UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Ingeniería

Diseño de un sistema de compresión de señales para velocidades de transmisión menores a 9600

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A

Pablo Valle Martínez

Director : Dr Federico Kuhlmann

MEXICO, D. F. 1988

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

I N D I C E

R E S U M E N

Una de las áreas de investigación en el procesamiento digital de señales, es la que se refiere a señales de voz, de donde han surgido un gran número de aplicaciones, dentro de las cuales destaca la codificación eficiente de voz. EL objetivo de ésta, es reducir al mínimo la cantidad de información a transmitir o almacenar de una señal de voz, sin sacrificar calidad más allá de lo tolerable (puede variar de acuerdo con la aplicación).

Se desarrolló un sistema de compresión de señales de voz para velocidades de transmisión menores a 9600 bits por segundo, con el cual se logra una calidad aceptable a tasas de hasta 2400 bits por segundo.

Se describe el origen, desarrollo y aplicaciones del procesamiento digital de señales en diferentes áreas, así como la forma en que se produce la voz en un ser humano y los diferentes tipos de sonidos que se pueden generar.

Se definen algunos conceptos importantes para la realización del procesamiento digital, tales como muestreo y cuantización. Dentro de ésta se mencionan algunas de las técnicas más utilizadas; se explican algunas técnicas de compresión de señales de voz reportadas en la literatura y la forma de calcular algunas características importantes de las señales de voz como su correlación y espectro, por transformada de Fourier y por técnicas paramétricas.

Se presenta el esquema de compresión desarrollado, detallando cada uno de los algoritmos que se utilizan, las consideraciones particulares para la aplicación de ellos, además de explicar la forma como opera el sistema, las diferentes simulaciones realizadas y los resultados obtenidos.

Finalmente, se propone una arquitectura para la realización física del sistema de compresión, basada en un procesador de señales de la familia TMS320.

A G R A D E C I M I E N T O S

Deseo expresar mi sincero agradecimiento al Dr. Federico Kuhlmann Rodriguez por haber aceptado dirigir este trabajo, así como por sus valiosas criticas y comentarios.

Agradezco de manera especial al Dr. Andrés Buzo de la Peña por su asesoria e invaluables sugerencias, para el desarrollo del sistema presentado en este trabajo.

I. INTRODUCCION AL PROCESAMIENTO DIGITAL DE SEÑALES

1.1 EVOLUCION

El procesamiento digital de señales es un área de investigación que tiene sus orígenes en las matemáticas de los siglos XVII y XVIII, que al paso del tiempo se ha convertido en una herramienta muy importante para la ciencia y tecnología. Las técnicas y aplicaciones de este campo son tan antiguas como Newton y Gauss, o tan modernas como las computadoras digitales y los circuitos integrados.

El procesamiento digital de señales involucra, tanto a la representación de señales por medio de sucesiones de numeros o simbolos, como al procesamiento de estas sucesiones, entendiendo por procesamiento al hecho de efectuar algún tipo de operaciones sobre las componentes de estas sucesiones.

El propósito del procesamiento, puede ser estimar parametros

características de una señal o para transformarla a una forma más conveniente o deseable por la aplicación.

Las fórmulas clásicas de análisis numérico, como interpolación, integración y diferenciación, son algoritmos de procesamiento digital de señales. Por otro lado, la disponibilidad de computadoras de alta velocidad de operación ha impulsado el desarrollo de algoritmos más complejos y sofisticados para el procesamiento digital de señales y actualmente, gracias a los avances logrados en las tecnologías de circuitos integrados, se han podido realizar físicamente sistemas cuyo propósito principal es el procesamiento digital de señales.

La importancia del procesamiento de señales es evidente en áreas tan diversas como: comunicaciones, biomedicina, audio, sonar, radar, sismología, geofísica, y muchas más. En muchas aplicaciones, como por ejemplo, en análisis de electroencefalogramas, sistemas de transmisión y reconocimiento de voz, lo que se desea es extraer de la señal algunos parámetros característicos. Alternativamente, el objetivo puede ser eliminar interferencia en la señal o modificar la misma para presentarla en una forma que permita ser interpretada más fácilmente por un experto. Como otro ejemplo, una señal transmitida en un canal de comunicaciones es modificada generalmente en diferentes formas, que incluyen distorsiones de canal y ruido; uno de los objetivos del receptor es el compensar o eliminar estas modificaciones y en cualquier caso, se requiere procesar la señal.

El procesamiento de señales no está restringido a problemas

unidimensionales; por ejemplo, en procesamiento de imagenes se requiere la utilización de procesamiento en dos dimensiones.

Años atras, el procesamiento de señales se realizaba con equipo analógico. Algunas excepciones a esto fueron evidentes en los años cincuenta, particularmente en áreas donde se requeria procesar señales sofisticadas. Este fué el caso, por ejemplo, en el análisis de algunos datos geofisicos que podian ser grabados en cintas magnéticas para su posterior análisis en grandes computadoras digitales. Esta clase de problemas son de las primeras aplicaciones en que se utilizaron computadoras digitales para procesar señales.

Durante el mismo periodo, se incrementó rápidamente la utilización de computadoras digitales en este campo. Aprovechando la gran versatilidad que ofrecen, resultaba atractivo simular sistemas de procesamiento digital de señales en ellas, antes de instrumentar el sistema en forma analógica. Por lo tanto, un nuevo algoritmo de procesamiento, o un nuevo sistema, podia ser estudiado, analizado, diseñado y probado experimentalmente antes de ser construido. Un ejemplo de lo anterior, fue el codificador de voz realizado en los Laboratorios Bell. En la realización de un codificador de voz, las caracteristicas de un filtro frecuentemente afectan la calidad de la señal de voz generada en forma aleatoria. A través de simulaciones de computadora, las caracteristicas del filtro fueron ajustadas y se pudo evaluar la calidad del sistema antes de la construcción del equipo analógico.

En todos los ejemplos de procesamiento que utilizan computadoras

digitales mencionados hasta aqui, se observan las enormes ventajas que ofrecen en cuanto a flexibilidad. Sin embargo, el procesamiento generalmente no podia ser hecho en tiempo real, es decir, se almacenaba la señal a ser procesada, posteriormente se aplicaba el algoritmo y finalmente se analizaba. Asi pues, en ese tiempo las computadoras digitales se utilizaban para simular o aproximar un sistema de procesamiento de señales analógico, despues o durante su diseño, pero antes de su construccion.

A medida que fue aumentando el uso de computadoras digitales en esta novedosa area, surgió también la tendencia natural a experimentar con algoritmos más sofisticados para obtener mejores desempeños. Algunos de estos algoritmos, sobrepasaron el rango de aplicabilidad de las computadoras digitales y aparentemente no tenían instrumentación práctica en equipo analógico. Como consecuencia, muchos de estos algoritmos fueron catalogados como interesantes, pero imprácticos y poco aplicables. Un ejemplo de esta clase de algoritmos fue el conjunto de técnicas conocidas como análisis de cepstrum y filtrado homomorfico. La implantacion de estas técnicas requiere la evaluación de la transformada inversa de Fourier del logaritmo de la transformada de Fourier de la señal de entrada. La resolución y precisión requerida para la transformada de Fourier eran tales, que los analizadores de espectro analógicos no podian utilizarse. Sin embargo, se continuó investigando en este tipo de algoritmos con la esperanza de que en algún momento pudieran llegar a ser realizados fisicamente y/o aplicados (en la práctica) con alguna ventaja real en algún sentido.

Un cambio de rumbo en la evolución del procesamiento digital de señales fué provocado por el invento, en 1965, de una clase de algoritmos muy eficientes para el cálculo de la transformada de Fourier, conocidos como transformada rápida de Fourier (FFT : Fast Fourier Transform). Se puede decir que estos algoritmos generaron el gran desarrollo que actualmente tiene el procesamiento digital de señales, pues las ventajas que presentaron fueron muy importantes desde muchos puntos de vista. Estos algoritmos permiten reducir el tiempo de cálculo de una transformada de Fourier en órdenes de magnitud, facilitando la realización de algoritmos sofisticados con tiempos de proceso atractivos para la interacción con un sistema. Así, muchos de los algoritmos que hasta ese momento parecían ser imprácticos, comenzaron a ser utiles para su implantación en hardware digital de propósito particular.

Una caracteristica importante de los algoritmos de transformada rápida de Fourier, fue el hecho de ser un concepto que implica el uso de señales discretas en el tiempo. Fué pensado para el cálculo (aproximado en general) de la transformada de Fourier de una señal discreta en el tiempo e involucra el uso inteligente de una serie de propiedades y técnicas que son discretas en el dominio del tiempo. La importancia de ésto, radica en que generó una reformulación de muchos conceptos del procesamiento de señales y algoritmos, en términos de matemáticas de tiempo discreto y estas técnicas formaron una serie de relaciones exactas, en el dominio del tiempo discreto. [1]

Las técnicas y aplicaciones del procesamiento digital de señales están aumentando a gran velocidad, gracias a los avances en las

tecnologías de alta escala de integración, la reducción resultante en costo y tamaño de los componentes digitales, y al aumento en velocidad de los mismos. Procesadores de propósito especial para la realización de la transformada rápida de Fourier están disponibles comercialmente, se pueden encontrar filtros digitales en circuitos integrados; los procesadores digitales forman parte de los equipos modernos de radar y sonar.

El desarrollo de los procesadores digitales de propósito especial, permitió la creación de computadoras cuya arquitectura está especialmente diseñada para tal fin. Estas computadoras están siendo de gran utilidad en el procesamiento de señales en tiempo real, así como en simulaciones en tiempo real, para el posterior diseño del hardware apropiado.

En resumen, la importancia del procesamiento digital de señales va incrementándose al paso del tiempo, y al parecer va a provocar en algunas áreas de aplicación avances aún más revolucionarios que lo presenciado hasta el momento.

I.2 COMPARACION CON EL PROCESAMIENTO ANALOGICO.

Como se mencionó en la sección anterior, el procesamiento de señales durante mucho tiempo fue analógico y en algunas aplicaciones sigue siendo así. Sin embargo, cada vez es más evidente la transición de procesamiento analógico a digital. Por ejemplo, un tema muy conocido de

la electrónica analógica son los filtros, los cuales se pueden diseñar con elementos pasivos como resistencias y capacitores o con elementos activos como amplificadores operacionales. Actualmente, existen comercialmente circuitos integrados que contienen filtros digitales, con los cuales se pueden obtener caracteristicas de respuesta en frecuencia, cuya realización con elementos analógicos seria de una alta complejidad. Sin embargo, existen aún aplicaciones en las cuales el procesamiento analógico sigue siendo indispensable e indiscutible; por ejemplo, la conversión de una señal analógica a digital y viceversa o la adecuación de la señal a muestrear proveniente del transductor (amplificación, filtrado).

Por otro lado, hay algoritmos de procesamiento que no se han implantado analógicamente, como es el caso de la transformada de Fourier, y muchos más que se han desarrollado para ser utilizados digitalmente.

Un resumen de las ventajas que ofrece el procesamiento digital de señales, es el siguiente :

. Reproducibilidad. La inmunidad de circuitos digitales a pequeñas imperfecciones y efectos parásitos, implica que los circuitos pueden ser producidos con caracteristicas de operación consistentes sin la necesidad de ajustes finos.

. Programabilidad. Una arquitectura base puede ser utilizada para una gran variedad de aplicaciones, con solo cambiar el algoritmo o sus parámetros que se encuentran en memoria.

. Tiempo compartido. Un circuito diseñado para el procesamiento digital de señales puede ser utilizado para muchas señales, almacenando los resultados temporales de cada proceso en memoria RAM y procesando cada señal en un tiempo determinado.

. Comparaciones automáticas. Debido a que tanto la entrada como la salida de un circuito, para el procesamiento digital de señales, están ya cuantizadas, se pueden realizar comparaciones de los resultados directamente, con patrones discretos almacenados en memoria.

Versatilidad. El procesamiento digital de señales puede realizar funciones que son imposibles o imprácticas con equipo analógico.[2]

Algunas ventajas del manejo digital de señales :

. Permite el multiplexaje de señales. Este se refiere a la reducción de costos de transmisión, al enviar diferentes mensajes por un mismo canal. El multiplexaje de señales analógicas también se realiza fácilmente, pero la transmisión de una señal analógica es más vulnerable al ruido, distorsión e interferencia que la de una señal digital.

. En las señales digitales es fácil agregar información de control, la cual se puede insertar codificada en el mensaje, en el receptor se recibe el mensaje y se decodifica la información de control. Los sistemas digitales permiten que la información de control pueda ser incluida o extraida de un mensaje independientemente de la naturaleza del medio de transmisión, mientras que en el caso de señales analógicas el agregar información de control depende de la naturaleza del medio de transmisión y del equipo de recepción, lo que provoca en algunos casos que la información de control tenga que ser cambiada de formato para poder ser interpretada.

. El gran desarrollo que ha tenido la tecnología de circuitos integrados para aplicaciones digitales, permiten la sustitución de diferentes subsistemas analógicos por equivalentes digitales. El costo relativamente bajo y alta eficiencia de los circuitos digitales, facilita su utilización en aplicaciones que son prohibitivas, desde el punto de vista costo, para el empleo de componentes analogicos.

. Regeneracion de la señal. La representación de cualquier señal analógica en forma digital, involucra la conversión de la señal continua a una sucesión de muestras. Cada muestra es representada por algun número de digitos binarios. Si durante la transmisión de una señal digital, se agregan a la misma pequeñas cantidades de ruido, interferencia o distorsión, los datos binarios en el receptor pueden ser recuperados más fácilmente que si se se tratara de una señal analógica. Una caracteristica fundamental de los sistemas digitales es la posibilidad de reducir la probabilidad de errores en la transmisión, agregando etapas de regeneración de señal a lo largo de la linea de transmisión.

. Las señales digitales, pueden ser criptografiadas más fácilmente que las analogicas.

En contraste a las ventajas mencionadas del manejo digital de señales, ahora se mencionarán algunas desventajas:

. Incremento en el ancho de banda requerido para la transmisión. El aumento en ancho de banda, se genera cuando las muestras son codificadas de alguna forma y transmitidas con un pulso por bit de código.

. Conversión analógica-digital. Como se mencionó anteriormente, los sistemas digitales utilizados en la actualidad están instalados en

ambientes analógicos, por lo tanto, siempre que se desee trabajar digitalmente con una señal analógica ésta debe convertirse a digital. Esto lleva a incluir en el costo total de un sistema, el correspondiente a la etapa de conversión.

. Necesidad de sincronización. En cualquier caso, en el que se transmite información digitalmente de un lugar a otro, se requiere un reloj de referencia que controle la transmisión.

. Incompatibilidad con equipos analógicos existentes. Por ejemplo, si se quiere utilizar el sistema telefónico convencional, para realizar la transmisión de información digital, se requiere diseñar la interfaz entre el sistema digital y el teléfono. En muchas ocasiones, estas interfases representan un mayor costo que el sistema digital en si mismo. [2]

1.3 APLICACIONES

Como se ha mencionado, el gran desarrollo que ha tenido el procesamiento digital de señales ha generado un gran número de nuevas aplicaciones .

Las técnicas utilizadas para el procesamiento digital de señales, no son esclusivas de un área determinada, es decir, tanto es útil una técnica en el procesamiento de señales biomédicas como en el procesamiento de señales de voz. Las diferencias que existen en estas técnicas, para su aplicación en diferentes áreas, son debidas a las consideraciones y restricciones que en cada aplicación particular se requieren, como pueden ser: velocidad de transmisión, necesidad de

tiempo real.

El objetivo de esta seccion es mencionar algunas de las aplicaciones en las que se utiliza el procesamiento digital de señales, en areas como telecomunicaciones, audio, imágenes y voz. [3]

TELECOMUNICACIONES. En equipos de transmisión, las funciones de filtrado para selección de banda, en las que ahora se utilizan filtros analógicos, serán en parte realizadas con filtros digitales para multiplexajes tanto por division de tiempo como por división de frecuencia.

Otra aplicación, es la generación y detección de tonos. Por ejemplo, en los Estados Unidos, al marcar un numero telefónico, el aparato envia tonos codificados que representan al anterior. Esta técnica es conocida como DTMF (Double Tone Multi Frecuency). En la central telefónica se identifica este tono y se realiza la conexión con el destino. Tanto la generación como la detección se efectua utilizando técnicas digitales.

En la industria telefónica, el control de eco en circuitos de transmisión de larga distancia se puede llevar a cabo con tecnicas digitales.

Actualmente existen algunas impantaciones con procesadores digitales de señales que realizan las siguientes funciones :

. Generación de tonos con muy alta precisión

. Osciladores

. Decodificadores de tonos

. Sintetizadores de voz

. Generadores de sonidos

. Filtros

. Canceladores de eco.

AUDIO. El área del audio comprende todo el procesamiento (grabacion, almacenamiento, transmisión y reproducción) de señales perceptibles por el oido humano. En la practica, estas señales son fundamentalmente musica y voz.

Las grabaciones profesionales de música se realizan en múltiples segmentos, donde cada uno de estos contiene la contribución de un musico o grupo de músicos, los cuales posteriormente se combinan para generar la grabación final. Con un equipo convencional de grabación analógica la señal de cada aparato se degrada en cada reproducción, debido a las caracteristicas del material magnético utilizado en la fabricación de las cintas, las cabezas grabadoras y las imperfecciones del mecanismo que realiza el transporte de la cinta a velocidad constante; así, la grabación final incluye la suma de todo el ruido y distorsión generado en cada etapa de la edición y mezclado de los diferentes segmentos. Es aqui donde los equipos de grabacion digital estan encontrando gran aceptación. El proceso de grabación en estos equipos es básicamente el siguiente: cada canal de entrada es filtrado, muestreado y convertido a palabras digitales, los grupos de bits son multiplexados y se les agregan bits de control para correción de errores. Posteriormente, el conjunto resultante de bits se transforma a una señal analógica apropiada por medio de modulación, señal que se graba en la cinta.

Durante la edición de las diferentes grabaciones, se requieren realizar ciertas modificaciones a la señal o agregar efectos como

reverberación, eco, procesos de compresión o reducción de ruido, mismos que pueden ser efectuados con la ayuda de procesadores digitales de señales.

Una de las grandes ventajas de los sistemas digitales, es que el tipo de proceso realizado puede ser cambiado por otro, sin la necesidad de efectuar cambios en hardware, si la arquitectura del sistema es mas o menos general; cada proceso es simplemente otro programa que será almacenado en la memoria del sistema.

Algunas técnicas más sofisticadas de procesamiento digital de señales, se utilizan actualmente (no comercialmente); por ejemplo el procesamiento homomorfico. Esta técnica consiste en someter a una operación no lineal, señales que han sido convolucionadas o multiplicadas, para producir una suma de ellas. Esta representación aditiva, se procesa con filtros clásicos y se regresa al dominio original utilizando la función inversa de la operación no lineal empleada. Esta secuencia de operaciones es la base de un sistema compansor, que puede ser utilizado para separar el sonido de sus reflexiones. Es aplicable en la restauracion de viejas grabaciones, eliminación de eco, etc.

IMAGENES. Es un área en la que actualmente se estan realizando importantes avances. Las imagenes son ampliamente utilizadas en gran número de disciplinas cientificas como medicina, percepción remota, experimentacion fisica.

Una imagen tipica contiene una gran cantidad de información redundante. La transmisión de una imagen en forma digital, requiere de un canal cuyo ancho de banda es función del número de muestras que se

tomen de la imagen, el número de bits por muestra, el tiempo permitido para la transmisión y la potencia utilizada. Una aplicación consiste en usar esquemas de compresión para reducir la cantidad de información a transmitir o almacenar a partir de una imagen. Una de las técnicas más empleadas es la cuantización diferencial (DPCM).

También en esta área, se utilizan técnicas digitales para lograr la restauración de imágenes, para aumentar la calidad de una imágen, reconocimiento de imágenes o para detectar movimiento.

VOZ. La voz es el medio natural de comunicación entre personas, ha sido ampliamente estudiada en su forma y contenido.

Al igual que en todas las aplicaciones que se han mencionado, el desarrollo de los equipos digitales generó una serie de aplicaciones atractivas como :

. comunicación entre personas y máquinas utilizando voz

. comunicación efectiva entre humanos utilizando computadoras

En la aplicación de las técnicas de procesamiento digital de señales a voz, se consideran tres tópicos interesantes: La representación de las señales de voz en forma digital, la instrumentación de sofisticadas técnicas de procesamiento, y la serie de aplicaciones que recaen en el procesamiento digital.

Algunas de las aplicaciones más importantes del procesamiento digital de señales de voz son: [4]

. Transmisión y almacenamiento digital de voz.

El codificador de voz (vocoder). El objetivo del codificador de voz es reducir el ancho de banda requerido para transmitir una señal de

voz. La necesidad de reducir el ancho de banda existe actualmente, aun cuando están disponibles canales de transmisión de gran ancho de banda como en vía satélite, microondas y fibras opticas.

Inclusive, existe la tendencia a desarrollar sistemas que realicen la compresión de voz para su transmisión a bajas tasas con muy buena calidad, para optimar la utilización de un canal dado.

Esto abre la posibilidad de transmitir digitalmente señales de voz que han sido criptografiadas utilizando sofisticados algoritmos .

. Sistemas de sintesis de voz.

Los sistemas de sintesis de voz están pensados para lograr la comunicación entre las maquinas y las personas. Por ejemplo, un sistema automatico de información, utilizando una computadora, puede ser requerido por una persona que desea algún tipo de información y con solo oprimir una tecla del telefono, la computadora realizara la sintesis del mensaje a enviar al usuario después de ejecutar alguna acción. Los sistemas de sintesis de voz son aplicables también en el aprendizaje del habla, por ejemplo, para personas que no pueden hablar porque no oyen, por medio de sintesis de voz se muestran patrones que estas personas deben tratar de generar y de esta forma aprenden a hablar.

. Sistemas de identificación y verificacion voz.

Los sistemas de verificación de voz, deben decidir si la voz es de la persona que dice ser. Estos sistemas son aplicables en situaciones donde se requiere el control de la entrada en áreas restringidas o el acceso a información confidencial

. Sistemas de reconocimiento de voz.

El reconocimiento de voz, es una de las aplicaciones más importantes del procesamiento de señales de voz, pues permite la

comunicación con las maquinas. por ejemplo, una máquina de escribir operada a través de voz, es decir que uno puede ir dictando una carta y la maquina la va escribiendo; indicar a una computadora los comandos a ejecutar, controlar una silla de ruedas con solo mencionar la dirección hacia la cual se quiere desplazar, etc.

. Mejoramiento de la calidad de las señales de voz.

En muchas ocasiones, las señales de voz se degradan y pierden su eficiencia en la comunicación. En estos casos se pueden utilizar técnicas del procesamiento digital de señales para mejorar su calidad. Ejemplos de esto, son : eliminar ruido de la señal, eliminar reverberación, restauracion de voz grabada, etc.

II. CARACTERISTICAS DE LAS SEÑALES DE VOZ

II.1 ANATOMIA DE LA VOZ.

Para aplicar las tecnicas del procesamiento digital de señales a señales de voz, es conveniente entender la forma en que ésta se produce, ya que sin este conocimiento son pequeñas las tasas de compresión logradas.

Las señales de voz están compuestas por una sucesión de sonidos. Tanto estos sonidos como las transiciones entre ellos, sirven como una representación simbólica de información. La secuencia en que se producen estos sonidos está gobernada por las reglas del lenguaje. El estudio de estas reglas y sus implicaciones en la comunicación entre personas pertenece a la linguistica, y del estudio y clasificación de los sonidos de voz se encarga la fonética.

Debido al desarrollo de las teorias modernas de acustica, el

entendimiento del sistema humano de producción de voz esta aumentando cada vez mas. El principio fundamental de la generación de sonidos en el sistema vocal y el funcionamiento de filtro acustico del sistema vocal, están actualmente bien definidos. Sin embargo, algunas no linealidades en la vibracion de las cuerdas vocales y en la interacción entre la fuente de información y el sistema vocal aun son objeto de investigación.

La figura 2.1 es un esquema que muestra el sistema vocal humano. El sistema vocal es un tubo acustico de aproximadamente 17 cm de longitud. Está indicado en el esquema con la linea punteada: comienza en la glotis y termina en los labios. El sistema vocal consiste entonces, de la faringe (la conexión entre el esófago y la boca) y la boca. El área de una sección del sistema vocal, esta determinada por la posición de lengua, labios, mandibula y velum, y varia de cero (completamente cerrada) a cerca de 20 cm^2. El sistema nasal comienza en el velum y termina en los orificios nasales. Cuando el velum se baja, el sistema nasal se acopla acusticamente con el sistema vocal, produciendo los sonidos nasales de la voz. [5] y [6]

Los sonidos pueden ser generados en el sistema vocal de tres formas. Los sonoros se producen al elevar la presión de aire en los pulmones, forzando un flujo de aire a través del orificio de las cuerdas vocales (la glotis) provocando que las cuerdas vocales vibren. El flujo de aire interrumpido, produce pulsos cuasi-periódicos de gran ancho de banda los cuales excitan al sistema vocal. La estructura fisiológica de las cuerdas vocales, que son quienes producen este flujo de aire pulsante, se muestra en la figura 2.2. Los ligamentos que vibran

de las cuerdas vocales tienen una longitud de aproximadamente 18 mm y la abertura media es de 5 mm^2.

Los sonidos fricativos son generados formando una oclusión en algun punto del sistema y forzando al aire a pasar a través de ella con una velocidad lo suficientemente grande para provocar turbulencia. Se crea así una fuente de sonido a presión. Los sonidos explosivos resultan de realizar una oclusión completa seguida de una liberación abrupta de la presión generada.

Todas estas fuentes vocales - tanto sonoras como no sonoras - generan señales con gran ancho de banda. El sistema vocal, actúa como un filtro que impone sus caracteristicas resonantes a la fuente.

El sistema vocal y el sistema nasal se muestran en la figura 2.3 como tubos de área en sección transversal no uniforme. Como el sonido, generado como se menciono anteriormente, se propaga a traves de estos tubos, el espectro de frecuencia es recortado por la respuesta en frecuencia del tubo. Este efecto es muy similar al observado en los instrumentos de viento. En el contexto de producción de voz, las frecuencias de resonancia del tubo del sistema vocal se llaman frecuencias formantes o simplemente formantes. Las frecuencias formantes dependen de la forma y dimensión del tubo del sistema vocal; cada forma está caracterizada por un conjunto de frecuencias formantes. Los diferentes sonidos se producen al variar la forma del sistema vocal. De esta manera, las propiedades espectrales de las señales de voz varian con el tiempo así como varia la forma del sistema vocal, figura 2.4.

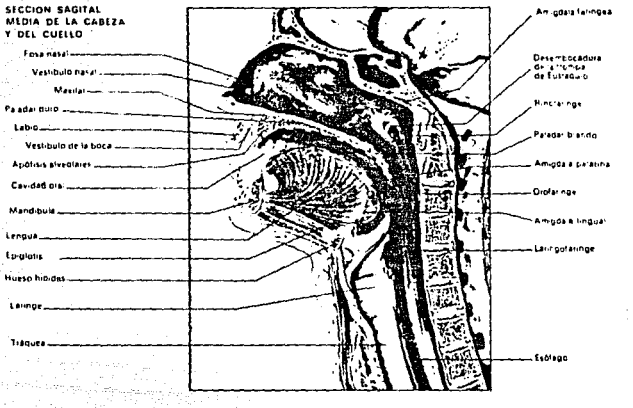


Figura 2.1 Esquema del sistema vocal.

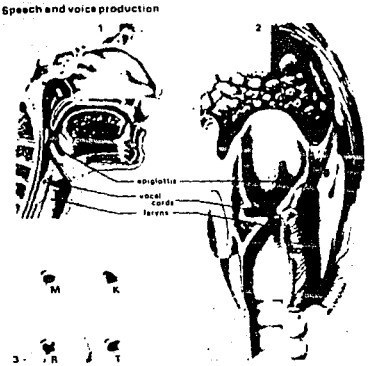


Figura 2.2 Esquema de las cuerdas vocales

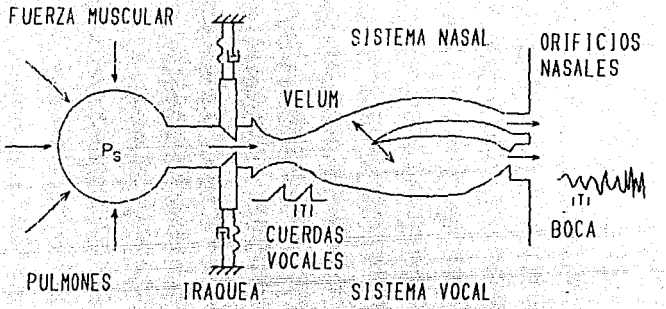


Figura 2.3 Diagrama de un modelo del sistema vocal.[5]

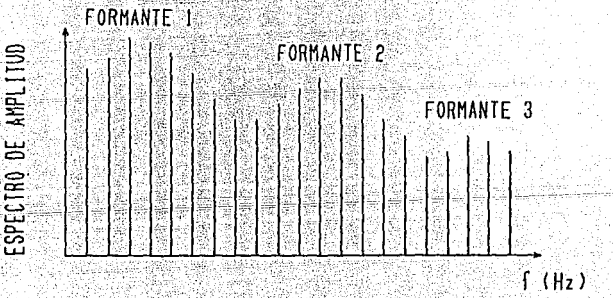


Figura 2.4 Espectro de una señal de voz.

II.2 CLASIFICACION DE LOS SONIDOS DE VOZ

La mayoria de los lenguajes pueden ser descritos en terminos de un conjunto de sonidos distintivos o fonemas. Cada fonema puede ser clasificado como sonido continuo o no continuo. Los sonidos continuos son producidos por una configuración del sistema vocal constante (no varia su forma con el tiempo), excitado por una fuente apropiada. Existen sonidos que son producidos cambiando la configuracion del sistema vocal; estos son clasificados como sonidos no continuos.

Se mencionarán algunas de las caracteristicas acusticas de diferentes sonidos, incluyendo el lugar y la forma como se articulan. [4]

VOCALES. Los sonidos vocales son producidos excitando una configuración fija del sistema vocal con pulsos cuasi-periódicos de aire, debidos a la vibración de las cuerdas vocales. La forma en la cual el area de la sección transversal varia a lo largo del sistema vocal, determina la frecuencia resonante del sistema (formante) y asi del sonido que se produce. La relación entre el área de sección transversal y la longitud del sistema vocal se llama función de área del sistema vocal. La función de área de un sonido vocal en particular, está determinada por las posiciones de la lengua, mandibula, labios y velum. Asi, cada sonido vocal puede ser caracterizado por una configuración del sistema vocal (funcion de area) que se utiliza para su producción.

DIPTONGOS. Aunque existe algo de ambiguedad y desacuerdo entre lo

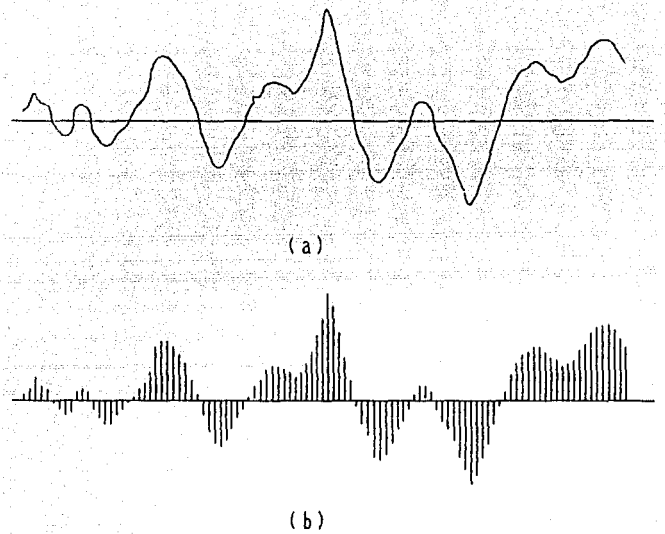


Figura 3.1 Señal de voz . (a) original (b) muestreada

entonces

$x_a(t)$ puede ser reconstruida únicamente por muestras igualmente espaciadas $x_a(nT)$, $-\infty < n < \infty$, si $1/T > 2F_N$.

El teorema anterior viene del hecho de que si la transformada de Fourier de la señal $x_a(t)$ se define como :

$$X_a(j\Omega) = \int_{-\infty}^{\infty} x_a(t)\, e^{-j\Omega t}\, dt \qquad (3.2)$$

y la transformada de Fourier de la secuencia x(n) está definida por :

$$X(e^{j\omega}) = \sum_{-\infty}^{\infty} x(n)\, e^{-j\omega n} \qquad (3.3)$$

entonces, si $X(e^{j\omega})$ se evalúa para frecuencias $\omega = \Omega T$, entonces $X(e^{j\Omega T})$ se relaciona con $X_a(j\Omega)$ [4] por

$$X(e^{j\Omega T}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} x_a\left(j\Omega + j\,\frac{2\pi}{T} k \right) \qquad (3.4)$$

Para observar las implicaciones que tiene la ecuación 3.4, consideremos que $X_a(j\Omega)$ es de la forma que se muestra en la figura 3.2; considerando que $X_a(j\Omega) = 0$ para $|\Omega| > \Omega_N = 2\pi F_N$. La frecuencia F_N es conocida como la frecuencia de Nyquist. De acuerdo con la ecuación 3.4, $X(e^{j\Omega T})$ es la suma de un número infinito de réplicas de $X_a(j\Omega)$, cada una centrada en multiplos enteros de $2\pi/T$. En la figura 3.3 se muestra cuando $1/T > 2F_N$, así que las imágenes de la transformada de Fourier no se translapan en la banda base $|\Omega| < 2\pi F_N$.

Por otro lado, en la figura 3.4, se muestra cuando $1/T < 2F_N$. En este caso, la imágen centrada en $2\pi/T$ se translapa con la de banda base.

