UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

CODIFICACION DE VOZ BASADA EN CUANTIZACION VECTORIAL DE LA TRANSFORMADA DISCRETA DE FOURIER DE DURACION CORTA

TESIS

QUE PARA OPTAR POR EL GRADO DE:

MAESTRÍA EN INGENIERÍA DE CONTROL

PRESENTA:

HE NING

1985

0912
duplicado

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

DIVISION DE ESTUDIOS DE POSGRADO
FACULTAD DE INGENIERIA

CODIFICACION DE VOZ

BASADA EN CUANTIZACION VECTORIAL DE

LA TRANSFORMADA DISCRETA DE FOURIER

DE DURACION CORTA

Créditos asignados a la tesis 12 (doce)

APROBADO POR EL JURADO:

Presidente: Dr. Federico Kuhlmann Rodriguez 11/11/85

Vocal: Dr. Andrés Buzo de la Peña 11/Nov/85

Secretario: Dr. Guillermo Rebolledo Cortizo 11/Nov/85

Suplente: Dr. Stanislaw Raczynski 11 Nov. 85

Suplente: Dr. Francisco García Ugalde 11. Nov. 85

UNIVERSIDAD NACIONAL AUTÓNOMA

DIVISION DE ESTUDIOS DE POSGRADO
FACULTAD DE INGENIERIA

Profr. LUIS ANDRES BUZO DE LA PEÑA
P r e s e n t e

Comunico a usted que a propuesta del SUBJEFE DEL AREA DE ELECTROMECANICA ha sido designado como director de tesis del alumno(a) HE NING para obtener el grado de M EN I EN CONTROL.

Mucho he de agradecerle su comunicación, por escrito, de la aceptación a esta designación y el nombre de la tesis a desarrollar.

Atentamente,
"POR MI RAZA HABLARA EL ESPIRITU"
Cd. Universitaria a 5 de agosto de 1985.
EL JEFE DE LA DIVISION

DR. GABRIEL ECHAVEZ ALDAPE

E.5.1

CODIFICACION DE VOZ BASADA EN CUANTIZACION VECTORIAL DE LA TRANSFORMADA DISCRETA DE FOURIER DE DURACION CORTA

CONTENIDO

RESUMEN

La técnica de codificación de formas de onda en el dominio de la frecuencia y la cuantización vectorial han sido aplicadas respectivamente a la compresión de la señal de voz a tasas de bit (bit rate) baja y mediana [1,2,3,4]. No obstante, al conocimiento del autor, existen pocos trabajos que combinen ambas técnicas de una manera directa para codificar voz. La principal dificultad de usar los cuantizadores vectoriales (VQ) para codificar formas de onda es el elevado requerimiento de recursos de computación. Para reducir este requerimiento, se han propuesto varios tipos de VQ con estructura interna permitiendo un rápido acceso a su vocabulario, entre ellos, se tienen los VQ de ganancia y forma (GS-VQ) y de producto cartesiano (PC-VQ) [2,5,6]. En esta tesis describimos el diseño y la simulación del un sistema de compresión de voz basado en cuantización vectorial de la transformada discreta de Fourier de duración corta (discrete short-time Fourier transform, DSTFT). El VQ usado es una combinación de PC-VQ y GS-VQ, el cual cuantiza directamente el espectro de voz obtenido mediante la DSTFT bajo el criterio de error cuadrático en términos de las componentes en la frecuencia. La simulación es realizada usando una minicomputadora, y los resultados experimentales muestran que este sistema de codificación alcanza una buena "calidad de comunicación" a la tasa de 6.4 Kb/s (1 bit/muestra).

RECONOCIMIENTO

Deseo expresar mi sincero agradecimiento al Dr. Andrés Buzo, director de la tesis, por su orientación, sus valiosas sugerencias y asesoría; así como al Laboratorio de Computación de la División de Estudios de Posgrado de la Facultad de Ingeniería, por las facilitades de cómputo y la colaboración continua que me ha brindado durante el desarrollo del proyecto.

También agradezco profundamente a la Embajada de la República Popular de China en México por su estímulo y apoyo que siempre me ha brindado, con lo cual, la realización de esta tesis ha sido posible.

He Ning

Septiembre 30,1985

Ciudad Universitaria, México, D.F.

