de voz ideal que supere los problemas de los sistemas existentes, como la dependencia del hablante y del tiempo de pronunciación de la palabra, la baja robustez y la necesidad de software especializado que limita la aplicación del sistema únicamente a la interacción con computador debido a los altos requerimientos de memoria y velocidad de procesamiento. De la misma manera en que aún existen muchos inconvenientes por resolver en este campo, también existen numerosos métodos con los cuales abordar dichos problemas y muchos de ellos todavía se encuentran en exploración, debido a los inconvenientes existentes en los sistemas actuales y a las diferentes alternativas que han surgido para superarlos el reconocimiento automático de voz continua siendo un área en continuo desarrollo y abierta a la investigación; adicionalmente el estado de evolución tecnológica actual permite tener a disposición una serie de herramientas que facilitan la investigación por lo que el desarrollo de sistemas de reconocimiento de voz se encuentra en auge, atendiendo además a la solución de problemas en diferentes campos, principalmente en el desarrollo de interfaces humano – máquina que permiten hacer uso de la voz para la manipulación de dispositivos.

Los sistemas de reconocimiento de voz actuales han sido desarrollados en su mayoría a través de modelos estadísticos conocidos como modelos ocultos de Markov, cuya aplicación es principalmente el reconocimiento de patrones, sin embargo, a pesar de que los resultados obtenidos han permitido una gran evolución, la aplicación de estos modelos implica realizar un cierto número de suposiciones en el modelado que limitan su efectividad, por ello la implementación del sistema a través de redes neuronales artificiales podría suponer una solución que permita mayor precisión gracias a características como la habilidad de clasificación, adaptabilidad y capacidad de solución a problemas complejos.

Adicionalmente, el desarrollo de un sistema capaz de activar dispositivos electrónicos mediante comandos de voz permite proporcionar una nueva alternativa a las herramientas que hasta el momento han sido desarrolladas para dar solución al problema de la interacción humano-máquina, ofreciendo mayor comodidad a la hora de manipular un dispositivo electrónico y brinda facilidades

como la de ejecutar actividades multitarea. El sistema también sería de gran utilidad para permitir acceso tecnológico a personas con cierto tipo de discapacidades físicas que les impidan el uso de herramientas existentes como teclados, tableros de control, pantallas táctiles u otras.

1. REVISIÓN SOBRE EL RECONOCIMIENTO DE VOZ

En este capítulo se presenta una revisión de los conceptos fundamentales del reconocimiento de voz. Se hace una breve introducción al campo mediante la redacción de la realidad del tema en la actualidad, luego se describe la estructura general de un sistema de reconocimiento de voz y se explican con cierto detalle las partes que lo componen, finalmente se describen dos algoritmos muy usados como reconocedores: el alineamiento de tiempo dinámico y los modelos ocultos de Markov como posibles soluciones para el reconocimiento de voz.

1.1 EL RECONOCIMIENTO DE VOZ

Los sistemas de reconocimiento automático de voz (ASR por sus siglas en inglés, *Automatic Speech Recognition*) están empezando a aparecer en una gran variedad de sistemas de información. En automóviles o en pequeños celulares, el ASR (o en español, reconocimiento automático de voz) permite a los usuarios controlar dispositivos electrónicos sin usar molestos teclados. En otras aplicaciones, tales como en la búsqueda de archivos audio visuales no estructurados, el ASR, promete acceder a la información que de otro modo sería inaccesible debido a la dificultad de búsqueda a través de cientos de horas de grabaciones.

Cuando ASR se incorpora en un sistema de información se convierte solo en un aspecto de un complejo e interrelacionado conjunto de procedimientos automáticos. El rendimiento de los sistemas no se medirá por la taza de error por palabra, sino a través de un criterio de tareas específicas. Cuando se usa en teléfonos, por ejemplo, un objetivo típico podría ser identificar la persona a quien el usuario desea llamar mientras al mismo tiempo ignora todas las palabras que el usuario pronuncia. En otras aplicaciones tales como la minería de audio, el rendimiento de todos los sistemas se puede juzgar a través de la medida de la precisión y la memoria, más comúnmente usadas en la recuperación de información que en el ASR. Dado que diferentes medidas de rendimiento son probables para diferentes aplicaciones, es deseable crear sistemas ASR que se ajusten al criterio específico de la aplicación. De cualquier manera, las técnicas más difundidas para la medida del rendimiento recaen sobre el proceso de entrenamiento y decisión en la mayoría de sistemas ASR que no son susceptibles para objetivos de aplicación específica.

El reconocimiento de voz forma parte del ámbito general del entendimiento y de la inteligencia, siendo su objetivo último que una máquina sea capaz de comprender lo que se está pronunciando. Por tanto, lo importante no es detectar la secuencia de fonemas, sílabas o palabras pronunciadas, sino extraer las ideas que se

estructuran en palabras para que la máquina dé una respuesta en consecuencia y se pueda establecer un diálogo.

Todavía se está muy lejos de alcanzar este objetivo, aunque ya se están dando los primeros pasos, evolucionando desde reconocedores de palabras aisladas con vocabularios pequeños a reconocedores de grandes vocabularios con restricciones gramaticales, e incluso semánticas, estas últimas todavía muy incipientes. Al comparar el funcionamiento de los reconocedores de voz con grandes vocabularios en situaciones adecuadas con las capacidades humanas, se sabe que el humano comete aproximadamente diez veces menos errores que estos, si además el ambiente presenta inconvenientes o la voz se encuentra degradada la diferencia es aún mayor.

Generalmente, se utilizan técnicas que intentan solventar esa pérdida de las propiedades utilizando muestras de voz degradada. Sin embargo, estos mecanismos resultan muy rudimentarios si se comparan con los robustos tipos de adaptación que los seres humanos presentan frente a una gran multitud de situaciones, por ejemplo, con señales de voz recortadas o eliminación de ciertas bandas de frecuencia. Además, los seres humanos presentan una adaptación muy rápida a las variabilidades que se producen de manera natural, ya sean causadas por nuevos locutores, variaciones de la velocidad del habla, reverberación, la acústica de la sala y las características del canal debido a la recepción de señales reflejadas y otros efectos acústicos.

Finalmente se puede concluir que la capacidad de los seres humanos supera con creces a los reconocedores actuales en cuanto a la habilidad para distinguir nuevas palabras y sonidos ambientales no vocálicos de palabras correctas.[1]

1.2 TÉRMINOS Y DEFINICIONES EN EL RECONOCIMIENTO DE VOZ

La precisión del reconocimiento de voz depende de muchas características, entre las cuales se incluyen los usuarios potenciales, el ambiente físico y las restricciones impuestas por la tarea. En esta sección se pretende dar a conocer algunos de los parámetros que definen e influyen en el comportamiento y funcionamiento del reconocimiento de voz.

[1] CHOU, Wu y JUANG, Biing. Pattern recognition in speech and language processing. USA: CRC Press, 2003. p 64-65.

1.2.1 Independencia del locutor. Uno de los parámetros que diferencian a los sistemas de ASR es la cantidad de datos que el sistema debe almacenar para funcionar satisfactoriamente dependiendo de las características de voz de los usuarios potenciales. Los sistemas que requieren muestras específicas de la voz de un usuario se definen como sistemas dependientes del locutor. Estos sistemas deben tener una muestra de la manera en que un usuario dirá cada elemento del vocabulario antes de que el sistema pueda reconocer una palabra. Los sistemas dependientes del hablante, en general, demuestran una precisión más alta que otros tipos de sistemas, debido a que el sistema se calibra para una voz específica.

Los sistemas de reconocimiento de voz que son capaces de reconocer el lenguaje hablado por cualquier usuario potencial, se definen como independientes del locutor. Estos sistemas no requieren muestras de la voz de cada usuario potencial, en lugar de esto, el sistema usa algoritmos que son suficientemente robustos para reconocer a cualquier persona que hable un idioma determinado. Los sistemas independientes del hablante pueden alcanzar niveles de precisión similares a los sistemas dependientes del hablante; de cualquier manera, esto es usualmente verdad sólo cuando el conjunto de vocabulario de los sistemas es pequeño. Los sistemas independientes del locutor son útiles cuando múltiples usuarios pueden acceder a la aplicación de manera no muy frecuente, como por ejemplo en el encendido de la iluminación del pasillo de un hotel.

Los sistemas de reconocimiento de voz que empiezan como sistemas independientes del hablante, pero se modifican dependiendo de la voz de usuarios individuales a medida que el sistema recibe muestras de voz se definen como sistemas adaptativos dependientes del usuario. Estos tipos de sistemas ASR se diseñan para usarse por un conjunto estable de operadores que accederán al sistema repetidamente durante un tiempo determinado. Para estos sistemas la precisión del reconocimiento es relativamente alta, ya que las muestras de las características de voz para cada elemento del vocabulario son actualizadas continuamente en cada utilización del sistema.

1.2.2 Variabilidad de voz. Se describen cinco niveles de variabilidad en el lenguaje hablado. Estos niveles de variabilidad de voz incluyen familias de lenguas, idiomas distintos, dialectos, idiolectos, y las variaciones del individuo en el tiempo debido al estrés, la fatiga, o el estado de la salud física. Los dos niveles de variabilidades del habla que más impactan a los sistemas son el dialecto y los idiolectos (un idiolecto se define como el lenguaje o el patrón de voz de una persona en particular).

Los sistemas dependientes del hablante se ajustan mejor a estas variabilidades ya que ellos deben tener muestras de la manera en que cada usuario dirá cada elemento del vocabulario. Los sistemas independientes del hablante no presentan estas características. Esta es la razón por la cual los sistemas dependientes del

hablante presentan mayor precisión que los sistemas independientes, ya que el idiolecto y sus variaciones no se consideran en los algoritmos. Los sistemas adaptativos han resuelto algunas de las dificultades asociadas con la variabilidad de la voz humana.

1.2.3 Modo de hablar de los usuarios. Un tercer parámetro que diferencian los sistemas ASR es la manera en la cual un usuario tiene que hablar al sistema. Los sistemas que requieren que el usuario haga pausas cortas entre cada palabra del vocabulario, se conocen como reconocedores de palabras aisladas (en ocasiones también se les denomina reconocedores de entrada discreta de voz).

Los reconocedores que no requieren de pausas artificiales entre cada palabra se definen como reconocedores de palabras conectadas. Estos sistemas requieren que el usuario hable con el mismo patrón de entonamiento que se usaría si las palabras se estuviesen leyendo de una lista. Existe una distinción entre los reconocedores de palabras conectadas y los reconocedores de voz continua; un sistema de reconocimiento de voz continua permite al usuario hablar en su tono natural sin tener que pausar entre cada palabra del vocabulario.

Algunos sistemas ASR son capaces del entendimiento de voz continua. Tales sistemas tratan de cumplir las tareas usando entradas de voz continúa. La medida del rendimiento de tales sistemas no es el reconocimiento de palabras o el reconocimiento de la precisión del mensaje, sino en cambio, la precisión de la respuesta. Estos sistemas tratan de asignar un significado al mensaje que se les dijo y tratan de responder de acuerdo a ello.

Es importante notar que el modo de hablar de los usuarios de un sistema de reconocimiento de voz cambia de un simple reconocedor de palabras aisladas a un reconocedor de voz continua, de igual manera los requerimientos de procesamiento para tales tareas cambian y los de este último son muy elevados.

1.2.4 Tamaño del vocabulario. Un cuarto parámetro que diferencian los sistemas de reconocimiento de voz es el tipo y el tamaño del vocabulario del sistema. En general, hay dos tipos de vocabulario: limitado e ilimitado. Los sistemas de vocabulario limitado requieren que la persona diga muestras para cada elemento del vocabulario que el sistema es capaz de reconocer. Las aplicaciones de reconocimiento de voz se diseñan extrayendo elementos de vocabulario de diccionarios de posibles comandos de voz. Se realiza un apareamiento de los patrones acústicos entre las palabras pronunciadas por el usuario y las palabras almacenadas en los diccionarios. El tamaño de estos diccionarios puede variar en un rango de unas cuantas palabras a miles de ellas. Los diseñadores construyen los diccionarios de vocabulario dentro del sistema de reconocimiento y pueden ser modificados usando una estrategia dependiente o adaptativa al hablante. Un sistema de vocabulario ilimitado no requieren un diccionario de palabras, en lugar de ello, utiliza algoritmos que analizan las palabras pronunciadas en segmentos fonéticos. Una vez se tienen los segmentos fonéticos el sistema intenta determinar cual palabra fue realmente pronunciada y genera una respuesta apropiada. Sistemas de vocabulario ilimitado son útiles para aplicaciones tales como el dictado donde el número de posibles palabras puede ser muy grande.

1.2.5 Barreras de implementación. La tecnología ASR ha progresado inmensamente durante las dos últimas décadas. De cualquier manera, la tecnología se encuentra todavía en estado de desarrollo, todavía se está muy lejos de las capacidades de reconocimiento y entendimiento que presentan los humanos. Los diseñadores de sistemas complejos deben realizar sus tareas cuidadosamente antes de implementar tecnologías ASR como método de interfaz de un operador con una máquina. La tarea a ser desarrollada por los operadores, y las características de la población de los usuarios deben ser analizadas y entendidas antes de la implementación. El entender la naturaleza de la tarea a ser realizada y las características de los usuarios, pueden permitir la modificación de parámetros del sistema lo cual a su vez optimizara la idoneidad y la aceptación de este tipo de interfaces.[2]

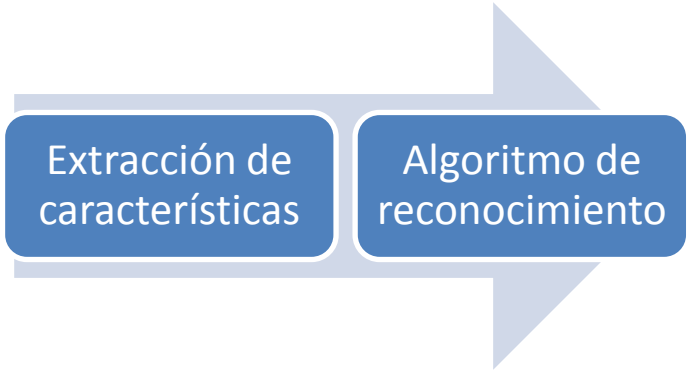
1.3 EXTRACCIÓN DE CARACTERÍSTICAS DE LA VOZ

Un reconocedor de voz se compone de dos bloques (ver Figura 1), el primero de ellos un extractor de características y el segundo un algoritmo de reconocimiento. La salida de un extractor de características es en cierto modo ciega, ya que no se preocupa por la palabra que se está analizando, este bloque solo transforma una señal de entrada en forma de onda sonora en una trayectoria representada mediante una serie de valores en un espacio de características. Específicamente, este bloque realiza la medida de los parámetros que conforman la voz y representa los eventos acústicos relevantes en la señal en términos de un

[2] GELLATLY, Andrew William. The use of speech recognition technology in automotive applications. Tesis de Doctorado. Virginia: Virginia Tech, 1997.

compacto y eficiente conjunto de parámetros. Y por otro lado, el algoritmo de reconocimiento se encarga de la compleja tarea de descubrir las relaciones y patrones que se presentan en los parámetros que conforman la voz, para reconocer las palabras diferenciándolas del resto que tiene almacenadas en su vocabulario.

Figura 1. Bloques de un sistema de reconocimiento de voz.



Como hipótesis se manifiesta que el algoritmo de reconocimiento depende directamente de la calidad con la que se extraigan los parámetros de la señal de voz, por eso es necesario buscar procedimientos que garanticen la entrega de datos significativos, por lo cual se han desarrollado muchas aproximaciones con buen grado de satisfacción, entre los algoritmos más usados se encuentran los *coeficientes cepstrales de frecuencia de Mel* y el modelo de *banco de filtros*.

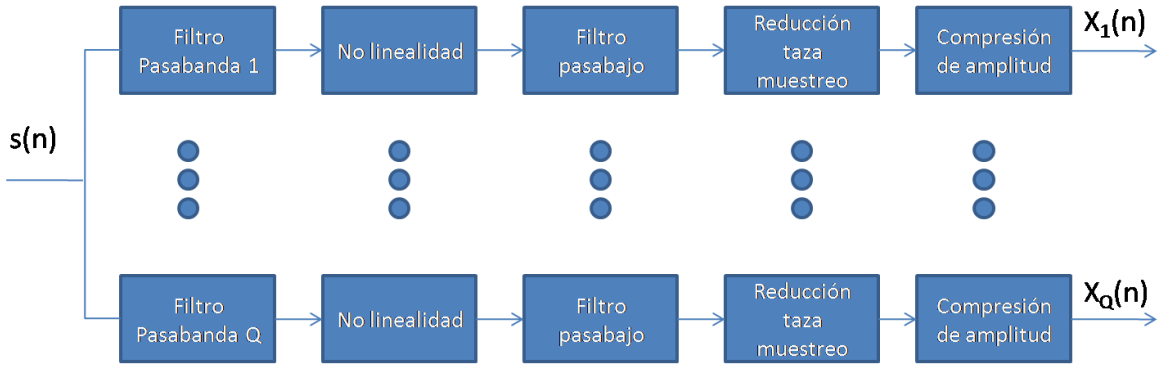
En un modelo de banco de filtros, la señal de voz previamente digitalizada se pasa a través de un banco de Q filtros pasabanda, cuya cobertura abarca el rango de frecuencias de interés en la señal (de 100 a 3000Hz para señales de calidad telefónica, de 100 a 8000Hz para señal con buen ancho de banda). Los filtros individuales generalmente se traslapan en frecuencia. Cada uno de los filtros pasabanda procesa los datos de manera individual y produce una representación diferente de la señal. Mientras que por otro lado, el modelo de los coeficientes cepstrales de frecuencia de Mel inicia con la realización un análisis espectral de bloques de voz, lo que generalmente se realiza mediante la transformada rápida de Fourier o la codificación de predicción lineal. Luego del análisis espectral se realiza una conversión de los parámetros obtenidos mediante un algoritmo de transformación y se generan los coeficientes cepstrales de frecuencia de Mel que son quienes contienen la información más representativa de la señal.

1.4 BANCO DE FILTROS

Como se describió previamente, un banco de filtros en el procesamiento de voz (ver Figura 2), es una estructura de Q filtros pasabanda que se utilizan para tener un análisis discriminativo en cada banda de frecuencia, pero ellos no pueden

funcionar solos, ya que si fuese de este modo, la información de salida sería Q veces mayor a la información de entrada y esto dificultaría aún más el proceso de reconocimiento, por lo que se añade un procesamiento posterior, como se ilustra a continuación:

Figura 2. Diagrama de bloques del banco de filtros.



Una muestra de señal $s(n)$ (que representa una señal de voz previamente digitalizada) se pasa a través de Q bancos de filtros, obteniendo las señales:

$$s_i(n) = s(n) * h_i(n), \quad 1 \leq i \leq Q \qquad (1)$$

$$s_i(n) = \sum_{m=0}^{M_i-1} h_i(m)s(n-m) \qquad (2)$$

Donde se asume que $h_i(m)$ es la respuesta al impulso del i-ésimo filtro pasabanda con una duración de M_i muestras; se usa la forma de convolución de la operación de filtrado para dar una representación explicita a la señal filtrada por uno de los filtros pasabanda, lo cual se muestra mediante la expresión $s_i(n)$. Ya que el propósito de un analizador de banco de filtros es dar una medida de la energía de la señal en una banda de frecuencias dada, cada una de las señales filtradas $s_i(n)$, se pasa a través de un elemento no lineal, tal como un rectificador de onda completa o un rectificador de media onda. La no linealidad cambia el espectro de la banda de paso, a una banda baja de frecuencias y crea imágenes de alta frecuencia. Se usa un filtro pasabajo para eliminar las imágenes de alta frecuencia, dando un conjunto de señales que representan la energía de la señal en cada una de las Q bandas de frecuencia. La reducción de la taza de muestreo representa un algoritmo de compresión de información, ya que lo que se pretende es generar un conjunto compacto de parámetros que representen la señal,

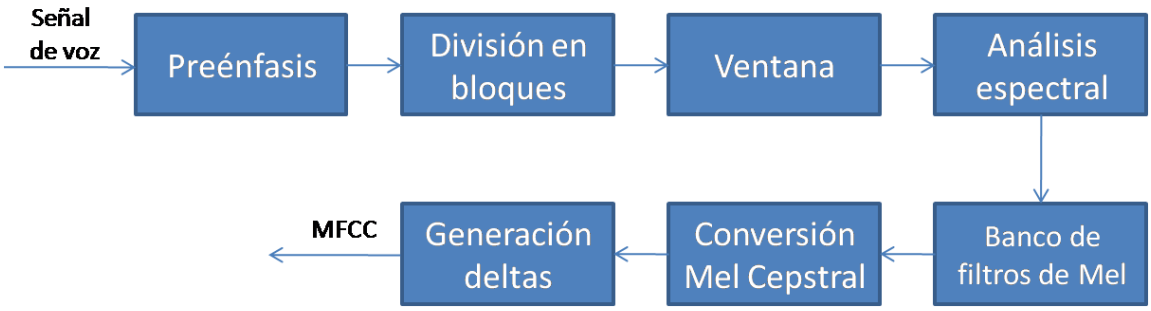
además se utiliza la compresión en amplitud para mantener la información en un rango de valores adecuado para su posterior procesamiento.[3]

1.5 COEFICIENTES CEPSTRALES DE FRECUENCIA DE MEL

Los coeficientes cepstrales de frecuencia de Mel (MFCC por sus siglas en inglés, *Mel Frecuency Cepstral Coefficients*) son la manera más común y más usada de extraer las características de una señal de voz, se basan en el comportamiento del oído humano a bandas de frecuencia específicas. Esta técnica hace uso de dos tipos de filtros, llamados filtros linealmente espaciados y filtros logarítmicamente espaciados. Para capturar las características fonéticas importantes de la voz, la señal se expresa en la escala de frecuencias de Mel. Esta escala posee un espaciamiento lineal debajo de los 1000Hz y un espaciamiento logarítmico por encima de los 1000Hz correspondiente al comportamiento del oído humano. Ya que la forma de onda normal de la voz puede variar dependiendo de la condición física de las cuerdas vocales del hablante, es aconsejable usar los MFCC que son menos susceptibles a dicho tipo de variaciones.

La Figura 3, muestra un diagrama de bloques del proceso que se lleva a cabo para implementar el análisis de coeficientes cepstrales de frecuencia de Mel:

Figura 3. Diagrama de bloques MFCC.



Preénfasis: la señal digitalizada de voz $s(n)$, se pasa a través de un sistema digital de bajo orden, para "pulir" espectralmente la señal y hacerla menos susceptible a efectos de precisión finita en el posterior procesamiento de la señal. La red de preénfasis más usada en el reconocimiento de voz es:

$$H(z) = 1 - az^{-1}, \quad 0.9 \leq a \leq 1 \quad (3)$$

[3] RABINER, Lawrence y JUANG Biing. Fundamentals of Speech Recognition. New Jersey: Prentice Hall International, 1993.

Para ese caso, la señal de salida de la red de preénfasis $s_1(n)$, se relaciona con la entrada $s(n)$, por la ecuación de diferencias:

$$s_1(n) = s(n) - as(n-1) \quad (4)$$

División en bloques: en este paso la señal de salida del preénfasis $s_1(n)$, se divide en bloques de N muestras. Usualmente las N muestras de los bloques de señal usados para reconocimiento de voz representan de $10\ a\ 20\ mS$, y además se recomienda que los bloques se encuentren traslapados para conservar toda la información que se transporta en la señal.

Ventana: El siguiente paso en el procesamiento es aplicar una ventana a cada bloque individual, con la finalidad de minimizar las discontinuidades de la señal en el comienzo y fin de cada bloque. La ventana más común para este paso, es la ventana de Hamming, la cual tiene la forma:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1 \quad (5)$$

Análisis espectral: Para realizar el análisis espectral se puede escoger entre varios procedimientos, los más utilizados en el reconocimiento de voz son la FFT (una descripción detallada del algoritmo y la implementación se puede mirar en la sección 4.3) y el LPC.

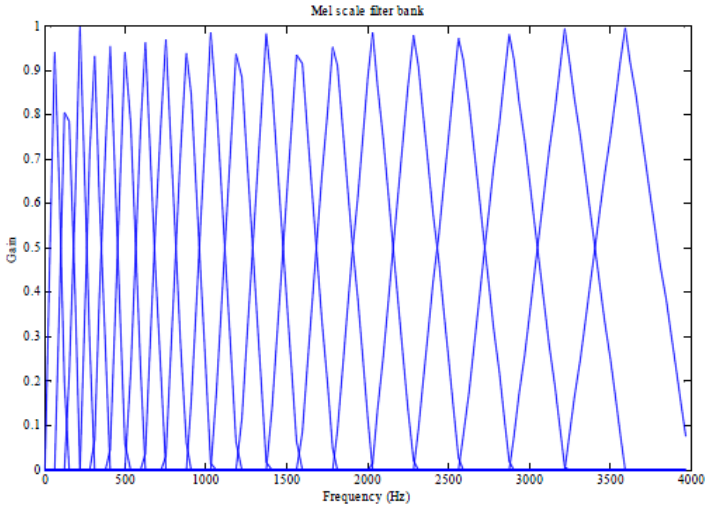
Banco de filtros de Mel: La señal de voz consiste de tonos con diferentes frecuencias. Para cada tono con una frecuencia real f, medida en Hz, es posible medir un tono subjetivo en la escala de Mel. La escala de frecuencias de Mel posee un espaciamiento aproximadamente lineal debajo de los $1000Hz$ y un espaciamiento logarítmico encima de los $1000Hz$. Como un punto de referencia se plantea que un tono de $1kHz$, de $40dB$ por encima del umbral de escucha perceptual que equivale a $1000\ mels$. Se puede usar la siguiente fórmula para computar los $mels$ para una frecuencia determinada en Hz:

$$mel(f) = 2595\ log_{10}\left(1 + \frac{f}{700}\right) \quad (6)$$

Una de las aproximaciones usadas para simular este espectro es usar un banco de filtros. Se implementa un filtro para cada componente de frecuencia en la escala de Mel. El banco de filtros usado tiene una respuesta de banda de paso

triangular, y el espaciamiento al igual que el ancho de banda se determina por un intervalo constante de frecuencias en la escala de Mel.

Figura 4. Banco de filtros de Mel.



Fuente. THRASYVOULOU, T. y BENTON, S. Speech parameterization using the Mel scale Part II [online] 2003 [cited: Diciembre 2009]. Available from Internet: http://www-2.cs.cmu.edu/~mseltzer/sphinxman/

Por consideraciones prácticas, se crean filtros solo hasta $4kHz$ ya que el espectro de la voz humana solo abarca hasta los $3300Hz$, además ya que el presente proyecto se desarrolla sobre un sistema embebido la taza de muestreo no permite realizar análisis superiores a los $6KHz$. La cantidad de filtros a emplear es un parámetro a discusión, pero se han probado sistemas de reconocimiento de voz con resultados satisfactorios que emplean entre 10 y 25 filtros. Se considera $S(k)$ como la salida del análisis espectral realizado, y $M_l(k)$ el l-ésimo filtro del banco de filtros, L la cantidad de filtros que conforman al bando de filtros, N la cantidad de datos obtenida del análisis espectral, de esta manera se obtiene una aproximación del espectro en la escala de Mel mediante la ecuación:

$$S_f(l) = \sum_{k=0}^{N/2} S(k) M_l(k)\,, \qquad 0 \le l < L-1 \quad (7)$$

Conversión Mel cepstral: En este paso, se pretende convertir el espectro logarítmico de frecuencias de Mel de nuevo al espacio del tiempo. La representación cepstral del espectro de la voz provee una buena grafía de las propiedades locales del espectro de la señal para cada bloque de voz tratado. Ya que los coeficientes son números reales, ellos se pueden convertir al dominio del

tiempo mediante la transformada discreta del coseno (DCT, por sus siglas en inglés *Discrete Cosine Transformation*).

$$c(i) = \sqrt{\frac{2}{L}} \sum_{m=1}^{L} log\big(S_f(m)\big) cos\left(\frac{\pi i}{L}(m - 0.5)\right), \quad 0 \leq i < C \quad (8)$$

Donde C es la cantidad de coeficientes deseada. Se ha probado experimentalmente que resultados satisfactorios son obtenidos cuando se escogen entre 10 y 20 coeficientes como representación espectral.[4]

Generación de deltas: La derivada en tiempo de los coeficientes cepstrales se aproxima a un polinomio ortogonal de primer orden sobre una ventana de longitud finita de $2k + 1$ bloques de voz, centrados alrededor del bloque actual. Se han probado resultados experimentales satisfactorios con $k = 2$.

$$\Delta c(m) = \left[\sum_{k=-K}^{K} k c_{t-k}(m)\right] G, \quad 1 \leq m \leq C \quad (9)$$

El vector definitivo de características del bloque de voz analizado, corresponde a concatenar los datos de Δc y c, en un vector de la forma $Q = \{c\ , \Delta c\}$.[5]

1.6 MODELOS OCULTOS DE MARKOV

Los modelos ocultos de Markov (HMM por sus siglas en inglés, *Hidden Markov Model*), son un método estadístico de caracterización usado en el reconocimiento de voz para determinar las propiedades que muestran patrones en cada uno de los bloques de señal procesados. El supuesto subyacente de este y todos los modelos estadísticos es que la señal de habla puede ser caracterizada como un proceso aleatorio paramétrico, y que los parámetros del proceso estocástico pueden ser determinados o estimados de alguna manera concreta.

