Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica

Instituto Costarricense de Electricidad

ICE

Reconocimiento de voz

Informe de Proyecto de Graduación para optar por el grado de Bachiller en Ingeniería Electrónica

Roberto Calvo Arias

Cartago, Enero del 2002

Dedicatoria

Dedico este trabajo de una forma muy especial a mi familia que siempre a estado ahí, apoyando sin importar lo largo y duro que sea el camino y nunca me han dejado solo ni por un momento. Gracias por todo el amor, cariño y esfuerzo que le han inyectado a mi vida.

A mi hija por ser la fuente de inspiración en todos los momentos de mi vida y la razón de seguir luchando día con día sin desmayar a pesar de que muchas veces el camino se torna muy difícil. Gracias por todas las sonrisas y los abrazos que me das, razón más que suficiente para seguir luchado. Este trabajo va dedicado especialmente para ti.

Agradecimientos

Agradezco a Moya y a Alex, por permitirme trabajar con ellos y crear un habiente de trabajo confortable. Gracias porque hicieron crecer mi confianza y pude trabajar a gusto. Gracias porque siempre sentí apoyo y confianza en el trabajo que realizaba.

Un agradecimiento muy especial para Bernal, por el regaño que me hizo seguir adelante el día en que quise tirar la toalla.

Resumen

Este proyecto sobre Reconocimiento de voz se desarrollo con el objetivo de crear un precedente sobre el tema, utilizable como documento de soporte y referencia en futuras investigaciones. Gira en torno a dos puntos principales: una investigación teórica y la implementación de un algoritmo de reconocimiento de palabras aisladas.

La etapa investigativa trata generalidades sobre Tecnología del Habla, principalmente las técnicas de extracción de parámetros, y se dedica un capítulo al estudio de los Procesadores Digitales de señales (DSP).

En la etapa de implementación se trabajo con dos algoritmos diferentes, utilizando como unidad gramatical la palabra. La primera con un algoritmo de programación dinámico DTW y la segunda con una red neuronal.

El sistema con algoritmo dinámico, consta de dos fases: una fase de entrenamiento, donde se crea una base de datos con las palabras a reconocer y la fase propia de reconocimiento.

El sistema con Red Neuronal, mostró excelentes resultados, acompañado de una fácil realización y simplicidad de reentrenamiento para cambiar el conjunto de palabras a reconocer.

Palabras claves: reconocimiento de voz, tecnología del habla, técnicas de extracción de parámetros, DSP, algoritmo de programación dinámica DTW, red neuronal.

Abstract

This is a project about speech recognition, had been developing with de objective to establish a precedent to be utilizable on future investigations, like a document of reference and support. Base on two main points: a theory investigations and an implementation of an algorithm to recognition of isolated words.

The investigation part is about speech technologies, speech signal processing and a DSP study, this ending because is an important tool in a real implementation.

On the implementations, the work was base on two techniques. The first implementation characteristic was the use of DTW algorithm, and the second one used a neural network.

The system with the first technique is based on two phases, the training phase, where the database with the words to recognition is created, and the recognition phase itself.

The Neural Network system was more flexibility and with less functional blocks, which makes implementation simplify. Without requiring a database this system is easy to training for the recognition of different groups of words.

Keywords: speech recognition, speech technology, signal processing, DSP, DTW, neural networks.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPITULO 1
INTRODUCCIÓN

1.1 DESCRIPCIÓN DE LA EMPRESA

A continuación se brinda una descripción general de la empresa donde se desarrollo el proyecto.

1.1.1 DESCRIPCIÓN GENERAL

El Instituto Costarricense de Electricidad (ICE) es una empresa costarricense, la cual fue creada como institución autónoma en el año 1949 y concebida desde su origen como el ente rector y principal ejecutor del desarrollo y administración de la industria eléctrica nacional.

Con la creación del ICE, el país avanzó notablemente en el desarrollo del sector eléctrico, en la segunda mitad del siglo XX. En los primeros 10 años se triplicó la capacidad instalada nacional y hoy en día la misma es superior al millón de kilovatios, o sea 30 veces mayor que la que se tenía en 1950. Asimismo, el servicio es confiable, de alta calidad, de cobertura nacional, sustentado en el uso racional de los recursos energéticos nacionales y soportado en políticas y planes de expansión a largo plazo.

El ICE actualmente brinda servicios en las áreas de energía y telecomunicaciones. Para brindar un mejor servicio al cliente, se encuentra dividido en dos grandes sectores: ICE-Energía e ICE-Telecomunicaciones.

ICE-Energía se encuentra subdividido en 6 Unidades Estratégicas de Negocios (UEN), las cuales son:

- Producción de electricidad
- Transporte de electricidad
- Servicio al cliente
- Proyectos y servicios asociados
- Centro nacional de control de energía
- Centro nacional de planificación de energía

1.1.2 DESCRIPCIÓN DEL DEPARTAMENTO DONDE SE REALIZÓ EL PROYECTO

El proyecto se realizó para el Centro de Servicio Investigación y Desarrollo, el cual pertenece a la UEN Proyectos y Servicios Asociados, del ICE Energía.

La Unidad Estratégica de Negocios Proyectos y Servicios Asociados es un área estratégica del ICE Energía, en la cual se llevan a cabo proyectos que le permiten al ICE continuar siendo líder en el desarrollo del Sistema Eléctrico Nacional y Regional, proyectando su capacidad dentro y fuera de las fronteras de Costa Rica, con un compromiso con el medio ambiente.

Los proyectos incluyen el diseño y construcción de hardware electrónico y también el desarrollo de aplicaciones de software en diferentes plataformas de desarrollo (Unix, Dos, Windows).

Este Centro de Servicio cuenta con los siguientes laboratorios: electrónica de corrosión, circuitos impresos y sistemas de potencia.

1.2 DEFINICIÓN DEL PROBLEMA Y SU IMPORTANCIA

En este apartado se describe la necesidad a solventar y la importancia que tiene para la empresa la realización de este.

1.2.1 DESCRIPCIÓN DEL PROBLEMA

El departamento donde se realizó la práctica, no cuenta con un estudio sobre la tecnología del reconocimiento de voz, a la cual hacer referencia y tomarse como base, para solventar dudas en un desarrollo futuro de una aplicación real de reconocimiento de voz.

1.2.2 IMPORTANCIA DEL PROBLEMA

El ICE es una institución concebida desde su origen como el ente rector y principal ejecutor del desarrollo y administración de la industria eléctrica nacional. Para seguir manteniendo esta postura vanguardista en un mundo tan cambiante e innovador, se debe invertir en la investigación de nuevas tecnologías y la aplicabilidad de estas, a la realidad nacional.

El Centro de Investigación y Desarrollo está destinado a la realización de herramientas a la medida y de bajo costo, que sean propiedad del ICE, para el uso inmediato o futuro.

La tecnología del reconocimiento automático de voz, a pesar de tener medio siglo de investigación, esta encontrando su mayor auge en estos años. El inmenso número de aplicaciones de un proyecto de esta envergadura, aplicables a necesidades con que cuenta actualmente el ICE, justifica y convierte en una necesidad inmediata un estudio profundo de esta tecnología.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar un algoritmo para el reconocimiento automático de voz.

1.3.2 OBJETIVOS ESPECÍFICOS

1. Realizar una investigación sobre las técnicas actuales empleadas en tecnología de reconocimiento automático de voz.

2. Realizar una Investigación sobre las principales características arquitectónicas y funcionales de los procesadores digitales de señales (DSP).

3. Diseñar un algoritmo para el reconocimiento de palabras aisladas.

4. Escribir un informe técnico del proyecto para la empresa y el informe de proyecto de graduación para el Instituto Tecnológico de Costa Rica**.**

CAPITULO 2
ANTECEDENTES

2.1 REQUERIMIENTOS DE LA EMPRESA

El resultado esperado al finalizar este proyecto consta de tres secciones principales:

1. Una investigación sobre las técnicas empleadas para el procesamiento automático de señales de voz.

2. Un estudio sobre las principales características arquitectónicas y funcionales, de los procesadores digitales de señales (DSP).

3. Un sistema para reconocimiento de palabras aisladas.

Este proyecto forma parte de una iniciativa para introducir la investigación de la tecnología del habla, en específico, reconocimiento automático de voz, para posibles aplicaciones de está tecnología, en equipos ya existentes.

Como parte de un plan piloto, este proyecto debe introducir al desarrollo de tecnologías económicas y realizadas propiamente por parte del ICE, en lo que a tratamiento de habla se refiere.

Además, introducir el uso de procesadores digitales de señales, dadas las enormes ventajas que brinda esta herramienta para el procesamiento de señales en tiempo real, superiores a las ofrecidas por los microprocesadores o microcontroladores y principalmente, por ser la herramienta con mejores prestaciones para la implementación real.

2.2 ANTECEDENTES DEL PROYECTO

El ICE no cuenta con un estudio previo sobre reconocimiento de voz; sin embargo, las aplicaciones de reconocimiento de voz son ampliamente estudiadas, en la actualidad se cuenta con aplicaciones comerciales de reconocimiento de voz en diferentes campos.

Haciendo una breve reseña histórica sobre los avances de esta tecnología, desde sus inicios, en los años cincuentas, hasta la década de los noventas y principios del siglo 2000, se tiene lo siguiente:

La investigación de tecnologías en reconocimiento de voz empezó a finales de los 50's con la llegada del computador digital. Esto combinado con herramientas para capturar y analizar la voz, (convertidores de señal análogo-digital y espectrogramas de sonidos), permitió a los investigadores buscar otras maneras de extraer características de la voz que mostraran las diferentes propiedades de las palabras.

En los 60's se dieron avances en la segmentación automática de voz en unidades lingüísticas relevantes (fonemas, sílabas y palabras) y en los algoritmos de pattern-matching y clasificación.

Por los 70's surgió un número de técnicas importantes hasta la fecha, creadas en parte por investigación de “Defense Advanced Research Projects Agency” (DARPA). Se hicieron reconocedores que manejaban un dominio de reconocimiento mayor y que estaban basados en el enfoque de reconocimiento de patrones.

La década de los 80´s trajo consigo un cambio del enfoque de reconocimiento de patrones hacia métodos de modelado probabilístico. A principios de los años 80 más de 10 compañías de Estados Unidos ofrecían reconocedores de palabras aisladas dependientes del locutor con un vocabulario de hasta 300 palabras. Sólo las firmas VERBEX y NEC ofrecían un sistema independiente del locutor con posibilidades de reconocimiento de palabras conectadas. En ese momento, la situación del Reconocimiento del Habla podría resumirse de la siguiente manera:

- Reconocedores de palabras aisladas dependientes del locutor como tecnología asentada.
- Reconocedores independientes del locutor y reconocedores de palabras conectadas, como tecnologías nacientes.

Por otro lado, debido a las limitaciones en el ancho de banda y la sensibilidad frente al ruido, sólo un número muy reducido de estos reconocedores trabajaba sobre la línea telefónica. En esta época sólo encontramos en la literatura referencia a tres aplicaciones del Reconocimiento del Habla dentro del ámbito de las telecomunicaciones:

Dos prototipos de reconocedores de palabras aisladas independientes del locutor, para aplicaciones como:

- Marcación por voz en la red privada.
- Reconocimiento de letras.
- Un reconocedor de palabras aisladas dependiente del locutor aplicado a la marcación de números de teléfono por voz.

En los 90´s la innovación tecnológica ha permitido una notable mejoría en sistemas de reconocimiento de voz. Unido a esto, las técnicas de hace algunos años han sido refinadas hasta el grado que actualmente se han obtenido muy buenos resultados de reconocimiento, y hay en el mercado sistemas comerciales a precios razonables, actualmente son muchas las compañías que cuentan con reconocedores de palabras aisladas (dígitos, más un número reducido de comandos) independiente del locutor. Sistemas diseñados, en su mayor parte, para incorporarse en aplicaciones de telecomunicaciones.

Las prestaciones obtenidas para palabras aisladas, vocabularios con un número de palabras inferior a 200, e independencia del locutor, dependen en gran medida de las características acústicas de las palabras del vocabulario. Así, mientras que el reconocimiento de los diez dígitos puede presentar una tasa de error de palabra inferior al 2%, el reconocimiento de 39 caracteres alfanuméricos (dígitos y letras) en inglés supone un 7% de error.

El reconocimiento de dígitos conectados es otra de las tareas con mayores posibilidades de utilización en diversas aplicaciones. Los resultados que proporcionan los mejores sistemas desarrollados para el inglés, por los Laboratorios Bell de AT&T y por el Centro de Investigación Informática de Montreal (CRIM), suponen una tasa de error de palabra inferior al 1% cuando trabajan en condiciones de laboratorio. Sin embargo, sobre la red telefónica la taza se reduce de forma importante, hasta tasas de error de palabra cercanas al 4%.

Entre los proyectos experimentales más importantes anteriores a 1994, se encuentran:

- **BYBLOS**

  Desarrollado por BBN. Byblos es el nombre de una ciudad fenicia donde se descubrió la primera muestra de escritura fonética. Este detalle marca el énfasis que se pone actualmente en desarrollar sistemas sobre una base fonética. Aunque se trata de un sistema dependiente de locutor, este sistema ha aportado un nuevo y eficiente procedimiento de reconocimiento rápido (búsqueda rápida) basado en algoritmos N-best.

- **TANGORA**

  Desarrollado en IBM. También se trata de un sistema dependiente de locutor para grandes vocabularios. Su principal interés es un proceso de adaptación a un nuevo locutor que requiere 20 minutos para leer 100 frases de 1.200 palabras, 700 de las cuales son distintas.

- **SPHINX-II**

  Desarrollado en la Universidad de Carnegie-Mellon (CMU). Es un sistema pionero en reconocimiento independiente de locutor para grandes vocabularios. Su más reciente innovación es el procedimiento VOCIND para hacer al sistema independiente del vocabulario.

- **LINCOLN**

  Desarrollado en el laboratorio del mismo nombre. Su principal aportación es el modelado de voz rápida, con emoción, tensión, etc.

- **DECIPHER**

  Desarrollado en SRI International. Su principal novedad fue la representación detallada de aspectos fonéticos importantes, tales como la coarticulación entre palabras.

- **ATR HMM-LR**

  Sistema japonés desarrollado en ATR. Está basado en procedimientos específicos de modelado de sonidos que no utilizan estructuras intermedias de modelos de fonema o palabra.

- **CSELT**

  Desarrollado en el centro italiano del mismo nombre. Su principal innovación es un sistema de búsqueda rápida basada en un primer descifrado fonético simple y rápido, seguido por una búsqueda más detallada.

- **PHILIPS**

  Desarrollado por la empresa del mismo nombre. Es un sistema pionero en procesos de reconocimiento rápidos para habla continua y vocabularios de hasta 10.000 palabras.

Sistemas telefónicos de AT&T y Bell Northern Research (BNR), ambos incorporan procedimientos específicos para aplicaciones de automatización de servicios telefónicos.

2.3 SOLUCIÓN PROPUESTA

Para poder alcanzar los resultados finales establecidos para el proyecto, se proponen los siguientes pasos:

Realizar una investigación sobre las diferentes técnicas actuales utilizadas para el reconocimiento automático del habla y como se combinan estas técnicas con diferentes tecnologías existentes para el procesamiento digital de señales acústicas. Brindar una reseña sobre las diferentes variables a contemplar para realizar un modelo para procesamiento de voz, y los diferentes enfoques que se tienen al respecto.

El procesamiento digital de señales requiere una demanda masiva de cálculos; realizar una aplicación de reconocimiento de voz en tiempo real requiere de una herramienta especializada para el procesamiento de señales. Los DSP son herramientas especialmente diseñadas para estos fines, por lo que una implementación de un sistema de reconocimiento automático del habla podría hacer uso de uno de estos procesadores, por ello, en este proyecto se incluye un capítulo sobre DSP, cuyo fin es conocer a fondo las diversas características que disponen estos procesadores y que sirva de guía de referencia para su selección, dependiendo de la aplicación, por lo que se exponen diferencias arquitectónicas y de desempeño entre diferentes casas fabricantes.

Diseño de dos reconocedores, como comprobación de las técnicas expuestas. El primero utilizando un algoritmo de programación dinámico DTW, en un DSP y el segundo empleando una red neuronal, en el software de simulación Matlab.

CAPITULO 3
PROCEDIMIENTO METODOLÓGICO

1. Se realizará una investigación bibliográfica sobre las diferentes técnicas que se utilizan en la actualidad en el ámbito de la tecnología del habla, prestando atención principalmente en las líneas orientadas hacia el diseño de sistemas de reconocimiento. La investigación tendrá como fin principal mostrar un panorama de la forma en que se logra procesar la voz, conociendo los diferentes aspectos y variables que se deben considerar al momento de desarrollar una aplicación de este tipo; además, de una descripción de las principales técnicas de modelado empleadas en la actualidad.

2. Se realizará una investigación sobre los procesadores digitales de señales (DSP), haciendo énfasis principalmente en las características arquitectónicas, de desempeño, manejo de memoria, interfaces de entrada / salida, programación, consumo y costo, que permitan establecer las ventajas y desventajas de estos sobre lo microcontroladores y los microprocesadores, para poder definir los casos en los que sea más conveniente la utilización de un DSP, que una de las otras herramientas. Con esta información se confeccionará una guía que presente las principales diferencias entre los diferentes fabricantes, que servirá como referencia para la selección del procesador DSP más idóneo para la aplicación que se desee implementar.

3. Diseñar el algoritmo de reconocimiento que emplea DTW.

4. Estudiar la programación del DSP-2181 de Analog Device.

5. Diseñar el algoritmo de reconocimiento que emplea la red neuronal.

6. Estudiar la programación en Matlab.

7. Estudiar la programación en SIMULINK.

8. Realizar el informe final del proyecto.

CAPITULO 4
RECONOCIMIENTO AUTOMÁTICO DEL HABLA

4.1 INTRODUCCIÓN

El proceso de reconocimiento automático del habla (RAH), dota a las máquinas con la capacidad de recibir mensajes orales. Tomando como entrada la señal acústica recogida por un micrófono, el objetivo final, es decodificar el mensaje contenido en la onda acústica, para realizar las acciones pertinentes. Para lograr este fin, un sistema de RAH necesita conjugar una gran cantidad de conocimientos acerca del sistema auditivo humano, sobre la estructura del lenguaje, la representación del significado de los mensajes y sobre todo el autoaprendizaje de la experiencia diaria. Actualmente se esta lejos de lograr un sistema completo que pueda comprender cualquier mensaje oral en cualquier contexto, tal y como lo podría hacer un ser humano. Sin embargo, la tecnología actual permite realizar sistemas de RAH que pueden trabajar, con un error aceptable, en entornos semánticos restringidos.

Las investigaciones y desarrollos en estas tecnologías tienen distintas metas; una de las mayores es simplificar la comunicación entre el usuario y la máquina. Así como los usuarios consideran al “mouse” una gran mejora en la interfaz con las computadoras personales, el reconocimiento y compresión de la voz, tiene un gran potencial para simplificar la forma de trabajar los seres humanos y las maquinas en general.

En general las aplicaciones tecnológicas basadas en el procesamiento automático del lenguaje hablado, se denominan comúnmente: “Tecnología del Habla” y se estructura en cuatro tecnologías básicas principales:

- **RECONOCIMIENTO DE VOZ O RECONOCIMIENTO DEL HABLA**

  Es la identificación de las palabras, y de las estructuras lingüísticas complejas que se forman y que componen el lenguaje hablado. El reconocimiento de voz puede ser dependiente o independiente del locutor (según sea capaz de reconocer a un único locutor o a un conjunto de locutores que hable el mismo idioma), y de palabras aisladas o conectadas (dependiendo de si es necesario realizar pausas entre cada palabra o no). Otros factores que se deben tener en cuenta son el tamaño del vocabulario, es decir, el número de palabras diferentes que es posible identificar, y las posibles combinaciones en que éstas pueden encontrarse.

- **Conversión Texto-Voz**

  Permite pasar de un texto en formato electrónico a lenguaje hablado. La conversión texto-voz está especialmente indicada cuando se pretende suministrar información que varía frecuentemente o cuando el volumen de información es elevado. En ambos casos presenta ventajas frente a la utilización de voz codificada, ya que la actualización de la información suministrada es inmediata debido a que, normalmente, está almacenada como texto en formato electrónico y que requiere del orden de 200 veces menos memoria de almacenamiento que la voz codificada.

- **Codificación de Voz**

  Convierte la señal analógica de voz en formato digital y viceversa, aplicando un factor de compresión que trata de reducir en mayor o menor medida el número de bits necesarios. Una vez digitalizada la señal de voz puede ser procesada, transmitida por canales digitales apropiados, almacenada en soportes informáticos y/o convertida nuevamente en señal analógica por un ordenador.

- **Reconocimiento de Locutores**

  Es el proceso de identificación o verificación de la identidad del hablante de forma automática a partir de la señal de voz. El grado de desarrollo de esta tecnología es inferior al de las anteriores, quizás como consecuencia de lo crítico que puede ser las aplicaciones en las que se inserte.

4.2 RECONOCIMIENTO AUTOMÁTICO DEL HABLA

Las principales características que diferencian a los sistemas basados en reconocimiento del Habla frente a otras alternativas son: la naturalidad que supone utilizar el habla en las operaciones de comando y control, y la robustez y precisión en la comunicación para diferentes usuarios y diferentes entornos. Los resultados que esta tecnología proporcione, deben contrastar con los derivados de otras alternativas como: teclados, mouse, paneles y otros.

Dado el amplio campo de aplicación de la tecnología del habla, se ha propuesto una clasificación, en tres grupos diferentes:

- **Aplicaciones locales**

  En esta área se pretende realizar interfaces hombre-máquina, que sustituyan la utilización de teclado y mouse, para dotar al usuario de la movilidad, que estos le restan, también facilitar el uso de máquinas a los usuarios discapacitados.

- **Respuesta vocal interactiva**

  En estas aplicaciones se involucra la difusión o captura de información, por parte de un gran número de usuarios, en particular se utiliza la red telefónica como vehículo de acceso a la información. Como por ejemplo para sustituir interfaces de detección de tonos DTMF, como lo son las consultas de cuentas bancarias, mensajería vocal, transmisión de información general, movimientos de cuentas y otros.

- **Automatización de sistemas telefónicos**

  En estas hallamos la marcación por voz, manejo de agendas, directorio público, entre otras.

4.3 PRINCIPALES ÁREAS DE TRABAJO EN RECONOCIMIENTO DEL HABLA

Un diagrama simplificado para un modelo de reconocimiento automático del habla, mostrado en la figura 4.1, sugiere algunos puntos importantes de considerar a la hora de llevar a cabo una implementación. Entre las principales áreas de trabajo que intervienen en el diseño y especificación de sistemas de Reconocimiento del Habla actuales, se encuentran las siguientes:

- Procesamiento de la señal de voz.
- Técnicas de reconocimiento de patrones.
- Diferentes estilos de habla.
- Dependencia del locutor.
- Tarea de reconocimiento.
- Bases de datos para entrenamiento y reconocimiento.

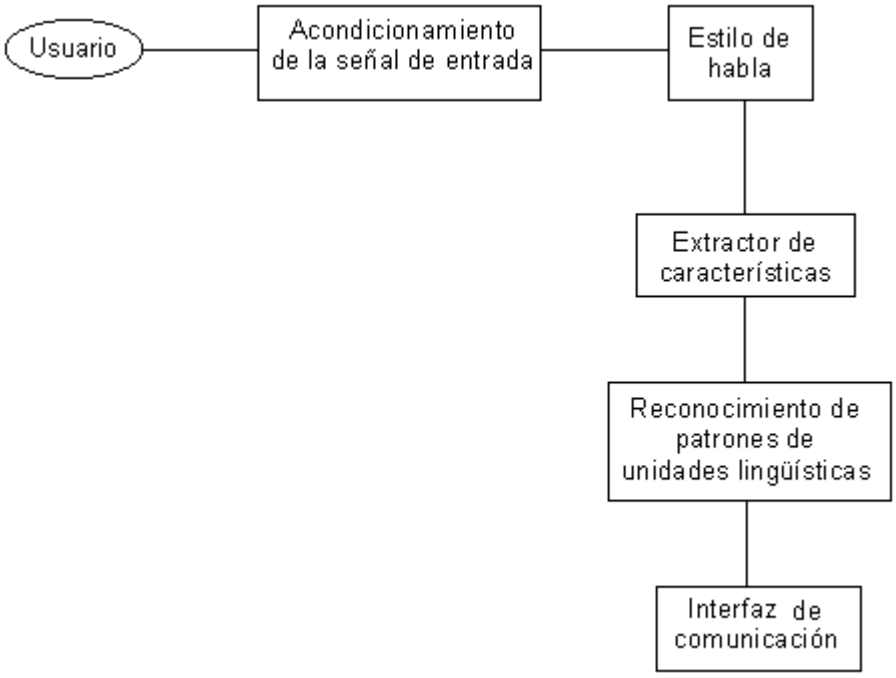

Figura 4.1 Modelo genérico de comunicación para reconocimiento del habla.

PROCESAMIENTO DE LA SEÑAL DE VOZ

La primera operación que debe realizar un reconocedor es procesar la señal de voz de entrada al sistema, con objeto de extraer la información acústica relevante para la tarea que debemos realizar.

Los rasgos o características que se deben extraer de la señal de voz, son el resultado de un largo proceso de investigación sobre diferentes procedimientos de parametrización de la voz. Planteándose como solución actual más extendida una parametrización de la envolvente espectral que incluye consideraciones preceptúales a partir del funcionamiento del oído.

Para reducir el número de parámetros posibles, la parametrización se combina con la utilización de técnicas discriminativas, seleccionándose el subconjunto con los parámetros más eficientes o distintivos.

La señal de entrada puede venir acompañada por efectos perturbadores, los cuales se desea sean eliminados, para ello se ha generado tres líneas principales de trabajo:

- **Detección robusta de voz:** apareciendo innumerables procedimientos de discriminación entre voz o ruido (silencio) para diferentes tipos de ruido.
- **Reducción de ruido:** distinguiéndose procedimientos que actúan directamente sobre la señal de voz y procedimientos que buscan compensar el efecto del ruido sobre la parametrización de la voz.
- **Cancelación de ecos:** incorporando técnicas de filtrado adaptativo que permitan al usuario comenzar a hablar mientras, desde el terminal remoto, se le está comunicando un mensaje que puede provocar un eco en la voz que entra al reconocedor.

