Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería Electrónica y Eléctrica
Escuela Profesional de Ingeniería Electrónica

Reconocedor y analizador de voz

TESIS

Para optar el Título Profesional de Ingeniero Electrónico

AUTORES

Anibal COTRINA ATENCIO

Fernando Raphael PERALTA REYES

ASESOR

Guillermo TEJADA MUÑOZ

Lima, Perú

2002

Referencia bibliográfica

Cotrina, A. & Peralta, F. (2002). *Reconocedor y analizador de voz*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería Electrónica y Eléctrica, Escuela Profesional de Ingeniería Electrónica]. Repositorio institucional Cybertesis UNMSM.

ANIBAL COTRINA ATENCIO
FERNANDO RAPHAEL PERALTA REYES

RECONOCEDOR Y ANALIZADOR DE VOZ

Tesis presentada a la facultad de Ingeniería Electrónica de la Universidad Nacional Mayor de San Marcos para obtener el Título Profesional de Ingeniero Electrónico.

Área: Sistemas Digitales
Asesor: Ing. Guillermo Tejada Muñoz.

Lima – Perú
2002

Dedico esta tesis a mi madre la única persona que siempre confió en mi.

Fernando Peralta Reyes

A Saúl y Priscila.

Aníbal Cotrina Atencio

AGRADECIMIENTOS

Agradecemos, a nuestro asesor, el Ing. Guillermo Tejada Muñoz, quien mostró una gran dedicación y una valiosa y estricta dirección, a quien le reconocemos gran parte del éxito en el desarrollo de nuestro trabajo. Al Ing. Bruno Vargas por su confianza, y constante apoyo a nuestra labor e interés en la investigación en la facultad. Al Ing. Ezequiel Zavala por brindar su apoyo a los alumnos deseosos de investigar. Al Ing. Victor Coronel quien desde el extranjero apoyo a nuestra facultad, donando bibliografía sobre Procesamiento Digital de Señales. A nuestros amigos del grupo de Procesamiento de Voz Jorge, Alex, David, Jossel y Edgard con quienes participamos en varios eventos dando a conocer nuestro trabajo y mostrando el nivel de nuestra facultad, también a Dennis quien siempre apoyó en la conformación de grupos de investigación. A nuestros amigos Irving, Pepe y Douglas, quienes nos brindaron su apoyo desinteresado, colaborando con equipos de computo durante el desarrollo de nuestra tesis. Al Instituto de Investigación de nuestra facultad que nos apoyó con material bibliográfico, equipos de computo y medición. Finalmente agradecemos a todos nuestros familiares, amigos y profesores, quienes participaron directa o indirectamente en el desarrollo de este trabajo.

Lista de Figuras

Capitulo 7

Lista de Tablas

Anexos

RESUMEN

El presente trabajo de Tesis consiste en la elaboración de un Sistema de Reconocimiento y Análisis de Voz de palabras aisladas, independiente del locutor y con un procesamiento *On Line*. La etapa de reconocimiento del sistema esta basado en el uso de Redes Neuronales Artificiales, que realiza la clasificación de las palabras por medio de sus patrones característicos que están conformados por los coeficientes Mel-Cepstrum.

Como resultado final del desarrollo del proyecto se ha diseñado un software denominado Reconocedor y Analizador de Voz (RAV), el cual, realiza el reconocimiento de las palabras y además, permite analizar gráficamente las principales características de la Señal de Voz. Este software constituye una herramienta importante para futuras investigaciones en el área del Reconocimiento de Voz.

INDICE GENERAL

1. DESCRIPCION DEL SISTEMA

1.1. *Introducción*

El presente trabajo de tesis consiste en el desarrollo de un software que permite realizar el análisis y el reconocimiento de la Señal de Voz (Ver figura 1.1), basándose en el uso de las Redes Neuronales Artificiales como clasificador de las palabras. Este software ha sido diseñado utilizando librerías dinámicas creadas en Visual C++, en la cuales se ejecutarán los algoritmos, y la interface gráfica ha sido desarrollada en Visual Basic.

Figura 1.1 Pantalla Principal del Reconocedor y Analizador de Voz

Un Sistema de Reconocimiento de Voz es una herramienta computacional capaz de procesar la Señal de Voz emitida por el ser humano y reconocer la información contenida en ésta, convirtiéndola en texto u órdenes que actúan sobre un proceso. En su desarrollo intervienen diversas disciplinas, tales como: la Fisiología, la Acústica, el Procesamiento de Señales, la Inteligencia Artificial y la Ciencia de la Computación.

En la figura 1.2 se muestran las principales etapas de un Sistema de Reconocimiento de Voz. La etapa de hardware está constituida por los dispositivos de y el procesador.

Las palabras pronunciadas son convertidas en señales eléctricas a través de un micrófono, luego mediante una tarjeta de sonido, esta señal es acondicionada y convertida a un formato digital para poder ser interpretada por el Procesador del Sistema.

La etapa de software, es un conjunto de algoritmos computacionales que tiene como función el procesamiento y reconocimiento de las palabras. En síntesis, la etapa de software posee tres fases: Detección de Extremos, Extracción de Patrones y Reconocimiento.

La Detección de Extremos, se encarga de detectar el inicio y final de la pronunciación de una palabra.

La Extracción de Patrones, extrae parámetros característicos de la Señal de Voz, los cuales permiten diferencian una palabra de otra.

El Reconocimiento, determina la correspondencia entre un patrón característico y una determinada palabra. Existen diversas técnicas con las cuales se puede efectuar esta última fase; tales como: la Distancia Euclidiana, el Alineamiento Temporal, los Modelos Ocultos de Markov y las Redes Neuronales Artificiales, entre otros.

El Reconocimiento de Voz puede simplificar el acceso a diversos sistemas, crear la posibilidad de que la población en general (incluyendo personas con discapacidad física) pueda usar las computadoras mediante el habla, para el manejo de transacciones, mensajes, información y controlar dispositivos de un proceso.

En el resto de este capítulo se describirán los principales objetivos del presente trabajo, así como los antecedentes y aportes del mismo. El Capítulo 2 trata sobre la teoría de la señal de Voz, centrándonos en su naturaleza, formas de onda y sus principales características. El Capítulo 3 presenta un marco teórico sobre los principios y elementos de un Sistema de Reconocimiento de Voz, en el cual se describirá la digitalización de la señal de voz, así como las principales técnicas de extracción de características y de clasificación. El capítulo 4 está dedicado a los parámetros característicos denominados Coeficientes Mel Cepstrum. El capítulo 5 muestra la teoría sobre las Redes Neuronales Artificiales. El capítulo 6 trata sobre el desarrollo del proyecto, muestra con detalle cada una de las etapas implementadas, así como todas las técnicas y algoritmos utilizados. En el Capítulo 7 trata sobre las pruebas y resultados experimentales. En el Capítulo 8 se muestran las conclusiones. En el Capitulo 9 se presentan las recomendaciones y perspectivas. En el Capítulo 10 se encuentran los anexos. En el capitulo 11 se muestra la bibliografía consultada. Finalmente se muestra el apéndice donde se encuentran detalles técnicos sobre la programación de la tarjeta de sonido.

1.2. Objetivos

- Elaborar un Sistema de Reconocimiento de Voz de palabras aisladas que sea independiente del tipo de voz del hablante, y que permita efectuar un procesamiento O*n Line*, ya sea para realizar una conversión Voz a Texto o interactuar con la computadora mediante comando de voz.
- Aplicar tecnologías de punta, tal como las Redes Neuronales Artificiales en el desarrollo de Sistema de Reconocimiento de Voz, cuyo conocimiento nos permita acercarnos a tecnologías vigentes de los países industrializados.
- Diseñar un Instrumento Virtual que permita visualizar y analizar las características de la Señal de Voz; que sirva como herramienta de desarrollo en estudios posteriores sobre Procesamiento de Voz, tanto en niveles de Pre y Post Grado.

1.3. Antecedentes

Actualmente, en el ámbito global se está dando gran importancia al estudio del Análisis y Reconocimiento de la Señal de Voz. Existen grandes empresas de software que como resultado de varias décadas de investigación vienen desarrollando sistemas de conversión Voz a Texto, como el caso de la IBM y Microsoft.

Se ha incrementado de manera significativa el desarrollo de trabajos de investigación en esta área, tanto en las Universidades como en los Institutos especializados alrededor del mundo, los cuales aportan continuamente nuevos conocimientos sobre el tema.

Existe mas información sobre este tema en artículos y en medios como Internet. Se puede tener acceso a paquetes de software que constituyen herramientas de Análisis y Reconocimiento de Voz (Shareware o Freeware), tales como el CLSU TOOLKIT, entre otros, que demuestran gran eficiencia, pero sin embargo son consideradas "Cajas Negras", ya que no ofrecen mayor información sobre las técnicas y algoritmos utilizados en su construcción.

Por esto, considerando la importancia del estudio del Análisis de la Señal de Voz y además el avance tecnológico vigente, se decidió efectuar un estudio mas profundo del tema, que tiene como objetivo final el diseño e implementación de un Sistema Reconocedor y Analizador de Voz, de tecnología de punta que pueda ser difundida entorno del área académica, también comercialmente e industrialmente en nuestro país.

1.4. Metodología

Inicialmente se investigó sobre el Estado del Arte de la Tecnología del Habla, para esto se consultó sitios especializados en Internet y artículos científico-técnicos recientes, lo cual nos permitió obtener una visión general de las últimas técnicas utilizadas en el Procesamiento y Reconocimiento de Voz. Seguidamente se procedió al estudio de libros de Procesamiento Digital de Señales para profundizar sobre aspectos teóricos de la matemática utilizada en este campo, luego de lo cual se inició el estudio de bibliografía más especializada sobre Procesamiento de Voz; también se inició el estudio de técnicas de Reconocimiento, tales como las Redes Neuronales Artificiales. Con todo el conocimiento adquirido como consecuencia de la revisión bibliográfica se empezó a modelar el sistema y diseñar cada una de sus etapas.

La primera fase del desarrollo del sistema, consistió en realizar un análisis *Off Line* de la Señal de Voz y sus características. La adquisición muestras de voz en esta fase se realizó por medio de la Grabadora de Sonidos de Windows y el análisis se realizó utilizando el programa Matlab v5.3, este programa fue fundamental para el desarrollo de esta fase, ya que además de permitir fácilmente analizar las muestras de voz, sirvió para desarrollar los algoritmos de procesamiento y reconocimiento de una manera sencilla y confiable.

En la siguiente fase se procedió a realizar la adquisición *On Line* de la Señal de Voz, para lo cual se elaboraron rutinas en MS Visual C++ que permiten el control de algunas funciones de la tarjeta de sonido Sound Blaster

que constituye el hardware que se realiza esta adquisición; asimismo se empezó a codificar en ANSI C los algoritmos necesarios para el procesamiento de la señal de voz.

Paralelamente se empezó a trabajar en el algoritmo de Reconocimiento del Sistema, que consiste en la implementación de una Red Neuronal Artificial. Dicho trabajo se dividió en dos etapas, en la primera se decidió modelar una red tipo Perceptron Multicapa con aprendizaje Backpropagation, y se comprobó su funcionamiento efectuando pruebas de Reconocimiento de Caracteres sobre pequeños mapas de bits, obteniendo el resultado esperado. Luego se ajustó el algoritmo para reconocer palabras en lugar de caracteres; pero este presentó problemas de aprendizaje, como consecuencia de ello se realizaron modificaciones para adaptarlo a las exigencias de nuestro sistema, lo que dio lugar a una nueva propuesta sobre la manera en que se debe realizar el entrenamiento de la red, la cual ha sido implementada mediante el algoritmo denominado Algoritmo EBHA (Entrenamiento por Bloques de Hablantes). Las pruebas de la Red Neuronal para reconocimiento de palabras se realizaron sobre una Computadora Pentium III de 750 MHz, facilitada por el Instituto de Investigación de la FIE. La codificación final del algoritmo se realizo usando el lenguaje MS Visual C++.

Finalmente, una vez probado que el sistema de reconocimiento *On Line* y cada una de sus etapas funcionaban correctamente, se procedió a desarrollar la Interface Gráfica del proyecto, la cual permite monitorear y analizar la Señal de Voz. Como se mencionó en los párrafos anteriores, los algoritmos de adquisición y procesamiento de la señal, se han implementado en Visual C++, pero específicamente en el standard ANSI C con el fin de aprovechar su portabilidad hacia otras plataformas. Las funciones que realizan estos procesos han sido "empaquetadas" en librerías de enlace dinámico (dll's), que son utilizadas por la Interface Gráfica. Esto permite fácilmente programar nuevas técnicas de análisis e integrarlas fácilmente a dicha Interface.

En lo que concierne a la eficiencia del Sistema de Reconocimiento, esta ha sido cuantificada mediante la Tasa de Acierto de Reconocimiento [Llamas, Cardeñoso. 1995], que es una cifra expresada en términos de porcentaje que indica la proporción entre las palabras reconocidas positivamente y el número total de palabras utilizadas en la evaluación. Además, se realizaron las pruebas necesarias para probar la eficiencia de nuestros algoritmos propuestos.

En la elaboración del sistema se han utilizado las siguiente herramientas:

- Computadora Pentium de 100Mhz, memoria de 32MB RAM, disco duro de 6 GB y tarjeta de sonido Sound Blaster 16.
- Computadora Pentium de 200Mhz, memoria de 32MB RAM, disco duro de 3.2 GB.
- Computadora Pentium III de 750 MHz, memoria de 64MB RAM, disco duro de 20GB.
- Borland C++ 3.0
- Matlab v5.3
- Microsoft Visual Basic 6.0
- Microsoft Visual C++ 6.0
- Microsoft Access
- 1 Micrófono Dinámico

1.5. Aportes

- Establecer en nuestra Universidad un primer paso en el estudio de la Señal de Voz y sus ventajas, con la finalidad de sentar las bases para el desarrollo investigaciones futuras relacionadas con el tema, tanto en Pre-Grado y Post-Grado; tales como el reconocimiento de habla continua, el reconocimiento de locutor y la síntesis de voz, entre otros.

- Entregar a la facultad una herramienta de Procesamiento y Reconocimiento de Voz desarrollado en un entorno visual que permitirá estudiar de manera mas detallada la Señal de Voz y sus aplicaciones.

- Generar artículos, publicaciones y ponencias que permiten difundir el nivel de conocimiento de nuestra facultad. Así se ha tenido oportunidad de exponer estos temas en congresos, tales como Intercon 2001 y Coneimera 2001 en donde se expuso el tema Reconocimiento de voz a través de la línea telefónica para ayuda a personas con discapacidad auditiva, o el concurso Inducontrol 2001, promovido por National Instruments, con el tema Telemando de procesos industriales, así como publicar en la revista Nº8 de nuestra facultad, el artículo Algoritmo COPER para la detección automática de Voz, quedando pendiente futuras publicaciones y presentaciones a nombre de la universidad y la facultad.

- Aplicar tecnologías de punta, como las Redes Neuronales Artificiales, las cuales, tomando como punto de partida nuestra experiencia, pueden ser implementados en otras áreas del Procesamiento Digital de Señales, tales como, el Reconocimiento de Imágenes, el Reconocimiento de Señales Ultrasónicas, Procesamiento y Reconocimiento de Señales Biológicas, entre otras.

- Crear tecnología nacional en el área de Procesamiento y Reconocimiento de Voz, dando conocer los algoritmos desarrollados y sus detalles de implementación, los cuales pueden ser optimizados y utilizados en otras aplicaciones.

- Proponer nuevas técnicas de Procesamiento y Reconocimiento, implementando un nuevo algoritmo de detección de extremos de una palabra, desarrollando una técnica propia para hacer la adquisición *On Line,* y además formulando una nuevo algoritmo de entrenamiento de la RNA.

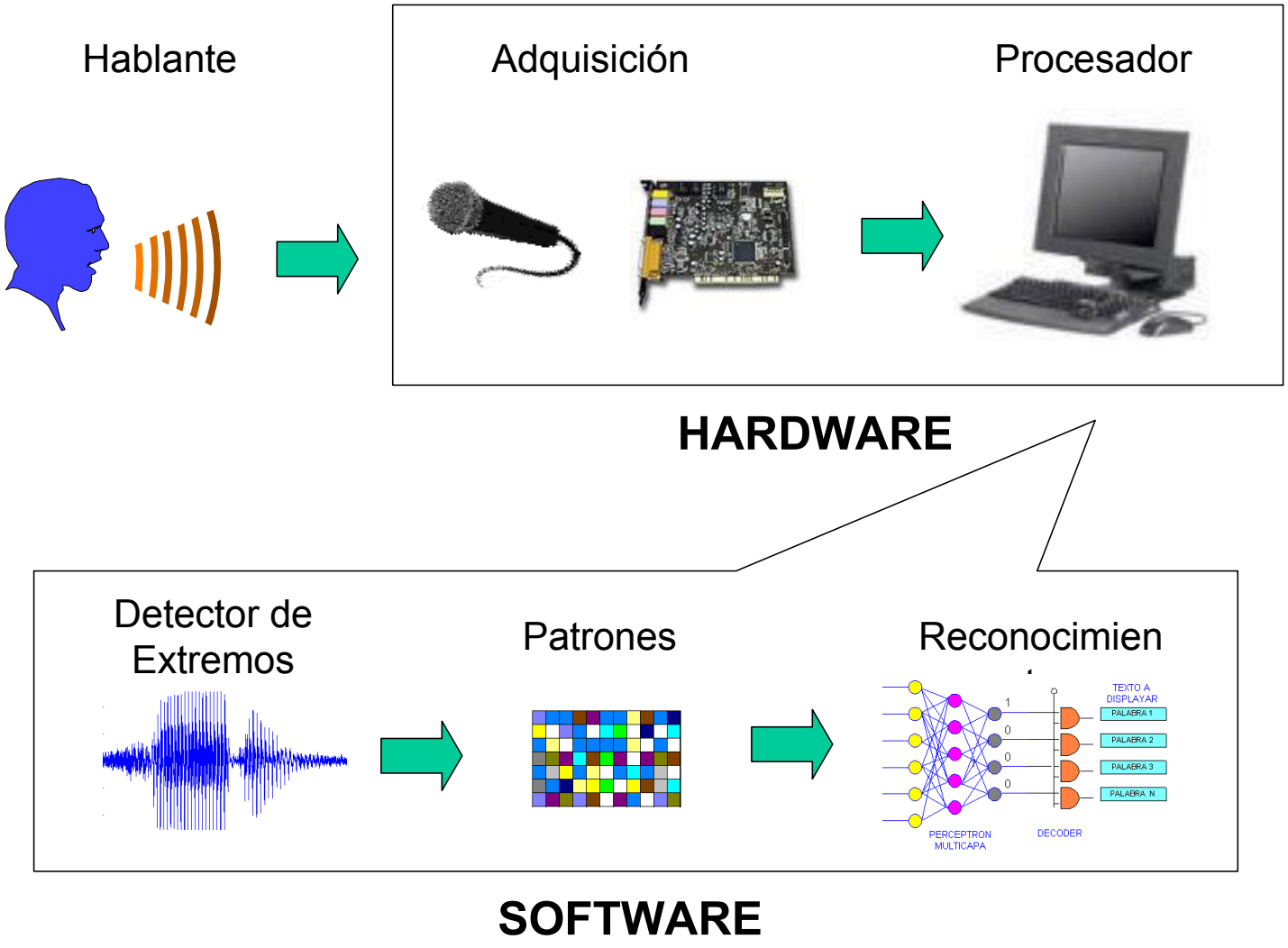


Figura 1.2 Etapas del Reconcimiento Sistema

2. FUNDAMENTOS DE LA SEÑAL DE VOZ

Las ondas sonoras son ondas mecánicas longitudinales, se originan por el movimiento de alguna porción de un medio elástico (sólido, liquido o gaseoso) con respecto a su posición de equilibrio, y debido a las propiedades elásticas del medio, esta perturbación puede desplazarse de un lugar a otro. Existe un gran margen de frecuencias entre las cuales se puede generar ondas mecánicas longitudinales. Las ondas sonoras se reducen a los límites de frecuencia que pueden estimular el oído humano para ser percibidas en el cerebro como una sensación acústica. Estos límites de frecuencia se extienden de aproximadamente 20 Hz a cerca 20 KHz y se llaman límites de audición. Las ondas audibles son producidas por cuerdas en vibración (por ejemplo el violín y las cuerdas vocales), por columnas de aire en vibración (el órgano y el clarinete) y por placas y membranas en vibración (el caso del tambor) [Resnick, Halliday. 1965].

Este capítulo trata sobre el estudio de las ondas sonoras generadas por el sistema fonador humano, las cuales se denominan señales de voz. En la primera parte se describe el proceso de generación de la Señal de Voz, seguidamente se describe sus propiedades en el dominio del tiempo y de la frecuencia, así como sus principales tipos. Finalmente se describe el modelo matemático del tracto vocal y los factores que afectan un Sistema de Procesamiento de Voz.

2.1. Descripción del Aparato Fonador Humano

El aparato fonador es el conjunto de órganos que tienen como función producir la voz humana, lo conforman los pulmones, los cuales producen un flujo de aire; la laringe, que contiene las cuerdas vocales, la faringe, las cavidades oral y nasal y una serie de elementos articulatorios como los labios, los dientes, el alvéolo, el paladar, el velo del paladar y la lengua. La figura 2.1 muestra el aparato fonador y cada una de sus partes.

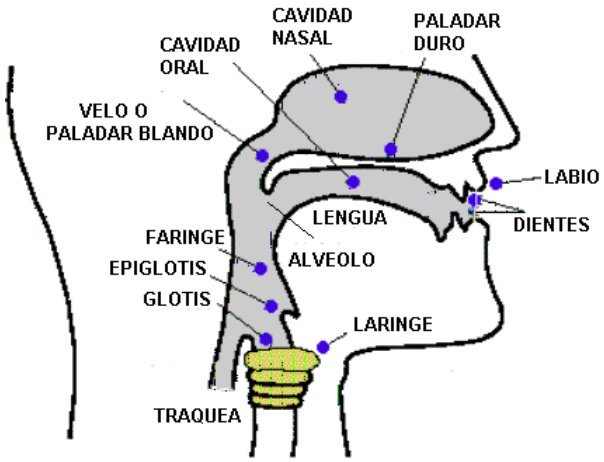


Figura 2.1 Corte esquemático del aparato fonador humano

En el proceso de generación de la voz, el sonido inicial proviene de la vibración de las cuerdas vocales conocida como vibración glotal, es decir, el efecto sonoro se genera por la rápida apertura y cierre de las cuerdas vocales conjuntamente con el flujo de aire emitido desde los pulmones. Las cuerdas vocales son dos membranas ubicadas dentro de la laringe, la abertura entre ambas cuerdas se denomina glotis. Cuando la glotis comienza a cerrarse, el aire proveniente desde los pulmones experimenta una turbulencia, emitiéndose un ruido de origen aerodinámico. Al cerrarse más las cuerdas vocales comienzan a vibrar a modo de lengüetas, produciéndose un sonido tonal, es decir periódico y cuya frecuencia varia en forma inversa al tamaño de las cuerdas. Este sonido es propio del hablante y es más agudo para el caso de mujeres y niños. Carece de información lingüística.

Luego de atravesar la glotis el sonido pasa a través de la cavidad supraglótica, que es la porción del aparato fonador que permite modificar el sonido dentro de márgenes muy amplios. Está conformado principalmente por tres cavidades, la cavidad oral, la cavidad labial y la cavidad nasal, correspondientes a la garganta, los labios y la nariz respectivamente. Estas cavidades constituyen resonadores acústicos, los cuales modifican los sonidos de acuerdo a la forma que adopten, la lengua y los labios permiten efectuar esta variación de manera voluntaria. En la figura 2.2 se muestra un modelo simplificado de estas tres cavidades.

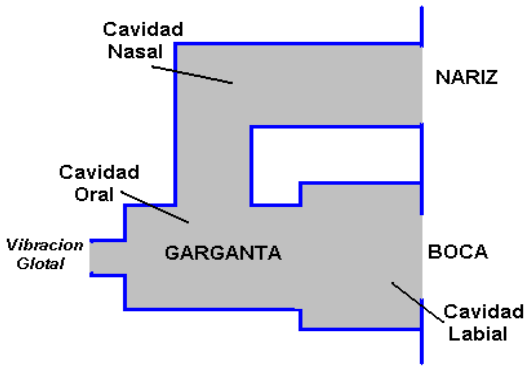


Figura 2.2 Modelo simplificado de las cavidades oral, labial y nasal

2.2. Características fundamentales de la Señal de Voz

2.2.1. Forma de onda de la Señal de Voz

La Señal de Voz está constituida por un conjunto de sonidos generados por el aparato fonador. Esta señal acústica puede ser transformada por un micrófono en una señal eléctrica. La señal de voz en el tiempo puede ser representada en un par de ejes cartesianos. Como todo los sonidos, está formado esencialmente por curvas elementales (senos y cosenos) pero las posibles combinaciones de éstas pueden ser complejas. A manera de ejemplo, en la figura 2.3 se muestra la forma de onda de la palabra 'explorador'. La representación de la Señal de Voz en función del tiempo es importante puesto que brinda información sobre características, tales como la Energía y los Cruces por Cero, las cuales facilitan su estudio y análisis.

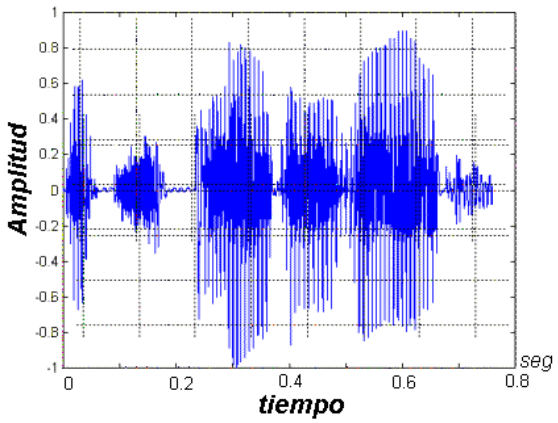


Figura 2.3 Forma de onda de la palabra 'Explorador'

2.2.2. Energía y Cruces por Cero

La Energía de una señal x(t) representa la energía disipada por una resistencia de 1 ohm cuando se le aplica un voltaje x(t). En una señal continua, la Energía total E en el intervalo de tiempo t_1 a t_2 esta definida como

$$E = \int_{t1}^{t2} |x(t)|^2 dt \qquad (2.1)$$

Para el caso de las señales discretas la Energía se define por

$$E = \sum_{m=0}^{N-1} x(m)^2 \qquad (2.2)$$

Donde:

N es el número de muestras de la señal.

La variación de Energía en la Señal de Voz se debe a la variación de la presión subglotal y de la forma del tracto vocal. La Energía es útil para distinguir segmentos sordos y sonoros en la Señal de Voz, debido a que los valores de esta característica aumentan en los sonidos sonoros respecto a los sordos.

Los Cruces por Cero indican el número de veces que una señal continua toma el valor de cero. Para las señales discretas, un cruce por cero ocurre cuando dos muestras consecutivas difieren de signo, o bien una muestra toma el valor de cero. Consecuentemente, las señales con mayor frecuencia presentan un mayor valor de esta característica, el ruido también genera un gran número de cruces por cero. La figura 2.4 ilustra esta idea para la señal continua y para la señal discreta.

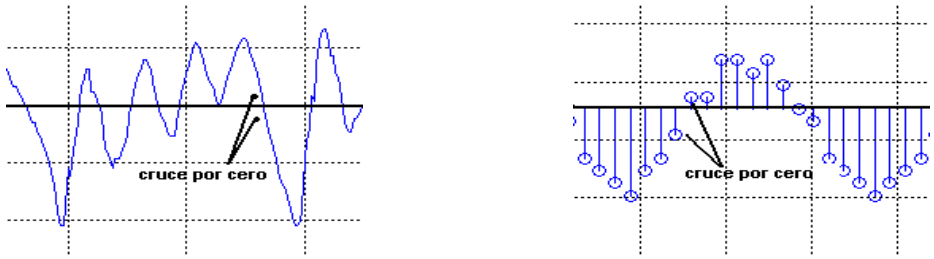


Figura 2.4 Cruces por Cero. a) Señal Continua. b)Señal Discreta

La formulación matemática de la Densidad de Cruces por Cero para señales discretas esta representa en la siguiente fórmula, en la cual, sign es la función signo y N es el número de muestras de la señal.

.

$$z = \sum_{m=0}^{N-1} \left| \text{sign}[x(m)] - \text{sign}[x(m-1)] \right| \qquad (2.3)$$

A continuación en la figura 2.5, se muestra las gráficas de Energía y Cruces por Cero de la palabra 'Seis'. Como se puede observar, el valor de la Energía varía en relación directa con la amplitud de la señal. La función de Densidad de Cruces por Cero alcanza sus valores más altos cuando se trata de sonidos tales como la 's', que son conocidos como sonidos fricativos. (Ver ítem 2.3.2)

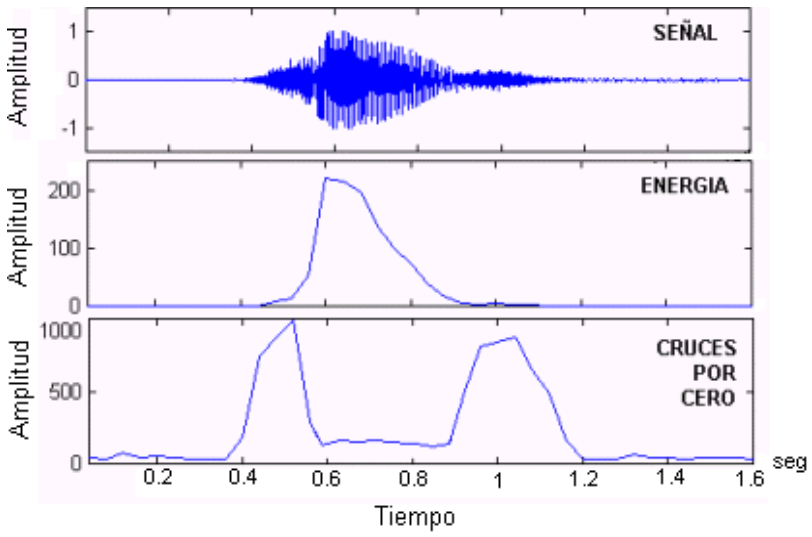


Figura 2.5 Energía y cruces por Cero de la palabra 'Seis'

2.2.3. Espectro de Frecuencia

Se realiza el estudio de la Señal de Voz en el dominio de la frecuencia, con la finalidad de conocer sus características espectrales. La figura 2.6 muestra el espectro de una señal de voz correspondiente a la palabra "Dos".

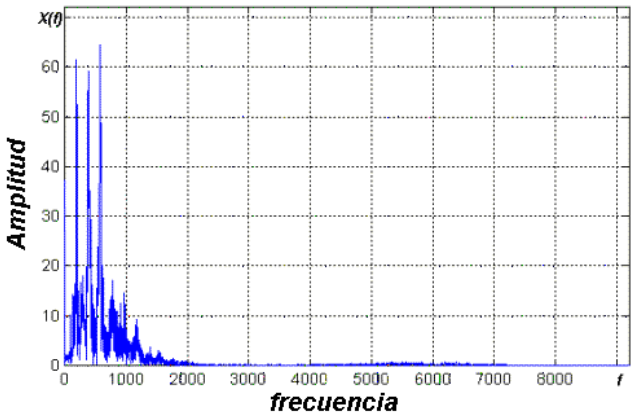


Figura 2.6 Espectro de frecuencia de la palabra 'dos'

La frecuencia fundamental o también denominada pitch, brinda información sobre la velocidad a la que vibran las cuerdas vocales al producir un sonido, el cual es generado por la rápida apertura y cierre de las cuerdas vocales con pequeños soplidos de aire, produciendo un espectro de frecuencia similar al mostrado en la figura 2.7. Este espectro podría ser obtenido si se colocara un micrófono de amplio rango directamente en la garganta, encima de las cuerdas vocales, pero debajo de las estructuras resonantes del tracto vocal.

El espectro está conformado de armónicos de periodo pitch, el cual es el rango fundamental de frecuencia producidas por las cuerdas vocales. Si bien el espectro lleva un gran componente cerca de la frecuencia pitch (aprox. 50 Hz), tiene gran cantidad de armónicos, y así tiene componentes de frecuencia que se extiende hasta pasado los 5 KHz. [Flores. 1993].

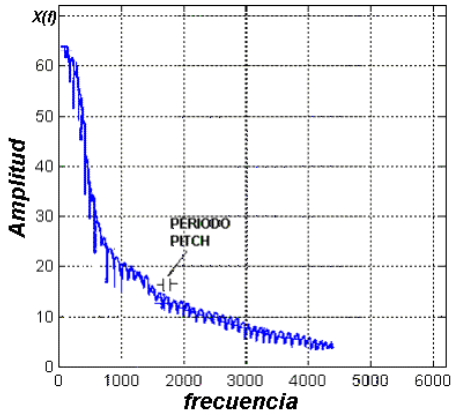


Figura 2.7 Frecuencia Fundamental

Otra característica importante es la envolvente espectral. Un análisis adecuado sobre esta característica permite obtener información sobre los diferentes tipos de sonido. La figura 2.8 muestra la envolvente espectral de la Señal de Voz, la cual es analizada con mayor detalle más adelante.

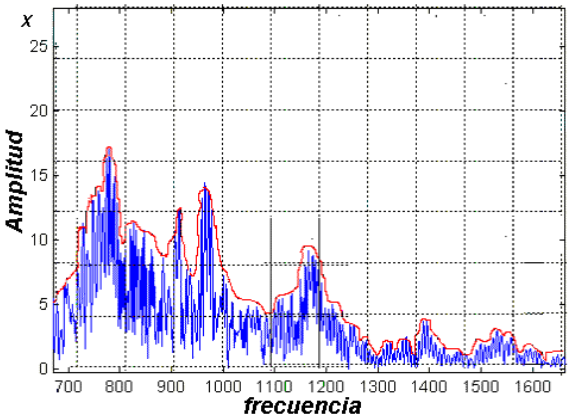


Figura 2.8 Envolvente Espectral

2.2.4. Frecuencias Formantes

Las cavidades que conforman la cavidad supraglótica actúan como resonadores acústicos. Si se realiza un análisis espectral del sonido luego de haber atravesado estas cavidades, el efecto de la resonancia produciría un énfasis en determinadas frecuencias del espectro obtenido, a las que se les denominara Formantes. Existen tantas Formantes como resonadores posee el tracto vocal. Sin embargo, se considera que sólo las tres primeras, asociadas a

la cavidad oral, bucal y nasal respectivamente proporcionan la suficiente cantidad de información para poder diferenciar los distintos tipos de sonido. En la figura 2.9, se muestra el espectro de la palabra 'Uno', y se denominan F1, F2 y F3 a sus tres principales frecuencias formantes. La amplificación de cada una de estas tres frecuencias depende del tamaño y forma que adopta la cavidad bucal y la cavidad oral, y si el aire pasa o no por la nariz.

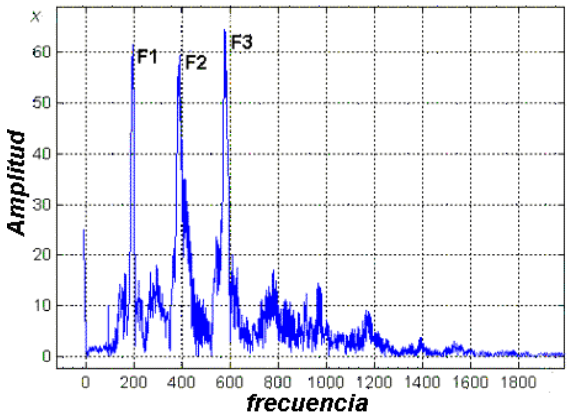


Figura 2.9 Frecuencias Formantes

2.3. Tipos de Señales de Voz

Básicamente, la Señal de Voz puede clasificarse en los siguientes tipos, Sonora, No Sonora y Plosiva [Flores. 1993].

2.3.1. Señal Sonora

La señal sonora se genera por la vibración de las cuerdas vocales manteniendo la glotis abierta, lo que permite que el aire fluya a través de élla. Estas señales se caracterizan por tener alta Energía y un contenido frecuencial en el rango de los 300 Hz a 4000 Hz presentando cierta periodicidad, es decir son de naturaleza cuasiperiódica. El tracto vocal actúa como una cavidad resonante reforzando la Energía en torno a determinadas frecuencias (formantes). En la figura 2.10 se muestra el comportamiento de este tipo de señales en el tiempo. Toda las vocales se caracterizan por ser sonoras pero existen consonantes que también lo son, tales como, la b, d y la m, entre otras.

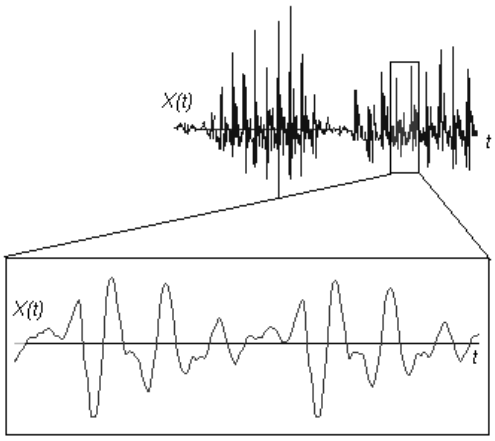


Figura 2.10 Señal Sonora

2.3.2. Señal No Sonora

A esta señal también se le conoce como señal fricativa o sorda, y se caracteriza por tener un comportamiento aleatorio en forma de ruido blanco. Tienen una alta densidad de Cruces por Cero y baja Energía comparadas con las señales de tipo sonora. Durante su producción no se genera vibración de las cuerdas vocales, ya que, el aire atraviesa un estrechamiento, y genera una turbulencia. Las consonantes que producen este tipo sonidos son la 's', la 'f' y la 'z' entre otras. La figura 2.11 muestra la forma de onda de una señal No Sonora

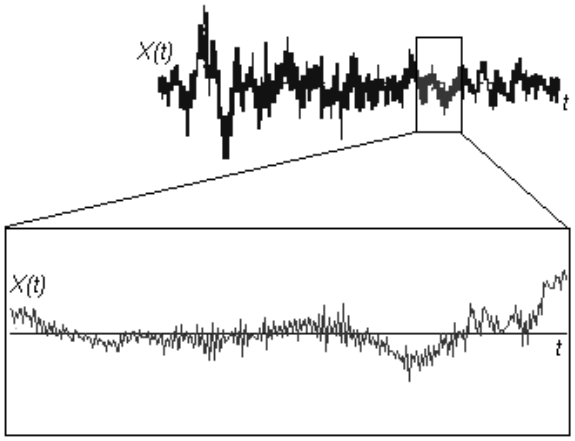


Figura 2.11 Señal No Sonora

2.3.3. La Señal Plosiva

Esta señal se genera cuando el tracto vocal se cierra en algún punto, lo que causa que el aire se acumule para después salir expulsado repentinamente (explosión). Se caracterizan por que la expulsión de aire está precedida de un silencio. Estos sonidos se generan por ejemplo, cuando se pronuncia la palabra 'campo'. La p es una consonante de carácter plosivo, y existe un silencio entre las sílabas 'cam' y 'po'. Otras consonantes que presentan esta característica son 't', y 'k', entre otras. La figura 2.12 muestra el comportamiento de este tipo de señal.

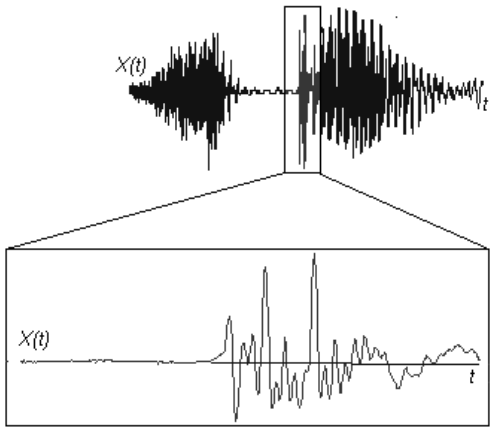


Figura 2.12 Señal Plosiva

2.4. Modelado del tracto vocal

El tracto vocal se comporta como un filtro, cuyos parámetros varían en el tiempo en función de la acción consciente que se realiza al pronunciar una palabra. En la figura 2.13 se muestra el diagrama de bloques del modelo del tracto vocal. Se consideran dos posibles entradas que dependerán del tipo de señal a reproducir, sonora o no sonora. Para señales sonoras, la excitación será un tren de impulsos de frecuencia controlada, mientras que para las señales no sonoras la excitación será ruido aleatorio. La combinación de estas señales modela el funcionamiento de la glotis. El espectro de frecuencias de la Señal de Voz puede obtenerse a partir del producto del espectro de la excitación por la respuesta en frecuencia del filtro.

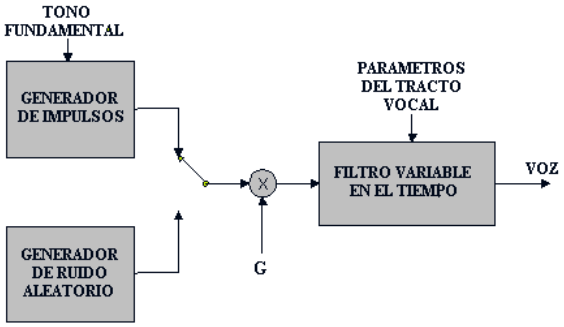


Figura 2.13 Modelo del tracto vocal

El control de ganancia G, determina la intensidad de la excitación. El tracto vocal manifiesta un número muy grande de resonancias, pero como se afirmó en el ítem 2.2.4, sólo se consideran tres y en algunos casos cuatro, esto es debido a que las resonancias de alta frecuencia son atenuadas por la característica frecuencial del tracto que tiende a actuar como un filtro pasabajo. Este modelo es una simplificación del proceso del habla. Los sonidos fricativos, no se filtran por el tracto con la misma extensión en que lo hacen las señales sonoras, por lo que el modelo no es muy preciso para este tipo de señales. Además, el modelo supone que las dos señales pueden separarse sin considerar ninguna interacción entre ellas, lo que no es del todo cierto, ya que la vibración de las cuerdas vocales es afectada por las ondas de presión dentro del tracto. Sin embargo, estas consideraciones pueden ser ignoradas, resultando el modelo lo suficientemente adecuado.

2.5. Factores que afectan a la Señal de Voz

Existen muchos factores que afectan la correcta percepción de las Señales de Voz, tales como el ruido, la acústica y la calidad del micrófono. El ruido, se define como aquellos sonidos aleatorios que de forma "oculta" transforman y enmascaran el sonido. Dado que, es poco probable encontrar un entorno de audio digital en perfecto silencio, es importante conocer la cantidad de ruido, en relación con la señal que se introduce en el equipo de sonido, especialmente en la tarjeta de sonido. La fuerza de cualquier sonido (hablar por

ejemplo), comparada con la fuerza promedio del ruido, se conoce como relación señal a ruido (SNR). A medida que aumenta la relación SNR, es mejor el trabajo realizado en grabación.

- Acústica de la habitación (ecos).- La acústica dentro de una habitación, puede crear cambios en el espectro de la Señal de Voz, debido a las resonancias de la habitación. Puesto que, cualquier ambiente cerrado tendría resonancias inherentes, su énfasis cuando interfiere con una señal de habla puede crear rangos anormales de frecuencias. Debido a esto, se producen dos cambios básicos en la acústica de una habitación, el primero es causado por el retardo en el tiempo del retorno de la señal original de una superficie reflectante, tal como una pared o una ventana. Cuando la onda es reflejada, regresa con mucho menor amplitud, y retardada en el tiempo, ésta interactúa con la forma de onda originalmente hablada para crear un nuevo espectro compuesto del habla. El segundo, está relacionado con la reflexión de una superficie rugosa de una pared, lo cual tiende a atenuar en altas frecuencias, pero a reforzar en el rango de bajas frecuencias. [Cater. 1984]

- Ruido del ambiente.- Si el usuario del sistema está operando el dispositivo en cualquier lugar que no sea una habitación tranquila, existe la posibilidad de la interferencia del ruido con las formas de onda. No obstante sin ruido externo, el sistema es susceptible de captar ruido a través del micrófono, y aunque suene extraño, muchas veces el ruido proviene desde la boca durante la pronunciación del mensaje.

En el caso de los sonidos plosivos, si el micrófono es ubicado directamente enfrente de la boca del hablante, entonces es muy susceptible de ser bombardeado por pequeñas ráfagas de aire ocasionadas por los sonidos plosivos. La mejor forma de tratar el problema es de rodear el micrófono con un material esponjoso transparente acústico, que rápidamente disipe la velocidad del viento de las pronunciaciones plosivas, permitiendo a las vibraciones acústicas normales pasar a través del micrófono.

Otras fuentes de ruido externo, tal como los ventiladores en las computadoras, aire acondicionados, teléfonos, y otras personas hablando puede también causar problemas con la exactitud del sistema de reconocimiento. Otra técnica para cancelar el ruido externo es filtrar la señal de audio antes procesarla. Debido a que las frecuencias de voz que contienen información relevante están dentro de un rango relativamente estrecho desde 200 a 3000 Hz, el espectro de audio puede ser filtrado a través de un filtro pasabanda para rechazar las señales acústicas fuera de ese rango de frecuencias.

- Calidad del Micrófono.- Probablemente, el factor que más influye en la adquisición electrónica de señales del habla es el tipo de micrófono que se esta usando. Existen, principalmente, cuatro tipos de micrófonos disponibles en el mercado, los cuales son el Electreto, el Dinámico, el de Cristal y el de Carbón.

Para percibir fácilmente las diferencias entre estos tipos de micrófonos, sus características principales son comparadas en la tabla 2.1.

Tabla 2.1 Comparación de los tipos de micrófonos

	Tipo de microfono			
Parámetro	Electreto	Dinámico	Cristal	Carbón
Respuesta en Frecuencia	Excelente	Excelente	Bien	Regular
Distorsión	Muy bajo	Muy bajo	Bajo	Alto
Cancelación de ruido	Excelente	Bien	Regular	Regular
Tamaño	Pequeño	Medio	Grande	Grande
Peso	Bajo	Medio	Bajo	Medio
Costo	Alto	Alto	Medio	Bajo
Nivel de Salida	Bajo (voltaje)	Medio (voltaje)	Alto (voltaje)	Alto (Resistencia)
Impedancia	Alto	Bajo	Alto	Bajo

Los dos parámetros más importantes en la lista, son las comparaciones de respuesta en frecuencia y la distorsión. Basados en estas comparaciones es recomendable el uso del Micrófono Dinámico y el Electreto [Cater. 1984].

3. RECONOCIMIENTO DE VOZ

El Reconocimiento de Voz, es el proceso por el cual un conjunto de algoritmos computacionales son capaces de traducir fielmente los sonidos de una unidad lingüística (palabra, sílaba o fonema) a un código simbólico que representa al mensaje. El sistema desarrollado en el presente trabajo, utiliza la palabra como unidad lingüística, lo que supone que el hablante pronuncia las palabras con pequeñas pausas entre ellas, las cuales son detectadas por el sistema. Es de aquí, que proviene el nombre de Sistema de Reconocimiento de Voz de palabras aisladas. Además se trata de un Sistema independiente del locutor, es decir, el Sistema puede reconocer voces de distintos locutores con gran eficiencia.

3.1. Elementos del Sistema de Reconocimiento de Voz

En la figura 3.1 se muestra los principales elementos de un sistema de Reconocimiento de Voz, los cuales serán descritos en los siguiente párrafos.

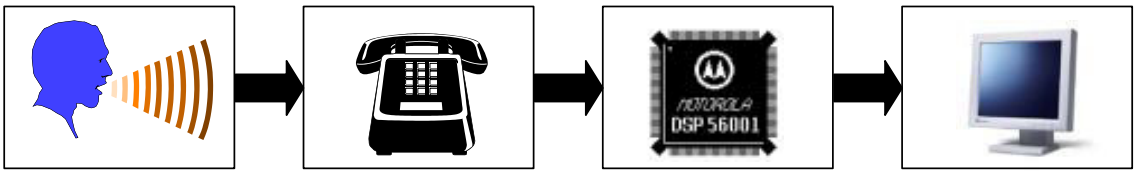


Figura 3.1 Elementos de un Sistema de Reconocimiento de Voz

- **Hablante o Locutor**: Es el individuo que emite el mensaje. Este es uno de los elementos que introduce mayor variabilidad en la forma de onda de entrada. Una persona no pronuncia siempre de la misma forma, debido a distintas situaciones físicas y psicológicas. Existe además gran variedad entre distintos locutores (hombres, mujeres, niños), diferencias según la edad o la región de origen (variabilidad de interlocutor). Es mucho más sencillo diseñar un sistema que efectúe el reconocimiento para un solo

locutor (se dice que el sistema es dependiente del locutor), a que un sistema funcione para cualquier locutor (sistema independiente del locutor).

- **Canal**: Es el medio físico apto para la transmisión de los sonidos de voz, y que pone en contacto el sistema fonador del locutor y el sistema de procesamiento. Por ejemplo el aire o la línea telefónica.

- **Procesamiento**: Es la etapa que se encarga de digitalizar la señal analógica del hablante a fin de extraer patrones característicos que representan a la unidad lingüística utilizada en el sistema, así como de realizar un proceso de clasificación para determinar los resultados.

- **Receptor**: Interpreta el resultado obtenido en el Procesamiento, y dependiendo de la aplicación, puede ejecutar un comando de control, un dato de entrada a una aplicación, o simplemente mostrar en pantalla el resultado de una conversión Voz a Texto, entre otras.

A continuación se describirá en detalle la etapa de procesamiento, la cual es el tema central del trabajo desarrollado.

3.2. Procesamiento

En la figura 3.2. se muestra las etapas en la cuales se divide el procesamiento. Seguidamente se describen cada una de estas etapas.

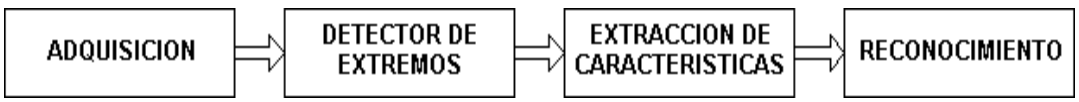


Figura 3.2 Etapas del Procesamiento de Voz

3.2.1. Adquisición de la Señal de Voz

El proceso consiste en convertir la señal analógica de la voz en una cadena de datos binarios, para lo cual se debe realizar un proceso de acondicionamiento y digitalización de la señal. La figura 3.3 muestra las etapas del proceso, la señal de voz es capturada a través de un micrófono que convierte las ondas acústicas del sonido en señales eléctricas, es decir, corriente o voltaje. Los niveles de voltaje obtenidos son amplificados a valores que puedan ser utilizados para su subsecuente digitalización. Inmediatamente un filtro pasa bajo define el ancho de banda y asegura un muestreo correcto. La salida del filtro pasa bajo es procesada por el circuito de muestreo y retención, el cual durante pequeños intervalos de tiempo mantiene el voltaje analógico presente, para que así el Convertidor Analógico-Digital (ADC) ejecute la cuantificación del nivel de voltaje retenido. El proceso de muestreo, retención y cuantificación es repetido sucesivamente hasta que la forma de onda sea completamente capturada.

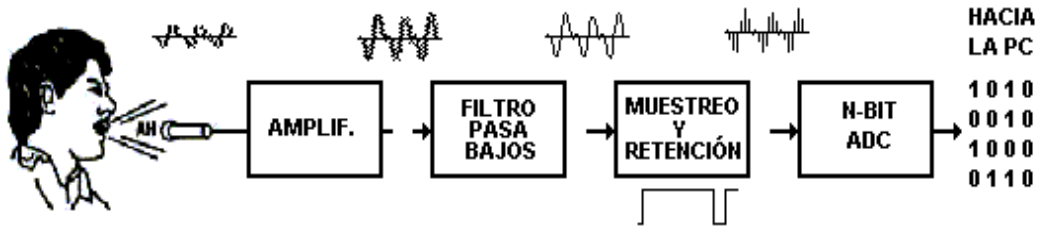


Figura 3.3 Digitalización de la Señal de Voz

El número de bits utilizados en la cuantificación afecta la calidad de la señal adquirida. Si asumimos una cuantificación uniforme[Shuzo, Kazuo. 1985], se obtiene que la relación de señal a ruido (SNR) en decibeles (dB) es igual a 1.76 + 6.02N, donde N es el número de bits utilizados.

Así cada bit adicional en el cuantificador contribuye en una mejora de aproximadamente 6 dB en el rango dinámico. El habla exhibe un rango dinámico de 50 a 60 dB y por esto una cuantificación de 8 o 9 bits debería proveer la necesaria relación señal a ruido para obtener una buena calidad. Pero en aplicaciones de Procesamiento de Voz comerciales, se debería efectuar una cuantificación de por lo menos 12 bits, esto debido a que diferentes hablantes producen niveles de amplitud muy fluctuantes, por lo que la SNR sufriría importantes variaciones.

En lo que respecta al filtro pasa bajo, se debe tener en cuenta que la necesidad de este filtro radica en que las frecuencias del habla pueden fácilmente extenderse más allá de la frecuencia de muestreo. En la mayoría de los sistemas, las frecuencias del habla por encima de 3 o 4 KHz son redundantes y proveen mucho menos información que las que están debajo de ese rango, mas aún si se quisiera realizar Sistemas de Reconocimiento a través de la línea telefónica, se debe considerar que las señales de voz transmitidas sobre la red telefónica están usualmente limitadas a un rango de frecuencias por debajo de los 3.3Khz. Así, el teorema de Nyquist indica que la tasa de muestreo debería ser mantenida a una frecuencia de por los menos 6.5 o 7KHz. Por esto es necesario utilizar un filtro pasa bajo, de tal manera que permita asegurar que el espectro del habla nunca sobrepase mas de un medio de la frecuencia de muestreo.

3.2.2. Detector Automático de Extremos o Detector de Actividad

Esta etapa es la encargada de realizar la detección automática de los instantes de comienzo y final de una pronunciación, debe ser capaz de distinguir la voz y el ruido de fondo. Un mal funcionamiento de esta etapa puede ocasionar:

-La pérdida de una trama de la señal de voz.
-La aceptación de sonidos no deseados que puedan ser confundidos con unidades lingüísticas.

De los dos tipos de errores mencionados, el primero es irreversible, por lo que el diseño de un Detector Automático de Extremos suele hacerse buscando que la pérdida de información sea menos frecuente, aún a costa de permitir la aceptación de sonidos no deseados. Motivo por el cual es importante la incorporación de procedimientos de rechazo de sonidos no deseados.