La teoría básica de los HMM fue publicada en una serie de documentos por Baum y sus colegas a finales de los sesentas y comienzos de los setentas, y fue implementada para aplicaciones de reconocimiento de habla por Baker en el CMU, y por Jelinek y sus colegas en IBM en los setentas.

---

[4] RASHIDUL, Hassan y MUSTAFA, Jamil. Speaker identification using Mel frecuency cepstral coefficients. Dhaka: 3rd International Conference on Electrical & Computer Engineering, 2004.

[5] RABINER, Lawrence. A tutorial on hidden Markov models and selected applications in speech recognition. USA: Proceedings of the IEEE, 1989. P 277.

1.6.1 Procesos discretos de Markov. Considere un sistema en el cual se puede saber en todo momento en cuál de los $\boldsymbol{N}$ diferentes estados se encuentra. En tiempos discretos espaciados regularmente, el sistema sufre un cambio de estado (aunque existe la posibilidad de quedarse en el mismo estado) de acuerdo a un conjunto de probabilidades asociadas con el estado. Se denotan los instantes de tiempo asociados con los cambios de estado como $\boldsymbol{t} = \mathbf{1}, \mathbf{2}, \ldots$, y se denota el estado actual en el tiempo $\boldsymbol{t}$ como $\boldsymbol{q_t}$. Una completa descripción probabilística del sistema descrito, requeriría especificaciones del estado actual, tanto como de los estados predecesores. Para el caso especial de una cadena de Markov discreta de primer orden, esta descripción probabilística se trunca solo para el estado actual y el predecesor.

$$P\left[q_t = S_j \,|\, q_{t-1} = S_i, q_{t-2} = S_k, \ldots\right] = P\left[q_t = S_j \,|\, q_{t-1} = S_i\right] \quad (10)$$

Asimismo, solo se considera los procesos en los que el lado derecho de la ecuación anterior, es independiente del tiempo, de esta manera se genera un conjunto de probabilidades de transición a_{ij} de la forma:

$$a_{ij} = P\left[q_t = S_j \,|\, q_{t-1} = S_i\right], \;\; 1 \leq i, \;\; j \leq N \quad (11)$$

Donde los coeficientes de probabilidad de transición tienen las propiedades

$$a_{ij} \geq 0 \quad (12)$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad (13)$$

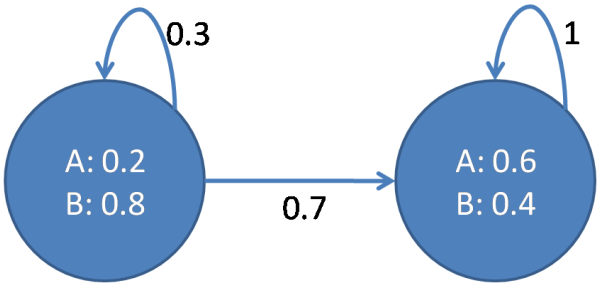
Ya que estos obedecen a las restricciones estándar de los procesos estocásticos.

El proceso estocástico descrito anteriormente podría ser considerado como un modelo observable de Markov, ya que la salida del proceso es el conjunto de estados en cada instante de tiempo, donde cada estado corresponde a un evento físicamente observable.

1.6.2 Extensión a modelos ocultos de Markov. Hasta ahora se ha considerado modelos de Markov en los cuales cada estado corresponde a un evento físicamente observable. Este modelo es demasiado restrictivo para ser aplicable a muchos problemas de interés. La expansión de este tipo de modelos es llamada Modelos Ocultos de Markov.

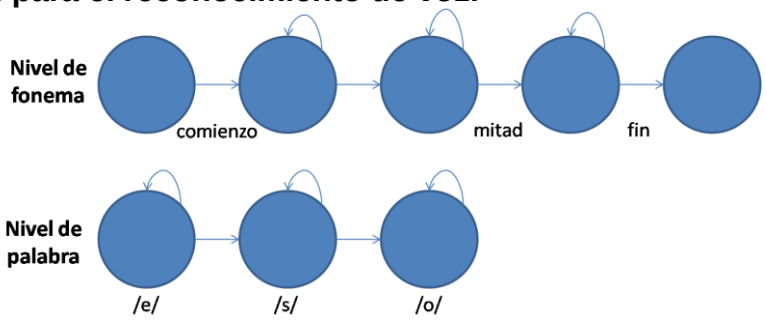
Un HMM es un conjunto de estados conectados por transiciones, como se muestra en la Figura 5, el modelo empieza en un estado inicial designado y en cada paso discreto de tiempo, realiza una transición a un nuevo estado, y se genera un símbolo de salida en ese estado. La selección de la transición y el símbolo de salida son aleatorios, pero se rigen por una distribución de probabilidad. El HMM se puede pensar como una caja negra, donde la secuencia de símbolos generados en el tiempo es observable, pero la secuencia de estados visitados sobre el tiempo es oculta. Esta es la razón por la cual se denominan modelos ocultos de Markov.

Figura 5. Ejemplo de HMM.



Los HMM tienen una gran variedad de aplicaciones dentro del reconocimiento de patrones. Cuando un HMM se utiliza en el reconocimiento de voz, los estados son interpretados como modelos acústicos, indicando que sonidos tienen probabilidad de ser escuchados durante un segmento específico de voz; mientras que las transiciones proveen las restricciones temporales, indicando la manera en que los estados pueden estar en secuencia. Ya que el habla siempre va hacia adelante en el tiempo, las transiciones en una aplicación siempre van hacia adelante o hacen un ciclo sobre el mismo estado de duración arbitraria. La Figura 6, muestra algunos tipos de HMM usados en el reconocimiento de voz.

Figura 6. HHM para el reconocimiento de voz.



Formalmente un HMM se caracteriza por los siguientes elementos:

- N, el número de estados en el modelo. Aunque los estados son ocultos, para muchas aplicaciones prácticas existe alguna relevancia significativa adjunta a los estados, como por ejemplo si se trata de un reconocedor en el nivel de palabra cada estado puede corresponder a un fonema.

- M, la cantidad de símbolos diferentes que puede llegar a emitir un estado. Los símbolos de las observaciones corresponden a las características físicas de la salida del sistema que está siendo modelado.

- La distribución de probabilidad de transición de estados $A = \{a_{ij}\}$ donde

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], \ 1 \leq i, \ j \leq N \quad (14)$$

- La distribución de probabilidad de los símbolos de observación en el estado j, $B = \{b_j(k)\}$, donde

$$b_j(k) = P[v_k \ en \ t | \ q_t = S_j], \ 1 \leq j \leq N, \ 1 \leq k \leq M \quad (15)$$

- La distribución inicial de probabilidad $\pi = \{\pi_i\}$ donde

$$\pi_i = P[q_1 = S_i], \ 1 \leq i \leq N \quad (16)$$

1.6.3 Los tres problemas básicos de los HMM. Conociendo la manera en que está compuesto un HMM, existen tres problemas básicos que deben ser resueltos para ser útiles en problemas reales. Estos problemas son:

- Dada una secuencia de observaciones de la forma $\boldsymbol{O} = O_1 O_2 O_3 \dots O_T$, y un modelo $\lambda = (A, B, \pi)$, ¿Cómo se calcula eficientemente $P(\boldsymbol{O}|\lambda)$, la probabilidad de una secuencia de observación dado el modelo?

- Dada una secuencia de observaciones de la forma $\boldsymbol{O} = O_1 O_2 O_3 \dots O_T$, y un modelo $\lambda = (A, B, \pi)$, ¿Cómo se selecciona una secuencia de estados $\boldsymbol{Q} = q_1 q_2 q_3 \dots q_T$ la cual sea óptima en alguna manera significativa (aquella que mejor explica las observaciones)?

- ¿Cómo ajustar los parámetros del modelo $\lambda = (A, B, \pi)$ para maximizar $P(\boldsymbol{O}|\lambda)$?

El primer problema es denominado problema de evaluación, dado un modelo y una secuencia de observaciones, como se calcula la probabilidad de que la secuencia observada fue producida por el modelo. Aunque también se puede pensar el problema en cómo calificar la probabilidad del modelo ante una secuencia de observaciones. Esto último es extremadamente útil, ya que si se analiza el caso de un reconocedor de palabras aisladas, se determina la palabra que se ha dicho escogiendo aquella que tenga la calificación más alta.

En el segundo problema se trata de destapar la parte oculta del modelo, para encontrar una secuencia "correcta" de estados. Se puede mirar que para la mayoría de modelos, *no existe* una secuencia *correcta* de estados. Por lo tanto, para situaciones prácticas, se usa un criterio de optimización para solucionar el problema en la mejor manera posible. Desafortunadamente, existen muchos criterios de optimización que se ajustan para dar solución al problema, y entonces la selección del criterio es una función que depende fuertemente del uso que se le vaya a dar a la solución.

En el tercer problema se trata de optimizar los parámetros del modelo para describir de una forma óptima, la manera en que ocurre una determinada secuencia de observación. La secuencia de observación usada para ajustar los parámetros del modelo es denominada secuencia de entrenamiento ya que es usada para *entrenar* al HMM. El problema de entrenamiento es el más importante para la mayoría de aplicaciones, ya que permite modificar los parámetros de los modelos para ajustarse a los datos de observación, lo cual hace posible crear modelos que se ajusten mejor al fenómeno.

La solución al primer problema es conocida como el algoritmo directo (ó en inglés, *forward algorithm*) y es supremamente útil en la elaboración de reconocedores de palabras aisladas.

La solución al segundo problema es conocida como el algoritmo de Viterbi (o en inglés, *Viterbi algorithm*) y es supremamente útil en la elaboración de reconocedores de habla continua.

La solución al tercer problema es conocida como el algoritmo directo inverso (o en inglés, *forward backward algorithm*) y es usada para el entrenamiento de los modelos de las dos aplicaciones anteriores[6].

1.6.4 Limitaciones y problemas de implementación. Aunque la tecnología de los HMM ha contribuido enormemente al desarrollo del reconocimiento de voz, hay algunas limitaciones inherentes de este tipo de modelo estadístico. Una gran limitante es el hecho de asumir que las observaciones sucesivas (bloques de voz) son independientes, y que por lo tanto la probabilidad de que una secuencia de observaciones sea producida, se pueda escribir como el producto de las probabilidades individuales.

$$P(O_1 O_2 O_3 \dots O_T) = \prod_{i=1}^{T} P(O_i) \quad (17)$$

Otra limitante, es la suposición de que las distribuciones de los parámetros de las observaciones individuales pueden ser bien representadas como una mezcla de modelos de densidades auto regresivos. Y finalmente, la suposición de Markov en sí misma, que propone que la probabilidad de encontrarse en un estado en el tiempo t, solo depende del estado en $t-1$, lo cual es claramente inapropiado en la señal de voz donde las dependencias a menudo se extienden a través de muchos estados.

También existen problemas relacionados con la implementación sobre las tecnologías actuales. Uno de ellos es que se hace necesaria la incorporación de un modelo que permita crear una escala para la realización de operaciones en el cálculo de las probabilidades. Ya que para la implementación del *forward algorithm* las probabilidades de transición y emisión se multiplican constantemente y se encuentran en el rango de 0 a 1, por lo que cada término generado en el proceso crece exponencialmente hacia cero. Si el número de bloques de habla que representan una determinada palabra es sustancialmente grande el rango dinámico de las computaciones excederá la precisión de cualquier máquina. Lo

[6]Ibid., p. 257- 286.

cual obliga a que la única manera razonable de realizar el computo sea incorporando un procedimiento de escalado.

Los modelos más usados dentro del reconocimiento de habla son los llamados modelos de izquierda a derecha (o en inglés, *left-right models*) los cuales obligan a usar más de una sola secuencia para poder entrenar el modelo. Esto se debe a que la naturaleza transitoria de los estados dentro del modelo solo permite un pequeño número de observaciones para cualquier estado (hasta que se hace la transición al estado siguiente). Por lo cual, para tener datos suficientes para lograr estimados confiables de los parámetros del modelo, se debe usar muchas secuencias de observación.

Cuando se utiliza el algoritmo directo inverso para reestimar los parámetros del HMM, teóricamente se tiene que corresponde a un máximo local de la función de probabilidad. Una pregunta clave es, ¿cómo seleccionar valores estimados de los parámetros del modelo para que el máximo local sea tan cercano como sea posible al máximo global de la función de probabilidad? Realmente, no existe una respuesta directa o simple. La experiencia ha demostrado que la matriz de emisiones puede ser inicializada en valores aleatorios cualquiera logrando un resultado muy semejante al final del proceso, pero en cambio la matriz de transmisiones puede lucir muy diferente dependiendo de los valores en los cuales se inicializa.

Otro problema asociado con el entrenamiento de los parámetros de los HMM a través de reestimación es que la secuencia de observación usada es necesariamente finita. Así que, muchas veces existe un número inadecuado de ocurrencias de eventos con muy baja probabilidad e incluso valores con probabilidad cero que dificultan aún más el problema de escalamiento. Una de las soluciones es aumentar el tamaño del conjunto de datos de observación, lo cual a menudo es impráctico. Una segunda solución es reducir el tamaño del modelo (número de estados, número de símbolos por estado). Aunque esto, siempre es posible, muchas veces hay razones físicas por las cuales se usa un modelo dado, y por lo tanto este no puede ser cambiado. Una tercera solución es buscar algoritmos estadísticos no convencionales que de alguna manera puedan mejorar la confiabilidad de los parámetros estimados basados en un conjunto de entrenamiento limitado. Aunque no exista una solución determinada para el problema, cualquiera de las estrategias anteriores puede implementarse para atenuar esta dificultad y lograr la generación de un modelo con resultados satisfactorios.

De cualquier manera, a pesar de las limitaciones y dificultades de implementación, este tipo de modelo estadístico ha funcionado extremadamente bien en muchos sistemas de reconocimiento de habla durante muchos años y es uno de los más usados en los dispositivos móviles debido a que requiere un nivel bajo de operaciones.

1.7 ALINEAMIENTO DE TIEMPO DINÁMICO

El alineamiento de tiempo dinámico (DTW por sus siglas en inglés, *Dynamic Time Warping*) es una técnica de comparación de patrones. Está técnica aventaja a las técnicas normales debido a que permite que la palabra a ser reconocida tenga un tamaño diferente al vector de referencia que se usa para comparar. La diferencia de tamaño se debe a las variaciones de la longitud de una palabra en el lenguaje hablado normal. Estas variaciones causan que los bloques de habla estén fuera de alineamiento. DTW opera seleccionando cual bloque del vector de referencia se ajusta mejor a cada bloque del vector de entrada, de tal manera que el error resultante total sea minimizado. [7]

1.7.1 Bloque de comparación de patrones, una aproximación clásica. El bloque de comparación de patrones es donde todas las decisiones se hacen, de tal manera que la precisión del reconocimiento depende de que tan precisas sean estas decisiones. El proceso de clasificación de patrones se realiza con la idea de seleccionar a que clase pertenece una determinada entrada de voz, y dependiendo de esto, se genera la palabra reconocida como salida.

Un componente clave de la mayoría de algoritmos de comparación de patrones es una medida de las diferencias entre dos vectores de características. Escrito de manera sencilla, la distorsión representa una distancia entre los vectores de características. La manera más fácil de medir la distancia entre dos vectores de características de dos señales de voz, es la distancia euclidiana. Para dos vectores de características $\boldsymbol{x}$ y $\boldsymbol{y}$, la distancia euclidiana entre ellos se encuentra dada por:

$$d(x,y) = \sqrt{(x-y)^2} \quad (18)$$

Esto representa una distancia local entre los vectores de características $\boldsymbol{x}$ y $\boldsymbol{y}$, de las señales de voz $\boldsymbol{X}$ y $\boldsymbol{Y}$. Aunque la métrica euclidiana es un poco más costosa computacionalmente que otras métricas, da más peso a grandes diferencias en una sola característica. También se ha mostrado que esta métrica tiene muchas propiedades teóricas deseables cuando se comparan coeficientes cepstrales.

La distancia local ayuda a determinar la similitud total entre dos señales de voz $\boldsymbol{X}$ y $\boldsymbol{Y}$. Si ambas, $\boldsymbol{X}$ y $\boldsymbol{Y}$ consisten de n vectores de características de cualquier orden, entonces la distancia global entre estos vectores está dada por:

[7] MUZAFFAR, Fariha, et al. DSP Implementation of Voice Recognition Using Dynamic Time Warping Algorithm. Karachi: Student Conference on Engineering Sciences and Technology, 2005.

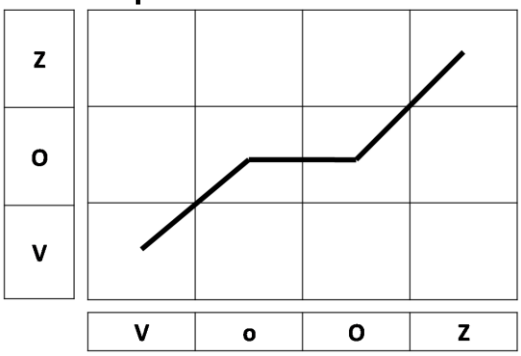
$$d(x,y) = \sum_{i=1}^{n} \sqrt{(x_i - y_i)^2} \qquad (19)$$

La distancia global expresa la diferencia computacional entre las dos señales de voz. Esta medida de distorsión puede ser usada como un comparador de patrones para sistemas de reconocimiento, pero posee ciertos inconvenientes. Mide la distancia local entre dos vectores de características de una manera completamente lineal. No toma en cuenta las fluctuaciones del hablante, o la situación donde las señales de voz tienen diferentes longitudes (las señales de voz generalmente no tienen la misma duración). En estas situaciones, esta medida de distorsión falla y la precisión alcanzada es muy baja. En general, se sabe que debido a la naturaleza del habla las señales tienen diferentes tamaños, por lo que se hace necesaria la implementación de un algoritmo que permita solucionar este inconveniente.

1.7.2 La solución DTW. El habla es un proceso dependiente del tiempo. Por lo tanto las repeticiones de una misma palabra pueden tener duraciones diferentes, además iteraciones de la misma palabra con la misma duración pueden diferir en la zona media, debido a que partes de las palabras pueden estar siendo pronunciadas a diferente velocidad. Para obtener una distancia global entre dos patrones de voz (representados como una secuencia de vectores) debe realizarse un alineamiento en el tiempo.

DTW es una técnica de programación dinámica, en la cual el problema completo se divide en un pequeño número de pasos, donde cada uno requiere una decisión basada en las medidas de distancia local. La decisión total se hace dependiendo de todas las pequeñas decisiones, por lo cual se considera que DTW se basa en la aproximación *divide y vencerás*. DTW garantiza encontrar el camino con menor distancia a través de la matriz, mientras minimiza la cantidad de computaciones requeridas.

Figura 7. Alineamiento de tiempo mediante DTW.



El problema se muestra en la figura 7, en el cual se usa una matriz "tiempo-tiempo" para visualizar el alineamiento. El patrón de referencia se ubica en el eje vertical y el patrón de entrada en el horizontal. En esta ilustración la entrada "VoOz" es una versión con ruido del patrón de referencia "VOZ". La idea es que "o" está más cercana a "O" comparada con cualquier otro elemento en la referencia. Para la realización de un sistema de reconocimiento la entrada "VoOz" debe compararse con todas las referencias que se tengan en el repositorio.

La referencia que mejor se ajusta es aquella que tenga el menor camino de distancia de alineamiento de la entrada con referencia al patrón almacenado. Un puntaje para una distancia global simple de un camino es la suma de las distancias locales que conforman el camino. Si $D(i,j)$ es la distancia global hasta (i,j) y la distancia local en (i,j) está dada por $d(i,j)$ entonces:

$$D(i,j) = min[D(i-1,j-1), D(i-1,j), D(i,j-1)] + d(i,j) \quad (20)$$

Se asume que $D(1,1) = d(1,1)$ es una condición inicial del problema. Este es un algoritmo recursivo muy eficiente para realizar el computo de $D(i,j)$. La distancia global obtenida al final del proceso es $D(n,N)$ que muestra la distancia total entre los dos patrones. La palabra reconocida es entonces aquella con el menor puntaje de distancia global. El camino empieza en la esquina inferior izquierda y termina en la esquina superior derecha, con el fin de evitar análisis en presencia de inversión en el tiempo.[8]

1.8 DETECCIÓN DE VOZ

El objetivo de la detección de voz (en inglés, *Speech Endpoint Detection*) consiste en separar los eventos acústicos de interés en una señal de habla continua de otras partes de la señal (tales como el ruido del medio). La necesidad de la detección de voz ocurrió en muchas aplicaciones de telecomunicaciones. Por ejemplo, en sistemas de transmisión análogos multicanal, existe una técnica llamada asignación de tiempo por interpolación de voz (TASI, por sus siglas en inglés *Time Assignment Speech Interpolation*), usada para tomar ventaja del tiempo muerto de cada canal a través de la detección de la presencia de voz de un hablante y la asignación de un canal solo cuando se detecta la voz, con el objetivo de permitir que más clientes puedan usar el sistema de transmisión. Por otro lado, en el ASR, la detección de voz se utiliza para aislar la voz de interés del ruido ambiental con la finalidad de crear patrones de voz para su reconocimiento.

La detección de voz en segmentos de habla se vuelve relativamente difícil en ambientes ruidosos, pero es definitivamente importante para el reconocimiento

[8] BIN AMIN, Talal y MAHMOOD Iftekhar. Speech Recognition Using Dynamic Time Warping. Islamabad: 2nd International Conference on Advances in Space Technologies, 2008.

robusto de voz. La energía espectral ha sido el elemento más usado como parámetro de característica para diferenciar la voz de otro tipo de señales. De cualquier manera, estas características se vuelven menos confiables y robustas en ambientes ruidosos, especialmente en la presencia de ruido no estacionario, respiración fuerte, golpe de labios, entre otras. Para solucionar este tipo de problemas se han presentado muchos métodos, entre ellos, el método basado en la entropía en los dominios del tiempo y la frecuencia, a lo cual se le denomina entropía espectral. En esta aproximación, inicialmente se estima la función de densidad de probabilidad (PDF, por sus siglas en inglés *Probability Density Function*) de cada bloque de voz, en el cual se define y mide la entropía espectral. Se ha encontrado experimentalmente que el valor de la entropía espectral es muy útil para distinguir los segmentos de voz en una expresión, de los segmentos de no voz, especialmente bajo condiciones de ruido.

1.8.1 Detección de voz basada en entropía. En los algoritmos convencionales de detección de voz, la energía espectral se usa como fuente primaria para generar parámetros de características, usualmente con la adición de la taza de cruce por cero, tono e información de duración. Pero estas se vuelven menos confiables en presencia de ruido no estacionario o con algunos artificios de sonido. La detección de voz mediante entropía soluciona algunos de estos inconvenientes. Lo primero que se realiza en este procedimiento es un espectrograma de la voz. Para cada bloque, se obtiene el espectro mediante la FFT. El espectro de la FFT puede verse como un vector de coeficientes. La PDF para el espectro puede estimarse por normalización sobre todos los componentes de frecuencia mediante:

$$p_i = \frac{s(f_i)}{\sum_{k=1}^{N} s(f_k)}, \quad 1 \leq i \leq N \quad (21)$$

Donde $s(f_i)$ es la energía espectral para el componente de frecuencia f_i, p_i es la densidad de probabilidad correspondiente a esa frecuencia, y N es el número total de componentes de frecuencia de la FFT. Para mejorar la discriminación de la PDF entre señales de voz y no voz, se han desarrollado muchas restricciones empíricas. Primero, solo los componentes de frecuencia entre $250Hz$ y $6000Hz$ son considerados:

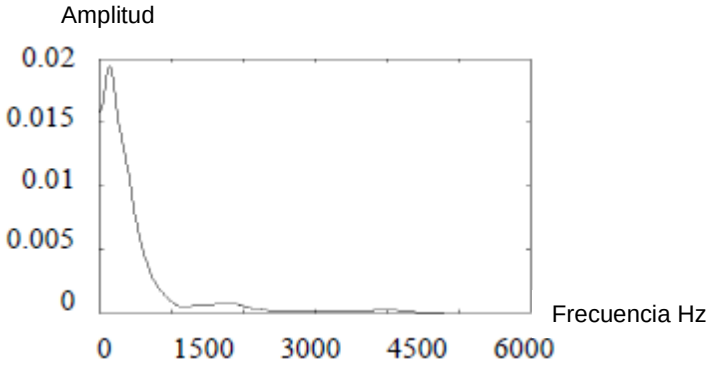
$$s(f_i) = 0, si\ f_i < 250Hz \text{ ó } f_i > 6000Hz \quad (22)$$

Esto se debe a que en esta región se cubre la mayoría de componentes de frecuencia de una señal de voz. Segundo, se aplican una barrera superior y una barrera inferior a las densidades de probabilidad:

$$p_i = 0, \qquad si\ p_i < \delta_2\ \text{ó}\ p_i > \delta_1 \quad (23)$$

Donde la barrera inferior δ_2 se usa para cancelar el ruido espectral con valores de densidades de potencia espectral casi constantes sobre todas las frecuencias, como por ejemplo el ruido blanco, mientras que la banda superior δ_1 se usa para eliminar el ruido concentrado en algunas de las bandas específicas de frecuencia. De igual manera, se puede añadir un conjunto de pesos para ajustar el componente de frecuencia a la entropía espectral. Estos pesos son calculados estadísticamente a partir de un gran conjunto de señales de voz.

Figura 8. Pesos utilizados en la detección de voz



Fuente. SHEN, Jia-lin; HUNG, Jeigh-weigh y LEE Lin-shan. Robust Entropy-based Endpoint Detection for Speech Recognition in Noisy Environments. Taipei: Institute of Information Science, 1998.

Después del proceso de normalización y mejora, la entropía espectral correspondiente para cada bloque se puede definir como:

$$H = -\sum_{k=1}^{N} w_k p_k log(p_k) \quad (24)$$

En el proceso de detección de voz, inicialmente se evalúa la suma de los valores de la entropía espectral sobre la duración de un bloque de voz y se suaviza mediante un filtro a través de la palabra. Luego, se utilizan niveles de mantenimiento para detectar el inicio y el fin de los segmentos de habla dentro del sonido que se está analizando. Un periodo de ruido ambiental se toma como

referencia para la detección de las barreras iníciales, y los niveles de mantenimiento se calculan de acuerdo al análisis de algunas señales de voz que se usen como calibración del sistema.[9]

1.9 RECONOCIMIENTO DE VOZ EN SISTEMAS EMBEBIDOS

La voz provee un método natural y directo para la interacción humano-máquina, ya que es una de las principales maneras de comunicación humana. La voz tiene muchas ventajas sobre otros métodos de comunicación, lo que hace que los sistemas ASR sean muy atractivos. Las personas pueden hablar entre 4 y 5 veces más rápido de lo que pueden escribir, lo cual permite una comunicación más efectiva, además cuando alguien está hablando las manos y los ojos están libres para realizar otro tipo de actividades. Algunos dispositivos con reconocimiento de voz, tales como los micrófonos y los celulares son pequeños y sencillos permitiendo la posibilidad de tener una tecnología con movilidad. Por último, ya que el lenguaje es parte de la cultura humana, no se requiere una práctica específica para el reconocimiento de voz, por lo cual la interacción con los dispositivos se puede extender a todo tipo de usuarios.

En los últimos años, diferentes tipos de implementaciones de algoritmos de reconocimiento de voz han aparecido en algunos dispositivos. En celulares y sistemas telefónicos automáticos, el reconocimiento de voz se usa para manejar un número limitado de comandos y controles de aplicaciones. Idealmente, un sistema de reconocimiento de voz con gran precisión, amplio vocabulario, independiente del hablante, continuo y de funcionamiento en tiempo real tendría enormes posibilidades para aplicaciones. Poder conversar libremente con las máquinas a través de diálogos naturales podría dar a muchas aplicaciones una interfaz mucho más amigable para el usuario. Desafortunadamente, hasta ahora, debido a la limitación de recursos, su uso ha sido restrictivo. Para crear sistemas de reconocimiento de voz que sean robustos, con gran vocabulario y continuos, se deben señalar tres problemas fundamentales: la precisión en el reconocimiento, velocidad de decodificación y complejidad.

Hasta hace poco, la investigación en el reconocimiento de voz se ha preocupado principalmente por lograr una gran precisión en sistemas continuos, de gran vocabulario e independientes del hablante, usualmente perdiendo velocidad y aumentando la complejidad como resultado del proceso. Para mantener tasas de reconocimiento elevadas a medida que el vocabulario incrementa, la investigación se ha concentrado en modelar el discurso hablado en dos niveles, el nivel acústico y el nivel de lenguaje. A pesar del incremento en el reconocimiento, la complejidad de usar un modelado acústico y un modelado de lenguaje ha hecho que la velocidad de reconocimiento y la eficiencia computacional vayan en decremento. A medida que la investigación en el campo del reconocimiento de voz se lleva a una

[9] SHEN, Jia-lin; HUNG, Jeigh-weigh y LEE Lin-shan. Robust Entropy-based Endpoint Detection for Speech Recognition in Noisy Environments. Taipei: Institute of Information Science, 1998.

escala mayor, la velocidad, la memoria necesaria, y el tamaño del hardware computacional requerido se vuelve problemático. Cuando el reconocimiento se hace en una máquina con altas prestaciones (como un computador de escritorio) muchas de estas preocupaciones pueden ser ignoradas. Pero para pequeños dispositivos, estos problemas limitan la usabilidad de los algoritmos para los sistemas portables y móviles. La implementación de sistemas de reconocimiento de voz en arquitecturas diseñadas de hardware/software permite el apalancamiento de la flexibilidad de la solución de software y también permite un mejor entendimiento de cómo se debería hacer el reconocimiento de voz para la maximización de su rendimiento, de tal manera que pueda ser colocado en sistemas embebidos móviles. En una aproximación de solo software, la precisión, rapidez y costo computacional se debe ajustar, esto depende del tipo de algoritmos corriendo sobre la CPU. El desarrollo de una solución donde se mezcle la arquitectura de hardware y software, la velocidad de decodificación y el costo computacional pueden ser optimizados a través de la explotación de las sinergias del hardware y software.