TÉCNICAS DE RECONOCIMIENTO DE PATRONES

El reconocimiento de patrones es la técnica más específica de todo sistema de reconocimiento. De ahí que muchos reconocedores se identifiquen a partir de la técnica de reconocimiento de patrones que incorporan. A partir de la representación paramétrica de la voz, este módulo realiza un proceso de clasificación utilizando una serie de patrones. Estos patrones se obtienen en una fase de entrenamiento del sistema y son representativos de un conjunto de unidades lingüísticas (palabras, sílabas, sonidos, fonemas). La peculiaridad más característica de este proceso, que marca su dificultad, es la variabilidad temporal que puede presentar una misma unidad lingüística al ser pronunciada de diferentes modos y/o velocidades de habla.

Las primeras técnicas de reconocimiento de patrones utilizadas fueron las basadas en un Alineamiento Temporal a través de algoritmos de Programación Dinámica, técnicas DTW (o por su nombre en inglés Dynamic Time Warping). Posteriormente, se recurrió a la mayor flexibilidad que el modelado de procesos estocásticos permite, para la representación de secuencias de duración variable. Concretamente la alternativa a las técnicas DTW fueron los Modelos Ocultos de Markov (HMM), que pueden verse como una generalización de algoritmos DTW , que han demostrado mejores prestaciones en multitud de sistemas de reconocimiento. La potencia y excelentes capacidades de clasificación mostradas por las denominadas Redes Neuronales Artificiales (RNA), las sitúa como una poderosa herramienta, muy utilizada en la actualidad. Hasta el momento las Redes Neuronales han permitido obtener los mejores resultados en Reconocimiento de Locutores, sin embargo en Reconocimiento del Habla encuentran como mayor dificultad la forma de afrontar la variabilidad temporal del habla.

MODELADO DEPENDIENTE DEL ESTILO DE HABLA

Existen tres modos fundamentales a la hora de hablar frente a un sistema de reconocimiento:

- **Palabras aisladas**

  Supone que el usuario pronuncia una sola palabra o comando que el sistema deberá reconocer.

- **Habla conectada**

  El usuario pronuncia de forma fluida un mensaje utilizando un vocabulario muy restringido; el ejemplo más típico sería la pronunciación de un número telefónico.

- **Habla continua**

  Corresponde al modo más avanzado de funcionamiento de un reconocedor, y supone la pronunciación de frases de forma natural para un vocabulario amplio de palabras.

Además de los tres modos fundamentales anteriores, los reconocedores de voz tienen que afrontar, para un modelado robusto del habla, los tres aspectos siguientes:

- **Reconocimiento en contexto o "word spotting"**

  Técnica especialmente utilizada en reconocimiento de palabras aisladas, encaminada a detectar la presencia de palabras del vocabulario a reconocer en el contexto de otras palabras o pronunciaciones. La mayoría de las veces el contexto es resultado de la dificultad que encuentra el usuario para ceñirse a la pronunciación de una única palabra aislada. En otras ocasiones, el reconocimiento en contexto es la solución apropiada para robustecer el reconocimiento en ambientes acústicamente hostiles; por ejemplo, cuando la palabra que pronuncia el usuario viene acompañada de ruidos telefónicos, urbanos, etc. En cualquier caso, se trata de una técnica importante para robustecer los sistemas en aplicaciones reales.

- **Rechazo**

  Otro efecto de la presencia de sonidos indeseados (ruidos, sonidos o palabras fuera del vocabulario), es provocar el reconocimiento de palabras que realmente no han sido pronunciadas. Los procedimientos conocidos como técnicas de rechazo tienen como objetivo permitir incluir entre los resultados de reconocimiento la identificación de esos sonidos indeseados. Nos encontramos ante un problema de gran importancia de cara a la operatividad de un sistema de reconocimiento, que aún hoy por hoy no cuenta con una clara solución.

- **Múltiples candidatos**

  El proceso de reconocimiento de patrones que realiza un reconocedor se basa en identificar el patrón que ofrezca la puntuación más alta para decidir cuál es la mejor palabra o secuencia de palabras reconocida. Este proceso se basa en información exclusivamente acústica, sin tener en consideración otras posibles fuentes de conocimiento que podrían utilizarse para completar las puntuaciones de las diferentes palabras o secuencias candidatas. En la mayoría de los casos, la aplicación en que se encuentra el reconocedor es la que posee la información necesaria que permitiría seleccionar entre varias hipótesis de reconocimiento. Pensemos, por ejemplo, en una aplicación basada en el reconocimiento de números telefónicos; en esa situación, ante las dos hipótesis mejores de reconocimiento, una compuesta de cinco dígitos y otra de siete, la aplicación seleccionaría esta última independientemente de quién obtuviese la mayor puntuación "acústica" en el proceso de clasificación. Los procedimientos que permiten a un reconocedor disponer de la flexibilidad que supone manejar N hipótesis de reconocimiento se denominan N-best.

DEPENDENCIA DEL LOCUTOR

El grado de dependencia del locutor, define si el sistema incorpora patrones de unidades lingüísticas adaptados a un locutor determinado, y, por tanto, sólo funcionará correctamente para él, o si los patrones pretenden ser válidos para cualquier hablante. En el primer caso se habla de reconocimiento dependiente del locutor, mientras que en el segundo el reconocimiento es independiente del locutor. A parte de las actividades específicas que se desarrollan para sistemas dependientes e independientes del locutor, existe un importante número de esfuerzos dirigidos a conseguir la adaptación de un reconocedor a un locutor específico con la menor cantidad de voz posible.

DEPENDENCIA DEL VOCABULARIO

Las prestaciones de un reconocedor dependen fuertemente del tamaño y grado de dificultad del vocabulario. Es decir, del número de palabras que el sistema es capaz de reconocer, y de la mayor o menor dificultad de su reconocimiento, basado en las relaciones de similitud fonética entre palabras. En la actualidad se diseñan sistemas tanto para vocabularios pequeños (menos de 50 palabras) y medios (entre 50 y 500 palabras), como para grandes vocabularios (más de 500 palabras), llegándose hasta 50.000 palabras para aplicaciones de dictado o acceso a bases de datos mediante lenguaje natural.

Otra importante dimensión, en relación con el vocabulario, es la que afecta a la distinción entre vocabularios fijos y flexibles. Una determinada aplicación, cuando esté reconociendo, siempre actuará sobre un vocabulario fijo. Pero en muchos casos ese vocabulario deberá variarse o actualizarse para eliminar y/o dar cabida a nuevas palabras. Tradicionalmente, una variación del vocabulario suponía comenzar un largo y costoso proceso de recogida de una nueva base de datos y re-entrenamiento de los patrones del sistema. En la actualidad hay diversas aproximaciones para conseguir un sistema con vocabulario flexible, que no necesite re-entrenarse para cada nuevo vocabulario.

GRAMÁTICAS DE RECONOCIMIENTO

Según aumenta el número de palabras del vocabulario, el número de posibles combinaciones crece exponencialmente. Por tanto, se hace imprescindible la incorporación de restricciones, en cuanto al número de combinaciones válidas, según la tarea en que se inserte el sistema. Restricciones que suelen incorporarse en forma de gramáticas basadas en reglas sintácticas y/o semánticas destinadas a reducir el número de palabras susceptibles de ser reconocidas en cada momento. La medida utilizada para definir el grado de dificultad que supone una determinada tarea es la denominada perplejidad, de modo que un nivel de perplejidad bajo supone que en cada momento el número de posibles palabras candidatas es bajo, mientras que una perplejidad alta supone que ese número es alto, y consiguientemente el reconocimiento será más difícil.

4.4 ALGORITMOS DE EXTRACCIÓN DE CARACTERÍSTICAS

La técnica de reconocimiento empleada para extraer características de la señal, debe cumplir con ciertas condiciones para aumentar la eficiencia del sistema. La técnica empleada debe entregar resultados simples y fáciles, que se calculen de forma rápida, con lo que se disminuye el uso de recursos computacionales y se aumenta la velocidad, lo cual es de suma importancia para una implementación en tiempo real. Si el patrón de características además de contar con un número reducido de elementos, son los adecuadas, habremos simplificado muchísimo tanto el sistema, como el proceso de aprendizaje.

Los temas expuestos en esta sección suponen entendimiento de teoría de filtros digitales por parte del lector. Si no se posee tal conocimiento se recomiendo el estudio de filtros FIR e IIR, para una mayor comprensión del contenido.

La técnica de parametrización debe tener la *mayor relación posible* con el problema que tratamos. Conviene tener en cuenta las características del oído si es posible, el cual posee mayor sensibilidad a distintas frecuencias, por lo que algunas técnicas trabajan sobre ámbitos de frecuencias no lineales y con bancos de filtros, simulando la capacidad auditiva. Estas técnicas se enfocan desde el punto de vista del receptor, sin embargo hay otras que lo hacen enfocadas en la fuente generadora de señal. Interesa que los elementos extraídos tengan valores parecidos, para entradas parecidas (voces que suenen de manera similar: palabras que queremos reconocer) y sobre todo que tengan valores distintos para salidas deseadas distintas.

La representación matemática de una señal de voz, se centra en la descomposición de la señal como una fuente pasando a través de un filtro variable en el tiempo, partiendo de un modelo que aproxima el funcionamiento del órgano humano productor de la voz, como se muestra en la figura 4.2.

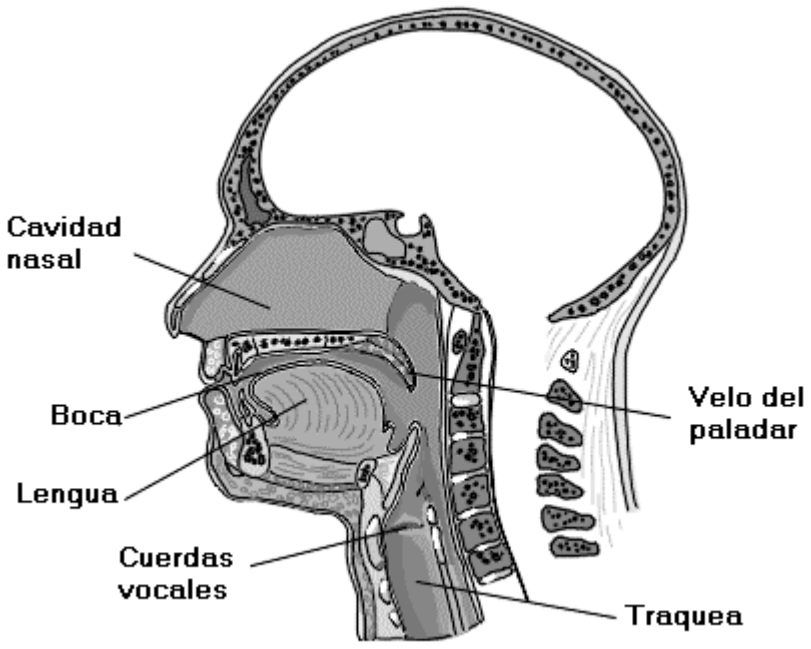

Figura 4.2 Corte transversal del órgano humano donde ser producen los sonidos.

Este filtro puede ser derivado de modelos de producción de voz basados en la teoría acústica donde la fuente representa el flujo de aire que atraviesa las cuerdas vocales y el filtro representa la resonancia del tracto vocal, con cambios en el tiempo. Este modelo fuente-filtro, se ilustra en la figura 4.3, existen métodos para calcular la fuente o excitación e[n] y el filtro h[n], a partir de la señal de voz x[n].

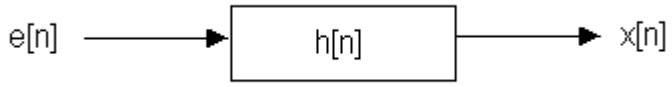

Figura 4.3 Modelo básico fuente-filtro para señales de voz.

Para estimar el filtro existen métodos inspirados en modelos de producción de voz (tales como Linear Predictive Coding y Análisis Cepstral) y modelos de percepción de voz (como Mel- frequency Cepstrum). Una vez que el filtro ha sido estimado, la fuente puede ser obtenida pasando la señal de voz a través del filtro inverso. La separación entre filtro y fuente, es uno de los retos más difíciles en el procesamiento de voz.

La información extraída mediante la parametrización, se representa mediante coeficientes estáticos o dinámicos. Los primeros se obtienen a partir del análisis de pequeños fragmentos de la señal de voz (5-20 ms), formando una matriz de coeficientes y los segundos son el producto de la combinación de componentes de diversos vectores.

Entre los métodos de extracción paramétrica tenemos: Análisis por Bancos de Filtros Digitales, Análisis de Transformada de Fourier de Tiempo Corto, Predicción Lineal, Análisis Cepstral, Predicción lineal perceptual.

ANÁLISIS POR BANCOS DE FILTROS

Un banco de filtros pasa banda puede entenderse como un modelo sencillo de etapas iniciales del sistema auditivo humano. La señal se descompone en un conjunto discreto de muestras espectrales, que contienen una información similar a la que se encuentra en los niveles superiores del sistema auditivo.

Con el objeto de acercarse a la sensibilidad del oído humano, que no tienen una respuesta lineal en frecuencia, existen diferentes escalas. Un ejemplo es la escala Mel, que es la más usual en aplicaciones de tratamiento de voz, generalmente se emplea con otros métodos, con es el caso del cálculo de coeficientes cepstrales.

Escala Mel:

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

Un banco de filtros esta constituido por un conjunto de filtros, cada uno de los cuales retiene la información de una serie determinada de frecuencia del espectro. En la figura 4.4 se muestra un ejemplo de un banco de filtros empleando escalas Mel y 19 filtros.

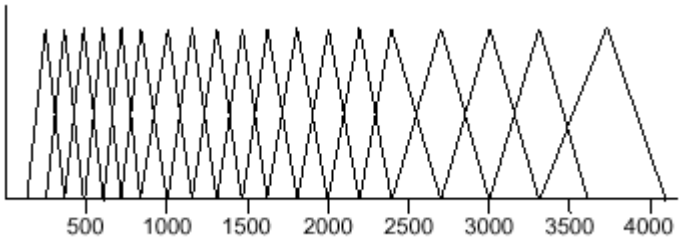

Figura 4.4 Banco de filtros de escalas Mel.

ANÁLISIS DE TRANSFORMADA DE FOURIER DE TIEMPO CORTO

Una señal de voz no presenta periodicidad cuando se analizan segmentos grandes de tiempo; además, no puede utilizarse la definición exacta de transformada de Fourier, dado que esta requiere conocimiento de la señal en un tiempo infinito.

Por ambas razones se desarrollo una nueva técnica denominada “Short-Time Analysis”, esta técnica descompone la señal en una serie de pequeños segmentos de tiempo, referidos como “Frame analysis” y se procesa cada segmento por separado. Este análisis requiere que durante la duración del Frame la señal sea aproximadamente constante.

Similar a los bancos de filtros, dada una señal de voz x[n], se define la señal de “tiempo corto” del frame m como:

$$x_m[n] = x[n] * \omega_m[n] \tag{4.1}$$

el producto de x[n], por una función ventana $\omega_m[n]$, que es cero para todos los puntos fuera de la región de interés. La función ventana puede poseer diferentes valores para diferentes frames, sin embargo, una opción es mantenerla constante para todos los frames:

$$\omega_m[n] = \omega[m-n] \tag{4.2}$$

en la práctica la longitud de la ventana esta en el orden de los 20 a 30 milisegundos.

Con base en lo anterior, se define “Short-Time Fourier Transform”, para un frame m, manteniendo las propiedades de la Transformada de Fourier, como:

$$X_m\left(e^{j\omega}\right)=\sum_{n=-\infty}^{\infty}x_m[n]e^{-j\omega n}=\sum_{n=-\infty}^{\infty}\omega[m-n]x[n]e^{-j\omega n} \tag{4.3}$$

En la figura 4.5 se muestra varios espectros de tiempo corto, para una señal de voz. Se nota como existe una serie de picos en el espectro. Para interpretar esto, se asume que las propiedades de $X_m[n]$ persisten fuera de la ventana y que, por consiguiente, la señal es periódica con periodo M, en el sentido verdadero. En este caso, conociendo las propiedades de la Transformada de Fourier, este espectro es una sumatoria de impulsos de la forma:

$$X_m\left(e^{j\omega}\right)=\sum_{k=-\infty}^{\infty}X_m[k]\delta\left(\omega-\frac{2\pi k}{M}\right) \tag{4.4}$$

dada la transformada de $\omega[n]$ como:

$$W\left(e^{j\varpi}\right)=\sum_{n=-\infty}^{\infty}\omega[n]e^{-j\omega n} \tag{4.5}$$

de manera que la transformada de $\omega[m-n]$ es $W\left(e^{-j\omega}\right)e^{-j\omega n}$. Por consiguiente, usando la propiedad de convolución, la transformada de $x[n]\omega[m-n]$, para un frame m fijo, es la convolución en el dominio de la frecuencia, obteniendo:

$$X_m\left(e^{j\omega}\right)=\sum_{k=-\infty}^{\infty}X_m[k]W\left(e^{j\left(\omega-\frac{2\pi k}{N}\right)}\right)e^{j\left(\omega-\frac{2\pi k}{N}\right)m} \tag{4.6}$$

Dando como resultado una sumatoria de pesos $W\left(e^{j\omega}\right)$, corridos sobre cada armónica, parecidos a los picos angostos observados en la figura 4.5b, con una ventana rectangular. El espectro corto de una señal periódica exhibe picos (igualmente espaciados a 2π/M) que representan los armónicos de la señal. Estimando $X_m[k]$ del espectro de tiempo corto $X_m\left(e^{j\omega}\right)$ vemos la importancia de la longitud de la ventana.

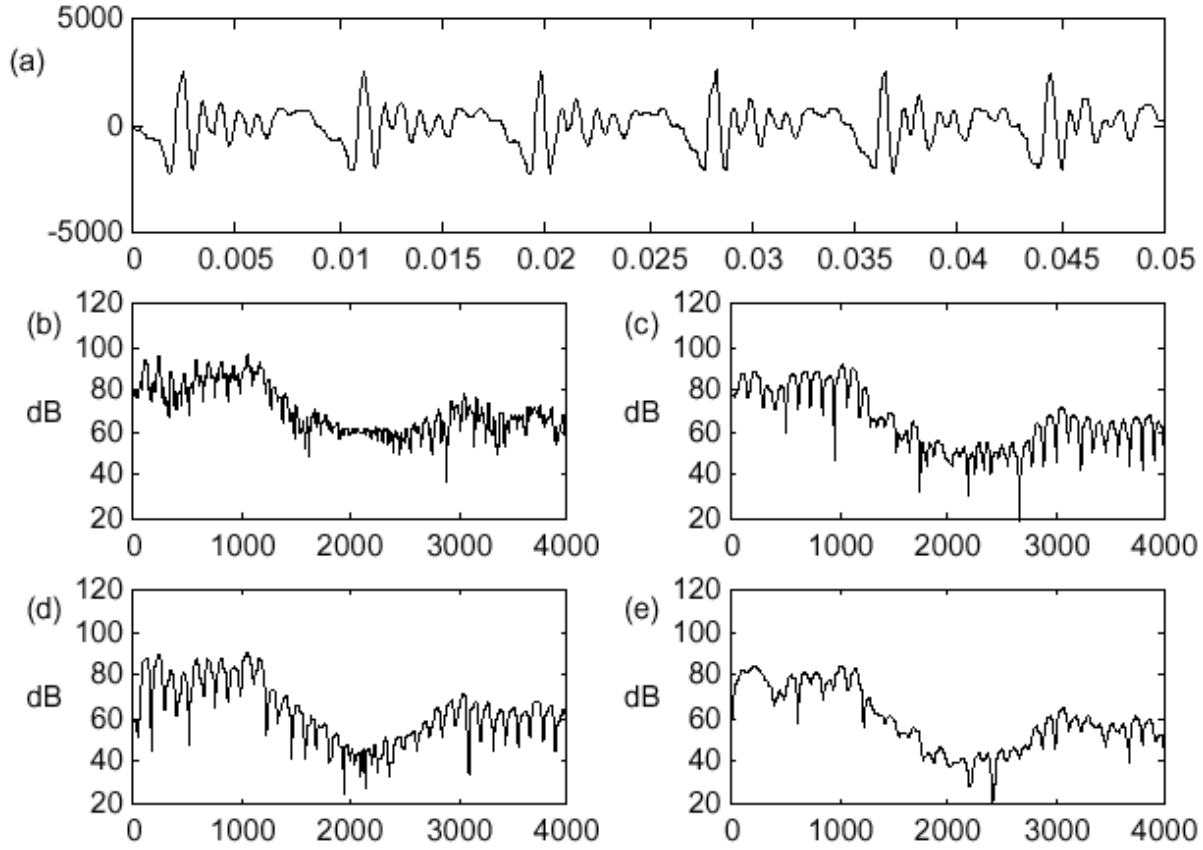

Figura 4.5 Espectro de tiempo corto de una señal de voz masculina (vocal /a/ con un tono de frecuencia de 110 Hz), a) representación en el tiempo, espectros obtenidos con ventanas de tiempo: b) ventana rectangular de 30 ms y c) 15 ms, d) ventana de Hamming de 30 ms y e) 15 ms.

Los lóbulos de la ventana no son visibles en e), porque la duración de la ventana es menor a 2 veces el periodo del tono. En la figura 4.5b se nota un goteo del espectro.

En la ecuación 4.6 se indica que no se puede recobrar $X_m[k]$ simplemente recobrando $X_m\left(e^{j\omega}\right)$ aunque la aproximación puede ser razonable si hay un pequeño valor λ tal que:

$$W\left(e^{j\omega}\right) \approx 0 \quad \text{para} \quad |\omega - \omega_k| \rangle \lambda \tag{4.7}$$

que es el caso fuera del lóbulo principal, de la respuesta en frecuencia de la ventana.

Para una ventana rectangular de duración N, $\lambda=2\pi/N$ (tomado de la teoría sobre filtros), la ecuación 4.7 se satisface si N>M, por ejemplo: la ventana rectangular contiene al menos un periodo del tono. El ancho del lóbulo principal de la respuesta de frecuencia de la ventana es inversamente proporcional a la duración de la ventana. En la figura 4.5 el periodo del tono es M=71, a una frecuencia de muestreo de 8 kHz. En la figura 4.5c se usa una ventana menor, lo que resulta en un mayor análisis de lóbulos, aunque siguen visibles.

Para una ventana Hamming de longitud N, la cual posee dos veces el ancho de una ventana rectangular ($\lambda=4\pi/N$), lo que ocasiona que $N \geq 2M$.

En la práctica no se puede saber que periodo de tono esta adelantado y que ofrece los recursos necesarios para prepararse para el menor periodo de tono. Un tono bajo de voz con F_0=50 Hz requiere una ventana rectangular de al menos 20 ms y una ventana de Hamming de 40 ms para que la ecuación 4.7 se satisfaga. Si la voz no es estacionaria dentro de los 40 ms, tomar semejante tamaño de ventana, implica obtener el espectro promedio durante ese segmento en lugar de distintos espectros. Por esta razón, la ventana rectangular provee mejor resolución de tiempo, que una ventana de Hamming.

Sin embargo, la respuesta en frecuencia no es completamente cero fuera del lóbulo principal, así que se necesita ver el efecto de esta suposición incorrecta. De la teoría de filtros, se nota como el segundo lóbulo esta aproximadamente 17 dB por debajo del lóbulo princiapal, por consiguiente, para el k-ésimo armónico el valor de $X_m\left(e^{j2\pi k/M}\right)$ no contiene $X_m[k]$, pero si tiene una suma ponderada de $X_m[l]$. Este fenómeno se llama “Goteo espectral”, porque la amplitud de una armónica gotea sobre el resto y enmascara su valor. Si la señal pose un espectro blanco, el goteo espectral no causa mayor problema, subsecuentemente el efecto del segundo lóbulo sobre el armónico es solo 0.08 dB. Por otro lado, si el espectro de la señal decae más rápidamente en frecuencia que la pendiente de la ventana, el goteo espectral resulta en estimaciones incorrectas.

El segundo lóbulo de una ventana de Hamming esta aproximadamente a 43 dB, lo que significa que el efecto del goteo es menos pronunciado. Otras ventanas como Hanning o ventanas triangulares, también ofrecen menor goteo espectral que la ventana rectangular. Por esta razón, a pesar de ofrecer una mejor resolución en el tiempo, las ventanas rectangulares son raramente utilizadas en análisis de voz. En la práctica, la duración de las ventanas esta en el orden de los 20 a 30 milisegundos, esta escogencia es un compromiso entre la suposición de que es estacionaria y su resolución en frecuencia.

En la práctica, la transformada de Fourier, ecuación 4.3, se obtiene a través de una Transformada Rápida de Fourier (TRF). Si la ventana tiene una longitud N, la TRF tiene que tener una duración mayor o igual a N.

La ST-FT es utilizada para calcular “Espectrogramas”, otra importante técnica para el análisis de voz, esta estima solo la energía y no la fase de la ST-FT, y se calcula la energía de la siguiente manera:

$$\log\left|X[k]\right|^2 = \log(X_r{}^2[k] + X_i{}^2[k]) \tag{4.8}$$

PREDICCIÓN LINEAL

Es una técnica de codificación del habla que modela el tracto vocal humano. Para sonidos sordos y sonoros, el tracto vocal puede ser modelado como una serie de cilindros con diferente radio y una cantidad diferente de energía a los limites de los cilindros. Matemáticamente, este modelo se puede representar como un filtro lineal excitado por un tono (sonido sonoros) o ruidos aleatorios (sonidos sordos).

La producción de la voz humana puede ser ilustrada por un modelo sencillo, figura 4.6a. Aquí los pulmones son reemplazados por una fuente CD, las cuerdas vocales por un generador de impulsos, las articulaciones del tracto por un filtro lineal y un generador de ruido produce las articulaciones sordas. Por supuesto, la relación entre estos varía dependiendo del sonido a ser pronunciado, en el modelo la relación se ajusta con potenciómetros.

Basados en este modelo, se puede poner un interruptor que seleccione entre articulaciones sonoras y sordas, las articulaciones del tracto se pueden simular con un filtro (un filtro digital recursivo) su comportamiento resonante (tono de respuesta) es definida por un juego de coeficientes del filtro. La estimación de los coeficientes del filtro se optimiza matemáticamente del procedimiento LPC, llamado coeficientes LPC y el modelo completo es llamado un vocalizador LPC. En la práctica es muy usado en telefonía, la pequeña taza de bits necesarios para la transmisión (7.8 kbits/s o 2.4 kbits/s) es muy baja comparada con PCM (64 kbits/s).

Una ventaja del vocalizador LPC es la facilidad de manipulación y la estrecha analogía con la producción de la voz humana, así los principales parámetros de la producción de la voz, expresados en los coeficientes LPC, son directamente accesibles. Las características de la voz pueden ser influenciables ampliamente, como el tono y la velocidad.