Usualmente, la Detección de Extremos se basa en el análisis de la evolución de dos propiedades de la Señal de Voz, estas son la Energía y los Cruces por Cero.

Para detectar el comienzo de una pronunciación se exige que la Energía o los Cruces por Cero superen ciertos umbrales durante un período de tiempo, y para la detección del final de la pronunciación los niveles de Energía y Cruces por Cero deben caer por debajo de éstos. Los niveles de umbral se obtienen experimentalmente analizando el contenido de Energía y Cruces por Cero que poseen tanto las palabras pronunciadas, como el ruido de fondo.

Básicamente, la Detección Automática de Extremos consiste en adquirir una trama de la Señal de Voz, e inmediatamente analizarla en función a la Energía y Cruces por Cero que contiene. El Proceso de Adquisición-Análisis se

repite permanentemente resultando en un sistema de barrido on-line. La figura 3.4 ilustra este proceso.

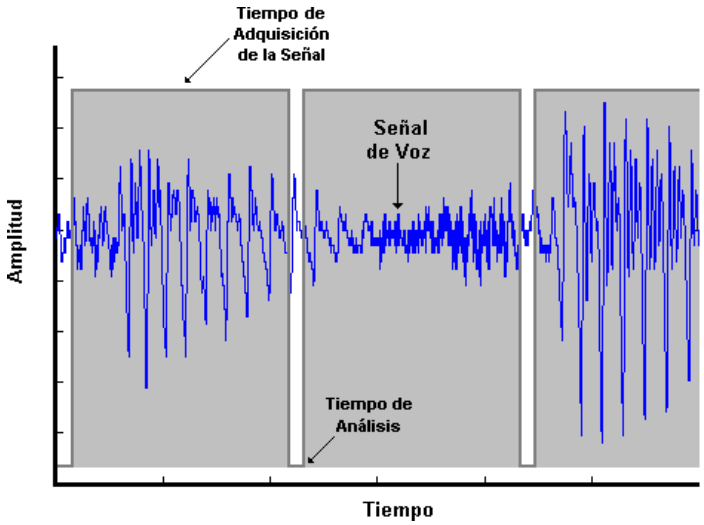


Figura 3.4 Detección Automática de Extremos

La duración del tiempo de adquisición está relacionado con el carácter cuasiestacionario que posee la señal de voz, así como de la precisión que se debe obtener en su delimitación, lo cual será descrito en el ítem 3.2.2.2. Respecto al tiempo de análisis, éste debe ser lo suficientemente pequeño, para evitar una pérdida significativa de información, lo cual se logra con un algoritmo eficiente y un procesador lo suficientemente veloz.

A continuación, se detallará el proceso de Detección Automática de Extremos. La figura 3.5 ilustra la obtención del inicio y final de una pronunciación. La figura 3.5(a) muestra la palabra a delimitar que en este caso es la palabra tres. En la figura 3.5(b) se muestra la evolución de la Energía por cada trama adquirida, y se aprecia que entre la muestra 5000 y 6000 existe un incremento significativo en la Energía, el cual, supera el nivel de umbral de inicio (E_{UI}), esto debido a que la palabra empieza con un sonido plosivo /t/. Si analizamos la figura 3.5(c) se observa que mientras la Energía ha sufrido un cambio importante, los cruces por cero aún mantienen valores por debajo del

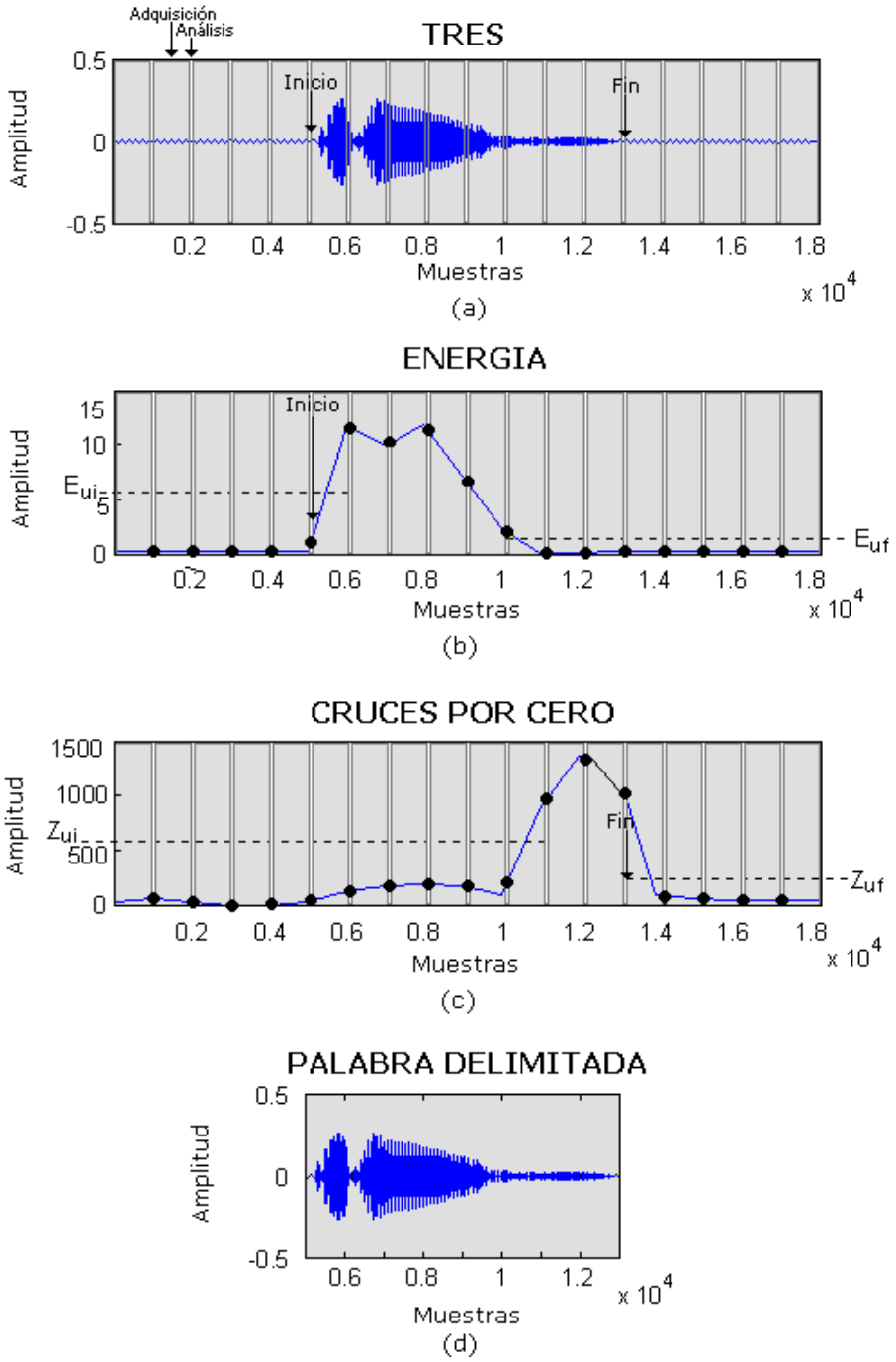


Figura 3.5 Proceso de Detección Automática de Extremos

umbral de inicio (Z_{UI}), pero en la detección de inicio de pronunciación es suficiente que se supere cualquiera de los dos umbrales, o bien E_{UI} o Z_{UI}, para que se establezca la posición de inicio, y en este caso la Energía superó el umbral E_{UI}, estableciéndose así, el inicio de la palabra en 5000.

Los umbrales de inicio de palabra deben tener valores relativamente altos para que sólo sean superados cuando realmente haya sido pronunciada una palabra, y para que no se produzcan un gran número de falsas alarmas ocasionadas por ruidos espurios. Sin embargo, si el nivel de umbral de inicio es demasiado alto puede ignorar tramas de algunas palabras, ocasionando pérdidas de información.

Los umbrales para el final de la pronunciación de una palabra se establecen cerca de los niveles de Energía y Cruces por cero que presenta el ruido de fondo, cuando tanto los niveles de Energía y Cruces por Cero de la señal están por debajo de los umbrales preestablecidos, se detecta el final de la pronunciación de la palabra. En figura 3.5(b) se puede ver que entre las muestras 11000 y 12000 la Energía cae por debajo del umbral (E_{UF}), mientras que si observamos la figura 3.5(c) apreciaremos como el nivel de cruces por cero se incrementa significativamente, debido a la presencia del sonido fricativo /s/, por lo que aún no se produce el fin de la palabra, ahora si analizamos entre las muestras 14000 y 15000 se observa que la densidad de cruces por cero ha caído por debajo del nivel de umbral (Z_{UF}), y que además la Energía también se encuentra por debajo de su umbral, cumpliéndose así, la condición para detección de final de la pronunciación de una palabra en la posición 14000. Finalmente en la figura 3.5(d) se muestra la palabra "tres" delimitada entre las muestras 5000 y 14000.

Esta técnica de Detección Automática de Extremos, ha venido siendo utilizada en muchos Sistemas de Reconocimiento, pero tiene la desventaja que sólo funciona razonablemente bien, cuando la relación señal a ruido es superior a 30 dB, pero fallan considerablemente cuando la voz se encuentra inmersa en un entorno ruidoso [Crespo, et. all. 2001].

En esta parte, se debe aclarar que tipo de ruido es el que afecta a la correcta delimitación de una pronunciación. Existen distintos tipos de fuentes de ruido que pueden interferir con la Señal de Voz, por ejemplo los provenientes de la red eléctrica, de motores, etc, los cuales poseen bajas componentes de frecuencia, y pueden ser eliminados fácilmente por un proceso de filtrado, pero existen otras fuentes de ruido, como el generado por las voces de otras personas que pueden encontrarse conversando cerca del Sistema de Reconocimiento, o el sonido producido por un equipo de música, o también por el timbre de una casa, los cuales son muy difíciles de evitar, debido a que son propios del entorno. Estos ruidos, son captados como un ruido aleatorio, de baja amplitud con componentes de frecuencia que están dentro del rango que el sistema analiza, por lo que resulta imposible realizar algún tipo de filtrado. Su amplitud relativamente pequeña hace que no ocasionen una gran distorsión a la señal, pero sus altas componentes de frecuencia, hacen que su densidad de cruces por cero aumente, lo cual ocasionaría que se confundan fácilmente con señales fricativas y serían detectados como tales, generando una delimitación incorrecta de la palabra pronunciada.

A continuación, en la Figura 3.6 se muestra la gráfica de la palabra "Cinco", la cual está afectada por un ruido de fondo, que ocasiona que se produzca una delimitación incorrecta.

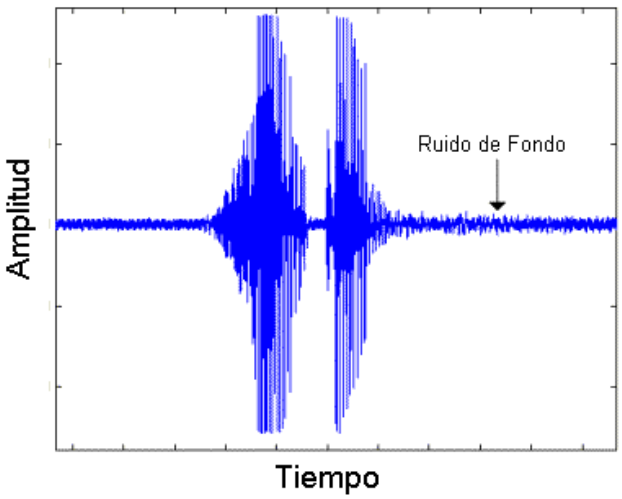


Figura 3.6 Palabra con Ruido de Fondo.

Para tener una mejor idea de que es lo que causa esta incorrecta delimitación, en la figura 3.7, se muestra como evolucionan la Energía y Cruces por Cero de la misma palabra.

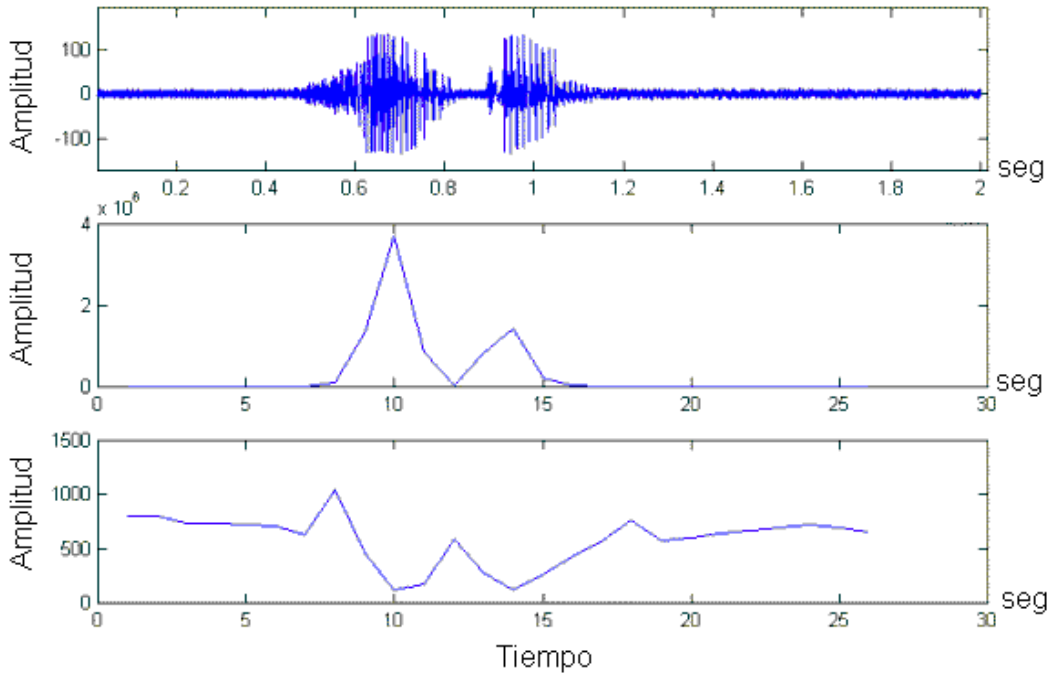


Figura 3.7 Gráfica de Energía y Cruces por Cero.

Se puede apreciar que la Energía sigue proporcionando información correcta sobre el inicio de los sonidos sonoros, mientras que los Cruces por Cero no pueden distinguir entre los sonidos fricativos y el ruido de fondo, la densidad de Cruces por Cero del ruido es mayor que el de la palabra, así que en este caso se haría una delimitación incorrecta, e inclusive es probable que detecte el inicio de pronunciación pero que nunca detecte el final de pronunciación, debido a que siempre se mantendría una alta densidad de Cruces por Cero.

A continuación se muestra la formulación matemática de los Cruces por Cero.

$$z = \sum_{m=0}^{L} \left| \text{sign}\,(y[m]) - \text{sign}\,(y[m-1]) \right| \qquad (3.1)$$

Esta fórmula analiza los cambios de signo de la señal y los va acumulando, y si el ruido de fondo es de alta frecuencia acumulará una gran cantidad de éstos. Por esto, se propone un nuevo algoritmo que no sólo relaciona los cambios de signo, sino que a la vez proporciona información sobre los cambios de Energía de la señal. Los autores del presente trabajo consideran muy importante el nuevo algoritmo propuesto, el cual no se ha encontrado en ninguna referencia bibliográfica consultada, por este motivo la expresión ha sido denominada con las siglas de nuestros apellidos Cotrina-Peralta (COPER), el cual será descrito a continuación.

3.2.2.1. El Algoritmo COPER (Cotrina-Peralta)

Básicamente este algoritmo es similar al de los Cruces por Cero, pero a las funciones signo, se les ha multiplicado por la Energía de la muestra analizada, es así que la densidad acumulada no sólo dependerá del cambio de signo de las muestras sino también de su amplitud, por esto, el ruido de fondo no logrará una gran acumulación de densidad, debido a que posee una pequeña amplitud comparada con las palabras pronunciadas.

Su formulación matemática se muestra a continuación:

$$\mathrm{COPER} = \sum_{m=0}^{L} \left| y[m] \cdot |y[m]| - y[m-1] \cdot |y[m-1]| \right| \qquad (3.2)$$

En la figura 3.8, se muestra la el análisis del parámetro COPER para la palabra CINCO en presencia de ruido de fondo, similar al caso anteriormente mostrado.

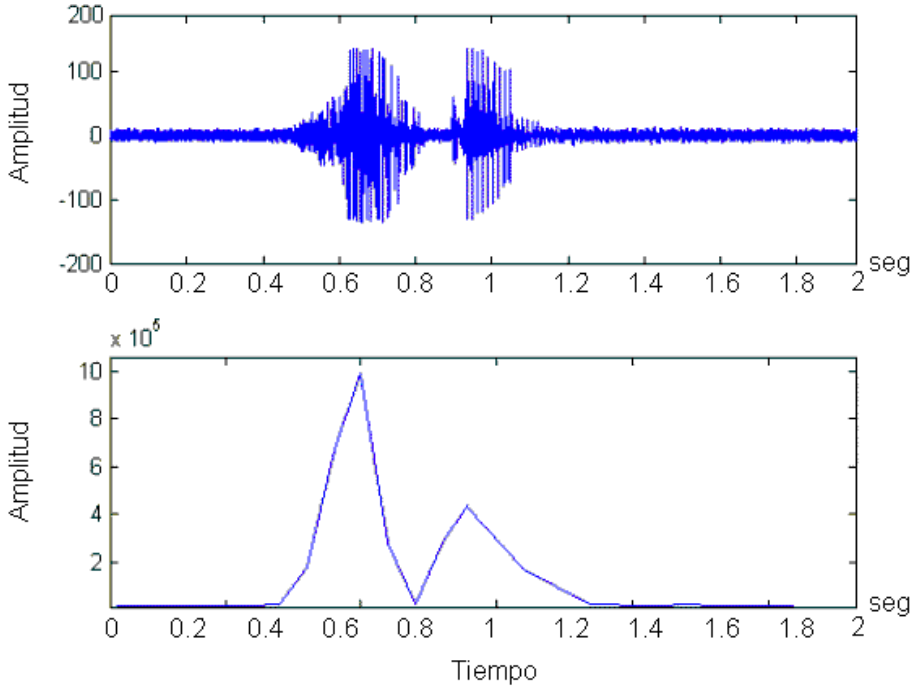


Figura 3.8 Evolución del parámetro COPER.

En esta figura se aprecia, que sólo existe una gran acumulación del parámetro COPER entre los instantes que dura la pronunciación, y que el ruido no contiene información significativa. Si se analiza con detenimiento se puede notar como el parámetro COPER va siguiendo con detalle los cambios temporales que se llevan a cabo durante la pronunciación de la palabra. Así el algoritmo COPER proporciona la suficiente información para ser utilizado como único parámetro en la Detección de Extremos.

Debido a que el algoritmo COPER es un nuevo algoritmo propuesto, en el Capítulo 7 se muestran las pruebas experimentales necesarias que permiten afirmar su validez.

3.2.2.2. Tiempo de Adquisición

Para establecer la duración del tiempo de adquisición de la señal, algunos autores, recomiendan adquirir una trama lo suficientemente grande, para así evitar adquirir sólo el silencio intermedio que se produce en algunas palabras, generado por la pronunciación de fonemas plosivos, tal como se

aprecia en la figura 3.9, en donde se muestra la gráfica de la palabra “Siete”, que posee un silencio intermedio entre la 'e' y 't', de aproximadamente de 1730 muestras o 157 ms (para una frecuencia de muestreo de 11KHz), el cual puede ser confundido como fin de pronunciación, produciéndose así una delimitación incorrecta.

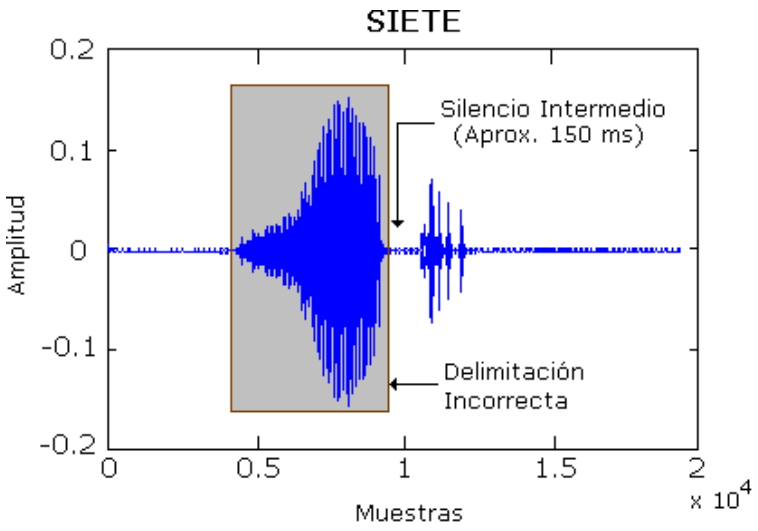


Figura 3.9 Silencio Intermedio.

Teniendo como referencia la palabra "Siete" se podría establecer un tiempo de adquisición de 180 ms (> 157 ms), que equivale a adquirir aproximadamente 2000 muestras por trama. Este método es utilizado, sobretodo, cuando se trabaja *Off-Line*, y los datos son obtenidos de ficheros previamente grabados, ya que es necesario traslapar las tramas de análisis[Flores. 1993]. Pero si este método quiere ser utilizado para hacer una implementación *On-Line*, la delimitación no será del todo eficiente, debido al gran tamaño de la trama de análisis, ya que se podría adicionar más información de la necesaria, o en otros casos la información podría ser recortada, tal como se muestra en la figura 3.10. Se aprecia que luego de delimitar la palabra, el inicio ha sido recortado, y en el final se ha captado ruido de fondo adicional.

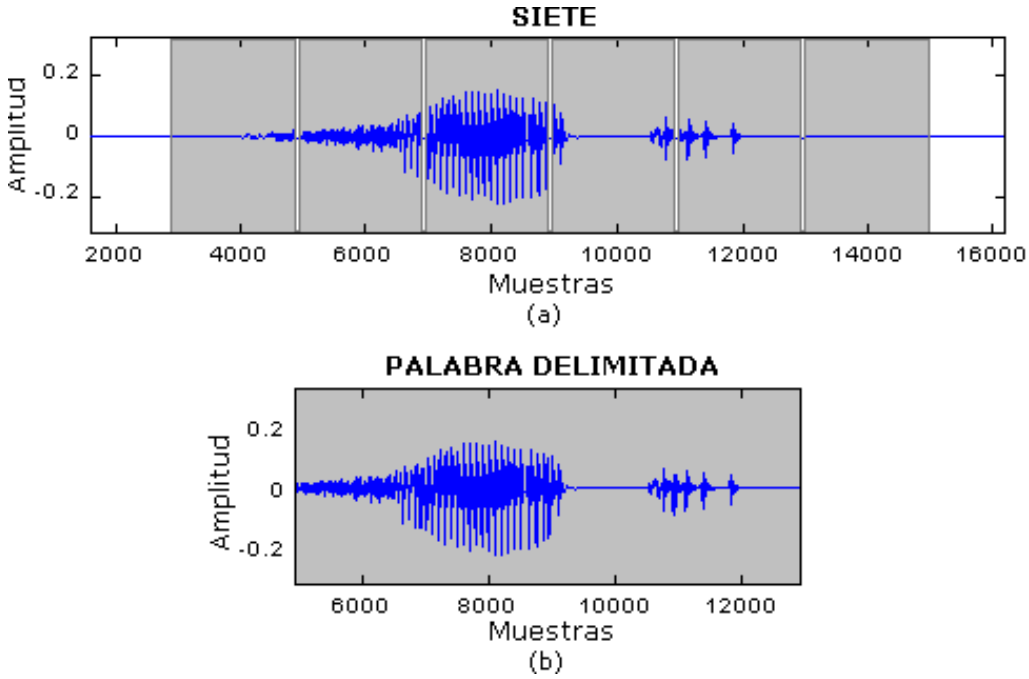


Figura 3.10 Delimitación clásica.

Por esto, debido a que uno de los objetivos del presente trabajo es realizar un reconocimiento *On-Line*, los autores del presente trabajo han desarrollado un método propio, para realizar la delimitación de la palabra, al cual se le denomina el método NVENT. En el siguiente ítem se detallará este proceso.

3.2.2.3. Método NVENT

En este método, se toma un tiempo de adquisición lo suficientemente pequeño, para así, permitir una mejor delimitación de la palabra. Como se sabe, la señal de voz posee una variación lenta en el tiempo, lo que permite dividir su análisis en tramas de duración relativamente cortas (entre 5 y 100 mseg). [Minh, 2001]

Tal como se vio en el ítem anterior, con este tiempo más corto, es probable que durante una trama de adquisición se capte sólo el silencio intermedio de algunas palabras pronunciadas, sin embargo, esto no será interpretado como el final de una pronunciación hasta que un número

preestablecido de tramas consecutivas nVent cumpla con las condiciones de fin de pronunciación (Ver figura 3.11a). El valor de nVent se establece según cuantas tramas consecutivas estén contenidas dentro de un silencio intermedio. La figura 3.11 muestra la delimitación de una palabra pronunciada, utilizando esta técnica.

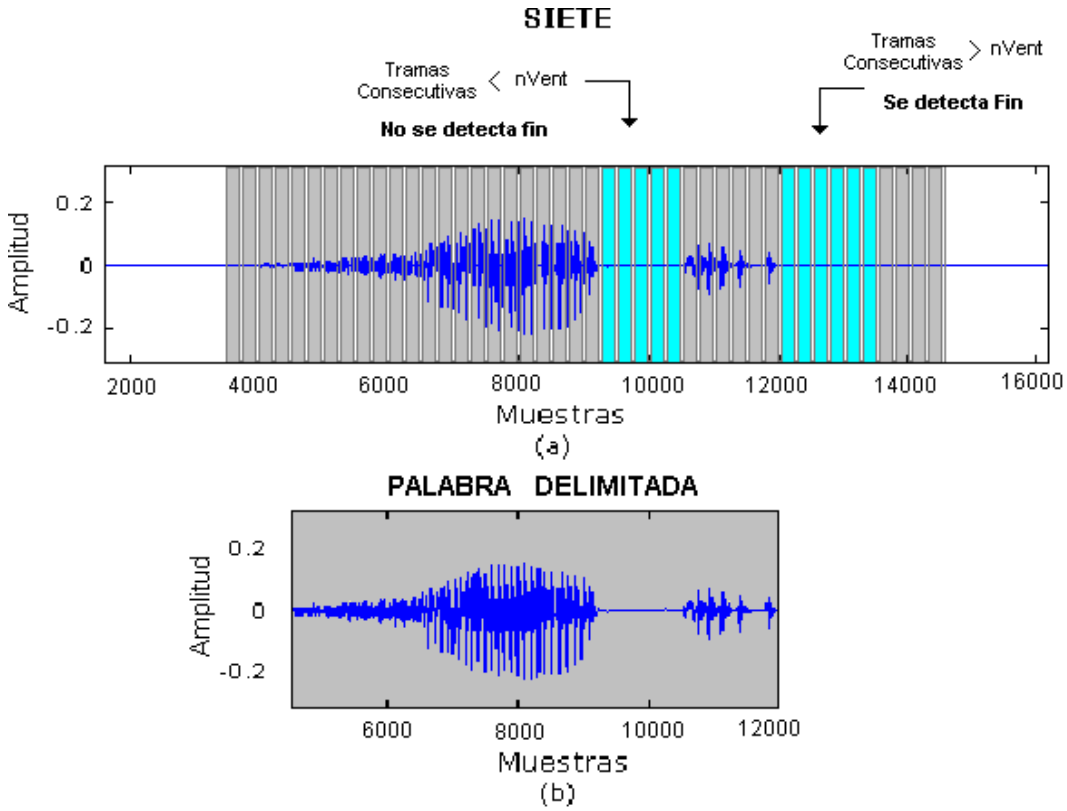


Figura 3.11 Método NVENT.

Comparando las figura 3.10b y 3.11b, se puede apreciar que con el método NVENT, se ha realizado una delimitación más exacta de la palabra pronunciada.

3.2.3. Extracción de Características

La extracción de características de la Señal de Voz consiste en la conversión de las muestras de voz a un conjunto de vectores, denominados patrones característicos, que representan eficientemente la información acústica de la señal de voz y que permiten reducir el número de datos por clasificar.

Los patrones característicos que vienen dando mejores resultados, son los relacionados con el cálculo de la envolvente espectral de la Señal de Voz calculada en tramas de 10 a 20 ms. de duración. [Hernández, et. al. 2001]

Dentro de los posibles patrones que permiten estimar la envolvente espectral se tiene los coeficientes obtenidos mediante un análisis de Predicción Lineal (LPC), que predice un valor de la forma de onda del habla a través de la sumatoria de las p muestras pasadas, cada una de las cuales es multiplicada por una constante α (promedio ponderado de las p muestras pasadas).

El análisis de Predicción Lineal [Boaz. 1997] se basa en obtener los coeficientes:

$\{\alpha_i\}$, i=1,2, ... , p, utilizando el criterio del error cuadrático promedio mínimo.

El valor predicho de y_n es denotado por $\hat{y}_n$; esto es

$$\hat{y}_n \approx \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + \cdots + \alpha_P y_{n-p} \qquad (3.3)$$

Otra técnica utilizada para obtener patrones característicos, es la de obtener los denominados coeficientes Cepstrales, los cuales han venido mostrando mejores resultados y son utilizados extensivamente en Sistemas del Reconocimiento de Voz. El Cepstrum se define como la transformada inversa

del logaritmo del módulo de la transformada de la señal, y proporciona información sobre las variaciones espectrales de la Señal de Voz.

$$c(t) = F^{-1}\left[\log|X(w)|\right] \qquad \text{,donde} \quad \mathrm{X(w)} = \mathrm{F[x(t)]} \tag{3.4}$$

En los últimos años, han aparecido innovaciones en la obtención de patrones de la Señal de Voz, una de éstas es la utilización de transformaciones que proporcionan una representación de la envolvente espectral, realizadas tratando de emular el tipo de procesamiento que realiza el sistema auditivo humano[Hernández, et. al. 2001]. Es decir, así como en acústica es habitual trabajar en bandas de octavas, que reflejan el hecho de que el oído humano presenta una mayor resolución espectral en bajas que a altas frecuencias, en reconocimiento se recurre a transformaciones que representen la envolvente espectral según un esparcimiento o resolución semejante a la del oído humano. Una de las transformaciones más utilizadas es la conocida como transformación con la escala Mel, que da paso a definir y utilizar los coeficientes denominados Mel-Cepstrum (MFCC), los cuales debido a su importancia serán tratados en el capítulo 4.

3.2.4. Reconocimiento de Patrones

El objetivo del Reconocimiento de Patrones es de clasificar los patrones característicos de la Señal de Voz, utilizando otros como referencia.

Entre las principales técnicas de reconocimiento tenemos:

-Comparación de Patrones.
-Modelos Automáticos Paramétricos.

3.2.4.1. Comparación de Patrones

Es una técnica que ha sido muy utilizada en los reconocedores de Voz tradicionales. A continuación en la figura 3.12, se muestra el diagrama de bloques de esta técnica.

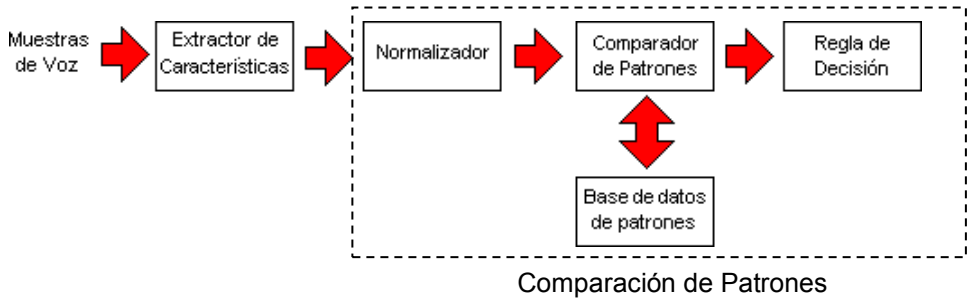


Figura 3.12 Reconocimiento del habla utilizando comparación de patrones.

Los comparadores de patrones basan su funcionamiento en el establecimiento de una distancia matemática entre vectores, de tal manera que se puede calcular lo cercano que se encuentra cada patrón proveniente de las muestras de voz de entrada con todos los patrones existentes en una base de datos. Como ejemplo de esta idea, en la Figura 3.13. se muestra un gráfico ilustrativo:

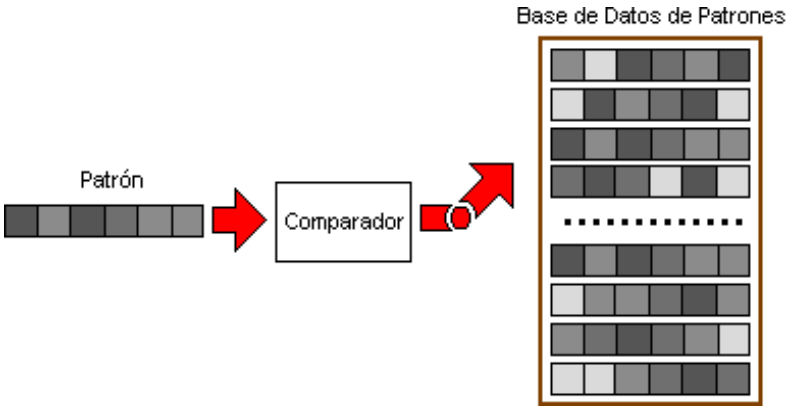


Figura 3.13 Funcionamiento general de un comparador de patrones.

La base de datos de patrones se obtiene seleccionando unidades lingüísticas del idioma que se pretenden reconocer. Seguidamente, se realizan grabaciones de estos sonidos y se obtienen sus características espectrales (parámetros LPC, MFCC u otros)

Antes de comparar cada patrón con la base de datos, resulta adecuado realizar un proceso de normalización con el fin de asegurar en la medida de lo posible la coincidencia en el tiempo de los patrones.

El proceso de normalización es necesario para ajustar los tamaños temporales de las unidades lingüísticas. Tomando la palabra como unidad lingüística, se encuentra que la duración de una misma palabra puede variar según sean los hablantes, contextos, estados de ánimo, etc. Esta situación lleva a obtener patrones de distintas longitudes, los cuales no podrían ser comparados. Se debe conseguir que los patrones de entrada tengan la misma longitud que los almacenados en la base de datos. El proceso de normalización se ilustra en la Figura3.14.

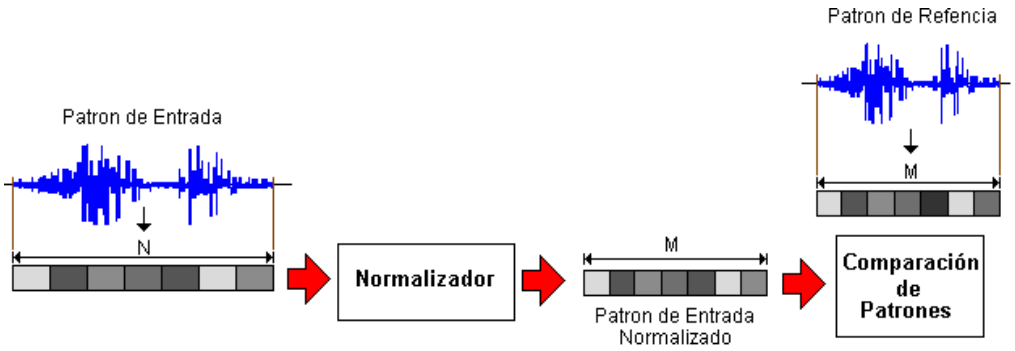


Figura 3.14 Normalización.

Las técnicas de reconocimiento por comparación de patrones, aunque forman parte de la materia básica en el área del tratamiento de la señal, han sido mayoritariamente sustituidas por los modelos de reconocimiento automático paramétricos, entre los que se encuentran los algoritmos genéticos, las cadenas de Markov y las Redes Neuronales.

3.2.4.2. Modelos Automáticos Paramétricos

Estos modelos, se caracterizan por la manera automática en la que se extraen las características espectrales de los sonidos y la forma en que las representan en su estructura interna.

En primer lugar se realiza una selección de los sonidos representativos del habla y luego, estos se organizan en grupos, conformando una base de datos de patrones, los cuales se ingresan a los algoritmos, que mediante un procedimiento previo, denominado aprendizaje, hacen que el conocimiento de las características espectrales correspondientes a los sonidos recolectados, quede representada en las estructuras internas de los modelos. Entre estos modelos se tienen a las Redes Neuronales, a los Modelos Ocultos de Markov y a los Algoritmos Genéticos entre otros, los cuales representan su estructura interna mediante neuronas, estados internos y estructuras de datos, respectivamente.

El proceso de reconocimiento a través de estos modelos consta de dos fases, la fase de aprendizaje y la fase de reconocimiento propiamente dicha. La primera de estas consiste en entrenar de manera supervisada al modelo con todo los patrones recolectados, con el fin de modificar su estructura y representar internamente las características de cada patrón entrenado. El objetivo del modelo es la clasificación de los patrones.

Este proceso se muestra en la Figura 3.15, en la cual los sonidos de la base de datos pasan por una etapa de Extracción de Características Espectrales antes de formar parte del entrenamiento del modelo paramétrico.

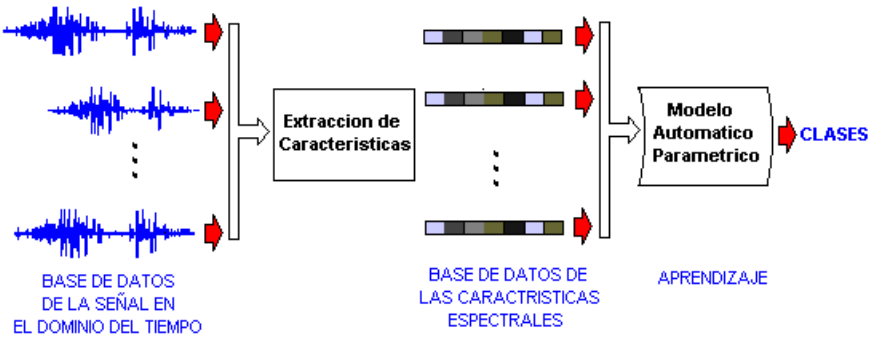


Figura 3.15 Etapa de Aprendizaje.

La segunda fase del proceso, consiste en utilizar el resultado de la fase anterior, en el reconocimiento de patrones que no estaban en la base de datos, pero que se encuentran dentro de las clases establecidas durante el aprendizaje. El modelo fundamenta su aprendizaje con el reajuste de sus parámetros internos. Este proceso se ilustra en la Figura 3.16: La señal en el tiempo, que se pretende reconocer, pasa por la misma etapa de Extracción de Características utilizada en el proceso de entrenamiento, antes de ingresar al modelo paramétrico, el cual determinara la clase a la que pertenece dicho sonido. Es importante remarcar que las condiciones de entrenamiento deben ser las mismas usadas durante el proceso de reconocimiento.

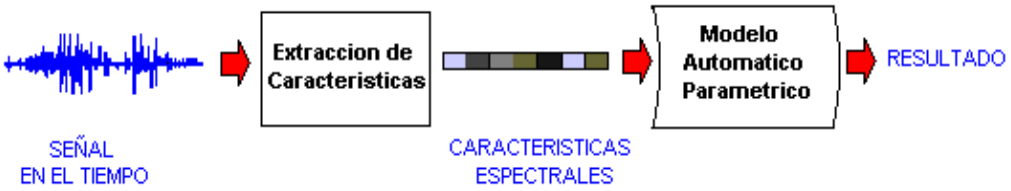


Figura 3.16 Etapa de Reconocimiento.

Una forma de explicar estas técnicas de reconocimiento, es por medio de la representación de los patrones en un espacio n-dimensional. Donde n es el número de elementos del vector que representa cada patrón

La figura 3.17a, muestra el proceso de reconocimiento a través de la Comparación de Patrones, en el cual se muestran las distancias a comparar en el espacio n-dimensional entre el patrón a reconocer (Px) con cada uno de los

dos patrones de la base de datos (P1 y P2). La distancia más corta es la que determina que patrón se trata.

Con respecto a la técnica de reconocimiento por medio de Modelos Automáticos Paramétricos, en la etapa de entrenamiento se establece una separación del espacio n-dimensional de a cuerdo a la disposición de los patrones de entrenamiento, la clase1 y la clase2, y durante la etapa de reconocimiento, el modelo se encarga de resolver a que clase pertenece un patrón que se quiere reconocer, de acuerdo a su ubicación en el espacio.

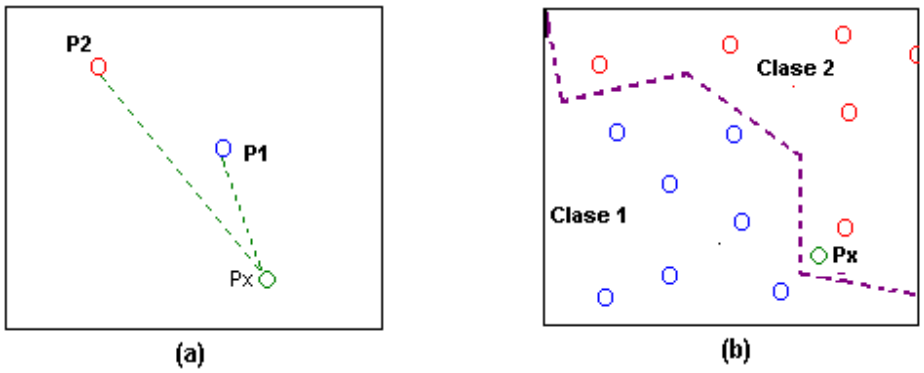


Figura 3.17 Representación en el espacio n-dimensional

4. COEFICIENTES MEL CEPSTRUM

En el ítem 3.2.3 se mencionó sobre los diversos tipos de patrones característicos que puede ser extraídos de la Señal de Voz. En este Capítulo se describirá una de las técnicas más utilizadas en los últimos años para extraer patrones característicos, la cual consiste en obtener los denominados Mel Frecuency Cepstrum Coefficients (MFCC) o Coeficientes Mel-Cepstrum. Al aplicar esta técnica, se trasforma las muestras de la señal de voz a un conjunto de coeficientes que representan eficientemente las propiedades espectrales y concentraciones de Energía de la Señal de Voz, tratando de emular el tipo de procesamiento que realiza nuestro sistema auditivo. Al tener en cuenta las características del oído, se trata de asemejar el sistema al reconocimiento hecho por una persona. Este análisis se basa en el uso de la escala de frecuencia Mel, la cual es un espaciamiento lineal de la frecuencia por debajo de los 1000Hz y un espaciamiento logaritmo por arriba de los 1000Hz.

El diagrama de bloques para obtener los MFCC es mostrado en la Figura 4.1.

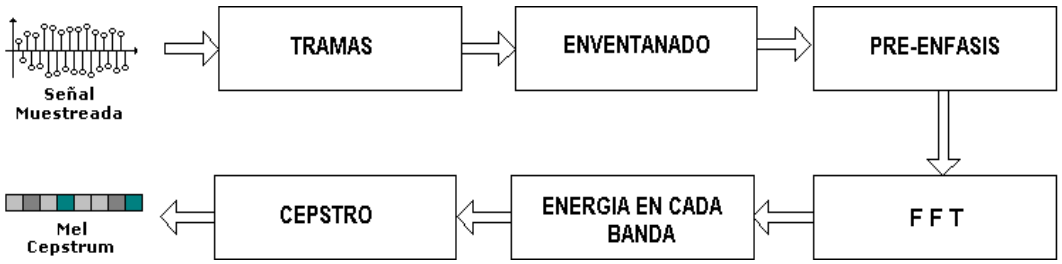


Figura 4.1 Diagrama de Bloques para obtener los MFCC

4.1. Secuencia de Tramas

Se considera que la señal del habla es cuasi-estacionaria (variación lenta en el tiempo), lo que permite dividir su análisis en tramas de duración corta. Estudios previos, han demostrado que los mejores resultados son obtenidos analizando tramas de 10 a 20 ms de duración, ya que es ahí, donde el análisis espectral muestra información distintiva entre los diferentes tipos de sonidos.

La señal de voz es segmentada en tramas de N muestras(20ms), con un paso de M muestras (M<N), tal como se representa en la figura 4.2. Para nuestro caso, al utilizar una frecuencia de muestreo de 11 KHz, se obtiene que el tamaño de N es de 220 muestras, y M, de 110 muestras. Pero debido a que la Transformada Rápida de Fourier utilizada en una etapa posterior, exige que el número de elementos de la trama sea potencia de 2, se establece el valor de N a 256 y M a 128 muestras, lo que equivale a 23.2 y 11.6 ms respectivamente.

La primera trama consiste de las N primeras muestras. La segunda empieza M muestras después que la primera, y la traslapa en N-M muestras. Similarmente, la tercera trama empieza 2M muestras después de la primera trama (o M muestras luego de la segunda trama) y la traslapa en N-2M muestras. Este proceso continúa hasta que se hace un barrido total de la palabra adquirida.

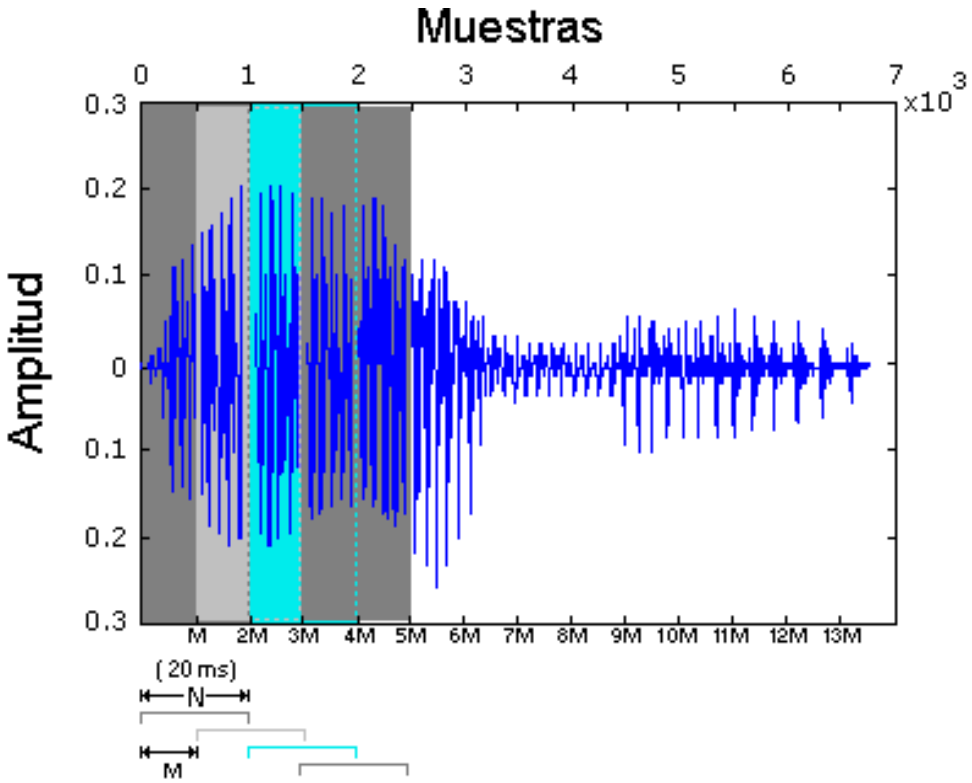


Figura 4.2 Secuencia de Tramas

4.2. Enventanado

Luego de obtener una trama de la señal, ésta posee discontinuidades en el inicio y final, lo que se puede apreciar en la Figura 4.3.

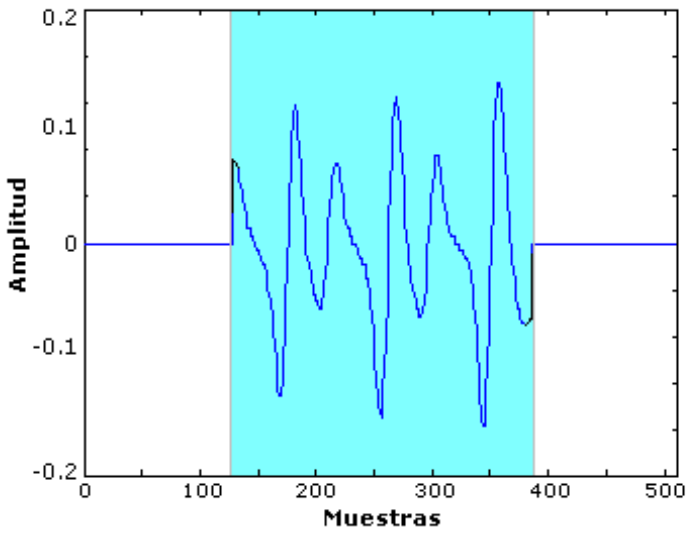


Figura 4.3 Trama con discontinuidades.

Por lo tanto, el siguiente paso consiste en multiplicar cada trama individualmente por una ventana temporal, a fin de minimizar las discontinuidades al inicio y al final de ésta. Este proceso se conoce como Enventanado. El efecto que se produce es el de convolucionar el espectro de la señal muestreada con el espectro de la ventana, produciendo una distorsión de la Transformada de Fourier de la señal original. Por ello, conviene elegir un tipo de ventana que produzca la menor distorsión posible.

Si se define una ventana como w(n), $0 \leq n \leq N-1$, donde N es el número de muestras de cada trama, entonces el resultado del enventanado de la trama $x_i(n)$ es:

$$y_l(n) = x_l(n)w(n), \quad 0 \leq n \leq N-1 \tag{4.1}$$

Típicamente se utiliza la ventana de Hamming (Ver Figura 4.4), la cual tiene la siguiente formulación matemática:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1 \tag{4.2}$$

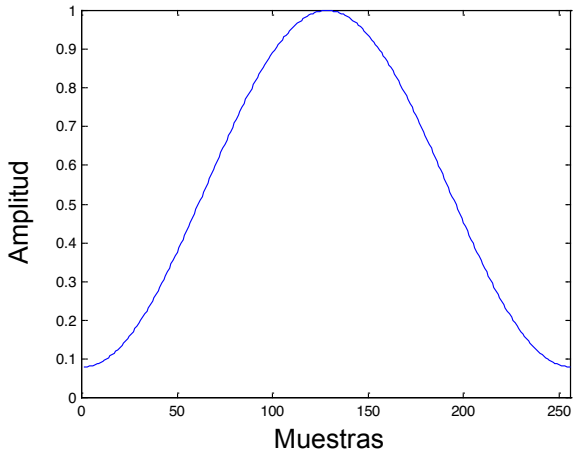


Figura 4.4 Ventana de Hamming

Luego de aplicar una ventana de Hamming a la trama original (ver Figura 4.5), en la señal resultante se aprecia, como las discontinuidades de inicio y final son reducidas a un valor muy cercano a cero.

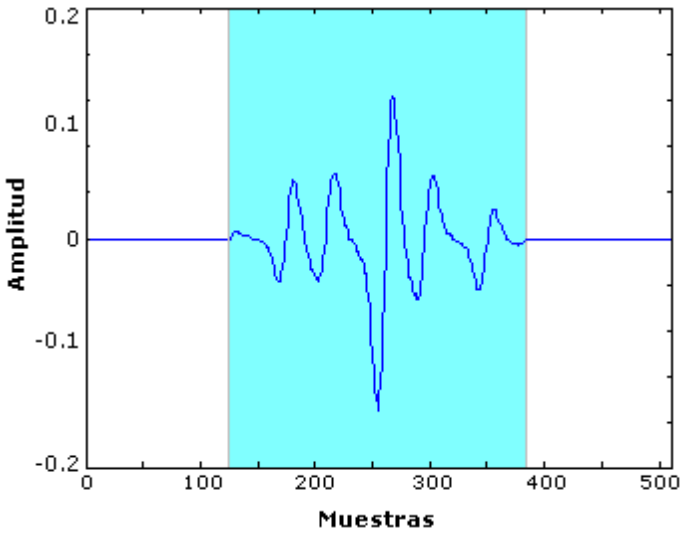


Figura 4.5 Señal enventanada.

4.3. Preenfasis

Debido a que la señal de voz se atenúa a 6 dB/octava conforme aumenta la frecuencia, es necesario introducir un filtrado cuya función es incrementar la relevancia de las componentes de alta frecuencia. Este proceso se conoce con el nombre de preénfasis y puede ser diseñado a través de un filtro digital paso alto. Este filtro paso alto puede implementarse con la siguiente ecuación en diferencias:

$$y[n]=x[n]-a \cdot x[n-1] \qquad (4.3)$$

donde a es una constante que varía entre 0 y 1.

4.4. Transformada Rápida de Fourier

En esta etapa se convierte cada trama de N muestras en el dominio del tiempo al dominio de la Frecuencia. Esto se realiza mediante el uso de la Transformada Discreta de Fourier(DFT), la cual esta definida por

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \qquad , k = 0,1,2, ..., N-1 \qquad (4.4)$$

La implementación directa de la ecuación (4.4) es muy ineficiente, especialmente cuando la secuencia de longitud N es larga. Para obtener una muestra de X(k), se necesita N multiplicaciones complejas y (N-1) sumas complejas. Así, para obtener un conjunto completo de coeficientes DFT, se necesita N^2 multiplicaciones complejas y $N(N-1) \approx N^2$ sumas complejas, lo cual es inconveniente en la práctica. Por esto se debe implementar un algoritmo que reduzca considerablemente el número de multiplicaciones y sumas que se necesitan para resolver la ecuación de la DFT. De aquí la **Transformada Rápida de Fourier** (**FFT**), es un algoritmo rápido, que permite implementar eficientemente la DFT.