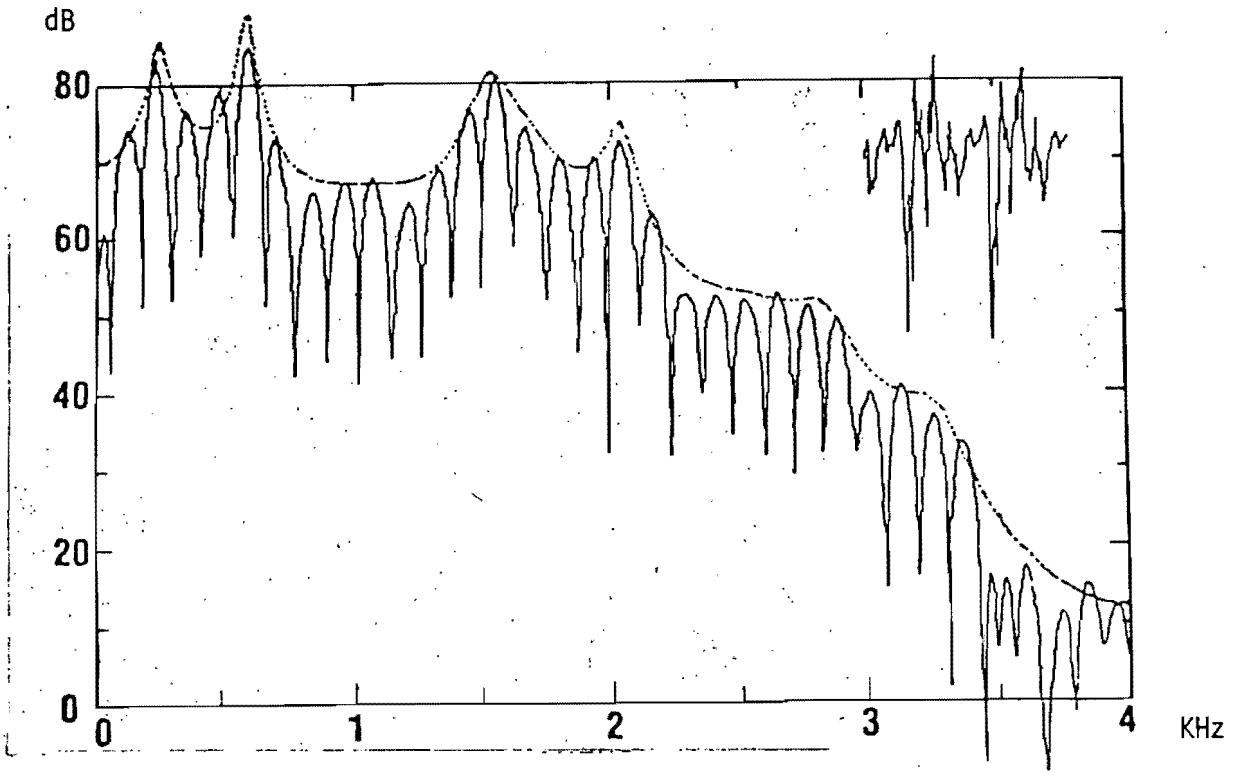
1

INTRODUCCION

Un tópico importante en diversas aplicaciones relacionadas con la voz es la codificación de esta señal. El objetivo de codificar la voz es el de reducir la tasa de estas señales, manteniendo cierto nivel de fidelidad necesario, ya sea para su transmisión sobre canales de comunicación o su almacenamiento en memorias. Esto es factible ya que la voz contiene considerable redundancia debida a las características y restricciones de nuestro aparatos de producción y de percepción de la voz, y a las estructuras inherentes del lenguaje.

En términos generales, la voz es una señal localmente estacionaria, algunas características cambian "lentamente" con respecto a la frecuencia de la señal misma, y posee un espectro de energía bien definido en una duración corta, típicamente de 10 a 50 ms (milisegundos). Durante un periodo largo (varios segundos, por ejemplo), se pueden observar segmentos de señal de diferente naturaleza que corresponden a diferentes patrones del espectro. En la Fig.1 se muestran dos patrones típicos del espectro y sus correspondientes señales en el tiempo. Por otra parte, la teoría de la percepción nos sugiere que el espectro de energía de duración corta juega un papel fundamental en la percepción, porque el oído humano actúa, de alguna manera, como un analizador de espectro o un banco de filtros [7,8].

Los sistemas de compresión de voz, o simplemente codificadores



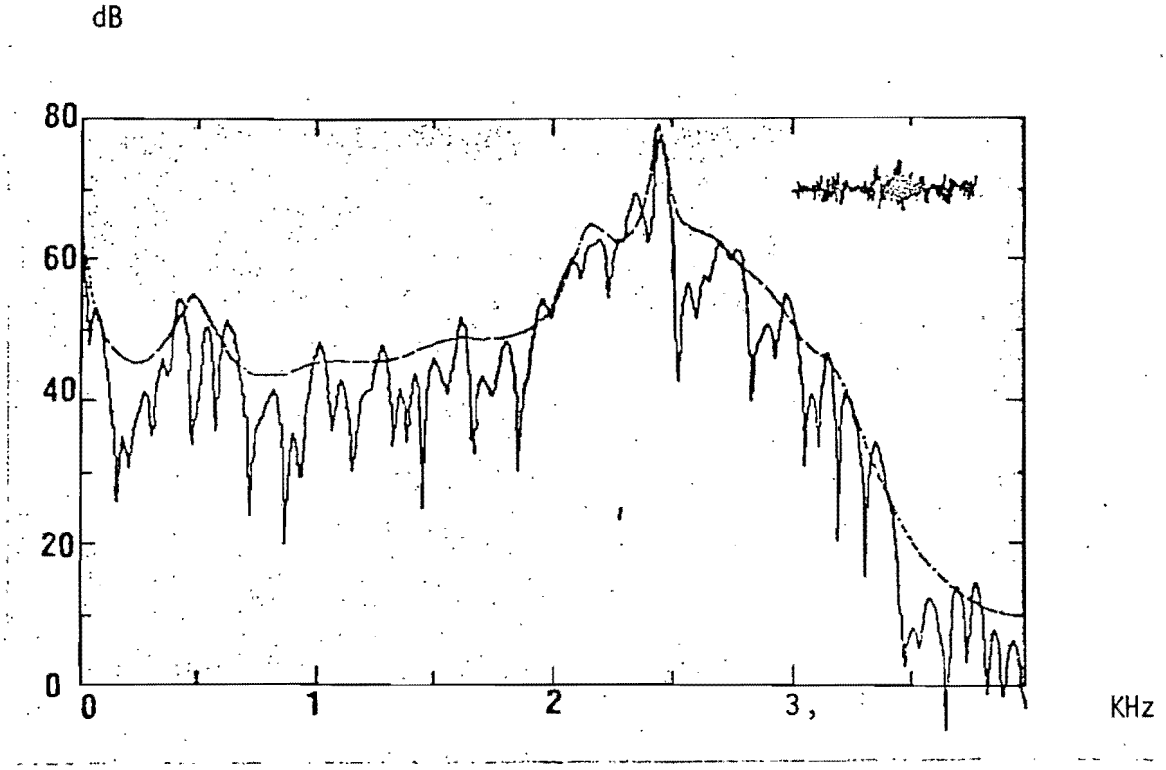


Fig.1 Dos espectros típicos de energía y sus correspondientes señales de voz.