Bajo la condición de que $1/T > 2F_N$, es claro que la transformada de Fourier de la secuencia de muestras es proporcional a la transformada de Fourier de la señal analógica en banda base :

$$X(e^{j\Omega T}) = \frac{1}{T}\, X_a(j\Omega) \qquad |\Omega| < \frac{\pi}{T} \qquad (3.5)$$

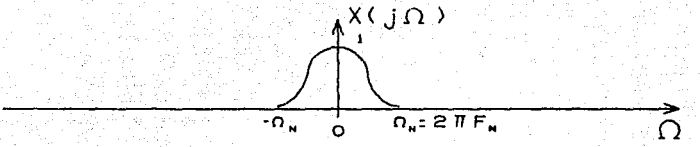


Figura 3.2 Espectro de una señal.

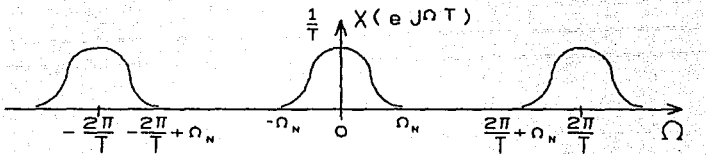


Figura 3.3 Espectro resultado de muestrear con f > 2FN

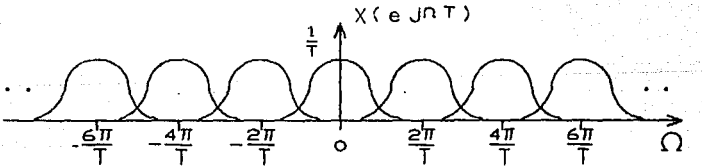


Figura 3.4 Espectro resultado de muestrear a f < 2FN

Utilizando este resultado, se puede demostrar [4], que la señal original puede obtenerse a partir de la secuencia de muestras, mediante la fórmula de interpolación :

$$x_a(t) = \sum_{n=-\infty}^{\infty} x_a(nT) \left[\frac{\text{sen}\ [\ \pi(t-nT)/T\ ]}{\pi(t-nT)/T} \right] \qquad (3.6)$$

Así, dado un conjunto de muestras de una señal analógica con ancho de banda limitado, que fueron tomadas a una frecuencia por lo menos igual a la frecuencia de Nyquist, es posible reconstruir la señal analógica original, empleando la ecuación 3.6. Los convertidores digital-analógico buscan aproximar ésta.

En la mayoría de las aplicaciones de codificación de voz, la frecuencia de muestreo está determinada por las características de la red telefónica. El ancho de banda de la red telefónica está entre 300 y 3300 Hz, por lo tanto, la frecuencia de muestreo que se utiliza generalmente es de 8 kHz (8000 muestras/segundo).Figura 3.5.

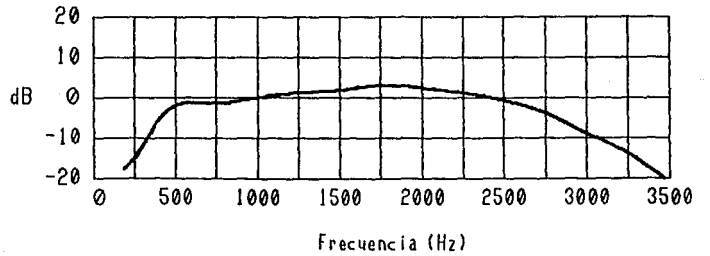


Figura 3.5 Respuesta en frecuencia de un canal telefónico

Una consideración importante, es que aun cuando la señal a ser muestreada tenga ancho de banda limitado, a ésta se le puede agregar ruido, cuyo ancho de banda es muy grande, en estos casos, debe filtrarse la suma de la señal con el ruido, de tal forma que no se produzca el efecto de traslape al muestrear la señal.

El filtrado, el muestreo y la cuantización (figura 3.6), se agrupan bajo el nombre de conversión analógico-digital.

Cuando la señal digital ha sido procesada, puede convertirse a señal analógica por medio de la conversión digital-analógico (figura 3.7), la cual consiste de un circuito muestreador-retenedor y de un filtro paso-bajas.

Figura 3.6 Diagrama de bloques de un convertidor A/D

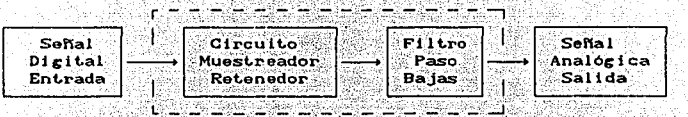


Figura 3.7 Diagrama de bloques de un convertidor D/A

III.2 ESTADISTICA DE LAS SEÑALES DE VOZ.

Para representar digitalmente una señal de voz es necesario considerarla como un proceso aleatorio. Aceptando lo anterior, si la señal $x_a(t)$ es una función de muestras de un proceso aleatorio continuo en el tiempo, entonces la secuencia de muestras originadas por un muestreo periódico puede pensarse como una secuencia de muestras discretas en el tiempo de un proceso aleatorio.

Comunmente, en el análisis de sistemas de comunicaciones, se utiliza la función densidad de probabilidad de primer orden $p(x)$ para caracterizar una señal analógica, así como la función de autocorrelación del proceso aleatorio, la cual se define por :

$$\phi_a(\tau) = E [x_a(t)\, x_a(t+\tau)] \qquad (3.7)$$

donde $E [x]$ denota la esperanza matemática de x . El espectro de potencia es la transformada de Fourier de $\phi_a(t)$:

$$\Phi_a(\Omega) = \int_{-\infty}^{\infty} \phi_a(\tau)\, e^{-j\Omega\tau}\, d\tau \qquad (3.8)$$

La señal discreta en tiempo, obtenida al muestrear la señal aleatoria $x_a(t)$ tiene una función de autocorrelación

$$\phi(m) = E [x(n)\, x(n+m)]$$

$$= E [x_a(nT)\, x_a(nT+mT)] = \phi_a(mT) \qquad (3.9)$$

Así, debido a que $\phi(m)$ es la versión muestreada de $\phi_a(t)$, entonces el espectro de potencia de $\phi(m)$ está dado por

$$\Phi\ (e^{j\Omega T}) = \sum_{m=-\infty}^{\infty} \phi(m)\ e^{-j\Omega T m}$$

$$= \frac{1}{T} \sum_{k=-\infty}^{\infty} \Phi a\ (\Omega + \frac{2\pi}{T}\ k\ ) \qquad (3.10)$$

La ecuación 3.10 muestra que para el proceso aleatorio que modela la voz, el espectro de potencia de la señal discretizada es una versión suavizada del espectro de potencia de la señal original.

La función densidad de probabilidad de las amplitudes x(n), es la misma que la de las amplitudes $x_a(t)$, puesto que $x(n) = x_a(nT)$. Así pues, la media y la variancia son las mismas tanto para las muestras como para la señal original.

Para aplicar modelos estadísticos a señales de voz, es necesario estimar la densidad de probabilidad y la función de correlación de las ondas de voz. La densidad de probabilidad se estima determinando un histograma de amplitudes para un gran número de muestras; se ha demostrado que una buena aproximación es la distribución gama de la forma [7]

$$p(x) = \left[\frac{\sqrt{3}}{8\pi\ \sigma_x |x|} \right]^{1/2} e^{-\frac{\sqrt{3}\ |x|}{2\ \sigma_x}} \qquad (3.11)$$

Una aproximación más simple, es utilizar una densidad Laplace

$$p(x) = \frac{1}{\sqrt{2}\ \sigma_x}\ e^{-\frac{\sqrt{2}\ |x|}{\sigma_x}} \qquad (3.12)$$

En la figura 3.8 se muestra una comparación entre las densidades de probabilidad gama, Laplace y de una señal de voz; normalizadas de tal

forma que la media es cero y la variancia (σ_x^2) es igual a uno. Se observa que la densidad de probabilidad gama es una mejor aproximación que la de Laplace.

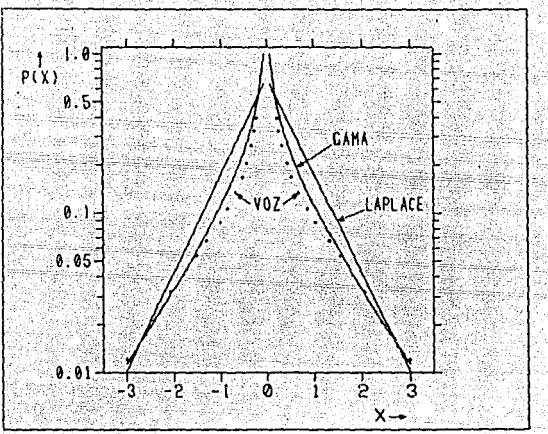


Figura 3.8 Comparación entre las densidades de probabilidad y la probabilidad de una señal de voz.

III.3. CUANTIZACION

Una vez que la señal de interés ha sido filtrada y muestreada a una tasa adecuada, (x(n)), para transmitir esta secuencia de muestras por un canal digital de comunicación, almacenarlas en una memoria digital o

usarlas como entrada a un algoritmo de procesamiento digital de señales, los valores de las muestras deben ser cuantizados a un conjunto finito de amplitudes de tal forma que puedan ser representados por un número finito de símbolos. Este proceso se muestra en la figura 3.9. Es útil separar el proceso de representar las muestras $\{x(n)\}$, por un conjunto de símbolos, $\{c(n)\}$, en dos etapas: la etapa de cuantización la cual produce la secuencia de amplitudes cuantizadas $\{\hat{x}(n)\} = \{Q[x(n)]\}$ y una etapa de codificación la cual representa a cada amplitud cuantizada por una palabra de código, $c(n)$ (figura 3.9a). En el receptor (figura 3.9b), el decodificador toma una secuencia de palabras de código $c'(n)$, y las transforma a una secuencia de muestras cuantizadas $\{\hat{x}'(n)\}$. Si $c'(n)=c(n)$, no hubo errores en la transmisión y $\hat{x}(n)=\hat{x}'(n)$.

Es conveniente utilizar números binarios para representar las muestras cuantizadas. Con B bits es posible representar 2^B niveles de cuantización. La capacidad de información requerida para transmitir o almacenar la representación digital es

$$I = B \cdot F_s \text{ [bits/segundo]} \qquad (3.13)$$

donde F_s es la frecuencia de muestreo (muestras/segundo) y B es el número de bits/muestra. Es importante que la capacidad de información sea lo menor posible, manteniendo un nivel de calidad aceptable. Para un ancho de banda de voz dado, el valor mínimo de frecuencia de muestreo queda establecido por el teorema del muestreo, así pues, la única forma de reducir la cantidad de información es reducir el número de bits por muestra. [4]

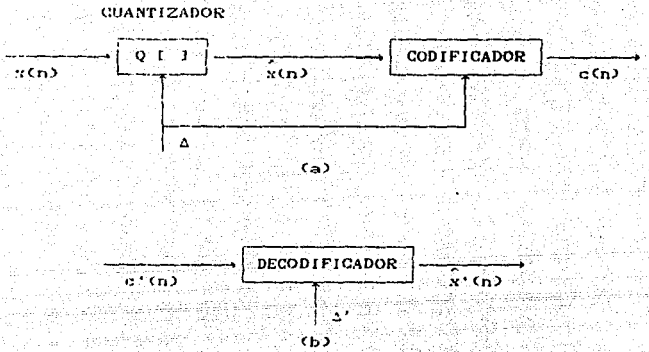


Figura 3.9 Proceso de cuantización y codificación. (a) Codificador (b) decodificador.

En general, es razonable considerar que el conjunto de muestras {x(n)} pertenece a un conjunto finito de amplitudes tal que

$$|x(n)| \leq X_{max} \qquad (3.14)$$

Si consideramos que una señal de voz tiene una función densidad de probabilidad de Laplace, se puede demostrar [4], que solo el 0.35% de las muestras de la señal de voz están fuera del intervalo

$$-4\sigma_x \leq x(n) \leq 4\sigma_x \qquad (3.15)$$

Así, es conveniente asumir que el intervalo pico a pico de la señal de voz es proporcional a su desviación estandar.

Las amplitudes de las muestras son cuantizadas dividiendo el

intervalo total de amplitudes en un conjunto finito de intervalos, asignando la misma amplitud a todas las muestras que caen en un mismo intervalo.

En el diseño de un cuantizador se debe tomar en cuenta lo siguiente: (1) Con un número fijo de niveles de cuantización se debe buscar minimizar el ruido de cuantización. Este ruido, o distorsión, se produce al representar todo un intervalo de amplitudes con un solo valor.(2) Con un ruido de cuantización fijo, se debe buscar minimizar el numero promedio de bits por muestra tal que cumpla con la distorsión especificada.

Básicamente existen tres formas de realizar la cuantización : [9]

1. Cuantizacion escalar.
2. Cuantización por bloques.
3. Cuantización secuencial.

La escalar se refiere a la cuantización de muestra por muestra, mientras que la cuantización por bloques es la representación de un bloque de muestras con un bloque de valores seleccionados de un conjunto discreto de bloques, y finalmente, la secuencial es la cuantización de una secuencia de muestras utilizando información de las muestras vecinas ya sea en bloques o escalarmente.

A continuación se mencionarán algunos ejemplos de sistemas de cuantización tanto de tipo escalar como de bloque.

III.3.1 CUANTIZACION UNIFORME

Los intervalos y niveles de cuantización pueden seleccionarse de diferentes maneras dependiendo de la aplicación. Cuando la representación obtenida será procesada por un sistema digital, los niveles e intervalos de cuantización se distribuyen generalmente de manera uniforme. Para definir un cuantizador uniforme, se tiene que

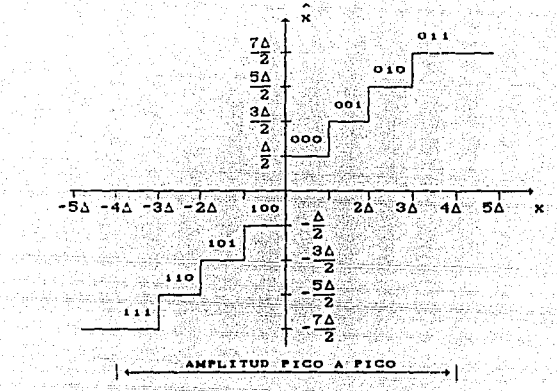
$$x_i - x_{i-1} = \Delta \qquad (3.16)$$

$$\hat{x}_i - \hat{x}_{i-1} = \Delta \qquad (3.17)$$

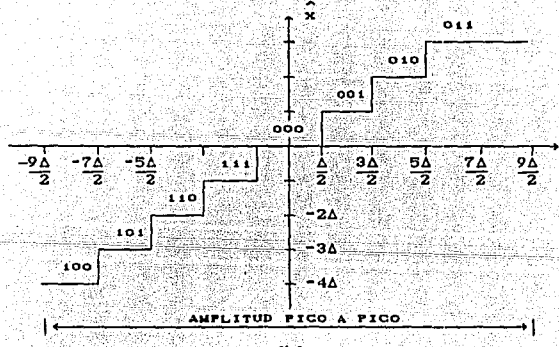
donde Δ es el paso de cuantización. Dos tipos de cuantizadores uniformes se muestran en la figura 3.10, para 8 niveles de cuantización.

En caso de que el número de niveles seleccionado sea una potencia de 2, puede observarse que el cuantizador por truncamiento tiene el mismo número de niveles positivos y negativos, posicionados simétricamente respecto al origen; mientras que en el cuantizador por redondeo hay un nivel negativo más que el número de niveles positivos, aunque en este caso se dispone de un nivel cero, cosa que no ocurre en el cuantizador por truncamiento.

Para el diseño de cuantizadores uniformes existen dos parámetros : el número de niveles y el paso de cuantización Δ. El número de niveles generalmente se escoge de la forma 2^B . Δ y B deben seleccionarse de tal forma que cubran todo el intervalo dinámico de la señal, esto es, si consideramos que la señal tiene una función densidad de probabilidad



(a)



(b)

Figura 3.10 Dos tipos de cuantizadores uniformes : (a) truncamiento (b) redondeo

simétrica, entonces

$$2\ X_{max} = \Delta\ 2^{B} \qquad (3.18)$$

Una forma de comparar los sistemas de cuantización es a través de la relación señal/ruido (SNR). Se define como la relación entre la energia de la señal con respecto a la energia del ruido, está dada por

$$SNR = \frac{\sum_{n} x^{2}(n)}{\sum_{n} e^{2}(n)} \qquad (3.19)$$

donde la suma es sobre todas las muestras de la señal. La variable e(n) es el error de cuantización que se define como

$$e(n) = \hat{x}(n) - x(n) \qquad (3.20)$$

Se puede demostrar [4] que la relación señal a ruido de un cuantizador uniforme, expresado en dB está dada por

$$SNR(dB) = 6B + 4.77 - 20 \log \left[\frac{X_{max}}{\sigma_x}\right] \qquad (3.21)$$

Evaluando la expresión anterior con 3.15 se obtiene

$$SNR(db) = 6B - 7.2 \qquad (3.22)$$

En otras palabras, por cada bit que se agrega, lo cual duplica el número de niveles de cuantización, se obtiene un mejor desempeño en 6 dB.

La condición para la obtención de la ecuación 3.22, es que el valor máximo del cuantizador sea $X_{max} = 4\sigma_x$, donde σ_x es la variancia de la señal, que puede calcularse a partir de la siguiente ecuación

$$\sigma_x^2 = \frac{1}{N} \sum_{n=1}^{N} \left[x(n) - \bar{x} \right]^2 \qquad (3.23)$$

Es deseable tener un cuantizador cuya relación señal a ruido sea independiente del nivel de la señal, para tener un porcentaje de error constante. Esto se puede lograr utilizando niveles de cuantización distribuidos no uniformemente. [4] y [8]

III.3.2 CUANTIZACION NO UNIFORME

Un problema de utilizar cuantización uniforme en una señal de voz, es que ésta cambia con el tiempo, y su variancia σ_x^2 puede ser muy diferente de un segmento a otro. Si el cuantizador está diseñado para manejar señales de gran magnitud con σ_x^2 grande, el paso de cuantización será grande y las señales de pequeña amplitud tendrán grandes errores de cuantización. Para resolver este problema, se diseña un cuantizador cuyo paso de cuantización se incrementa de la misma forma que la amplitud de la señal, ahora el cuantizador uniforme opera sobre el logaritmo de la señal de voz, la cual es una versión comprimida de la señal original. La señal se reconstruye en el receptor expandiendo la señal recibida. Este proceso de compresión-expansión se conoce como companSión (COMpresión-exPANSION) .

Existen dos métodos para realizar la companSión: la ley μ y la ley A. La primera es la que se utiliza en América y la segunda en Europa. En la figura 3.11 se muestra un diagrama de bloques del sistema de codificación utilizando companSión.

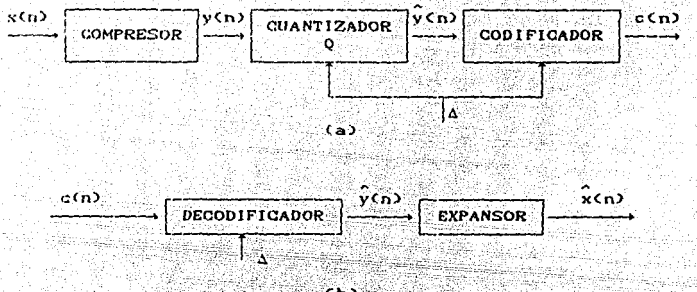


Figura 3.11 Cuantización utilizando compansión . (a) Transmisor (b) Receptor

Para la cuantización utilizando ley μ se tiene la siguiente ecuacion :

$$y(n) = X_{max} \frac{\log\left(1 + \mu \frac{|x(n)|}{X_{max}}\right)}{\log(1+\mu)} \operatorname{sgn}[x(n)] \qquad (3.24)$$

donde

X_{max} es el valor maximo de la señal

$\operatorname{sgn}[x(n)] = \pm 1$ cuando $x(n)$ es positivo o negativo

$|x(n)|$ es el valor absoluto de $x(n)$

μ es un parámetro que determina el nivel de compresión.

Se ha encontrado que μ=255 da buenos resultados para la compansión de señales de voz. El expansor debe realizar la función inversa de la ecuacion 3.24.

En lugar de aplicar las ecuaciones de compresión y expansión a cada

muestra, es recomendable utilizarlas una vez para calcular las diferentes fronteras de cuantizacion y los diferentes niveles de cuantización, generando así el cuantizador no lineal deseado. En la figura 3.12 se muestra una aproximación de la curva característica de la transformación ley μ.

Utilizando el mismo tipo de consideraciones hechas en el análisis de cuantizadores uniformes, en [4] se menciona una fórmula para calcular la relación señal a ruido de un cuantizador ley μ

$$SNR(dB) = 6B + 4.77 - 20 \log [\ln(1+\mu)] - 10 \log \left[1+\left(\frac{X_{max}}{\mu\sigma_x}\right)^2 + \sqrt{2}\left(\frac{X_{max}}{\mu\sigma_x}\right)\right] \qquad (3.25)$$

Comparando la expresión anterior con la ecuación 3.21 se observa una dependencia menos severa de la relación señal a ruido con el valor (X_{max}/σ_x). Puede observarse que mientras μ aumenta, la relación señal a ruido disminuye y es menos sensible a cambios en (X_{max}/σ_x).

La cuantización utilizando ley A se define de forma diferente, pero el resultado es muy similar al obtenido con la ley μ. Un compresor por ley A se define por

$$y(n) = \frac{A\,x(n)}{1 + \log(A)} \qquad \text{para } 0 \le |x(n)| \le \frac{X_{max}}{A}$$

$$y(n) = \frac{1 + \log\left[\frac{A|x(n)|}{X_{max}}\right]}{1 + \log(A)} \operatorname{sgn}[x(n)] \qquad \text{para } \frac{X_{max}}{A} < |x(n)| \le X_{max} \qquad (3.26)$$

El valor de A que se utiliza en los sistemas europeos de comunicaciones es de 87.56. En la figura 3.13 se muestra una aproximacion de la curva caracteristica de la ley A. [4] y [8]

1	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	1
1	1	0	0	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1

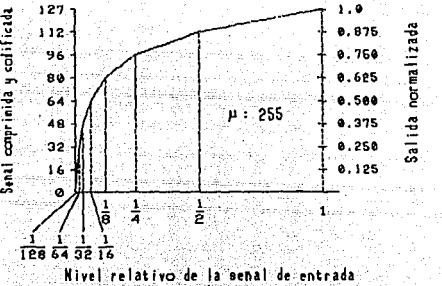


Figura 3.12 Caracteristica ley μ

1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0
1	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0

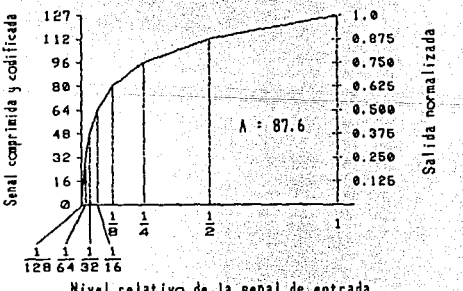


Figura 3.13 Caracteristica ley A

La cuantización logaritmica de la voz, ha demostrado ser muy efectiva, el asignar 8 bits/muestra con este metodo, equivale a utilizar un cuantizador uniforme de 13 bits/muestra. La cuantización logaritmica se encuentra implantada en circuitos integrados llamados CODEC. [8]

III.3.3 CUANTIZACION ADAPTABLE.

La amplitud de la señal de voz, como se ha mencionado, puede variar en un gran intervalo dependiendo de quien habla, del ambiente donde se desarrolla la comunicación y de la expresion utilizada. Una aproximación para considerar estas fluctuaciones en la amplitud de la señal de voz, es utilizar un cuantizador no uniforme. Otra aproximación es adaptar las caracteristicas del cuantizador al nivel de la señal de entrada, esto es lo que se conoce como cuantización adaptable; cuando se aplica directamente sobre las muestras se le conoce como PCM adaptable o simplemente APCM.

Una clasificación de los cuantizadores adaptables es en base al lugar en donde se efectúa la adaptación, si ésta se realiza de la señal de entrada al codificador se conoce como adaptación por avance, si se realiza de la señal de salida del codificador al canal se conoce como adaptación por realimentación. [4]

ADAPTACION POR AVANCE. En la figura 3.14 se muestra un sistema que opera con este tipo de adaptación. En este caso, el tamaño del paso de cuantización en un instante n está dado por

$$\Delta(n) = \Delta \quad \sigma(n) \qquad (3.27)$$

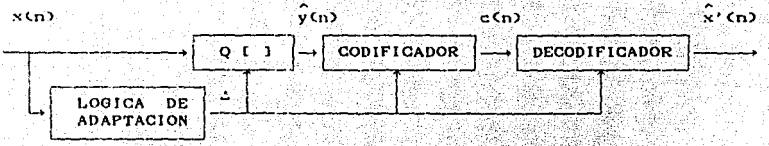


Figura 3.14 Sistema de cuantización adaptable por avance

donde $\sigma^2(n)$ es la variancia de la señal. y Δ es el valor seleccionado de paso de cuantización por unidad de variancia. La variancia de la señal puede calcularse del segmento de señal que será cuantizado

$$\sigma^2(n) = \frac{1}{N} \sum_{k=n}^{n+N-1} x^2(k) \qquad (3.28)$$

Puede calcularse la variancia también de una forma recursiva utilizando

$$\sigma^2(n) = a\ \sigma^2(n-1) + x^2(n-1) \qquad (3.29)$$

donde $0 < a < 1$. Mientras menor es el valor de a, el cuantizador puede registrar mejor los cambios en la señal. Un valor típico de a es 0.9; el valor de $\Delta(n)$ usualmente está restringido al intervalo $\Delta_{min} \leq \Delta(n) \leq \Delta_{max}$. La relación $\Delta_{max}/\Delta_{min}$ determina el intervalo dinámico del sistema.

Para lograr una reconstrucción precisa de la señal, la lógica de decodificación adaptable debe ser similar a la del codificador. Debido a que el decodificador no tiene forma de realizar la

adaptación a partir de los valores cuantizados, es necesario transmitir información acerca de $\Delta(n)$. Esto incrementa la velocidad de transmisión.

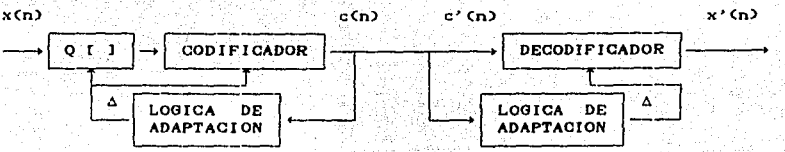


Figura 3.15 Sistema de cuantización adaptable por realimentación.

ADAPTACION POR REALIMENTACION. La figura 3.15 muestra un sistema de cuantización adaptable por realimentación, en la cual se puede observar que la variancia de la señal de entrada se calcula a partir de la señal ya codificada. De esta manera tiene la ventaja de que no requiere transmitir información adicional para la decodificación de la señal. En este caso el paso de cuantización se adapta de acuerdo con

$$\Delta(n) = P\ \Delta(n-1) \qquad (3.30)$$

El valor de P depende de la magnitud de la palabra de codigo anterior $|c(n-1)|$. Una tabla para valores típicos de P se encuentra en [3]. La razón de esta multiplicación es que, para pequeños valores de $c(n-1)$, la señal es pequeña y se utiliza $P < 1$ para reducir el paso de cuantización, mientras que para valores grandes de $c(n-1)$, $P > 1$ ya que la señal es grande y hay peligro de que el cuantizador la recorte.

III.3.4. CUANTIZACION DIFERENCIAL.

Examinando las señales de voz, especialmente en segmentos sonoros, se observa que existe un cambio suave entre muestras consecutivas. La diferencia entre muestras adyacentes debe tener una variancia pequeña y un intervalo dinámico menor que las muestras de voz en si mismas. Aprovechando ésto, en un sistema de cuantización diferencial (figura 3.16), la señal de entrada al cuantizador es [4],[8] y [10]

$$d(n) = x(n) - \tilde{x}(n) \qquad (3.31)$$

la diferencia de la señal de entrada, x(n), con una estimación, o predicción de la muestra de entrada, denotada por $\tilde{x}(n)$.

Este valor predicho es la salida de un sistema predictor P, cuya entrada es una versión cuantizada de la señal de entrada x(n). La señal d(n) es la señal que será cuantizada y la cuantización puede ser fija o adaptable, uniforme o no uniforme, pero en cualquier caso los parámetros deben ajustarse a la señal d(n).

La señal de diferencia cuantizada puede representarse como

$$\hat{d}(n) = d(n) + e(n) \qquad (3.32)$$

donde e(n) es el error de cuantización. Por otro lado

$$\hat{x}(n) = \tilde{x}(n) + \hat{d}(n) \qquad (3.33)$$

sustituyendo 3.31 y 3.32 en 3.33 se tiene que

$$\hat{x}(n) = x(n) + e(n) \qquad (3.34)$$

Esto es, independientemente de las propiedades del sistema P, la muestra de voz cuantizada difiere de la entrada por el error de cuantización de

la señal de diferencia. Si la predicción es buena, la variancia de d(n) será menor que la de x(n) de tal forma que el cuantizador con un numero de niveles dado, puede ser ajustado para dar un menor error de cuantización, comparado con el error mínimo de cuantización que se podría obtener al cuantizar la señal directamente.

En la figura 3.16 se muestra un diagrama de bloques de un sistema de cuantización diferencial.

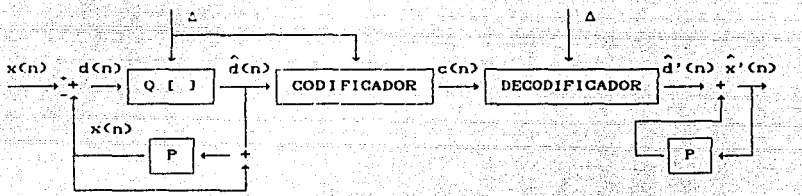


Figura 3.16 Sistema de cuantización diferencial

La relación señal a ruido del sistema de la figura 3.16 se define como

$$SNR = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\sigma_x^2}{\sigma_d^2}\frac{\sigma_d^2}{\sigma_e^2} = G_p\, SNR_Q \qquad (3.35)$$

La cantidad SNR_Q depende del cuantizador que se utilice, y conociendo las propiedades de d(n), SNR_Q puede maximizarse utilizando cualquiera de las técnicas descritas anteriormente. La cantidad G_p, si es mayor que uno, representa la ganancia en la relación señal a ruido debida al esquema diferencial; el objetivo es ahora maximizar G_p haciendo una selección adecuada del sistema P. Para una señal dada, σ_x^2

es una cantidad fija de tal forma que G_p puede maximizarse, minimizando σ_d^2.

Una opción para el sistema predictor P, es utilizar una combinación lineal de los valores cuantizados anteriormente, esto es

$$\tilde{x}(n) = \sum_{k=1}^{p} a_k \hat{x}(n-k) \qquad (3.36)$$

donde p es el orden del sistema predictor.

Se puede utilizar la cuantización diferencial junto con otros tipos de cuantización de los que se han mencionado, como por ejemplo : la combinación de la cuantización diferencial con la cuantización adaptable recibe el nombre de cuantización adaptable diferencial (ADPCM), dentro de la cual hay dos tipos , la que utiliza adaptación por avance y la que utiliza adaptación por realimentación, para lograr un mejor desempeño.

III.3.5 MODULACION DELTA

Es un caso especial de la cuantización diferencial. En esta clase de sistemas, la frecuencia de muestreo se selecciona de tal forma que sea varias veces mayor que la fecuencia de Nyquist de la señal original. Como resultado, las muestras adyacentes están muy correlacionadas.

III.3.5.1 MODULACION DELTA LINEAL

En la figura 3.17 se muestra un diagrama del sistema de modulación delta. En este caso, el cuantizador sólo tiene dos niveles y el paso de cuantización es fijo. El nivel positivo de cuantización se representa por c(n)=0 y el negativo por c(n)=1. Así [8]

$$\hat{d}(n) = \Delta \quad \text{si } c(n) = 0$$
$$= -\Delta \quad \text{si } c(n) = 1 \qquad (3.37)$$

En la figura 3.17 se incluye un sistema predictor de primer orden. Como se mencionó anteriormente, para utilizar esta técnica eficientemente, la señal debe mostrar una gran correlación entre muestras adyacentes, de tal forma que sea suficiente un bit para representar las diferencias entre muestras; esto se logra sobre-muestreando la señal.

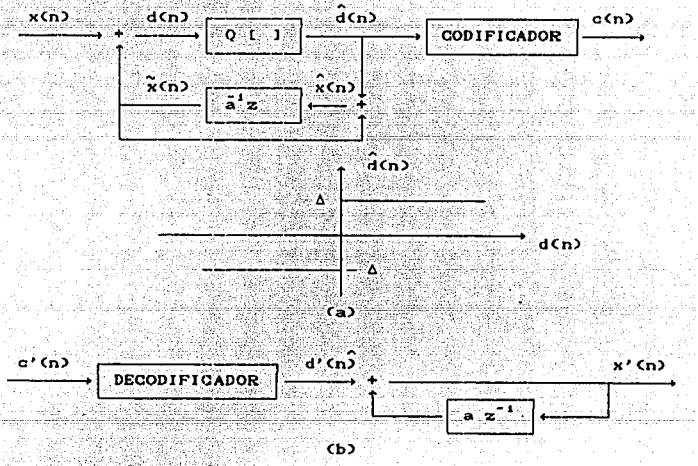


Figura 3.17 Sistema de modulación delta . (a) codificador (b) decodificador

En la figura 3.18 se muestra un ejemplo de modulación delta, con a = 1 en el sistema predictor. En general, $\hat{x}(n)$ satisface la ecuación

$$\hat{x}(n) = a\,\hat{x}(n-1) + \hat{d}(n) \qquad (3.38)$$

Debido a que la señal reconstruida puede crecer con una pendiente maxima de Δ/T , a este tipo de codificación se le conoce como modulación delta lineal. De la figura 3.18, se puede observar que si la pendiente de la señal es mayor que Δ/T , que es la mayor pendiente que el sistema puede seguir, se produce un error llamado de sobrecarga por pendiente. Se puede ver también que en zonas donde la señal es relativamente constante, el codificador delta provoca brincos en la señal reconstruida alrededor de la señal original, a este tipo de ruido se le conoce como ruido granular.