Los distintos algoritmos utilizados para elaborar una FFT explotan las propiedades de simetría y periodicidad del factor de fase W_N[Proakis, Manolakis. 1998]. Se basan en el uso de la estrategia “divide y vencerás”, la cual consiste en la descomposición de una DFT de N muestras en DFTs más pequeñas, donde N se puede representar por el producto de dos enteros, esto es

$$N=LM \tag{4.5}$$

Por tanto, la secuencia x(n) puede almacenarse en una matriz bidimensional indexada por l y m, así los índices n y k de la ecuación (4.4) pueden ser escritos como

$$\begin{aligned} &n=Ml+m, && 0 \le l \le L-1, \quad 0 \le m \le M-1 \\ &k=p+Lq, && 0 \le p \le L-1, \quad 0 \le q \le M-1 \end{aligned} \tag{4.6}$$

Seguidamente, la secuencia x(n) se divide en M pequeñas secuencias de longitud L, se toma M pequeñas DFT's de L puntos, y se combinan en una DFT más grande usando L pequeñas DTFs de M puntos. Las secuencias x(n) y X(k) se pueden escribir como matrices x(l,m) y X(p,q), respectivamente. Entonces la ecuación (4.4) puede ser escrita como:

$$X(p,q)=\underbrace{\sum_{m=0}^{M-1}\left\{W_N^{mp}\underbrace{\left[\sum_{l=0}^{L-1}x(l,m)W_N^{lp}\right]}_{\text{DFTL-puntos}}\right\}W_M^{mq}}_{\text{DFT M-puntos}} \tag{4.7}$$

Esta ecuación puede ser desarrollada con un cálculo en tres pasos:

1. Primero se calcula las DFTs de L puntos.

$$F(P,m)=\sum_{l=0}^{L-1}x(l.m)W_L^{lp} \qquad ; 0 \le p \le L-1 \tag{4.8}$$

para cada una de las filas m=0,1, ..., M-1

2. Segundo, se calcula la nueva matriz rectangular G(p,m) definida como

$$G(p,m) = W_N^{pm} F(p,m) \qquad ; 0 \le m \le M\text{-}1 \qquad (4.9)$$

$$0 \le p \le L\text{-}1$$

3. Finalmente, se calcula las DFTs de L puntos

$$X(p,q) = \sum_{m=0}^{M-1} G(p,m) W_M^{mq} \qquad ; 0 \le q \le M\text{-}1 \qquad (4.10)$$

para cada columna p=0,1, ..., M-1, de la matriz G(p,m)

El número total de multiplicaciones complejas para esta aproximación puede ser dada por

$$C_N = ML^2 + N + LM^2 < N^2 \qquad (4.11)$$

En particular, el método es muy eficiente cuando N es compuesto, esto es, cuando N puede factorizarse como $N=r_1 r_2 r_3 \cdots r_v$, donde los $\{r_j\}$ son primos.

El caso en que $r_1=r_2=r_3= \cdots =r_v \equiv r$, de manera que $N=r^v$, es de particular importancia. En dicho caso las DFTs son de tamaño r, de manera que el cálculo de las DFTs de N puntos sigue un patrón regular. El número r se denomina base (radix) del algoritmo para las FFT. Los algoritmos en base 2 (radix 2), son los más usados por los algoritmos para la FFT.

La Figura 4.6 muestra el espectro de una señal luego de aplicarle la FFT.

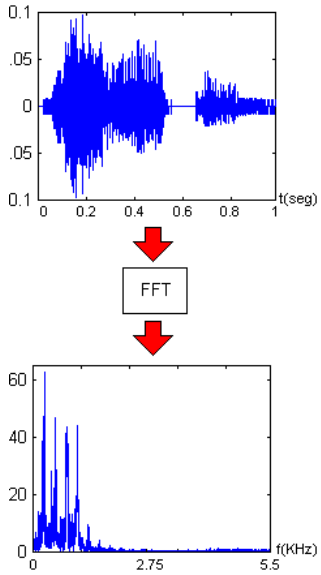


Figura 4.6 Transformada Rápida de Fourier.

4.5. Energía en cada Banda

Estudios científicos han mostrado que la percepción humana del contenido de las frecuencias de los sonidos de la señal de voz no sigue una escala lineal, como se sabe la frecuencia es una entidad física y por tanto puede ser medida de forma objetiva por diferentes medios. Por el contrario la altura o tono[Borja, 1997] de un sonido, es un fenómeno totalmente subjetivo y por tanto, no es posible medirlo de forma objetiva. Normalmente, cuando se aumenta la frecuencia de un sonido, su altura también sube, sin embargo esto no se da de forma lineal, o sea, no se corresponde la subida del valor de la frecuencia con la percepción de la subida de tono.

Por procedimientos estadísticos sobre un determinado número de personas sin conocimientos musicales se fijó el valor de la escala subjetiva mediante una ley empírica que define una nueva escala de tonos. Para medir los intervalos de esta escala se utiliza la unidad Mel, llamada también Melio. Por definición un sonido de 1000 Hz, con 40 dB por encima del umbral de percepción, tiene un tono de 1000 mels.

Así, para computar las unidades Mels del sonido a una frecuencia f(Hz), se utiliza la siguiente fórmula.

$$Mel(f) = 2595*\log_{10}(1+f/700) \quad (4.12)$$

En la figura 4.7, se muestra una gráfica Mel Vs Frecuencia, como se aprecia, la escala de Frecuencia Mel tiende a un espaciamiento lineal de frecuencia por debajo de los 1000 Hz y a un espaciamiento logarítmico sobre los 1000Hz.

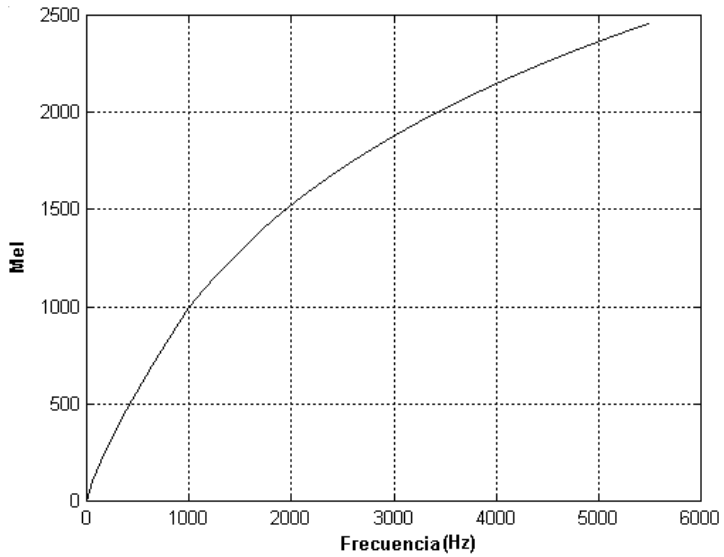


Figura 4.7 Gráfica Mel vs Frecuencia

Una manera de aproximarse a este espectro subjetivo es utilizar un banco de filtros, mucho más estrechos y linealmente espaciados hasta aproximadamente 1KHz (Ver figura 4.8a), y muy amplios y logarítmicamente espaciados a partir de 1KHz (Ver figura 4.8b). De este modo, se da mayor importancia a la información contenida en las bajas frecuencias en concordancia con el comportamiento del oído humano.

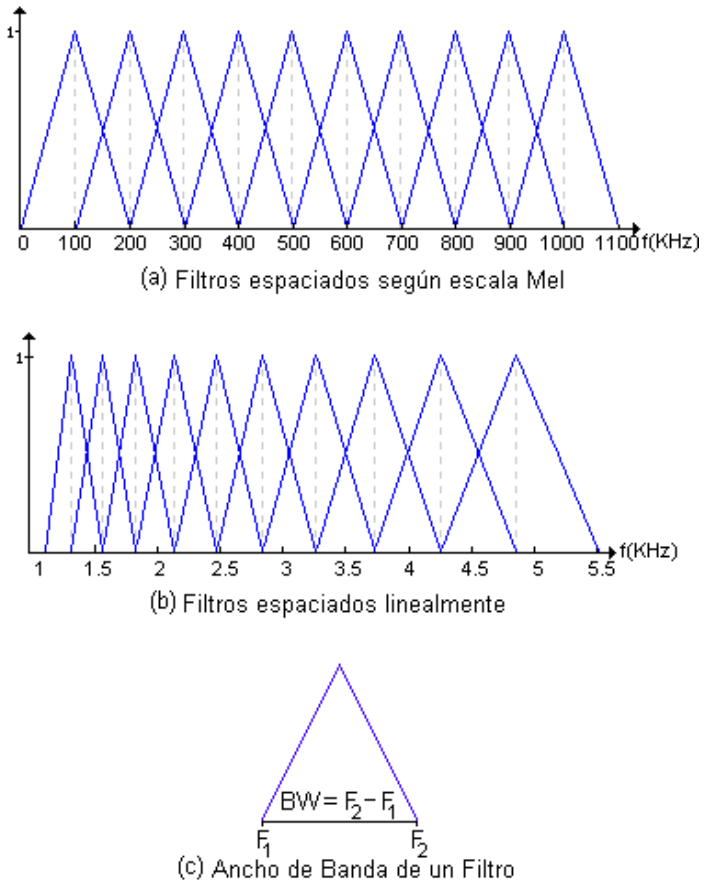


Figura 4.8 Banco de Filtros Triangulares.

Los filtros son aplicados directamente en el dominio de la frecuencia, y su respuesta está dada por la siguiente ecuación:

$$\omega(f) = \begin{cases} \dfrac{2f}{BW - 1}, & 0 \le f \le \dfrac{BW - 1}{2}\,(Hz) \\ 2 - \dfrac{2f}{BW - 1}, & \dfrac{BW - 1}{2} \le f \le BW - 1\,(Hz) \end{cases} \qquad (4.13)$$

Donde BW es el ancho de banda del filtro triangular y f es la frecuencia en Hz. (Ver figura 4.8c).

Típicamente se toma un banco de 20 filtros triangulares, los diez primeros filtros se ubican hasta 1 KHz, son linealmente espaciados y dividen el espectro en 10 espacios iguales. Poseen un ancho de banda de 200Hz, y se traslapan en 100Hz, tal como se muestra en figura 4.9.

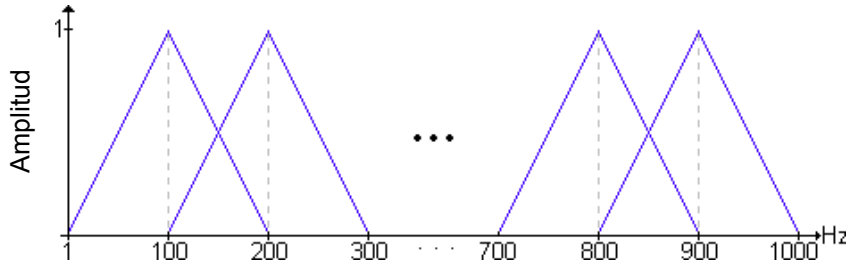
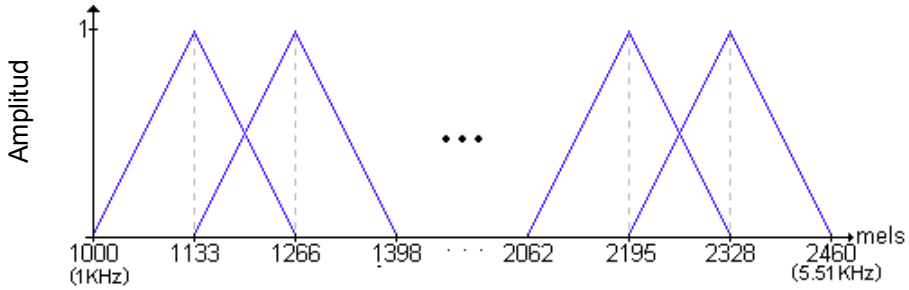
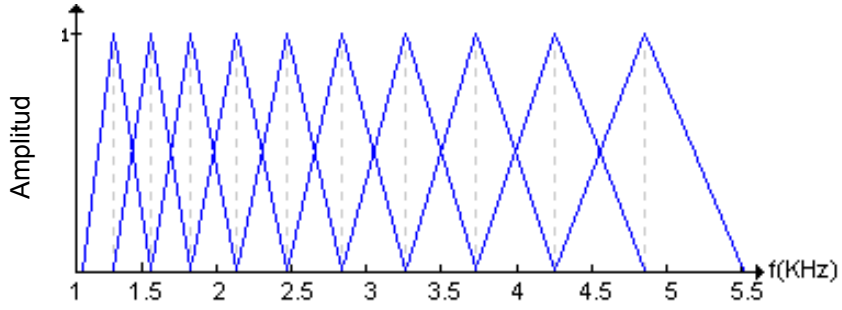


Figura 4.9 Filtros en escala lineal.

Los siguientes 10 filtros triangulares se ubican después de 1KHz, hasta 5.5KHz (Fs/2, para Fs=11KHz), 1000 y 2460 mels, respectivamente, aplicando la ec. 4.12. Los filtros son espaciados uniformemente en la escala de frecuencia Mel (Ver figura 6.10a), pero en el dominio de la frecuencia, estos filtros se encuentran espaciados logarítmicamente (Ver figura 4.10b). En el ítem 4.5.1 se explicará en detalle los criterios para el diseño del banco de filtros.



(a) Filtros Triangulares en escala Mel



(b) Filtros Triangulares en el dominio de la frecuencia

Figura 4.10 Filtros en Escala Mel

Seguidamente, se calcula la Energía en cada una de las bandas de frecuencias en las que el banco de filtros divide el espectro. La cual está definida por la siguiente fórmula.

$$E(i) = \int_{F_1}^{F_2} |X(F)|^2 dF \quad ; \quad i=1, 2, ..., L \qquad (4.14)$$

Donde:

F_1 : Frecuencia de inicio del filtro. (Ver figura 4.8c)

F_2 : Frecuencia de fin del filtro.

$|X(F)|^2$: Densidad Espectral de Energía.

L : Número de Filtros

4.5.1. Diseños del banco de filtros triangulares

Para diseñar el banco de filtros debe determinarse las frecuencias de Inicio(fStart), Centro(fCent) y Final(fStop) de cada filtro triangular, tal como se puede apreciar en la figura 4.11.

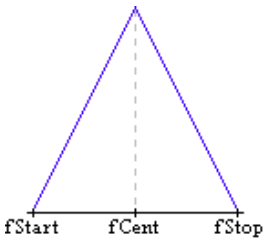


Figura 4.11 Filtro Triangular

Como se sabe, la trama de análisis, posee un tamaño N de 256 muestras(Ver ítem 4.1), luego de aplicarle la Transformada Rápida de Fourier(FFT), el espectro de esta señal poseerá un tamaño de 128 muestras, lo que equivale a un ancho de banda de Fs/2, es decir, 5.5 KHz. Así, el objetivo del diseño de los filtros es establecer una correspondencia entre las frecuencias de inicio, centro y final del filtro y las muestras del espectro obtenido, esto se aprecia mejor en la figura 4.12.

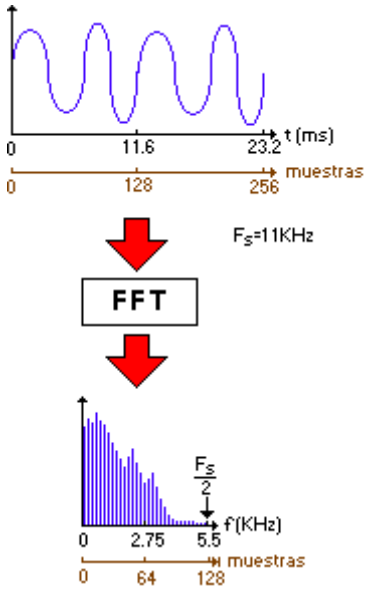


Figura 4.12 Frecuencias y número de muestras

La siguiente fórmula establece una relación entre un intervalo de frecuencia y el número de muestras correspondiente.

$$n = \frac{N \cdot \Delta f}{F_S} \tag{4.15}$$

donde:

Δf: es el intervalo de frecuencia en Hz.

Fs: es la frecuencia de muestreo en Hz.

N: es el número total de muestras de la trama.

n: es el número de muestras.

Para determinar las frecuencias de inicio, centro y fin de los filtros se deben considerar dos casos: Los bancos de filtros por debajo de 1Khz, los cuales poseen un comportamiento lineal, y el banco de filtros por encima de 1KHz espaciados en escala Mel.

- **Diseño de los filtros en escala lineal**

Inicialmente se debe establecer la frecuencia de Inicio (FrecInicio) y Final (FrecFinal) del banco de filtros, según lo explicado en los párrafos anteriores se tiene que los valores son

FrecInicio = 0Hz

FrecFinal = 1kHz

Con estos valores se determina que el tamaño del banco de filtros Δf es igual a 1KHz, y reemplazando en la ec. 4.15 se obtiene un número de muestras aproximado de 23.22.

Como se sabe, se utilizarán 10 filtros para dividir el espectro hasta 1Khz, en 10 espacios uniformes, así que el número n de muestras debe ser

redondeado al múltiplo de 10 más cercano, es decir, 20 o 30, pero como n=23.22. se aproxima a 20 (n=20).

Al tener n=20, los índices se crean por una simple regla de tres simple.

$$Indice=n*f/T \tag{4.16}$$

De donde se obtiene la siguiente tabla

Tabla 4.1 Relación entra la Frecuencia y Número de Muestras

Hertz			Índice		
fStart	fCent	fStop	nStart	nCent	NStop
0.00	100.00	200.00	0	2	4
100.00	200.00	300.00	2	4	6
200.00	300.00	400.00	4	6	8
300.00	400.00	500.00	6	8	10
400.00	500.00	600.00	8	10	12
500.00	600.00	700.00	10	12	14
600.00	700.00	800.00	12	14	16
700.00	800.00	900.00	14	16	18
800.00	900.00	1 000.00	16	18	20
900.00	1 000.00	1 100.00	18	20	22

- **Diseño de los filtros en escala Mel**

En primer lugar, se debe determinar la frecuencia de Inicio (FrecInicio) y fin (FrecFinal) del banco de filtros, y realizar su conversión a Mels mediante la ec. (4.12), de donde se obtiene que

FrecInicio = 1 KHz $\rightarrow$ MelInicio = 1000 mels

FrecFinal = 5.5 KHz $\rightarrow$ MelFinal = 2460.5 mels

Seguidamente, se divide el espacio, en escala Mel, comprendido entre MelInicio y Melfinal en 10 segmentos de igual tamaño (10 filtros triangulares), de donde se obtiene el Inicio, Centro y Final de cada filtro en Melios. Estos datos son mostrados en la Tabla 4.2.

Tabla 4.2 Indices de los filtros en Escala Mel

Mel		
mStart	mCent	mStop
999.99	1 132.76	1 265.53
1 132.76	1 265.53	1 398.31
1 265.53	1 398.31	1 531.08
1 398.31	1 531.08	1 663.85
1 531.08	1 663.85	1 796.63
1 663.85	1 796.63	1 929.40
1 796.63	1 929.40	2 062.18
1 929.40	2 062.18	2 194.95
2 062.18	2 194.95	2 327.72
2 194.95	2 327.72	2 460.50

Luego, se convierten los valores obtenidos en escala Mel a Hertz mediante la siguiente fórmula.

$$f = 700 \cdot (10^{\,Mel\,/\,2595} - 1) \;\; Hz \qquad (4.17)$$

De donde se obtiene la siguiente tabla.

Tabla 4.3 Indices de los filtros en Hertz

Mel			Hertz		
mStart	mCent	MStop	fStart	fCent	FStop
999.99	1 132.76	1 265.53	1 000.00	1 212.56	1 451.69
1 132.76	1 265.53	1 398.31	1 212.56	1 451.69	1 720.72
1 265.53	1 398.31	1 531.08	1 451.69	1 720.72	2 023.39
1 398.31	1 531.08	1 663.85	1 720.72	2 023.39	2 363.90
1 531.08	1 663.85	1 796.63	2 023.39	2 363.90	2 746.99
1 663.85	1 796.63	1 929.40	2 363.90	2 746.99	3 177.98
1 796.63	1 929.40	2 062.18	2 746.99	3 177.98	3 662.85
1 929.40	2 062.18	2 194.95	3 177.98	3 662.85	4 208.35
2 062.18	2 194.95	2 327.72	3 662.85	4 208.35	4 822.06
2 194.95	2 327.72	2 460.50	4 208.35	4 822.06	5 512.50

Una vez obtenidas las frecuencias de Inicio, Centro y Final de los filtros, es necesario establecer a que número de índice corresponde. Ya que, en este caso, no se trata de un espaciamiento lineal, se debe obtener la distancia de una frecuencia a otra y realizar su conversión a número de muestras.

El primer valor con el que se toma, es el último valor utilizado en los índices de la escala lineal, es decir nStart(1) = 22 (Ver Tabla 4.1). Así, los siguientes índices se obtiene utilizando las siguientes fórmulas

Fórmula para obtener los nStart

$$\mathrm{nStart[i]} = \mathrm{floor}\left(\mathrm{nStart[i-1]} + \frac{\mathrm{nStart[i]} - \mathrm{nStart[i-1]}}{\mathrm{Fs}} \cdot \mathrm{N} \right) \quad ; \text{i=2,3,…,10} \qquad (4.18)$$

Fórmula para obtener los nCent

$$\mathrm{nCent[i]} = \mathrm{floor}\left(\mathrm{nStart[i-1]} + \frac{\mathrm{nCent[i]} - \mathrm{nCent[i-1]}}{\mathrm{Fs}} \cdot \mathrm{N} \right) \quad ; \text{i=1,2,3,…,10} \qquad (4.19)$$

Fórmula para obtener los nStop

$$nStop[i] = floor\left(nCent[i-1] + \frac{nStop[i] - nStop[i-1]}{Fs} \cdot N \right) \quad ; i=1,2,3,\ldots,10 \qquad (4.20)$$

donde

floor( ) : Es el redondeo mas cercano a cero.

Fs : Es la frecuencia de muestreo

N : Es el número de muestras de cada trama.

Aplicando las fórmulas anteriores se obtiene la siguiente tabla.

Tabla 4.4 Indice de los filtros en Número de muestras

Mel			Hertz			Indice		
mStart	mCent	mStop	fStart	fCent	fStop	NStart	nCent	NStop
1 064.40	1 191.32	1 318.24	1 100.00	1 314.56	1 554.70	22.00	26.00	31.00
1 191.32	1 318.24	1 445.15	1 314.56	1 554.70	1 823.47	26.00	31.00	37.00
1 318.24	1 445.15	1 572.07	1 554.70	1 823.47	2 124.27	31.00	37.00	43.00
1 445.15	1 572.07	1 698.99	1 823.47	2 124.27	2 460.93	37.00	43.00	50.00
1 572.07	1 698.99	1 825.91	2 124.27	2 460.93	2 837.72	43.00	50.00	58.00
1 698.99	1 825.91	1 952.83	2 460.93	2 837.72	3 259.42	50.00	58.00	67.00
1 825.91	1 952.83	2 079.74	2 837.72	3 259.42	3 731.40	58.00	67.00	77.00
1 952.83	2 079.74	2 206.66	3 259.42	3 731.40	4 259.63	67.00	77.00	89.00
2 079.74	2 206.66	2 333.58	3 731.40	4 259.63	4 850.83	77.00	89.00	102.00
2 206.66	2 333.58	2 460.50	4 259.63	4 850.83	5 512.50	89.00	102.00	117.00
2 333.58	2 460.50	2 587.41	4 850.83	5 512.50	6 253.04	102.00	117.00	134.00
2 460.50	2 587.41	2 714.33	5 512.50	6 253.04	7 081.86	117.00	134.00	153.00

Una vez finalizado el diseño de los filtros, estos índices obtenidos son utilizados para obtener la función de transferencia de cada filtro y realizar el proceso de filtrado de la señal.

4.6. Cepstrum

El Cepstrum se define como la transformada inversa del logaritmo del módulo de la Transformada de Fourier de la señal.

$c(n)=F^{-1}[\log|X(w)|]$ donde $X(w)=F[x(n)]$ (4.21)

La representación cepstral del espectro del habla, provee una buena representación de las propiedades espectrales locales de la señal para cada trama. Tal como se vio en el Capitulo 2, en el modelo del tracto vocal, la voz se genera por una excitación producida por dos fuentes, la cual pasa a través de un filtro, cuya respuesta en frecuencia, modifica el espectro adicionando información lingüística al sonido. Así, para realizar el reconocimiento de las palabras pronunciadas, bastaría conocer solamente las características del tracto vocal (o el filtro que lo modela), ya que la información proveniente de las cuerdas vocales (excitación) sólo proporciona información acerca del locutor. Precisamente, este tipo de análisis denominado cepstral, se utiliza para realizar la separación de estos dos parámetros.

Siguiendo con el modelo del tracto vocal, si se analiza la excitación proveniente de la fuente que genera sonidos sonoros, es decir, la fuente que es representada por un tren de pulsos de periodo T, la forma de onda obtenida puede ser aproximada por la siguiente ecuación.

$$s(t) = \sum_{n=-\infty}^{\infty} s_0(t-nT) = S_0(t) * \left[\sum_{n=-\infty}^{\infty} \delta(t-nT) \right] \quad (4.22)$$

Donde $s_0(t)$ es la respuesta al impulso del sistema de generación de voz, $\sum_{n=-\infty}^{\infty} \delta(t-nT)$ el tren de pulsos de periodo T, y '$*$' denota la convolución en el tiempo.

Aplicando la Transformada de Fourier a la ecuación (4.22), se tiene

$$S(\omega) = S_0(\omega)\left\{\frac{\sin\left[(2N+1)\frac{1}{2}\omega T\right]}{\sin\left(\frac{1}{2}\omega T\right)}\right\}^2 \qquad (4.23)$$

donde $S(\omega)$, $S_0(\omega)$ son los espectros de energía de s(t) y $s_0(t)$ respectivamente, y el término que se encuentra dentro de las llaves representa el espectro armónico[Shuzo, Kazuo. 1985] con una frecuencia $\omega=2\pi/T$.

Si se aplica el logaritmo a ambos términos de la ecuación (4.23) el producto se transforma en sumas tal como se muestra a continuación

$$\ln\left[S(\omega)\right] = \ln\left[S_0(\omega)\right] + 2\ln\left\{\frac{\sin\left[(2N+1)\frac{1}{2}\omega T\right]}{\sin\left(\frac{1}{2}\omega T\right)}\right\} \qquad (4.24)$$

El primer término de la derecha, $S_0(\omega)$ es el espectro de Energía del Sistema de generación de voz, el cual demuestra un cambio relativamente lento en la frecuencia, por tanto, contiene información de la envolvente del espectro, que se relaciona con la respuesta en frecuencia del filtro que modela el tracto vocal.

El segundo término, es el espectro armónico lineal, con una frecuencia fundamental $\omega=2\pi/T$, demuestra un cambio rápido en la frecuencia, y por tanto corresponden al rizado del espectro, el cual se relaciona estrechamente con la frecuencia fundamental y el carácter periódico de la excitación aplicada al tracto vocal.

Si denotamos por E(w) y H(w) a las transformaciones de la excitación y el filtro respectivamente, se cumple:

$$X(w)=E(w)\cdot H(w) \qquad (4.25)$$

entonces reemplazando en la ecuación (4.21) se tiene:

$$c(t) = F^{-1}[\log(|E|*|H|)] = F^{-1}[\log(|E|)] + F^{-1}[\log(|H|)] = c_e(n) + c_h(n) \quad (4.26)$$

Esto significa que el cepstrum de una señal es la suma del cepstrum de la excitación y el cepstrum del filtro (de la respuesta al impulso del filtro). De lo anteriormente descrito, se concluye que las bajas componentes cepstrales están relacionadas con las características del tracto vocal y las altas componentes cepstrales, con la información sobre el locutor. Entonces es posible separar $\{c_e(n)\}$ de $\{c_h(n)\}$ utilizando ventanas "pasa bajo" y "pasa alto" apropiadas, tal como se muestra en la figura 4.13. Para realizar, el reconocimiento de palabras aisladas, sólo se necesita obtener información sobre las características del tracto vocal, por lo tanto, se analizará las bajas componentes cepstrales.

$$c_e(n) = c(n)*w_{lp}(n) \quad (4.27)$$
$$c_h(n) = c(n)*w_{hp}(n)$$

$$w_{lp} = \begin{cases} 1, & |n| \le N_1 \\ 0, & \text{en otro caso} \end{cases} \quad (4.28)$$

$$w_{hp} = \begin{cases} 0, & |n| \le N_1 \\ 1, & |n| > N_1 \end{cases}$$

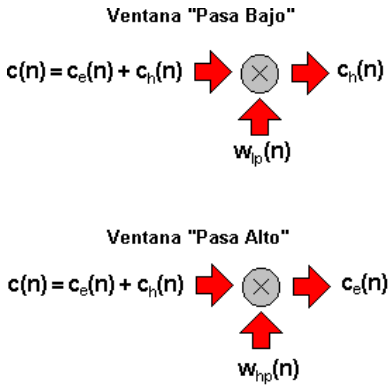


Figura 4.13 Ventanas "Pasa Bajo" y "Pasa Alto"

En la figura 4.14, se muestra las etapas diferentes etapas para obtener información sobre la envolvente espectral, realizando un análisis cepstral. Mientras en la figura 4.15 se muestra las gráficas de las señales más importantes obtenidas en diversas etapas.

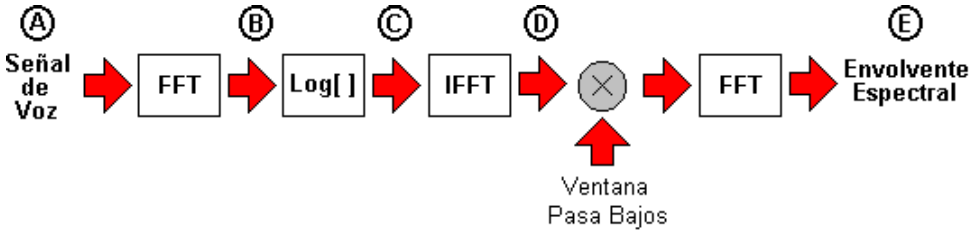


Figura 4.14 Diagrama de Bloques para obtener la Envolvente Espectral.

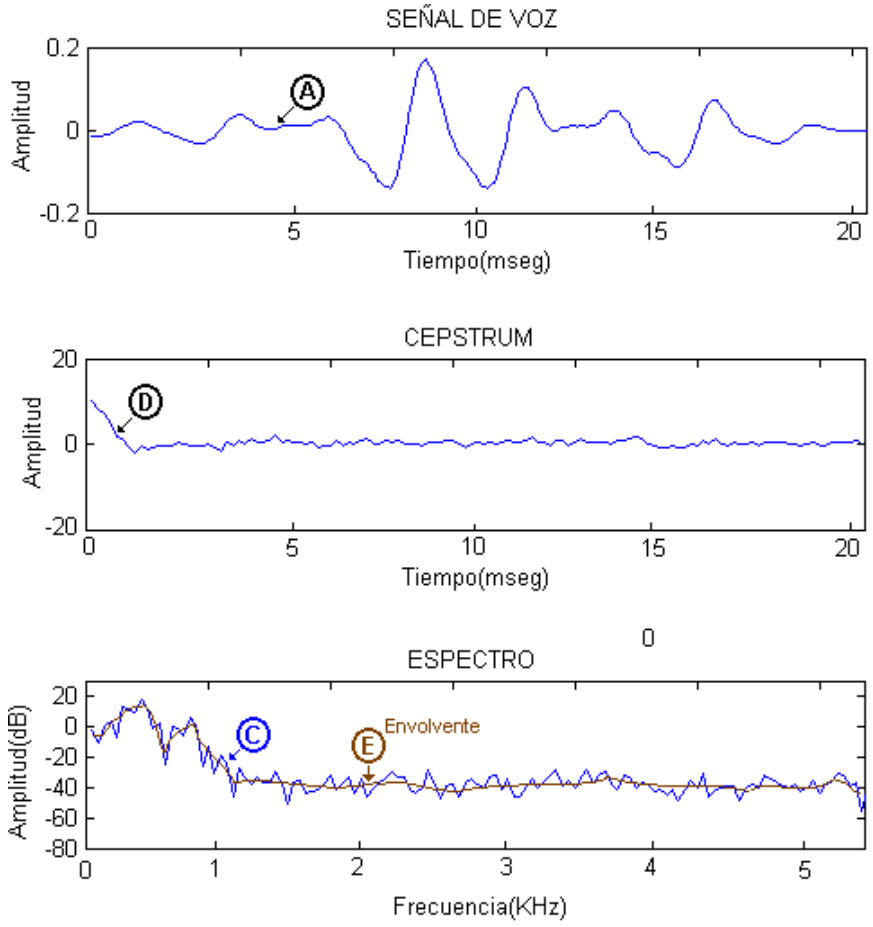


Figura 4.15 Gráficas de las distintas etapas.

El tamaño adecuado de la ventana “Pasa Bajo”, está relacionado con la frecuencia fundamental (pitch) de la palabra pronunciada[Shuzo, Kazuo. 1985], es decir, se debe extraer el pitch de la trama analizada, y según este valor separar las bajas componentes cepstrales de las altas componentes cepstrales.

En lo que respecta a la Transformada Inversa, debido a que los coeficientes del espectro y su logaritmo son número reales, se pueden convertir al dominio del tiempo utilizando la Transformada Coseno Discreta (DCT)[Boaz. 1997], que hace las veces de Transformada Inversa de Fourier. La ecuación de la Transformada Discreta del Coseno se muestra a continuación.

$$x[n] = \sqrt{\frac{2}{N-1}} \sum_{k=0}^{N-1} a[k] \cdot a[n] \cdot X^{C1}[k] \cdot \cos\left(\frac{\pi n k}{N-1}\right) \qquad (4.29)$$

donde:

$\mathbf{X^{C1}}$: Es la transformada de la secuencia x[n].

$$a[n] = \begin{cases} 2^{-1/2}, & n = 0, N-1 \\ 1, & \text{Otro Caso} \end{cases}$$

Aplicando la ecuación (4.29), se convierte el logaritmo de las concentraciones de energía (Ec. 4.14) del espectro Mel al dominio del tiempo, obteniéndose como resultado los denominados coeficientes Mel-Cepstrum (Ver figura 4.16). El cálculo de los MFCC, responde a la siguiente expresión:

$$MFCC_j(i) = \sum_{k=1}^{NF} \log[E(j,k)] \cos\left[i\left(k - \frac{1}{2}\right)\frac{\pi}{L}\right]; \quad i=1,2,3,...,P \qquad (4.30)$$

Donde :

k : es la banda de frecuencias.

j : es la trama en curso.

E(j,k) : es la energía de la banda k en la trama j.

NF : es el número de bandas o filtros.

P : es el número total de coeficientes MFCC (10, en nuestro caso).

Típicamente, se evalúan 10 puntos de la transformada inversa, para así, sólo obtener información sobre las bajas componentes cepstrales.

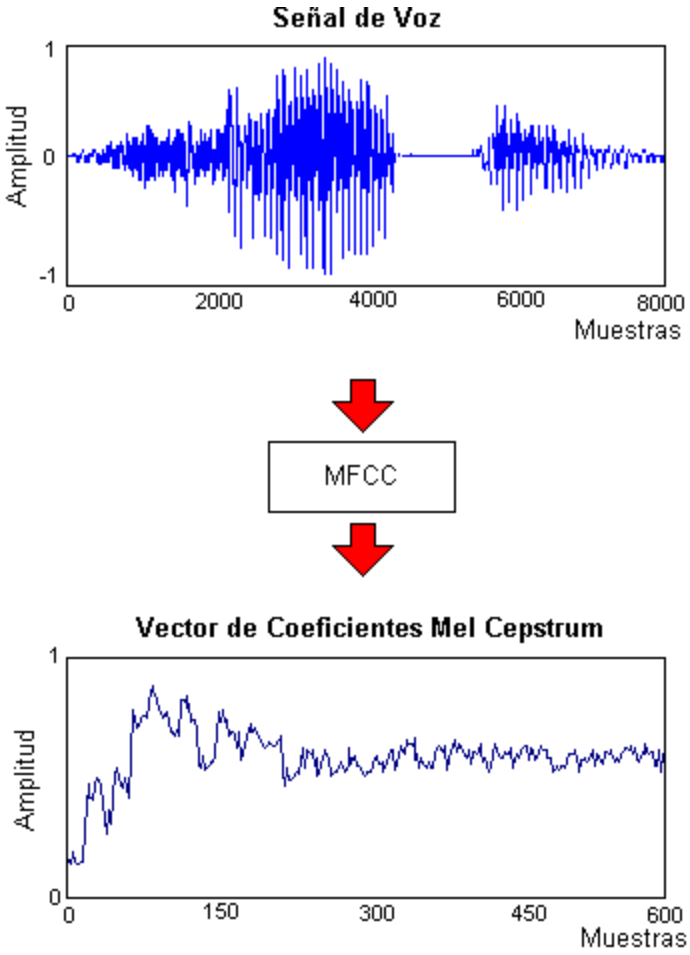


Figura 4.16 Vector de Coeficientes Mel-Cepstrum

5. FUNDAMENTOS DE LAS REDES NEURONALES

5.1 Introducción

Las Redes Neuronales Artificiales (RNA) han surgido de la evolución de los sistemas de computación inspirados en el cerebro humano y dotados de cierta 'inteligencia', con las cuales se intenta dar solución a problemas complejos. Las RNA son la combinación de elementos simples interconectados (neuronas), que operando paralelamente, consiguen resolver problemas relacionados con el reconocimiento de formas o de patrones, predicción de eventos, control de procesos entre otros.

Tal como se afirmó en el capítulo 3, el proceso de Reconocimiento de Voz constituye un problema clásico de reconocimiento de patrones que puede ser solucionado utilizando las RNA. En el caso particular del presente trabajo se ha usado red neuronal tipo Perceptrón Multicapa (MLP). En consecuencia, en este capítulo se analizan los fundamentos de las RNA, partiendo del análisis de la neurona, su modelo matemático, hasta llegar a la descripción de la estructura de la red. Finalmente, se efectúa una revisión particular del MLP y el algoritmo de aprendizaje utilizado sobre éste, denominado *Backpropagation.*

5.2 Modelo de la Neurona

5.2.1 La Neurona

El cerebro humano está compuesto principalmente por un número muy grande de neuronas (alrededor de 10^{10}), masivamente interconectadas, con un promedio de unos miles de interconexiones por cada neurona. Cada neurona es una célula especializada que puede propagar una señal electroquímica. La neurona tiene estructuras de entrada en forma de ramificaciones llamadas dendritas, un cuerpo celular o soma y una ramificación de salida llamada axón (ver figura 5.1). El axón de una célula se conecta a la dendrita de otra vía una sinapsis. Cuando una neurona es activada, ésta envía una señal

electroquímica a lo largo del axón. Esta señal cruza la sinapsis hacia otra célula la cual se activa y repite el proceso. Una neurona se activa sólo si el total de la señal recibida por el soma desde las dendritas excede un cierto nivel denominado 'umbral de disparo'.

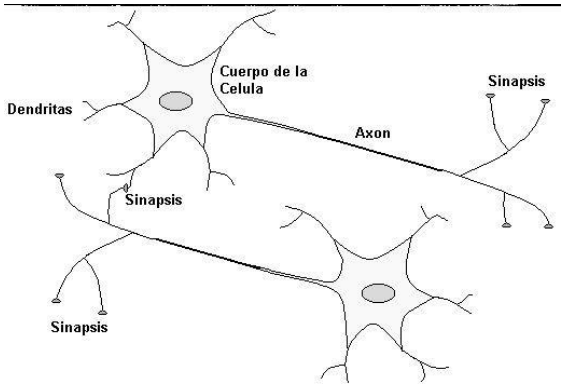


Figura 5.1 Neurona Biológica

La magnitud de la señal recibida por una neurona depende críticamente de la eficacia de la sinapsis, cada una de éstas contiene una abertura con un neurotransmisor químico balanceado, para transmitir una señal a través de la abertura. Donald Hebb postuló que el aprendizaje consistía principalmente en alterar la magnitud de las conexiones sinápticas. Por ejemplo, en el clásico experimento del condicionamiento Pavloviano, cuando se hace sonar una campanilla justo antes de servir la comida al perro, este aprende rápidamente a relacionar (asociar) el sonido con la acción de comer. Las conexiones sinápticas entre las partes apropiadas de la corteza auditiva y las glándulas salivales son reforzadas, por esto, cuando la corteza auditiva es estimulada por el sonido de la campanilla, el perro comienza a salivar. Así, a partir de un número elevado de sencillas unidades de proceso, el cerebro gestiona la ejecución de tareas extremadamente complejas.

La manera por la cual una red de neuronas biológicas procesa la información, está basada en la descomposición paralela de elementos complejos en elementos básicos, y no de manera secuencial [Freeman, Skapura. 1993]. Es decir, la fisiología de una red neuronal, consiste en descomponer la información compleja en sus elementos fundamentales, y almacenar estos elementos y sus

interrelaciones en un banco de memoria del cerebro. Por ejemplo, cuando se mira una pintura, el cerebro no almacena en la memoria una matriz de pixeles, en lugar de esto almacena las características fundamentales de la pintura tales como líneas, puntos, formas colores, etc. Claro está, que hay un mayor grado de complejidad en el funcionamiento del cerebro que lo discutido aquí, pero es interesante y resaltante que las RNA puedan lograr algunos resultados remarcables, usando modelos no más complicados que el biológico.

5.2.2 *El Modelo Básico de la Neurona*

La Neurona Artificial también denominada Unidad de Proceso es el elemento fundamental de una RNA cuya función, simple y única, consiste en recibir las entradas de células vecinas y calcular un valor de salida. En cualquier sistema que se esté modelando, es útil caracterizar tres tipos de unidades: entrada, salida y ocultas. Las unidades de entrada (que constituyen a la vez las entradas de la red), son las que reciben las señales desde el entorno proveniente de sensores o de otros sectores del sistema; las unidades de salida envían la señal fuera del sistema (salidas de la red); estas señales pueden controlar directamente actuadores u otros sistemas. Y las unidades ocultas no tienen contacto con el exterior, es decir sus entradas y salidas se encuentran dentro del sistema

Para describir el Modelo Básico de una Neurona se ha establecido un modelo matemático de una neurona artificial genérica [Stamatios, Kartalopoulos. 1996]. Para establecer una similitud directa con el modelo biológico, en las RNA se manejan los siguientes conceptos: Se denomina sinapsis; a la interconexión entre dos unidades de proceso y al parámetro que cuantifica la intensidad de la conexión se le llama peso. Las señales que llegan a la sinapsis, constituyen las entradas de la neurona, éstas son ponderadas (atenuadas o amplificadas) a través del peso, asociado a la sinapsis correspondiente. Estas señales de entrada, pueden excitar a la neurona (sinapsis con peso positivo) o inhibirla (peso negativo). Si la suma de estas entradas ponderadas es mayor o igual que el valor umbral de la neurona,

entonces la neurona se activa. La figura 5.2 muestra el esquema básico de una neurona artificial.

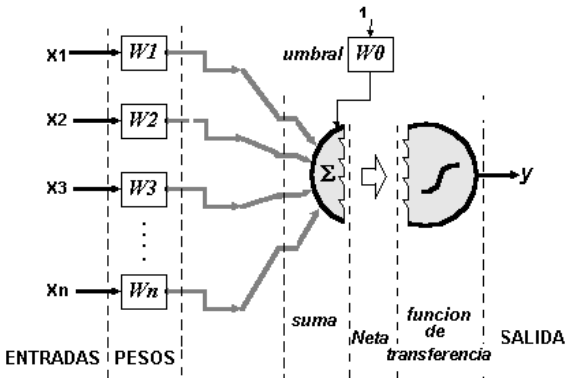


Figura 5.2 Esquema de una Neurona Artificial

Donde, Xi (i=1,2..n), representa las entradas, los bloques Wi (i=1,2..n) representan a las sinapsis y la salida de la neurona esta representada por *y*, el valor del umbral viene representado por W0, el cual tiene como entrada a la unidad. El cuerpo esta dividido en dos etapas, la primera indica la suma de las entradas ponderadas, tomando el valor del umbral como una entrada más, y dando como resultado una Entrada Neta (o simplemente Neta). La expresión matemática que expresa este proceso es:

$$\text{Neta} = \sum_{i=0}^{N} \text{Xi.Wi} \tag{5.1}$$

La segunda toma el valor de la Entrada Neta la evalúa mediante una función de transferencia, el resultado de esta etapa constituye la respuesta de salida **y**, de la neurona. La siguiente ecuación muestra este proceso:

$$y = f(\text{Neta}) = f.\left(\sum_{i=0}^{N} \text{Xi.Wi} \right) \tag{5.2}$$

Cabe resaltar, que así como la magnitud de salida de una neurona es función directa de la magnitud de entrada, el comportamiento de esta función

no es lineal con la excitación de entrada, es decir, cuando esta excitación es suficientemente alta, la neurona entra en la zona refractaria, donde la salida permanece sin variación significativa a pesar de los cambios en la excitación.

*La Función de Transferencia e*s la función que actúa sobre el valor de la Entrada Neta de una neurona. El uso de esta función tiene el propósito de asegurar que la respuesta de salida de la neurona, se encuentre entre ciertos valores establecidos durante el diseño. Esta acotación tiene como fin acondicionar la respuesta de salida, especialmente cuando los valores de los estímulos de entrada son considerablemente altos o demasiado pequeños. Existen Funciones de Transferencias comúnmente usadas durante en diseño de una red neuronal, el uso de cada una de ellas depende del modelo que se pretende implementar y el algoritmo de aprendizaje en uso. A continuación se citan algunas funciones de transferencia típicas.

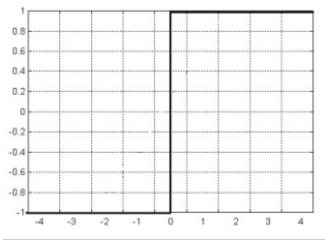
La Función Escalón produce dos valores, 1 si la Entrada Neta es mayor o igual a cero, y 0 en caso contrario. La gráfica de esta función se muestra en la figura 5.3(a).

La Función Signo a diferencia de la función Escalón, produce el valor de 1, si la Entrada Neta es mayor que cero y el valor de -1, si es menor que cero. Las neuronas con estas características en la salida, son denominadas Neuronas de salida Binarias, estas funciones se muestran en las figuras 5.3(a) y 5.3(b).

La Función Lineal (figura 5.3(c)), otorga una salida proporcional a la Entrada Neta. Una variante de esta función, la constituye la Función Lineal-Mixta, es esta función, la salida está obligada a permanecer dentro de un intervalo de valores reales prefijados, con un limite inferior y superior, de n1 y n2 respectivamente, en la figura 5.3(d), se muestra la gráfica de esta función.

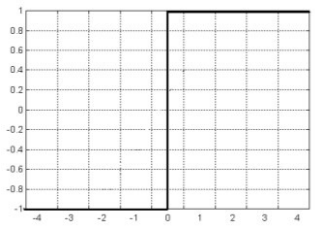
La Función Sigmoidal es una función continua y su importancia radica en su derivada, ya que siempre es positiva y cercana a cero para los valores

grandes positivos o negativos, además toma su valor máximo, cuando la Neta es igual a cero. Esto hace que se pueda utilizar las reglas de aprendizaje definidas para el escalón, con la ventaja de que su derivada esta definida en todo su dominio. En las figuras 5.3(e) y 5.3(f) se muestran dos variantes de esta función.



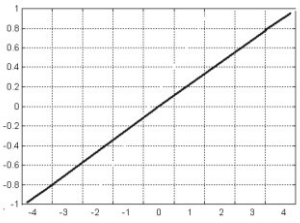
$$f = \begin{cases} 1 & si\ \ Neta \geq 0 \\ 0 & si\ \ Neta < 0 \end{cases}$$

(a) Función Escalón



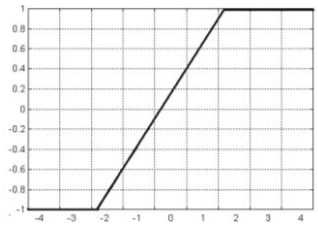
$$f = \begin{cases} +1 & si\ \ n \geq 0 \\ -1 & si\ \ n < 0 \end{cases}$$

(b) Función Signo



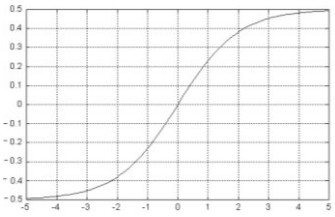
$$f = Neta$$

(c) Función Lineal



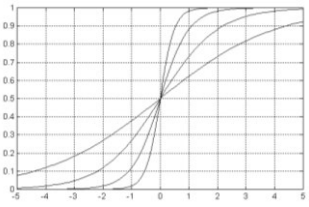
$$f = \begin{cases} +1 & si\ \ Neta > n2 \\ n & si\ \ n1 \leq Neta \leq n2 \\ -1 & si\ \ Netan < n1 \end{cases}$$

(d) Función Lineal Mixta



$$f = \frac{1}{1+e^{-k.Neta}} - 0.5$$

(e) Función Sigmoidal (acotada entre -0.5 y 0.5)



$$f = \frac{1}{1+e^{-k.Neta}}$$

(f) Función Sigmoidal (acotada entre 0 y 1)

Figura 5.3 Funciones de Transferencia de una Neurona

5.3 La Red Neuronal

Una Red Neuronal Biológica, está constituida por un número considerable de células denominadas neuronas, las cuáles se encuentran masivamente conectadas unas a otras. Estas conexiones entre neuronas, se realizan mediante contactos especiales denominadas Sinapsis y la intensidad de estos contactos es diferente en cada conexión.

Una Red Neuronal Artificial, trata de emular el funcionamiento de su correspondiente biológico. Consiste en un modelo matemático, compuesto por un gran número de unidades de proceso sencillas, a manera de neuronas, interconectadas unas con otras y distribuidas en niveles. De otra manera, se puede decir también, que se trata de una estructura de procesamiento de datos en paralelo, basada en grupos de elementos de proceso interconectados entre sí, y en las cuales se realizan un cálculo sencillo sobre los datos provenientes de las conexiones. De manera elemental, el modelo matemático es capaz de realizar algún tipo de procesamiento sobre estímulos de entrada, y basándose en cálculos internos ofrecer una respuesta a tales estímulos [Llamas, Cardeñoso. 1995] (ver figura 5.4).

Figura 5.4 Funcionamiento elemental de una Red Neuronal Artificial

Las Redes Neuronales Artificiales tienen la cualidad de aprender de la experiencia, de generalizar partiendo de casos anteriores, y de abstraer características esenciales de un objeto, debido a esto, tienen diversas características ventajosas tales como el Aprendizaje Adaptivo, que es la capacidad de aprender a realizar tareas, basada en un entrenamiento o experiencia inicial, y la Autoorganización, que permite a una red crear su propia organización de la información. Por otro lado, la capacidad de otorgar una respuesta en tiempo real en una red, está relacionada directamente con la tecnología destinada a su desarrollo, en la actualidad se pueden diseñar chips de estado sólido especializados, que mejoran su capacidad en muchas tareas. Finalmente, se debe resaltar la cualidad de Portabilidad, es decir, se podrían usar computadoras sofisticadas durante el entrenamiento de la red y equipos no muy potentes durante su operación.

Con respecto a la disposición de las unidades de proceso en una red, se conoce como capa o nivel a un conjunto de neuronas cuyas entradas provienen de la misma fuente (que puede ser otra capa de neuronas), y cuyas salidas dan origen al mismo destino (que puede ser otra capa de neuronas). Se pueden distinguir tres tipos. La Capa de entrada, la cual recibe directamente la información proveniente de las fuentes externas a la red, las capas Ocultas, estas son internas y no tienen contacto directo con el entorno exterior y la Capa de Salida, la cual envía información desde la red hacia el exterior. Los aspectos que caracterizan a una RNA son la topología y el mecanismo de aprendizaje.

5.3.1 *Topología de Las Redes Neuronales*

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

5.3.1.1 *Redes Monocapa*

Las redes monocapas suelen utilizarse frecuentemente en tareas relacionadas con la regeneración de información de entrada que se presenta a la red incompleta o distorsionada.

Estas redes podrían tener conexiones hacia adelante, o se podrían establecer conexiones laterales entre las neuronas que pertenecen a la única capa como se muestra en la figura 5.5(a), o bien podrían establecer conexiones recurrentes. Se dice que una red es Recurrente, cuando existe un lazo de reglamentación en un nivel, es decir cuando el conjunto de salidas de una capa o nivel se retroalimenta a manera de señales de entradas a neuronas de la misma capa, tal es el caso de la red de *Hopfield*, que se muestra en la figura 5.5(b).

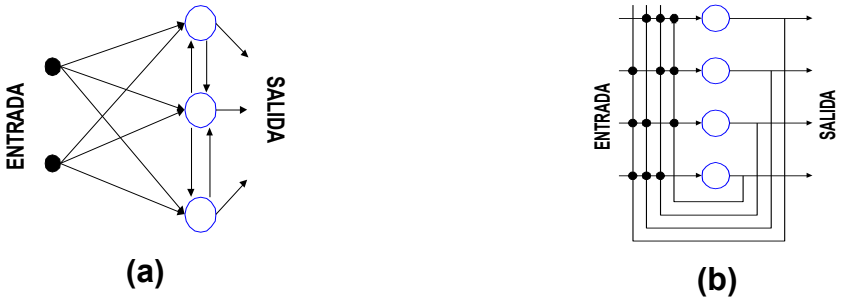


Figura 5.5 Redes Monocapa (a) Con conexiones laterales (b) Red Recurrente

5.3.1.2 Redes Multicapa

En las redes multicapa (ver figura 5.6), las neuronas de una capa reciben señales de entrada de otra capa anterior, (más cercana a las entradas de la red), y envían las señales de salida a una capa posterior, (más cercana a la salida de la red). A estas conexiones se las denomina conexiones hacia adelante o *feedforward* (figura 5.6a) y dan origen a las redes de tipo *feedforward*, en las cuales, todas las señales se propagan hacia adelante a través de las capas de la red.

Sin embargo, también existe la posibilidad de conectar las salidas de las neuronas de capas posteriores a las entradas de las capas anteriores, a estas conexiones se las denomina conexiones hacia atrás o *feedback* (figura5.6b). Este tipo de conexiones dan origen a las redes *feedforward/feedback*, en las cuales la información circula tanto hacia delante *(feedforward)* como hacia atrás *(feedback*) durante el funcionamiento de la red. Generalmente estas redes suelen ser de dos capas, existiendo por lo tanto dos tipos de pesos: los que corresponden a las conexiones *feedforward* de la primera capa hacia la segunda y los que corresponden a las conexiones *feedback* de la segunda a la primera capa. Estos valores no tienen por qué coincidir y de hecho en la mayoría de los casos son diferentes.

