183

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

"DISEÑO Y REALIZACION DE UN SISTEMA DE SINTESIS DE SEÑALES DE VOZ."

T E S I S

QUE PARA OBTENER EL TITULO DE :

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A :

JOSE EDUARDO TORRES FERNANDEZ

MEXICO, D. F. 1995.

Agradecimientos

A la Universidad Nacional Autónoma de México, en especial a la Facultad de Ingeniería.

A la División de Estudios de Posgrado de la Facultad de Ingeniería.

Al Dr. Rogelio Alcántara Silva por dirigir este trabajo, por su amistad, y por su apoyo.

A la Dirección General de Asuntos del Personal Académico de la U.N.A.M. por su apoyo económico.

A mis compañeros del Departamento de Eléctrica de la División de Estudios de Posgrado de la Facultad de Ingeniería.

Dedicatoria

A mis padres José Eduardo y María Florentina por el apoyo incondicional para todo y siempre.

En recuerdo de mi abuela María Florentina y de la Madre Patria.

A toda mi familia.

A mis amigos en especial a Alejandro Nieto y a Verónica Andrade.

A mis maestros en especial a mi Maestra Blanca Arcelia.

A los corredores de los Viveros de Coyoacan, porque gracias a ellos aprendí que las carreras no son tan sólo una lucha física, sino también un ajedrez mental. De tal suerte, que el vencedor no lo es, porque corrió más aprisa sino porqué realmente creyó ganar.

A todos los aventureros y vagabundos de corazón.

En recuerdo de los aventureros George Mallory y Andrew Irvine quienes en 1924 perdierón la vida, intentando lo que en áquel entonces parecía imposible: La Conquista del Everest.

A los que no creen en los imposibles.

A áquel lugar del centro, espacio cerrado de media luz, viejo laberinto de piedra con pretensiones de grandeza.

"Las batallas de la vida raramente son ganadas por el hombre más fuerte o por el que corre más aprisa; por lo regular, el que gana es quien cree que puede ganar."

F. Arthur Clark

Indice

"Introducción."

La dificultad principal en el desarollo de cualquier aplicación, en la cuál se efectue algún tipo de procesamiento digital de señales, proviene de la gran cantidad de datos resultante de la conversión analogica digital (A/D). El procesamiento de un gran volúmen de datos requiere la toma de decisiones fundamentales en el desarollo e implementación de cualquier sistema que efectúe algún tipo de procesamiento. Estas decisiones son relativas a la cantidad de memoria requerida, el tiempo de procesamiento y, la tasa de transmisión. Resulta obvio que al aumentar el número de datos que requieren de un procesamiento, mayor será el espacio en memoria requerido, el tiempo de procesamiento y. la tasa de transmisión. El aumento de la cantidad de memoria encarece el sistema, el aumento en el tiempo de procesamiento no es deseable ya que la tendencia actual es hacia la implementación de sistemas en tiempo real y, el aumento en la tasa de transmisión mermará el número de señales que puedan transmitirse por el mismo canal de comunicaciones.

A pesar de los avances en la microelectrónica y de las técnicas de procesamiento en paralelo, siempre será de fundamental importancia la extracción de la información necesaria y suficiente para la reconstrucción de la señal, así como la eliminación de la información reduntante. Para tal efecto existen algoritmos de compresión y síntesis, los cuales producen una disminusión notable del número de datos a almacenar, procesar y/o transmitir por un sistema. Esta reducción trae como consecuencia un decremento del número de circuitos integrados utilizados por el sistema, reduce el tiempo de procesamiento, lo cuál abre paso a la posibilidad de una ejecución en tiempo real; y se reduce la tasa de transmisión, lo cuál es beneficioso si se toma en cuenta el hecho de que siempre se dispondrá de un ancho de banda limitado para la transmisión de las señales y, que se puede utilizar el mismo canal de comunicaciones para la transmisión de otras señales.

En el caso de las señales voz, éstas presentan redundancia, la cuál puede eliminarse mediante métodos de compresión y/o síntesis. La elección de uno de los dos métodos dependerá de la relación entre la calidad perceptual de la señal y de la tasa de transmisión deseable. Si se desea una tasa de transmisión mínima se tendrá que recurrir a un método de síntesis de voz; de otra manera, se utilizará algún método de compresión.

Los métodos de síntesis de voz recurren a un modelo para llevar a cabo la síntesis de voz. Dicho modelo trata de emular tanto el comportamiento del conducto vocal humano, así como las características del sistema perceptual humano. A pesar de que este modelo funciona y da resultado, gran parte de las dificultades derivadas por el uso de este modelo se deben a que la mayoría de los procesos perceptuales que ocurren a nivel cerebral son desconocidos y, por lo tanto no pueden ser tomados en cuenta por dicho modelo.

Dentro de los algoritmos de síntesis de voz existen aquellos que necesitan determinar si el segmento en procesamiento presenta o no periodicidad, con el objeto de determinar si la señal de excitación para el modelo es un tren de pulsos periodicos ó si es ruido blanco y; existen aquellos metodos que no necesitan determinar si el segmento en procesamiento presenta periodicidad, ya que la excitación consiste en un número determinado de pulsos de diferentes amplitudes y posiciones (métodos de excitación multipulso).

Los métodos de excitación multipulso poseen una tasa de transmición mayor que la correspondiente a los métodos de síntesis en los cuales se determina si el segmento de voz en procesamiento presenta periodicidad (se detecta el pitch). Esto se debe a que en lugar de transmitir un párametro que indique la periodicidad del segmento de voz y, otro parámetro que indique el valor de dicha periodicidad; en los métodos de excitación multipulso se debe transmitir tanto las amplitudes de todos los pulsos como sus posiciones. Este aumento en la tasa de transmisión, que con llevan los metodos de excitación multipulso, es a cambio de una mejor calidad perceptual de las señales sintéticas.

En este trabajo se realiza el estudio de los siguientes algoritmos de síntesis de voz:

I)Síntesis de voz mediante una codificación lineal predictiva (LPC), y estimación del pitch mediante los siguientes algoritmos:

a)Algoritmo de autocorrelación con sujetador central para la estimación del pitch (AUTOC).

b)"Simplified Inverse Filter Transform filter" (SIFT).

c) Método cepstral para la estimación del pitch.

II) Método de excitación Multipulso.

III) "Regular Pulse Excitation" (RPE).

Se efectuarón diversas pruebas con el objeto de evaluar el desempeño de cada uno de los anteriores algoritmos. Las pruebas principalmente consistierón en la variación de los parámetros más significativos de de cada uno de los algoritmos mencionados anteriormente; con el objeto de determinar un rango para estos parametros, dentro del cuál la calidad perceptual de las señales sintéticas sea la más alta posible y no varie de una manera notable.

También se efectuó una comparación de resultados entre los métodos estudiados. En esta comparación no sólo se tomó en cuenta la calidad perceptual de las señales sintéticas, sino también se tomarón en cuenta los factores fundamentales en el desarollo de cualquier algoritmo de síntesis de voz: el tiempo de procesamiento y, la tasa de transmisión.

En el capítulo uno se presentan, de una forma breve los algoritmos pertenecientes al procesamiento digital de señales en general (no sólamente señales de voz), que se utilizarón en la implementación de los algoritmos de síntesis de voz. Estos algoritmos son: la transformada rápida de Fourier (FFT), la transformada rápida de Hartley, la correlación de señales temporales y la estimación espectral.

En el capítulo dos se presentan las características de las señales de voz, el sistema humano para la producción y percepción de la voz y, se finaliza con una exposición somera de las técnicas para el modelado de las señales de voz como lo son: el énfasis espectral, el análisis espectral, la transformación de parametros y, la modelización estadística.

En el capítulo tres se hace una descripción detallada de cada uno de los algoritmos estudiados en este trabajo. Esta descripción se acompaña de los diagramas de bloques correpondientes a cada uno de los métodos de síntesis, además de comentarios que ilustran las ventajas y desventajas de cada uno de los métodos.

En el capítulo cuatro se presentan una serie de pruebas relativas a la variación de los parametros más significativos de cada uno de los métodos de síntesis estudiados y cómo, dicha variación afecta a la calidad perceptual de las señales sintéticas generadas por los diversos algoritmos.

También en el capítulo cuatro se presenta una comparación a nivel de tiempo de procesamiento, tasas de transmisión y calidad perceptual de las señales sintéticas; entre los métodos de síntesis de voz estudiados.

Finalmente se presentan las conclusiones que se obtuvieron y, se comparten algunas de las experiencias obtenidas durante su desarollo.

Con este trabajo se busca seleccionar el mejor algoritmo de síntesis para poder aplicarlo en compresión, reconocimiento y, encriptado de señales de voz, así como también a otras areas del procesamiento digital de señales como lo es el análisis espectral.

Capítulo Uno

"Algoritmos para el Procesamiento de Señales."

I.1 "Introducción."

Tanto en el desarrollo teorico como en la implementación de sistemas, en los que se realiza algún tipo de procesamiento digital de señales, un factor de importancia fundamental para el desarollo de dichos sistemas, entre otros, el tiempo de cálculo, la cantidad de recursos computacionales que requiere dicho procesamiento, así como también el costo de los mismos.

En la actualidad, debido al desarollo de técnicas y tecnologias tales como el procesamiento en paralelo, "algoritmos rápidos", las técnicas de segmentación de algoritmos secuenciales en tareas que se puedan ejecutar en paralelo, desarollo de procesadores dedicados al procesamiento digital señales, superconductores cerámicos y otras maravillas de la civilización del hombre moderno; lo que se persigue es que la mayoría de las aplicaciones tales como: dispositivos controlados por voz, sistemas de seguridad que requieren la identificación del parlante, sistemas multimedia para la transmisión de voz e imágenes en tiempo real, etc.; se ejecuten en tiempo real.

Además, independientemente de que una ejecución en tiempo real sea requerida ó no, otro factor que impone el uso de algoritmos eficientes es el gran volumen de datos que usualmente intervienen en un sistema de procesamiento. Esto proviene del "Teorema de Muestreo", el cuál nos dice que para evitar los efectos del "aliasing", la frecuencia de muestro debe ser de por lo menos el doble de la frecuencia máxima presente en la señal a ser muestreada. De esta manera según el tipo de aplicación será la cantidad de datos a procesar, por ejemplo: en aplicaciones relativas a la voz, la frecuencia máxima de la voz se considera de 4000 a 5000 Hz, lo que implica una frecuencia de muestreo de unos 8 a 10 kHz., es decir, 8000 o 10 000 datos por segundo. En aplicaciones de audio la frecuencia usual de muestreo es de 40 kHz, por lo que estamos hablando de 40000 datos por segundo y, finalmente en una aplicación de video se llega al orden de millones de datos por segundo. Este volumen de datos hace necesario el uso de algoritmos eficientes, ya que de lo contrario el sistema puede resultar irrealizable.

Afortunadamente existen una serie de algoritmos denominados "rápidos", ya que efectúan la misma tarea que los algoritmos a partir de los cuales fueron desarollados, solamente que en un número menor de operaciones y por tanto en un tiempo mucho menor. Estos algoritmos son indispensables en el desarollo e implementación de nuevos esquemas para el procesamiento de cualquier tipo de señales. Aquí se discutirán de una manera breve los algoritmos que se emplearon en el desarollo de las técnicas de síntesis implementadas; estos son: transformada rápida de Fourier, transformada rápida de Hartley, algoritmos de correlación y algunas técnicas de estimación espectral.

I.2 "Transformada Rápida de Fourier."

I.2.1 "Introducción."

La evaluación directa de la transformada discreta de Fourier (DFT), involucra N multiplicaciones complejas y N-1 adiciones complejas para cada valor de la secuencia de entrada x[n], y dado que se determinarán N valores, N^2 multiplicaciones y N(N-1) adiciones son necesarias. Cosecuentemente, para valores grandes de N, la evaluación directa de la DFT involucrá una gran cantidad de opreraciones a realizar.

La manera efeciente de evaluar la DFT, es a través del uso de los algoritmos rápidos, comúnmente referidos como "transformada rápida de Fourier (FFT)". Aquí sólamente se describirá el algoritmo de decimación temporal, para una mayor referencia consúltese [ANT79].

I.2.2 "Algoritmo FFT. Radix-2 con Decimación Temporal."

La DFT está dada por:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{-kn}$$

donde $W_N = e^{j2\pi/N}$
Asumamos que:

$$N = 2^r$$

donde r es un entero. La suma anterior se puede partir en dos:

$$X(k) = \sum_{n_{par}=0}^{N-1} x(n) W_N^{-kn} + \sum_{n_{impar}=0}^{N-1} x(n) W_N^{-kn}$$

Alternativamente:

$$X(k) = \sum_{n=0}^{N/2-1} x_{10}(n) W_N^{-2kn} + W_N^{-k} \sum_{n=0}^{N/2-1} x_{11}(n) W_N^{-2kn} \qquad (1.1)$$

donde:

$$x_{10}(n) = x(2n)$$
$$x_{11}(n) = x(2n+1) \qquad (1.2)$$

para $0 \leq n \leq N/2\text{-}1$. Dado:

$$W_N^{-2kn} = W_{N/2}^{-kn}$$

La ec.(1.1) puede ser expresada como:

$$X(k) = \sum_{n=0}^{N/2-1} x_{10}(n) W_{N/2}^{-kn} + W_N^{-k} \sum_{n=0}^{N/2-1} x_{11}(n) W_{N/2}^{-kn}$$

Evidentemente:

$$X(k) = X_{10}(k) + W_N^{-k} X_{11}(k) \tag{1.3}$$

y dado que $X_{10}(k)$ y $X_{11}(k)$ son periodicas, cada una con un periodo N/2, tenemos:

$$\begin{aligned} X\left(k + \frac{N}{2}\right) &= X_{10}\left(k + \frac{N}{2}\right) + W_N^{-(k+N/2)} X_{11}\left(k + \frac{N}{2}\right) \\ &= X_{10}(k) - W_N^{-k} X_{11}(k) \end{aligned} \tag{1.4}$$

Lo que se ha hecho hasta este punto, es expresar la DFT de N puntos como función de dos DFT's de (N/2) elementos. El proceso anterior se repite para X_{10} y X_{11}, de tal manera que estas se expresan como DFT's de (N/4) elementos. Este proceso se continua hasta llegar al r-ésimo ciclo, donde $r = log_2 N$. De esta manera el m-ésimo ciclo del procedimiento da los siguientes resultados intermedios:

$$X_{(m-1)0}(k) = X_{m0}(k) + W_N^{-2(m-1)k} X_{m1}(k)$$

$$X(m-1)0\left(k + \frac{N}{2^m}\right) = X_{m0}(k) - W_N^{-2(m-1)k} X_{m1}(k)$$

$$X_{(m-1)1}(k) = X_{m2}(k) + W_N^{-2(m-1)k} X_{m3}(k)$$

$$X_{(m-1)1}\left(k + \frac{N}{2^m}\right) = X_{m2}(k) - W_N^{-2(m-1)k} X_{m3}(k)$$

donde:

$$\begin{aligned} x_{m0}(n) &= x_{(m-1)0}(2n) \\ x_{m1}(n) &= x_{(m-1)0}(2n+1) \\ x_{m2}(n) &= x_{(m-1)1}(2n) \\ x_{m3}(n) &= x_{(m-1)1}(2n+1) \end{aligned} \tag{1.5}$$

Al llegar al r-ésimo ciclo todas las secuencias de datos constan de un sólo elemento:

$$X_{ri}(0) = x_{ri}(0)$$

para i = 0, 1, $\cdots$, N - 1. Los valores de las penúltimas DFT's pueden ser obtenidas de las ecuaciones (1.5):

$$X_{(r-1)0}(0) = x_{r0}(0) + W_N^0 x_{r1}(0)$$

$$X_{(r-1)0}(1) = x_{r0} - W_N^0 x_{r1}(0)$$
$$X_{(r-1)1}(0) = x_{r2}(0) + W_N^0 x_{r3}(0)$$
$$X_{(r-1)1}(1) = x_{r2}(0) - W_N^0 x_{r3}(0)$$

Asumiendo que la sequencia $\{x_{r0}(0), x_{r1}(0), \cdots\}$ está disponible en un arreglo, los valores de $X_{(r-1)i}(k)$ para i = 0, 1, $\cdots$ pueden ser calculados. Después los valores de $X_{(r-2)i}(k)$, $X_{(r-3)i}(k)$, $\cdots$ pueden ser calculados en secuencia, y finalmente los valores de X(k) son obtenidos.

La única tarea que resta es la identificación de los elementos $x_{r0}(0), x_{r1}(0), \cdots$!. Como se puede mostrar, $x_{rp}(0)$ está dado por:

$$x_{rp}(0) = x(q)$$

donde q es la representación binaria en r bits de p en orden inverso.

I.3 "Transformada Rápida de Hartley."

I.3.1 "Introducción."

La transformada discreta de Hartley (DHT) usa la variable real $cos(2\pi kn/N) + sin(2\pi kn/N)$ como el kernel de transformación, mientrás que la transformada discreta de Fourier (DFT) usa la exponencial compleja, $exp(i2\pi kn/N)$, como el kernel de transformación. Por tanto, la transformada de Hartley mapea una función real del tiempo x[n] en una función real de la frecuencia H(k), debido a lo cuál, sólo se necesitan operaciones artiméticas simples para su cálculo [ONE88].

Además, los arreglos de datos reales requieren de la mitad de memoria que utilizan los arreglos de datos complejos, lo cuál es de importancia en el desarollo de sistemas, en los que, la memoria sea limitada.

I.3.2 "Definición de la Transformada de Hartley."

La ecuación (1.6) es la forma analítica de la transformada de Hartley para el tiempo continuo, y la ec.(1.7) es la transformada inversa de Hartley para la frecuencia

continua:

$$H(f) = \frac{1}{2\pi}\int_{-\infty}^{\infty} x(t)cas(2\pi ft)dt \tag{1.6}$$

$$X(t) = \int_{-\infty}^{\infty} H(f)cas(2\pi ft)df \tag{1.7}$$

donde $cas(2\pi ft) = cos(2\pi ft) + sin(2\pi ft)$.

Para el caso discreto, la transformada y transformada inversa de Hartley, se definen de la siguiente manera:

$$H(k) = \frac{1}{N}\sum_{n=0}^{N-1} X(n)cas(2\pi kn/N) \tag{1.8}$$

y

$$X(n) = \sum_{k=0}^{N-1} H(k)cas(2\pi kn/N) \tag{1.9}$$

La transformada discreta de Hartley (DHT), se puede expresar matricialmente:

$$H = \frac{1}{\sqrt{N}}TX \tag{1.10}$$

donde:

$$H^T = [k_0\ k_1\ k_2\ \cdots\ k_{N-1}]$$

$$T_{[NXN]} = \left[\ cas(2\pi kn/N)\ \right]$$

$$X^T = [x_0\ x_1\ x_2\ \cdots\ x_{N-1}]$$

I.3.3 "Transformada Rápida de Hartley."

El cálculo de la transformada discreta de Hartley (DHT), presenta problemas análogos al cálculo de la transformada discreta de Fourier (DFT). Esto es, para el cálculo de una DHT para N datos, es necesario realizar N^2 operaciones aritméticas.

La transformada rápida de Fourier utiliza un proceso de permutación para bisectar los datos hasta obtener pares de datos, para los cuáles, el cálculo de la transformada de Fourier es trivial [ONE88].

La idea de efectuar un proceso de permutación, es que resulta más rápido dividir los datos en pares, calcular la transformada de dichos pares, y recombinar los pares transformados, para obtener la transformada del conjunto de N datos, que calcular la transformada para el conjunto entero de datos.

El proceso de permutación es particularmente efectivo y rápido cuando el número de datos a transformar es grande. Si se sobreponen todos los pares en

una esctructura llamada "mariposa" [ONE88], se puede calcular la transformada de Fourier para el conjunto de N datos. Este cálculo requiere de Nlog(N) operaciones aritméticas.

Bracewell demostró que es factible emplear una metodología similar en el caso de la transformada discreta de Hartley. Nuevamente, se utiliza el proceso de permutación para bisectar los datos, hasta obtener pares de datos.

El cálculo de la transformada de Hartley para un par de datos es trivial, por lo que, se pueden superponer las secuencias de dos elementos para calcular la transformada de Hartley del conjunto de datos de entrada. Sin embargo, para llevar a cabo dicho cálculo, se requiere de una fórmula que exprese la transformada de Hartley del conjunto de datos, en términos de sus subsecuencias de media longitud [ONE88].

Bracewell demostró mediante la aplicación del teorema del desplazamiento y del teorema de la similaridad que, la ecuación (1.11) es la fórmula general de descomposición para la DHT. Esta fórmula genera la DHT deseada, mediante la bisección de los datos.

Dicho de otra manera, es la regla usada para generar los elementos a ser usados en la mariposa, para el cálculo de la transformada. También se puede aplicar una metodología similar a la de la transformada de Fourier, para llegar a la ecuación (1.12):

$$H(k) = H_1(k) + H_2(k)cos(2\pi k/N_s) + H_2(N_s - k)sin(2\pi k/N_s) \qquad (1.11)$$

$$F(k) = F_1(k) + F_2(k)e^{j2\pi k/N_s} \qquad (1.12)$$

donde N_s es el número de elementos en la secuencia de media longitud, esto es $N_s = N/2$ para un conjunto de N elementos.

La fórmula de descomposición para la FHT difiere de la de la FFT, en un aspecto importante: los elementos multiplicados por los términos trigonométricos no son simétricos.

una esctructura llamada "mariposa" [ONE88], se puede calcular la transformada de Fourier para el conjunto de N datos. Este cálculo requiere de Nlog(N) operaciones aritméticas.

Bracewell demostró que es factible emplear una metodología similar en el caso de la transformada discreta de Hartley. Nuevamente, se utiliza el proceso de permutación para bisectar los datos, hasta obtener pares de datos.

El cálculo de la transformada de Hartley para un par de datos es trivial, por lo que, se pueden superponer las secuencias de dos elementos para calcular la transformada de Hartley del conjunto de datos de entrada. Sin embargo, para llevar a cabo dicho cálculo, se requiere de una fórmula que exprese la transformada de Hartley del conjunto de datos, en términos de sus subsecuencias de media longitud [ONE88].

Bracewell demostró mediante la aplicación del teorema del desplazamiento y del teorema de la similaridad que, la ecuación (1.11) es la fórmula general de descomposición para la DHT. Esta fórmula genera la DHT deseada, mediante la bisección de los datos.

Dicho de otra manera, es la regla usada para generar los elementos a ser usados en la mariposa, para el cálculo de la transformada. También se puede aplicar una metodología similar a la de la transformada de Fourier, para llegar a la ecuación (1.12):

$$H(k) = H_1(k) + H_2(k)cos(2\pi k/N_s) + H_2(N_s - k)sin(2\pi k/N_s) \qquad (1.11)$$

$$F(k) = F_1(k) + F_2(k)e^{j2\pi k/N_s} \qquad (1.12)$$

donde N_s es el número de elementos en la secuencia de media longitud, esto es $N_s = N/2$ para un conjunto de N elementos.

La fórmula de descomposición para la FHT difiere de la de la FFT, en un aspecto importante: los elementos multiplicados por los términos trigonométricos no son simétricos.

I.4 "Correlación de Señales en el Tiempo Discreto."

I.4.1 "Introducción."

La correlación entre dos señales es una medida del grado de similitud entre estas [PROA88]. Para ser específicos, supongamos que tenemos dos señales, x(n) e y(n), las cuales queremos comparar. En aplicaciones de radar o sonar, x(n) puede representar la señal transmitida e, y(n) representaría la señal recibida. Si un objeto está presente en el espacio de búsqueda del radar o sonar, la señal recibida y(n) es una versión retrasada de la señal transmitida (es la señal refejada por el objeto y además distorsionada por ruido aditivo). La señal recibida se puede expresar como [PROA88]:

$$y(n) = \alpha x(n - D) + w(n) \tag{1.13}$$

donde α es un factor de atenuación, que representa las pérdidas, que ocurren en la transmisión y rebote de x(n). D es el retraso, el cuál se asume que es un múltiplo del periodo de muestreo, y w(n) representa el ruido aditivo. Por otro lado, si no está presente algún objeto en el espacio de búsqueda del radar o sonar, la señal recibida, y(n), consiste solamente de ruido.

La secuencia x(n) es llamada señal de referencia o señal transmitida, y la secuencia y(n) es llamada señal recibida. El problema de la deteción por radar o sonar consiste en comparar y(n) con x(n), para determinar si existe un objeto presente y, en caso afirmativo, determinar el retraso temporal D, del cuál se cálcula la distancia a dicho objeto. En la práctica, la señal x(n - D) está fuertemente "contaminada" por ruido, por lo cuál, una inspección visual de y(n) no revelará la presencia o ausencia de la señal reflejada por el objeto. La correlación provee un medio importante para extraer dicha información de y(n) [PROA88].

Las comunicaciones digitales es otra área, dónde la correlación es frecuentemente usada. En las comunicaciones digitales, la transmición de información se realiza generalmente en forma binaria. Para transmitir un cero, se transmite la secuencia $x_0(n)$ para $0 \leq n \leq L - 1$, y para transmitir un uno, se transmite la secuencia $x_1(n)$ para $0 \leq n \leq L - 1$, dónde L es un entero que denota el número de bits en cada una de las dos secuencias. Frecuentemente, $X_1(n)$ se selecciona para que sea el negativo de $x_0(n)$. La señal recibida por el supuesto receptor, se puede representar como:

$$y(n) = x_i(n) + w(n) \qquad i = 0, 1 \qquad 0 \leq n \leq L - 1 \tag{1.14}$$

donde la incertidumbre consiste en saber si $x_0(n)$ ó $x_1(n)$ es la señal componente en y(n) y, w(n) representa el ruido.

I.4 "Correlación de Señales en el Tiempo Discreto."

I.4.1 "Introducción."

La correlación entre dos señales es una medida del grado de similitud entre estas [PROA88]. Para ser específicos, supongamos que tenemos dos señales, x(n) e y(n), las cuales queremos comparar. En aplicaciones de radar o sonar, x(n) puede representar la señal transmitida e, y(n) representaría la señal recibida. Si un objeto está presente en el espacio de búsqueda del radar o sonar, la señal recibida y(n) es una versión retrasada de la señal transmitida (es la señal refejada por el objeto y además distorsionada por ruido aditivo). La señal recibida se puede expresar como [PROA88]:

$$y(n) = \alpha x(n - D) + w(n) \tag{1.13}$$

donde α es un factor de atenuación, que representa las pérdidas, que ocurren en la transmisión y rebote de x(n). D es el retraso, el cuál se asume que es un múltiplo del periodo de muestreo, y w(n) representa el ruido aditivo. Por otro lado, si no está presente algún objeto en el espacio de búsqueda del radar o sonar, la señal recibida, y(n), consiste solamente de ruido.

La secuencia x(n) es llamada señal de referencia o señal transmitida, y la secuencia y(n) es llamada señal recibida. El problema de la deteción por radar o sonar consiste en comparar y(n) con x(n), para determinar si existe un objeto presente y, en caso afirmativo, determinar el retraso temporal D, del cuál se cálcula la distancia a dicho objeto. En la práctica, la señal x(n - D) está fuertemente "contaminada" por ruido, por lo cuál, una inspección visual de y(n) no revelará la presencia o ausencia de la señal reflejada por el objeto. La correlación provee un medio importante para extraer dicha información de y(n) [PROA88].

Las comunicaciones digitales es otra área, dónde la correlación es frecuentemente usada. En las comunicaciones digitales, la transmición de información se realiza generalmente en forma binaria. Para transmitir un cero, se transmite la secuencia $x_0(n)$ para $0 \leq n \leq L - 1$, y para transmitir un uno, se transmite la secuencia $x_1(n)$ para $0 \leq n \leq L - 1$, dónde L es un entero que denota el número de bits en cada una de las dos secuencias. Frecuentemente, $X_1(n)$ se selecciona para que sea el negativo de $x_0(n)$. La señal recibida por el supuesto receptor, se puede representar como:

$$y(n) = x_i(n) + w(n) \quad i = 0, 1 \quad 0 \leq n \leq L - 1 \tag{1.14}$$

donde la incertidumbre consiste en saber si $x_0(n)$ ó $x_1(n)$ es la señal componente en y(n) y, w(n) representa el ruido.

El receptor conoce las posibles secuencias transmitidas $x_0(n)$ y $x_1(n)$, por lo que su tarea es comparar la señal recivida y(n) con $x_0(n)$ y $x_1(n)$, para determinar cúal de las dos es la señal recivida. Esta comparación se realiza mediante la correlación de y(n) con $x_0(n)$ y $x_1(n)$ [PROA88].

I.4.2 "Secuencias de Autocorrelación y Correlación Mutua."

Supongamos que tenemos dos secuencias, x(n) e y(n), de energía finita. La correlación mutua de x(n) e y(n), se define como:

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l) \qquad l = 0, \pm 1, \pm 2, \cdots \tag{1.15}$$

o de manera equivalente:

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n+l)y(n) \qquad l = 0, \pm 1, \pm 2, \cdots \tag{1.16}$$

El índice l representa el atraso, el subíndice xy en la secuencia de correlación mutua $r_{xy}(l)$, respresenta las secuencias que son correlacionadas. El orden de los subíndices, x precediendo a y, indica la dirección en la cuál una secuencia es desplazada con respecto a la otra.

Si invertimos los papeles de x(n) e y(n) en (1.15) y (1.16), es decir, invertimos el orden de los indices xy, obtenemos:

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y(n)x(n-l) \tag{1.17}$$

de manera equivalente:

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y(n+l)x(n) \tag{1.18}$$

Al comparar (1.15) con (1.17) y (1.16) con (1.18), se concluye que:

$$r_{xy}(l) = r_{yx}(-l) \tag{1.19}$$

Por lo tanto, $r_{yx}(l)$ provee exactamente la misma información que $r_{xy}(l)$ con respecto a la similitud existente entre las secuencias x(n) e y(n) [PROA88].

Un caso de especial importancia ocurre cuando y(n) = x(n):

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n-l) \tag{1.20}$$

de manera equivalente:

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n+l)x(n) \tag{1.21}$$

donde $r_{xx}(l)$ se conoce como la secuencia de autocorrelación de x(n).

I.4.3 "Propiedades de las Secuencias de Autocorrelación y Correlación Mutua."

Sea la combinación lineal de las secuencias, x(n) e y(n), ambas de energía finita:

$$ax(n) + by(n-l)$$

donde a y b son constantes arbitrarias y, l es el atraso. La energía de esta señal es:

$$\sum_{n=-\infty}^{\infty} [ax(n) + by(n-l)]^2 = a^2 \sum_{n=-\infty}^{\infty} x^2(n) + b^2 \sum_{n=-\infty}^{\infty} y^2(n-l) + 2ab \sum_{n=-\infty}^{\infty} x(n)y(n-l)$$

$$= a^2 r_{xx}(0) + b^2 r_{yy}(0) + 2ab r_{xy}(l) \tag{1.22}$$

Primero, se debe hacer notar que $r_{xx}(0) = E_x$ y $r_{yy}(0) = E_y$, son las energías de x(n) e y(n), repectivamente [PROA88]. Es obvio que:

$$a^2 r_{xx}(0) + b^2 r_{yy}(0) + 2ab r_{xy}(l) \geq 0 \tag{1.23}$$

Si $b \neq 0$, podemos dividir (1.23) entre b^2:

$$r_{xx}(0)\left(\frac{a}{b}\right)^2 + 2r_{xy}(l)\left(\frac{a}{b}\right) + r_{yy}(0) \geq 0$$

Esta es una ecuación cuadrática con coeficientes $r_{xx}(0)$, $2r_{xy}(l)$, y $r_{yy}(0)$. Dado que la ecuación cuadrática es no negativa, se sigue que el discriminante debe ser no positivo [PROA88], esto es:

$$4\left[r_{xy}^2(l) - r_{xx}(0)r_{yy}(0)\right] \leq 0$$

Por lo tanto, la secuencia de correlación mutua satisface la siguiente condición [PROA88]:

$$|r_{xy}(l)| \leq \sqrt{r_{xx}(0)r_{yy}(0)} = \sqrt{E_x E_y} \tag{1.24}$$

En el caso especial de y(n) = x(n), la ecuación anterior se reduce a:

$$|r_{xx}(0)| \leq r_{xx}(0) = E_x \tag{1.25}$$

Esto significa que la secuencia de autocorrelación presenta su máximo valor cuando el retraso es cero. Este resultado es consistente con el hecho de que una señal se ajusta perfectamente a si misma cuando el retraso es cero. Al aumentar el retraso se espera que el valor de $r_{xx}(l)$ decresca, es decir, $r_{xx}(l)$ debería tender a cero cuando l tiende a infinito [PROA88].

En el caso de una secuencia de correlación mutua, el valor máximo de esta, está dado por (1.24). En caso de que:

$$y(n) = \pm cx(n - n_0) \tag{1.26}$$

donde c es un factor de escala arbitrario y n_0 es un retraso temporal, la correlación mutua $r_{xy}(l)$ se convierte en:

$$r_{xy}(l) = \pm cr_{xx}(l - n_0) \tag{1.27}$$

y la autocorrelación de y(n) en n = 0 es:

$$r_{yy}(0) = c^2 r_{xx}(0) \tag{1.28}$$

Sustituyendo (1.27) y (1.28) en (1.24), la desigualdad se reduce a:

$$|r_{xy}(l)| = |\pm cr_{xx}(l - n_0)| \leq cr_{xx}(0)$$

En este caso el rango de valores de $r_{xy}(l)$ es:

$$-cr_{xx}(0) \leq r_{xy}(l) \leq cr_{xx}(0) \tag{1.29}$$

En la práctica es deseable normalizar la secuencias de autocorrelación y correlación mutua al rango [-1,1] [PROA88]. En el caso de la secuencia de autocorrelación se aplica:

$$\rho_{xx}(l) = \frac{r_{xx}(l)}{r_{xx}(0)} \tag{1.30}$$

y en el caso de la correlación mutua:

$$\rho_{xy}(l) = \frac{r_{xy}(l)}{\sqrt{r_{xx}(0) r_{yy}(0)}} \tag{1.31}$$

Finalmente, las secuencias de autocorrelación y de correlación mutua son funciones pares [PROA88]:

$$\begin{aligned} r_{xy}(l) &= r_{yx}(-l) \\ r_{xx}(l) &= r_{xx}(-l) \end{aligned} \tag{1.32}$$

Por lo tanto, basta con calcular $r_{xx}(l)$ para $l \geq 0$.

I.5 "Estimación Espectral."

I.5.1 "Introducción."

El término de "Estimación Espectral" se refiere al conjunto de metodos utilizados para obtener el contenido frecuencial de una señal aleatoria, cuando sólo se tiene una pequeña parte de esta [MULLIS].

Antes de continuar, se debe definir lo que se entiende por "el contenido frecuencial" de una señal. Para una banda determinada de frecuencias Ω, podemos aislar la parte de la señal que cae en la banda Ω utilizando un filtro ideal paso banda, cuyo ancho de banda sea igual a Ω:

$$H(z) = \frac{Y(z)}{X(z)} \tag{1.33}$$

donde:

$$H(e^{j\theta}) = \begin{cases} 1, & \theta \in \Omega \\ 0, & \textit{cualquier otro caso} \end{cases} \tag{1.34}$$

En este experimento idealizado la señal **y**, representa lo que queremos decir por "la parte de **x** que cae en la banda Ω. Si la energía de **y** es finita, entonces esa energía es: "la energía de **x** en la banda Ω" [MULLIS]. Si **x** y **y** son señales estacionarias con potencia finita, entonces el promedio de potencia de **y** es "la potencia de **x** en la banda Ω" [MULLIS].

Se tendrá una respuesta completa a la pregunta del contenido frecuencial, si se pueden calcular las variables mencionadas en el párrafo anterior, para cualquier banda de frecuencias Ω [MULLIS]. Este es el propósito de la función de densidad espectral [MULLIS], la cuál tiene la siguiente propiedad:
"La energía o potencia de la señal en la banda Ω es la integral de la función de densidad sobre Ω."

Estamos interesados en dos clases de señales: las señales de energía finita, y las señales aleatorias y estacionarias. La primera clase tendrá una densidad de energía espectral [MULLIS]. La segunda clase tendrá una densidad de potencia espectral [MULLIS].

Densidad Espectral de Energía (ESD): asumamos que la señal:

$$x(k) \quad \longleftrightarrow \quad X(e^{j\theta}) \tag{1.35}$$

tiene energía finita:

$$||x||^2 = \sum_{k=-\infty}^{\infty} |x(k)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\theta})|^2 \, d\theta \tag{1.36}$$

Para extraer la porción de la energía que cae en la banda Ω, filtramos x a través de (1.34), de esta manera obtenemos una señal, y, de salida. Entonces, la energía total de y, es la parte de la energía de x que cae en Ω y, está dada por:

$$||y||^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |Y(e^{j\theta})|^2\, d\theta \tag{1.37}$$

Pero:

$$y = h * x \quad \longleftrightarrow \quad Y(e^{j\theta}) = H(e^{j\theta})X(e^{j\theta}) \tag{1.38}$$

y aplicando (1.34) obtenemos:

$$\text{"}energía\ de\ x\ en\ \Omega\text{"} = ||y||^2 = \frac{1}{2\pi}\int_{\Omega} |X(e^{j\theta})|^2\, d\theta \tag{1.39}$$

Esta es la propiedad de la función de densidad que requerimos, por lo tanto la ec.(1.36) es la función de densidad espectral de energía para la señal x.