La investigación ha mostrado que los mejores sistemas de reconocimiento de voz utilizan información acústica y de lenguaje de manera conjunta para descifrar palabras. La idea detrás de este esquema es trabajar en nivel de fonemas y en el contexto de las palabras. Esta aproximación de dos niveles se ha desarrollado para lidiar con los problemas de ambigüedad y de mala articulación. En particular, el modelado acústico es necesario para las tareas de amplio vocabulario y los modelos de lenguaje son necesarios para manejar la ambigüedad de las palabras, ya que estas ayudan a diferenciar entre palabras confusas.

Las restricciones de complejidad y velocidad obstaculizan el uso de los sistemas de reconocimiento de amplio vocabulario. Mucha investigación se ha hecho para incrementar la velocidad y reducir la complejidad a través de software. Esto ha tenido algo de resultados satisfactorios, pero para lograr mayor velocidad y potencia de los sistemas, la introducción de más hardware parece inevitable. Investigaciones recientes en el reconocimiento de voz han resuelto el problema de rendimiento a través de un diseño de software cuidadoso y un proceso de calibración. En software, correr el reconocimiento de voz sobre un procesador de alto nivel aún es limitante porque la estructura de memoria del procesador no se encuentra optimizada para el trabajo sobre las señales de voz.[10]

[10] AMONE, Luigi; BOCCHIO, Sara y ROSTI, Alberto. On embedded system architectures for speech recognition applications: the gap between the status and the demand. Italy: *Fourth IEEE International Symposium on Signal Processing and Information Technology*, 2004.

2. REVISIÓN SOBRE REDES NEURONALES ARTIFICIALES

En el capítulo anterior se introdujo la temática referente al reconocimiento de voz, ahora se va a presentar las redes neuronales como una atractiva solución al problema de reconocimiento de patrones, especialmente al reconocimiento de fonemas en el campo del ASR. Se empieza con una descripción de las generalidades de las redes neuronales artificiales, se hace una breve descripción de su historia, su desarrollo y su inspiración biológica. Luego se avanza a la especificación de las características principales de las redes neuronales, se muestra un procedimiento de entrenamiento y se termina con una explicación del uso de las redes neuronales en el campo del reconocimiento de voz.

2.1 FUNDAMENTOS DE LAS REDES NEURONALES

Las redes neuronales artificiales (ANN, por sus siglas en inglés *Artificial Neural Networks*) se componen de elementos simples operando en paralelo. Estos elementos se inspiran en el sistema nervioso humano. Al igual que en la naturaleza, las conexiones entre los elementos determinan la función que realiza la red. Es posible entrenar una red neuronal para realizar una determinada función ajustando los valores que caracterizan las conexiones entre los elementos.

Usualmente, las redes neuronales se ajustan, o entrenan, para que una entrada particular conduzca a una salida u objetivo específico. Normalmente se requieren muchos pares de entrada-salida para entrenar una red neuronal.

Las redes neuronales han sido entrenadas para realizar funciones complejas en muchos campos, incluyendo el reconocimiento de patrones, la identificación, la clasificación, la voz, la visión y los sistemas de control.

El conexionismo o estudio de las redes neuronales artificiales, estaba inicialmente inspirado por la neurobiología, pero ahora se ha vuelto un campo interdisciplinario incluyendo la ciencia de computación, ingeniería eléctrica y electrónica, matemáticas, física, sicología, lingüística entre otras. Algunos investigadores aún se encuentran estudiando la neurofisiología del cerebro humano, pero se está prestando mucha atención a las propiedades de la neuro computación, usando modelos neuronales simplificados. Dentro de estas propiedades se incluyen:

1. Entrenabilidad. Se puede enseñar a las redes a formar asociaciones entre alguna entrada y ciertos patrones de salida. Esto puede usarse, por ejemplo, para enseñarle a una red a clasificar patrones de habla en categorías de fonemas.

2. Generalización. Las redes no solo memorizan los datos de entrenamiento, sino que aprenden los patrones que subyacen de estos, así pueden generalizar de un cierto grupo de datos a nuevos ejemplos. Esto es esencial en el reconocimiento de voz, porque los patrones de voz nunca son exactamente los mismos.

3. No linealidad. Las redes pueden operar funciones no paramétricas y no lineales de los datos de entrada, lo que les permite realizar transformaciones complejas de los datos de entrada. Esto es útil si se considera que la voz es un elemento altamente no lineal.

4. Robustez. Las redes toleran daños físicos y datos ruidosos; incluso los datos ruidosos pueden permitir que las redes generen mejores generalizaciones. Esta es una característica muy importante, porque los patrones de voz capturados suelen ser ruidosos.

5. Uniformidad. Las redes ofrecen un paradigma de cómputos uniformes lo cual les permite integrar restricciones a diferentes tipos de entradas.

6. Paralelismo. Las redes son altamente paralelas por naturaleza siendo muy ajustables para implementación sobre computadores de características paralelas. Esto permite un procesamiento más veloz de cualquier tipo de datos.

Hay muchos tipos de modelos conexionistas, con diferentes arquitecturas, procedimientos de entrenamiento y aplicaciones, pero todos se basan en ciertos principios comunes. Una red neuronal artificial consiste de un alto número de elementos simples de procesamiento (neuronas), cada uno de los cuales influencia el comportamiento de las demás a través de una red excitadora o inhibidora de pesos. Cada neurona simplemente computa una función no lineal de la suma del producto de las entradas por los pesos y envía el resultado a través de sus conexiones de salida a otras neuronas. Un conjunto de entrenamiento consiste de patrones de valores que se asignan a las unidades de entrada y a las de salida. Ya que los patrones se adoptan del conjunto de entrenamiento, es necesaria una regla de entrenamiento que modifique la fuerza de los pesos de tal manera que la red gradualmente aprenda el conjunto de entrenamiento.[11]

2.2 CARACTERÍSTICAS DE LAS ANN

A pesar de que existen muchos tipos de redes neuronales artificiales, todos ellos tienen cuatro atributos básicos:

- Un conjunto de unidades de procesamiento

[11] JAIN, Anil y MAO, Jianchang. Artificial Neural Networks: A Tutorial. En: IEEE Computer Science Magazin. Marzo 1996. Vol 29, no 3.

- Un conjunto de conexiones
- Un procedimiento de computo
- Un procedimiento de entrenamiento

2.2.1 Unidades de procesamiento. Una red neuronal contiene un gran número de unidades de procesamiento sencillo, que tratan de asemejarse a las neuronas del cerebro. Todas estas unidades pueden operar *simultáneamente*, siendo capaces de procesar en paralelo. Todas las computaciones realizadas por el sistema son basadas en estas unidades; no existe otro procesador que supervise o coordine su actividad. En cada momento, cada unidad computa una función escalar de sus entradas locales, y genera el resultado llamado valor de activación que se transmite a sus unidades vecinas.

Las unidades en una red son usualmente divididas en unidades de entrada, las cuales reciben la información del ambiente (tal como un arreglo de sensores); unidades ocultas, las cuales internamente transforman la representación de la información; y las unidades de salida, las cuales representan decisiones o señales de control.

En gráficas de redes neuronales, dichas unidades se representan por círculos. El estado de una red en cualquier momento se representa por el conjunto de valores de activación sobre todas las unidades; el estado de la red usualmente varía en cada instante, al igual que el cambio de las entradas y la realimentación en el sistema, lo cual causa que la red siga una trayectoria dinámica a través del espacio de estados.

2.2.2 Conexiones. Las unidades que conforman una red se organizan en una determinada topología por un conjunto de conexiones, o pesos, mostradas como líneas en los diagramas. Cada peso tiene un valor real, usualmente en el rango de $-\infty$ a ∞, aunque en algunos problemas el rango es limitado. El valor (o la fuerza) del peso describe cuanta influencia una unidad tiene sobre su vecina; un valor positivo causa que una unidad excite a la siguiente, mientras que un peso negativo causa que una unidad inhiba a la siguiente.

Los valores de los pesos determinan la reacción computacional de la red neuronal a cualquier patrón de entrada arbitrario, así se puede decir que los pesos codifican la memoria de largo plazo, o el conocimiento de la red. Los pesos pueden cambiar como resultado de un entrenamiento, pero tienden a cambiar lentamente, ya que el conocimiento acumulado cambia de esa manera en el cerebro humano.

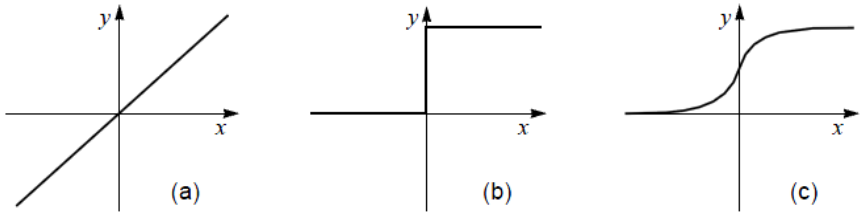
2.2.3 Cómputo. El cómputo siempre comienza presentando un patrón de entrada a la red, o un patrón de activación a las neuronas (unidades) de entrada. Luego la activación de todas las unidades restantes se computa, ya sea de manera síncrona (si se cuenta con una unidad con capacidad de procesamiento paralelo) o asíncrona (en una máquina como un micro procesador, el orden en el cual se opera solo depende de la manera de programación realizada). Este procedimiento se conoce como propagación directa (en inglés, *forward propagation*), ya que progresa de la capa de entrada a la capa de salida. Los cómputos de una red neuronal terminarán tan pronto como la propagación alcance la capa de salida, aunque en casos especiales como las redes recurrentes, las activaciones nunca se estabilizarán, pero podrían seguir una trayectoria específica a través del espacio de estados cuando las unidades son actualizadas continuamente.

Una unidad se activa típicamente en dos etapas: la primera es el cómputo de la entrada de la red, y luego el computo de la salida de activación como función de la entrada de la red. Esto se puede expresar matemáticamente como:

$$x_j = \sum_i y_i w_{ji} \quad (25)$$

Donde y_i es la salida de activación de una unidad que actúa como entrante para la unidad actual, w_{ji} es el peso de la unidad i a la unidad j, x_j representa el valor de entrada a la neurona, con este último valor se calcula la salida de activación y_j como una función de x_j. Las funciones de activación (o funciones de transferencia) de cada neurona pueden tomar diferentes formas, pero entre las más usadas se tienen:

Figura 9. Funciones de activación (a) lineal (b) mantenimiento (c) sigmoidal



La Figura 9 (a), se caracteriza por la ecuación $y = x$, la Figura 9 (b) es una función definida a tramos:

$$y = \begin{cases} 0 \; si \; x \leq 0 \\ 1 \; si \; x > 0 \end{cases} \quad (26)$$

Y la Figura 9 (c), es la más usada en el reconocimiento de patrones se define como:

$$y = \frac{1}{1+e^{-x}} \quad (27)$$

Esta última forma parte de las llamadas funciones sigmoidales, que se caracterizan por presentar las ventajas de no linealidad, continuidad, diferenciabilidad, lo cual permite a una ANN multicapa computar cualquier valor real, además de soportar diferentes tipos de algoritmos de entrenamiento.

2.2.4 Entrenamiento. Entrenar una red, en el sentido más general, significa adaptar sus conexiones para que la red muestre un comportamiento de salida deseado para todos los patrones de entrada. El proceso involucra la modificación de los pesos. Encontrar un conjunto de pesos que permita a la red lograr un objetivo específico no es un procedimiento trivial. Una solución analítica solo existe en el caso sencillo de asociación de patrones, cuando la red es lineal y el objetivo es mapear un conjunto de vectores de entrada ortogonales a ciertos vectores de salida.

La mayoría de los procedimientos de entrenamiento se encuentran basados, en la regla de Hebb (en inglés, *Hebb rule*), la cual refuerza la conexión entre dos unidades si sus salidas de activación están correlacionadas:

$$\Delta w_{ji} = \varepsilon y_i y_j \quad (28)$$

Reforzando la correlación entre pares de unidades activos durante el entrenamiento, la red se prepara para activar a la segunda unidad si se conoce la primera durante la etapa de prueba.

Una variación importante de la anterior regla, es la regla Delta, la cual aplica cuando existe un valor objetivo para alguna de las dos unidades. Esta regla refuerza la conexión entre las dos unidades si existe una correlación entre la activación de la primera unidad y_i y el error de la segunda unidad relativo a su objetivo t_j:

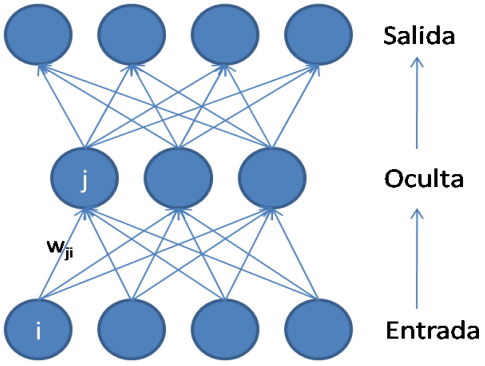
$$\Delta w_{ji} = \varepsilon y_i (t_j - y_j) \quad (29)$$

Esta regla disminuye el error relativo, haciendo que la red compute una salida y_j más cercana a t_j conociendo el valor de activación de y_i durante el procedimiento.

2.3 RETRO PROPAGACIÓN

El algoritmo de retro propagación, también conocido como retropropagación del error o regla Delta generalizada, es el algoritmo de entrenamiento más usado para redes neuronales artificiales.

Figura 10. Modelo de red neuronal artificial.



Suponga que se tiene una red multicapa de unidades no lineales (típicamente sigmoidales), como la mostrada en la figura 10. Se quiere encontrar los valores de los pesos que permitan que la red compute una función deseada desde los vectores de entrada hacia los vectores de salida. Ya que las neuronas o unidades computan funciones no lineales, no es posible solucionar el problema analíticamente; por lo tanto se debe resolver el problema mediante un procedimiento que busque una reducción del error global E.

Si se define i, j, k como índices arbitrarios para las neuronas, O como el conjunto de neuronas de salida, p como los índices de los patrones de entrenamiento (donde cada patrón de entrenamiento contiene un vector de entrada y un vector objetivo), x_j^p como la entrada a la neurona j para el patrón p, y_j^p como la salida de activación de la neurona j para el patrón p, w_{ji} como el peso entre la neurona i y la neurona j, t_j^p como la activación objetivo para la unidad j en el patrón p, E^p como el error global de la salida para el patrón de entrenamiento p, y E como el error global para el conjunto completo de entrenamiento. Asumiendo el tipo más común de red neuronal con activación sigmoidal, se tiene:

$$x_j^p = \sum_i w_{ji}\, y_j^p \quad (30)$$

$$y_j^p = \sigma\left(x_j^p\right) = \frac{1}{1+e^{-x_j^p}} \quad (31)$$

Es esencial que esta función de activación $y_j^p = \sigma\left(x_j^p\right)$ sea diferenciable, porque dentro del algoritmo de retropropagación es necesario calcular el gradiente de la misma.

La selección de la función mediante la cual se va a medir el error es algo arbitraria, pero para ejemplificación se utiliza una medida mediante suma del error cuadrático.

$$E^p = \frac{1}{2}\sum_j \left(y_j^p - t_j^p\right)^2 \quad (32)$$

$$E = \sum_p E^p \quad (33)$$

Para modificar cada peso w_{ji} en proporción a su influencia sobre el error E, en la dirección en la cual se reduce E:

$$\Delta^p w_{ji} = -\varepsilon \frac{\partial E^p}{\partial w_{ji}} \quad (34)$$

Donde ε es una constante con valor pequeño, denominada taza de aprendizaje.

Por la regla de la cadena, es posible expandir la derivada como sigue:

$$\frac{\partial E^p}{\partial w_{ji}} = \frac{\partial E^p}{\partial y_j^p}\frac{\partial y_j^p}{\partial x_j^p}\frac{\partial x_j^p}{\partial w_{ji}} \quad (35)$$

$$\frac{\partial E^p}{\partial w_{ji}} = \gamma_j^p \sigma'\left(x_j^p\right) y_i^p \quad (36)$$

El primero de estos tres términos, $\gamma_j^p = \frac{\partial E^p}{\partial y_j^p}$ permanece sin ser expandido. La manera exacta en la cual va a ser expandida depende de si j es una neurona de salida o no. Si j es una unidad de salida, entonces:

$$j \in O \Rightarrow \gamma_j^p = \frac{\partial E^p}{\partial y_j^p} = y_j^p - t_j^p \quad (37)$$

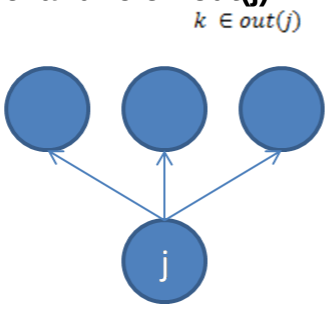
Pero, si j no es una unidad de salida, entonces afecta directamente a un conjunto de $k \in out(j)$ unidades, y por la regla de la cadena se obtiene:

$$j \notin O \Rightarrow \gamma_j^p = \frac{\partial E^p}{\partial y_j^p} = \sum_{k \in out(j)} \frac{\partial E^p}{\partial y_k^p} \frac{\partial y_k^p}{\partial x_k^p} \frac{\partial x_k^p}{\partial y_j^p} \quad (38)$$

$$j \notin O \Rightarrow \gamma_j^p = \frac{\partial E^p}{\partial y_j^p} = \sum_{k \in out(j)} \gamma_k^p \sigma'(x_k^p) w_{kj} \quad (39)$$

La recursividad en esta ecuación, en la cual γ_j^p se refiere a γ_k^p, significa que los γ (y por lo tanto los Δw) en cada capa pueden ser derivados directamente de los γ en la siguiente capa. Así entonces, se derivan los γ en una red multicapa si se empieza en la capa de salida (37) y trabajando el recorrido hacia atrás hasta la capa de entrada, una capa a la vez (39) Este procedimiento de aprendizaje se denomina retro propagación debido a que los términos de error γ se propagan a través de la red en la dirección inversa.

Figura 11. Explicación de la función out(j)



Si la neurona j no es una salida, entonces afecta a k unidades en la siguiente capa

En resumen, se tiene:

$$\Delta^p w_{ji} = -\varepsilon\, \gamma_j^p \sigma'\left(x_j^p\right) y_i^p \quad (40)$$

Donde:

$$\gamma_j^p = \begin{cases} y_j^p - t_j^p, & si\ j \in O \\ \sum_{k \in out\,(j)} \gamma_k^p \sigma'\left(x_k^p\right) w_{kj}\ , & si\ j \notin O \end{cases} \quad (41)$$

El aprendizaje del algoritmo puede ser acelerado incrementando la taza de aprendizaje ε, pero solo hasta cierto punto, porque cuando este valor se vuelve muy grande, los pesos se vuelven excesivos, las unidades se saturan y por lo tanto el aprendizaje se vuelve imposible. Por esta razón, se han desarrollado gran número de procedimientos heurísticos para acelerar el aprendizaje. Estas técnicas generalmente están motivadas por una imagen intuitiva de que la retro propagación es un procedimiento que maneja un gradiente descendente. Esto quiere decir que si se imagina un paisaje montañoso que representa la función de error E sobre el espacio de los pesos, entonces el algoritmo trata de encontrar un valor mínimo local de E tomando pasos mediante valores incrementales Δw_{ji} hacia abajo en dicho lado de la colina, en la dirección proporcionada por $-\partial E^p / \partial w_{ji}$. Esta imagen ayuda a mirar, por ejemplo, que si se toma un paso muy largo, se cae en el riesgo de moverse tan lejos hacia un lado de la colina que se pueda pasar a otra colina vecina. [12]

2.4 ANN EN EL RECONOCIMIENTO DE VOZ

En algunas ocasiones se piensa que las ANN solo han sido usadas recientemente para aplicaciones con datos reales. Pero la verdad es que ha existido una explosión de interés y de aplicaciones sobre ANN durante las 3 últimas décadas, y aún más las ANN han sido utilizadas para el reconocimiento de patrones por más de 40 años. En la década de los 80, un número de investigadores empezaron a aplicar aproximaciones con ANN para clasificación de voz, y particularmente para aplicaciones ASR. A pesar de que se tiene un conocimiento relativamente profundo sobre las señales de voz, el ASR aún es un reto de ingeniería complicado. Esto se debe a diversas razones, pero en parte se debe a que este campo se mueve bajo la premisa de rendimiento a nivel humano bajo condiciones reales, lo cual aún continúa siendo un problema sin solución.

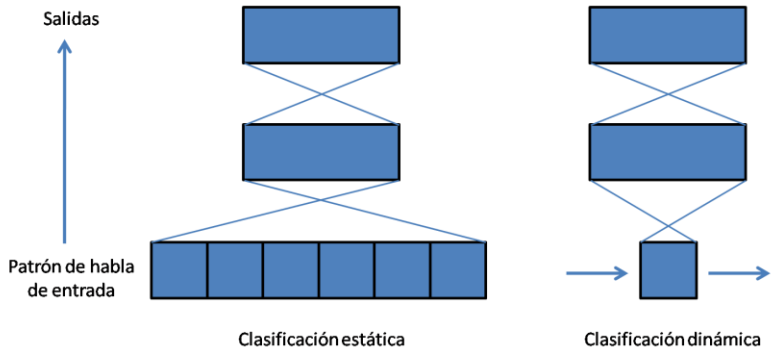
[12] TEBELSKIS, Joe. Speech recognition using neural networks. Tesis de Doctorado. Pittsburgh: Carnegie Mellon University, 1995.

Aunque es verdad que ASR es un problema difícil y que las ANN pueden ser útiles en el reconocimiento de patrones complejos, se debe ser cauteloso antes de pensar que la *magia* de las ANN puede implementar un sistema ASR completo. Problemas de reconocimiento de patrones, son raramente implementados por un componente monolítico, es decir no solo basta con usar una red neuronal o cualquier otro componente homogéneo. En particular, aún no se ha encontrado una manera para implementar de manera satisfactoria un sistema basado solo en redes neuronales para el reconocimiento de voz. Pero de cualquier manera, se ha aprendido la manera en que se pueden usar las ANN como componente clave en tales sistemas.

Los primeros sistemas con ANN involucraban tareas muy sencillas como el determinar las secciones de voz y secciones de no voz en una grabación. El éxito en estas aplicaciones motivó a los investigadores a buscar la solución de la clasificación de fonemas; esta tarea resultó ser la prueba de que las redes neuronales poseen las características necesarias para la tarea de reconocimiento de voz. Las mismas técnicas se implementaron para el reconocimiento en el nivel de palabra con un poco de éxito, aunque se volvió evidente que existen problemas de escala y duración en esta aproximación.

Existen dos aproximaciones para la clasificación de voz usando ANN: la aproximación estática y la aproximación dinámica. En clasificación estática, la red neuronal mira toda la entrada de voz al mismo tiempo y toma una sola decisión. En contraste, en clasificación dinámica, la red neuronal solo ve una pequeña ventana (o bloque de voz), y esta ventana se desliza sobre la señal de entrada mientras que la red realiza una serie de decisiones locales, las cuales deben ser integradas en un sistema de decisión global en un paso posterior. La clasificación estática presenta un buen rendimiento para el reconocimiento individual de fonemas, pero se escala pobremente para el nivel de palabras u oraciones; mientras que la clasificación dinámica se escala mejor.

Figura 12. Clasificación estática y dinámica.



El reconocimiento a nivel de fonemas es una de las maneras más usadas para crear reconocedores de palabras aisladas, pero para esta tarea involucra la implementación de un sistema que permita tomar una decisión global sobre los fonemas. Es aquí donde empiezan a tomar gran importancia los sistemas híbridos de reconocimiento, entre los más importantes se encuentran los ANN/HMM y los ANN/DTW donde la red neuronal es la encargada de clasificar los patrones de entrada en probabilidades de fonemas y un modelo de decisión (un algoritmo HMM o DTW) es el encargado de retomar estos datos y mirar cual patrón de palabra es el que más se asemeja a la información almacenada. Básicamente la tarea de la red neuronal se convierte en ser un facilitador para el sistema de decisión final, ya que se encarga de transformar el complejo conjunto de datos obtenidos mediante el pre procesamiento (o extracción de características) y transformarlos en otro conjunto de datos que presenta características más definidas y sencillas, lo que permite lograr mejores resultados y permite elevar el nivel de rendimiento y reconocimiento del sistema en general.

Se han desarrollado una gran cantidad de trabajos sobre redes neuronales en el reconocimiento de voz, en el 2000, Lin Cong y sus colegas[13] elaboraron un híbrido implementado mediante ANN y HMM para el reconocimiento de dígitos, su trabajo supuso un avance interesante porque se elaboró un sistema independiente del hablante y además con robustez ante el ruido presentado en automóviles. En el 2002, Salwa H. El-Ramly y sus colegas[14] presentaron su investigación sobre la aplicación de redes neuronales recurrentes y perceptrones multicapa para el reconocimiento de fonemas en el lenguaje Árabe obteniendo excelentes resultados, principalmente con las redes neuronales recurrentes. En el 2006, Iman Abuel Maaly y Manal El-Obaid[15] realizaron un trabajo mediante redes neuronales para el reconocimiento de fonemas independientemente del hablante en Árabe preocupándose principalmente por el porcentaje de reconocimiento y logrando valores superiores al 99%. En el 2009, Lalith Kumar y Kishore Kumar[16], desarrollaron un reconocedor de dígitos que en su bloque de reconocimiento solo usaba redes neuronales, pero sus porcentajes de reconocimiento aún se encuentran muy por debajo de los sistemas híbridos. También en 2009, Xiang Tang[17], implementó un reconocedor de dígitos independiente del hablante mediante un hibrido ANN/HMM y lo comparó con un sistema mediante HMM, se demostró que el hibrido presenta mejores resultados en condiciones de no ruido, y

[13] CONG, Lin, et al. Robust Speech Recognition Using Neural Networks and Hidden Markov Models. Las Vegas: Information Technology Coding and Computing, 2000.

[14] EL-RAMLY, Salwa, et al. Neural Networks Used for Speech Recognition. Alexandria: Nineteenth National Radio Science Conference, 2002.

[15] MAALY, Iman y EL-OBAID, Manal. Speech Recognition using Artificial Neural Networks. Damascus: Information and Communications Technologies, 2006.

[16] KUMAR, Kishore y KUMAR, Lalithl.Speech Recognition Using Neural Networks. Singapore: International Conference on Signal Processing Systems, 2009.

[17] TANG, Xiang. Hybrid Hidden Markov Model and Artificial Neural Network for Automatic Speech Recognition. Wuhan City: Communications and System Pacific-Asia Conference on Circuits, 2009.

cuando aumenta el nivel de ruido, la diferencia en reconocimiento es aún mayor entre los dos sistemas.

En los últimos tres años ha existido una tendencia a la exploración de nuevas metodologías para el reconocimiento de fonemas y el reconocimiento de voz en general, entre los temas existentes se puede mencionar las maquinas de soporte vectorial (SVM, por sus siglas en inglés *support vector machines*) y los algoritmos basados en cúmulos de partículas (PSO, por sus siglas en inglés *particle swarm optimization*) que se utilizan como complemento o modificación para otros algoritmos.

La perspectiva futura en el desarrollo del ASR, no solo depende del reconocimiento a nivel de fonemas, palabras u oraciones, sino que ahora se empieza a pensar en reconocedores que funcionen con características a nivel humano, esto incluye vocabularios muy amplios, la posibilidad de adquirir nuevas palabras, funcionamiento en tiempo real, diferenciación en ambientes ruidosos, entre otras características; donde entran a jugar estrategias de procesamiento en niveles más altos tales como el semántico y el sintáctico, donde la inteligencia computacional se vuelve parte fundamental y el entendimiento del discurso hablado humano es el objetivo final, así se pretende que las interfaces humano máquina se vuelvan mucho más naturales e intuitivas. Aunque en la actualidad el estudio de las redes neuronales en estos niveles ya se encuentra en proceso, el desarrollo es muy limitado y el futuro del ASR se verá beneficiado cuando estas y otras estrategias se trabajen a fondo y se optimicen, logrando acercarnos a ese entendimiento del lenguaje humano que se ha perseguido durante las últimas décadas.