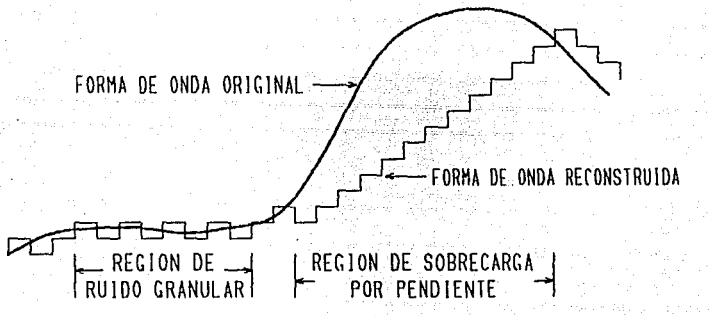


Figura 3.18 Ejemplo de modulación delta lineal.

III.3.5.2 MODULACION DELTA ADAPTABLE.

En la modulación delta adaptable, usualmente se utiliza la adaptación por realimentación. En la figura 3.19 se muestra un esquema

general de un sistema de este tipo.

Existen algunos algoritmos para realizar la modulación delta adaptable, uno de ellos es el siguiente :

Se trata de una modificación del esquema de cuantización adaptable mencionado anteriormente. El paso de cuantización se calcula a partir del paso utilizado en el instante anterior, esto es

$$\Delta(n) = M \ \Delta(n-1) \qquad (3.39)$$

donde $\Delta_{min} \leq \Delta(n) \leq \Delta_{max}$. La variable M puede tomar los valores P y Q

$$M = P > 1 \quad \text{si } c(n) = c(n-1)$$
$$M = Q < 1 \quad \text{si } c(n) \neq c(n-1) \qquad (3.40)$$

Debe notarse que la palabra de código para la muestra presente se utiliza antes de que la muestra sea cuantizada. Esto es posible ya que c(n) está determinada por el signo de la diferencia $\hat{d}(n) = x(n) - \tilde{x}(n)$. La figura 3.20 muestra que se puede determinar si x(n) está arriba o abajo de $\tilde{x}(n)$ antes de la cuantización de $\hat{d}(n)$, lo cual hace que c(n) sea conocida antes. Entonces, cuando se calcula d(n), el valor de $\Delta(n)$ puede ser utilizado.

Los parámetros del sistema de modulación adaptable son P, Q, Δ_{min} y Δ_{max}. Los límites del paso de cuantización deben seleccionarse de manera que incluyan el intervalo dinámico de la señal. La relación $\Delta_{max}/\Delta_{min}$ debe ser suficientemente grande para mantener una relación señal a ruido grande comparada con el nivel de la señal de entrada. Por otro lado el

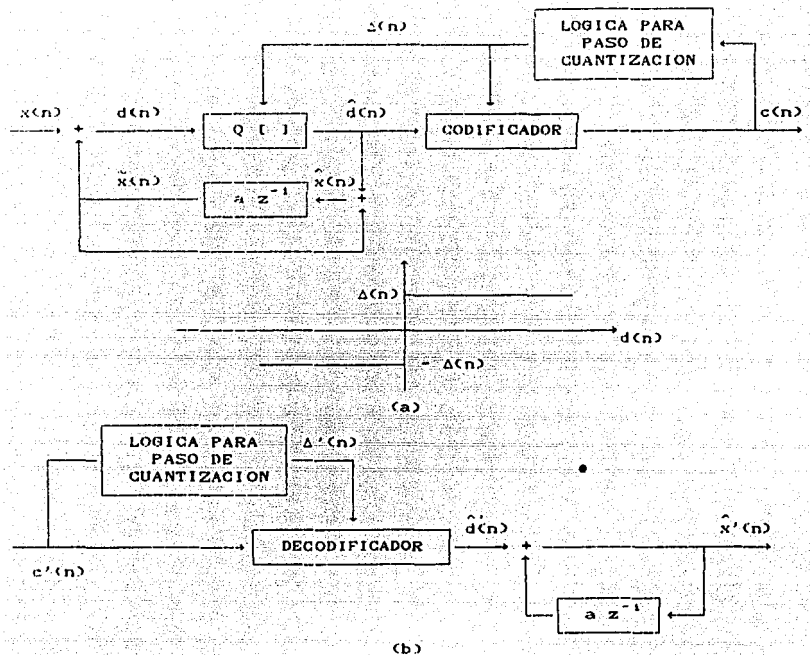


Figura 3.19 Sistema de modulación delta adaptable. (a) codificador (b) decodififcador.

paso de cuantización debe ser pequeño para reducir el ruido de cuantización. Jayant [8] ha demostrado que P y Q deben satisfacer la relación PQ ≤ 1 para tener estabilidad. Un ejemplo se muestra en la figura 3.20 donde P = 2.0 y Q = 0.5.

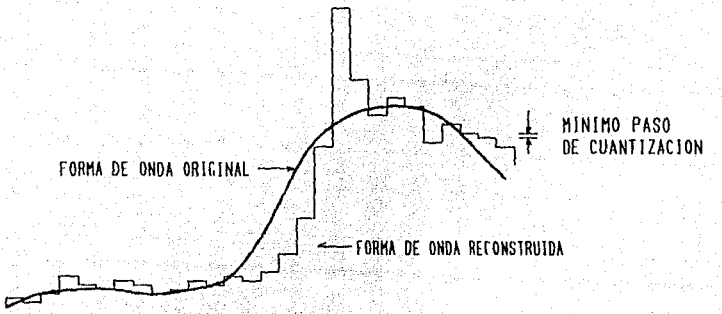


Figura 3.20 Ejemplo de modulación delta adaptable.

III.3.6 PCM DIFERENCIAL

Cualquier sistema que utilice el esquema mencionado en la sección III.3.4 puede llamarse PCM diferencial (DPCM). Por ejemplo, la modulación delta, es un sistema DPCM de 1 bit. Por otro lado, el término DPCM se utiliza más bien para sistemas de cuantización diferencial en los cuales el cuantizador tiene mas de dos niveles.

De la figura 3.21 [11], se puede observar que los sistemas DPCM con numero fijo de predictores pueden suministrar de 4 a 11 dB de ganancia sobre la cuantización directa (PCM) . El beneficio se obtiene al pasar de un sistema sin predictor a un sistema con predictor de primer orden . Esta ganancia en relación señal a ruido, implica que un sistema DPCM

puede alcanzar un valor dado de relación señal a ruido utilizando un bit menos del que se requeriría utilizando el mismo cuantizador directamente sobre la forma de onda.

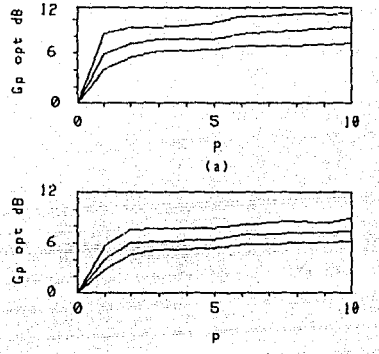


Figura 3.21 Ganancia vs numero de coeficientes del predictor
(a) Señal filtrada paso bajas (b) Señal filtrada paso banda
Las lineas de los extremos limitan el rango obtenido con cuatro pruebas, la del centro es el promedio.

III.3.7 PCM DIFERENCIAL ADAPTABLE.

La cuantización adaptable discutida en la sección III.3.3, puede ser aplicada al sistema DPCM, a estos sistemas se les conoce como cuantizadores diferenciales adaptables PCM (ADPCM). Como se mencionó en la sección citada, existen dos esquemas de cuantización adaptable, por

realimentación y por avance.

Utilizando cuantización adaptable por avance, el paso de cuantización es proporcional a la variancia de la entrada al cuantizador. Se ha demostrado que estos procedimientos de adaptación mejoran en 5 dB la relación señal a ruido sobre la cuantización con ley μ PCM. Este mejor desempeño junto con los 6 dB que se obtienen de utilizar el esquema diferencial con predictor generan una relación señal a ruido de 10-11 dB más que la que se podría obtener de un cuantizador fijo con el mismo número de niveles.

En el caso de la cuantización adaptable por realimentación, el mejoramiento es de 4 - 6 dB sobre el cuantizado con ley μ con el mismo numero de bits.

Otra forma de lograr un mejor desempeño con este tipo de sistemas es sustituir el sistema predictor de primer orden por un predictor adaptable, el cual puede proporcionar de 10 - 12 dB de ganancia. En la figura 3.22 se muestra un sistema DPCM con cuantización adaptable y predictor adaptable. La linea punteada indica que en ambos sistemas, tanto cuantizador como predictor pueden utilizarse el esquema de realimentación o el de avance.

Los coeficientes del sistema predictor se consideran dependientes del tiempo, así el valor predicho es

$$\tilde{x}(n) = \sum_{k=1}^{p} \alpha_k(n)\, \hat{x}(n-k) \qquad (3.43)$$

Para adaptar los coeficientes del predictor $\alpha(n)$ es común considerar que las caracteristicas de las señales de voz permanecen fijas durante

pequeños intervalos de tiempo.

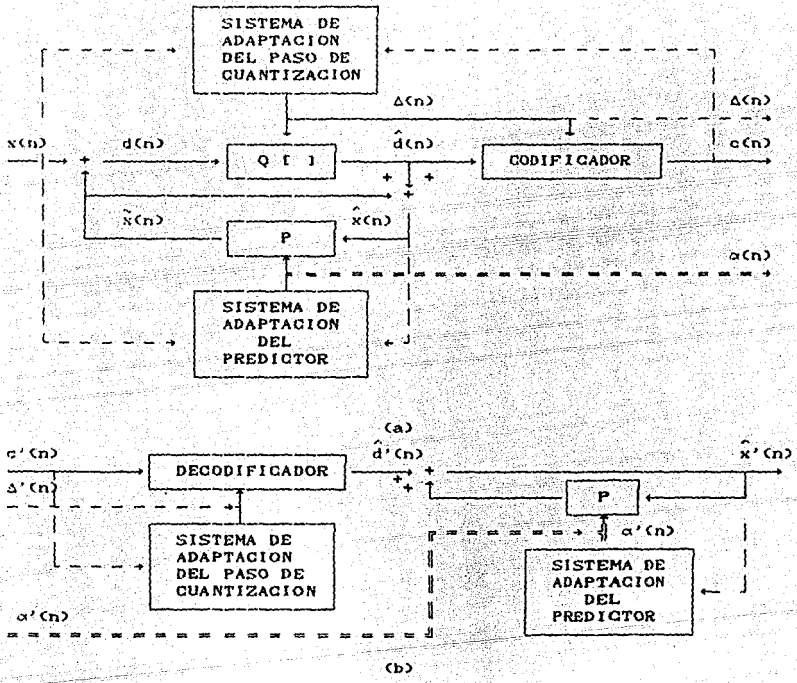


Figura 3.22 Sistema ADPCM con cuantización adaptable y predictor adaptable: (a) codificador (b) decodificador

En [4] se menciona que el desempeño de un sistema DPCM con predictor adaptable es de 14 dB de ganancia como máximo. Otra característica importante de estos sistemas es que los predictores fijos

son sensibles a quien habla y a lo que se habla, mientras que los predictores adaptables no.

III.3.7 CUANTIZACION VECTORIAL.

Un cuantizador vectorial es un sistema que mapea una secuencia de vectores continuos o discretos, a una secuencia digital adecuada para ser transmitida o almacenada. La principal característica de este sistema es permitir la compresión de información, reduciendo así, la cantidad de informacion a transmitir o los requerimientos de memoria para su almacenamiento. [12]

El mapeo de cada vector puede o no tener memoria en el sentido de depender de las acciones pasadas del codificador. Matemáticamente, un cuantizador vectorial de dimensión k sin memoria, consiste de dos mapeos: un codificador γ que asigna a cada vector de entrada $x = (x_0, x_1, ..., x_{k-1})$ un simbolo $\gamma(x)$ de un conjunto de símbolos M, y un decodificador β que asigna a cada simbolo de entrada $v \in M$ un valor del alfabeto de reproducción $\hat{A}$. El conjunto de símbolos, por conveniencia, es un espacio de vectores binarios; por ejemplo, M puede ser un conjunto de 2^R vectores binarios de dimensión R.

La aplicación de un cuantizador en la compresión de información se muestra en la figura 3.23. Los vectores de entrada pueden ser muestras consecutivas de una señal o conjuntos consecutivos de parámetros. Así, la fuente de datos o información es una secuencia de vectores aleatorios x_n, el codificador produce una secuencia de simbolos U_n; el

decodificador recibe una secuencia $\hat{U}_n$ y la mapea a la secuencia final de reproduccion $\hat{x}_n$.

Figura 3.23 Sistema de compresión de información

Debe observarse que la cuantización vectorial permite el manejo de velocidades de transmisión de fracciones de bit/muestra, cosa que no ocurre en el caso escalar. Por ejemplo, un sistema PCM escalar debe tener una tasa de al menos 1 bit/muestra, mientras que un cuantizador vectorial de dimensión k puede tener una tasa de 1/k bit/muestra, ya que se tiene un simbolo para representar un vector de dimensión k.

El propósito de todo cuantizador es proporcionar la mejor secuencia de reproducción posible para una tasa dada. Para cuantificar el desempeño de un cuantizador, se requiere la utilización de medidas de distorsión.

Se considera que la distorsión provocada al reproducir un vector de entrada x_n por un vector de reproducción $\hat{x}_n$ esta dada por una medida de distorsión $d(x,\hat{x})$. En la literatura se han propuesto muchas medidas de distorsión. Dada una medida de distorsión, se puede cuantificar el desempeño de un sistema con una distorsión promedio $E\{d(x,\hat{x})\}$ entre la entrada y la reproducción final. Un sistema tendrá buen desempeño si su

distorsión promedio es pequeña. En la práctica, el promedio más importante es el promedio en tiempo

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} d (x_i , \hat{x}_i) \qquad (3.41)$$

si el límite existe. Si el proceso utilizado es estacionario y ergódico, entonces, con probabilidad uno, el límite existe y es igual al valor esperado de la distorsión E $\langle d(x,\hat{x}) \rangle$.

A continuación se mencionarán dos tipos de medidas de distorsión comunmente utilizadas en la codificación de señales de voz.

(1) *Error cuadrático*. En este caso tanto el espacio de entrada como el espacio de reproducción son espacios Euclidianos de dimensión k, siendo la medida de distorsión

$$d(x,\hat{x}) = ||x - \hat{x}||^2 = \sum_{i=0}^{k-1} (x_i - \hat{x}_i)^2 \qquad (3.42)$$

el cuadrado de la distancia Euclidiana entre los vectores. Al utilizar esta medidad de distorsión es común expresarla en términos de la relación señal a ruido de cuantización

$$SNR = 10 \log \frac{E\langle ||X||^2 \rangle}{E\langle d(X,\hat{X}) \rangle} \qquad (3.43)$$

Esto corresponde a normalizar la distorsión promedio con el promedio de energía en una escala logaritmica: un valor grande (pequeño) de relación señal a ruido corresponde a un valor pequeño (grande) de distorsión promedio.

(2) *Distorsión de Itakura-Saito*. Esta medida de distorsión es un caso especial de la medida de mínima entropía relativa.

Considerando que el vector de entrada es una colección de muestras

sucesivas de una forma de onda, y que el vector de salida es de la forma $\hat{x} = (\alpha , a_1 , a_2 , a_3 , \ldots , a_n)$, donde α es positivo y representa a la ganancia o energia residual y a_i con $a_0 = 1$ son los coeficientes de un filtro en el sentido de que si

$$A(z) = \sum_{i=0}^{p} a_i z^{-i} \qquad (3.44)$$

entonces el filtro con transformada z igual a $1/A(z)$ es un filtro estable.

La distorsión de Itakura-Saito entre el vector de entrada y el vector de salida se define en el dominio del tiempo como

$$d(x,\hat{x}) = \frac{a^t R(x) a}{\alpha} - \ln \frac{\alpha_p(x)}{\alpha} - 1 \qquad (3.45)$$

donde $a^t = (1 , a_1 , \ldots , a_p)$, $R(x)$ es la matriz de autocorrelación del vector de entrada x de dimensión $(p + 1)(p + 1)$, y donde $\alpha_p(x)$ es una ganancia de entrada definida como el valor minimode $b^t R(x) b$, donde el minimo se toma sobre todos los vectores b con la primer componente igual a uno.

Existen formas equivalentes de expresar esta medida de distorsión, algunas especialmente útiles para la teoria y otras más prácticas para el cálculo. La fórmula anterior es una de las más simples.

Propiedades de un cuantizador óptimo. Un cuantizador vectorial es óptimo si minimiza la distorsión promedio $E [X, \beta(\gamma(X))]$. La colección de posibles vectores de reproducción $C = \{$ todos los vectores $y : y = \beta(v)$, para algunos $v \in M \}$, se le llama el alfabeto de reproducción o

simplemente alfabeto del cuantizador y a sus componentes palabras de alfabeto.

Propiedad 1. Con el objetivo de minimizar la distorsión promedio y dado un decodificador especifico β, ningún conjunto cuantizador codificador sin memoria puede hacer mejor trabajo que seleccionar la palabra del alfabeto v de M que genere la menor distorsión posible a la salida, esto es, seleccionar el simbolo que genere la minima

$$d\{x,\beta[\gamma(x)]\} = \min_{v \in M} d[x,\beta(v)] = \min_{y \in C} d(x,y) \qquad (3.46)$$

Esto es, para un decodificador dado en un cuantizador vectorial sin memoria el mejor codificador es el que realice un mapeo de minima distorsión o mapeo de vecino más cercano

$$\gamma(x) = \min_{v \in M}{}^{-1} d[x,\beta(v)] \qquad (3.47)$$

donde la notación de minimo inverso significa que seleccionamos v dado el minimo de 3.46.

Un codificador γ puede pensarse como la partición de un espacio en subespacios donde todos los vectores de entrada con un mismo vector de reproducción están agrupados. Este tipo de partición de acuerdo con la regla de minima distorsión se le conoce con el nombre de partición de Voronoi o de Dirichlet. Un codificador con cuantización vectorial con distorsión minima se muestra en la figura 3.24.

Un ejemplo sencillo de este tipo de partición se muestra en la figura 3.25. Asi como la regla de minima distorsión optimiza el

codificador de un cuantizador vectorial sin memoria, también se puede optimar el decodificador dado un codificador.

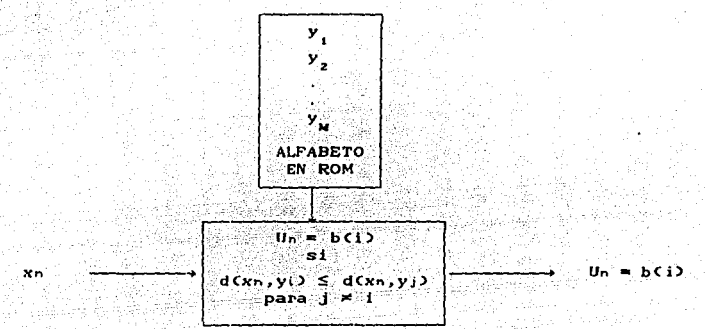


Figura 3.24 Codificador con cuantización vectorial. Se calcula la distorsión entre el vector de entrada y cada palabra de alfabeto almacenada. La salida codificada es la representación binaria del índice de la palabra de alfabeto de mínima distorsión.

Propiedad 2. Dado un codificador γ, ningún decodificador puede hacer mejor trabajo que asignar a cada símbolo v, el centroide generalizado (centro de gravedad o baricentro) de todos los vectores fuente codificados en ese símbolo,

$$\beta(v) = cent(v) = \min_{\hat{x} \in \hat{A}}{}^{-1} E\left\{ d(X,\hat{x}) \mid \gamma(X) = v \right\} \quad (3.48)$$

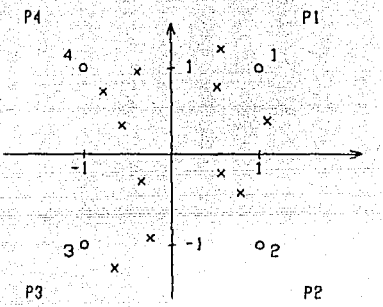
$\beta(v)$ es el vector que genera la mínima distorsión promedio dado el vector de entrada que se mapeó en v.

Por ejemplo, el centroide en el caso de una distribución de muestras y medida de distorsión de error cuadrático es simplemente el

centroide euclideano, la suma vectorial de todos los vectores de entrada codificados por un simbolo dado. Dada una distribución de muestras definida por la secuencia de entrenamiento $\{ x_i; i = 0, 1, 2, \ldots, L-1 \}$ entonces

$$cent(v) = \frac{1}{i(v)} \sum_{x_i : \gamma(x_i)=v} x_i \qquad (3.49)$$

donde $i(v)$ es el numero de indices i para los cuales $\gamma(x_i) = v$.



x : VECTORES DE ENTRENAMIENTO
o : PALABRAS DE ALFABETO
Pi : REGION CODIFICADA POR LA PALABRA i

Figura 3.25 Partición dos dimensiones con minima distorsión. Los cuatro circulos son las palabras del alfabeto de dos dimensiones. Las regiones de Voronoi son los cuadrantes que contienen a los circulos. Cada vector de entrada se mapea a la palabra de alfabeto mas cercana, esto es, el circulo en el mismo cuadrante.

Los centroides euclidianos del ejemplo de la figura 3.25 se muestran en la figura 3.26. Las nuevas palabras del alfabeto representan

mejor a los vectores de entrenamiento que las palabras de alfabeto anteriores. generan ademas una nueva particion de minima distorsión como se muestra en la figura 3.26 con linea punteada. Esta es la clave del algoritmo: iterativamente, optimar el alfabeto del codificador anterior y a continuación optimar un codificador de minima distorsión para el nuevo alfabeto.

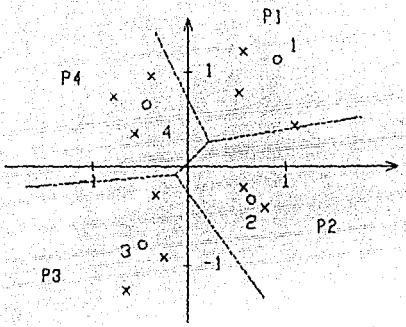


Figura 3.26 Centroides de la figura 3.24 .Los nuevos centroides de las regiones Voronoi se muestran en circulos. Nótese que el cálculo de los centroides ha desplazado a las palabras del alfabeto para que representen mejor a los vectores de entrada.

El hecho de que un codificador pueda optimarse utilizando el decodificador y viceversa forma la base del algoritmo de Lloyd. Los algoritmos de diseño de cuantizadores vectoriales se basan en la observación de que el algoritmo de Lloyd es valido para vectores, para distribuciones de muestras y para una variedad de medidas de distorsión. La unica restricción sobre la medida de distorsión es que se puedan

calcular los centroides, el algoritmo es el siguiente :

Paso 0. Dada una secuencia de entrenamiento de un decodificador inicial.

Paso 1. Codificar la secuencia de entrenamiento en una de simbolos utilizando la regla de minima distorsión y el decodificador dado. Si la distorsión promedio es suficientemente pequeña, termina .

Paso 2. Remplazar las palabras del alfabeto de reproducción anteriores del decodificador para cada simbolo v, por el centroide de todos los vectores de entrenamiento quo quedan incluidos en v en el paso 1. Regresar a 1.

Este algoritmo fue desarollado para cuantizadores vectoriales, secuencias de entrenamiento y medidas de distorsión generales por Linde, Buzo y Gray [13], se conoce como el algoritmo LBG.

El algoritmo mencionado anteriormente requiere un alfabeto inicial para su operación. Se han desarrollado básicamente dos aproximaciones : una inicia con un alfabeto simple del tamaño deseado y el otro inicia con un alfabeto simple pequeño y recursivamente genera más grandes.

Un ejemplo de la primera aproximación es utilizar un código escalar, como un cuantizador uniforme, sucesivamente k veces y adecuar el alfabeto de vectores resultantes al tamaño correcto. El modelo matemático para este tipo de códigos es un código producto.

En aplicaciones de codificación de forma de onda donde los alfabetos de reproducción y entrada son los mismos - espacios euclideanos de dimensión k - los códigos producto permiten generar a

partir de alfabetos de dimensión pequeña, otros de dimensiones mayores. Iniciando con un cuantizador escalar Co y utilizando un código producto de dos dimensiones Co x Co, como alfabeto inicial para el diseño de un cuantizador vectorial de dos dimensiones. Al completar el diseño tendremos un código de dos dimensiones, C^2. Para formar un alfabeto inicial para el diseño de un cuantizador vectorial de tres dimensiones, se toman todas las posibles parejas de C^2 y los escalares de Co, esto es, se utiliza el código producto C^2 x Co . Continuando de esta forma, se obtiene un cuantizador vectorial de dimensión k-1 descrito por un alfabeto C^{k-1} ; un alfabeto inicial para el diseño de un cuantizador de dimensión k es el código producto C^{k-1} x Co.

En lugar de construir grandes códigos, a partir de códigos de dimensión menor, se puede construir un alfabeto utilizando la técnica de división (splitting). Este método puede utilizarse para cualquier dimensión incluyendo códigos escalares y consiste en lo siguiente:

Primero se encuentra el código óptimo 0 - el centroide de toda la secuencia de entrenamiento -, como se muestra en la figura 3.26a para un alfabeto de entrada de dos dimensiones. Esta palabra de alfabeto se divide en dos para formar dos palabras de alfabeto, figura 3.26b. Por ejemplo, puede perturbarse la energía ligeramente para formar la segunda palabra o puede encontrarse la palabra más lejana a la primera. Es conveniente tener la palabra de alfabeto original como miembro del nuevo par para asegurar que la distorsión no se incrementará. El algoritmo se repite para obtener un código de un bit por vector como se muestra en la figura 3.26c. El diseño continúa de esta forma, el código final de una etapa se divide para formar el código inicial de la siguiente. Figura

3.26 d.

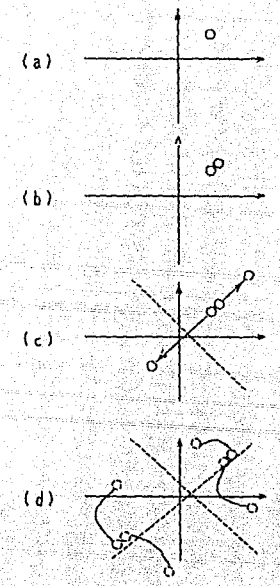


Figura 3.26 Ejemplo de generación de un alfabeto inicial por división (splitting)

En este capítulo se han mencionado las bases de la digitalización de señales, enfocado a señales de voz. A continuación se describen los sistemas de compresión y la forma de calcular algunas características de las señales de voz.

IV. TECNICAS PARA LA COMPRESION DE SEÑALES DE VOZ.

IV.1 INTRODUCCION

La idea de compresión de información en general no es nueva, ya que siempre ha existido un interés economico en ello. Aún se utilizan abreviaciones y acronimos tanto en material hablado como escrito. Una de las primeras técnicas conocidas de compresion de información es el código Morse utilizado en la telegrafía.

La compresión de información es la reducción, en la cantidad de recursos que deben ocupar un conjunto de mensajes o muestras, sin sacrificar calidad más allá de lo tolerable. Estos recursos pueden ser un volumen fisico, como un medio de almacenamiento de datos, por ejemplo la cinta magnética; un intervalo de tiempo, como el requerido para transmitir un conjunto de mensajes dado; o una porción del espectro electromagnético, como el ancho de banda requerido para transmitir un conjunto de mensajes. Todas estas formas de espacio de señal - volumen.

tiempo y ancho de banda - están relacionadas por

$$\text{Volumen} = f(\text{tiempo} \times \text{ancho de banda}) \qquad (4.1)$$

Así, una reducción en volumen puede traducirse en una reducción en el tiempo de transmisión o en el ancho de banda. El parámetro a reducir o comprimir usualmente determina el dominio en el cual se realizará la operación de compresión de la información en el sistema. [9]

La importancia de estos parámetros depende totalmente de la aplicación, pues mientras para cierta aplicación es de gran interés el reducir el ancho de banda requerido para transmitir una señal, como en telefonía, para otras el interés puede ser transmitir información a mayor velocidad o en otros casos, el volumen de información es el parámetro crítico que necesita ser reducido.

Han existido varios intentos de agrupar las técnicas de compresión de información en diferentes clases, pero no existe una clasificación que incluya a todas la técnicas que se han desarrollado. Una clasificación es la siguiente: técnicas reversibles y técnicas irreversibles.

REVERSIBLES	IRREVERSIBLES
Reducción de redundancia	Reducción de entropía

Una operación de reducción de entropía es una reducción de información, debido a que la entropía se define como el promedio de la información. La información perdida no puede ser recuperada, así pues, la operación de reducción de entropía es una operación irreversible.

Por otro lado, puede considerarse que los datos están formados de dos partes: información y redundancia. Una operación de reducción de redundancia elimina, o al menos la reduce, de tal forma que pueda ser posteriormente insertada en los datos. Así pues, la reducción de redundancia es un proceso reversible.

Actualmente, este conjunto de técnicas hacen atractiva la aplicabilidad de sistemas de almacenamiento y comunicación de voz, pues permiten representar una mayor cantidad de señal de voz con un número dado de dígitos binarios, sin perder la calidad natural de la voz. Estas nuevas técnicas pueden lograr tasas de transmisión de 16 kbits por segundo y eventualmente 4 kbits por segundo, en comparación con los valores estándar de 64 y 32 kbits por segundo.

No obstante la creciente disponibilidad de canales de transmisión con grandes anchos de banda, como las fibras ópticas, la codificación de voz para su transmisión a baja velocidad no ha perdido su importancia. Una de las razones es el deseo de tener sistemas de almacenamiento eficiente de voz, para correo electrónico o equipos con respuesta audible. La codificación de voz a bajas tasas de transmisión, es crítica para acomodar mas usuarios en canales con limitaciones de ancho de banda o potencia, o para compartir canales de transmisión con voz y datos.

En el desarrollo de sistemas de compresión de señales de voz, los parámetros a optimar son: la velocidad de transmisión, la calidad, la complejidad y el retraso. Mientras se reduce la velocidad de transmisión, la calidad disminuye pero la complejidad aumenta;

generalmente un aumento en complejidad involucra un aumento en costo y en muchos tipos de codificación también aumenta el retraso, lo cual no es problema en sistemas de almacenamiento de voz, pero puede serlo en otras aplicaciones.

A velocidades de transmisión de 64 kbits/s, la calidad no es un problema; por ejemplo, con un sistema PCM se obtiene una muy buena calidad, de hecho, es muy difícil o imposible reconocer cuando la transmisión es analógica o digital.

El único algoritmo especial utilizado en la codificación PCM a 64 kbits/s es la cuantización no uniforme, en la cual el paso de cuantización aumenta a medida que la amplitud de la señal de voz aumenta, esta no linealidad favorece a la amplitudes pequeñas, que predominan en las señales de voz.

Para lograr menores tasas de transmisión se utilizan técnicas más elaboradas. En general la función de éstas es eliminar redundancias en la señal de voz y utilizar los bits disponibles para codificar la parte no redundante de la señal de una forma eficiente.

Un método es la predicción lineal, la cual comprime una señal de voz al estimarla como una función lineal de las salidas anteriores del sistema de codificación. El error de predicción tiende a tener menor energía que la señal original, y puede ser codificado utilizando menos bits, para un nivel permisible de error en la reproducción. Otra aproximación es utilizar codificación adaptable de subbandas, el cual

separa la señal de voz en bandas de frecuencia y asigna los bits disponibles de tal forma que en las bandas de baja frecuencia, donde aparecen el tono y los formantes de la señal, se tenga muy buena precisión, mientras que en las bandas de alta frecuencia, donde se encuentran los sonidos fricativos, se utilicen pocos bits.

Estas técnicas se pueden combinar con otras, como lacuantización vectorial, para lograr una reducción significativa en la velocidad de transmisión .

A las técnicas que producen, a la salida del sistema, una aproximación de la señal de voz de entrada se les conoce como codificadores de forma de onda. Por otro lado, existen técnicas, que presentan una descripción compacta de la señal de entrada y codifican los parámetros de ésta (esta descripción está basada en la noción de una señal alimentada a un filtro lineal - un modelo que trata de simular el proceso de excitación en el sistema vócal humano). A estas técnicas se les conoce como codificadores de voz (Vocoders).

En estos codificadores de voz, el resultado es un sonido artificial en el cual las palabras son perfectamente entendibles pero dificilmente se identifica a quien las dijo. [14]

Finalmente, han surgido técnicas basadas en la combinación de los codificadores de voz y los codificadores de formas de onda, que se conocen como codificadores híbridos. Con ellos se pueden lograr tasas de transmisión aún menores, al alimentar una señal de excitación optimada a

un filtro con predicción lineal. En la figura 4.1 se muestran diferentes tasas de transmisión en una escala unidimensional, y una asignación aproximada de la calidad de la voz que se puede obtener. [8]

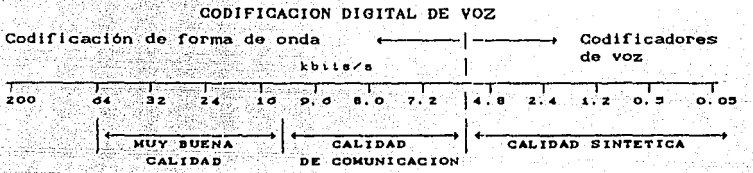


Figura 4.1 Espectro de velocidades de transmisión de codificación de voz en escala no lineal con su calidad asociada.

TABLA 4.1 COMPLEJIDAD RELATIVA DE SISTEMAS DE COMPRESION DE VOZ

COMPLEJIDAD RELATIVA	CODIFICADOR
1	ADM : Modulación Delta Adaptable
1	ADPCM : PCM Diferencial Adaptable
5	SUBBAND : Codificador por Subbandas
5	P-P ADPCM : ADPCM con Predicción de Tono
50	APC : Codificador por Predicción Adaptable
50	ATC : Codificador por Transformación Adaptable
50	ϕV : Codificador de Voz por Fase
50	VEV : Codificador de Voz por Excitación de Voz
100	LPC : Codificación por Predicción Lineal
100	CV : Codificador de Voz de Canal
200	ORTHOG : LPC con Coeficientes Ortogonalizados
500	FORMANT : Codificador de Voz por Formantes
1000	ARTICULATORY : Sintetizador del Sistema Vocal

Como se puede esperar, con el objetivo de lograr la mejor calidad posible mientras se reduce la velocidad de transmisión, es necesario utilizar algoritmos más sofisticados, lo cual en este contexto, implica una gran cantidad de cálculos y en consecuencia grandes tiempos de ejecución. En la Tabla 4.1, se muestra una comparación cualitativa de la complejidad de algunos sistemas de codificación de señales de voz, algunos de los cuales utilizan técnicas de cuantización que se mencionaron en el capitulo anterior y otras se mencionarán más adelante.