Densidad Espectral de Potencia: Una señal aleatoria y estacionaria no tiene energía finita, pero puede tener un promedio finito de potencia. Si una señal, x, de este tipo además es ergódica; entonces los promedios temporales se aproximarán a los valores esperados [MULLIS]:

$$promedio\ de\ potencia = \lim_{L\to\infty} \frac{1}{L}\sum_{k=0}^{L-1} |x(k)|^2 = E|x(k)|^2 \tag{1.40}$$

Por tanto, el promedio de potencia es igual a la variancia de x(k), ó al primer elemento de la siguiente secuencia de autocorrelación [MULLIS]:

$$r_{xx}(k) = Ex(k+l)x^*(l) \quad \longleftrightarrow \quad S_{xx}(\theta) \tag{1.41}$$

Para hallar el promedio de potencia en la banda Ω, aplicamos la ec.(1.34) y aplicamos el teorema Wiener-Khintchine [MULLIS]:

$$r_{yy}(k) \quad \longleftrightarrow \quad S_{yy}(\theta) = S_{xx}(\theta)|H(e^{j\theta})|^2 \tag{1.42}$$

Por lo tanto,

$$\text{"}promedio\ de\ potencia\ de\ x\ en\ la\ banda\ \Omega = r_{yy}(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{yy}(\theta)\, d\theta \tag{1.43}$$

$$= \frac{1}{2\pi}\int_{\Omega} S_{xx}(\theta) d\theta \tag{1.44}$$

Por lo tanto, $S_{xx}(\theta)$ es la función de densidad espectral de potencia.

I.5.2 "Aproximaciones a la Estimación Espectral."

Sea **y[n]**, una señal estacionaria con densidad espectral de potencia $S(\theta)$. Supongamos que una porción, de L muestras, de la señal ha sido registrada y almacenada:

$$y[0, L-1] = \{y(0), y(1), \cdots, y(L-1)\} \tag{1.45}$$

El problema es estimar la función $S(\theta)$ dadas las L muestras. Por tanto, un estimador es un mapeo de los datos al espectro de potencia [MULLIS]:

$$\hat{S}(\theta) = \mathcal{L}(\theta; y[0, L-1]) \tag{1.46}$$

donde $\mathcal{L}$ es el estimador y, $\hat{S}$ es el estimado.

Estimación Espectral Usando la Función ESD: cuando sólo se conocen un número finito de muestras de la señal y[0, L-1], podemos construir una señal de energía finita, de la siguiente manera:

$$w(k) = \begin{cases} 1, & 0 \leq k \leq L-1 \\ 0, & \textit{en cualquier otro caso} \end{cases} \tag{1.47}$$

y

$$x(k) = w(k)y(k), \quad -\infty < k < \infty \tag{1.48}$$

donde w es una ventana (en este trabajo se define el término ventana como cualquier función, cuyo objetivo es ponderar a favor de las muestras del centro del segmento de voz (frame), sobre el que actúa. La duración de la ventana es excatamente igual a la duración del frame a menos que se indique lo contrario) y, **x** es una señal limitada en el dominio del tiempo. Dado que **x** es de energía finita, se puede calcular su densidad espectral de energía (ESD). La ESD con un factor de escala apropiado se usa para estimar la densidad espectral de potencia (PSD) de la señal **y** [MULLIS]:

$$\hat{S}(\theta) = \frac{1}{L}|X(e^{j\theta})|^2 \tag{1.49}$$

Este estimado se conoce con el nombre de "peridiograma" [MULLIS].

Estimación Espectral de los Valores de Autocorrelación: se tienen n valores de autocorrelación, a partir de los cuáles se desea estimar el espectro de potencia. Entonces, si asumimos que el espectro es de promedio móvil, el estimado está dado por [MULLIS]:

$$\hat{S}(\theta) = \sum_{k=-n}^{n} \hat{r}(k)e^{-jk\theta} \tag{1.50}$$

Este estimado se conoce con el nombre de "estimado de Blackman-Tukey".

Si asumimos que el espectro obedece un modelo autoregresivo (ver apéndice C), el estimado está dado por [MULLIS]:

$$\hat{S}(\theta) = \frac{\alpha}{|\sum_{k=0}^{n} a_k e^{-jk\theta}|^2} \tag{1.51}$$

donde $(\alpha, a_1, \cdots, a_n)$ se obtiene de la secuencia de autocorrelación mediante el algoritmo de Levinson (ver apéndice C).

Capítulo Dos

"Generalidades de las Señales de Voz."

II.1 "Percepción y Producción de Voz."

A continuación se describe brevemente el aparato vocal humano. El conducto vocal comienza en la glotis (abertura superior de la laringe) y término en los labios, consiste de la faringe (conexión entre el esófago y la boca), y la cavidad oral (boca). Las dimensiones del conducto vocal varían de persona a persona, no obstante para la longitud total del conducto vocal se puede hablar de un promedio de 17 cm para los hombres. Sin embargo en lo que respecta al área de la sección transversal del conducto vocal, esta varía acorde a la pronunciación de las palabras, pero depende de la posición de la lengua, labios, mandíbula y el velo del paladar. El rango de variación del área transversal del conducto vocal es de 0 a 20 cm^2. Otro conducto de importancia en el aparato vocal humano es el conducto nasal, el cuál principia en el velo del paladar y término en las fosas nasales. Cuando el velo del paladar está relajado, el conducto nasal está acústicamente acoplado al conducto vocal para producir los sonidos nasales [RAB93].
La voz se produce por el flujo estable de aire, que al escapar de los pulmones es parcialmente convertido en pulsos de acuerdo a los siguientes mecanismos [SCHR66]:

1) Por la acción de las cuerdas vocales, la cuál consiste en entrecortar el flujo estable de aire proveniente de los pulmones, dando lugar a pulsos de corta duración.
2)Por las constricciones del conducto vocal.
3)Por la liberación repentina de presión excesiva seguida de un bloqueo del conducto vocal, en algún punto del mismo.

De esta manera se gernera una señal acústica de ac [SCHR66], llamada función de excitación de la voz. Esta toma tres formas distintas en concordancia con el mecanismo por el cual fué generada [SCHR66]:

1)En caso de que la función de excitación halla sido generada por la acción de las cuerdas vocales, esta consiste en pulsos quasi-periodicos.
2)En caso de que la función de excitación halla sido generada por las constricciones del conducto vocal, esta consiste en ruido continuo.
3)En caso de que la función de excitación se deba a una liberación repentina de presión, esta consistirá de un sólo pulso.

La mayoría de los sonidos son generados por sólo una de las anteriores formas de la función de excitación. El modo 1 corresponde a los sonidos sonoros, el modo 2 corresponde a los sonidos no sonoros y el modo 3 corresponde a cierto tipo de sonidos no sonoros. Algunos sonidos se deben a una combinación de dos modos de la función de excitación [SCHR66].

El espectro de la función de excitación cubre un amplio rango de frecuencia, con componentes espectrales significativas a lo largo de la mayor parte del rango de audio frecuencia. Dado que las señales de voz tienen un ancho de banda limitado a 5 kHz aproximadamente, se deduce que el conducto vocal filtra la señal de excitación, además de modularla. La respuesta en frecuencia del conducto vocal es caracterizada por un número de resonancias o "formantes", los cuales modulan el espectro de la función de excitación y dan el timbre característico de cada sonido. El valor de los formants depende de la respuesta en frecuencia del conducto vocal, que a su vez depende de la posición de la lengua, labios y otros organos artículatorios; por lo que varía conforme a la pronunciación de los distintos sonidos, aunque generalmente son tres fomants por debajo de 3 kHz [SCHR66].

II.1.1 "Percepción Auditiva Humana."

Existen tres propiedades básicas del oído humano que se relacionan con el análisis y síntesis de señales de voz:

1)El oído ejecuta un análisis espectral en tiempo corto.
2)Para una percepción monoauditiva, el oído es relativamente insensible a la fase.
3)El oído es excesivamente sensitivo a la periodicidad (pitch) de las señales de voz.

Ohm y von Helmholtz propusieron que el oído es un analizador de espectros insensitivo a la fase [SCHR66]. De esta manera, modelaron la membrana basilar en el oído interno como un arreglo de resonadores sintonizados. Este modelo fué abandonado en favor de una línea de transmisión no uniforme con resolución espectral limitada [SCHR66], al quedar demostrada, por von Békésy, la existencia de ondas viajeras en la membrana basilar. La discriminación de frecuencias llevada a cabo por el sistema auditivo humano no puede ser atribuida a medios mecánicos, de manera que se asume ocurre a nivel neuronal.

La propuesta original según la cuál, el oído es totalmente "sordo" a la fase ha tenido que ser revisada. Sin embargo el hecho sigue siendo válido en el caso de la percepción monoauditiva, en la cuál la fase se puede considerar como un factor de menor importancia con alguna influencia en la calidad de la voz "sonora" escuchada en audífonos, pero sin afectar la inteligibilidad.

La percepción auditiva de la frecuencia fundamental (Pitch), se basa en un poderoso mecanismo de discriminación frecuencial, el cuál se desconoce pero se sabe que es un proceso a nivel cerebral.

II.2 "Técnicas para el Modelado de Señales de Voz."

El modelado de señales puede dividirse en cuatro operaciones básicas [PIC93]: énfasis espectral, análisis espectral, transformación de parámetros y modelización estadística. Existen tres consideraciones básicas en la elaboración de modelos para señales: en primer lugar, las paramétrizaciones tratan de representar aspectos salientes de las señales de voz. Aquí se prefieren parámetros que sean análogos a los usados por el sistema auditivo humano. Por tanto nos referimos a estos parámetros como parámetros perceptualmente significativos. En segundo lugar se desea que las paramétrizaciones sean robustas a variaciones en el canal de comunicaciones, parlante y transductor. Esto último es referido como el problema de la invariancia. Finalmente los parámetros que representan la dinámica del espectro ó cambios espectrales en el tiempo son deseables. Nos referimos a esto último como el problema de la correlación temporal.

Es necesario mencionar, el hecho de que los modelos que son buenos para determinado tipo de aplicación, pueden dar por resultado una solución subóptima para otro tipo de aplicación [PIC93].

II.2.1 "Énfasis Espectral."

El énfasis espectral consiste de dos operaciones básicas: la conversión A/D y el filtrado digital (énfasis de las componentes frecuenciales más importantes de la señal).

Una vez que la conversión A/D ha sido completada, el último paso del post-filtrado digital es llevado a cabo mediante un filtro de respuesta al impulso finita (FIR):

$$H_{pre}(z) = \sum_{k=0}^{N_{pre}} a_{pre}(k) z^{-k} \tag{2.1}$$

Normalmente se utiliza un sólo coeficiente para el filtro digital, comúnmente llamado filtro de preénfasis:

$$H_{pre}(z) = 1 + a_{pre}(1) z^{-1} \tag{2.2}$$

Un rango de valores típicos para a_{pre} es [-1.0,-0.4]. El propósito del filtro de preénfasis es levantar el espectro de la señal 20 dB por década apróximadamente. Existen dos argumentos para justificar el uso de este filtro. El primero explica que las secciones "sonoras" de la voz tienen una pendiente espectral negativa de 20 dB por década

apróximadamente, esta es debida a características fisiológicas del sistema de producción de voz [PIC93]. Entonces el filtro de preénfasis sirve para compensar los efectos de dicha pendiente negativa, antes de efectuar el análisis espectral, mejorando de esta manera la eficiencia de este último.

El segundo argumento se refiere a la mayor sensibilidad del sistema auditivo a componentes frecuenciales mayores a un kHz, la cuales son amplificadas por el filtro de preénfasis, ayudando de esta manera al algoritmo de análisis espectral en el modelado de los aspectos perceptuales más importantes del espectro de la señal de voz.

II.2.2 "Análisis Espectral."

Por razones pedagógicas, se clasificarán los tipos de mediciones espectrales utilizadas en el procesamiento digital de señales de voz en: mediciones de potencia, las cuales se refieren a la potencia de la señal; y mediciones de la amplitud espectral, las cuales se refieren a mediciones de la potencia de la señal sobre un determinado rango de frecuencia.

"Frecuencia Fundamental."

La frecuencia fundamental se define como la frecuencia a la cual vibran las cuerdas vocales durante un sonido "sonoro".
La frecuencia fundamental (f_0) es un parámetro, del cuál es difícil obtener una estimación confiable a partir de la señal de voz. Normalmente, $50Hz \leq f_0 \leq 500Hz$ para voz "sonora". Para voz no sonora, f_0 no está definida.

La frecuencia fundamental es comúnmente llamada frecuencia pitch, pero si se desea ser más correcto, el término pitch se define como una calidad subjetiva de la voz, la cuál está relacionada con la frecuencia fundamental [PAPA].

"Potencia."

La potencia se cálcula de la siguiente forma:

$$P(n) = \frac{1}{N_s} \sum_{m=0}^{N_s-1} [w(m)x(n - \frac{N_s}{2} + m)]^2 \qquad (2.3)$$

donde N_s es el número de muestras usadas para calcular la potencia, x(n) denota la señal, w(m) denota una función de ponderación, y n denota el tiempo discreto medido a partir del centro de la ventana. La mayoría de los algoritmos para el procesamiento digital de voz, en lugar de usar la potencia directamente, usan el logaritmo de la potencia multiplicado por 10 (en dB's), en un esfuerzo por emular la respuesta logarítmica del sistema auditivo.

La función de ponderación en la ec. (2.3) es una ventana. Existen varios tipos de ventanas que incluyen la rectangular, Hamming, Hanning, Blackman, Bartlett y Kaiser [PIC93]. La ventana generalizada de Hanning está dada por:

$$w(n) = \frac{\alpha_w - (1 - \alpha_w)cos(\frac{2\pi n}{N_s-1})}{\beta_w} \qquad (2.4)$$

para $0 \leq n \leq N_s$, y $w(n) = 0$ en cualquier otro caso. α_s se define como una constante de la ventana en el rango [0,1], y N_s es la duración de la ventana en muestras. Para implementar la ventana de Hamming $\alpha_u = 0.54$.

β_u es una constante de normalización definida de tal manera que el valor rms de la ventana sea igual a la unidad. β_u se define como:

$$\beta_w = \sqrt{\frac{1}{N_s} \sum_{n=0}^{N_s-1} w^2(n)} \qquad (2.5)$$

El propósito de la ventana es favorecer las muestras del centro de la misma. Esta función de la ventana sumada al análisis de traslape conduce a la obtención de estimaciones suavizadas. Es importante que la anchura del lóbulo principal en la respuesta en frecuencia de la ventana sea lo más pequeña posible, o de lo contrario la ventana tendrá efectos perjudiciales en el análisis espectral.

La potencia como la mayoría de los parámetros en los algoritmos de procesamiento de voz se cálcula frame por frame. La duración del frame T_f se define como el tiempo en el cuál el conjunto de parametros que caracteriza al frame es válido. El periodo del frame (o duración del frame T_f) denota el tiempo entre dos cálculos sucesivos de parametros. La tasa de frame ($1/T_f$) se refiere al número de frames calculados por segundo.

La duración de los frames depende de la velocidad de articulación del sistema productor de voz, o dicho de otra manera de la tasa de cambio de la forma del conducto vocal.

Es de importancia el intervalo sobre el cuál la potencia es calculada. El número de muestras que se utiliza para calcular la potencia, N_s, es conocido como la duración de la ventana (sólo en este caso la duración de la ventana es diferente a la duración del frame). La duración de la ventana, N_s, controla el grado de suavización aplicado en el cálculo de la potencia. La duración del frame y de la ventana controlan la razón a la cual los valores de potencia monitorean la dinámica de la señal [PIC93]. En la figura 2.1 se muestra la difrencia entre un frame y una ventana, la cuál sólo es válida para el cálculo de la potencia, en el resto de este trabajo dicha diferencia no existe.

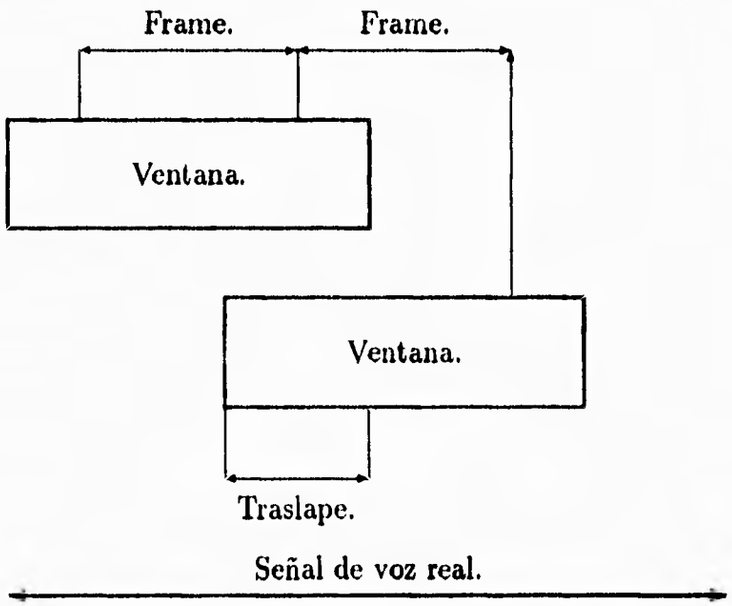


Figura 2.1: Difrencia entre un frame y una ventana.

Este tipo de análisis es llamado "análisis de traslape" debido a que con cada nuevo frame, sólo una fracción de la señal cambia. La cantidad de traslape controla, hasta cierto punto, la velocidad a la que los parámetros pueden cambiar de frame a frame [PIC93]. El porcentaje de traslape está dado por:

$$\%Traslape = \frac{T_w - T_f}{T_w} x\, 100\% \tag{2.6}$$

donde T_w es la duración de la ventana en segundos y T_f es la duración del frame.

"Algoritmos de Análisis Espectral."

En la figura 2.2 se muestran las seis clases principales de algoritmos de análisis espectral utilizados en los sistemas que procesan voz. Actualmente tanto la transformada rápida de Fourier como la predicción lineal son bastante populares en las aplicaciones que hacen uso del procesamiento digital de voz. A continuación se discutirán cada una de estas seis técnicas:

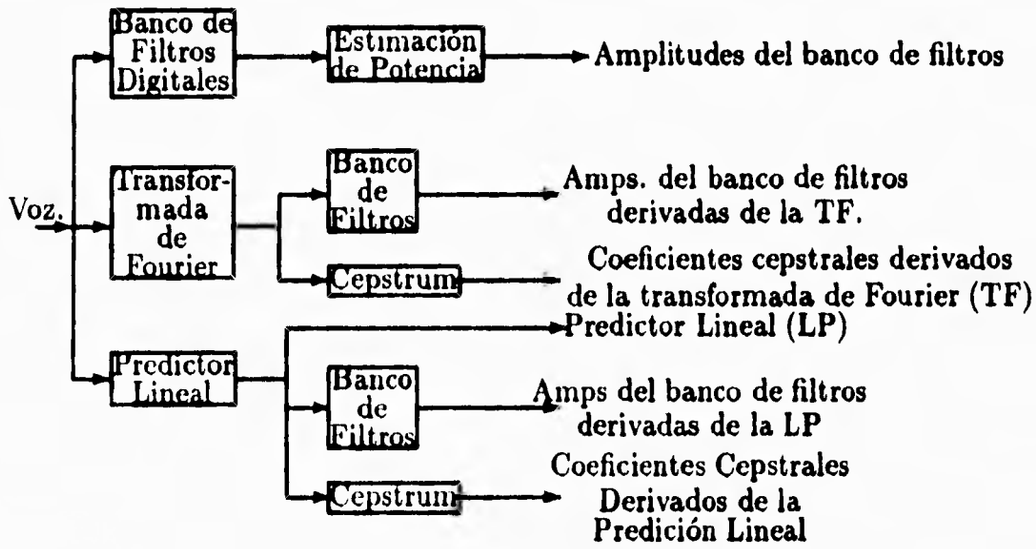


Figura 2.2: Las seis clases principales de algoritmos de análisis espectral.

1)**Banco de Filtros Digitales:** Es un modelo simplificado de las primeras etapas de la transducción del sistema auditivo. Existen dos razones principales para la utilización del banco de filtros digitales: 1) El desplazamiento de la membrana basilar para estímulos como tonos puros es proporcional al logaritmo de la frecuencia del tono, y 2) Los experimentos en percepción auditiva humana han mostrado que las componentes frecuenciales de los sonidos complejos no pueden ser individualmente identificadas dentro de cierto ancho de banda . Pero cuando una componente frecuencial cae fuera de éste ancho de banda, puede ser individualmente distinguida. A este ancho de banda se le llama ancho de banda crítico [PIC93].

Se puede definir un mapeo de la frecuencia acústica, f, a una escala perceptual

de frecuencia, de la siguiente manera:

$$Bark = 13atan\left(\frac{0.76f}{1000}\right) + 3.5atan\left(\frac{f^2}{(7500)^2}\right) \tag{2.7}$$

Las unidades de esta escala perceptual se refieren como la tasa de banda crítica o Bark.
Una escala perceptual de frecuencia más popular, intenta mapear la frecuencia percivida de un tono (pitch) en una escala lineal, matemáticamente:

$$frecuencia\ mel = 25 + 75[1 + 1.4(f/1000)^2]^{0.69} \tag{2.8}$$

Esta transformada puede ser usada para calcular anchos de banda en una escala perceptual para filtros cuya frecuencia está dada en escala de Bark ó de mel.

Las escalas de Bark y mel pueden ser interpretadas como la transformación de una escala de frecuencia en una escala perceptual y lineal. De lo anterior surge una técnica de análisis espectral conocida como el banco de filtros de banda crítica. Este es simplemente un banco de filtros pasobanda de fase lineal (FIR), arreglados linealmente a lo largo de la escala de Bark o de mel [PIC93]. Los anchos de banda de los filtros que componen el banco de filtros se escogen de tal manera que sean iguales al ancho de banda crítico para la correspondeiente frecuencia central [PIC93].

Cada filtro del banco es implementado como un filtro de fase lineal de tal manera que el retraso producido por el banco de filtros sea igual a cero, y las señales de salida de todos los filtros esten sincronizadas en el dominio del tiempo. Las ecuaciones de los filtros para una fase lineal son:

$$x_i(n) = \sum_{j=-(N_{FB_i}-1)/2}^{(N_{FB_i}-1)/2} \alpha_{FB_i}(j)x(n+j) \tag{2.9}$$

donde $\alpha_{FB_i}(j)$ denota el j-ésimo coeficiente para el i-ésimo filtro.

Es de importancia el hecho de que ciertas salidas del banco de filtros puedan estar correlacionadas con determinadas clases de sonidos.

El banco de filtros digitales se usa generalmente en sistemas que tratan de emular el proceso auditivo. De este análisis se obtiene como salida un vector cuyas componentes son los valores de potencia de cada frame de datos. Esta técnica es robusta al ruido ambiente.

2)**Realización del Banco de Filtros Digitales Mediante la Transformada de Fourier:** Una de las maneras más faciles y eficientes de calcular un banco de filtros espaciados no uniformemente es llevar a cabo la transformada de Fourier

de la señal y muestrearla a las frecuencias deseadas. La transformada discreta de Fourier se define como:

$$X(k) = \sum_{n=0}^{N_s-1} x(n)e^{-j(2\pi k/f_s)n} \tag{2.10}$$

donde k denota la frecuencia en hertz, f_s denota la frecuencia de muestreo de la señal, y N_s denota la duración de la ventana en muestras.

Usualmente el espectro es sobremuestreado y cada salida del banco de filtros es calculada como la suma ponderada de sus valores adyacentes:

$$X_{avg}(k) = \sum_{n=0}^{N_{os}} w_{FB}(n)X(k + \delta F(k,n)) \tag{2.11}$$

donde N_{os} representa el número de muestras usadas para obtener el valor promedio, $w_{FB}(n)$ representa una función de ponderación y $\delta F(k,n)$ representa alguna función que describe las frecuencias en la vecindad de f, que serán utilizadas para calcular el promedio.

La transformada rápida de Fourier (FFT) también puede ser utilizada como un método alternativo para calcular el espectro de la señal. La principal ventaja de la FFT es su rapidez: requiere de NlogN sumas y de Nlog(N/2) multiplicaciones; mientrás que la transformada discreta de Fourier (DFT) requiere N^2 operaciones.

Generalmente se limita el rango dinámico del espectro, para lo cual se fija un umbral que limite las regiones de menor amplitud del espectro, es decir, en lugar de utilizar las estimaciones ruidosas de las regiones de pequeña amplitud del espectro, estas son recortadas de acuerdo a dicho umbral, fijado a partir de un cierto porcentaje del pico máximo del espectro de la señal de voz.

Es importante que la envolvente espectral sea relativamente plana antes de aplicarle a la señal los algoritmos de limitación del rango dinámico de espectro, por que de otra manera regiones utiles, cuya energá espectral es baja, pueden ser erróneamente eliminadas.

3)**Coeficientes Cepstrales:** Los sistemas homomórficos son útiles para el procesamiento de señales de voz por que ofrecen una metodología para la separación de la señal de excitación de los efectos del conducto vocal (véase sección III.3.2).

Para calcular el cepstrum, primero se cálcula la magnitud logarítmica del espectro y después, se cálcula la transformada inversa de Fourier del espectro logarítmico:

$$c(n) = \frac{1}{N_s}\sum_{k=0}^{N_s-1} log_{10}|X_{avg}(k)|e^{(2\pi/N_s)kn} \quad 0 \leq n \leq N_s - 1 \tag{2.12}$$

Nos referimos a los coeficientes cepstrales calculados mediante la transformada de Fourier como los "coeficientes cepstrales derivados de la transformada de Fourier".

La ecuación (2.12) representa la DFT inversa de espectro logarítmico, la cuál puede ser simplificada, si se tiene en cuenta que la magnitud del espectro logarítmico es una función real y simétrica. De aquí que:

$$c(n) = \frac{2}{N_s} \sum_{k=1}^{N_s} X_{avg}(I(k)) cos \frac{2\pi}{N_s} kn \qquad (2.13)$$

donde c(n) es usualmente truncada a un orden mucho menor que N_s. I(k) es una función que mapea el entero k a la muestra apropiada de S_{avg}.

El cepstrum como se define en (2.13) puede ser fácilmente modificado a una escala de mel, mediante el muestreo de la transformada de Fourier a las frecuencias apropiadas [PIC93].

4)**Coeficientes Lineales Predictores:** Aquí sólo se discutirá de una manara breve esta técnica, para una mayor referencia véase la sección III.2. Esta es una técnica parámetrica para la representación de señales de voz, la cual trata de modelar el espctro de dichas señales como un proceso autoregresivo. A pesar de que los modelos paramétricos no son muy populares para el reconocimiento de voz, estas técnicas son ampliamente utilizadas en los sistemas de compresión de señales [PIC93]. Los modelos paramétricos fueron los que dieron impulso a la transción hacia las poderosas técnicas estadísticas para el modelado de señales [PIC93].

Los coeficientes de reflexión (véase sección III.2.1) son extremadamente útiles en sistemas, donde la memoria para el almacenamiento de datos sea limitada, ya que mediante una técnica de síntesis se puede generar la señal. Otro hecho importante sobre estos coeficientes es que su obtención para un orden N, también implica que se ha obtenido la solución para modelos de orden menor que N, lo cual es de gran utilidad en sistemas de procesamiento donde se requiera de una estimación del orden del modelo.

El modelo trata de reproducir en términos generales, tan fielmente como sea posible, el espectro de la señal de voz. Al incrementar el orden del modelo, la resolución con que la que el espectro de la señal es reproducido aumenta, mientrás que al disminurir el orden del modelo dicha resolución disminuye hasta que con un orden bajo se lográ reproducir solamente las características más sobresalientes del espectro de la señal original.

Una característica muy importante de este modelo es su inexactitud en regiones donde el espectro de la señal tiene una energía baja, por lo que sería deseable limitar en su parte baja el rango dinámico de la señal [PIC93]. Existen diversos métodos para lograr dicha reducción del rango dinámico en un modelo LPC: el método estabilizado de covarianza, el cuál reduce el rango dinámico en el dominio de la frecuencia (espectro); un método de ponderación percepetual, el cuál amplia ligeramente los

anchos de banda del modelo LP; y el método estabilizado de autocorrelación, en el cuál una pequeña cantidad de ruido es adicionada a la función de autocorrelación.

El último de estos métodos es el más simple y efectivo. La función de autocorrelación es modificada antes del cálculo de los coeficientes del filtro ó de los cxoeficientes de reflexión:

$$R_{nw} = (1 + \gamma_{nw})R_n(0)$$

$$R_{nw}(i) = R_n(i), \quad i > 0 \tag{2.14}$$

El umbral para limitar el rango dinámico de la señal generalmente se expresa en decibeles:

$$\gamma_{nw_{dB}} = 10log_{10}\gamma_{nw} \tag{2.15}$$

Este proceso de estabilización es equivalente a adicionar ruido blanco no correlacionado a la señal de voz, antes de efectuar el análisis LP. El efecto de adicionar este ruido a la función de autocorrelación es prevenir la modelización de ceros "inexistentes" o "falsos" en el espectro de la señal. Sin embargo, se introduce alguna distorsión en la forma de una suavización espectral, en las regiones del espectro de alta energía [PIC93]. Una desventaja debida a la no linealidad de este método es que su funcionamiento se vuelve problemático en ambientes ruidosos [PIC93].

5)**Amplitudes del Banco de Filtros Derivadas del Análisis LPC:** Son definidas como las amplitudes del banco de filtros resultantes del muestreo del modelo espectral LP (en lugar del espectro de la señal), a las frecuencias apropiadas.

Una técnica directa para el cálculo de las amplitudes del banco de filtros derivadas del análisis LP, es la evaluación directa del modelo LP:

$$X_{LP}(k) = \frac{G_{LP}}{\sum_{i=0}^{N_{LP}} a_{LP}(i)e^{-j2\pi(k/f_s)}} \tag{2.16}$$

donde f_s representa la frecuencia de muestreo.

Otro enfoque, es calcular el espectro de potencia usando la función de autocorrelación de la respuesta al impulso del modelo LPC, $H_{LP}(Z)$. La respuesta al impulso de $H_{LP}(z)$ puede calcularse directamente de los coeficientes LP[PIC93]:

$$R_{LP}(n) = \sum_{m=0}^{N_{LP}-|n|} a_{LP}(m)a_{LP}(m + |n|), \quad |n| \leq N_{LP}$$

$$= 0, \quad |n| > N_{LP} \tag{2.17}$$

La densidad espectral de potencia puede ser eficientemente calculada de la función de autocorrelación. Dado que ésta es una función par y real, la transformada de

Fourier también es real, y está dada por:

$$X_{LP}(k) = R_{LP}(0) + 2\sum_{k=1}^{N_{LP}} R_{LP}(n)cos\left(2\pi\frac{k}{f_s}n\right) \tag{2.18}$$

6)Coeficientes Cepstrales Derivados del Análisis LPC: Un paso lógico es el cálculo de los coeficientes cepstrales a partir de los coeficientes LPC. Si el filtro LPC es estable, el logarítmo del filtro inverso puede expresarse como una serie de potencias en z^{-1} [PIC93]:

$$\begin{aligned} C_{LP}(z) &= \sum_{i=0}^{N_c} c_{LP}(i)z^{-1} \\ &= logH(z) \\ &= log\left(\frac{G_{LP}}{\sum_{j=0}^{N_{LP}} a_{LP}(j)z^{-j}}\right) \end{aligned} \tag{2.19}$$

Podemos resolver para los coeficientes de la siguiente manera: diferenciar ambos lados de la ecuación con respecto a z^{-1}, e igualar los coeficientes de los polinomios resultantes. Esto resulta en la siguiente recursión:
inicialización:

$$c_{LP}(1) = -a_{LP}(1) \tag{2.20}$$

Para $2 \leq i \leq N_c$ {

$$c_{LP}(i) = -a_{LP}(i) - \sum_{j=1}^{i-1}\left(1 - \frac{j}{i}\right)a_{LP}(j)c_{LP}(i-j) \tag{2.21}$$

}

Los coeficientes c_{LP} son llamados los coeficientes cepstrales derivados del análisis LP. Una desventaja de estos coeficientes es que se basan en una escala lineal de frecuencia y se debe de trabajar para introducir una escala no lineal de frecuencias. Por lo que se prefiere la siguiente recursión:

Para $0 \leq n \leq N_c\{$

$$c_{bt}^{(n)}(0) = \alpha_{bt}[c_{bt}^{(n-1)}(0) - 0] + c_{LP}(N_c - n) \tag{2.22}$$

$$c_{bt}^{(n)}(1) = \alpha_{bt}[c_{bt}^{(n-1)}(1) - 0] + (1 - a_{LP}^2)c_{bt}^{(n-1)}(0) \tag{2.23}$$

Para $2 \leq k \leq N_{bt}\{$

$$c_{bt}^{(n)}(k) = \alpha_{bt}[c_{bt}^{(n-1)}(k) - c_{bt}^{(n)}(k-1)] + c_{bt}^{(n-1)}(k-1) \tag{2.24}$$

$\}$

$\}$

donde α_{bt} es un parámetro para la frecuencia de prewarping, el cual varía entre 0.4 y 0.8. Las condiciones iniciales son cero. El procesamiento puede empezar con $c_{LP}(1)$ en $n = 0$, ya que $c_{LP}(0) = 0$.

Frecuencia de prewarping: La transformada bilineal dá lugar a una compresión de frecuencias, la cuál es compensada mediante un escalamiento de frecuencias previo a la aplicación de dicha transformada. Este escalamiento se efectúa mediante:

$$u_0 = \frac{2}{T} tan\left(\frac{\omega_0 T}{2}\right)$$

donde T es el periodo de muestreo, ω_0 es la frecuencia angular a escalar y, u_0 es la frecuencia angular escalada ó frecuencia de prewarping.

II.2.3 "Transformaciones de Parametros."

En las secciones previas de éste capítulo se discutierón varios métodos para calcular "mediciones absolutas" sobre las señales de voz. En esta sección se tratará el siguiente paso en el procesamiento digital de señales: la transformación de parámetros. Los parametros que caracterizan a las señales, se generan a partir de las mediciones efectúadas sobre las mismas, mediante dos operaciones fundamentales: la diferenciación y la concatenación. La salida de esta etapa del procesamiento digital de señales, es un vector cuyas componentes consisten en estimaciones burdas de la señal.

"Diferenciación."

Se puede decir que las mediciones absolutas, anteriormente discutidas, son operadores derivativos de orden cero [PIC93]. De aquí que se desee investigar el efecto sobre los modelos de las señales de voz, al aplicar operadores derivativos de mayor orden a dichas mediciones.

Existen diversas maneras en las que una derivada de primer orden en el tiempo continuo puede ser realizada en el tiempo discreto:

$$\dot{s}(n) \equiv \frac{d}{dt}s(n) \approx s(n) - s(n-1) \tag{2.25}$$

$$\dot{s}(n) \equiv \frac{d}{dt}s(n) \approx s(n+1) - s(n) \tag{2.26}$$

$$\dot{s}(n) \equiv \frac{d}{dt}s(n) \approx \sum_{m=-N_d}^{N_d} ms(n+m) \tag{2.27}$$

Las primeras dos ecuaciones son conocidas como las diferencias hacia atrás y hacia delante respectivamente. La ecuación (2.27) representa un filtro de fase lineal, el cuál es una aproximación a un diferenciador ideal. La ecuación (2.25) es la ecuación en diferencias del filtro de preénfasis. Estas ecuaciones se refieren al análisis de regresión.

La señal de salida de este proceso de diferenciación se denota como un parámetro delta. Así mismo la salida de un proceso de diferenciasión de segundo orden se refiere como un parámetro delta-delta.

Debemos hacer notar que la diferenciación es un proceso inherentemente ruidoso, por ejemplo: los filtros de diferenciación tienden a amplificar el ruido en las mediciones de las señales. Frecuentemente es necesario calcular las derivadas de los parámetros suavizados, en lugar de las derivadas de las mediciones "burdas", de tal manera que el nivel de ruido en la medición de salida sea reducido [PIC93].

"Concatenación."

La mayoría del postprocesamiento de señales puede ser explicado en términos de la teoría de los filtros lineales. Esta noción será generalizada en la forma de un operador matricial. Definamos una matriz de medición de señales como:

$$X = \begin{bmatrix} x(0,0) & x(0,1) & \cdots & x(0,N_x-1) \\ x(1,0) & x(1,1) & \cdots & x(1,N_x-1) \\ \cdots & \cdots & \cdots & \cdots \\ x(N_f-1,0) & x(N_f-1,1) & \cdots & x(N_f-1,N_x-1) \end{bmatrix} \tag{2.28}$$

donde x(n,m) denota la m-ésima medición de la señal del n-ésimo frame ó en el instante $\left(n + \frac{1}{2}\right) T_f$. N_f denota el número total de frames en la señal, y N_x denta el número total de mediciones para cada frame.

La matriz de medición de la señal X, contiene todas las mediciones efectúadas sobre la señal para cualquier instante de tiempo comprendido por la duración de la señal.

Se debe hacer notar que la matriz de mediciones usualmente estará compuesta no solamente de una medición, sino de varias mediciones como por ejemplo: la potencia y los coeficientes cepstrales. N_x representará la dimensión del vector que esta compuesto de dichas mediciones.