(coders), son los medios que realizan la codificación y están caracterizados por tres factores fundamentales: tasa de bit, distorsión, y complejidad. Idealmente, un sistema de compresión debe minimizar las tres figuras de mérito conflictivas; esto es, se desea diseñar codificadores que eliminen eficientemente las redundancias de la señal de voz sin introducir demasiado ruido de cuantización y manteniendo un mínimo requerimiento computacional (tiempo de procesamiento y memoria). En la actualidad, la tecnología digital se ha desarrollado un cierto nivel, que es posible diseñar sistemas para procesamiento digital de señales considerablemente complejos, es de gran interés diseñar e implementar codificadores (generalmente complejos) que produzcan voz de calidad telefónica ("toll" quality, equivale a Log-PCM a 64 Kb/s) a tasas entre 4.8-9.6 Kb/s.

Tradicionalmente, los sistemas de compresión de voz han sido clasificados en dos categorías amplias: codificadores de formas de onda (waveform coders) y codificadores de voz (voice coders o vocoders). Los codificadores de formas de onda intentan obtener la aproximación (versión cuantizada) más parecida a la señal original (en principio, no está restringida a la señal de voz, puede ser música, por ejemplo) de acuerdo a algún criterio de fidelidad, mientras que los codificadores de voz intentan modelar el proceso de producción de voz mediante algún modelo paramétrico y generar voz sintética usando tal modelo. Los ejemplos más conocidos son los sistemas PCM y sus versiones diferencial y adaptiva, los sistemas de codificación por predicción lineal (LPC), etc. Generalmente, los codificadores de formas de onda son más exitosos para reproducir voz de buena calidad (de comunicación o telefónica) a tasas de bit mayores de 16 Kb/s. Por otro lado, los codificadores de voz son menos robustos, pero son capaces de lograr alto grado de compresión,

producen voz de calidad sintética a tasas entre 2 y 5 Kb/s. La Fig.2 muestra a grosso modo la calidad subjetiva como función de las tasas que pueden alcanzarse con ambos tipos de codificadores [8]. Es informativo observar que en la región de 3 a 10 Kb/s, ni los codificadores de voz ni los codificadores de formas de onda pueden producir voz de alta calidad; por lo tanto, el problema de diseñar sistemas de compresión que produzcan voz de buena calidad a una tasa entre 4.8 y 9.6 Kb/s ha sido la meta principal de nuestra investigación.

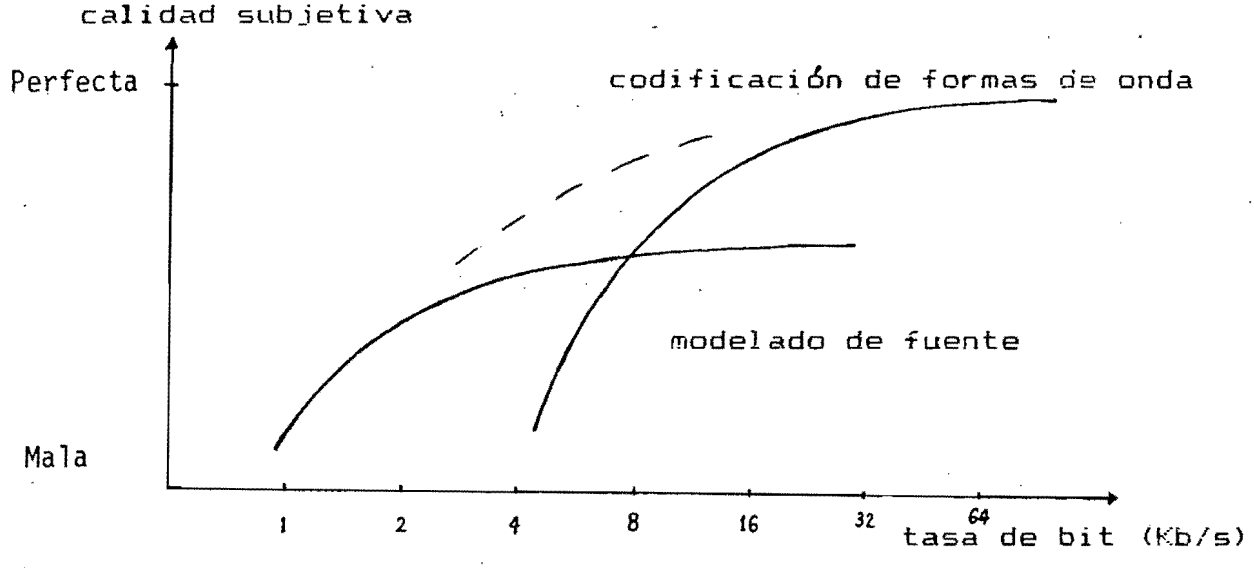


Fig.2

La clase de codificadores de formas de onda se puede separar nuevamente en dos tipos dependiendo del dominio donde se realiza la cuantización: métodos en el tiempo y en la frecuencia. La razón más importante de codificar una representación adecuada en el dominio de la frecuencia en lugar de codificar la señal misma, es el hecho de que la preservación de la estructura del espectro de duración corta (el cual puede describirse convenientemente mediante la transformada discreta de Fourier de duración corta) es esencial para lograr una buena calidad