En la figura 4.2 se muestran en diagramas de bloque, diferentes sistemas de codificación de voz : (A) PCM, (B) PCM diferencial adpatable ADPCM, (C) Codificación adptable por subbandas, (D) Codificación por predicción lineal multipulso, (E) Codificación por predicción lineal excitando estocásticamente, (F) Codificador de voz (Vocoder) por predicción lineal (LPC). [14]

IV.2 CORRELACIONES.

La función de autocorrelación de una señal discreta en el tiempo x[n] se define como [10]

$$\phi(m) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n]\, x[n+m] \qquad (4.2)$$

La función de autocorrelación es útil para mostrar la estructura de cualquier forma de onda [10] . Por ejemplo, si una señal es periódica con periodo P , es fácil demostrar que

$$\phi(m) = \phi(m+P) \qquad (4.3)$$

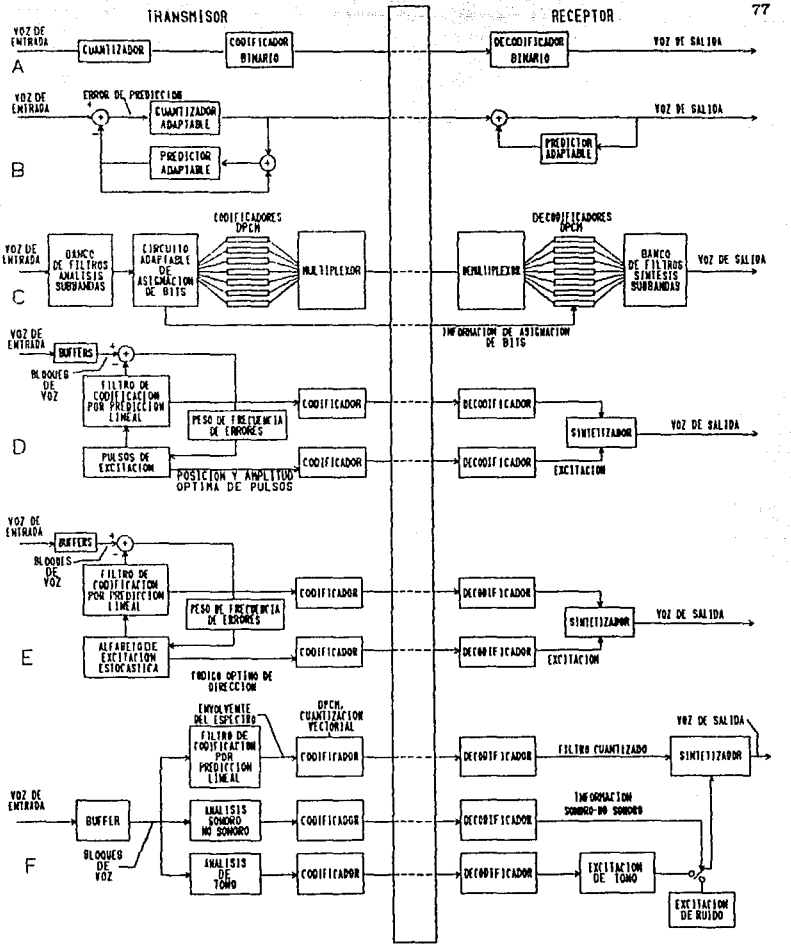


Figura 4.2 Diferentes esquemas de codificación de voz.

Así, la periodicidad en la función de autocorrelación indica periodicidad en la señal. Una función de autocorrelación que tiene un pico en m=0 y decae rápidamente a cero cuando m aumenta, indica que la señal no tiene una estructura predecible [10].

La señal de voz no es estacionaria, pero sin embargo, las propiedades de la señal de voz no cambian en intervalos de tiempo relativamente grandes. Esto lleva a la noción de técnicas de análisis de intervalos de tiempo, que operan sobre pequeños segmentos de la señal de voz. Por ejemplo, considerando un segmento de N muestras de una señal

$$x_i[n] = x[n+i], \qquad 0 \le n \le N-1 \tag{4.4}$$

donde i denota el inicio del segmento. Entonces la función de autocorrelación por intervalo de tiempo se define como

$$\phi_i(m) = \frac{1}{N} \sum_{n=0}^{N'-1} x_i[n]\, x_i[n+m], \qquad 0 \le m \le M_0-1 \tag{4.5}$$

donde M_0 es el máximo retraso de interés; por ejemplo, si se desea observar la periodicidad de una señal $M_0 > P$. El entero N' se especificará más adelante.

Se puede interpretar la expresión 4.5 como la autocorrelación de un segmento de señal de voz de longitud N muestras, iniciando en la muestra i. Si N'=N, entonces se utilizan datos fuera del intervalo $i \le n \le N+i-1$ para el cálculo. Si N'=N-m, entonces sólo se requieren de los datos dentro del intervalo. En este caso, frecuentemente se pesa el segmento con una ventana que suavemente hace tender los extremos del mismo a cero [10]. Utilizando la función de autocorrelación para detectar

periodicidad en una señal de voz, cualquier selección es adecuada, sin embargo, la selección del N' es importante para métodos basados en predicción lineal, el cálculo directo de $\phi(m)$ para $0 \leq m \leq Mo-1$, requiere un esfuerzo computacional proporcional a MoN.

IV.3 TRANSFORMADA DISCRETA DE FOURIER.

TRANSFORMADA RAPIDA DE FOURIER.

La transformada discreta de Fourier (TDF) es el equivalente en el tiempo discreto de la transformada de Fourier de tiempo continuo. Mientras que la transformada de Fourier en el tiempo continuo opera sobre una señal continua en el tiempo x(t), la transformada discreta de Fourier opera sobre muestras de x(t). Si X(f) es la transformada de Fourier continua de x(t), entonces la TDF de x(t) (muestreada) es una secuencia de muestras de X(f), igualmente espaciadas en frecuencia. La ecuación 4.6 calcula la TDF de una señal discreta. X[k] , es la transformada de Fourier discreta, x(n) es la secuencia de muestras de la señal de entrada x(t). El término W_N, se define como $e^{-j2\pi/N}$, que corresponde al término $e^{-j2\pi ft}$ utilizado para calcular la transformada continua de Fourier. [15]

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{nk} \qquad k = 0,\ldots,N-1 \qquad (4.6)$$

donde $W_N = e^{-j\,2\pi/N}$

La expresión 4.6 muestra que en el caso de que x[n] sea una secuencia compleja, la evaluación completa de una TDF de N puntos requiere de $(N-1)^2$ multiplicaciones de números complejos y N(N-1) sumas de numeros complejos. La idea detrás de la Transformada Rápida de

Fourier es dividir la secuencia original de N puntos en dos secuencias, de las cuales se pueden obtener sus TDF, que combinadas darán la TDF de la secuencia original de N puntos.

Por ejemplo, para una secuencia de entrada de N puntos x[n], donde N es potencia de 2, se definen dos secuencias de (N/2) puntos x1[n] y x2[n] como los elementos pares y nones de x[n] respectivamente,

$$\begin{aligned} x_1[n] &= x[2n] \qquad n = 0,1,\ldots,\frac{N}{2} - 1 \\ x_2[n] &= x[2n+1] \qquad n = 0,1,\ldots,\frac{N}{2} - 1 \end{aligned} \tag{4.7}$$

La TDF de x[n] de N puntos se puede expresar en términos de sus componentes pares e impares como:

$$\begin{aligned} X[k] &= \sum_{\substack{n=0 \\ n\ par}}^{N-1} x[n]\ W_N^{nk} + \sum_{\substack{n=0 \\ n\ non}}^{N-1} x[n]\ W_N^{nk} \\ &= \sum_{n=0}^{N/2-1} x[2n]\ W_N^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1]\ W_N^{(2n+1)k} \end{aligned} \tag{4.8}$$

Observando que W_N^2 se puede escribir como

$$W_N^2 = [\ e^{j(2\pi/N)}]^2 = e^{j(2\pi/(N/2))} = W_{N/2} \tag{4.9}$$

que sustituyendo en la ecuación 4.8 resulta

$$\begin{aligned} X[k] &= \sum_{n=0}^{N/2-1} x_1[n]\ W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x_2[n]\ W_{N/2}^{nk} \\ &= X_1[k] + W_N^k\ X_2[k] \end{aligned} \tag{4.10}$$

donde X1[k] y X2[k] son las TDF de x1[n] y x2[n], respectivamente. La ecuación 4.10 muestra que la TDF X[k] de N puntos se puede descomponer en dos TDF de N/2 puntos.

Debido a que X[k] está definido para $0 \leq k \leq N-1$ y X1[k] , X2[k] están definidas para $0 \leq k \leq N/2-1$, se debe buscar la expresión para cuando k es mayor que N/2. Observando que $W_N^{k+N/2} = -W_N^k$ la expresión 4.10 se puede expresar como

$$X[k] = X_1[k] + W_N^k \, X_2[k]$$
$$X[k+N/2] = X_1[k-N/2] - W_N^{k-N/2} \, X_2[k-N/2] \qquad (4.11)$$

El proceso de reducir una TDF de L puntos (donde L es potencia de 2) a una TDF de L/2 puntos se continúa hasta que quedan TDF de 2 puntos. Una TDF de 2 puntos , F[k], k = 0, 1, ... puede evaluarse como

$$F(0) = f(0) + f(1) \, W_8^0$$
$$F(1) = f(0) + f(1) \, W_8^4 \qquad (4.12)$$

donde f(n) , n=0,1, es la secuencia de dos puntos que se transforma. Debido a que $W_8^0 = 1$ y $W_8^4 = -1$, no se requieren multiplicaciones para evaluar 4.12. Por cada división de la TDF en dos TDF se requieren N/2 multiplicaciones complejas para combinar los resultados de la etapa anterior. Debido a que hay ($\log_2 N$) etapas, el número de multiplicaciones complejas requeridas para evaluar una TDF de N puntos es de aproximadamente $N/2 \log_2 N$.

El algoritmo que se acaba de describir se conoce como la Transformada Rápida de Fourier (TRF) con decimación en tiempo, debido a que en cada etapa del proceso la secuencia de entrada (en el tiempo) se divide en secuencias menores para su proceso, es decir, la secuencia se decima en cada etapa. Otra forma de evaluar la TDF a partir de la TRF es con decimación en frecuencia.

Generalmente, se ordena la secuencia de entrada para el cálculo de la TRF debido a que de esta forma las potencias de W_N^k se incrementan en orden creciente, y se produce una salida en orden secuencial. El método para acomodar la secuencia de entrada para el algoritmo de TRF se le conoce como inversión de bits.

Para el calculo de la Transformada Inversa Discreta de Fourier se puede utilizar el mismo algoritmo de la TRF. La TDF inversa de una secuencia de N puntos (X[k], k = 0,1,...,N-1) se define como :

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, W^{-nk} \qquad (4.13)$$

Si se toma el complejo conjugado de la ecuación 4.13 y se multiplica por N se tiene

$$N\, x^*[n] = \sum_{k=0}^{N-1} X^*[k]\, W^{nk} \qquad (4.14)$$

El lado derecho de la expresión 4.14 es la TDF de la secuencia ($X^*[k]$) que se puede calcular utilizando el algoritmo de la TRF que se ha descrito. La secuencia de salida (x[n]) se puede obtener calculando el complejo conjugado de la TDF de la ecuación 4.14 y dividiendo entre N para dar

$$x[n] = \frac{1}{N} \left[\sum^{N-1} X^*[k]\, W^{nk} \right]^* \qquad (4.15)$$

IV.4 CODIFICACION POR PREDICCION LINEAL.

Una de las técnicas más poderosas en el análisis de señales de voz

es el método de predicción lineal. Este método se ha utilizado para la estimación de los parámetros básicos de la voz, el periodo de tono, formantes, espectro, funciones de área del tracto vocal y para representar señales de voz para su transmisión a bajas tasas o para su almacenamiento. La importancia de este método radica en su habilidad de proveer estimaciones precisas de los parámetros de voz, y su relativa velocidad de calculo. [4].[8] y [16]

La idea detras de la predicción lineal es que una muestra de señal de voz puede aproximarse por la combinacion lineal de muestras pasadas. Minimizando la suma del cuadrado de las diferencias (en un intervalo finito) entre las muestras actuales y las linealmente predichas, se puede obtener un conjunto de coeficientes para el predictor. La codificación por predicción lineal se basa en el modelo de producción de voz, mostrado en la figura 4.3.

Para la aplicación de este modelo, se requiere la determinación de si la señal es sonora o no sonora, y si es sonora, con que periodo de tono. La diferencia fundamental entre la codificación por predicción lineal y otros codificadores de voz, es la forma como modela el tracto vocal (y en segundo lugar, el calculo de la ganancia G). En la codificación por predicción lineal, el tracto vocal se modela como un filtro digital solo polos. Agregando la ganancia G, en el filtro, se puede expresar como

$$H(z) = \frac{G}{1 + a_1 z^{-1} + \ldots + a_p z^{-p}} = \frac{S(z)}{E(z)} \qquad (4.16)$$

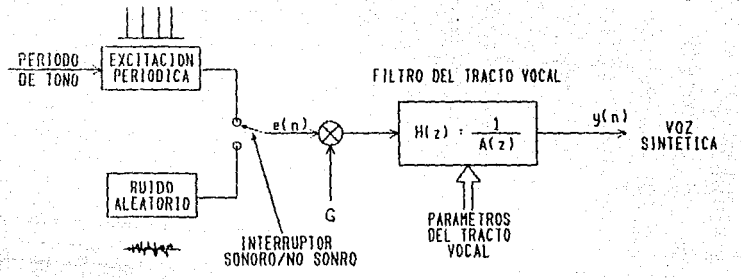


Figura 4.3 Modelo de producción de voz para codificación por predicción lineal.

donde p es el orden del modelo. Si y[n] es la señal de voz de salida del modelo, y e[n] es la entrada de excitación, la expresión 4.16 en el dominio del tiempo es

$$y[n] = G\,e[n] - a_1\,y[n-1] - \ldots - a_p\,y[n-p] \qquad (4.17)$$

Cada muestra de voz se calcula como una combinación lineal de las muestras anteriores más una contribución de la excitación.

Un predictor lineal con coeficientes de predicción a_k se define como un sistema cuya salida es

$$\tilde{y}[n] = \sum_{k=1}^{p} a_k\,y[n-k] \qquad (4.18)$$

El problema fundamental en el análisis por predicción lineal es determinar el conjunto de coeficientes $\{ a_k \}$ directamente de la señal

de voz de tal forma que se obtenga una buena estimación de las propiedades espectrales de la señal de voz utilizando la ecuación 4.16. Para lograr lo anterior se tiene

$$\tilde{y}[n] = - a_1 y[n-1] - \ldots - a_p y[n-p] \qquad (4.19)$$

como la estimación de y[n] a partir de las muestras anteriores, y se determinan los coeficientes a_i, tal que el error

$$\sum_n (y[n] - \tilde{y}[n])^2 \qquad (4.20)$$

sea mínimo sobre todas las muestras disponibles. La minimización del error cuadrático total, con respecto a los coeficientes a_i generan el siguiente conjunto de ecuaciones lineales :

$$\begin{aligned} a_1 r(0) + a_2 r(1) + \ldots + a_p r(p-1) &= - r(1) \\ a_1 r(1) + a_2 r(0) + \ldots + a_p r(p-2) &= - r(2) \\ &\vdots \\ a_1 r(p-1) + a_2 r(p-2) + \ldots + a_p r(0) &= - r(p) \end{aligned} \qquad (4.21)$$

O en forma matricial

$$R \; a = - r \qquad (4.22)$$

En las expresiones anteriores se ha definido

$$r(i) = r(-i) = \sum_{n=0}^{N-i-1} y[n] \, y[n+i] \qquad (4.23)$$

como la i-ésima autocorrelación. En ésta formulación, la señal y[n] se multiplica por una ventana, para asegurar que y[n] = 0 para $n < 0$ y $n \geq N$. Esto es debido a que se considera a la señal de voz como estacionaria, sin embargo sabemos que no lo es, de tal forma que para aplicar el método, se segmenta la señal de voz en bloques llamados

marcos (frames), los cuales son cuasi-estacionarios. En la figura 4.4 se muestra un ejemplo de segmentación de una señal de voz. El tipo de ventana mas utilizado es la de Hamming que está dada por

$$w[n] = 0.54 - 0.46 \cos \frac{2\pi n}{N}, \quad 0 \leq n \leq N-1$$
$$w[n] = 0 \qquad \text{otro caso} \qquad (4.24)$$

En este caso N es la longitud de la ventana en muestras y se selecciona generalmente entre 20 y 40 ms.

La técnica descrita se conoce como el método de autocorrelación, el cual utiliza una matriz R de tipo Toeplitz. Una matriz Toeplitz es aquella cuyas diagonales están compuestas por el mismo elemento. Esta matriz no es singular y puede ser siempre invertida. Por lo tanto, siempre se puede obtener la solución

$$a = - R^{-1} r \qquad (4.25)$$

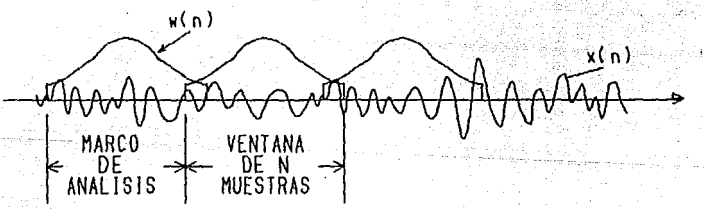


Figura 4.4 Segmentación de una señal de voz en marcos cuasi-estacionarios multiplicando la señal por una ventana.

Existen otros métodos para obtener los coeficientes del modelo de predicción lineal, como el de covariancia, en el cual la señal no se requiere que sea multiplicada por ventanas, pero genera un sistema de ecuaciones que no siempre tiene inversa; o el método de máximo de verosimilitud, que bajo ciertas características obtiene el mismo resultado que el método de autocorrelación.

La solución del sistema de ecuaciones 4.21 se obtiene explotando la característica de la matriz R de ser tipo Toeplitz, se han desarrollado varios algoritmos recursivos bastante eficientes para resolver este tipo de sistemas. El más conocido es el método recursivo desarrollado por Durbin, definido por :

$$
\begin{aligned}
&E(0) = r(0) \\
&k_i = \left[r(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} r(i-j) \right] / E^{(i-1)} \quad , 1 \le i \le p \\
&a_i^{(i)} = k_i \\
&a_j^{(i)} = a_j^{(i-1)} - k_i\, a_{(i-j)}^{(i-1)} \quad , 1 \le j \le i-1 \\
&E^{(i)} = (1-k_i^2)\, E^{(i-1)}
\end{aligned}
\qquad (4.26)
$$

El conjunto de ecuaciones 4.26 se resuelven recursivamente para $i = 1, 2, \ldots, p$ y la solución final está dada por

$$a_j = a_j^{(p)} \qquad (4.27)$$

Debe notarse que la cantidad $E^{(i)}$ del conjunto de expresiones 4.26 es el error de predicción de orden i. Si los coeficientes de autocorrelación se normalizan por r(0) la solución de la matriz no se altera, y el error normalizado se puede expresar como

$$\frac{E^{(i)}}{r(0)} = (1-k_1^2)\ldots(1-k_p^2) \qquad (4.28)$$

$E^{(p)}$ es el cuadrado de la ganancia G necesaria para el modelo de síntesis.

También, es importante mencionar que al mismo tiempo que con esta recursión se obtienen los coeficientes a_i del predictor, se obtienen también los valores k_i conocidos como los coeficientes de reflexión, con los cuales se puede determinar la estabilidad del sistema, se puede demostrar que cuando $|k_i| < 1$ para $1 \leq i \leq p$, las raíces del polinomio denominador de H(z) se encuentran dentro del círculo unitario garantizando así la estabilidad del sistema. [4]

Por otro lado, los coeficientes del sistema predictor son los coeficientes del denominador de una función de transferencia que modela los efectos de la respuesta del tracto vocal combinada con la forma de onda producida por la glotis y la radiación. Así pues, dado un conjunto de coeficientes del predictor podemos encontrar la respuesta en frecuencia del modelo de producción de voz evaluando H(z) para $z=e^{j\omega}$

$$H(e^{j\omega}) = \frac{G}{1 - \sum_{k=1}^{p} a_k\, e^{-j\omega k}} = \frac{G}{A\,(e^{j\omega})} \qquad (4.29)$$

El algoritmo de FFT mencionado en la sección anterior, puede utilizarse para obtener el espectro a partir de los coeficientes del sistema predictor.[4]

IV.5 DETECCION DEL PERIODO DE TONO.

La determinación de la periodicidad en un segmento de voz es muy importante para la aplicación de varios algoritmos de codificación de voz. La frecuencia fundamental de una señal de voz generalmente se designa como Fo y usualmente se le llama frecuencia de tono. El inverso de la frecuencia de tono es el periodo de tono que se expresa en milisegundos, o si se conoce la frecuencia de muestreo en muestras.

Una medida precisa del periodo del tono de una señal de voz a partir de la onda acústica de presion es difícil de realizar por varias razones; por ejemplo, la señal de excitación no es un tren de pulsos perfecto o por la interacción entre el tracto vocal y la excitación de la glotis. Como resultado de los diferentes problemas asociados en la determinación del periodo del tono, se han propuesto, en la literatura del procesamiento de voz, una gran variedad de algoritmos para la detección del mismo. Básicamente, un detector de tono es un sistema que determina si un segmento de voz es sonoro o no sonoro, y durante periodos sonoros, provee una medida del periodo de tono. Debe mencionarse, que existen algoritmos que sólo determinan el periodo del tono. Una posible clasificación de los algoritmos de detección del periodo de tono la presenta Rabiner, et.al. [17] que las divide en tres categorías:

1. Aquellos algoritmos que utilizan, para la detección del tono, las propiedades en el dominio del tiempo de las señales de voz. Las medidas más utilizadas son cruces por cero, valles y crestas y pico en

autocorrelación.

2. Aquellos algoritmos que utilizan las características en frecuencia de las señales de voz, ya que si una señal es periódica en el dominio del tiempo, su espectro de frecuencia consistirá de una serie de impulsos en la frecuencia fundamental y sus armónicas, pudiendo determinar fácilmente el periodo de tono.

3. Aquellos que utilizan tanto las características en el dominio del tiempo como en el dominio de la frecuencia de las señales de voz; por ejemplo, puede utilizar técnicas en el dominio de la frecuencia para obtener un espectro plano y entonces aplicar una medida de autocorrelación para estimar el periodo de tono.

A continuación se mencionarán algunos algoritmos de los reportados en la literatura para la estimacion del periodo de tono. [4], [8] y [17]

A. Método de autocorrelación modificado. Está basado en el método de recortador-central de Sondhi. En la figura 4.5 se muestra un diagrama de bloques de este algoritmo. El método requiere que la señal de voz pase a través de un filtro paso bajas con frecuencia de corte en 900 Hz. La señal filtrada se digitaliza a 10 kHz y se segmenta en bloques de 300 muestras. La primera etapa de procesamiento es el calculo del nivel de recorte c_L para el bloque de 300 muestras. El valor de recorte se establece en el que corresponde al 64% del valor absoluto menor de pico entre las 100 primeras y 100 ultimas muestras del bloque en análisis. A continuación, la seccion de 300 muestras se recorta "centralmente", resultando una señal que toma uno de tres posibles valores +1 si la muestra excede el nivel positivo de recorte, -1 si la muestra es menor

que el nivel de recorte negativo, y cero en caso contrario. A continuación, se calcula la funcion de autocorrelación para el bloque de 300 muestras con retrasos de 20 a 200 muestras. Adicionalmente, se calcula la correlación cero para posteriormente normalizar. Se busca el valor máximo normalizado de la función de autocorrelación y se compara con 0.3, si es mayor se clasifica entonces como un segmento sonoro y la posición donde se encuentra el máximo es el valor del periodo de tono, si es menor se clasifica como no sonoro.

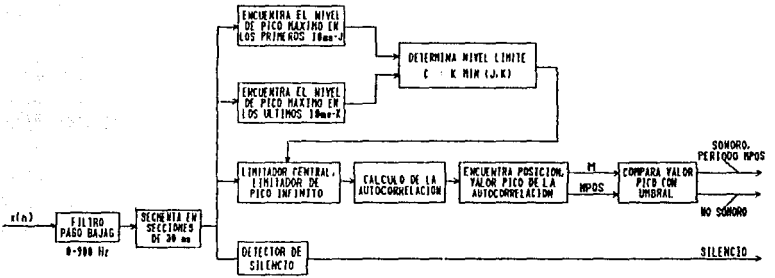


Figura 4.5 Diagrama de bloques del detector de periodo de tono por el método de autocorrelación modificado.

B. Método Cepstral. En la figura 4.6 se muestra un diagrama de bloques del algoritmo de este tipo de detección de periodo de tono. Cada bloque de 512 muestras se multiplica por una ventana de Hamming de 512 puntos, y se calcula el "cepstro" que es la transformada inversa de Fourier del logaritmo del espectro de amplitud del bloque de muestras en

análisis. Se encuentra el valor máximo del "cepstro" y si éste excede un umbral establecido, se determina que la sección es sonora y el periodo de tono se localiza donde se encontró el máximo. Si el máximo no excede el umbral, se cuenta el número de cruces por cero que ocurren en el segmento, si éste excede a un umbral, se clasifica al segmento como no sonoro, en caso contrario se clasifica como sonoro con periodo de tono igual a la posición del valor máximo del "cepstro".

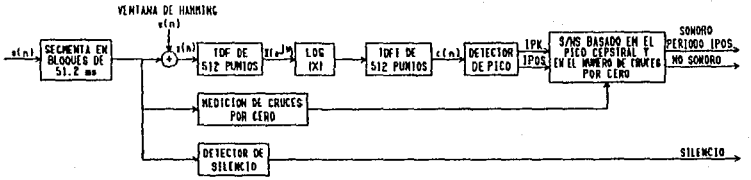


Figura 4.6 Diagrama de bloques del detector de periodo de tono , método cepstral.

C. Técnica simplificada de filtrado inverso (TSFI) . El diagrama de bloques de ésta técnica se muestra en la figura 4.7. Se filtra un bloque de 400 muestras a un ancho de banda de 900 hz y se decima en una relación de 5:1. Los coeficientes del filtro inverso de 4° orden se obtienen utilizando el método de análisis de voz por predicción lineal. La señal decimada pasa a través de este filtro inverso, dando una señal con espectro plano en la cual se calcula la función de autocorrelación. El periodo de tono se determina interpolando la función de autocorrelación en la vecindad del valor máximo de la misma. La decisión

sobre si el segmento es sonoro o no , se basa en la amplitud del valor maximo de la funcion de autocorrelacion, el umbral en este caso es 0.4.

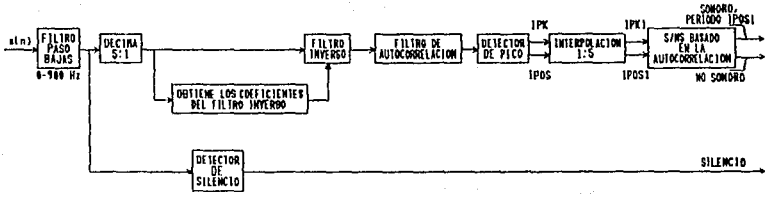


Figura 4.7 Diagrama de bloques del detector de periodo de tono, técnica de filtrado inverso.

D. Método de procesamiento paralelo. En la figura 4.8 se muestra el diagrama de bloques de este método. La señal de voz se filtra a un ancho de banda de 900 hz, después de lo cual, se realizan una serie de mediciones sobre los picos y valles de la señal filtrada generando seis funciones. Se procesa cada una de estas funciones con un estimador elemental de periodo de tono, obteniéndose 6 estimaciones de éste. Finalmente, se combinan las estimaciones con un algoritmo sofisticado para determinar el periodo de tono. La decisión de sonoro-no sonoro se realiza en base a la semejanza entre los 6 valores estimados de periodo de tono.

E. Función de diferencia de magnitud promedio. El diagrama de bloques de esta técnica se muestra en la figura 4.9. La señal de voz muestreada a 10 kHz se decima a 6.67 kHz. Se realiza una medición del

numero de cruces por cero de la señal antes de filtrarla (NCC) y una medición de la energia de la señal ya filtrada (ENE). Se calcula entonces, la sumatoria de los valores absolutos de las diferencias de la señal filtrada, con la misma retrasada de 16 a 124 muestras. La función de diferencia de magnitud promedio es el valor de cada sumatoria para cada retraso. El periodo de tono se determina como la posición donde la función de diferencia toma su valor minimo. Adicionalmente, se calcula el cociente del valor máximo entre el valor minimo de la función de diferencia que combinado con las medidas de energia y cruces por cero determinan si el segmento es sonoro o no sonoro.

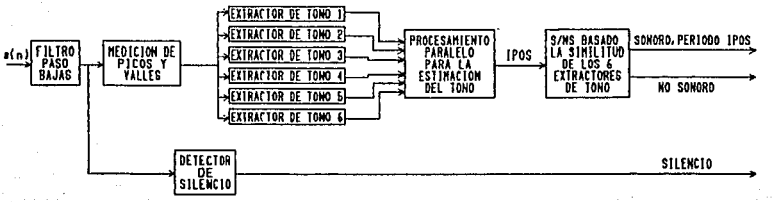


Figura 4.8 Diagrama de bloques del detector de periodo de tono, método de procesamiento paralelo.

En [17] se menciona que en todas las técnicas descritas se incluye un detector de silencio, para distinguir cuando se trata de un segmento con sonido y cuando sólo es ruido, con el fin de no procesar los segmentos que contienen silencio.

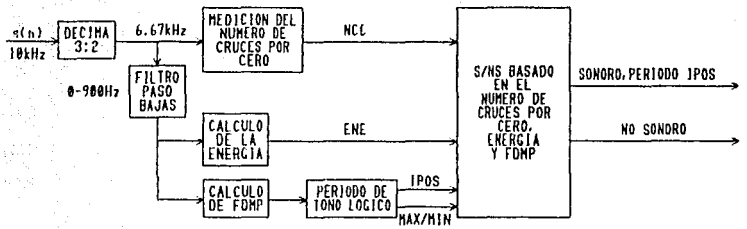


Figura 4.9 Diagrama de bloques del detector de periodo de tono, función diferencia de magnitud promedio.

IV.6. ESCALAMIENTO ARMONICO EN EL DOMINIO DEL TIEMPO. (EADT)

El método de escalamiento armónico en el dominio del tiempo (EADT) es una técnica de codificación de forma de onda que se puede utilizar para reducir la velocidad de transmisión en un factor de 2. [8], [18] [19] y [20].

La función básica de EADT es tomar dos periodos consecutivos de tono en el transmisor y comprimirlos en uno. En el receptor, la señal se expande para recobrar el periodo de tono perdido. De lo anterior se deriva la necesidad de tener conocimiento del periodo de tono del bloque de señal a ser procesado, para ello se puede utilizar cualquiera de los algoritmos que se mencionaron en la sección IV.5; en este caso no es necesaria la decisión sobre si el segmento es sonoro o no. Los errores debidos a la mala detección del periodo de tono no son tan catastróficos

en esta técnica, como en el caso de los codificadores de voz.

En la figura 4.10 se muestra un diagrama de codificación de forma de onda utilizando EADT. El bloque comprimido de salida del EADT se codifica para su transmisión, utilizando cualquiera de las técnicas mencionadas anteriormente como modulación delta, predicción lineal u otra. En el receptor se aplica el proceso inverso.

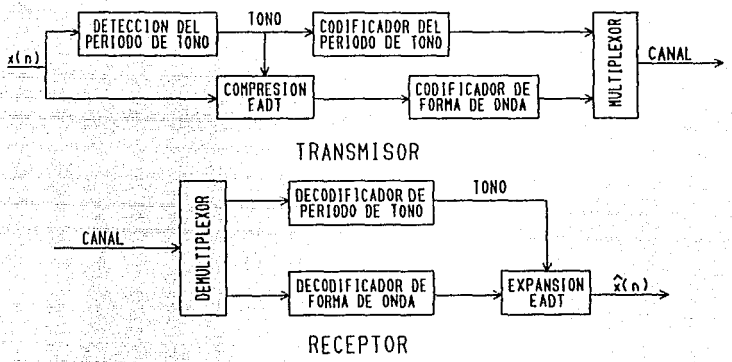


Figura 4.10 Sistema EADT con transmisión del periodo de tono.

En el caso de la figura 4.10, el periodo de tono también se codifica y se transmite al receptor, debido a que se requiere para el proceso de expansión. Sin embargo, ésto incrementa la velocidad de transmisión. Otro esquema es el que se muestra en la figura 4.11, donde no se transmite la información del periodo de tono; éste se calcula

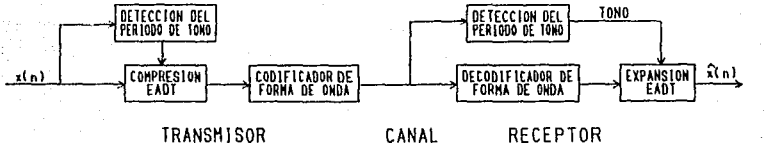


Figura 4.11 Sistema EADT sin transmisión del periodo de tono.

nuevamente en el receptor a partir de la señal recibida. Ahora no existe el aumento en velocidad de transmisión pero el desempeño del sistema se degrada debido a que el detector de periodo de tono opera sobre un menor numero de muestras. Generalmente se obtiene mejor calidad con el primer esquema.