A continuación se definirán dos matrices auxiliare relacionadas con el proceso de suavización de parámetros. Primero, definimos una matriz de retrasos:

$$\tau = \begin{bmatrix} \tau(0,0) & \tau(0,1) & \cdots & \tau(0,N_{\tau 0}-1) \\ \tau(1,0) & \tau(1,1) & \cdots & \tau(1,N_{\tau 1}-1) \\ \cdots & \cdots & \cdots & \cdots \\ \tau(N_p-1,0) & \tau(N_p-1,1) & \cdots & \tau(N_p-1,N_{\tau p}-1) \end{bmatrix} = [\bar{\tau}_0\ \bar{\tau}_1\ \cdots\ \tau_{\bar{N_p}-1}] \tag{2.29}$$

donde $\bar{\tau}_i$ denota el i-ésimo vector de atraso, N_p denota el número total de parámetros de la señal, y N_τ denota la dimensión de cada renglón.

A continuación se define una matriz de ponderación W, que contiene los coeficientes de los filtros, que se aplicarán a las mediciones. Estos coeficientes tienen una correspondencia uno a uno con los elementos de la matriz de atrasos:

$$W = \begin{bmatrix} w(0,0) & \cdots & w(0,N_{\tau 0}-1) \\ \cdots & \cdots & \cdots \\ w(N_p-1,0) & \cdots & w(N_p-1,N_{\tau p}-1) \end{bmatrix} = [\bar{w}_0\ \bar{w}_1\ \cdots\ w_{\bar{N_p}-1}] \tag{2.30}$$

donde $\bar{w}_i$ denota el i-ésimo vector de coeficientes cuya dimensión es igual a la del correspondiente vector τ.

También se define una matriz de indexación I, la cuál para cada renglón de W, define su correspondiente columna en X:

$$\bar{I} = [I_0\ I_1\ \cdots\ I_{N_x-1}] \tag{2.31}$$

Se define el filtrado del vector de mediciones como un operador de pseudo-convolución [PIC93]:

$$V = X * W \tag{2.32}$$

donde el operador * se define de la siguiente manera:

Para $0 \leq n \leq N_f\text{-}1\{$

Para $0 \leq i \leq N_x\text{-}1\{$

$$V[n,i] = \sum_{j=0}^{N_{rj}-1} W(i,j)X[n + \tau(i,j), I_i] \tag{2.33}$$

}

}

Notése que mediante el uso de la matriz de indexación $\bar{I}$, se pueden derivar varios parámetros a partir de la misma medición. Llamamos a la operación descrita por la ecuación (2.33): "concatenación". Esta se define como la generación de un sólo vector por frame, el cuál contiene todos los parámetros, que describen a dicho frame de la señal.

Finalmente se discutirá una forma particular de ponderar parámetros, usada con los coeficientes cepstrales. La investigación en las técnicas cepstrales de procesamiento, suguiere un medio para efectuar operaciones de filtrado lineal directamente sobre los coeficientes cepstrales, con el objeto de realzar las porciones del cepstrum que contienen información sobre el conducto vocal. Esta técnica se conoce como "liftering".

El proceso de liftering es el que sigue:

$$c_{Lift}(m) = c(m)w_{Lift}(m) \tag{2.34}$$

donde:

$$w_{Lift}(m) = 1 + \frac{N_c}{2} sin \frac{\pi m}{N_c} \tag{2.35}$$

La ecuación (2.34) describe una operación de ponderación en el dominio del tiempo, mientrás que la ecuación (2.35) describe la función de ponderación (ventana). Cabe hacer notar que ésta función estática de ponderación puede ser directamente aplicada a cualquier conjunto de coeficientes cepstrales.

II.2.4 "Modelado Estadístico."

En esta sección se asume que los parámetros que describen a la señal, son generados por un proceso aleatorio. Para descubrir la naturaleza de dicho proceso, se establecerá un modelo para la señal basado en los datos disponibles, se optimizará dicho modelo, y finalmente, se medirá la calidad de dicho modelo. La única información

disponible sobre el proceso, son sus salidas observadas, es decir, los parametros de la señal que han sido calculados. Por ésta razón, el vector de parametros de salida, de esta etapa del procesamiento, se llama "observaciones de la señal". El conjunto de dichos vectores, para la señal entera, recibe el nombre de "matriz de observación".

"Modelos Estadísticos Multivariados."

Antes de proseguir con la explicación pertinente, cabe mencionar que, frecuentemente en un conjunto de parámetros homogéneos de la señal, se mezclen distintas cantidades tales como: potencia y coeficientes cepstrales, las cuales tienen escalas numéricas completamente diferentes: el rango dinámico y la variancia de la potencia son mucho mayores que los correspondientes al conjunto de coeficientes cepstrales. De aquí que, si se compara dos vectores de parámetros usando un operador simple tal como: la distancia Euclideana, el resultado será "dominado" por los términos con mayores amplitudes y variancias, a pesar de que exista información valiosa en los parámetros de amplitud pequeña.

Mientrás que el problema de la ponderación debido a la variancia de los parámetros de la señal es fácil de solucionar, existe otro problema cuya solución es más sutil. Este problema es la eliminación de la correlación en las mediciones de la señal. Este problema es importante por dos razones: primero, la correlación implica redundancia, esto quiere decir que el número de parámetros necesarios para describir la información puede ser mucho menor que el número de mediciones; segundo, se requiere, preferentemente, de técnicas simples para comparar vectores. La presencia de parámetros correlacionados vuelve más difícil el desarollo de una métrica estadística.

1)**Transformaciones de Pre-blanqueado:** Existe un método directo para decorrelacionar los parámetros de la señal, de una manera estadísticamente óptima para un proceso multivariado Gaussiano. Sea la distribución de probabildad para un proceso multivariado gaussiano:

$$\begin{aligned} p(\bar{\nu}) &= N[\bar{\nu}, \bar{u}_\nu, C_\nu] \\ &= \frac{1}{(2\pi)^{N_\nu}|C_\nu|} e^{(-1/2)(\bar{x}-\bar{u}_\nu)C_\nu^{-1}(\bar{x}-\bar{u}_\nu)} \end{aligned} \tag{2.36}$$

Asumimios que los parametros de la señal obedecen este tipo de modelo estadístico [PIC93].

Se puede hallar una transformación lineal, la cuál decorrelacione y normalice

los parámetros, de una manera simultánea. Definamos el vector transformado como:

$$\bar{y} = \Psi(\bar{\nu} - \bar{\mu}_\nu) \tag{2.37}$$

donde $\bar{\nu}$ denota el vector de parametros de entrada, y $\bar{\mu}_\nu$ denota el valor medio de dicho vector. Definamos Ψ como una transformación de prewhitening, basándonos en el hecho de que, se desea que el resultado de esta transformación sea un vector aleatorio, gaussiano y decorrelacionado. Para lograr este resultado, se debe mostrar que Ψ está dada por:

$$\Psi = \Lambda^{-1/2}\Phi \tag{2.38}$$

donde Λ denota una matriz diagonal de valores propios, Φ denota una matriz de vectores propios de la matriz de covariancia de $\bar{\nu}$.

Los valores propios y los vectores propios son la base de todo el cálculo, ya que describen una transformación lineal del espacio del vector de entrada en un nuevo espacio, en el cuál se pueden utilizar distancias euclideanas, como se calculan normalmente, para la comparación de vectores.

Se puede mostrar que los valores propios y los vectores propios, satisfacen la siguiente relación:

$$C_\nu = \Phi\Lambda\Phi \tag{2.39}$$

donde C_ν es la matriz de covariancia para ν. Cada elemento de C_ν, $C_\nu(i,j)$, puede ser calculado de la siguiente manera:

$$C_\nu(i,j) = \frac{1}{N_f}\sum_{m=0}^{N_f-1}(\nu_m(i) - \mu_\nu(i))(\nu_m(j) - \mu_\nu(j)) \tag{2.40}$$

Existe una simplificación muy importante de la ecuación (2.39), la cuál necesita explicación. Si los parámetros no están correlacionados, entonces la matriz de covariancia en (2.39), se reduce a una matriz diagonal. En éste caso, la transformación dada por (2.39), se simplifica a una matriz diagonal [PIC93]:

$$\Psi = \begin{bmatrix} \frac{1}{\sigma_{\nu(0)}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_{\nu(1)}} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \frac{1}{\sigma_{\nu(N_\nu-1)}} \end{bmatrix} \tag{2.41}$$

donde $\sigma_\nu(i)$ es la desviación estandar de la i-ésima componente del vector de parámetros ν. La transformación descrita por (2.41), también se conoce como: la normalización de los parámetros por sus desviaciones estandar; por ello cada parámetro tiene la misma importancia en el cálculo.

Por último cabe mencionar, que en el caso de que una distribución Gaussiana simple, no sea suficiente para la modelización de los parámetros, entonces podemos recurrir a una suma ponderada de distribuciones gaussianas [PIC93].

2)**Cuantización Vectorial:** es posible realizar un ajuste no paramétrico de los datos, al forzar al sistema para que "aprenda" una distribución discreta de probabilidad, cuya forma es arbitraria [PIC93]. A este tipo de modelización se le conoce como cuantización vectorial.

Uno de los argumentos más convincentes para el empleo de la cuantización vectorial, se basa en un modelo para la producción de voz [PIC93], el cuál propone a las dimensiones del conducto vocal, como las principales mediciones. Por tanto, éste modelo asume la existencia de un conjunto pequeño de sonidos elementales dentro de un idioma. De aquí que, deberíamos de ser capaces de modelar un vector continuamente valuado, con un conjunto finito de vectores que represente dichos sonidos elementales.

Otro argumento igualmente convincente, basado en la teoría de la tasa de distorsión, muestra que deberíamos de ser capaces de modelar el vector de parámetros, con un número finito de valores discretos.

Definamos que un cuantizador vectorial se compone de dos términos: una matriz Q, de tamaño N_{vq} x N_y; y una distribución discreta de probabilidad denotada por: $p(\bar{q}_i)$. Q se conoce como "vector quantization codebook". Sus renglones son vectores de parametros. $p(\bar{q}_i)$ se llama "a priorix symbol probability distribution". Sus elementos ($0 \leq i \leq N_{vq}$ - 1) son las probabilidades de observar determinado vector de parametros (o renglón) en Q. Denotamos al proceso de cuantización vectorial del vector de entrada $\bar{y}$, con $Q[\bar{y}]$.

También necesitamos definir una medida general para la distancia entre dos vectores:

$$D(\bar{y}_1, \bar{y}_2) = f(\bar{y}_1, \bar{y}_2) \tag{2.42}$$

El proceso de cuantización vectorial consiste de dos tareas principales. Primero, la estimación de Q[ ], de tal manera que la distorsión introducida al reemplazar el vector de entrada por un codebook vector, sea mínima. Segundo, ¿ cómo se puede estimar la probabilidad de observar $\bar{y}$, del codebook ?.

El último problema es relativamente simple: escogeremos el índice i, de acuerdo con la regla:

$$i = min[D(\bar{y}, \bar{q}_j)], \quad 0 \leq j < N_{vq} \tag{2.43}$$

y hacemos:

$$P(\bar{y}|Q) = p(\bar{q}_i) \tag{2.44}$$

$P(\bar{y}|Q)$ puede ser estimada, calculando la probabilidad de ocurrencia de cada vector, en el codebook.

El primer problema es ligeramente más complicado, ya que se requiere de una secuencia de entrenamiento. No existe una forma cerrada para el cálculo del conjunto óptimo de vectores, que conforman el codebook. Afortunadamente, existen varias técnicas iterativas para el cálculo de los vectores que formarán el codebook. La técnica más popular es el algoritmo K-MEANS, cuyo nombre, alude al hecho de que, el algoritmo trata de agrupar los datos en k grupos y, sustituir los datos de cada grupo por su respectiva media ó centroide. El algoritmo es el siguiente [PIC93]:

2)**Cuantización Vectorial:** es posible realizar un ajuste no paramétrico de los datos, al forzar al sistema para que "aprenda" una distribución discreta de probabilidad, cuya forma es arbitraria [PIC93]. A este tipo de modelización se le conoce como cuantización vectorial.

Uno de los argumentos más convincentes para el empleo de la cuantización vectorial, se basa en un modelo para la producción de voz [PIC93], el cuál propone a las dimensiones del conducto vocal, como las principales mediciones. Por tanto, éste modelo asume la existencia de un conjunto pequeño de sonidos elementales dentro de un idioma. De aquí que, deberíamos de ser capaces de modelar un vector continuamente valuado, con un conjunto finito de vectores que represente dichos sonidos elementales.

Otro argumento igualmente convincente, basado en la teoría de la tasa de distorsión, muestra que deberíamos de ser capaces de modelar el vector de parámetros, con un número finito de valores discretos.

Definamos que un cuantizador vectorial se compone de dos términos: una matriz Q, de tamaño $N_{\nu q}$ x N_y; y una distribución discreta de probabilidad denotada por: $p(\bar{q}_i)$. Q se conoce como "vector quantization codebook". Sus renglones son vectores de parametros. $p(\bar{q}_i)$ se llama "a priorix symbol probability distribution". Sus elementos ($0 \leq i \leq N_{\nu q} - 1$) son las probabilidades de observar determinado vector de parametros (o renglón) en Q. Denotamos al proceso de cuantización vectorial del vector de entrada $\bar{y}$, con $Q[\bar{y}]$.

También necesitamos definir una medida general para la distancia entre dos vectores:

$$D(\bar{y}_1, \bar{y}_2) = f(\bar{y}_1, \bar{y}_2) \tag{2.42}$$

El proceso de cuantización vectorial consiste de dos tareas principales. Primero, la estimación de Q[ ], de tal manera que la distorsión introducida al reemplazar el vector de entrada por un codebook vector, sea mínima. Segundo, ¿ cómo se puede estimar la probabilidad de observar $\bar{y}$, del codebook ?.

El último problema es relativamente simple: escogeremos el índice i, de acuerdo con la regla:

$$i = min[D(\bar{y}, \bar{q}_j)], \quad 0 \leq j < N_{\nu q} \tag{2.43}$$

y hacemos:

$$P(\bar{y}|Q) = p(\bar{q}_i) \tag{2.44}$$

$P(\bar{y}|Q)$ puede ser estimada, calculando la probabilidad de ocurrencia de cada vector, en el codebook.

El primer problema es ligeramente más complicado, ya que se requiere de una secuencia de entrenamiento. No existe una forma cerrada para el cálculo del conjunto óptimo de vectores, que conforman el codebook. Afortunadamente, existen varias técnicas iterativas para el cálculo de los vectores que formarán el codebook. La técnica más popular es el algoritmo K-MEANS, cuyo nombre, alude al hecho de que, el algoritmo trata de agrupar los datos en k grupos y, sustituir los datos de cada grupo por su respectiva media ó centroide. El algoritmo es el siguiente [PIC93]:

Inicialización:
Asignar un conjunto de N_{vq} vectores iniciales a $\bar{q}_i^{(0)}$

$$\varepsilon^{(0)} = 1$$

Para 1 ≤ m ≤ M_{max}{

$$\eta[j] = 0, \qquad 0 \leq j < N_{vq}$$

Para 0 ≤ n < N_f{

$$i = min[D(\bar{y}_n, \bar{q}_j^{(m-1)}], \qquad 0 \leq j < N_{vq}$$
$$\bar{q}^{(m)}[j] = \bar{q}^{(m)}[j] + \bar{y}_n[j], \qquad 0 \leq j < N_v$$
$$\eta[i] = \eta[i] + 1$$

}

$$\bar{q}_i^{(m)} = \bar{q}_i^{(m)}/\eta[i], \qquad 0 \leq i < N_{vq}$$
$$\varepsilon^{(m)} = \varepsilon^{(m)} + Q[\bar{y}_n], \qquad 0 \leq n < N_f$$

If $(\varepsilon^{(m)}/\varepsilon^{(m-1)}) < \Delta\varepsilon_{vq}$, romper ciclo
}
Terminación:

$$p(\bar{q}_i) = \frac{\eta[i]}{N_{db}}, \qquad 0 \leq i \leq N_{vq}$$

La inicialización es un proceso importante, ya que las suposiciones iniciales para los centroides de grupo deberían atravesar el espacio de datos. Un procedimiento simple para seleccionar dicho centroides, es buscar en lo datos, los vectores iniciales N_{vq} cuya distancia del uno al otro sea ε. Inicialmente el valor de ε es grande, pero se va reduciendo a medida que los vectores N_{vq}, que satisfacen el requerimento mínimo de distancia, son encontrados.

Es necesario mencionar que no existe garantía de que éste algoritmo converja a una soluición óptima. La calidad del codebook puede calcularse, mediante el promedio de la distorsión sobre la base de datos para el "entrenamiento":

$$\varepsilon_{avg} = \frac{1}{N_f} \sum_{n=0}^{N_f-1} D\left(D(\bar{y}_n, Q[\bar{y}_n]\right) \tag{2.45}$$

Este valor es calculado para cada iteración del algoritmo K-MEANS. El promedio de la distorsión usualmente decrece logarítmicamente con relación al tamaño del codebook. Mientrás más grande sea el codebook menor será el promedio de distorsión.

El proceso de cuantización se convierte en un problema de búsqueda, una vez que sea calculado el codebook. La regla del vecino más cercano, ecuaciones (2.43) y (2.44), es una búsqueda lineal, la cuál requiere de N_{vq} comparaciones de distancia por cada vector de entrada.

"Medición de Distancias."

Cualquier método para la medición de distancias, debería de tener las siguientes propiedades:
1)No negativo:

$$D(\bar{x}_1, \bar{x}_2) > 0, \quad x_1 \neq x_2$$
$$D(\bar{x}_1, \bar{x}_2) = 0, \quad x_1 = x_2$$

2) Simetría:

$$D(\bar{x}_1, \bar{x}_2) = D(\bar{x}_2, \bar{x}_1)$$

3) Desigualdad del triángulo:

$$D(\bar{x}_1, \bar{x}_3) \leq D(\bar{x}_1, \bar{x}_2) + D(\bar{x}_2, \bar{x}_3)$$

La más famosa medición de distancia, que posee estas propiedades es la euclideana.

Una de las primeras mediciones introducidas en el reconocimiento de voz, fué la medición de probabilidad. Está medición calcula la energía de la diferencia espectral de dos conjuntos de parametros LP. Esta medición evalúa la probabilidad de que los datos de prueba sean generados de un modelo estadístico, el cuál está basado en el conjunto de parametros LPC, de referencia. Esta medición está dada por:

$$D(\bar{y}_1, \bar{y}_2) = \frac{a_{LP_1} R_2 a_{LP_1}}{a_{LP_2} R_2 a_{LP_2}} \tag{2.46}$$

R_2 representa la matriz de autocorrelación usada para generar los parametros LP para $\bar{x}_2$.

La medición de probabilidad es asimétrica, lo cuál no es un problema significativo, sin embargo, actualmente no es muy frecuente su utilización.

En una derivación de la medición de Mahalanobis, se muestra que la medición de probabilidad de un vector que obedece una distribución Gaussiana multivariada, se puede expresar como una distancia euclideana ponderada [PIC93]:

$$D(\bar{y}, \bar{\mu}) = (\bar{y} - \bar{\mu}) C^{-1} (\bar{y} - \bar{\mu}) \tag{2.47}$$

donde μ y C son la media y la covariancia de la distribución. Parte de la popularidad de las distancias euclideanas se debe a la utilización de conjuntos de parametros decorrelacionados. En varias aplicaciones, tales como la cuantización vectorial, es posible utilizar una forma factorizada de la distancia euclideana:

$$\begin{aligned} D(\bar{y}_1, \bar{y}_2) &= \|\bar{y}_1 - \bar{y}_2\|^2 \\ &= (\bar{y}_1 - \bar{y}_2)(\bar{y}_1 - \bar{y}_2) \\ &= \|\bar{y}_1\|^2 + \|\bar{y}_2\|^2 - 2(\bar{y}_1 \cdot \bar{y}_2) \end{aligned} \tag{2.48}$$

"Medición de Distancias."

Cualquier método para la medición de distancias, debería de tener las siguientes propiedades:
1)No negativo:

$$D(\bar{x}_1, \bar{x}_2) > 0, \quad x_1 \neq x_2$$
$$D(\bar{x}_1, \bar{x}_2) = 0, \quad x_1 = x_2$$

2) Simetría:

$$D(\bar{x}_1, \bar{x}_2) = D(\bar{x}_2, \bar{x}_1)$$

3) Desigualdad del triángulo:

$$D(\bar{x}_1, \bar{x}_3) \leq D(\bar{x}_1, \bar{x}_2) + D(\bar{x}_2, \bar{x}_3)$$

La más famosa medición de distancia, que posee estas propiedades es la euclideana.

Una de las primeras mediciones introducidas en el reconocimiento de voz, fué la medición de probabilidad. Está medición calcula la energía de la diferencia espectral de dos conjuntos de parametros LP. Esta medición evalúa la probabilidad de que los datos de prueba sean generados de un modelo estadístico, el cuál está basado en el conjunto de parametros LPC, de referencia. Esta medición está dada por:

$$D(\bar{y}_1, \bar{y}_2) = \frac{a_{LP_1} R_2 a_{LP_1}}{a_{LP_2} R_2 a_{LP_2}} \tag{2.46}$$

R_2 representa la matriz de autocorrelación usada para generar los parametros LP para $\bar{x}_2$.

La medición de probabilidad es asimétrica, lo cuál no es un problema significativo, sin embargo, actualmente no es muy frecuente su utilización.

En una derivación de la medición de Mahalanobis, se muestra que la medición de probabilidad de un vector que obedece una distribución Gaussiana multivariada, se puede expresar como una distancia euclideana ponderada [PIC93]:

$$D(\bar{y}, \bar{\mu}) = (\bar{y} - \bar{\mu}) C^{-1} (\bar{y} - \bar{\mu}) \tag{2.47}$$

donde μ y C son la media y la covariancia de la distribución. Parte de la popularidad de las distancias euclideanas se debe a la utilización de conjuntos de parametros decorrelacionados. En varias aplicaciones, tales como la cuantización vectorial, es posible utilizar una forma factorizada de la distancia euclideana:

$$\begin{aligned} D(\bar{y}_1, \bar{y}_2) &= \|\bar{y}_1 - \bar{y}_2\|^2 \\ &= (\bar{y}_1 - \bar{y}_2)(\bar{y}_1 - \bar{y}_2) \\ &= \|\bar{y}_1\|^2 + \|\bar{y}_2\|^2 - 2(\bar{y}_1 \cdot \bar{y}_2) \end{aligned} \tag{2.48}$$

Capítulo Tres

"Descripción de los Esquemas Clásicos para la Síntesis de voz"

III. "Síntesis de Señales de Voz Mediante la Codificación Lineal Predictiva"

III.1 "Introducción."

En este capítulo se hace una descripción de los siguientes esquemas para la síntesis de voz:

1)Método de síntesis basado en una codificación lineal predictiva (LPC), con el algoritmo de autocorrelación con recorte central (AUTOC), para la estimación del pitch.
2)Método de síntesis basado en LPC, con el algoritmo "Simplified Inverse Filter Tracking" (SIFT), para la estimación del pitch.
3)Método de síntesis basado en LPC, con un método cepstral para la estimación del pitch.
4)Método de síntesis "Excitación Multipulso".
5)Método de síntesis "Regular Pulse Excitation" (RPE).

La diferencia fundamental entre todos estos métodos radica en el cálculo de la secuencia de excitación para el filtro de síntesis. Para los tres primeros métodos (AUTOC, SIFT y el cepstral), dicha secuencia de excitación consiste de ruido blanco ó de un tren de pulsos equidistantes. Esta secuencia de excitación depende del algoritmo de estimación del pitch, si éste declara el frame de voz como no sonoro, la secuencia de excitación consistirá de ruido blanco. En caso contrario la secuencia de excitación será un tren de pulsos, cuyo periodo es igual al pitch estimado por el algoritmo.

En el caso de los métodos: excitación multipulso y RPE, la secuencia de excitación para el filtro de síntesis consiste de una serie de pulsos, cuyas amplitudes y posiciones son determinadas de acuerdo con el respectivo algoritmo. Evitándose de esta manera la decisión Sonoro-No Sonoro.

Es necesario aclarar que todos los métodos de síntesis de voz, aquí descritos utilizan el mismo modelo paramétrico (LPC), para la representación de las señales de voz. Por tanto, no se hace una división entre los métodos de excitación multipulso y, aquéllos en los que se detecta el pitch.

III.2"Un Modelo para la Representación Paramétrica de Señales de Voz."

El primer paso en el análisis de señales es establecer un modelo para la representación de las mismas.

La voz es producida al excitar el conducto vocal. Durante la producción de sonidos sonoros, el conducto vocal es excitado por una serie de pulsos quasi-periodicos generados por las cuerdas vocales. En el caso de sonidos no sonoros, éstos se producen por las turbulencias de aire generadas por las constricciones del conducto vocal. Un modelo simple del conducto vocal, se logra mediante su representación como un filtro lineal, discreto y variante en el tiempo. Si suponemos que las variaciones de forma del conducto vocal a través del tiempo, pueden aproximarse con suficiente excatitud, por una sucesión de formas estacionarias, es posible definir una función de transferencia en el domino complejo z, para el conducto vocal. Es bastante conocido el hecho, de que para sonidos sonoros no nasales, la función de transferencia del conducto vocal no tiene ceros [ATAL71]. Por lo tanto, para está clase de sonidos, el conducto vocal, puede ser adecuadamente representado por un filtro todo polar. En el caso de los sonidos no sonoros y de los sonidos nasales, se presentan las antiresonancias (ceros) y las resonancias (polos) del conducto vocal [ATAL71]. Dado que los ceros, de la función de transferencia del conducto vocal, para este último caso, caen dentro del círculo unitario del plano z [ATAL71], cada factor en el numerador de la función de transferencia, puede ser aproximado por varios polos en el denominador de dicha función de transferencia [ATAL71]. Dado que la localización de un polo es considerablemente más importante, desde el punto de vista de la perceptuabilidad, que la localización de un cero [ATAL71]; una representación explícita, de las antiresonancias mediante ceros, en la función de transferencia, no es necesaria.

La transformada z del flujo glotíco volumétrico, durante un sólo periodo pitch, puede modelarse mediante un modelo polar [ATAL71]. Tomando en cuenta, está aproximación, la transformada z del flujo glotíco puede ser representada por:

$$U_g(z) = \frac{K_1}{(1 - z_a z^{-1})(1 - z_b z^{-1})} \tag{3.1}$$

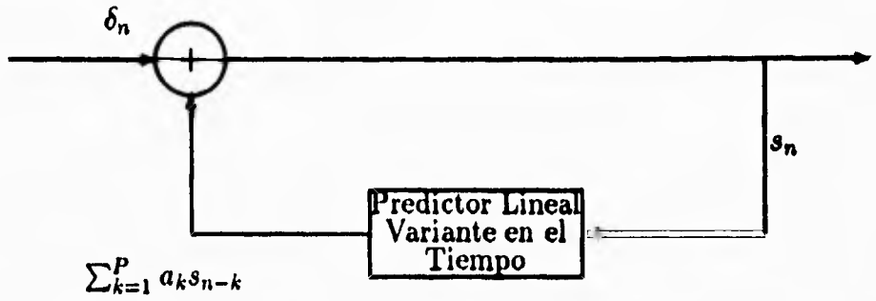


Figura 3.1: Modelo para la producción de voz basado en la codificación lineal predictiva

donde:
K_1 es una constante relacionada a la amplitud del flujo glotíco.
z_a, z_b son los polos.

Si la emisión de ondas sonoras a través de la boca, se aproxima a la emisión de ondas sonoras de una sóla fuente esférica, entonces la razón entre la presión del sonido en el micrófono, a la velocidad del flujo glotíco en los labios, es representada en el dominio z, como: $K_2(1 - z_{-1})$, donde K_2 es una constante relacionada a la amplitud del flujo glotíco volumétrico en los labios, y a la distancia de los labios al micrófono [ATAL71]. Entonces la contribución del flujo volumétrico glotíco, junto con la radiación, puede ser representada en la función de transferencia, por el factor:

$$\frac{K_1 K_2(1 - z^{-1})}{(1 - z_a z^{-1})(1 - z_b z^{-1})}$$

Por tanto, la ecuación de transferencia puede ser expresada como:

$$\frac{K_1 K_2}{[1 + (1 - z_a)z^{-1}](1 - z_b z^{-1})} \tag{3.2}$$

El error que se introduce debido a esta aproximación está dado por:

$$\frac{K_1 K_2 z^{-2}(1 - z_a)}{(1 - z_a z^{-1})[1 + (1 - z_a)z^{-1}](1 - z_b z^{-1})}$$

Una de las características más importantes de este modelo consiste en que la combinación de las contribuciones del flujo glotíco, el conducto vocal, y la radiación son representados por un sólo filtro recursivo [ATAL71]. Por lo tanto, la dificultad de separar la contribución de la función de excitación (fuente) de la del conducto vocal, ha sido salvada [ATAL71].

Esta representación de las señales de voz se ilustra en la figura 3.1. La excitación para sonidos sonoros es producida mediante un generador de pulsos, de periodo y amplitud variable. La excitación para sonidos no sonoros es una fuente de ruido blanco. El predictor lineal P, es un filtro transversal con p atrasos de una muestra cada intervalo; forma una suma ponderada de las pasadas p muestras a la entrada del predictor. La salida del filtro lineal al n-ésimo instante de muestreo, está dada por:

$$s_n = \sum_{k=1}^{P} a_k s_{n-k} + \delta_n \tag{3.3}$$

donde: a_k son los coeficientes predictores.

y δ_n representa la muestra n-ésima de la excitación. La función de transfrencia del filtro lineal de la figura 1. está dado por:

$$T(z) = \frac{1}{1 - \sum_{k=1}^{P} a_k z^{-k}} \tag{3.4}$$

El número de coeficientes p requeridos para representar cualquier segmento de voz adecuadamente está determinado por el número de resonancias y antiresonancias del conducto vocal, en el rango de frecuencias de interés, la naturaleza de la función que representa el flujo glotíco volumétrico y la radiación.

En [ATAL71] se muestra, que para poder representar adecuadamente los polos de la función de transferencia del conduct vocal, la memoria del predictor lineal debe ser igual a dos veces el tiempo requerido para que las ondas sonoras viajen de la glotis a los labios.

Los coeficientes de predicción a_k, junto con el periodo pitch, la energía del frame de voz, y un parámetro binario que indica, si el frame de voz es sonoro o no sonoro, proveen una representación completa de dicho frame de voz. Dado que la forma del conducto vocal es variante en el tiempo, es necesario reajustar dichos parámetros periodicamente.

III.2.1"Determinación de los Parámetros de Predicción."

Antes de proceder con la explicación de la determinación de los parámetros predictores, es necesario hacer notar, que el método de LPC se aplica a señales estacionarias; dado que para señales de voz este no es el caso, debemos segmentar las señales de voz, de tal manera que dichos segmentos de voz, llamados "frames", sean quasi-estacionarios. Esto se efectua mediante la multiplicación de la señal de voz, s(n), por una ventana, w(n), la cuál tiene el valor de cero fuera del intervalo de interés. Generalmente se evita aplicar una ventana de tipo rectangular, dado que en el domino de la frecuencia la señal de voz se "contamina" [PAPA]. En este trabajo, se aplicó la ventana de Hamming:

$$w(n) = 0.54 - 0.46\,cos\frac{2\pi n}{N} \quad 0 \leq n \leq N-1 \tag{3.5}$$

donde w(n) = 0, en cualquier otro caso.

Otro aspecto de importancia, en la determinación de los parámetros predictores, es la necesidad de tomar en cuenta, que las componentes de alta frecuencia de la voz en magnitud son menos significativas que las componentes de baja frecuencia; lo cuál trae como consecuencia que el método dé resultados satisfactorios en baja frecuencia, mientras que en alta frecuencia no tendrá un desempeño muy adecuado. Para prevenir este efecto se filtra la señal de voz a través de un filtro de énfasis, el cuál enfátiza las altas frecuencias antes de procesar la señal. El filtro de síntesis está dado por las ecuaciones:

$$s'(n) = s(n) - 0.9375s(n-1) \tag{3.6}$$

donde s'(n) es la señal preenfatizada y s(n) es la señal de entrada.
A la salida del sintetizador la señal debe ser deenfatizada mediante:

$$s(n) = s'(n) + 0.9375s(n-1) \tag{3.7}$$

Volviendo a la figura 3.1., observamos que, excepto para la muestra inicial de cada periodo pitch, las muestras de voz sonora son linealmente predecibles, en términos de las pasadas p muestras de voz. Esta propiedad de las señales de voz, es utilizada para determinar los coeficientes predictores. El error de predicción E_n, se define como la diferencia entre la muestra S_n y su valor estimado $\hat{s}$, el cuál está dado por:

$$\hat{s}_n = \sum_{k=1}^{P} a_k s_{n-k} \tag{3.8}$$

Entonces E_n está dado por:

$$E_n = s_n - \hat{s}_n = s_n - \sum_{k=1}^{P} a_k s_{n-k} \tag{3.9}$$

Se define el error cuadrático medio $< E_n^2 >_{av}$ como el promedio de E_n^2 sobre las n muestras del marco de voz a análizar, excepto aquéllas que se encuentran al principio de cada periodo pitch, esto es:

$$< E_n^2 >_{av} = < (s_n - \sum_{k=1}^{P} a_k s_{n-k})^2 >_{av} \tag{3.10}$$

Los coeficientes predictores a_k de la ec.3 son aquéllos que minimizan el error cuadrático medio $< E_n^2 >_{av}$. El mismo procedimiento es utilizado para los sonidos no sonoros. Los coeficientes a_k, que minimizan $< E_n^2 >_{av}$ son obtenidos al igualar con cero, la derivada parcial de $< E_n^2 >_{av}$ con respecto a cada coeficiente a_k. Se puede mostrar [ATAL71], que los coeficientes a_k, se obtienen al resolver el siguiente conjunto de ecuaciones:

$$\sum_{k=1}^{P} \varphi_{jk} a_k = \varphi_{j0}, \quad j = 1, 2, \ldots, p, \tag{3.11}$$

donde:

$$\varphi_{jk} = < s_{n-j} s_{n-k} >_{av} \tag{3.12}$$

Es la función de autocorrelación estimada.

Dado que la matriz de coeficientes es una matriz Toeplitz, existen métodos más eficientes de resolver dicho conjunto de ecuaciones qué la eliminación Gauss-Jordan. Dos de los métodos comúnmente utilizados en el procesamiento de voz son: la recursión de Levinson-Durbin (ver apéndice C) y, la recursión de Leroux-Gueguen [PAPA].