3. RECONOCEDOR DE DÍGITOS EN LENGUAJE DE ALTO NIVEL

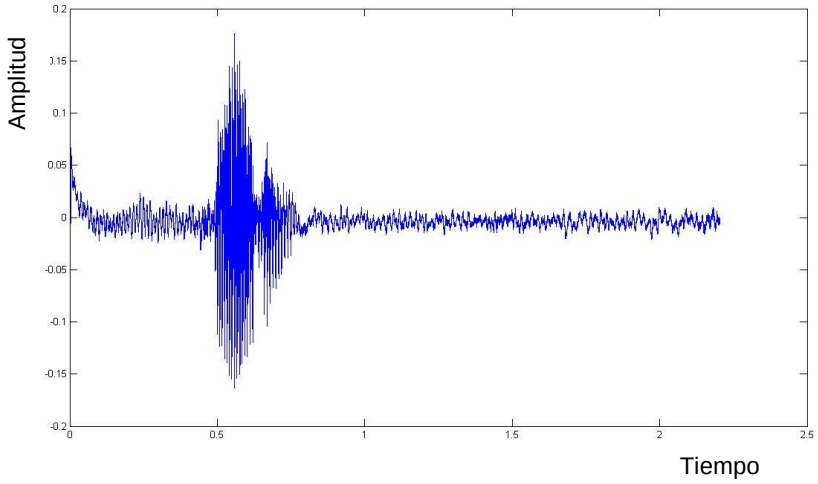
Se ha presentado una revisión de las temáticas que involucra el reconocimiento de voz y algunas de las temáticas de las redes neuronales artificiales y su aplicación en el ASR. Ahora, en este capítulo se describen la implementación y las pruebas experimentales del reconocedor fonético de voz en lenguaje de alto nivel.

3.1 EXTRACCIÓN DE CARACTERÍSTICAS DE LA VOZ

Para el desarrollo del presente proyecto se seleccionó los Coeficientes Cepstrales de Frecuencia de Mel como algoritmo para extraer las características de la señal digitalizada de voz. La selección de los MFCC se hace considerando que estos presentan un comportamiento muy similar al del oído humano, además porque a pesar de no ser muy robustos ante el ruido, son capaces de disminuir la diferencia entre las pronunciaciones de los diferentes hablantes. Los MFCC destacan las características que componen la voz humana y así facilitan el trabajo que debe realizar cualquier algoritmo de reconocimiento.

La mayoría de la temática trabajada en la extracción de características de voz es nueva en el ambiente de pregrado, por lo cual se considera ilustrativo presentar un ejemplo donde se muestre el desarrollo de cada una de las etapas por las que se pasa para obtener los MFCC. A continuación se presentará el procesamiento realizado a una grabación de dos segundos con una frecuencia de muestreo de $11025Hz$ que contiene la pronunciación de la palabra uno.

Figura 13. Representación en el tiempo de la palabra uno.

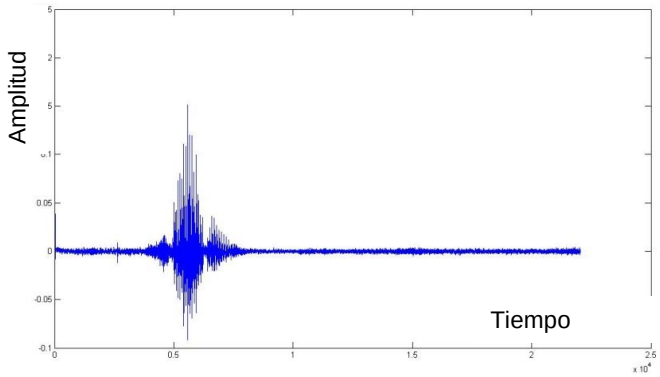


El preénfasis se realizó mediante la implementación de la función:

$$s_1(n) = s(n) - as(n-1) \quad (42)$$

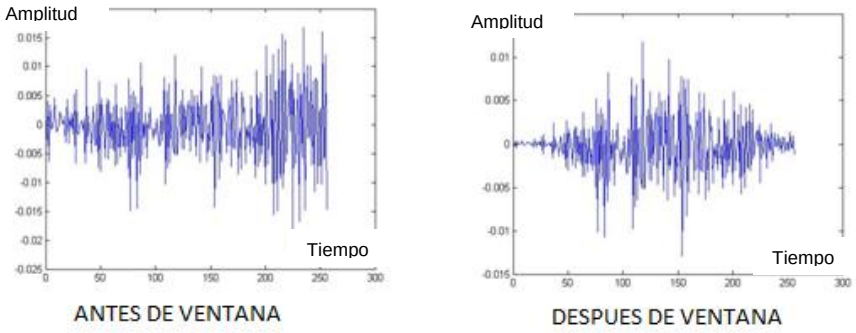
Donde $a = 0.95$, el objetivo del preénfasis como ya se explicó anteriormente es hacer la señal menos susceptible a malinterpretaciones en el posterior procesamiento. La figura 14 muestra la transformación que sufre la señal al pasar por la red de preénfasis.

Figura 14. Representación de la palabra uno con preénfasis.



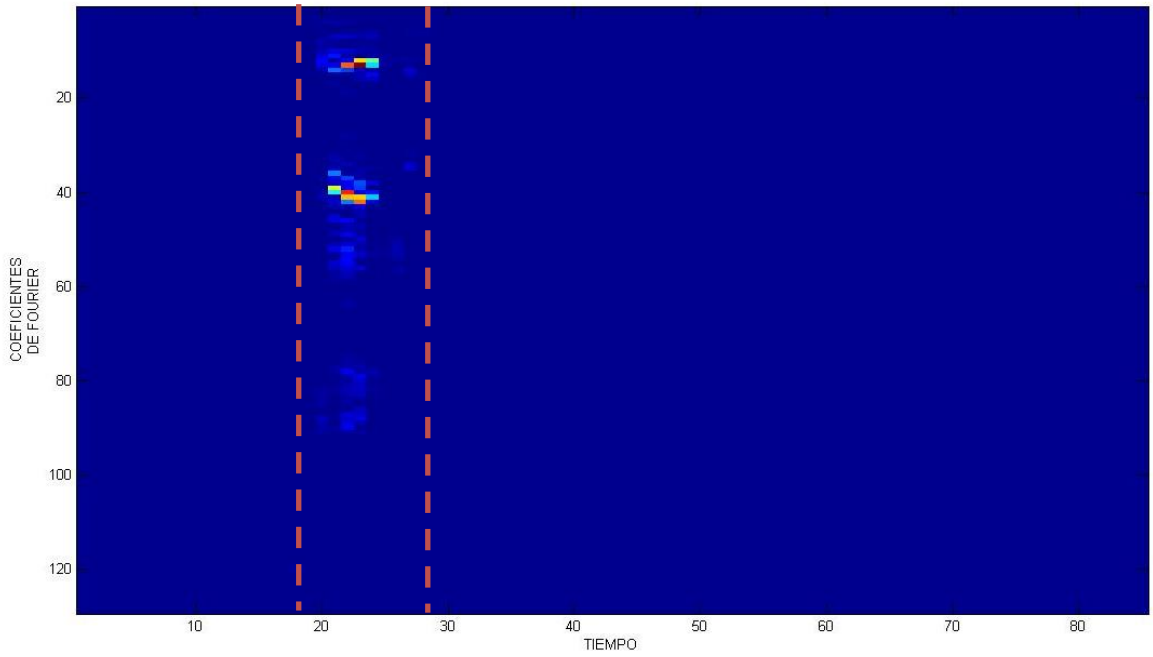
Para la división en bloques se decidió utilizar 256 muestras lo que corresponde a 21mS para formar un bloque. Los bloques de voz se traslaparon 32 muestras, factor a tener en cuenta ya que es una condición que causó inconveniente en la implementación en bajo nivel. Después se aplicó una ventana de Hamming a cada bloque individual.

Figura 15. Bloque de voz antes y despues de la ventana de Hamming.



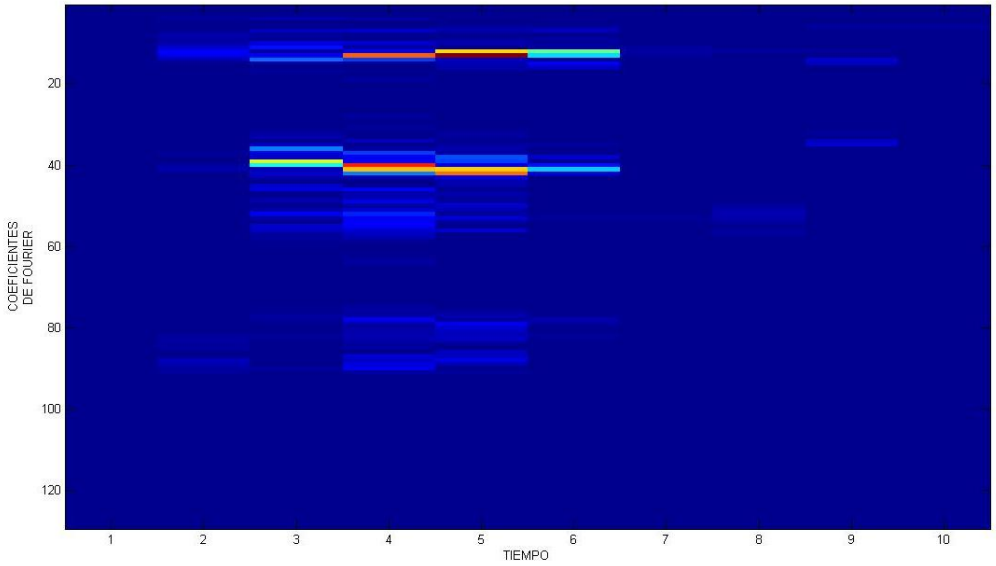
El siguiente paso dentro del procesamiento es la aplicación de un método de análisis espectral. Se escogió la transformada rápida de Fourier en lugar de la codificación de predicción lineal debido a que este algoritmo también se utiliza para la realización de la detección de voz y las limitaciones en cuanto a memoria de programa de los sistemas embebidos hacen necesario que en lo posible se reutilice código.

Figura 16. Representación espectral de los bloques de tiempo.



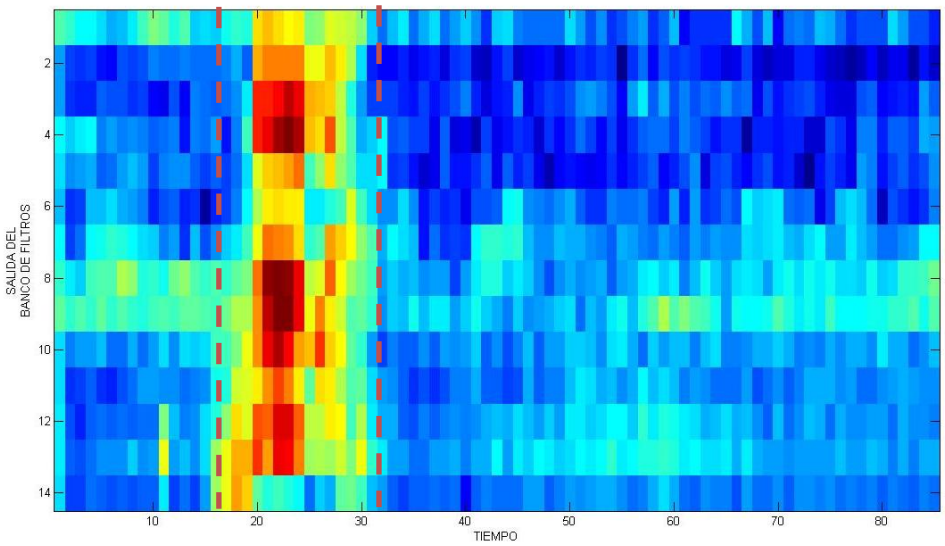
La figura 16 muestra el análisis espectral hecho a toda la palabra, en el eje x se ubica el tiempo, esto significa que cada una de las columnas equivale a un bloque de señal de voz. En el eje y se ubican los coeficientes de Fourier, esto quiere decir que cada fila corresponde al análisis de una banda de frecuencia para la grabación. Un color tendiente al rojo indica que el coeficiente de Fourier tiene un valor grande, y un valor tendiente al azul significa que el valor tiende a un valor cercano a cero, las líneas punteadas indican el segmento que contiene voz.

Figura 17. Representación espectral segmento hablado de la palabra uno.



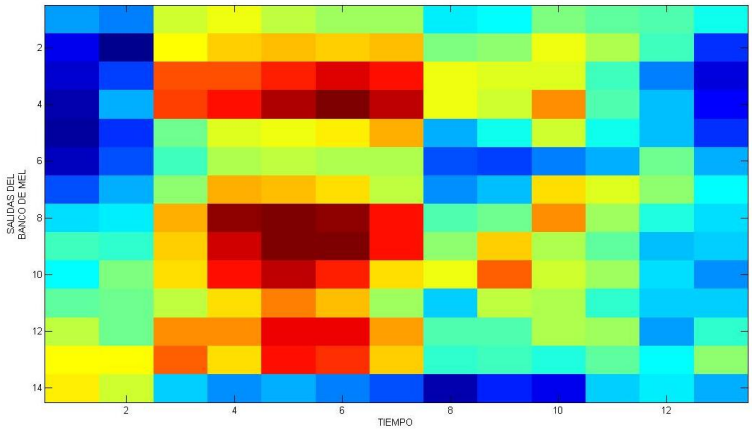
Al análisis de Fourier de la señal de voz se le debe aplicar el banco de filtros de Mel, para esta aplicación se utilizó un banco de 14 filtros distribuidos de manera logarítmica. En la figura 18, las líneas punteadas indican el segmento que contiene voz.

Figura 18. Salida banco de filtros de Mel.



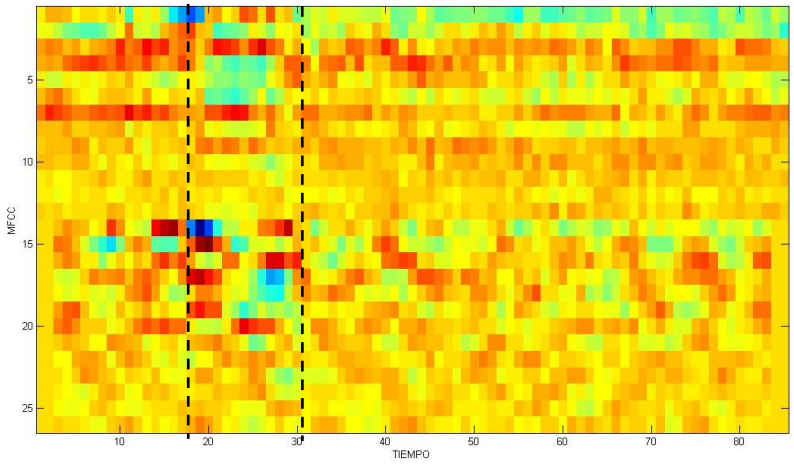
Al igual que en el análisis de Fourier un color rojo indica un valor grande, y un color azul indica un valor tendiente a cero. En el eje x se ubica el tiempo, es decir cada columna es un bloque de voz y en el eje y se ubican las salidas de cada filtro del banco de Mel, por lo que cada fila indica el análisis de una banda de frecuencias.

Figura 19. Salida banco de filtros segmento hablado de la palabra uno.



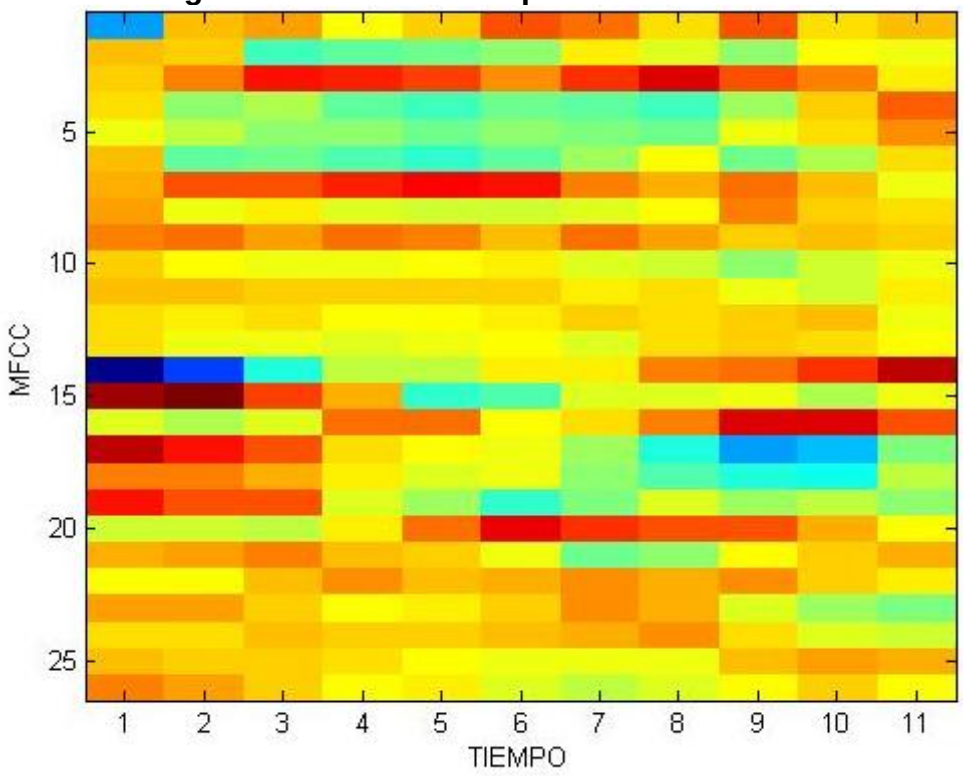
Finalmente, se debe regresar del espacio de frecuencias al espacio del tiempo, lo cual se logra mediante la transformada discreta del coseno; estos datos de salida representan los coeficientes cepstrales de frecuencia de Mel. Además se deben calcular los deltas de los MFCC para formar un conjunto de 26 datos que son quienes caracterizan la señal de voz.

Figura 20. Coeficientes cepstrales de frecuencia de Mel (MFCC).



Maneja las mismas convenciones que las gráficas anteriores, con la diferencia de que en el eje y se ubican los coeficientes cepstrales de frecuencia de Mel.

Figura 21. MFCC segmento hablado de la palabra uno.



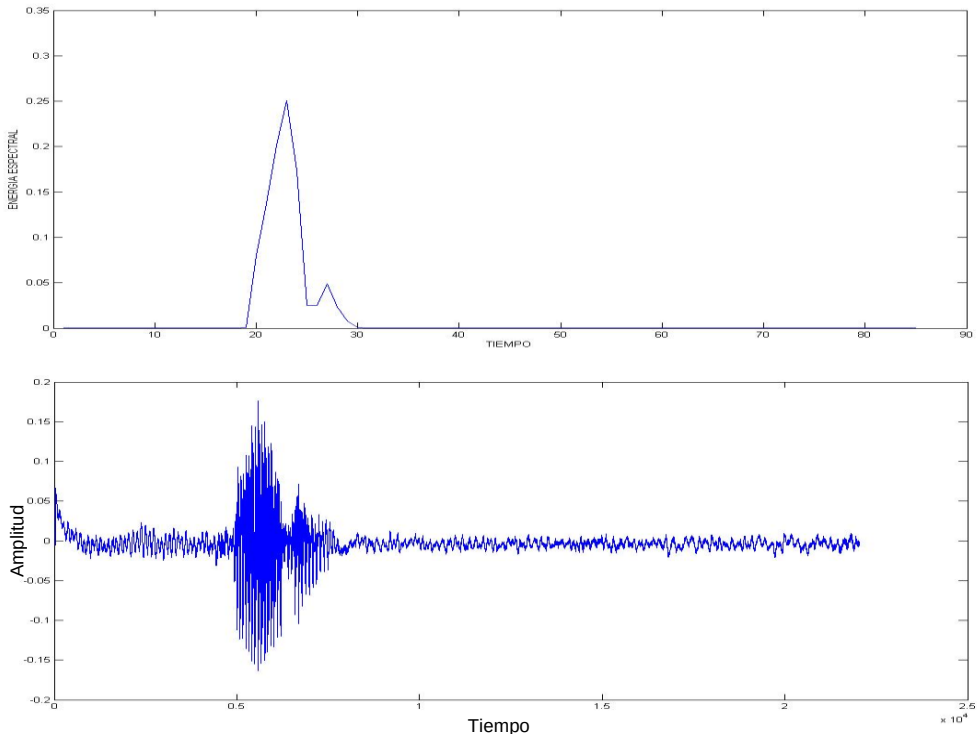
3.2 DETECCIÓN DE VOZ

Como se mencionó en capítulos anteriores el objetivo de la detección de voz es separar los eventos acústicos de interés, es decir, separar las palabras pronunciadas en una señal de audio continua que contiene voz, diferenciándola del ruido ambiental. La importancia de la detección de voz para un reconocedor de palabras aisladas radica en que el algoritmo de reconocimiento depende directamente de la precisión con que la detección de voz se realiza; además en un sistema embebido, no es posible procesar en tiempo real toda la información, por lo cual la detección de voz es la que permite identificar cuando es necesaria iniciar el algoritmo de reconocimiento.

Para el desarrollo del presente proyecto se utilizó el algoritmo de detección de voz basada en entropía espectral (…véase sección 1.8…), el cual permite garantizar algo de robustez ante ruidos sin comprometer en gran medida la complejidad algorítmica ni tampoco el tiempo de procesamiento necesario. Como se mencionó en la sección 1.8 la detección de voz está basada en determinar una función de energía espectral y hallar unos niveles de mantenimiento para determinar los

eventos acústicos de interés. La figura 22 muestra el procesamiento de una palabra y su gráfica de energía espectral correspondiente.

Figura 22. Entropía espectral para la palabra uno.



La gráfica de energía espectral muestra la cantidad de energía que tiene la palabra en cada bloque de voz. La figura 22 corresponde a la pronunciación de un "uno", la zona inicial sube rápidamente y muestra un pico debido a que la palabra comienza con una vocal y por eso tiene alta energía. La zona donde se encuentra la consonante tiene una energía más baja y por eso se produce la bajada de la energía espectral. El último pico de la gráfica corresponde a la pronunciación de la última vocal, pero debido a que la palabra está finalizando su pico es bajo.

3.3 ALGORITMOS DE RECONOCIMIENTO

Para completar el sistema de reconocimiento de voz hace falta implementar la segunda parte, que corresponde al algoritmo de reconocimiento. Es aquí donde se concentra la parte fundamental del desarrollo del trabajo, por esto, se consideró conveniente realizar un pequeño estudio comparativo entre algunos algoritmos, específicamente entre un sistema mediante la metodología estadística tradicional de modelos ocultos de Markov, un sistema hibrido de dos etapas conformadas por una red neuronal artificial y alineamiento de tiempo dinámico, y finalmente, un sistema hibrido conformado por una red neuronal y modelos ocultos de Markov;

con la finalidad de determinar cuál es el más eficaz en cuanto a porcentaje de reconocimiento.

Para tal fin, se decidió realizar un estudio sobre una de las aplicaciones tradicionales de prueba para sistemas de reconocimiento de voz, el reconocimiento de dígitos, es decir los números desde el cero hasta el nueve.

Para garantizar que las pruebas sobre los tres algoritmos de reconocimiento se realizan bajo condiciones similares se utilizó el mismo procedimiento de extracción de características (…véase sección 1.5…), se utilizó el mismo algoritmo de detección de voz y además se creó una pequeña base de datos con voces de 3 personas, dos hombres y una mujer. La base de datos contenía 250 grabaciones. Las primeras 150 grabaciones corresponden a 5 pronunciaciones de cada uno de los dígitos de cada uno de los hablantes, denominadas en el proyecto como V1, estas se utilizaron como base del entrenamiento de los algoritmos. Las siguientes 50 grabaciones son de hombres, denominadas en el proyecto como V2, y se utilizaron para probar los sistemas. Y las últimas 50 grabaciones son de una mujer, se denominan V3, y también se utilizaron para probar los sistemas.

3.4 ALGORITMO DE RECONOCIMIENTO MEDIANTE HMM

Los modelos ocultos de Markov (…véase sección 1.6…), son un conjunto de estados conectados por transiciones. Formalmente se componen de: el número de estados N, el número de posibles emisiones M, una matriz de probabilidades transmisión A, una matriz de probabilidades de emisión B y una matriz de distribución inicial de probabilidades π; que definen el comportamiento del algoritmo.

Los M símbolos de emisión son valores discretos, estos son los datos observables, sin embargo, la salida de la extracción de características son vectores y no valores discretos, por lo cual no es posible entrenar directamente los modelos ocultos de Markov teniendo como entrada los MFCC. Para lograr el entrenamiento de los modelos de Markov a partir de los MFCC se hace necesario incorporar un algoritmo intermedio que se encargue de agrupar los vectores en valores discretos de acuerdo a sus características. La solución a este problema son los algoritmos de agrupamiento, más conocidos como *clustering,* que consisten en la distribución de un conjunto de observaciones en subconjuntos, a los que se les denomina *clusters,* de tal manera que las observaciones en cada *cluster* sean similares en alguna manera representativa. Para el presenta trabajo se utilizó el algoritmo denominado *k-means* que pretende dividir un conjunto de n observaciones en k *clusters,* en donde cada observación corresponde con el *cluster* con el promedio más cercano.

El entrenamiento del modelo se realizó mediante el algoritmo denominado *forward backward algorithm*, que se encarga de tomar los valores de la salida del algoritmo

de *clustering* de las grabaciones de audio, y a través de estas modificar los valores de las matrices de transmisión y emisión para maximizar la probabilidad de que una muestra de voz corresponda al modelo. Para realizar un reconocedor de palabras aisladas, se debe crear un modelo para cada palabra. En nuestro caso se crearon diez modelos, cada uno correspondiente a un dígito. Para el entrenamiento se utilizó las grabaciones almacenadas en V1.

Se evaluó el algoritmo de reconocimiento con diferentes cantidades de *clusters,* variándolas entre 8 y 36, los resultados más significativos en cuanto a porcentaje de reconocimiento de palabras en los diferentes conjuntos de la base de datos se muestran en la tabla 1.

Tabla 1. Porcentajes de Reconocimiento por número de *clusters.*

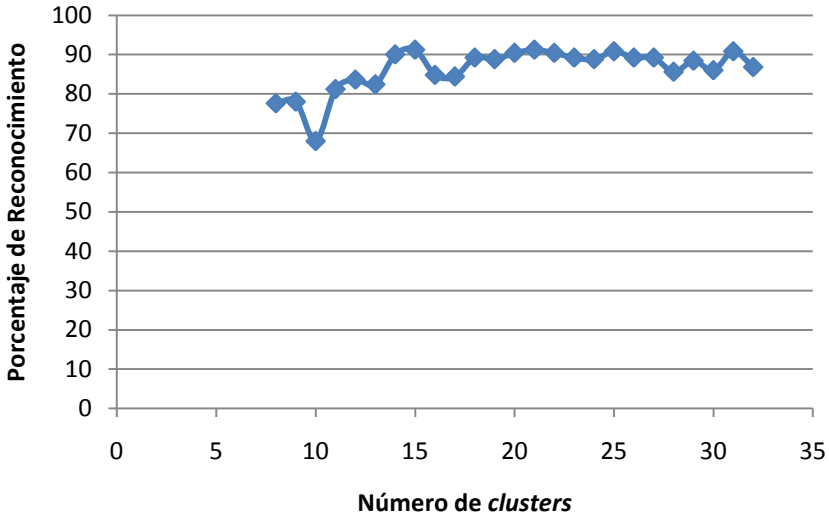
N	V1	V2	V3	Total
9	86,7	74,0	56,0	78,0
10	80,0	48,0	52,0	68,0
14	96,0	86,0	76,0	90,0
15	94,7	92,0	80,0	91,2
16	94,7	74,0	66,0	84,8
19	97,3	80,0	72,0	88,8
20	97,3	88,0	72,0	90,4
21	98,0	90,0	72,0	91,2
22	98,0	86,0	72,0	90,4
31	99,3	84,0	72,0	90,8
32	98,7	72,0	66,0	86,8
33	98,7	88,0	70,0	90,8

La columna marcada con N indica la cantidad de *clusters* con la que se entrenó el algoritmo. Las columnas marcadas con V1, V2, V3 y Total representan el porcentaje de reconocimiento de palabras sobre cada grupo de la base de datos.

En esta prueba se observó que el peor resultado se observó con $N = 10$ correspondiente a un 68% de reconocimiento. Por otro lado, los mejores resultados se obtuvieron con $N = 15$ y $N = 21$ con un 91.2% de reconocimiento. Es interesante mirar una de las tendencias que tienen los datos con respecto al número de *clusters*, a medida que se aumentan, el porcentaje de reconocimiento sobre los datos de entrenamiento aumenta considerablemente, pero la generalización del sistema sobre los datos de prueba disminuye.

Otro análisis de la tabla 1, muestra que la cantidad de *clusters* y el porcentaje de reconocimiento total no están relacionados de manera lineal como se pudiese pensar. La Figura 23, muestra esto de una manera más clara:

Figura 23. Porcentaje de reconocimiento por número de *clusters*.



3.5 ALGORITMO DE RECONOCIMIENTO MEDIANTE HIBRIDO ANN/HMM

Una de las ideas detrás del desarrollo del presente proyecto, es investigar el rendimiento que presentan las redes neuronales como algoritmo de reconocimiento para el ASR, lastimosamente, las redes neuronales sólo pueden ser usadas a través de clasificación estática (…véase sección 2.4…) para el reconocimiento de palabras; existen estudios que han demostrado que la clasificación estática presenta tasas de reconocimiento de palabras muy bajas en comparación con la clasificación dinámica. Por esto se decidió usar las ANN como clasificador de fonemas y realizar híbridos con algoritmos complementarios que ayuden a solucionar la característica dinámica que posee la voz y de esta manera lograr el reconocimiento de palabras.