perceptual de reproducción. Específicamente, las sub-bandas en donde se concentra la información de frecuencia fundamental (pitch) y de resonancias (formants) deben ser cuantizadas con mayor exactitud; en cambio, en las porciones de alta frecuencia donde ocurre sonido de tipo ruido, se puede permitir un nivel relativamente alto de ruido de cuantización, puesto que el ruido de cuantización es eficientemente "enmascarado" por estos sonidos [1,7,8]. En otras palabras, el método en la frecuencia nos ofrece la oportunidad de cuantizar la señal de voz de manera más selectiva y flexible, y consecuentemente un posible mejoramiento de la calidad de reproducción. Dos ejemplos típicos son, la codificación por sub-bandas y la codificación por transformación [15]. El primero divide el espectro de voz en un conjunto de 4 a 8 sub-bandas contiguas por medio de un banco de filtros paso-bandas (las fugas espectrales son despreciables), mientras que el segundo divide la señal en bloques consecutivos (64-512 muestras) y calcula la transformada discreta de Fourier (DFT) u otras transformadas para cada bloque de señal (con posible traslape entre los bloques consecutivos, pero el traslape en frecuencia predomina). Ambas técnicas cuantizan su respectiva representación espectral escalarmente, y los espectros resultantes son usados para sintetizar una señal en tiempo ya sea por un sintetizador o por tranformación inversa.

Tradicionalmente, los sistemas de compresión usan exclusivamente técnicas de cuantización escalar (muestra por muestra). Durante los últimos años, sin embargo, la técnica de cuantización vectorial (VQ) ha ténido un gran impacto en el campo de codificación de voz. la LPC basada en VQ ha logrado una reducción considerable de la tasa de bit [2]. Así mismo, la VQ ha sido aplicada para diseñar codificadores de formas de onda en el dominio del tiempo [3,4,5]; sin embargo, a nuestro

conocimiento, existe poco trabajo publicado sobre la aplicación de VQ a codificación en el dominio de la frecuencia [9]. Como consecuencia, nos dedicaremos a investigar este tópico en particular.

La idea fundamental de la VQ está basada en un resultado básico de la teoría de tasa-distorsión (rate distortion theory, es la rama de la teoría de información que estudia los fundamentos de la compresión de datos), la cual nos indica que la máxima compresión no se alcanza con cuantización escalar sino vectorial (con vectores de dimensión arbitrariamente grande, que son tratados como un solo objeto), aunque las componentes de los vectores sean estadísticamente independientes! Entonces, no es sorpredente que una reducción significativa de la tasa puede obtenerse cuantizando una representación adecuada de voz, bloque por bloque (de tamaño razonable), en lugar de cuantizar muestra por muestra.

Desgraciadamente, el uso directo de VQ tiene una seria dificultad: requiere enorme esfuerzo computacional tanto para su diseño como para su operación. Se necesitan almacenar voluminosos vocabularios de búsqueda total (full searched codebooks) y enorme velocidad de operación para realizar el cálculo de distorsión y búsqueda del mejor vector de reproducción. Por ejemplo, un sistema de VQ que trabaja con vectores de 128 muestras (muestreada a 6.4 KHz) a tasa de bit igual a 6.4 Kb/s (1 bit/muestra) requiere almacenar y buscar un vocabulario de $2^{128} \cong 10^{38}$ diferentes vectores de reproducción (codewords)! Obviamente, el tiempo para accesar exhaustivamente el vocabulario es del mismo orden astronómico; de hecho, ni siquiera se puede almacenar en memorias de las máquinas contemporáneas (teóricamente se puede hacer despreciable el tiempo de procesamiento por sacrificar aún más la memoria: para cada posible vector de entrada, asignar un vector de reproducción

correspondiente y almacenar su índice. En el ejemplo anterior, suponga que cada muestra es de 12 bits, entonces existen $2^{1536} = 10^{462}$ posibles vectores de entrada y se necesita almacenar una tabla del mismo número de índices de 128 bits, además del vocabulario de vectores de reproducción anterior). Esta restricción es el principal obstáculo que nos prohibe diseñar sistemas con VQ de gran escala (tasa de bit y dimensión de vector grandes), lo cual implica una limitación fundamental del desempeño de sistemas prácticos que se pueden alcanzar en el contexto de la teoría de tasa-distorsión.

En la práctica, desde luego, los vocabularios más útiles no son los de búsqueda total, sino los que permiten su acceso eficientemente (por lo tanto, son subóptimos en el sentido de la teoría de tasa-distorsión) para criterios de distorsión razonablemente complejos. Esto implica que los vocabularios deben poseer suficiente estructura interna de naturaleza geométrica o algebraica; por ejemplo, los vocabularios que poseen estructura de arbol, los códigos permutacionales son de esta naturaleza. En esta tesis, sólo usamos un esquema de organización de vocabulario, el cual "factoriza" un vocabulario como un producto cartesiano de varios vocabularios (de búsqueda total) de menor dimensión y por tanto más rápidos de accesar; o bien, como un producto cartesiano de un vocabulario de escalares por un vocabulario de vectores normalizados.