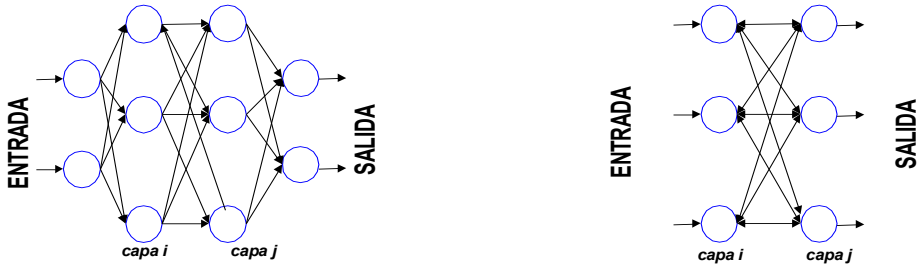


Figura 5.6 Redes Multicapa. a) *Feedforward.* b) *Feedforward/feedback.*

5.3.2 *Mecanismos de Aprendizaje*

Biológicamente, se suele aceptar que la información memorizada en el cerebro, está más relacionada con los valores sinápticos de las conexiones entre las neuronas, que con las neuronas mismas; es decir, el conocimiento se encuentra en las sinapsis. En el caso de las redes neuronales artificiales sucede algo similar, ya que se puede considerar que el conocimiento se encuentra representado en los pesos de las conexiones entre neuronas. Todo proceso de aprendizaje implica un cierto número de cambios en éstas conexiones, que consisten en la destrucción, modificación y creación de conexiones entre neuronas. En los sistemas biológicos, existe una continua creación y destrucción de conexiones. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma, pasa a tener un valor distinto de cero; de modo similar, una conexión se destruye cuando su peso pasa a ser cero. Durante el proceso de aprendizaje se modifican los pesos de las conexiones entre las neuronas de la red, por lo tanto, se puede decir que este proceso ha finalizado, es decir, que la red ha aprendido, cuando los valores de los pesos permanecen estables.

Un aspecto importante del aprendizaje de las RNA, es conocer cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red aprenda una nueva información. Estos criterios determinan lo que se conoce como regla de aprendizaje de la red, y establece una de las clasificaciones de las redes neuronales. Las RNA con Aprendizaje Supervisado y las RNA con Aprendizaje no Supervisado. La diferencia

fundamental entre ambos radica en la existencia o no de un agente externo (supervisor) que controle el proceso de aprendizaje de la red.

Otro criterio que se puede utilizar para diferenciar las reglas de aprendizaje, se basa en considerar si la red puede aprender durante su funcionamiento habitual, lo que se denomina ON LINE, o si el aprendizaje supone la desconexión de la red, denominado OFF LINE. Cuando el aprendizaje es OFF LINE se distingue entre una fase de aprendizaje o entrenamiento y una fase de operación o funcionamiento, existiendo un conjunto de datos de entrenamiento y un conjunto de datos de prueba. En estas redes los pesos de las conexiones permanecen fijos después que termina la etapa de entrenamiento. En las redes de aprendizaje ON LINE no se distingue entre fases de entrenamiento y de operación, por lo cual los pesos varían dinámicamente siempre que se presente nueva información en el sistema. Debido al carácter dinámico de estas redes, el estudio de estabilidad suele ser un aspecto fundamental.

5.3.2.1 Aprendizaje Supervisado

El aprendizaje supervisado se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo, que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor comprueba la salida de la red, y en caso de que ésta no coincida con la deseada, procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida obtenida se aproxime a la deseada. Hay tres formas de llevar a cabo este tipo de aprendizaje, el Aprendizaje por Corrección de Errores, el Aprendizaje por Refuerzo y el Aprendizaje Estocástico.

El Aprendizaje por Corrección de Errores consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores obtenidos de la salida de la red y los valores deseados, es decir, en función del

error obtenido en la salida. Un algoritmo simple de aprendizaje por corrección de error podría ser el siguiente:

$$\Delta w_{ab} = \alpha \cdot y_a (d_b - y_b) \qquad (5.3)$$

Donde, de acuerdo a la figura 5.7:

ΔW_{ab}: Variación del peso de la conexión entre las neuronas a y b.

y_a: valor de la salida de la neurona a

d_b: valor de la salida deseado para la neurona b

y_b: valor de la salida de la neurona b

α: factor de aprendizaje que regula el aprendizaje

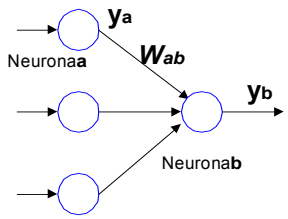


Figura 5.7 Esquema de conexión entre dos neuronas

El Aprendizaje por Refuerzo, es más lento que el anterior, ya que se basa en la idea de que no se conoce exactamente el comportamiento deseado, es decir, no se puede indicar qué salida se desea ante una determinada entrada. En este tipo de aprendizaje, la tarea del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida por la red se ajusta a la deseada o no (éxito = +1 o fracaso = -1), y en función de ello se ajustan los pesos. Se podría decir que en este tipo de aprendizaje, la función del supervisor se asemeja más a la de un crítico (que opina sobre la respuesta de la red), que a la de un maestro (que indica a la red la respuesta concreta que debe dar), como ocurriría en el caso de aprendizaje por corrección de error.

Y el Aprendizaje Estocástico, consiste en realizar cambios aleatorios en los pesos de las conexiones y evaluar su efecto a partir del objetivo deseado y

de distribuciones de probabilidad. En la tabla 5.1, se muestra una clasificación de los modelos de RNA con aprendizaje supervisado más importantes.

Tabla 5.1 Redes con aprendizaje Supervisado

TIPOS DE APRENDIZAJE SUPERVISADO		***MODELO DE RED***
POR CORRECCION DE ERROR	**OFF LINE**	PERCEPTRON ADALINE/MADALINE PERCEPTRON MULTICAPA BRAIN-STATE-IN-A BOX COUNTERPROPAGATION
POR REFUERZO	**ON LINE**	LINEAR REWARD PENALTY ADAPTIVE HEURISTIC CRITIC
ESTOCASTICO	**OFF LINE**	BOLTZMANN MACHINE CAUCHY MACHINE

5.3.2.2 Aprendizaje No Supervisado

Este tipo de aprendizaje, las redes no reciben ninguna información por parte del entorno que le indique si la salida generada en respuesta de una determinada entrada es correcta o no, por esto, suele decirse que estas redes son capaces de autoorganizarse. Existen varias posibilidades en cuanto a la interpretación de la salida de estas redes, que dependen de su estructura y del algoritmo de aprendizaje empleado. En algunos casos, la salida representa el grado de similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado hasta entonces. En otro caso, podría establecer categorías (*clustering*), indicando a la salida, a qué categoría pertenece la información suministrada, siendo la propia red quien debe encontrar las categorías apropiadas, a partir de correlaciones entre las informaciones presentadas. Una variación de esto es el prototipado, en donde la red obtiene ejemplares representantes de las clases a las que pertenecen las informaciones de entrada.

En cuanto a los algoritmos de aprendizaje no supervisado, en general se suelen considerar dos tipos, el Aprendizaje Hebbiano y el Aprendizaje Competitivo y Cooperativo. En el primer caso, generalmente se pretende medir

la familiaridad o extraer características de los datos de entrada, mientras que el segundo suele orientarse hacia la clusterización o clasificación de dichos datos.

El Aprendizaje Hebbiano consiste básicamente en el ajuste de los pesos de las conexiones, de acuerdo con la correlación de los valores de activación (salidas) de las dos neuronas conectadas, y la modificación de los pesos se realiza en función de los estados (salidas) de las neuronas, obtenidos luego de la presentación de ciertos estímulos, sin tener en cuenta si se deseaba o no, obtener dichos estados de activación.

En el Aprendizaje Competitivo y Cooperativo, las neuronas compiten y cooperan entre sí con el fin de llevar a cabo una tarea dada. Este tipo de aprendizaje, cuando se presenta cierta información de entrada, sólo una neurona, o un pequeño número de neuronas, se activa (alcanza su valor máximo), por lo tanto, las neuronas compiten por activarse, quedando finalmente una, o un grupo, como ganadora y obligando a las otras neuronas a bajar sus valores de respuestas a los mínimos. En la tabla 5.2 se muestra una clasificación de los modelos de RNA con aprendizaje no supervisado más importantes.

Tabla 5.2 Redes con aprendizaje No Supervisado

TIPO DE APRENDIZAJE NO SUPERVISADO		***MODELO DE RED***
HEBBIANO	**OFF LINE**	HOPFIELD LINEAR ASSOCIATIVE MEMORY TEMPORAL ASSOC. MEMORY
	ON LINE	ADDITIVE GROSSBERG SHUNTING GROSBERG BIDIRECCIONAL ASSOC. MEMORY
COOPERATIVO/ COMPETITIVO	**OFF LINE**	LEARNING VECTOR QUANTIZATION COGNITRON/NEOCOGNITRON
	ON LINE	ADAPTIVE RESONANCE THEORY

5.3.3 Tipos de Asociación entre la información de entrada y de salida

Además de la topología y el mecanismo de aprendizaje de una red neuronal, existen otros aspectos que las caracterizan, los cuales son, el tipo de asociación realizada entre la información de entrada y de salida, así como la representación de estas informaciones.

Existen dos formas básicas de realizar la asociación entre las informaciones de entrada y salida, la primera se denomina Heteroasociación y se refiere al caso en el que la red aprende parejas de datos *[(A1,B1), (A2,B2)...]*, es decir, cuando se presenta una información de entrada A1, deberá responder generando la correspondiente salida asociada B1. A la segunda forma se le conoce como Autoasociación y en este caso la red aprende cierto conjunto de datos de entrada *A1, A2,..., An*, así, cuando se le presenta una información de entrada realizará una Autocorrelación, con cada uno de los datos almacenados, respondiendo con aquel que tenga mayor grado de correlación al de la entrada. Estos dos mecanismos de asociación dan lugar a dos tipos de redes neuronales: Las redes Heteroasociativas y Las redes Autoasociativas

Con respecto a la representación de la información de entrada y de salida, en un gran número de redes, tanto los datos de entrada como los de salida son de naturaleza analógica, es decir, son valores reales continuos, generalmente normalizados y cuyo valor absoluto es menor que la unidad (datos fuzzy) [Hilera, Martínez. 1995]. Cuando esto ocurre, las funciones de activación de las neuronas serán también continuas, del tipo lineal o sigmoidal. Otras redes, por el contrario, sólo admiten valores discretos o binarios, generando también en la salida datos binarios. En este caso, las funciones de activación de las neuronas serán del tipo escalón o la función signo. Existe también un tipo de redes, que podrían denominarse híbridas, en las que las informaciones de entrada pueden ser valores continuos, aunque las salidas de la red son discretas.

5.3.4 Algunos modelos de Redes Neuronales

Los diferentes tipos de RNA que han sido desarrollados hasta este momento son denominados paradigmas o modelos. Se han elaborado gran cantidad de paradigmas, algunos más especializados que otros, pero la búsqueda del mejor paradigma representativo, es decir, el que emule verdaderamente el funcionamiento de una red neuronal biológica, está aún en camino. A continuación se citan algunos de los modelos de red neuronal más conocidos.

- El modelo de Mc Culloc Pitts
- El Perceptrón
- El Perceptrón Multicapa (MLP)
- Los modelos Adaline y Madeline
- El modelo de Hopfield
- Brain State in a Box
- Counterpropagation
- Neocognitron
- Teoría de resonancia Adaptiva (ART)
- Memoria Asociativa Lineal (LAM)
- Memoria Asociativa Lineal (BAM)
- Memoria Asociativa Temporal (TAM)
- Mapas Autoorganizativos (SOM)
- Redes Neuronales Probabilisticas (PNN)
- Funciones de Base Radial
- Redes Neuronales con Retardo en el Tiempo (TDNN)
- Maquina de Boltzman
- Maquina de Cauchy, entre otros.

5.4 El Perceptrón Multicapa y el Algoritmo Backpropagation

Entre los modelos de RNA que se han desarrollado hasta ahora, el Perceptrón Multicapa (MLP) ha sido elaborado y probado con éxito en la solución de varios tipos de problemas, especialmente, ha demostrado una gran eficiencia en solucionar problemas de clasificación de patrones, también puede solucionar problemas de predicción, y además se puede usar como una herramienta de transformación [Hilera, Martínez. 1995]. Como ya se mencionó anteriormente, el Reconocimiento de Palabras Aisladas es un problema de Reconocimiento de Patrones, donde cada palabra representa un patrón. El presente trabajo utiliza en la etapa de reconocimiento un subsistema basado en el uso del MLP, el cual será un clasificador de palabras. El algoritmo de aprendizaje que se usa sobre el MLP es el Backpropagation el cual permite aprender de manera supervisada las diferentes formas que podría tomar el vector de características (representación del patrón) de una palabra y elaborar una regla de clasificación adecuada.

5.4.1 El Perceptrón Multicapa

El MLP es un modelo de RNA estructurada en niveles, con una capa de entrada, una capa de salida y de una o más capas ocultas. Una forma de conceptualizar el MLP es el siguiente, “Un Perceptrón Multicapa esta compuesto por varios Perceptrones en una estructura jerárquica de capas conformando una topología *feedforward*” [Stamatios, Kartalopoulos. 1996].

Si se analiza el Perceptrón de una capa (ver figura 5.8), éste se encuentra formado por una estructura de varias salidas a partir de unidades básicas, para ello se ubican en paralelo tantas neuronas como salidas se requiere, todas ellas tendrán las mismas entradas y cada neurona dará como resultado una salida diferente porque tendrá diferentes pesos de conexión y umbrales.

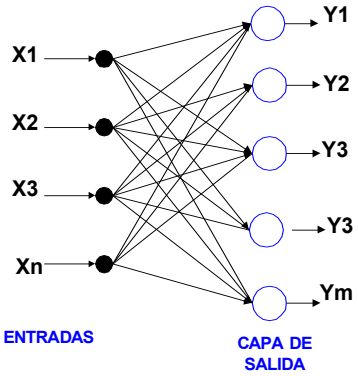


Figura 5.8 Perceptrón de una capa

El Perceptrón Multicapa consiste básicamente en poner varias capas elementales, como las descritas en el Perceptrón de una capa, interconectadas sucesivamente con el objeto de dotar a la red de la capacidad suficiente para realizar tareas mas complicadas (ver figura 5.9). Este modelo contiene una capa de entrada, una de salida y por lo menos una capa oculta. El número de nodos de la capa de entrada depende del número de entradas de la señal de estímulo, y el número de nodos de la capa de salida, está en función del número de salidas que tendrá la red. Con respecto al número de capas ocultas y la cantidad de neuronas por cada una de estas capas es relativo y por lo general estos parámetros dependen exclusivamente de la aplicación, y son establecidos por el diseñador tomando las limitaciones de tipo computacional.

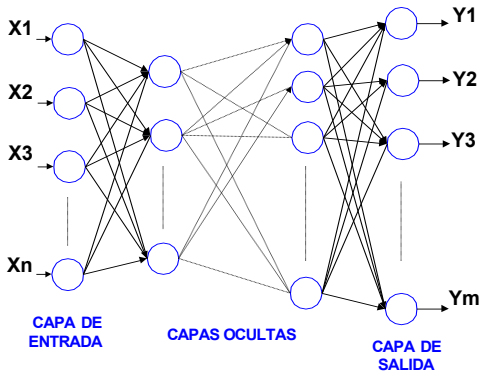


Figura 5.9 Perceptrón Multicapa

Una manera de comprender el comportamiento de las Redes Neuronales como el MLP, consiste en representar los patrones en un espacio n-dimensional (ver figura 3.17(b)). Como resultado del proceso de entrenamiento, la red establece regiones que agrupan patrones de una misma clase, y al conjunto de regiones establecidas se le denomina Mapa de Regiones de Decisión. Es mediante estas regiones que se determina la clase a la que pertenece un patrón que no ha participado en el entrenamiento.

Un Perceptrón elemental que consta de dos niveles (la de entrada con neuronas lineales y la de salida con función de activación escalón) solo puede establecer dos tipos de regiones separadas por una frontera lineal en el espacio multidimensional de patrones de entrada. Un Perceptrón con tres niveles puede formar cualquier región convexa en el espacio. Un Perceptrón de cuatro capas puede formar regiones de decisión arbitrariamente complejas. A medida que el numero de capas en un Perceptrón se incrementa, las regiones de decisión se vuelven más complejas, pero algunos autores [Hilera, Martínez. 1995] sugieren el uso de hasta una cuarta capa, como máximo, ya que un incremento del número de capas no produce mejoras significativas.

Se puede considerar, que en la actualidad el MLP es el modelo más frecuentemente utilizado y de manera más versátil. Dada esta amplitud de su estudio, se divide la utilización del MLP en tres grandes grupos, teniendo en cuenta su función.

El MLP actúa como clasificador, cuando la salida de la red es directamente una estimación de la probabilidad de pertenencia de la entrada a una determinada clase. Una de las tareas en las que ha demostrado un alto rendimiento es como generador de funciones discriminantes: la salida se puede considerar como la probabilidad de que la muestra pertenezca o no a una determinada clase.

El MLP actúa como predictor cuando, dada una secuencia de vectores de características correspondientes a los instantes de tiempo *(t-n), (t-n-1), …, (t-1),* la red trata de predecir el valor del vector de características en el instante siguiente, o sea *(t)*. La eficacia del MLP como discriminante ha sido de sobra demostrada, sin embargo, es conveniente recordar que para que las superficies de separación entre clases puedan ser perfectamente definidas, es necesario presentarle a la red, en entrenamiento, un conjunto de ejemplos de cada una lo suficientemente representativo.

Y finalmente cabe la posibilidad de usar el MLP como una herramienta de transformación sobre los datos entrada. Se podría afirmar que un caso más específico de transformación de datos de entrada es el uso del MLP en la compresión de datos.

5.4.2 Backpropagation

El algoritmo Backpropagation, también denominado algoritmo de Retropropagación del Error o Generalización de la Regla Delta [Hilera, Martínez. 1995], se caracteriza por ser de entrenamiento supervisado, y realiza la tarea de actualización de pesos, basándose en el Error Medio Cuadrático.

La importancia de una red neuronal con aprendizaje Backpropagation, radica en su capacidad de autoadaptar los pesos de las neuronas de las capas intermedias, para aprender la relación que existe entre un conjunto de patrones (vectores de entrada) dados como ejemplos y sus salidas correspondientes, con el fin de aplicar esa misma relación, después del entrenamiento, a nuevos

patrones con ruido o incompletos. Esta red debe encontrar una representación interna, que le permita generar salidas deseadas cuando se le dan entradas de entrenamiento, y que se pueda aplicar además a entradas no presentadas durante la etapa de aprendizaje, para clasificarlas según las características que compartan con los ejemplos de entrenamiento. Esta característica importante que se exige a los sistemas de aprendizaje, es la capacidad de generalización, entendidas como la facilidad de dar salidas satisfactorias, a entradas que el sistema no ha visto durante su fase de entrenamiento.

El algoritmo Backpropagation es un aprendizaje supervisado y por tanto, necesita un conjunto de entrenamiento que describa cada entrada y su valor de salida esperado de la siguiente forma:

$$(p_1,d_1),(p_2,d_2),(p_3,d_3),\ldots,(p_q,d_q) \tag{5.4}$$

Donde:
p_q: es una entrada a la red (patrón) y
d_q: es la correspondiente salida deseada para el patrón q-ésimo.

La idea central de este algoritmo, se encuentra en calcular los errores para las unidades de las capas ocultas, a partir de los errores de las unidades de la capa de salida, siendo propagados capa tras capa hacia atrás hasta llegar a la capa de entrada. El algoritmo debe ajustar los parámetros de la red para minimizar el error medio cuadrático entre la salida deseada y la salida real de la red.

Para realizar el entrenamiento de una red neuronal, se debe tener inicialmente definida la topología de la red, definir el número de neuronas en la capa de entrada, la cantidad de capas ocultas y el número de neuronas de cada una de ellas, el número de neuronas en la capa de la salida y también se debe definir las funciones de transferencia requeridas en cada capa. Basándose en la topología escogida, se asignan valores iniciales a cada uno de los parámetros que conforma la red.

El proceso de entrenamiento para el caso de especifico de un MLP con aprendizaje Backpropagation se muestra en la figura 5.10. en este se observa que el patrón de entrenamiento se propaga a través de la red, desde la capa de entrada hasta la de salida produciéndose una respuesta, denominada salida real. Esta se compara con el valor de salida deseada y el error producido se transmite hacia la capa anterior. Enseguida se calcula la contribución de cada neurona de esta capa al error en la salida. Se repite este proceso hasta alcanzar la capa de entrada. Por esto se afirma que el error en la red se Retropropaga.

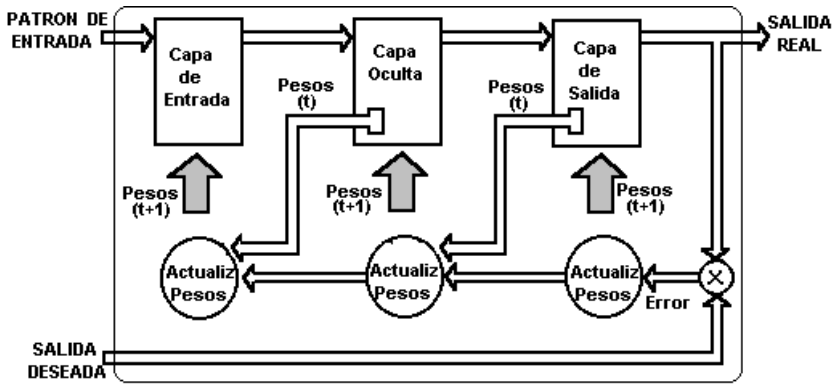


Figura 5.10 Esquema del efecto de Retropropagación del error

Los pesos de conexión entre dos capas consecutivas de la red se reajustan, basándose en el valor del error recibido, de manera que la siguiente vez que se propague el mismo patrón, la salida real esté más próxima a la deseada, y el error disminuya.

En este proceso de aprendizaje el error marca el camino más adecuado para la variación de los pesos de conexión, que al final del entrenamiento producirán una respuesta satisfactoria en la salida, para todos los patrones.

5.4.2.1 *Funcionamiento del algoritmo*

El método que sigue la Regla Delta Generalizada para ajustar los pesos radica en que éstos se actualizan en forma proporcional a la delta (δ),

producida en la salida como efecto de la comparación del valor de la salida real con el de la salida deseada.

Dada una neurona de la i-esima capa (cuyo valor de salida es yi), conectada a otra de la j-esima capa, el cambio que se produce en el peso de la conexión para un patrón de aprendizaje p determinado es:

$$\Delta w_{ji}(t+1) = \alpha \cdot \delta_{pj} \cdot y_{pi} \qquad (5.5)$$

En donde el subíndice **p** se refiere al patrón de aprendizaje de turno y α es una constante que expresa representa la tasa de aprendizaje.

Si la j-esima capa es la de salida, tal como se muestra en la figura 5.11(a), entonces el término delta (δ), está dado por la siguiente fórmula.

$$\delta_{pj} = (d_{pj} - y_{pi}).f(neta_j) \qquad (5.6)$$

En caso de una neurona de la j-esima capa que no sea la de salida (figura 5.11(b)), el error que se produce esta en función del error que se comete en las neuronas de la capa siguiente es decir la que recibe como entradas las salidas de la capa j-esima.

$$\delta_{pj} = (\sum_{k} \delta_{pk}.w_{kj}).f'(neta_j) \qquad (5.7)$$

Donde el rango k de la sumatoria cubre toda las neuronas a las que está conectada la salida de neurona de estudio. De esta forma, el error que se produce en una neurona oculta es la suma de los errores que se producen en las neuronas a las que esta conectada la salida de ésta, multiplicando cada una de ellas por el peso de la conexión.

(a) (b)

Figura 5.11 Conexión de una neurona de la capa i con una de la capa j

A continuación se presentan, a modo de síntesis, los pasos y fórmulas utilizadas durante el proceso de elaboración del algoritmo *Backpropagation*.

a) Inicializar los pesos de la red con valores pequeños aleatorios.
b) Presentar un patrón de entrada y especificar la salida que debe generar la red
c) Calcular la salida actual de la red, para ello se presenta las entradas de la red se calcula la salida de cada capa hasta llegar a la ultima (salida), las expresiones siguientes describen este proceso:

- Se calculan las entradas netas ($neta_h$) y las salidas (O_h) para las neuronas de la capa oculta:

$$neta_h = \sum_{i=1}^{N} w_{ai} . x_{pi} \qquad (5.8)$$

$$o_h = f_h(neta_h) \qquad (5.9)$$

Donde:
Wa: representa los pesos entre la capa de entrada y la oculta
h: refiere a la capa oculta
p: p-esimo vector de entrenamiento
i : iesimo elemento del vector de entrada

- Se calculan las entradas netas ($neta_o$) y las salidas (O_o) para las neuronas de la capa de salida

$$\text{neta}_o = \sum_{i=1}^{N} w_{bj} . y_{pj} \tag{5.10}$$

$$o_o = f_o(\text{neta}_o) \tag{5.11}$$

donde:
Wb: representa los pesos entre la capa de entrada y la oculta
o: se refiere a la capa de salida
j: j-esima neurona oculta

d) Calcular los términos de error para toda las neuronas. Se asume la que la k-esima es la capa de salida. El valor delta en esta capa se obtiene mediante la fórmula:

$$\delta_{ok} = (t_{ok} - o_{ok}).f_o'(\text{neta}_{ok}) \tag{5.12}$$

Donde:
f_o es la FT correspondiente a la capa de salida (función sigmoidal)
ok indica la k-esima neurona de salida

- Entonces los términos de error para las neuronas de salida quedan:

$$\delta_{ok} = (t_{ok} - o_{ok}).o_{ok}(1 - o_{ok}). \tag{5.13}$$

- Para el caso de las neuronas ocultas (j-esima capa), la expresión obtenida es:

$$\delta_{hj} = f_h'.(\text{neta}_{hj}) \sum_k \delta_{ok} w_b \tag{5.14}$$

- En particular para la función sigmoidal, como es el caso, se tiene:

$$\delta_{hj} = o_{hj}(1 - o_{ah}).\sum_{k} \delta_{ok} w_b \qquad (5.15)$$

e) Para la actualización de pesos se utiliza el algoritmo recursivo, comenzando por las neuronas de salida y trabajando hacia atrás hasta llegar a la capa de entrada ajustando los pesos de la forma siguiente:

- Para los pesos de las neuronas de la capa de salida:

$$w_b(t+1) = w_b(t) + \alpha.\delta_{ok}.o_{oj} \qquad (5.16)$$

- Y para los pesos de las neuronas de la capa oculta:

$$w_a(t+1) = w_a(t) + \alpha.\delta_{hj}.o_{ai} \qquad (5.17)$$

f) El proceso se repite hasta que el termino de error Ep, en la capa de salida, es muy cercano a cero en cada uno de los patrones aprendidos.

$$E_p = \frac{1}{2}\sum_{k=1}^{M} \delta_{ok}^2 \qquad (5.18)$$

5.4.3 Consideraciones del algoritmo de aprendizaje

El algoritmo de Backpropagation encuentra un valor mínimo de error, mediante la aplicación de pasos descendentes (gradiente descendiente). Cada punto de la función de error corresponde a un conjunto de valores de los pesos de la red (ver figura 5.12). Con el gradiente descendente, siempre que se realiza la actualización de pesos de la red, se asegura el descenso del valor de la función hasta encontrar el valor mínimo más cercano. Esto podría ocasionar muchas veces, que se alcance un mínimo local antes que el mínimo global de la función, y consecuentemente, haría que el proceso de aprendizaje se detenga antes de tiempo.

Por tanto, uno de los problemas que presenta este algoritmo de entrenamiento de redes multicapa es, que busca minimizar la función de error, pudiendo caer en un local o en algún punto estacionario, con lo cual no se llega a encontrar el mínimo global de la función de error. Sin embargo se debe tener en cuenta que para algunas aplicaciones no es necesario alcanzar el mínimo global, sino que puede ser suficiente con un error mínimo local preestablecido [Hilera, Martínez. 1995].

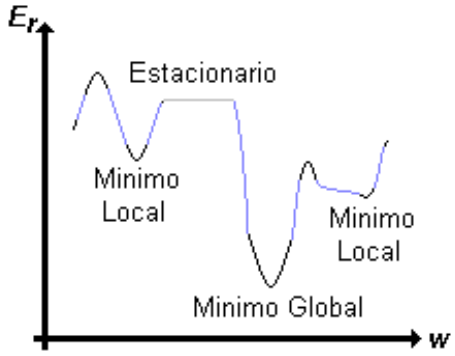


Figura 5.12 Función típica de error

- Control de Convergencia - En las técnicas de gradiente descendiente, como es el caso del algoritmo Backpropagation, es conveniente avanzar por la superficie de error con incrementos pequeños de los pesos; esto se debe, a que se tiene una información local de la superficie y no se sabe lo lejos o lo cerca que se está del punto mínimo. Por otro lado, con incrementos grandes, se corre el riesgo saltar el valor mínimo del error y estar oscilando alrededor de él, pero sin poder alcanzarlo. El parámetro que ajusta el valor del incremento es la taza de aprendizaje.

Es recomendable aumentar el valor de la tasa de aprendizaje a medida que disminuye el error de la red durante la fase de entrenamiento, para así garantizar, una rápida convergencia, teniendo la precaución de no tomar valores demasiado grandes que hagan que la red oscile alejándose demasiado del valor mínimo.

En el desarrollo matemático que se ha realizado para llegar al algoritmo Backpropagation, no se asegura en ningún momento que el mínimo que se encuentre sea global, una vez la red se asiente en un mínimo sea local o global cesa el aprendizaje. En todo caso, si la solución es admisible desde el punto de vista del error, no importa si el mínimo es local o global, o si se ha detenido en algún momento previo a alcanzar un verdadero mínimo.

- Dimensionamiento de la red - El número de nodos en la capa de entrada es igual al número de elementos de la señal de entrada, y el número de nodos de salida es el mismo número de salidas del sistema. No existe un método que indique de manera óptima cual debe ser el número de capas ocultas a utilizar, para una determinada aplicación, ni tampoco el número de nodos por cada capa oculta.

El número de nodos en las capas ocultas interviene en la eficacia del aprendizaje y de generalización de la red, y para determinar el número adecuado se debe ensayar con distintos números de neuronas, para representar la organización interna y elegir la más adecuada. Es posible eliminar neuronas ocultas si la red converge sin problemas, determinando el número final en función del rendimiento global del sistema. Por otro lado, si la red no converge es posible que sea necesario aumentar el número de nodos ocultos. Con respecto al número de capas, en general tres son suficientes, entrada, oculta y salida [Hilera, Martínez. 1995], sin embargo existen problemas que son más fáciles de resolver, con mas de una capa oculta.

- Inicialización y cambio de pesos - Sería ideal, para una rápida adaptación del sistema, durante el proceso de aprendizaje, inicializar los pesos con una combinación de valores muy cercano al punto mínimo del error buscado. Pero esto no es posible puesto que no se conoce a priori donde esta situado el punto mínimo. Es por eso que para inicializar los valores de los pesos se eligen valores aleatorios, es decir un punto cualquiera del espacio. Es conveniente que estos valores elegidos se sitúen dentro de un rango. El cual se elige de tal

modo que, el resultado de la suma de las entradas ponderadas por los pesos, es decir el valor de la Entrada Neta (ecuación 5.1), no sea muy grande y no sature la función de activación. Por esto usualmente se elige un rango entre [-0.5,0.5], o [0,1]. La modificación de los pesos puede realizarse cada vez que un patrón ha sido presentado, o bien, después de haber acumulado los cambios de los pesos en un número de iteraciones, el momento adecuado para el cambio de los pesos depende de la aplicación.

6. IMPLEMENTACIÓN DEL SISTEMA

Tal como se vio en el capítulo 1, el Sistema de Reconocimiento de Voz desarrollado consta de dos etapas: Hardware y Software. La etapa de Hardware está conformada por la Adquisición de la señal y el Procesador del sistema, que para nuestro caso está conformado por una Computadora Personal Pentium. En lo que respecta a la etapa de software se ha diseñado una interface gráfica, la cual ha sido denominada Reconocedor y Analizador de Voz (RAV), y tiene como función el desarrollo de los algoritmos propuestos, además de funcionar como un instrumento virtual especializado en Señales de Voz, que permite analizar sus características, así como analizar y determinar otros parámetro de interés.

A continuación se describirá en detalle las etapas de implementación de nuestro trabajo, así en el ítem 6.1 se explicara el Hardware, en el ítem 6.2 los algoritmos propuestos, para que finalmente en el ítem 6.3 se describa el RAV propiamente dicho.

6.1 Descripción del Hardware

El sistema ha sido implementado en una computadora personal Pentium de 100MHz. Se utiliza un micrófono dinámico AIWA 600Ω de Impedancia. Para la digitalización de la señal de voz se utiliza la tarjeta de Sonido Creative Labs Sound Blaster 16 AWE64, la cual ha sido programada para una adquisición en Modo Directo, a una resolución de 8 bits y frecuencia de muestreo de aproximadamente 11KHz. En la figura 6.1 se muestra un esquema del hardware del sistema.

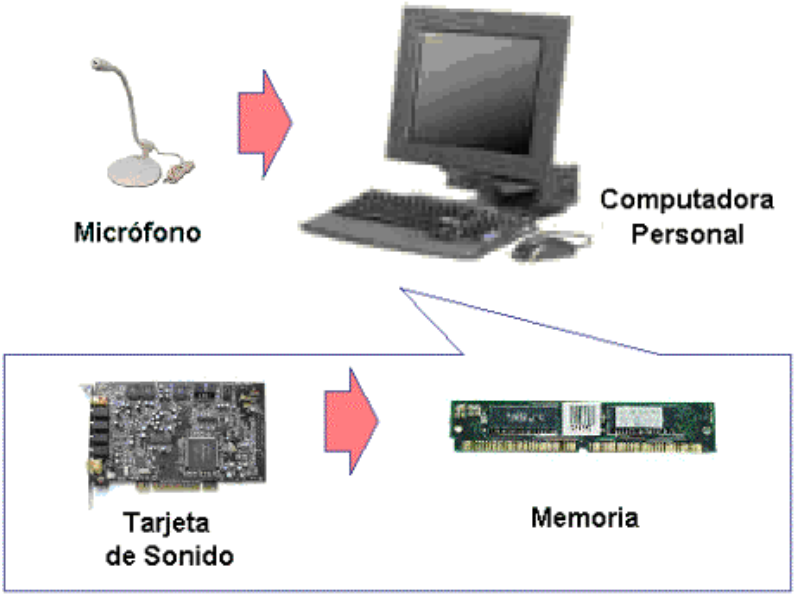


Figura 6.1 Etapas del Sistema de Adquisición

6.1.1 La tarjeta de sonido Sound Blaster

En la figura 6.2 se muestra el diagrama de bloques de las tarjetas de sonido pertenecientes a la familia Sound Blaster 16 de Creative Labs.

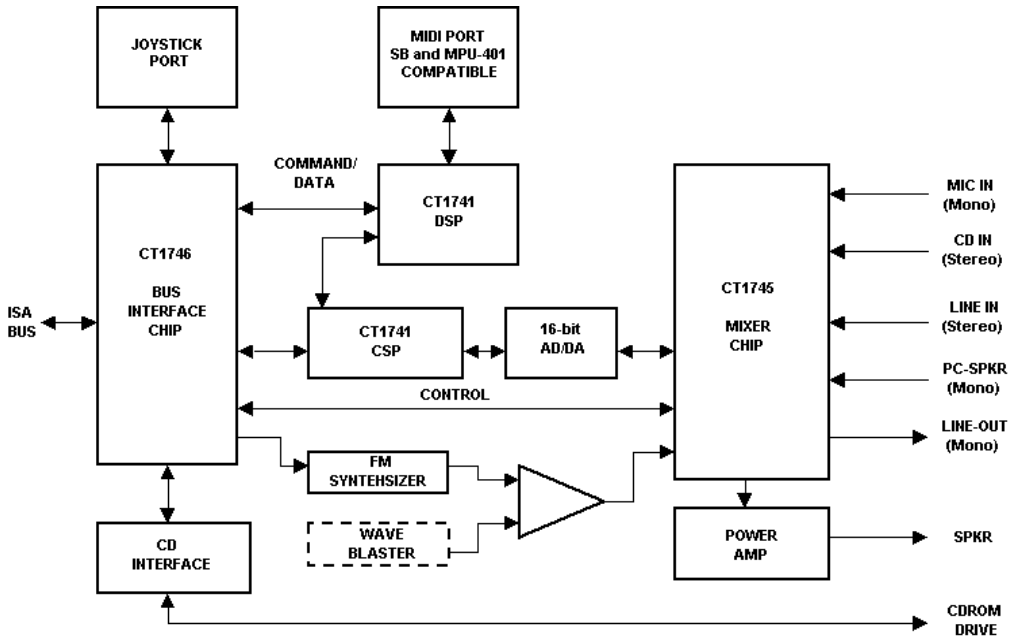


Figura 6.2 Diagrama de bloques de la tarjeta de sonido

Entres sus bloques más importantes se encuentran el DSP y el circuito Mezclador CT1745, los cuales han sido utilizados para realizar la adquisición de la señal de entrada.

6.1.2 Programación del DSP

El DSP de la tarjeta Sound Blaster se programa a través de 4 puertos o direcciones E/S, que se calculan sumando el valor de offset que se indica en la Tabla 6.1, a la dirección base que este configurada la tarjeta. Por defecto la dirección base es 220h. Si la dirección base fuera 240h los puertos serían, respectivamente, 246h, 24Ah, 24Ch y 24Eh. Si fuera 260h serían 266h, 26Ah, etc. Para simplificar a la dirección base se le denotará 2x0h.

Tabla 6.1 Puertos del DSP

OFFSET	DIRECCION	TIPO	FUNCION
6h	2x6h	Sólo escritura.	Inicializa el DSP a su estado por escritura defecto. Puerto de Inicialización.
Ah	2xAh	Sólo lectura	Puerto de Lectura de datos del DSP.
Ch	2xCh	Escritura	Usado para mandar datos y comandos al DSP. Puerto de Escritura de datos y comandos. Informa si el DSP está listo para recibir datos o comandos.
		Lectura	
Eh	2xEh	Sólo Lectura	Informa el estado del buffer de lectura de datos. Nos indica si hay datos para leer desde el DSP. Puerto de Disponibilidad de datos.

6.1.2.1 Inicializar el DSP - Subrutina IniSound

Es necesario realizar una rutina de inicialización antes de empezar a trabajar con el DSP[Creative,1996], para ello se debe seguir los siguientes pasos:

1. Escribir 01h en el puerto 2x6h.
2. Esperar al menos 3 microsegundos.

3. Escribir 00h en el puerto 2x6h.
4. Leer del puerto 2xEh para ver si es posible leer los diferentes datos.
5. Leer el puerto 2xAh, si se obtiene el valor 0AAh la inicialización ha sido correcta, caso contrario, se ha producido algún error y se debe finalizar indicándolo.

La figura 6.3 muestra la subrutina IniSound, la cual se encarga de realizar el proceso de inicialización del DSP.

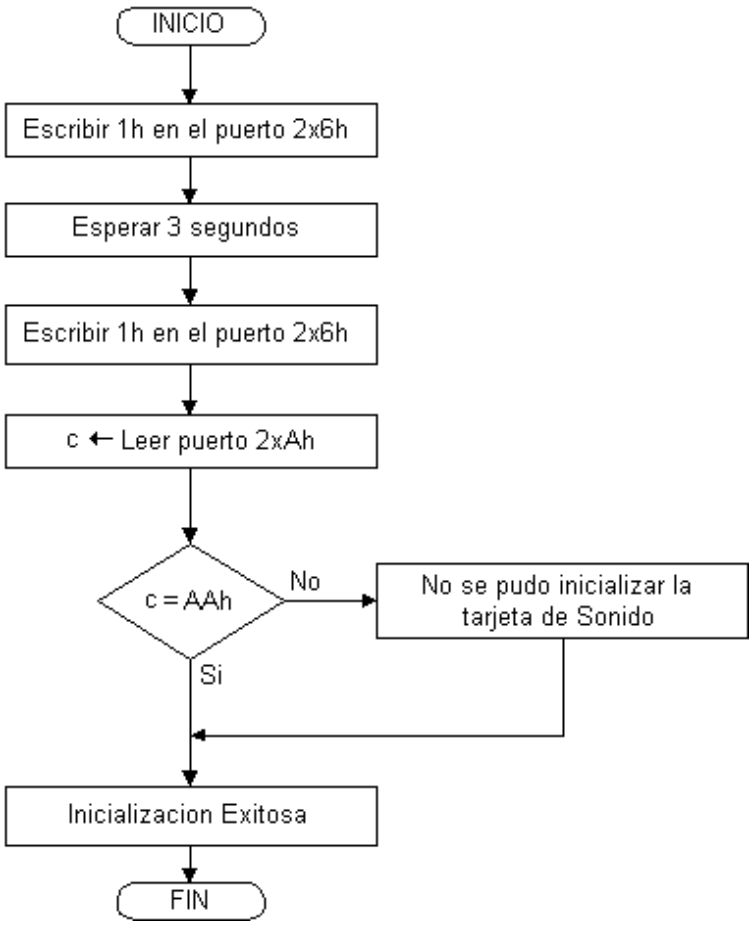


Figura 6.3 SubRutina IniSound

6.1.2.2 Leer del DSP - Subrutina LeerDsp

Antes de leer el puerto de lectura 2xAh, se debe comprobar si el DSP está listo para enviar datos, por lo que hay que comprobar el bit 7 del puerto 2xEh. Si hay datos para leer, el bit 7 está a 1, de lo contrario estará a 0. El

proceso ha sido implementado en la subrutina LeerDsp, cuyo diagrama de flujo se muestra en la figura 6.4, en el cual se utiliza la variable DatoFlag para comprobar el estado del bit 7 del puerto 2xEh, si éste bit es 1, se almacena el valor del puerto de lectura 2xAh en la variable Dato, caso contrario, se vuelve a realizar la comprobación hasta que se cumpla la condición necesaria. Se debe notar que al finalizar la subrutina esta retorna como resultado la variable Dato, en lo sucesivo, se utilizará esta nomenclatura para indicar que una subrutina devuelve un determinado valor.

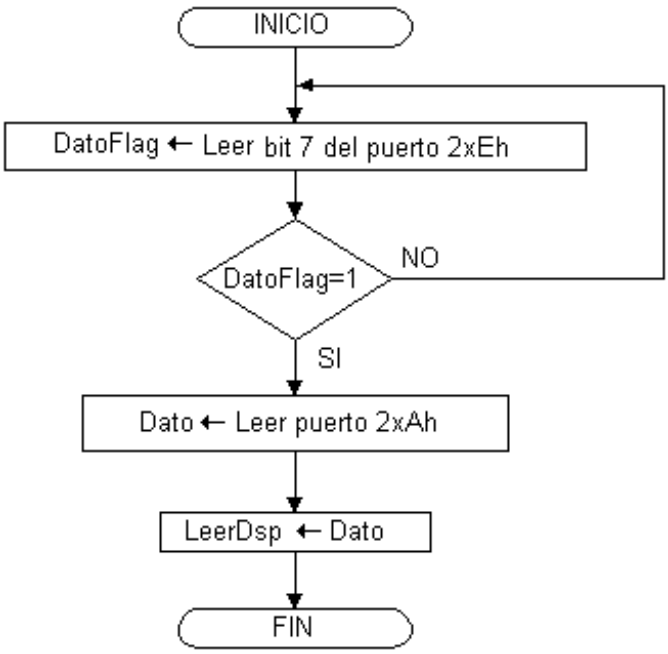


Figura 6.4 Subrutina LeerDsp

6.1.2.3 Escribir al DSP - Subrutina EscribirDsp

Antes de escribir sobre el puerto 2xCh, primero se debe comprobar si es posible hacerlo, para ello se lee el estado de su bit 7, si está a 0, el DSP está listo para recibir datos o comandos. La figura 6.5, muestra el diagrama de flujo de la subrutina que ha sido denominada EscrbirDsp, donde la variable Valor, contiene el dato que será escrito en el DSP y la variable DatoFlag se utiliza para comprobar el estado del bit 7 del puerto 2xCh.

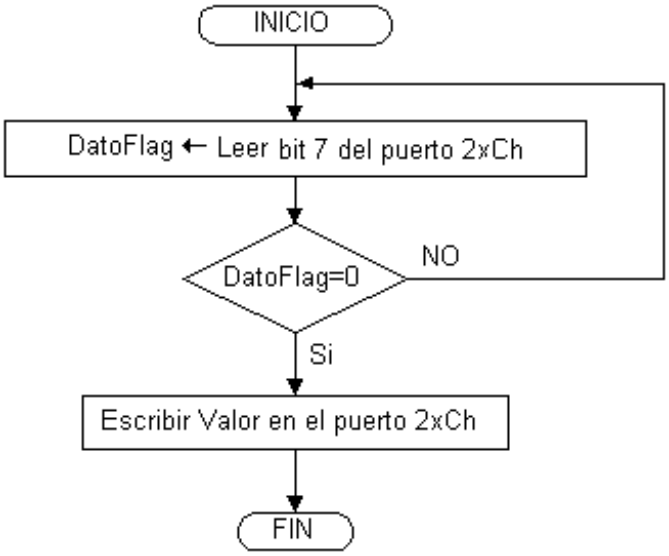


Figura 6.5 Subrutina EscribirDsp

6.1.2.4 Adquisición en Modo Directo - Subrutina Adquisición

Este modo de operación proporciona una adquisición a una resolución de 8 bits y la frecuencia de muestreo debe ser controlada por el programador. En este modo la adquisición se realiza sin intervención del DMA de la computadora. Los pasos a seguir son los siguientes:

1. Escribir 20h en 2xCh. (Se llama a la subrutina EscribirDsp)
2. Leer el dato de 8 bits de 2xAh (Se llama a la subrutina LeerDsp)
3. Esperar el tiempo apropiado. (Para establecer la frecuencia de muestreo)

Repetir los pasos 1 al 3 hasta terminar la adquisición.

Este proceso se ha implementado en la subrutina Adquisición, cuyo diagrama de flujo se aprecia en la figura 6.6.

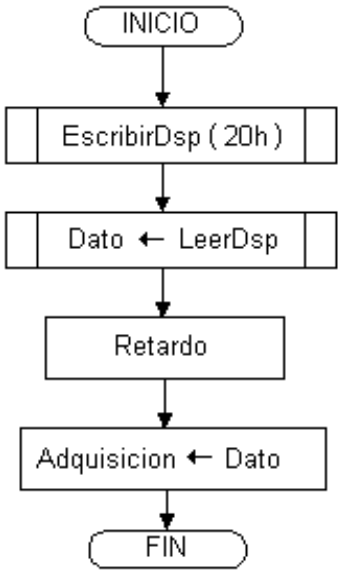


Figura 6.6 Subrutina Adquisición

6.1.3 Programación del Circuito Mezclador - Subrutina Mixer

El circuito mezclador usa dos puertos E/S consecutivos: 2x4h y 2x5h donde x depende de la dirección base E/S seleccionada. El puerto 2x4h es de sólo lectura y es denominado el Puerto de Direccionamiento. El Puerto 2x5h es tanto de lectura y escritura y es denominado Puerto de Datos.

La secuencia de programación del Circuito Mezclador es la siguiente:

1. Escribir el índice del registro del mezclador al puerto de Direccionamiento.
2. Escribir/Leer el valor del registro del mezclador hacia/desde el puerto de Datos.

La programación del circuito Mezclador ha sido implementada en la subrutina Mixer. En la figura 6.7, se muestra el diagrama de flujo para realizar este proceso, los parámetros de entrada son el índice del registro (Index) y el valor a establecer en el registro (Setting). La variable Oper se utiliza para establecer la escritura o lectura del mezclador. Para mayores detalles referirse al APENDICE X-X.

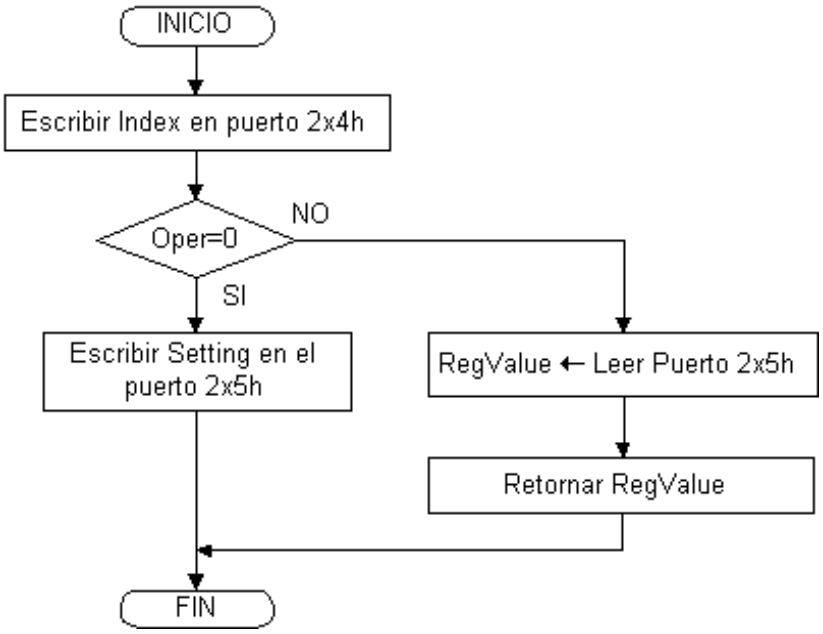


Figura 6.7 Subrutina Mixer

6.2. Descripción de los Algoritmos

En esta sección de describe los algoritmos correspondientes para implementar el Sistema de Reconocimiento de Voz. En la figura 6.8 se muestra el diagrama de bloques.

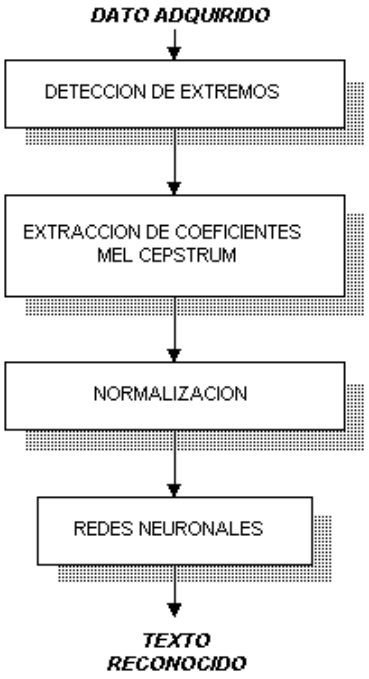


Figura 6.8 Diagrama de Flujo del Sistema de Reconocimiento

Tal como se aprecia en la esta figura, el sistema consta de las siguientes etapas:

- Detección de Extremos - Esta etapa realiza la detección de inicio y fin de una palabra (ver ítem 3.2.2), la cual a sido implementada en la Librería de Enlace Dinámico "Detector.dll".

- Extracción de Coeficientes Mel Cepstrum - En esta etapa se obtiene los coeficientes Mel Cepstrum de la palabra adquirida (ver Capitulo 4), obteniendo como resultado el vector de características, se ha sido implementada en la librería denominada "MelCep.dll".

- Normalización - En esta etapa se realiza la normalización del numero de elementos del vector de características a un numero preestablecido(ver ítem 3.2.4.1). Esto se realiza mediante la librería "Normaliza.dll".

- Redes Neuronales - Esta es la etapa en la cual se realiza la clasificación(Ver Capítulo 5) del Vector de Características Normalizado, y se identifica la palabra a la que pertenece mediante la librería "RNA.dll".

En los siguientes ítems se describe en forma detallada cada uno de los algoritmos utilizados en la implementación de estas librerías de enlace dinámico.

6.2.1 Detector de Extremos - Detector.dll

El Detector Automático de Extremos implementado, se basa en el análisis de la evolución del parámetro COPER en el tiempo (Ver ítem 3.2.2.1), para esto, permanentemente se adquiere un conjunto de tramas de longitud L equivalente a 100 muestras (aprox. 9 ms, para una frecuencia de muestreo de 11KHz) de la señal proveniente del micrófono, y se obtiene el parámetro COPER de éstas, el cual es comparado con los niveles de umbral predefinidos (CP_{UI} y CP_{UF}), para así determinar los instantes de inicio y final de la pronunciación de una palabra. Las muestras adquiridas son almacenadas temporalmente en un buffer de memoria de tamaño fijo.

Se ha determinado que una palabra pronunciada, normalmente, tiene una duración no mayor a 1.8 seg, entonces al utilizar una frecuencia de muestreo de 11KHz, los datos adquiridos no ocupan más de 20000 muestras. Por lo tanto, se establece el buffer con un tamaño 25000 elementos para asegurar que este pueda contener una palabra completa.

Cuando es detectado el inicio de pronunciación, los datos empiezan a almacenarse secuencialmente en el buffer de memoria, en tramas de longitud de L muestras, hasta que sea detectado el final de la misma. La figura 6.9 ilustra este proceso, donde el final de pronunciación se produce luego de haber adquirido 144L tramas, es decir, 14400 muestras, valor con el que se crea un vector dinámico, en el cual se almacena la palabra delimitada.

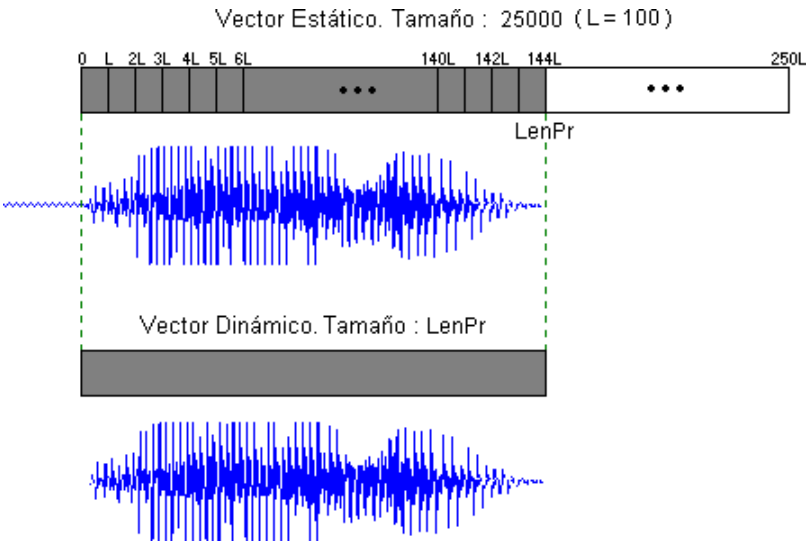


Figura 6.9 Detección de Extremos

En la figura 6.10 se muestra el diagrama de flujo general del Detector de Extremos. Básicamente consiste en la detección del inicio de pronunciación mediante el proceso DetectorInicio, seguidamente se determina el fin de pronunciación con el proceso DetectorFin, y finalmente se retorna un vector dinámico **Palabra[ ]**, el cual contiene la palabra delimitada.