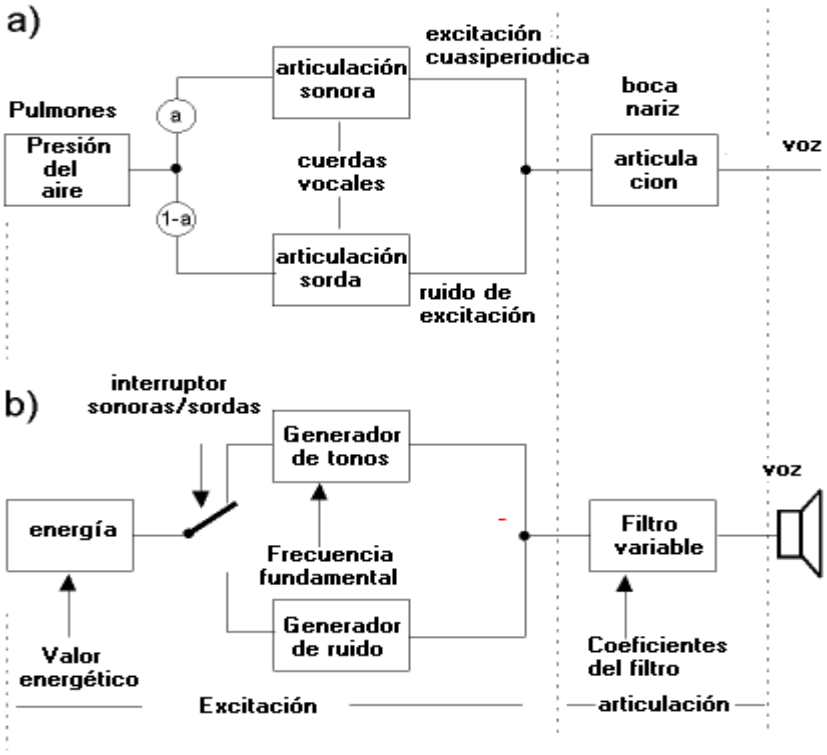

Figura 4.6 Ilustración de un modelo de simulación de la producción de la voz humana.

La meta del análisis de predicción lineal es derivar los parámetros necesarios para reconstruir los sonidos: decidir entre sonidos sonoros o sordos, frecuencia fundamental, ganancia del sistema y los coeficientes que describen al filtro. El objetivo de LPC es predecir la próxima salida del sistema, basado en las entradas y salidas previas. Esta es una efectiva técnica de codificación, ya que las señales de voz son altamente correlacionadas cuando se consideran en un intervalo de corta duración (tamaño de la ventana). Es decir, dada una sucesión de muestras de voz, subsecuentes muestras de voz pueden predecirse con un mínimo de error sobre un corto periodo de tiempo.

El filtro se puede modelar de la siguiente manera:

$$H(z) = \frac{X(z)}{E(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

donde p denota el orden del análisis LPC.

Si tomamos la Transfomada Z Inversa obtenemos:

$$x[n] = \sum_{k=1}^{p} a_k x[n-k] + e[n]$$

la estimación del error de predicción cuando se usa esta aproximación, puede ser calculado por diferentes métodos y viene dado de la siguiente manera:

$$e[n] = x[n] - \hat{x}[n] = x[n] - \sum_{k=1}^{p} a_k x[n-k]$$

Existen tres métodos principales para solucionar las ecuaciones LPC: el método de covarianza, el método de autocorrelación y la Formulación Lattice. Estos métodos requieren que la señal sea segmentada en periodos cortos de tiempo (usando una ventana de Hamming por ejemplo). Para conocer más sobre este método referirse a la bibliografía.

CEPSTRUM

El procesamiento Cepstral, son técnicas sobre sistemas Homomórficos, una clase de sistemas no lineales que obedecen a un principio de superposición, de estos los sistemas lineales son un caso especial, es una transformada Homomórfica $\hat{x}[n] = D(x[n])$, esta transformación convierte una convolución:

$$x[n] = e[n] * h[n]$$

en una suma:

$$\hat{x}[n] = \hat{e}[n] + \hat{h}[n]$$

Cepstrum es una Transformación Homomórfica que permite separar la fuente del filtro. Se puede demostrar que existe un valor para N, tal que el Cepstrum del filtro $\hat{h}[n] \approx 0$ para n>=N y que el Cepstrum de la excitación $\hat{e}[n] \approx 0$ para n<N. Con esta suposición, se puede aproximadamente recobrar ambos e[n] y h[n] de $\hat{x}[n]$ por filtrado homomórfico. En la figura 4.7 se muestra como recobrar h[n] con un filtro homomórfico.

$$l[n] = \begin{cases} 1 & |n| < N \\ 0 & |n| \geq N \end{cases}$$

donde D es el operador Cepstrum

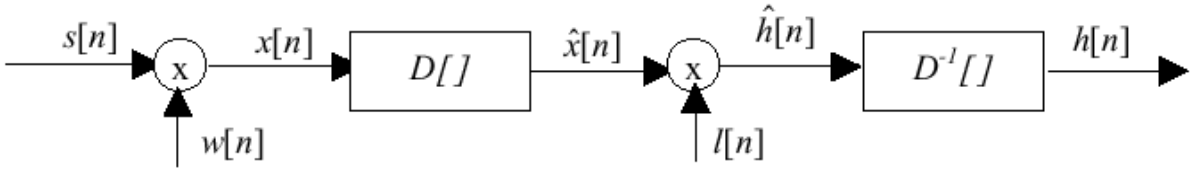

Figura 4.7 Filtrado Homomórfico para recobrar la respuesta del filtro de una señal periódica.

La señal de excitación puede ser recobrada similarmente con un filtro homomórfico dado por:

$$l[n] = \begin{cases} 1 & |n| \geq N \\ 0 & |n| < N \end{cases}$$

La motivación para realizar un procesado homomórfico del habla viene resumido en la figura 4.8.

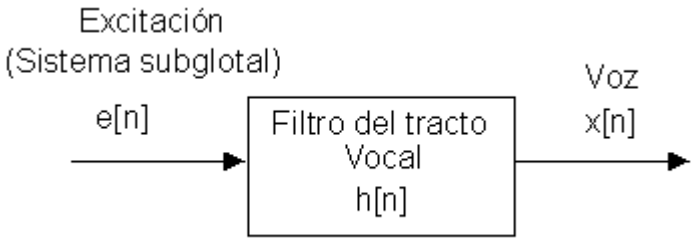

Figura 4.8 Ilustración que motiva el uso de técnicas homomórficas para el procesamiento de voz.

En el dominio de la frecuencia tenemos:

$$X\left(e^{j\omega}\right) = H\left(e^{j\omega}\right)E\left(e^{j\omega}\right)$$

Para la mayoría de las aplicaciones de voz sólo necesitamos la amplitud espectral:

$$\log\left[X\left(e^{j\omega}\right)\right] = \log\left[H\left(e^{j\omega}\right)\right] + \log\left[E\left(e^{j\omega}\right)\right]$$

En el dominio logarítmico, los dos componentes anteriores pueden separarse empleando técnicas convencionales de procesamiento de señales, a partir de la Transformada Rápida de Fourier y su inversa. Tal como se muestra en la figura 4.9.

Figura 4.9 Análisis Cepstral partiendo de la Transformada Rápida de Fourier.

$$c(n) = \frac{1}{N_s}\sum_{k=0}^{N_s-1}\log_{10}\left|X_{med}(k)\right|e^{j\frac{2\pi}{N_s}kn} \qquad 0 \le n \le N_s - 1$$

El valor c(n) se conoce como coeficientes cepstrales derivados de la Transformada de Fourier, N_s es el número de puntos con los que se calcula la FFT. Esta ecuación también se conoce como la inversa de la FFT del espectro logarítmico.

Existen el Real Cepstrum y el Complex Cepstrum, se diferencian en que el Complex Cepstrum utiliza la forma compleja del logaritmo. Si la señal x[n] es real, ambas el real y el complex cepstrum son también reales.

MEL FREQUENCY CEPSTRUM

Los coeficientes Cepstrum en escala de frecuencia Mel (MFCC) es una representación definida como el Real Cepstrum de una señal de entrada de tiempo corto, pasada a través de un banco de filtros digitales y derivada a partir de la FFT. La diferencia con Real Cesptrum radica en que utiliza una escala de frecuencia no lineal, que aproxima el principio de la audición humana.

Para derivar el algoritmo se toma la FFT de la señal de entrada:

$$X_a[k] = \sum_{n=0}^{N-1} x[n] e^{-j 2\pi mk / N} \quad , \; 0 \leq k < N$$

se define un banco de filtros con M filtros (m=1,2,...,M), donde m es un filtro triangular dado por:

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])} & f[m-1] \leq k \leq f[m+1] \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m] - f[m-1])} & f[m] \leq k \leq f[m+1] \\ 0 & k > f[m+1] \end{cases}$$

Tales filtros calculan el espectro promedio alrededor de cada frecuencia central con incrementos en el ancho de banda, como se muestra en la figura 4.10.

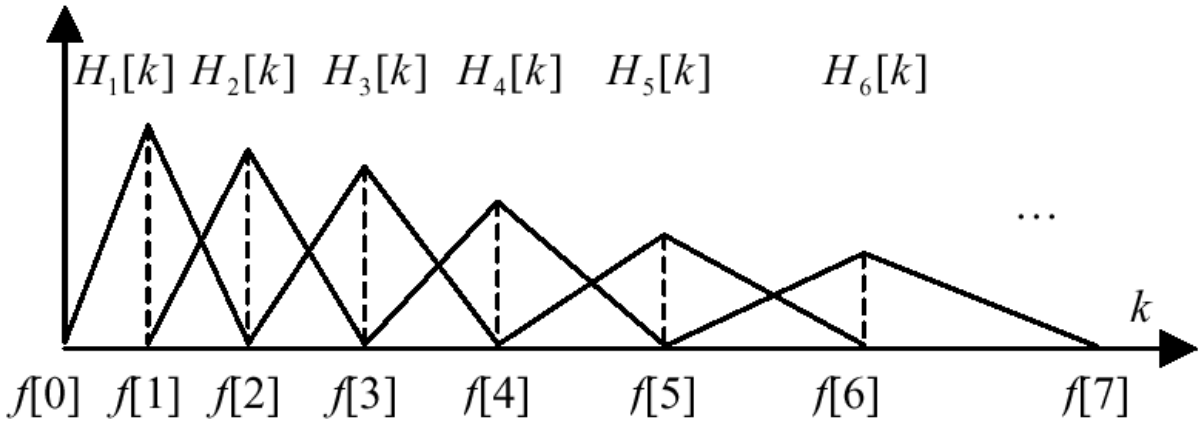

Figura 4.10 Filtros usados en el calculo de Mel Cepstrum.

Se define f_i y f_s como las frecuencias inferior y superior del banco en Hz, F_s la frecuencia de muestreo en Hz, M el número de filtros y N el tamaño de la FFT. Los puntos límite $f[m]$ son uniformemente espaciados en la escala Mel:

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_i) + \frac{B(f_s) - B(f_i)}{M+1}\right), \quad 0 \le n < M$$

donde la escala mel inversa B^{-1} es dada por:

$$B^{-1}(b) = 700\left(e^{b/1125} - 1\right)$$

y B por su inversa.

Entonces se calcula el logaritmo de la energía de la salida de cada filtro como:

$$S[m] = \log\left[\sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k]\right],\ 0 \le m < M$$

Mel Frequency Cepstrum es entonces la Transformada Coseno Discreta de la salida del filtro M:

$$c[n] = \sum_{m=0}^{N-1} S[m]\cos\left(\pi n\left(m + \frac{1}{5}\right)/M\right),\ 0 \le n < M$$

M varía para diferentes implementaciones de 24 a 40. Para reconocimiento típicamente solo los primeros 13 coeficientes son usados.

Una implementación de esta técnica se muestra en la figura 4.11.

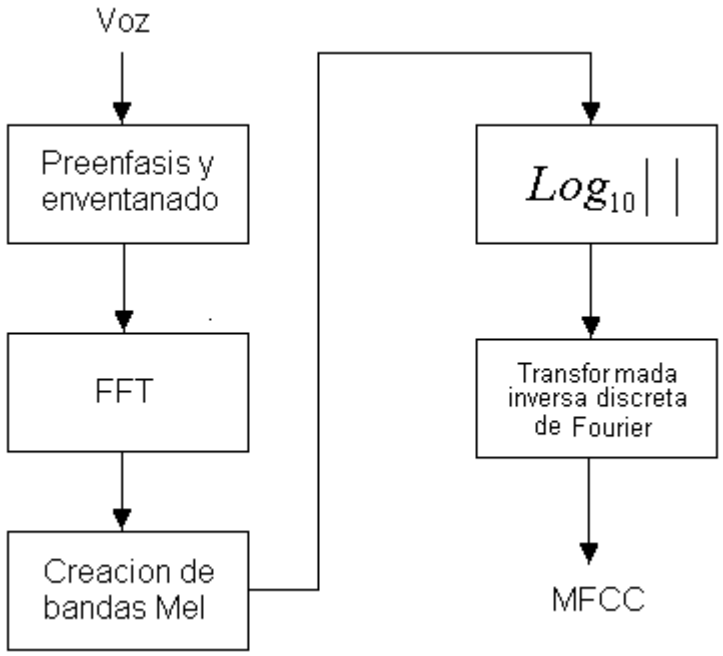

Figura 4.11 Esquema de parametrización para la obtención de MFCC.

Partiendo de la predicción lineal también es posible obtener los coeficientes cepstrum y la expresión de los coeficientes cepstrales asociados:

$$c(0) = \log(1) = 0$$

$$c(i) = -a(i) - \sum_{j=1}^{i-1}\left(1 - \frac{j}{i}\right)a(j)c(i-j), \quad 1 \le i \le Nc$$

Una transformación usual sobre este tipo de coeficientes es lo que se conoce como coeficientes cepstrales delta o coeficientes delta cepstrum, que se obtienen a partir de:

$$\Delta c_j(i) = \frac{1}{2T+1}\sum_{k=-T}^{T} kc_{j+k}(i)$$

REDES NEURONALES

Abstrayendo su funcionamiento interno pueden considerarse como una caja negra que devuelve unas salidas para unas entradas dadas de forma que las salidas se aproximan suficientemente (si la red ha sido bien entrenada anteriormente) a las salidas que queremos obtener. Además son tan flexibles que pueden "aprender sobre la marcha" y "adaptarse según las necesidades del momento", es decir, partir de una situación donde lo hacen todo mal y mejorar poco a poco hasta funcionar suficientemente bien y una vez que esto está conseguido si cambian las condiciones también se pueden adaptar a estos cambios. Es claro que algo así puede tener aplicaciones muy diversas.

Para utilizar una red neuronal para reconocimiento de voz debemos elegir las entradas, las salidas y la estructura necesaria para que produzca las salidas deseadas para las entradas dadas. Después se elige un algoritmo de entrenamiento entre los posibles y unos parámetros para después realizar el/los entrenamientos.

Una red neuronal podría ser entrena para comprender la señal de voz en el tiempo, sin embargo, al estar esta sometida a perturbaciones, es mejor usar algún algoritmo de los estudiados anteriormente y usar un híbrido. Es preferible que las entradas de la red cumplan con las mismas condiciones que se pidieron para las técnicas de modelado, en especial que tenga un número reducido de elementos.

Lo recomendado en el uso de redes neuronales para reconocimiento de voz, es diseñar una red con varias capas ocultas (tres o cuatro) y pocas neuronas por capa (unos 10), luego es preferible tener pocas salidas y varias redes. Por ejemplo, si se desea reconocer los números del 1 al 10, es mejor construir cinco redes que pueden diferenciar dos números cada una, que una sola que reconozca los diez dígitos, esto para no crear una red muy grande y tener un sistema lento.

Una red neuronal (RNA, en castellano, y ANN, en inglés) puede definirse como un modelo artificial basado en la conexión de varios procesadores elementales (neuronas artificiales) para que conjuntamente realicen una función común.

Aunque puede pretender imitar las características y el funcionamiento de las neuronas naturales presentes en el sistema nervioso de muchos seres vivos (cuyo máximo exponente es el cerebro humano) en realidad el funcionamiento de las redes de neuronas naturales es mucho más complejo y su funcionamiento interno es muy poco conocido.

El modelo y funcionamiento de una red neuronal es sencillo y se debe a las contribuciones individuales de cada una de las neuronas que conforman la red. La neurona artificial es el elemento básico de procesamiento de la red neuronal artificial. Es un sistema con varias entradas y una salida, la cual depende del estado interno de la neurona.

La conexión (o dependencia) entre la salida y las entradas se puede modificar y ello se consigue mediante unos factores (pesos) que se aplican a cada una de las entradas. El resultado de tal aplicación es un solo valor a partir de varios, (varias entradas sopesadas, es decir, multiplicadas cada una por un peso) lo cual se consigue mediante una suma y una función no lineal.

$$y = f_{ac}(x_i * w_i - Umb)$$

La neurona sólo se activa (valor alto a la salida) si la suma de las entradas ponderada mediante unos factores llamados pesos o, lo que es lo mismo, el producto escalar del vector de pesos por el vector de entradas supera un umbral, es decir, si el producto escalar de los vectores pesos y entradas es suficientemente alto. No ocurrirá esto cuando entradas y pesos sean ortogonales (producto escalar nulo), opuestos (producto escalar < 0) o de módulo pequeño.

Su funcionamiento, por tanto, consiste en tomar las entradas, multiplicarlas cada una por un factor llamado peso (cada entrada tiene el suyo), sumar todos estos productos, restar un umbral y aplicar al resultado una función no lineal.

Las variable que intervienen en el proceso son:

- Entradas. (x_i)
- Pesos. (w_i)
- Umbral. (Umb)
- Función de activación. (f_{ac})
- Salida. (y)

Una de las ventajas de la función de activación es que permite concentrar los valores de salida en un conjunto acotado (típicamente de 0 a 1 o de -1 a 1).

La elección de la función de activación junto con la forma de ponderación (variantes del producto escalar) determinan las características de la neurona artificial.

Perceptrón de una capa.

Consiste en formar una función de varias salidas a partir de las unidades básicas vistas anteriormente. Para ello se colocan en paralelo tantas neuronas artificiales como salidas deseemos (ya que cada neurona artificial va a darnos una salida). Todas ellas tendrán las mismas entradas (entradas de la capa) y cada neurona dará en general una salida diferente porque tendrá diferentes pesos y umbrales.

Perceptrón multicapa (MLP).

Consiste básicamente en poner varias capas elementales, como las anteriormente descritas, interconectadas sucesivamente con el objeto de dotar a la red de la complejidad suficiente para realizar la tarea requerida, tal como se muestra en la figura 4.12.

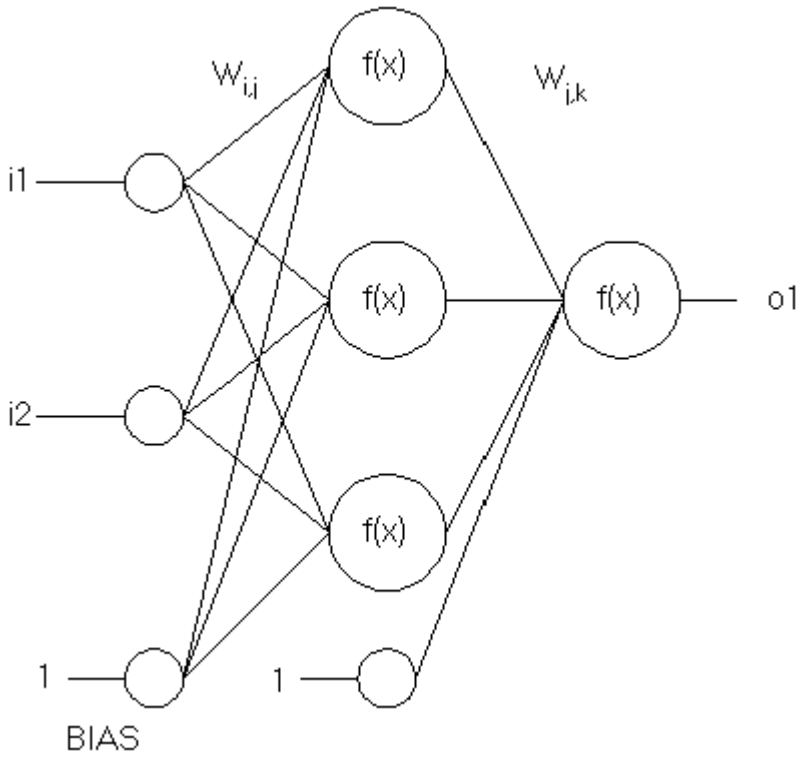

Figura 4.12 Red neuronal de tres capas (3x3x1).

Perceptrón multicapa con capas de retardo.

Es el modelo más complejo. Consiste en la inclusión de bloques (o capas) de neuronas que toman como entradas las salidas de otro/s bloque/s en el instante anterior. Esto permite que la red sea un sistema con memoria a corto plazo (o en fase operativa). Esta memoria es muy diferente a la memoria debida al entrenamiento que presenta el sistema global (que podría llamarse memoria a largo plazo o memoria en fase de entrenamiento).

Teniendo en cuenta que los patrones que pasamos a la red varían con el tiempo se puede comprender la importancia de estos retardos. Si la red no presenta retardos y la entrenamos mediante unos parámetros variables con el tiempo, impedimos que la red tenga en cuenta el orden en que llegan los parámetros (no podría distinguir "uno" de "onu" por ejemplo; la red pudiera ser entrenada para distinguir unos ejemplos pero sería imposible que generalizara). Los retardos permiten distinguir características incrementales de primer orden (diferencias o derivada numérica respecto al tiempo) o de segundo orden (derivada segunda numérica).

En lo referente al entrenamiento de la red, para un sistema de reconocimiento se puede tomar la activación de una única neurona de salida por palabra reconocida, además contar con un buen número de muestras. Si el resultado no cumple con lo esperado se debe aumentar el número de neuronas en la red. El proceso de creación de la estructura de la red es un proceso de prueba y error, se parte de una base y se crean modificaciones hasta alcanzar los resultados deseados. Entre los modelos de redes recomendados se hallan las redes de retropropagación.

Matlab ofrece distintos tipos de redes neuronales, para las cuales incorpora sus algoritmos de entrenamiento, entre los tipos de redes se tienen las siguientes:

- ADALINE (Adaptive Linear Neuron Networks)
- Retropropagación (Backpropagation networks)
- Self-Organizations networks
- Recurrent networks

ADALINE: consiste en una red de una sola capa (perceptón), con funciones de activación lineales. En la figura 4.13 se muestra el modelo general para este tipo de red. En las entradas de la red se pueden introducir retardos con lo que se pueden realizar filtros lineales. Este tipo de redes solo pueden solucionar problemas lineales.

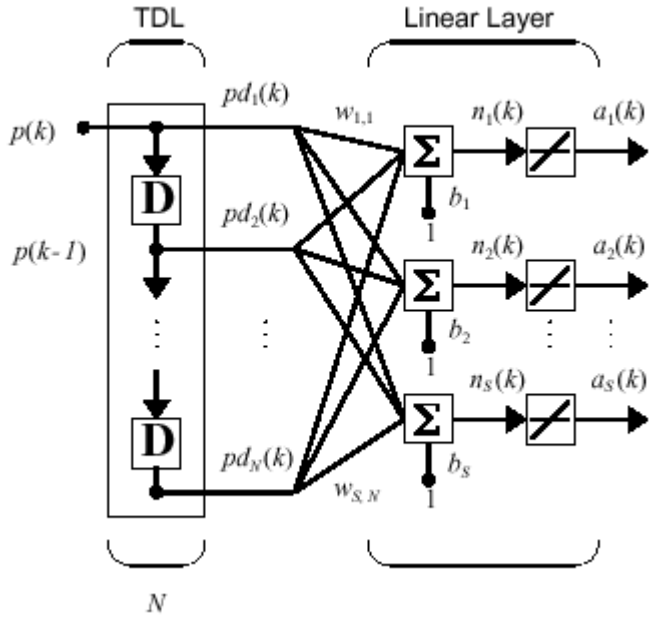

Figura 4.13 Arquitectura general de una red ADALINE.

RED DE RETROPROPAGACIÓN: son redes de múltiples capas con funciones de activación no lineales diferenciables. Son capaces de aproximar cualquier función con un número finito de discontinuidades. En la figura 4.14 se muestra un modelo de arquitectura para una red de dos capaz.

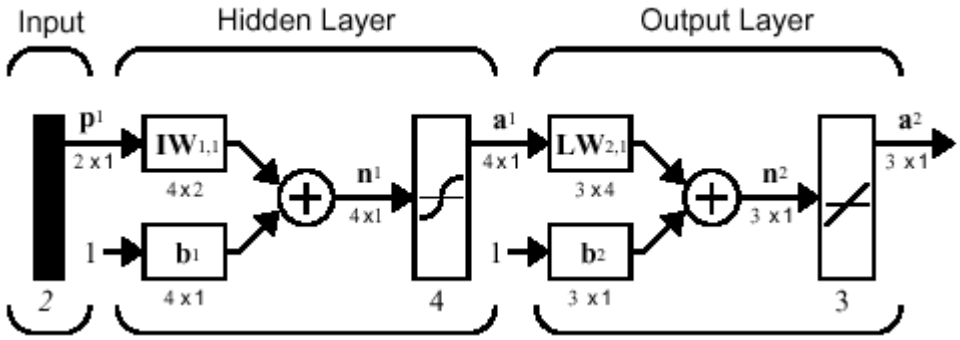

Figura 4.14 Modelo de la arquitectura de una red de Retropropagación de 2 capas.

RADIAL BASIS NETWORKS: estas redes requieren más neuronas que las redes de retropropagación, pero pueden ser diseñadas en una fracción del tiempo que toma entrenar una red de retropropagación. Trabajan mejor cuando hay muchas muestras disponibles para entrenamiento. En la figura 4.15 se muestra la arquitectura usada por estas redes, mediante el entrenamiento se determina automáticamente la cantidad de neuronas que conformarán la red.