La primera aproximación al reconocimiento de voz implementada con redes neuronales artificiales fue un híbrido con modelos ocultos de Markov. La idea detrás de esta aproximación, es crear una red neuronal que clasifique en fonemas, las salidas representadas en forma de coeficientes cepstrales de Mel de la etapa de extracción de características y estudiar la manera en la que las ANN afectan el rendimiento de los HMM como algoritmo de reconocimiento.

Los veintiséis MFCC que representan las características de la voz, son las entradas a la red neuronal artificial. La aplicación de reconocimiento de dígitos involucra trece fonemas diferentes para diferenciar e identificar cada uno de los

dígitos. Estos trece fonemas son las salidas que debe tener la red neuronal. La tabla 2 muestra los fonemas.

Tabla 2. Fonemas del reconocedor de dígitos

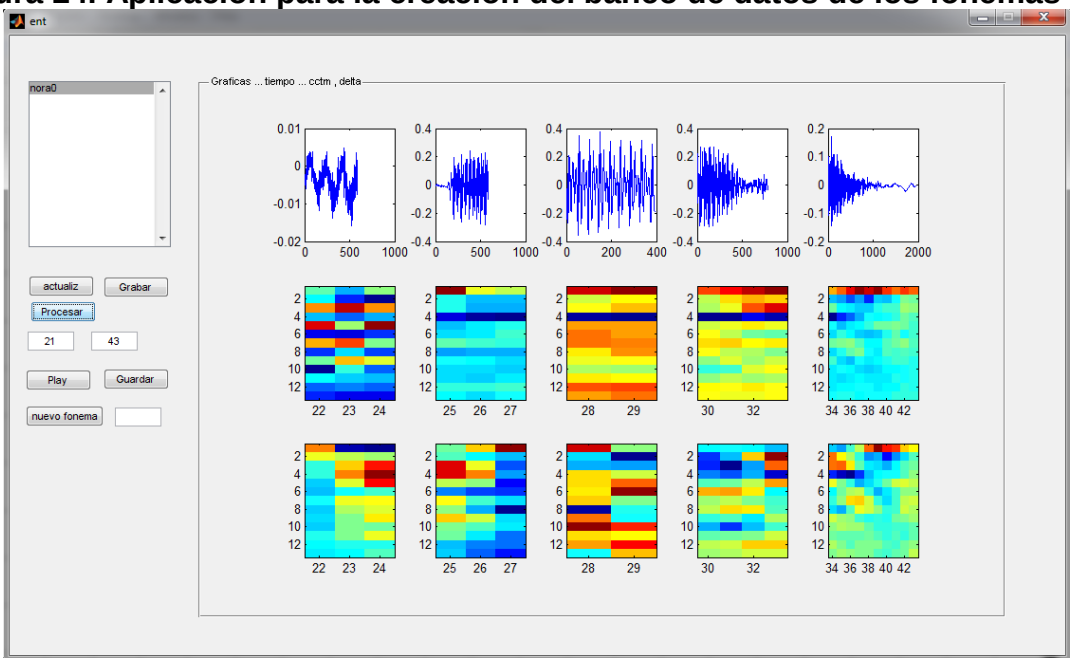
Nomenclatura	Sonido
1	a
2	k
3	ch
4	d
5	e
6	i
7	n
8	o
9	r
10	s
11	t
12	u
13	v

Para hacer que la red neuronal sea capaz de clasificar los MFCC en fonemas es necesario añadir una capa oculta que sirva como procesamiento de la información. De esta manera se estructuró una red neuronal con 26 entradas, conectadas a 35 neuronas en una capa oculta con función de activación sigmoidal y estas conectadas a 13 neuronas en la capa de salida también con función de activación sigmoidal.

El procedimiento de entrenamiento de una ANN implica la identificación de manera clara de que datos corresponden a las entradas y que datos corresponden a las salidas, esto no supone una dificultad cuando se tratan palabras completas, ya que la detección de voz ayuda a solucionar este problema de manera sencilla. Pero cuando se trabaja a nivel de fonemas, detectar que bloques del audio corresponden a cada uno de los fonemas representa una dificultad significativa. Existen algunos métodos experimentales que tratan de lidiar con este problema mediante modificaciones a algoritmos de detección de voz, entre ellos se cuenta el trabajo con las derivadas de la función de energía espectral, derivadas de la función de entropía espectral, entre otras; pero más allá de eso, el problema de trabajar a nivel de fonema, radica en que no existe una definición clara de lo que representa un fonema en cada lengua, sino que dependen de la percepción de la persona que realiza el estudio y de la población o el lugar en el cual se realiza, por lo tanto, las divisiones realizadas por un algoritmo, pueden ser poco adecuadas para el tratamiento de este tipo de señales. Por esto se decidió crear una pequeña

aplicación que facilite la creación de bancos de datos que contengan cada uno de los fonemas mencionados en la tabla 2. La aplicación muestra la representación de la palabra en forma de MFCC además, genera una división a nivel de fonemas basada en entropía espectral y también permite reproducir bloques específicos de la grabación, con la finalidad de ayudar a diferenciar cada uno de los fonemas, pero la decisión final de la asignación de cada bloque a un fonema específico la toma el usuario (en nuestro caso los investigadores).

Figura 24. Aplicación para la creación del banco de datos de los fonemas



Los bancos de datos están formados por los MFCC que corresponden a cada fonema. Estos bancos de datos fueron los utilizados para entrenar la red neuronal. La idea de los modelos ocultos de Markov en un híbrido con ANN es muy similar a la idea que hay detrás de la implementación de solo modelos ocultos de Markov, esto quiere decir que para cada palabra del vocabulario (cada uno de los dígitos) se debe crear un modelo oculto de Markov. El entrenamiento de los modelos ocultos de Markov se hace teniendo como entradas las salidas del algoritmo de *clustering* sobre las salidas de la red neuronal, se utilizó un procedimiento de entrenamiento igual al descrito en la sección anterior.

Al igual que en la aproximación anterior, se evaluó el algoritmo de reconocimiento con diferentes cantidades de *clusters,* variándolas entre 8 y 32, los resultados más significativos en cuanto a porcentaje de reconocimiento de palabras en los diferentes conjuntos de la base de datos se muestran en la tabla 3, la nomenclatura manejada es la misma de las otras secciones.

Tabla 3. Resultados del algoritmo ANN/HMM para diferentes cantidades de clusters

N	V1	V2	V3	Total
8	98,7	48	68	82,4
9	100	48	70	83,6
10	100	54	62	83,2
11	99,3	44	60	80,4
12	100	60	76	87,2
13	100	64	68	86,4
14	99,3	46	54	79,6
15	100	60	70	86,0
16	100	54	62	83,2
32	100	40	38	75,6

En esta prueba se observó que los resultados se mantuvieron relativamente constantes para diferentes cantidades de *clusters*. De cualquier manera, los mejores resultados se obtuvieron con $N = 12$ con un 87.2% de reconocimiento. Se puede observar en la tabla 3, que la inclusión de la red neuronal hace que el porcentaje de reconocimiento sobre los datos usados para el entrenamiento sea de valores muy cercanos o incluso iguales al 100%, pero la generalización del sistema se ve muy afectada ya que el porcentaje de reconocimiento sobre las otras dos partes de la base de datos tiene valores muy bajos.

3.6 ALGORITMO DE RECONOCIMIENTO MEDIANTE HIBRIDO ANN/DTW

La tercera aproximación realizada fue un híbrido con una red neuronal y el algoritmo de alineamiento del tiempo dinámico. Para la implementación de este algoritmo no fue necesario reentrenar la red neuronal, sino que se utilizó la misma red de la aproximación mediante hibrido ANN/HMM, ya que el objetivo de la red sigue siendo el de clasificación de características de voz en fonemas.

El DTW es un algoritmo que permite manejar el problema que se presenta con las diferentes duraciones en tiempo de las palabras pronunciadas en iteraciones distintas, en otras palabras, la cantidad de bloques de voz no es constante, incluso sobre la misma palabra. El algoritmo DTW garantiza encontrar el camino con mínima distancia sobre la matriz de distancias locales que compara los vectores que conforman la matriz de referencia con los vectores que conforman la matriz de entrada (…véase sección 1.7…). El algoritmo DTW requiere la creación de una matriz de referencia por cada palabra que se desee reconocer. La ventaja que presenta inicialmente el DTW frente a los modelos de Markov es que no necesita

un entrenamiento para funcionar; ya que las matrices de referencia son creadas teniendo en cuenta la salida ideal de la red neuronal.

Tabla 4. Matriz de referencia para el número uno

FONEMA	U	N	O
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	1	0
8	0	0	1
9	0	0	0
10	0	0	0
11	0	0	0
12	1	0	0
13	0	0	0

El sistema se evaluó sobre las 3 partes de la base de datos que almacena las voces. Los resultados de este sistema fueron extremadamente satisfactorios, la tabla 5, muestra los resultados obtenidos en cuanto porcentaje de reconocimiento.

Tabla 5. Porcentaje de reconocimiento del algoritmo ANN/DTW

V1	V2	V3	Total
100	100	98	99,6

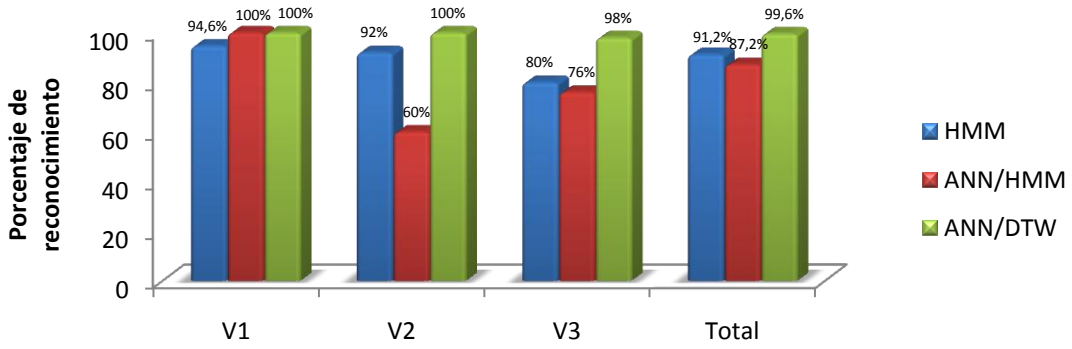
Aquí se observó que el hibrido ANN/DTW superó con creces a los otros algoritmos implementados, además la capacidad de generalización de la red neuronal se traslada con facilidad por el algoritmo de alineamiento del tiempo dinámico logrando altos porcentajes de reconocimiento incluso sobre las bases de datos que nunca formaron parte del entrenamiento.

3.7 COMPARACIÓN DE LOS TRES ALGORITMOS

Los tres algoritmos de reconocimiento presentados en este capítulo pueden ser usados en sistemas de reconocimiento de voz, el más utilizado en aplicaciones debido a su facilidad de entrenamiento y baja complejidad algorítmica es el de Modelos Ocultos de Markov. Muchas investigaciones se han centrado en explotar las características ventajosas que presentan los HMM dentro del campo del ASR. Por otro lado, a pesar de que existen investigaciones que conciernen a las ANN en

el ASR su alcance en el campo aplicativo todavía es limitado. Las ANN presentan unas características muy interesantes en el desarrollo de sistemas ASR por sus propiedades que ayudan a facilitar el manejo de inconvenientes que presentan las señales de audio, tal es el caso de la generalización que permite el manejo de un amplio rango de hablantes con el entrenamiento de solo unas cuantas personas (sistemas con múltiples hablantes), o el manejo de no linealidades que permite la clasificación de muestras difíciles en fonemas. Las ANN a pesar de que no pueden conformar un algoritmo completo de reconocimiento, permiten solucionar problemas complejos como la entonación, el timbre, el acento, la intensidad, la diferencia de frecuencia en la pronunciación, entre otras, lo que facilita la creación de sistemas más robustos para la elaboración de aplicaciones.

Figura 25. Comparación de los algoritmos implementados.



En la figura 25 se puede observar que el algoritmo implementado mediante híbrido ANN/HMM es el que presenta menor rendimiento en la aplicación. La razón para que esto ocurra no está directamente reflejada en el rendimiento como tal de los algoritmos. La verdadera razón para que esto ocurra es que los modelos ocultos de Markov manejan una serie de multiplicaciones de valores muy pequeños para la selección de la palabra pronunciada (…véase sección 1.6.4…). La red neuronal en esta aproximación actúa como un filtro de manera que los datos que entran al HMM se encuentran o muy cercanos a cero o muy cercanos a uno, por lo cual el algoritmo de *clustering* genera sus grupos muy diferenciados y esto a su vez hace que las matrices de emisión y transmisión del modelo de Markov contengan muchos valores cercanos a cero, lo que conduce a un problema de escalamiento en las probabilidades de cada una de las palabras. Cabe destacar que para la implementación de este algoritmo se utilizó el paquete de estadística de Matlab el cual trabaja con una precisión de 64 bits lo cual es superior a lo ofrecido por la mayoría de compiladores que trabajan en elementos embebidos, además la función usada para obtener los datos también maneja un algoritmo de escalamiento y los resultados demostraron que no fue suficiente para conseguir desempeño satisfactorio, por eso, se decidió no profundizar en la búsqueda de más algoritmos de escalamiento ya que la dificultad técnica a la hora de

implementarlos en un dispositivo que maneja registros de 16 bits como un microprocesador o un DSP es muy alta.

De los resultados se observó que los modelos ocultos de Markov como algoritmo de reconocimiento presentan buen comportamiento, además son fáciles de entrenar e implementar. Los inconvenientes que presentan se deben a las variaciones que se tienen en la voz, como la frecuencia, la intensidad (que puede llevar al ya mencionado problema de escalamiento), el timbre, entre otras. Estas presentan muy buenas características para ser implementadas en sistemas donde es admisible tener niveles de error de alrededor del 15% en cuanto al porcentaje de reconocimiento.

El hibrido ANN/DTW, por otro lado, demostró superar a los demás algoritmos en las pruebas sobre todas las partes de la base de datos. La complejidad algorítmica del DTW no es muy elevada, por lo cual es factible su implementación bajo un sistema embebido. Debido a su mayor precisión y robustez ante las señales de prueba se decidió usar este híbrido como algoritmo de reconocimiento.

4. IMPLEMENTACIÓN DEL SISTEMA DE RECONOCIMIENTO EN DSP

Hasta ahora, se ha presentado una descripción completa del funcionamiento de un sistema de reconocimiento de voz en lenguaje de alto nivel. En este capítulo se muestra la manera en que el sistema de reconocimiento de voz fue implementado en el eZDSP USB Stick VC5505 de Texas Instruments.

4.1 SELECCIÓN DEL ELEMENTO

Para la selección del dispositivo embebido se tuvieron en cuenta varias consideraciones, a saber: la necesidad de una memoria suficiente para almacenar las matrices que componen la ANN, las matrices de referencia para el DTW, los coeficientes involucrados en el pre procesamiento y los datos de voz durante un periodo de dos segundos, además la necesidad de una memoria de programa grande debido a la complejidad algorítmica, una velocidad de procesamiento superior a 50 MIPS y que el dispositivo cuente con una interfaz para simular o emular la aplicación en ejecución con la finalidad de depurar los algoritmos implementados. Otro factor influyente en la decisión de compra es el costo de la herramienta, ya que los recursos para la financiación del presente proyectos son asumidos completamente por los investigadores.

La herramienta seleccionada de las encontradas en el mercado fue el eZDSP USB Stick VC5505 fabricado por Texas Instruments, es una herramienta de pequeño tamaño y bajo costo, que puede ser usada como instrumento de desarrollo mediante conexión con el puerto USB, además incluye todo el hardware y software necesario para trabajar aplicaciones de 16 bits con bajo consumo de potencia. El puerto USB provee toda la potencia necesaria para manejar la herramienta de manera que en el proceso de desarrollo no hace falta una fuente de poder externa. La herramienta además tienen incrustado el emulador XDS510 para depurar el código de manera sencilla y soporta la conexión con el software Code Composer Studio V4.

El elemento incluye: Un DSP TMS320C5505 de punto fijo, un emulador integrado XDS100, una memoria EEPROM y un códec de audio de 32 bits programable TLV320AIC3204.

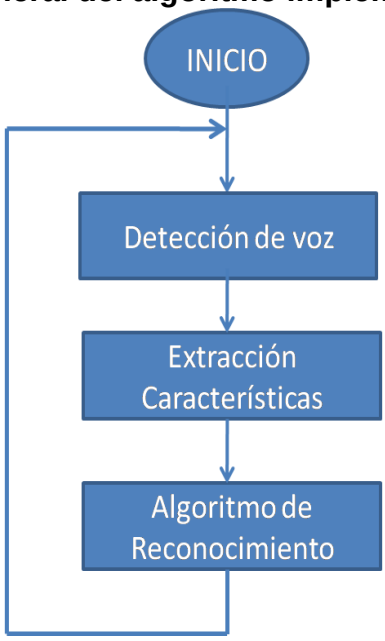
El TMS320VC5505 es uno de los procesadores de 16 bits de más bajo consumo usado en aplicaciones a nivel industrial, lo que le permite una duración de batería mucho más larga. Con 200MIPS de velocidad de procesamiento, 320KB de memoria integrada, hasta 500KB de código de programa el TMS320VC5505 provee una buena base para el desarrollo de aplicaciones de audio, procesamiento de señales de instrumentos musicales, soluciones médicas y de seguridad.

4.2 DESCRIPCIÓN GENERAL DEL ALGORITMO

La implementación del sistema en lenguaje de alto nivel es la base de la implementación del sistema en el dispositivo embebido. La gran diferencia entre los dos procedimientos radica en que el procesamiento de la información en alto nivel se hace sobre grabaciones, mientras que en el dispositivo embebido se busca lograr un funcionamiento en tiempo real, y por lo tanto, la adquisición y el procesamiento de los datos de audio se hacen de manera continua.

Como se mencionó en la sección anterior, el dispositivo usado es el eZDSP USB Stick VC5505 cuya velocidad de procesamiento es de 200MIPS. Debido a la complejidad algorítmica y la operatoria involucrada en el procesamiento de los datos adquiridos, no es posible realizar todo el reconocimiento de voz en tiempo real. Por lo cual, se decidió implementar la detección de voz (…véase sección 1.8…) en tiempo real y dejar para un procesamiento un poco más lento, todo lo concerniente al reconocimiento, es decir, la extracción de características (…véase sección 1.5…) y el algoritmo de reconocimiento (…véase secciones 1.7 y capítulo 2…).

Figura 26. Descripción general del algoritmo implementado.



Para la codificación de los algoritmos en el DSP, se utilizó el software Code Composer Studio V4 que acompaña al dispositivo, este facilitó mucho la programación, ya que funciona como compilador de lenguaje C para el elemento, por lo cual, solo fue necesario realizar algunas modificaciones a los algoritmos implementados en alto nivel. Como se describió en los capítulos 1 y 2 el procesamiento de los datos es completamente matemático e involucra operaciones de punto flotante, además requiere una buena precisión. El DSP utilizado es de punto fijo y cuenta con una unidad lógica de procesamiento de 16

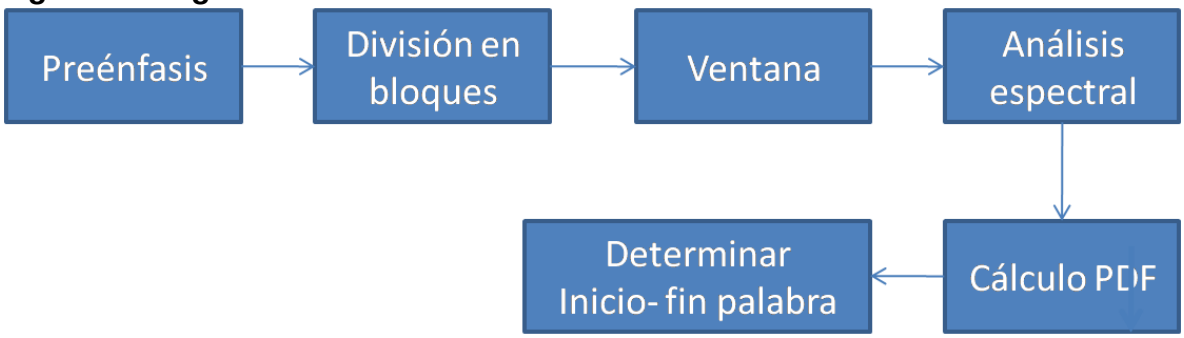
bits, lo que hace que no es óptimo para trabajar con los algoritmos. Code Composer Studio maneja una librería con operaciones matemáticas que están optimizadas para aprovechar el hardware de cada dispositivo, la precisión máxima que ofrecen es de 40 bits, lo cual hace que la velocidad de procesamiento sea lenta ya que la ALU trabaja a solo 16 bits, además esto representa una disminución en la precisión en comparación con el sistema en alto nivel que se realizó con procesamiento a 64 bits.

El eZDSP USB Stick VC5505 tiene con un códec de audio TLV320AIC3204 que cuenta con un ADC estéreo con velocidad de muestreo de hasta 48 Ksps (kilo muestras por segundo), con entradas y salidas programables. Además, cuenta con bloques integrados dentro de su hardware que permiten la implementación de algunos elementos de procesamiento de señales como amplificadores, filtros digitales y efectos de sonido. El códec permite el control del nivel de *bias* (cantidad de nivel de voltaje directo que lleva la señal) de la entrada de micrófono, así como su pre amplificación. Los filtros pueden eliminar el ruido audible introducido por el acople mecánico o por ruido en los elementos. Todas estas funcionalidades del elemento se utilizaron para reducir la cantidad de procesamiento en el DSP; pero a pesar de esto, es necesario añadir un micro controlador externo para construir la interfaz que permita la conexión a los dispositivos electrónicos a manejar.

4.3 ALGORITMO DE DETECCIÓN DE VOZ

El primer paso dentro del reconocimiento de palabras aisladas es separar los eventos acústicos que representan cada una de las palabras pronunciadas, para esto se utiliza un algoritmo de detección de voz.

Figura 27. Algoritmo de detección de voz



La primera acción realizada por este bloque es la adquisición de los datos de audio que entran por el micrófono, esta se hace aprovechando la conexión interna existente entre el DSP y el códec de audio, mediante protocolo I2S, los datos se almacenan en un vector que es usado para el procesamiento. El bloque se encarga de realizar el pre énfasis, la aplicación de la ventana de haming y el análisis espectral sobre bloques de 256 datos. Finalmente, se utilizó el algoritmo

de entropía espectral para detectar la voz, los límites de detección de inicio y fin de palabra se definieron de manera experimental.

Cuando se encuentra un inicio de palabra se empieza a grabar los coeficientes de Fourier en un vector para su posterior procesamiento y al encontrar el fin de palabra se genera una llamada para la ejecución de la extracción de características y del algoritmo de reconocimiento.

El preénfasis se realiza para "pulir" espectralmente la señal y hacerla menos susceptible a efectos de precisión en el posterior procesamiento de la señal. La red de preénfasis usada en el DSP, es:

$$H(z) = 1 - 0.95z^{-1} \quad (43)$$

La cual se puede describir mediante la ecuación:

$$s_1(n) = s(n) - 0.95s(n-1) \quad (44)$$

Esto en el DSP implica que por cada dato capturado es necesario realizarle la resta del dato anterior, de acuerdo con la ecuación 44. El códec de audio envía bloques de 256 datos cada $21.3mS$, el preénfasis se realiza a través de un ciclo donde para $n = 1$ se tiene en cuenta el último dato del bloque inmediatamente anterior, por lo cual es necesario que el dato $n = 256$ se guarde en cada bloque de procesamiento.

La aplicación de la ventana de Hamming se hace con la finalidad de minimizar las discontinuidades de la señal al comienzo y fin de cada bloque, su forma es:

$$w(n) = 0.54 - 0.46\,cos\left(\frac{2\pi n}{255}\right), \quad 0 \leq n \leq 255 \quad (45)$$

El valor $w(n)$ se debe multiplicar por $s_1(n)$. La manera óptima para implementar el procedimiento en el DSP, fue crear una tabla con los valores correspondientes a $w(n)$ y estos multiplicarlos por $s_1(n)$ mediante un ciclo.

$$s_2(n) = s_1(n)w(n), \;\; 0 \leq n \leq 255 \quad (46)$$

El siguiente paso dentro de la detección de voz es realizar el análisis espectral, para lo cual se utilizó la trasformada rápida de Fourier. El algoritmo utilizado para el cálculo de FFT fue creado por Jens Jorsen Nielsen, quien distribuye de manera libre, una función en lenguaje de programación C para el cálculo de una transformada de un valor N arbitrario de puntos[*]. La entrada de la transformada es considerada como una secuencia de valores complejos, al igual que la secuencia de salida. Cada secuencia de valores complejos se maneja con dos vectores, el primero que contiene la parte real y el segundo contiene la parte imaginaria de la señal.

Debido a la longitud del código de la función y a la necesidad de una mejoría sobre la velocidad de procesamiento de los datos fue necesario realizar una disminución del tamaño del código de la función y aprovechar el hecho de que la secuencia de entrada es de tipo real para disminuir el tiempo de procesamiento necesario para el cálculo de la FFT.

Por definición la transformada discreta de Fourier, se calcula como:

$$y_j = \sum_{k=0}^{N-1} e^{\frac{-2\pi i}{N} jk} x_k = \sum_{k=0}^{N-1} w_N^{jk} x_k \quad (47)$$

Donde x_k es la secuencia de valores complejos de entrada, y_j es la secuencia de valores complejos de salida y $w_N^{jk} = e^{\frac{-2\pi i}{N} jk}$. De acuerdo con Chris Lomont[18] y la teoría de análisis de Fourier, si x_k es una serie de valores reales entonces se cumple que:

$$y_j^* = \sum_{k=0}^{N-1} w_N^{-jk} x_k^* = \sum_{k=0}^{N-1} w_N^{(N-j)k} x_k = y_{N-j} \quad (48)$$

Como se mira en la ecuación la salida tiene simetría conjugada, por lo cual se dice que solo es necesario calcular la mitad de la salida para describir la salida compleja. Si se toma la secuencia real $x_0, x_1, x_2, \dots, x_{N-1}$de longitud $N = 2^n$ y se trata como una secuencia de valores complejos $z_k = x_{2k} + ix_{2k+1}$ de longitud $N/2$,

[*] Una copia del algoritmo se puede encontrar en http://www.corix.dk/html/fft.html

[18] LOMONT, Chris. The Fast Fourier Transform. Chris Lomont organization [online]. Enero de 2010. Available from www.lomont.org

donde $k = 0,1,2, \dots N/2 - 1$ entonces es posible calcular la FFT de z_k que tiene la mitad de la longitud de la secuencia original.

Si se llama t_j a la transformada de Fourier de z_k, entonces:

$$t_j = \sum_{k=0}^{\frac{N}{2}-1} w_{\frac{N}{2}}^{jk} (x_{2k} + i x_{2k+1}) = \sum_{k=0}^{\frac{N}{2}-1} w_{\frac{N}{2}}^{jk} x_{2k} + i \sum_{k=0}^{\frac{N}{2}-1} w_{\frac{N}{2}}^{jk} x_{2k+1} \quad (49)$$

Y tomando el aporte de Chris Lomont se tiene que:

$$y_j = \sum_{k=0}^{\frac{N}{2}-1} w_{\frac{N}{2}}^{jk} x_{2k} + w_N^j \sum_{k=0}^{\frac{N}{2}-1} w_{\frac{N}{2}}^{jk} x_{2k+1} \quad (50)$$

Y resolviendo y_j en términos de t_j, se obtiene que:

$$y_j = \frac{1}{2}\left(t_j + t^*_{\frac{N}{2}-j}\right) - \frac{i}{2}\left(t_j - t^*_{\frac{N}{2}-j}\right) w_N^j, \qquad 0 < j < \frac{N}{2} \quad (51)$$

$$y_0 = \frac{1}{2}(t_0 + t_0^*) - \frac{i}{2}(t_0 - t_0^*) \quad (52)$$

$$y_{N/2} = \frac{1}{2}(t_0 + t_0^*) + \frac{i}{2}(t_0 - t_0^*) \quad (53)$$

Las ecuaciones 51,52 y 53 permiten calcular una FFT de longitud N a través del cálculo de una FFT de longitud $N/2$, siempre y cuando se cuente con un procesador que pueda realizar operaciones complejas. En este caso, el DSP solo cuenta con la posibilidad de realizar operaciones de tipo real por lo cual fue necesario expandir aún más estas ecuaciones tomando en cuenta por separado la secuencia que contiene la parte imaginaria y la parte real de t_j.