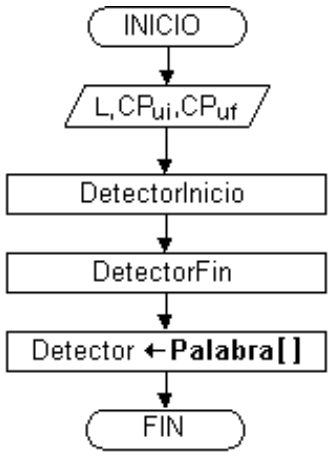


Figura 6.10 Proceso de Detección de Extremos

En los siguientes ítems se detalla tanto el algoritmo utilizado para detección de inicio de pronunciación, así como el de final de pronunciación. Previamente se muestra el algoritmo utilizado para determinar el parámetro COPER de las L muestras adquiridas.

6.2.1.1 El Parámetro COPER - Subrutina COPER

La figura 6.11 muestra el diagrama de flujo de la subrutina COPER, que tiene como función adquirir L muestras (100) de la señal proveniente del micrófono, las cuales son almacenadas en el vector **Dato[ ]**, y seguidamente obtener el parámetro COPER de éstas muestras. La variable 'n' indica la posición a partir de la cual se almacenarán las L muestras adquiridas en el buffer **Dato[ ]**. La subrutina retorna la variable CP, la cual contiene el parámetro COPER de las L muestras adquiridas.

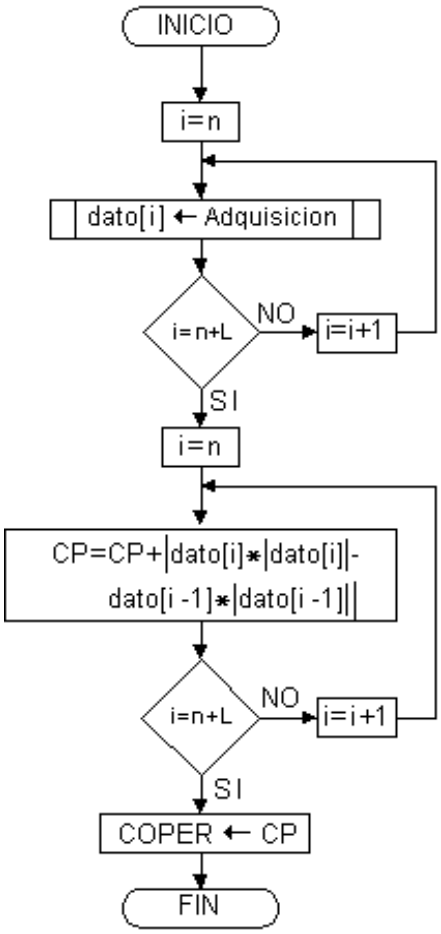


Figura 6.11 Subrutina COPER

6.2.1.2 Detector de Inicio de Pronunciación

Para la detección de inicio de pronunciación permanentemente se adquieren tramas de L muestras y se almacenan a partir del primer elemento del buffer **Dato[ ]**, concluida la adquisición, se determina su parámetro, si el umbral no es superado, nuevamente se vuelven a adquirir la trama siguiente a partir del primer elemento del buffer, sobrescribiendo los datos anteriormente almacenados. En el caso de que el umbral sea superado, es decir, se detecte inicio de pronunciación, se procede a almacenar los datos en el resto del buffer hasta que se detecte fin de pronunciación. (Ver figura 6.12).

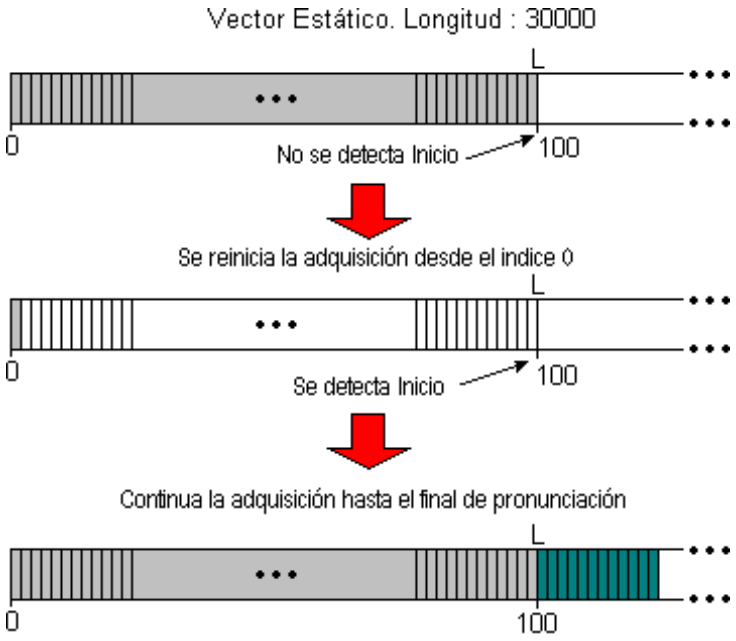


Figura 6.12 Detección de Inicio: Buffer de Memoria

En la figura 6.13 se muestra el diagrama de flujo del proceso, en el cual se obtiene el parámetro COPER de L muestras (CP), y luego se compara con el umbral de inicio(CP_{UI}).

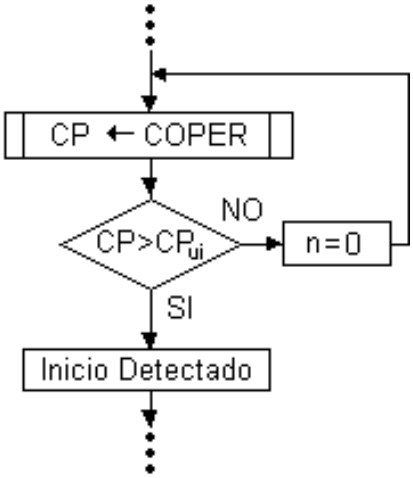


Figura 6.13 Subrutina DetectorInicio

6.2.1.3 Detector de Final de pronunciación

La detección de final de pronunciación se activa inmediatamente después de la detección de inicio de palabra. La figura 6.14 muestra su diagrama de flujo. Tal como se vio en el ítem 3.2.2, en el buffer(**dato[ ]**) se almacenan tramas consecutivas de L muestras, luego de la adquisición de cada trama, se analiza su parámetro COPER, el cual es almacenado en la variables CP, y que es comparado con el nivel de umbral CP_{uf}, hasta que se cumpla la condición de fin de palabra. Como se mencionó en el ítem 3.2.2.3, para que la condición de fin de pronunciación se cumpla, un número preestablecido de tramas consecutivas denominado nVent, deben poseer características de final de pronunciación. Por esto luego de detectarse un final de pronunciación, existe una variable t, que cuenta el número de tramas consecutivas que han cumplido con la condición de fin de pronunciación, luego este valor se compara con la variable nVent, si son iguales, significa que se ha detectado el fin de pronunciación es detectado; en caso contrario, se continúa con la adquisición. Una vez detectado el final de pronunciación, la variable n contiene el número de muestras adquiridas, valor con el cual se crea un vector dinámico **Palabra[ ]**, en el cual se almacena la palabra delimitada.

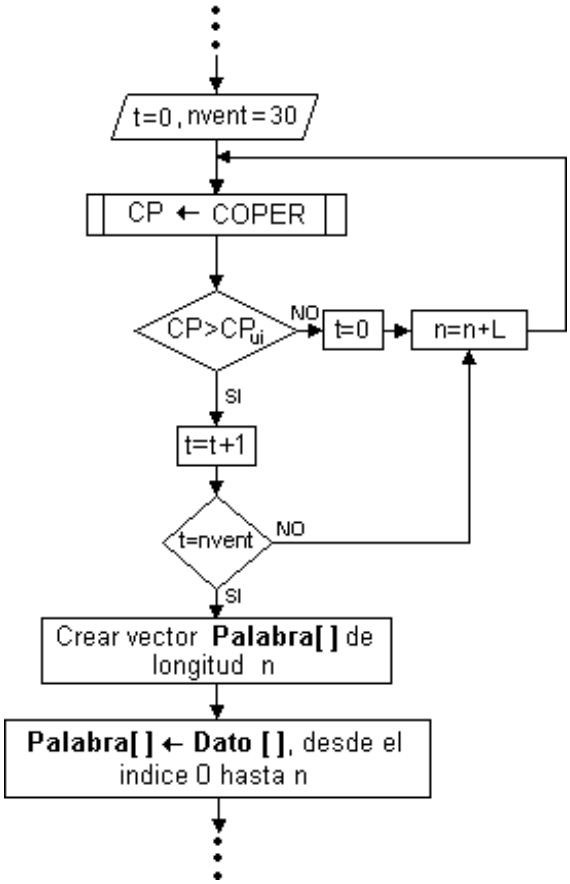


Figura 6.14 Subrutina DetectorFin

6.2.1.4 Diagrama de flujo completo

A continuación, en la figura 6.15, se muestra el diagrama de flujo completo del detector automático de extremos.

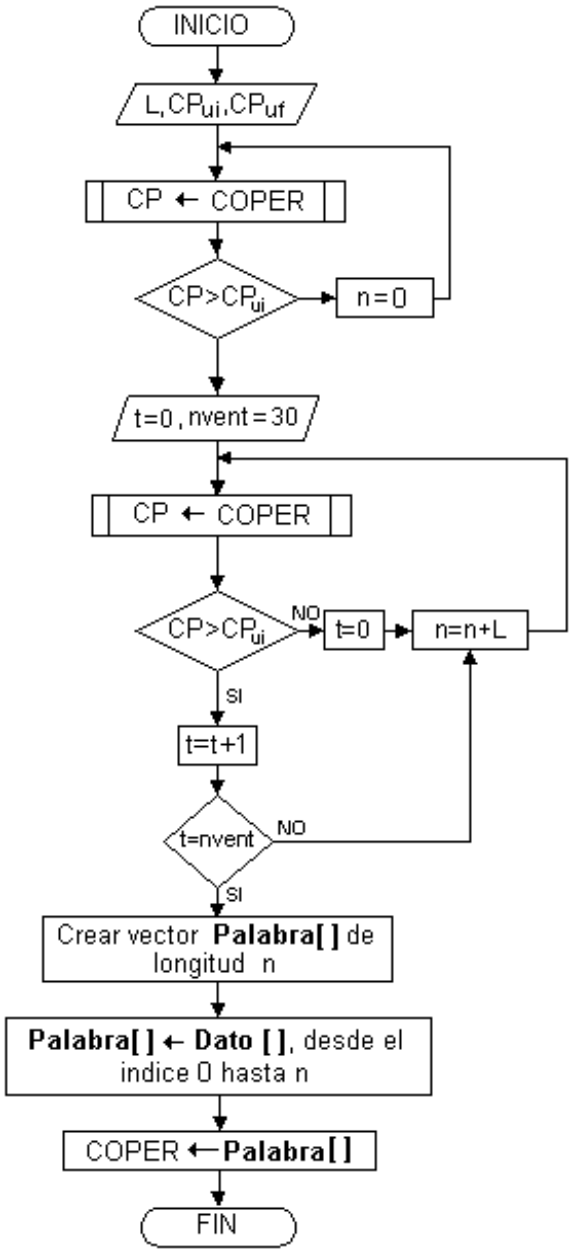


Figura 6.15 Diagrama de Flujo Completo

6.2.2 Coeficientes Mel Cepstrum - MelCep.dll

En la figura 6.16 se muestra el diagrama de bloques de la librería *Melcep.dll*, la cual tiene como objetivo obtener los coeficientes Mel-Cepstrum, para este fin, se analiza la palabra delimitada, almacenada previamente en el vector **Palabra[ ]**, obtenido en la etapa de detección de extremos. El análisis del vector se realiza en tramas de longitud N, en pasos de M muestras (Ver ítem 4.1). El proceso de análisis se detiene, cuando se supera el número de muestras que posee la palabra delimitada.

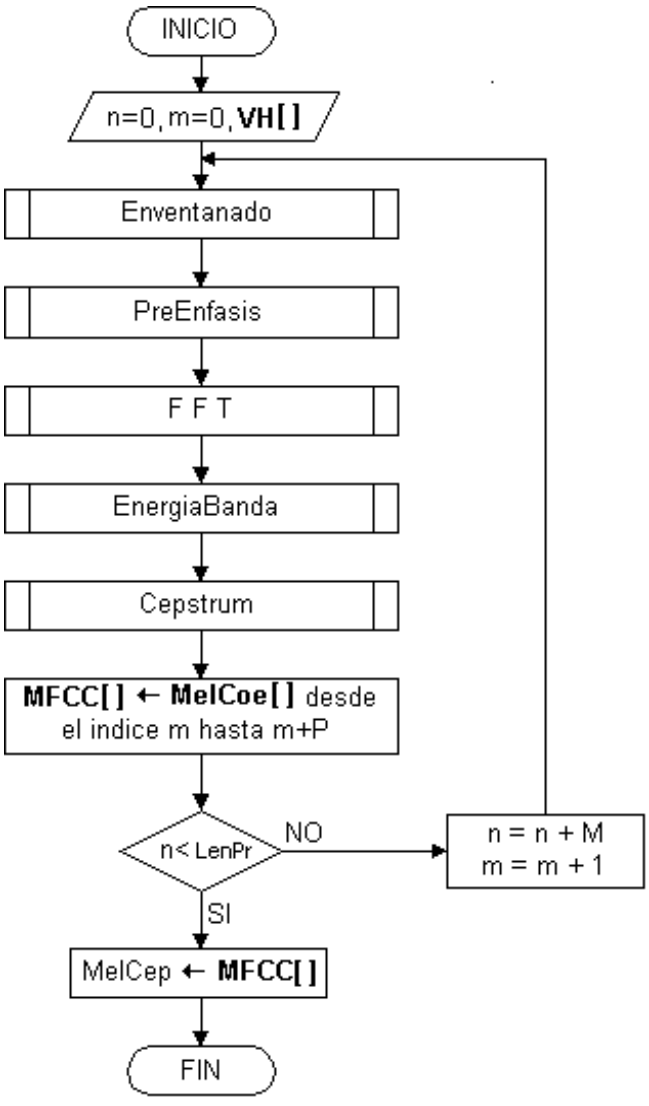


Figura 6.16 Subrutina MelCep

Tal como se puede ver en la figura 6.16, la librería *MelCep.dll* posee las siguientes subrutinas:

- Enventanado - Enventanado de la trama en análisis. (Ver ítem 4.2)
- PreEnfasis - Filtro de preeenfasis. (Ver ítem 4.3)
- FFT - Transformada Rápida de Fourier. (Ver ítem 4.4)
- EnergiaBanda - Energía en cada banda del banco de filtros. (Ver ítem 4.5)
- Cepstrum - Obtiene el Cepstrum de las concentraciones de energía en cada banda. (Ver ítem 4.6)

A continuación se describirá en detalle cada una de estas subrutinas.

6.2.2.1 Subrutina Enventanado

Tiene como función, multiplicar termino a termino, la trama de análisis (vector **Palabra[ ],** desde el índice n hasta n+N) por una ventana de Hamming de N elementos (vector **VH[ ]**), dando como resultado el vector **WinSec[ ]**. El diagrama de flujo se muestra en la figura 6.17.

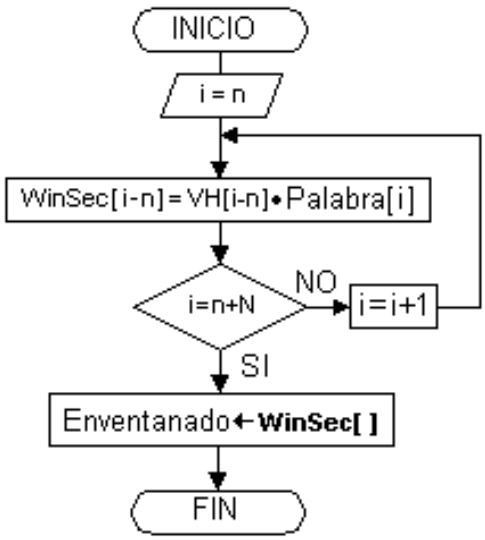


Figura 6.17 Subrutina Enventanado

6.2.2.2 Subrutina PreEnfasis

Esta subrutina realiza un filtrado de preenfasis a la secuencia **Winsec[ ],** obtenida luego del enventanado. La secuencia filtrada es almacenada en el vector **WinSecPre[ ]**. El diagrama de flujo se muestra en la figura 6.18.

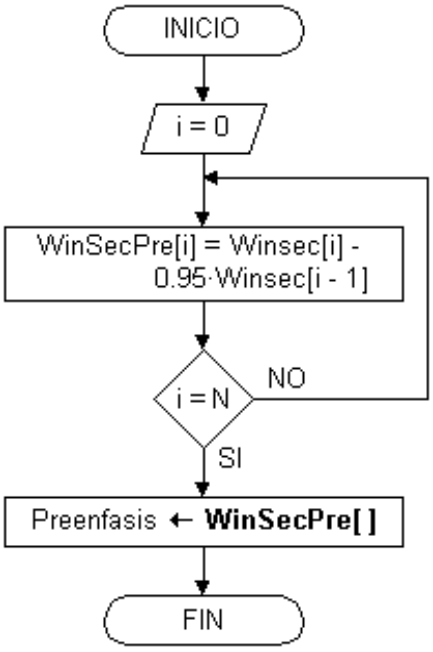


Figura 6.18 Subrutina Preenfasis.

6.2.2.3 La Transformada Rápida de Fourier - Subrutina FFT

La FFT es utilizada como una herramienta, y la explicación detallada del algoritmo utilizado está fuera de los objetivos del presente trabajo. La subrutina utilizada ha sido descarga desde una dirección de Internet, donde el código se distribuye gratuitamente. Para mayores referencia se puede consultar las referencias bibliograficas [Proakis, Manolakis. 1998] [Ingle, Proakis. 1997] [Takuya. 1998].

Luego de aplicar la FFT a la secuencia **WinSecPre[ ],** obtenida luego del preenfasis, se tiene como resultado el espectro de la trama en análisis que es almacenado en el vector **Espec[ ]**.

6.2.2.4 Subrutina EnergiaBanda

El objetivo de esta subrutina es de procesar el espectro de cada trama, a través de un banco de 20 filtros triangulares, y obtener la Energía de la salida de cada banda. El diagrama de flujo se muestra en la figura 6.19.

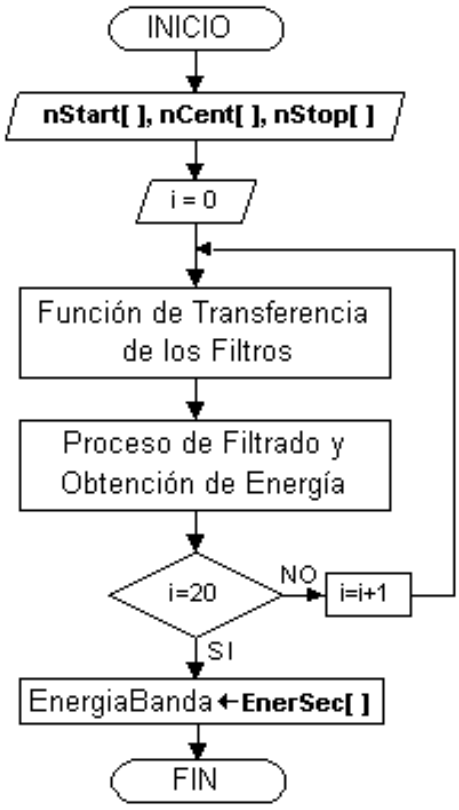


Figura 6.19 Subrutina EnergiaBanda

Según como se vio en el ítem 4.5.1, en primer lugar, se deben inicializar los índices de cada uno de los filtros triangulares (**nStart[ ]**, **nCent[ ]**, **nStop[ ]**), para así, obtener su función de transferencia. Seguidamente se realiza el proceso de filtrado y obtención de la Energía en cada banda. A continuación se describirán estos procesos.

6.2.2.5 Función de Transferencia de los Filtros

Una vez determinados los índices de los filtros, éstos son utilizados para obtener la función de transferencia de cada filtro. En la figura 6.20 se muestra el diagrama de flujo, en el cual se implementa la función de transferencia del filtro triangular (Ec. 4.13).

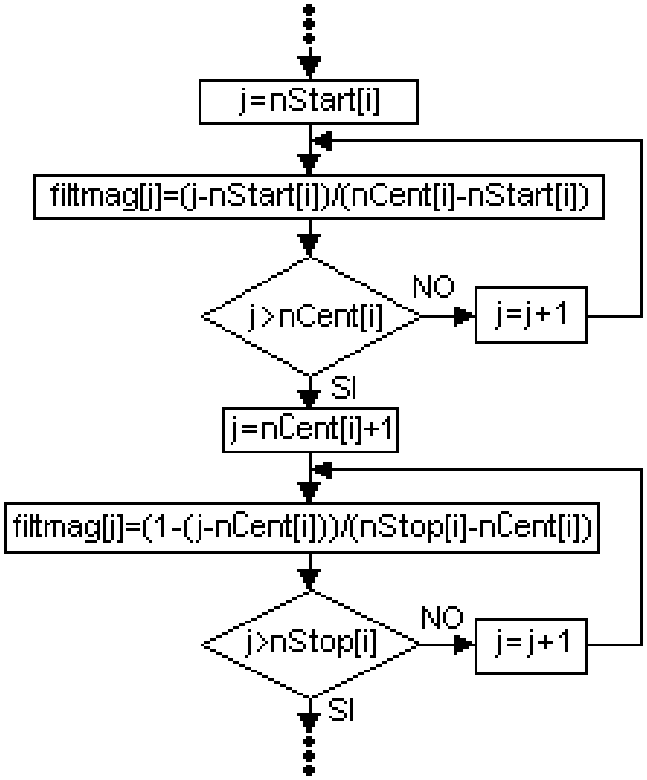


Figura 6.20 Filtro Triangular.

6.2.2.6 Proceso de Filtrado y obtención de Energía

Finalmente, se realiza el proceso de filtrado y se obtiene la Energía de la salida de cada banda. El diagrama de flujo se muestra a continuación, en la figura 6.21

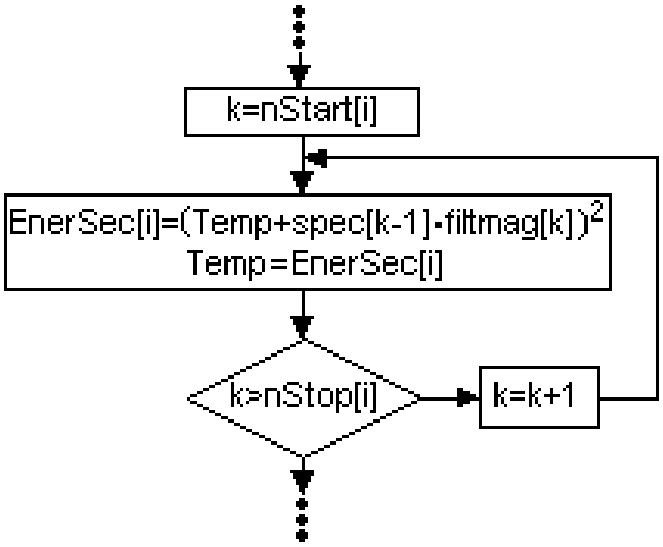


Figura 6.21 Energía en cada banda.

En la Figura 6.22 se muestra el diagrama de bloques completo de la subrutina *MelCep.dll*.

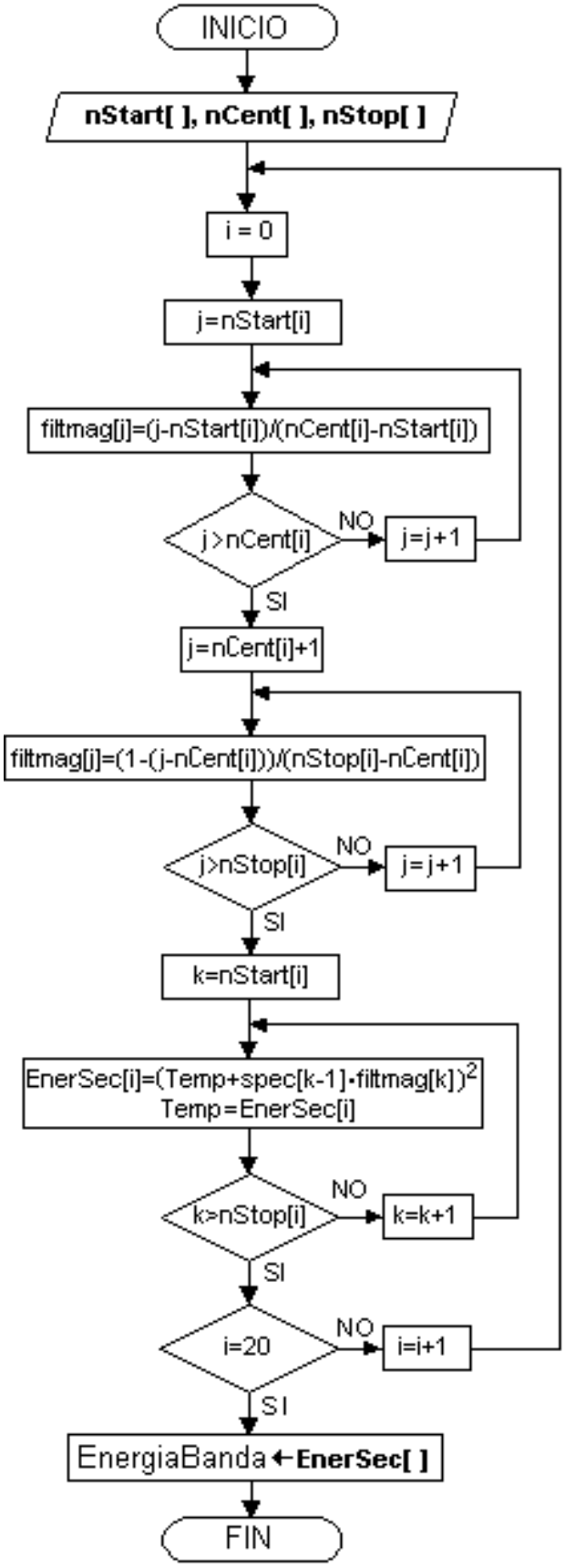


Figura 6.22 Diagrama de Flujo Completo.

6.2.2.7 Subrutina Cepstrum

Finalmente, para obtener los Coeficientes Mel-Cepstrum, se implementa la ec. (4.30). El diagrama de flujo para implementar esta ecuación se muestra en la figura 6.23, donde NF es el número de filtros utilizados y P es la cantidad de coeficientes a obtener.

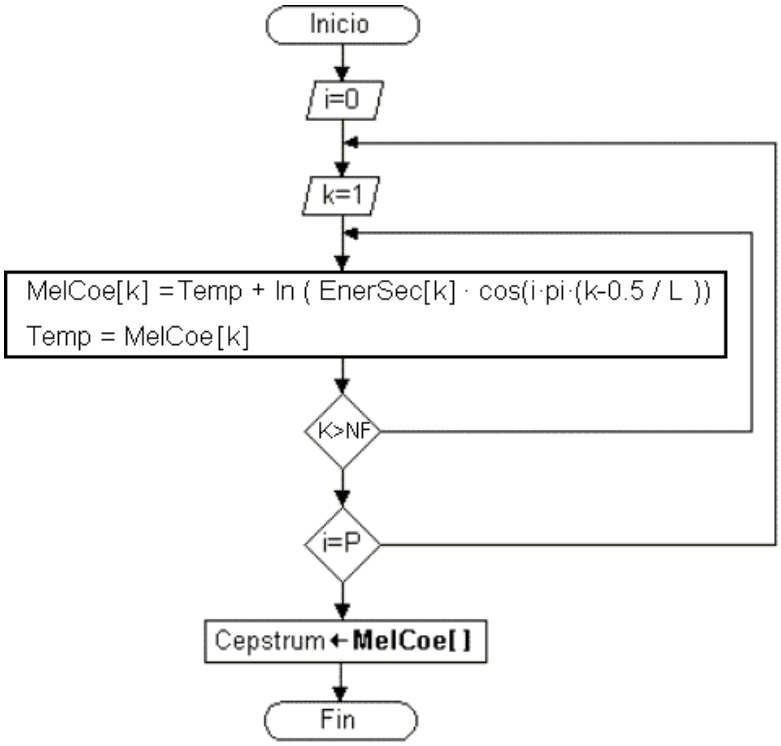


Figura 6.23 Subrutina Cepstrum.

6.2.3 Normalización - Normaliza.dll

Como se vió en el Capítulo 3, es poco probable que una persona pueda volver a pronunciar una palabra con igual duración. Además, se sabe que el número de elementos del vector que devuelve la Etapa de Extracción de Características, está relacionado directamente con el tiempo de duración de una palabra, produciéndose así vectores de distintos tamaños, lo cual no es compatible con la etapa de Reconocimiento, que exige que el tamaño del vector sea constante, sin importar el tiempo de duración de la palabra. Por esto se construye la librería *Normaliza.dll*, como una etapa intermedia entre la Extracción de Características y el Reconocimiento, que se encarga de expandir o comprimir el número de elementos del vector sin perder de manera considerable la información contenida en la secuencia original, a un valor preestablecido de 100 elementos (véase fundamento en el capítulo 7). La figura 6.24, muestra el diagrama de flujo que resume este proceso.

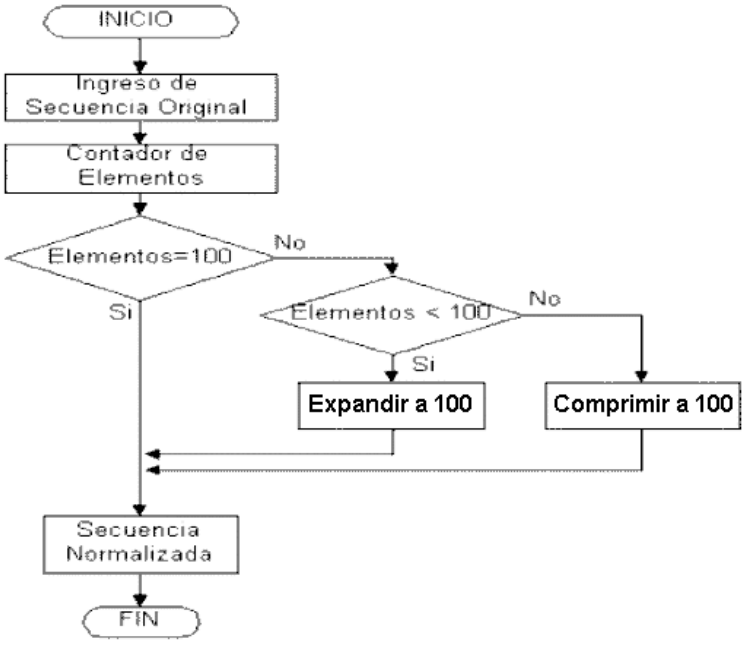
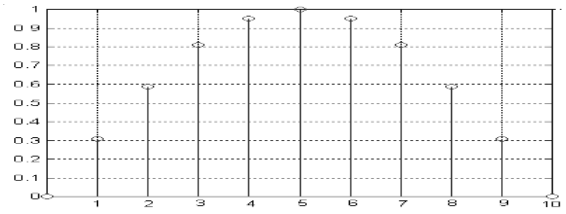


Figura 6.24 Funcionamiento de la Etapa de Normalización.

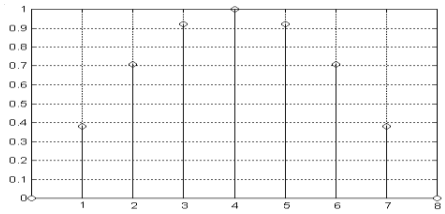
Para una mejor explicación de este proceso, se muestra un ejemplo ilustrativo en el cual se detallan los pasos a seguir, antes de describir el diagrama de flujo de la librería *Normaliza.dll.*

Sea *Xn,* una secuencia de 11 elementos (figura 6.25(a)), la cual será normalizada a un valor predefinido de 9 (figura 6.25(b))

Xn = {0, 0.3, 0.59, 0.8, 0.95,1, 0.95, 0.8, 0.59, 0.3, 0};



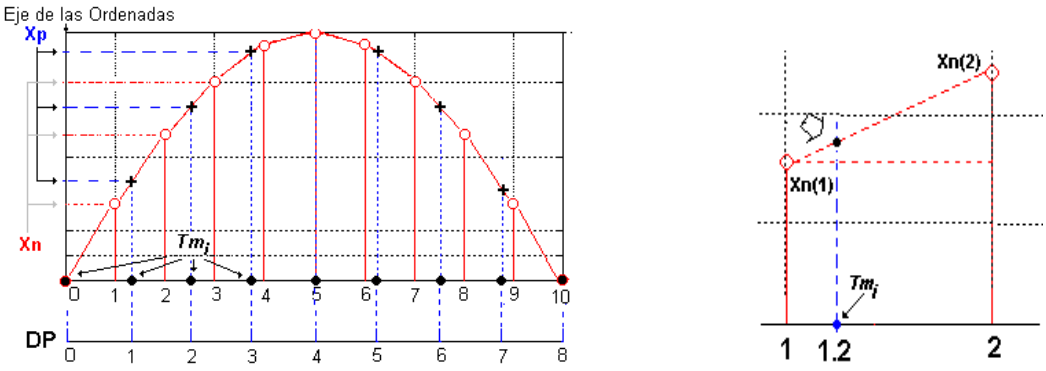
(a)



(b)

Figura 6.25 Secuencias de Entrada y de Salida de la Etapa de Normalizacion. (a) Secuencia Original (b) Secuencia Comprimida

Los 11 elementos tienen su correspondiente valor en el eje de las ordenadas; el proceso normalización consiste en hallar el valor de las nuevas ordenadas, para los 9 puntos que constituirán los elementos de la secuencia normalizada, esto se ilustra en la figura 6.26(a)



(a) **(b)**

Figura 6.26 Proceso de Interpolacion Lineal

En la figura, *Tm* es el vector que representa la proyección de estos 9 puntos, sobre el eje del dominio los que se hallan mediante la siguiente ecuación:

$$Tm(i) = \frac{nn-1}{np-1} \times i \qquad i = 0,1, \ \ldots \ 8 \tag{6.1}$$

Donde:

nn=11, es el número de elementos de la secuencia original.

np=9, es el número preestablecido de elementos.

Los valores de las nuevas ordenadas de los puntos *Tm,* proyectados, se hallan mediante la formula de Interpolación Lineal (ver figura 6.26(b)), que viene dada por:

$$Xn(Tm(i)) = Xn(k(i)+1) + \left(Xn\left(k(i)+2\right) - Xn\left(k(i)+1\right)\right) \times \left(Tm(i). - Xn\left(k(i)+1\right)\right) \tag{6.2}$$

Siendo:

$$k(i) = \|Tm(i)\| \qquad i=0,1,2....8 \tag{6.3}$$

La secuencia normalizada Xp, esta dada por:

$$Xp(i) = Xn(Tm(i)) \qquad i=0,1,2....8 \tag{6.4}$$

El resultado de aplicar las ecuaciones 6.1, 6.2, 6.3 y 6.4 sobre la secuencia Xn, es la secuencia normalizada *Xp*.

Xp = {0, 0.38, 0.70, 0.92, 1, 0.92, 0.7, 0.38, 0}.

La figura 6.27, muestra el diagrama de flujo de la librería "Normaliza.dll" que representa la etapa de Normalización del sistema. En dicho diagrama *Xn* constituye el parametro de entrada y el vector de salida viene dado por *Xp*. En primer lugar se inicializa la variable *np* con el valor de 100, la que indica la

longitud de la preetablecida, luego se guarda en *nn* la longitud de la secuencia de entrada, enseguida se traducen las ecuaciones 6.7, 6.8 y 6.9 descritas anteriormente.

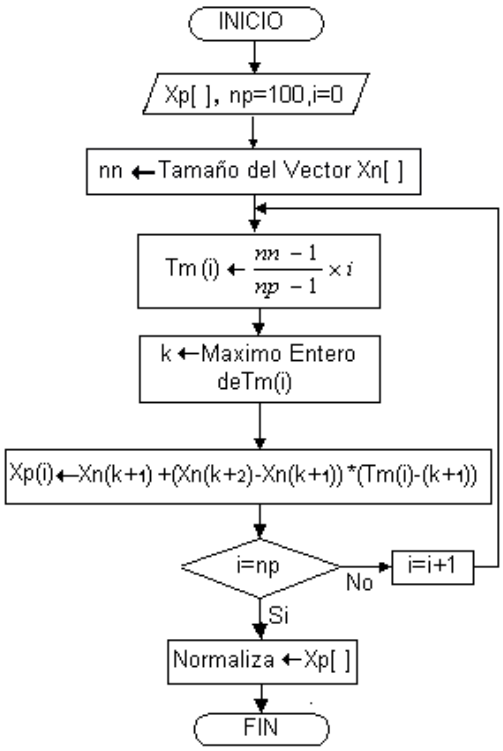


Figura 6.27 Diagrama de flujo de la subrutina *Normaliza.dll*

6.2.4 Redes Neuronales - RNA.dll

Esta librería corresponde a un conjunto de subrutinas que implementa una Red Neuronal (previamente entrenada) que forma parte de la etapa de Reconocimiento del Sistema.

Se ha tomado como ejemplo el reconocimiento de 10 palabras. La RNA utilizada es una del tipo Perceptrón Multicapa de 3 niveles que tiene las características presentadas en la tabla 6.2.

Tabla 6.2 Características del MLP utilizado

Capa	*Entrada (1)*	*Oculta (2)*	*Salida (3)*
Número de Nodos	*100*	*32*	*10*
Función de Transferencia	*Lineal*	*Sigmoidal*	*Sigmoidal*

El número de nodos en la capa de entrada es igual al número de elementos del vector de características, el cual, como resultado de la etapa de Normalización posee 100 elementos, cada uno de los cuales puede tomar un valor acotado entre cero y uno.

Los nodos en la capa de salida dependen directamente del número de patrones que el sistema puede reconocer. Por lo general, se asigna un nodo de salida por cada patrón a reconocer, sin embargo cuando se cuenta con un gran número de patrones, se opta por realizar una codificación en la salida.

En el sistema desarrollado en este trabajo, el número de salidas es igual al número de palabras que conforman el vocabulario de reconocimiento. Como consecuencia de esto, la red planteada para la prueba (tabla 6.2) es capaz de reconocer 10 palabras. Durante el diseño se asignó un nodo de salida a cada palabra, así, si la red reconoce a la Palabra 1, entregará como respuesta el vector V1 (figura 6.28), es decir, sólo la neurona asignada a esa palabra tendrá

el valor de uno, las demás tendrán cero. Si reconociera la Palabra 2, sólo la neurona asignada a ella tendrá el valor de uno y así sucesivamente.

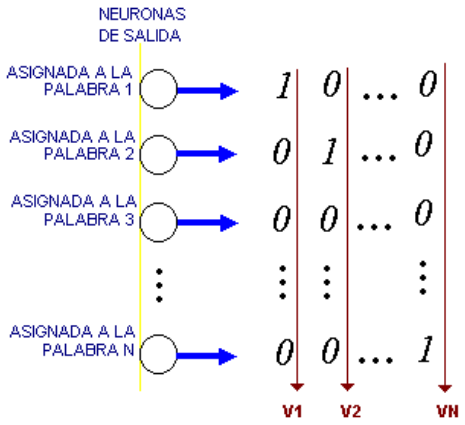


Figura 6.28 Expresión de las salidas en una red

En lo que respecta al número de nodos en la capa oculta, no existe una regla general para definirlo, este depende en gran medida de la aplicación especifica de la red y del criterio del diseñador. Teóricamente se sabe que, mientras más neuronas posean una capa, la redundancia es mayor y la red se vuelve mas tolerante a fallos, sin embargo algunos autores afirman que en la práctica puede resultar inconveniente usar gran cantidad de neuronas. [Hilera, Martínez. 1995]. Por esto, antes de definir dicho número, es necesario efectuar pruebas previas, en las que se evalúa el grado de aprendizaje de la red con diferentes números de nodos ocultos. Para el caso de la red que reconoce 10 palabras, se realizó dicha evaluación y se concluyó utilizar 32 nodos en la capa esta capa (el detalle de la prueba se muestra en al capítulo 7)

Las neuronas de Entrada, tienen como Función de Transferencia (FT) a la función Lineal de pendiente unitaria, debido a que estas, sólo se encargan de captar el estímulo externo y distribuirlas hacia el interior de la red. Es decir, prácticamente, no realizan un proceso sobre los datos de entrada.

Las neuronas de la capa Oculta y de Salida, tienen como FT, a la función Sigmoidal, ya que las estas a diferencia de las neuronas de entrada si

procesan los datos de entrada. El valor de k de la función Sigmoidal de cada una de las capas es diferente. Se ha realizado una prueba que evalúa el tiempo de entrenamiento de la red para distintos valores de k, en cada capa. Para el caso de la red planteada en la tabla 6.2, el valor de k en la capa oculta es de 0.091551 y en la capa de salida es 0.156945, como se muestra en la tabla 7.3. Esta prueba se detalla en el capítulo 7

Se ha convenido asignar cero al valor de Umbral de cada neurona que conforma el Perceptrón Multicapa con el propósito de disminuir la carga computacional de la red durante la etapa de entrenamiento y principalmente durante la etapa de reconocimiento (denominada propagación). Cabe resaltar que los valores de los pesos iniciales de las conexiones del MLP están acotados entre los valores de cero y uno [Freeman, Skapura. 1993].

En la figura 6.29 se ilustra este proceso que realiza la red para efectuar el Reconocimiento (denominado también Propagación), en donde se observa que el parámetro de entrada está dado por el Vector de Características Xp y la salida está constituida por el vector Ob; Wa y Wb constituyen el resultado del entrenamiento de la Red Neuronal (véase ítem 6.2.4.1). Wa es la Matriz de pesos entre las capas de Entrada y Oculta; y Wb entre las capas Oculta y de Salida

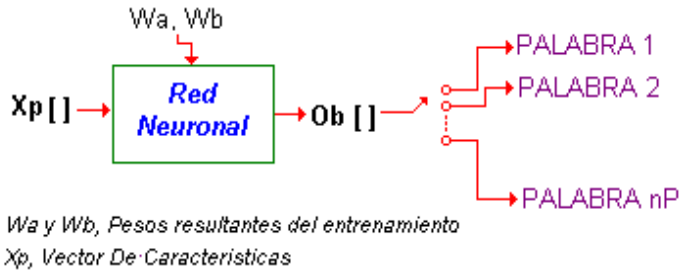


Figura 6.29 Proceso de Reconocimiento

En la figura 30 se muestra el diagrama de bloques de la librería *RNA.dll*, la cual realiza la propagacion de la red tipo MLP de tres capas.

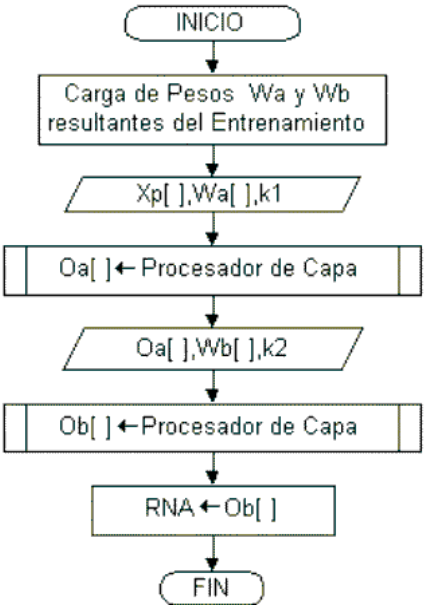


Figura 6.30 Proceso de Propagación

Inicialmente se cargan los pesos resultantes de la etapa de Entrenamiento (Wa y Wb), luego se realizan las operaciones en la capa Oculta, con los parámetros de entrada a Xp (Vector de Caracteristicas), k1 (Constante de la FT sigmoidal de la capa Oculta) y Wa; al final de la operación denominada Procesador de Capa (que se detalla más adelante como subrutina *Opca*) se devuelve el vector Oa que agrupa los valores de Salida de las neuronas de dicha capa. Luego se realizan las operaciones en la capa de Salida, con los parámetros Oa, valor previamente calculado, k2 (Constante de la FT sigmoidal de la capa de Salida) y Wb; al finalizar esta operación se devuelve el vector Ob que agrupa los valores de Salida de la red. Conforme a lo descrito anteriormente esta salida de la red (el vector Ob) está en términos de Ceros y Unos, por eso se hace necesario efectuar una decodificación que permita mostrar la palabra reconocida.

6.2.4.1 Algoritmo de Entrenamiento EBHA - Subrutina EntTotal

Se ha desarrollado un algoritmo de entrenamiento para la Red Neuronal tipo Perceptron Multicapa denominado Algoritmo de Entrenamiento por Bloques de Hablantes (EBHA), que se basa en el uso de la regla de aprendizaje Backpropagation (descrita en el capítulo 5). Este algoritmo se encuentra codificado en la subrutina *EntTotal*, que se describe mas adelante.

Inicialmente se efectúo el proceso de entrenamiento con métodos sugeridos en la bibliografía de referencia, sin embargo no se obtuvieron resultados satisfactorios ya que se presentaron problemas en la convergencia del error y también en el grado de generalización de clases, así como en el tiempo de entrenamiento. Por este motivo se decidió idear un método de entrenamiento coherente con el tipo de información que se deseaba clasificar, es decir palabras pronunciadas por distintos hablantes.

La figura 6.31 muestra una vision global de este sistema de entrenamiento, en el cual, los parámetros de entrada están conformados por el número de hablantes nH, el número nP de palabras que el sistema puede reconocer, el conjunto de patrones de entrenamiento Xp y los valores de salida están dados por los pesos modificados, Wa y Wb.

Figura 6.31 Vision global del sistema de entrenamiento EBHA

El algoritmo consiste en entrenar la red por bloques, cada uno de los cuales esta conformado por los patrones de un mismo hablante; al proceso de entrenamiento que se realiza sobre cada uno de estos bloques se ha denominado Entrenamiento por Hablante.

Para describir éste proceso se analiza la figura 6.32, en la cual se puede apreciar el bloque que contiene los nP patrones característicos de un mismo hablante (X1, X2, …, XnP). Al iniciarse el entrenamiento, el switch se encuentra en la posición 1, así, se tendrá como pesos iniciales a Wo (que representa tanto a Wa y Wb).

Teniendo como referencia los valores de Wo se realiza el entrenamiento del primer patrón (X1), mediante la regla de aprendizaje Backpropagation (BACKPR), que devuelve como resultado W1, el cual contiene los pesos modificados, los mismos que serán utilizados como referencia en el entrenamiento del segundo patrón (X2). Luego se continúa con el tercero, y así sucesivamente hasta llegar a entrenar el último patrón (XnP) que dará como resultado los pesos WnP.

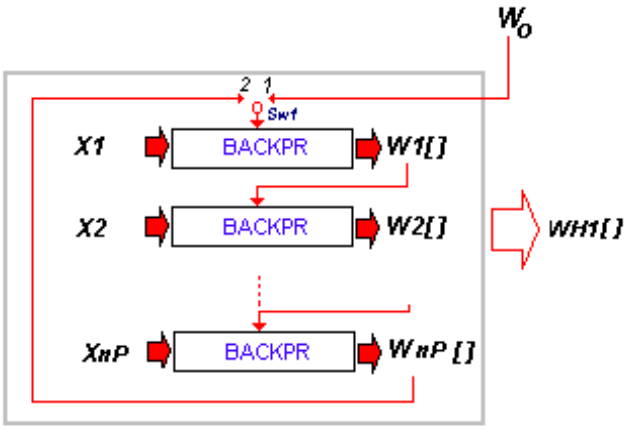


Figura 6.32 Entrenamiento de un hablante

Seguidamente se medirá el grado de similitud entre todos los pesos obtenidos, en caso de que sean diferentes, se repetirá el procedimiento anterior, pero el switch pasará a la posición 2, es decir, el primer patrón (X1) volverá a ser entrenado pero esta vez teniendo como referencia los pesos WnP. Este procedimiento se repite las veces que sea necesario, hasta que exista un alto grado de similitud entre todo los pesos obtenidos durante el desarrollo del proceso (W1=W2=W3...=WnP). Una vez que esto se ha logrado

se devuelve como resultado los pesos obtenidos en el último entrenamiento, los cuales se almacenan en WH1, que es el resultado de entrenar el primer hablante, este resultado es utilizado como referencia para entrenar un bloque siguiente, que corresponderá a un segundo hablante, en el cual se realizará un procedimiento análogo al anterior, y por consiguiente, tendrá como resultado una nueva matriz de pesos, WH2. Este proceso se continuara hasta que se termine de entrenar al número preestablecido de hablantes, nH. Esto se puede apreciar mejor en la figura 6.33.

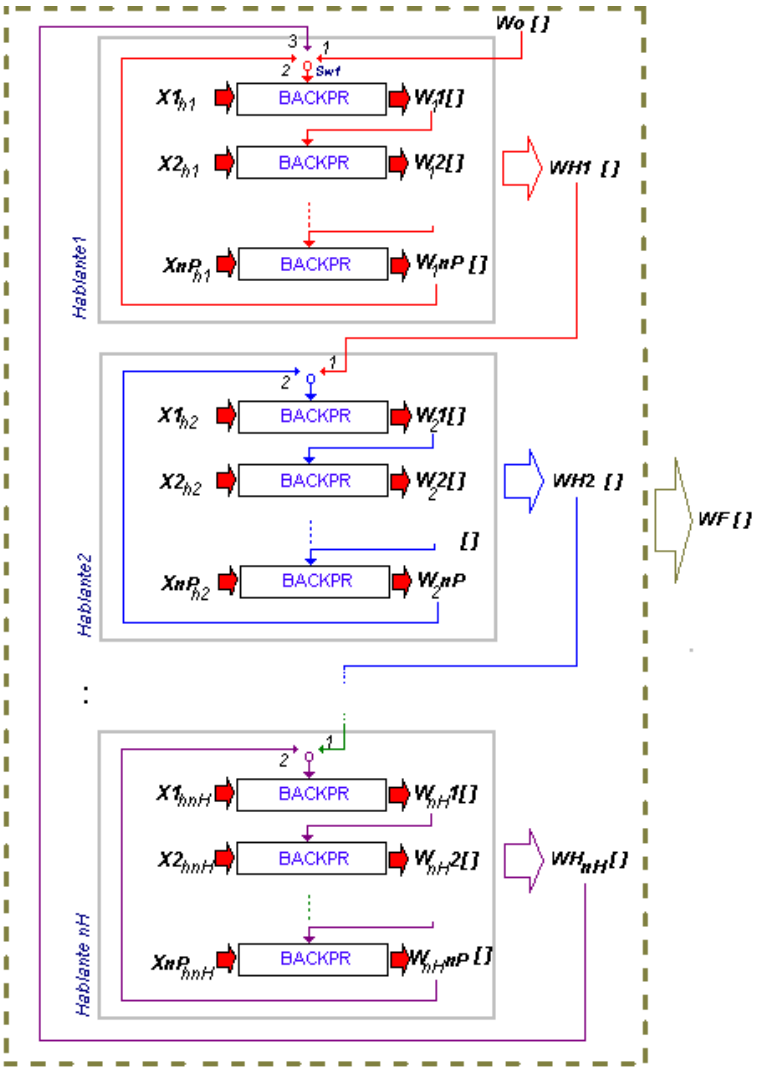


Figura 6.33 Entrenamiento de todo los hablantes (EBHA)

Después se efectúa una comparación entre los pesos resultantes del entrenamiento de cada hablante, con la finalidad de analizar la similitud entre

estos, si dichos pesos son diferentes, la secuencia de entrenamiento con cada uno de los bloques se repite, pero esta vez, con WH_{nH} como pesos de referencia en el primer hablante. (switch en posición 3)

Estas secuencias se repetirán hasta que el grado de similitud entre los pesos resultantes de cada bloque alcance un valor muy alto. Cuando esto sucede el proceso de entrenamiento finaliza y se devuelve como resultado los pesos del último entrenamiento (WH_{nH}) que constituyen los pesos entrenados de la red para todo los hablantes, y los cuales serán usados en el proceso de reconocimiento.

A continuación, en la figura 6.34 se muestra el diagrama de flujo de la subrutina *EntTotal*.

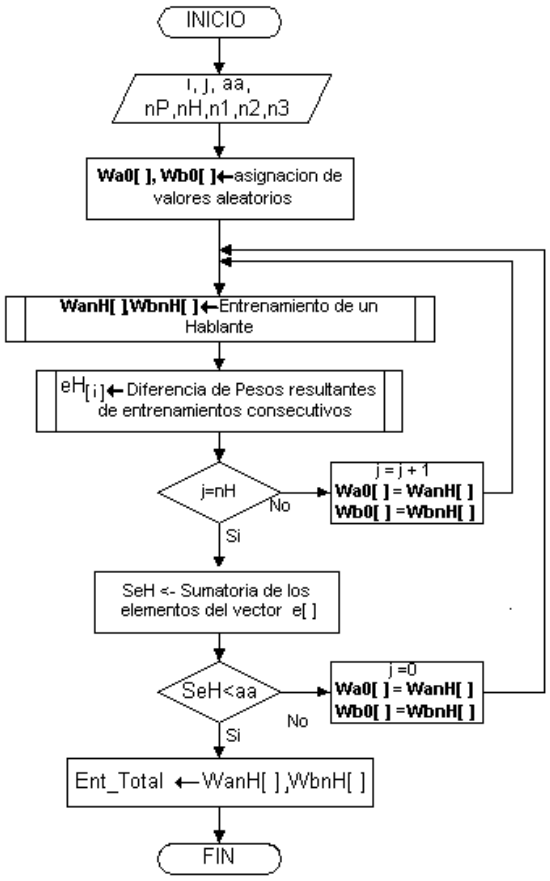


Figura 6.34 Subrutina de Entrenamiento de todo los hablantes (*EntTotal*)

En este algoritmo primero se inicializan los parámetros de entrenamiento n1, n2 y n3 (número de nodos de la capa de Entrada, Oculta y de Salida respectivamente); nH (número de hablantes), nP (número de palabras), y el parámetro de truncamiento *aa* (cercano a cero), que es el valor utilizado para establecer una cota inferior a la diferencia entre pesos del entrenamiento de dos hablantes consecutivos. Las variables i y j son contadores que se utilizan para seleccionar los hablantes y las palabras a entrenar.