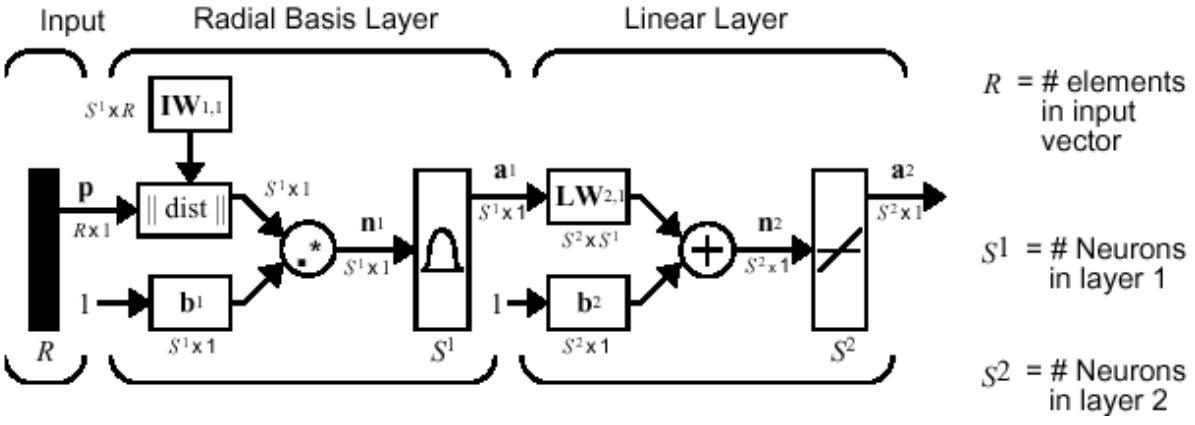

Figura 4.15 Arquitectura de una red Radial Basis

En este tipo de red también se encuentran las redes de Regresión Generalizada y Probabilística. La primera a menudo es usada para aproximación de funciones. Utiliza una capa radial basis y una capa lineal especial. En la figura 4.16 se muestra la arquitectura de esta red, similar a la red Radial Basis, solo se diferencian en la segunda capa. Al igual que la anterior se autoconstruyen. La segunda usada para clasificación de problemas, ofrece un máximo de 3 posibilidades, se autoconstruye, en la figura 4.17 se muestra su arquitectura.

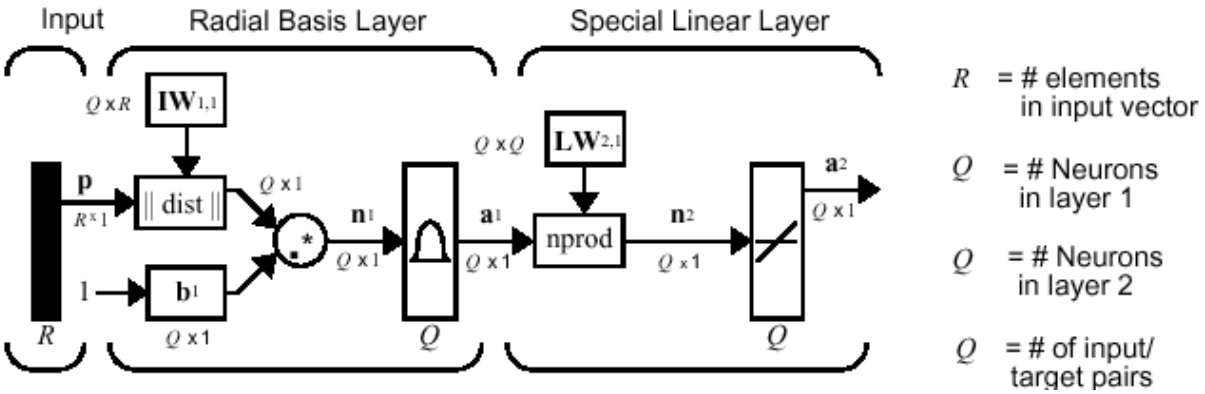

Figura 4.16 Arquitectura para una red de regresión generalizada.

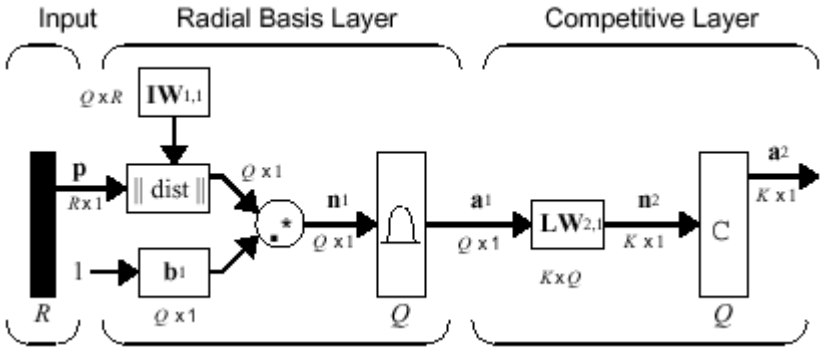

Figura 4.17 Arquitectura para una red probabilística.

SELF-ORGANIZATIONS NETWORKS: estas redes pueden aprender a detectar regularidades y correlaciones en sus entradas y adaptar sus respuestas futuras de acuerdo a estas entradas. Las neuronas de redes competitive aprenden a reconocer grupos de vectores de entrada similares. Self-Organizations maps aprenden a reconocer grupos de vectores de entradas similares de manera que físicamente las neuronas se juntan en la capa de respuestas a vectores de entrada similares. Si la red sólo necesita aprender a categorizar, estas redes pueden ser empleadas. Su arquitectura se muestra en la figura 4.18.

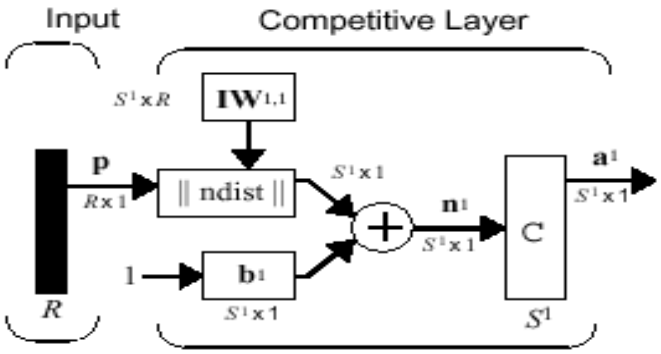

Figura 4.18 Arquitectura de una red Self-organizations.

LEARNING VECTOR QUANTIZATION (LVQ): es un método para entrenar competitive layers de manera supervisada. Una capa competitive automáticamente aprende a clasificar los vectores de entrada. Sin embargo, la forma en que los encuentra sólo depende de la distancia entre los vectores de entrada. Si dos vectores de entrada son muy similares, seguramente se pondrán en la misma clase. No tienen un mecanismo estricto para decidir si dos vectores pertenecen o no a la misma clase. LVQ aprende a clasificar los vectores los vectores de entrada en clases escogidas por el usuario, en la figura 4.19 su arquitectura. A diferencia de los perceptrones, LVQ pueden clasificar cualquier conjunto de vectores de entrada, no sólo los conjuntos de entradas linealmente separables. El único requerimiento es que la capa competitive debe tener bastantes neuronas y a cada clase se le deben asignar bastantes neuronas competitive.

Para asegurar que a cada clase se le asignó una cantidad aproximada de neuronas competitive, es importante que el vector meta utilizado para iniciar la red, tenga la misma distribución de objetivos con los datos de entrada utilizados para entrenar la red. Si esto se hace, las clases meta con más vectores serán la unión de subclases.

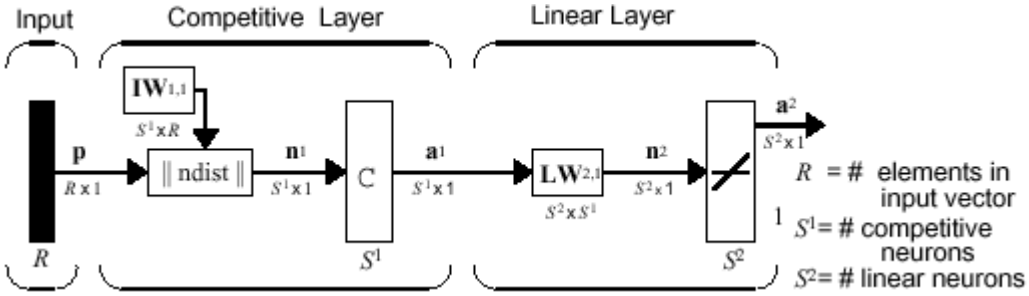

Figura 4.19 Arquitectura de una red LVQ.

REDES RECURRENTES: Matlab Cubre dos tipos, Elman y Holpfield. Las redes Elman son 2 capas de redes de retropropagación, con una retroalimentación de las salidas de la capa oculta a la entrada. Esta retroalimentación permite aprender a reconocer y generar patrones temporales. Son capaces de aprender a detectar y generar patrones de tiempo, lo que las hace útiles en áreas como procesamiento y predicción de señales donde el tiempo juega un rol dominante. Necesitan muchas neuronas en la capa recurrente para ajustarse a funciones complejas, la arquitectura de esta red se muestra en la figura 4.20.

Las redes Holpfield no son muy usadas en la práctica.

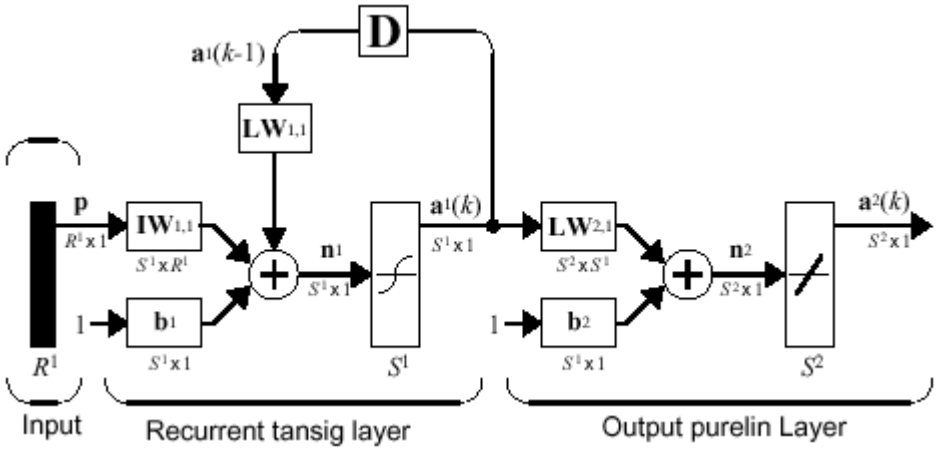

Figura 4.20 Arquitectura de una red de retropropagación Elman.

4.5 ANÁLISIS DE RESULTADOS

Una de las principales consideraciones a definir al momento de iniciar la confección de un sistema de reconocimiento de voz, es saber en que forma se le va a hablar al sistema, existen tres modos fundamentales: palabras aisladas, habla conectada y habla continua. En el primero las palabras se pronuncian de forma aislada, con pausas entre las palabras, el segundo modo reconoce una palabra dentro de un contexto, por último se da el reconocimiento de frases en forma natural, que representa la forma más avanzada. Para una implementación el reconocimiento de palabras aisladas, es el modo más sencillo.

Un sistema independiente del locutor presenta un mayor reto de generalización que uno dependiente, lo que se busca es un reconocimiento independiente del locutor con un número reducido de muestras de entrenamiento, estas muestras dependen del tipo de unidad gramatical que este usando el reconocedor: palabras, fonemas, sílabas.

Las técnicas de reconocimiento Cesptrum y Mel Cepstrum incorporan las ventajas que posee la transformada rápida de Fourier, por lo que representan una técnica confiable de parametrización, además la técnica Mel Cepstrum, por usar una escala no lineal, que asemeja el funcionamiento del oído, resulta ser la mejor para emplear en reconocimiento, también ofrece un número reducido de elementos, el cual depende del número de coeficientes que se deseen por frame y del tamaño de este.

En las redes neuronales queda excluidas para el procesamiento de voz las siguientes: ADALINE, Hopfield, Self-Organization, la probabilística, la LVQ y la red de recurrencia Elmam. La primera por resolver sólo operaciones lineales, la segunda es una red poco usada, un modelo en desarrollo. Las redes Self-Organization, probabilística y la LVQ, son redes para clasificación o categorización,. Las redes Elman tienen un retardo, lo cual impone un tiempo de muestreo, utilizando un sistema híbrido este retardo no es necesario, son especialmente usadas para señales para señales que varían en el tiempo.

Quedan dos redes por analizar: la Red de Retropropagación y la red Radial Basis, esta requiere más neuronas que la primera, como se muestra en la tabla 4.1, para los resultados obtenidos en el experimento 1 (el procedimiento del experimeto 1 se muestra en los apéndices), lo que aumentaría el costo computacional y el tiempo de ejecución, por lo que la mejor opción es la Red de Retropropagación. La Red de retropropagación tiene múltiples capaz, realiza cualquier función con un número finito de irregularidades, además, es una de las más usadas.

Tabla 4.1 Resultados experimentales obtenidos al entrenar redes para reconocer voz, dependiente del locutor.

Tipo de red	Número de neuronas por capa	Porcentaje de éxitos (%)
Retropropagación	5x5x2	100
Radial Basis	27x2	100
Regresión Generalizada	28x2	0

Un sistema entre una red neuronal y una técnica de parametrización, es una solución que puede llegar a dar excelentes resultados.

4.6 CONCLUSIONES

Un reconocedor de palabras aisladas representa la forma más sencilla de un reconocedor y un reconocedor de habla continua la más avanzada.

Los reconocedores se pueden basar en diferentes unidades lingüísticas, como palabras, fonemas y sílabas.

Mel Cepstrum es la técnica de extracción de parámetros más eficaz, se basa en la densidad espectral de energía y utiliza una base de frecuencia no lineal que simula la capacidad auditiva humana.

Las redes neuronales de retropropagación representan la mejor alternativa para implementar el algoritmo de reconocimiento, por ofrecer mejor rendimiento en el cálculo computacional.

Un sistema híbrido entre una red neuronal y un sistema de parametrización representa una alternativa, con alta probabilidad de éxito para un sistema de reconocimiento, que no requiere base de datos.

CAPITULO 5
PROCESADORES DIGITALES DE SEÑALES

5.1 INTRODUCCIÓN

Un DSP es un procesador especializado para desarrollar cálculos masivos y aplicaciones en tiempo real, por lo que son ampliamente usados en el tratamiento digital de señales. Estas aplicaciones demandan un gran número de cálculos aritméticos como sumas sucesivas, multiplicación y multiplicación-acumulación, que son operaciones típicas de los algoritmos empleados en los DSPs. Entre estos algoritmos se encuentran: transformada rápida de Fourier, correlación, algoritmos para transmisión de voz, control de motores y otros.

PROCESADORES DIGITALES DE SEÑALES VS MICROPROCESADORES UNIVERSALES

Los Procesadores Digitales de Señales tienen aproximadamente el mismo nivel de integración y la misma frecuencia de reloj, que los microprocesadores de propósito general, incluso, generalmente los microprocesadores llevan ventaja en este campo; pero, en tareas de procesamiento de señales, los DSPs son de 2 a 3 veces más veloces, esta ventaja se logra mediante una arquitectura optimizada para tareas especializadas. Se puede decir que algunas características de los DSPs algunas veces son incluidas en los más recientes microprocesadores universales, pero no viceversa.

Algunas de las diferencias arquitectónicas que se presentan entre un DSP y un microcontrolador universal son:

- Arquitectura de la unidad aritmética
- Unidades especializadas (multiplicadores, multiplicadores-acumuladores y otros)
- Ciclos de instrucción regular (como la arquitectura RISC)
- Procesamiento paralelo
- Arquitectura de Bus Harvard
- Organización de la memoria interna
- Organización multiprocesamiento
- Interconexión de procesadores
- Interconexión de bancos de memoria

RASGOS QUE CARACTERIZAN UN PROCESADOR DSP

El procesamiento digital de señales demanda un fuerte desempeño por parte del procesador; pero, una alta eficiencia no puede ser medida únicamente por la velocidad de procesamiento de cálculos de multiplicación-acumulación o MIPS (millones de instrucciones por segundo) que efectúe un DSP, aunque generalmente un procesador DSP es caracterizado principalmente por el número de MIPS capaz de realizar, tomando como referencia la forma en que el DSP ejecuta una instrucción, que no es necesariamente igual a la de otro, el medir el desempeño de un DSP, sólo por el número de MIPS que realice, es engañoso. Otras características arquitectónicas de procesamiento como lo son: cálculos de operaciones aritméticas, direccionamiento y secuencia del programa son más importantes.

Para distinguir los puntos altos en el desempeño y las características arquitectónicas que diferencian un procesador DSP, de un microprocesador y un microcontrolador. Se desarrollará sobre algunas de los principales rasgos que caracterizan un buen DSP.

a. ARITMÉTICA RÁPIDA Y FLEXIBLE

La operación más común en el procesamiento digital de señales es la suma de productos. La operación más compleja en la técnica digital es la multiplicación, por ejemplo en un microprocesador 8086 una suma toma 3 pulsos de reloj y una multiplicación de 134-160, como usualmente una operación de multiplicación tiene una probabilidad de realizarse del 1%, comparada con la frecuencia de otras operaciones, un multiplicador es innecesario en un microprocesador, contrario a un DSP en donde es indispensable. Para incrementar la velocidad, los DSPs usualmente tienen unidades aritméticas especializadas, que pueden operar simultáneamente. Todos los DSPs tienen un multiplicador-acumulador y dos operaciones de multiplicación y suma, pueden implementarse en el mismo ciclo. Otro aspecto es el muestreo de señales, que debe hacerse a una taza fija, lo que hace necesario un ciclo de instrucción regular. Un microprocesador lo logra con un ciclo de instrucción reducido (RISC, reduced instruction set computer), un DSP lo logra mediante hardware. Un DSP debe realizar un cálculo de multiplicación, multiplicación con acumulación (una cantidad arbitraria de corrimientos) y operaciones aritméticas y lógicas, en un solo ciclo de procesador. En la suma, las unidades aritméticas deben permitir cualquier secuencia de cálculo, de manera que un algoritmo dado de DSP pueda ser ejecutado sin ser reformulado.

b. RANGO EXTENDIDO PARA MULTIPLICACIÓN CON ACUMULACIÓN

Extensas sumas de productos son fundamentales para los algoritmos de DSPs, por lo que protección contra desbordamiento en acumulaciones sucesivas, asegura que no se presenten perdidas de datos o de rango.

Estas dos secciones tratan sobre características aritméticas. Un buen indicador de una buena arquitectura aritmética es la habilidad para desempeñar una gran variedad de cálculos matemáticos. Estos cálculos pueden ser manejados de una manera flexible, de manera que el algoritmo puede ser implementado sin restaurar el orden de las operaciones matemáticas o los operandos. Si la arquitectura es fija, para aplicaciones muy específicas o limitada y el algoritmo debe de ser restaurado, esto supone trabajo extra para el diseñador o programador y retraza el trabajo.

c. BÚSQUEDA DE DOS OPERANDOS EN UN SOLO CICLO DE PROCESAMIENTO (DENTRO O FUERA DEL CHIP)

Dos operandos son siempre necesarios para alimentar las operaciones de cálculo, por lo que estos deben estar listos, por ejemplo para agilizar una multiplicación con acumulación. También, un direccionamiento flexible para buscar datos de diferentes localidades de memoria es importante.

d. ARQUITECTURA DE BUFFER CIRCULAR (INTERNA Y EXTERNA)

Muchos algoritmos para DSPs requieren de buffers circulares, entre ellos los filtros. Hardware para manejar punteros de direccionamiento y módulos de direccionamiento, reducen los tiempos de espera (incrementan el desempeño) y simplifican la implementación. Los microprocesadores universales fueron diseñados para computadoras personales simples y baratas, tienen una arquitectura de bus Von-Newman: espacio de datos y comandos común y un bus de datos y direcciones. Cuando un microprocesador realiza cualquier instrucción debe leerla de memoria, decodificarla, y leer el operando, luego ejecutar el cálculo, lo cual conlleva mucho tiempo. Todos los DSPs tienen una arquitectura Harvard modificada, con buses de datos y programa diferentes. Casi todos los DSPs tienen arquitectura Harvard modificada con tres buses: uno de programa y dos de datos. Esto permite al DSP leer una palabra de instrucción y dos operandos simultáneamente.

Estos dos apartados tratan sobre la habilidad de direccionamiento. Desempeñar cálculos aritméticos rápidos, requiere que los datos sean buscados a la misma velocidad con que se da el procesamiento. El hardware de direccionamiento debe soportar la búsqueda dual de dos operandos, necesarios para la total utilización de la arquitectura encontrada en la mayoría de los DSPs. Un buen DSP debe tener la habilidad para guardar dos tipos de operandos, típicamente un coeficiente y una palabra de datos. La máxima eficiencia puede ser obtenida si dos diferentes espacios de memoria son provistos para los operandos, de manera que se puedan buscar en el mismo ciclo. Usando la memoria de datos y la memoria de programa para guardar los datos, permitirá una máxima eficiencia.

e. LAZOS SIN GASTO (NÚMEROS DE CICLOS / ACCESO A MEMORIA) Y BIFURCACIONES

Los algoritmos para DSPs son repetitivos y pueden ser expresados fácilmente como lazos. Secuencias de programación que soportan código enlazado, sin tiempos de espera, proporcionan un mejor funcionamiento y una fácil implementación. Igualmente, los tiempos de espera en programas condicionados son inaceptables en aplicaciones de procesamiento de señales.

Esto se refiere a la eficiencia en la secuencia del programa, la cual tiene muchos aspectos, en los que destacan: la ejecución de lazos, por ejemplo los ciclos “DO UNTIL” y los saltos condicionales o incondicionales.

f. MULTIPROCESAMIENTO

Los DSPs son diseñados para cálculos en tiempo real, algunas veces el número de operaciones es muy extenso y se llega a necesitar más de un procesador. Los sistemas basados en microprocesadores tienen una arquitectura de bus común, todas las unidades (memorias, controladores) están conectadas al procesador y al bus de datos y direcciones. Cuando se incrementa el número de microprocesadores, conectados al bus común, se reducen las posibilidades de intercambio de información entre microprocesadores. Es difícil desarrollar sistemas multiprocesador basados en microprocesadores de propósito general, sin hardware adicional. Los DSPs tienen conexiones especializadas para interconexión de varios DSPs, los que trabajan con su propia memoria, sin interferir a los otros.

Las aplicaciones con DSPs requieren memoria RAM de alta velocidad. Los sistemas diseñados utilizando DSPs deben usar un DSP con suficiente memoria interna o incluir suficiente memoria externa, la cual debe ser accesada lo suficientemente rápido, para no entorpecer la operación del procesador.

En la siguiente tabla se listan algunos algoritmos típicos utilizados en aplicaciones de procesadores DSPs y la cantidad aproximada de memoria requerida. Esta información es tomada de aplicaciones desarrolladas utilizando DSPs de la familia ADSP-2100, de Analog Device.

Tabla 5.1 Algoritmos típicos para DSPs y requerimientos de memoria aproximados.

Algoritmo	**Memoria RAM requerida aproximada**
V.32bis o V.32terbo Fax / Modem	16K words PM, 16k words DM
20 Voice Music Synthesis	6K words PM, 4k words DM
Full Duplex Speaker Phone	2K words PM, 2k words DM
Digital Answering Machine	5K words PM, 1k words DM
GMS, 13 kbps Compression / Decompression	2K words PM, 1k words DM
CELP, 4.3 o 7.5 kbps Compression / Decompression	1.5K words PM, 1.5k words DM
MPEG Layer2, 64 kbps Compression	5K words PM, 4k words DM
MPEG Layer2, 64 kbps Decompression	1K words PM, 8.5k words DM
Dolby AC-2, 117 kbps Compression / Decompression	4K words PM, 8.5k words DM

Nota: datos tomados de la casa fabricante Analog Device.

Hay dos factores que influyen los tiempos de acceso de la memoria RAM: las instrucciones del DSP y la lógica de decodificación. Una taza de instrucciones de 25 MIPS o mayores, con memorias tipo SRAM con tiempos de acceso de 15 ns o menos son requeridos para la operación del DSP a máxima velocidad (sin estados de espera). Sistemas que usan una lógica externa de decodificación, requieren memoria SRAM con tiempos de acceso muy rápidos para compensar los retardos incurridos por la decodificación.

Otro problema importante del uso de memoria RAM externa en un sistema con DSPs, es el tiempo envuelto en el diseño, la depuración y uso de la interface para la memoria RAM externa. Sistemas que usan interfaces con RAM externa, tienen un desempeño de cuello de botella.

Una técnica para reducir el número de ciclos/acceso requeridos para acceder la memoria RAM, es el acceso directo de memoria (DMA, por sus siglas en inglés, para Direct Memory Accessing). Usando DMA, un dispositivo externo puede accesar la memoria interna del DSP sin ninguna interferencia del DSP. Este tipo de acceso permite enlazar un dispositivo sin incurrir en un gasto del DSP en una transferencia de datos.

Los sistemas con DSPs tienen algún tipo de dispositivos externos (Convertidores analógico-digital, digital-analógico, codificadores) conectados, muchos incluyen interfaces especializadas (puerto serie, paralelo o ambos) para estos periféricos. Para conectar dispositivos de E/S en paralelo, un DSP con espacio separado de memoria, líneas de control de E/S y un conjunto de instrucciones específicas de E/S, proporciona ventajas. Tener por separado los dispositivos de E/S implica que no se tiene que usar direccionamiento de memoria del DSP para mapear los dispositivos externos. Las líneas de control separadas para estos dispositivos externos reducen la cantidad de decodificadores y lógica de control necesaria en el sistema. Un conjunto de instrucciones específicas para el control de los dispositivos de E/S facilitan la programación.

En resumen un buen DSP debe tener la habilidad para desempeñar con facilidad las siguientes tres características básicas:

- Realización rápida de cálculos matemáticos
- Búsqueda de datos a alta velocidad
- Secuencia efectiva a través de operaciones repetitivas

ASPECTOS RELEVANTES HA CONSIDERAR PARA LA ELECCIÓN DE UN DSP

- Bajo costo
- Bajo consumo de potencia
- Frecuencia del reloj
- Memoria interna
- Herramientas de desarrollo
- Soporte para lenguajes de alto nivel.