El proceso de compresión utilizando el metodo de EADT se ilustra en la figura 4.12. Se considera que el periodo de tono es de P muestras de longitud y se comprimen 2 bloques consecutivos de P muestras a un bloque de P muestras. La compresión se realiza multiplicando el primer periodo de tono por una ventana triangular w[n]

$$w[n] = 1 - \frac{n}{P - 1} \qquad (4.30)$$

El segundo periodo de tono se multiplica por la ventana triangular 1 - w[n]. Los dos periodos resultantes se suman para generar la forma de onda comprmida. Debido a la manera como se comprime la forma de onda, la continuidad de la misma no se altera, ya que el principio y el final del nuevo segmento son iguales al segmento original.

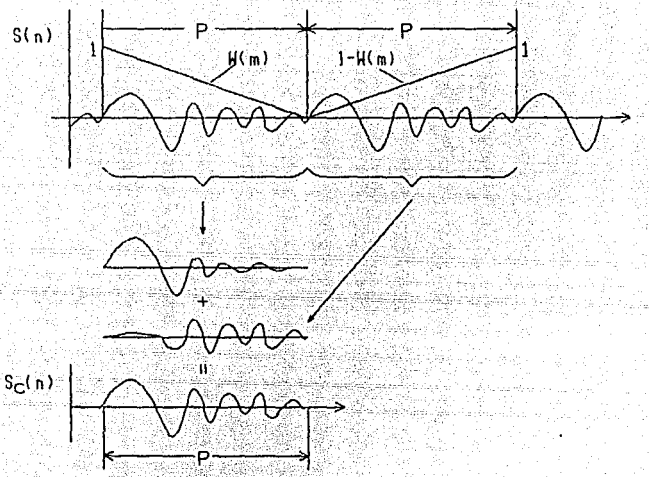


Figura 4.12 Compresion utilizando EADT

En el receptor se realiza la expansión, como se muestra en la figura 4.13. La ventana w[n] se aplica a dos periodos de tono consecutivos de la señal comprimida, y lo mismo se hace con la ventana 1 - w[n] . Sin embargo, las dos ventanas se traslapan sobre el periodo de tono que se quiere expander. Los dos segmentos se suman para generar un bloque de 2P muestras de señal expandida.

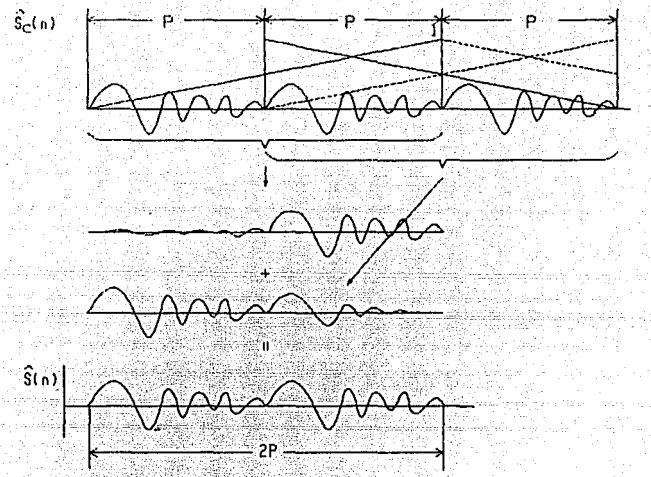


Figura 4.13 Expansión utilizando EADT.

Esta técnica de escalamiento, a pesar de haber sido utilizada hace tiempo, no es muy conocida, actualmente comienza a explotarse para la compresion de señales de voz. Existen comercialmente en los Estados Unidos tarjetas que realizan compresión de voz a tasas de 9600 bits por segundo, utilizando esta técnica.

V. SISTEMA DE COMPRESION.

V.1. ALGORITMO DE COMPRESION.

Una vez realizada la revisión bibliográfica de las diferentes técnicas de compresion de señales en general, y en particular de señales de voz, se propone una técnica híbrida de compresión, basada en la transformada de Fourier de la señal, escalamiento armónico en el dominio del tiempo, codificación por predicción lineal y cuantización vectorial, la cual permite obtener buena calidad en la señal recibida, con tasas de transmision menores a 9600 bits por segundo.

Se mencionó en el capitulo anterior, que las técnicas que codifican a las señales de voz, para su transmisión a tasas menores a 9600 bits por segundo, son las conocidas como codificadores de voz (Voice coders), cuya calidad en la señal reproducida es mala, pues se entiende el mensaje pero no se puede identificar a la persona que habia; y por otro lado, que existen técnicas, llamadas de codificación de forma de onda,

que presentan una buena calidad en la señal en el receptor, pero la tasa de transmisión no es menor a 9600 bps; por ejemplo PCM, que utiliza 64 Kbits por segundo.

De acuerdo con lo anterior, si lo que se desea es un sistema de compresión para tasas de transmisión menores a 9600 bps lo indicado es utilizar un esquema del tipo de los codificadores de voz. Por otro lado, si lo importante es mantener una buena calidad en la señal recibida , el esquema a seleccionar es un esquema de codificación de forma de onda. La idea de combinar las técnicas de codificadores de voz con las de forma de onda, presenta una solución a mantener un buen nivel de calidad en la señal recibida mientras se disminuye la tasa de transmisión, de ahí, que en el sistema que se presenta se utilicen de alguna forma, la combinación de técnicas de forma de onda y codificación de voz.

Se decidió utilizar la cuantización vectorial, para lograr una reducción en la tasa de transmisión y aprovechar su mejor representación de las señales, así como la transformada de Fourier de la señal y una aproximación del espectro de la señal, obtenido a partir de la codificacion por predicción lineal. En [21] se menciona el desempeño de la cuantización vectorial.

Antes de describir el algoritmo de compresión, se mencionará brevemente la forma como se llegó a él. El primer esquema fué obtener la transformada de Fourier de la señal, para su codificación por cuantización vectorial; la asignación de bits era constante para cada marco de señal de voz; en este caso, para obtener una buena calidad en

la señal recibida, se requeria una tasa de transmision superior a 9600 bits por segundo.

Un intento por mejorar el esquema anterior fue modificar la distribucion de bits, tratando ahora de realizar la codificación por bandas, esto es, dividir las componentes del espectro en grupos, y a cada grupo asignar un numero fijo de bits, tratando de hacer un mejor trabajo en la codificación de las bajas frecuencias, debido a que se considera que la mayor parte de la información de una señal de voz se encuentra en estas frecuencias. Con esta modificación mejoró la calidad de la señal de voz en el receptor, sin embargo, la tasa de transmisión para obtener buena calidad continuó siendo alta.

Entonces se decidió utilizar el algoritmo de escalamiento armónico en el dominio del tiempo, el cual, como se menciomó en el capitulo anterior, realiza una reducción de la información en un 50% . Para aplicar éste, era necesario tener la información sobre el periodo de tono, de tal forma, que primero se implantaron dos algoritmos de detección de periodo de tono: el método de filtrado inverso y el de diferencia de magnitud promedio, los cuales se mencionaron en el capitulo anterior. Al escuchar la señal, resultado de aplicar este algoritmo a una señal de voz, utilizando los dos metodos de deteccion de periodo de tono, se notó que la señal no habia perdido su información, esto es, es perfectamente entendible y se puede identificar a la persona que habla, la diferencia radica en que se escucha como si la persona estuviera hablando muy rapido. Al aplicar el algoritmo inverso, no se notó pérdida de calidad con respecto a la señal original, por lo que, el

siguiente paso fué incluirlo como primer algoritmo en el esquema de compresion. El resultado a esto fue una importante reduccion en la tasa de transmision, pues ahora solo se requiere enviar la mitad de la información que se enviaba al receptor. Algo importante fué, que el trabajar con la señal de voz ya comprimida en el tiempo, con el algoritmo de transformada de Fourier y cuantización vectorial, no degradó sensiblemente la calidad de la señal en el receptor con respecto al esquema anterior.

Observando los resultados obtenidos con la técnica de escalamiento armónico en el tiempo, y la codificación por bandas de la transformada de Fourier, el paso siguiente fue trabajar en el mejoramiento de ambos algoritmos.

Revisando el algoritmo de escalamiento armónico en el dominio del tiempo, surgio la siguiente idea: de acuerdo con el algoritmo, si existen dos segmentos de señal de P muestras, que en base a una medida de diferencia se parecen mucho, estos dos segmentos se pueden representar por uno solo de P muestras, donde P es el periodo de tono; puede ocurrir en ocasiones que sean más de dos, los segmentos que se parezcan, esto debido a que la señal de voz es periódica en ciertos intervalos de tiempo, en los cuales pueden estar nP muestras y tal vez se pueden reducir un numero mayor a 2 periodos de tono a uno solo. Se probó ésto, encontrando que pocas veces ocurrian más de tres periodos de P muestras que se parecian, pero que tres periodos si ocurren considerablemente.

Se hicieron pruebas con los dos esquemas de detección de periodo de tono, obteniendo los mismos resultados, en eso momento se tomó la decision de utilizar para la deteccion del periodo de tono el esquema de diferencia de magnitud promedio, pues es más facil de implantar que el de filtrado inverso, y observando que en el caso del algoritmo de escalamiento armonico no es relevante la precision de la deteccion del periodo de tono, lo cual concuerda con la literatura.

Se probaron diferentes medidas para la determinación de si un periodo de P muestras se parece al siguiente segmento de P muestras, en el algoritmo de detección de periodo de tono.

Otro detalle referente a ésto, es que el algoritmo propone la multiplicación de las muestras de la señal por una ventana triangular, en el proceso de reduccion de dos, a un segmento de P muestras, como se explicó en el capitulo anterior; pero ahora hay momentos en los que hay que realizar una reducción de tres segmentos de P muestras a uno sólo. ¿ qué tipo de ventana es la adecuada para este caso ?. Se probaron diferentes formas de ventanas: triangulares, trapezoidales; finalmente, se decidio utilizar una triangular similar a la utilizada en la reducción de dos a uno, buscando que mantenga la continuidad de la señal de voz, pues se considera que la presencia de discontinuidades en la señal en el receptor decrementa la calidad de la misma. También se probaron diferentes ventanas en la parte de expansion de este algoritmo, con el fin de encontrar la que ayudara a obtener una mejor calidad en la señal recibida.

El algoritmo final de escalamiento armonico en el dominio del tiempo se detallará mas adelante, tanto la parte de reducción como la de expansión.

Se definieron también algunos parametros en cuanto a la obtención de la transformada de Fourier de la señal, como el orden de la transformada, sobre cuantos puntos se calcularia, si se utilizaria algún tipo de ventana y si existiria traslape entre ellas. En cuanto al orden de la transformada de Fourier utilizando la TRF, sabemos que mientras mayor es el orden de la transformada, se tienen un mayor numero de componentes a codificar, por otro lado, se desea tener un buen numero de componentes para obtener una mejor aproximación del espectro; existe pues un compromiso, se seleccionó el orden igual a 7, con lo cual se tienen 65 componentes de amplitud y 65 componentes de fase. Se utilizó una ventana trapezoidal de 128 muestras con traslape de 8 muestras entre marcos de análisis.

En cuanto a la asignación de bits para la codificación de la señal, debido a que al modificar la forma de asignación de bits, se modificó la calidad de la señal resultante, se hicieron diferentes pruebas buscando la mejor forma de codificar las componentes de la transformada de Fourier, algunas son las siguientes: dar un mayor numero de bits a las amplitudes y menos a las fases, en grupos de cuantas componentes dividir el espectro de la señal, codificar directamente las componentes del espectro o alguna relación a partir de ellas y diferentes medidas de distorsión para la cuantización vectorial. A continuación se harán algunos comentarios respecto a estas pruebas.

Como se mencionó en el capitulo III, para la utilizacion de la cuantizacion vectorial, es necesario disponer del alfabeto, pues con el se realizará la cuantización. Practicamente para todas las pruebas mencionadas, se crearon alfabetos ya que no es el mismo para cada caso, tampoco es el mismo cuando se utilizan en la cuantización diferentes medidas de distorsion. La forma de crear los alfabetos se detallara cuando se describa el algoritmo final.

En cuanto a codificar directamente las componentes del espectro de amplitud o alguna relación a partir de ellas, se probó el codificar una relación entre el espectro de amplitud obtenido a partir de la TRF y el espectro de amplitud obtenido a partir de la representación por prediccion lineal, ambos normalizados; la relación consistió en calcuar el cociente de cada componente del espectro de amplitud normalizado obtenido por TRF entre cada componente del espectro de amplitud obtenido por predicción lineal, también normalizado. Esto se probó debido a que mientras que con la TRF se obtiene una aproximación del espectro de amplitud de la señal, el espectro de amplitud obtenido a partir de la predicción lineal es una aproximación de la respuesta en frecuencia del sistema de producción de voz, entonces, el cociente de ambos espectros, representa una medida de que tan parecidos son ambos. El codificar esta relación podria tener beneficios, pues su rango dinámico es menor que el de las componentes de amplitud directamente, sólo que en este caso se requeria realizar el cálculo de los coeficientes del sistema de predicción lineal, así como la ganancia. Como el resultado obtenido fue un aumento en la calidad de la señal se incluyó dentro del sistema de compresión. Para calcular los coeficientes del sistema de predicción

lineal se utilizó el algoritmo de Levinson Durbin, con el cual, al mismo tiempo que calcula los coeficientes a_i, se obtienen los coeficientes k_i y la ganancia del sistema. Con estos coeficientes y la ganancia, se puede obtener el espectro de amplitud del sistema de producción de voz para ese marco de análisis, como se mencionó en el capítulo anterior.

Ahora es necesario transmitir información acerca del espectro de amplitud obtenido a partir de los coeficientes de predicción lineal. Para ello se decidió cuantizar vectorialmente las componentes de la correlación, pues con ello, tenemos un conjunto de espectros de amplitud finito, que solo se deben calcular una vez. En cuanto al cálculo de la correlación de la señal, ésta se realiza para segmentos de 128 elementos, segmentando la señal utilizando ventanas de tipo Hamming con traslape de 8 muestras de cada lado, de la misma forma que en el caso de la ventana trapezoidal en el cálcuo de la TRF. Se utiliza orden 11 en las correlaciones, para con ellas calcular los coeficientes de predicción lineal con orden 10, que es el orden del sistema de predicción lineal que se utiliza más frecuentemente cuando se trabaja con voz. En Estados Unidos se tiene estandarizado un sistema de codificación de señales de voz, basado en predicción lineal, en el cual se utiliza orden 10 para los coeficientes del sistema predictor.

Como consecuencia de tener como primer algoritmo, en el esquema de compresión de voz, al de escalamiento armónico en el dominio del tiempo, con la modificación que se ha mencionado para cada bloque de análisis, en la transformada de Fourier o en las correlaciones, las 128 muestras requeridas para realizar dichos cálculos, representan un mayor número de

muestras. Para conocer el número total de bits que corresponden a un segmento en particular, se debe conocer el numero total de muestras que están representadas en ese segmento. Para calcular el numero total de muestras representadas en cada segmento, se utiliza la información del periodo de tono.

Conociendo el numero de bits para cada bloque, se distribuye un porcentaje para amplitudes y otro para fases. La distribución de bits para amplitudes, se realizó dependiendo del porcentaje que ocupa del total, cada componente de amplitud del espectro obtenido por TRF. Sin embargo, se obtuvieron mejores resultados haciendo la misma asignación empleando las componentes del espectro de amplitud obtenido a partir del modelo de predicción lineal, en lugar de utilizar las componentes de la TRF.

En cuanto al porcentaje de bits para codificar amplitudes y el porcentaje para fases, después de varias pruebas se fijó en 50% a cada una, ya que no se encontró una razón para hacerlo de otra forma.

Hasta aquí, se ha mencionado brevemente el trabajo realizado para la obtención del esquema final de compresión, el cual detallamos a continuacion.

DETECCION DEL PERIODO DE TONO Y ESCALAMIENTO ARMONICO EN EL DOMINIO DEL TIEMPO.

Para la detección del periodo de tono se utiliza un algoritmo similar al de diferencia de magnitud promedio.

El periodo de tono se busca entre 30 y 128 muestras, pues después de diferentes pruebas se encontró que éste pocas veces está fuera de este intervalo. En [4] se muestra una gráfica del periodo de tono para una oración, en la cual los límites se encuentran entre 50 y 90 muestras.

El algoritmo verifica si son tres los periodos de tono que se parecen, en cuyo caso los representará por un sólo periodo.

La detección se realiza de la siguiente forma: (x[n] es la señal de voz)

- Se calculan las siguientes variables

$$SUMA1[k] = \sum_{i=n}^{k+n-1} | x[i] - x[i+k] | \qquad \text{para } k = 30,\ldots,128$$

$$SUMA2[k] = \sum_{i=n}^{k+n-1} | x[i+k] - x[i+2k] | \qquad \text{para } k = 30,\ldots,128$$

$$A[k] = \sum_{i=n}^{k+n-1} | x[i] | \qquad \text{para } k = 30,\ldots,128$$

$$B[k] = \sum_{i=n}^{k+n-1} | x[i+k] | \qquad \text{para } k = 30,\ldots,128$$

$$C[k] = \sum_{i=n}^{k+n-1} | x[i+2k] | \qquad \text{para } k = 30,\ldots,128$$

(5.1)

- Se efectúan las siguientes normalizaciones :

$$SUMA1[k] = \frac{2\ SUMA1[k]}{A[k] + B[k]} \qquad \text{para } k = 30,\ldots,128$$

$$SUMA2[k] = \frac{2\ SUMA2[k]}{B[k] + C[k]} \qquad \text{para } k = 30,\ldots,128$$

(5.2)

- Se encuentra para cada variable (SUMA1 y SUMA2) su valor mínimo y en que posición ocurre.

MIN1 = valor mínimo (SUMA1[k]) para k = 30,...,128

PO1 = k1 donde k1 es la posición donde ocurrió el mínimo

MIN2 = valor mínimo (SUMA2[k]) para k = 30,...,128

PO2 = k2 donde k2 es la posición donde ocurrió el mínimo

-A continuación se determina si existe periodo de tono. Si existe, se verifica si es el mismo para dos o tres periodos.

Si MIN1 ≥ 1.25 entonces no hay periodo de tono.P0=0

Si MIN1 < 1.25 entonces

Si $|PO1 - PO2| \leq 3$ entonces

$$P0 = \frac{PO1 + PO2}{2}$$

NDS = 1

(Existen tres periodos de P0 muestras)

Si $|PO1 - PO2| > 3$ entonces

P0 = PO1

NDS = 0

(Existen dos periodos de P0 muestras)

El valor de umbral de 1.25, se determinó después de varias pruebas con diferentes señales de voz, buscando el valor para el cual; una medida de distorsión de error cuadrático medio entre las señales, tomaba su valor mínimo.

Este algoritmo genera dos datos, P0 y NDS;

- P0 = 0 indica que no hay periodo de tono,
- P0 ≠ 0, NDS = 0 indica que existen dos periodos de P0 muestras que se parecen,
- P0 ≠ 0, NDS = 1 indica que existen tres periodos de P0 muestras que se parecen.

Con la informacion sobre el periodo de tono, se puede aplicar el algoritmo de escalamiento armonico en el dominio del tiempo, al cual como se menciono, se le hizo una modificacion para hacer reducciones de tres periodos a uno o no hacer reduccion.

Si el algoritmo de detección de periodo de tono le indica que en el segmento en análisis no encontró periodo de tono (P0=0), solamente recorre el índice de muestras de la señal 32 muestras y regresa al algoritmo de detección de tono para otro segmento.

Si el algoritmo de detección de periodo de tono indica que sí encontro periodo de tono (P0≠0), entonces

- Si NDS = 0

$$y[n] = x[i] * \left(\frac{M - i}{P0 - 1} + 1\right) + x[i+P0] * \left(\frac{i - M}{P0 - 1}\right)$$

para i = M,..,P0 + M - 1 donde M es la posición de inicio del segmento en análisis.

Se toma un nuevo segmento de voz, a partir de la última muestra considerada para el cálculo anterior y se regresa al algoritmo de detección de periodo de tono.

- Si NDS = 1

$$y[n] = x[i] * F1 + x[i+P0] * F2 + x[i+2P0] * F3$$

donde

$$F1 = \begin{cases} \dfrac{M - i}{\dfrac{P0}{2} - 1} + 1 & \text{si } i-M+1 \leq \dfrac{P0}{2} \\ 0 & \text{c.c.} \end{cases}$$

$$F2 = \begin{cases} \dfrac{i - M}{\dfrac{P0}{2} - 1} & \text{si } i-M+1 \leq \dfrac{P0}{2} \\ \dfrac{M - i + \dfrac{P0}{2}}{\dfrac{P0}{2} - 1} + 1 & \text{c.c.} \end{cases}$$

$$F3 = \begin{cases} 0 & \text{si } i-M+1 \leq \dfrac{P0}{2} \\ \dfrac{i - M - \dfrac{P0}{2}}{\dfrac{P0}{2} - 1} & \text{c.c.} \end{cases}$$

(5.3)

son los ventanas por las que se multiplica la señal, para realizar la reducción de tres periodos a uno.(Véase la figura 5.1). Para i = M, ..., P0+M-1, donde M es la posición de inicio del segmento en análisis.

Se toma un nuevo segmento de señal de voz a partir de la última muestra utilizada para el cálculo anterior y se repite el algoritmo.

En la figura 5.1 se muestra la forma como se hace la reducción de tres periodos de señal a uno . Para pasar de dos periodos a uno se hace

de la misma forma que se describió en el algoritmo de EADT en el capitulo anterior (figura 4.12).

La operación de estos dos algoritmos, genera dos resultados básicamente: una señal de voz resultado de la reducción de aquellos periodos de señal muy parecidos y el periodo de tono de cada segmento presente en la señal obtenida, además de la indicación del número de periodos de tono reducidos.

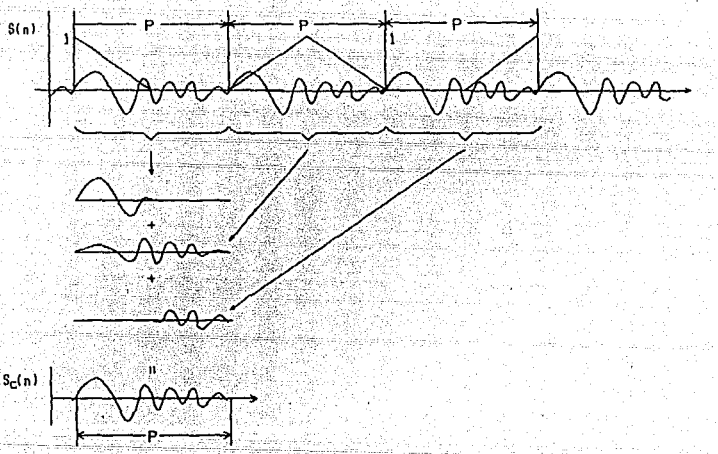


Figura 5.1 EADT Reducción 3:1.

CALCULO DE CORRELACIONES.

Para el calculo de las correlaciones de la señal resultado del primer paso de compresión, se multiplica la misma por una ventana de Hamming de la forma

$$w[n] = \begin{cases} 0.54 - 0.46 \cos \frac{2\pi n}{N} & 0 \le n \le N-1 \\ 0 & \text{c.c.} \end{cases} \qquad (5.4)$$

Se considera una ventana de 128 muestras con traslape de 8 muestras, por ejemplo: se toman de la muestra 113 a la 241, se multiplica por la ventana de Hamming y se calcula de estas 128 muestras su correlación; el siguiente bloque de señal a considerar seria de la muestra 226 a la 354.

En este algoritmo se calcula la autocorrelación de la señal, pues a partir de ella se obtendrán los coeficientes del sistema de predicción lineal. La autocorrelación se calcula de la siguiente manera :

$$r(i) = \sum_{k=0}^{N-i-1} x[k]\, x[k+i] \qquad (5.5)$$

Se calculan 11 correlaciones para cada segmento, debido a que el algoritmo que determina los coeficientes del sistema de predicción lineal, requiere una correlación más que el orden del sistema, el cual generalmente es 10.

CALCULO DE LA TRANSFORMADA DE FOURIER DE LA SEÑAL.

En este caso se emplea el algoritmo de la Transformada Rápida de

Fourier ya descrita. Se utiliza la técnica de inversión de bits, para ordenar las muestras de entrada, de tal forma que las componentes del espectro que se obtienen estén ordenadas.

Se utiliza también el multiplicar la señal resultado de la primera compresión por una ventana, en este caso de tipo trapezoidal. La ventana es de 128 muestras y sus valores corresponden a

$$w[n] = \begin{cases} \dfrac{n}{M+1} & 1 \leq n \leq M \\ 1.0 & M+1 \leq n \leq N-M \\ 1.0 - \dfrac{n-N+M}{M+1} & N-M+1 \leq n \leq N \end{cases} \qquad (5.6)$$

donde M es el número de muestras de pendiente, que en nuestro caso fueron 16. Se utiliza traslape de 8 muestras, entre ventana y ventana (figura 5.2). Se seleccionó orden 7 para el cálculo de la TRF, disponiendo de esta forma de 65 componentes de amplitudes y 65 de fase.

Para el calculo de la TRF se utiliza la rutina que se encuentra en [15], debida a Cooley et.al.; la cual recibe un vector complejo con las muestras de entrada, las ordena de acuerdo con la inversión de bits y calcula la transformada, regresando en el mismo vector complejo las componentes del espectro .

Se descompone el vector complejo en amplitud y fase, ya que de esta forma es como se va a cuantizar el espectro de la señal.

Después de ejecutar las rutinas mencionadas anteriormente, se tiene

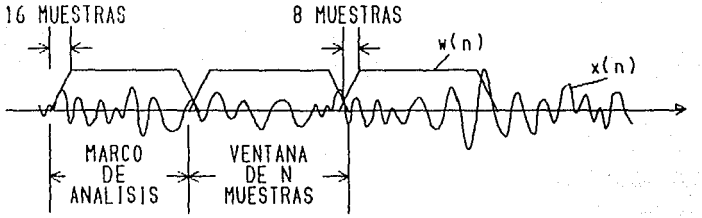


Figura 5.2 Segmentación de la señal.

la información que hay que codificar para su transmisión, de tal forma que se pueda recuperar la señal original en el receptor. La información que se necesita enviar es la siguiente :

- El periodo de tono,
- el indicador de número de periodos de tono reducidos por cada periodo de tono,
- las correlaciones de cada bloque de 128 muestras, y
- las componentes de amplitud y fase de cada bloque de 128 muestras, además de la suma de las amplitudes de la TRF.

Para la codificación de las correlaciones, las componentes del espectro de amplitud y fase, y la sumatoria de las amplitudes de la TRF se utiliza la cuantización vectorial, que como se mencionó en el capitulo III, requiere de la existencia de los alfabetos. Para el periodo de tono se utilizan los códigos de Huffman.

CODIFICACION DEL PERIODO DE TONO.

Huffman desarrolló un procedimiento para codificar fuentes estadísticamente independientes, de tal forma de obtener una mínima longitud de palabra en promedio. Este código tiene la propiedad de decodificación instantánea. El procedimiento de codificación es el siguiente : [22]

1. Se calcula la probabilidad de ocurrencia de cada símbolo a transmitir y se ordenan de mayor a menor.
2. Se suman las dos probabilidades menores, el resultado sustituye a ambas y se ordena nuevamente el conjunto de probabilidades. Este procedimiento se repite hasta que la suma de las probabilidades menores dé como resultado la unidad. Esto ocurrirá cuando sólo se tengan dos probabilidades.
3. Se asigna un cero al primer elemento de las últimas dos probabilidades y a la otra uno.
4. De acuerdo a como se fueron sumando las probabilidades, se asigna a cada sumando la secuencia de bits del resultado y se agrega además un cero a la más alta y uno a la más baja; este procedimiento se repite hasta conocer la secuencia asignada a cada probabilidad del conjunto original.

Este procedimiento se ilustra en la figura 5.3.

Utilizando este procedimiento se realizó la codificación del periodo de tono. Se tomaron 8400 muestras de periodos de tono, los cuales se obtuvieron de tres señales de voz, a partir de ellas se calculó la probabilidad de ocurrencia de cada periodo de tono y se aplicó el procedimiento. Se debe recordar que el periodo de tono se

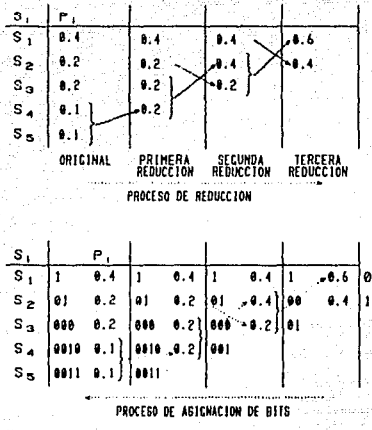


Figura 5.3 Ejemplo del proceso de codificación de Huffman.

encuentra entre 30 y 128 muestras, de acuerdo con el algoritmo de detección del periodo de tono, y puede tomar el valor de 0 cuando no hay periodicidad en la señal. En la tabla 5.1 se muestra para cada periodo de tono, su probabilidad de ocurrencia y su palabra de código correspondiente.

Se dispone de esta forma de la palabra de código para cada periodo de tono, en el caso de que el periodo de tono sea diferente de cero, se agrega un bit a cada palabra de código, para indicar si corresponde a una reducción 2:1 (0) o a una reducción 3:1 (1).

TABLA 5.1 CODIFICACION DEL PERIODO DE TONO

PERIODO DE TONO	PROBABILIDAD	PALABRA DE CODIGO
0	3.6904763E-02	00100
30	3.0833334E-02	01110
31	2.6309524E-02	11000
32	2.9642858E-02	10001
33	3.2023810E-02	01011
34	2.5952380E-02	11010
35	2.4880953E-02	11101
36	2.0357143E-02	000100
37	1.9404761E-02	000110
38	1.6904762E-02	010001
39	1.7261906E-02	001110
40	1.3809524E-02	101010
41	1.2023809E-02	111111
42	1.3809524E-02	101011
43	1.5476190E-02	011010
44	1.2976190E-02	110110
45	1.3690476E-02	101101
46	1.1428571E-02	0000100
47	1.5595238E-02	011001
48	1.5238095E-02	011111
49	1.4523810E-02	100101
50	1.4523810E-02	100110
51	1.6666668E-02	010010
52	1.5119048E-02	100001
53	1.3095238E-02	110010
54	1.2142858E-02	111110
55	1.1666667E-02	0000010
56	1.3690476E-02	101110
57	1.4285714E-02	100111
58	1.5000000E-02	100100
59	1.2857143E-02	110111
60	1.5476190E-02	011011
61	1.2023809E-02	0000000
62	1.5476190E-02	011110
63	1.6666668E-02	010011
64	1.7619047E-02	001100
65	1.7500000E-02	001101
66	1.8095238E-02	001011
67	1.8333333E-02	001010
68	1.5952380E-02	010101
69	1.7023809E-02	010000
70	1.3095238E-02	110011
71	1.4166667E-02	101000
72	1.2857143E-02	111000
73	1.3809524E-02	101100
74	1.2023809E-02	0000001
75	1.2261905E-02	111101
76	1.3452381E-02	101111
77	9.9999998E-03	0001010
78	1.0238095E-02	0000111
79	7.6190475E-03	1000000

TABLA 5.1 CONTINUACION

PERIODO DE TONO	PROBABILIDAD	PALABRA DE CODIGO
80	9.8809525E-03	0001011
81	8.0952384E-03	0110000
82	8.3333338E-03	0101000
83	6.5476191E-03	1010011
84	8.4523810E-03	0011111
85	5.5952379E-03	00000111
86	6.1904760E-03	1111000
87	6.0714288E-03	00000110
88	5.3571430E-03	00001011
89	4.8809522E-03	00011101
90	3.0952380E-03	11100111
91	4.1666669E-03	00111101
92	3.6904763E-03	10100100
93	3.5714286E-03	10100101
94	3.8095238E-03	01100011
95	2.3809525E-03	000111001
96	3.2142857E-03	11100101
97	2.3809525E-03	000111110
98	2.7380954E-03	000010100
99	2.4999999E-03	000111000
100	2.6190476E-03	000011010
101	3.2142857E-03	11100110
102	2.6190476E-03	000011011
103	3.0952380E-03	11110010
104	2.7380954E-03	000010101
105	3.0952380E-03	11110011
106	2.3809525E-03	000111111
107	1.1904762E-03	0011110000
108	1.3095238E-03	0000110010
109	1.3095238E-03	0000110011
110	1.9047619E-03	011000101
111	2.0238096E-03	010100101
112	1.0714285E-03	0011110001
113	2.3809525E-03	000111100
114	2.3809525E-03	000111101
115	2.0238096E-03	010100110
116	1.9047619E-03	100000110
117	9.5238094E-04	1000001010
118	9.5238094E-04	1000001011
119	1.0714285E-03	0101001000
120	1.6666667E-03	111001000
121	1.6666667E-03	111001001
122	1.9047619E-03	100000111
123	2.0238096E-03	010100111
124	1.9047619E-03	100000100
125	1.0714285E-03	0101001001
126	2.1428571E-03	001111001
127	2.0238096E-03	011000100
128	2.6190476E-03	000011000

CODIFICACION DE LAS CORRELACIONES.

Como se mencionó anteriormente, la información a codificar es una relación entre el espectro de amplitud obtenido con la TRF y el mismo a partir de predicción lineal. El espectro de amplitud por TRF se ha explicado detalladamente en el inciso anterior. Para obtener el espectro de amplitud por predicción lineal, es necesario calcular primero los coeficientes del sistema de predicción lineal, para a partir de la función de transferencia del filtro inverso calcular la respuesta en frecuencia del sistema. Para el cálculo de los coeficientes del sistema de predicción lineal, se utiliza el algoritmo de Levinson Durbin. Este algoritmo requiere como datos de entrada las correlaciones de la señal, de ahí que primero se realice ese cálculo. Con éstas pueden calcularse los coeficientes del sistema de predicción lineal, a partir de los cuales se obtiene el espectro de amplitud. Con el objetivo de simplificar la operación del sistema se decidió cuantizar las correlaciones , ya que haciendo ésto, se tiene un número finito de conjuntos de coeficientes de predicción lineal, y por lo tanto, un numero finito de espectros de amplitud. En operación, el sistema procede a cuantizar las correlaciones y obtiene como resultado directamente el espectro de amplitud por predicción lineal, ya que se dispone de un alfabeto de espectros.