"Método de Análisis Leroux-Gueguen

para la Obtención de los Parámetros Predictores"

Los parámetros K_i, i = 1, ..., p son llamados coeficientes de reflexión, y juegan un papel central en el método de LPC (Linear Predictive Coding). Tienen las siguientes propiedades:

1.- Son equivalentes a los coeficientes predictores a_i. Las siguientes ecuaciones muestran dicha relación:

De K's a A's:

$$a_i^{(i)} = K_i \tag{3.13}$$

$$a_j^{(i)} = a_j^{(i-1)} + K_i a_{i-j}^{(i-1)} \tag{3.14}$$

donde: i = 1, ...,p
j = 1, ...,i-1

De A's a K's:

$$K_i = a_i^{(i)} \tag{3.15}$$

$$a_j^{(i-1)} = \frac{a_j^{(i)} - a_i^{(i)} a_{i-j}^{(i)}}{1 - K_i^2} \tag{3.16}$$

donde: i = p, ..., 1
j = 1, ..., i-1

2.- Para un filtro estable se cumple que:

$$-1 < K_i < 1 \quad i = 1, ..., p \tag{3.17}$$

3.- Para implementar el filtro que modeliza al conducto vocal, no es necesario convertir los coeficientes de reflexión (K's) a los coeficientes predictores (A's). Para esto, se implementa el filtro lattice (ver figura 2), mediante las ecuaciones:

$$f^{(i-1)}(n) = f^{(i)}(n) - K_i b^{(i-1)}(n-1) \tag{3.18}$$

$$b^{(i)}(n) = K_i f^{(i-1)}(n) + b^{(i-1)}(n-1) \tag{3.19}$$

con:

$$f^{(p)}(n) = Gu(n) \tag{3.20}$$

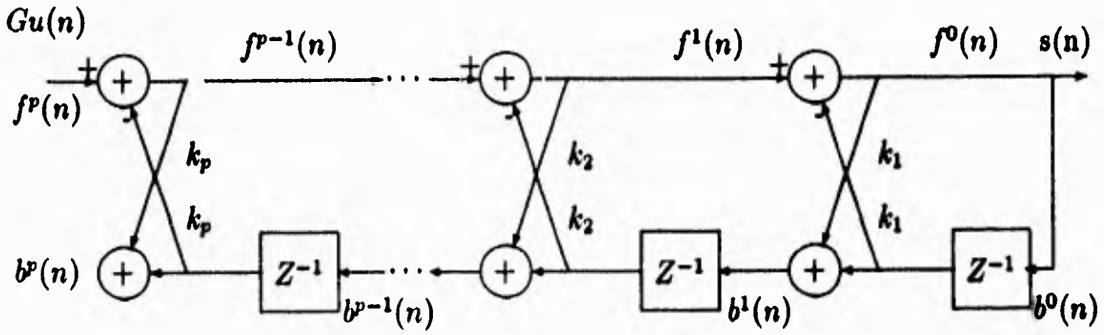


Figura 3.2: Filtro Lattice para sintetizar voz usando los coeficientes de reflexión K_i

donde u(n) es la función de excitación. El superíndice indica la etapa en el filtro lattice, mientrás que el argumento es el índice temporal. La salida del filtro lattice es:

$$s(n) = f^{(0)}(n) \tag{3.21}$$

"Algoritmo de Leroux-Gueguen"

Este algoritmo calcula diectamente los coeficientes de reflexión (K's), a partir de la secuencia de p autocorrelaciones de la señal de voz, sin necesidad de calcular los coeficientes predictores (A's) como resultado intermedio[PAPA]. Leroux y Gueguen, lograrón esto al introducir la variable:

$$e^j(i) = r(i) + a_1^{(j)} r(i-1) + \ldots + a_j^{(j)} r(i-j) \tag{3.22}$$

donde r(i) son los coeficientes de autocorrelación. Entonces, los coeficientes de reflexión pueden ser calculados de las fórmulas:

$$K_j = \frac{-e^{j-1}(j)}{e^{j-1}(0)} \quad j = 1, \ldots, p \tag{3.23}$$

$$e^j(i) = e^{j-1}(i) + K_{j-1} e^{j-1}(j-i) \quad i = -p+j, \ldots, p \tag{3.24}$$

con la condición inicial:

$$e^0(i) = r(i) \quad i = -p, \ldots, p \tag{3.25}$$

donde r(-i) = r(i). La energía residual es:

$$E(i) = e^j(0) \tag{3.26}$$

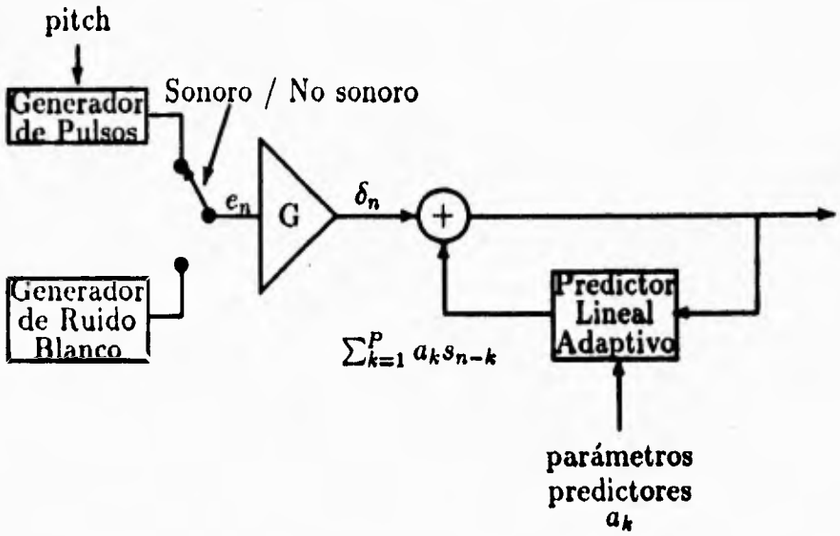


Figura 3.3: Modelo para sintetizar voz basado en la codificación lineal predictiva

Hasta este punto, la figura 3.3 ha quedado totalmente explicada. Los únicos parámetros, de los cuales no se ha tradado son: el periodo pitch y el parámetro binario que indica, si el frame en cuestión es sonoro o no sonoro. Por lo cuál, dichos parámetros no aparecen en la figura.

III.3"Extracción del Pitch

III.3.1 "Introducción"

La determinación de la existencia de cierta periodicidad en un frame de voz, es de crucial importancia en los vocoders (codificadores de voz), así como en la mayoría de los algoritmos codificadores de señales de voz. Esta periodicidad o falta de la misma, determinan si el marco de voz es sonoro o no sonoro, y en caso de que este sea sonoro, determina el valor de la frecuancia fundamental. La determinación de esta última, juega un papel crítico en la calidad de la voz sintetizada mediante vocoders.

La frecuancia fundamental de un marco de voz, se designa como F_0, y es comúnmente llamada frecuencia "pitch". Sin embargo, si se desea ser más correcto, el término pitch se define como una calidad subjetiva de la voz, la cuál está relacionada a F_0. El inverso de la frecuencia pitch:

$$\tau = \frac{1}{F_0} \tag{3.27}$$

es el periodo pitch.

La estimación del pitch permanece como la parte más vulnerable de los vocoders; dado que todavía no se ha propuesto un algoritmo, el cuál se desempeñe robustamente en la mayoría de los casos y en concordancia con la percepción humana [PAPA].

III.3.2 "El Problema de la Extracción del Pitch"

Para darse una idea de este problema, véase la figura 3.4, en la cuál se esquematiza un modelo simplificado [NOLL67], para la producción de sonidos sonoros, el cuál consiste de una fuente sonora (excitación), s(t), y del modelo del conducto vocal, h(t).

Entonces, de acuerdo con la teoría de sistemas la señal de voz f(t), está dada en el dominio del tiempo, por la convolución de la señal de excitación, s(t), con la respuesta al impulso del conducto vocal, h(t), matemáticamente:

$$f(t) = s(t) * h(t) \tag{3.28}$$

Figura 3.4: Modelo simplificado para la producción de sonidos "sonoros".

Alternativamente en el dominio de la frecuencia, si $S(\omega)$ es la transformada de Fourier de s(t) y, $H(\omega)$ es la transformada de Fourier de la respuesta al impulso del conducto vocal, h(t), entonces la transformada de Fourier o espectro de la señal de voz, se puede expresar como:

$$F(\omega) = S(\omega)\,H(\omega) \tag{3.29}$$

Entonces, extraer el pitch de la señal de voz, f(t), es equivalente a determinar si la señal de excitación, s(t), presenta periodicidad o no, y en caso afirmativo estimar el valor de la frecuencia fundamental (pitch). Sin embargo, el pitch se extrae de la señal de voz, f(t), y no de la señal de excitación, s(t), lo cuál trae la siguiente complicación: la señal de excitación está convolucionada con la respuesta al impulso del conducto vocal, esto significa que la señal de voz contiene información tanto del pitch como de las resonancias (formants) del conducto vocal. Entonces para lograr una buena estimación del pitch, es necesario reducir al máximo los efectos de las resonancias (formants) del conducto vocal sobre la señal de voz. Entonces, los distintos algoritmos para la extracción del pitch intentan eliminar mediante diversos métodos, los efectos de la estructura formant sobre la señal de voz. Sin embargo, no se ha logrado una soluciòn satisfactoria por lo que las investigaciones en el tema continúan.

III.3.3 "Método de Autocorrelación para la Estimación del Pitch"

Una de las dificultades en lograr una estimación confiable del pitch, a través de un amplio rango de expresiones y parlantes, es el efecto de la estructura formant sobre las mediciones relativas a la periodicidad de la señal de voz [RAB76]. Por tanto, para lograr una detección confiable del pitch, es deseable que los efectos de las resonancias del conducto vocal sobre la señal de voz sean reducidos al máximo.

El método de autocorrelación, véase figura 3.5, propone un recorte central de la señal de voz, para reducir los efectos de las resonancias del conducto vocal sobre la periodicidad de la señal. El proceso de recorte central consiste en suprimir la señal entre ciertos niveles, matemáticamente:

$$\begin{aligned} s(n) &= 1 \quad \text{si s(n)} \geq C_L \\ s(n) &= -1 \quad \text{si s(n)} \leq C_L \\ s(n) &= 0 \quad \text{en cualquier otro caso} \end{aligned}$$

Debido al amplio rango dinámico de la señal de voz, el nivel de recorte se escoge, de tal manera que se prevengan perdidas de información, sobre todo en los frames de voz, en los que la señal está incrementando o decrementando su amplitud.

El modo en el cuál, el nivel de recorte se fija, es el siguiente: el frame de voz, el cuál consta de 256 muestras, se divide en tres partes iguales. A continuación el algoritmo hallará el máximo absoluto, tanto de la primera parte como de la tercera, de tal manera que se tendrán dos máximos. Entonces el nivel de recorte se fija como un porcentaje del menor de estos dos máximos. Extensivas simulaciones por computadora han mostrado que dicho porcentaje es del 80% para la mayoría de los casos [RAB76].

El siguiente paso en el algoritmo es, el cálculo de la función de autocorrelación para el frame de voz:

$$R_x(m) = \sum_{n=0}^{256-m} x(n)x(n+m) \qquad m = M_i, M_i+1, \cdots, M_f \tag{3.30}$$

donde: M_i es el retraso inicial y M_f es el retraso final, para los cuales la función de autocorrelación es calculada. Los valores típicos para M_i y M_f son 25 y 200 respectivamente, los cuales corresponden a un rango de Pitch de 25 a 200 Hz. para una frecuencia de muestreo de 10 000 Hz. [RAB76]. Adicionalmente $R_x(0)$ es calculada, con el objeto de normalizar la función de autocorrelación.

Se debe hacer notar que en el cálculo de la función de autocorrelación, se asume que las muestras que estan fuera del frame son cero. Esto implica, que la

función de autocorrelación sufre de una ponderación lineal, la cuál es uno para m = 0 y decae linealmente hasta llegar a cero para m = 255. Está ponderación lineal tiene el efecto de realzar el pico correspondiente al periodo pitch, con respecto a los picos que corresponden a múltiplos de dicho periodo, reduciendo de esta manera la posibilidad de que el pitch estimado se duplique o triplique [RAB76].

El último paso en la estimación del pitch consisten en hallar el valor máximo de la función de autocorrelación (normalizada con respecto a $R_x(0)$), en el intervalo $[M_i, M_f]$. Tanto la localización como el valor del pico máximo son guardados en memoria. Si el valor del pico máximo excede un umbral sonoro - no sonoro (en el orden de 0.30 [RAB76]), el frame de voz es clasificado como sonoro, y el periodo pitch es igual a la posición del pico máximo. Si el pico máximo cae por debajo del valor de dicho umbral, el frame de voz es declarado como no sonoro.

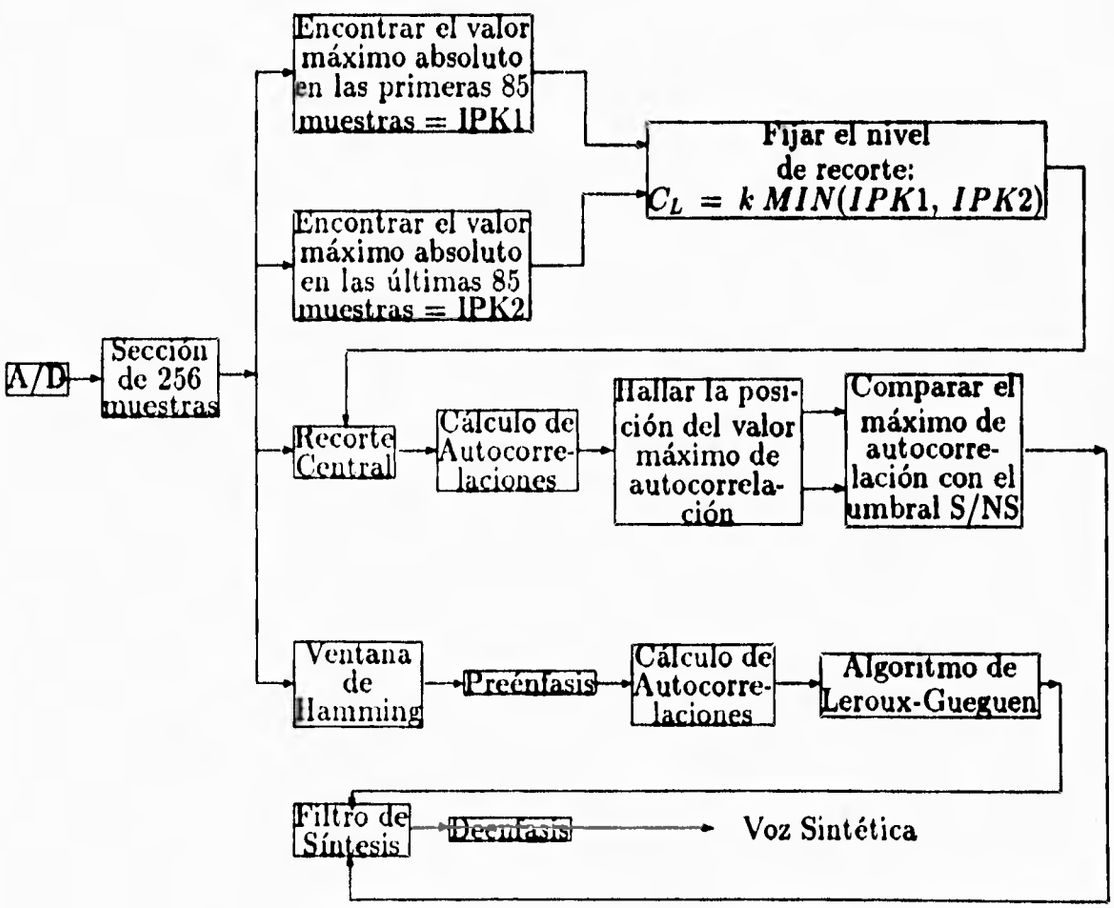


Figura 3.5: Modelo para sintetizar voz cuyo detector de pitch se basa en el método de autocorrelación con recorte central.

III.3.4 "Método Cepstral para la Estimación del Pitch."

La figura 3.6, muestra el diagrama de bloques para la síntesis de voz, mediante LPC con el método cepstral para la estimación del pitch.
La propuesta del método cepstral para la estimación del pitch consiste en concebir

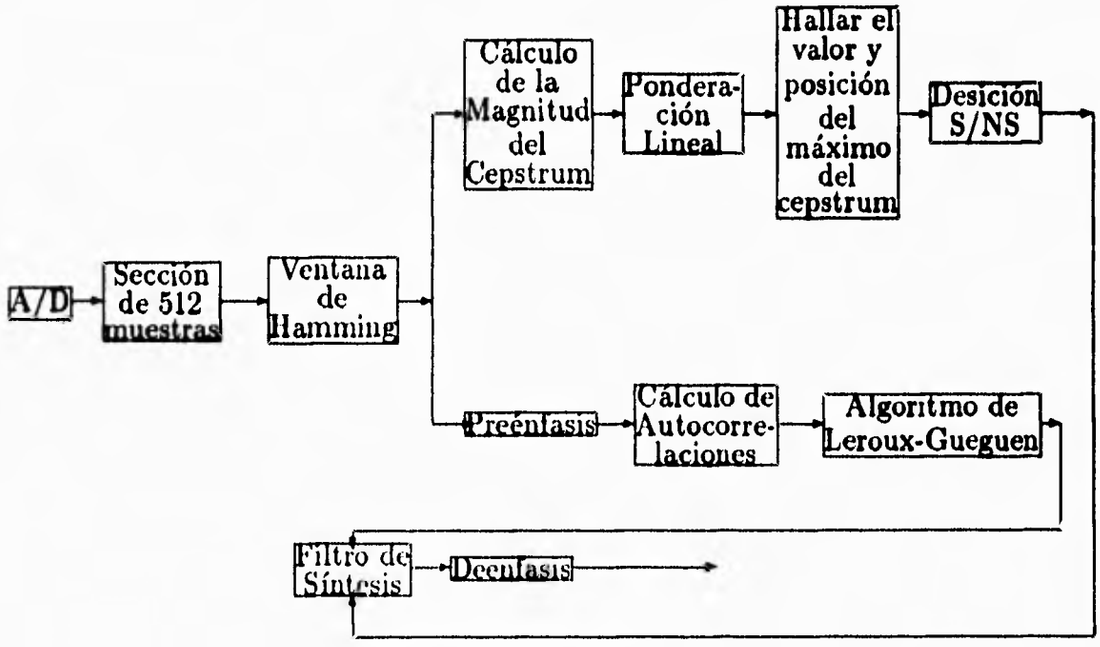


Figura 3.6: Sintetizador de voz basado en la codificación lineal predictiva, cuyo detector de pitch se basa en el método cepstral.

una nueva función, en la cuál los efectos de la fuente sonora (excitación) y del conducto vocal, sean casi independientes o fácilmente identificables y separables. La transaformada de Fourier del espectro logarítmico de potencia es dicha función. Está separa los efectos del conducto vocal, de los de la fuente sonora (excitación). Matemáticamente: De la ec.(3.29):

$$log|F(\omega)|^2 = log[\,|S(\omega)|^2\,|H(\omega)|^2\,] \tag{3.31}$$

$$log|F(\omega)|^2 = log|S(\omega)|^2 + log|H(\omega)|^2 \tag{3.32}$$

De la ecuación anterior, se observa que los efectos del conducto vocal y la fuente sonora (excitación) son aditivos, en lugar de estar convolucionados como en la función de autocorrelación [NOLL67]. La importancia de este hecho se puede explicar como sigue: el conducto vocal produce una ondulación de "baja frecuencia" en el espectro logarítmico (envolvente de este último), mientrás que la periodicidad de la fuente sonora se manifiesta como un rizo de "alta frecuencia" en el espectro logarítmico. Por lo tanto, el espectro del espectro logarítmico de potencia tiene un pico pronunciado, el cuál se debe a los rizos de alta frecuencia del espectro logarítmico, y un pico más achatado con respecto al anterior, el cuál se debe a la estructura formant de baja frecuencia en el espectro logarítmico. Si se desea, el pico correspondiente a la periodicidad de la señal de voz puede hacerse más pronunciado al elevar al cuadrado el segundo espectro. Esta función, el cuadrado de la transformada de Fourier del espectro logarítmico de potencia, es llamada el "Cepstrum" [NOLL67].

Para prevenir la posible confusión entre las componentes frecuenciales de una función temporal, y la frecuencia de los rizos, en el espectro logarítmico, Tukey usó el término "quefrency" para la "frecuencia" de los rizos espectrales. La quefrency tiene unidades de ciclos por hertz o, simplemente segundos. Adoptando está terminología, el cepstrum consiste de un pico en alta quefrency, igual al periodo pitch en segundos, e información en baja quefrency, debida a la estructura formant en el espectro logarítmico.

Las figuras 3.7 y 3.8, muestran el diagrama de flujo del algoritmo usado para determinar, a partir del pico cepstral del n-ésimo cepstrum, si el n-ésimo segmento de voz es sonoro o no sonoro. En este documento sólo se señalarán los puntos más importantes, para aquéllos interesados en una descripción más destallada la podrán encontrar en [NOLL67].

Después de la obtención del cepstrum, este sufre una ponderación lineal, en el intervalo [1,15]ms, con un factor de 1 para 1ms, y de 5 para 15ms. De está manera se compensa el decremento de la amplitud de los picos cepstrales, al incrementarse la quefrency. Luego, se halla el pico máximo en el intervalo de [1,15]ms, el cuál se compara con un umbral sonoro - no sonoro, para determinar si el frame de voz es sonoro o no sonoro.

Dado que los picos cepstrales tienden a decrementar su amplitud al final de los segmentos sonoros de voz, es necesario compensar dicho efecto para evitar la posibilidad, de que un pico máximo caiga erróneamente por debajo del umbral sonoro - no sonoro. Esta compensación consiste en reducir el umbral por un factor de 2, en el intervalo de ± 1 ms del anterior periodo pitch. De aquí que, sea necesario almacenar en memoria el pitch del frame anterior. Cabe señalar, que está compensación sólo se aplica en el caso de una serie consecutiva de frames sonoros.

Para cada frame i, además del pitch estimado del frame anterior i - 1, es necesario un estimado preliminar del pitch del siguiente frame i + 1. De está manera, si se detecta que el presente frame, i, es sonoro, mientrás que el anterior (i - 1) y el siguiente (i + 1) son no sonoros, se descarta el resultado y se declara el presente

frame (i), como no sonoro. Otra razón para necesitar de una estimación preliminar del pitch del siguiente frame es el doblamiento del pitch, esto es, estimar un periodo pitch como el doble del valor "verdadero". Un procedimiento para evitar el doblamiento del pitch es el siguiente:

1.- Al determinar que un pico cepstral excede el umbral, es necesario determinar si su coordenada temporal es $\geq$ a la estimación del periodo pitch del intervalo anterior.

2.- En caso afirmativo, existe la posiblidad de doblar el pitch. Se procede a hallar el pico máximo en un rango de $\pm$ 0.5ms la mitad de la coordenada temporal del pico correspondiente al presente (i) cepstrum.

3.- Si este último pico máximo cae dentro del intervalo de $\pm$1ms del periodo pitch del frame anterior, se reduce el umbral sonoro - no sonoro a la mitad.

4.- Si este pico excede dicho umbral, entonces el periodo pitch es igual a su coordenada temporal.

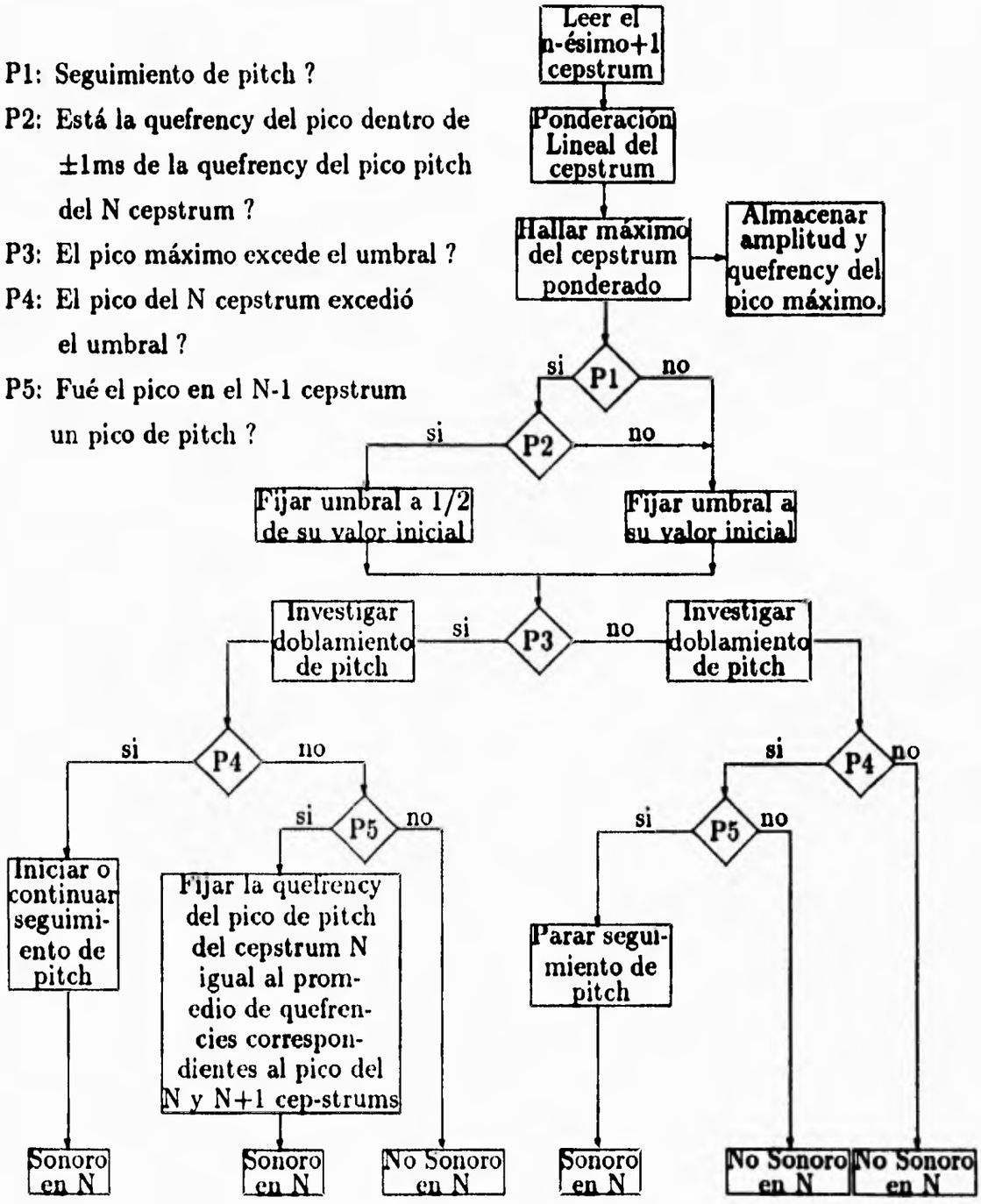
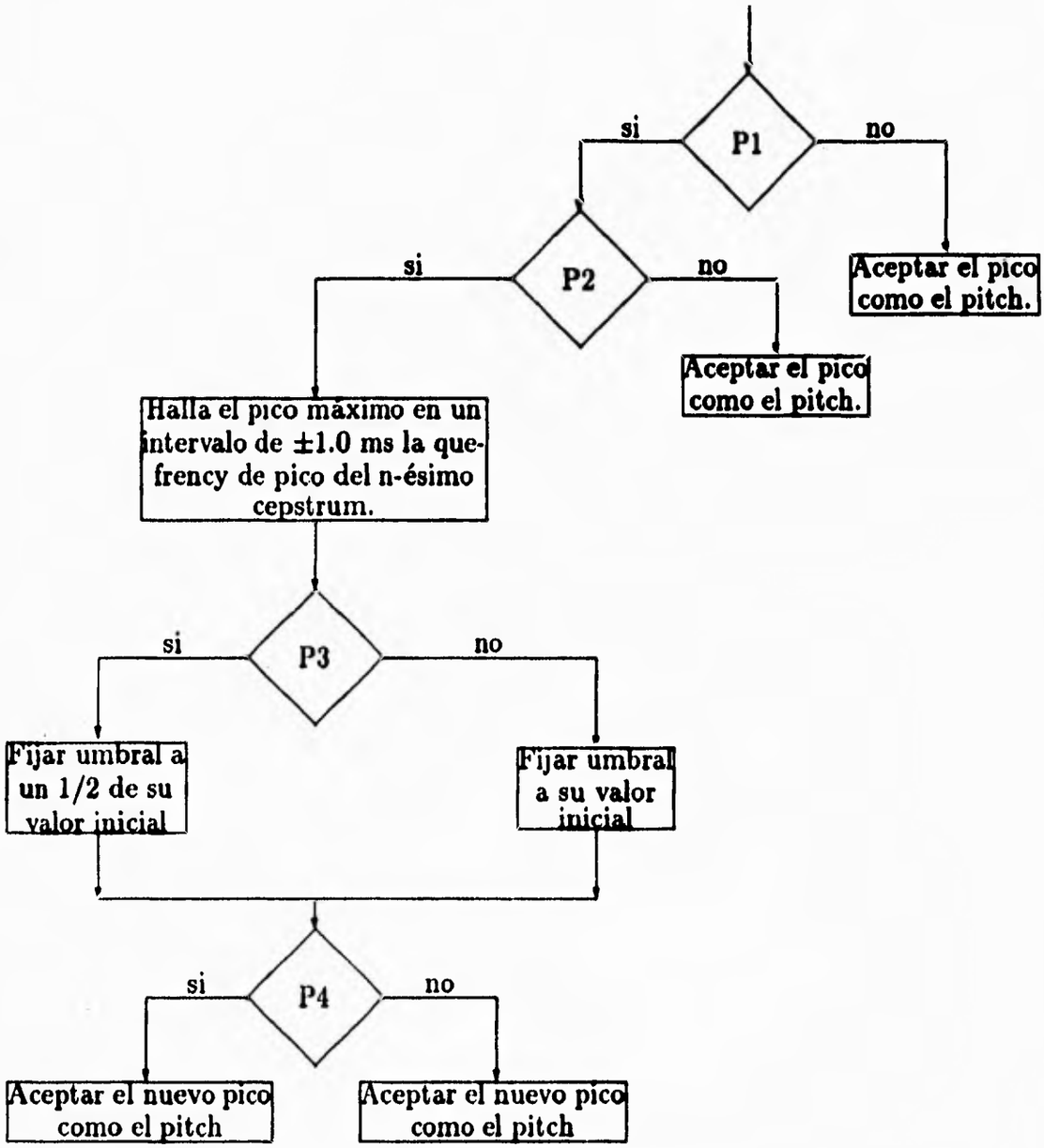


Figura 3.7: Algoritmo de decisión Sonoro - No Sonoro, utilizado en la estimación del pitch mediante método cepstral



P1: Seguimiento del Pitch?

P2: Es la quefrency del pico $\geq$ a la quefrency del pico del n-ésimo cepstrum?

P3: Está la quefrency del pico máximo dentro de un intervalo de ±1.0ms la quefrency del pico del n-ésimo cepstrum?

P4: El máximo excede el umbral?

Figura 3.8: Subrutina para Investigar Doblamiento de pitch.

"III.3.5 "Método SIFT (Simplified Inverse Filter Tracking) para la Estimación del Pitch."

La propuesta de este método, (véase figura 3.9.), para reducir los efectos de la estructura formant sobre la periodicidad de la señal consiste en utilizar la señal de error residual, la cuál contiene información sobre la excitación de la señal, incluyendo el periodo pitch. Pero su contenido de información acerca de la estructura formant es mínimo. La señal residual de error, se obtiene de filtrar la señal de voz a través del filtro inverso, el cuál está dado por el denominador de la ec.(4):

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_p z^{-p} \tag{3.33}$$

Los coeficientes a_i son obtenidos como se indica en III.2.1. Entonces el filtro inverso A(z), reduce los efectos del conducto vocal sobre la periodicidad de la señal; dado que la estructura formant es la envolvente del espectro logarítmico de potencia, podemos afirmar que el filtro A(z) es un aplanador de espectros. En la figura 9., se muestra el diagrama de bloques del sintetizador de voz, que utiliza el método SIFT para la estimación del pitch.

El primer paso para la estimación del pitch es filtrar y decimar la señal de voz. Esto se debe a que una estimación exacta del periodo pitch puede obtenerse aún con una frecuencia de muestreo de 2 KHz, lo cuál reduce considerablemente el número de operaciones. Es necesario de que la señal sea de banda limitada (0 - 1 kHz), para evitar el aliasing, por lo que una frecuencia de corte de 800 Hz para el filtro paso bajas es una razonable elección, dado que provee la atenuación suficiente a 1 kHz. El fitro digital está dado por:

$$u_n = a_1 s_n + a_2 u_{n-1} \tag{3.34}$$

$$x_n = a_3 u_n + a_4 x_{n-1} + a_5 x_{n-2} \tag{3.35}$$

donde:

$$\begin{aligned}
a_1 &= 1 - e^{-\alpha_1 T} \\
a_2 &= e^{-\alpha_1 T} \\
a_3 &= 1 - 2e^{-\alpha_2 T} \cos(\beta_2 T) + e^{-2\alpha_2 T} \\
a_4 &= 2e^{-\alpha_2 T} \cos(\beta_2 T) \\
a_5 &= -e^{-2\alpha_2 T} \\
\alpha_1 &= (0.3572)2\pi f_c \\
\alpha_2 &= (0.1786)\pi f_c \\
\beta_2 &= (0.8938)\pi f_c \\
u_n &= 0, \quad n < 0 \\
x_n &= 0, \quad n < 0
\end{aligned}$$

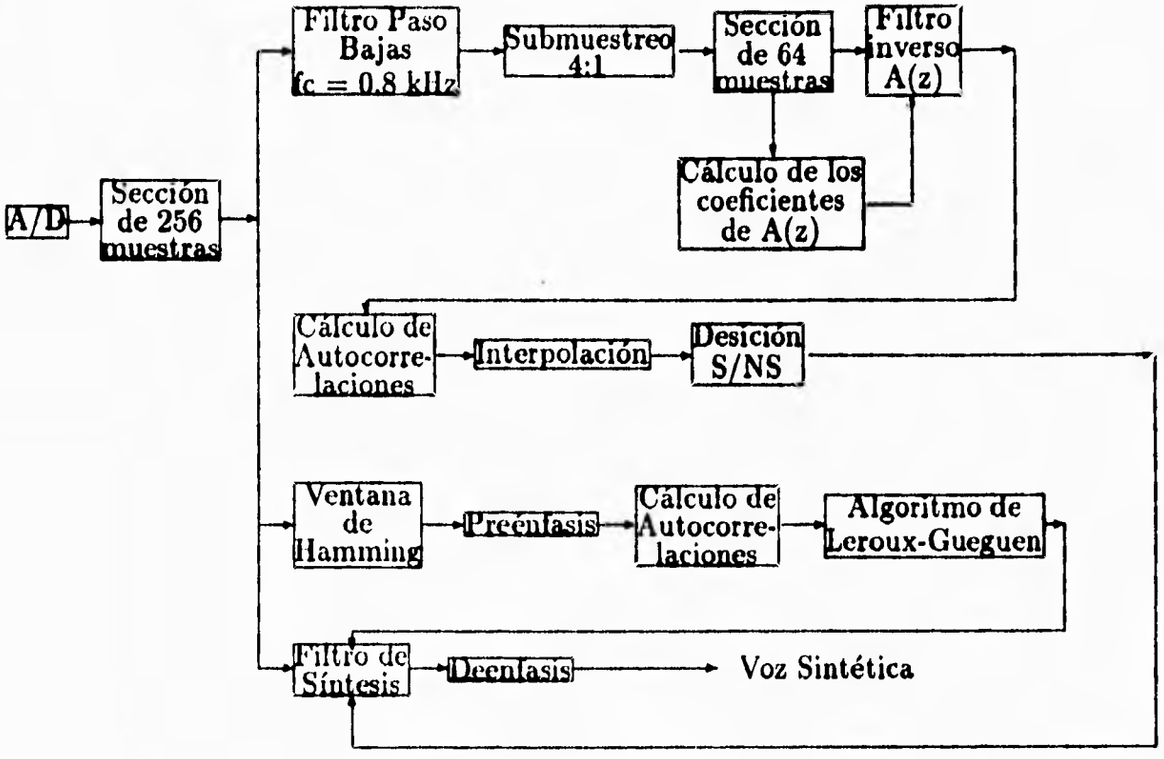


Figura 3.9: Sintetizador de voz basado en la codificación lineal predictiva, cuyo detector de pitch se basa en el método SIFT

donde s_n y x_n son las sequencias de entrada y salida, respectivamente, $f_c = 0.8$ kHz, y T = 0.125 ms. Para convertir las muestras a una razón de 2 kHz, es necesario aplicar una decimación, la cuál consiste en obtener una nueva sequencia de datos w_n, a partir de la secuencia prefiltrada x_n de la siguiente forma: si suponemos que la frecuencia de muestreo es de 8 kHz, de cada cuatro datos de x_n tomamos uno, y lo escribimos en w_n.

El siguiente paso en la estimación del pitch consiste en filtrar la señal a través del filtro inverso A(z), con lo cuál se obtendrá la señal del error residual. Dado que la señal que se filtrará a través de A(z) ha sido filtrada y decimada, es suficiente un modelo LPC de cuarto orden. El análisis LPC es aplicado a segmentos de 64 muestras, lo que corresponde a 32ms de voz. En un intervalo como tal, es razonable esperar tres periodos pitch para una voz masculina típica.

El siguiente paso consiste en calcular la autocorrelación de la señal del error residual e(n). Si P es el periodo pitch estimado del frame anterior, se halla el pico

máximo de la función de autocorrelación en el intervalo: P + ±7 retrasos. Este pico determina el periodo pitch estimado del presente frame. En el caso de un frame no sonoro. En caso de que el frame anterior fuese no sonoro, el valor de P se resetea a un valor inicial.

El último paso consiste en una interpolación, esta se debe realizar debido a que una medición excata del pitch, requiere de una resolución temporal de 0.1 a 0.15 ms. Supongamos que T = 0.125 ms. y que el Pitch "verdadero" (P) sea de 6ms, entonces el error máximo de cuantización está dado por: $T/2P_2 = 1.74$ Hz. Para el valor de T = 0.5 ms, el error máximo de cuantización es de 7.0 Hz., lo cuál es lo suficientemente grande para afectar la calidad de la voz sintética [MARK72]. Una solución simplificada a este problema se obtiene al derivar una función de interpolación trigonométrica para la secuencia de autocorrelación r_n, obtenida del filtro inverso. La ecuación matricial para la interpolación está dada por:

$$[\gamma_{\pm 3/4}\ \gamma_{\pm 1/2}\ \gamma_{\pm 1/4}]^T = A\,[\gamma_{\pm 1}\ \gamma_0\ \gamma_{\mp 1}]^T \qquad (3.36)$$

donde:

$$A = \begin{bmatrix} 0.879124 & 0.321662 & -0.150534 \\ 0.637643 & 0.636110 & -0.212208 \\ 0.322745 & 0.878039 & -0.158147 \end{bmatrix}$$

donde: γ_0 es igual al pico máximo de la secuencia de autocorrelación normalizada r_{max}; $\gamma_{+1} = r_{max+1}$; $\gamma_{-1} = r_{max-1}$; y $\gamma_{\pm 3/4}$, $\gamma_{\pm 1/2}$, y $\gamma_{\pm 1/4}$ son los puntos a interpolar en la secuencia de autocorrelación normalizada.

Para decidir si un frame es sonoro o no sonoro, el pico máximo de la secuencia de autocorrelación es hallado, después de lo cuàl se aplica la ecuación anterior. El valor máximo resultante se compara con un unbral sonoro - no sonoro, si el valor máximo rebasa dicho umbral el segmento de voz es declarado sonoro, de otro modo es declarado no sonoro. Para evitar errores causados por valores pequeños de autocorrelación, si un intervalo no sonoro se halla entre dos que han sido declarados sonoros, se redeclara este último como sonoro, y su periodo pitch es igual a la media aritmética de los dos periodos pitch correspondientesa los segmentos adyacentes.

III.4 "Excitación Multipulso."

III.4.1 "Introducción."