APLICACIONES TÍPICAS DE LOS DSPS:

- Procesamiento de señales de audio
- Procesamiento de habla
- Comunicaciones
- Instrumentación y Medición
- Equipo médico
- Procesamiento óptico e imágenes
- Control industrial (Motores)

PRINCIPALES FABRICANTES:

- Analog Device
- Texas Instruments
- Motorola

5.2 ANALOG DEVICE

DSPs DE 16 BITS DE PUNTO FIJO

Tabla 5.2 Lista genérica de DSPs de 16 bits más vendidos

DSP	Max MIPS	RAM words de programa	RAM words de datos	Vcc / V	Precio USA
ADSP-2192	320	32K	100K	2.5	53.20
ADSP-2188M	75	48K	56K	2.5	28.00
ADSP2188N	80	48K	56K	1.8	26.00
ADSP-2189M	75	32K	48K	2.5	23.00
ADSP-2189N	80	32K	48K	1.8	21.00
ADSP-2187N	80	32K	32K	1.8	17.00
ADSP-2185M	75	16K	16K	2.5	10.00
ADSP-2185N	80	16K	16K	1.8	9.50
ADSP-2186M	75	8K	8K	2.5	7.50
ADSP-2186N	80	8K	8K	1.8	7.25
ADSP-2184N	80	4K	4K	1.8	5.75

Tabla 5.3 Lista de herramientas de desarrollo para DSPs de 16 bits

Procesador	Plataforma de evaluación	Emulador	Software de desarrollo
ADSP-218XM	ADDS-2189M-EZLITE $295	ADDS-218X-ICE-1.8V $1995	VDSP-21XX-PC-FULL $2995
ADSP-218XN	ADDS-2189N-EZLITE $295	ADDS-218X-ICE-1.8V $1995	VDSP-21XX-PC-FULL $2995
ADSP-2192	ADDS-2192-12EZLITE $295	ADDS-APEX-ICE $4995 ADDS-TREK-ICE $5995 ADDS-SUMMIT-ICE $3995	VDSP-21XX-PC-FULL $2995

Tabla 5.4 Lista completa de DSPs de 16 bits de Analog Device

Producto	Descripción
ADSP-21535	aplicaciones de internet
ADSP-2184N	80MIPS, 1.8V, 2 Puertos serie, Host Port, 20KB RAM
ADSP-2185N	80MIPS, 1.8V, 2 Puertos serie, Host Port, 80KB RAM
ADSP-2186N	80MIPS, 1.8V, 2 Puertos serie, Host Port, 40KB RAM
ADSP-2188N	80MIPS, 1.8V, 2 Puertos serie, Host Port, 256KB RAM
ADSP-2191M	aplicaciones en Telecomunicaciones, 160 MIPS, sistema de E/S reforzado
ADSP-2192	DSP Microcomputador
ADSP-2187N	80MIPS, 1.8V, 2 Puertos serie, Host Port, 160KB RAM
ADSP-2189N	80MIPS, 1.8V, 2 Puertos serie, Host Port, 192KB RAM
ADSP-2188M	75 MIPS, 2.75v, 2 puertos serie, host port, 256 KB RAM
ADSP-2186M	75 MIPS, 2.5V, 2 Puertos serie, Host Port, 40 KB RAM
ADSP-2185M	75 MIPS, 2.5v, 2 puertos serie, host port, 80 KB RAM
ADSP-2141L	SafeNet DSP Security System on a Chip
ADSP-2184	40 MIPS, 5v, 2 puertos serie, host port, 20KB RAM
ADSP-2184L	40 MIPS, 3.3v, 2 puertos serie, host port, 20KB RAM
ADSP-2189M	75 MIPS, 2.5v, 2 puertos serie, host port, 192 KB RAM
ADSP-2187L	52 MIPS, 3.3v, 2 puertos serie, host port, 160 KB RAM
ADSP-2186L	40 MIPS, 3.3 v, 2 puertos serie, host port, 40 KB RAM
ADSP-2185	33 MIPS, 5 v, 2 puertos serie, host port, 80 KB RAM
ADSP-2186	40 MIPS, 5v, 2 puertos serie, host port, 40 KB RAM
ADSP-2104	20 MIPS, 5v, 2 puertos serie
ADSP-2104L	13 MIPS, 3.3v, 2 puertos serie
ADSP-2183	52 MIPS, 3.3 v, 2 puertos serie, host port, 80 KB RAM
ADSP-2181	40 MIPS, 5v, 2 puertos serie, host port, 80 KB RAM
ADSP-21msp58	Fully-integrated, single-chip DSP
ADSP-2165	DSP Microcomputers Con ROM
ADSP-2166	DSP Microcomputers Con ROM
ADSP-2171	33 MIPS, 5v, 2 puertos serie, host port
ADSP-2173	20 MIPS, 3.3v, 2 puertos serie, host port
ADSP-2161	DSP Microcomputers Con ROM
ADSP-2163	DSP Microcomputers Con ROM
ADSP-2164	DSP Microcomputers Con ROM
ADSP-2103	10.2 MIPS, 3.3v, 2 puertos serie
ADSP-2162	DSP Microcomputers Con ROM
ADSP-2115	25 MIPS, 5v, 2 puertos serie

DSPs de 32 Bits SHARC®

Tabla 5.5 Lista genérica de DSPs de 32 bits, más vendidos

DSP	Max MIPS	SRAM interna	Vcc / V	Precio $ US
ADSP-21160N	540	4 MBits	1.9/3.3	145.00
ADSP-21160M	480	4 MBits	2.5/3.3	145.00
ADSP21065L	198	544 KBits	3.3	30.00
ADSP-21161N	600	1 MBits	1.8/3.3	34.32

Tabla 5.6 Lista de herramientas de desarrollo para DSPs de 32 bits

Procesador	Plataforma de evaluación	Emulador	Software de desarrollo
ADSP-21065L	ADDS-21065L-EZLITE $299	ADDS-APEX-ICE $4995 ADDS-TREK-ICE $5995 ADDS-SUMMIT-ICE $3995	VDSP-SHARC-PC-FULL $2995
ADSP-21160M	ADDS-21160M-EZLITE $595	ADDS-APEX-ICE $4995 ADDS-TREK-ICE $5995 ADDS-SUMMIT-ICE $3995	VDSP-SHARC-PC-FULL $2995
ADSP-21161N	ADDS-21161N-12EZLITE	ADDS-APEX-ICE $4995 ADDS-TREK-ICE $5995 ADDS-SUMMIT-ICE $3995	VDSP-SHARC-PC-FULL $2995

Tabla 5.7 Lista completa de DSPs de 32 bits de Analog Device

Producto	Descripción
ADSP-21161N	Bajo costo, 100 MHz, 600 MFLOPS, 3.3 V I/O, 1.8 V CPU, 32/40 Bit Punto flotante, 32 Bit punto fijo
ADSP-21160M	80 MHz, 600 MFLOPS, 3.3v I/O, 2.5v DSP, punto flotante
ADSP-21065L	Bajo costo, 60 MHz, 180 MFLOPS, 3.3v, punto flotante
ADSP-21061L	44MHz, 150 MFLOPS, 3.3v, punto flotante
AD14160	480-MFLOP, Quad DSP, 5v, encapsulado CBGA
AD14060L	480-MFLOP, Quad DSP, 3.3v, encapsulado CQFP
AD14160L	480-MFLOP, Quad DSP, 3.3v, encapsulado CBGA
AD14060	480-MFLOP, Quad DSP, 5v, encapsulado CQFP
ADSP-21061	50 MHz, 150 MFLOPS, 5v, punto flotante
ADSP-21060L	120 MFLOPS, 3.3 v, punto flotante
ADSP-21062L	40 MHz, 120 MFLOPS, 3.3v, punto flotante
ADSP-21060	40 MHz, 120 MFLOPS, 5v, punto flotante
ADSP-21062	40 MHz, 120 MFLOPS, 5v, punto flotante

RECOMENDACIÓN POR MERCADO Y APLICACIÓN

Conforme las capacidades de procesamiento de los DSPs se fueron incrementando, se fueron usando más y más en diversas aplicaciones

Para el fabricante Analog Device la integración de los DSPs en una gran variedad de aplicaciones, es facilitada por la disponibilidad de librerías de rutinas (incluidas con el compilador C), que proporcionan código preparado para muchos algoritmos.

A continuación se lista un gran número de mercados que están siendo invadidos por los DSPs y la aplicación que desempeñan en este.

Procesamiento de Señales de Audio	
Función del DSP	**Aplicación**
• Reverb • Control de tono • Eco • Filtrado • Compresión de audio • Ecualización • Efectos especiales • Sonido Surround	• Instrumentos musicales y amplificadores • Consolas de mezcla de audio • Equipo de grabación • Consolas para Disc Jockey • Equipo de transmisión • Equipo de TV por cable • Equipo de audio y tableros para PCs • Juguetes • Sistemas de sonido para autos • Digital audio tape players • Compact disk players • Equipo de HDTV • TV digital
	DSPs recomendados
	• ADSP-21065L • ADSP-21160M • ADSP-21161N

Procesamiento de voz	
Función del DSP	**Aplicación**
• Síntesis de voz • Reconocimiento de voz • Compresión de voz • Texto a voz • Filtrado • Grabación de voz y playback	• Grabadoras sin cinta • Equipo de grabación de voz • Correo telefónico • Sistemas de seguridad activados por voz • Intercom systems • Sistemas de identificación de personas • Juguetes y juegos
	DSPs recomendados
	• ADSP-218XM/N • ADSP-21065L • ADSP-21161N

Comunicaciones	
Función del DSP	**Aplicación**
• Modulación y transmisión • Demodulación y recepción • Compresión de voz • T1 switching • DTMF • Data Encryption • Recuperación de señales • Cancelación de eco • Datos sobre voz	• Modems • Maquinas de Fax • Sistemas PBX • Sistemas de correo telefónico • Sistemas de comunicación de datos privados • Automatic Teller machines • Equipo de transmisión • Telefonía móvil • Digital pagers • GPS • Secure, Speaker, Video Telephones • Maquinas contestadoras digitales • Telefonía satelital • Wireless Local Loop • Telecom Infrastructure
	DSPs recomendados
	• ADSP-218XM/N • ADSP-21065L

Equipo Médico	
Función del DSP	**Aplicación**
• Filtrado • Cancelación de eco • FFT • Beam forming	• Equipo de monitoreo respiratorio • Monitoreo del pulso cardiaco • Equipo de ultrasonido • Medical Imaging equipment • Análisis sanguíneo • Monitoreo de fetos y niños • Monitoreo de pacientes • Monitoreo de flujo sanguíneo • CAT scanners
	DSPs recomendados • ADSP-218XM/N • ADSP-2116X • ADSP-2106X

Medición e Instrumentación	
Función del DSP	**Aplicación**
• (FFT) • Filtrado • Síntesis de formas de Onda • Adaptive filters • Cálculos numéricos de alta velocidad	• Equipo de pruebas y de medición • Equipo de análisis de vibración • Automative Engine Analyzers • Balanceo automático de llantas • Industrial scales and measurement • Active mufflers • Oil Drilling equipment • Instrumentos sísmicos • Medidores de energía • Máquinas de ejercicios • Analizadores de señales • Generadores de funciones y señales
	DSPs recomendados • ADSP-218XM/N • ADSP-21065L • ADSP-21161N

Procesamiento óptico y de imágenes	
Función del DSP	**Aplicación**
• Filtrado en 2 D • FFT • Control de Lazos • Reconocimiento de patrones • Image Smoothing	• Scanners de código de barras • Buscadores de objetos bajo el agua • Sistemas de inspección automática • TV digital • Reconocimiento de huellas digitales • Visión robótica • Sistemas de visión
	DSPs recomendados
	• ADSP-2106X • ADSP-2116X

Control Industrial	
Función del DSP	**Aplicación**
• Filtrado • FFT • Control de lazos • Cancelación de ruido	• Motores de electrodomésticos, robots y automatización de oficinas • Equipo de manejo de energía • Generadores • Elevadores • Aire acondicionado • Sistemas de control de tráfico • Navegación • Manejadores de disco • Control a alta velocidad • Analizadores de vibración
	DSPs recomendados
	• ADSP-218XM/N • ADSP-2106X • ADSP160M • ADMCXXX (DSPs especializados para control industrial)

BENEFICIOS IMPORTANTES DE LOS DSP ANALOG DEVICE

Característica	Beneficio
• Ejecución de instrucciones en un solo ciclo	• ADSP-218X no requiere ciclos extras de decisión en bifurcaciones, verificaciones condicionadas o llamadas a subrutinas • Bifurcaciones retardadas incrementan la eficiencia en las arquitecturas pipelined como lo son SHARC y ADSP-219X • Operaciones determinísticas hacen fácil desarrollar, perfilar y referirse a código
• Familias con código compatible	• Todos los miembros de la familia ADSP-2100 tienen la misma arquitectura básica y el mismo lenguaje ensamblador • Todos los miembros de la familia ADSP-21000 SHARC tienen la misma arquitectura básica y el mismo lenguaje ensamblador • No es necesario aprender o investigar sobre nuevas herramientas de desarrollo cuando se pasa de un miembro de la familia a otro • La inversión en software se mantiene
• Lenguaje de programación sencillo	• “Algebraic syntax assembly language” es fácil de usar, aprender y leer • A diferencia de los competidores que usan nemónicos como SPAC y XORX, el lenguaje ensamblador de ADI hace fácil programar en un lenguaje de alto nivel

• Centro balanceado, memoria y dispositivos de E/S integrados	• Rápida unidad de procesamiento, gran cantidad de memoria interna y un amplio ancho de banda de los dispositivos de E/S que simplifican el desarrollo de sistemas en tiempo real • Hasta 14 canales de DMA sin interrupciones que permiten el movimiento de datos sin interrumpir el procesamiento matemático
• Amplia memoria interna	• Proporciona amplia memoria para guardar las tareas más comunes en DSPs, tales como filtros digitales, FFT eliminando la necesidad de memoria externa
• Eficiente secuencia de programa y lazos sin gasto	• Minimiza los tiempos de acceso a la memoria externa • Hardware de administración de lazos interno y eficiente ejecución de código que no requiere programación extra para código recursivo • No se necesita de un software complejo para manejar los lazos de ejecución
• Miembros de la familia compatibles pin a pin	• Incrementa la velocidad o integración de memoria en una patilla de salida común • Aumenta flexibilidad sin requerimiento de rediseño de tarjetas

Rasgos comunes entre miembros de la misma familia

- Ejecución de instrucciones en un solo ciclo
- Bus interno de datos y programa separado
- Memoria de programa de doble propósito, tanto para instrucciones como para guardar datos
- 3 unidades computacionales independientes: ALU, multiplicador/acumulador y registro de desplazamiento (Barrel shifter)
- 2 generadores de direcciones de datos independientes
- El secuenciador de programa proporciona: lazos sin gasto y ejecución de instrucciones aritméticas condicionadas
- Generador de estados de espera programable
- Gran número de registros de direccionamiento
- Soporte de direccionamiento por módulo y bit-invertido
- Arranque automático de memoria interna de bajo costo, ejemplo EPROM o host interface port
- Cambio de contexto en un solo ciclo
- Instrucciones multifuncionales
- Interrupciones externas sensibles a nivel o flanco
- Amplio set de instrucciones, ejecución condicionada
- Hardware de interrupciones de buffer circular, anidados
- Soporte paralelo de movimiento de datos diferentes, incluidos movimientos de registro a registro
- El registro de desplazamiento permite corrimientos de 0-32 bits
- Juego de registros de ciclo simple para el cambio de contexto, mayoría de registros aritméticos ocultos con nivel simple.

5.3 TEXAS INSTRUMENTS

Los procesadores DSPs de la casa fabricante Texas Instruments se hallan divididos en tres plataformas principales, optimizados para desempeñar tareas específicas.

PLATAFORMA C6000 DSP DE ALTO DESEMPEÑO

Optimizados para aplicaciones de redes de transmisión y digitalización de imágenes

Tabla 5.8 Generaciones de la plataforma TMS320C6000

Generación	Características
C62X	16 bit, punto fijo, 1200-2400 MIPS
C64X	16 bit, punto fijo, 3200-4800 MIPS
C67X	32 bit, punto flotante, 600-1000 MFLOPS

PLATAFORMA C5000 DSPS DE BAJO CONSUMO DE POTENCIA

Optimizados para aplicaciones portátiles, inalámbricas y personales

Tabla 5.9 Generaciones de la plataforma TMS320C5000

Generación	Características
C54X	16 bit, punto fijo, 0.54 mW/MIPS, 30-532 MIPS
C55X	16 bit, punto fijo, 0.25 mW/MIPS, 320-400 MIPS

PLATAFORMA C2000 DSPS PARA CONTROL

Optimizados para aplicaciones control digital de motores

Tabla 5.10 Generaciones de la plataforma TMS320C2000

Generación	Características
C24X	16 bit, punto fijo, 20-40 MIPS
C28X	32 bit, punto fijo, hasta 400 MIPS

Cada una de estas generaciones posee características, que los remiten a una aplicación en específico. Sin perder de vista que este documento lo que busca es encontrar un procesador DSP, conveniente para una aplicación de reconocimiento de voz, se dará énfasis a la plataforma C5000, que es la que realiza tareas de este tipo. Como primer punto se listarán los DSPs que conforman las generaciones C54X y C55X.

TMS320C5000™ Plataforma De Eficiente Consumo De Potencia

Esta plataforma provee una combinación entre dispositivos periféricos de alto desempeño, tamaño y un eficiente consumo de potencia, para aplicaciones inalámbricas e internet. Con un consumo de potencia tan bajo como 0.9 V y 0.05mW /MIPS y un desempeño de hasta 600 MIPS, la plataforma C5000 está optimizada para productos personales y portátiles, como Digital Music Players, Telefonos celulares 3G y cámaras digitales, así como aplicaciones intensivas en voz y datos y aplicaciones multi-canal. Además los miembros de esta plataforma son compatibles en software.

Tabla 5.11 Características distintivas entre la plataforma C54x y la C55x

Característica	Generación C54X	Generación C55X	Mejora en la generación C55x
MW /MIPS	0.32	0.05	6x
MIPS / MMACS	30 a 532	288 a 600	5x
Densidad de código		Arquitectura de tamaño de instrucción variable	30%
Unidades funcionales MACs ALUs Acumuladores	1 1 2	2 2 4	El doble de unidades funcionales, duplica el programa de búsqueda y incrementa la flexibilidad del tamaño de las instrucciones
Buscador de programa	16 bits	32 bits	
Tamaño de instrucción	16 bits fijo	Variable 8 a 48 bits	

GENERACIÓN C54X

Tabla 4.12 Tabla genérica sobre DSPs de la generación c54x

Nombre Del Dispositivo	Frec MHz	Memoria Datos/ programa (words)	RAM/ ROM (words)	DMA/ timers	Total Puertos serie	Total Puertos Serie estandar	Puerto Serie TDM	Vcc (V)	Costo US$
TMS320C541-40	40	64k/64k	5k/28k	/1	2	2		5	17.58
TMS320C542-40	40	64k/64k	10k/2k	/1	2		1	5	21.64
TMS320LC541-66	66	64k/64k	5k/28k	/1	2	2		3.3	
TMS320LC541B-50	50	64k/64k	5k/28k					3.3	
TMS320LC541B-66	66	64k/64k	5k/28k					3.3	9.47
TMS320LC542-40	40	64k/64k	10k/2k	/1	2		1	3.3	21.64
TMS320LC542-50	50	64k/64k	10k/2k	/1	2		1	3.3	23.81
TMS320LC543-40	40	64k/64k	10k/2k	/1	2		1	3.3	20.58
TMS320LC543-50	50	64k/64k	10k/2k	/1	2		1	3.3	21.51
TMS320LC545A-50	50	64k/64k	6k/48k	/1	2	1		3.3	16.79
TMS320LC545A-66	66	64k/64k	6k/484k	/1	2	1		3.3	18.46
TMS320LC546A-40	40	64k/64k	6k/48k	/1	2	1		3.3	
TMS320LC546A-50	50	64k/64k	6k/48k	/1	2	1		3.3	15.54
TMS320LC546A-66	66	64k/64K	6k/48k	/1	2	1		3.3	17.09
TMS320LC549-80	80	64k/8M	32k/16k	/1	3		1	3.3	22.00
TMS320UC5402-80	80	64k/1M	16k/4k	6Ch int/2	2			1.8	6.79
TMS320UC5409-80	80	64k/8M	32k/16k	6Ch ext/1	3			1.8	17.10
TMS320UVC5402-30	30	64k/1M	16k/4k	6Ch int/2	2			1.2	7.75
TMS320UVC5409-30	30	64k/8M	32k/16k	6Ch ext/1	3			1.2	18.52
TMS320VC5401-50	50	64k/1M	8k/4k	6Ch int/2	2			1.8	
TMS320VC5402-100	100	64k/1M	16k/4k	6Ch int/2	2			1.8	5.66
TMS320VC5409-100	100	64k/8M	32k/16k	6Ch ext/1	3			1.8	19.03
TMS320VC5409-80	80	64k/8M	32k/16k	6Ch ext/1	3			1.8	9.92
TMS320VC5409A-120	120	64k/8M	32k/16k	6Ch ext/1	3			1.5	
TMS320VC5409A-160	160	64k/8M	32k/16k	6Ch ext/1	3			1.6	
TMS320VC5410-100	100	64k/8M	64k/16k	6Ch ext/1	3			2.5	30.40
TMS320VC5410-120	120	64k/8M	64k/16k	6Ch ext/1	3			2.5	36.49
TMS320VC5410A-120	120	64k/8M	64k/16k	6Ch ext/1	3			1.5	
TMS320VC5410A-160	160	64k/8M	64k/16k	6Ch ext/1	3			1.6	
TMS320VC5416-120	120	64k/8M	128k/16k	6Ch ext/1	3			1.5	
TMS320VC5416-160	160	64k/8M	128k/16k	6Ch ext/1	3			1.6	37.92
TMS320VC5420-200	100	256k/256k	192k/NA	12Ch int/2	6			1.8	60.58
TMS320VC5421-200	100	64k/256k	256k/4k	12Ch ext/2	6			1.8	120.95
TMS320VC5441-532	133	4X96k/2X128k	640K/	24Ch ext/4	12			1.5	190.64
TMS320VC549-100	100	64k/8M	32K/16K	/1	3		1	2.5	24.23
TMS320VC549-120	120	64k/8M	32K/16K	/1	3		1	2.5	

Nota: el último número en el nombre del dispositivo, se refiere al número de MIPS. El precio de los dispositivos que no lo poseen, se debe consultar al proveedor.

ARQUITECTURA TMS320C54x™ DSPs

Características principales
Acelerador Viterbi reduce Viterbi “butterfly update” para solo cuatro ciclos de instrucción para decodificar canales GSM, lo que libera MIPS de manera que el CPU puede hacer otras tareas. Cuatro buses internos y generadores de direccionamiento dual permiten multitareas y búsquedas de datos y reducen el efecto de embudo en la memoria
Un sumador de 40 bit y un acumulador de 40 bit soportan instrucciones cruciales paralelas que se ejecutan en un solo ciclo
Un segundo sumador de 40 bit permite a la salida del multiplicador operaciones MAC “unpipelined”, así como adición y multiplicación en paralelo
Single cycle normalization y codificación exponencial soportan aritmética de punto flotante, muy usada en codificación de voz
Un multiplicador 174 x 17 permite multiplicaciones de 16 bit con o sin signo, con redondeo y control de saturación, todo en un solo ciclo de instrucción
Nuevas instrucciones de ciclo simple, ejecutan eficientemente tareas comunes para DSPs tales como filtros simétricos FIR
Una unidad lógico aritmética (ALU) de 40 bit, caracterizada por una compatibilidad dual de configuración de 16 bit permitiendo sumas dobles en un solo ciclo
8 registros auxiliares y un software de pila (stack software) permiten una avanzada compilación en C de punto fijo
Versiones de desempeño multi-foco (C542x yC544x) diseñadas a satisfacer aplicaciones de infraestructuras de consumo eficiente, tales como puertos universales

GENERACIÓN C55x

Tabla 4.13 Lista genérica de DSPs de la generación C55x

Nombre Del Dispositivo	Frec MHz	MIPS	Espacio total de direccionamiento de memoria (words)	RAM/ ROM (words)	DMA/ timers	Total Puertos serie	Vcc (V)	Suministro E/S (V)	Costo US$
TMS320VC5502-200	200	400	8M (16M bytes)	32k/16k	6 Ch int-ext/3	3 McBSPs 12C,UART	1.5	3.3	
TMS320VC5509-144	144	288	8M (16M bytes)	128k/32k	6 Ch int-ext/2	3 McBSPs 12C,MS,MMC/SD	1.5	2.5-3.6	
TMS320VC5509-200	200	400	8M (16M bytes)	128k/32k	6 Ch int-ext/2	3 McBSPs 12C,MS,MMC/SD	1.5	2.5-3.6	
TMS320VC5510-160	160	320	8M (16M bytes)	160k/16k	6 Ch int-ext/2	3 McBSPs	1.6	3.3	
TMS320VC5510-200	200	400	8M (16M bytes)	160k/16k	6 Ch int-ext/2	3 McBSPs	1.6	3.33.3	

Nota: los precios no están disponibles, se debe contactar directamente al distribuidor

ARQUITECTURA TMS320C55x™ DSPs

Características principales

Manejo automático avanzado de consumo: los DSPs de la generación C55x proporcionan un control automático de consumo para todos sus periféricos, arreglos de memoria y unidades individuales del CPU. La unidad central de los DSPs de la generación C55x monitorea continuamente cuales partes del chip están en uso, cortándoles la energía cuando estas no están en uso.

Dominios Idle incrementados: para un mayor manejo de consumo, el dominio Idle es configurable por el usuario, configurando el consumo de energía dependiendo de la aplicación. El DSP C55x extiendo las tres ramas fijas Idle de la generación C54x a un total de 64 combinaciones configurables por el usuario, de los siguientes 6 componentes:

- CPU
- Caché
- DMA
- Periféricos
- Generador de reloj
- Interface de memoria externa (EMIF)

Una de las innovaciones más importantes es el soporte de instrucciones de tamaño variable, basado en un nuevo esquema de byte de direccionamiento.

- El tamaño de la instrucción puede ser de 8-16-24-32-40 o 48 bits
- Búsqueda de instrucciones incrementada de 16 bits a 32 bits
- Una unidad de buffer de instrucción interna automáticamente desempaca instrucciones para lograr dar la máxima eficiencia al uso de cada ciclo de reloj

La reducción de la actividad de la unidad del bus de memoria, disminuye el consumo de energía, aún cuando instrucciones grandes acarrean fuera más funciones pore ciclo de reloj, resultando en un incremento funcional y un sistema de menor costo

La unidad central de la generación C55x se centra en proporcionar paralelismo en los ciclos, mejorando:

- **Hardware adicional:** dos 17 x 17 bits MACs, una segunda ALU de 16 bit, 4 nuevos registros de datos (que pueden ser usados para cálculos simples) para realizar más de un trabajo por ciclo, disminuyendo el consumo
- **Capacidad de nuevas instrucciones:** instrucciones auto paralelas, instrucciones paralelas implícitas o en construcción, instrucciones paralelas del usuario, instrucciones extras para incrementar la ortogonalidad
- **Buses adicionales y direccionamiento expandido:** para asegurar la obtención de las máximas posibilidades, la unidad central tiene: 3 buses de lectura de datos de 16 bit, 2 buses de escritura de 16 bits, un bus de programa de 32 bits y 6 buses de direccionamiento de 24 bits

Control de densidad de código mejorado: nuevas características del control de código en la unidad central de los DSPs C55x hace posible integrar el control de código en el DSP, eliminando la necesidad de un microcontrolador por separado:

- **Nueva unidad de buffer de instrucción:** capacidad para manejar instrucciones de tamaño variable, lo que significa control de código compacto y manejo eficiente, reduciendo el consumo
- **Nuevos registros de datos y ALU:** 4 nuevos registros de datos, junto a una nueva ALU de 16 bits, hace posible acarrear fuera las operaciones aritméticas y lógicas simples, que son típicas de la unidad de control central
- **Ejecución condicionada:** muchas operaciones de control envuelven saltos condicionados. Para acelerar la ejecución, la unidad central se prepara para ambas posibilidades, de manera que cuando la ejecución ocurre, el DSP esta listo para actuar inmediatamente.