Para la codificación de las correlaciones se utiliza la cuantización vectorial, primero mencionaremos la generación del alfabeto de espectros de amplitud y posteriormente la codificación .

A su vez, para la generación del alfabeto de espectros, primero se

obtiene el alfabeto de correlaciones, a partir del cual se obtienen los coeficientes del sistema de predicción lineal y de estos, el espectro.

Para la generación del alfabeto de correlaciones, se obtuvieron las mismas de tres señales de voz, las cuales previamente fueron procesadas de acuerdo con el algoritmo aquí propuesto, esto es, el algoritmo de escalamiento armónico en el dominio del tiempo. Al proceso de obtención de los alfabetos se le conoce como entrenamiento.

Detallamos a continuación el proceso de entrenamiento en general, ya que se utiliza tanto para correlaciones como para amplitudes, fases, y sumatoria de amplitudes en el sistema de compresión. En cada caso se harán aclaraciones particulares.

1. Se establece el número de niveles máximo deseado $N = 2^R$, donde R es un entero. Se fijan también, el valor de K, que corresponde al numero de componentes de cada vector, n , que es la longitud de la secuencia de entrenamiento. ε , el umbral de mínima distorsión y se inicializa un contador M con 1, donde M indica el nivel actual.

2. Dada la secuencia de entrada $\{x_j ; j = 0, ..., n-1\}$, se define $A = \{x_j ; j = 0, ..., n-1\}$ como el alfabeto de la secuencia de entrenamiento. Se define también $\hat{A}(1) = \hat{x}(A)$, como el centroide de la secuencia de entrenamiento. El centroide se determina de acuerdo con una medida de distorsión. En el caso de la medida de distorsión de error cuadrático es el centroide Euclideano o la suma vectorial de todos los vectores de entrada codificados en un símbolo dado, y está dado por

$$x(A) = \frac{1}{||A||} \sum_{i=1}^{n} x_i \qquad (5.7)$$

donde $||A||$ denota al número de vectores de entrenamiento.

3. División (splitting). Dada $\hat{A}(M) = \{y_i,\ i = 1, \ldots, M\}$, se divide cada vector y_i en $y_i + \varepsilon$ e $y_i - \varepsilon$, donde ε es un vector de perturbación fijo. Se define $\hat{A}_0(2M) = \{y_i + \varepsilon, y_i - \varepsilon,\ i = 1, \ldots, M\}$ y se remplaza M por 2M.

4. Se inicializa m=0 y $D_{-1} = \infty$.

5. Dado $\hat{A}_m(M) = \{y_1, \ldots, y_M\}$, se encuentra su partición óptima $P(\hat{A}_m(M)) = \{ S_i;\ i = 1, \ldots, M \}$, esto es; $x_j \in S_i$ si $d(x_j,y_i) \leq d(x_j,y_l)$, para toda l.

6. Se calcula la distorsión resultante

$$D_m = D(\hat{A}_m(M), P(\hat{A}_m(M))) = n^{-1} \sum_{j=0}^{n-1} \min_{y \in A_m} d(x_j,y) \qquad (5.8)$$

7. Si $(D_{m-1} - D_m)/D_m \leq \varepsilon$, entonces paso 9. En caso contrario continua.

8. Se encuentra el alfabeto de reproducción óptimo $\hat{A}_{m+1}(M) = \hat{x}(P(\hat{A}_m(M))) = \{\hat{x}(S_i);\ i = 1, \ldots, M\}$ para $P(\hat{A}_m(M))$. Se remplaza m por m+1 y se repite desde el paso 5.

9. Se hace la asignación $\hat{A}(M) = \hat{A}_m(M)$. El cuantizador final de M niveles queda descrito por $\hat{A}(M)$. Si M = N , termina, con el cuantizador final $\hat{A}(N)$. En caso contrario repite desde el paso 3.

Este algoritmo se describe en [13] y [23], mientras que un análisis detallado de la cuantización vectorial en general, se encuentra en

[12] y [21].

Para el caso del alfabeto de correlaciones, el algoritmo se aplica de la siguiente manera :

Se utiliza un cuantizador de 9 bits, es decir, N=512, K=11, pues es el numero de correlaciones que tenemos, n=20000, y ε = 0.01. Se utiliza el error cuadrático medio como medida de distorsion para la separación del espacio

$$d(x,\hat{x}) = \sum_{i=0}^{k-1} |x_i - \hat{x}_i|^2 \qquad (5.9)$$

Con los datos anteriores se inicializa el algoritmo, con una modificación, en la etapa de división (splitting), en lugar de perturbar cada componente del vector de correlaciones que resultó ser el centroide, se perturban los coeficientes de reflexión obtenidos a partir de este vector de correlaciones utilizando Levinson. Una vez que se tienen los nuevos vectores de coeficientes de reflexión k_i, se calculan los coeficientes a_i, a partir de los k_i y a continuación se calcula la correlación de los coeficientes a_i; de esta forma tenemos los dos vectores de correlación para continuar con el algoritmo. [24]

Los coeficientes de reflexión y la ganancia, se obtienen al mismo tiempo que los coeficientes a_i, utilizando el método de Levinson. Para pasar de los coeficientes k_i a los coeficientes a_i se utiliza la siguiente relación : [8]

$$a_i^{(i)} = K_i$$

$$a_j^{(i)} = a_j^{(i-1)} + K_i\, a_{i-j}^{(i-1)} \qquad \begin{matrix} i = 1,\ldots,p \\ j = 1,\ldots,i-1 \end{matrix}$$

$$a_0 = 1 \qquad (5.10)$$

Para calcular la correlación de los coeficientes a_i se utiliza la misma rutina que para la correlación de la señal.

Una vez generado el alfabeto de correlaciones, se puede calcular el alfabeto de espectros. Para ello, se utiliza la rutina de TRF, solo que, como se menciono en el capitulo anterior, se calculan los coeficientes a_i a partir de las correlaciones, se forma un vector de 128 componentes, donde las primeras 10 son los coeficientes a_i y el resto se llena con ceros, y se utiliza la TRF. Para obtener el espectro de amplitud se divide la ganancia entre la magnitud de cada componente. Se tienen así dos alfabetos, el de correlaciones y el de espectros, donde para cada conjunto de correlaciones se tiene un espectro.

Estos alfabetos sólo se tienen que generar una vez, con ellos se procederá a codificar cualquier señal.

La codificación de las correlaciones se hace de la siguiente manera :

Para cada vector de correlaciones de orden 11, se busca el vector del alfabeto cuya distancia euclidiana al vector a codificar, es la menor de todas las distancias del vector a codificar a los vectores del

alfabeto. El indice que apunta al vector de distancia minima, indica tambien la posicion del vector de componentes de amplitud del espectro por prediccion lineal.

DETERMINACION DEL NUMERO DE BITS PARA CADA MARCO DE ANALISIS.

Para determinar el numero de bits que se tiene en cada marco de señal, se debe conocer la tasa de transmision del canal, por ejemplo 9600 bits por segundo, asi como la frecuencia de muestreo que se utilizó para la conversión de la señal analógica a digital, por ejemplo 8000 muestras por segundo. El cociente de las anteriores, determina el numero de bits por muestra, continuando con el ejemplo, 1.2 bits/muestra. Conociendo el número de muestras representado en cada segmento de análisis, al multiplicarlo por el número de bits/muestra, se obtiene el numero de bits disponible para codificar ese segmento de señal.

En este sistema, debido a que se utiliza la técnica de escalamiento armónico en el dominio del tiempo, en cada segmento de 128 muestras a codificar, están representadas mas muestras, es decir, el número de muestras originales por segmento de analisis varia de segmento a segmento, por lo cual, es necesario antes de realizar la codificación de las amplitudes y fases, calcular el número de bits para cada segmento .

Para determinar el numero de muestras que están representadas en cada segmento, se dispone de la información que genera el algoritmo de detección de periodo de tono; por ejemplo, si la información fuera la siguiente :

PO = 35 NDS = 1

PO = 55 NDS = 0

PO = 0 NDS = 0

PO = 90 NDS = 0

. . .

Implica que las primeras 35 muestras representan a 105 muestras, las siguientes 55 muestras, representan a 110 muestras, ya que PO = 0 , las siguientes 32 muestras no han sufrido ninguna modificacion y finalmente, las 90 muestras siguientes representan a 180 muestras, de las cuales sólo se utilizan 6 para formar el segmento de 128 muestras de analisis. Por lo tanto, esas 128 muestras representan realmente a 259 muestras de la señal original (NOTA : NDS=1 reduccion 3:1, NDS=0 reduccion 2:1) .

No. muestras = 3*35 + 2*55 + 1*32 + 2*6 = 259

donde 35 + 55 + 32 + 6 = 128 muestras del segmento de analisis.

Tenemos 259 muestras para el primer segmento de analisis, con esos bits hay que codificar los periodos de tono involucrados en ese segmento, el indice del espectro de amplitud por prediccion lineal, la relación de amplitud de TRF con amplitud del espectro por prediccion lineal y la sumatoria de las componentes de amplitud de la TRF.

Anteriormente se mencionó la forma como se codifica el periodo de tono, siguiendo con el ejemplo :

PO = 35 NDS = 1 se codifica con 6 bits,

PO = 55 NDS = 0 se codifica con 8 bits,

PO = 0 NDS = 0 se codifica con 6 bits,

PO = 90 NDS = 0 se codifica con 9 bits.

Asi pues, para codificar el periodo de tono se requieren 29 bits,

que hay que descontar de los 259 bits disponibles. También deben descontarse 9 bits, que corresponden a la codificación de las correlaciones; finalmente, se disponen de 221 bits para la codificación de la relación de espectros, fases y sumatoria de amplitudes de la TRF.

CODIFICACION DE AMPLITUDES DE LA TRF.

Para codificar las amplitudes de la TRF, como se ha venido mencionando, se utiliza una relación entre las amplitudes de los espectros obtenidos a partir de la TRF y a partir del sistema de predicción lineal. Del número de bits disponibles hasta este momento se toman 9 bits para codificar vectorialmente la sumatoria de las amplitudes de la TRF, utilizando como medida de distorsión el error cuadrático medio y el resto se divide, 50% para amplitudes y 50% para las fases.

La relación a codificar es la siguiente :

$$y[i] = \frac{z[i] \sum_{k=1}^{N} x[k]}{\sum_{k=1}^{N} z[k] \; x[i]} \qquad i = 1,\ldots,N \qquad (5.11)$$

donde z[i] es el vector de amplitudes del espectro de la TRF y x[i] es el vector de amplitudes del espectro obtenido por predicción lineal. La distribución de bits para las amplitudes se hace de acuerdo con el porcentaje que ocupa cada componente del espectro de predicción lineal. Esto es, para la iésima componente

$$\text{No. bits [i]} = \frac{x[i] \text{ No. bits para amplitudes}}{\sum_{k=1}^{N} x[k]} \qquad i = 1,\ldots,N \qquad (5.12)$$

Para codificar la relación, se suma el numero de bits en grupos de 8 componentes, se realiza la codificación del bloque de mayor numero de bits al menor; si el numero de bits para el bloque es menor a 8 se codifican esas componentes vectorialmente, si es mayor a 8, se divide en dos grupos de 4 componentes, si la suma del numero de bits de las cuatro componentes es menor a 8, se codifica vectorialmente, si es mayor de 8, se divide en grupos de 2 componentes, si la suma del numero de bits es menor de 8, se codifica vectorialmente, en caso contrario las componentes se codifican escalarmente, permitiendo como máximo 8 bits por componente, si sobran bits se utilizan para indicar en cada palabra codificada el indice del cuantizador y el número de bits correspondiente, si no son suficientes se toman de los que restan para codificar las demás componentes, haciendo una nueva asignación de bits para las componentes que faltan. Los bits que sobren, en caso de que suceda, se distribuyen en las componentes que aún no se han codificado. Con este procedimiento se codifican todas las componentes del espectro. Como se divide el espectro en grupos con numero par de componentes, la que no se incluye en ningun grupo se codifica escalarmente con los bits que sobraron (máximo 8). Si al terminar de codificar el segmento, sobran bits, estos se agregan a los asignados para codificar las fases. [24]

De lo anterior se deriva que deben existir alfabetos de la relación de amplitudes escalar, vectorial de 2 componentes, vectorial de 4 componentes y vectorial de 8 componentes, para numero de bits variable de 0 a 8 bits.

Para la generación de estos alfabetos, se utiliza el mismo

algoritmo descrito anteriormente; en este caso no se hace ninguna modificación. Se aprovecha la caracteristica de ese algoritmo de generar sucesivamente los alfabetos para diferente numero de bits, es decir, al calcular los alfabetos para 8 bits se tuvieron que generar los alfabetos de 0 a 7 bits. Se utiliza N = 256, K = 1, 2, 4 y 8; n = 50000, ε = 0.01. La medida de distorsion es error cuadrático medio.

El proceso de codificación, es de la misma forma que para las correlaciones. Se calcula la distancia euclidiana entre el vector a codificar y los existentes en el alfabeto, aquel vector del alfabeto, cuya distancia al vector a codificar es minima, se define como el vector de reproducción. Se transmite el indice que corresponde al vector seleccionado, el tipo de cuantizador: escalar, vectorial 2, 4 u 8 componentes y el numero de bits.

CODIFICACION DE LAS FASES DE LA TRF.

A diferencia de la codificación de las amplitudes, las fases se codifican directamente, escalar o vectorialmente, en grupos de 2 y 4 componentes.

La asignación de bits a cada componente se realiza de acuerdo con el porcentaje que ocupa su correspondiente componente de amplitud ya cuantizada, en el espectro de amplitud total. A la primera y ultima componente se les asigna un bit, ya que siempre toman el valor π o $-\pi$.

De manera similar a la codificación de las amplitudes, para las fases se forman grupos de 4 componentes, obteniendo el numero de bits para cada grupo. Se codifican en orden de mayor a menor, de acuerdo con

el numero de bits por grupo. La codificación de cada grupo es como sigue: si el numero de bits es mayor a 8 se divide en dos grupos de 2 componentes, y asi sucesivamente. Cuando el número de bits no excede a 8, se cuantiza de acuerdo con el numero de componentes involucradas. Para las primeras cuatro componentes, la codificación se hace de la siguiente manera: A la primera ya mencionamos que se le asigna 1 bit, verifica si puede cuantizar las componentes 2 y 3 vectorialmente. (No bits < 8), si se puede lo hace y la cuarta la cuantiza escalarmente, si no, verifica si agrupando las componentes 3 y 4, las puede cuantizar vectorialmente, en caso afirmativo lo hace y la componente 2 la codifica escalarmente, si no ocurrió ninguna de las anteriores, las componentes 2, 3 y 4 las codifica escalarmente. En cada caso dispone de los bits que sobran o toma de los restantes para indicar el tipo de cuantizador y el numero de bits con que se codifico. [24]

En cuanto a los alfabetos, se tienen para codificación escalar, vectorial de 2 y 4 componentes. Estos se generan a partir de las componentes del espectro de TRF en forma binomica. Para emplear el algoritmo, todas las componentes de entrenamiento se normalizan dividiendo tanto la parte real como la parte imaginaria entre la magnitud de la componente. El algoritmo se aplica normalmente, en el momento de dividir, se perturba la parte real de cada componente, se normaliza y se continua con el algoritmo. [24]

Como medida de distorsión se utiliza el error cuadrático, sólo que en este caso, se calcula de la siguiente manera :

$$d(x_i,\hat{x}_i) = \sum_{i=1}^{K} \sqrt{|Re(\hat{x}_i)-Re(x_i)|^2 + |Im(\hat{x}_i)-Im(x_i)|^2} \qquad (5.13)$$

Para la codificación de las fases, de la misma forma que en los casos anteriores de correlaciones y amplitudes, se busca al vector que de acuerdo con una medida de distorsión, mejor reproduce al vector a codificar. Se utiliza como medida de distorsión la siguiente [24]

$$d(x_i,\hat{x}_i) = \sum_{i=1}^{K} (1 - \cos (x_i - \hat{x}_i))^2 \qquad (5.14)$$

Después de utilizar los algoritmos descritos anteriormente, se tiene la información a transmitir. A continuación se describirá el proceso de decodificación.

TRANSFORMADA INVERSA DE FOURIER.

El receptor recibe los periodos de tono, el índice del espectro de predicción lineal de cada segmento, la suma de las componentes de amplitud del espectro de la señal obtenido a partir de la TRF, los índices de las relaciones de amplitud y los índices de las fases.

Con la suma de amplitudes del espectro y los índices de las relaciones de amplitud, se obtienen las amplitudes del espectro y con los índices de las fases, se obtienen las fases de cada segmento. Debemos recordar que el receptor también conoce todos los alfabetos.

Con las amplitudes y fases decodificadas, se puede calcular la transformada inversa de Fourier, de tal forma de obtener la señal comprimida. Para obtener la transformada inversa, se utiliza nuevamente la rutina de TRF. La forma de hacerlo es la siguiente :

-Se forma un vector complejo con las componentes de amplitud y fase decodificadas. En el sistema son 65 componentes.

-Las componentes 66 a 128 es la parte simétrica del espectro, la componente 66 es igual a la 64, la 67 a la 63 y así sucesivamente.

-Finalmente, se obtiene el conjugado de las primeras 65 componentes, formando así un vector complejo de 128 componentes, con el cual la rutina de TRF obtiene la señal discreta original. Debido a que para el cálculo de la TRF de la señal, se utilizaron ventanas con traslape, es necesario para la recuperación de la señal, el eliminar el efecto del traslape.

ALGORITMO DE EXPANSION DE ESCALAMIENTO ARMONICO EN EL DOMINIO DEL TIEMPO.

Una vez que tenemos la señal comprimida decodificada, con la información del periodo de tono, se puede realizar la expansión de la señal. Si el periodo de tono es cero, significa que las siguientes 32 muestras no fueron reducidas; si el periodo de tono es diferente de cero y NDS=0, entonces se debe repetir este bloque dos veces, pues corresponde a una reducción 2:1; en el caso de que el periodo de tono sea diferente de cero y NDS=1, entonces el bloque de PO muestras se repite 3 veces, ya que corresponde a una reducción de 3:1.

En el caso de PO≠0 y NDS=0, se utiliza el algoritmo de expansión normal de EADT, si PO≠0 y NDS=1, el bloque de PO muestras se repite 3 veces y se utiliza una ventana diferente a la expansión 2:1. Las expresiones para la obtención de la señal original son las siguientes :

$PO=0 \qquad y[k] = x[i]$

$PO\neq 0 \quad NDS=0$

$$y[k] = x[i]\,\frac{i-M}{2PO-1} + x[i+PO]\left(\frac{M-i}{2PO-1} + 1\right)$$

$PO\neq 0 \quad NDS=1$

$$y[k] = x[i]\,\frac{i-M}{2PO-1} + x[i+PO]\left(\frac{M-i}{2PO-1} + 1\right) \qquad i = n,\ldots,n+PO$$

$$y[k+PO] = x[i+PO]\,\frac{i-M}{2PO-1} + x[i+2PO]\left(\frac{M-i}{2PO-1} + 1\right) \qquad i = n,\ldots,N+2PO$$

(5.15)

En la figura 5.4 se muestra un ejemplo de expansión 3:1.

V.2 OPERACION DEL SISTEMA Y RESULTADOS.

Una vez detallados los algoritmos que se utilizan en el sistema de compresión de señales de voz, se mencionará ahora la forma como se realizó la simulación del sistema, y los resultados que se obtuvieron.

El esquema completo de compresión en diagrama de bloques, se muestra en la figura 5.5. Se debe mencionar, que los alfabetos son los mismos, tanto para el transmisor como para el receptor.

La simulación del sistema se realizó en una computadora VAX, utilizando como lenguaje de programación FORTRAN 77. El sistema está estructurado en módulos, esto es, está formado por un conjunto de

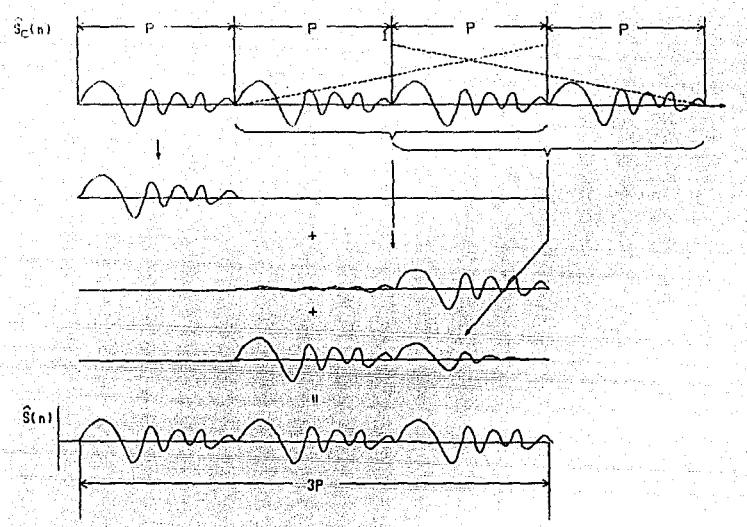


Figura 5.4 Expansión EADT 3:1

programas, donde cada uno corresponde a una etapa del esquema de compresión. Se implantó de esta forma ya que permite modificar alguno de los algoritmos, sin interferir con los demás, o incluir fácilmente otras etapas de procesamiento; ésto fue muy útil durante el desarrollo del sistema.

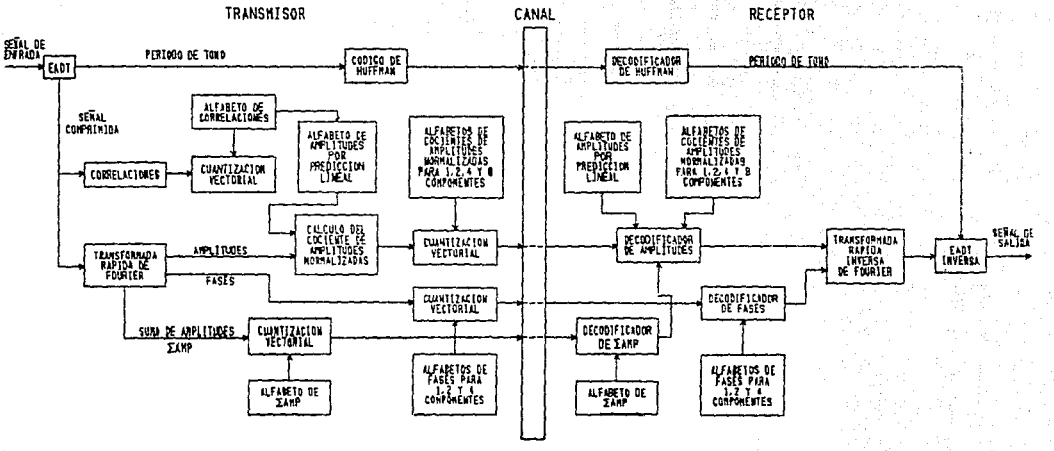


Figura 5.5. Diagrama de bloques del sistema de compresión.

La señales de voz se almacenaron en archivos en la misma computadora . Estas señales fueron tomadas de un amplificador de audio y se digitalizaron empleando una tarjeta de conversión A/D - D/A, diseñada especialmente para la adquisición de señales de voz. Esta tarjeta se conecta en un slot de una computadora del tipo PC. Las caracteristicas principales de esta tarjeta son :

- En la entrada tiene un filtro paso banda de 300 a 3300 Hz, el cual cumple con dos objetivos : limitar el ancho de banda de la señal de entrada, para efectuar adecuadamente el muestreo y para emular el efecto que produce la linea telefónica en las señales de voz.
- Se realiza la conversión A/D y D/A utilizando un CODEC, que es un convertidor que entrega o recibe una palabra digital de 8 bits, en forma serie y cuantizada bajo ley μ.
- Para realizar la transferencia de las muestras de la tarjeta a la memoria de la PC o de la PC a la tarjeta, utiliza acceso directo a memoria (DMA).
- La frecuencia de muestreo es programable (por software).

La señal de voz almacenada en la PC utilizando esta tarjeta, se transmite a la VAX por comunicación serie. Durante la transmisión de la señal, ésta se expande, ya que como se mencionó, la tarjeta comprime la señal de voz a 8 bits en ley μ; la expansión se realiza a 14 bits. En la VAX se realiza el procesamiento de la señal; el resultado del proceso de compresión expansión se transmite a la PC, para poder escuchar la señal y evaluar así el desempeño del sistema. En este caso, durante la transmisión de la señal procesada de la VAX a la PC, se comprime con ley μ a 8 bits, para poder utilizar la tarjeta de conversión.

El rango dinámico de las señales de voz a procesar, es de 8031 a -8031 que equivale a señales analógicas entre 5 y -5 volts. La razón de realizar la compresión de las señales de voz en la VAX y no en la PC, es el tiempo de proceso. Por ejemplo, si se desea comprimir una grabación de 50 segundos de duración, tomando una muestra cada 125 μs (8000 muestras/s), se tienen que procesar 400000 muestras, y considerando que son varios los algoritmos que se aplican a la señal, una PC tarda aproximadamente 12 horas en efectuar el proceso de compresión expansión, mientras que la VAX lo realiza en 30 minutos. Sin embargo, se debe recordar que el objetivo de este tipo de sistemas de compresión es su operación en tiempo real; para observar el funcionamiento del sistema en tiempo real, en el siguiente capítulo se propone una arquitectura basada en un procesador de señales.

Algunos datos cuantitativos de la operación del sistema de compresión, así como los resultados obtenidos se reportan a continuación.

Para el sistema final se utilizó una frecuencia de muestreo de 8000 muestras/s; de cada señal se almacenaron 393216 muestras, que corresponden a 49.15 segundos de grabación. El sistema se probó con señales de voz de diferentes personas, hablando en inglés y español, de grabaciones del radio y de un teléfono. Con algunas de estas señales se generaron los alfabetos que el sistema requiere. En la figura 5.6 se muestra un segmento de una señal de voz, la cual se adquirió de la forma descrita anteriormente. En esta figura se puede observar un segmento de ruido, que corresponde a las primeras muestras, en donde no hay voz,

seguido de un cambio abrupto en la señal, en el instante en que se presenta la voz, y finalmente, se nota una periodicidad en las últimas muestras, la cual debe ser detectada por el algoritmo de periodo de tono.

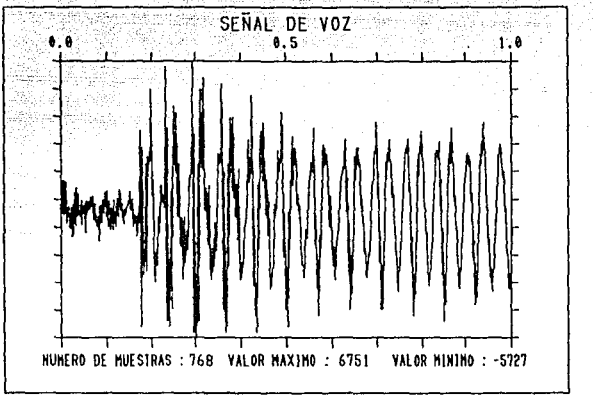


Figura 5.6 Segmento de una señal de voz, muestreada a 8000 Hz.

En cuanto al primer algoritmo del esquema de compresión, el escalamiento armónico en el dominio del tiempo, la relación de compresión obtenida es por lo menos 2 a 1, generalmente es mayor; ya que el número de veces que no se realiza reducción, es menor que el número de veces que se realiza reducción de tres bloques a uno y de dos bloques a uno, como se puede observar en los resultados siguientes, que corresponden a tres señales diferentes :

Señal 1	Bloques no reducidos.....	80
	Bloques con reducción 2:1....	1782
	Bloques con reducción 3:1....	2573
Señal 2	Bloques no reducidos.....	65
	Bloques con reducción 2:1....	3404
	Bloques con reducción 3:1....	973
Señal 3	Bloques no reducidos.....	177
	Bloques con reducción 2:1....	3267
	Bloques con reducción 3:1....	980

Con los archivos de periodo de tono generados con el algoritmo anterior, se diseñó el alfabeto de codificación de periodo de tono utilizando los códigos de Huffman. La longitud promedio del código es 6.2 bits/muestra, mientras que la entropía es 6.18 bits/muestra, lo que nos da una eficiencia de 99.6%. La distribución de probabilidades de cada periodo de tono se muestra en la figura 5.7, donde se puede observar que el periodo de tono, con mayor probabilidad de ocurrencia, es P0=0 (es el primer periodo de tono que aparece del lado izquierdo de la gráfica, seguido del periodo 30,31,...,128); debe notarse que los periodos de tono con mayor probabilidad son múltiplos y submúltiplos de valores alrededor de 60. En la figura 5.8 se muestra la evolución del periodo de tono, para los primeros 200 periodos detectados para la señal 2. Se observa una zona, alrededor de los primeros 100 periodos de tono, donde el periodo permanece prácticamente constante.

En las figuras 5.9 y 5.10 se ilustra el proceso de reducción de dos periodos de señal a uno. Para el segmento de señal de la figura 5.9, el

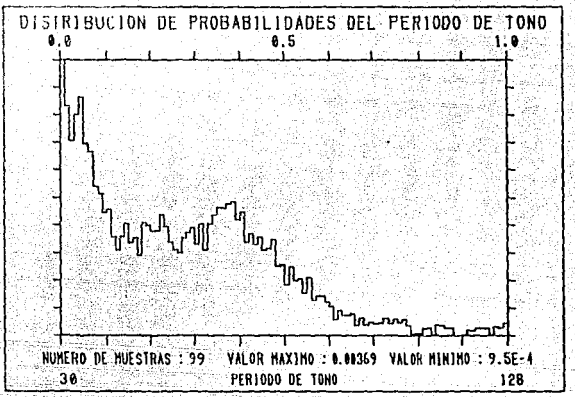


Figura 5.7 Distribución de probabilidades para los periodos de tono.

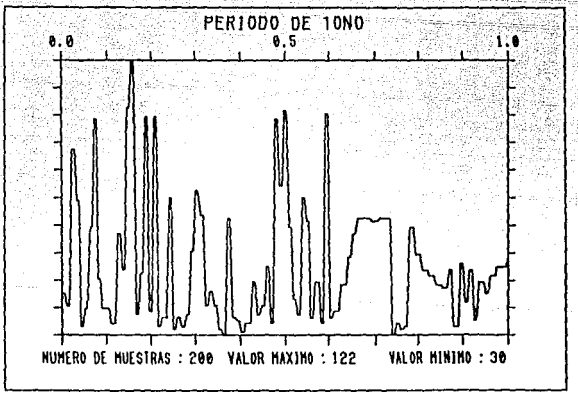


Figura 5.8 Evolución del periodo de tono para un segmento de señal.

algoritmo de detección del periodo de tono determina que existen dos periodos de 44 muestras que pueden ser reducidos a uno; estos dos periodos corresponden a las 88 muestras graficadas en esta figura. De acuerdo con el algoritmo, de estos dos periodos se obtiene un segmento de 44 muestras (figura 5.10), donde se puede observar que el principio del nuevo segmento se parece al principio del segmento de 88 muestras original y el final se parece al final del mismo. La parte central del nuevo segmento es un promedio de los dos segmentos de 44 muestras involucrados en la reducción.

A partir de aquí, tomaremos un segmento de señal para describir las siguientes etapas del procesamiento. El segmento de señal se muestra en la figura 5.11, mientras que la señal resultado del escalamiento se presenta en la figura 5.12.

El siguiente algoritmo que se aplica a la señal de voz es el cálculo de las correlaciones. En la figura 5.13, se muestra una gráfica del conjunto de 11 correlaciones calculadas para el segmento de señal seleccionado.

A continuación se realiza el cálculo de la transformada de Fourier de la señal, empleando la TRF, con orden 7 y ventanas trapezoidales. Para cada segmento de señal se obtienen 65 componentes de amplitud y 65 de fase.

Con la cuantización vectorial de las correlaciones, se obtiene directamente una aproximación del espectro de amplitudes por predicción

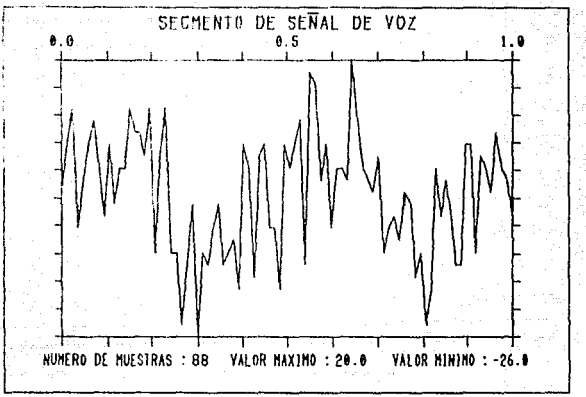


Figura 5.9 Segmento de 88 muestras de señal de voz a ser reducido.

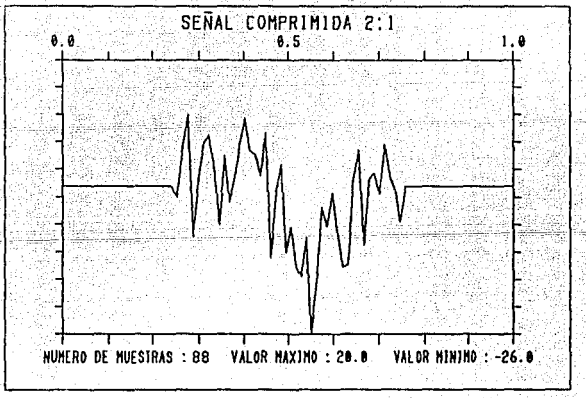


Figura 5.10 Segmento que resulta de la reducción.

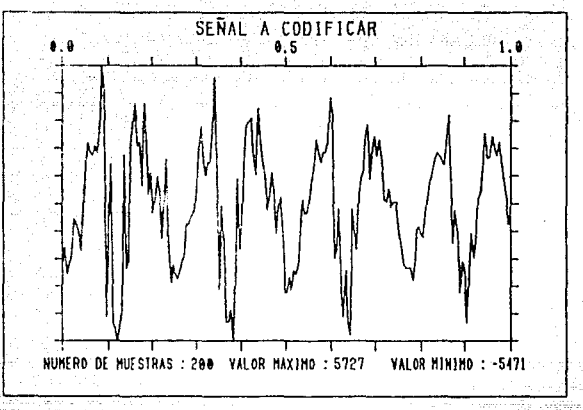


Figura 5.11 Señal de voz a codificar.