El tipo de codificadores que nos interesan son los vocoders, los cuáles sintetizan voz según un modelo paramétrico para la producción de voz. Los vocoders son eficientes para reducir la tasa de bits a valores menores de 4 kbits/s, sin embargo, lo logran a costa de la calidad e inteligibilidad.

El modelo usado para sintetizar voz se muestra en la figura 1. Este modelo asume que los segmentos de voz pueden ser clasificados en dos clases: sonoros y no sonoros; y que el periodo pitch para los segmentos sonoros es conocido.

La dificultad principal para producir voz sintética de alta calidad con este modelo es: la forma inflexible en que la señal de excitación es generada. Esto es, una clasificación confiable de los segmentos de voz, en dos categorías (sonoros y no sonoros), es difícil de llevar a cabo en la práctica. Además no sólo existen dos maneras de excitar al conducto vocal, ya que a menudo estas se mezclan para generar distintos sonidos [ATAL82].

Volviendo a las señales de voz, existen regiones de estas, las cuales no pueden ser clasificadas, ya sea por medios automáticos o manuales, dentro de una de las dos categorías antes mencionadas, ya que no resulta claro a cual de las dos pertenecen.

Volviendo a la naturaleza de la señal de excitación, existe evidencia según la cual, además de la excitación primaria, la cuál se efectúa al cerrase la glotis, existe una excitación secundaria, la que no sólo se genera al abrise y mantenerse abierta la glotis, sino también al cierre de la misma [ATAL82]. Estos resultados sugieren, que la excitación para segmentos sonoros debe de consistir de varios pulsos comprendidos dentro de un periodo pitch, en lugar de un sólo pulso al principio de dicho periodo. La dificultad principal en la aplicación de un modelo multipulso estriba, en el desarollo de un algoritmo que determine de una manera satisfactoria la posición y amplitud de dichos pulsos.

En los métodos de excitación multipulso no se clasifican los frames de voz en sonoros y no sonoros, sino que se determina la posición y amplitud de cierto número de pulsos por cada frame, de tal manera que la voz sintética resultante sea lo más parecida posible a la senñal original de voz.

III.4.2."Método de Excitación Multipulso."

Un procedimiento de análisis mediante síntesis para determinar las posiciones y amplitudes de los pulsos, se muestra en la figura 10.

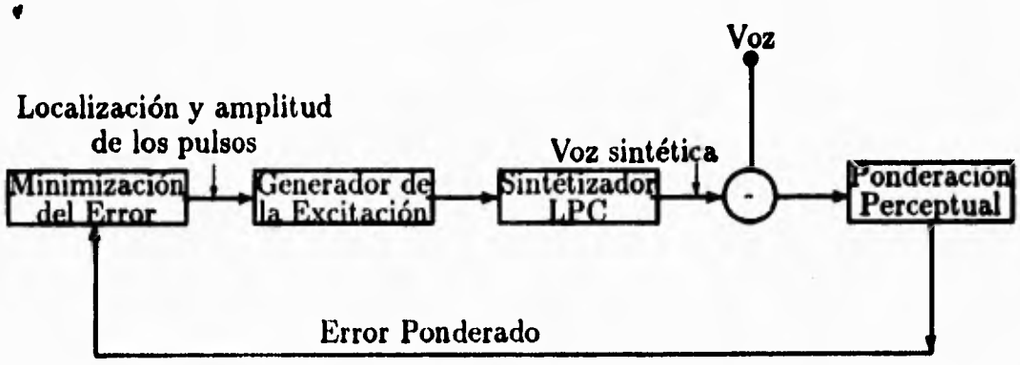


Figura 3.10: Diagrama de bloques de un procedimiento de análisis mediante síntesis para determinar la localización y amplitud de los pulsos para la excitación multi-pulso

Las muestras de la voz sintética son comparadas con la de la voz original, para producir la señal de error e_n. Esta señal de error no es significativa para la obtención de las posiciones y amplitudes de los pulsos, ya que debe ser modificada para tomar en cuenta el trato del error por la percepción humana. Esto se efectúa mediante un filtro lineal, el cuál atenúa la energía de la señal de error en las regiones de los formants. Dicho de otra forma, la señal de error sufre una ponderación, la cuál da por resultado una medida más significativa, en términos de la percepción humana, de la diferencia entre la señal de voz original y la sintétetica [ATAL82]. El siguiente paso es la obtención del error cuadrático medio ε, el cuál se obtiene al elevar al cuadrado y promediar la señal de error ponderada.

"Minimización del Error ε"

Si s(n) es la señal original y s'(n) la señal sintética, es deseable minimizar la energía de la señal de error:

$$e(n) = [s(n) - s'(n)] * w(n) \tag{3.37}$$

donde $w(n)$ es la función de ponderación, y el asterisco denota la convolución. En el dominio de frecuencia:

$$E(z) = [S(z) - S'(z)]\, W(z) \tag{3.38}$$

Si H(z) es la función de transfrencia del conducto vocal, entonces una elección natural para la función de ponderación es:

$$W(z) = \frac{H(bz)}{H(z)} \tag{3.39}$$

donde:
$$H(bz) = \frac{G}{[1 + a_1 b^{-1} z^{-1} + \cdots + a_p b^{-p} z^{-p}]}$$

b es un número entre cero y uno. La experiencia muestra que b = 0.8 produce una buena calidad de voz.

Notese que S(z)/H(z) es la señal residual y S'(z)/H(z) es la señal de excitación para generar la señal sintética. Esta excitación en nuestro caso es el pulso a ser determinado. Sea A_m la amplitud del puso con m indicando su posición. Tanto m como A_m deben ser determinados. También, sea h(n) la respuesta al impulso de H(bz). Con estas substituciones en la ec.(3.38), la energía del error a ser minimizada está dada por:

$$\sum_n e^2(n) = \sum_n [d(n) - A_m h(n - m)]^2 \tag{3.40}$$

donde d(n) es el resultado de filtrar la señal residual r(n) a través de H(bz). La minimización de la ecuación anterior se logra cuando:

$$A_m = \frac{\sum_n d(n)\, h(n - m)}{\sum_n h^2(n - m)} = \frac{C_m}{\phi} \tag{3.41}$$

donde $C_m = \sum d(n)h(n - m)$ es la correlación mutua entre h(n) y d(n), y $\phi = \sum h^2(n - m)$ es la energía de h(n). La correspondiente energía minima es:

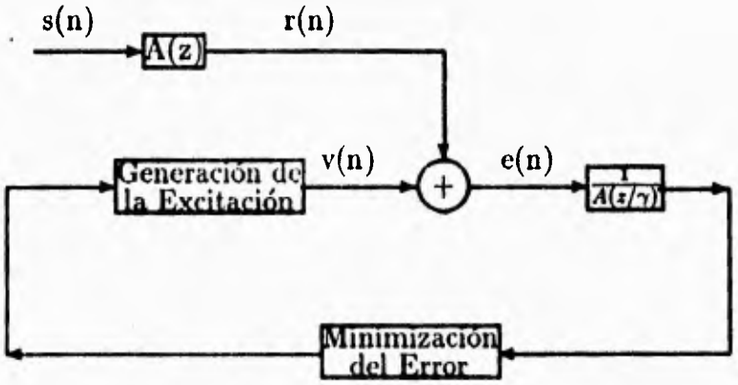
$$E = (\sum d^2(n)) - (\frac{C_m^2}{\phi}) \tag{3.42}$$

En la ecuación anterior, C_m es la única ecuación que depende de m. De aquí que, para la minimización de la energía del error residual, la posición m, del pulso es seleccionada de tal manera que, el valor absoluto de la correlación mutua C_m es maximizado. Entonces, la amplitud del pulso se determina mediante la ec.(3.41). Una vez determinado el pulso, este es substraído de la señal residual. Para determinar el siguiente pulso, el mismo procedimiento se repite, haciendo uso de los pulsos calculados anteriormente.

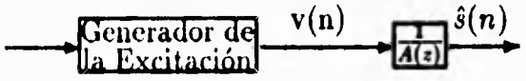
III.4.3"Método "Regular Pulse Excitation" "

"Estructura Básica del Codificador."

La estructura básica del codificador, vease la figura 3.11, puede ser vista como un modelo para la señal residulal, r(n).



[a] codificador



[b] decodificador

Figura 3.11: Diagrama de bloques de codificador regular pulse excitation RPE

La señal residual r(n), se obtiene al filtrar la señal de voz s(n), a través del filtro inverso A(z). La diferencia entre la señal residual y cierto modelo residual v(n)

[KROON86], es alimentada a través del filtro:

$$\frac{1}{A(z/\gamma)} = \frac{1}{1 + \sum_{k=1}^{P} a_k \gamma^k z^{-k}}, \quad 0 \leq \gamma \leq 1 \tag{3.43}$$

Este filtro, es en realidad una función de ponderación, la cuál hace que el error sea significativo en términos de la percepción humana. La diferencia ponderada resultante e(n), es elevada al cuadrado y acumulada, para ser usada como una medida de la efectividad del modelo v(n).

La secuencia de excitación v(n), se determina para frames adyacentes de L muestras cada uno, de la siguiente manera: a cada frame de L muestras, corresponde un cierto vector óptimo $b^k = (b(1), \cdots, b(Q))$, donde: k denotas la fase, y Q es la longitud del vector (Q < L). Entonces, cada segmento de la señal de excitación v(n), contiene Q muestras equidistantes de amplitud difrente de cero, mientras que las muestras restantes son iguales a cero. El espaciamiento entre muestras difrentes de cero es N = L / Q. Finalmente para cada valor de k se calculan los componentes del vector b^k que minimizen el error cuadrático medio. El vector que de por resultado el mínimo error es seleccionado y transmitido.

"Algoritmo de Codificación."

Sea M_k la matriz de posición, de tamaño Q x L, cuyos elementos están dados por:

$$m_{ij} = 1 \quad si \quad j = 1 * N + k - 1 \qquad 0 \leq i \leq Q - 1$$

$$m_{ij} = 0 \qquad \text{En cualquier otro caso} \qquad 0 \leq j \leq L - 1$$

El vector de la excitación $v^{(k)}$, correspondiente al k-ésimo defasamiento de la matriz M_k, se puede escribir como:

$$v^{(k)} = b^{(k)} M_k \tag{3.44}$$

Sea H la matriz triangular superior de tamaño L x L, cuya j-ésima columna contiene

la respuesta al impuso h(n) del filtro $1/A(z/\gamma)$:

$$H = \begin{bmatrix} h(0) & h(1) & \cdots & h(L-1) \\ 0 & h(0) & & h(L-2) \\ 0 & 0 & & h(L-3) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & 0 & h(0) \end{bmatrix}$$

Sea e_0 la salida del filtro $1/A(z/\gamma)$ debido a la memoria del mismo. Entonces la señal e(n) producida por el vector de entrada $b^{(k)}$, puede ser expresada como:

$$e^{(k)} = e^{(0)} - b^{(k)}H_k, \quad k = 1, \cdots, N \tag{3.45}$$

donde:

$$e^{(0)} = e_0 + rH \tag{3.46}$$

$$H_k = M_k H \tag{3.47}$$

donde el vector r, representa la señal residual r(n). El objetivo es minimizar el error cuadrático medio:

$$E^{(k)} = e^{(k)}e^{(k)t} \tag{3.48}$$

donde t denota la transpuesta. Para una determinada fase, las componentes óptimas de $b^{(k)}$ pueden ser calculadas de (3.45) y (3.48), mediante la igualación de $e^{(k)}H_k^t$ con cero. Entonces:

$$b^{(k)} = e^{(0)}H_k^t[H_k H_k^t]^{-1} \tag{3.49}$$

Sustituyendo (3.49) en (3.45), y el resultado de esta sustitución en (3.48), se obtiene la siguiente expresión para el error:

$$E^{(k)} = e^{(0)}[I - H_k^t[H_k H_k^t]^{-1}H_k]e^{(0)t} \tag{3.50}$$

El vector $b^{(k)}$ que de el mínimo valor de $E^{(k)}$ para toda k, es seleccionado.

"Algoritmo Simplificado."

La complejidad del codificador RPE, puede ser substancialmente reducida sin ninguna degradación significativa de la calidad de voz. Sea h(n) la respuesta al impulso del filtro invaraiante $1/C(z/\gamma)$ [KROON86]. Matemáticamente:

$$\frac{1}{C(z/\gamma)} = \frac{1}{1 + \sum_{k=1}^{9} c_k \gamma^k z^{-k}} \tag{3.51}$$

La matriz H se utilizará en (47), como se define a continuación [KROON86]:

$$H = \begin{bmatrix} h(0) & h(1) & \cdots & h(L-1) & 0 & \cdots & 0 \\ 0 & h(0) & \cdots & h(L-2) & h(L-1) & & 0 \\ 0 & 0 & \cdots & h(L-3) & h(L-2) & & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & \cdots & h(0) & \cdots & h(L-1) & 0 \end{bmatrix}$$

Se descarta la aproximación de orden cero e_0 en (3.46). Entonces (3.45) y (3.49) se expresan como:

$$e^{(k)} = rH - b^{(k)}H_k \tag{3.52}$$

y

$$b^{(k)}[H_k H_k^t] = rHH^t M_k^t \tag{3.53}$$

respectivamente. Denotando:

$$S = HH^t \tag{3.54}$$

y renombrando:

$$H_k H_k^t \approx r_0 I \tag{3.55}$$

con:

$$r_0 = \sum_{i=0}^{L-1} h^2(i) \tag{3.56}$$

como una constante de codificación, es fácil mostrar que:

$$b^{(k)} = \frac{1}{r_0} rSM_k^t \tag{3.57}$$

Interpretando M_k^t como un operador de submuestreo, en la ecuación anterior $b^{(k)}$ es la salida submuestreada del suavizador S, cuya entrada es la señal residual r(n) escalada. La selección de la excitación se basa en la minimización del error dado por la ec.(50). Bajo las condiciones mencionadas anteriormente, está ecuación se expresa como:

$$E^{(k)} = rHH^t r^t - r_0 b^{(k)} b^{(k)t} \tag{3.58}$$

De aquí que:

$$min\{E^{(k)}\} = max\{b^{(K)} b^{(k)t}\} \tag{3.59}$$

Ahora el procedimiento completo es extremadamente fácil. La señal residual r es suavizada por el suavizador S = HH^t. El vector resultante es submuestreado mediante la aplicación de M_k^t, y el vector $b^{(k)}$ para el cuál $b^{(k)}b^{(k)t}$ sea máximo es seleccionado.

Capítulo Cuatro

"Resultados de la Evaluación de los Esquemas Clásicos para la Síntesis de Voz."

IV.1 "Introducción."

Los dos objetivos que se persiguen en este capítulo son: 1) presentar una serie de evaluaciones del desempeño de cada uno de los algoritmos de síntesis de voz, que se discutieron en el capítulo tres y, 2) comparar el desempeño de dichos algoritmos entre sí y, seleccionar aquel que reuna las mejores propiedades.

En el caso de nuestro primer objetivo, las evaluaciones efectuadas consistirán en la variación de cada uno de los parámetros más significativos para el desempeño de los distintos algoritmos de síntesis de voz; mientras que el resto de dichos parámetros permanece constante, es decir, en los valores recomendados para los mismos por la literatura. Esta variación de parámetros genera un universo de señales sintéticas, dentro del cuál cada señal se corresponde, en una relación uno a uno, con un vector, cuyas componentes son los parámetros utilizados para generar dicha señal. Es de nuestro interés comparar las distintas señales cuyos vectores de parámetros tan sólo varíen en la misma componente; para determinar el rango dinámico de dicha componente o parámetro. Esta comparación es tan sólo de orden perceptual debido a que una comparación de espectros, señales temporales, etc., no será indicativa de las diferencias perceptuales existentes entre las señales sintéticas que están sujetas a dicha comparación.

El tratar de establecer la calidad perceptual de una señal sintética de voz con respecto a una señal sintética de referencia, no es una tarea sencilla, ya que no existe un criterio para evaluar de una forma mesurable la calidad de la voz sintética. Por tanto no se trata de una propiedad cuantitativa y mesurable, sino de una propiedad subjetiva, la cuál dependerá del sujeto que realice la medición. Sin embargo, cuando la diferencia perceptual entre dos señales de voz es grande la incertidumbre de cometer un error de percepción decrece, lo cuál nos sirve de ayuda para descartar aquellos valores del parámetro en cuestión que produzcan dicha diferencia perceptual. Por tanto, la determinación confiable de los rangos dinámicos para los distintos parámetros, dónde las diferencias perceptuales producidas por las variaciones de dichos parámetros sean minimas, es posible.

Finalmente, una vez que se han obtenido dichos rangos dinámicos, es de interés efectuar una comparación del desempeño entre los distintos algoritmos de síntesis de voz. Dicha comparación debe incluir dos aspectos fundamentales de cualquier sistema básado en algoritmos de síntesis de voz: 1) la calidad de la voz sintética y, 2) el número de parámetros que representan un frame de voz, ya que de ellos dependerá la cantidad de información a transmitir. Al aumentar la cantidad de parámetros que caracterizan un frame de voz, la cantidad de información a ser transmitida también aumentará, lo cuál trae como consecuencia que el número de señales que pueda transmitir determinado canal de comunicación se vea seriamente reducido, además la capacidad de memoria del sistema de procesamiento debe ser lo suficientemente grande para poder almacenar dichos parámetros y, el tiempo de procesamiento aumenta, lo cual dificulta, de ser necesario, una ejecución en tiempo

real. Por tanto, la eficiencia de un algoritmo de síntesis de voz es proporcianalmente inversa al número de parámetros necesarios para representar un frame de voz y, directamente proporcional a la calidad de voz sintética lograda por el algoritmo en cuestión.

IV.2 "Obtención de Datos Reales para la Síntesis de Voz."

En todo sistema, en el cuál se efectue algún procesamiento digital de señales, un factor de especial interés es el tipo de señales que intervienen en dicho sistema, así como la calidad percepetual de las mismas. Para evaluar la calidad perceptual de una señal tomamos en cuenta los siguientes factores: 1) el valor de la razón de la potencia de la señal a la potencia del ruido (SNR) y, 2) si la señal sufre de algún tipo de interferencia, generalmente de tipo electrónica, que provoque una distorsión o alteración significativa tanto de la envolvente de la señal como de su calidad perceptual.

El tipo de señal así como su calidad perceptual dependerá del tipo de aplicación que se desee implementar, por ejemplo: en los sistemas de reconocimiento de voz, se emplean algoritmos para la detección de principio y fin de palabra en ambientes ruidosos; mientrás que para los algoritmos de síntesis de voz, aquí evaluados, según mi experiencia es necesaria la obtención de señales de voz que cumplan con los siguientes requisitos: 1) una razón SNR elevada, 2) evitar cualquier clase de interferencia electrónica que cause una distorsión notable en la señal, ó una calidad perceptual baja, como: ruido de motores de ac, lamparas fluorescente y, en el diseño de los circuitos electrónicos que intervienen en la obtención de datos reales es muy importante cuidar los siguientes factores: acoplamiento de impedancias, amplificación de ruido, saturación de amplificadores, etc. En el presente trabajo se utilizó la tarjeta y software comercial "Sound Blaster". La frecuencia de muestreo de la tarjeta "Sound Blaster" puede ajustarse de 5000 a 44100 Hz., así que dicha frecuencia de muestreo se ajustó a 8000 Hz. Además del ajuste de frecuencia en la tarjeta "Sound Blaster", se puede elegirse entre una digitalización con 256 niveles (8 bits) o, una digitalización a 65536 niveles (16 bits). En este trabajo, se eligió la primera opción.

IV.3 "Señal de Prueba."

Es importante señalar el hecho de que para las evaluaciones de este capítulo se utilizó la misma señal de voz, a menos que se indique lo contrario. Esta señal dice: "Esta es una señal típica de voz". Esta señal tiene una duración de 7.09 seg., fué muestreada a 8000 Hz. y, se digitalizó a 8 bits.

IV.4 "Evaluación del Método de Síntesis: LPC con AUTOC para la Extracción del Pitch."

IV.4.1 "Introducción."

Este método se explicó en detalle en las secciones III.2 y III.3.3, aquí sólo se efectuará una evaluación perceptual de las señales sintéticas generadas por el algoritmo, al variar los parámetros de este último.

A continuación se describen los detalles de la implementación del algoritmo:

1) El método se programó en leguaje C y, se utilizó una estación de trabajo para el desarollo de las simulaciones. Esto es importante, ya que la velocidad de las estaciones de trabajo hace posible el análisis y síntesis de varios segundos de voz, lo cuál de otra manera sería una tarea muy díficil de ejecutar, ya que el procesamiento podría tardar horas.

2) Como ya se mencionó anteriormente, la obtención de datos reales para la evaluación del algoritmo, se efectuó mediante la tarjeta y/o software comercial "Sound Blaster". La frecuancia de muestreo de esta tarjeta se ajustó a 8000 Hz y, se digitalizó la señal de voz a 8 bits (256 niveles).

3)La longitud de frame en muestras para el análisis LPC y extracción de Pitch fué de 256 muestras, o lo que es equivalente a 32 ms.

4)Al frame para el análisis LPC se le aplico una ventana de Hamming.

5)En muchos algoritmos para el procesamiento digital de señales de voz se recomienda un traslape entre frames sucesivos, con el fin de evitar una especie de "brinco perceptual" debido al cambio brusco de parámetros entre un frame y su cosecutivo. Este traslape se escogió de una longitud de 86 muestras, o lo que es equivalente a 10.75 ms.

6) El algoritmo de AUTOC que se utiliza para determinar si el frame en cuestión

es sonoro ó no sonoro, tiene dos parámetros importantes: a) el parámetro K, que sirve para determinar el nivel de recorte que se aplicará al frame en cuestión y, 2) el umbral sonoro - no sonoro (S/NS), el cúal como su nombre lo indica sirve para determinar si el frame en cuestión es sonoro o no sonoro. En este caso se utilizaron los valores recomendados por [RAB76], es decir, K = 0.8 y S/NS = 0.3.
7) Dentro de método de AUTOC, se necesita hallar el pico máximo de las autocorrelaciones dentro de cierto intervalo. Este pico máximo se compara con el umbral S/NS para determinar si el frame en cuatión es sonoro o no sonoro. El intervalo de búsqueda para el pico máximo de la función de autocorrelación se escogio de 20 a 200, lo cuál correponde al intervalo de frecuencias para el pitch: [40, 400] Hz.
8) El modelo LPC que se utilizó fué de décimo orden.
9) No se detectaron silencios.

Las evaluaciones realizadas para este algoritmo consistieron en la variación de los siguientes parámetros:
a) El parámetro para determinar el nivel de recorte K.
b) El umbral sonoro - no sonoro S/NS.
c) El orden (P) del modelo LPC.
d) El número de muestras que se traslapan entre frames sucesivos.

IV.4.2 "Variaciones del Nivel de Recorte (K)."

En el algoritmo de autocorrelación con recorte central (AUTOC), uno de los parámetros más importantes es el denominado con la letra mayúscula K, el cuál sirve para determinar el nivel de recorte [RAB76]. Los resultados de las evaluaciones que a continuación se presentan, consisten en la variación de dicho parámetro K, mientras que el resto de parámetros permanecen sin variación alguna respecto de los valores recomendados por [RAB76], tal como se listaron en IV.4.1.

Dado que [RAB76] afirma que el valor óptimo para el parámetro K es de 0.8, la primera prueba que realice fué variar K de: 0.5 a 0.9. Los resultados de esta prueba se muestran a continuación:

K	Tiempo	Comentarios
0.5	26.3s	Las difrencias perceptuales con respecto a la señal generada con una K = 0.8, son meramente subjetivas.
0.6	26.9s	Aquí también las diferencias perceptuales con respecto a la señal generada con una K = 0.8, son subjetivas.
0.7	27s	Las diferencias perceptuales con respecto a la señal generada con una K de 0.8, son subjetivas.
0.75	27.1s	El resultado es prácticamente igual al obtenido con una K de 0.8.
0.8	27.3s	
0.85	27.5s	Las diferencias perceptuales con respecto a K = 0.8, son subjetivas.
0.9	27.8s	Aquí en la gráfica de la señal contra el tiempo discreto, aparecen picos indeseables de gran amplitud y corta duración, los cuales provocan una especie de "brinco perceptual" al pronunciarse la letra "s".

Dado que en la prueba anterior no se observan grandes diferencias salvo para K = 0.9, se decidió hacer una segunda prueba. En esta, el parámetro K tomaría los valores: 0.1, 0.2, 0.3, 0.4, 0.95. Los resultados se muestran a continuación:

K	Tiempo	Comentarios
0.1	26.4s	Las diferencias perceptuales con respecto a la señal generada con una K de 0.8, son subjetivas.
0.2	26.3s	Resultado prácticamente igual al anterior.
0.3	26.3s	Las diferencias perceptuales con respecto a la señal generada con una K = 0.8, son subjetivas.
0.4	26.8s	Las diferencias perceptuales con respecto a la señal generada con K = 0.8, son subjetivas.
0.95	27.8s	Aquí en la gráfica de la señal contra el tiempo discreto, aparecen picos indeseables tanto positivos como negativos de gran amplitud y corta duración en las palabras esta, es y tipica. Estos picos provocan "brincos perceptuales" en la pronunciación de dichas palabras.

El resumen de dichas pruebas se muestra en la sig. tabla:

K	Comentarios
0.1 - 0.85	Diferencias perceptuales subjetivas
0.9 - 0.95	"Brincos perceptuales" en las letras "s" debido a picos tanto positivos como negativos de gran amplitud y corta duración en la gráfica de la señal contra el tiempo discreto

El tiempo de ejecución no es un factor, cuya variación en estas pruebas sea de importancia, ya que la diferencia entre el tiempo máximo y mínimo de ejecución es de 1.4s, el cuál al ser comparado con el tiempo de ejecución para nuestra K de referencia (0.8) nos da: (1.4/27.1)*100 = 5.16% de variación.

IV.4.3 "Variaciones del Umbral Sonoro - No Sonoro (S/NS)."

En el algoritmo de autocorrelación con recorte central (AUTOC), uno de los parámetros más importantes es el umbral sonoro - no sonoro, el cuál sirve para determinar si el frame en cuestión es sonoro ó no sonoro [RAB76]. Los resultados de las evaluaciones que a continuación se presentan, consisten en la variación de dicho umbral S/NS, mientrás que el resto de parámetros permanecen sin variación alguna respecto de los valores recomendados por [RAB76], tal como se listaron en IV.4.1.

Dado que [RAB76] afirma que el valor óptimo para el umbral S/NS es de 0.3, la primera prueba que realicé fue variar S/NS de: 0.2 a 0.4. Los resultados de esta prueba se muestran a continuación:

S/NS	Tiempo	Comentarios
0.2	25.2s	Las diferencias perceptuales con respecto de la señal generada con un S/NS de 0.3, son subjetivas.
0.23	25.7s	Las diferencias perceptuales con respecto de la señal generada con un S/NS de 0.3, son subjetivas.
0.26	26.7s	Resultado prácticamente igual al de utilizar un S/NS de 0.3 para generar la señal.
0.3	27.5s	
0.33	27.8s	Aquí, en la gráfica de la señal contra el tiempo discreto aparece un pico indeseable de gran amplitud y corta duración en la palabra "es", el cuál provoca una especie de "brinco perceptual" en la pronunciación de la letra "s".
0.36	28.0s	El mismo resultado que el anterior, sólo que en este caso el pico se presenta en la palabra señal.
0.4	28.5s	El mismo resultado que el anterior.

Dado que en la prueba anterior no se observan grandes diferencias perceptuales salvo el surgimiento de picos indeseables de gran amplitud y corta duración, en las graficas de la señal contra el tiempo discreto, para valores del umbral sonoro - no sonoro (S/NS) mayores o iguales a 0.33; se decidio hacer una segunda prueba, en espera de mayores diferencias perceptuales. En esta, el parámetro S/NS tomaría los valores: 0.1, 0.5, 0.6, 0.7, 0.8 y 0.9. Los resultados se muestran a continuación:

S/NS	Tiempo	Comentarios
0.1	24.0s	Las diferencias perceptuales con respecto de la señal generada con un S/NS de 0.3, son subjetivas.
0.5	29.2s	Aquí, en la gráfica de la señal contra el tiempo discreto aparecen picos indeseables de gran amplitud y corta duración en las palabras:"esta", "es" y señal. Estos "picos" provocan una especie de "brincos perceptuales" cuando la letra "s" es pronunciada. Es necesario mencionar que salvo el anterior problema, la diferencias perceptuales con respecto de la señal generada con un S/NS de 0.3, son subjetivas.
0.6	30.1s	Aquí se presenta el mismo problema que en el caso anterior. Además en la gráfica de la señal contra el tiempo discreto, la amplitudes negaticas son mayores con respecto a las de la gráfica para un S/NS de 0.3
0.7	30.8s	El mismo resultado que para el caso anterior.
0.8	31.4s	El mismo resultado que en los dos casos anteriores, sólo que aquí la calidad de la voz sintética se degrado notablemente con respecto a la señal generada con un S/NS = 0.3
0.9	31.2s	Resultado prácticamente igual al anterior

A continuación se muestra una tabla, en la cuál, se resumen los resultados obtenidos de las dos pruebas anteriores:

S/NS	Comentarios
0.1 - 0.33	Diferencias perceptuales subjetivas
0.36 - 0.5	"Brincos perceptuales" en la pronunciación de las letras "s".
0.6 - 0.7	Lo mismo que en el caso anterior. Además las señales temporales tienen menores amplitudes que las correspondientes a la señal generada con un S/NS = 0.3
0.8 - 0.9	Lo mismo que en el caso anterior, sólo que aquí la calidad perceptual de la señal sintética se degrado notablemente con respecto a la señal generada con un S/NS. Aquí la voz es más grave.

Como se puede observar de las primeras dos tablas, el tiempo de ejecución del programa aumenta al aumentar el valor del umbral sonoro - no sonoro. Esto se debe a que al aumentar dicho umbral el porcentaje de frames no sonoros aumenta, lo cuál incrementa el tiempo de ejecución del programa dado que para un frame no sonoro, además de la ejecución de la subrutina de detección del pitch es necesario generar ruido blanco. Dicha generación de ruido blanco no es necesaria en el caso de que el frame halla sido declarado como sonoro, por tanto un frame sonoro se sintetiza mucho más rápido que un frame no sonoro.

IV.4.4 "Variación del Orden (P), del Modelo LPC."

La reducción o el aumento del orden P, del modelo LPC implica una reducción o disminución de la tasa de transmisión, lo cuál tiene un significado vital dentro de un sistema de comunicaciones. Mientrás menor sea la tasa de transmisión, menor será la cantidad de memoria que se requerirá para el alamacenamiento de dichos parámetros y, menor será la cantidad de información que se debe transmitir, por tanto el canal de comunicaciones podrá transmitir y/o recibir un mayor número de señales. De lo contrario, se requiere una mayor capacidad de memoria y, el número de señales que puede manejar dicho canal de comunicaciones decrece. Sin embargo al aumentar la tasa de transmisión, la calidad perceptual de la señal de voz sintética aumenta hasta cierto punto después del cuál, las diferencias perceptuales son meramente subjetivas.

Las pruebas aquí presentadas consistirán en la variación del orden (P) del modelo LPC y, tienen por objeto determinar si los valores recomendados por la literatura [ATAL71] son correctos. Por ello, la prueba aquí realizada constió en variar el orden del modelo LPC alrededor del valor recomendado por la literatura (P = 10). En la siguiente tabla se presentan los resultados de dicha prueba:

P	Tiempo	Comentarios
6	27.3s	La calidad perceptual de la señal sintética es menor con respecto a aquella sintetizada con una P = 10. Aquí la voz sintética es más grave. Aquí la gráfica de la señal contra el tiempo discreto los picos negativos son de mayor amplitud y menor duración que con respecto a aquellos correspondientes a la señal generada con P = 10. Además se presentan picos espurios que provocan un "brinco perceptual" en la pronunciación de las letras "s".
8	27.4s	La calidad perceptual de la voz sintética es semejante a la que se obtiene al utilizar una P = 10. La única diferencia en las gráficas de la señal contra el tiempo discreto, consiste en que la amplitud correspondiente al pico mínimo, en la palabra "voz" es mayor para una P = 8 que para P = 10.
10	27.5s	
12	27.6s	Las diferencias perceptuales entre la voz sintetizada con P = 12 y aquella sintetizada con P = 10, son subjetivas.

En este caso los mejores valores son 8 y 10, como lo recomienda la literatura,

IV.4.4 "Variación del Orden (P), del Modelo LPC."

La reducción o el aumento del orden P, del modelo LPC implica una reducción o disminución de la tasa de transmisión, lo cuál tiene un significado vital dentro de un sistema de comunicaciones. Mientrás menor sea la tasa de transmisión, menor será la cantidad de memoria que se requerirá para el alamacenamiento de dichos parámetros y, menor será la cantidad de información que se debe transmitir, por tanto el canal de comunicaciones podrá transmitir y/o recibir un mayor número de señales. De lo contrario, se requiere una mayor capacidad de memoria y, el número de señales que puede manejar dicho canal de comunicaciones decrece. Sin embargo al aumentar la tasa de transmisión, la calidad perceptual de la señal de voz sintética aumenta hasta cierto punto después del cuál, las diferencias perceptuales son meramente subjetivas.

Las pruebas aquí presentadas consistirán en la variación del orden (P) del modelo LPC y, tienen por objeto determinar si los valores recomendados por la literatura [ATAL71] son correctos. Por ello, la prueba aquí realizada constió en variar el orden del modelo LPC alrededor del valor recomendado por la literatura (P = 10). En la siguiente tabla se presentan los resultados de dicha prueba:

P	Tiempo	Comentarios
6	27.3s	La calidad perceptual de la señal sintética es menor con respecto a aquella sintetizada con una P = 10. Aquí la voz sintética es más grave. Aquí la gráfica de la señal contra el tiempo discreto los picos negativos son de mayor amplitud y menor duración que con respecto a aquellos correspondientes a la señal generada con P = 10. Además se presentan picos espurios que provocan un "brinco perceptual" en la pronunciación de las letras "s".
8	27.4s	La calidad perceptual de la voz sintética es semejante a la que se obtiene al utilizar una P = 10. La única diferencia en las gráficas de la señal contra el tiempo discreto, consiste en que la amplitud correspondiente al pico mínimo, en la palabra "voz" es mayor para una P = 8 que para P = 10.
10	27.5s	
12	27.6s	Las diferencias perceptuales entre la voz sintetizada con P = 12 y aquella sintetizada con P = 10, son subjetivas.

En este caso los mejores valores son 8 y 10, como lo recomienda la literatura,

ya que valores por debajo de 8 dan lugar a una degradación de la calidad de la voz sintética. Esta degradación consiste en los "brincos perceptuales" cuando se pronuncia la letra "s" y, un tono más grave de la voz sintética con respecto a la sintetizada con P = 10.

Mientras que valores mayores a 10, no producen una mejora sustancial en la calidad perceptual de la voz sintética, por lo tanto por razones de memoria y complejidad del programa se recomienda una P menor a 12.

El tiempo de procesamiento se incrementó al incrementarse el orden del filtro de síntesis (P). Aunque la magnitud de dicho incremento no es significativa para procesar 56000 muestras de voz (7 segundos de voz muestreados a 8000 Hz), si lo será cuando la cantidad de datos a procesar sea mucho mayor.

IV.4.5 "Variación de la Magnitud del Traslape Entre Frames Sucesivos de Voz."

La literatura recomienda un traslape entre frames sucesivos, para evitar un "brinco perceptual" debido al cambio brusco de parámetros entre un frame y su consecutivo. Este traslape evita dicho cambio brusco de parámetros, mediante la actualización de los mismos cada T ms:

T = Duración del frame - duración del traslape

donde tanto la duración del frame como la duración del traslape están dadas en milisegundos.

Dado que T es menor que la duración del frame, la tasa de transmisión sufre un incremento. Este incremento es inversamente proporcional al tiempo de actualización de los parámetros (T).

La longitud del traslape y por tanto, el aumento de la tasa de transmisión, dependerá del tipo de aplicación. Si el factor más importante es la calidad de la voz sintética, entonces se buscará un traslape que produzca una buena calidad de voz sin importar el aumento de la tasa de transmisión. Por otro lado, si lo que se desea es una disminución de la tasa de transmisión, quizás lo más idóneo sea no efectuar traslape alguno.

En esta sección la prueba que se efectuó consistió en variar el traslape, con el objeto de hallar la longitud en muestras de dicho traslape que produzca la mejor calidad perceptual para la voz sintética. Dado que la literatura recomienda un traslape de un 30% de la longitud del frame, nuestro traslape de referencia es de 86 muestras (ó 10.75 ms), lo cual representa el 33.6% de la longitud del frame utilizado para el análisis LPC. Los resultados de esta prueba se muestran a continuación:

Para la obtención de la tasa de transmisión se tomó en cuenta la siguiente cuantificación: un byte para cada uno de los diez coeficientes del filtro de síntesis,

Traslape	Tasa de transmisión (bits/seg.)	Tiempo	Comentarios
11 (1.4ms)	3167.35	22.8s	Aquí en la gráfica de la señal contra el tiempo discreto se presentan picos negativos, de gran amplitud y corta duración en la palabras "es" y "tipica", lo cuál provoca un "brinco perceptual" al pronunciar la letra "s". Las diferencias perceptuales con respecto a la señal de referencia, generada con un tralape de 86 muestras, son subjetivas.
64 (8.0ms)	4041.67	25.8s	Aquí también como en el caso anterior, se presentan picos en las palabras: "es", "señal" y "típica", lo cuál provoca problemas al pronunciar las letras "s". La calidad perceptual de la voz sintética es más ronca que con respecto a la voz sintética de referencia.
86 (10.75ms)	4564.71	28.5s	Voz Sintética de referencia.
128 (16.0ms)	6062.50	32.6s	En la gráfica de la señal contra el tiempo discreto, las amplitudes correspondientes a los picos negativos son mayores que las correspondientes a la señal de referencia. Así mismo se presentan picos indeseables que provocan "brincos perptuales al pronunciar las letras "s". La voz sintética tiene un tono un poco más grave con respecto a la señal de referencia.
192 (24.0ms)	12125.00	52.5s	La calidad de la voz sintética definitivamente es inferior con respecto a la señal de referencia, debido a la presencia de "brincos perceptuales" en las letras "s" y, a que la voz tiene un tono más grave

un bit para la desición S/NS, un byte para el valor del pitch y, un byte para la energía de la señal de voz real.