Interface de memoria externa: la unidad central de los DSPs C55x el EMFI incluido incrementa el ancho de banda, expande las opciones de memoria y una característica de corte de energía automático, más una variedad de interfaces de alta velocidad de 32 bits, bajo costo y memorias sincrónicas, como:

- Burst SRAM sincrónico y DRAM sincrónico
- SRAM, DRAM, ROM y Flash asincrónicos

Instrucciones caché con Burst Fill: la distancia entre la memoria externa e interna se vuelven factores para mantener la operación del chip a máxima velocidad. En la suma, esto solo toma un poco más de energía para acercar la información que esta fuera del chip. Cargar instrucciones desde la memoria externa en el caché interno ayuda a asegurar que las instrucciones podrán estar disponibles a la velocidad de reloj. Esta ayuda a disminuir el consumo, porque varias instrucciones pueden ser cargadas en caché a la misma vez y el CPU no tiene que tener acceso a la memoria en cada instrucción

Depuración Reducida: un nuevo hardware de emulación avanzado incluido en el chip, que trabaja con el software "eXpressDSP™ Real-Time", tecnología para aumentar la velocidad y simplificar la depuración.

- **Depuración sin interrupción:** se fija un "punto de observación" y se observan los cambios en ciertos registros mientras corre el código, sin detener el DSP
- **Real-Time Data eXchance (RTDX):** observa cual salida del programa se verá igual, sin detener el DSP
- **FIFOS trazadas:** trabajar con "TI´s XDS510 Emulator", permite al DSP salvar las últimas 16 discontinuidades y 32 valores de PC

CARACTERÍSTICAS PERIFÉRICAS DE LA PLATAFORMA C5000

El soporte de periféricos integrados en el chip, tales como: McBSPs, DMA, y 8/16 bit HPI, ofrecen gran flexibilidad para desarrollar sistemas portátiles eficientes y operados con bateria.

Multi-Channel Buffered Serial Ports (McBSPs)

Direct memory Access (DMA)

- Transferencia de datos entre puntos en el espacio de memoria sin la intervención del CPU
- Permite movimientos de datos a/desde memoria interna, memoria externa y periféricos para ocurrir en el fondo de la operación del CPU
- Operación independiente del CPU
- Puertos dedicados variables de 6/12/24 canales para transferencias directas EHIP entre EHIP y memoria (solo C55x™)

IOM-2 compatible

- AC97 compatibles
- IIS compatibles
- SPI™

Transmisión y recepción de hasta 32/128 canales

- Tamaño de datos seleccionables 8-12-16-20-24- o 32 bits

Reloj interno y generadores de Frame programables

Hardware interno (COMpress & exPAND) para compression/expansión en los formatos μ-law y A-law

8/16 bit Enhanced Host Port Interface (EHPI)

- Puerto paralelo para el procesador organizador para acceder directamente el espacio de memoria del DSP
- Acceso al espacio de memoria configurable para cada organizador (host) solo en modo HOM o acceso compartido en modo SAM
- El host y el DSP pueden intercambiar información vía memoria interna o externa
- Acceso completo a la memoria interna del DSP (acceso a una porción de la memoria externa del DSP vía memoria interna del bus del DMA)

Tabla 5.14 Herramientas de desarrollo disponibles para la plataforma C5000

Nombre de herramienta de desarrollo	Número de parte	Sistema operativo	Generación	Precio US$
C5402 DSP Starter Kit	TMDS320005402	Win98/2000/NT	Plataforma TMS320C5000	295.01
C5409 Evaluation Module (EVM) Bundle TI/Spectrum Digital	TMDS3P603122	Win 95, 98, NT	Plataforma TMS320C5000	3995.12
C5409 Evaluation Module Bundle - UK + European Power Cords	TMDS3P603122E	Win 95, 98, NT	Plataforma TMS320C5000	
C5416 Evaluation Module (EVM) Bundle TI/Spectrum Digital	TMDS3P603123	Win 95, 98, NT	Plataforma TMS320C5000	3995.12
C5416 Evaluation Module Bundle - UK + European Power Cords	TMDS3P603123E	Win 95, 98, NT	Plataforma TMS320C5000	
TMS320C5000 DSP Teaching Kit	TMDS3200154	Win98/NT/2000	Plataforma TMS320C5000	349.01
TMS320C5000 DSP Teaching Kit - European & UK Pwr Cords	TMDS3200154E	Win 98/2000/NT	Plataforma TMS320C5000	
TMS320C5000 DSP Teaching Kit	TMDS3200154	Win98/NT/2000	TMS320C54X	349.01
TMS320C5000 DSP Teaching Kit - European & UK Pwr Cords	TMDS3200154E	Win 98/2000/NT	TMS320C54X	
THS10064 Evaluation Module	THS10064EVM		TMS320C54X	99.00
THS10082 Evaluation Module	THS10082EVM		TMS320C54X	99.00
THS1206 Evaluation Module	THS1206EVM		TMS320C54X	99.00
THS12082 Evaluation Module	THS12082EVM		TMS320C54X	99.00

5.4 MOTOROLA DSP56000

Los DSP56000/100 son DSPs de propósito general, de punto fijo y de 24 y 16 bit. Dependiendo de la versión estos incluyen hasta 1k word, de memoria interna RAM datos/programa, una interface host paralela, dos puertos serie, un decodificador, un temporizador y algunas variaciones de codificadores. Las variaciones de la familia DSP56000 incluyen: DSP56001(24 bit con ROM interna y tablas de senos), DSP56002 (24 bit), DSP56156(16 bit) y DSP 56166 (16 bit con ROM interno para arranque), estos DSPs por sus características son más usados en el control de motores por lo que quedan fuera del análisis.

Tabla 4.15 Lista de referencia cruzada de DSPs de Analog Device y Texas Instruments

Texas Instruments Familia C54X	Analog Device	Número de núcleos	MMACs	RAM (Kwords)	Voltaje de operación (núcleo-E/S)
TMS320C541	ADSP-2181,2185	1	40	5	5v
TMS320C542	ADSP-2181,2185	1	40	10	5V
TMS320LC541	ADSP-2185L,2186L	1	66	5	3.3V
TMS320LC542	ADSP-2185L,2186L	1	50	10	3.3V
TMS320LC543	ADSP-2185L,2186L	1		10	3.3V
TMS320LC545A	ADSP-2185L,2186L	1	66	6	3.3V
TMS320LC546A	ADSP-2185L,2186L	1	66	6	3.3V
TMS320LC548	ADSP-2185L,2187L	1	66	32	3.3V
TMS320LC549	ADSP-2185L,2187L	1	80	32	3.3V
TMS320UC5402	ADSP-2186N	1	80	16	1.8V, 1.8v-3.3v
TMS320UC5409	ADSP-2185N	1	80	32	1.8V, 1.8v-3.3v
TMS320UVC5401	ADSP-2184N	1	50	8	1.8V, 3.3v
TMS320UVC5402	ADSP-2186N	1	30	16	1.2V, 1.2V-2.75V
TMS320UVC5409	ADSP-2185N	1	30	32	1.2V, 1.2V-2.75V
TMS320VC549	ADSP-2185N	1	120	32	2.5V, 3.3V
TMS320VC5402	ADSP-2186N	1	100	16	1.8V, 3.3V
TMS320VC5409A	ADSP-2185N	1	160	32	1.8V, 3.3V
TMS320VC5410A	ADSP-2187N	1	160	64	2.5V, 3.3V
TMS320VC5416	ADSP-2188N	1	160	128	1.5V, 3.3V
TMS320VC5420	ADSP-2188N	2	200	200	1.8V, 3.3V
TMS329VC5421	ADSP-2192	2	200	256	1.8V, 3.3V
TMS320CV5441	ADSP-2192	4	532	640	1.5V, 3.3V
TMS320VC5510	ADSP-2192X	1	400	160	1.6V, 3.3V

5.5 CONCLUSIONES

Los DSP son poderosas herramientas para el manejo de señales en tiempo real, razón por la cual son usados en muchas áreas interdisciplinarias, entre las que se encuentra el procesamiento de señales de voz, síntesis, transmisión, compresión y reconocimiento.

Dadas sus características de tamaño, consumo, costo y poder de procesamiento son especialmente útiles para aplicaciones portátiles, tales como la telefonía movil.

Los DSP Analog Device y Texas Instruments son compatibles con lenguaje C, al igual que el software MATLAB, por lo que el desarrollo de una implementación de reconocimiento de voz desarrollada en lenguaje MATLAB, se puede trasladar al DSP sin ningún problema, dando las modificaciones de formato necesarias. Además MATLAB ofrece una librería llamada “SIMULINK” con herramientas adaptadas para procesos con procesadores DSP, llamada DSP Blockset.

Los DSP son construidos y optimizados para realizar operaciones como FFT, Autocorrelación, Correlación y otros algoritmos muy utilizados en el procesamiento de señales de voz, además permiten el trabajo en frames lo que libera al procesador entre frames para realizar otras tareas. Su arquitectura con DMA permite lectura y escritura a puerto y memoria simultáneamente.

CAPITULO 6
IMPLEMENTACIÓN DE UN ALGORITMO DE RECONOCIMIENTO DE VOZ

Las características principales a las que se debe apegar este sistema de reconocimiento, son las siguientes:

- Sistema dependiente del locutor.
- Palabras aisladas
- Vocabulario pequeño
- Señal de entrada de calidad telefónica.

Para consultas sobre estas características en detalle, remitirse al capítulo 2, en donde se explican las diferentes características que tienen los sistemas de reconocimiento.

Se trabajó sobre dos posibles soluciones a este problema, la primera solución utilizó un DSP e implementaba un algoritmo con LPC y reconocimiento mediante comparación de patrones almacenados en una base de datos. En la segunda se utilizó una combinación de un método de parametrización y una red neuronal, obteniendo excelentes resultados.

6.1 ALGORITMO UTILIZANDO DTW

La primera solución implementada utiliza un principio de reconocimiento muy simple, se compara una palabra o comando de entrada desconocido, con palabras o comandos almacenadas en una base de datos, conocida como librería, se selecciona cual es la palabra más similar a la entrada, la que se convierte en la palabra reconocida y se activa la salida especificada para esa palabra, las palabras en la librería pueden tener uno o más candidatos, la diferencia es el tamaño de la base de datos y el algoritmo de búsqueda y reconocimiento. En la figura 6.1 se presenta un esquema simplificado de dicha solución. La entrada del sistema (voz), captada a través de un micrófono, se convierte a un voltaje digital, a través de un convertidor analógico digital (la señal se muestrea a 8 kHz) es procesada y la palabra reconocida. El proceso de reconocimiento lleva dos partes: la primera es construir una base de datos (fase de entrenamiento del sistema), con las palabras que el sistema será capaz de reconocer y la segunda parte, que es en sí el reconocimiento.

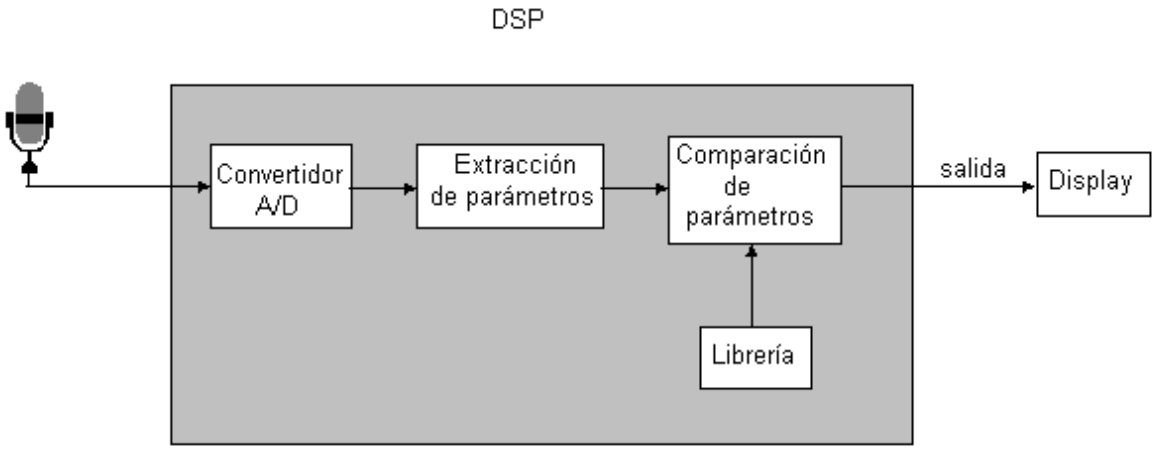

Figura 6.1 Esquema del sistema de reconocimiento de voz empleando base de datos.

Para el reconocimiento de voz, existen muchas técnicas de modelado disponibles, en el esquema anterior el bloque de extracción de parámetros podría ser cualquiera de las técnicas mencionadas en el capítulo 2. En esta solución se utilizará la técnica de “Predicción lineal”. Para este método en particular se hay desarrollado una serie de técnicas de comparación de patrones, por lo que es una solución eficaz para esta forma de solución. Este método estima las características de la envolvente del espectro de la forma de onda de la voz; usando generadores de coeficientes LPC, la redundancia e información innecesaria de la señal de voz es removida, dejando solamente la información esencial para el reconocimiento.

Se escogió un sistema de reconocimiento dependiente del locutor, porque requiere menos memoria para la base de datos. La operación de un sistema independiente del locutor y de uno dependiente del locutor, varía principalmente en la forma de buscar las palabras en la base de datos, el principio en el trato de la señal de voz sigue siendo el mismo. En un sistema dependiente del locutor, idealmente el entrenador y el usuario, deben ser la misma persona; de esta manera, la librería de reconocimiento será relativamente pequeña, ya que quedan por fuera las variaciones por dialecto, acento y otras. Un sistema independiente del locutor, usualmente necesita de muchas muestras de una misma palabra, por lo cual requiere de grandes cantidades de memoria. A pesar de esta diferencia el método de comparación en el reconocimiento sigue siendo el mismo.

Las palabras a reconocer se encuentran aisladas o fuera de contexto, esto para facilitar el diseño al dejar fuera parámetros, que se producen en el reconocimiento de voz en contexto, que requieren una depuración mayor del sistema, se requiere una pausa entre palabras.

Como se mencionó anteriormente un sistema desarrollado de esta forma requiere una fase de entrenamiento y una fase de reconocimiento. Ambas utilizan las mismas herramientas para la extracción de parámetros; pero la segunda utiliza unos algoritmos extras para la fase de comparación de patrones.

FASE DE ENTRENAMIENTO

La primer fase de creación del sistema de reconocimiento es la creación de la base de datos. En un sistema dependiente del locutor la palabra a reconocer puede tener solo un candidato almacenado o varios, sistema de múltiples candidatos, si el sistema es independiente del locutor, se requiere además, que hallan muestras de distintas personas, por lo que la base de datos puede crecer hasta cientos de palabras almacenadas, para unas pocas palabras a reconocer.

En la figura 6.2 se muestra un diagrama de bloques de la fase de entrenamiento del sistema, con las diferentes etapas de que deben realizarse antes de almacenar una palabra válida.

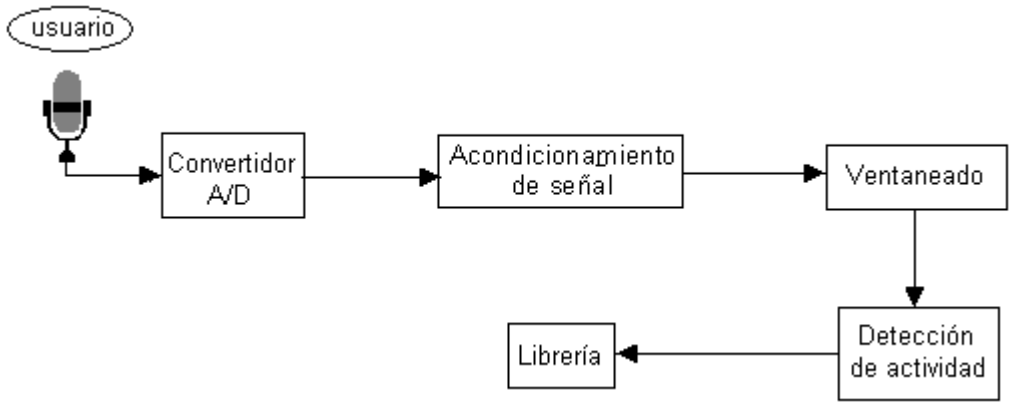

Figura 6.2 Diagrama de bloques de la fase de entrenamiento del sistema.

ACONDICIONAMIENTO DE SEÑAL

El objetivo del bloque de Acondicionamiento es conseguir, en lo posible, reducir la variabilidad presente en la señal de voz debida principalmente a diferentes micrófonos, ruidos y canales de comunicación. La robustez frente a entornos de trabajo diversos es uno de los aspectos principales a tener en cuenta en el diseño de sistemas de utilidad real. Este bloque muchas veces se encuentra dentro del detector de actividad o en la extracción de parámetros, sin embargo se explica por separado para una comprensión más a fondo. En esta aplicación se propuso un filtro pasabanda, para eliminar los sonidos que acompañan una señal de calidad telefónica. El filtro consistía en un filtro pasabanda, tipo Butterword de orden 10, con un ancho de banda limitado de 300 a 3400 Hz.

VENTANEADO

Este proceso lo que busca es fraccionar la señal en pequeños tramos, para la aplicación posterior de alguna de las técnicas de extracción de parámetros, como se explica en el capítulo 2. Para esta implementación se utilizó una ventana de Hamming de 20 ms.

DETECTOR DE ACTIVIDAD

Este módulo es el encargado de realizar una detección automática de los instantes de comienzo y final de la voz de entrada al sistema. Podemos decir que estamos ante un reconocedor de voz y silencio; de hecho, en algunas configuraciones no existe distinción alguna entre el reconocimiento de unidades lingüísticas y el reconocimiento de ruido o silencio. Su funcionamiento es crítico y especialmente difícil en entornos telefónicos. Un error del detector de actividad puede suponer:

La pérdida de parte del mensaje pronunciado por el usuario, sin posibilidad alguna de éxito en el proceso de reconocimiento.

La aceptación de sonidos indeseados capaces de ser confundidos con unidades lingüísticas.

De los dos tipos de errores anteriores el primero es irrecuperable, por lo que el diseño de un detector de actividad suele hacerse buscando que se produzca el menor número de veces posible, aún a costa de permitir una mayor aceptación de sonidos indeseados.

Este elemento es importante si se desea que el sistema no esté siempre reconociendo, sino, sólo en aquellos momentos en que hay voz entrante. De esta forma, al liberar al CPU de la tarea de reconocimiento en los momentos en que no hay voz para reconocer, se permite al sistema atender otras tareas, lo cual es muy importante en sistemas reales, en especial si se trabaja con varias líneas.

El detector de extremos es basado en la energía (suma de magnitudes) y la taza de cruces por cero (ZCR) de cada frame de datos de entrada, basada en una modificación al detector de extremos propuesto por Rabiner (observar en la bibliografía). Esta subrutina determina los extremos de la palabra, comparando los valores de entrada con diferentes umbrales, estos umbrales se adaptan para contener ruidos de fondo.

Esta rutina devuelve varias banderas que indican inicio de palabra, posible inicio de palabra, o fin de palabra. Hay dos tipos de umbrales para la energía y ZCR, umbral posible y umbral de inicio de palabra (IP). Los umbrales posibles simplemente son fijados sobre niveles de ruido de fondo y por esta razón, pueden ser excedidos. El Umbral de inicio de palabra se fija relativamente alto para que se exceda solo si se esta en presencia de una palabra, sin embrago, fijarlo muy alto puede causar que se pierdan algunos sonidos leves. Es necesario experimentar para conocer el valor apropiado.

Hay dos umbrales adicionales. El umbral de longitud de palabra mínima, fijado al menor número de frames que puede contener una palabra (10 para esta aplicación). Este debe ser largo, pero no demasiado. El umbral de tiempo es la duración de silencio que debe de transcurrir antes de que el extremo final de una palabra sea detectado. Esto es necesario para permitir silencio en el medio de palabras (sobre todo precediendo paradas, como “t” o “p”).

Al buscar el inicio de palabra, el algoritmo compara primero la energía del frame y la taza de cruces por cero, con el umbral de inicio de palabra, si se excede, la bandera de inicio de palabra se activa y el sistema inicia a almacenar frames. Si el umbral no se excede, se comparan con los umbrales posibles. Si la energía del frame o ZCR excede umbral posible, la bandera posible se activa y el sistema inicia a guardar frames. Para esto ser considerado, el umbral inicio de palabra debe excederse, antes de caer en los posibles umbrales

Una vez que una palabra es determinada, el algoritmo investiga para el extremo final de la palabra. La rutina encuentra el fin de la palabra cuando la energía y ZCR caen debajo de los posibles umbrales para tiempos más largos que el umbral de tiempo. Cuando esto pasa, la bandera de fin de palabra se activa.

EXTRACCIÓN DE CARACTERÍSTICAS

La extracción de características se hizo mediante la técnica de parametrización LPC, utilizando 12 coeficientes y la ganancia, para cada frame. El análisis se baso en frames de 240 muestras, 160 muestras del frame actual y 80 muestras de traslape, con una frecuencia de muestreo de 8 kHz.

ALMACENAMIENTO

El vector de características extraído se almacena en memoria, listo para ser utilizado en la fase de reconocimiento. Si el sistema acepta múltiples candidatos, se almacenarán tantas palabras como se permita.

FASE DE RECONOCIMIENTO

Durante la fase de reconocimiento el sistema compara una palabra de entrada desconocida con las palabras almacenadas en memoria, entonces la palabra en memoria más parecida es seleccionada como resultado y se activa la salida determinada para ella. El método implementado es basado en un sistema de algoritmo de programación dinámica (DTW), que alinea los ejes de tiempo de la palabra de entrada con respecto a la palabra almacenada, para tener control sobre los cambios de velocidad en la pronunciación de la palabra.

Esta fase mantiene la misma estructura para la adquisición de los datos y se agregan las rutinas necesarias para comparar los vectores característicos.

En la figura 6.3 se muestra el diagrama de bloques para la fase de reconocimiento del sistema. Esta fase requiere que la fase de entrenamiento halla sido realizada.

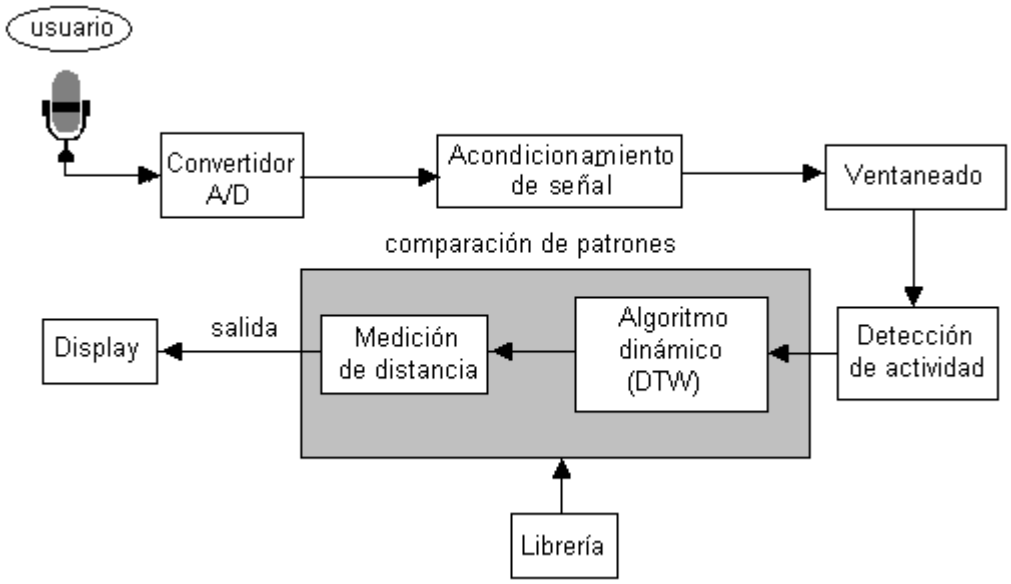

Figura 6.3 Diagrama de bloques de la fase de reconocimiento.

DYNAMIC TIME WARPING (DTW)

Se trata de ajustar la palabra desconocida **x** de longitud **N**, con una palabra o las palabras almacenadas en la librería **y** de longitud **M.** Los índices **y** y **x** denotan un punto particular de tiempo del dato de voz, representado por un vector característico. Una matriz de distancias puede ser calculada para representar las distancias entre el vector característico de **x** (palabra desconocida) y todos los vectores característicos **y** (palabras de la librería), evaluados para $0 \leq x \leq N$. Cada punto de la matriz de distancias tiene un valor que es la distancia entre un vector característico **x** y un vector característico **y**. La medida de distancia específica usada entre vectores característicos es arbitraria. La matriz de distancias es la única cosa que DTW necesita.

Para doblar el eje de tiempo, de la palabra en la librería, al eje de tiempo de la palabra desconocida, varias condiciones se deben fijar. El punto de inicio del alabeo es (0,0) y el punto final (N,M). La inclinación mínima de la torsión es ½ y la máxima 2. Finalmente dos inclinaciones consecutivas de 0 no son permitidas. La figura 6.4 presenta un diagrama de la matriz de distancia con estas condiciones.