Se asigna valores aleatorios entre 0 y 1 a los pesos de conexión Wa0 y Wb0, después se entrena cada uno de los nH hablantes, mediante la subrutina *EntHab* (o Entrenamiento de un Hablante). El vector eH almacena la diferencia entre los pesos resultantes del entrenamiento de dos patrones consecutivos, y SeH almacena la sumatoria de los elementos este vector.

Si el valor de SeH es mayor que aa, se vuelven a entrenar todo los hablantes, por el contrario, si después de muchas repeticiones SeH alcanza un valor menor que aa, el algoritmo detiene el proceso de aprendizaje y retorna como resultado los valores de WanH y WbnH como pesos finales de entrenamiento.

El cálculo de la diferencia de pesos resultantes del entrenamiento de dos hablantes consecutivos se realiza mediante la subrutina denominada Err_W, la que se muestra en la figura 6.35. Este proceso calcula el diferencia entre los pesos obtenidos como resultado del entrenamiento de dos hablantes consecutivos. Los parámetros de entrada son los pesos WM resultantes del entrenamiento del hablante m, y WN que son los pesos resultantes del entrenamiento del hablante m+1

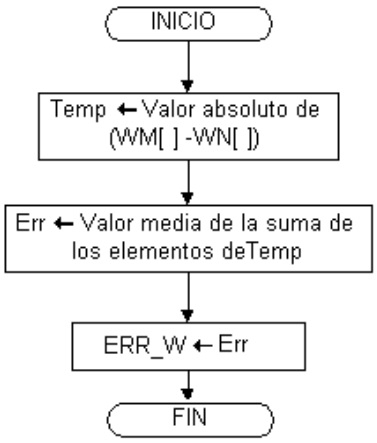


Figura 6.35 Diferencia de pesos resultantes del entrenamiento de dos hablantes consecutivos

Temp, es una matriz auxiliar que almacena el valor absoluto de la diferencia de WM[ ] y WN[ ], Err es el valor medio de la suma de los elementos de la matriz Temp.

La subrutina *EntHab* realiza el entrenamiento de cada uno de los nH hablantes sleccionados, por esto seguidamente se describe esta subrutina

6.2.4.2 Entrenamiento de un Hablante - Subrutina EntHab

En la figura 6.36 se muestra el diagrama de flujo que efectúa el proceso denominado Entrenamiento por Hablante. Los parámetros de entrada de esta subrutina están dados por las dos matrices Wa y Wb, que son los pesos iniciales de la red, y devuelve las matrices, WanP y WbnP que constituyen las matrices de pesos de conexión modificados correspondientes al entrenamiento de un hablante.

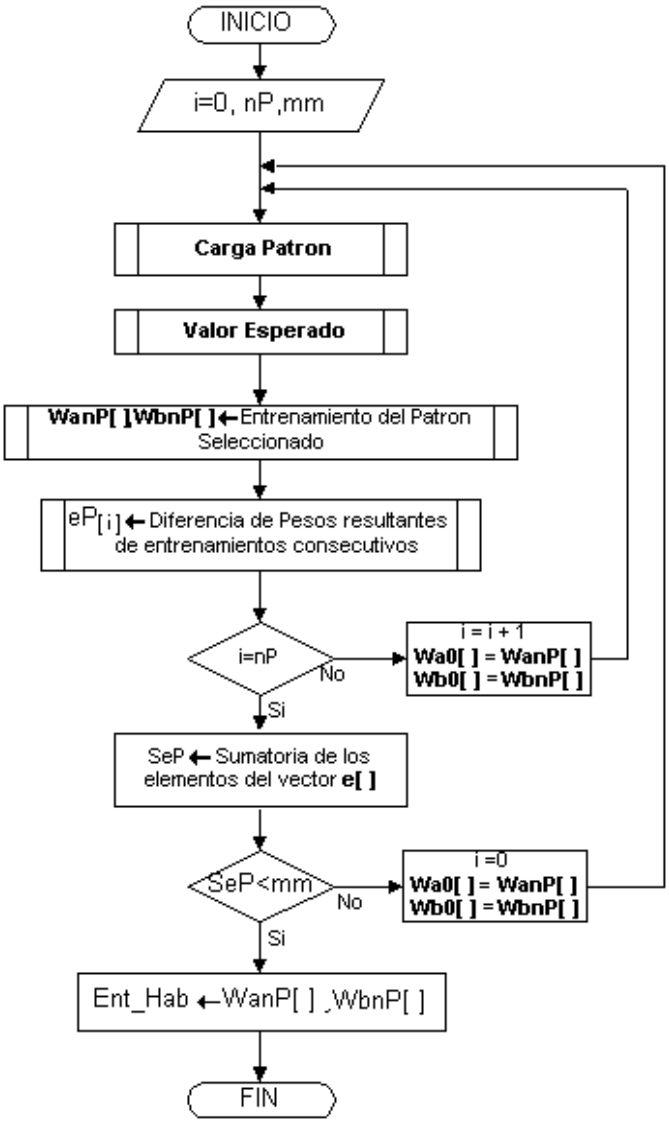


Figura 6.36 Entrenamiento por Hablante

El contador i indica el índice del patrón a entrenar, nP es el total de patrones a entrenar y mm es la cota de error para medir grado de similitud entre los pesos de los patrones, el cual es igual a 0.000001. El Entrenamiento del Patrón seleccionado se realiza con el procedimiento denominado *Backpr*, el cual aplica el algoritmo Backpropagatión al patrón seleccionado, que se explico anteriormente.

CargaPatron es la subrutina que selecciona el patrón que se va a entrenar y lo almacena en el vector X, ValorEsperado selecciona su correspondiente salida deseada en el vector vE. El vector eP almacena la

diferencia entre los pesos resultantes del entrenamiento de dos patrones consecutivos, y SeP es la variable que almacena la sumatoria de los elementos este vector. El valor de eP también se calcula mediante la Subrutina ErrW, descrita anteriormente.

Inicialmente se entrena la red para los nP patrones que corresponden a un hablante; si el valor de SeP es mayor que mm, se vuelven a entrenar todo los patrones, en caso contrario (SeP < mm) el algoritmo devuelve los valores de WanP y WbnP y finaliza el proceso de entrenamiento.

El entrenamiento de cada patrón correspondiente a un hablante se realiza mediante la subrutina *Backpr,* la cual se detallara en el próximo ítem.

6.2.4.3 Entrenamiento de un Patrón - Subrutina Backpr

En la figura 6.37 se muestra el diagrama de flujo de esta subrutina que corresponde al algoritmo de Aprendizaje Backpropagation, para un Perceptrón Multicapa de tres niveles. Dicho procedimiento esta constituido por un conjunto de subrutinas que efectúan las operaciones correspondientes al algoritmo, descrita en la sección 5.4.2.1.

Los parámetros de entrada de *Backpr* están dados por, las matrices de pesos Wa y Wb (si se trata del primer patron a entrenar ambas tienen valores aleatorios, en caso contrario, estos son los pasos resultantes del entrenamiento del patron anterior); también por el patrón de entrada X, asimismo por el valor de salida esperado vE correspondiente a dicho patrón, por las constantes k1 y k2 de la función Sigmoidal de las capas oculta y de salida respectivamente, además por la taza de aprendizaje nz y por ep1 que es la cota de error preestablecido para medir grado de similitud entre la salida real y la salida esperada de la red.

Los parámetros de salida están constituidos por valores modificados de los pesos de conexión expresados por WanP y WbnP, como se muestra en la siguiente expresión:

[WanP , WbnP] = ***Backpr*** *(Wa0,Wb0,X[ ],vE[ ],k1,k2,nz,ep1)*

Seguidamente se describen cada una de las subrutinas incluidas dentro de BACKPR.

6.2.4.3.1 Procesador de Capa - Subrutina Opca

Esta subrutina sirve para calcular el valor de salida de las neuronas, la cual actua como un procesador de la capa oculta o de salida, de acuerdo a los parametros de configuracion que se le provee. El resultado es almacenado en un vector llamado O (Ver figura 6.38).

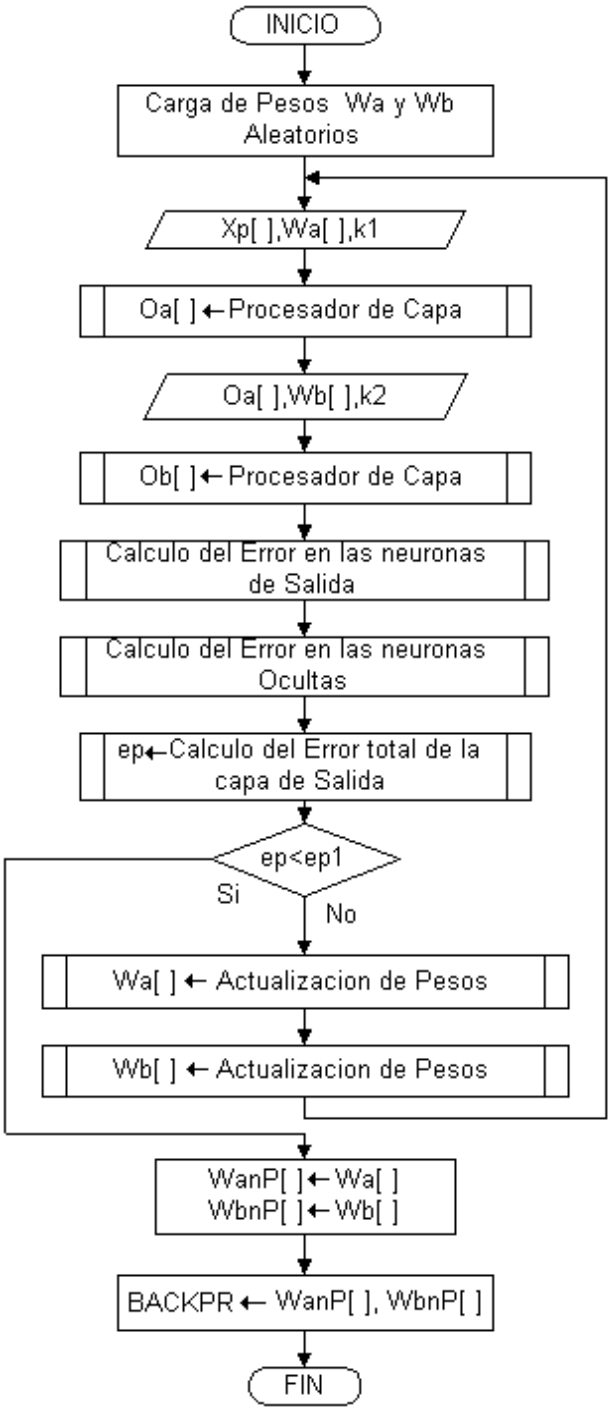


Figura 6.37 Algoritmo de Aprendizaje Backpropagation

En la figura 6.38, X y W son los parámetros de entrada, el primero representa el vector de entrada de cada neurona de la capa en cuestión y el segundo representa la matriz de pesos de conexión con la capa anterior; el vector R representa los valores de Entrada Neta de las neuronas.

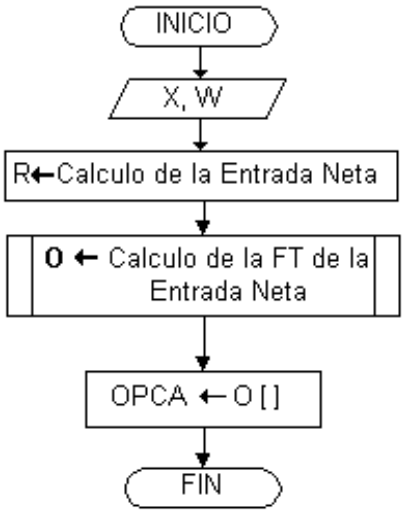


Figura 6.38 Procesador de Capa

6.2.4.3.2 Cálculo del Error en las neuronas de Salida - Subrutina Err2

Esta subrutina (figura 6.39) calcula el error producido en el cada una de las neuronas de la Capa de Salida y agrupa los resultados en el vector Db de acuerdo a la ec 5.13.

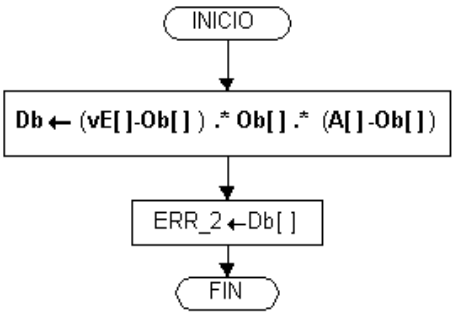


Figura 6.39 Error en las neuronas de Salida

Los parámetros de entrada son los vectores que agrupan los valores de salida real y deseada en cada neurona de la capa de Salida de la red, Ob y vE respectivamente, A es un vector de las mismas dimensiones que Ob, cuyos elementos son todos 1, y el símbolo “. * “ es un operador definido que realiza la multiplicación término a término de dos matrices equidimensionales.

6.2.4.3.3 Cálculo del Error en las neuronas de Ocultas - Subrutina Err1

Este procedimiento calcula el error producido en cada una de las neuronas de la Capa Oculta y agrupa los resultados en el vector Da en concordancia con la ec. 5.14, como se muestra en la figura 6.40.

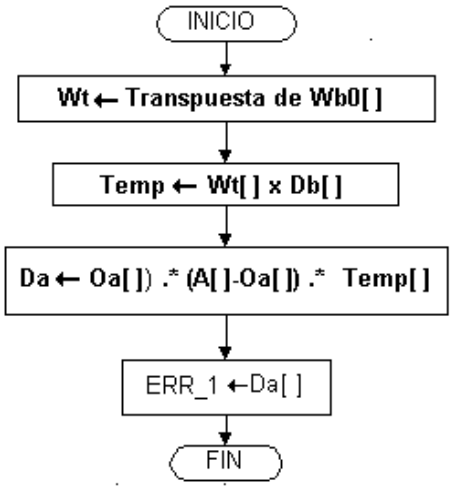


Figura 6.40 Error en las neuronas Ocultas

Los parámetros de entrada son, el vector Oa que agrupa los valores de salida en cada neurona de la capa Oculta, la matriz Wb0 de pesos de conexión con la capa de Salida y además el error producido en cada una de las neuronas de la Capa de Salida agrupados en el vector Db. Wt es la matriz transpuesta de Wb0, A es un vector de las mismas dimensiones que Oa, cuyos elementos son todos 1 y Temp una matriz auxiliar.

6.2.4.3.4 Actualizacion de pesos - Subrutina Actp

Esta Subrutina calcula el nuevo valor de una matriz de pesos W0, (Wa0 o Wb0), es decir actualiza los pesos de conexión entre dos capas (ec 5.16 y 5.17). En la figura 6.41, se muestra este procedimiento, cuyos parámetros de entrada son, el vector de entrada X de cada neurona de una determinada capa de la red, la matriz W0 de pesos de conexión de ésta capa con posterior y D es el vector en el que se almacena el valor de error producido en las neuronas de la capa posterior.

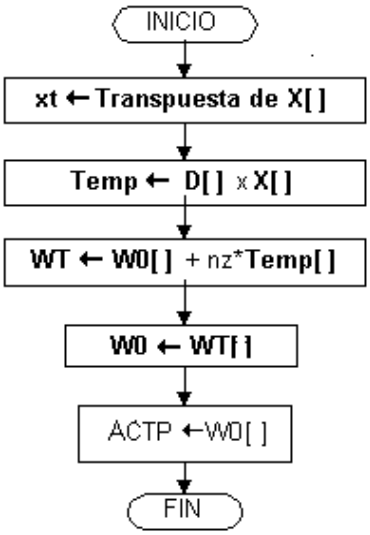


Figura 6.41 Actualización de Pesos

WT es la matriz en la que se almacenan el valor de los pesos actualizados temporalmente, xt es la Transpuesta del vector X, Temp es una variable temporal y nz es la taza de aprendizaje.

6.2.4.3.5 Error en la capa de Salida - Subrutina Mod

Esta subrutina aplica la ec 5.18 sobre el vector Db (que en el diagrama de flujo de la figura 6.42, esta representado por D), que agrupa los errores producidos en las neuronas de la capa de salida, el resultado es el error producido en la capa de salida, por la salida real con respecto a la salida deseada.

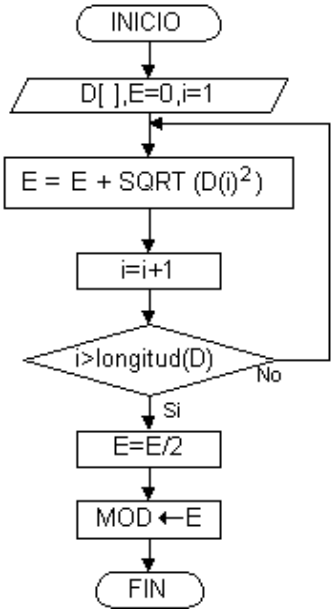


Figura 6.42 Cálculo del Error en la capa de Salida

6.3 Interface Gráfica del RAV

El RAV tiene el comportamiento funcional de un instrumento Virtual especializado para el análisis de las Señales de Voz. Permite graficar la señal de voz en el dominio del tiempo, asi como en el dominio de la Frecuencia, adicionalmente, muestra las gráficas de características tales como la Energía, la Densidad de Cruces por Cero, asi como la función COPER. También posee la opción para grabar palabras en formatos Wav y Csv. Otra de sus funciones es la de crear y almacenar los patrones característicos de las palabras, necesarios para el entrenamiento de la Red Neuronal.

La interface gráfica del RAV ha sido desarrollada en el lenguaje de Programación Visual Basic 6.0, y utiliza Librerías de Enlace Dinámico (DLLs) creadas en Visual C++ 6.0, las cuales contienen subrutinas que realizan tanto el Procesamiento como el Reconocimiento de Voz. El uso de estas librerías dinámicas también permite enlazar el sistema de reconocimiento con otros programas tales como Lab View ó Java, entre otros, debido a que los DLLs son compatibles con ellos.

En la figura 6.43, se muestra la pantalla de Inicio del RAV, luego de presionar el botón de Encendido, se despliega un grupo de fichas que contendrán las diferentes opciones del programa(Ver figura 6.44). Estas opciones son listadas a continuación, y serán detalladas en los ítems siguientes.

- **Señales**: Gráficas de la Señal de Voz y sus características.
- **Controles**: Establece el modo de operación y parámetros de la detección de extremos.
- **Grabar**: Almacena las palabras pronunciadas.
- **Coeficientes**: Crea patrones característicos.
- **RNA**: Establece los parámetros de la Red Neuronal , y realiza el proceso de entrenamiento.
- **Pantalla**: Permite controlar la visualización de las gráficas.

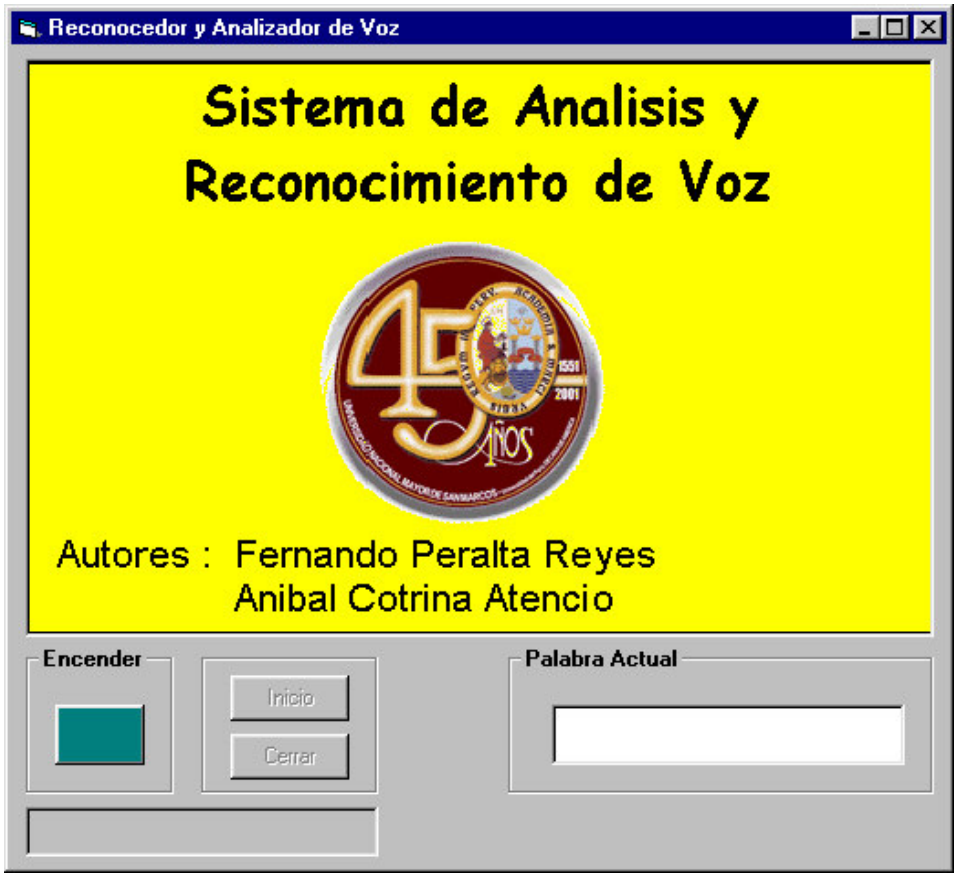


Figura 6.43 Pantalla de Inicio del RAV

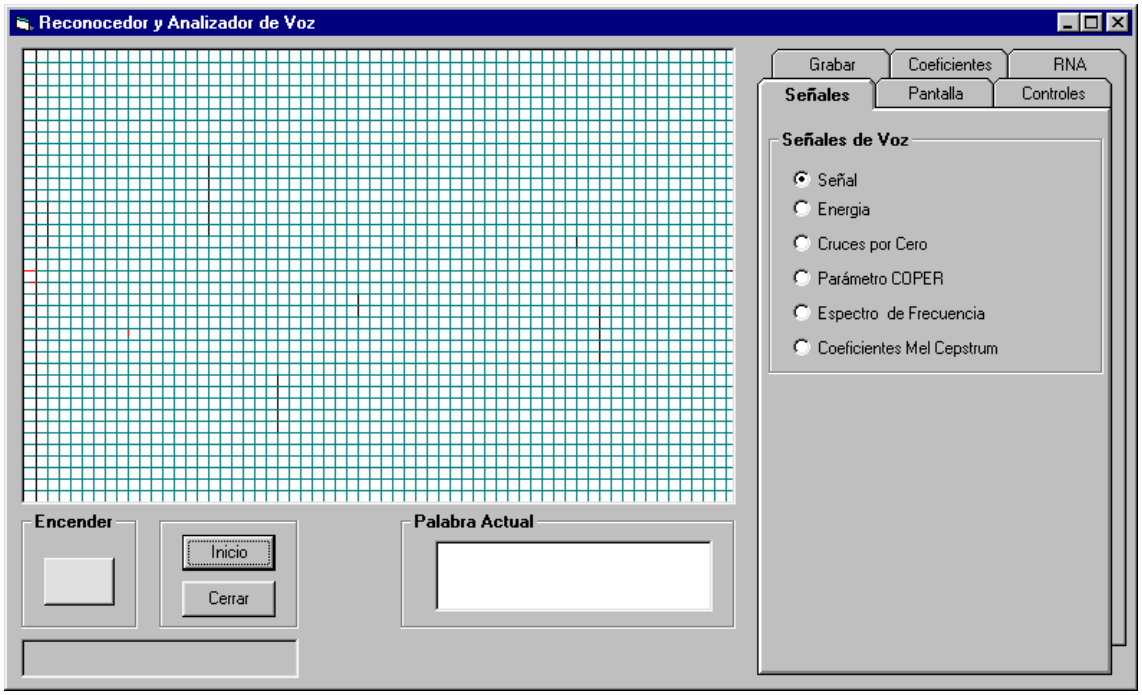


Figura 6.44 Pantalla desplegada del RAV

6.3.1 Señales

Esta ficha permite visualizar la gráfica de la Señal de Voz, así como sus principales características, entre las que tenemos Energía, Cruces por Cero, Función COPER, Espectro de Frecuencia y Coeficientes Mel Cepstrum. A continuación, a manera de ejemplo, se muestra las gráficas, para cada uno de estas opciones.

- ***Señal en el Tiempo***

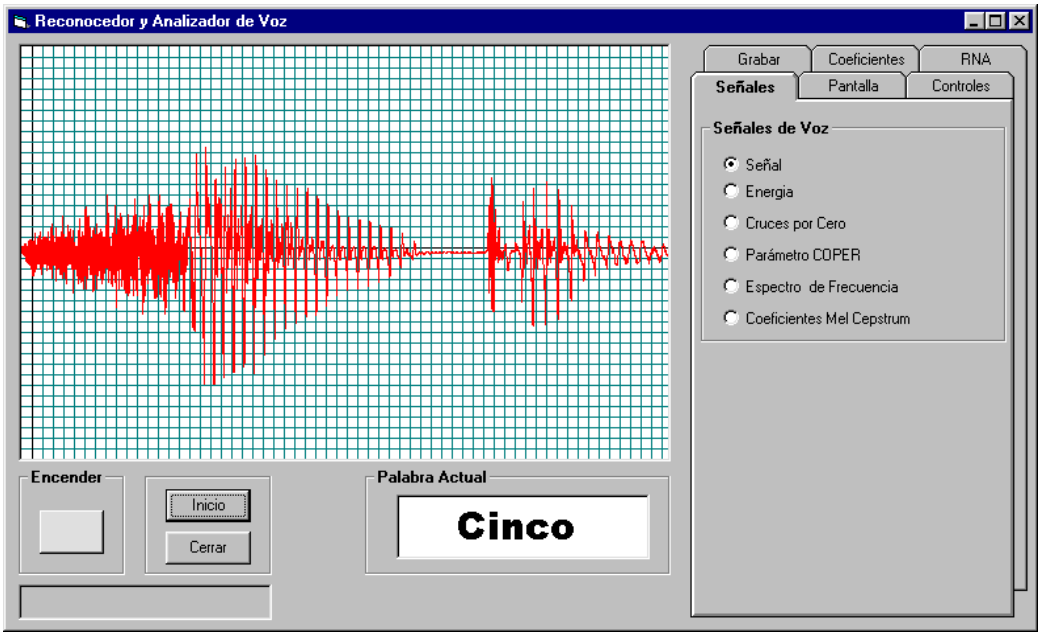


Figura 6.45 Gráfica de la Señal en el tiempo

- ***Energía***

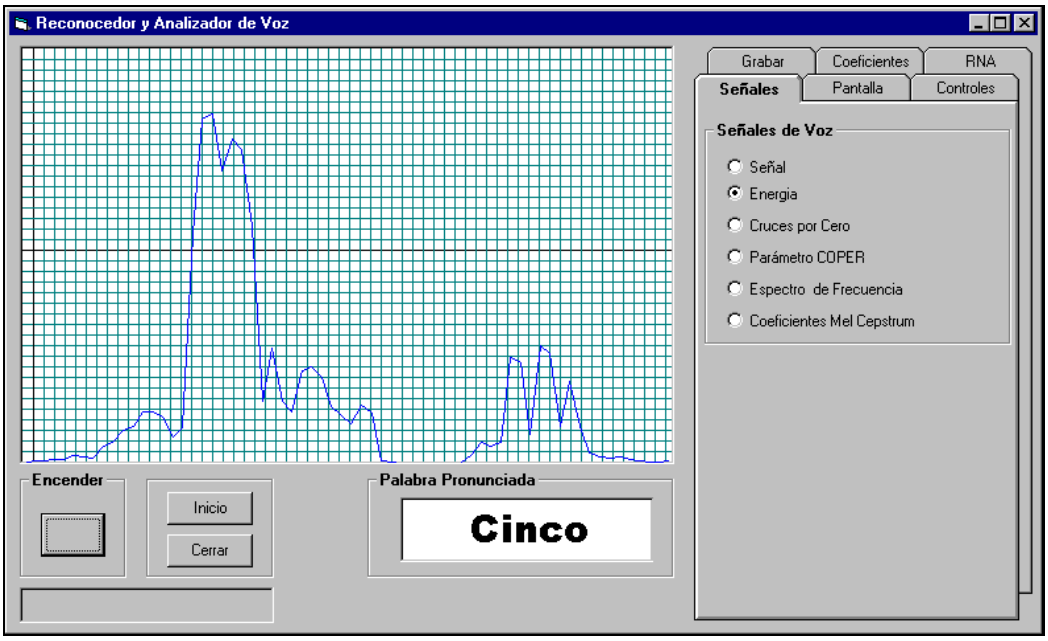


Figura 6.46 Gráfica de la Energía

- ***Cruces por Cero***

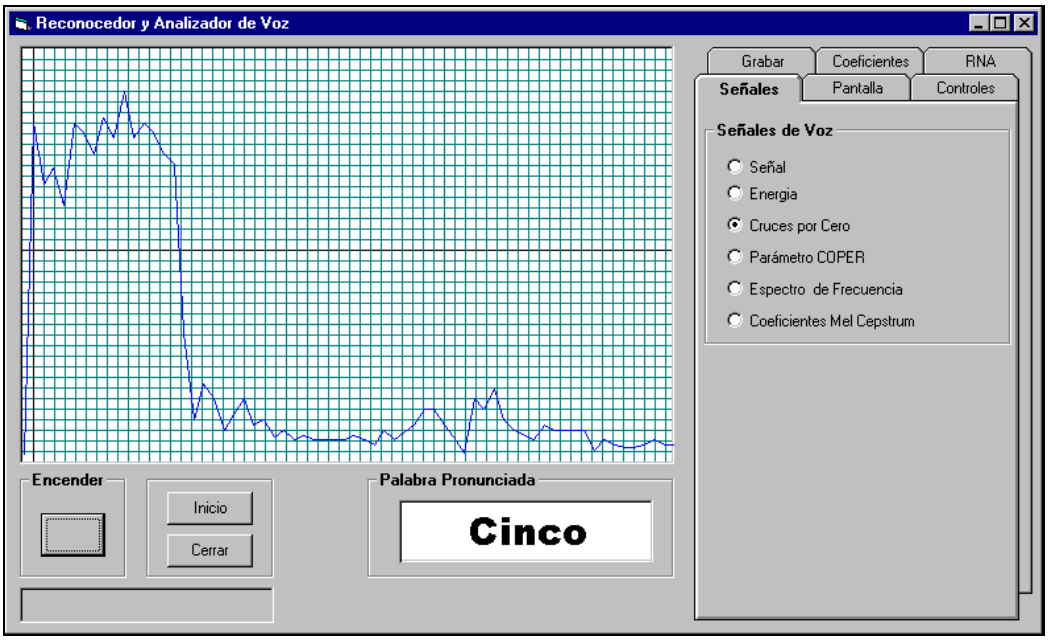


Figura 6.47 Gráfica de la Densidad de Cruces por Cero

- ***Función COPER***

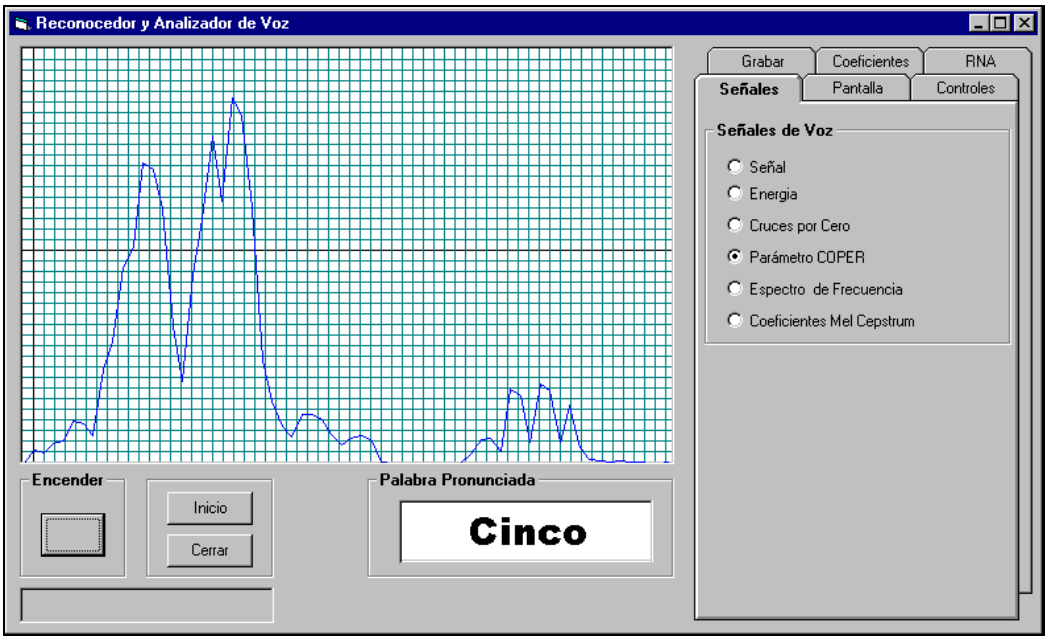


Figura 6.48 Gráfica de la función COPER

- ***Espectro de Frecuencia***

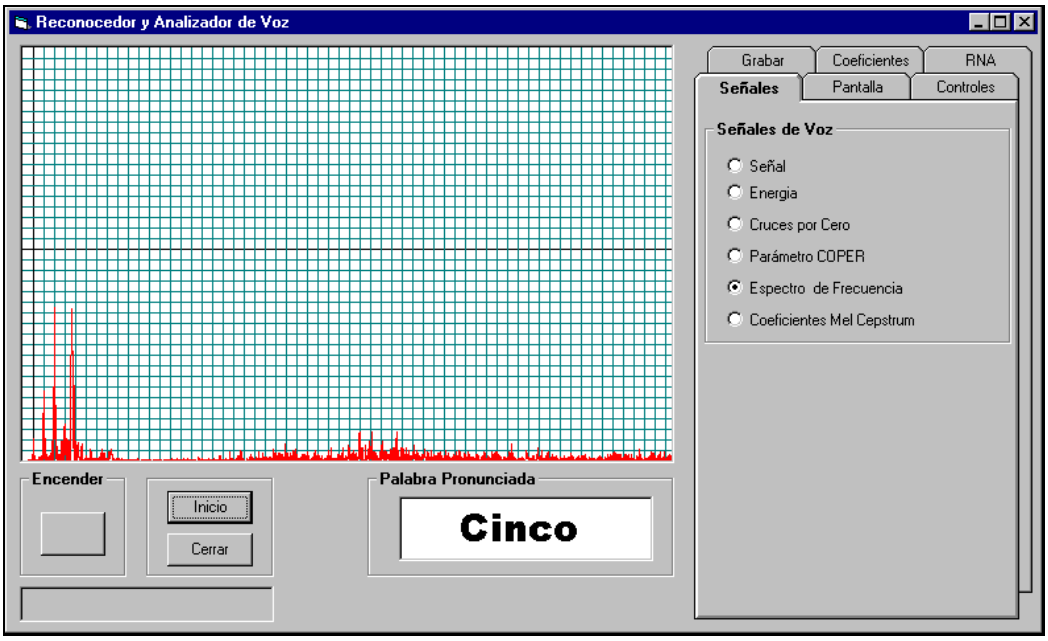


Figura 6.49 Gráfica del Espectro de Frecuencia

- ***Coeficientes Mel Cepstrum***

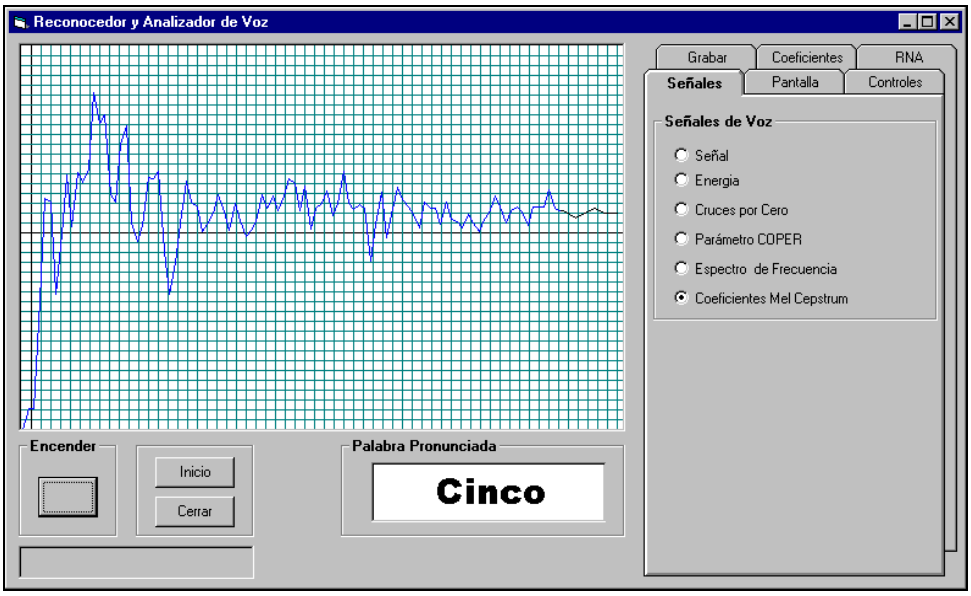


Figura 6.50 Gráfica de los Coeficientes Mel Cepstrum

6.3.2 Controles

En esta ficha se encuentran las opciones que controlan los modos de operación del Sistema, la Aplicación deseada, además permite establecer los parámetros para la Detección de Extremos.

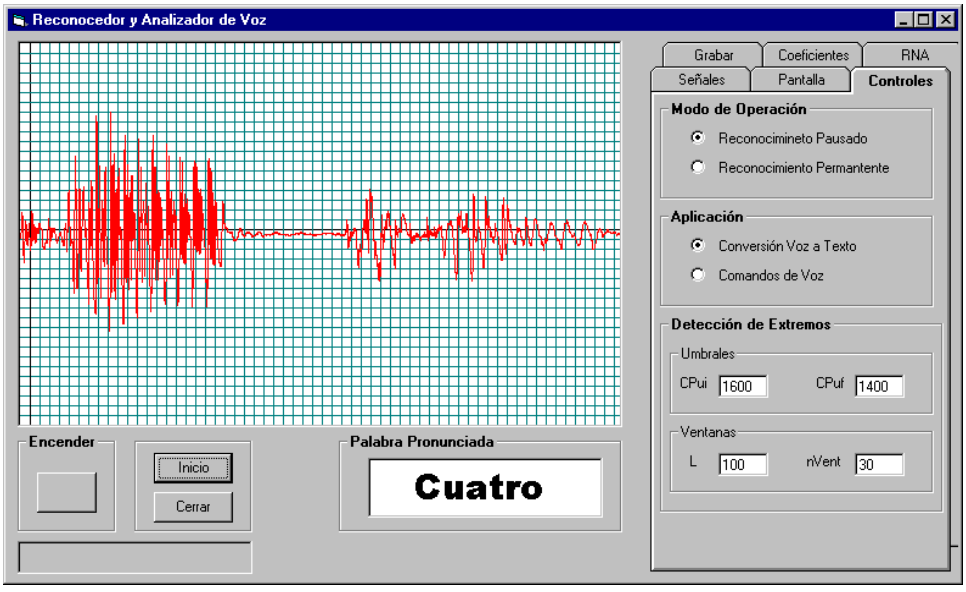


Figura 6.51 Ficha de Controles

Los modos de operación son:

- Reconocimiento Pausado: Esta opción permite adquirir la Señal de Voz para realizar el procesamiento y reconocimiento, cada vez que se presiona el botón de Inicio.

- Reconocimiento Permanente: En este modo de operación, una vez que ha sido presionado el botón de Inicio, el sistema, sensa permanentemente la entrada del micrófono, en busca de actividad de voz y realizar el proceso de reconocimiento.

La opción de Aplicación permite realizar la Conversión Voz a Texto, o efectuar un Comando de Voz, que específicamente es utilizado para ejecutar programas de Windows, específicamente: Word, Excel, Solitario y PaintBrush. Al dar el comando de Voz, el programa se ejecutará automáticamente, tal como se muestra en la siguiente figura, donde el programa Microsoft Word se activa luego de pronunciar el comando respectivo.

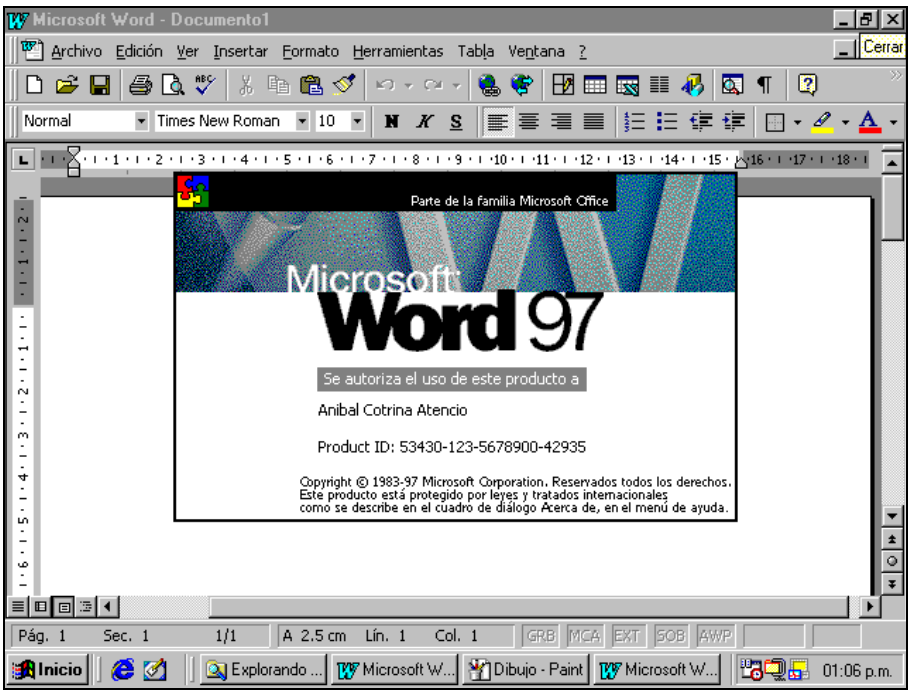


Figura 6.52 Ejecución de Word mediante un comando de Voz

Respecto a la Detección de Extremos, se tienen cajas de texto que permite modificar los niveles de umbral (CP_{UI}, CP_{UF}), así como el número de tramas consecutivas para determinar el fin de pronunciación (nVent) y la longitud de cada una de estas tramas (L).

6.3.3 *Grabar*

Esta ficha tiene como función almacenar las secuencias resultantes de la digitalización de las palabras, presenta dos opciones: Palabra y Proyecto, tal como se aprecia en la figura 6.53. La primera realiza el almacenamiento individual de una palabra y la segunda crea un Proyecto de Grabacion, que consiste en almacenar un conjunto de palabras, desde listas automáticas, o lista creadas manualmente.

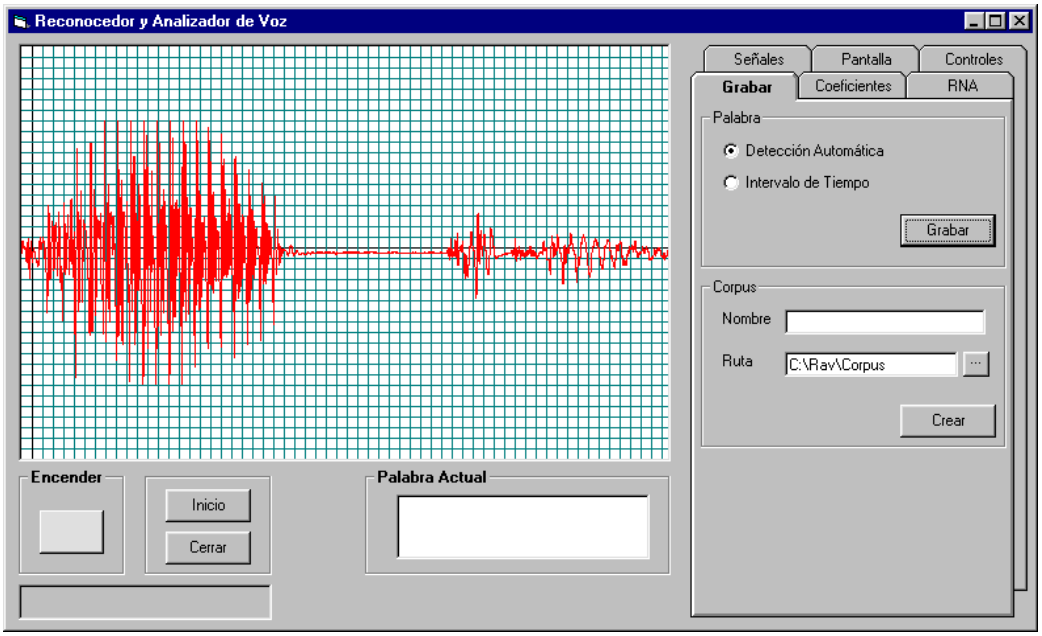


Figura 6.53 Ficha de Grabación

La opcion 'Palabra' presenta dos modos de almacenamiento:

- Detección Automática - Realiza la detección automática de extremos de las palabras pronunciadas, y luego almacena los datos en un archivo.

- Intervalo de tiempo - Permite almacenar en un archivo los datos adquiridos dentro de un intervalo de tiempo predefinido.

Por otro lado la opción proyecto presenta dos cajas de texto en las cuales se establece el nombre y la ruta del proyecto. El boton Crear carga una ventana en la cual se puede elegir una lista predefinida o generar una manualmente una de hasta 20 palabras (Ver figura 6.54). Además, en esta ventana se establecen el numero de palabras, el numero de hablantes y el numero de repeticiones de cada palabra.

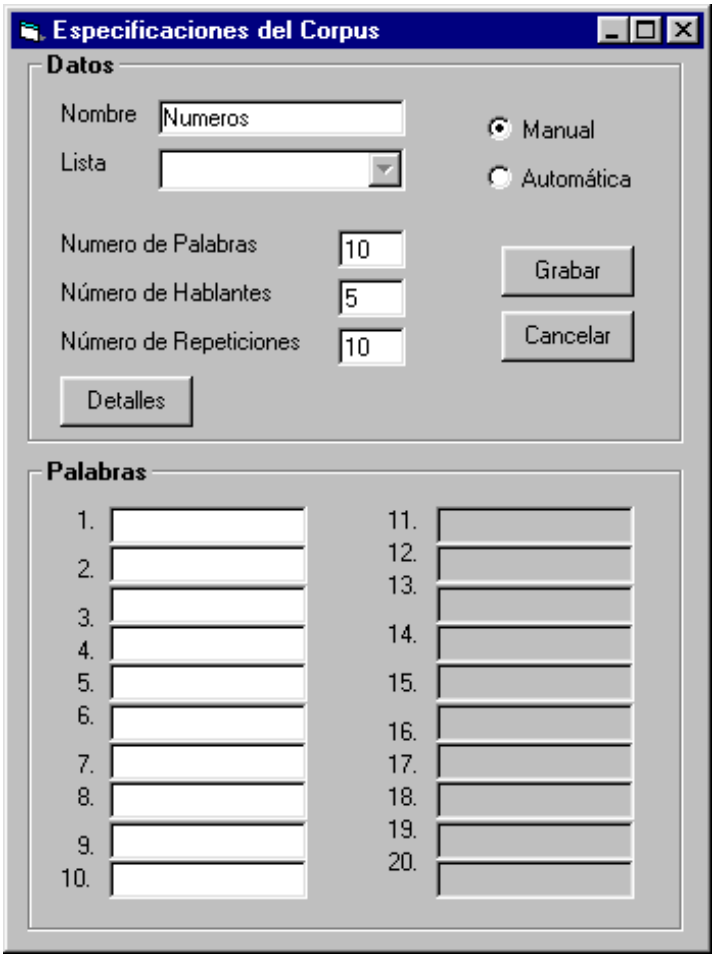


Figura 6.54 Pantalla de Creación de Proyecto

6.3.4 Coeficientes

En esta ficha se encuentran las opciones para crear los patrones característicos de palabras que han sido previamente almacenadas en archivos, con la finalidad de construir el corpus de datos necesario para el entrenamiento de la RNA.

Tal como se aprecia en la figura 6.55, existen dos opciones: Archivo y Proyecto. La opción Archivo permite crear un solo patrón característico seleccionando un archivo que contenga una palabra.

La opción Proyecto permite crear un corpus de entrenamiento, es decir un grupo de patrones característicos, tomando como referencia un grupo de palabras que hallan sido grabadas previamente en un proyecto creado mediante la opción Grabar.

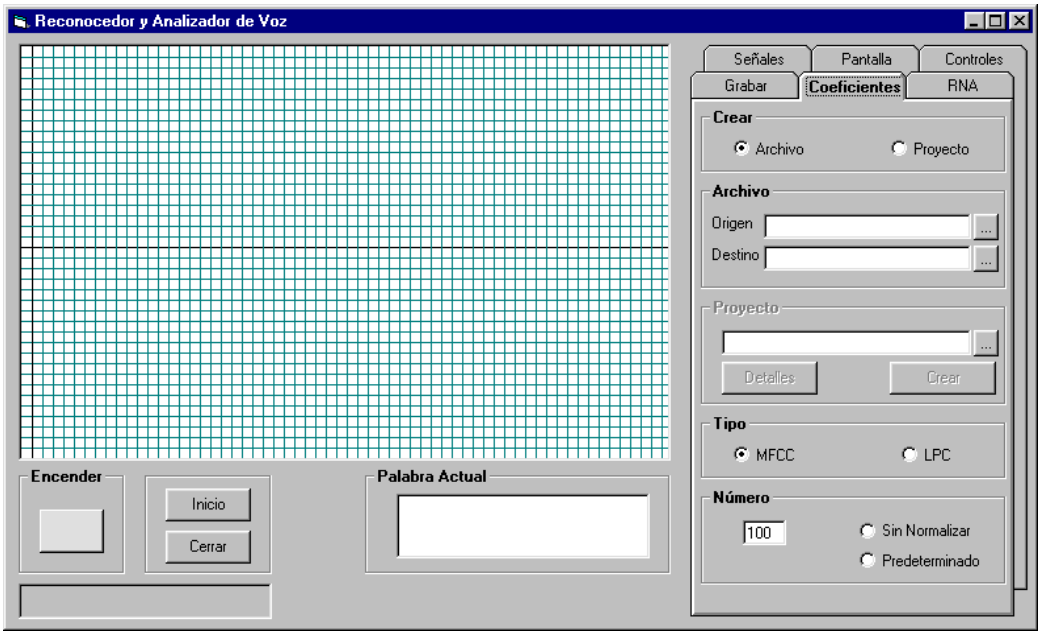


Figura 6.55 Ficha de Creación de Coeficientes

En la opción Tipo se debe indicar el tipo de Coeficiente a crear, ya sea, MFCC o LPC. El tamaño del vector de coeficientes característicos se establece mediante la opción Número, el cual tiene predefinido el valor de 100.

6.3.5 RNA

En esta ficha se muestra las opciones necesarias para establecer los parámetros de la Red Neuronal que se usara en la etapa de reconocimiento, la cual esta dividida en tres secciones como se muestra en la figura 6.56.

- Topologia: En esta sección se establecen los parámetros estructurales del MLP, los cuales son el número de nodos en cada capa, y la constante de la función Sigmoide de las capas Oculta (k1) y de Salida (k2)

- Entrenamiento: Establece los parámetros de entrenamiento de la red como son el error de truncamiento, el número de palabras y el número de hablantes, y el nombre de los pesos resultantes. El botón Entrenar carga una ventana que muestra el inicio del entrenamiento (figura 6.57).

- Propagación: En esta sección se indica los nombres de los archivos que contienen los pesos resultantes de un proceso previo de entrenamiento. El botón Ver muestra el resultado de la propagación en cada salida de la red.

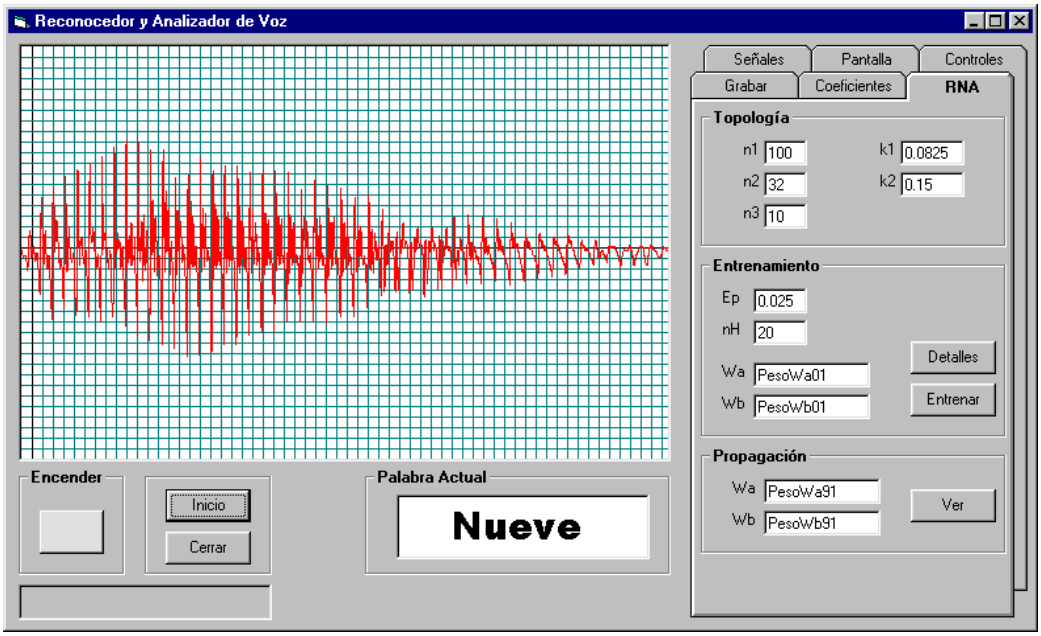


Figura 6.56 Ficha de la RNA

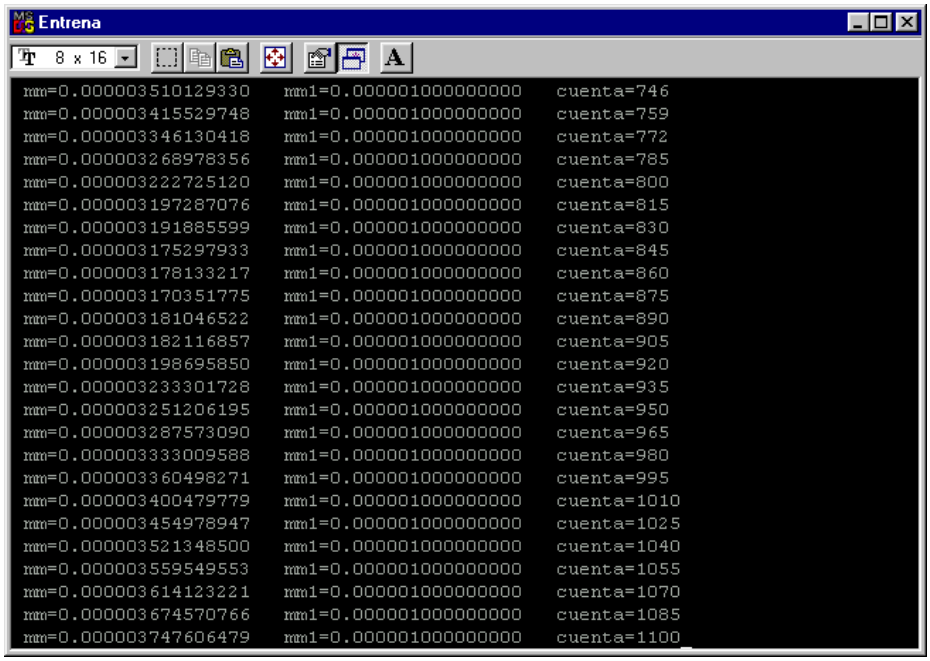


Figura 6.57 Pantalla de Entrenamiento de la RNA

6.3.6 Pantalla

Esta ficha, permite modificar la visualización de los gráficos correspondientes a la señal. Como se muestra en la figura 6.58, en esta ficha se distinguen las tres controles denominados Ejes, Cuadrícula y Traza de Color.