En conclusión el mejor valor para el traslape es 86 muestras (10.75 ms), ya que para valores muy grandes del traslape la degradación de la calidad perceptual de la voz sintética es bastante notable; mientras que para traslapes muy pequeños, la calidad de voz sólo sufre problemas con la pronunciación de las letras "s". Por lo tanto, tomando en cuenta el tiempo de procesamiento y la tasa de transmisión, los traslapes que recomiendo deben ser menores o iguales a un tercio de la longitud en muestras del frame para el análisis LPC.

Como resulta lógico, el tiempo de procesamiento aumentó al aumentar el traslape. Sin embargo, la calidad de voz no se incremento, por lo que no se recomiendan traslapes mayores a la mitad de muestras del frame de análisis LPC.

Por último haré mención a la siguiente prueba efectuada sobre el algoritmo: se decidió eliminar el traslape en favor de una reducción del tamaño del frame para el análisis LPC. Esto es el tamaño del frame se reduciría a 80 muestras (10 ms), pero no se efectuaría el traslape entre frames sucesivos.

Es necesario mencionar que el tamaño del frame para la estimación del pitch no varió su tamaño en muestras, es decir, se mantuvo constante. De esta manera al efectuar la síntesis para un frame de 256, los valores de los coeficientes de reflexión cambiarán cada 80 muestras, mientras que el pitch sería el mismo para las 256 muestras. El resto de los parámetros permanecen como se listaron en la sección IV.4.1. Los resultados de esta prueba se comentan a continuación:

1) La voz sintética resultante tiene algunos silencios menos ruidosos con respecto a la señal sintetizada con traslapes del frame de análisis LPC. Sin embargo, la mayoría de los silencios resultaron tan ruidosos como los de la señal sintetizada con traslapes.

2) Las voces resultaron de una calidad ligeramente inferior, ya que presenta una especie de reverberancia con respecto a la señal sintétizada con traslapes.

3) A pesar de las diferencias, se puede decir que perceptualmente hablando es un resultado similar al que se obtiene al traslapar los frames consecutivos de voz.

4) La tasa de transmisión es igual a 9700 bits/seg y, el tiempo de procesamiento es de 39.4s. Por tanto, dados estos factores y la similitud de resultados entre efectuar o no, él traslape, se decidió optar por la opción de traslapar los frames consecutivos de voz.

IV.4.6 "Presentación de las Curvas."

Los valores de los parámetros utilizados por el algoritmo de síntesis se listan en la sección IV.4.1. A continuación se muestran las gráficas de la señal real de voz contra el tiempo discreto y, la de la señal sintética de voz contra el tiempo discreto:

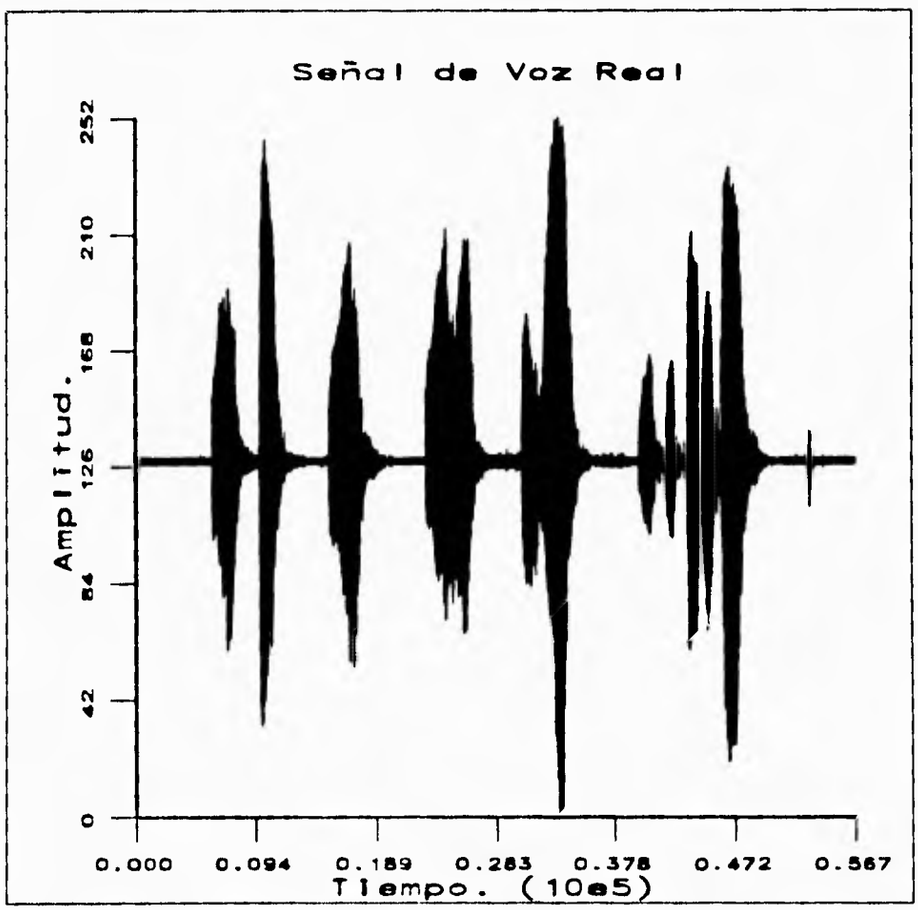


Figura 4.1: Gráfica de la señal de voz real contra el tiempo discreto.

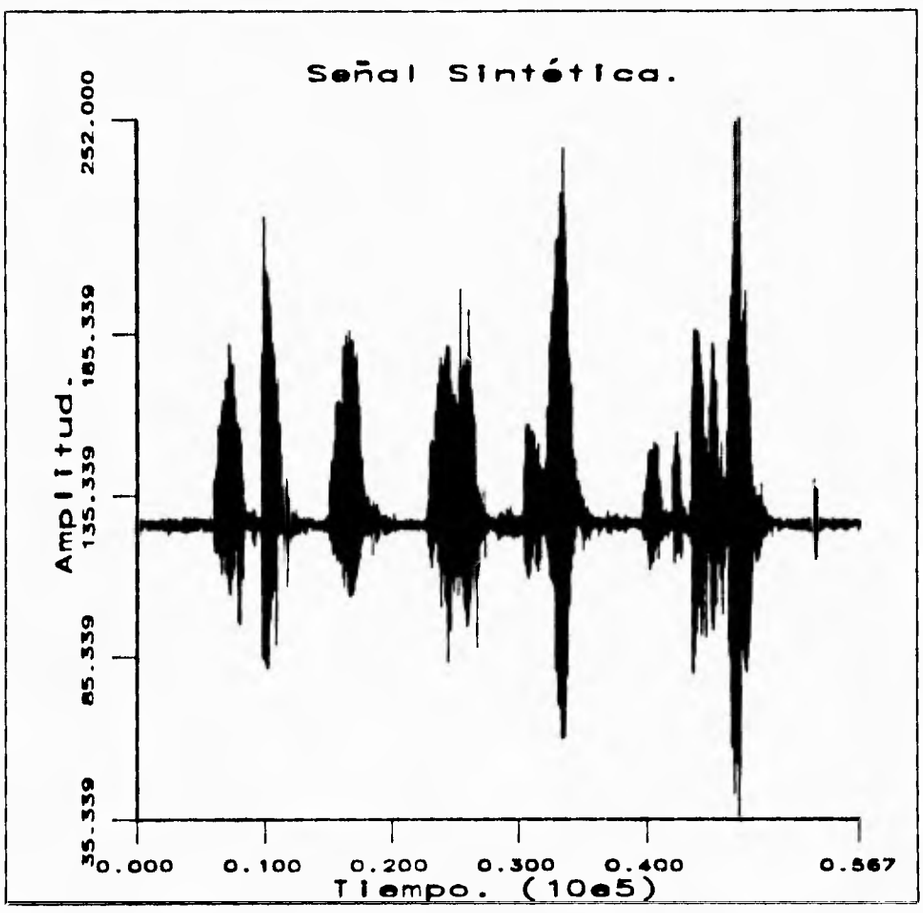


Figura 4.2: Gráfica de la señal de voz sintética contra el tiempo discreto.

A continuación se muestran: el espectro de la señal real de voz y, la estimación paramétrica del mismo (estimado de Levinson).

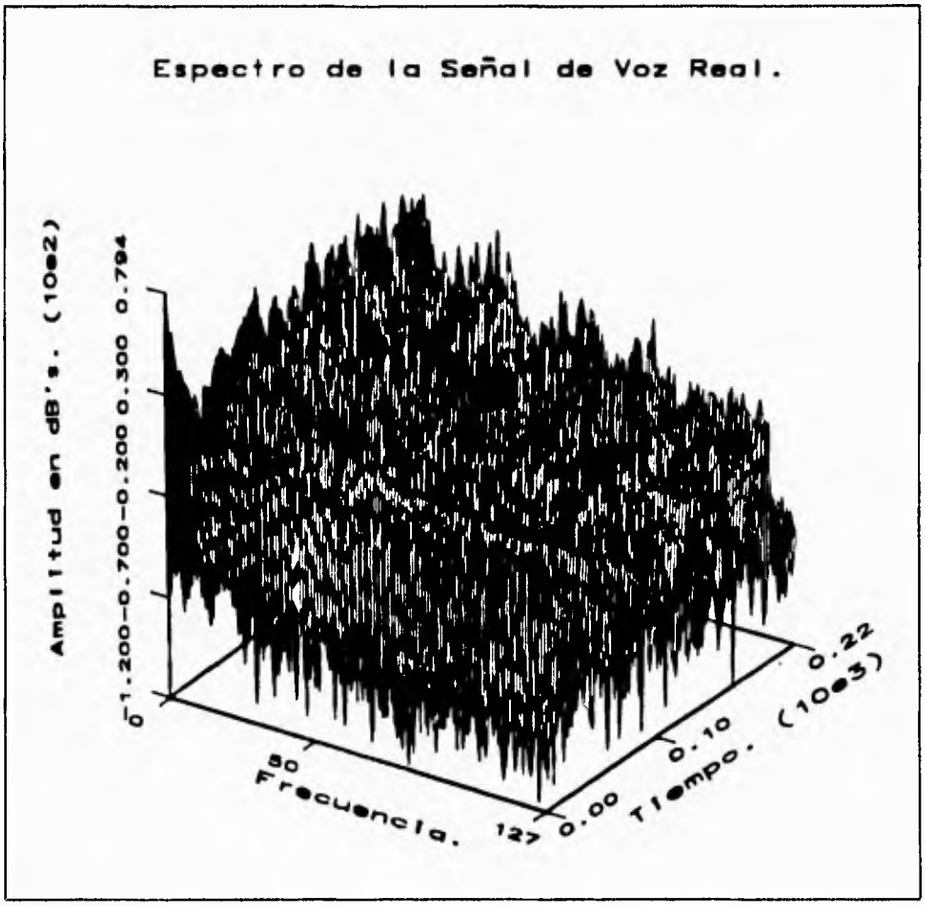


Figura 4.3: Espectro de la señal de voz real.

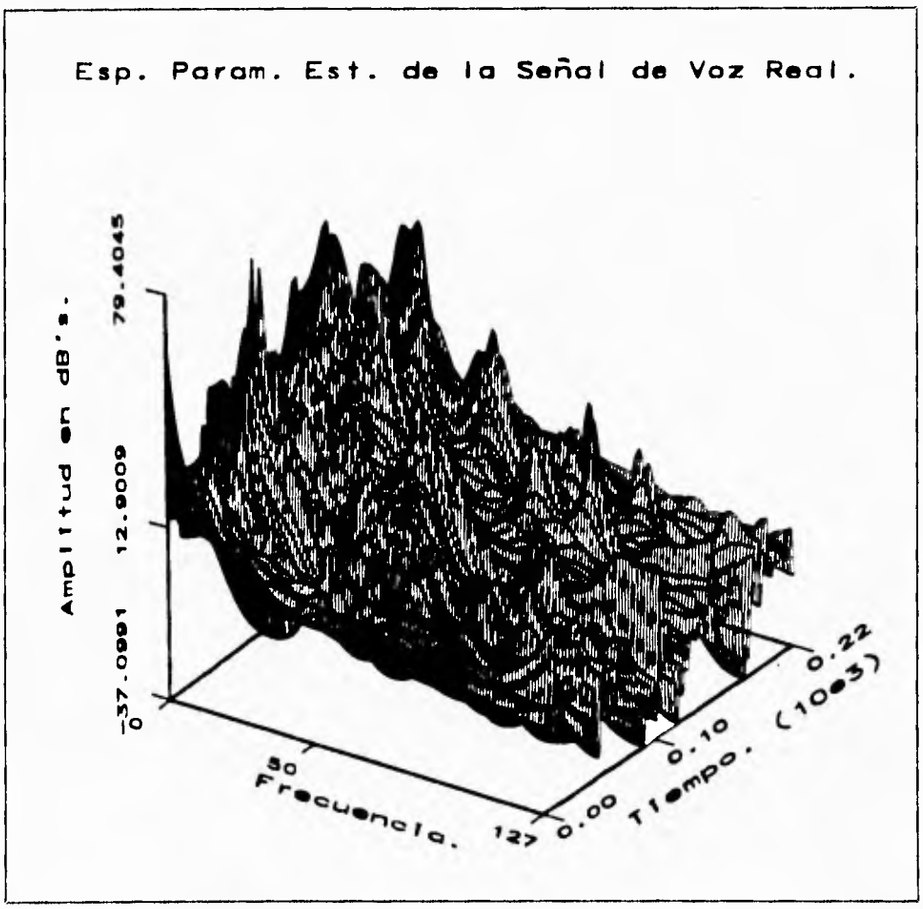


Figura 4.4: Espectro paramétrico estimado de la señal de voz real.

A continuación se muestra el espectro de la señal sintética de voz:

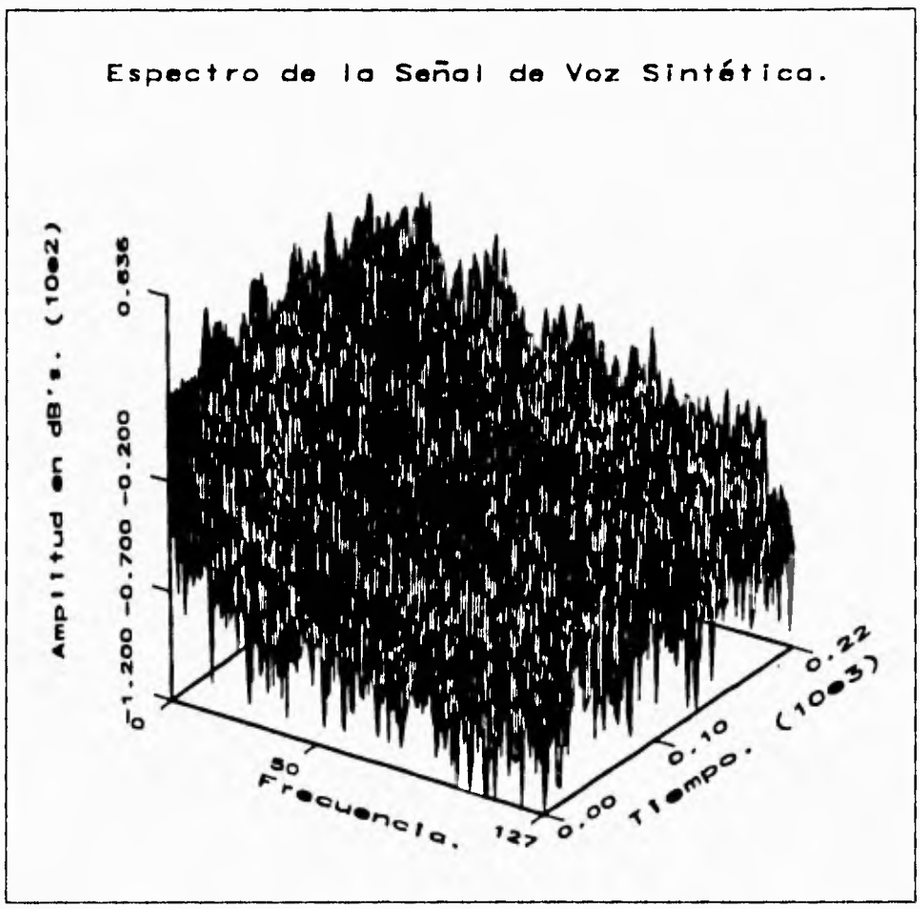


Figura 4.5: Espectro de la señal de voz sintética.

Una comparación entre los espectros FFT de la señal real de voz (fig. 4.3), y de la señal sinética de voz (fig. 4.5) no es muy conveniente debido a que ambas gráficas son tridimensionales, lo que conduce a que la posición y amplitud de los picos más notables de ambas gráficas no se puedan fijar con absoluta presición y certeza; y por tanto la comparación de dichos espectros es más cualitativa que cuantitativa. Por tanto, con el objeto de facilitar la percepción visual se ha optado por presentar el espectro paramétrico estimado de la señal de voz real (estimado de Levinson), para su comparación con el espectro FFT de la señal de voz real. Esta comparación visual es más sencilla, ya que el espetro paramétrico estimado de la señal de voz real es una superficie suave, es decir sin cambios abruptos de pendiente (picos).

Por otro lado, debido a que una comparación entre el espectro FFT de la señal real de voz y el espectro FFT de la señal sintética de voz, no es determinante sobre la calidad perceptual de la señal sintética de voz; por lo que de aquí en adelante se ha optado por presentar únicamente las gráficas de las señales sintéticas de voz contra el tiempo discreto.

Al hacer una comparación perceptual de la señal de voz real con su respectiva sintética sobresale el hecho de que las zonas de silencio son mucho má ruidosas, mientrás que para la voz se tiene tanto una calidad perceptual como inteligibilidad aceptables.

Se intentó eliminar dicho ruido mediante el diseño de varios filtros pasobanda de tipo FIR. Sin embargo no se eliminó dicho ruido, lo cuál se debe a las siguientes razones:

1) El ruido se debe a que el modelo LPC, no es bueno para modelar las regoines de voz que tienen una pequeña energía espectral, como lo son los silencios [PIC93].

2) El hecho de segmentar la señal de voz en segmentos de una duración constante, constituye un problema para el algoritmo. Esto se debe a que un segmento de voz puede contener distintas regiones, las cuales, pueden ser sonoras, no sonoras, silencios ó, incluso puede resultar que su clasificación dentro de una de las categorias anteriores, no sea tan obvia.

3)Dado que el algoritmo clasifica los segmentos de voz como sonoros y no sonoros, éste clasificará un segmento con distintas regiones de voz como sonoro o no sonoro, lo cuál, con lleva cierto error. Este indudablemente repercute en la calidad perceptual de la señal sintética de voz.

Por tanto, se llegó a la conclusión de que la calidad perceptual de las señales sintéticas de voz, obtenidas mediante este algoritmo, pueden ser substancialmente mejoradas si se efectúa una segmentación inteligente de la señal de voz. Por segmentación inteligente entendemos la detección de cambios bruscos en la señal, así como la detección de principio y fin de palabra y, la detección de silencios. Mediante dicha segmentación el modelo LPC modelará de una manera más eficiente los segmentos de voz, elevando por tanto la calidad perceptual de la señal sintética de voz.

IV.5 "Evaluación del Método de Síntesis:

LPC con SIFT para la Extracción del Pitch."

IV.5.1 "Introducción."

Este método se explicó en detalle en las secciones III.2 y III.3.5, aquí sólo se efectuará una evaluación perceptual de las señales sintéticas generadas por el algoritmo, al variar los parámetros de este último.

A continuación se describen los detalles de la implementación del algoritmo:
1) El método se programó en leguaje C y, se utilizó una estación de trabajo para el desarollo de las simulaciones. Esto es importante, ya que la velocidad de las estaciones de trabajo hace posible el análisis y síntesis de varios segundos de voz, lo cuál de otra manera sería una tarea muy difícil de ejecutar, ya que el procesamiento podría tardar horas.
2) Como ya se mencionó anteriormente, la obtención de datos reales para la evaluación del algoritmo, se efectuó mediante la tarjeta y/o software comercial "Sound Blaster". La frecuencia de muestreo de esta tarjeta se ajusto a 8000 Hz y, se digitalizó la señal de voz a 8 bits (256 niveles).
3)La longitud de frame en muestras para el análisis LPC y extracción de Pitch fué de 256 muestras, o lo que es equivalente a 32 ms.
4) El frame para el análisis LPC es filtrado a través de un filtro paso bajas (f_c = 800 Hz.) [MARK72]. Esto implica, según el teorema de Nyquist, que la frecuencia máxima es aproximadamente 1600 Hz, por lo que, en este punto la señal esta sobremuestreada. Con el objeto de reducir el tiempo de procesamiento, se efectúa un submestreo de 4:1, es decir, por cada cuatro datos que componen el frame se toma sólo uno, para formar el nuevo de frame, a partir del cuál se estimará el pitch.
5) Al frame para el análisis LPC se le aplicó una ventana de Hamming.
6)En muchos algoritmos para el procesamiento digital de señales de voz se recomienda un traslape entre frames sucesivos, con el fin de evitar una especie de "brinco perceptual" debido al cambio brusco de parámetros entre un frame y su cosecutivo. Este traslape se escogió de una longitud de 86 muestras, o lo que es equivalente a 10.75 ms.
7) El algoritmo de SIFT que se utiliza para determinar si el frame en cuestión es sonoro ó no sonoro, tiene el siguiente parámetro importante: el umbral sonoro - no sonoro (S/NS), el cúal como su nombre lo indica sirve para determinar si el frame en cuestión es sonoro o no sonoro. En este caso se utilizó el valor recomendado por [MARK72], es decir, S/NS = 0.4.
8) Dentro del método de SIFT, se necesita hallar el pico máximo de las autocorrelaciones dentro de cierto intervalo. El intervalo de búsqueda para el pico máximo de la función de autocorrelación se escogió de 6 a 50, lo cual correponde al intervalo de

frecuencias para el pitch: [40, 333.33] Hz.
9)Además de hallar el pico máximo de la función de autocorrelación, es necesario interpolar esta función alrededor del desplazamiento m, correpondiente a dicho valor máximo con el objeto de obtener una estimación del pitch con una resolución adecuada [MARK72]. Esta interpolación se efectuó de acuerdo con [MARK72].
10) Al interpolar se obtiene un pico, el cuál se compara con el umbral S/NS, con el objeto de determinar si el frame en cuestión es sonoro o no sonoro. El valor de dicho umbral, de acuerdo con [MARK72], varía de 0.378 a 4. Aquí se escogió un umbral S/NS de 0.4.
11) Al declarar un frame como sonoro o no sonoro, también se toma en cuenta, tanto el frame anterior como el posterior. Esto es, si un frame es declarado como sonoro pero se encuentra entre dos frames no sonoros, entonces el frame se redeclara como no sonoro; y viceversa.
12) El modelo LPC que se utilizó fué de décimo orden.
13) No se detectaron silencios.

Dado que el método de síntesis, a evaluar en esta sección, tan sólo difiere del anetrior (sección IV.4), en el método de estimación del pitch; no es necesario repetir las pruebas relacionadas con la codificación lineal predictiva. Por tanto, sólo se variará el umbral para la decisión sonoro - no sonoro.

IV.5.2 "Variaciones del Umbral Sonoro - No Sonoro (S/NS)."

En el algoritmo de autocorrelación con recorte central (SIFT), uno de los parámetros más importantes es el umbral sonoro - no sonoro, el cuál sirve para determinar si el frame en cuestión es sonoro ó no sonoro [MARK72]. Los resultados de las evaluaciones que a continuación se presentan, consisten en la variación de dicho umbral S/NS, mientras que el resto de parámetros permanecen sin variación alguna respecto de los valores recomendados por [MARK72], tal como se listaron en IV.5.1.

Dada la experiencia obtenida en IV.4.3, la primera prueba que realicé fué variar S/NS de: 0.2 a 0.6. Los resultados de esta prueba se muestran a continuación:

S/NS	Tiempo	Comentarios
0.2	27.4s	Aquí la gráfica de la señal contra el tiempo discreto difiere de aquella para un S/NS de 0.4, en que los picos negativos tienen mayor amplitud y, que en la palabra señal se presenta un pico indeseable. Este provoca problemas al pronunciar la letra "s". La voz sintética presenta cierta reverberancia con respecto a la voz sintética generada con un S/NS de 0.4
0.3	29.0s	Resultado prácticamente igual al obtenido con un umbral S/NS de 0.4
0.4	30.0s	Señal sintética de referencia.
0.5	30.8s	Aquí la gráfica de la señal contra el tiempo discreto difiere de aquella para un S/NS de 0.4, en que los picos tienen mayor amplitud y, en el surguimiento de picos indeseables de gran amplitud y corta duración que provocan problemas al pronunciar las letras "s". La voz sintética presenta cierta reverberancia con respecto a la voz sintética generada con un S/NS de 0.4
0.6	31.6s	Además de los problemas presentados en el caso anterior, la voz sintética se degrada demasiado con respecto a nuestra señal de referencia porque presenta un tono demasiado grave y reverberancia.

Como conclusión de la prueba anterior, se puede decir que el mejor valor para el umbral S/NS está entre 0.3 y 0.4. Sin embargo, para precisar estos límites con una mayor exactitud, se realizó la siguiente prueba:

S/NS	Tiempo	Comentarios
0.25	28.5s	Existe cierta reverberancia de la señal sintética con respecto a nuestra señal de referencia. La calidad perceptual es mejor con un umbral S/NS de 0.4
0.45	30.5s	La calidad perceptual de la señal sintética es un poco inferior con respecto de la señal de referencia, debido a la existencia de cierta reverberancia.

La conclusión de estas pruebas es que el mejor valor para el umbral S/NS está entre 0.3 y 0.4. El tiempo de ejecución del programa aumenta al aumentar el valor del umbral sonoro - no sonoro. Esto se debe a que al aumentar dicho umbral el porcentaje de frames no sonoros aumenta, lo cuál incrementa el tiempo de ejecución del programa dado que para un frame no sonoro, además de la ejecución de la subrutina de detección del pitch es necesario generar ruido blanco. Dicha generación de ruido blanco no es necesaria en el caso de que el frame halla sido declarado como sonoro, por tanto un frame sonoro se sintetiza mucho más rápido que un frame no sonoro.

IV.5.3 "Presentación de las Curvas."

Los valores de los parámetros utilizados por el algoritmo de síntesis se listan en la sección IV.5.1. A continuación se muestra la gráfica de la señal sintética de voz contra el tiempo discreto, la cual se puede compara con la señal real de voz de la figura 4.1:

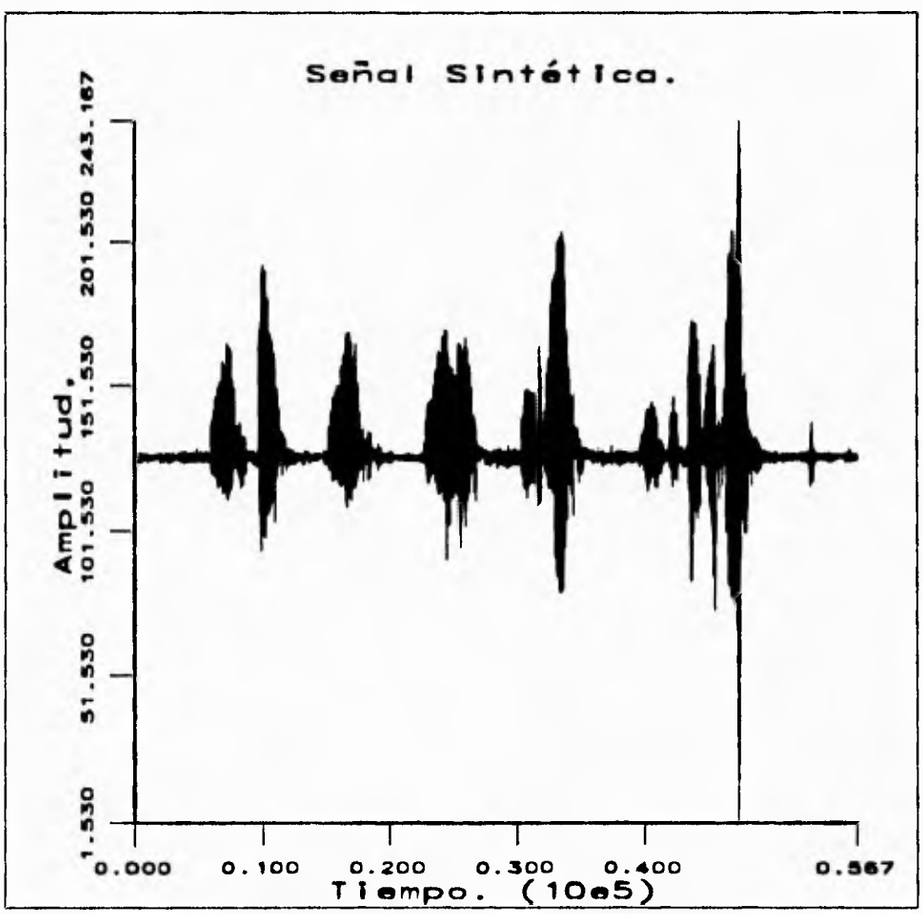


Figura 4.6: Gráfica de la señal de voz sintética contra el tiempo discreto.

Al hacer una comparación perceptual de la señal de voz real con su respectiva sintética sobresale el hecho de que las zonas de silencio son mucho más ruidosas, mientrás que para la voz se tiene tanto una calidad perceptual como inteligibilidad aceptables. El resultado y las conclusiones de la sección IV.4.6, se repiten en este caso. No existe una diferencia perceptual significativa entre el método de AUTOC y el del SIFT; sólo que el tiempo de procesamiento es menor para el primero.

IV.6 "Evaluación del Método de Síntesis:

LPC con un Método Cepstral para la Extracción del Pitch."

IV.6.1 "Introducción."

Este método se explicó en detalle en las secciones III.2 y III.3.4, aquí sólo se efectuará una evaluación perceptual de las señales sintéticas generadas por el algoritmo, al variar los parámetros de este último.

A continuación se describen los detalles de la implementación del algoritmo:

1) El método se programó en leguaje C y, se utilizó una estación de trabajo para el desarollo de las simulaciones. Esto es importante, ya que la velocidad de las estaciones de trabajo hace posible el análisis y síntesis de varios segundos de voz, lo cuál de otra manera sería una tarea muy díficil de ejecutar, ya que el procesamiento podría tardar horas.

2) Como ya se mencionó anteriormente, la obtención de datos reales para la evaluación del algoritmo, se efectuó mediante la tarjeta y software comercial "Sound Blaster". La frecuencia de muestreo de esta tarjeta se ajusto a 8000 Hz y, se digitalizó la señal de voz a 8 bits (256 niveles).

3)La longitud de frame en muestras para el análisis LPC fué de 256 muestras (32 ms) y, la longitud del frame para extracción de Pitch fué de 512 muestras (64 ms). Esta diferencia se debe a que el método cepstral para la extracción del pitch requiere de un intervalo de mayor duracción, el cuál incluya varios periodos fundamentales, para una estimación confiable del pitch [PAPA].

4) Tanto al frame para el análisis LPC como al frame para la extracción del pitch, se les aplicó una ventana de Hamming.

5)En muchos algoritmos para el procesamiento digital de señales de voz se recomienda un traslape entre frames sucesivos, con el fin de evitar una especie de "brinco perceptual" debido al cambio brusco de parámetros entre un frame y su cosecutivo. Este traslape se escogio de una longitud de 86 muestras, o lo que es equivalente a 10.75 ms.

6) El método cepstral que se utiliza para determinar si el frame en cuestión es sonoro ó no sonoro, tiene el siguiente parámetro importante: el umbral sonoro - no sonoro (S/NS), el cúal como su nombre lo indica sirve para determinar si el frame en cuestión es sonoro o no sonoro. En este caso se utilizó el valor de 0.2 para dicho umbral.

7) Dentro del método cepstral, se necesita hallar el pico máximo del cepstrum del frame en procesamiento, dentro de cierto intervalo. El intervalo de búsqueda para el pico máximo del cepstrum se escogio de 1 a 15 [ms], lo cual correponde al intervalo de frecuencias para el pitch: [66.67, 1000] Hz.

8) El pico máximo del cepstrum se compara con el umbral S/NS, con el objeto de determinar si el frame en cuestión es sonoro o no sonoro. El valor de dicho umbral es de 0.2.
9) Al declarar un frame como sonoro o no sonoro, también se toma en cuenta, tanto el frame anterior como el posterior. Esto es, si un frame es declarado como sonoro pero se encuentra entre dos frames no sonoros, entonces el frame se redeclara como no sonoro; y viceversa.
10) Se evita el llamado doblamiento de pitch mediante el algoritmo descrito en [NOLL67].
11) El modelo LPC que se utilizó fué de décimo orden.
12) No se detectaron silencios.

Dado que el método de síntesis, a evaluar en esta sección, tan sólo difiere de los dos anetriores (secciones IV.4 y IV.6), en el método de estimación del pitch; no es necesario repetir las pruebas relacionadas con la codificación lineal predictiva. Por tanto, sólo se variará el umbral para la decisión sonoro - no sonoro.

IV.6.2 "Variaciones del Umbral Sonoro - No Sonoro (S/NS)."

En el método cepstral para la extracción del pitch, uno de los parámetros más importantes es el umbral sonoro - no sonoro, el cuál sirve para determinar si el frame en cuestión es sonoro ó no sonoro [NOLL67]. Los resultados de las evaluaciones que a continuación se presentan, consisten en la variación de dicho umbral S/NS, mientrás que el resto de parámetros permanecen sin variación alguna respecto de los valores recomendados por [NOLL67], tal como se listaron en IV.8.1.

Dada la experiencia obtenida en IV.4.3, la primera prueba que realizé fué variar S/NS de: 0.1 a 0.6. Los resultados de esta prueba se muestran a continuación:

S/NS	Tiempo	Comentarios
0.1	93.4s	Aquí la gráfica de la señal contra el tiempo discreto presenta mayores amplitudes que aquella para S/NS de 0.2. Existen dificultades al pronunciar la vocal "u". La calidad perceptual de la señal sintética es aceptable a pesar de cierta reverberancia con respecto a la señal de referencia.
0.2	94.3s	Señal sintética de referencia.
0.3	95.3s	La calidad perceptual de la señal sintética es inferior con respecto a la calidad de la señal señal de referencia, ya que la primera tiene un tono más grave.
0.4	95.7s	La calidad perceptual de la señal sintética es ligeramente inferior con respecto a la calidad de la señal de referencia, ya que la primera tiene un tono ligeramente más grave.
0.5	95.6	Esta señal sintética definitivamente presenta un tono mucho más grave que la señal de referencia.
0.6	95.8	Resultado similar al anterior.

Como conclusión de la prueba anterior, se puede decir que el mejor valor para el umbral S/NS está alrededor de 0.2. Sin embargo, para precisar los límites, para el umbral sonoro - no sonoro con una mayor excactitud, se realizó la siguiente prueba:

S/NS	Tiempo	Comentarios
0.15	94.5s	Las diferencias perceptuales con respecto a la señal de referencia son subjetivas.
0.25	94.8s	La voz sintética tiene un tono ligeramente más grave con respecto a la señal de referencia.

La conclusión de estas pruebas es que el mejor valor para el umbral S/NS está entre 0.15 y 0.2. El tiempo de ejecución del programa aumenta al aumentar el valor del umbral sonoro - no sonoro. Esto se debe a que al aumentar dicho umbral el porcentaje de frames no sonoros aumenta, lo cuál incrementa el tiempo de ejecución del programa dado que para un frame no sonoro, además de la ejecución de la subrutina de detección del pitch es necesario generar ruido blanco. Dicha generación de ruido blanco no es necesaria en el caso de que el frame halla sido declarado como sonoro, por tanto un frame sonoro se sintetiza mucho más rápido que un frame no sonoro.

IV.6.3 "Presentación de las Curvas."