Si se define $t_r(j)$ como la secuencia que contiene los valores reales de t_j, $t_c(j)$ como la secuencia que contiene los valores imaginarios de t_j, $y_r(j)$ como la secuencia de valores de reales de y_j y $y_c(j)$ como la secuencia que contiene los

valores imaginarios de y_j, y despejando a partir de las ecuaciones 51, 52 y 53 se tiene:

$$y_r(j) = \frac{1}{2}\left(t_r(j) + t_r\left(\frac{N}{2} - j\right) + A(j)\left(t_c(j) + t_c\left(\frac{N}{2} - j\right)\right) + B(j)\left(t_r(j) - t_r\left(\frac{N}{2} - j\right)\right)\right), \quad 0 < j < \frac{N}{2} \quad (54)$$

$$y_c(j) = \frac{1}{2}\left(t_c(j) - t_c\left(\frac{N}{2} - j\right) - A(j)\left(t_r(j) - t_r\left(\frac{N}{2} - j\right)\right) + B(j)\left(t_c(j) + t_c\left(\frac{N}{2} - j\right)\right)\right), \quad 0 < j < \frac{N}{2} \quad (55)$$

$$y_r(0) = t_r(0) + t_c(0), \quad y_c(0) = 0 \quad (56)$$

$$y_r(N/2) = t_r(0) - t_c(0), \quad y_c(N/2) = 0 \quad (57)$$

Donde:

$$A(j) = cos\left(\frac{2\pi j}{N}\right), B(j) = -sen\left(\frac{2\pi j}{N}\right) \quad (58)$$

Por lo tanto, para calcular una transformada de Fourier de 256 puntos que es la que se necesita en el algoritmo basta con calcular un FFT de 128 puntos y aplicar las ecuaciones 54-57. El algoritmo implementado en el DSP consta de varios pasos:

a. Descomponer la señal de entrada $s_2(n)$ en dos señales $f_e(n) = s_2(2n)$ y $f_o(n) = s_2(2n+1)$.

b. Calcular la FFT de 128 puntos complejos mediante el algoritmo de Jens Jorsen Nielsen, donde $f_e(n)$ corresponde a la parte real de la señal y $f_o(n)$ corresponde a la parte imaginaria de la señal.

c. Calcular los valores de $y_r(j)$ y $y_c(j)$ mediante las ecuaciones 54-57.

Los valores de $A(j)$ y $B(j)$ se pueden almacenar en vectores ya que contienen valores fijos, para un número de puntos de la transformada previamente determinados. Y para calcular $y_r(j)$ y $y_c(j)$ solo es necesario hacer un ciclo que se encargue de operar de acuerdo con las ecuaciones pronunciadas.

Para realizar el proceso final de detección de voz es necesario calcular la función de densidad de probabilidad mediante la ecuación:

$$p_i = \frac{s(f_i)}{\sum_{k=0}^{127} s(f_k)}, \ 0 \leq i \leq 127 \quad (59)$$

Donde $s(f_i)$ es el valor de la energía espectral de la componente de frecuencia espectral f_i y p_i es la densidad de probabilidad correspondiente a esa frecuencia. La energía espectral se puede calcular como:

$$s(f_i) = y_r(i)^2 + y_c(i)^2 \quad (60)$$

La entropía espectral de la señal se obtiene mediante:

$$H = -\sum_{k=0}^{127} w_k p_k log(p_k) \quad (61)$$

Por lo tanto, para calcular el valor de la entropía espectral de un bloque de voz es necesario calcular el valor de la energía espectral de cada coeficiente. El denominador de la ecuación 59 se calcula una sola vez mediante un ciclo que realice la sumatoria de todos los valores de energía. El valor de la entropía espectral se calcula a través de un ciclo que realice la sumatoria de la multiplicación de los valores de los pesos y las densidades de probabilidad. Para tal fin es necesario tener un vector que almacene los valores de los pesos. Los límites de detección de inicio y fin de palabra son determinados de manera experimental, a través del ingreso de voces de prueba al sistema para medir la cantidad de entropía que generan.

4.4 ALGORITMO DE EXTRACCIÓN DE CARACTERÍSTICAS

Para la extracción de las características de la voz, al igual que en el sistema de alto nivel, se usaron los coeficientes cepstrales de frecuencia de Mel, porque su funcionamiento está basado en el comportamiento del oído humano a diferentes bandas de frecuencias y ya que han demostrado presentar mayor robustez ante condiciones no deseadas.

La extracción de características se hace de acuerdo al procedimiento descrito en la sección 1.5. Se debe tener en cuenta que en el algoritmo de detección de voz,

se realiza el análisis espectral, por eso, en busca de agilizar el proceso de reconocimiento, en la etapa de extracción de características solo se realizan: el banco de filtros de Mel y la conversión Mel Cepstral, que permiten obtener los coeficientes cepstrales de mel como características de la señal.

El banco de filtros de Mel, consiste en un conjunto de filtros espaciados logarítmicamente en frecuencia y se expande hasta los $4kHz$. El espectro de la escala de Mel se puede calcular mediante:

$$S_f(l) = \sum_{k=0}^{127} S(k)M_l(k)\,, \qquad 0 \leq l < 19 \quad (62)$$

Donde $S(k)$ es la energía espectral encontrada en la detección de voz, $M_l(k)$ el l-ésimo filtro del banco de filtros y $S_f(l)$ el l-ésimo coeficiente del análisis espectral en frecuencia de Mel. Para el cálculo de estos coeficientes en el DSP solo es necesario tener almacenados los valores de los 20 bancos de filtros y realizar un producto punto entre el vector de energía espectral y cada uno de los vectores del filtro. La velocidad del procedimiento se ve beneficiada ya que en cada banco de filtros existen muchos componentes que tienen valor cero.

Finalmente, se realiza la conversión Mel Cepstral con la finalidad de regresar del espacio logarítmico de frecuencias al espacio del tiempo. Para tal proceso se utiliza la transformada discreta del coseno:

$$c(i) = \sqrt{\frac{2}{L}} \sum_{m=0}^{19} log\big(S_f(m)\big) cos\left(\frac{\pi i}{L}(m + 0.5)\right), \qquad 0 \leq i < 14 \quad (63)$$

La implementación de esto en el DSP se hace calculando el logaritmo de cada uno de los coeficientes del análisis en frecuencia de Mel, almacenando los valores del coseno en una tabla y realizando un ciclo que se encargue de realizar las multiplicaciones correspondientes y vaya acumulando esos resultados.

4.5 ALGORITMO DE RECONOCIMIENTO

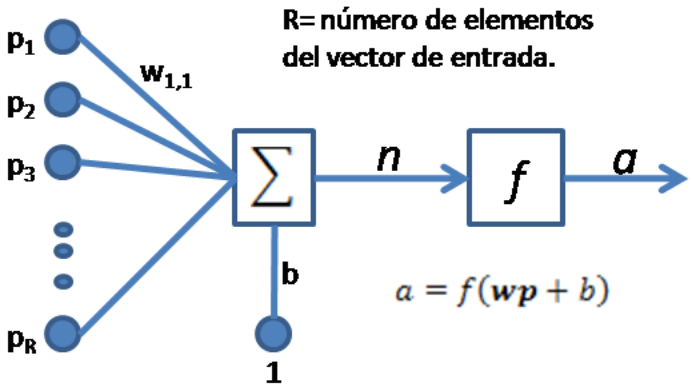
La parte final del reconocimiento de voz se implementó mediante el híbrido ANN/DTW, que como se mencionó en el capítulo 3, fue el que presentó mayor porcentaje de reconocimiento en las pruebas experimentales realizadas en el sistema en alto nivel. La red neuronal fue utilizada para el reconocimiento de fonemas y el DTW para manejar el problema de las duraciones diferentes de las

pronunciaciones de las palabras en iteraciones diferentes, incluso sobre la misma palabra.

Una red neuronal artificial (…véase capítulo 2…) consta de un conjunto de unidades de procesamiento, un conjunto de conexiones, un procedimiento de cómputo y un procedimiento de entrenamiento. La implementación de la red neuronal artificial se centra principalmente en el procedimiento de cómputo. Cabe decir que para nuestro caso se utilizó una red con 30 neuronas en la capa oculta.

El cómputo comienza presentando un patrón de entrada a la red neuronal artificial, específicamente los coeficientes cepstrales de frecuencia de Mel, que se propagan a través de las conexiones que tienen con las neuronas de la capa oculta, estas se encargan de aplicar una función de activación y obtener nuevos valores que a su vez se van a propagar a través de otra capa de neuronas encargadas de generar las salidas representando cada fonema.

Figura 28. Operaciones involucradas en una neurona



La figura 28 es una manera diferente de mirar el procedimiento realizado por una neurona. El patrón de entrada $p_1, p_2, p_3, \dots, p_R$ se puede representar de manera vectorial como $\boldsymbol{p}$. Las conexiones entre el patrón de entrada y la neurona $w_{1,1}, w_{1,2}, w_{1,3}, \dots, w_{1,R}$ como un vector $\boldsymbol{w}$. De esta manera, se calcula el valor n como:

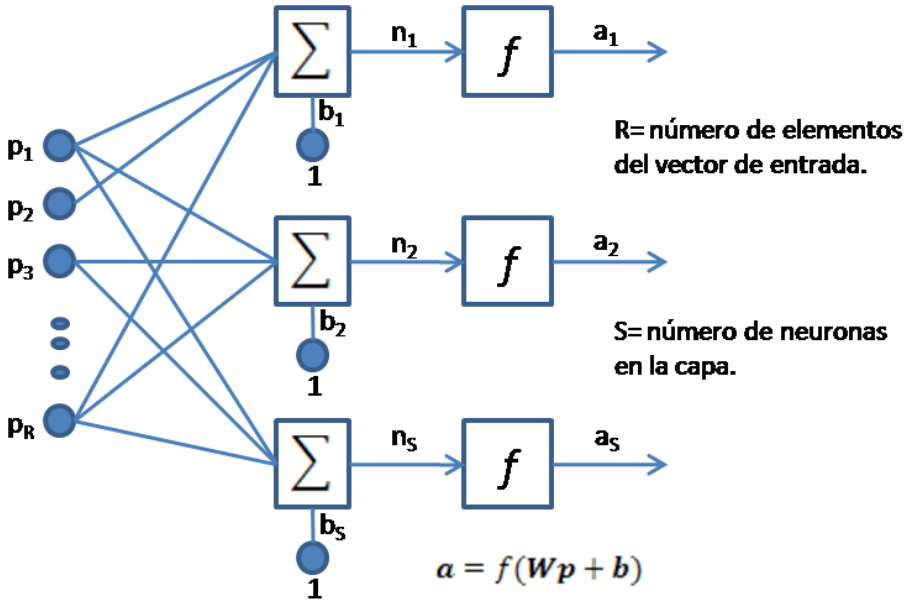
$$n = \boldsymbol{wp} + b \quad (64)$$

Donde b es el valor del *bias* de la neurona. Y el valor definitivo de salida de la neurona, por lo tanto, se puede representar, como:

$$a = f(\boldsymbol{wp} + b) \quad (65)$$

Donde f representa la función sigmoidal de activación de la neurona, para nuestro caso una función del tipo tangente sigmoidal. Ahora, si se mira el conjunto de neuronas que conforman una capa, se tiene lo mostrado en la figura 29:

Figura 29. Operaciones involucradas en una capa de neuronas



Es posible representar el patrón de entrada $p_1, p_2, p_3, \dots, p_R$ como un vector $\boldsymbol{p}$ de longitud R. Y las conexiones se pueden representar como una matriz $\boldsymbol{W}$.

$$\boldsymbol{W} = \begin{vmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{vmatrix} \quad (66)$$

Donde todos los pesos $w_{i,j}$ representan la conexión existente entre el patrón de entrada p_j y la neurona i. De esta manera, el vector $\boldsymbol{n}$ de longitud S se puede calcular como el producto matricial:

$$\boldsymbol{n} = \boldsymbol{W}\boldsymbol{p} + \boldsymbol{b} \quad (67)$$

Donde $\boldsymbol{b}$, es el vector que contiene los *bias* que se conectan a cada una de las neuronas de la capa. Por lo tanto, el vector de salida $\boldsymbol{a}$ se puede calcular como el producto matricial:

$$\boldsymbol{a} = f(\boldsymbol{W}\boldsymbol{p} + \boldsymbol{b}) \quad (68)$$

Donde f es la función de activación de cada una de las neuronas que conforman la capa, para este caso se utilizó una función tangente sigmoidal, que se describe mediante la ecuación:

$$y = \frac{2}{1 + e^{-x}} - 1 \quad (69)$$

Por lo tanto, para la implementación de la red neuronal en el DSP solo es necesario construir un procedimiento que permita la multiplicación de la matriz de pesos $\boldsymbol{W}$ por el vector de características $\boldsymbol{p}$, a eso adicionarle el vector de *bias* $\boldsymbol{b}$ y al vector de resultados aplicarle la función de activación tangente sigmoidal. Si se añaden más capas de procesamiento lo que se hace es repetir el proceso, teniendo en cuenta que los valores que hacen las veces de patrón de entrada son los de la salida de la capa inmediatamente anterior. Como se manifestó en el capítulo 2, el conocimiento de la red neuronal depende directamente de los pesos de sus conexiones, por lo cual se ha reservado la sección 5.1 para explicar del procedimiento de entrenamiento utilizado.

Después de que el vector de características ha pasado por la red neuronal artificial, se tiene un vector de las salidas de la red neuronal que representa los fonemas reconocidos. A este vector es necesario aplicarle el algoritmo DTW para manejar el problema de modelado temporal de las palabras y para reconocer cada una de las palabras (…véase sección 1.7…).

El algoritmo DTW consiste básicamente en realizar una comparación entre los vectores que representan los fonemas de una palabra pronunciada (agrupados en una matriz), con las matrices de referencia que se tienen para representar cada una de las palabras en el vocabulario del sistema. El procedimiento de realizar una comparación consiste inicialmente en construir una matriz de distancias locales, comparando cada vector de entrada con cada vector de la matriz de referencia a través de la medida de la distancia euclidiana entre ellos. Para ilustrar esta idea se puede considerar la figura 30.

Figura 30. Matriz de distancias locales

z	1	1.5	1.2	0.6
o	0.9	0.4	0.3	1.1
v	0.2	1.7	1.1	1.8
	v	o	o	z

La figura 30 muestra una relación tiempo-tiempo de las matrices donde en el eje vertical se ubica la matriz de referencia y en el eje horizontal la matriz de entrada. Cada campo de la matriz contiene la distancia local de la comparación de los dos vectores.

El siguiente paso es aplicar el algoritmo para determinar la distancia global a partir de las distancias locales. Como se describió en sección 1.7, si $D(i,j)$ es la distancia global hasta (i,j) y la distancia local en (i,j) está dada por $d(i,j)$ entonces:

$$D(i,j) = min[D(i-1,j-1), D(i-1,j), D(i,j-1)] + d(i,j) \quad (70)$$

Se asume que $D(1,1) = d(1,1)$ es una condición inicial del problema. Y además se define que si existen N vectores de entrada y la matriz de referencia está compuesta por n vectores de características, entonces la distancia total se encuentra dada por $D(n,N)$. Para llegar a calcular este valor es necesario recorrer la matriz desde la parte inferior izquierda hacia la derecha de tal manera que se puedan ir calculando los valores $D(1,2), D(1,3), \dots, D(1,N)$, a través de la búsqueda del mínimo de los valores que se encuentran en las posiciones especificadas en la ecuación. Con estos valores es posible calcular los valores de la siguiente fila, es decir $D(2,1), D(2,2), \dots, D(2,N)$; de esta manera se puede continuar hasta llegar a obtener los valores de la última fila, es decir $D(n,1), D(n,2), \dots, D(n,N)$. En la figura 31, se puede apreciar los valores obtenidos al realizar el proceso a la matriz

utilizada en el ejemplo, los valores en verde representan el camino con la menor distancia y el valor en azul es el valor de la distancia global.

Figura 31. Matriz de distancias globales

z	2.1	2.1	1.8	1.5
o	1.1	0.6	0.9	2.0
v	0.2	1.9	3.0	4.8
	v	o	o	z

Por lo tanto, para implementar el algoritmo DTW en el DSP, solo es necesario construir la matriz de distancias locales a través de las medidas de la distancia euclideana entre los vectores de entrada y la matriz de referencia. Luego recorrer esta matriz de distancias locales, de izquierda a derecha y de abajo hacia arriba apoyados de la búsqueda de un mínimo entre tres valores para construir la matriz de distancias globales. Al finalizar, el valor obtenido en la posición $D(n,N)$ se asigna a una posición de un vector que contiene la distancia global definitiva para cada palabra. El procedimiento se debe repetir por cada palabra del vocabulario del sistema. Finalmente, la palabra reconocida por el sistema se obtiene mediante la búsqueda del mínimo de los valores del vector que contiene las distancias globales definitivas.

4.6 ENTRENAMIENTO DE LAS REDES NEURONALES

En la sección anterior se describió lo necesario para implementar los algoritmos en el DSP, pero como se ha mencionado el conocimiento de la red neuronal y por lo tanto, el funcionamiento del sistema, depende directamente de los valores que tienen las conexiones de la red, por lo tanto, la parte fundamental del proceso de implementación esta etapa de la construcción de la ANN.

Para entrenar las redes neuronales es necesario definir los comandos a reconocer dentro de las aplicaciones, por eso se inició escogiendo las aplicaciones que se manejan con el sistema. Se decidió construir tres aplicaciones simples: la primera es el reconocimiento de dígitos, escogida por ser una de las aplicaciones

tradicionales en el ASR y que por lo tanto puede servir para comparar el sistema diseñado con sistemas existentes. La segunda aplicación fue el control de un carro de juguete con movimientos básicos y la tercera aplicación fue la elaboración del control remoto para un televisor sony.

Las palabras que se pronunciaron para entrenar la primera aplicación, son: cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho y nueve. Para la segunda aplicación: adelante, atrás, izquierda, derecha, pare. Y para la tercera aplicación: arriba, abajo, más, menos, encender, apagar.

Dentro del alcance en la propuesta, se había planteado la elaboración de un sistema de reconocimiento que permita el funcionamiento a múltiples usuarios. Se enfrentó el problema de las variaciones de voz debido a diferentes usuarios a través de la red neuronal artificial, partiendo de una de sus características inherentes, la generalización. Como se sabe, cuando dos personas pronuncian una misma palabra, los componentes de frecuencia son diferentes, ya que factores tales como el tono y el timbre de la voz cambian dependiendo de la persona, por lo cual es necesario buscar un procedimiento que permita reducir las diferencias espectrales. Los MFCC permiten disminuir la diferencia espectral entre las pronunciaciones de las personas, pero cuando se trata del análisis de señales de voz de hombres y mujeres se presenta una diferencia significativa, inclusive con la aplicación de los MFCC.

Para enfrentar el problema de reconocimiento de voz multiusuario se utiliza la generalización de la red neuronal, la manera en la que se pretende que la ANN aprenda a reconocer diferentes usuarios, incluso personas de ambos sexos, es la construcción de una base de datos que almacene las voces de seis usuarios diferentes. Estos seis usuarios se dividieron en un grupo de tres hombres y tres mujeres. Cada uno de los usuarios pronunció cada palabra de las tres aplicaciones, cinco veces, esto con la finalidad de tener suficientes muestras de cada uno de los fonemas para entrenar una red neuronal artificial con suficiente información. Como la frecuencia de muestreo en el dispositivo es diferente a la de la aplicación elaborada en Matlab y como también el dispositivo y el códec de audio no cuentan con todas las capacidades de procesamiento que tiene la tarjeta de sonido del computador, es necesario tomar las muestras de voz directamente del DSP para hacer el entrenamiento de las ANN. Por eso, se elaboró un dispositivo sencillo compuesto por el DSP y un microcontrolador con conexión USB para permitir la adquisición de los datos en el computador. El dispositivo se encarga de detectar voz (mediante el algoritmo descrito en la sección 1.8) y enviarla al pc a través del puerto USB; además se elaboró una aplicación en Visual C# que recibe los datos y los almacena en un documento de texto. Estos datos posteriormente se pasaron a MATLAB, en donde se realizó las modificaciones correspondientes a la aplicación presentada en la sección 3.5 para realizar un procedimiento de entrenamiento igual al allí descrito. Se crearon bancos de datos con la información correspondiente a cada uno de los fonemas

de cada aplicación y se entrenaron las redes neuronales para el reconocimiento de voz mediante el algoritmo *Resilient Backpropagation*[**].

4.7 DIFERENCIAS ENTRE LA IMPLEMENTACIÓN EN LENGUAJE DE ALTO Y BAJO NIVEL

La implementación del sistema en el DSP se intentó hacer exactamente igual a la implementación mediante la herramienta computacional pero existieron varios factores que lo imposibilitaron, por lo que existen diferencias en cuanto al análisis de Fourier, la precisión de las operaciones, el procesamiento en la parte análoga de la señal, la falta de traslapado entre los bloques de datos y el no cálculo de los deltas.

El análisis de Fourier, realizado mediante MATLAB, es una función incrustada dentro de su código, su funcionamiento está basado en la librería denominada FFTW[*], la cual usa el algoritmo de Cooley Tukey. Mientras que en el DSP se utilizó la función en C desarrollada por Jens Jorgen Nielsen, el algoritmo es una mezcla de *split radix* en el nivel alto y de varios algoritmos veloces en el bajo nivel. Incluso cuando los dos algoritmos calculan la misma función, la operatoria realizada en el bajo nivel es diferente y el resultado final tiene unas pequeñas variaciones en los coeficientes de Fourier.

Aunque la mayoría del procesamiento de la señal se hace de manera digital, no se puede dejar de lado el hecho de que el procesamiento realizado de manera análoga es un factor que facilita el proceso del reconocimiento de voz. La tarjeta de sonido del computador en el que se trabajo cuenta con varios filtros analógicos que hacen que la señal que se digitaliza tenga un bajo nivel de contaminación, mientras que el dispositivo utilizado no cuenta con ningún filtro análogo por lo cual la tarea de reconocimiento se dificulta. Otro factor a considerar es que el proceso de digitalización hecho en el computador se manejó a 24 bits, mientras que en el dispositivo se manejó a 16 bits.

Otro elemento a considerar es que las operaciones realizadas en Matlab cuentan con una precisión de 64 bits, mientras que en el DSP se trabajan a 40 bits, ya que esta es la precisión máxima que maneja el compilador. La diferencia de precisión sería un factor despreciable si el procesamiento de la señal no fuese secuencial o si no necesitase tanta operatoria, pero debido a que estas dos condiciones se cumplen, existe una pequeña diferencia en los resultados al procesar los mismos datos en los dos sistemas.

---

[*] La librería para realizar el cálculo de la FFT mediante computadora se puede conseguir en http://www.fftw.org

[**] El algoritmo *Resilient Backpropagation* es una derivación del *Backpropagation* explicado en el capítulo 5.

Una diferencia significativa en cuanto al tratamiento de la señal está en el traslapado de los bloques. Como se explicó en la sección 4.4, el tiempo disponible para procesamiento, inicialmente no fue suficiente para poder realizar la detección de voz, debido a que el algoritmo de Fourier requería más del existente. Para hacer posible la detección de voz en tiempo real fue necesario añadir el algoritmo expresado en el documento Chris Lomont para calcular una FFT de N puntos reales mediante una transformada de $N/2$ puntos complejos; esto permitió que sea posible la detección de voz en tiempo real, pero a pesar de ello el tiempo de procesamiento quedó en el límite, es decir, no sobró tiempo para añadir ninguna clase de operaciones extra. Después de lograr la detección de voz en tiempo real se intentó hacer el procesamiento con traslapado de 30 muestras en los bloques, igual que en el procesamiento en alto nivel, pero lamentablemente la velocidad del elemento no fue suficiente, razón por lo cual se tuvo que continuar aboliendo el traslapado de datos.

Finalmente, cuando se escogió el ezDSP VC5505 USB Stick como componente para la implementación, se sobrestimó su velocidad de procesamiento y al no contar con una función para calcular la transformada de Fourier no se pudo determinar la cantidad real de operaciones involucrada en el procesamiento. Para lograr que la implementación incluya el traslapado de los bloques de voz, se requiere un elemento con mayor velocidad. Como sugerencia se recomienda un procesador con capacidad de operaciones en punto flotante y con una ALU de 32 bits.

5. RESULTADOS

Durante los anteriores capítulos se ha preocupado por explicar los fundamentos de la construcción del sistema y la implementación del mismo. En el presente capítulo se presenta los resultados del algoritmo propuesto para el cálculo de la FFT y de las aplicaciones definitivas, realizando una medida del rendimiento mediante tres ópticas diferentes: el proceso de entrenamiento, la verificación del porcentaje de reconocimiento sobre las personas que crearon la base de datos de entrenamiento y la verificación de la generalización del sistema mediante pruebas a hablantes que no pertenecen al grupo de entrenamiento, finalmente se presenta una discusión sobre la diferenciación entre el reconocimiento de voz dependiente del género.

5.1 IMPLEMENTACIÓN DE LA FFT

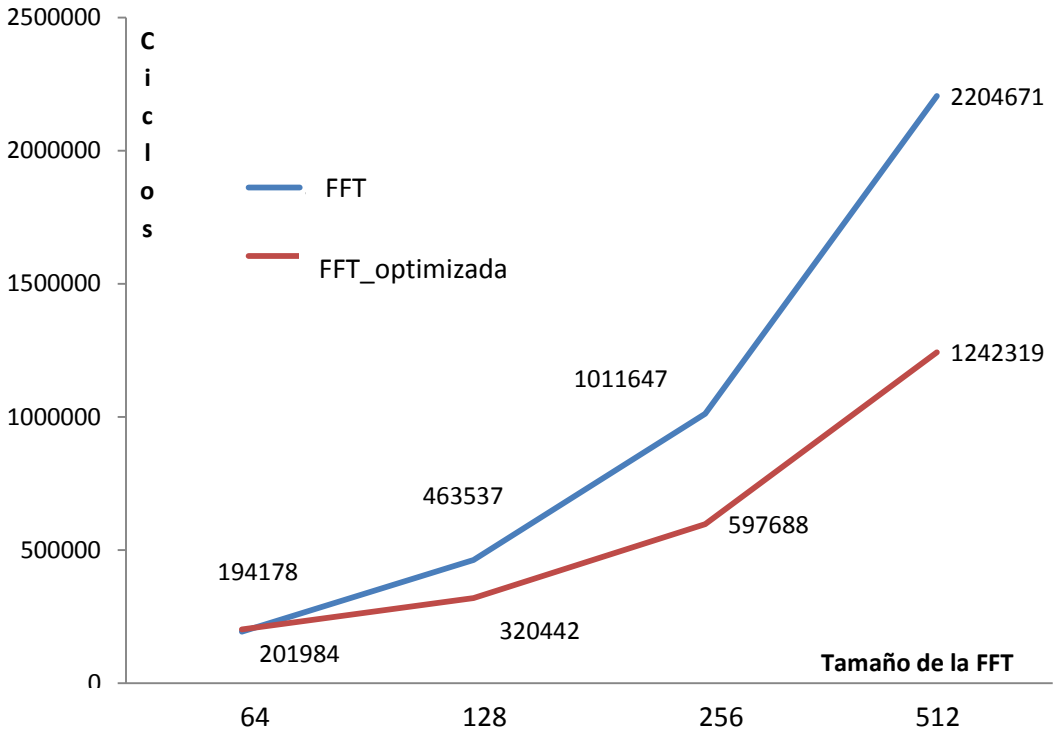
Como se explicó en la sección 4.3, para realizar la implementación de la detección de voz en tiempo real se hizo necesario optimizar el tiempo requerido para obtener una FFT. Con base en el documento de Chris Lomont se obtuvieron las ecuaciones que permiten obtener la FFT de N puntos reales mediante una FFT de $N/2$ puntos complejos, bajo una arquitectura que no soporta operaciones complejas. Para validar que las ecuaciones 54 – 58 permiten una reducción en el tiempo de procesamiento se realizó varias transformadas de Fourier con diferente cantidad de puntos. La tabla 6, muestra los resultados de la cantidad de ciclos de reloj que tarda el DSP en obtener la transformada.

Tabla 6. Comparación de los algoritmos implementados para el cálculo de la FFT

Número de Puntos	FFT	FFT optimizada
64	194178	201984
128	463537	320442
256	1011647	597688
512	2204671	1242319
1024	10954976	3109585

De acuerdo con la figura 32 se observa que entre más puntos contenga la transformada de Fourier a calcular, la diferencia en la cantidad de operaciones realizadas por los dos algoritmos es mayor. El algoritmo presentado en este trabajo, es útil para transformadas de más de 128 datos.

Figura 32. Comparación de los algoritmos implementados para el cálculo de la FFT



Para el presente trabajo, la implementación del algoritmo mostrado en el documento de Chris Lomont a través de las ecuaciones 54 – 58, significó una reducción del 40,92% del tiempo de procesamiento necesitado para el cálculo de la FFT y permitió la implementación del algoritmo de detección de voz en tiempo real.