El objetivo de la presente tesis es desarrollar un método de diseño de codificador de formas de onda, basado en cuantización vectorial de la transformada discreta de Fourier de duración corta (DSTFT), y aplicarlo para diseñar un sistema de compresión de voz con "buena calidad de comunicación" a la tasa de 6.4 Kb/s, mediante simulación en una minicomputadora. En el siguiente capítulo nos

dedicamos al estudio de la DSTFT considerando los aspectos más importantes para la realización de nuestro sistema de compresión. En el capítulo 3 se describen los cuantizadores vectoriales de tipo producto cartesiano que cuantizan la DSTFT con el criterio de mínimo error cuadrático. En el capítulo 4, consideramos un diseño práctico de sistema de compresión de voz y presentamos los resultados experimentales obtenidos mediante su simulación. Finalmente, en el último capítulo concluimos el estudio con algunos comentarios y recomendaciones para investigación futura.

2

ANALISIS Y SINTESIS DE VOZ MEDIANTE LA TRANSFORMADA DISCRETA DE FOURIER DE DURACION CORTA

CONCEPTOS BASICOS E INTERPRETACION DE LA DSTFT

El análisis de una señal se refiere a la descomposición de la señal en un conjunto de componentes en el domonio de la frecuencia y la síntesis es la reconstrucción de una señal usando su representación espectral obtenida mediante el análisis. Tanto el análisis como la síntesis pueden llevarse a cabo mediante técnicas de procesamiento digital de señales. Un esquema general de un sistema de análisis y síntesis se muestra en la Fig.1.

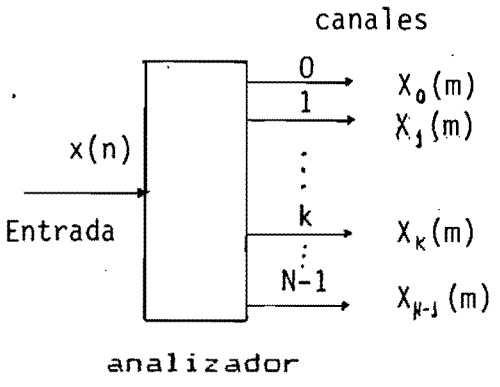


Fig. 1a

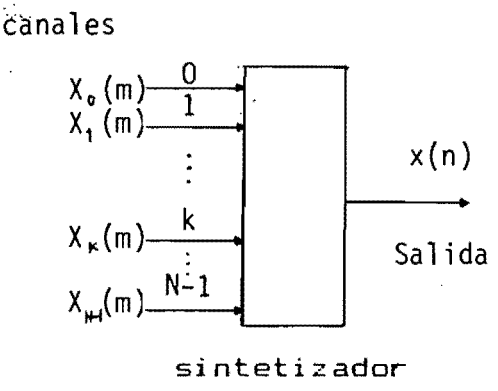


Fig. 1b

En el analizador (Fig.1a), la señal de entrada x(n) es convertida en un conjunto de señales de canal (generalmente complejas), $X_k(m)$, k = 0,1,...,N-1, que representan diferentes componentes de la señal original

en alguna región en el dominio de la frecuencia (sub-banda). Para cada m fijo, el vector $[X_0(m),\ldots,X_{N-1}(m)]$ se conoce como espectro de la señal x(n) en n=mM. En el sintetizador (Fig.1b), las señales de canal (quizás ya procesadas) $\hat{X}_K(m)$ son usadas para reproducir una señal sintética $\hat{x}(n)$. Una manera útil de definir las señales de canal está basada en la teoría de la transformada discreta de Fourier de duración corta (DSTFT). A continuación estudiaremos brevemente algunos aspectos más importantes de la DSTFT.

La DSTFT de una señal (real) x(n) está definida por:

$$X_K(m) = \sum_{n=-\infty}^{\infty} h(mM-n)\, x(n)\, e^{-j\frac{2\pi}{N}nK} \qquad k = 0,1,\ldots,N-1 \qquad (1)$$

y la fórmula general de síntesis es:

$$\hat{x}(n) = (1/N) \sum_{K=0}^{N-1} \sum_{m=-\infty}^{\infty} f(n-mM)\, \hat{X}_K(m)\, e^{j\frac{2\pi}{N}nk} \qquad (2)$$

donde h(n) y f(n) son llamados filtros o ventanas de análisis y de síntesis, respectivamente, y están relacionados a través de la siguiente condición de reconstrucción exacta:

$$\sum_{m=-\infty}^{\infty} h(m\,M - n + k\,N)\, f(n - m\,M)$$

$$= 1, \text{ si } k = 0;$$

$$= 0, \text{ si } k \neq 0 \qquad \text{para todo } n \qquad (3)$$

bajo esta condición, la señal sintética $\hat{x}(n)$ es exactamente igual a la señal de entrada x(n), siempre y cuando $\hat{X}_k(m) = X_k(m)$. La prueba de (3) se dará posteriormente al presentar un algoritmo eficiente para calcular la DSTFT.

Las ecuaciones anteriores pueden interpretarse mediante bancos de filtros o simplemente el esquema que calcula la transformada discreta de Fourier (DFT) de señales segmentadas por ventanas. En la Fig.2a tenemos el analizador correspondiente a (1) mientras la Fig.2b muestra el sintetizador correspondiente a (2). El número de canales del sistema es N, y M (M < N) representa el periodo de decimación (muestreo) del espectro, es decir, en cada periodo de M muestras de la señal, el analizador produce un espectro de N componentes complejas (contiene también exactamente N diferentes números reales). Si M = N, la representación espectral de la señal no contiene redundancia comparada con la señal misma, por lo que es el caso más interesante para diseño de sistemas de compresión.

La interpretación alterna se ilustra en la Fig.3. Se segmenta la señal de entrada en bloques sucesivos (con o sin traslape) por una ventana centrada en n = m M y se tranforma mediante la DFT obteniendo sucesivamente los espectros (N puntos, N > M, en n = m M). Esta última interpretación da origen al nombre de DSTFT. Note que los filtros de análisis y de síntesis usados en la práctica son de respuesta a impulso finita, pero la teoría general no está restringida a usar solo filtros de este tipo.