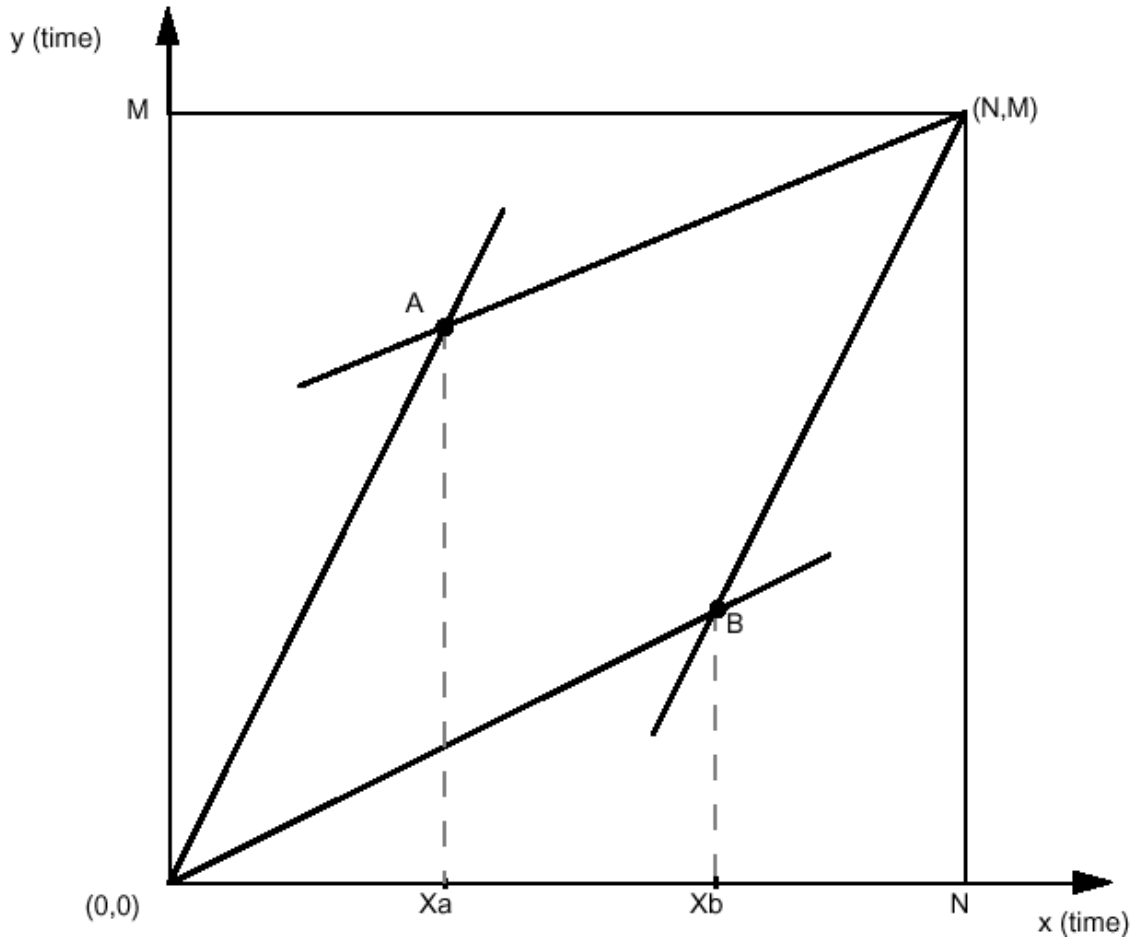

Figura 6.4 Matriz de distancias con las condiciones de declive.

Como se muestra en el diagrama, parte de la matriz de distancias es inválida cuando las condiciones de inclinación y alabeo son impuestas. Tiempo significativo de ejecución es ahorrado si sólo los alabeos permitidos son considerados y sólo los vectores de distancias dentro de los límites de torsión son calculados.

Para determinar los límites del alabeo, los puntos A y B (x_a o x_b) mostrados en el diagrama, deben ser calculados. La siguiente ecuación representa estos puntos:

$$x_a = \frac{1}{3}(2M - N)$$
$$x_b = \frac{2}{3}(2M - N)$$

dado que el proceso actual es desarrollado sólo para puntos donde ***x*** y ***y*** son enteros, los valores de x_a y x_b son redondeados hacia abajo al número entero más cercano en todos los casos, sin perdida de exactitud.

Los valores de x_a y x_b deben estar dentro del rango de $0 < x_a < N$ y $0 < x_b < M$. Esto impone una condición de longitud de la palabra desconocida y la palabra en la librería, la ecuación para estos requerimientos es:

$$2M - N \geq 3$$
$$2N - M \geq 2$$

en esta relación las dos palabras no pueden ser dobladas juntas con esa implementación.

Finalmente, los valores máximos y mínimos de **y** deben ser determinados para cada valor de x, la ecuación para esto es:

$$y_{\min} = \frac{1}{2}x, \;\; 0 \le x \le x_b$$

$$y_{\min} = 2x + (M - 2N), \;\; x_b \le x \le N$$

$$y_{\max} = 2x, \;\; 0 \le x \le x_a$$

$$y_{\max} = \frac{1}{2}x + \left(M - \frac{1}{2}N \right), \;\; x_a \le x \le N$$

El alabeo puede ser dividido en dos o tres secciones, basadas en las relaciones de x_a y x_b. Cada uno de los límites de cada sección, se presentan en la siguiente tabla:

Tabla 6.1 Límites de las secciones del alabeo de tiempo.

Sección	$x_a < x_b$	$X_b < x_a$	$x_a = x_b$
1	$0 \le x \le x_a$	$0 \le x \le x_b$	$0 \le x \le x_a, x_b$
2	$x_a \le x \le x_b$	$x_b \le x \le x_a$	$x_a, xb \le x \le N$
3	$x_b \le x \le N$	$X_a \le x \le N$	

Para cada caso, los límites de **y** son diferentes, pero el alabeo es el mismo. DTW busca la ruta de la palabra de mínima distancia a través de la matriz de distancias, mientras considera las condiciones dadas. Esto es hecho secuencialmente, empezando para x = 0 y terminando con x = N. La siguiente recursión muestra la ruta a través de la matriz que es sujeta a las condiciones de alabeo:

$$D(x,y) = d(x,y) + \min\left[D(x-1,y), D(x-1,y-1), D(x-1,y-2)\right] \quad 0 \le x \le N$$

D(x,y) representa el valor (intermedio) de la distancia de palabra a (x,y), d(x,y) es el valor (del vector de distancia) al punto (x,y).

Puesto que la recursión sólo envuelve los valores en las columnas (x-1) y x, no es necesario calcular la matriz de distancia completa antes de comenzar la recursión.

Para entender mejor el concepto se detallaran los algunos de los algoritmos empleados para hallar DTW. La figura 6.5 muestra con un ejemplo simple del proceso de DTW.

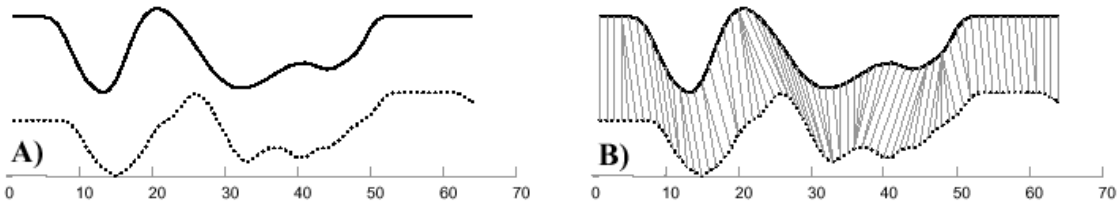

Figura 6.5 En este ejemplo se muestra la utilidad de DTW, a) Aquí se muestran dos secuencias para la palabra “pen”, grabadas en dos días diferentes, b) Se muestra un alineamiento logrado utilizando DTW.

En la figura 6.5a es notorio la similitud en la forma global de ambas secuencias, sin embargo no están alineadas en el eje de tiempo; una medida de la distancia a un punto arbitrario (x,y), produciría una valor diferente para ambas secuencias. Con la herramienta DTW se puede buscar un alineamiento entre las dos secuencias, que permite una medición de distancia más sofisticada, para calcular la distancia al punto (x,y).

Aunque DTW a sido usado con éxito en muchas disciplinas, este puede producir algunos resultados indeseables. La observación crucial es que el algoritmo trata de explicar la variación en el eje Y por el alabeo en el eje X, esto puede llevar a un alineamiento gratuito donde un simple punto en el mapa de una serie de tiempo, sobre una larga sucesión de otra serie de tiempo; estos ejemplos de comportamientos indeseables son llamados “singularidades”, una serie de medidas se han tomado para tratar con las singularidades, todas estas medidas condicionan un posible alabeo permitido. Sin embargo estas sufren del inconveniente que deben prever el “correcto” alabeo para ser encontradas.

En casos simulados, el alabeo correcto puede ser encontrado alabeando una serie de tiempo e intentando recobrar la serie de tiempo original. Naturalmente ocurren casos obviamente correctos, alineaciones “rasgo a rasgo”, con en la figura 6.5b.

Un problema adicional con DTW, es que el algoritmo puede fallar al buscar, alineamientos naturales en dos secuencias simples, porque un rasgo (por ejemplo: un pico, un valle, un punto de inflexión) en una secuencia es ligeramente más alto o pequeño que su rasgo característico en la otra secuencia, la figura 6.6 ilustra este punto.

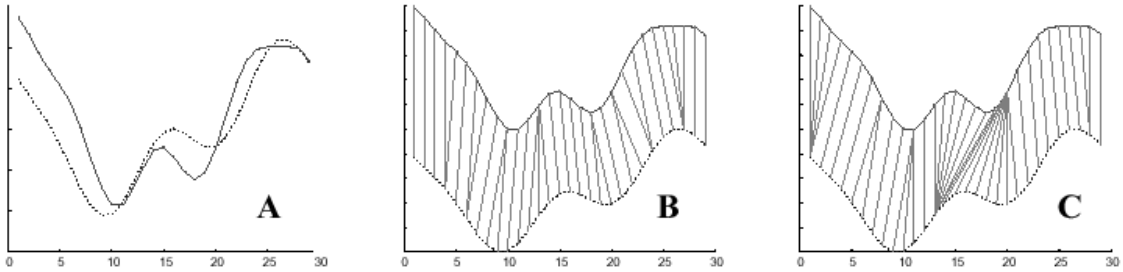

Figura 6.6 a) Dos señales sintetizadas (con la misma media y varianza), b) El alineamiento natural "rasgo a rasgo", c) La alineación producida por DTW.

Se nota como DTW falla en alinear los dos picos centrales, porque están ligeramente más separados en el eje Y. A continuación se detallan los algoritmos para el alabeo de tiempo dinámico.

DTW DINÁMICO SIMÉTRICO O CLÁSICO

Para obtener la distancia global entre dos patrones de voz (representados por una secuencia de vectores) se debe realizar un alineamiento de los ejes de tiempo. Una matriz tiempo-tiempo es usada para visualizar la alineación. En todos los ejemplos de alineación el patrón de referencia (plantilla) va en el eje vertical y el patrón de entrada en el eje vertical.

Se suponen dos series de tiempo Q y C, de longitud n y m respectivamente, donde:

$$Q = q_1, q_2, q_3, ..., q_n$$
$$C = c_1, c_2, c_3, ..., c_m \quad (31)$$

para alinear las dos secuencias usando DTW, se construye una matriz n X m donde los elementos de la matriz (i,j) contienen la distancia $d(q_i,c_j)$ entre dos puntos q_i y c_j.

Cada elemento (i,j) de la matriz corresponde a la alineación entre los puntos q_i y c_j. Esto es ilustrado en la figura 6.7. Una ruta de alabeo W, es un conjunto continuo de elementos de la matriz que define un mapa entre Q y C. El k-ésimo elemento de W es definido como $w_k = (i,j)_k$ así que tenemos:

$$W = w_1, w_2, w_3, ..., w_k \quad (32) \qquad \max(m,n) \le k \le m,n-1$$

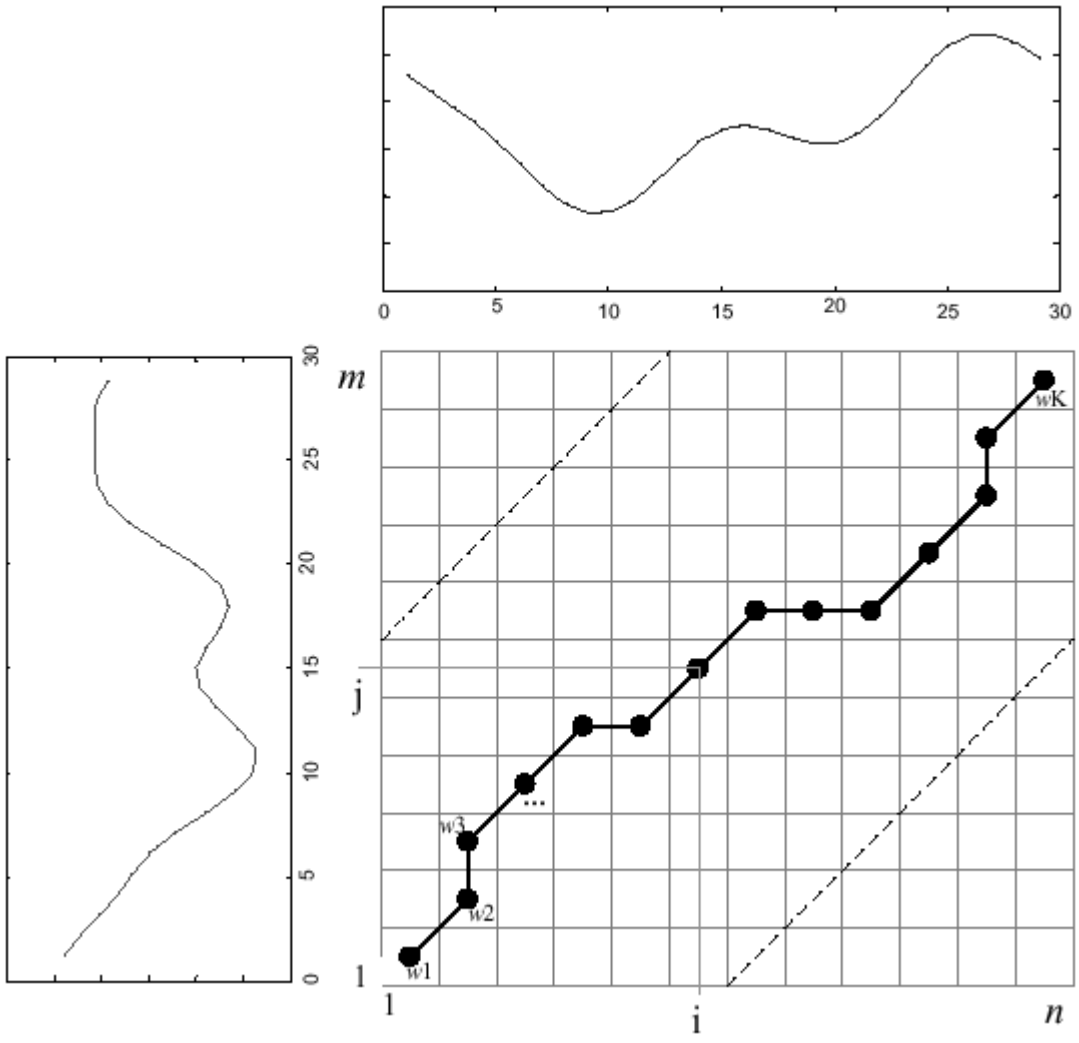

Figura 6.7 Ejemplo de una ruta de alabeo.

CONDICIONES DE LA RUTA DE ALABEO

- **Condiciones límite:** w_1 = (1.1) y w_k = (m,n), esto requiere que la ruta de alabeo inicie y termine en esquinas diagonales opuestas de la matriz.

- **Continuidad:** dado w_k = (a,b) entonces w_{k-1} = (a´,b´) donde a - a´ ≤ 1 y b – b´ ≤ 1. Esto restringe los pasos en la ruta del alabeo a celdas adyacentes (incluidas celdas diagonales adyacentes).

- **Monotidad:** dado w_k = (a,b) entonces w_{k-1} = (a´,b´) donde a - a´ ≥ 0 y b – b´ ≥ 0. Esto restringe a los puntos en W a estar monótonamente espaciados en el tiempo.

Hay muchas rutas de alabeo exponenciales que satisfacen estas condiciones, sin embargo, el interés es en la ruta que minimiza el costo de albeo:

$$DTW\ (Q, C) = \min \left\{ \frac{\sqrt{\sum_{k=1}^{k} w_k}}{k} \right\}$$

k en el denominador es usado para compensar el hecho de que la ruta de alabeo puede tener muchas longitudes.

Esta ruta puede ser encontrada eficientemente usando programación dinámica para evaluar la siguiente recurrencia que define la distancia cumulativa $\gamma(i, j)$ como la distancia d(i, j) encontrada en la celada actual y la mínima de la distancia cumulativa de los elementos adyacentes:

$$\gamma(i,j) = d(q_i, c_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}$$

Esto significa que si tomamos un punto (i,j) en la matriz de tiempo-tiempo (donde el indice i indica el patrón de la ventana de entrada y j el patrón de la ventana de la plantilla o referencia), el punto previo debe ser (i-1,j-1), (i,j-1) o (i-1,j), ver figura 6.8 La idea de DTW en la programación dinámica es que el punto (i,j) debe continuar por la ruta de menor distancia de (i-1,j-1), (i,j-1) o (i-1,j).

Para un reconocimiento básico DP (programación dinámica) tiene un pequeño requerimiento de memoria, el único almacenamiento requerido para la búsqueda es un arreglo que mantiene una columna sencilla de la matriz de tiempo-tiempo.

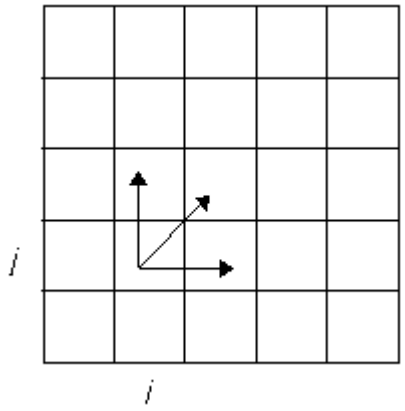

Figura 6.8 Rutas posibles más cercanas que puede seguir a partir del punto (i,j) para DTW simétrico.

Computacionalmente, la ecuación anterior puede ser programada por recursividad. Sin embargo, a menos que el lenguaje sea optimizado para recursión, este método puede ser un poco lento para patrones de tamaño relativamente pequeño. Otro método que es rápido y requiere poca memoria, usa dos ciclos “for” anidados, este método sólo necesita dos arreglos que mantienen las columnas adyacentes de la matriz.

El algoritmo para buscar la distancia menor es:

1. Calcular el inicio de la columna 0. El costo global a esta celda es sólo su costo local. Entonces, el costo global para cada célula sucesiva es el costo local para esa celda más el costo global a la celda debajo de él. Esto se llama el predCol (columna predecesora).

2. Calcular el costo global a la primera celda de la próxima columna (curCol), este es el costo local para la celda, más el costo global menor de la columna anterior.

3. Calcular el costo global del resto de las celdas de curCol. Por ejemplo, d (i,j) es la distancia local a (i,j) más el costo global mínimo a cualquiera de los puntos (i-1,j), (i-1,j-1) o (i,j-1).

4. El curCol se asigna al predCol y se repite el paso 2 hasta que todas las columnas se han calculado.

El costo global es el valor guardado en la celda mayor de la última columna.

CONDICIONANDO EL ALGORITMO CLÁSICO DE DTW

El problema de las singularidades fue notado temprano como en 1978; varios métodos fueron propuestos para aliviar este problema, un pequeño vistazo a estos:

1. La ventana: los elementos permitidos en la matriz se restringen para que calcen dentro de una ventana de alabeo, $|i-(n/(m/j))|<R$, donde R es un valor de ancho de columna entero positivo. Esto significa que las esquinas de la matriz son recortadas por consideraciones, como muestras las líneas punteados en la figura 6.7. Otros experimentaron con varias lotras formas de formar la ventana. Estas condiciones aprovechan el tamaño máximo de la singularidad, pero no previenen que vuelva a ocurrir.

2. Peso de las inclinaciones: si se cambia la ecuación para encontrar los puntos de la ruta, e la siguiente manera:

$$\gamma(i,j)=d(i,j)+\min\{\gamma(i-1,j-1),X\gamma(i-1,j),X\gamma(i,j-1)\}$$

donde X es un número real positivo, se puede condicionar el alabeo para cambiar el valor de X. Como X crece, la ruta de alabeo incrementa se inclinación hacia la diagonal.

3. **Pasos (condiciones de inclinación):** se puede visualizar las siguientes ecuaciones como un patrón de pasos permitidos, tal como se muestra en la figura 6.9A.

Figura 6.9A patrón: $\gamma(i,j) = d(i,j) + \min\{\gamma(i-1,j-1), \gamma(i,j-1), \gamma(i-1.j)\}$

Figura 6.9B patrón: $\gamma(i,j) = d(i,j) + \min\{\gamma(i-1,j-1), \gamma(i-1,j-2), \gamma(i-2,j-1)\}$

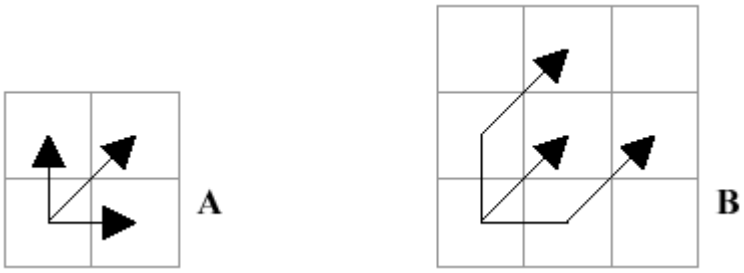

Figura 6.9 Representación pictórica de los pasos permitidos para la ruta de alabeo.

Se puede reemplazar la ecuación:

$$\gamma(i,j) = d(i,j) + \min\{\gamma(i-1,j-1), X\gamma(i-1,j), X\gamma(i,j-1)\}$$

por la ecuación:

$$\gamma(i,j) = d(i,j) + \min\{\gamma(i-1,j-1), \gamma(i-1,j-2), \gamma(i-2,j-1)\}$$

correspondiente al patrón mostrado en la figura 4.B. Usando este patrón la ruta de alabeo es forzada a moverse en pasos diagonales en cada paso paralelo a un eje.

Todo lo anterior puede ayudar a mitigar el problema de las singularidades, pero se corre el riesgo de perder el alabeo correcto.

ALABEO DE TIEMPO DINÁMICO ASIMÉTRICO

Aunque el algoritmo DP básico tiene el beneficio de la simetría (es decir, todas las ventanas, ambas, las de la entrada y la referencia deben ser usadas) este tiene el efecto lateral, que penaliza las transiciones verticales y horizontales al diagonal.

Un camino para evitar este efecto es doblar la contribución de d(i,j) cuando se da un paso en diagonal. Este tiene el efecto de no cobrar penalidades a los movimientos horizontales o verticales, sino a los diagonales. Esto tampoco es deseable, penalizaciones independientes d_h y d_v son aplicados a los movimientos horizontales y verticales.

$$D(i,j) = \min\{D(i-1,j-1) + 2d(i,j), D(i-1,j) + d(i,j) + d_h D(i,j-1) + d(i,j) + d_v\}$$

los valores apropiados para d_h y d_v deben ser encontrados experimentalmente.

Esta aproximación favorece más a las plantillas cortas por encima de las más largas, así que un refinamiento es normalizar el valor de distancia final por longitud de plantilla para reajustar el equilibrio. Si se restringen las transiciones aceptables para ser:

(i-1,j-2) a (i,j) - la diagonal extendida (saltar una ventana de la plantilla - diagonal inclinación 2).

(i-1,j-1) a (i,j) - la diagonal estándar (inclinación 1)

(i-1,j) a (i,j) - horizontal (reproduce una ventana de la plantilla - inclinación 0)

Si se asume que cada frame del patrón de entrada se usa una vez y sólo una vez. Esto significa que se puede distribuir por normalización de longitud de plantilla y no se requiere agregar la distancia local dos veces para la ruta de transición diagonal (cuesta 1). Este acercamiento es llamado programación dinámica asimétrica.

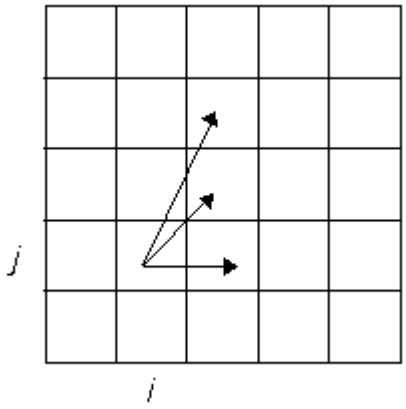

Figura 6.10 Las tres posibles direcciones en las cuales la ruta desde la cela (i,j) puede se tomada, en el alabeo de tiempo dinámico asincrónico

Sin embargo, se puede notar que ocurre un caso especial, considerando la ruta inicial desde (0,0), como la ruta se debe mover a la columna 1, entonces la distancia global a la fila 1, 2 de la primera columna pierde sentido, como se ilustra en la figura 6.11.

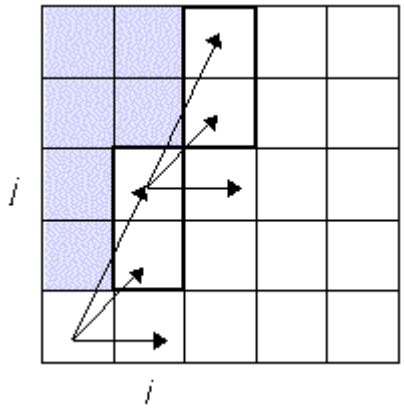

Figura 6.11 Esta figura muestra las tres posibles direcciones en las cuales se puede tomar la ruta desde la celda (i,j) en el alabeo de tiempo dinámico asimétrico.

Ahora considere el punto (i,j) en Figura 6.11. Este puede ser considerado un caso especial. Caso especial de elevación en virtud del hecho que el camino que debe seguirse a lo largo del patrón archivado es de un frame a la vez. Esto significa que la ruta más cercana sólo puede llegar como una celda de "caso especial" desde un número limitado de localidades. Como puede verse en la Figura 6.11, tales casos especiales aparecen en pares que se localizan progresivamente más alto en cada columna subsecuente hasta un par (o la mitad de un par si la altura de la columna es igual) localizado en la superficie de una columna. Todas las columnas restantes pueden repartirse como normales. Para celdas normales y especiales, las filas 0 y 1 se reparte con de una manera diferente de lo normal.

DTW DERIVATIVO (DDTW)

Si sólo se consideran los valores de los puntos en el eje Y, por ejemplo considerar dos puntos q_i y c_i que tienen idénticos valores, pero q_i es parte de un punto de subida y c_j es parte de un punto de caída, DTW considera un mapa ideal entre estos puntos, aunque intuitivamente preferiríamos no mapear un punto en dirección de subida como uno de bajada, para evitar este problema DDTW propone una modificación que no hace consideración de los valores de los puntos en el eje Y; prefiere considerar el rasgo de nivel más alto de "forma".