5.2 RESULTADOS DE LA PRIMERA APLICACIÓN

La primera aplicación implementada fue el reconocimiento de dígitos, para esto fue necesario reconocer las siguientes palabras:

Tabla 7. Palabras a reconocer en la primera aplicación

No	Palabra
1	Uno
2	Dos
3	Tres

Tabla 7. Palabras a reconocer en la primera aplicación (continuación)

No	Palabra
4	Cuatro
5	Cinco
6	Seis
7	Siete
8	Ocho
9	Nueve
10	Cero

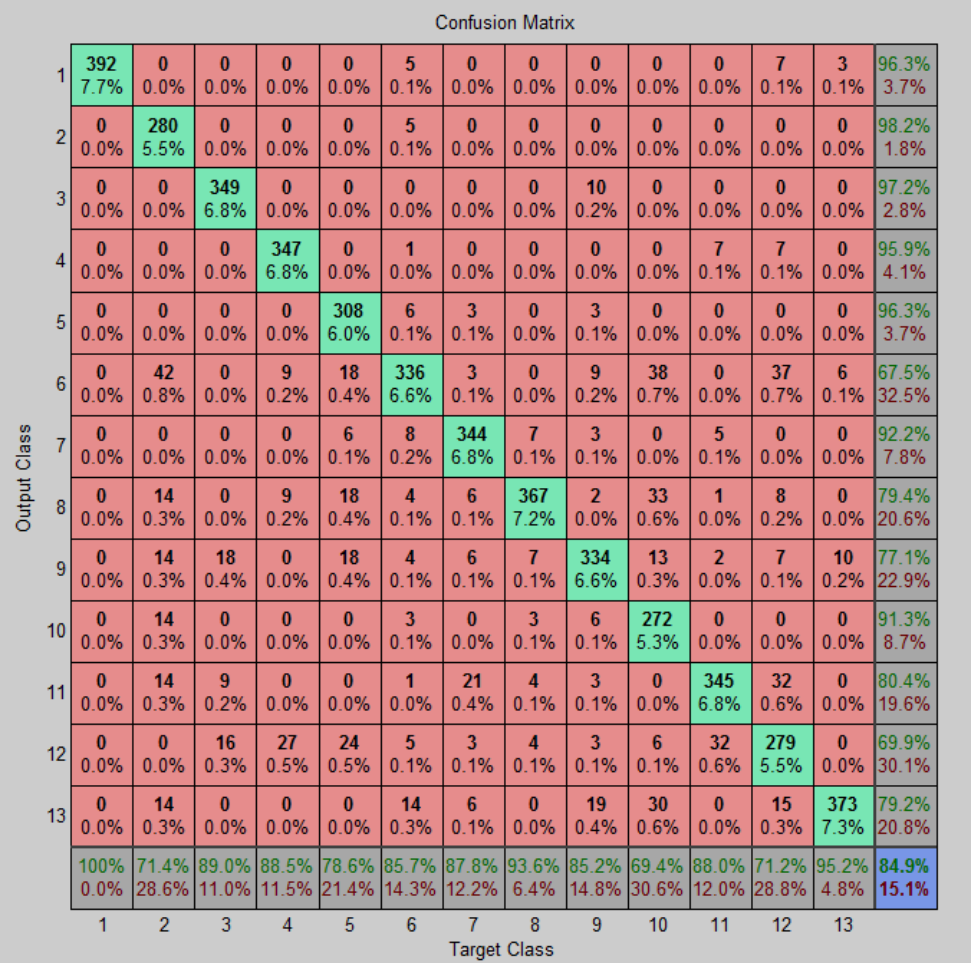
Para los cuales es necesario entrenar los fonemas:

Tabla 8. Fonemas involucrados en la primera aplicación

No	Fonema
1	A
2	B
3	C
4	CH
5	D
6	E
7	I
8	N
9	O
10	R
11	S
12	T
13	U

El nivel de rendimiento medido a través de error cuadrático medio (MSE, por sus siglás en inglés *Mean Squared Error*) fue de 0.0122. Un detalle más específico del entrenamiento se puede apreciar en la denominada *confusion matrix,* que muestra el porcentaje de acierto y error por fonema, además de especificar las equivocaciones de la salida de la red. Para esta red se obtuvo:

Figura 33. *Confusion Matrix* del entrenamiento de la primera aplicación



En la *confusion matrix*, los datos en verde de la fila inferior indican el porcentaje de acierto de las salidas de la red en comparación con los datos reales. Si miramos cada una de las columnas se mira una casilla en verde que indica la cantidad de aciertos que tuvo la red, las casillas en rojo indican la cantidad de datos que la red clasifico erróneamente. La casilla que se encuentra en la esquina inferior derecha muestra el porcentaje total de acierto y error de la red. Por otro lado si se observa la columna de la derecha los datos en verde indican el porcentaje de las salidas que clasificó la red como un fonema que de verdad corresponden a ese fonema, y el valor en rojo indica el porcentaje de veces que la red clasificó ese fonema cuando en verdad el fonema pronunciado era alguno diferente.

Para esta red, por ejemplo, se observa que el fonema 2, que de acuerdo a la tabla 8 corresponde al sonido de un B, presenta debilidad en la clasificación, si se mira la fila inferior solo el 71.4% de las veces que apareció en el conjunto de datos de entrenamiento se caracterizó correctamente, además se observa que cuando se

clasifica erróneamente, en la mayoría de las ocasiones se obtiene como salida el fonema 6 que corresponde a la pronunciación de una E, esto se debe a que los datos de la extracción de características de los MFCC que se utilizaron para entrenar el fonema B, se seleccionaron manualmente de la pronunciación de la palabra nueve, que es la única palabra con este fonema, por lo tanto debido a errores en el procedimiento de selección de los fonemas, algunos datos de entrenamiento corresponden al fonema de la letra E. Lo mismo observa para el fonema 12, correspondiente al sonido de una E, para el cual se presenta clasificaciones falsas con el fonema 6 (letra E), por que los datos de entrenamiento se obtienen de la palabra siete.

Cuando la red neuronal clasifica de manera errónea un fonema, pero el error corresponde al sonido del siguiente fonema dentro de la palabra pronunciada, no se genera un error significativo en el reconocimiento ya que gracias al modelado temporal del DTW la distancia global al finalizar el algoritmo no presentará diferencia significativa. Por ejemplo si el fonema 2 (B) se confundiese con el fonema 6 (E) en la palabra nueve, el alineamiento de tiempo se encargaría de que la distancia global se, conservase debido a la continuidad temporal de los fonemas 2 (B) y 6 (E). En contraposición, si la ANN clasifica erróneamente un fonema que no corresponde al sonido siguiente en la palabra pronunciada, la distancia global del DTW se verá afectada obteniendo un valor mayor. Por ejemplo, en la palabra siete, el fonema 12 (E) de acuerdo al entrenamiento puede tener tendencia a clasificarse erróneamente con el fonema 11 (S), obteniéndose un incremento en la distancia global, ya que este no existe en la parte final matriz de referencia para esta palabra.

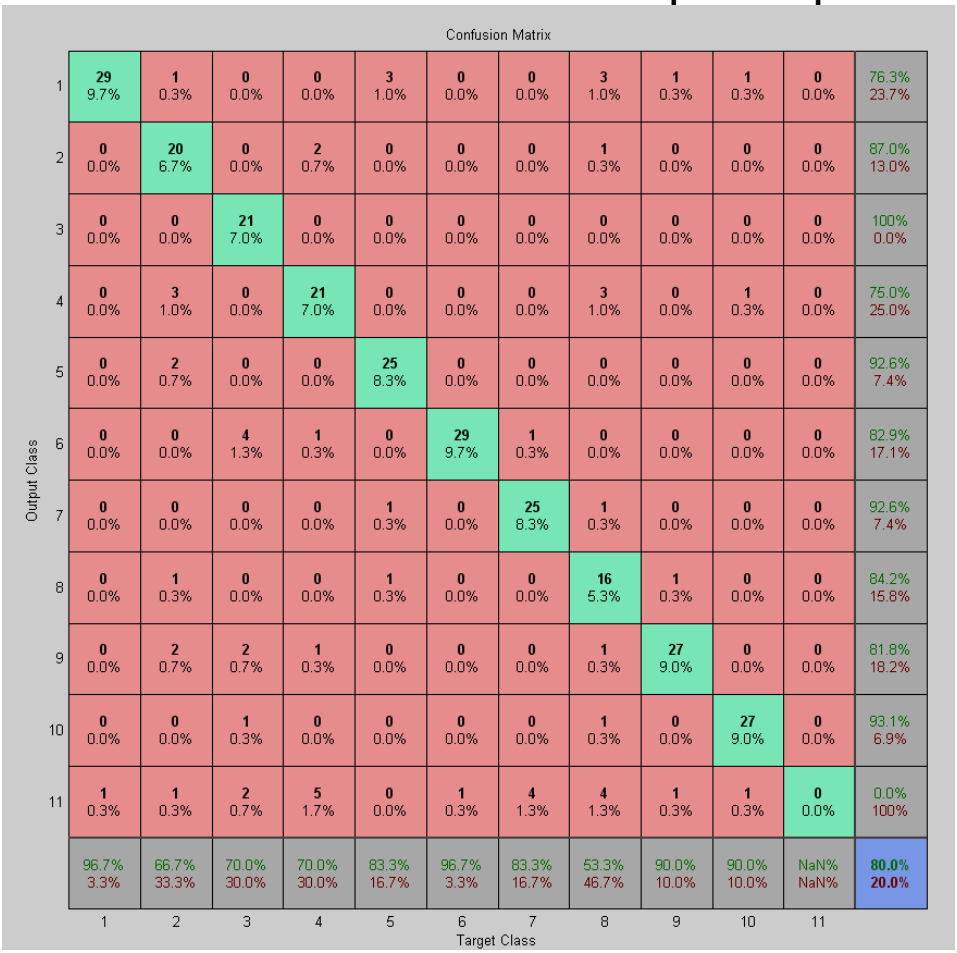
Tomando en cuenta lo anterior, para mejorar el funcionamiento del proceso de reconocimiento una de buena alternativa para mejorar el reconocimiento del sistema, es entrenar la red neuronal con el conjuntos de palabras que se desee reconocer, ya que de esta forma los errores de clasificación pueden llegar a jugar a favor del valor de distancia global obtenido mediante el DTW. De igual manera, analizar el comportamiento de salidas de la red neuronal permite la construcción de matrices de referencia que aumenten el reconocimiento del sistema, por ejemplo, se sabe que el fonema 6 (E) y el fonema 11 (S) poseen un alto porcentaje de reconocimiento, entonces duplicar la cantidad de apariciones de estos fonemas en la matriz de referencia conduce a una reducción de la distancia global de la matriz de referencia ya que reduce la medida de la distancia de los fonemas de las matrices de entrada y de referencia.

En ocasiones puede ocurrir que el algoritmo de detección de voz entregue entradas que no contienen voz, o puede ocurrir que solo se entregue solo parte de la palabra, cuando esto ocurre es mejor que el sistema detecte que la palabra no pertenece al vocabulario en lugar de entregar una palabra cualquiera. Para hacer que esto ocurra se fija un valor de distancia global que le permite al sistema decir que no entendió la palabra pronunciada. Haciendo que el sistema obtengan en su

mayoría salidas correctas o no entendidas se reduce el porcentaje de falsos positivos del sistema, lo cual tiende a mejorar el rendimiento del mismo.

El sistema se evaluó sobre las personas que conformaron el grupo de entrenamiento, para ello cada persona pronuncio cada palabra en 5 ocasiones, recolectando un total de 30 muestras[*]. Los resultados se presentan en forma de una *confusion matrix.*

Figura 34. *Confusion Matrix* de los resultados de la primera aplicación



Esta primera forma de evaluar el comportamiento del sistema se puede considerar como una aproximación de reconocimiento multi locutor, pero no independiente del hablante. El porcentaje más bajo de reconocimiento se encuentra en la palabra ocho, esto se debe a que en la red el fonema 4 (ch), se confunde con el sonido del de la t, por tanto la matriz de referencia con la que se compara la entrada no

[*] Vale la pena recordar que el grupo de entrenamiento se conforma por tres hombres y tres mujeres

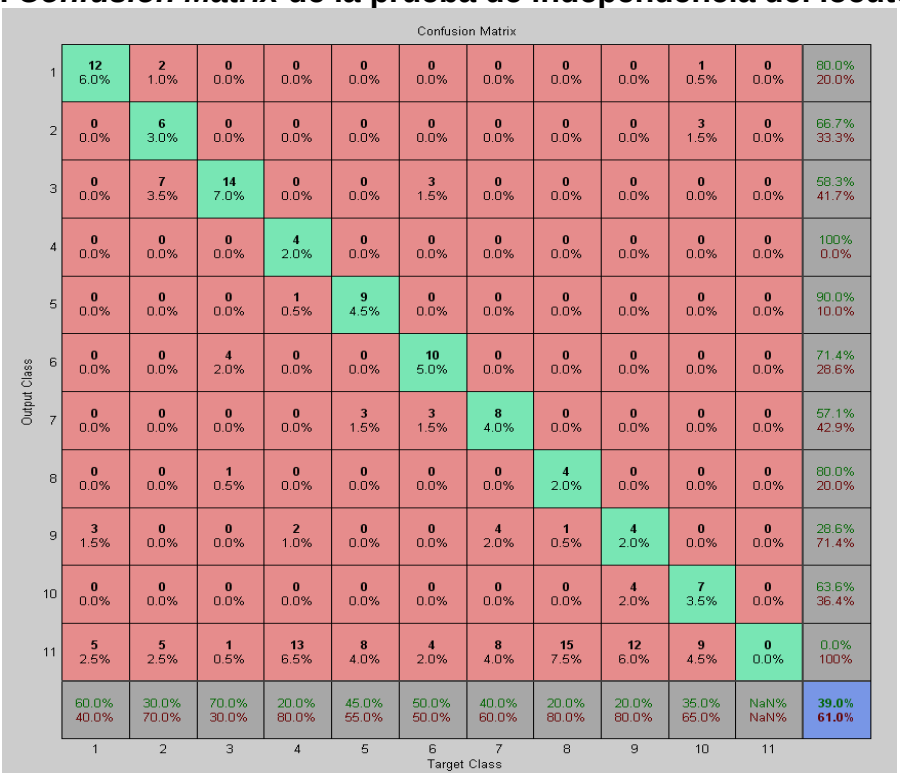
concuerda con las salidas de la red. En los resultados se observa que existieron varias palabras que no se entendieron y algunas otras que se reconocieron erróneamente como cuatros.

Tabla 9. Resultados de la primera aplicación

	Numero palabras	Porcentaje
Palabras clasificadas correctamente	240	80%
Palabras clasificadas incorrectamente	40	13.3%
Palabras sin clasificación	20	6.7%
Total de Palabras	300	100%

Para observar la capacidad de generalización de la red neuronal, se hizo una prueba similar de reconocimiento para un grupo de personas fuera del grupo de entrenamiento. Se probó con dos hombres y dos mujeres diciendo cada palabra 5 veces. Los resultados se muestran en la siguiente *confusion matrix*.

Figura 35. *Confusion Matrix* de la prueba de independencia del locutor



En la *confusion matrix* se observa que el porcentaje de reconocimiento es de solo el 39%, sin embargo la mayoría de las palabras no reconocidas se clasifican como no entendidas. Esto se debe a que la red neuronal no posee información sobre los fonemas de estas personas, por lo cual no se tiene una salida apropiada de la red neuronal, haciendo que la distancia total del DTW crezca, y supere el umbral de reconocimiento.

Tabla 10. Resultados de la prueba de generalización de la primera aplicación

	Numero palabras	Porcentaje
Palabras clasificadas correctamente	78	39%
Palabras clasificadas incorrectamente	42	21%
Palabras no clasificadas	80	40%
Total de Palabras	200	100%

5.3 RESULTADOS DE LA SEGUNDA APLICACIÓN

La segunda aplicación implementada fue un control para el manejo de un televisor, se manejan algunos comandos básicos, como encender y apagar el televisor, subir y bajar los canales al igual que el volumen; para esto fue necesario reconocer las siguientes palabras:

Tabla 11. Palabras a reconocer en la segunda aplicación

No	Palabra	Función
1	Encender	Enciende el televisor
2	Apagar	Apaga el televisor
3	Arriba	Sube el canal
4	Abajo	Baja el canal
5	Mas	Sube el volumen
6	Menos	Baja el Volumen

La aplicación se realizó para televisores Sony y lo que hace es enviar pulsos a través de un transmisor infrarrojo dependiendo de la palabra reconocida. Cabe resaltar que si el televisor esta encendido y el usuario dice el comando encender el televisor se apaga y viceversa, puesto que no se tiene control del estado inicial del televisor. Para esta aplicación es necesario entrenar los siguientes fonemas:

Tabla 12. Fonemas involucrados en la segunda aplicación

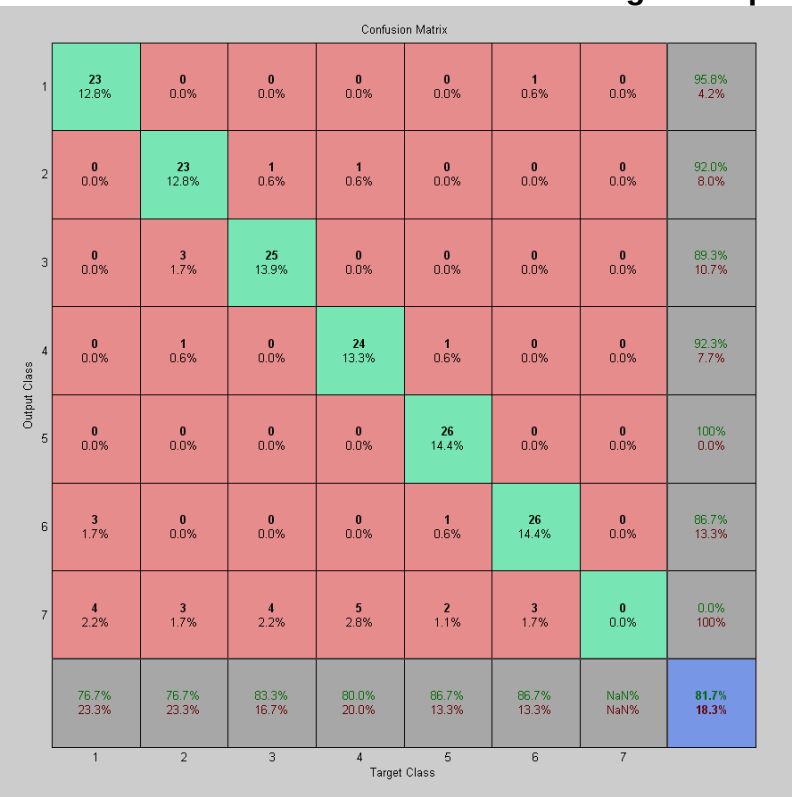
No	Fonema
1	A
2	B
3	D
4	E
5	G
6	I
7	J
8	M
9	N
10	O
11	P
12	R
13	S

El nivel de rendimiento del entrenamiento de la ANN, medido a través de error cuadrático medio fue de 0.0092. Un detalle más específico del entrenamiento se puede apreciar en la siguiente *confusion matrix.*

Figura 36. *Confusion Matrix* del entrenamiento de la segunda aplicación

El sistema se probó con las personas que conformaron el grupo de entrenamiento cada persona dijo cada palabra en 5 ocasiones. En la *confusion matrix* se muestra el porcentaje de reconocimiento para la aplicación.

Figura 37. *Confusion Matrix* de los resultados de la segunda aplicación



Para esta aplicación se tiene un porcentaje de reconocimiento del 81.7%, lo cual muestra la capacidad del sistema para manejar varios locutores. Se observa que la mayoría del porcentaje que no se reconoce, se encuentra clasificado como palabras que no entendió el sistema.

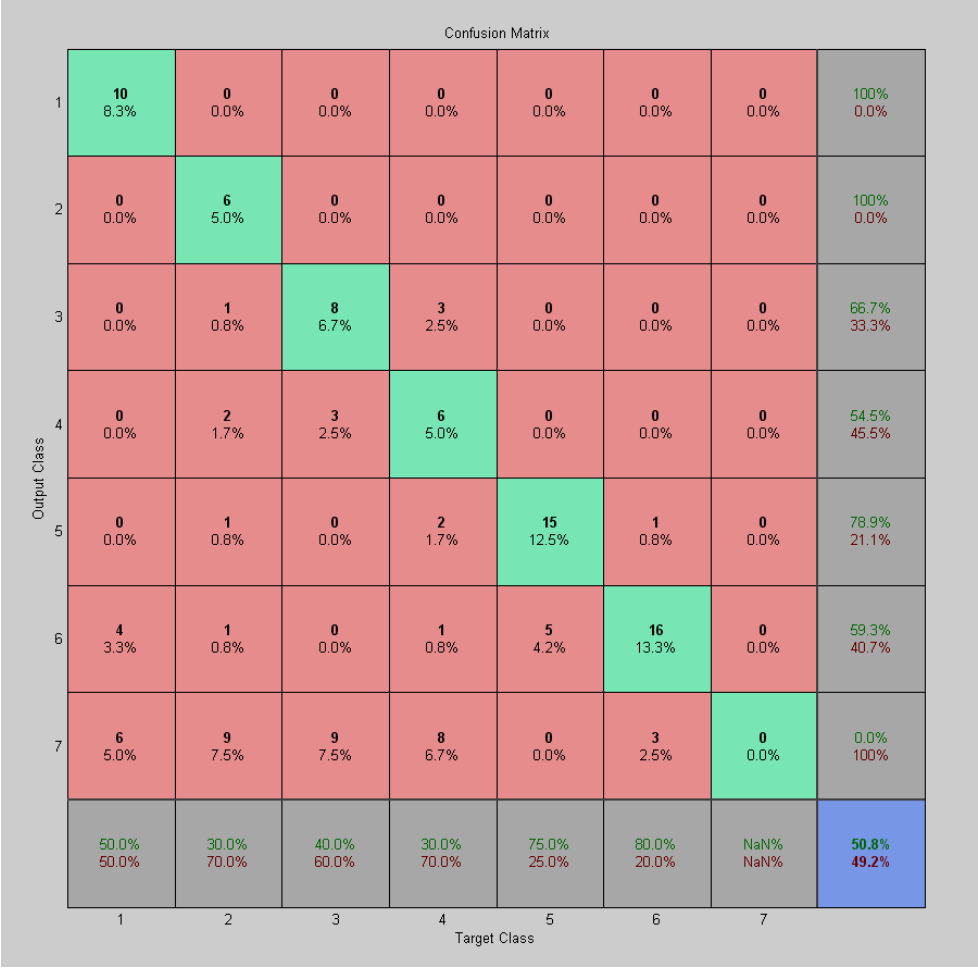
Tabla 13. Resultados de la segunda aplicación

	Numero palabras	Porcentaje
Palabras clasificadas correctamente	147	81.7%
Palabras clasificadas incorrectamente	12	6.7%
Palabras no clasificadas	21	11.6%
Total de Palabras	180	100%

El porcentaje de error del sistema, considerando solo las palabras que fueron clasificadas erróneamente es del 6.67%.

Para probar la independencia del hablante del sistema se probó con cuatro personas fuera del grupo de entrenamiento dos mujeres y dos hombres.

Figura 38. *Confusion Matrix* de la prueba de independencia del locutor



El porcentaje de clasificación correcta de 50.8% se debe a que las palabras más y menos, tienen un alto nivel de reconocimiento puesto que las dos tienen una vocal muy marcada y terminan en s, además son muy diferentes de las demás palabras del sistema. Se observa que el porcentaje de palabras que no son clasificadas es mayor que las palabras que se clasifican de manera equivocada.

Tabla 14. Resultados de la prueba de generalización de la segunda aplicación

	Numero palabras	Porcentaje
Palabras clasificadas correctamente	61	50.8%
Palabras clasificadas incorrectamente	24	20%
Palabras no clasificadas	35	29.2%
Total de Palabras	120	100%

5.4 RESULTADOS DE LA TERCERA APLICACIÓN

La tercera aplicación implementada fue el control del movimiento de un carro robótico; para esto se reconocen las siguientes palabras:

Tabla 15. Palabras a reconocer en la tercera aplicación

No	Palabra	Función
1	Adelante	Mueve el robot hacia adelante
2	Atrás	Mueve el robot hacia atrás
3	Derecha	Gira el robot a la derecha
4	Izquierda	Gira el robot a la izquierda
5	Pare	Detiene el robot

El sistema de reconocimiento da instrucciones de movimiento a un carro robótico a partir de los comandos pronunciados, se puede dar movimiento hacia adelante hacia atrás, girar a la derecha e izquierda y detener el carro.

Para la aplicación es necesario entrenar los siguientes fonemas:

Tabla 16. Fonemas involucrados en la tercera aplicación

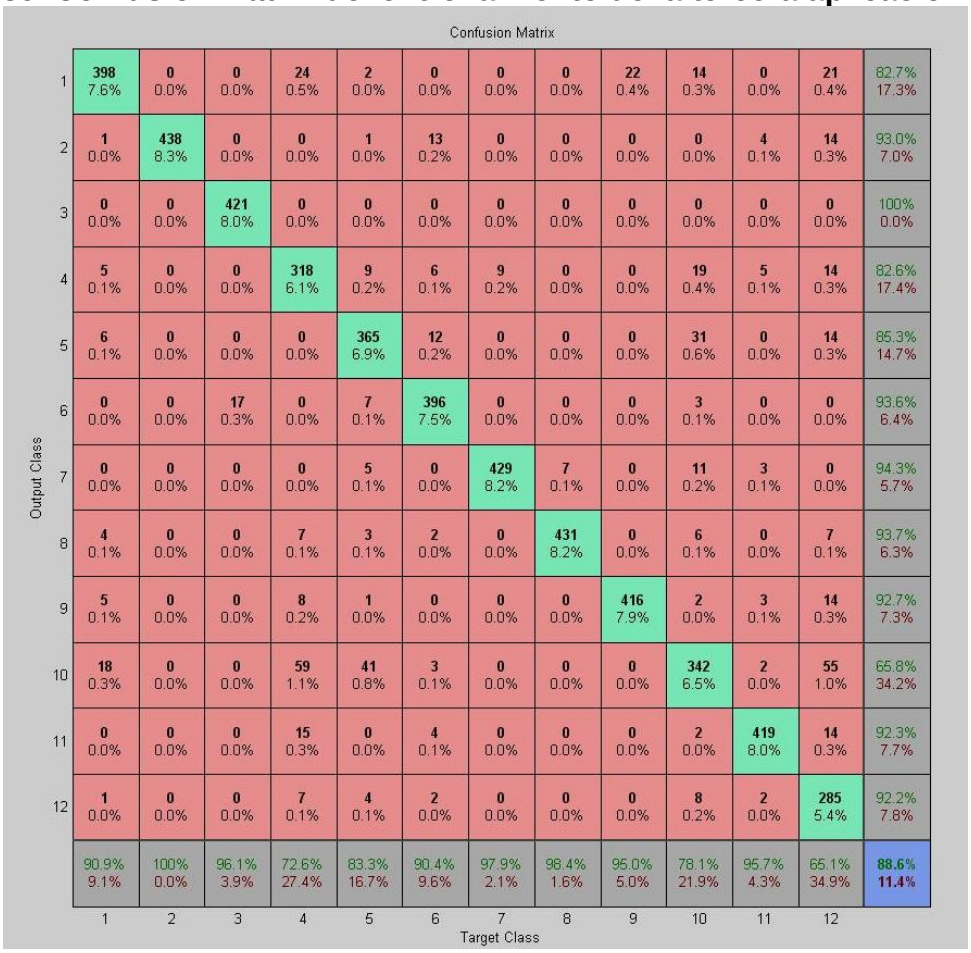
No	Fonema
1	A
2	C
3	CH
4	D
5	E

Tabla 16. Fonemas involucrados en la tercera aplicación (continuación)

No	Fonema
6	I
7	L
8	N
9	P
10	R
11	S
12	T

El nivel de rendimiento medido a través de error cuadrático medio fue de 0.0103. Un detalle más específico del entrenamiento se puede apreciar en la *confusion matrix,* que muestra el porcentaje de acierto y error por fonema, además de especificar contra que se equivocó la salida de la red. Para esta red se obtuvo:

Figura 39. *Confusion Matrix* del entrenamiento de la tercera aplicación



El sistema se probó con las personas que conformaron el grupo de entrenamiento, cada persona pronunció cada una de las palabras en 5 ocasiones. En la *confusion matrix* se muestra el porcentaje de reconocimiento para la aplicación y la tabla 18 muestra una descripción incluyendo el número de palabras que el sistema clasificó como no entendidas.

Figura 40. *Confusion Matrix* de los resultados de la tercera aplicación

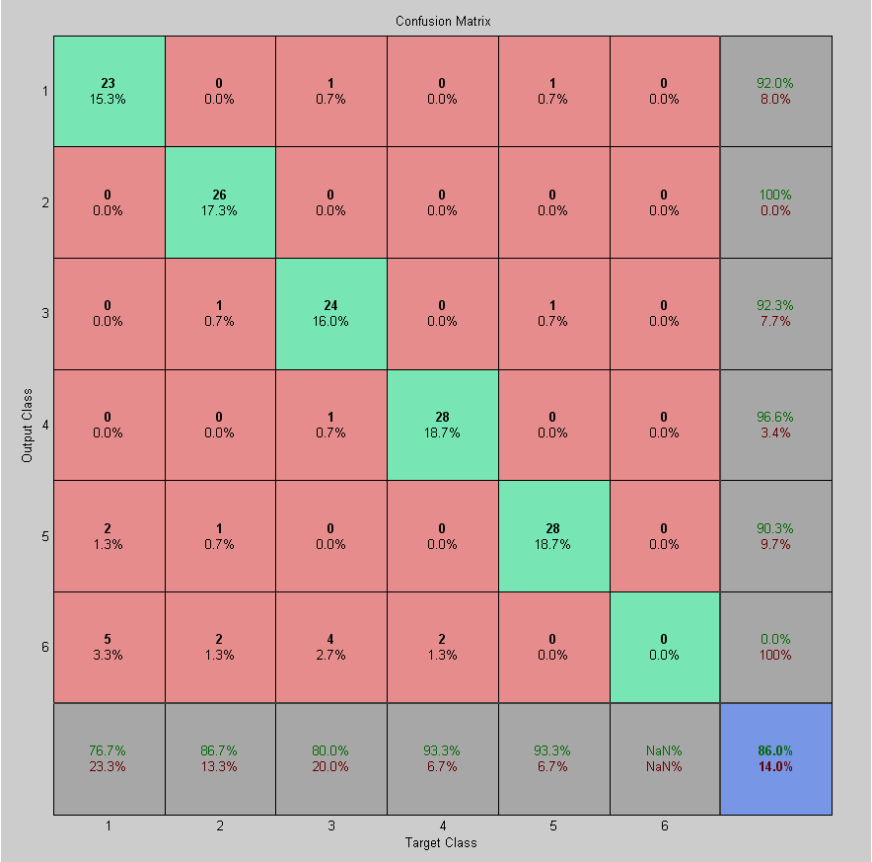


Tabla 17. Resultados de la tercera aplicación

	Numero palabras	Porcentaje
Palabras clasificadas correctamente	129	86%
Palabras clasificadas incorrectamente	8	5.33%
Palabras no clasificadas	13	8.67%
Total de Palabras	150	100%

Para esta aplicación se obtuvo un porcentaje de reconocimiento del 86%, se observa que el error del sistema es de 5.33%, el porcentaje de palabras no clasificadas es mayor que el de palabras erróneamente clasificadas.