CRITERIO DE DISEÑO DE FILTROS DE ANALISIS Y DE SINTESIS

Dos casos extremos son de gran interés. Es facil de comprobar que los filtros paso-bajas ideales

$$h(n) = f(n) = \mathrm{sinc}(n/N)$$
$$= \sin(\pi n/N)/(\pi n/N), \text{ si } n \neq 0;$$
$$= 1, \qquad \text{si } n = 0 \qquad (4)$$

satisfacen la condición de reconstrucción exacta (3). En este caso, los

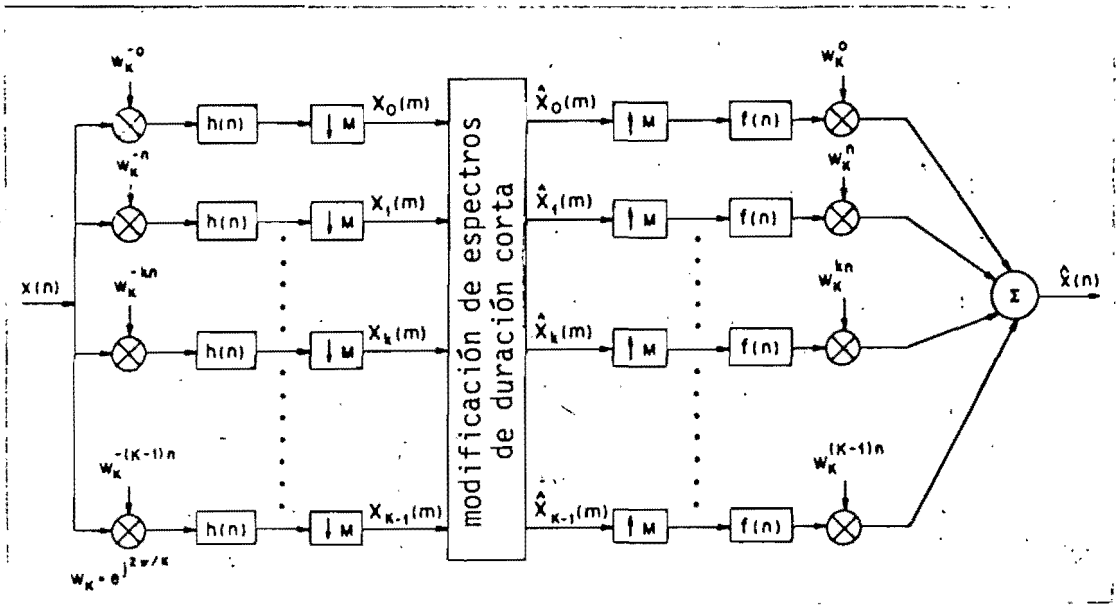


Fig.2 Modelo de bancos de filtros con modulación compleja.

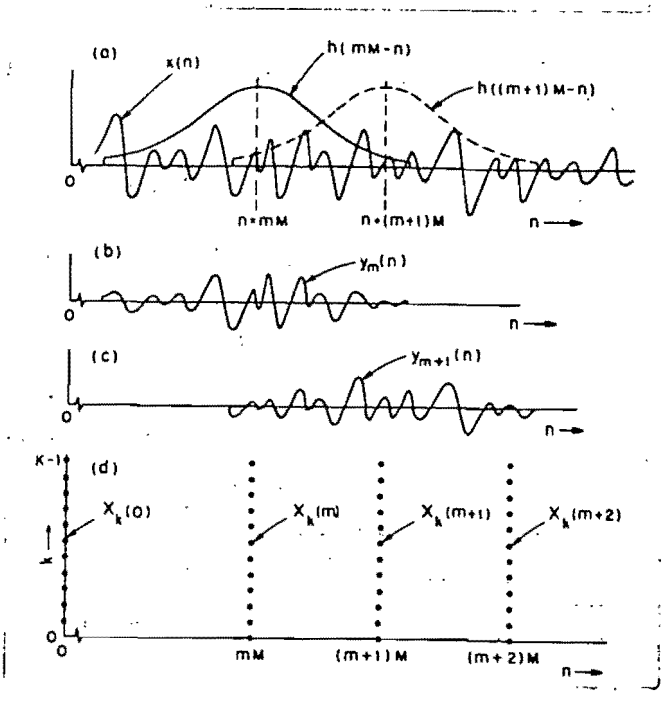


Fig.3 Interpretación de transformada de bloques consecutivos.

espectros obtenidos tienen una propiedad deseable, teóricamente, las diferentes componentes no contienen error de traslape; sin embargo estos filtros ideales sólo pueden ser aproximados en las implementaciones prácticas.

El otro caso extremo corresponde al método de transformada (DFT) de bloques sucesivos (sin traslape) usando ventana rectangular de longitud N:

$$h(n) = f(n) = 1, \text{ si } |n| \leq N/2;$$
$$= 0, \text{ si } |n| > N/2 \qquad (5)$$

Obviamente, la condición (3) se satisface ya que siempre podemos reconstruir la señal original por DFT inversa (IDFT). En este caso, no existe traslape de señal en el dominio del tiempo, sino en la frecuencia. Como la cuantización se llevará a cabo sobre las componentes de frecuencia, es preferible que los errores de cuantización en cada sub-banda sean aislados (minimizando el traslape en la frecuencia), ésta es una de las razones por las cuales usamos solamente los filtros paso-bajas ideales. Es posible diseñar filtros de análisis y síntesis que eliminan traslapes en ambos dominios, sin embargo, para lograr esto, se requiére que $M \leq N/4$, o sea, el espectro tiene un número de muestras por lo menos 4 veces mayor que el número de muestras de la señal misma [10].