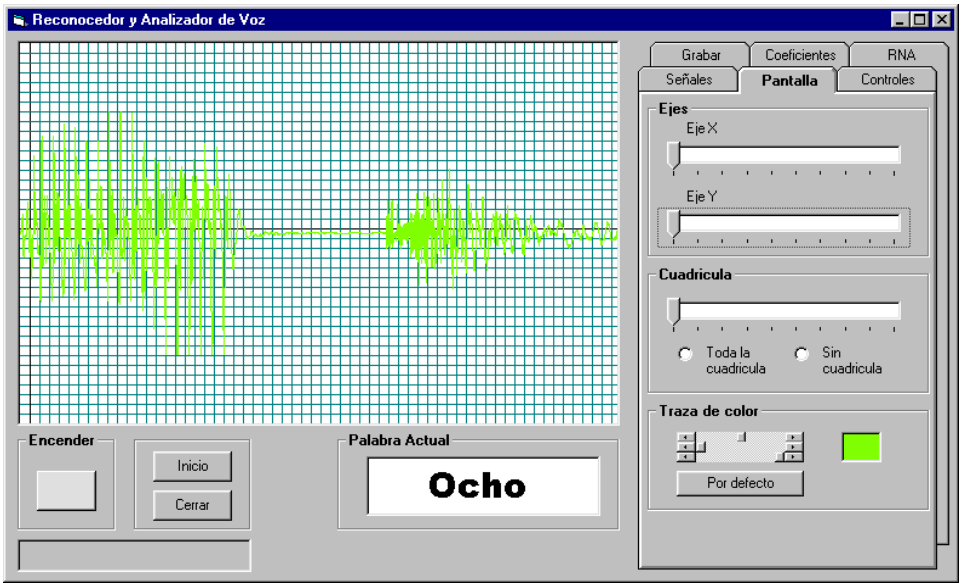


Figura 6.58 Ficha de control Visual de las Gráficas

- ***Ejes***

Este control permite realizar una Ampliacion y/o Reduccion de los ejes de coordenadas de la figura que se muestra en el display del RAV

- ***Cuadrícula***

Esta opción permite Mostrar u Ocultar la cuadrícula.

- ***Traza de color***

Esta opción permite modificar el color de la gráfica de las señales.

7. PRUEBAS Y RESULTADOS EXPERIMENTALES

En el presente capítulo se muestra los resultados de las principales pruebas experimentales que se realizaron en el desarrollo del sistema. Está dividido en dos partes; en la primera, se realiza la evaluación de los métodos y algoritmos propuestos durante la construcción del sistema, así, se evaluarán el Algoritmo COPER y el método ideado para la determinación de los parámetros de la Red Neuronal.

En la segunda parte de este capítulo, se mide la eficiencia del sistema en el Reconocimiento de Voz.

7.1. Evaluación de métodos y algoritmos propuestos

7.1.1. Evaluación del Algoritmo COPER

El objetivo de esta prueba es comprobar la validez de la función COPER(ver ítem 3.2.2.1), que surge como una propuesta de nuestro trabajo, para una detección de extremos más eficiente comparada con los métodos tradicionales. Previamente se describirá el proceso para establecer los niveles de umbral de Inicio y Fin de una pronunciación.

7.1.1.1. Niveles de Umbral

Para determinar los niveles de umbral se creo un corpus de 20 palabras, conformadas por los 10 dígitos del español, pronunciados dos veces. Cada palabra se encuentra mezclada con ruido de fondo, tal como se puede observar en la figura 7.1, en la cual se muestra la gráfica correspondiente a la palabra 'Cero'. El ruido adicionado a cada palabra es del tipo gaussiano, y poseen distintas amplitudes para cada caso, para así simular distintos entornos ruidos.

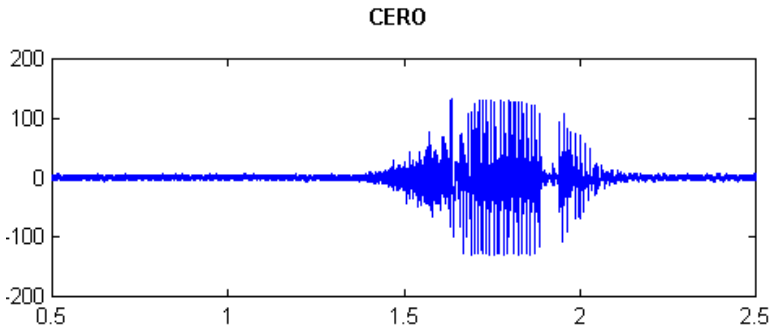
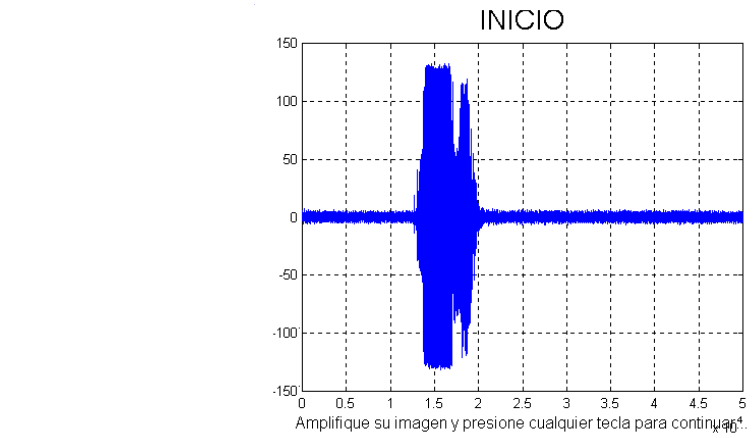
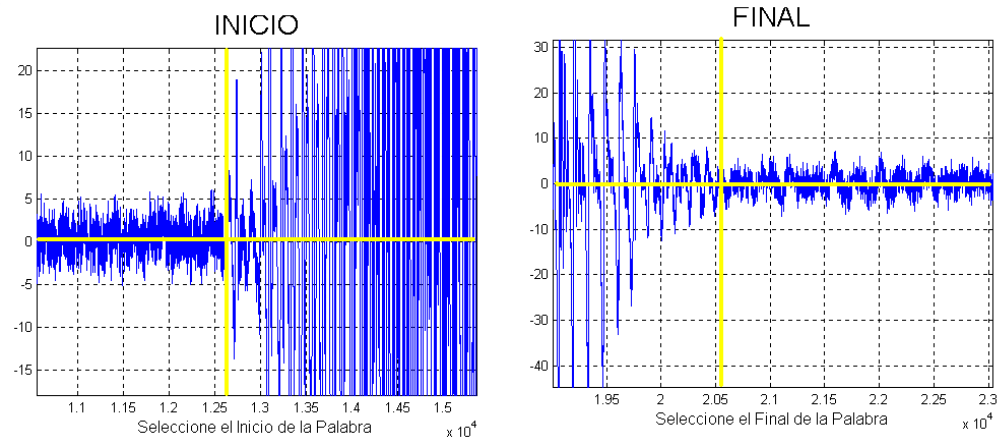


Figura 7.1 Muestra de la Palabra "Cero"

Como primer paso, se analiza la evolución de la función COPER en instantes previos y posteriores al inicio y fin de cada pronunciación. Para realizar esta operación, en Matlab se ha desarrollado el programa **AnalizaUmbral.m**, el cual muestra secuencialmente pantallas con cada una de las palabras previamente grabadas, y pide que manualmente se seleccione el inicio y fin de la palabra mediante el uso de mouse (Ver Figura 7.2).



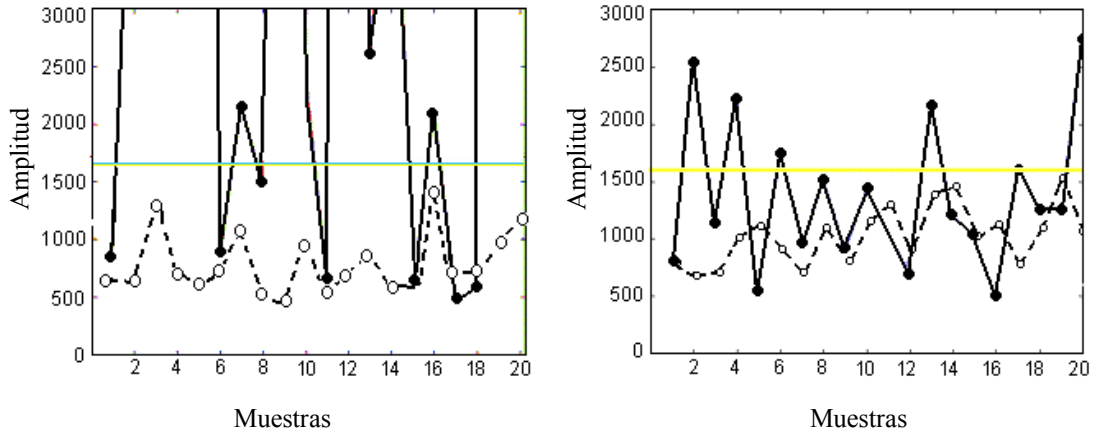
(a)



(b) (c)

Figura 7.2 Selección manual de extremos. (a) Palabra que va ser delimitada (b) Pantalla para seleccionar Inicio de Pronunciación (c) Pantalla para seleccionar Fin de Pronunciación

Inmediatamente después de seleccionado un punto de la gráfica, el programa almacenará en memoria información sobre la evolución de la función COPER una trama antes y otra después del punto seleccionado. Por cada palabra delimitada se obtendrán dos valores de la función COPER para el inicio de Pronunciación, y otros dos valores para el final de Pronunciación. Así luego de concluirse la delimitación de las palabras 20 palabras se obtendrán un total 40 puntos de análisis para el Inicio de Pronunciación, y otros 40 para el análisis del fin de Pronunciación. Con estos 2 conjuntos de valores se construyen dos gráficas que permiten analizar los umbrales (Ver Figura 7.3).



(a) (b)

Figura 7.3 Valores de la Función COPER (a) Valores de la Función COPER para el inicio de pronunciación (b) Valores de la Función COPER para el fin de pronunciación

La figura 7.3(a) muestra los valores de la función COPER instantes previos y posteriores al inicio de la pronunciación de cada palabra, la línea punteada muestra el conjunto de valores de la función COPER antes del inicio de la pronunciación. La línea continua, muestra los valores de la función COPER luego de haberse iniciado la pronunciación. Se aprecia que instantes previos a la pronunciación los valores se encuentran por debajo de 1600, mientras que, luego de iniciada la pronunciación, tiene valores muy superiores. Gráficamente se determina que el nivel de umbral de inicio CPui es igual a 1600.

La figura 7.3(b), muestra la gráfica de los valores de la función COPER instante previos y posteriores al final de la pronunciación. La línea punteada proporciona información sobre los valores de la función COPER, luego de finalizar la pronunciación. La línea punteada muestra los valores de la función COPER en un instante anterior al fin de la pronunciación. Se aprecia que luego del fin de pronunciación los valores de la función COPER posee valores por debajo de 1500, mientras que toma unos valores un poco más altos antes de que finalice la pronunciación. Gráficamente se determina que el nivel de umbral de final de pronunciación CPuf es igual a 1500.

7.1.1.2. Comparación entre COPER y Energía-Cruces por Cero

El objetivo de estas pruebas es comparar la Detección Automática de Extremos (delimitación) utilizando la función COPER frente al método de Energía - Cruces por Cero, para lo cual se realizarán pruebas de medida de la Precisión y el Tiempo de Procesamiento, en ambos casos, tal como se detallará a continuación.

7.1.1.2.1. Prueba de Precisión

Esta prueba consistió en medir la precisión obtenida en la delimitación automática de las palabras utilizando la Energía y Cruces por Cero frente al uso de la función COPER, teniendo como referencia los resultados obtenidos de una delimitación manual de las palabras. La delimitación manual, consiste en determinar los instantes de inicio y fin de la pronunciación de una palabra observando la gráfica de la señal en el tiempo, ajustando sus límites y escuchando su contenido sonoro.

Como se explicó en el ítem 3.2.3, durante el procesamiento de la Señal Voz, ésta es convertida en un vector denominado Patrón Característico, el cual es representativo de cada palabra. Por esto, para medir la similitud entre la delimitación manual y la automática (COPER y Energía-Cruces por Cero) se medirá el grado correlación entre los patrones característicos obtenidos para cada caso.

Para realizar esta prueba se almacenaron los dígitos del Cero al Nueve, bajo 7 niveles de ruido de fondo distintos, obteniéndose un total de 70 palabras a delimitar. En la tabla 7.1 se muestra la Relación Señal a Ruido (SNR) obtenida para cada uno de los niveles de ruido.

Tabla 7.1 SNR por cada nivel de ruido

Nivel	SNR
1	24dB – 32dB
2	23dB – 30dB
3	22dB – 28dB
4	20dB – 27dB
5	19dB – 27dB
6	18dB – 24dB
7	15dB – 22dB

Seguidamente se procedió a delimitar manualmente cada una de las palabras, luego se realizó la delimitación automática utilizando primero la Energía y Cruces por Cero, y después la función COPER, obteniéndose los patrones característicos para cada caso.

A continuación, se calcularon los coeficientes de correlación entre los patrones obtenidos de la delimitación manual y la delimitación automática utilizando la Energía y Cruces por Cero, los resultados se muestran en la figura 7.4, donde se puede apreciar que la correlación promedio entre los patrones característicos se mantiene con valores altos por arriba de 0.95 hasta SNR de 20 dB (Ver tabla 7.1), luego conforme la SNR disminuye, la correlación entre patrones también disminuye y empieza a caer por debajo de 0.9.

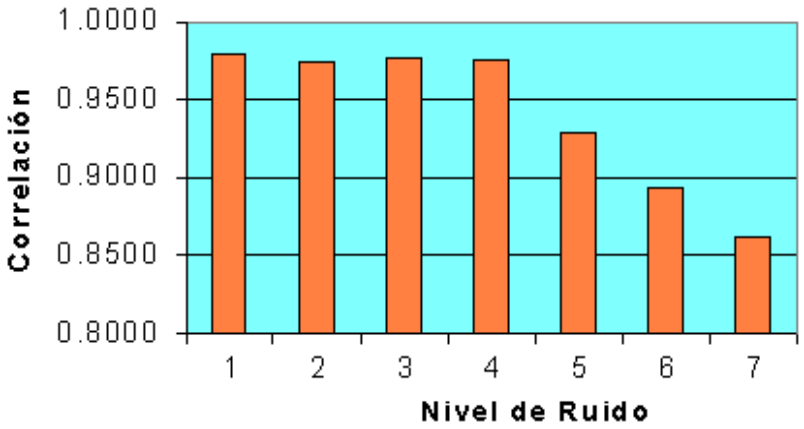


Figura 7.4 Promedio de la Correlación utilizando Energia-Cruces por Cero

Finalmente, se obtuvieron los coeficientes de correlación correspondientes a los patrones obtenidos de la delimitación manual y la automática utilizando función COPER, los resultados se muestran en la figura

7.5, donde se aprecia que para todos los casos la correlación promedio mantiene valores altos por encima de 0.95.

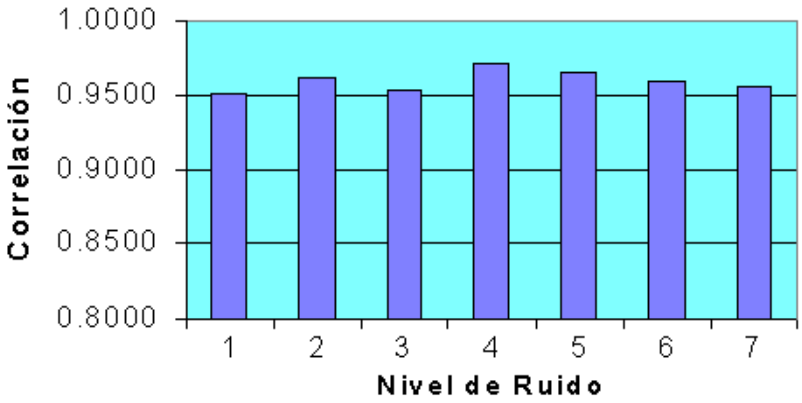


Figura 7.5 Promedio de la Correlación utilizando COPER

Así, comparando ambas gráficas, se concluye que los patrones característicos obtenidos utilizando la delimitación COPER muestran una mayor precisión, sobre todo en presencia de un ruido de fondo considerable. Para mayores detalles de la prueba se puede consultar las tablas mostradas en el Anexo A.

7.1.1.2.2. Tiempo de Procesamiento

Esta prueba consistió en medir el tiempo de procesamiento utilizado por cada una de dos los métodos, para lo cual, se creó un programa en Matlab denominado ComparaTiempos.m cuyos resultados se muestran a continuación.

```
********************
* Energia y Cruces *
********************
Tiempo =

   0.866999999999996

*********
* COPER *
*********
Tiempo =

   0.801999999999989

Resultado : Algoritmo COPER es más rápido
```

De los resultados obtenidos se puede apreciar que el algoritmo COPER posee un menor tiempo de procesamiento. Los detalles del código del programa ComparaTiempos.m se muestran en el anexo D.

7.1.1.3. Pruebas On Line

Esta prueba se realizó con el objetivo de verificar el funcionamiento *On-Line* del Detector Automático de Extremos utilizando la función COPER y su capacidad de poder delimitar palabras por debajo de una SNR de 30dB, determinando hasta que nivel de ruido de fondo el sistema opera correctamente. La prueba se llevó a cabo en una habitación de 4 x 4 x 5 m, en donde el ruido de fondo provenía de un equipo de música, la distancia entre el micrófono y los parlantes del equipo de música fue de aproximadamente 1 metro.

La prueba consistió en adquirir las palabras pronunciadas por el locutor (los dígitos de Cero al Nueve) bajo distintos niveles de ruido de fondo, y determinar la SNR para cada caso, los cuales se muestran en la tabla 7.2.

Tabla 7.2 Relación Señal a Ruido

Ruido de Fondo	SNR (dB)
Ruido 1	33 – 29
Ruido 2	27 – 22
Ruido 3	22 – 17
Ruido 4	15 – 20
Ruido 5	11 – 17

Estas pruebas experimentales fueron desarrolladas haciendo uso del RAV, utilizando el modo de Detección Automática en la Opción Grabar (Véase figura 7.6).

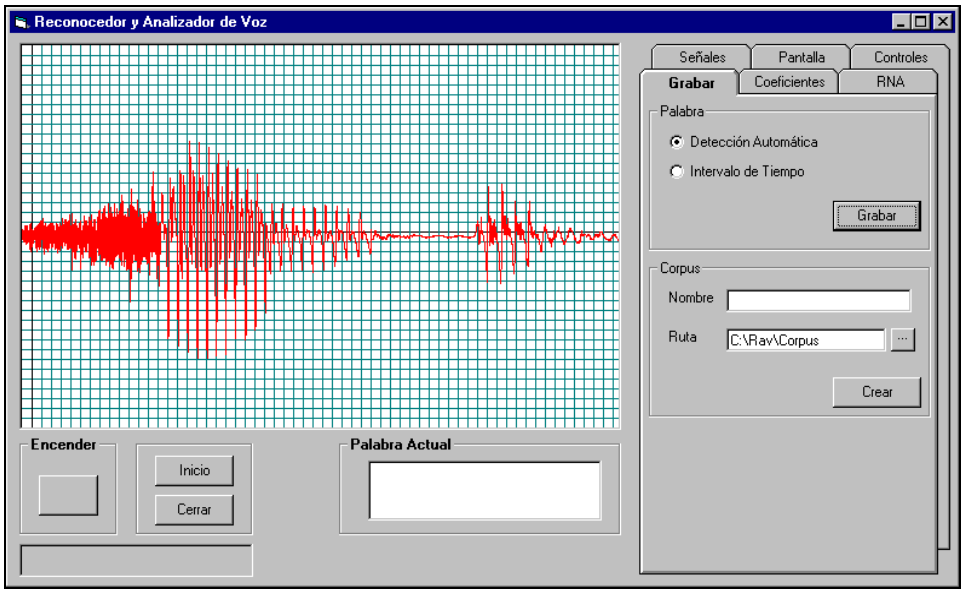


Figura 7.6 Pantalla de Grabación del RAV

Las pruebas experimentales, permitieron determinar que el sistema opera correctamente hasta una SNR de 15dB aproximadamente, luego cuando el ruido de fondo es más intenso y la SNR empieza a caer por debajo de los 15dB, la delimitación empieza a tener problemas ocasionando la saturación y deformación de la señal adquirida, tal como se puede apreciar en las figuras 7.7 y 7.8, en donde se muestran las gráficas resultantes luego del proceso de detección automático de extremos, para dos casos específicos. La figura 7.7 muestra la delimitación para una SNR entre 15 y 20 dB, y la figura 7.8 muestra las gráficas para SNR entre 11 y 17 dB, en donde se puede apreciar como en muchos casos la delimitación falla, debido al intenso ruido de fondo.

Además monitoreando las palabras delimitadas, y utilizando la opción Controles del RAV, se realizaron los ajustes finales de los niveles de umbrales, cuyos nuevos valores obtenidos son:

CPui = 1800

CPuf = 1600

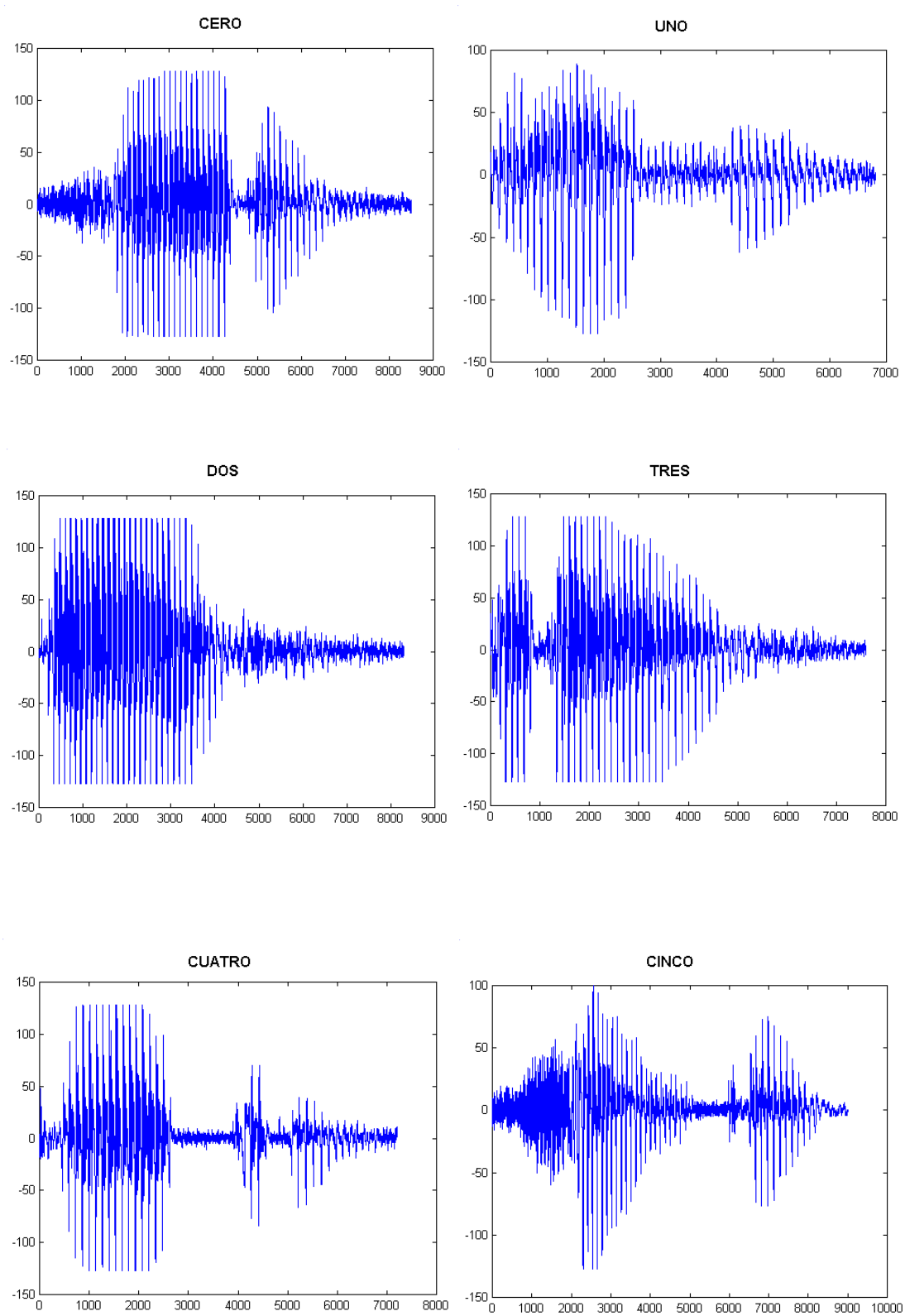


Figura 7.7 Palabra delimitadas para una SNR de hasta 15dB.

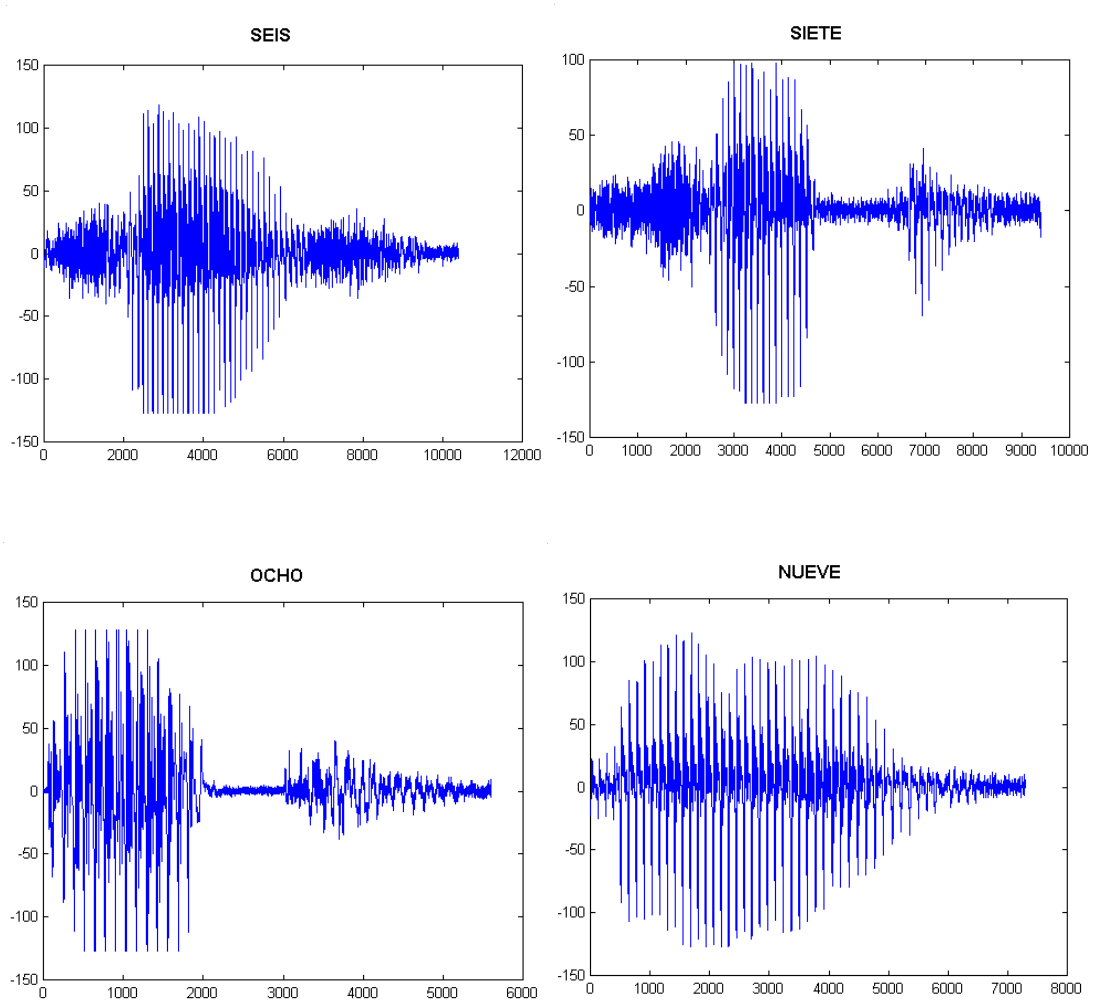


Figura7.7 (Continuación)

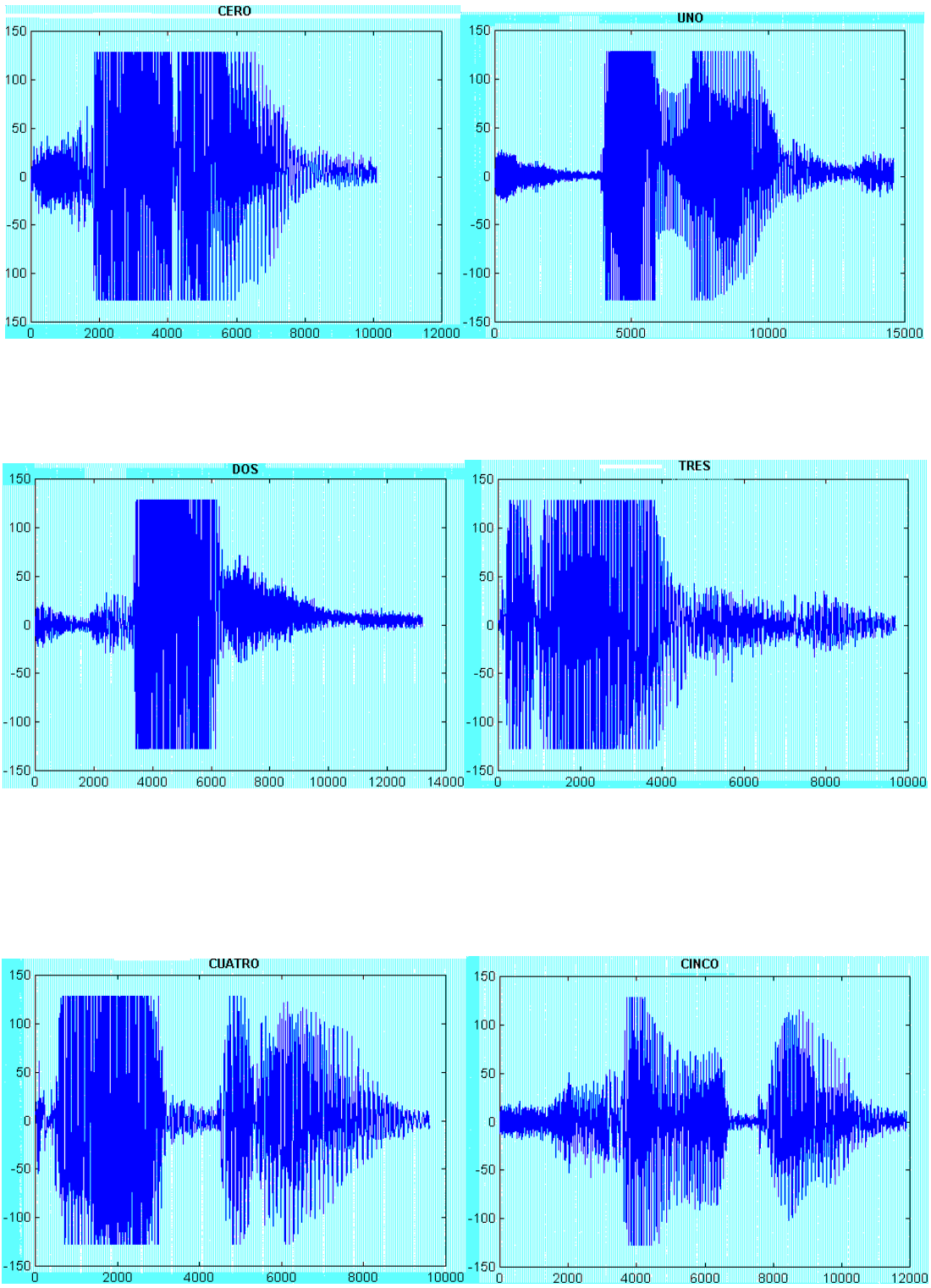


Figura 7.8 Palabra delimitadas para una SNR de hasta 11dB.

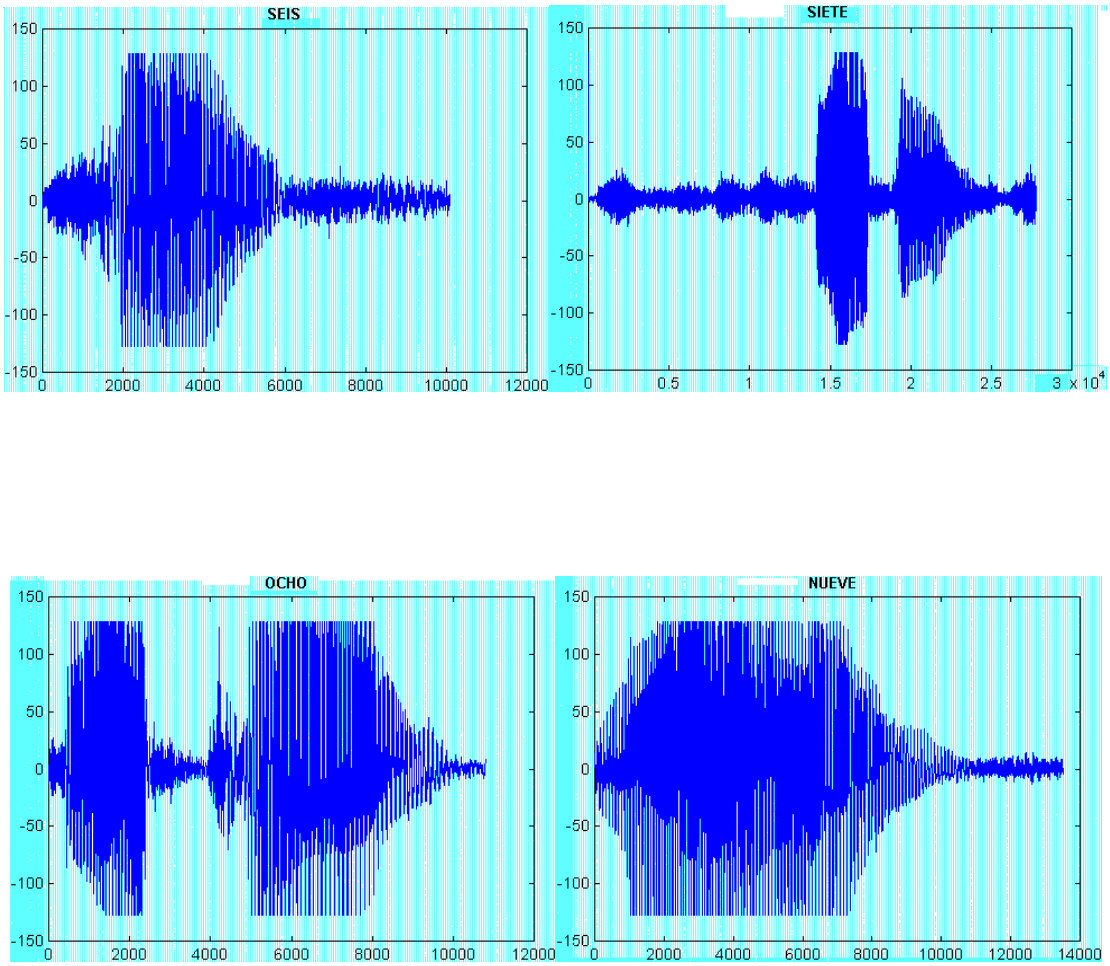


Figura 7.8 (Continuación).

7.1.2. Evaluación de los Parámetros de la RNA

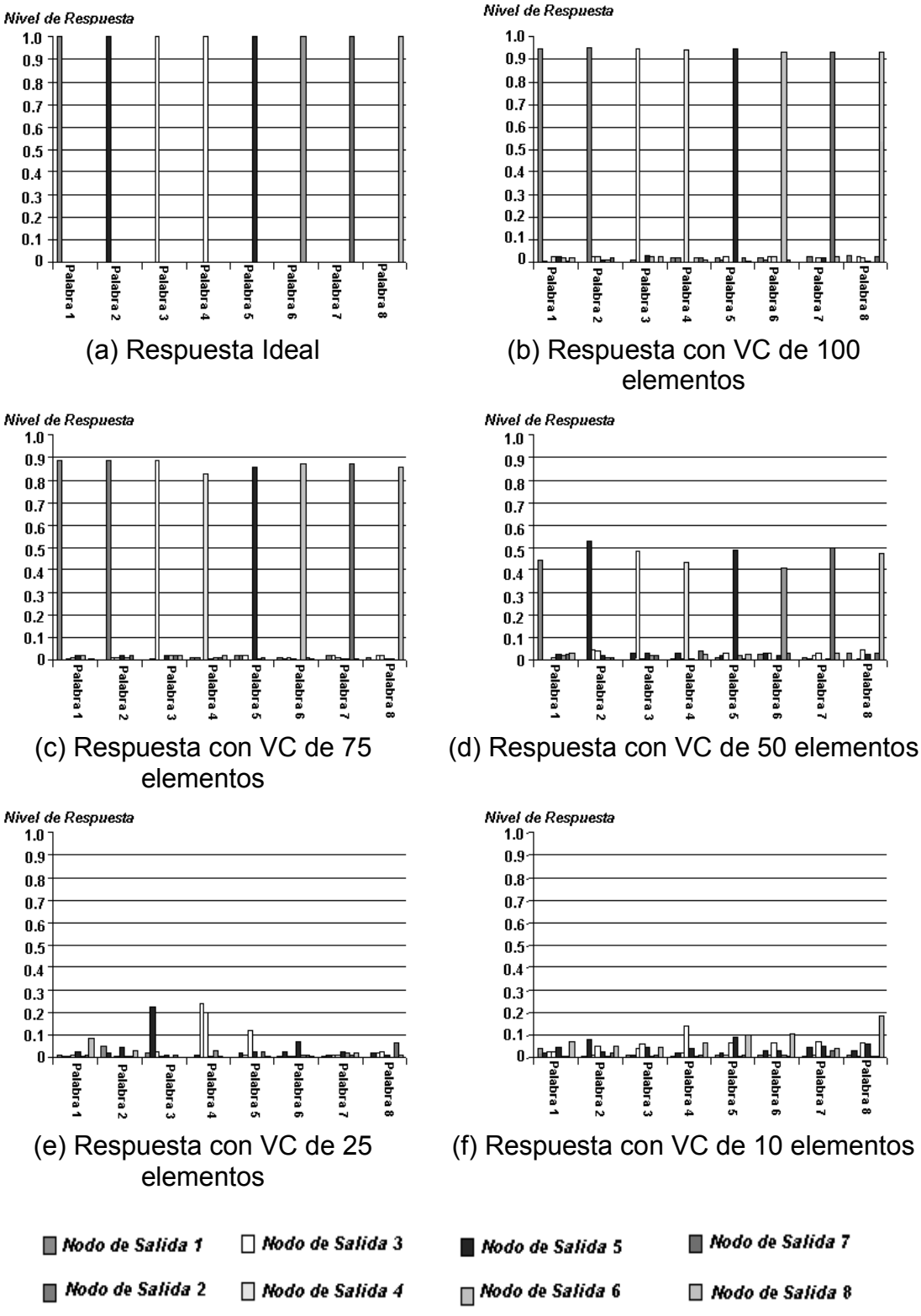
7.1.2.1. Número *de Elementos del Vector de Características*

Para determinar el número de coeficientes óptimo del Vector de Características (VC) se ha realizado cinco pruebas, en cada una de las cuales se varió el número de elementos de este vector (100, 75, 50, 25 y 10) y se procedió a implementar un Perceptrón Multicapa de 32 nodos ocultos, 8 nodos de salida y un numero de nodos de entrada igual al número de elementos de VC. Cada Perceptrón Multicapa fue entrenado para reconocer 8 palabras distintas (patrones).

En cada caso, la evaluación consiste en medir el grado de aprendizaje después de cuatro horas de entrenamiento para todo los casos. Se escogió el vector que ocasiono que las respuestas, después de este tiempo de entrenamiento, se acerquen a la ideal. Cabe agregar que como parte de esta evaluación se efectúa la propagación de la red con los ocho patrones de entrenamiento.

En la figura 7.9 se muestra el valor de las salidas de la red cuando son excitadas con los ocho patrones de entrenamiento para cada uno de los casos de evaluación, incluyendo el caso ideal que se muestra en la figura 7.9a. Los datos graficados se han obtenido de las tablas mostradas en el anexo B.

De los gráficos se puede observar que cuando el número de elementos del vector de características disminuye, los niveles de respuesta de salida se alejan del valor ideal. Podría escogerse entre 100 y 75 elementos del VC, ya que se acercan al valor ideal aceptablemente, pero se ha creído conveniente fijar este número de elementos en 100 porque presenta mayor eficiencia.



(a) Respuesta Ideal

(b) Respuesta con VC de 100 elementos

(c) Respuesta con VC de 75 elementos

(d) Respuesta con VC de 50 elementos

(e) Respuesta con VC de 25 elementos

(f) Respuesta con VC de 10 elementos

Figura 7.9 Valor en la neurona activa de la capa de Salida

7.1.2.2. Determinación del número *de Nodos Ocultos en el MLP*

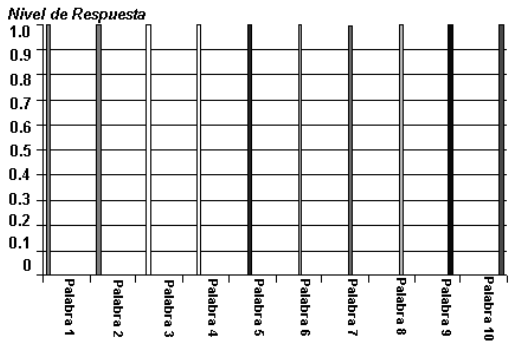
Para determinar el número óptimo (adecuado) de Nodos en la Capa Oculta se han realizado siete pruebas en las cuales se modelan redes con distintas cantidades de nodos en esta capa (150, 100, 80, 50, 30, 25 y 10) y se evalúa el nivel de aprendizaje de cada red 10 patrones distintos. Toda las redes tienen 100 nodos en la capa de entrada y 10 en la de salida. El criterio de evaluación utilizado es similar al de la prueba de Determinación del Número de elementos del Vector de Características.

La figura 7.10 muestra los gráficos de los valores de salida de la red en cada salida versus la palabra de excitación, para las distintas cantidades de nodos ocultos que se usaron en la evaluación. Las tablas correspondientes aparecen en el Anexo B. De estos gráficos se puede concluir que mientras más nodos ocultos posea la red, su respuesta de salida tiende a la ideal. Se podría inferir que se debería entrenar con la mayor cantidad de nodos ocultos posible, pero esto no es del todo óptimo, como analiza con la figura 7.11, en la cual se muestra el número de conexiones de la red versus el número de nodos en la Capa Oculta.

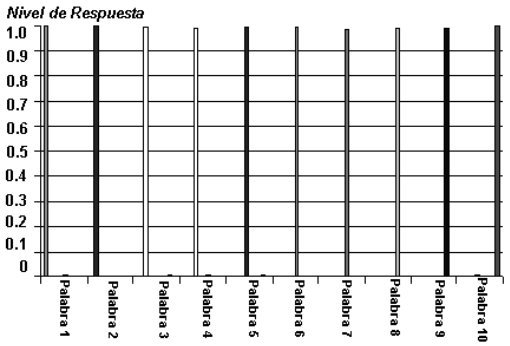
El número de conexiones del Perceptrón de Tres capas, en el cual todas las neuronas entrada (n1) están conectadas con toda las neuronas ocultas (n2) y éstas a su vez, están conectadas con todas las neuronas de salida (n3), viene dada por la siguiente expresión: n1xn2+n2xn3.

De las figuras 7.10 y 7.11 se observa que la respuesta de la red con 150 nodos ocultos esta muy próximo al ideal, pero esto implica 16000 conexiones en la red. Por otro lado, la red con 30 nodos en la capa oculta otorga una respuesta que esta alrededor de 0.9, que es un valor bastante aceptable, con la ventaja de que tiene aproximadamente 3000 conexiones, lo que significa que empleara menos tiempo de procesamiento (en concordancia con la ec. 5.8).

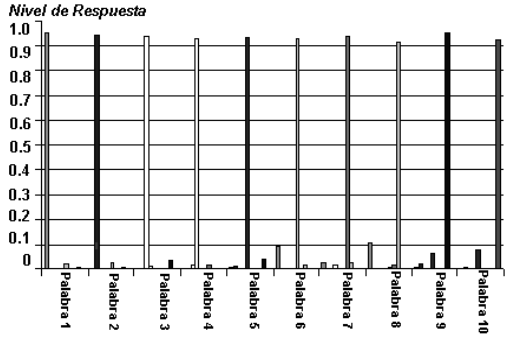
Por esta razón es que se ha creído conveniente usar alrededor de 30 nodos en la capa oculta del MLP.



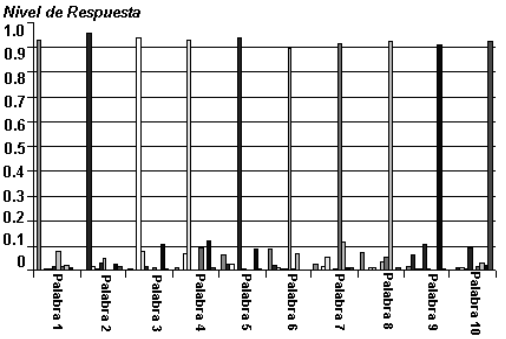
(a) Respuesta con 150 nodos ocultos



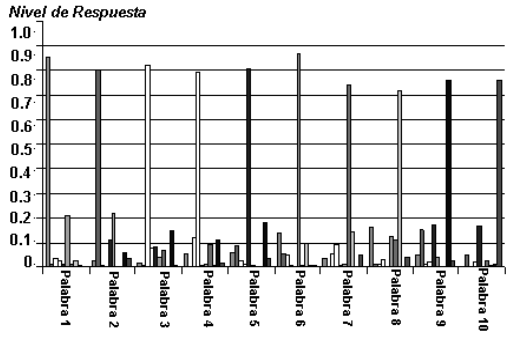
(b) Respuesta con 80 nodos ocultos



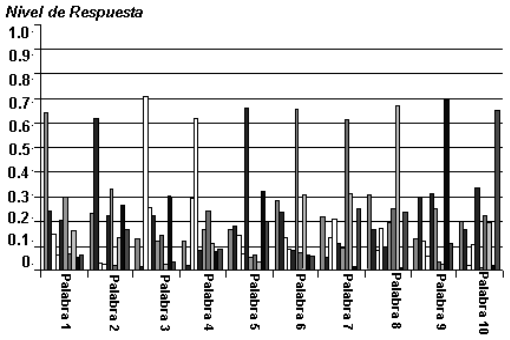
(c) Respuesta con 50 nodos ocultos



(d) Respuesta con 30 nodos ocultos



(e) Respuesta con 25 nodos ocultos



(f) Respuesta con 10 nodos ocultos

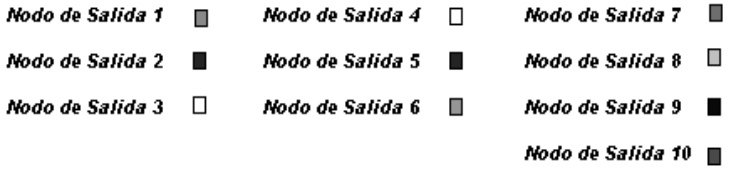


Figura 7.10 Valor en la neurona activa de la capa de Salida

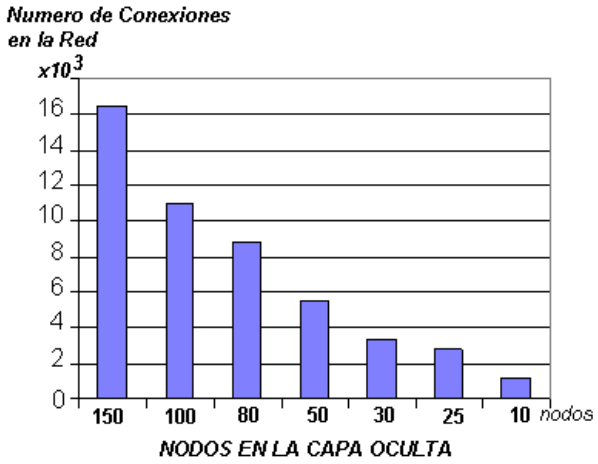


Figura 7.11 Conexiones de una red en función del número de nodos ocultos

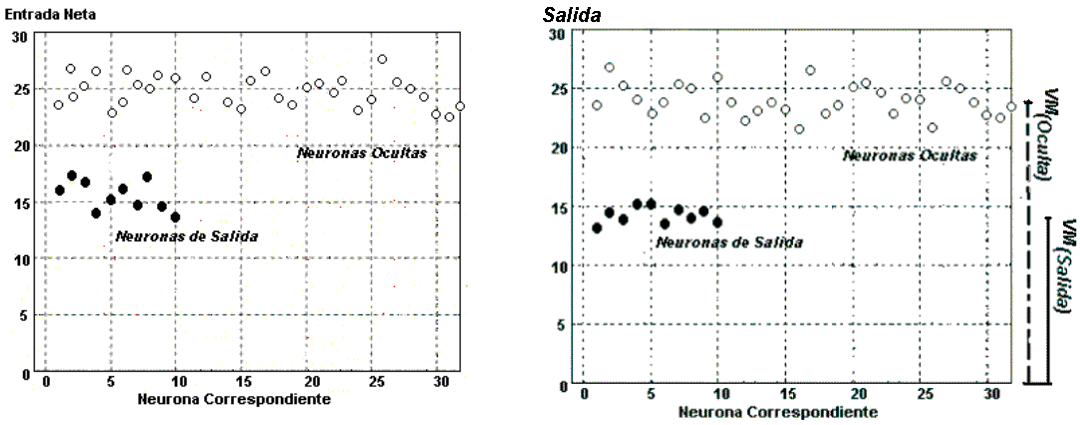
7.1.2.3. Elección de la Función de Transferencia en las Capa de la RNA

Esta prueba consiste en determinar la Función de Transferencia (FT) más óptima para las capas de la red. En el capítulo 6 se mencionó que la capa de entrada tiene como FT a la función lineal de pendiente unitaria, y que las capas Oculta y de Salida poseen la función Sigmoidal de constante k. El objetivo es determinar un valor adecuado de la constante k con la finalidad de reducir el tiempo de entrenamiento.

El Perceptrón Multicapa bajo prueba tiene 100 capa de Entrada 32 en la Oculta y 10 en la de Salida. La figura 7.12a muestra el resultado de la primera propagación de la red, cuando los pesos de conexión tienen valores aleatorios. Los 32 círculos de la parte superior muestran los valores de Entrada Neta (ecuación 5.1) que asumen las 32 neuronas de la capa Oculta; la abscisa del gráfico muestra la neurona correspondiente. Análogamente los 10 círculos negros de la parte inferior muestran la Entrada Neta de las 10 neuronas de la capa de Salida.

La figura 7.12b muestra la salida de las neuronas como respuesta a las Entradas Netas de la figura anterior por cada capa, como es obvio, tienen un comportamiento proporcional (ecuación 5.2). Obsérvese del gráfico que la

salida de cada capa tiende a un valor medio igual a $VM_{(Oculto)}$ para la capa Oculta y $VM_{(Salida)}$ para la capa de Salida



(a) **(b)**

Figura 7.12 Resultados de la primera propagación de la red. (a) Valores de Entrada Netas (b) Respuesta de Salida

Para que el entrenamiento de la red sea lo más rápido posible, el valor inicial de $VM_{(Salida)}$ debe estar en un valor equidistante entre Cero y Uno, las cuales son las salidas finales que se debe alcanzar, de esta manera el valor inicial podrá alcanzar su valor final en un tiempo corto. Se podría pensar que el valor de $VM_{(Salida)}$ inicial podría ser 0.5, pero esto no es cierto debido a que la Función de Transferencia de las capas son no lineales (ver figura 5.3e).

Se ha variado k1 y k2 de cada función de transferencia (de las capas Oculta y de Salida respectivamente) y se ha obligado a que las salidas tengan valores iniciales arbitrarios tal como se muestra en la tabla 7.3. A partir de estos valores se ha entrenado la red y se ha medido el tiempo, cuyo resultado se muestra en la misma tabla.