De igual manera, el sistema se probó con un grupo de 4 personas para mirar la generalización del sistema hacia la independencia del locutor. Los resultados obtenidos se muestran en la figura 41 y tabla 18.

Figura 41. *Confusion Matrix* de la prueba de independencia del locutor

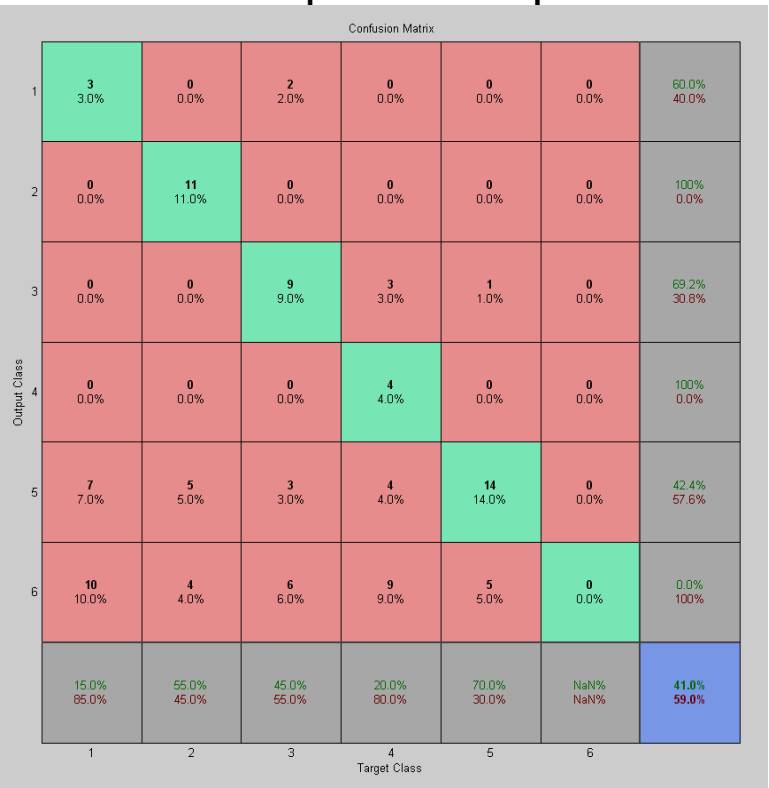


Tabla 18. Resultados de la prueba de generalización de la tercera aplicación

	Numero palabras	Porcentaje
Palabras clasificadas correctamente	41	41%
Palabras clasificadas incorrectamente	25	25%
Palabras no clasificadas	34	34%
Total de Palabras	100	100%

Se observa que el porcentaje de reconocimiento es bajo para personas fuera del grupo de entrenamiento, de igual manera el porcentaje de palabras no clasificadas supera a las palabras clasificadas de manera errónea.

5.5 RESULTADOS DEL ALGORITMO DE RECONOCIMIENTO

La implementación del algoritmo en DSP obtuvo un porcentaje de reconocimiento promedio del 83% sobre el grupo de personas que conformaron el conjunto de entrenamiento.

Figura 42. Porcentaje de reconocimiento de las aplicaciones

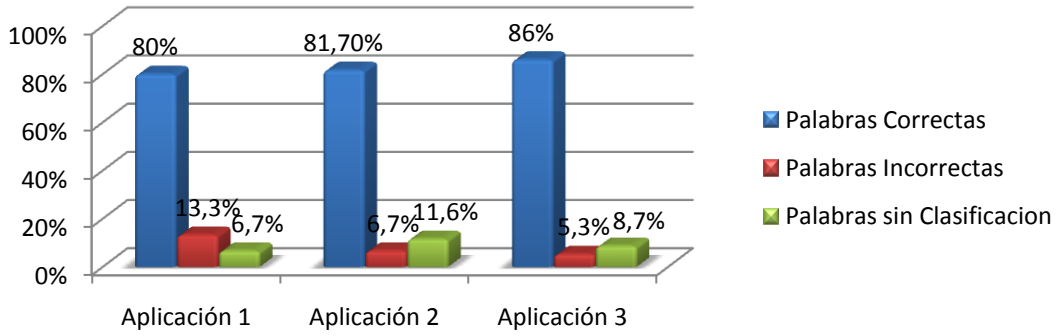


Figura 43. Valores promedio de reconocimiento de las aplicaciones

Promedio de las Aplicaciones	Porcentaje
Palabras clasificadas correctamente	83%
Palabras clasificadas incorrectamente	8%
Palabras sin Clasificación	9%
Total de Palabras	100%

El sistema presenta un error promedio del 8% y un promedio de palabras sin clasificación del 9%. Tener palabras sin clasificación mejora el desempeño del sistema, pues aunque el usuario tenga que repetir la palabra de control, el dispositivo que se está controlando no realiza una acción equivocada. Las pruebas de reconocimiento, demuestran que la red neuronal está en capacidad de generalizar los fonemas con los cuales ha sido entrenada y la metodología del presente proyecto permite desarrollar sistemas multiusuario.

Por otro lado, las pruebas que se realizaron para ver la capacidad de manejo que cuenta el sistema frente a la independencia del hablante muestran que el sistema presenta dificultad en el reconocimiento en este aspecto. Se obtuvo un porcentaje de reconocimiento promedio del 44%, aunque el criterio de palabras no entendidas por el sistema permite que el promedio de error del sistema sea del 34%. Esto se debe a que las personas fuera del grupo de entrenamiento poseen fonemas que la red neuronal no está en capacidad de clasificar, por lo cual en la mayoría de los casos la medida de la distancia global del DTW sobrepasa el umbral para indicar que no se reconoció la palabra. De este modo, se obtuvo un porcentaje de error de 22%, sobre las personas fuera del grupo de entrenamiento.

Figura 44. Porcentaje de reconocimiento de las aplicaciones con locutores externos

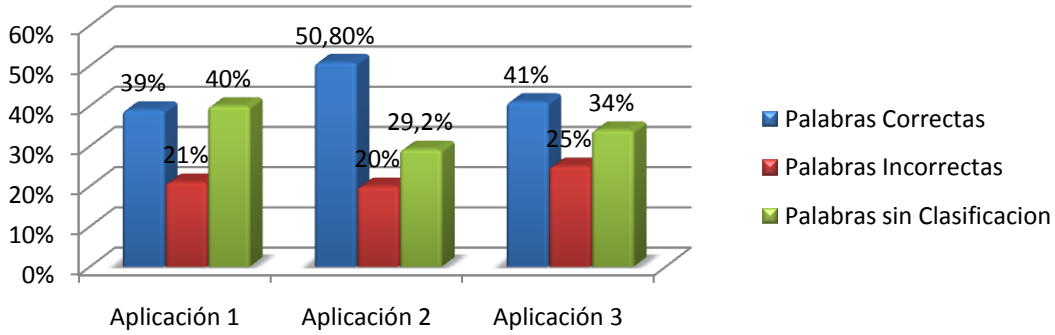


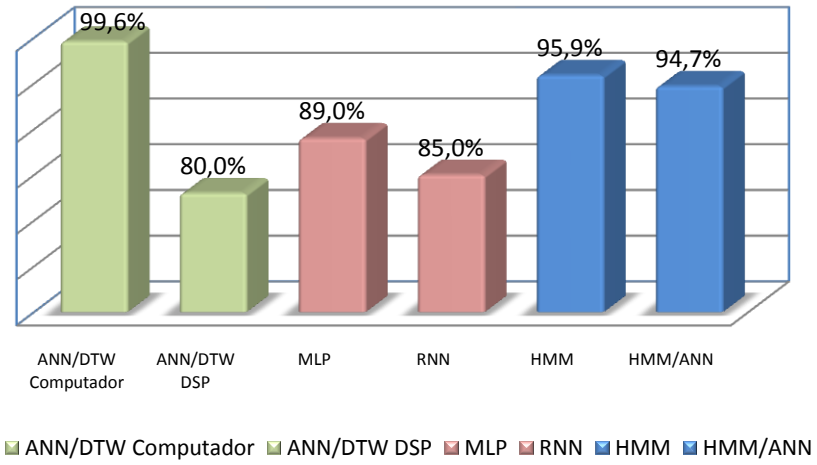
Tabla 19. Porcentaje de reconocimiento promedio para locutores externos

	Porcentaje
Palabras clasificadas correctamente	44%
Palabras clasificadas incorrectamente	22%
Palabras sin Clasificación	34%
Total de Palabras	100%

5.6 COMPARACIÓN DEL SISTEMA CON OTROS TRABAJOS

Como se mencionó durante los capítulos anteriores, el reconocimiento de dígitos es una aplicación que es muy común dentro del reconocimiento de voz. La figura 45 compara las implementaciones realizadas en el presente trabajo con implementaciones de otros dos trabajos.

Figura 45. Comparación de algoritmos de reconocimiento de dígitos



Los datos en verde muestran los porcentajes de reconocimiento para el algoritmo diseñado en el presente trabajo. Los datos en rojo fueron obtenidos de los sistemas de reconocimiento de dígitos basados en redes neuronales, propuestos por Lalith Kumar y Kishore Kumar[19], el primero de ellos basado en el perceptrón multicapa y el segundo en una red neuronal recurrente. Y los datos en azul corresponden al modelo propuesto por Xian Tang[20], para el reconocimiento de dígitos mediante HMM y un hibrido entre HMM/ANN.

El algoritmo implementado en el presente proyecto mediante el híbrido ANN/DTW en computadora muestra un nivel superior de reconocimiento en comparación con todas las otras implementaciones, validando la suposición de que las ANN pueden realizar el modelado acústico si están acompañadas del DTW para el modelado temporal, además exhiben mejores resultados que las otras implementaciones.

Por otro lado, el modelo del presente proyecto implementado sobre el DSP, a pesar de que el porcentaje de reconocimiento es comparable con los otros sistemas, tiene el menor rendimiento de los mostrados en la figura 45. Como se describió en la sección 4.7, debido a la velocidad del elemento seleccionado, la implementación de los algoritmos en el dispositivo embebido sufrió varias modificaciones causando así la reducción en el reconocimiento que se presenta en la figura 45.

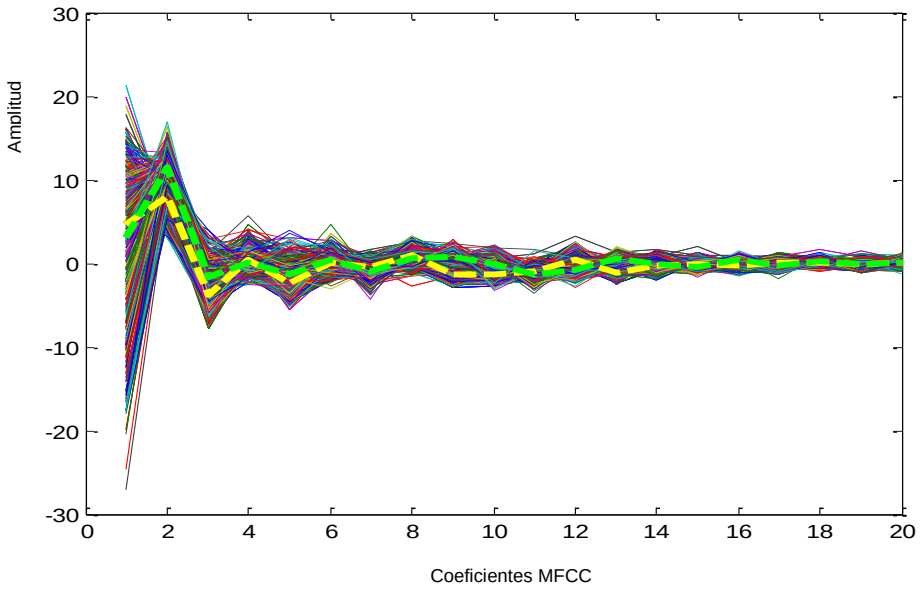
[19] KUMAR, Kishore y KUMAR, Lalith. Speech Recognition Using Neural Networks. Singapore: International Conference on Signal Processing Systems, 2009.
[20] TANG, Xiang. Hybrid Hidden Markov Model and Artificial Neural Network for Automatic Speech Recognition. Wuhan City: Communications and System Pacific-Asia Conference on Circuits, 2009.

5.7 ANÁLISIS DE FONEMAS POR GÉNERO DEL HABLANTE

En el capítulo 4, se manifestó la hipótesis de que iba a ser posible enfrentar el problema de independencia del locutor a partir de la propiedad de generalización que caracteriza a las redes neuronales. Como se puede observar a lo largo de las pruebas realizadas para mirar el grado de independencia del locutor alcanzada con el algoritmo implementado, la generalización de la red neuronal artificial no es suficiente para permitir caracterizar todas las representaciones existentes de los fonemas. De cualquier manera, como se manifestó durante los resultados del presente capítulo, sí es posible crear un sistema de reconocimiento de voz multiusuario bajo las condiciones del presente proyecto.

A lo largo del desarrollo del proyecto se tuvo gran contacto con las representaciones espectrales de la voz y se observó de manera empírica que existe una diferencia entre las representaciones para hombres y para mujeres. De acuerdo con lo observado, se propone para un futuro trabajo realizar la implementación de redes neuronales diferentes para hombres y mujeres. Para dar una muestra de la validez de la hipótesis, a continuación se realiza un análisis sobre los datos del fonema a, para los dos géneros.

Figura 46. Representación en forma de MFCC para el fonema A pronunciado por hombres y mujeres



En el eje vertical se ubica la magnitud de cada coeficiente, y cada valor del eje horizontal representa un coeficiente cepstral de frecuencia de Mel. El conjunto de líneas delgadas que se muestran en la figura 46 corresponden a la representación espectral de todos los fonemas que se clasificaron como el fonema A. La línea amarilla es la medida de la concentración del fonema a para las pronunciaciones

de mujeres, y la línea verde es la medida de la concentración del mismo fonema para hombres.

Figura 47. Representación en forma de MFCC para el fonema A pronunciado por hombres

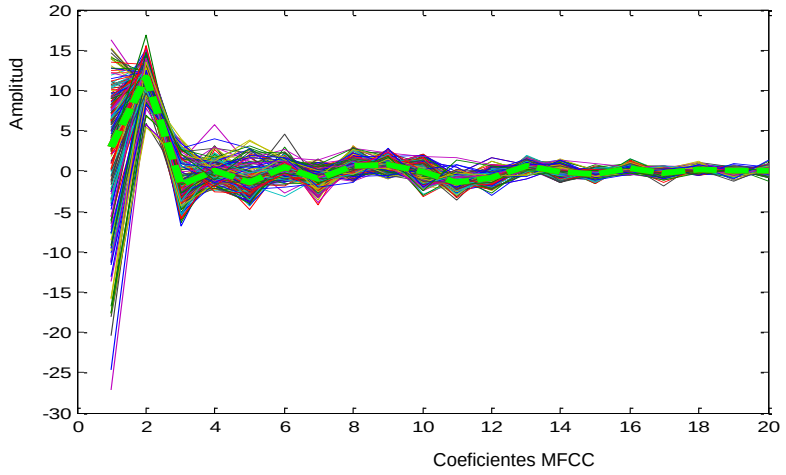
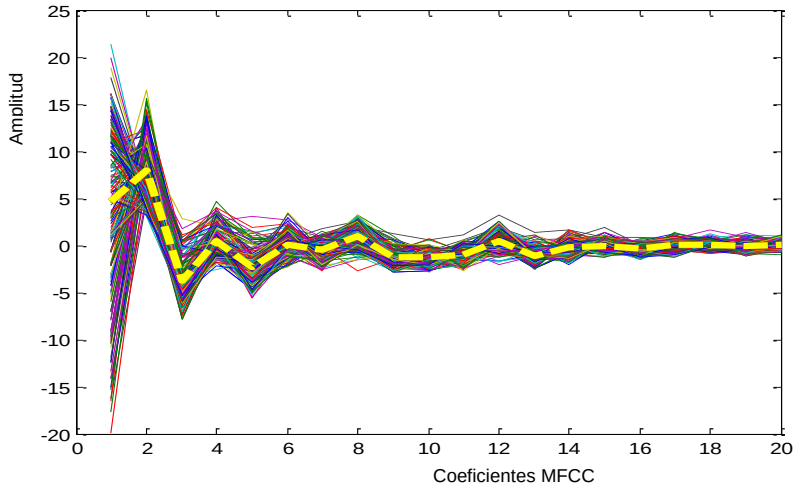
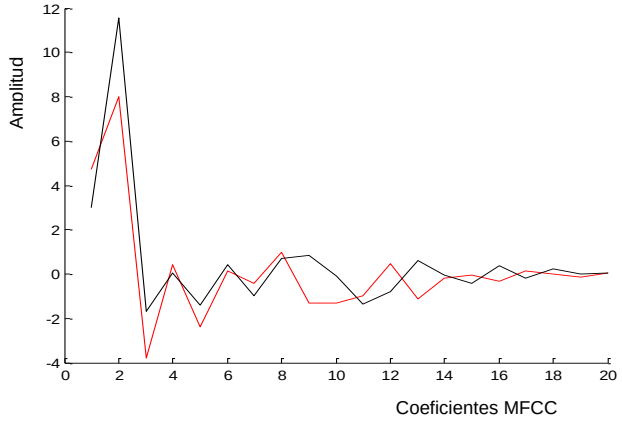


Figura 48. Representación en forma de MFCC para el fonema A pronunciado por mujeres



Para notar de manera más clara las diferencias en las representaciones de los fonemas de la vocal A para los dos géneros, a continuación se muestra la figura 49 que compara las concentraciones.

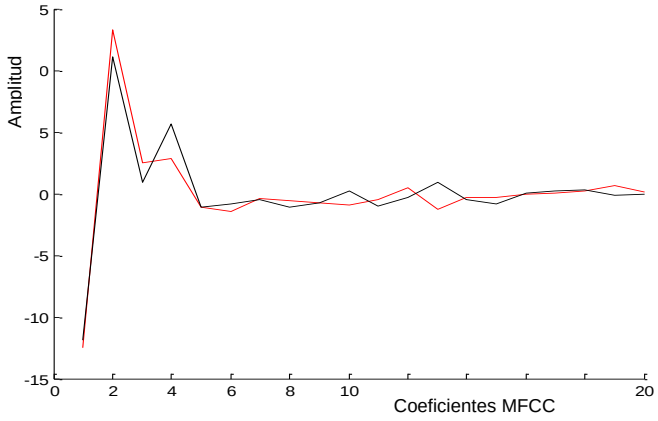
Figura 49. Comparación de las concentraciones de las representaciones en forma de MFCC



La línea en color rojo representa la concentración de las representaciones en forma de MFCC para los hombres y la línea en negro para mujeres. Analizando los resultados se observa que los primeros ocho coeficientes tienen diferencia en cuanto a amplitud, lo cual no representa un problema tan importante para la red neuronal ya que la forma del patrón se conserva; pero a partir del octavo coeficiente se nota que los picos de la figura 49 para hombres y mujeres se encuentran corridos, lo que representa una dificultad significativa para la red neuronal ya que la forma del patrón es diferente.

El problema se manifiesta de manera similar para la mayoría de fonemas, para confirmar esta afirmación a continuación en la figura 50 se compara las concentraciones del fonema del sonido de una N, para hombres y mujeres.

Figura 50. Comparación de las concentraciones de las representaciones en forma de MFCC para una N



De igual manera, los picos cambian en amplitud durante los primeros coeficientes, y se manifiestan en diferentes coeficientes después del octavo coeficiente. Por lo anterior, es lógico suponer que una red neuronal entrenada dependiendo del género seguramente llevará a una mejor clasificación de los fonemas y por tanto a un mayor porcentaje de reconocimiento.

6. CONCLUSIONES

Se demostró que las redes neuronales artificiales ofrecen una solución muy atractiva para el modelado acústico. Para lo cual se implementaron en computadora tres algoritmos para el reconocimiento de dígitos, específicamente el algoritmo mediante HMM, un híbrido ANN/DTW y un híbrido ANN/HMM. Se mostró que el algoritmo mediante ANN/DTW supera en porcentaje de reconocimiento a los otros, validando así que el modelado acústico mediante ANN supone una mejoría para los sistemas ASR.

En cuanto a la comparación de los sistemas ANN/DTW contra el ANN/HMM podemos decir que la razón principal por la cual el primer modelo presentó resultados superiores es que la operatoria requerida para el cálculo de probabilidades en los HMM conlleva a un problema de escalamiento, el cual, incluso es capaz de desbordar el manejo realizado por la computadora a 64bits. Dado que el dispositivo embebido utilizado para la implementación solo maneja operaciones a 40bits es lógico pensar que el problema de escalamiento es aún más complicado de manejar; de cualquier manera para realizar una afirmación más acertada de cual algoritmo presenta mejores características para la realización de modelado temporal es necesario buscar algoritmos que permitan manejar el problema de escalamiento de una mejor manera.

El problema de implementación de reconocimiento de voz mediante redes neuronales artificiales en dispositivos embebidos se manejó mediante la traducción de las ecuaciones matemáticas involucradas en los algoritmos a operaciones sencillas. Debido a que la velocidad del ezDSP VC5505 USB Stick no fue suficiente para realizar el procesamiento en tiempo real de la detección de voz existen diferencias en cuanto a la implementación, entre el tratamiento en computadora y el DSP. De cualquier manera, se exhibe que las acciones necesarias para elaborar reconocimiento de voz son posibles en tiempo real y en dispositivos embebidos.

Las ecuaciones 54 – 58 presentadas en la sección 4.3 fueron desarrolladas en este proyecto basadas en la teoría presentada en el documento de Chris Lomont y permiten el cálculo de una transformada de Fourier de N puntos reales en menor tiempo, ya que se realizan mediante una transformada de $N/2$ puntos complejos, y facilita la codificación cuando el algoritmo FFT se implementa sobre una arquitectura que no soporta operaciones complejas, para señales reales de 256 puntos se logró reducir el tiempo de procesamiento en un 40.92% al calcular la FFT.

Mediante las pruebas experimentales se determinó que las redes neuronales entrenadas mediante un grupo de seis personas no son capaces de afrontar el problema de independencia del locutor.

La independencia de locutor depende principalmente del género, mediante las gráficas de concentración del espectro para algunos fonemas, se presentó la hipótesis de que el manejo de hombres y mujeres en una misma red neuronal lleva a que sea más complejo la identificación de patrones dentro del conjunto de datos de entrenamiento.

Mediante la implementación en el DSP del sistema de reconocimiento de voz se construyeron tres aplicaciones: un reconocedor de dígitos, un control remoto para televisor Sony y un carro robótico. Para la primera aplicación se obtuvo un porcentaje de reconocimiento del 80%, para la segunda 81.7% y para la tercera 86%; estos resultados a pesar de ser menores al de la aplicación en computadora (99.6%) permiten concluir que es posible desarrollar algoritmos de reconocimiento de palabras aisladas mediante redes neuronales artificiales en dispositivos embebidos.

La presente investigación es uno de los primeros trabajos que se han desarrollado sobre reconocimiento de voz en nuestra región, por lo cual, se ayudará a la realización de futuras investigaciones sobre el tema, ya que presenta una revisión completa de la teoría necesaria para la elaboración de sistemas ASR.

7. RECOMENDACIONES

Construir un algoritmo de escalamiento que permita dar más soporte al algoritmo elaborado mediante HMM para poder realizar una comparación más precisa con el algoritmo DTW y así, determinar cuál de los algoritmos permite realizar un mejor modelado temporal y a su vez obtener mayor porcentaje de reconocimiento.

Implementar el algoritmo de reconocimiento de voz, propuesto en esta investigación, sobre una arquitectura de 32 bits que cuente con operaciones de punto flotante y permita traslapar los bloques de voz a reconocer.

Realizar una investigación para enfrentar la independencia del locutor a través de la creación de redes neuronales dependientes del género, y también, crear un algoritmo que permita determinar el género del locutor.

Implementar diferentes tipos de algoritmos de extracción de características para determinar si esto conduce a una mejoría en el porcentaje de reconocimiento de un sistema ASR.

Construir un sistema embebido para reconocimiento de voz de amplio vocabulario.

BIBLIOGRAFÍA

AMONE, Luigi; BOCCHIO, Sara y ROSTI, Alberto. On embedded system architectures for speech recognition applications: the gap between the status and the demand. Italy: Fourth IEEE International Symposium on Signal Processing and Information Technology, 2004.

BIN-AMIN, Talal y MAHMOOD, Iftekhar. Speech Recognition Using Dynamic Time Warping. Islamabad: 2nd International Conference on Advances in Space Technologies, 2008.

CHOU, Wu y JUANG, Biing. Pattern recognition in speech and language processing. USA: CRC Press, 2003. p 64-65.

CONG, Lin, et al. Robust Speech Recognition Using Neural Networks and Hidden Markov Models. Las Vegas: Information Technology Coding and Computing, 2000

EL-RAMLY, Salwa, et al. Neural Networks Used for Speech Recognition. Alexandria: Nineteenth National Radio Science Conference, 2002.

GELLATLY, Andrew William. The use of speech recognition technology in automotive applications. Tesis de Doctorado. Virginia: Virginia Tech, 1997.

JAIN, Anil y MAO, Jianchang. Artificial Neural Networks: A Tutorial. En: IEEE Computer Science Magazin. Marzo 1996. Vol 29, no 3.

KUMAR, Kishore y KUMAR, Lalithl. Speech Recognition Using Neural Networks. Singapore: International Conference on Signal Processing Systems, 2009.

LOMONT, Chris. The Fast Fourier Transform. Chris Lomont's homepage[online]. Enero de 2010. Available from www.lomont.org

MAALY, Iman y EL-OBAID, Manal. Speech Recognition using Artificial Neural Networks. Damascus: Information and Communications Technologies, 2006.

MUZAFFAR, Fariha, et al. DSP Implementation of Voice Recognition Using Dynamic Time Warping Algorithm. Karachi: Student Conference on Engineering Sciences and Technology, 2005.

RABINER, Lawrence y JUANG Biing. Fundamentals of Speech Recognition. New Jersey: Prentice Hall International, 1993.

RABINER, Lawrence. A tutorial on hidden Markov models and selected applications in speech recognition. USA: Proceedings of the IEEE, 1989. P 277.
RASHIDUL, Hassan y MUSTAFA, Jamil. Speaker identification using Mel frecuency cepstral coefficients. Dhaka: 3rd International Conference on Electrical & Computer Engineering, 2004.

TANG, Xiang. Hybrid Hidden Markov Model and Artificial Neural Network for Automatic Speech Recognition. Wuhan City: Communications and System Pacific-Asia Conference on Circuits, 2009.

TEBELSKIS, Joe. Speech recognition using neural networks. Tesis de Doctorado. Pittsburgh: Carnegie Mellon University, 1995.

THRASYVOULOU, T. y BENTON, S. Speech parameterization using the Mel scale Part II [online] 2003 [cited: Diciembre 2009]. Available from Internet: http://www-2.cs.cmu.edu/~mseltzer/sphinxman/

BIBLIOGRAFÍA COMPLEMENTARIA

CEPISCA, Costin, et al. About the efficiency of real time sequences FFT computing. Bucharest: Design and Diagnostic of Electronic Circuits and Systems, 2007.

GALLARDO, Ascensión. Reconocimiento de Habla robusto frente a condiciones de ruido auditivo y convolutivo. Tesis de Doctorado. Madrid: Universidad Politécnica de Madrid, 2002.

HEUNGSUK, Chin, et al. Realization of speech recognition using DSP. Pusan:IEEE International simposium in Industrial Electronics, 2001.

HUA-TAN, Zheng y LINDBERG, Borge. Automatic Speech Recognition on Mobile Devices and over Communication Networks. UK:Springer, 2008.

ICONTEC, Norma Técnica Colombiana 1486 (sexta actualización). Documentación. Presentación de tesis, trabajos de grado y otros trabajos de investigación.

PICONE, Joseph. Signal Modeling Techniques in Speech Recognition. En: Proceedings of the IEEE. Septiembre 1993. vol. 81, no 9.

PUERTAS TERA, Jose Ignacio. Robustez en reconocimiento fonético de voz para aplicaciones telefónicas. Tesis de Doctorado. Madrid: Universidad Politécnica de Madrid, 2000.