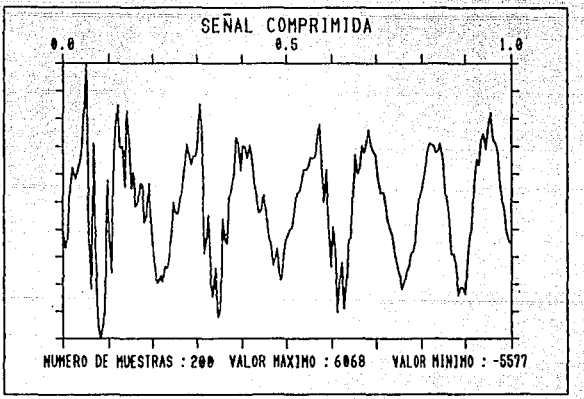


Figura 5.12 Señal resultado de la primera compresión.

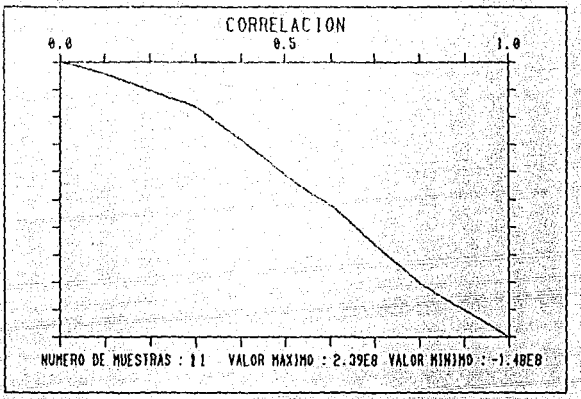


Figura 5.13 Correlaciones de un segmento de señal de voz.

lineal. En la figura 5.14, se comparan gráficamente los espectros de amplitud obtenidos a partir de la TRF (línea discontinua) y de predicción lineal (línea continua). Se observa claramente que el espectro de amplitud por predicción lineal, es una versión suavizada del espectro de la TRF. En la figura 5.15 se muestra la fase obtenida por TRF para el mismo segmento de señal.

El siguiente paso es codificar el cociente de los espectros de amplitud normalizados (figura 5.16), dada la asignación de bits basada en el espectro de amplitud por predicción lineal. Para este segmento, de acuerdo con el número de componentes que representa, se le asignaron

369.4 bits, para una tasa de trasmisión de 9600 bits por segundo, de los cuales la mitad (184.7 bits), se asignan a amplitudes y el resto a las fases.

En la figura 5.17, se presenta la distribución de bits para amplitudes y en la figura 5.18, la distribución de bits para las fases. La distribución de bits para amplitudes y fases no es la misma, ya que para las primeras se utiliza el espectro de amplitudes por predicción lineal y para las fases, el espectro de amplitud cuantizado. Se codifica también la ganancia de la TRF utilizando cuantización vectorial con un alfabeto de 9 bits.

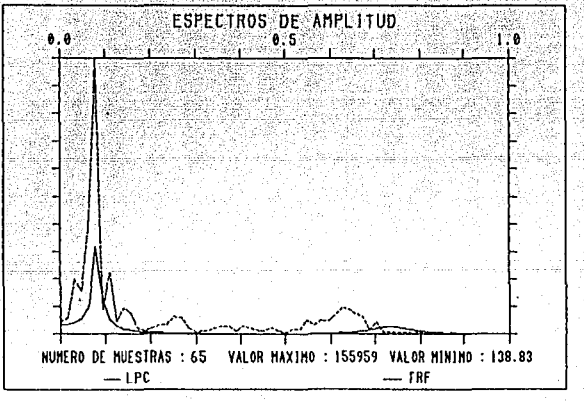


Figura 5.14 Comparación de espectros de amplitud.

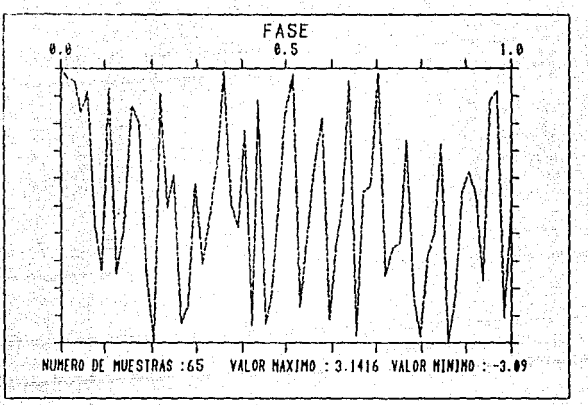


Figura 5.15 Espectro de fase del segmento de señal.

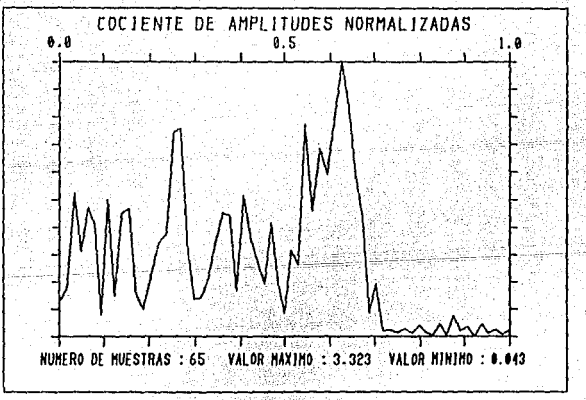


Figura 5.16 Cociente de espectros normalizados.

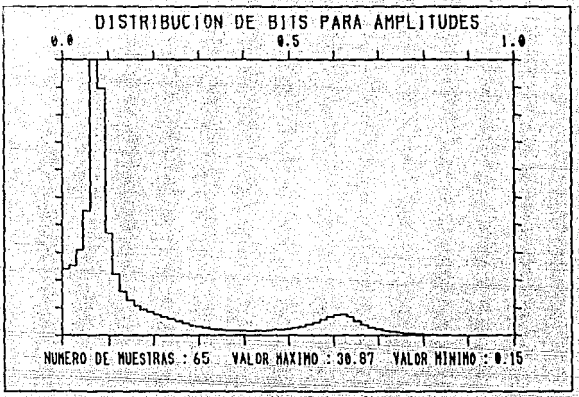


Figura 5.17 Distribución de bits para amplitudes a 9600 bps.

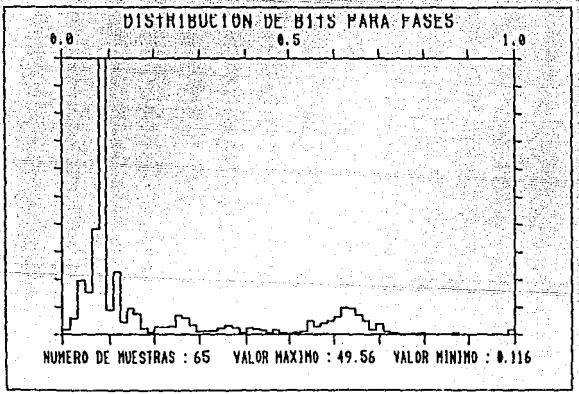


Figura 5.18 Distribución de bits para fases a 9600 bps.

Hasta aquí termina la operación de codificación en el transmisor y a continuación se describe el proceso de decodificación en el receptor.

Para efectuar el proceso de codificación se utilizaron dos tasas de transmisión: 9600 y 4800 bps. En el siguiente grupo de gráficas se puede observar el resultado de la codificación tanto de amplitudes como de fases. La figura 5.19 compara las amplitudes originales con las codificadas a 9600 bps, mientras que la figura 5.20 compara lo mismo para 4800 bps. De igual forma se presentan las figuras 5.21 y 5.22 para las fases correspondientes. Es evidente la mejor reproducción de las amplitudes codificadas a 9600 bps. Por lo que respecta a las fases, el resultado de la codificación a 9600 bps, aproxima mejor a las fases originales que las codificadas a 4800 bps, aunque se aprecia un mejor trabajo en la codificación de las amplitudes que en la de las fases.

Se calcula ahora la transformada inversa de Fourier y se elimina el efecto del traslape de las ventanas, lo que da como resultado las gráficas de las figuras 5.23 y 5.24, donde se comparan las señales codificadas a 9600 y 4800 bps respectivamente, con la señal equivalente original. (Nota: Son las señales antes del aplicar el algoritmo de escalamiento armónico inverso)

Finalmente, se muestra la comparación gráfica de la señal codificada a diferentes tasas de transmisión, para 9600, 7200, 4800, 3600 y 2400 bps, con la señal original. En estas gráficas se puede observar que la distorsión en la señal codificada aumenta conforme disminuye la velocidad de transmisión, ya que a menor velocidad de transmisión, se dispone de un menor número de bits para codificar.

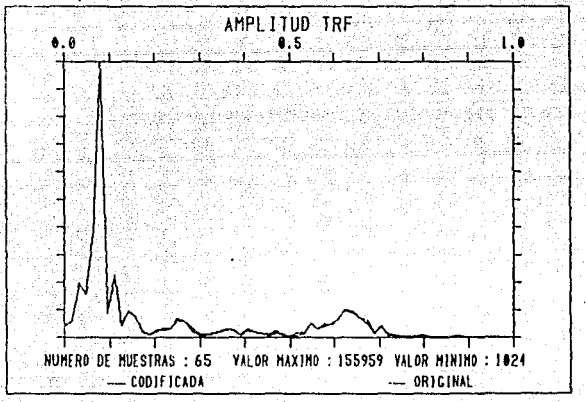


Figura 5.19 Comparación de amplitudes original y codificadas a 9600 bps.

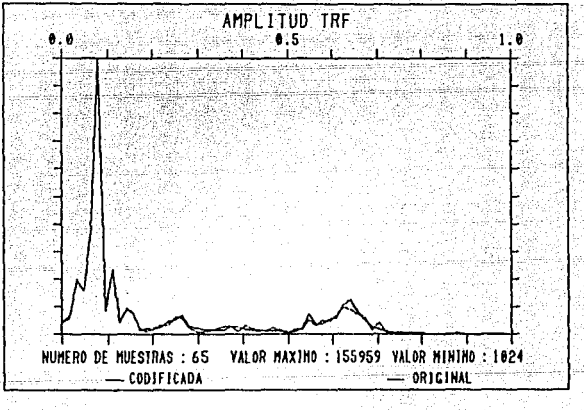


Figura 5.20 Comparación de amplitudes original y codificadas a 4800 bps.

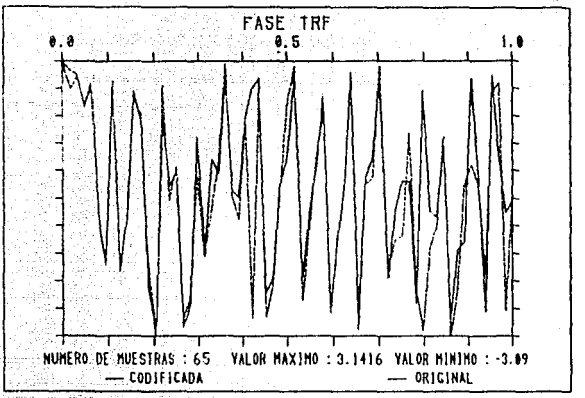


Figura 5.21 Comparación de fases original y codificadas a 9600 bps.

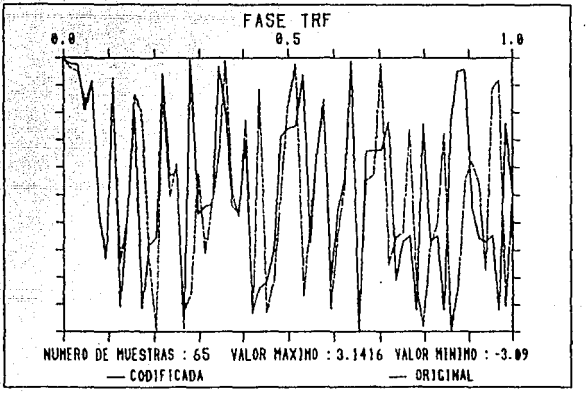


Figura 5.22 Comparación de fases original y codificadas a 4800 bps.

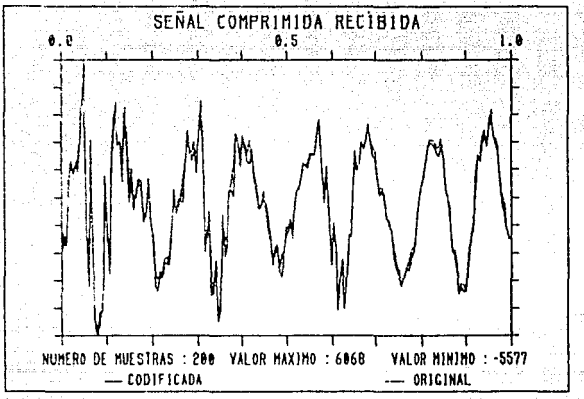


Figura 5.23 Comparación de señal original y sintética reducida a 9600 bps.

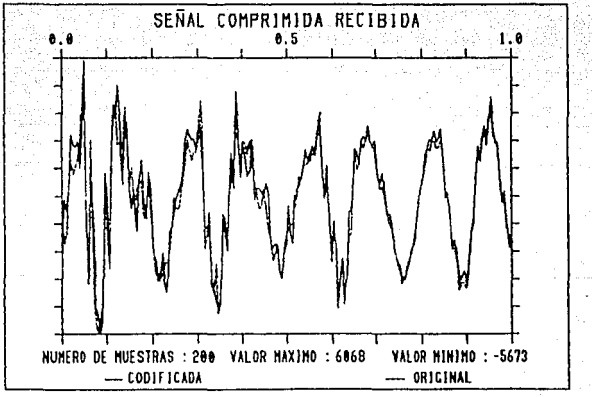


Figura 5.24 Comparación de señal original y sintética reducida a 4800 bps.

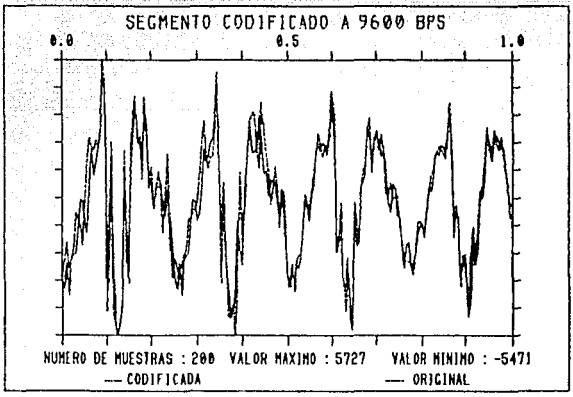


Figura 5.25 Comparación señal original con señal codificada a 9600 bps.

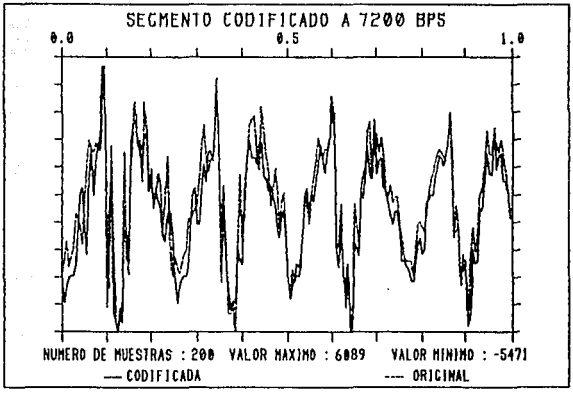


Figura 5.26 Comparación señal original con señal codificada a 7200 bps.

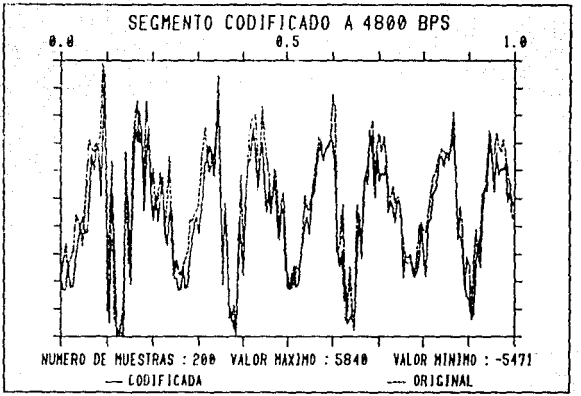


Figura 5.27 Comparación señal original con señal codificada a 4800 bps.

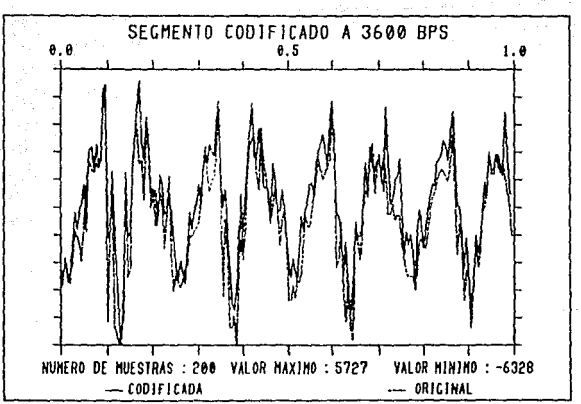


Figura 5.28 Comparación señal original con señal codificada a 3600 bps.

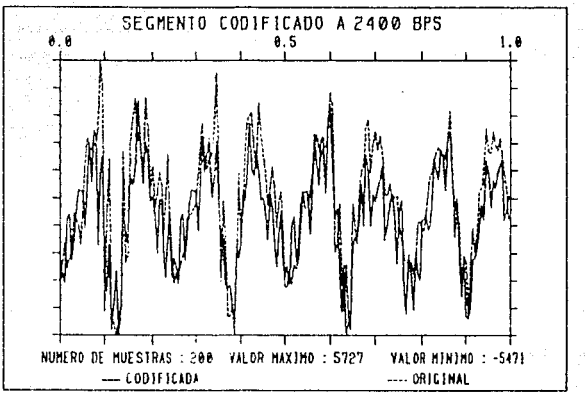
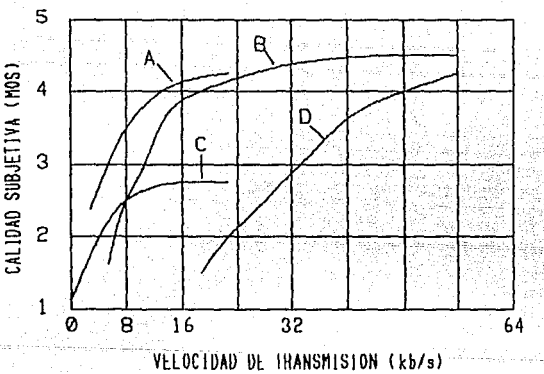


Figura 5.29 Comparación señal original con señal codificada a 2400 bps.

Con las gráficas anteriores se puede observar la similitud entre la señal original y la señal codificada a diferentes velocidades de transmisión, aún a 2400 bps la señal es parecida a la señal original. Sin embargo, la similitud gráfica no es el mejor indicador del desempeño del sistema de compresión. La mejor forma de evaluar un sistema de compresión de señales de voz es someter al mismo a una prueba subjetiva de la calidad de la señal codificada.

En la literatura se presentan comparaciones gráficas de los sistemas de codificación de voz, como la mostrada en la figura 5.30, en donde se califica la calidad de un sistema, utilizando una medida conocida como calificación de opinión media (mean opinion score MOS), otorgada por un grupo de personas. Una calificación de 5 indica calidad perfecta, una calificación de 4 o más, representa buena calidad,

significa que las personas encuentran la voz tan entendible como la original y sin distorsión. Las calificaciones de 3 a 4 se conocen como calidad de comunicación. En estos valores, la distorsión está presente pero no es obvia, la voz se mantiene entendible. Al final de la escala está la calidad sintética tipica de los codificadores de voz, en donde se entienden las palabras, pero generalmente no se identifica a la persona que habla; otra caracteristica de estos sistemas es el bajo nivel de robustez bajo diferentes condiciones, como ruido, o muchas personas hablando. [8] y [25]



A - CODIFICADOR HIBRIDO ALTA COMPLEJIDAD
B - CODIFICADOR DE FORMA DE ONDA ALTA COMPLEJIDAD
C - VOCODER ALTA COMPLEJIDAD
D - CODIFICADOR DE FORMA DE ONDA BAJA COMPLEJIDAD

Figura 5.30 Calidad subjetiva de los sistemas de codificación de voz

Se realizó una prueba de este tipo, solicitando a 16 personas su participación, se buscó que las personas hubieran escuchado a la persona que habla en las grabaciones en alguna ocasión.

La prueba consistió en reproducir un conjunto de 8 grabaciones de igual duración, donde el texto de cada una es el mismo pero codificado a diferente velocidad de transmisión. Se informó a los participantes que las grabaciones no seguían un orden en particular, por ejemplo, que la primera no era mejor que la segunda y se pidió que trataran de identificar, después de escuchar la primera grabación, a la persona que habla en la misma. Después de haber reproducido la primera grabación, se reprodujo la grabación original, con el fin de que la calificación de las señales codificadas se hiciera comparando con la original. Es importante mencionar, que se reprodujo como señal original la grabada directamente de un micrófono, y no la señal original almacenada en la computadora PC.

Los resultados fueron los siguientes :

Grabación	Velocidad de Transmisión	Calificación promedio
1	3600	3.8296
2	4800	3.6392
3	7200	3.8080
4	9600	3.8134
5	4800 en promedio	2.9921
6	3200 en promedio	3.0733
7	2400 en promedio	2.8296
8	9600,7200,4800,3600,4800,7200	3.6762

En las primeras 4 grabaciones, permanece constante la velocidad de transmisión, mientras que en las siguientes tres grabaciones, se simula la existencia de varias conversaciones por un canal de transmisión, por ejemplo, la primera simula dos conversaciones, que comparten un canal de 9600 bps, correspondiendo en promedio a cada una 4800 bps; en la última grabación se modifica la velocidad de transmisión cada 8 segundos aproximadamente. Para la simulación de varias conversaciones por un canal de transmisión, la distribución de bits para cada una se realiza en base a que tan parecido es cada segmento a ruido blanco, por ejemplo, para tres conversaciones, se aplica el algoritmo descrito hasta antes de realizar la asignación de bits. El algoritmo de asignación de bits, distribuye los bits disponibles para las tres señales de acuerdo con el porcentaje de error que representa cada segmento; el error se calcula con los coeficientes de reflexión obtenidos por Levinson.

De las 16 personas que participaron, 11 identificaron a la persona que habla en la primera grabación. Esto es importante, pues indica que aún en la menor tasa de transmisión utilizada en la prueba, se puede identificar a la persona que habla, característica deseable en los sistemas de codificación de voz.

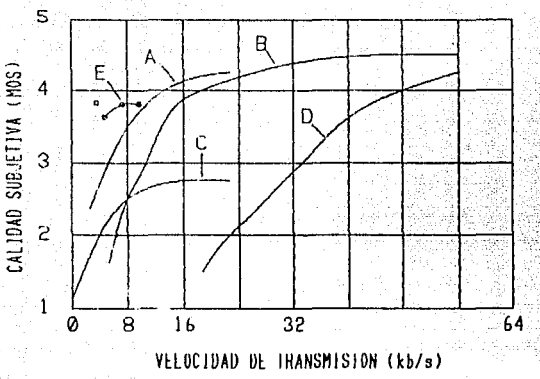
Puede observarse que los participantes otorgan una calificación alta, en promedio, a la primera grabación, siendo que no es la mejor, tal vez debido a que de las 16 personas, sólo 5 han tenido contacto con sistemas de este tipo, por lo que los demás adquieren experiencia durante la prueba. En la realización de ésta se notó lo siguiente: durante la reproducción de las primeras grabaciones, los participantes

ponían toda su atención en todo el tiempo que duraban las mismas, para asignar una calificación, pero para las últimas, sólo escuchaban el principio e inmediatamente calificaban; por ejemplo, nadie hizo comentarios sobre la última grabación, donde paulatinamente se modifica la velocidad de transmisión.

Si estos resultados se consideran en la gráfica de calidad subjetiva de señales de voz, mostrada anteriormente, se observa que el presente sistema tiene una buena calidad para tasas de transmisión menores a 9600 bps (figura 5.31), sin embargo para poder generalizar el resultado es conveniente enviar una cinta con señales codificadas a un lugar especializado en este tipo de pruebas.

En cuanto a la complejidad del sistema, el trabajo del transmisor es mayor que el del receptor, ya que debe detectar el periodo de tono, codificarlo, calcular y cuantizar las correlaciones, calcular la Transformada de Fourier y cuantizar amplitudes, fases y sumatoria de amplitudes; mientras que el receptor, decodifica el periodo de tono, decodifica amplitudes, fases y ganancia, calcula la TRF inversa y expande la señal. De todas las etapas del sistema, la más compleja es el cálculo de la Transformada de Fourier, directa e inversa, pues la cuantización más que compleja es lenta, ya que se trata de búsqueda completa, además de que los alfabetos solo se calculan una vez. El sistema debe tener un buen manejo de memoria, para accesar eficientemente a todos los alfabetos involucrados, tanto en el codificador como en el decodificador. Una sugerencia del manejo de memoria para un proceso de este tipo, se describe en el siguiente

capitulo.



A - CODIFICADOR HIBRIDO ALTA COMPLEJIDAD
B - CODIFICADOR DE FORMA DE ONDA ALTA COMPLEJIDAD
C - VOCODER ALTA COMPLEJIDAD
D - CODIFICADOR DE FORMA DE ONDA BAJA COMPLEJIDAD
E - SISTEMA DE COMPRESION

Figura 5.31 Comparación de la calidad subjetiva del sistema de compresión de señales de voz.

En este capitulo se presentó el esquema de compresión desarrollado, detallando cada uno de los algoritmos empleados, así como los resultados obtenidos con el mismo. En el capitulo de conclusiones, se harán algunas recomendaciones del trabajo a realizar con este sistema, además de sugerir modificaciones para su mejoramiento.

VI. ARQUITECTURA PARA EL SISTEMA DE COMPRESION.

VI.1. DESCRIPCION DEL HARDWARE DISEÑADO PARA EL PROCESAMIENTO DIGITAL DE SEÑALES.

En la actualidad, existen procesadores que han sido diseñados especialmente para el procesamiento digital de señales, de marcas comerciales de circuitos integrados como National, Motorola, NEC, Analog Devices, Texas Instruments y Signetics. [26]

Los procesadores diseñados para el procesamiento digital de señales comparten una característica: la velocidad en la ejecución de ciertos algoritmos. Tienen un reducido juego de instrucciones optimizado, para realizar lo más rápido posible sumas, restas, multiplicaciones y corrimientos. En los procesadores digitales de señales (PDS) recientes, acompaña al reducido juego de instrucciones, el cual puede ser implantado en un área pequeña de silicio, un multiplicador que realiza la operación en un ciclo de reloj, para lo cual dispone de gran parte

del área de silicio [26]. En contraste, los procesadores de propósito general realizan la multiplicación en varios ciclos de reloj, ellos dedican una mayor área de silicio para el conjunto de instrucciones, entre las cuales se incluyen manejo de memoria caché, operaciones de punto flotante, manejo de bases de datos.

Por ejemplo, para la implantación de un filtro digital FIR (de respuesta a impulso finita) de n etapas, se requiere aproximadamente la acumulación de n multiplicaciones. Esta operación se ejecuta por cada muestra de entrada. La mayoría de los nuevos PDS ejecutan la instrucción de multiplicación y acumulación en un ciclo de reloj de aproximadamente 100 ns. Para un procesador de los más rápidos de propósito general, como el 80386 de 16MHz - el cual efectúa la suma de registro a registro de 16 bits en 125 ns -, requiere cerca de 1250 ns para una multiplicación de 16 por 16 bits, un procesador 8088 de 5 MHz requiere cerca de 32,000 ns para realizar la misma instrucción [26]. Otros algoritmos del PDS - por ejemplo la transformada rápida de Fourier TRF - requieren un número mayor de sumas y restas que de multiplicaciones, sin embargo aún para este tipo de algoritmos el multiplicador relativamente lento de los procesadores de propósito general, representa una gran limitación.

De lo anterior es evidente la gran ventaja de utilizar los PDS, en lugar de los procesadores de propósito general, para la implantación de algoritmos de procesamiento digital de señales.

Una de las desventajas de los primeros PDS es que no disponen de una forma eficiente de comunicación con otro procesador para realizar la

función de coprocesador, sin embargo, en los PDS más recientes este problema se ha resuelto, pues incluyen medios de comunicación con otros procesadores y para manejo de memoria (DMA).

Uno de los PDS de la familia de Texas Instruments, de la primera generación de procesadores digitales de señales, es el TMS32010, del cual a continuación se describen sus principales características.

Los procesadores de la familia TMS320 tienen como características fundamentales las siguientes : [27] y [28]

. Arquitectura Harvard,
. pipeline extensivo,
. multiplicador en hardware,
. instrucciones especiales para el procesamiento digital de señales,
. rápidos ciclos de instrucción.

La familia TMS320 utiliza una arquitectura Harvard modificada por su velocidad y flexibilidad. En una arquitectura Harvard estricta, las memorias de datos y programa están separadas, lo que permite un traslape total de los ciclos de fetch y ejecución. La modificación que introducen los procesadores de la familia TMS320 permite la transferencia de datos, entre la memoria de datos y la de programa, incrementando así la flexibilidad del dispositivo.

Junto con la arquitectura Harvard, se utiliza extensivamente el "pipelining", para reducir al mínimo el ciclo de instrucción. El

"pipeline" va de 2 a 4 niveles, dependiendo de la generación a la cual pertenece el procesador. Para los de la primera generación es de 2 niveles, ésto quiere decir que el procesador está ejecutando dos instrucciones en paralelo, y cada instrucción está en una etapa diferente de su ejecución. Por ejemplo, mientras una instrucción está siendo decodificada, la otra se puede estar ejecutando.

En cuanto al multiplicador en hardware, se ha mencionado anteriormente la gran velocidad que éste permite, en la ejecución de algoritmos de procesamiento digital de señales. Otra característica de la familia TMS320 son las instrucciones especiales que se pueden utilizar, como es el caso de instrucciones que permiten cargar un registro con un dato, mover los datos en la memoria para producir un retraso, hacer la multiplicación del dato almacenado en el registro con el dato después del retraso y sumarlo con el resultado de la muestra anterior, todo en un ciclo.

La velocidad de ejecución de instrucciones en estos procesadores, abre la posibilidad de ejecutar algoritmos en tiempo real, ya que por ejemplo, la primera generación de PDS de la familia TMS320 tienen ciclos de instrucción entre 160 y 200 ns, mientras que los de la tercera generación tiene ciclos de instrucción de 60 a 75 ns.

El TMS3210 es el primer miembro de la familia TMS320, elaborado en 1982; fué el primer procesador capaz de realizar 5 millones de instrucciones por segundo. Contiene un multiplicador en hardware que realiza la multiplicación de dos números de 16 bits, con resultado de 32

bits en un ciclo de instrucción de 160 ns. Tiene un registro de corrimiento que permite hacer corrimientos de los datos que van hacia la unidad aritmético-lógica. Se incluye hardware extra para dos registros auxiliares que permiten el direccionamiento indirecto de los datos en memoria RAM, pueden ser configurados para auto-incrementarse o auto-decrementarse facilitando el manejo de tablas de datos.

Se disponen de dos modos de operación: microprocesador o microcomputadora, seleccionable a través del pin MC/$\overline{MP}$. Cuando se utiliza el modo microcomputadora, se equipa al TMS con una memoria ROM de 1536 palabras, la cual graba el fabricante de acuerdo con el programa del usuario. Cuando se utiliza en el modo microprocesador, puede accesar 4096 palabras de programa localizadas en memoria externa. La memoria de datos está incluida en el dispositivo y son 144 localidades de 16 bits. Existen cuatro elementos básicos en el procesador: la unidad aritmético-lógica (ALU), el acumulador, el multiplicador, y los registros de corrimiento. Todas las operaciones se realizan utilizando aritmética complemento a dos.

La unidad aritmético-lógica, es de propósito general y opera con palabras de 32 bits. Puede hacer sumas, restas y operaciones lógicas.

El acumulador almacena el valor que proviene del ALU y generalmente también es entrada al ALU, opera con palabras de 32 bits. Se divide en dos partes, alta (bits 31 al 16) y baja (bits 15 al 0), disponiendo de instrucciones para almacenar valores en la parte alta o en la parte baja de éste.

El multiplicador consiste de tres unidades: el registro T, el registro P y el multiplicador propiamente dicho. El registro T es un registro de 16 bits que almacena al multiplicando, mientras que el registro P es de 32 bits para almacenar el producto. Para utilizar el multiplicador, se almacena el multiplicando en el registro T a partir de la memoria de datos en RAM, en la instrucción de multiplicación se indica el multiplicador que puede ser un valor (instrucción MPYK) o una localidad de memoria de datos (instrucción MPY).

Los registros de corrimiento son 2, uno que permite realizar corrimientos a la izquierda de 0 a 16 bits, a los datos que se van a almacenar, a substraer o sumar al acumulador y otro, que realiza corrimientos de 0, 1 y 4 bits, para el manejo de los bits de signo, en los cálculos de aritmética complemento a dos.

El bus de datos de 16 bits del TMS32010 se puede utilizar para realizar funciones de entrada salida a velocidades de 50 millones de bits por segundo. Se disponen de 8 puertos de entrada y 8 de salida, además de una entrada para manejo de interrupciones por poleo y otra para interrupción directa.

Contiene también un stack de 4 niveles para almacenar al contador de programa, durante la atención a interrupciones o saltos a subrutinas.

En la figura 6.1 se muestra un diagrama de bloques del TMS32010.