Los valores de los parámetros utilizados por el algoritmo de síntesis se listan en la sección IV.6.1. A continuación se muestra la gráfica de la señal sintética de voz contra el tiempo discreto, la cual se puede compara con la señal real de voz de la figura 4.1:

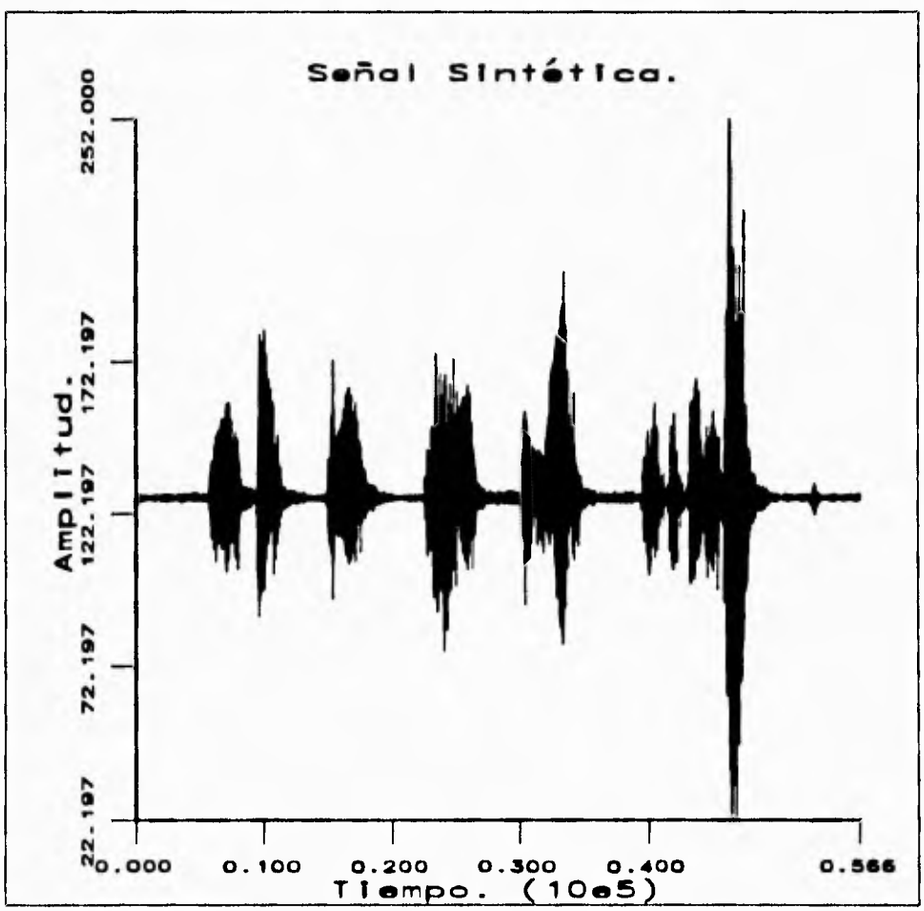


Figura 4.7: Gráfica de la señal de voz sintética contra el tiempo discreto.

Al hacer una comparación perceptual de la señal de voz real con su respectiva sintética sobresale el hecho de que las zonas de silencio son mucho más ruidosas, mientrás que para la voz se tiene tanto una calidad perceptual como inteligibilidad aceptables. El resultado y las conclusiones de la sección IV.4.6, se repiten en este caso. No existe una diferencia perceptual significativa entre los metodos de AUTOC, el del SIFT y, el método cepstral para la extracción del pitch; sólo que el tiempo de procesamiento que es menor para el primer caso y, mayor para el método cepstral.

IV.7 "Evaluación del Método de Síntesis de Voz Excitación Multipulso."

IV.7.1 "Introducción."

Este método se explicó en detalle en las secciones III.2 y III.4.2, aquí sólo se efectuará una evaluación perceptual de las señales sintéticas generadas por el algoritmo, al variar los parámetros de este último.

A continuación se describen los detalles de la implementación del algoritmo:

1) El método se programó en leguaje C y, se utilizó una estación de trabajo para el desarollo de las simulaciones. Esto es importante, ya que la velocidad de las estaciones de trabajo hace posible el análisis y síntesis de varios segundos de voz, lo cuál de otra manera sería una tarea muy difícil de ejecutar, ya que el procesamiento podría tardar horas.

2) Como ya se mencionó anteriormente, la obtención de datos reales para la evaluación del algoritmo, se efectuó mediante la tarjeta y software comercial "Sound Blaster". La frecuencia de muestreo de esta tarjeta se ajusto a 8000 Hz y, se digitalizó la señal de voz a 8 bits (256 niveles).

3) La longitud en muestras del frame para el análisis LPC fué de 80 muestras (10 ms).

4) Una de las ventajas del método de excitación multipulso sobre los métodos de síntesis de voz que hacen uso de un algoritmo para la estimación del pitch, es la eliminación de la necesidad de clasificar un frame de voz como sonoro o no sonoro, para determinar el tipo de excitación necesaria para sintetizar dicho frame de voz. En el método de excitación multipulso, la excitación consiste en una serie de pulsos cuyas amplitudes y posiciones, se determinan mediante algún criterio para la minimización del error cuadrático medio y, que tome en cuenta la percepción auditiva humana. En este caso se obtienen 5 pulsos por cada 10ms (80 muestras) de voz.

5) Al frame para el análisis LPC se le aplicó una ventana de Hamming.

6) Dentro de cualquier método de síntesis de voz es necesario tomar en cuenta las características de la percepción auditiva humana y, el método de excitación multipulso no es la excepción.

En este método es necesario filtrar la señal residual a través de un filtro de ponderación perceptual con el objeto de evitar un grave error debido a la omisión de las características perceptuales del sistema auditivo humano. Este filtro de ponderación perceptual está dado por:

$$W(z) = \frac{H(bz)}{H(z)}$$

donde $H(z)$ es el filtro de síntesis y, $H(bz)$ está dada por:

$$H(bz) = \frac{G}{[1 + a_1 b^{-1} z^{-1} + \cdots + a_p b^{-p} z^{-p}]}$$

Esta ecuación se obtiene de sustituir z por bz en la ecuación del filtro de síntesis. b es un parámetro que varía entre cero y uno. Las pruebas que se efectuarán consisten en la variación de este parámetro alrededor del valor recomendado por [PAPA], es decir, b = 0.8.

Dado que se sigue utilizando una codificación lineal predictiva (LPC), al igual que en los métodos anteriores, no se realizarán pruebas relacionadas con dicha codificación, ya que los resultados obtenidos serían similares y repetitivos. Por tanto las pruebas se realizarán sobre la variación de los parámetros relativos al método de excitación multipulso. Estos parámetros son el número de pulsos por determinada cantidad de tiempo y, la variación del parámetro "b" del filtro de ponderación perceptual.

IV.7.2 "Varaiación del Número de Pulsos."

El número de pulsos por frame es un parámetro de suma importancia, ya que se relaciona directamente con la tasa de transmisión. Por tanto, el número de pulsos por frame es un parámetro crítico para el diseño, desarollo e implementación de cualquier sistema de síntesis de señales de voz que utilice algún método de excitación multipulso.

La tasa de transmisión es directamente proporcional al número de pulsos que se utilizen para representar un frame de voz. Sin embargo, el número de pulsos por frame no es un parámetro que se pueda establecer de ante mano, sin tomar en cuenta las condiciones y restricciones impuestas por la aplicación, sino que está determinada por la relación que existe entre la calidad perceptual y, el tiempo de procesamiento así como de la cantidad de memoria disponible en el sistema de procesamiento. De esta manera mientras mayor sea el número de pulsos que se utilicen para sintetizar un frame, mayor será la calidad de la voz sintética pero también, mayor serán los requerimientos de memoria y, el tiempo de procesamiento.

La prueba que se realizó consistió en variar el número de pulsos por frame de 5 a 2. Cabe señalar que dado que el programa se lleva aproximadamente 2 hr. en procesar siete segundos de voz, se optó por utilizar solamente una porción de la señal de prueba (sección IV.3), de 10996 muestras de longitud. Esta porción de la señal de prueba corresponde a la palabra "señal". Los resultados son los siguientes:

Número de Pulsos por Frame	Tiempo	Comentarios
5	23.3m	Las difrencias perceptuales entre las distintas señales sintéticas son subjetivas. Sin embargo al disminuir el número de pulsos por frame la razón SNR disminuye ligeramente sin llegar a degradar la calidad perceptual de una manera significativa.
4	18.8m	
3	14.1m	
2	9.5m	
-	-	
-	-	
-	-	

IV.7.3 "Variaciones del Parámetro b."

La prueba que aquí se presenta, consiste en la variación del parámetro "b". Este parámetro está relacionado con la ponderación perceptual de la señal residual (error). De esta manera "b" indica el grado de filtrado perceptual para la señal de error.

El paramétro b, se varió de 0.3 a 0.9 pasando por el valor recomendado por la literatura [PAPA], que es de 0.8. Los resultados se muestran en la siguiente tabla:

b	Tiempo	Comentarios
0.3	23.3m	Las difrencias perceptuales entre las distintas señales sintéticas generadas al variar el parámetro "b" son de caracter subjetivo.
0.4	23.3m	
0.5	21.2m	
0.6	23.3m	
0.7	23.1m	
0.8	23.3m	
0.9	23.3m	

IV.7.4 "Presentación de las Curvas."

Los valores de los parámetros utilizados por el algoritmo de síntesis se listan en la sección IV.7.1. Las gráficas que a continuación se presentan, son la señal sintética de voz y, el espectro de la misma. Cabe señalar que para la obtención de estos resultados, se consideró completa la señal de prueba (sección IV.3). Por lo que estos resultados se pueden comparar con las figuras 4.1 y 4.3 respectivamente.

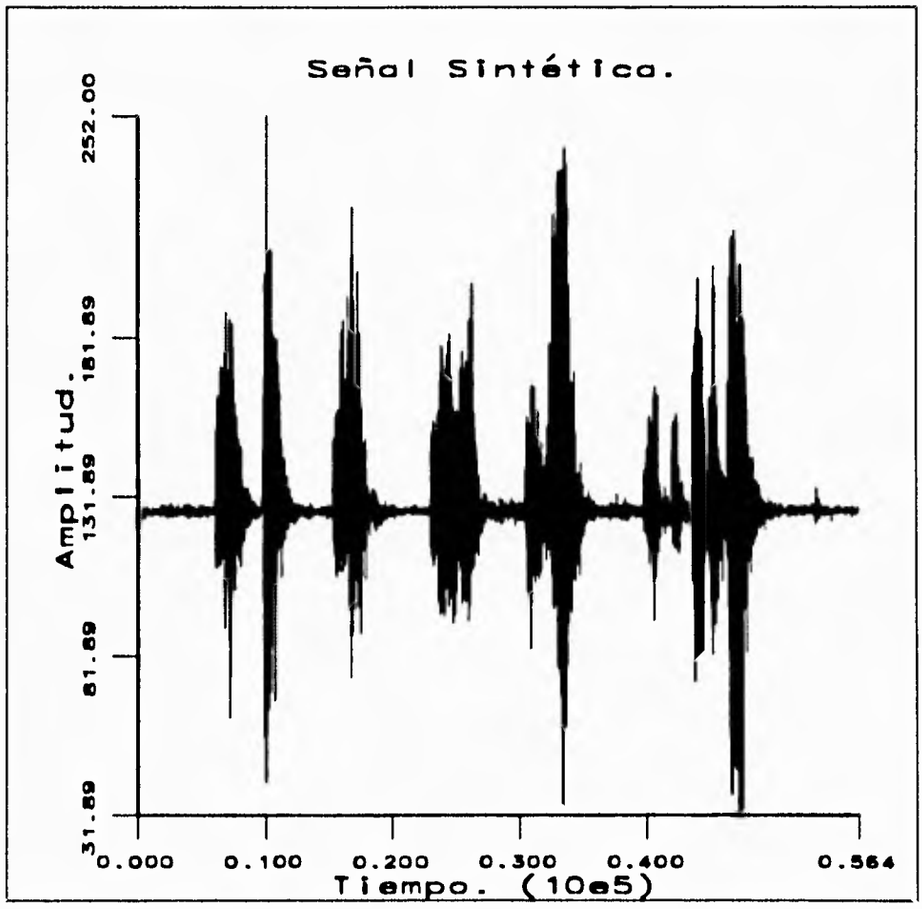


Figura 4.8: Gráfica de la señal de voz sintética contra el tiempo discreto.

Al comparar la señal de voz sintética con la señal de voz real, se observa el mismo problema que existe con los métodos anteriores (secciones IV.4.6, IV.5.3 y, IV.6.3), es decir, en las zonas de silencios de la señal sintética la potencia del ruido es mucho mayor que la potencia del ruido para las zonas de silencio de la señal real de voz. Nuevamente, una segmentación que tome en cuenta los cambios bruscos en la señal real de voz, así como los silencios mejorará la calidad perceptual de las señales sintéticas de voz.

El método de excitación multipulso, no presenta un mejor resultado con respecto de los resultados obtenidos con los métodos anteriores (secciones IV.4.6, IV.5.3 y, IV.6.3), como era de esperarse. Incluso el resultado obtenido con el método LPC con el algoritmo AUTOC para la estimación presnta un mejor resultado que el método multipulso, ya que la razón SNR es más grande para el primer método que para el método de excitación multipulso.

Al comparar la calidad perceptual de la señal sintética generada con el método de excitación multipulso con las señales generadas con el método LPC en combinación con algún algoritmo para la estimación del pitch, no se observan diferencias perceptuales significativas. A pesar de la similitud de los resultados, el tiempo de procesamiento es muy diferente ya que el algoritmo de excitación multipulso tarda 2 hrs. en sintetizar la señal de prueba (sección IV.3), cuya duración es de 7 seg. Mientrás que el tiempo de procesamiento para la señal de prueba es de 27.1s para el método LPC con el algoritmo de AUTOC, de 30.0s para el método LPC con el algoritmo de SIFT para la estimación y de 94.3s para el método cepstral. Por lo tanto dado que el tiempo de procesamiento para el método de excitación multipulso es mucho mayor que el tiempo de procesamiento para los algoritmos de síntesis que utilizan estimadores del pitch, no se recomienda su utilización.

IV.8 "Resultados Obtenidos del Método de Síntesis: Regular Pulse Excitation (RPE)".

IV.8.1 "Introducción."

Este método se explicó en detalle en las secciones III.2 y III.4.3. Los detalles de la implementación se describen a continuación:
1) El método se programó en leguaje C y, se utilizó una estación de trabajo para el desarollo de las simulaciones. Esto es importante, ya que la velocidad de las estaciones de trabajo hace posible el análisis y síntesis de varios segundos de voz, lo cuál de otra manera sería una tarea muy difícil de ejecutar, ya que el procesamiento podría tardar horas.
2) Como ya se mencionó anteriormente, la obtención de datos reales para la evaluación del algoritmo, se efectuó mediante la tarjeta y software comercial "Sound Blaster". La frecuencia de muestreo de esta tarjeta se ajusto a 8000 Hz y, se digitalizó la señal de voz a 8 bits (256 niveles).
3) La longitud en muestras del frame para el análisis LPC fué de 256 muestras (32ms).
4) De acuerdo con [KROON86], se obtienen 10 pulsos para cada 40 muestras (5ms), con un espaciamiento uniforme entre los pulsos de 4 muestras.
5) Se implementó el algoritmo simplifiado del método RPE (Sección III.4.3), ya que de esta manera se evita la inversión de matrices, la cuál de llevarse a cabo produciría un aumento del tiempo de procesamiento y, aumentaría la complejidad del programa.
6) Se utilizó el siguiente filtro de ponderación perceptual:

$$\frac{1}{A(z/\gamma)} = \frac{1}{1 + \sum_{k=1}^{10} a_k b^k z^{-k}}$$

donde: a_k son los coeficientes LPC y, b es un prámetro que varía de cero a uno.

Como puede observarse de la ecuación (3.39), éste filtro es idéntico al utilizado por el método de excitación multipulso de la sección anterior. Por tanto, las pruebas relacionadas con la variación del parámetro b, no se repetirán.

IV.8.2 "Presentación de Curvas."

A continuación se muestra la gráfica de la señal sintética de voz contra el tiempo discreto, la cual se puede compara con la señal real de voz de la figura 4.1:

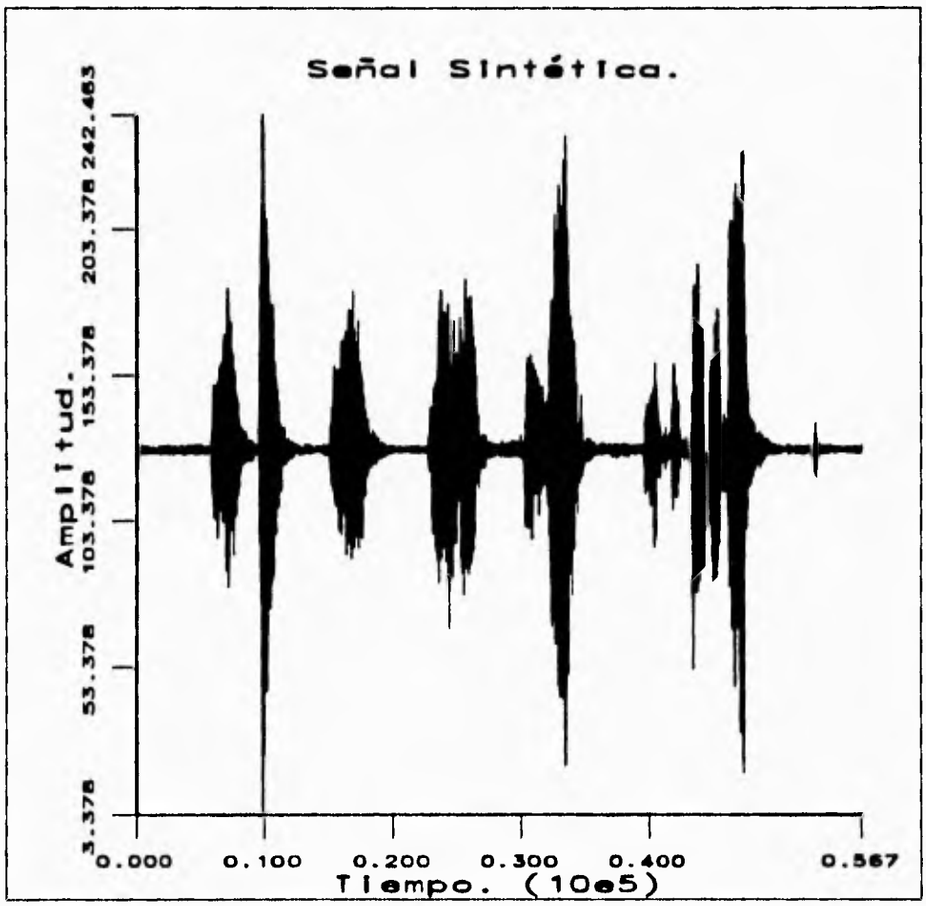


Figura 4.9: Gráfica de la señal de voz sintética contra el tiempo discreto.

Los resultados que se obtienen del método aquí presentado son muy similares a los obtenidos con el método de excitación multipulso. Las diferencias perceptuales no son muy notorias, sin embargo el tiempo de procesamiento del método RPE es mucho menor que el correspondiente para el método de excitación multipulso.

El tiempo de procesamiento del algoritmo RPE, al sintetizar la señal de prueba (sección IV.3), fué de 11.07m. Mientrás que el tiempo de procesamiento del método de excitación multipulso para sintetizar la misma señal fué de 2hr. Por lo tanto, se recomienda la utilización del método RPE sobre la utilización del método de excitación multipulso convencional.

A pesar de que el método RPE es mucho más rápido que el método de excitación multipulso, es lento comparado con los métodos que utilizan algún algoritmo para la estimación del pitch y, dado que los resultados obtenidos mediante los métodos de síntesis, discutidos en el capítulo tres, no presentan diferencias perceptuales notorias entre sí, recomiendo la utilización de los algoritmos más rápidos.

IV.9 "Comparación De Resultados Entre Los Distintos Metodos De Síntesis de Voz."

IV.9.1 "Introducción,"

En esta sección se presenta una comparación de los resultados obtenidos al sintetizar la señal de prueba (sección IV.3), mediante los cinco algoritmos de síntesis de voz discutidos en el capítulo tres. Estos métodos son los siguientes:
1)Método LPC con el algoritmo denominado "AUTOC" para la estimación del pitch.
2)Método LPC con el algoritmo denominado "SIFT" para la estimación del pitch.
3)Método LPC con un método cepstral para la estimación del pitch.
4)Método de excitación multipulso.
5)Método RPE (Regular Pulse Estimation).
El resultado de esta comparación se presenta en un cuadro sinóptico, en el cuál se pueden observar los tiempos de procesamiento así como las tasas de transmisión.

IV.9.2 "Comparación de Resultados."

Antes de presentar el cuadro sinóptico, cabe hacer notar que los valores de los parámetros distintivos de cada algoritmo de síntesis de voz, se fijaron de acuerdo a las recomendaciones dadas por: [RAB76], [NOLL67], [MARK72], [ATAL82] y, [KROON86]. Estas especificaciones se listan para una mayor claridad en: IV.4.1, IV.5.1, IV.6.1, IV.7.1 y, IV.8.1.

Método	Tiempo	Tasa de Transmisión bits/seg.
AUTOC	27.1s.	4564.71
SIFT	30.0s.	4564.71
Cepstral	94.3s.	4564.71
Multipulso	2 hrs.	12800.00
RPE	11.07m.	33600.00

La cuantificación que se utilizo fué la siguiente: un byte para cada uno de los diez coeficientes del filtro de síntesis (a_i), un byte para la energía del frame de voz real (r[0]), un byte para el valor del pitch (métodos AUTOC, SIFT y, Cepstral), un

bit para la desición S/NS (métodos AUTOC, SIFT Y, Cepstral) y, un byte para cada pulso (métodos Multipulso y RPE). El método que da por resultado una mejor calidad perceptual de las señales sintéticas, sin lugar a dudas, es el AUTOC. Esta mejor calidad se debe a que su SNR es mayor con respecto a las SNR's correspondientes a los cuatro métodos restantes. Sin embargo, es necesario aclarar que salvo una mayor SNR del método de AUTOC, las diferencias perceptuales restantes existentes entre éste método y los cuatro restantes son meramente subjetivas.

La mayor diferencia radica en los tiempos de procesamiento y, en las tasas de transmisión. Dado que no existen diferencias perceptuales notables entre los cinco métodos, se prefieren aquellos que su tiempo de procesamiento y, su tasa de transmisión sean menores. Por tanto, el mejor método es el de AUTOC seguido por los métodos: SIFT y cepstral. Dado que el tiempo de procesamiento para el método de excitación multipulso es muy grande en comparación con el tiempo de procesamiento requerido por lo métodos restantes, no se recomienda su uso. En cuanto al método RPE no se aconseja su utilización debido a su gran tasa de transmisiónla, lo cuál implica una gran capacidad de almacenamiento en memoria y/o una mayor cantidad de información a transmitir por el sistema. Esto implica que el canal de comunicaciones utilizado para la transmisión, deberá tener una capacidad mayor, es decir, un ancho de banda mayor ó de lo contrario transmitrá un número menor de señales con respecto a las que transmitiría si se usa algún otro método de síntesis con una tasa de transmisión menor.

"Conclusiones."

Los algoritmos para sintetizar voz tratados en este trabajo pueden dividirse en dos clases: la primera clase consiste en aquellos algoritmos que clasifican los segmentos de voz como sonoros o no sonoros y, por tanto requieren de un estimado del periodo fundamental (pitch). La segunda clase de algoritmos de síntesis de voz no realiza dicha clasificación de los segmentos de voz, sino que obtienen la secuencia de excitación para el filtro de síntesis a partir de la minimización de alguna medida del error perceptual.

Los algoritmos para la estimación del pitch aquí estudiados fueron: el método de autocorrelación con sujetador central para la estimación del pitch (AUTOC) [RAB76], simplified inverse filter transform (SIFT) [MARK72] y, el método cepstral para la estimación del pitch (CEP) [NOLL67].

Los métodos de multipulso estudiados fueron: método de excitación multipulso (MP) [ATAL82] [PAPA] y, el método de regular pulse excitation (RPE).

Según la literatura [MARK72] y [NOLL67], el método AUTOC tiene la ventaja de su sencillez en el cálculo y, por tanto su inplementación en sistemas de tiempo real. Sin embargo, tiene la desventaja teórica de que el recorte central que propone para separar los efectos de la estructura formant sobre la periodicidad de la señal, no garantiza completamente dicha separación ni mucho menos la justifica matemáticamente como en el caso del método cepstral [NOLL67] [MARK72]. Sin embargo, de acuerdo a las simulaciones realizadas, el método de AUTOC sin lugar a dudas es el mejor, ya que produce la voz sintética de mejor calidad, tiene la mayor SNR (razón de señal a ruido), su tiempo de procesamiento es el menor de todos los métodos de síntesis estudiados y, su tasa de transmisión es menor que con respecto a los métodos de excitación multipulso.

En cuanto a las pruebas realizadas con respecto a la variación de parámetros del método AUTOC, se pudó comprobar que el rango, para dichos parámetros, dentro del cuál las diferencias perceptuales son subjetivas, es mayor tanto para el umbral sonoro - no sonoro así como para el nivel de recorte, que el recomendado por [RAB76]. Al variar estos parámetros fuera de sus respectivos rangos, la inteligibilidad de la señal no se ve afectada, sin embargo la pronunciación del sonido de la letra "s" se ve afectada.

De acuerdo con [MARK72], en el estimado del pitch dado por el método de AUTOC siempre existirá un pequeño error debido a la interacción de la estructura formant con la señal de excitación. Por otro lado [MARK72], asegura que la no linealidad introducida por el uso de logaritmos en el método CEP trae como consecuencias: que el pico en el origen del cepstrum no pueda ser utilizado como un valor de referencia para la normalización y, que la amplitud de los picos cepstrales no solamente sea una función del número de periodos pitch contenidos en el frame de

voz, sino también sea una función de la envolvente espectral. Por lo cual [MARK72], recomienda el uso del algoritmo SIFT, el cuál retiene las ventajas tanto del método AUTOC como del método cepstral [MARK72].

De acuerdo con la serie de pruebas y evaluaciones relacionadas con el método SIFT, no se observan diferencias muy notorias con repecto al método cepstral y, el método de AUTOC produce un mejor resultado. El resultado producido por el método SIFT es muy similar al que se obtiene con el método AUTOC, sin embargo este último tiene una SNR mayor.

El tiempo de ejecución del método SIFT es apróximadamente 5 segundos mayor al del método AUTOC. La tasa de transmisión del método SIFT es igual a la de los métodos de AUTOC y cepstral.

La variación del umbral sonoro - no sonoro corroboró que los valores recomendados por [MARK72] son los correctos, sin embargo la utilización de otros valores no afecta a la inteligibilidad de la señal sintética. Esta solo se ve afectada por el surguimiento de cierta reverberancia y, de la dificultad al pronunciar el sonido de la letra "s". Conforme el umbral S/NS crece, la voz se vuelve más ronca.

El método cepstral es ampliamente usado dentro del campo de la investigación con motivos de comparación con otros métodos, ya que justifica matemáticamente la separación de los efectos de la estructura formant de la señal de excitación. Sin embargo, su programación es más compleja y, su tiempo de ejecución mayor que los correspondientes a los métodos AUTOC y SIFT.

De acuerdo a las simulaciones realizadas se observa un resultado más pobre que el obtenido con los metodos: AUTOC y SIFT; dado que la voz sintética es más ronca, la SNR es similar a la del método SIFT e inferior a la del método AUTOC; y que el tiempo de ejecución es aproximadamente un minuto mayor que los correspondientes al método AUTOC y al método SIFT.

De acuerdo a las pruebas realizadas, el rango para el umbral sonoro - no sonoro es: [0.15, 0.2]. Valores fuera de este rango no afectan la inteligibilidad de las señales sintéticas, sin embargo surgen problemas en la pronunciacioón del sonido correpondiente a la letra "s" y mientras más se aleje el valor del umbral S/NS del rango [0.15, 0.2], la voz sintética se vuelve más ronca.

En cuanto a las pruebas realizadas sobre la Codificación Lineal Predictiva (LPC) se corroborarón las recomendaciones dadas por [ATAL71] y [PAPA], en el sentido de que el orden para el filtro de síntesis debe de ser entre 8 y 12; y que la longitud del traslape entre frame sucesivos debe de ser aproximadamente igual a un tercio de la longitud del frame en muestras.

En cuanto al orden del filtro de síntesis se observó que los valores de 8, 10 y 12 producen resultados muy similares, en los que las diferencias perceptuales son subjetivas. Mientras que para un valor del orden del filtro de síntesis igual a seis produce una voz sintética más ronca que con respecto a las voces sintetizadas con ordenes mayores. Además se presentan problemas con la pronunciación del sonido de la letra "s".

Los métodos de excitación multipulso suponen la ventaja de no tener que

realizar una clasificación del segmento de voz en sonoro o no sonoro, además de evitar el tono robotizado de la voz sintética atribuible al error del periodo pitch estimado. Sin embargo, de acuerdo con las simulaciones realizadas el método de excitación multipulso no produce un resultado superior al de los métodos: AUTOC, SIFT y cepstral. De hecho los métodos: AUTOC y SIFT dan mejor resultado que el método de excitación multipulso; dado que este último tiene un tiempo de ejecución bastante considerable además de presentar una tasa de transmisión mucho mayor que la de los métodos: AUTOC, SIFT y cepstral.

Las pruebas de variación de parámetros no corroborarón lo expuesto en [ATAL82] y [PAPA]; ya que la variación del paramétro "b" del filtro de ponderación perceptual no produce cambios percetuales significativos en las señales sinteticas de voz. Mientras que la variación del número de pulsos utilizados para sintetizar un frame de voz no produce cambios perceptuales significativos en las señales de voz, salvo la disminución de la SNR con la disminución del número de pulsos utilizados para sintetizar un frame de voz.

Finalmente en cuanto al método: "regular pulse excitation" (RPE) las pruebas realizadas indican que los resultados obtenidos con este método no son mejores que los resultados obtenidos con los métodos anteriores. Esto se debe a que la señal sintética de voz es bastante ronca con respecto a las producidas por los otros métodos.

El tiempo de ejecución de este método es 12 veces menor que el tiempo de procesamiento para el método de excitación multipulso, 20 veces mayor con respecto a los métodos: AUTOC y SIFT; y 7 veces mayor con respecto al método cepstral.

La tasa de transmisión es la más elevada de los cinco métodos, por lo que no recomiendo este método.

Es de interés hacer notar que en todos los métodos de síntesis de voz aquí presentados, los silencios son mucho más ruidosos que los de la señal real de voz, lo cuál atribuyo a los siguientes hechos:

1) El método LPC no modela adecuadamente las regiones de voz con una baja energía espectral, como lo son las zonas de silencio [PIC93].

2) Presumiblemente la segmentación de la señal de voz en frames de duración constante, no sea adecuada para modelizar correctamente aquellos frames en los que ocurren cambios abruptos en la señal de voz como por ejemplo: pricipio de palabra, fin de palabra, pausas de silencio existentes entre dos sonidos que constituyen una misma palabra y, cambio de un sonido sonoro a un sonido no sonoro y vicevesra. Por tanto, se ha pensado que un estudio para la segmentación inteligente de las señales de voz sería de bastante provecho para su aplicación en problemas no sólo de síntesis de voz, sino también para aplicaciones de reconocimiento de voz. Sin embargo, debido a la extensión y complejidad del tema, éste podría constituir material suficiente para la elaboración de un trabajo futuro.

3) En el caso de los métodos de excitación multipulso se hace uso de un filtro de ponderación perceptual, propuesto por B.S. Atal de una manera un tanto cuanto empírica ya que hasta el día de hoy muchos de los procesos perceptuales que

ocurren a nivel neuronal son desconocidos. Por tanto se puede argumentar que no se lograrán avances significativos en cuanto a la calidad perceptual de las señales sintéticas, mientrás no se comprendan mejor dichos procesos neuronales.
4) Finalmente los modelos parámetricos para la representación de las señales de voz tal vez no sean los más adecuados, ya que estos modelos presuponen un pleno conocimiento tanto de los procesos físicos como neuronales del sistema humano de producción y percepción de la voz. Dado que estos procesos no son del todo comprendidos, cualquier modelo parámetrico para la representación de las señales de voz con lleva cierto error, el cuál influye en la calidad perceptual de la señal sintética de voz.

"Bibliografía."

[ALC89] Alcántara, Rogelio. Apuntes de Procesamiento Digital de Señales. DEPFI-UNAM. México, D.F. 1989.

[ANT79] Antoniou, Andreas. Digital Filters: Analysis and Design. Mc-Graw Hill. International Editions. New York. 1979.

[ATAL71] Atal, B.S. y Hanaver, Suzanne L. "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave". Journal Acoustical Society of America. Vol. 50. No. 2. Agosto 1971. p.p. 637 - 655.

[ATAL82] Atal, Bishnu S. y Remde R. Joel. "A New Model of LPC Excitation for Producing Natural Sounding Speech at Low Bit Rates". Proceedings of the IEEE Int. Conf. Acoust., Speech and Signal Proc., p.p. 614 - 617. Paris, Francia 1982.

[KHOROS] Para información sobre Khoros se puede enviar un mail electrónico a la siguiente dirección: khoros@chama.cccc.unm.edu

[KROON86] Kroon, Peter; Deprettere, ED. F. y Sluyter, Rob J. "Regular Pulse Excitation - A Novel Approach to Effective and Efficient Multipulse Coding of Speech". IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol. ASSP - 34. No. 5. Octubre 1986. p.p. 1054 - 1063.

[MARK72] Markel, John D. "The SIFT Algorithm for Fundamental Frequency Estimation", IEEE Transactions on Audio Electroacoustics. Vol. AU-20. Diciembre 1972. p.p. 367 - 377.

[MULLIS] Mullis, Clifford T. y Roberts, Richard A. Digital Signal Processing", Reimpresión. Addison-Wesley Publishing Company. 1987

[NOLL67] Noll, Michael A. "Cepstrum Pitch Determination". The Journal of the Acoustical Society of America. Vol. 41. No. 2. Agosto 1967. p.p. 293-308.

[ONE88] O'Neill Mark A. "Faster Than Fast Fourier". Byte. Abril 1988. p.p. 293 - 300.

[PAPA] Papamichalis, Panos E. Practical Approaches to Speech Coding. Prentice Hall Inc. Englewood Cliffs, New Jersey. 1987.

[PIC93] Picone, Joseph W. "Signal Modeling Techniques in Speech Recognition". Proceedings of the IEEE. Vol. 81. No. 9. Septiembre 1993. p.p. 1215-1247.

[PROA88] Proakis, John G. Digital Signal Processing: principles, algorithms and applications. Segunda Edición. Macmillan. New York. 1988.

[RAB76] Dubnowski, John J.; Schafer, Ronald W. y Rabiner Lawrence R. "Real Time Digital Hardware Pitch Detector". IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol. ASSP-24. No. 1. Febrero 1976.

"Apéndice A."

A.1 "Guía para la Utilización de la Tarjeta Sound Blaster."

A.1.1 "Introducción."

Existen dos formas de utilizar esta tarjeta, una es entrando a **WINDOWS** y, la otra es desde la **LINEA DE COMANDOS DE DOS**. Los archivos de datos generados en ambiente **WINDOWS** tienen un formato diferente a los generados desde la **LINEA DE COMANDOS DE DOS**, ya que tanto el encabezado como los bytes de control, que caracterizan a los archivos generados bajo ambiente **WINDOWS**, no son iguales al respectivo encabezado y bytes de control, que caracterizan a los archivos generados desde la **LINEA DE COMANDOS DE DOS**. Sin embargo se pueden utilizar los siguiente comandos para efectuar una conversión de formatos: VOC2WAV y, WAV2VOC. Estos comandos se explicarán en una sección posterior de este mismo apéndice.

La principal diferencia que existe entre ambos modos de operación consiste en el rango de frecuencias de muestreo para digitalizar las señales de voz y/o audio. Mientrás que desde la **LINEA DE COMANDOS DE DOS** es posible variar la frecuenciade muestreo desde los 5000 Hz. hasta los 44100 Hz.; bajo ambiente **WINDOWS** sólo es posibles escoger entre las siguientes tres frecuencias de muestreo: 11000, 22000 ó 44000 Hz.

A.1.2 "Manejo de Archivos Bajo Ambiente Windows."

Es necesario ejecutar el comando **WIN** para entrar al ambiente WINDOWS. Una vez que se entra al ambiente de trabajo **WINDOWS** aparecerán una serie de "icons", los que representan diversas aplicaciones. Para trabajar la tarjeta "Sound Blater" es necesario selecionar el icon marcado con la leyenda "Sound Blaster 16". Al seleccionar este icon, se abrirá una ventana que mostrá una serie de icons, los cuales representan todas las aplicaciones posibles de esta tarjeta. De este menu gráfico seleccionaremos la opción "WAVE STUDIO". Al seleccionar esta última opción aparecerá una inteface gráfica para grabar, reproducir, editar y, alterar señales de voz y/o

audio.

En la parte superior de la pantalla a parecerá un primer menu con las siguientes opciones:

1)**FILE**: el menu de file tiene las siguientes opciones: NEW, OPEN, CLOSE, SAVE y, SAVE AS.

2)**EDIT**: el menu de edit tiene las siguientes opciones: CUT, COPY, PASTE, PASTE MIX, CROP TO SELECTION, DELETE y, SELECT ALL.

3)**VIEW**: este menu incluye opciones para la visualización del espacio de trabajo.

4)**SPECIAL**: este menu incluye opciones para realizar efectos especiales sobre las señales de voz y/o audio.

5)**WINDOW**: este munu incluye opciones para acomodar las distintas ventanas que aparecen en pantalla y, que continen las gráficas de las señales de voz y/o audio.

6)**INFO**: aquí se puede obtener información sobre el sistema de cómputo y, la aplicación WAVE STUDIO.