El método más sencillo para diseñar los filtros de análisis y de síntesis es el método de ventanas:

$$h(n) = f(n) = \text{sinc}(n/N) \cdot W(n) \qquad (6)$$

donde $W(n)$ representa una ventana de longitud $L = 2 K N + 1$. Una elección razonable es la ventana de Hamming:

$$W(n) = 0.54 + 0.46 \cos(\pi n/KN) \qquad |n| \leq KN$$
$$= 0 \qquad |n| > KN$$

donde el parámetro K es usado para controlar la fuga espectral y debe de ser escogido de tal manera que satisfaga también la restricción de retraso del sistema.

Técnicas más sofisticadas para diseño de interpoladores pueden usarse para diseñar filtros que ofrecen mayor flexibilidad de operación [11,10].

METODO EFICIENTE DE IMPLEMENTACION USANDO EL ALGORITMO FFT

Una razón fundamental, por la cual la DSTFT ha tenido tantas aplicaciones en diversas áreas, es que su cálculo puede efectuarse eficientemente usando el algoritmo de transformada de Fourier rápida (FFT). Existen principalmente dos algoritmos equivalentes [10], el algoritmo que usamos es el siguiente [12].

Análisis: substituyendo $n = m\ M + l\ N + r$, $0 \leqslant r \leqslant N-1$ en (1), tenemos,

$$X_k(m) = e^{-j\frac{2\pi}{N}mMk} \sum_{r=0}^{N-1} \left[\sum_{l=-\infty}^{\infty} h(-lN-r)\ x(mM+lN+r) \right] e^{-j\frac{2\pi}{N}rk}$$

$$= e^{-j\frac{2\pi}{N}mMk} \left\{ \sum_{r=0}^{N-1} \tilde{x}_r(m)\ e^{-j\frac{2\pi}{N}rk} \right\} \qquad (7)$$

donde: $\tilde{x}_r(m) = \sum_{l=-\infty}^{\infty} h(-lN-r)\,x(mM+lN+r)$ (8)

es la secuencia obtenida usando el traslapé en tiempo (time-aliasing), y es periódica en r con periodo N. Nótese que la cantidad encerrada por { } en (7) tiene la forma de DFT, y por lo tanto se calcula eficientemente usando FFT. Para evitar la multiplicación compleja en (7) (cuando M<N) es suficiente recorrer circularmente la secuencia $\tilde{x}_r(m)$ y trabajar con

la secuencia recorrida:

$$X_K(m) = \sum_{n=0}^{N-1} x_n(m)\ e^{-j\frac{2\pi}{N}nk} \qquad (9)$$

donde: $x_n(m) = \tilde{x}_r(m), \quad n = (mM+r) \bmod N \qquad (10)$

Síntesis: la ecuacion (2) puede reescribirse como:

$$\hat{x}(n) = \sum_{k=-\infty}^{\infty} f(n-mM)\ \hat{x}_n(m) \qquad (11)$$

donde: $$\hat{x}_n(m) = (1/N) \sum_{k=0}^{N-1} \hat{X}_k(m)\ e^{j\frac{2\pi}{N}nk} \qquad (12)$$

la cual es la IDFT de $\hat{X}_k(m)$ y por tanto puede evaluarse eficientemente mediante FFT. Es importante interpretar la señal $\hat{x}_n(m)$ como una secuencia periódica con periodo N, cuando se calcula la señal sintética usando (11).

Obviamente, $\hat{X}_k(m) = X_k(m)$ implica que $\hat{x}_n(m) = x_n(m)$, usando (10) y (8), se tiene que

$$\hat{x}_n(m) = \sum_{k=-\infty}^{\infty} h(mM-n+kN)\ x(n-kN)$$

substituyendo en (11) e igualando a $x(n)$ obtenemos la condición de reconstrucción (3). En las Fig.5a-b se ilustran esquemáticamente los procedimientos de análisis y de síntesis.

Finalizamos este capítulo con la siguiente fórmula de Parseval. Sea $x(n)$ una señal de energía finita, entonces,

$$\sum_{n=-\infty}^{\infty} |x(n)|^2 = \sum_{m=-\infty}^{\infty} (1/N) \sum_{k=0}^{N-1} |X_K(m)|^2 \qquad (13)$$