De aquí se puede apreciar que el menor tiempo de entrenamiento es de 130 segundos, que corresponde a un valor de k1 de 0.091551 y de k2 igual a 0.1569450, los cuales han sido adoptados en nuestro caso. Para mayor

claridad véase la figura 7.13 en donde se muestra la evolución de los tiempos de entrenamiento versus los valores establecidos de $VM_{(Salida)}$ y $VM_{(Oculto)}$

Tabla 7.3 Valores de prueba de $VM_{(Salida)}$

	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5	Prueba 6	Prueba 7
VM (Oculta)	0.6	0.7	0.8	0.85	0.9	0.95	0.99
VM (Salida)	0.6	0.7	0.8	0.85	0.9	0.95	0.99
k1 (Capa Oculta)	0.01689	0.03530	0.05776	0.07228	0.09155	0.12269	0.19146
k2 (Capa de Salida)	0.04224	0.07498	0.10830	0.12849	0.15695	0.19500	0.28720
Tiempo de Entrenamiento (seg)	538	291	164	152	130	153	380

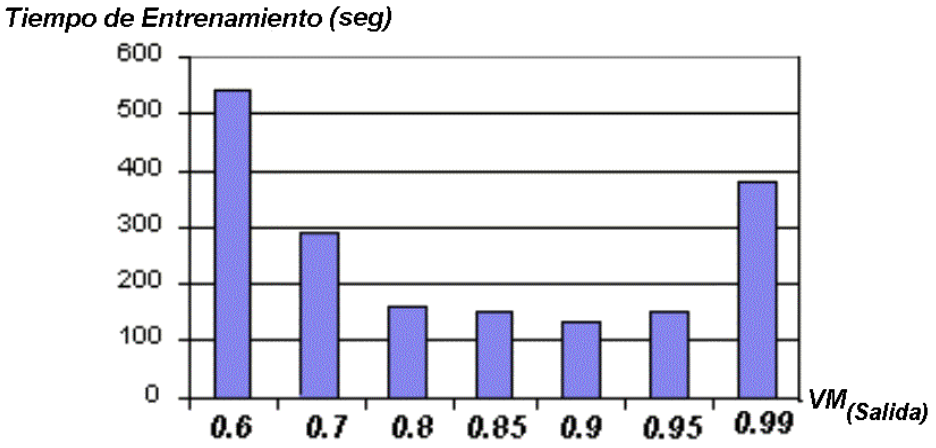


Figura 7.13 Evolución del Tiempo de Entrenamiento para dos patrones

7.2. Eficiencia del Sistema

En este apartado se muestran los resultados que corresponden a la prueba final del Sistema de Reconocimiento con el propósito de medir su eficiencia. Una vez que se realiza el entrenamiento del sistema, se calcula esta eficiencia en dos modos denominados *On Line y Off Line*. En el modo *Off Line* se mide la eficiencia con palabras cuyos patrones se encuentran guardados en archivos con la finalidad de comprobar que las condiciones establecidas para el entrenamiento del sistema han sido los correctos. En el modo *On Line,* se trabaja con palabras pronunciadas directamente por el locutor. La eficiencia en

ambos casos se calcula mediante el cálculo de la Tasa de Acierto, medida como la proporción entre el número de aciertos de palabras reconocidas y el número total de palabras utilizadas; su expresión matemática se presenta en la ecuación 7.1.

$$tasa\_de\_acierto\_\% = \frac{\#de\_aciertos}{\#de\_pruebas} \times 100 \qquad (7.1)$$

Las palabras pronunciadas fueron los 10 dígitos del idioma español (Cero, Uno, Dos, Tres, Cuatro, Cinco, Seis, Siete, Ocho, Nueve).

7.2.1. Entrenamiento del Sistema

7.2.1.1. Obtención de Muestras de Voz para el Entrenamiento

Se recolectaron un conjunto de 56 hablantes, los cuales han sido clasificados, teniendo en cuenta el sexo y edad, tal como se muestra en la tabla 7.4.

Tabla 7.4 Hablantes Recolectados

SEXO	M	F
Niño (8-11 años)	4	4
Adolescente (12-17 años)	6	6
Joven (18-30 años)	10	10
Adulto (30-60 años)	6	10
Total	**26**	**30**

Cada una de estas personas pronuncio las diez palabras establecidas para la prueba (dígitos del Cero al Nueve). Es decir, al momento de culminar la grabación, se contaba con una base de datos formada por 560 patrones característicos (muestras). Luego de la cual se efectúo un análisis de correlación de los patrones característicos con la finalidad de seleccionar a los

hablantes que comparten más características en común dentro del grupo [Llamas, Cardeñoso. 1995].

Para el análisis de correlación se utilizó la ecuación 7.2 [Chou, 1968], en la cual r_{xy} es el coeficiente de correlación entre las secuencias *x* e *y*.

$$r_{xy} = \frac{\sum x' y'}{\sqrt{\sum x'^2 \, y'^2}} \tag{7.2}$$

En donde:

$$x' = x - \bar{x} \tag{7.3}$$

$$y' = y - \bar{y} \tag{7.4}$$

Se aplicó esta fórmula a todos los patrones recolectados (secuencias), tomados de dos en dos, y aquellos que no alcanzaron un coeficiente de correlación mayor a 0.75 fueron considerados como patrones dispersos y fueron separados.

Como consecuencia, se obtuvo un nuevo grupo conformado por 8 hablantes, los cuales se muestran en la tabla 7.5.

Tabla 7.5 Hablantes Seleccionados

SEXO	M	F
Niño (8-11 años)	1	0
Adolescente (12-17 años)	0	1
Joven (18-30 años)	3	1
Adulto (30-60 años)	1	1
Total	**5**	**3**

Con palabras pronunciadas por estas personas se forma una nueva base de datos formada por 800 patrones correspondientes a 10 repeticiones por cada palabra; a esta base de datos que sirve para entrenar a la Red Neuronal se le denomina Corpus de Entrenamiento. A la base de datos

formada por los patrones de los 48 hablantes que no fueron seleccionados se le denomina Corpus de Evaluación y sirve para la evaluar el sistema en modo *Off Line* (véase ítem 7.2.2.1).

7.2.1.2. Resultados del Entrenamiento y Características de la RNA

En la tabla 7.6 se muestra los parámetros de la estructura del Perceptrón Multicapa, el cual consta de tres capas (una de Entrada, una de Salida y una Oculta). La tabla 7.7 muestra los parámetros y tiempo de entrenamiento de la Red Neuronal.

Tabla 7.6 Parámetros de la RNA

Capa	Entrada (1)	Oculta (2)	Salida (3)
Número de Nodos	100	32	10
Función de Transferencia	Lineal	Sigmoidal	Sigmoidal
Constante de la Sigmoidal (k)	...	0.09155	0.15695

Tabla 7.7 Parámetros de Entrenamiento.

Tiempo de Entrenamiento	8h 25min
Numero de Patrones	800
Numero de Palabras (nP)	10
Numero de Hablantes	8
Hablantes Relativos (nH)	80
Error de patron (ep)	0.025
Error de Hablante (mm)	0.0001
Error de grupo de hablantes (aa)	0.00001

Para definición de los parámetros que figuran en esta tabla, el lector puede revisar el ítem 6.2.4.

7.2.2. Medición de la Eficiencia del Sistema

7.2.2.1. Eficiencia Off Line

La eficiencia *Off Line* mide el desempeño del sistema con los patrones que pertenecen al Corpus de Evaluación con la finalidad de comprobar que los parámetros de la red modelada y el método de entrenamiento detallado en el capítulo 6 funcionan correctamente, así también acreditaría que el criterio de selección de hablantes que se usó para formar el Corpus de Entrenamiento ha sido el acertado.

La figura 7.14 muestra una gráfica de barras que expresa el resultado de la eficiencia *Off Line*. Cada una de estas barras indica la Tasa de Acierto obtenida en cada palabra. Como se puede observar, el sistema presentó una Tasa de Acierto de 100% en casi toda las palabras, excepto en las palabras 'Tres', 'Cuatro' y 'Seis', donde el índice es algo menor, pero aceptable.

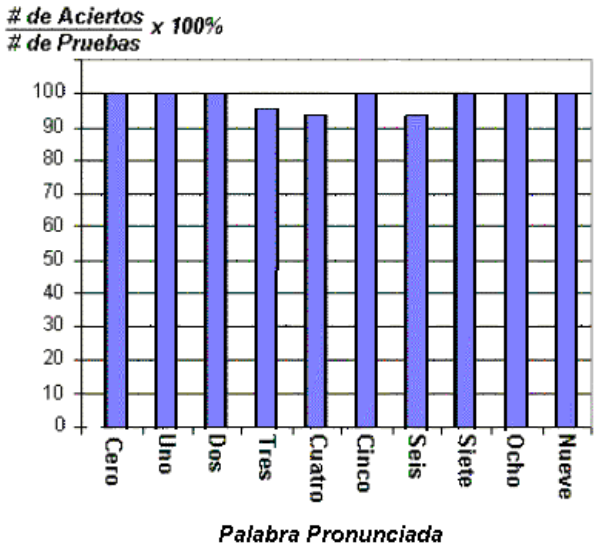


Figura 7.14 Resultados de la evaluación *Off Line*

La tasa de acierto global del sistema es de 98.125 %, lo cual indica que el resultado de la prueba fue exitoso. La tabla correspondiente a este gráfico se encuentra en el Anexo C.

7.2.2.2. Eficiencia On Line

El objetivo de esta prueba es medir la eficiencia del sistema en modo *On Line*, que constituye el principal objetivo del trabajo de tesis. Para realizar esta evaluación, se ha reunido a 20 personas, distribuidas en las categorías que se muestran en la tabla 7.8, quienes pronunciaban directamente la palabra.

Tabla 7.8 Hablantes de prueba

SEXO	M	F
Niño (8-11 años)	2	2
Adolescente (12-17 años)	2	3
Joven (18-30 años)	3	3
Adulto (30-60 años)	2	3
Total	**9**	**11**

Esta prueba se realizó en dos etapas; en la primera se evalúo el sistema en un entorno que se considera casi aislado del ruido, ya que el mayor nivel de ruido proviene de los ventiladores de la computadora; en este caso se obtuvo como resultado una Tasa de Acierto de 91.65 %. La tabla 7.9 muestra los resultados obtenidos, en la cual se detalla la eficiencia obtenida por cada categoría de hablante.

Tabla 7.9 Eficiencia *On Line* sin ruido

	# de Hablantes	# de palabras Pronunciadas	# de palabras Acertadas	Tasa de Acierto %
Total	20	2000	1833	91.65
Niño (8-11)	4	400	346	86.5
Adolescente (12-17)	5	500	465	93
Joven (18-30)	6	600	561	93.50
Adulto (30-60)	5	500	461	92.2
Promedio	**---**	**---**	**---**	**91.65**

La segunda etapa de la prueba consistio en evaluar el sistema en un ambiente con ruido (proveniente de un equipo de música) que producía una figura de ruido que oscilaba entre 20 dB y 30 dB; lógicamente en estas condiciones, la Tasa de Acierto del sistema disminuyó, pero no de manera considerable. ya que esta se situó en el valor de 87.4 %, tal como se muestra en la tabla 7.10

Tabla 7.10 Evaluación *On Line* con ruido

	# de Hablantes	# de palabras Pronunciadas	# de palabras Acertadas	Tasa de Acierto %
Total	20	2000	1748	87.4
Niño (8-11)	4	400	336	84
Adolescente (12-17)	5	500	445	89
Joven (18-30)	6	600	533	88.83
Adulto (30-60)	5	500	434	86.8
Promedio	**---**	**---**	**---**	**87.4**

Las figuras 7.15 y 7.16 muestran la eficiencia alcanzada por cada palabra pronunciada para la prueba *On Line*. La figura 7.15 corresponde a la evaluación del sistema aislado de ruido, y la figura 7.16 en un ambiente con ruido.

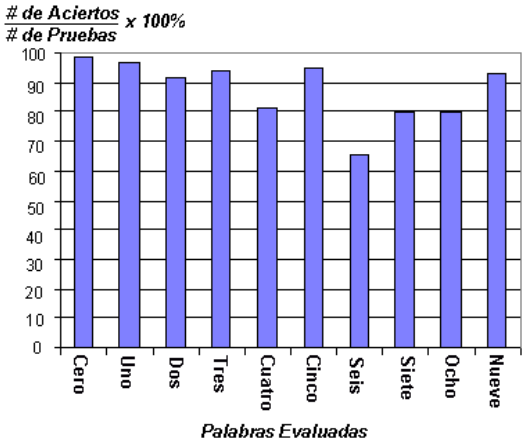


Figura 7.15 Eficiencia del reconocimiento *On Line* sin ruido

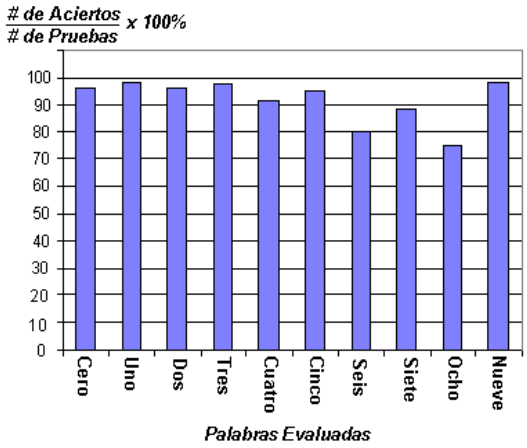


Figura 7.16 Eficiencia del reconocimiento *On Line* con ruido

8. CONCLUSIONES

- A partir de los cálculos sobre la eficiencia del sistema (Ver ítem 7.2), se puede concluir que el objetivo de diseñar un sistema de reconocimiento de voz independiente del locutor, ha sido logrado satisfactoriamente. Así, analizando la tabla 7.9, se puede observar que la Tasa de Acierto es de 91.65% cuando las pruebas fueron realizadas en un ambiente prácticamente sin ruido; por otro lado, en entornos ruidosos con figuras de ruido entre 15dB y 30dB se alcanzó una Tasa de Acierto de 87.4%, tal como lo muestran los resultados de la tabla 7.10. Demostrándose así, que el sistema opera eficientemente aún en presencia de ruido de fondo intensos.

- Se ha creado un algoritmo eficiente de adquisición de la Señal de Voz, con la finalidad de realizar el Reconocimiento de Voz *On Line*, al cual se le ha denominado NVENT(ver ítem 3.2.2.3).

- El algoritmo de detección de Extremos denominado COPER, desarrollado como otro aporte del presente trabajo, presenta gran eficiencia ya que permite detectar el inicio y fin de pronunciación de una palabra en entornos ruidosos que presentan una figura de ruido de hasta 15 dB (Véase ítem 7.1.1.2.1), además debido a que su formulación requiere pocas operaciones matemáticas, el tiempo de procesamiento de este algoritmo es pequeño, lo que permite su implementación en aplicaciones en tiempo real (Véase ítem 7.1.1.2.2).

- Se ha diseñado una interface gráfica amigable que permite analizar las características de la Señal de Voz, así como crear proyectos de Reconocimiento de Voz (con nuevas palabras por reconocer), teniendo como aplicaciones la Conversión Voz a Texto o la ejecución de Comandos a través de Voz (véase ítem 6.3.2), así como otras utilidades que hace de ésta interface una herramienta importante para futuros trabajos de Investigación, tanto al nivel de Pre o Post-Grado.

- Se ha aplicado tecnologías de punta en el proceso de Reconocimiento de Voz dando buenos resultados en la implementación del sistema. Así, se ha comprobado que la RNA tipo Perceptrón Multicapa (MLP) tiene una gran eficiencia como clasificador de patrones. Además, el uso de los Coeficientes Mel-Cepstrum, que representan de manera eficiente la información característica de cada palabra, ayudan a que el proceso de clasificación se lleve a cabo con gran efectividad.

- Se ha creado un algoritmo de entrenamiento del MLP, al que se le ha denominado “EBHA”, descrito en el capítulo 6. Los resultados que se aprecian en el gráfico de barras de la figura 7.12, muestran que la Tasa de Acierto del sistema evaluado en modo *Off Line* es de 98.125 % (muy cercano al ideal, 100%), demostrando que el algoritmo planteado tiene una gran efectividad en el establecimiento de clases y en el tiempo de entrenamiento (Véase la tabla 7.7).

- Se ha comprobado la eficiencia del criterio de selección del Corpus de Entrenamiento descritos en la sección 7.2.2.1, según los resultados de la tabla 7.13.

- Se ha probado que el tiempo de entrenamiento de un Perceptrón Multicapa de tres niveles, con función de transferencia Lineal en la capa de Entrada, y función Sigmoidal en las capas Oculta y de Salida depende directamente de los valores de la constante k de cada una de estas funciones sigmoidales. Esto se puede apreciar en la tabla 7.3. Por esto se ha elegido el valor de esta constante, de tal manera que las neuronas otorguen una respuesta de salida entre los valores de 0.85 y 0.95, tanto en la capa oculta como en la de salida.

9. RECOMENDACIONES Y PERSPECTIVAS

9.1 Recomendaciones

- La tarjeta de sonido de la PC es una herramienta eficiente y especializada para la adquisición de señales sonoras (de 20 Hz a 20000 Hz), por esto, se recomienda su uso en lugar de construir una etapa de adquisición propia. Se debe profundizar en la programación de éstas tarjetas, para así lograr desarrollar rutinas que permitan realizar la adquisición desde cualquier tarjeta de sonido, pudiéndose utilizar la API de Windows, o alguna herramienta equivalente en otros lenguajes de programación tales como Java. Con esto se lograría, mejores detalles en la adquisición de la señal de voz, como por ejemplo lograr una resolución de 16 bits y una adquisición más eficiente, probando con distintas frecuencias de muestreo.

- En el caso que se desee implementar el sistema con un procesador DSP debería diseñarse una etapa de acondicionamiento adecuado, para poder adquirir la señal con gran calidad, ya que pese a que las tarjetas de desarrollo de estos procesadores poseen su propia adquisición de datos, estás no están diseñadas específicamente para trabajar con voz. Para el diseño de la etapa de acondicionamiento se recomienda consultar la referencia bibliográfica[Cater. 1984]

- Se debe efectuar una selección previa de los patrones con los cuales se procederá a entrenar una red neuronal, ya que es necesario que patrones pertenecientes a una misma clase tengan gran similitud, de lo contrario podría ocasionar que la red no efectúe la correcta separación de clases.

9.2 Perspectivas

- Se podría realizar otros estudios experimentando otras técnicas de reconocimiento tales como los Modelos Ocultos de Markov o la Lógica Fuzzy para contrastar resultados con los del presente trabajo, también se puede realizar investigaciones con la Redes Neuronales Dinámicas (TDNN), las cuales vienen siendo utilizadas con eficiencia en reconocimiento de Voz, sobre todo para análisis fonético.

- Orientarse al Reconocimiento de Voz utilizando el análisis fonético, para así lograr un Reconocedor de palabras, de vocabulario ilimitado, y así también empezar a experimentar con el reconocimiento del habla continua.

- Implementar los algoritmos sobre un procesador DSP, para lograr aplicaciones en tiempo Real propiamente dicha, y poder optimizar los algoritmos, muchos de los cuales inclusive podrían ser implementados en circuitería de estado sólido, por ejemplo el algoritmo COPER o el método NVENT podrían ser implementados en VHDL.

- Podría diseñarse interfaces para conectar la línea telefónica a la tarjeta de sonido de la computadora, y poder realizar reconocimiento de voz a larga distancia.

10. ANEXOS

ANEXO A – Resultados de la correlación entre patrones característicos

Tabla 10.1 Correlación entre Patrones de delimitación Manual y COPER

	SNR (dB)						
Palabra	**24 - 32**	**23 - 30**	**22 - 28**	**20 - 27**	**19 - 27**	**19 - 24**	**15 – 22**
Cero	0.9495	0.9469	0.9479	0.9668	0.9507	0.9295	1.0000
Uno	0.9433	0.9457	0.9342	1.0000	0.9320	0.9253	0.9364
Dos	0.9897	0.9892	0.9892	0.9897	0.9904	0.9887	0.9898
Tres	0.9442	0.9892	0.9474	0.9846	0.9914	0.9875	0.9886
Cuatro	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Cinco	0.9480	0.9462	0.9444	0.9477	0.9632	0.9602	0.9702
Seis	0.9782	0.9809	0.9800	0.9755	0.9713	0.9764	0.9159
Siete	0.9442	0.9493	0.9399	0.9606	0.9693	0.9380	0.9299
Ocho	0.9353	0.9330	0.9464	0.9763	0.9378	0.9423	0.9379
Nueve	0.8883	0.9475	0.9049	0.9170	0.9506	0.9403	0.8947
Promedio	**0.9521**	**0.9628**	**0.9534**	**0.9718**	**0.9657**	**0.9588**	**0.9563**

Tabla 10.2 Correlación entre Patrones de delimitación Manual y *Energía y Cruces por Cero*

	SNR (dB)						
Palabra	**24 - 32**	**23 - 30**	**22 - 28**	**20 - 27**	**19 - 27**	**19 - 24**	**15 - 22**
Cero	0.9628	0.9601	0.9619	0.9477	0.8623	0.8877	0.8993
Uno	1.0000	1.0000	1.0000	1.0000	1.0000	0.7522	0.7177
Dos	1.0000	0.9892	1.0000	0.9904	1.0000	0.9627	0.8982
Tres	0.9884	0.9465	0.9893	0.9914	0.8480	0.8992	0.9096
Cuatro	1.0000	1.0000	1.0000	1.0000	1.0000	0.8317	0.7391
Cinco	0.9691	0.9682	0.9670	0.9632	0.8916	0.9380	0.9367
Seis	0.9903	0.9908	0.9904	0.9900	0.9429	0.9599	0.9746
Siete	0.9854	0.9862	0.9633	0.9761	0.9355	0.9416	0.9557
Ocho	1.0000	1.0000	1.0000	1.0000	1.0000	0.9139	0.7979
Nueve	0.8976	0.9006	0.8987	0.8996	0.8012	0.8442	0.7929
Promedio	**0.9794**	**0.9742**	**0.9771**	**0.9758**	**0.9282**	**0.8931**	**0.8622**

ANEXO B – Resultados obtenidos en la determinación de parámetros de la Red Neuronal

- **Determinación del Numero de elementos del Vector de Características**

Tabla 10.3 Resultados de la propagación para un VC de 100 elementos

		Palabra en Evaluacion							
		Palabra 1	**Palabra 2**	**Palabra 3**	**Palabra 4**	**Palabra 5**	**Palabra 6**	**Palabra 7**	**Palabra 8**
Nodos de Salida de la Red	*Nodo 1*	0.947	0.002	0.001	0.017	0.023	0.023	0.003	0.030
	Nodo 2	0.010	0.952	0.015	0.018	0.014	0.016	0.027	0.001
	Nodo 3	0.001	0.025	0.944	0.000	0.025	0.025	0.002	0.028
	Nodo 4	0.026	0.024	0.001	0.939	0.001	0.024	0.022	0.020
	Nodo 5	0.026	0.012	0.030	0.003	0.943	0.001	0.019	0.009
	Nodo 6	0.018	0.014	0.025	0.022	0.003	0.935	0.001	0.002
	Nodo 7	0.004	0.019	0.003	0.022	0.023	0.012	0.935	0.024
	Nodo 8	0.020	0.000	0.025	0.016	0.005	0.003	0.029	0.930

Tabla 10.4 Resultados de la propagación para un VC de 75 elementos

		Palabra en Evaluacion							
		Palabra 1	**Palabra 2**	**Palabra 3**	**Palabra 4**	**Palabra 5**	**Palabra 6**	**Palabra 7**	**Palabra 8**
Nodos de Salida de la Red	*Nodo 1*	0.889	0.000	0.003	0.011	0.017	0.014	0.000	0.014
	Nodo 2	0.000	0.886	0.010	0.016	0.017	0.009	0.019	0.001
	Nodo 3	0.006	0.015	0.884	0.000	0.017	0.016	0.018	0.022
	Nodo 4	0.010	0.012	0.000	0.824	0.001	0.009	0.014	0.019
	Nodo 5	0.019	0.020	0.018	0.004	0.860	0.001	0.010	0.005
	Nodo 6	0.018	0.016	0.017	0.015	0.004	0.874	0.010	0.006
	Nodo 7	0.001	0.017	0.017	0.016	0.013	0.014	0.874	0.003
	Nodo 8	0.008	0.001	0.017	0.018	0.002	0.008	0.006	0.862

Tabla 10.5 Resultados de la propagación para un VC de 50 elementos

		Palabra en Evaluacion							
		Palabra 1	**Palabra 2**	**Palabra 3**	**Palabra 4**	**Palabra 5**	**Palabra 6**	**Palabra 7**	**Palabra 8**
Nodos de Salida de la Red	*Nodo 1*	0.440	0.001	0.001	0.008	0.016	0.029	0.011	0.034
	Nodo 2	0.003	0.532	0.035	0.034	0.020	0.033	0.009	0.003
	Nodo 3	0.002	0.046	0.482	0.008	0.032	0.031	0.020	0.006
	Nodo 4	0.015	0.040	0.007	0.437	0.001	0.002	0.032	0.047
	Nodo 5	0.028	0.019	0.030	0.004	0.490	0.018	0.002	0.028
	Nodo 6	0.017	0.011	0.022	0.002	0.021	0.408	0.007	0.001
	Nodo 7	0.030	0.016	0.023	0.043	0.004	0.033	0.494	0.034
	Nodo 8	0.036	0.001	0.003	0.030	0.029	0.001	0.031	0.477

Tabla 10.6 Resultados de la propagación para un VC de 25 elementos

		Palabra en Evaluacion							
		Palabra 1	**Palabra 2**	**Palabra 3**	**Palabra 4**	**Palabra 5**	**Palabra 6**	**Palabra 7**	**Palabra 8**
Nodos de Salida de la Red	*Nodo 1*	0.015	0.057	0.017	0.001	0.001	0.008	0.009	0.003
	Nodo 2	0.007	0.020	0.229	0.011	0.019	0.028	0.011	0.021
	Nodo 3	0.005	0.003	0.026	0.240	0.016	0.006	0.013	0.023
	Nodo 4	0.015	0.004	0.004	0.204	0.120	0.006	0.011	0.027
	Nodo 5	0.026	0.045	0.015	0.004	0.024	0.075	0.027	0.011
	Nodo 6	0.008	0.007	0.002	0.031	0.003	0.011	0.021	0.002
	Nodo 7	0.015	0.009	0.014	0.004	0.026	0.011	0.012	0.065
	Nodo 8	0.089	0.034	0.001	0.001	0.007	0.009	0.020	0.013

Tabla 10.7 Resultados de la propagación para un VC de 10 elementos

		Palabra en Evaluacion							
		Palabra 1	**Palabra 2**	**Palabra 3**	**Palabra 4**	**Palabra 5**	**Palabra 6**	**Palabra 7**	**Palabra 8**
Nodos de Salida de la Red	*Nodo 1*	0.038	0.006	0.015	0.006	0.011	0.013	0.005	0.013
	Nodo 2	0.020	0.083	0.014	0.022	0.021	0.033	0.049	0.031
	Nodo 3	0.026	0.012	0.039	0.017	0.015	0.014	0.015	0.009
	Nodo 4	0.025	0.053	0.064	0.139	0.064	0.065	0.071	0.066
	Nodo 5	0.048	0.029	0.048	0.043	0.096	0.032	0.052	0.059
	Nodo 6	0.009	0.008	0.008	0.007	0.004	0.014	0.004	0.008
	Nodo 7	0.006	0.018	0.013	0.016	0.012	0.006	0.034	0.005
	Nodo 8	0.071	0.053	0.046	0.070	0.101	0.109	0.043	0.188

- **Determinación del Numero Nodos Ocultos del MLP**

Tabla 10.8 Resultados de la propagación con 150 nodos Ocultos

		Palabra en Evaluación									
		Cero	**Uno**	**Dos**	**Tres**	**Cuatro**	**Cinco**	**Seis**	**Siete**	**Ocho**	**Nueve**
Nodos de Salida de la Red	*Nodo 1*	0.9999	0.0000	0.0000	0.0000	0.0000	0.0010	0.0000	0.0016	0.0000	0.0000
	Nodo 2	0.0000	0.9999	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0000
	Nodo 3	0.0000	0.0000	0.9999	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Nodo 4	0.0000	0.0000	0.0001	0.9940	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Nodo 5	0.0000	0.0000	0.0000	0.0000	0.9994	0.0000	0.0000	0.0000	0.0006	0.0028
	Nodo 6	0.0083	0.0090	0.0000	0.0000	0.0000	0.9999	0.0000	0.0001	0.0000	0.0000
	Nodo 7	0.0000	0.0000	0.0000	0.0004	0.0000	0.0000	0.9979	0.0000	0.0000	0.0000
	Nodo 8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.9544	0.0000	0.0000
	Nodo 9	0.0000	0.0000	0.0039	0.0000	0.0065	0.0000	0.0000	0.0000	0.9992	0.0000
	Nodo 10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9990

Tabla 10.9 Resultados de la propagación con 100 nodos Ocultos

		Palabra en Evaluación									
		Cero	**Uno**	**Dos**	**Tres**	**Cuatro**	**Cinco**	**Seis**	**Siete**	**Ocho**	**Nueve**
Nodos de Salida de la Red	*Nodo 1*	0.9998	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001	0.0002	0.0000	0.0000
	Nodo 2	0.0000	0.9993	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Nodo 3	0.0000	0.0000	0.9998	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Nodo 4	0.0000	0.0000	0.0001	0.9979	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Nodo 5	0.0000	0.0000	0.0000	0.0000	0.9991	0.0000	0.0000	0.0000	0.0002	0.0002
	Nodo 6	0.0002	0.0002	0.0000	0.0000	0.0000	0.9989	0.0000	0.0000	0.0000	0.0000
	Nodo 7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9970	0.0000	0.0000	0.0000
	Nodo 8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0002	0.9990	0.0000	0.0000
	Nodo 9	0.0000	0.0000	0.0001	0.0000	0.0003	0.0000	0.0000	0.0000	0.9994	0.0000
	Nodo 10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9996

Tabla 10.10 Resultados de la propagación con 80 nodos Ocultos

		Palabra en Evaluación									
		Cero	**Uno**	**Dos**	**Tres**	**Cuatro**	**Cinco**	**Seis**	**Siete**	**Ocho**	**Nueve**
Nodos de Salida de la Red	*Nodo 1*	0.9977	0.0000	0.0000	0.0000	0.0000	0.0012	0.0002	0.0007	0.0005	0.0000
	Nodo 2	0.0000	0.9986	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0013	0.0000
	Nodo 3	0.0000	0.0000	0.9970	0.0012	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002
	Nodo 4	0.0000	0.0000	0.0011	0.9909	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000
	Nodo 5	0.0000	0.0000	0.0000	0.0000	0.9932	0.0000	0.0000	0.0000	0.0017	0.0039
	Nodo 6	0.0027	0.0003	0.0000	0.0000	0.0000	0.9931	0.0000	0.0001	0.0000	0.0000
	Nodo 7	0.0001	0.0000	0.0001	0.0028	0.0000	0.0000	0.9846	0.0004	0.0000	0.0000
	Nodo 8	0.0001	0.0000	0.0000	0.0000	0.0000	0.0020	0.0015	0.9904	0.0000	0.0000
	Nodo 9	0.0002	0.0000	0.0027	0.0001	0.0031	0.0000	0.0000	0.0000	0.9878	0.0000
	Nodo 10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9981

Tabla 10.11 Resultados de la propagación con 50 nodos Ocultos

		Palabra en Evaluación									
		Cero	**Uno**	**Dos**	**Tres**	**Cuatro**	**Cinco**	**Seis**	**Siete**	**Ocho**	**Nueve**
Nodos de Salida de la Red	*Nodo 1*	0.9800	0.0007	0.0001	0.0012	0.0072	0.0908	0.0255	0.1025	0.0050	0.0000
	Nodo 2	0.0005	0.9719	0.0001	0.0000	0.0121	0.0020	0.0000	0.0001	0.0213	0.0038
	Nodo 3	0.0002	0.0004	0.9404	0.0137	0.0003	0.0008	0.0020	0.0004	0.0009	0.0002
	Nodo 4	0.0008	0.0000	0.0110	0.8669	0.0012	0.0006	0.0148	0.0013	0.0002	0.0008
	Nodo 5	0.0000	0.0020	0.0011	0.0000	0.9257	0.0000	0.0000	0.0000	0.0615	0.0794
	Nodo 6	0.0215	0.0279	0.0004	0.0003	0.0000	0.8896	0.0001	0.0063	0.0007	0.0000
	Nodo 7	0.0004	0.0000	0.0004	0.0136	0.0000	0.0000	0.8928	0.0155	0.0000	0.0012
	Nodo 8	0.0005	0.0000	0.0000	0.0001	0.0000	0.0136	0.0266	0.8022	0.0000	0.0001
	Nodo 9	0.0029	0.0028	0.0340	0.0014	0.0432	0.0019	0.0001	0.0001	0.8870	0.0000
	Nodo 10	0.0000	0.0018	0.0004	0.0002	0.0008	0.0000	0.0017	0.0006	0.0001	0.9011

Tabla 10.12 Resultados de la propagación con 30 nodos Ocultos

		Palabra en Evaluación									
		Cero	**Uno**	**Dos**	**Tres**	**Cuatro**	**Cinco**	**Seis**	**Siete**	**Ocho**	**Nueve**
Nodos de Salida de la Red	*Nodo 1*	0.9258	0.0001	0.0066	0.0128	0.0594	0.1642	0.0277	0.1409	0.0141	0.0001
	Nodo 2	0.0011	0.8956	0.0017	0.0016	0.0234	0.0181	0.0006	0.0007	0.0625	0.0083
	Nodo 3	0.0065	0.0154	0.8668	0.0685	0.0249	0.0109	0.0176	0.0111	0.0029	0.0123
	Nodo 4	0.0058	0.0032	0.0785	0.8271	0.0015	0.0053	0.0508	0.0104	0.0050	0.0051
	Nodo 5	0.0178	0.0294	0.0136	0.0003	0.8370	0.0027	0.0001	0.0018	0.1591	0.1218
	Nodo 6	0.1345	0.1469	0.0017	0.0017	0.0063	0.7642	0.0029	0.0346	0.0034	0.0000
	Nodo 7	0.0160	0.0004	0.0093	0.0942	0.0007	0.0056	0.7422	0.0523	0.0025	0.0163
	Nodo 8	0.0190	0.0000	0.0000	0.0005	0.0000	0.0690	0.1116	0.7368	0.0000	0.0293
	Nodo 9	0.0099	0.0249	0.1018	0.1169	0.1688	0.0010	0.0086	0.0006	0.8812	0.0198
	Nodo 10	0.0001	0.0146	0.0054	0.0105	0.0042	0.0004	0.0106	0.0088	0.0046	0.9222

Tabla 10.13 Resultados de la propagación con 25 nodos Ocultos

		Palabra en Evaluación									
		Cero	**Uno**	**Dos**	**Tres**	**Cuatro**	**Cinco**	**Seis**	**Siete**	**Ocho**	**Nueve**
Nodos de Salida de la Red	*Nodo 1*	0.8492	0.0250	0.0157	0.0530	0.0544	0.1390	0.0348	0.1591	0.0465	0.0022
	Nodo 2	0.0127	0.7998	0.0026	0.0010	0.0852	0.0497	0.0019	0.0124	0.1518	0.0437
	Nodo 3	0.0357	0.0047	0.8181	0.1187	0.0263	0.0466	0.0516	0.0080	0.0122	0.0010
	Nodo 4	0.0248	0.0006	0.0778	0.7902	0.0107	0.0050	0.0900	0.0297	0.0212	0.0228
	Nodo 5	0.0109	0.1081	0.0828	0.0057	0.8048	0.0018	0.0054	0.0014	0.1694	0.1652
	Nodo 6	0.2121	0.2152	0.0390	0.0087	0.0075	0.8660	0.0124	0.1240	0.0386	0.0002
	Nodo 7	0.0094	0.0001	0.0664	0.0909	0.0015	0.0048	0.7391	0.1091	0.0002	0.0276
	Nodo 8	0.0261	0.0005	0.0001	0.0027	0.0000	0.0987	0.1429	0.7160	0.0000	0.0035
	Nodo 9	0.0062	0.0549	0.1483	0.1083	0.1792	0.0076	0.0014	0.0007	0.7604	0.0106
	Nodo 10	0.0016	0.0384	0.0068	0.0174	0.0354	0.0031	0.0453	0.0387	0.0239	0.7594

Tabla 10.14 Resultados de la propagación con 10 nodos Ocultos

		Palabra en Evaluación									
		Cero	**Uno**	**Dos**	**Tres**	**Cuatro**	**Cinco**	**Seis**	**Siete**	**Ocho**	**Nueve**
Nodos de Salida de la Red	*Nodo 1*	0.6430	0.2324	0.1301	0.1160	0.1658	0.2812	0.2146	0.3073	0.1264	0.1991
	Nodo 2	0.2422	0.6176	0.0156	0.0187	0.1795	0.2360	0.0495	0.1645	0.2988	0.1656
	Nodo 3	0.1476	0.0321	0.7075	0.2900	0.1422	0.1328	0.1349	0.0810	0.1162	0.0189
	Nodo 4	0.0628	0.0236	0.2541	0.6160	0.0670	0.0849	0.2094	0.1669	0.0557	0.1020
	Nodo 5	0.2054	0.2209	0.2183	0.0807	0.6613	0.0811	0.1088	0.0972	0.3135	0.3343
	Nodo 6	0.2998	0.3306	0.1205	0.1657	0.0512	0.6587	0.0918	0.1963	0.2517	0.0087
	Nodo 7	0.0690	0.0197	0.1424	0.2393	0.0592	0.0716	0.6083	0.2490	0.0342	0.2228
	Nodo 8	0.1579	0.1338	0.0278	0.1061	0.0353	0.3098	0.3129	0.6672	0.0258	0.1943
	Nodo 9	0.0493	0.2664	0.3051	0.0763	0.3228	0.0599	0.0178	0.0107	0.6912	0.0216
	Nodo 10	0.0633	0.1662	0.0356	0.0897	0.1989	0.0563	0.2507	0.2370	0.1063	0.6525

ANEXO C - Resultados de la Evaluación del Sistema

- **Resultados de la Evaluación *Off Line***

En esta sección del anexo se muestran los resultados de la evaluación del sistema en modo Off Line con los 10 dígitos del idioma español. La tabla 10.15 muestra los resultados obtenidos con los patrones que pertenecen al Corpus de Evaluación. En esta tabla la columna vertical muestra los dígitos a los cuales corresponden los patrones de evaluación y la horizontal corresponde a las palabras reconocidas por el sistema. Y la tabla 10.16 muestra los resultados de la evaluación con los 8 hablantes entrenados.

Tabla de evaluación de los hablantes no Entrenados (Corpus de Evaluación)

Tabla 10.15 Evaluación de los patrones no Entrenados: (48 hablantes)

		Palabra reconocida Por el Sistema										
		Cero	Uno	Dos	Tres	Cuatro	Cinco	Seis	Siete	Ocho	Nueve	% Acierto
Patrones de palabras Evaluadas	Cero	48										100
	Uno		48									100
	Dos			48								100
	Tres			2	46							95.83
	Cuatro					45				3		93.75
	Cinco						48					100
	Seis							45	3			93.75
	Siete								48			100
	Ocho									48		100
	Nueve										48	100

Total	**98.125**

Tabla de evaluación de los hablantes Entrenados

Tabla 10.16 Evaluación de los patrones Entrenados: (8 hablantes)

		Palabra reconocida Por el Sistema										
		Cero	Uno	Dos	Tres	Cuatro	Cinco	Seis	Siete	Ocho	Nueve	**% Acierto**
Patrones de palabras Evaluadas	Cero	8										100
	Uno		8									100
	Dos			8								100
	Tres				8							100
	Cuatro					8						100
	Cinco						8					100
	Seis							8				100
	Siete								8			100
	Ocho									8		100
	Nueve										8	100

Total	**100**

- **Resultados de la Evaluacion On Line**

En esta sección se muestran a manera de ejemplo, en mas detalle los resultados de la evaluación del sistema en modo *On Line*. El formato de las tablas utilizadas son similares a las utilizadas en la seccion anterior

La tabla 10.17 y 10.18 muestran los resultados obtenidos en el caso de un hablante que ha participado en el entrenamiento del sistema (sexo masculino y 24 años de edad). La tabla 10.17 muestra los resultados obtenidos en un ambiente aislado de Ruido y la tabla 10.18 en un ambiente con Ruido de fondo.

Tabla 10.17 Evaluacion de un Hablante Entrenado, sin Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	10										100
Uno		10									100
Dos			10								100
Tres				10							100
Cuatro					9				1		90
Cinco	1					9					90
Seis				1			9				90
Siete								10			100
Ocho					1				9		90
Nueve										10	100

%Total	**96**

Tabla 10.18 Evaluacion de un Hablante Entrenado, con Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	10										100
Uno		10									100
Dos			10								100
Tres				10							100
Cuatro	1				9						90
Cinco						10					100
Seis							7	3			70
Siete								10			100
Ocho									10		100
Nueve				1						9	90

%Total	**95**

Las tablas siguientes muestran los resultados obtenidos al evaluar el sistema con algunos hablantes que han sido considerados representativos

Tabla 10.19 Hablantes Representativos de la evaluación *On Line*

	Sexo	*Edad*
Hablante Evaluado 1	Masculino	23
Hablante Evaluado 2	Masculino	50
Hablante Evaluado 3	Femenino	52
Hablante Evaluado 4	Femenino	40
Hablante Evaluado 5	Masculino	10

Hablante Evaluado 1

Tabla 10.20 Resultados del Hablante Evaluado 1, sin Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	10										100
Uno		10									100
Dos			10								100
Tres				10							100
Cuatro					8				2		80
Cinco						10					100
Seis				1			7	2			70
Siete								10			100
Ocho	1								9		90
Nueve										10	100

%Total	**94**

Tabla 10.21 Resultados del Hablante Evaluado 1, con Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	10										100
Uno		10									100
Dos			9		1						90
Tres				10							100
Cuatro	3				7						70
Cinco						10					100
Seis				2			6	2			60
Siete	3							7			70
Ocho					1				9		90
Nueve				1						9	90

%Total	**89**

Hablante Evaluado 2

Tabla 10.22 Resultados del Hablante Evaluado 2, sin Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	9			1							90
Uno		10									100
Dos			6	1					3		60
Tres			1	9							90
Cuatro					10						100
Cinco						10					100
Seis				2			8				80
Siete								10			100
Ocho					1				9		90
Nueve										10	100

%Total	**91**

Tabla 10.23 Resultados del Hablante Evaluado 2, con Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	9							1			90
Uno		10									100
Dos			8						2		80
Tres			1	9							90
Cuatro		3			4				3		40
Cinco	1					9					90
Seis				4			6				60
Siete	2			2				6			60
Ocho									10		100
Nueve				1						9	90

%Total	**80**

Hablante Evaluado 3

Tabla 10.24 Resultados del Hablante Evaluado 3, sin Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	10										100
Uno		9		1							90
Dos			10								100
Tres				10							100
Cuatro					8				2		80
Cinco						10					100
Seis	1		1	1			7				70
Siete	2							8			80
Ocho					1				9		90
Nueve					1					9	90

%Total	**90**

Tabla 10.25 Resultados del Hablante Evaluado 3, con Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	10										100
Uno		9				1					90
Dos			10								100
Tres			1	9							90
Cuatro					10						100
Cinco						10					100
Seis				2			8				80
Siete	4							6			60
Ocho					6				4		40
Nueve					1					9	90

%Total	**85**

Hablante Evaluado 4

Tabla 10.26 Resultados del Hablante Evaluado 4, sin Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	10										100
Uno		10									100
Dos			10								100
Tres				10							100
Cuatro					10						100
Cinco						10					100
Seis							9	1			90
Siete								10			100
Ocho					5				5		50
Nueve										10	100

%Total	**94**

Tabla 10.27 Resultados del Hablante Evaluado 4, con Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	10										100
Uno		9			1						90
Dos			10								100
Tres				10							100
Cuatro	1				9						90
Cinco						10					100
Seis				1			7	2			70
Siete								10			100
Ocho					2				8		80
Nueve										10	100

%Total	**93**

Hablante Evaluado 5

Tabla 10.28 Resultados del Hablante Evaluado 5, sin Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	9				1						90
Uno		10									100
Dos			10								100
Tres				10							100
Cuatro					10						100
Cinco	2					8					80
Seis	1		1				8				80
Siete	3				2			5			50
Ocho					6				4		40
Nueve										10	100

% Total	**84**

Tabla 10.29 Resultados del Hablante Evaluado 5, con Ruido

	Cero	*Uno*	*Dos*	*Tres*	*Cuatro*	*Cinco*	*Seis*	*Siete*	*Ocho*	*Nueve*	*% Numero*
Cero	10										100
Uno		10									100
Dos			8		1				1		80
Tres			1	9							90
Cuatro					10						100
Cinco	2					8					80
Seis			1	2			5	2			50
Siete	1							9			90
Ocho					3				7		70
Nueve										10	100

%Total	**86**

ANEXO D –Programa ComparaTiempos.m

```
close all
clear all
clc

LenRuido=2000;
L=100;
nVent=20;

%%%%%%%%%%%
% Umbrales %
%%%%%%%%%%%

EUI=11000;
EUF=3500;
ZUI=100;
ZUF=80;
CPUI=1600;
CPUF=1200;

Pass=0;
t=0;

Palabra=csvread('vcp007.csv');
LenPalabra=length(Palabra);

Ruido=csvread('rcp001.csv');

PalabraRuido=Ruido(1:1:2000);
PalabraRuido(LenRuido+1:LenRuido+LenPalabra)=Palabra(1:LenPalabra)+
Ruido(LenRuido+1:LenRuido+LenPalabra);
PalabraRuidoBase=PalabraRuido(LenRuido+1:LenRuido+LenPalabra);
nc=length(PalabraRuidoBase);
Ruido0=Ruido(LenRuido+1:LenRuido+LenPalabra);
PalabraRuido(LenRuido+1+LenPalabra:LenRuido+1+LenPalabra+LenRuido+2000)=
Ruido(LenRuido+1+LenPalabra:LenRuido+1+LenPalabra+LenRuido+2000);

LenPalabraRuido=length(PalabraRuido);

% Algoritmo Energía y Cruces por Cero

Inicio=length(PalabraRuido)-1;
Fin=length(PalabraRuido);
Pass=0;

Tiempo1=cputime;

for i=1:L:LenPalabraRuido-L


   E=0;
   Z=0;

   for j=i:1:i+L
      if j==1
         E=E+PalabraRuido(1)*PalabraRuido(1);
         Z=Z+abs(sign(PalabraRuido(1)));
```

```
        else
            E=E+PalabraRuido(j)*PalabraRuido(j);
            Z=Z+abs(sign(PalabraRuido(j))-sign(PalabraRuido(j-1)));
        end
    end

    if(Pass==1)
        if(E<EUF & Z<ZUF)
            t=t+1;
            if t==nVent
                Fin=i-L*nVent;
                Pass=2;
                Tiempo2=cputime;
            end
        else
            t=0;
        end
    end

    if((E>EUI | Z>ZUI) & Pass==0)
        Inicio=i;
        Pass=1;
    end
end

PalabraDelimitada=PalabraRuido(Inicio:1:Fin);

disp('********************')
disp('* Energia y Cruces *')
disp('********************')
disp('')
disp('Tiempo')
TiempoEC=Tiempo2-Tiempo1

Pass=0;
t=0;

% Algoritmo COPER

Inicio=length(PalabraRuido)-1;
Fin=length(PalabraRuido);
Pass=0;

Tiempo1=cputime;

for i=1:L:LenPalabraRuido-L
    CP=0;
    for j=i:1:i+L
        if j==1
            CP=CP+abs(PalabraRuido(1)*abs(PalabraRuido(1)));
        else
            CP=CP+abs(PalabraRuido(j)*abs(PalabraRuido(j))-
            PalabraRuido(j-1)*abs(PalabraRuido(j-1)));
        end
    end

    if(Pass==1)
        if(CP<CPUF)
            t=t+1;
```

```
            if t==nVent
               Fin=i-L*nVent;
               Pass=2;
               Tiempo2=cputime;
            end
         else
            t=0;
         end
      end

      if(CP>CPUI & Pass==0)
         Inicio=i;
         Pass=1;
      end
   end

PalabraDelimitadaCOPER=PalabraRuido(Inicio:1:Fin);

disp('*********')
disp('* COPER *')
disp('*********')
disp('')
disp('Tiempo')
TiempoCP=Tiempo2-Tiempo1

disp('')

if (TiempoEC-TiempoCP>0)
   disp('Algoritmo COPER es mas rápido')
elseif (TiempoEC-TiempoCP==0)
   disp('Algoritmo COPER es mas rápido')
elseif (TiempoEC-TiempoCP<0)
   disp('Algoritmo Energía y Cruces por Cero es mas rápido')
end
```

11. BIBLIOGRAFÍA

- John G. Proakis, Dimitris G. Manolakis. "Tratamiento digital de señales". 3a Edición. Editorial Prentice Hall. 1998.
- Boaz Porat. "A Course in Digital Signal Processing". John Wiley & Sons, Inc. 1997.
- Vinay K. Ingle, John G. Proakis. "Digital Signal Processing Using Matlab V.4". PWS Publishing Company. 1997.
- Jesús Bernal Bermúdez, Jesús Bobadilla Sancho, Pedro Gómez Vilda. "Reconocimiento de Voz y Fonética Acústica". Ediciones Alfaomega. 2000.
- Cesar Llamas Bello, Valentín Cardeñoso. “Reconocimiento Automático del Habla. Teoría y Aplicaciones”. Universidad de Valladolid. 1995.
- Andrés Flores Espinoza, “Reconocimiento de Palabras Aisladas en Castellano”, Inictel. Dirección de Investigación y Desarrollo. 1993.
- Shuzo Saito, Kazuo, Kazuo Nakta. "Fundamentals of Speech Signal Processing". Academic Press. 1985.
- John P. Cater. "Electronically Hearing: Computer Speech Recognition" 1st Edition. Howard W. Sams & Co., Inc. 1984.
- Stamatios V. Kartalopoulos, Ph. D. "Understanding Neuronal Networks and Fuzzy Logic". IEEE Press. 1996.
- José Ramón Hilera González, Víctor José Martínez Hernando. "Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones". Editorial Addison-Wesley Iberoamericana. 1995.
- Freeman J.A., Skapura D.M., “Redes Neuronales, Algoritmos, Aplicaciones y Técnicas de Programación”, ADDISON-WESLEY. 1993.
- Francisco Charte. "Programación con Visual Basic 5.0". Ediciones Anaya Multimedia. 1997.
- Microsoft Press, Microsoft Visual Basic 6.0. Manual del Programador, Mc. Graw-Hill.
- Chuck Sphar. "Aprenda Microsoft Visual C++ 6.0 Ya". Mc. Graw-Hill. 1999.
- Herbert Schildt. “Turbo C/C++. Manual de Referencia”.

- Duane Hanselman, Bruce Littlefield. "Matlab edición de estudiante. Guía de usuario Versión 4". Editorial Prentice Hall. 1996.
- Robert Resnick, David Halliday. “Física para estudiantes de Ciencias e Ingeniería. Parte 1”. /ma edición. John Wiley & Sons, Inc. 1965.
- Ya-Lun Chou. “Análisis Estadístico”. Editorial Interamericana. 1968.
- C. Crespo Casas, C. de la Torre Munilla, J.C. Torrecilla Merchán. "Detector de extremos para reconocimiento de voz". Telefónica Investigación y Desarrollo. Publicación de Telefónica I+D. S:A. Madrid España 2001. EDICIONES ON-LINE : http://www.tid.es/presencia/publicaciones/comsid/esp/home.html Miércoles, 10 de enero de 2001, 10:22 horas.
- M, J. Poza Lara, J. F. Mateos Díaz, J. A. Siles Sánchez. " Design of an isolated-word recognition system for the Spanish Telephone Network". Telefónica Investigación y Desarrollo. Publicación de Telefónica I+D. S:A. Madrid España .2001. EDICIONES ON-LINE : http://www.tid.es/presencia/publicaciones/comsid/esp/home.html Miércoles, 10 de enero de 2001, 10:33 horas.
- L. Hernández Gómez, F. J. Caminero Gil, C. de la Torre Munilla, L. Villarrubia Grande." Estado del arte en Tecnología del Habla**".** Telefónica Investigación y Desarrollo. Publicación de Telefónica I+D. S:A. Madrid España .2001. EDICIONES ON-LINE : http://www.tid.es/presencia/publicaciones/comsid/esp/home.html Miércoles, 10 de enero de 2001, 16:06 horas.
- M. J. Poza Lara, L. Villarrubia Grande, J A. Siles Sánchez. " Teoría y aplicaciones del reconocimiento automático del habla**".** Telefónica Investigación y Desarrollo. Publicación de Telefónica I+D. S:A. Madrid España .2001. EDICIONES ON-LINE : http://www.tid.es/presencia/publicaciones/comsid/esp/home.html Jueves, 11 de enero de 2001, 12:28 horas.
- Jay Land. Isolated Word Speech Recognition of the English Digits. http://www.gerc.eng.ufl.edu/STUDENTS/Jland/proj1/proj1.html Jueves, 11 de enero de 2001, 12:20 horas.

- Minh N. Do . "Digital Signal Processing Mini-Project: An Automatic Speaker Recognition System". Audio Visual Communications Laboratory . Swiss Federal Institute of Technology, Lausanne, Switzerland. http://lcavwww.epfl.ch/~minhdo/asr_project/asr_project.html Jueves, 11 de enero de 2001, 12:15 horas.
- Alberto Cid Esteban, Norberto Mateos Carrascal. "Reconocimiento de Voz mediante el Perceptrón Multicapa". Laboratorio de Electroacústica II (SSR). Curso 95-96. E.T.S.I.T.M. http://www.intersaint.org/acid/rvpm0.htm Martes, 23 de enero de 2001, 10:13 horas.
- Creative Technology Ltd. "Sound Blaster Series. Hardware Programming Guide". http://developer.creative.com/scripts/DC_D&H_Games-Downloads.asp?opt=3 Lunes, 4 de Junio del 2001, 17:56 horas
- Borja Azpiroz. “Acústica Básica” 1997. http://personal.redestb.es/azpiroz/ Sábado, 19 de Mayo del 2001, 12:27 horas
- Takuya Ooura. “General Purpose FFT (Fast Fourier/Cosine/Sine Transform) Package” 1996-1998 http://momonga.t.u-tokyo.ac.jp/~ooura/fft.html Lunes, 26 de Marzo del 2001. 22:13 horas.