Inmediatamente abajo del menu que se acaba de describir aparece un menu gráfico, del cuál, los icons más importantes son los que representan los botones de una grabadora. Estos incluyen: el play, record, pausa, paro y, un equalizador.

Para grabar una señal de voz y/o audio basta oprimir el boton de record e inmediadatamente aparece una pantalla con un menu gráfico. Dentro de este menu gráfico se puede elegir: si se grabará en mono o en stereo, el número de bits por muestra (8 ó 16) y, la frecuencia de muestreo (11, 22 ó 44 kHz.). Una vez que se han selecionado las opciones deseadas y, se esta listo para iniciar la grabación se seleciona el boton marcado con "START". Inmediatamente después de lo cuál aparece una pantala que indica el porcentaje de la memoria libre que ocupará la señal, que se está grabando. Para detener la grabación basta con seleccionar "STOP".

Para reproducir la señal basta seleccionar el incon con el simbolo de PLAY, para detener la reproducción momentáneamente se oprime el boton con el incon de "PAUSE" y, para detener la reproducción se selecciona el icon de "STOP".

Finalmente es importante señalar que los archivos de datos, ya sea de voz o de audio, generados bajo ambiente **WINDOWS** tendrán la extensión ".WAV".

A.1.3 "Manejo de Archivos Desde la Línea de Comandos."

Es necesario estar en el directorio C:\SB16\VOCUTIL, para poder ejecutar los distintos comandos que nos permitirán: grabar, reproducir, agregar cabecera y bytes de control; cambios de formato a versiones anteriores y; cambios de formato para poder utilizar archivos creados fuera de windows en windows y viceversa.

Los archivos creados fuera de windows tienen la extensión .VOC, cuyo formato varía ligeramente según la versión del software de la tarjeta "Sound Blaster". Este formato consta de un encabezado, el cuál es necesario para la identificación del archivo como un archivo de datos de voz y/o audio; el valor del byte que sigue a la cabecera indica el tipo de bloque de datos que sigue a continuación. Por tanto, existen 8 tipos de indicadores, de los cuales los más importantes se describen a continuación:

INDICADOR 1:
El primer byte es el identificador de bloque y, su valor es uno. Esto indica que se trata de un bloque de datos "nuevo". Los tres siguiente bytes dan el tamaño del bloque de datos, el siguiente byte da la frecuencia de muestreo, el siguiente byte indica el tipo de compresión utilizada y, finalmente comienzan los datos.

INDICADOR 0:
El primer byte es el identificador de bloque y es igual a cero, lo cuál significa que no existen más bloques de datos. Por tanto, este es el último byte del archivo.

"Comandos para Manejar Archivos de Datos Desde la Línea de Comandos de Dos."

Los comandos que se describen a continuación nos sirven para el manejo y la creación de archivos de voz y/o audio fuera del ambiente **WINDOWS**, es decir, desde la línea de comandos de dos.

VREC: este comando se utiliza para grabar archivos de voz y/o audio, su sintaxis es la siguiente:

vrec <archivo> /b: /r: /a: /s: /c: /m: /t:

dónde:

/b: es el tamaño del buffer utilizado para almacenar los datos (2-32). El valor de default es igual a 16.

/r: es el número de bits por muestra (8 o 16). El valor de default es ocho.

/a: indica la fuente de grabación. En este trabajo se utilizó la macro MIC.
/s: indica la frecuencia de muestreo (5000 a 44100 Hz.).
/c: indica el tipo de compresión utilizada para almacenar los datos. El valor cero indica que no se realizó compresión alguna.
/m: indica si la grabación es mono o stereo. Se deben utilizar las macros MONO ó STEREO.
/t: indica el tiempo de grabación (1 - 65000 seg.).

VPLAY: este comando se utiliza para reproducir los archivos de voz y/o audio. Su sintaxis es la siguiente:

vplay <archivo> /b: /t:

Dónde las opciones **/b:** y **/t:** tienen el mismo significado que para el comando **vrec**.

VOCHDR: este es el comando a utilizar cuando se tiene un archivo de datos y, se desea reproducirlo en la tarjeta; pués agrega el encabezado y los bytes de control necesarios para su identificación y uso como archivo de datos de voz y/o audio. La sintaxis de este comando es la siguiente:

vochdr <archivo de entrada> <archivo de salida.VOC> /s: /t: /r: /c:

Las opciones para este comando tienen el mismo significado que para los comandos anteriores, exceptuando:
/t: indica si la grabación se efectúo en mono o en stereo. Se puede utilizar tanto las macros: MONO ó STEREO, como sus respectivos valores que son: uno y dos. El valor de default es uno, o sea, MONO.

VOC2WAV: Este comando se utiliza para efectuar una conversión del formato .VOC al .WAV. La sintaxis de este comando es la siguiente:

voc2wav <archivo de entrada (.VOC)> <archivo de salida (.WAV)> /r:

La opción **/r:** tiene el mismo significado que para el comando **vrec**

WAV2VOC: Este comando efetúa la conversión del formato .WAV al formato .VOC. La sintaxis de este comando es la siguiente:

wav2voc <archivo de entrada (.WAV)> <archivo de salida (.VOC)>

Las opciones para este comando no son de importancia para el desarollo de éste trabajo, así que no se toman en cuenta.

VOCN20: este comando efectúa la conversión del formato .VOC versión 1.20, al mismo formato pero versión 1.10. La sintaxis de este comando es la siguiente:

vocn2o <archivo.voc versión 1.20> <archivo.voc versión 1.10>

/a: indica la fuente de grabación. En este trabajo se utilizó la macro MIC.
/s: indica la frecuencia de muestreo (5000 a 44100 Hz.).
/c: indica el tipo de compresión utilizada para almacenar los datos. El valor cero indica que no se realizó compresión alguna.
/m: indica si la grabación es mono o stereo. Se deben utilizar las macros MONO ó STEREO.
/t: indica el tiempo de grabación (1 - 65000 seg.).

VPLAY: este comando se utiliza para reproducir los archivos de voz y/o audio. Su sintaxis es la siguiente:

vplay <archivo> /b: /t:

Dónde las opciones **/b:** y **/t:** tienen el mismo significado que para el comando **vrec.**

VOCHDR: este es el comando a utilizar cuando se tiene un archivo de datos y, se desea reproducirlo en la tarjeta; pués agrega el encabezado y los bytes de control necesarios para su identificación y uso como archivo de datos de voz y/o audio. La sintaxis de este comando es la siguiente:

vochdr <archivo de entrada> <archivo de salida.VOC> /s: /t: /r: /c:

Las opciones para este comando tienen el mismo significado que para los comandos anteriores, exceptuando:
/t: indica si la grabación se efectúo en mono o en stereo. Se puede utilizar tanto las macros: MONO ó STEREO, como sus respectivos valores que son: uno y dos. El valor de default es uno, o sea, MONO.

VOC2WAV: Este comando se utiliza para efectuar una conversión del formato .VOC al .WAV. La sintaxis de este comando es la siguiente:

voc2wav <archivo de entrada (.VOC)> <archivo de salida (.WAV)> /r:

La opción **/r:** tiene el mismo significado que para el comando **vrec**

WAV2VOC: Este comando efetúa la conversión del formato .WAV al formato .VOC. La sintaxis de este comando es la siguiente:

wav2voc <archivo de entrada (.WAV)> <archivo de salida (.VOC)>

Las opciones para este comando no son de importancia para el desarollo de éste trabajo, así que no se toman en cuenta.

VOCN20: este comando efectúa la conversión del formato .VOC versión 1.20, al mismo formato pero versión 1.10. La sintaxis de este comando es la siguiente:

vocn2o <archivo.voc versión 1.20> <archivo.voc versión 1.10>

Este comando no presenta opciones.

VOCO2N: este comando efectúa la conversión del formato .VOC versión 1.10, al mismo formato pero versión 1.20. La sintaxis de este comando es la siguiente:

voco2n <archivo.voc versión 1.10> <archivo.voc versión 1.20>

Al igulal que en el caso anterior, éste comando no presenta opciones.

"Apéndice B."

B.1 "Introducción."

Todos los programas en lenguaje C realizados para los distintos métodos de síntesis de voz tienen la misma estructura. Esto se debe a que todos los métodos de síntesis estudiados en este trabajo hacen uso del mismo modelo para la representación paramétrica de las señales de voz (veáse III.2).

Existe un programa principal a través del cuál se hace llamada a las diversas subrutinas, las cuales se encuentran en archivos idependientes al archivo en el que se encuentra almacenado el programa principal. Estas subrutinas no reciben parámetros sino que operan sobre variables globales. La estructra del programa principal se ilustra en la siguiente tabla:

Sección de librerías.
Declaración de Variables Globales

Programa Principal{
Normalización del archivo de voz
Determinación del número de frames a procesar (N)
for(i = 1; i ≤ N; i++){
Subrutinas pertenecientes al método de síntesis en particular
}
Normalización del archivo de voz sintética.
Obtención de espectro y autocorrelaciones tanto de la señal de voz real como de la señal estimada.
}

La sección de librerías es aquella dónde se incluyen las librerías del lenguaje

C, que utiliza el programa principal. Además se incluye el archivo con extensión ".h", en el cuál se han utilizado macros para establecer el valor de los parámetros más significativos para el desempeño de los métodos de síntesis programados; de tal modo que cuando sea necesario modificar el valor de un parámetro en particular, tan sólo baste con modificar la instrucción "# define".

La sección de declaración de variables globales incluirá todas las variables sobre las cuales operarán las distintas subrutinas. Los arreglos y variables más importantes son:

1)**inter**: arreglo que almacena el frame de voz real para el análisis del pitch.
2)**As**: arreglo que almacena el frame de voz real para el análisis LPC.
3)**vozsin**: arreglo que almacena la salida del filtro de síntesis.
4)**r**: arreglo que almacena la secuencia de autocorrelación.
5)**A**: arreglo que almacena los coeficientes del filtro de síntesis.
6)**KK**: arreglo que almacena los coeficientes de reflexión.
7)**G**: variable que almacena la ganancia del filtro de síntesis.
8)**entrada**: arreglo que almacena la secuencia de excitación para el filtro de síntesis.
9)**maximo**: variable que almacena el valor máximo del archivo de voz real sin normalizar.
10)**promedio**: variable que almacena el valor promedio del archivo sin normalizar.
11)**num_interval**: variable que almacena el número de frames a procesar.
12)**contints**: variable que se utiliza como contador del número de frames procesados.
13)**veces**: variable que indica el número de muestras por frame.
14)**start** y **end**: variables que sirven para medir el tiempo de procesamiento del programa.
15)**nomarchi**: variable que almacena el nombre del archivo que contiene los datos de voz real.
16)**salida_voznorm**: variable que almacena el nombre del archivo que contiene la voz sintética.
17)***invoz**: puntero al archivo que contiene los datos de voz real.
18)***voz_outnorm**: puntero al archivo que contiene la voz sintética.

El programa principal siempre comienza con la subrutina de normalización del archivo de datos de voz real. Esta subrutina tiene por objetivos: la eliminación de cualquier offset en la señal de voz real y, su normalización, de tal manera que el rango dinámico de la señal quede comprendido dentro del intervalo [-1,1]. El offset que trae originalmente la señal de voz, se almacena en la variable **promedio**; mientras que el valor máximo absoluto que traía originalmente la señal de voz, se almacena en la variable **maximo**.

El siguiente paso en el programa principal es la obtención del número de frames a procesar. Este se obtiene del tamaño en muestras del archivo de voz real, del tamaño en muestras del frame de voz y, del tamaño en muestras del traslape existente entre frames sucesivos.

El siguiente paso del programa principal es la ejecución del único loop existente, cuya variable de control es la variable **contints**, la cuál varía desde uno hasta el número de frames a procesar. El cuerpo principal del loop consiste de las subrutinas particulares a cada método de síntesis, las cuales serán descritas en la secciones subsiguientes de este apéndice.

Una vez ejecutado el loop anterior se obtiene un archivo que contiene los datos de voz sintética. Sin embargo, para que los datos de voz sintética puedan ser leídos correctamente por la tarjeta "Sound Blaster", primero la señal debe tener una media igual a la que utiliza la tarjeta para la lectura de datos y, en segundo lugar las amplitudes de la señal sintética no pueden ser mayores al rango de valores que admite la tarjeta. La subrutina de normalización de la señal de voz sintética se encarga de las operaciones anteriormente mencionadas.

La última subrutina se encarga de obtener: el espectro, secuencia de autocorrelación y espectro estimado de la señal de voz real; así como el espectro y secuencia de autocorrelación de la señal de voz sintética.

B.2 "Documentación del Programa para el Método de Síntesis: LPC con AUTOC."

El programa se compone de un programa principal, del cuál se llama a las distintas subrutinas. Estas subrutinas se encuentran en diferentes archivos y, no hacen uso de parámetros sino de variables globales para la ejecución de sus respectivas tareas.

El nombre del archivo que contiene a la función main se llama **"estpitch.c"** y en este se llama a las siguientes subrutinas: **nombres()**, **norm()**, **numints()**, **carga_matriz()**, **recorte_central()**, **autocor()**, **ventaneo()**, **preenfasis()**, **obten_excitación()**, **corel()**, **coeficientes()**, **sintetisa()**, **deemfasis()**, **arch_vozsin()**, **norminv()** y, **esparch()**.

La subrutina **nombres()** pide el nombre para los siguientes archivos: archivo de datos de entrada (voz real), archivo para el espectro de la señal de voz real, archivo para la estimación parámetrica del espectro de la señal de voz real, archivo para las autocorrelaciones de la señal de voz real, archivo para la señal sintética sin normalizar, archivo para la señal sintética normalizada, archivo para las autocorrelaciones de la señal sintética y finalmente, si se desea una escala normal o logarítmica para los espectros.

La subrutina **norm()** tiene por objeto eliminar cualquier offset de la señal de voz real y, normalizar la misma a ± 1.

La subrutina **numints()** tiene como propósito determinar el número de frames que contiene la señal de voz real, en base a la longitud del archivo en muestras, el tamaño del frame en muestras y, el tamaño del traslape en muestras. El número de frames contenidos en la señal de voz real determinará el número de ciclos, en los cuales se ejecuatarán las siguientes subrutinas: **carga_matriz()**, **recorte_central()**, **autocor()**, **ventaneo()**, **preenfasis()**; **obten_excitación()**, **corel()**, **coeficientes()**, **sintetisa()**, **deemfasis()** y, **arch_vozsin()**.

La subrutina **carga_matriz()** lee el archivo de datos de entrada (voz real) y carga los vectores **inter**, **As** y, **vozorig** con los datos de voz real correspondientes al frame en procesamiento. El vector **inter** se utiliza para la estimación del pitch, el vector **As** se utiliza para el análisis LPC y, el vector **vozorig** se utiliza para almacenar temporalmente los datos comunes al frame en procesamiento y, al siguiente frame.

La subrutina **recorte_central()** tiene por objeto llevar a cabo el recorte central de la señal requerido por el algoritmo de estimación del pitch. Dentro de esta subrutina se llama a la subrutina halla_max, la cuál tiene por objeto hallar el valor máximo de un vector. Dicha subrutina recibe los siguientes parámetros: el nombre del vector, el límite inferior del intervalo de búsqueda, el límite superior del intervalo de búsqueda y, un parámetro tipo caracter que indica si se desea obtener el valor máximo positivo ó el valor máximo absoluto.

La subrutina **autocor()** tiene por objeto el cálculo de las autocorrelaciones

para el análisis del pitch, para lo cuál llama a la subrutina **ar()**. Esta subrutina recibe los siguientes parámetros: el vector de entrada, el valor inicial y final del atraso para el cálculo de las autocorrelaciones y, un vector de salida, dónde se almacenan los resultados.

La subrutina **ventaneo()** aplica una ventana de Hamming al vector As para el análisis LPC.

La función **preenfasis()** tiene por objeto aplicar el filtro de énfasis al vector As para el análisis LPC.

La función **obten_excitacion()** utiliza las variables **v_ uv** y **pospitch** para generar la excitación correspondiente al filtro de síntesis. La variable **v_uv** indica si el frame en procesamiento es sonoro ó no sonoro, mientras que la variable **pospitch** indica, en caso de que resulte sonoro el frame, el valor del pitch. Finalmente la secuencia de excitación se almacena en el vector **entrada**.

La subrutina **corel()** cálcula las autocorrelaciones necesarias para el análisis LPC, para lo cuál llama a la función **ar()**.

La subrutina **coeficientes** cálcula a partir de las autocorrelaciones los coeficientes del filtro de síntesis y, los coeficientes de rflexión. Los coeficientes del filtro de síntesis se almacenan en el vector **A** y, los coeficientes de reflexión se almacenan en el vector **KK**.

La subrutina **sintetiza** utiliza los arreglos **A** y **entrada**, además de la ganancia del filtro de síntesis (**G**); para generar la voz sintética.

La subrutina **archvozsin()** tiene por objeto archivar la voz sintética generada para el frame en procesamiento.

El contador **contints** se incrementa y, si su valor alcanza el número de frames determinados por la subrutina **numints()**, el programa continua con la subrutina **norminv()**, en caso contrario se vuelven a ejecutar las subrutinas: **carga_matriz()**, **recorte_central()**, **autocor()**, **ventaneo()**, **preenfasis()**; **opbten_excitación()**, **corel()**, **coeficientes()**, **sintetisa()**, **deemfasis()** y, **arch_vozsin()**.

La subrutina **norminv** tiene por objeto la eliminación de cualquier offset de la señal sintética de voz, la normalización de dicha señal a ± 1 y por último dar el offset y ganacia adecuados a la señal sintética de voz, para que pueda ser leída por la tarjeta "Sound Blaster".

La subrutina **esparch** tiene como objetivo generar y archivar los espectros y autocorrelaciones, tanto de la señal de voz real como de la señal sintética de voz. Además genera el espectro parámetrico estimado para la señal real de voz.

B.3 "Documentación del Programa para el Método de Síntesis: LPC con SIFT."

Dado que este programa tan sólo varía en la estimación del pitch, con respecto al programa documentado en la sección B.1, aquí sólo se documentarán las subrutinas pertinentes.

La estructura del programa es igual a la del programa anterior, es decir, las subrutinas utilizan variables globales y, cada subrutina se almacena en un archivo idependiente de aquel utilizado para almacenar el programa principal.

La principal diferencia rádica en que primero se efectúa el análisis del pitch para todos los frames de voz y, después se efectua el análisis LPC para todos los frames de voz.

La subrutina **prefiltra()** tiene dos objetivos: 1) filtrar la señal real de voz a través de un filtro paso bajas (f_c = 900 Hz.) y, 2) efectuar una decimación de 4 a 1. El vector con los datos de la señal de voz real es **inter** y, el vector con los datos filtrados y decimados es **intersm**.

La subrutina **residual()** tiene por objetivo filtrar la señal de voz a través del filtro inverso **A(z)**. Los vectores sobre los que opera esta subrutina son **intersm** y, **resi**. En el vector **resi** se almacena la salida del filtro inverso **A(z)**.

La subrutina **autocorelmax()** tiene por objeto hallar el valor máximo de la secuencia de autocorrelación. El valor máximo de la secuencia de autocorrelación se almacena en la variable **magpitch** y, la posición de dicho valor máximo se almacena en **pospitch**.

La subrutina **decide()** tiene por objeto declarar el frame en procesamiento como sonoro o no sonoro, de arcuerdo con el umbral de desición (S/NS). Además, la subrutina **decide()** archiva las variables: pospitch, magpitch y, voiced. Esta última variable es de tipo char e indica si el frame es sonoro o no sonoro.

Una vez que se ha completado el análisis del pitch para todos los frames de voz, se ejecuta la subrutina **ajusta()**. Esta subrutina tiene por objeto redeclarar como sonoro o no sonoro ciertos intervalos de voz. Esta redeclaración toma en cuenta tanto la decisión que se tomó en el frame anterior como la que se tomó en el siguiente frame.

Las subrutinas para el análisis LPC son idénticas a las que se documentarón en la sección B.1.

B.4 "Documentación del Programa para el Método de Síntesis: LPC con un Método Cepstral para la Estimación del pitch."

La estructura de este programa es excatamente igual al de la sección B.2, es decir, primero se efectúa el análisis del pitch para todos los frames de voz y, después se efectúa el análisis LPC para todos los frames de voz.

Nuevamente como en los casos anteriores las subrutinas no reciben parámetros sino que operan sobre variables globales. Además cada subrutina se almacena en un archivo, el cuál es independiente con respecto al archivvo que contiene el programa principal.

Dadas las semejanzas mencionadas en los dos parrafos anteriores, aquí sólo se procederá a documentar las subrutinas pertinentes.

La subrutina **cepstrum** cálcula el cepstrum del frame en procesamiento. Para dicho cálculo llama dos veces a la función fft la cuál recibe como parámetros: el vector que contiene los datos reales a transformar, un vector que contiene los datos imaginarios a transformar, el número de datos a transformar y, la potencia de dos del número de datos a transformar.

La subrutina **pondera** realiza una ponderación lineal sobre el cepstrum del frame en procesamiento y, archiva dicho cepstrum ponderado.

Una vez que se ha completado el cálculo del cepstrum para todos los frames, se ejecuta la subrutina **ajusta()**, la cuál tiene la función de declarar los frames de voz como sonoros o no sonoros. Esta declaración toma en cuenta las decisiones tomadas tanto para el frame anterior como para el siguiente frame.

Las subrutinas utilizadas para efectuar el análisis LPC son idénticas a las de las secciones B.1 y B.2.

B.5 "Documentación del Programa para el Método Excitación Multipulso."

La estructura del programa difiere de las anteriores (secciones B.1, B.2 y, B.3), en que aquí no existe algoritmo alguno para la estimación del pitch. Sin embargo, ciertas subrutinas relativas al análisis LPC son las mismas que las utilizadas en los programas anteriores. Por lo tanto, sólo se decumentarán las subrutinas pertinentes.

La subrutina **residual()** filtra la señal real de voz, almacenada en el vector **As**, a través del filtro inverso **A(z)**. La salida de dicho filtro se escribe en el vector **resi**.

La subrutina **obten_excitacion** no es la misma que se utilizó para los programas anteriores (secciones B.1, B.2 y, B.3); ya que en este caso se calculan las posiciones y amplitudes de los pulsos que constituyen la señal de excitación para el filtro de síntesis. Para dicho cálculo se recurre a la subrutina **filtinv()**. Esta subrutina recibe como parámetro el nombre del vector que contiene los datos a ser filtrados a través del filtro de ponderación perceptual H(bz).

B.6 "Documentación del Programa para el Método Regular Pulse Excitation."

Para este programa ciertas subrutinas relativas al análisis LPC son idénticas que aquéllas correspondientes a los programas anteriormente discutidos (secciones B.1, B.2, B.3 y, B.4), por lo que sólo se documentarán las subrutinas pertinentes.

La subrutina **residual** filtra la señal real de voz a través del filtro inverso **A(z)**. Esta subrutina opera sobre el vector de datos **As** y, escribe la salida del filtro inverso **A(z)** en el vector **resi**.

La subrutina **math()** obtinene la matriz **H**, cuyo j-ésimo renglón contiene la respuesta al impulso h(n), del filtro de ponderación perceptual $1/A(\gamma z)$, cuando éste último es excitado con el impulso δ(n-j).

Las subrutina **math** llama a su vez a la subrutina **filtrar()**, la cuál recibe como parámetro el nombre del vector de datos a ser filtrado a través del filtro de ponderación perceptual $1/A(\gamma z)$.

La subrutina **transpuesta** como su nombre lo indica, obtiene la transpuesta de la matriz **H** y, el resultado lo escribe en la matriz **HT**.

La subrutina **suavizar()** efectúa la multiplicación de matrices: HH^t y, escribe el resultado en el arreglo **S**.

La subrutina **vectorb()** efectúa la operación matricial:

$$b^{(k)} = \frac{1}{r_0} r S M_k^t$$

donde: r es un vector que representa la señal residual, M_k^t es la transpuesta de la matriz de posiciones y, b respresenta la secuencia de excitación [KROON86].

Finalmente la subrutina **obten_excitación** difiere de las subrutinas para los casos anteriores, ya que el propósito de esta subrutina es determinar el valor del defasamiento k para el vector que representa la secuencia de excitación $b^{(k)}$, mediante:

$$min\{E^{(k)}\} = max\{b^{(k)} b^{(k)t}\}$$

B.7 "Presentación de Resultados."

B.7.1 "Introducción."

En este trabajo se utilizó el sistema Khoros para la presentación de las curvas. Estas fueron presentadas con ayuda del lenguaje visual de cantata y de las rutinas: **asc2viff**, **xprism2** y **xprism3**, existentes dentro del sistema Khoros. La rutina **asc2viff** sirvió para convertir los datos del formato ascii al formato viff. La rutina **xprism2** sirvió para graficar los datos pertenecientes a las señales tanto de voz real como sintéticas. Mientras que la rutina **xprism3** graficó los espectros tridimensionales.

El sistema Khoros integra varias interfases, generadores de código, ayudas, visualización de datos, cálculos dsistribuidos y, procesamiento de información. La infraestructura consiste de seis subsistemas principales [KOROS]:

1) Un lenguaje visual de alto nivel.

2) Un sistema de desarollo de interfase usuario que consiste de una Especificación de Interfase Usuario (UIS) y, generadores de código que hacen uso de la UIS para generar código para todos los programas creados dentro del sistema Khoros.

3) Un formato interoperable para el intercambio de datos (VIFF), el cuál es soportado por una librería de conversión.

4)Librerías para el despliegue y procesamiento de datos; procesamiento de imagenes, procesamiento digital de señales, análisis numérico, conversión de datos y archivos, despliegue de graficas e imágenes.

5) Un conjunto de programas interactivos para X Windows relativos al despliegue de imágenes, manipulación de los mapas de colores, animación, graficación, compresión de imágenes y, visualización de superficies.

6)Un conjunto de "sistemas meta", los cuales dan fundamento a los cálculos distribuidos y, a la transferencia eficiente de datos.

B.7.2 "El lenguaje Visual."

El lenguaje visual de Khoros se llama **cantata**. El usuario realiza una aplicación en **cantata** al conectar nodos de procesamiento (llamados "glyphs") para dar lugar a un diagrama de flujo. Los Glyphs se seleccionan de las distintas librerías de rutinas disponibles en Khoros; el usuario también puede crear sus propias librerías por medio de la utilización del sistema de desarollo de interfase usuario.

Cada Glyph representa un programa entero, además existen glyphs de control de flujo (como condicionales, ciclos de repetición, etc.), que extienden la funcionalidad de los diagramas de flujo.

Los procedimientos visuales son un conjunto de glyphs que realizan una determinada tarea dentro del diagrama de flujo. Estos procedimientos visuales están representados por un glyph, de tal manera que el espacio de trabajo no se sature de glyphs y por tanto. se complique todavía más el diagrama de fuljo.

B.7.3 "Sistema de Desarollo Interfase-Usuario."

La estructura de cualquier programa en Khoros está definida por una Especificación Interfase-Usuario (UIS). Esta especificación de alto nivel se usa junto con los generadores de código para permitir que los programas desarollados en Khoros combinen una interfase de comando de línea tipo UNIX (CLUI) con una interfase gráfica basada en X-Windows (GUI). Distintas subrutinas programadas por el usuario pueden ser fácilmente incorporadas al lenguaje visual. Las herramientas del software citadas a continuación sirven a este último propósito:
preview: herramienta de despliegue GUI.
composer: editor GUI.
conductor: herramienta para la generación de código para una GUI.
ghostwriter: herramienta para la generación de código para la CLUI.
kinstall: herrramienta para la configuración y manejo de la subrutina para khoros.

B.7.4 "Formato Interoperable para el Intercambio de Datos."

El formato utilizado por Khoros para el despliegue de imagenes así como para los archivos de imagenes (VIFF) soporta objetos geométricos, datos multidimensionales, y un esquema robusto de mapeo. La conversión de tipos entre diferentes arquitecturas se efectúa de una forma automática por medio de las utilerías del "meta-system".

Khoros soporta los siguientes formatos [Khoros]: TIFF, pbm, BIG, DEM, DLG, ELAS, FITS, Matlab, SUN raster, TGA, y xbm.

B.7.5 "Librerías para el Procesamiento de Datos."

Khoros incluye librerías de programas que pueden operar en datos, vectores, matrices, datos multibanda o vectores de dimensión N [KHOROS]. Estos operadores son polimórficos, es decir, actúan sobre bits, bytes, shorts, ints y tipos complejos. El término también implica que dichos operadores actúan de manera diferente dependiendo de las dimensiones y/u organización de los datos.

Existen dos niveles de interfases definidos en las librerías de funciones para Khoros: la interfase del programa o proceso y, la interfase de la llamada a la función o interfase del procedimiento. La interfase del programa está completamente determinada por la especificación de alto nivel interfase-usuario descrita en la sección B.7.3. La interfase de procedimiento permite que las subrutinas sean combinadas en un sólo programa.

B.7.6 "Aplicaciones en X-Windows."

Los programas con interfase gráfica interactiva que forman parte de Khoros están basados en los wigdets Athena y MIT X11R4. Las aplicaciones escritas bajo el sistema Khoros son automáticamente dotadas de un registro periodico y de una capacidad de reproducción que es útil para efectuar demostraciones y, para propósitos de instrucción. Esta capacidad de registro periodico y reproducción fué extendida para trabajar con despliegues multiples simultáneamente con un controlador de interfase de usuarios distribuidos. Esta interface permite no solamente ejecutar y desplegar una aplicación en Khoros a varios usuarios simultáneamente, sino que también puede recibir datos de entrada provenientes de cada usuario.

B.7.7 "Base del Sistema Meta."

Khoros está diseñado para operar y utilizar un ambiente computacional heterogéneo. La librería del sistema meta provee un conjunto de llamadas a funciones que extienden las llamadas standard al sitema Unix para proveer una interfase a nivel red.

"Apéndice C."

C.1 "Representación de los Sistemas Lineales e Invariantes en el Tiempo Discreto."

Un sistema lineal e invariante en el tiempo discreto es aquel cuya relación entre la señal de entrada x[n], y la señal de salida y[n] esta dada por la transformación L, que se caracteriza por las propiedades de linealidad e invariancia temporal [ALC89].

Existen varias formas de representar matemáticamente a un sistema lineal e invariante en el tiempo discreto. Estas formas son: mediante su respuesta al impulso unitario $\delta(n)$, mediante su función de transferencia H(z), mediante su ecuación en diferencias y, mediante su respuesta en frecuencia [ALC89].

C.1.1 "Representación de los Sistemas Lineales Mediante su Respuesta al Impulso Unitario."

La respuesta y[n], de un sistema lineal e invariante en el tiempo discreto L, cuando la señal de entrada x[n], es igual al impulso unitario $\delta[n]$ está dada por:

$$y[n] = L\{\delta[k]\}$$

Definamos a h[n] como la respuesta al impulso unitario $\delta[n]$:

$$L\{\delta[n]\} = h[n]$$

Una vez estimada la respuesta al impulso unitario, h[n], la salida del sistema para cualquier entrada x[n] estará dada por:

$$y[n] = \sum_{i=-\infty}^{\infty} x[i]h[n-i] \qquad \text{(C.1)}$$

C.1.2 "Representación de los Sistemas Lineales Mediante La Función de Transferencia."

La función de transferencia, H(z), para un sistema lineal se puede obtener al aplicar la transformada Z a su respuesta al impulso h[n]:

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$$

Recordando que [ALC89]:

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

donde el operador * denota la convolución de las señales temporales.

Si aplicamos la transformada Z, a la ecuación anterior obtenemos:

$$Y[z] = Z\{x[n] * h[n]\}$$

Recordando que: $Z\{x[n] * h[n]\} = Z\{x[n]\}Z\{h[n]\}$, obtenemos la representación por medio de la función de transferencia del sistema lineal e invariente en el tiempo discreto:

$$Y[z] = H[z]X[z] \qquad \text{(C.2)}$$

C.1.3 "Representación de los Sistemas Lineales Mediante su Ecuación en Diferencias."

La expresión más general para llevar a cabo esta representación es la siguiente:

$$y[n] = -\sum_{i=1}^{p} a_i y[n-i] + \sum_{m=0}^{q} b_m x[n-m] \qquad \text{(C.3)}$$

donde los a_i y los b_m son los coeficientes que definen el sistema.

Si aplicamos la transformada Z a la ecuación anterior obtenemos:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_{q-1} z^{-q+1} + b_q z^{-q}}{1 + a_1 z^{-1} + \cdots + a_{p-1} z^{-p+1} + a_p z^{-p}} \qquad \text{(C.4)}$$

Esta última ecuación relaciona el modelo por ecuación en diferencias con la función de transferencia [ALC89].

C.1.4 "Representación de los Sistemas Lineales Mediante su Respuesta en Frecuencia."

La respuesta en frecuencia de un sistema lineal, con respuesta al impulso, h[n], se obtiene evaluando la función de transferencia H(z) para $z = e^{j\omega}$ [ALC89],

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n)e^{-j\omega n}$$

Puesto que $H(e^{j\omega})$ es una función compleja, podemos escribir:

$$H(e^{j\omega}) = \left|H(e^{j\omega})\right| e^{j\theta_h(\omega)} \tag{C.5}$$

donde $|H(e^{j\omega})|$ y $e^{j\theta_h(\omega)}$ son la magnitud y la fase, respectivamente, de la respuesta en frecuencia del sistema.

C.2 "Modelos Autoregresivo, de Promedio Móvil y, Autoregresivo de Promedio Móvil."

Es importante señalar que en el caso de que la secuencia de entrada, x[n], a la ecuación (C.3), sea una secuencia de variables aleatorias independientes se presenta los siguienes casos:
1)Cuando q = 0, tenemos la ecuación de un modelo autoregresivo (AR):

$$y[n] = -\sum_{i=1}^{p} a_i y[n-i] \tag{C.6}$$

2)Cuando p = 0, tenemos la ecuación de un modelo de promedio móvil (MA):

$$y[n] = \sum_{m=0}^{q} b_m x[n-m] \tag{C.7}$$

3) Cuando $p \neq 0$ y $q \neq 0$, tenemos la ecuación de un modelo autoregresivo y de promedio móvil (ARMA):

$$y[n] = -\sum_{i=1}^{p} a_i y[n-i] + \sum_{m=0}^{q} b_m x[n-m] \tag{C.8}$$

C.3 "Densidad Espectral de Potencia".

En el caso de las señales aleatorias estacionarias no se hace referencia a la transformada de Fourier sino a la densidad espectral de potencia [PROA88]. Esto se debe a que las señales aleatorias estacionarias no tienen energía finita y por tanto, la transformada de Fourier no existe. En lugar de hablar de la energía de dichas señales se hace referencia al término: densidad espectral de potencia [PROA88]. Esto se debe a que las señales aleatorias estacionarias tienen un promedio de potencia finito [PROA88].

En el caso general, cuando una señal aleatoria estacionaria es representada por un modelo ARMA, su función de densidad espectral de potencia se obtiene de la transformada Z de la ec.(C.8):

$$S(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_q z^{-q}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}}$$

Al sustituir $z = e^{j\theta}$, y obtener el modulo de $S(j\theta)$ se obtiene:

$$S(\theta) = b_0 \frac{|\sum_{i=1}^{q} c_i cos(i\theta)|^2 + |\sum_{i=1}^{q} c_i sen(i\theta)|^2}{|\sum_{i=1}^{p} a_i cos(i\theta)|^2 + |\sum_{i=1}^{p} a_i sen(i\theta)|^2}$$

donde $c_i = \frac{b_i}{b_0}$ para $i = 1, \cdots, q$

Al utlizar $e^{j\theta} = cos(\theta) + j sen(\theta)$, se obtiene:

$$S(\theta) = b_0 \frac{|\sum_{i=0}^{q} c_i e^{-ji\theta}|^2}{|\sum_{i=0}^{p} a_i e^{-ji\theta}|^2} \qquad \text{(C.9)}$$

Para el caso de un modelo autoregresivo q = 0 y, la ec (C.8) se convierte en la ec.(1.51):

$$\hat{S}(\theta) = \frac{\alpha}{|\sum_{i=0}^{p} a_i e^{-ji\theta}|^2}$$

donde los coeficientes a_i, se obtienen a partir de la secuencia de autocorrelación, r(n), mediante la recursion de Levinson-Durbin [PAPA] [PROA88].

C.3.1 "Algoritmo de Levinson-Durbin."

Partiendo de la secuencia de autocorrelación r(i), $i = 0, \cdots, p$ se pueden calcular de una manera recursiva los coeficientes a_i de la ec.(1.51) mediante la siguiente recursión [PAPA]:

$$E(0) = r(0) \tag{C.10}$$

$$K_i = -\frac{r(i) + a_1^{(i-1)} r(i-1) + \cdots + a_{i-1}^{(i-1)} r(1)}{E(i-1)} \quad para\, i = 1, \cdots, p \tag{C.11}$$

$$a_i^{(i)} = K_i \tag{C.12}$$

$$a_j^{(i)} = a_j^{(i-1)} + K_i a_{i-j}^{(i-1)} \quad j = 1, \cdots, i-1 \tag{C.13}$$

$$E(i) = (1 - K_i^2) E(i-1) \tag{C.14}$$

Los coeficientes $a_j^{(i)}$, $j = 1, \cdots, i$ son los coeficientes de un modelo de orden i. De aquí que los coeficientes del modelo deseado de orden p sean:

$$a_j = a_j^{(p)}, \quad j = 1, \cdots, p$$