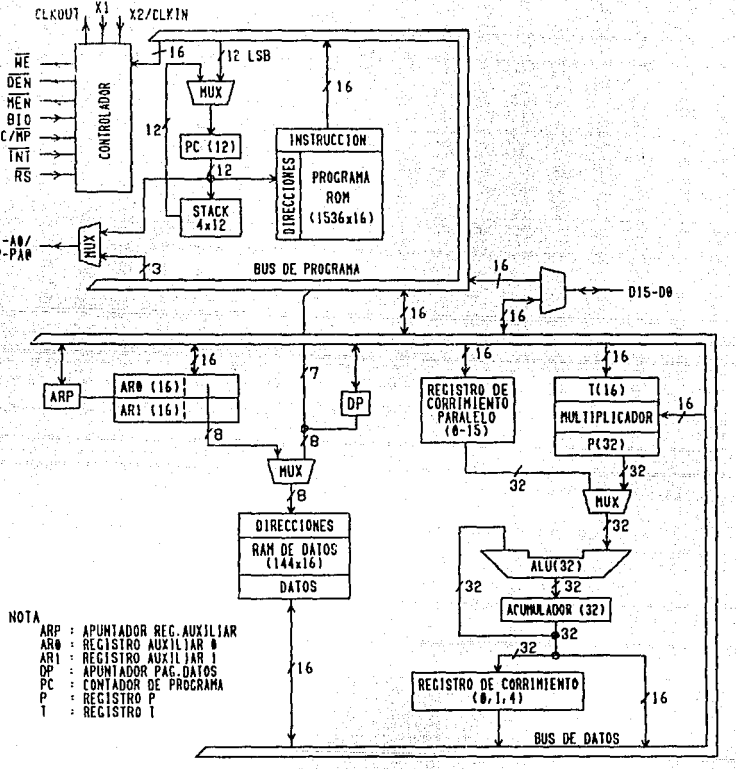


Figura 6.1 Diagrama de bloques del procesador TMS32010.

VI.2. ARQUITECTURA PARA LA IMPLANTACION DEL SISTEMA DE COMPRESION.

En esta sección se propone una arquitectura basada en el procesador TMS32010, para la implantación en hardware del sistema de compresión digital de señales de voz. El objetivo de la arquitectura, es poder probar el desempeño del sistema operando en un procesador de punto fijo. Para el diseño de esta arquitectura, se considera que la señal de voz se encuentra almacenada en archivos en una computadora del tipo PC, esto es porque se dispone de archivos de este tipo.

Las características que se desean en la arquitectura, son las siguientes:

- Comunicación eficiente con la computadora personal.
- Detener lo menos posible la ejecución del procesador de señales.
- Simulación en tiempo real.

Debido a que el algoritmo diseñado para el sistema de compresión es "pesado" en términos computacionales, se va a requerir la operación de varios procesadores de señales en paralelo, y considerando que la programación de los algoritmos, en el lenguaje del procesador de señales, no forma parte de este trabajo, la arquitectura que a continuación se presenta es modular, de tal forma que a ella se pueden agregar módulos de procesamiento iguales al propuesto aquí, permitiendo que la arquitectura pueda crecer fácilmente.

Esta arquitectura se conoce como procesamiento concurrente-datos

compartidos (Shared Data Concurrent Procesor SDCP). En la figura 6.2, se muestra un diagrama de bloques simplificado de este tipo de arquitectura. Como se mencionó anteriormente, pueden existir N módulos de proceso concurrente (MP), llamados módulo 1 a módulo N. El sistema utiliza un controlador maestro (CM). Los datos están disponibles para su uso por todos los módulos de proceso, a través de un bus de alta velocidad (BCD Bus compartido de datos). En la inicialización del sistema, el CM utiliza el BCD para enviar los programas a cada módulo. En caso de que falle alguno de los módulos, el controlador puede deshabilitar al módulo y reasignar la tarea que éste realizaba a otro y continúa así funcionando [29].

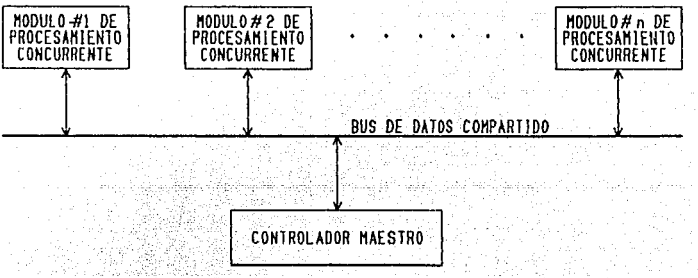


Figura 6.2 Sistema de procesamiento concurrente-bus de datos compartido.

De acuerdo con el sistema de compresión, para la ejecución de cada uno de los algoritmos involucrados, se requiere un marco de datos, el resultado de un algoritmo, generalmente son datos de entrada al siguiente algoritmo, por lo cual, se pensó en almacenar en parte de la

memoria de programa de cada procesador, el programa o programas a ejecutar por él, mientras que en el resto de la memoria se pueden almacenar los datos necesarios para la operación de ese procesador.

Uno de los problemas a resolver, fue el hecho de que si un módulo de proceso requiere de los resultados de otro módulo de proceso para operar, este procesador debe detener su operación, para esperar a que el controlador realice la transferencia de datos de un procesador a otro. Para no detener la ejecución de los procesadores se propone tener dos bloques de memoria de la mitad de capacidad de la memoria total, es decir de 2k palabras, para que mientras el procesador está accesando datos de uno de los bloques de memoria, el controlador puede transferir datos al otro bloque de memoria; cuando el procesador termine de operar con un bloque de datos, el controlador se encarga de cambiar el bloque de memoria sobre el cual debe operar el procesador. De esta forma, se puede evitar el detener la ejecución de los procesadores, sólo será necesario detenerlos en la inicialización del sistema. Para la adecuada operación del sistema se requiere que el controlador opere a una velocidad superior a la de los procesadores. En la figura 6.3, se muestra un diagrama de bloques de esta configuración.

La forma en que operaría el sistema es la siguiente: primero se transfiere a la memoria de programa de cada procesador, el programa que deberá ejecutar y a continuación se transfieren los datos a la otra parte de la memoria. Por ejemplo, uno de los módulos de procesamiento se puede encargar de ejecutar el algoritmo EADT, junto con la codificación del período de tono, otro para calcular la TRF, otro para calcular las

correlaciones y cuantizarlas generando el espectro por predicción lineal y otro para la codificación por cuantización vectorial de amplitudes, fases y sumatoria de amplitudes.

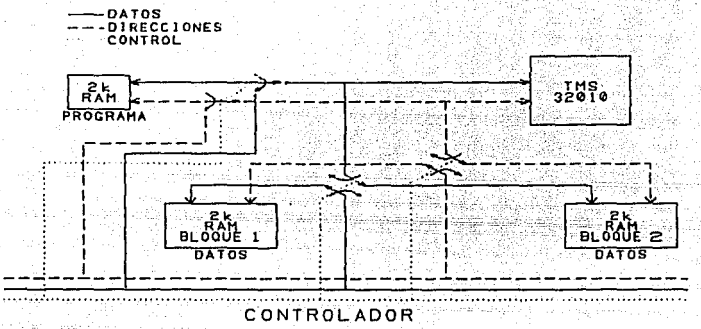


Figura 6.3 Sistema de manejo de memoria para el TMS32010

Una vez almacenados en memoria tanto el programa como los datos, el controlador se encarga de indicar a los procesadores en que momento inicien su operación, ya que puede ser que para que operen unos de los módulos requieran que el primer módulo haya terminado ya de procesar los primeros datos. El controlador informará de ésto a los procesadores, a través del registro de estado del controlador. Siguiendo con el ejemplo, en este caso, para que los módulos que calculan la TRF y la correlación operen, necesitan la señal resultado del algoritmo EADT, que efectúa el primer procesador, mientras que el procesador que realiza la codificación vectorial de amplitudes, fases y ganancia, necesita que

tanto el módulo de TRF como el de correlación, hayan terminado.

Para saber el estado de cada módulo de proceso, se incluyen en el diseño unos registros que el procesador de señales puede accesar, a través de sus puertos de entrada y salida. En este registro, el procesador de señales indicará al controlador, que ha terminado de operar con un bloque de memoria de datos, de la misma manera que el controlador, indica al procesador que puede inicar el proceso con el nuevo bloque de memoria.

El controlador tiene un papel fundamental en la adecuada operación del sistema, es el encargado de realizar las siguientes funciones :

- Indicar el inicio de operación a cada módulo de proceso,
- Hacer la conmutación de bloques de memoria en cada módulo,
- Transferir datos y programas de la computadora personal al sistema,
- Transferir resultados a la computadora personal,
- Transferir datos de un modulo de proceso a otro,
- Verificar condiciones de operación de los módulos de proceso,
- Manejo de direcciones en los bloques de memoria que no está accesando el procesador aritmético en cada módulo.

En el ejemplo, el controlador después de haber almacenado los programas de EADT, TRF, correlaciones y codificación por cuantización vectorial, ademas de los primeros marcos datos a procesar al módulo 1, indica al primer módulo que inicie su operación, y mantiene en espera a

los módulos restantes, mientras tanto transfiere más datos al bloque de memoria (2), que no está accesando el TMS del módulo 1 en operación, de tal forma que cuando éste le avise que ha terminado de procesar el bloque de memoria (1), el controlador realice la conmutación de bloques de memoria para el TMS del módulo 1. La comunicación entre el controlador y la computadora personal se realiza a través de un puerto paralelo conectado en un slot de la misma, de acuerdo con ésto, un programa en la computadora personal, es el encargado de enviar los programas y datos al sistema, así como de recibir los resultados obtenidos por éste.

Una vez realizada la conmutación de los bloques de memoria en el módulo 1, el controlador debe transferir los datos del bloque de memoria (1) del módulo de proceso 1, a los bloques de memoria (1) de los módulos encargados de calcular la TRF y las correlaciones (módulos 2 y 3, respectivamente). A continuación, el controlador debe indicar a ambos módulos que pueden iniciar su operación. Para la transferencia de los datos de un módulo a otro, es necesario generar las direcciones de cada bloque de memoria, así como habilitar los dispositivos involucrados en el flujo de datos (circuitos tres estados, buffers). Para generar las direcciones, el controlador dispone de dos grupos de contadores, uno para generar las direcciones de uno de los módulos y otro para las direcciones del otro módulo o módulos. En el caso de la transferencia de los datos del módulo 1 a los módulos 2 y 3, un grupo de contadores genera las direcciones del bloque de memoria (1) del módulo 1, y el otro grupo de contadores se encarga de direccionar a los bloques de memoria (1) de los módulos 2 y 3, ya que los datos que deben recibir ambos

módulos son los mismos.

A continuación, el controlador debe llenar el bloque de memoria (1) del módulo 1 con nuevos datos, tenerlo listo para que cuando el procesador del módulo 1 termine de operar con el bloque de memoria (2), inmediatamente continue trabajando con el bloque de memoria (1) de nuevo.

El funcionamiento del sistema en el caso de los módulos 2 y 3, es un poco diferente al del módulo 1, pues en este caso se debe esperar a que terminen de hacer sus cálculos ambos módulos antes de que se realice la transferencia de datos al módulo 4, que realizará la codificación por cuantización vectorial. Es importante mencionar que la distribución de las tareas en cada módulo, es la parte que más influye en el desempeño del sistema, en cuanto a su operación en tiempo real, pues si uno de los módulos termina mucho antes que el módulo que le proporciona los datos, éste deberá esperar hasta que el otro módulo concluya, provocando un retraso significativo en el proceso en conjunto.

Regresando al ejemplo, el módulo 4 iniciará su operación, cuando los módulos 2 y 3 hayan terminado de procesar los bloques de memoria (1) de cada uno; previamente se han almacenado en el bloque de memoria (1) del módulo 4, los periodos de tono generados en el módulo 1. Una vez que el módulo 4 ha terminado de efectuar la codificación por cuantización vectorial del primer bloque de datos, el controlador debe enviar las palabras codificadas obtenidas a la computadora personal para su almacenamiento.

El intercambio de datos, ya sea entre la computadora personal y el sistema, o entre los módulos de proceso, debe realizarse a una alta velocidad, pues el controlador debe renovar la información en todos los módulos de memoria que no se están accesando; debe hacer todo este trabajo mientras los procesadores están trabajando, pues si los procesadores tuvieran que esperar mucho tiempo a que el controlador les transfiera la información a procesar, el sistema se haria demasiado lento y se perderia la efectividad de este tipo de arquitectura.

Ahora se mencionarán algunas consideraciones importantes en el diseño del hardware del sistema, empleando la arquitectura antes mencionada.

Por reducción del hardware necesario para la implantación de los módulos de proceso, se utilizaron memorias de tipo estático, para los bloques de memoria para los procesadores TMS32010. Se utilizan las memorias 2148H3 con tiempo de acceso de 45 ns, debido a que los TMS requieren memorias con tiempo de acceso menor a 50 ns. [30]

La conmutación de los bloques de memoria, así como la selección de bus de direcciones y datos, se realizan utilizando circuitos de tres estados, bidireccionales en el caso de datos y unidireccionales en el caso de direcciones.

Para direccionar las localidades de memoria se utilizan contadores con carga paralelo, con los cuales se pueden generar direcciones incrementándolas de una en una, decrementándolas, iniciando a partir de

la dirección cero o de algún otro valor.

La parte central del controlador está diseñada en base al 2910, un secuenciador "bit slice", diseñado para controlar la ejecución de microinstrucciones almacenadas en su memoria de programa. Junto con su capacidad de acceso secuencial, se dispone de saltos condicionales a cualquier microinstrucción dentro de su rango de 4096 micro-palabras. Se dispone de un stack de tipo LIFO (último en entrar, primero en salir) para manejar saltos a subrutinas. [31]

Durante cada microinstrucción, el controlador de microprograma genera una dirección de 12 bits provenientes de una de 4 fuentes : 1) el registro de direcciones de microprograma, que usualmente contiene una dirección mayor en una unidad a la dirección anterior (PC); 2) una entrada externa (D); 3) un registro/contador (R), que mantiene el dato almacenado durante la microinstrucción anterior; o 4) de la pila de nueve niveles del tipo último en entrar, primero en salir.

En la figura 6.4 se muestra un diagrama de bloques del secuenciador 2910.

El dispositivo tiene salidas tipo tres estados. Dispone de un conjunto de 16 instrucciones, las cuales seleccionan la dirección de la siguiente microinstrucción a ser ejecutada. Cuatro de ellas son incondicionales; diez, dependen parcialmente de condiciones externas y las restantes dependen del contenido del registro/contador.

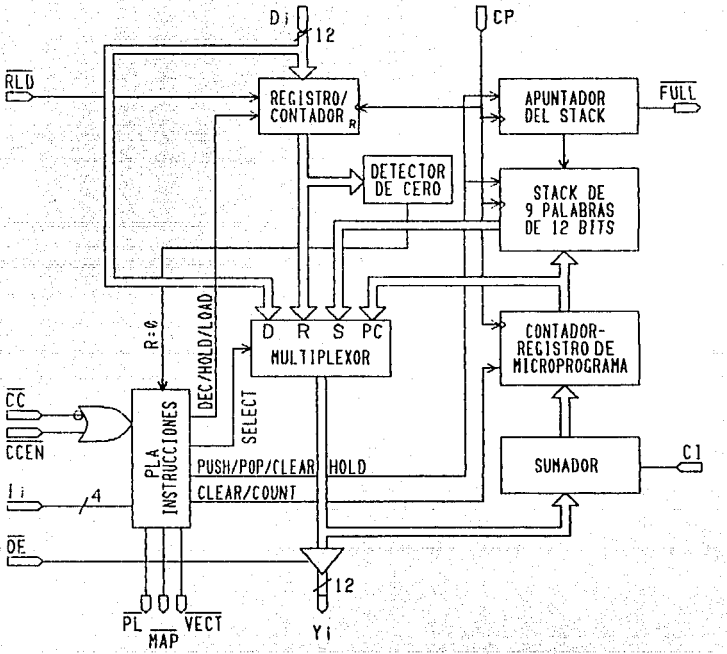
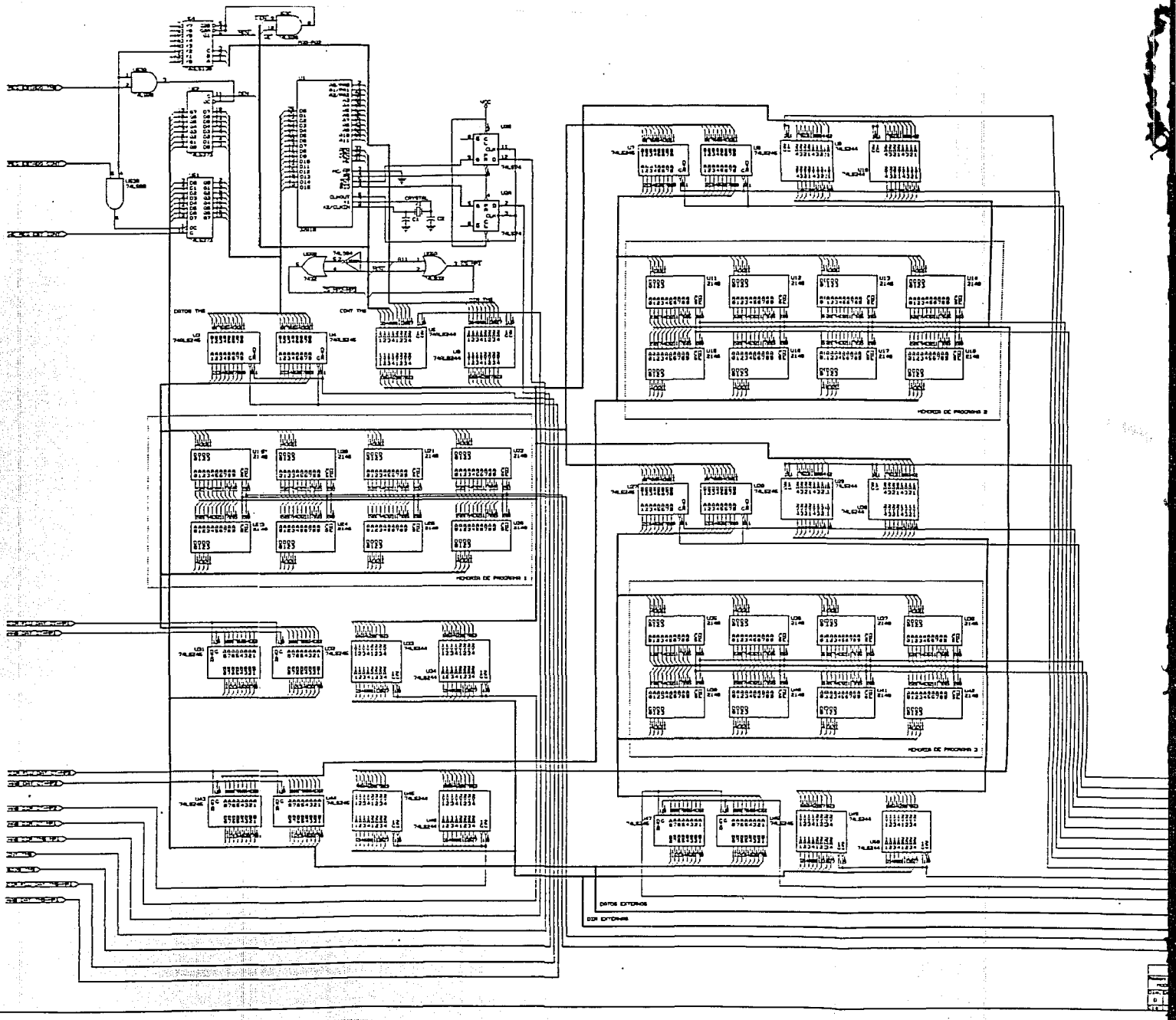
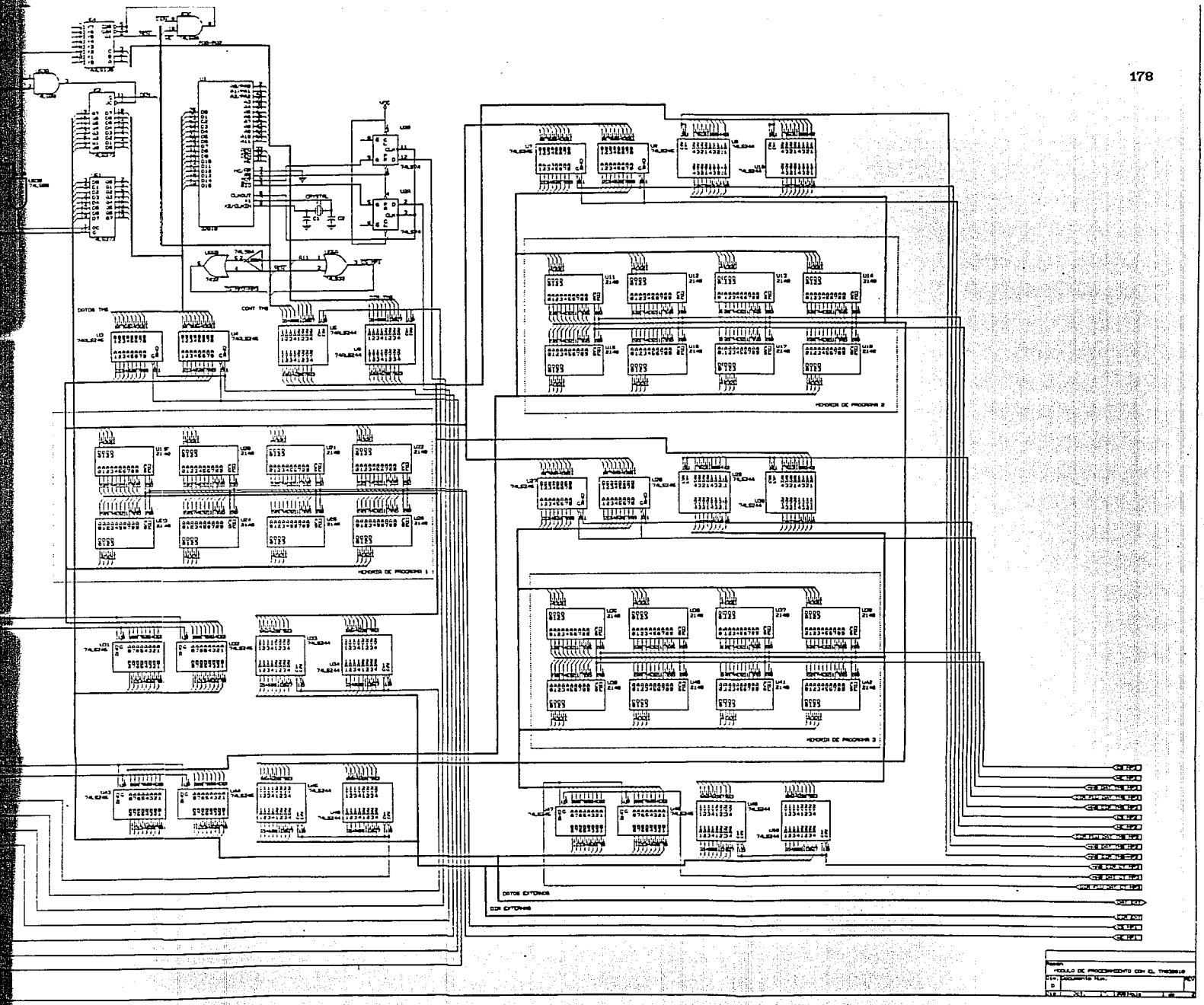
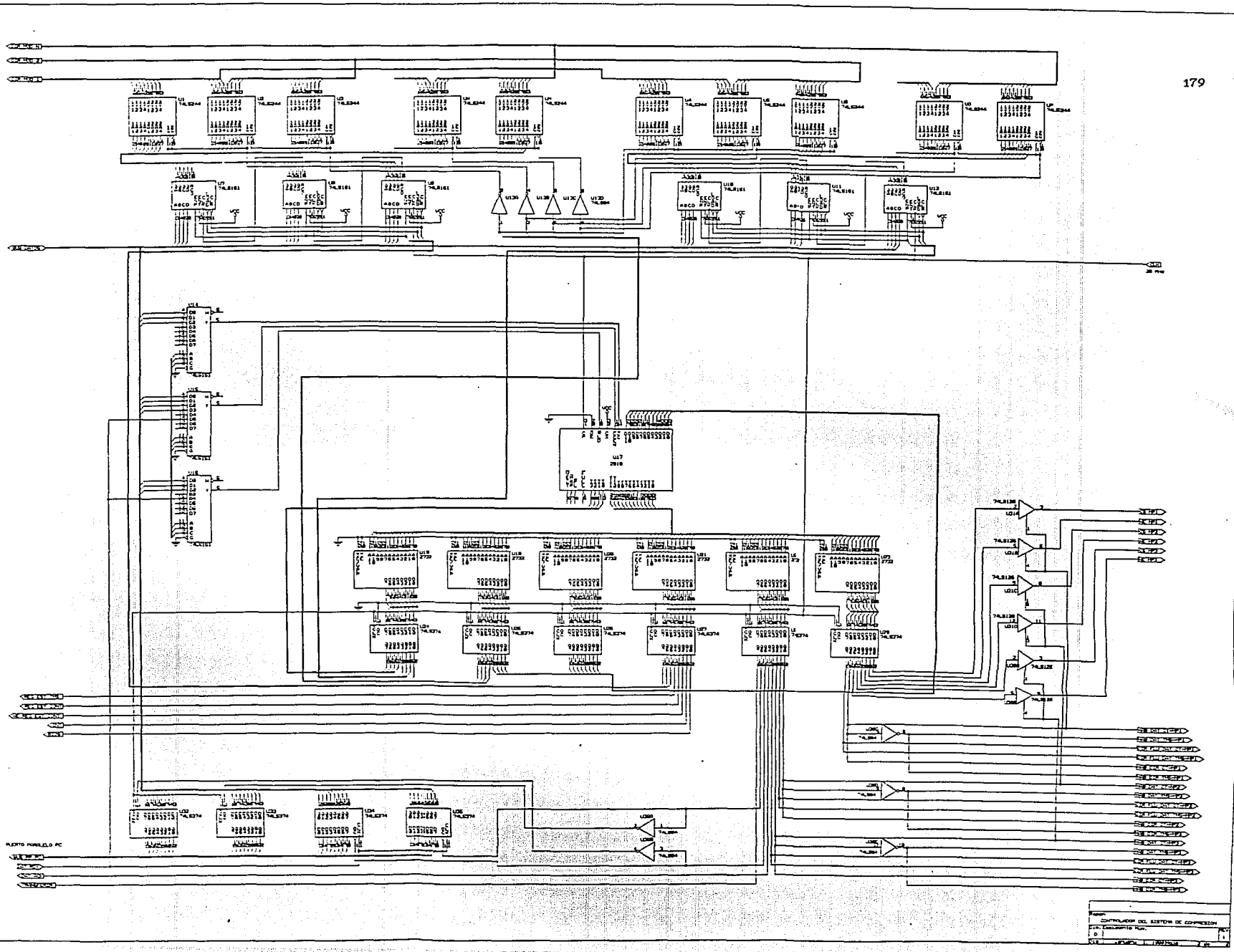


Figura 6.4 Diagrama de bloques del secuenciador 2910.

Finalmente, a continuación se presentan los diagramas del sistema descrito en este capítulo, uno de los diagramas corresponde a los módulos de proceso y el otro al controlador.


MEMORIA DE PROGRAMA 1
MEMORIA DE PROGRAMA 2
MEMORIA DE PROGRAMA 3
DATOS EXTERNOS
DIR EXTERNAS


DATOS TMS
CONT TMS
MEMORIA DE PROGRAMA 1
MEMORIA DE PROGRAMA 2
MEMORIA DE PROGRAMA 3
DATOS EXTERNOS
CCR EXTERNO


PUERTO PARALELO PC
CONTROLADOR DEL SISTEMA DE COMPRESION

VII. CONCLUSIONES Y RECOMENDACIONES.

1. Se presentó un esquema de compresión de señales de voz, que permite obtener una calidad aceptable a tasas de transmisión de hasta 2400 bits por segundo.

2. Modificando el algoritmo de escalamiento armónico en el dominio del tiempo, se obtiene una mayor relación de compresión comparada con la que se logra a partir del algortimo normal.

3. Al codificar los espectros de amplitud de la señal de voz, obtenidos a partir de la Transformada Discreta de Fourier y por predicción lineal (LPC), y efectuar la asignación de bits basados en el espectro de amplitud por predicción lineal, se logra recuperar con muy buena precisión en el receptor el espectro de amplitud de la señal.

4. Para la realización física del sistema de compresión se propone una arquitectura basada en el procesador aritmético TMS32010, con la

cual se puede observar la operación del sistema en tiempo real.

5. Se recomienda realizar los siguientes puntos para mejorar el sistema :

- Simulaciones en lenguaje de alto nivel del sistema operando en punto fijo, ya que de esta forma trabajará en el procesador aritmético.

- Cambiar el esquema de cuantización vectorial por búsqueda completa a cuantización vectorial por árbol, para reducir el tiempo de proceso.

- Buscar una forma de asignación de bits más eficiente para las fases, pues como se observa en las figuras 5.21 y 5.22, no se realiza un buen trabajo.

- Observar el efecto de codificar parte real y parte imaginaria, en lugar de amplitud y fase de la Transformada de Fourier, con el mismo patrón de asignación de bits.

B I B L I O G R A F I A

1. A.V.Oppenheim and R.W.Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ : Prentice Hall 1975.
2. J.C. Bellamy, *Digital Telephony*. Wiley Interscience, 1982.
3. A.V.Oppenheim, *Applications of Digital Signal Processing*. Englewood Cliffs, NJ : Prentice Hall 1978.
4. L.R.Rabiner and R.W.Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ : Prentice Hall 1978.
5. J.L.Flanagan, C.H.Coker, L.R.Rabiner, R.W.Schafer, and N.Umeda, " Synthetic voices for computers ", *IEEE Spectrum*, OCTUBRE 1970, 22-45.
6. I.H.Witten, *Principles of Computer Speech*. Academic Press 1982.
7. M.D.Paez and T.H.Glisson, " Minimun mean - squared - error quantization in speech PCM and DPCM systems ", *IEEE Transactions on Communications*, ABRIL 1972, 225-230.
8. P.E.Papamichalis, *Practical Approaches to Speech Coding*. Englewood Cliffs, NJ : Prentice Hall, 1987.
9. T.J.Lynch, *Data Compression Techniques and Applications*. Lifetime Learning Publications Belmont California, 1985.
10. R.W.Schafer and L.R.Rabiner, " Digital representations of speech signals ", *Proceedings of the IEEE*, VOL 63, NO. 4 , ABRIL 1975, 662-677.
11. J.L.Flanagan, M.R.Schroeder, B.S.Atal, R.E.Crochiere,N.S.Jayant and J.M.Tribolet, " Speech coding ", *IEEE Transaction on Communications*, VOL COMM 27, NO.4, ABRIL 1979, 710-736.

12. R.M.Gray, " Vector quantization ", *IEEE ASSP Magazine*, ABRIL 1984, 4-29.

13. Y.Linde, A.Buzo and R.M.Gray, " An algorithm for vector quantizer design ", *IEEE Transaction on Communications*, VOL COMM 28, NO.1, ENERO 1980, 84-95.

14. N.S.Jayant, " Coding speech at low bit rates ", *IEEE Spectrum*, AGOSTO 1986, 58-63.

15. L.R.Rabiner and B.Gold, *Theory and Applications of Digital Signal Processing* . Englewood Cliffs, NJ : Prentice Hall, 1975.

16. J.Makhoul, " Linear prediction: a tutorial review ", *Proceedings of the IEEE*, ABRIL 1975, 561-580.

17. L.Rabiner, M.J.Cheng, A.E.Rosenberg, and C.A.McGonegal, " A comparative performance study of several pitch detection algorithms ", *IEEE Transactions on Acoustics, Speech and Signal Processing*, VOL. ASSP 24, No. 5, OCTUBRE 1976, 399-417.

18. D.Malah, " Time domain algorithms for harmonic bandwith reduction and time scaling of speech signals ", *IEEE Transactions on Acoustics, Speech and Signal Processing*, VOL ASSP 26, ABRIL 1978, 121-133.

19. D.Malah, R.E.Crochiere and R.V.Cox, " Performance of transform and subband coding systems combined with harmonic scaling of speech ", *IEEE Transaction on Acoustics, Speech and Signal Processing*, VOL ASSP 29, NO.2, ABRIL 1981, 273-283.

20. R.E.Crochiere, R.V.Cox and J.D.Johnston, " Real-time speech coding ", *IEEE Transaction on Communications*, VOL COMM 30, NO.4, ABRIL 1982, 621-634.

21. J.Makhoul, S.Roucos and H.Gish, " Vector quantization in speech coding ", *Proceedings of the IEEE*, VOL 73, NO.11, NOVIEMBRE 1982, 1551-1558.

22. R.W.Hamming, *Coding and Information Theory*. Englewood Cliffs, NJ : Prentice Hall, 1980.

23. A.Buzo. A.H.Gray Jr, R.M.Gray and J.D.Markel, " Speech coding based upon vector quantization ", *IEEE Transaction on Acoustics, Speech and Signal Processing*, VOL ASSP 28, NO.5, OCTUBRE 1980, 562-574.

24. A.Buzo. Comunicación personal.

25. N.Kitawaki, H.Nagabuchi and K.Itoh," Objetive qualty evaluation for low-bit-rate speech coding systems ", *IEEE Journal on Selected Areas in Communications*, VOL 6, NO.2, FEBRERO 1988, 242-247.

26. Varios, *IEEE Micro*, DICIEMBRE 1986, 6-48.

27. K.S.Lin, G.A.Frantz and R.Simar Jr., " The TMS320 family of Digital Signal Processors ", *Proceedings of the IEEE*, VOL 75, NO.9. SEPTIEMBRE 1987, 1143-1159.

28. Texas Instrument, *Digital Signal Processing Applications with the TMS320 Family*, Volumen 1. Englewood Cliffs, NJ : Prentice Hall 1987.

29. A.S.Jackson, " A unique concurrent processor architecture for high speed simulation ", Motorola Inc., Orange, CA. AR234

30. *TMS32010 User's Guide*. Texas Instrument.

31. Hojas de datos del Am2910A, Advanced Micro Devices. DICIEMBRE 1984.