Obtenemos esta información de la forma considerando la primera derivada de la secuencia. Este método a diferencia del clásico en que mide la distancia a un punto (i,j) que contiene la distancia $d(q_i, c_i)$ entre dos puntos q_i, y c_i, este método prefiere calcularla como el cuadrado de la diferencia de las derivadas estimadas de q_i y c_i. Existen métodos sofisticados para calcular las derivadas, se puede estimar de forma simple y general de la siguiente forma:

$$D_x[q] = \frac{(q_i - q_{i-1}) + ((q_{i+1} - q_{i-1})/2)}{2} \qquad 1 < i < m$$

Este estima simplemente el promedio de las inclinaciones de las líneas a través de los puntos en cuestión y su vecino izquierdo, y la inclinación de la línea a través del vecino izquierdo y el vecino derecho. Empíricamente esta estimación es más robusta para contornos que cualquier otra estimación que considere sólo dos puntos. Nótese que la estimación no está definida por el primero y el último punto de la secuencia, en cambio se usan el segundo y el penúltimo elemento respectivamente. Para ruidos se usa un suavizador (para allanar) antes de estimar las derivadas.

En la figura 6.12 se muestra una comparación entre DTW clásico y DDTW, se nota como en la figura 6.12d el algoritmo para DDTW ofrece mejores resultados.

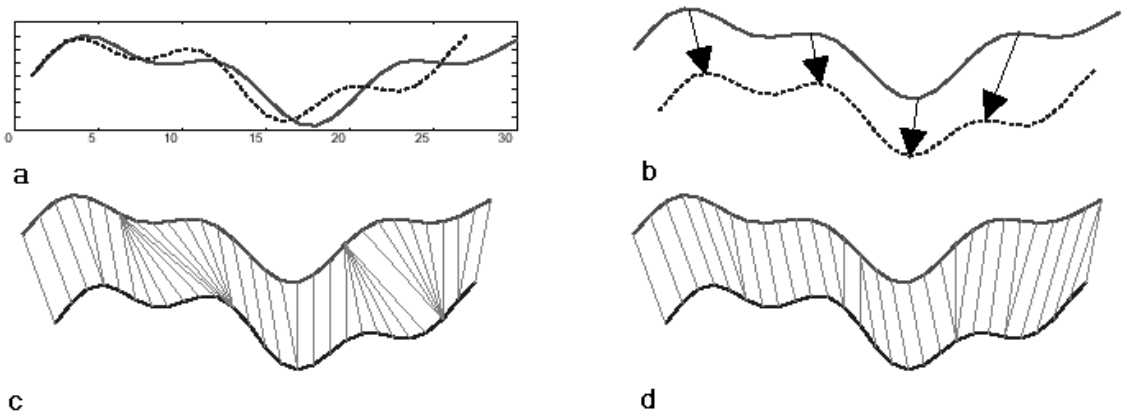

Figura 6.12 a) Dos señales artificiales, b) Alineamiento intuitivo rasgo a rasgo, c) DTW, d) DDTW.

La entrada desconocida y las palabras en la librería, son representadas como una serie de vectores característicos. Para comparar dos palabras una medida de la similitud entre estas es necesaria. Al nivel más básico, el sistema debe medir la similitud entre dos vectores característicos; a esto se le conoce como medida de distancia o distorsión. Muchos métodos de medición de distorsión son propuestos en la literatura, dos de los populares y que parten de los coeficientes LPC, son:

- Itakura log-likelihood ratio
- Bandpass cepstral distortion measure

Durante la acción de reconocimiento, la medición de distorsión es integrada dentro de la rutina de DTW. El “mejor ajuste” entre la palabra de entrada desconocida y la almacenada en la librería es calculado, el sistema compara la entrada desconocida con cada palabra almacenada en la librería, en turno. El sistema tiene una puntuación para cada plantilla (vector característico, que contiene los coeficientes LPC obtenidos) almacenada en la librería. Para un sistema de una palabra por plantilla (usualmente un sistema dependiente del locutor) típicamente, el sistema escoge la puntuación más baja como el resultado del reconocimiento. Para sistemas independientes del locutor, donde más de una plantilla por palabra es almacenada, la puntuación más baja para cada palabra de la librería es promediada, este es un valor medio de la puntuación de la palabra de reconocimiento, para cada palabra en la librería, el sistema sigue seleccionando la puntuación más baja como el resultado del reconocimiento.

MEDICIÓN LOG LIKELIHOOD RATIO (LLR)

El LLR es referido como la medición de distancia Itakura. La distancia LLR para un segmento de voz es basada en la convicción de que un segmento de voz puede ser expresado mediante un modelo de polos LPC de orden p de la forma en que ya lo conocemos:

$$x[n] = \sum_{m=1}^{p} a_m x[n-m] + G_x u[n]$$

donde: $x[n]$: es la n-ésima muestra de voz

a_m : (donde m = 1,2,3,...,p) son los coeficientes del filtro de polos

$x[n-m]$: representan las salidas anteriores consideradas.

G_x: la ganancia del sistema para la señal de entrada (constante)

$u[n]$: la entrada actual.

$$LLP = \log\left(\frac{\vec{a}_x \overline{R}_y \vec{a}_x^T}{\vec{a}_y \overline{R}_y \vec{a}_y^T}\right)$$

donde:

$\vec{a}_x$: vector de los coeficientes LPC $(1,-a_x(1),-a_x(2),...,-a_x(p))$ de la señal original $x[n]$.

$\vec{a}_y$: vector de coeficientes LPC $(1,-a_y(1),-a_y(2),...,-a_y(p))$ para la señal "deformada" (mediante DTW) de voz $y[n]$.

$\overline{R}_y$: es la matriz de autocorrelación para la señal de voz "deformada"

Como la LLR esta basada bajo la concepción de que la señal de voz esta bien representada mediante el modelo de polos LPC, el desempeño de la LLR esta limitado por las condiciones de distorsión donde esta concepción es válida. Esta concepción puede no ser valida si la señal de voz es pasada a través de un sistema de comunicación que cambia significativamente las estadísticas de la señal de voz original.

MEDICIÓN DE PARÁMETROS LPC

Motivado por la predicción lineal de la voz, medidas objetivas de la calidad de la voz pueden comparar los parámetros de la predicción lineal de la señal de voz original y la voz distorsionada. Los parámetros usados en la medición de parámetros LPC pueden ser los coeficientes de predicción o transformaciones de los coeficientes de predicción tales como relaciones de coeficientes de área.

Una forma para medir los parámetros de distancia, presenta matemáticamente la siguiente forma:

$$d(Q,p,m)=\left(\frac{1}{N}\sum_{i=1}^{N}\left|Q(i,m,x)-Q(i,m,y)\right|^{p}\right)^{\frac{1}{p}}$$

donde:

$d(Q,p,m)$: es la medida de la distancia del análisis de la ventana *m*

p: en la energía en la normal

N: es el orden de análisis LPC

$Q(i,m,x)$ y $Q(i,m,y)$: son el i-ésimo parámetro de la correspondiente ventana de la voz original y la distorsionada, respectivamente.

La correspondiente medida de distancia para cada ventana se asume para todas las ventanas, de la siguiente manera:

$$D(p) = \frac{\sum_{m=1}^{M} W(m) d(Q, p, m)}{\sum_{m=1}^{M} W(m)}$$

donde:

$D(p)$: es el resultado de la distorsión estimada

M: es el número total de ventanas (frames)

W(m): es el peso asociado con la medida de distancia para la m-ésima ventana.

El peso podría, por ejemplo, ser la energía en el análisis de la ventana de referencia. Esta medición a sido investigada para varias formas de parámetros LPC, entre ellos, la medición logarítmica de área a reportado tener la correlación más alta con subjetiva calidad. Esta fórmula general, pude ser usada en mediciones objetivas de calidad de voz en el cálculo de distorsión para una probar una muestra.

MEDICIÓN DE DISTANCIA CEPSTRAL (CD)

La distancia cepstral (por sus siglas en inglés, CD) es otra forma de medición de parámetros LPC, utilizando los coeficientes de la predicción lineal para calcular los coeficientes cepstral de la diferencia global entre la original y su correspondiente codificación de voz cepstrum. El cálculo de cepstrum a partir de los coeficientes LPC, a diferencia del cálculo directo a partir de la forma de onda, resulta en una estimación del espectro llano de voz, puede ser escrito como:

$$\log\left(\frac{1}{A(z)}\right) = \sum_{k=1}^{\infty} c(k) z^{-k}$$

donde:

A(z): es el filtro polinomial de análisis LPC

C(k): denota el k-ésimo coeficiente cepstral

Z puede ser tomado igual a e^{-jw}

También hay otra forma para calcular los coeficientes cepstral a partir de los coeficientes LPC, siguiendo la siguiente formula:

$$nc(n) - na(n) = \sum_{k=1}^{n-1} (n-k) c(n-k) a(k) \quad \text{para n=1,2,3,..}$$

donde:

a(0) y *a(k) = 0* para *k > p.* En esta expresión, *a(k)* es el coeficiente LPC y p es el orden LPC. Los coeficientes son calculados por recursión a parte de esta ecuación.

Una medición de calidad objetiva de voz basada en los coeficientes cepstral calcula la distorsión de una ventana:

$$d\left(c_x, c_y, 2, m\right) = \left[\left(c_x(0) - c_y(0)\right)^2 + 2\sum_{k=1}^{L} \left(c_x(k) - c_y(k)\right)^2 \right]^{\frac{1}{2}}$$

donde d es la distancia L de la ventana m y c_x(k) y c_y(k) son los coeficientes cepstral para la señal de voz original y la distorsionada, respectivamente. Finalmente la distorsión se calcula sobre todas las ventanas usando la ecuación:

$$D(p) = \frac{\sum_{m=1}^{M} W(m) d(Q, p, m)}{\sum_{m=1}^{M} W(m)}$$

Esta implementación requiere de una base de datos, la que crece dependiendo del número de palabras que se desea reconocer y si es dependiente o no del locutor.

El sistema de reconocimiento que utiliza el algoritmo descrito, no se implementó satisfactoriamente, debido principalmente a que se intentó programarlo directamente en el DSP, si contar con una fase previa de simulación que garantizara dos cosas: el correcto desempeño del código implementado para las diferentes subrutinas del programa y el funcionamiento de este en el ambiente y las condiciones establecidas. En segundo lugar el software de ensamblador utilizado, no contaba con un simular o un depurador que permitiera localizar con facilidad los puntos en donde se encontraban los errores del código, por lo que buscar un error se convertía en una tarea tediosa y un proceso de prueba y error.

Por esta razón no se cuenta con un registro de las prestaciones que ofrece este algoritmo para el reconocimiento de voz, en las condiciones establecidas.

6.2 ALGORITMO EMPLEANDO REDES NEURONALES

Aprovechando las ventajas que brindan las redes neuronales, se llevó a cabo un nuevo algoritmo de reconocimiento, en el que se simplifica considerablemente el número de elementos funcionales involucrados, con respecto a la solución anterior. Los bloques reemplazados corresponden a la base de datos, el logaritmo de alineamiento temporal DTW, la medición de distorsión y los algoritmos de búsqueda y almacenamiento en la base de datos. Todos estos bloques son sustituidos por la red neuronal, logrando una simplificación en el tamaño del sistema, aumentando la velocidad de cálculo.

El funcionamiento del sistema es muy simple, se toma una señal de voz, se extrae el patrón de características y se alimenta a la red neuronal, se discrimina entre las alternativas y se obtiene el resultado, tal como se muestra en la figura 6.13.

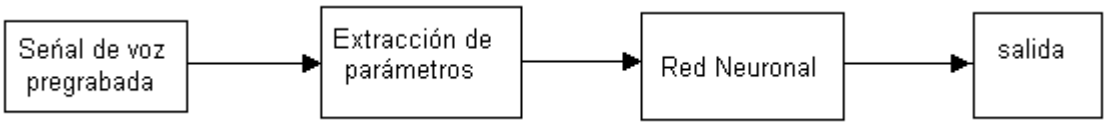

Figura 6.13 Diagrama de bloques general del sistema de reconocimiento de voz empleando una red neuronal.

Para comprobar la eficiencia del algoritmo, se implementaron tres versiones, utilizando diferentes técnicas de parametrización: Real Cepstrum, Complex Cepstrum y Mel Frequency Cepstrum, obteniendo mejores resultados empleando Complex Cepstrum como se muestra en las tablas 6.2 y 6.3.

La red neuronal empleada fue una red de retropropagación, con un algoritmo de entrenamiento “Resilent Backpropagation” (ver manual de redes neuronales en MATLAB), la creación de la red y la fase de entrenamiento son procedimientos simples, determinar la cantidad de neuronas y capaz, que debe contener la red, es un trabajo de intuición, mediante prueba y error. En este trabajo se comenzó con 2 capas ocultas con 5 neuronas y dos neuronas en la capa de salida.

El sistema se formuló para poder reconocer dos palabras, dentro de un conjunto de palabras desconocidas; logrando el éxito en esta red, aumentar la capacidad del sistema es cuestión de colocar nuevas redes en paralelo, entrenadas para reconocer palabras diferentes, sin embargo, no se asegura que la misma configuración sea válida.

Los vectores de entrada al sistema son señales de voz en formato “wav”, pregrabadas en disco duro, el sistema no admite silencios antes del inicio de la señal, se considera la duración de la palabra de 0.5 segundos y con una frecuencia de muestreo de 8 kHz.

El sistema no está implementado para reconocimiento en tiempo real. En Matlab es difícil conseguirlo, no así en Simulink, sin embargo, para ello se necesitaría construir un detector de extremos,

El primer paso de entrenamiento consistió en estructurar la red, con los pasos mostrados en el experimento 2 (Apéndice A.3), las pruebas se realizaron con el mismo entrenador, los resultados se muestran en la tabla 6.2. Para la parametrización Real y Complex Cepstrum, el entrenamiento fue 100% exitoso, sin embargo, al probar el sistema con nuevos vectores, la red que empleaba Complex Cepstrum, tuvo mejor eficiencia 60%. El entrenamiento con Mel Cepstrum, ofreció un 7.5, a pesar de utilizar una cantidad mucho mayor a las empleadas a las otras técnicas.

Tabla 6.2 Resultados del entrenamiento realizado con diferentes parametrizaciones.

Método de parametrización	Número de neuronas por capa	Porcentaje de éxito en el entrenamiento	Porcentaje de éxito en pruebas
Real Cepstrum	8x8x8x2	100%	40%
Complex Cepstrum	8x8x8x2	100%	60%
Mel Frequency Cepstrum	20x20x20x2	7.5%	0%

La segunda prueba realizada a los sistemas obtenidos, fue medir la capacidad de generalización, para ello se utilizaron voces diferentes a la del entrenador. El porcentaje de éxitos obtenidos se muestra en la tabla 6.3

Tabla 6.3 Porcentaje de éxito de generalización presentado por el sistema.

Método de parametrización	**Hombres**	**Mujeres**
Real Cepstrum	42%	20%
Complex Cepstrum	42%	60%

De los datos anteriores se desprende claramente, como este método es 100% efectivo para reconocer un vocabulario fijo, con el cual fue entrenado, con un pequeño número de elementos en el algoritmo. Al momento de reconocer palabras diferentes a las que fue sometido durante el entrenamiento, el algoritmo no presentó los resultados esperados. Sin embargo, el escaso número de muestras de entrenamiento juega un papel importante en la generalización del sistema, los porcentajes sólo reflejan una muestra muy pequeña de lo que es realidad en la población total.

El algoritmo con red neuronal absorbe los bloques de alineamiento temporal DTW, base de datos y comparación de patrones, que contiene el otro algoritmo. Las funciones de almacenamiento y comparación son bien asimiladas por la red neuronal, sin embargo, la variaciones temporales no, esto se nota al momento de comparar el resultado del entrenamiento, en los que se obtiene un éxito del 100% y al probarse con nuevas muestras, los porcentajes disminuyen considerablemente.

6.3 CONCLUSIONES

El sistema implementado utilizando red neuronal no requiere de base de datos, lo que simplifica considerablemente su tamaño.

En el algoritmo con red neuronal las funciones de alineamiento temporal, búsqueda en la base de datos y comparación son absorbidas por la red neuronal.

El algoritmo con red neuronal ofrece una 100% de éxito en el reconocimiento de patrones fijos, si estos se contemplaron en el entrenamiento.

El algoritmo con red neuronal, no ofrece una eficiente capacidad de generalización.

El entrenamiento de una red neuronal es un proceso simple en Matlab, gracias a los algoritmos de entrenamiento que este incorpora.

CAPITULO 7
RECOMENDACIONES

La creación de un sistema para reconocimiento de voz, requiere la comprobación previa de la funcionalidad bajo las condiciones de trabajo del algoritmo de reconocimiento que va emplear, por lo que se recomienda trabajar primero en un programa de simulación como MATLAB, para comprobar la eficiencia del algoritmo, antes de intentar implementarlo.

Implementar un detector de extremos, con lo cual se completaría el sistema para poder trabajar en tiempo real. Existen muchas formas de implementar un detector de extremos una forma recomendada por telefonía I+D con muy buenas prestaciones, utiliza Modelos Ocultos de Markov.

Crear una base de datos amplia, que incluya múltiples candidatos, con un vocabulario extenso, para realizar mejores entrenamientos y comprobaciones de reconocimiento de múltiples locutores.

Para reconocer tramos de voz mayores o disminuir el número de entradas de la red, se puede partir la señal de tiempo en varios tramos (por ejemplo tramos de 250 ms de duración) y analizarla con varias redes.

Utilizar otros métodos de parametrización, como: Transformada Rápida de Fourier, Mel Cepstrum, Modelos Ocultos de Markov, Predicción Perceptual.

Trabajar utilizando MATLAB, este software ofrece un ambiente de trabajo flexible y permite pasar los programas realizados en el a lenguaje C, lenguaje aceptado por los DSP.

Por último implementar el algoritmo, utilizando una PC, un DSP o circuitos integrados RNA.

CAPITULO 8
BIBLIOGRAFÍA

L. Hernández Gómez, F.J. Caminero Gil.”Estado del arte en tecnología del habla”. 1994. < http://www.tid.es/presencia/publicaciones/comsid/esp/articulos/vol52/artic1/1.html>. (23 julio 2001).

C. Crespo Casas, C. de la Torre Munilla, J. C. Torrecilla Merchán.”Detector de extremos para reconocimiento de voz”.1994. < http://www.tid.es/presencia/publicaciones/comsid/esp/articulos/vol52/artic6/6.html >. (19 enero 2002).

E. Gonzáles Berbés, J. Calero Gonzáles. “Aplicaciones de la tecnología del habla”.1994. < http://www.tid.es/presencia/publicaciones/comsid/esp/articulos/vol52/artic2/2.html > (23 julio 2001).

“Redes neuronales artificiales”.< http://www.gc.ssr.upm.es/inves/neural/ann2/anntutorial.html >.(10 dic 2001).

“Uso de redes neuronales para el reconocimiento de voz”.< http://www.intersaint.org/acid/rvpm5.htm >(10 dic 2001).

Don cros.”Fast Fourier Transforms”.15 febrero 2000.< http://www.intersrv.com/~dcross/fft.html >.(30 julio 2001).

Robinson, Tony. “Speech analysis”. 1998.< http://svr-www.eng.cam.ac.uk/~ajr/SA95 >.(30 julio 201).

“DSP markets and examples aplications”.< http://www.adaptyv.com/en/print.php?sid=2 >.(10 agosto 2001).

Texas Instruments.< http://www.ti.com / >.

Analog Device.”Guide”.< http://search.analog.com/search97cgi/s97is.dll >(23 julio).

Analog Device.< http://www.analog.com >.

Manual de Redes Neuronales. Matlab.

Manual de programación en Simulink. Matlab.

Manual de programación en Matlab. Matlab.

Manual de programación ADSP-2181 EZ LITE. Analog Device Inc.

CAPITULO 9
APÉNDICES Y ANEXOS

Apéndice A.1: Abreviaturas

A/D	Convertidor analógico digital
ALU	Arithmetic logic unity
ANN	Artificial Neural Networks
CP	Bandpass Cepstral distortion
CPU	Control processor unity
D/A	Convertidor digital analógico
DDTW	Derivative Dynamic Time Warping
DMA	Direct Memory Accessing
DSP	Digital Signal Processor
DTW	Dynamic Time Warping
$E(e^{J\omega})$	Señal de entrada en el dominio de la frecuencia
e(z)	Señal de entrada en el dominio Z
E/S	Entrada/Salida
e[n]	Señal de entrada en el tiempo discreto
EPROM	Erase programmable read only memory
f_{ac}	Función de activación de la neurona
FFT	Fast Fourier Transform
FIR	Finite impulse response

$H(e^{J\omega})$	Función del filtro en el dominio de la frecuencia
h(z)	Función del filtro en el dominio Z
h[n]	Función del filtro en el dominio del tiempo discreto
HMM	Hidden Markov Models
ICE	Instituto Costarricense de Electricidad
IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response
LLP	Log likelihood ratio
LPC	Linear predictive coding
LPP	Linear perceptual prediction
MAC	Operaciones de multiplicación y acumulación
MIPS	Millones de instrucciones por Segundo
RAH	Reconocimiento automático del habla
RAM	Random access memory
RISC	Reduced instruction set computer
ROM	Read only memory
SRAM	Static random access memory
ST-FT	Short-time Fourier Transforms
UEN	Unidad Estratégica de Negocios
Umb	Umbral
W[n]	Función ventana en el dominio del tiempo discreto
$X(e^{J\omega})$	Función de salida en el dominio de la frecuencia
x(z)	Función de salida en el dominio Z
x[n]	Función de salida en el dominio del tiempo discreto
ZCR	Zero cross rate

Apéndice A.2: Experimento 1.

Objetivo: medir el número de neuronas necesarias para lograr reconocimiento de voz.

Funcionamiento: la red creada posee dos salidas, la primera se activa si la entrada corresponde a un uno y la segundo con un dos, de ser diferentes debe haber un cero en la salida.

Vectores de entradas: corresponden a señales de voz, muestreadas a una frecuencia de 8 kHz, se toma medio segundo de señal (4000 muestras), se parametrizan empleando real cepstrum.

Condiciones de entrenamiento: las muestras de entrenamiento provienen de un único locutor, la red debe responder efectivamente a estas muestras, no se mide la capacidad de generalizar para nuevas muestras.

Rango de las salidas: valores de salidas en el rango de 0.99 a 1, se consideran éxitos. El valor de la salida para considerarse un fracaso y asegurar una buena discriminación tiene que ser menor a 0.

Palabras usadas en el entrenamiento: 5 muestras del uno, 5 muestras del dos, 2 de une, 2 de un, 2 de una, 2 de unidad, 2 de ungir, 2 de doscientos, 2 de dos mil, 2 de doce y 2 de don.

Apéndice A.3 Experimento 2.

Objetivo: entrenar una red neuronal para el reconocimiento de dos palabras.

Funcionamiento: se construye el modelo de reconocimiento mostrado en la figura 6.13, se utilizan tres métodos diferentes de parametrización: Real Cepstrum, Complex Cepstrum y MFCC.

Vectores de entrada: la señal de voz es pregrabada en disco duro a una frecuencia de muestreo de 8 kHz, con una duración de 0.5 segundos

Condiciones de entrenamiento: las muestras de entrenamiento provienen de un único locutor, la red debe responder efectivamente a estas muestras, no se mide la capacidad de generalizar para muestras de diferentes locutores.

Rango de las salidas: valores de salidas en el rango de 0.99 a 1, se consideran éxitos. El valor de la salida para considerarse un fracaso y asegurar una buena discriminación tiene que ser menor a 0.

Se usa una red de retropropagación, inicialmente con dos capas ocultas de 5 neuronas cada una y una capa de salida con 2 neuronas.

Palabras usadas en el entrenamiento: 20 muestras del uno, 20 muestras del dos, 2 de une, 2 de un, 2 de una, 2 de doscientos, 2 de dos mil, 2 de doce.

Terminado del entrenamiento, se prueba la efectividad, con un vector de pruebas, que contiene: 5 muestras del uno y 5 del dos.

Apéndice A..4 Experimento 3.

Objetivo: probar la capacidad de generalización para el reconocimiento de distintos locutores de la red construida.

Vectores de entrada: la señal de voz es pregrabada en disco duro a una frecuencia de muestreo de 8 kHz, con una duración de 0.5 segundos

Condiciones de entrenamiento: las muestras de entrenamiento provienen de 6 locutores, 3 mujeres y tres hombres.

Rango de las salidas: valores de salidas en el rango de 0.99 a 1, se consideran éxitos. El valor de la salida para considerarse un fracaso y asegurar una buena discriminación tiene que ser menor a 0.

Palabras usadas en el entrenamiento: 12 muestras del uno,12 de dos.

Apéndice A.5: Glosario.

Automatización de sistemas telefónicos: marcación por voz, manejo de agendas, directorio público.

Codificación de voz: conversión de la señal analógica de voz en formato digital y viceversa, aplicando un factor de compresión que trata de reducir en mayor o menor mediada el número de bits necesarios.

Conversión texto a voz: conversión de un texto en formato electrónico a lenguaje hablado.

Dynamic Time Warping: técnica de alineamiento temporal a través de un algoritmo de programación dinámico.

Frame análisis: técnica de descomposición de la señal en una serie de pequeños segmentos de tiempo.

Habla conectada: pronunciación fluida de un mensaje utilizando un vocabulario muy restringido.

Habla continua: pronunciación de palabras de forma natural para un vocabulario amplio de palabras.

Modelos Ocultos de Markov: generalización de los algoritmos DTW mediante modelado de procesos estocráticos.

Palabras aisladas: supone la pronunciación de palabras con silencios entre ellas.

Parametrización: representación matemática mediante parámetros de una señal.

Procesamiento de señales: extraer la información relevante de la señal para la tarea a realizar.

Reconocimiento automático del habla: proceso que dota a las máquinas con la capacidad de recibir y comprender mensajes orales.

Reconocimiento en contexto: detección de la palabra a reconocer dentro del contexto de otras palabras.

Reconocimiento de locutores: proceso de identificación o verificación de la identidad del hablante de forma automática a partir de la señal de voz.

Reconocimiento de voz: identificación de las palabras y de las estructuras lingüísticas complejas que forman y componen el lenguaje hablado.

Redes Neuronales Artificiales: modelo artificial basado en la conexión de varios procesadores elementales (neuronas artificiales) que conjuntamente realizan una función común, simulando el desempeño de las neuronas naturales.

Respuesta vocal interactiva: difusión o captura de información por parte de un gran número de usuarios.

Tecnología del habla: aplicaciones tecnológicas basadas en el procesamiento automático del lenguaje hablado.

Anexo B.1: Hoja de información.

Información del estudiante

Nombre: Roberto Calvo Arias
Cédula: 3-346-182
Carné ITCR: 9512356
Dirección: 75 metros oeste Escuela Las Américas, Turrialba, Costa Rica.
Teléfono: 556 9389 / 3750861.

Información del proyecto

Nombre del proyecto: Reconocimiento de voz.
Profesor Asesor: Victorino Rojas.

Información de la empresa

Nombre: Instituto Costarricense de Electricidad
Zona: Sabana Norte.
Dirección: Sabana Norte.
Teléfono: 220 8279
Actividad principal: Energía y telecomunicaciones.