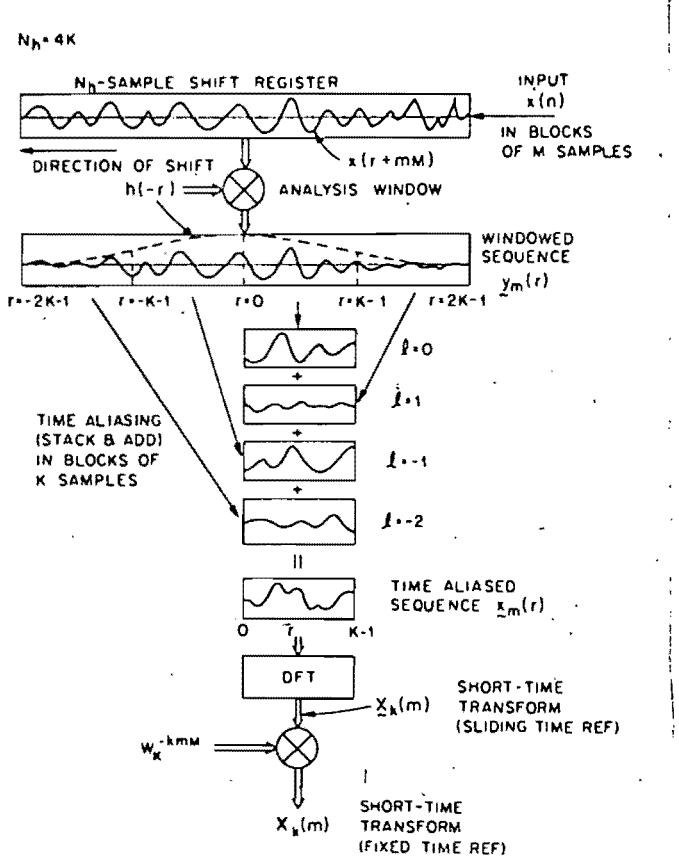


Fig.4 Secuencia de operaciones de análisis

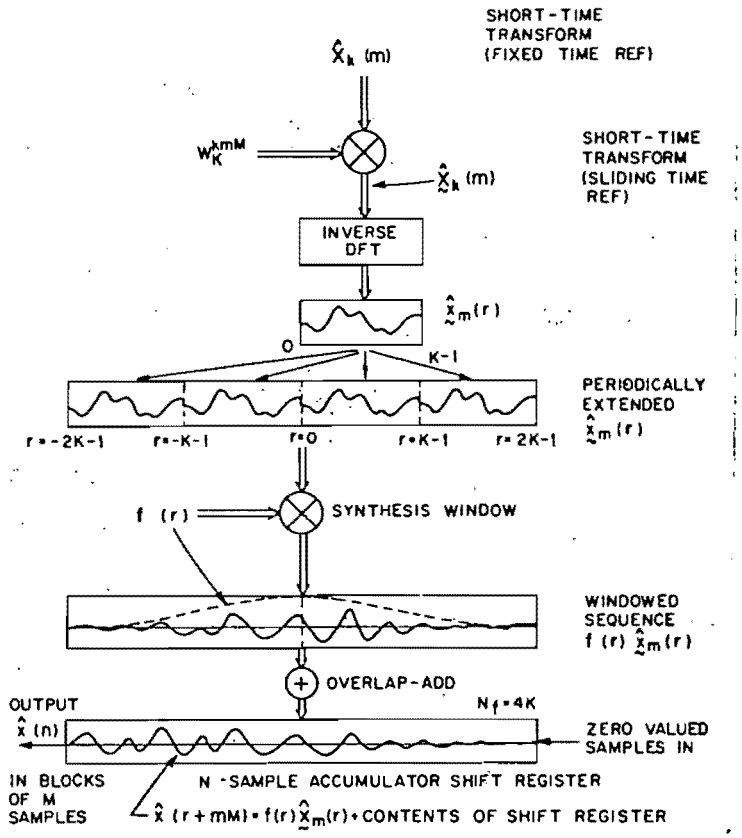


Fig.5 Secuencia de operaciones de síntesis

aquí el filtro de análisis está dado por (4) o (5). En el caso de (5), la fórmula anterior resulta obvia; en el caso de (4), la prueba está basada en la identidad [13]:

$$\sum_{k=-\infty}^{\infty} \text{sinc}[(kN-i)/N]\ \text{sinc}[(kN-j)/N] = \text{sinc}[(i-j)/N]$$

Una versión alterna de (13) es usada en el siguiente capítulo.

3

DISEÑO DE CUANTIZADOR VECTORIAL DE ESTRUCTURA DE PRODUCTO CARTESIANO

DESCRIPCION GENERAL DEL CUANTIZADOR VECTORIAL (VQ)

El funcionamiento de un VQ se puede describir mediante un modelo básico ilustrado en la Fig.1:

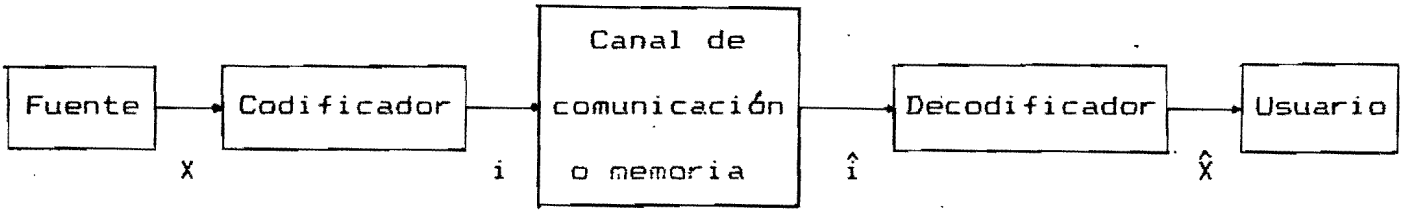


Fig.1

El modelo consiste de los siguientes módulos funcionales:

1) Una fuente de información que emite periódicamente vectores aleatorios, X, los cuales serán cuantizados.

2) Un codificador (encoder), el cual almacena un conjunto finito de vectores de reproducción (codewords) llamado vocabulario (codebook), y genera, como resultado de la cuantización, un índice para cada vector de entrada; o sea, el codificador busca en su vocabulario y determina el vector de reproducción más "cercano" al vector de entrada. Este vector representa entonces al vector original y es identificado por su índice. Note que el codificador es el dispositivo en donde ocurre la compresión.

3) Un canal de comunicación o medio de almacenamiento, mediante el cual, los índices generados por el codificador son transmitidos o almacenados.
4) Un decodificador (decoder), junto con un vocabulario (esencialmente el mismo que el del codificador) que convierte los índices en los correspondientes vectores de reproducción.
5) Un usuario quien recibe la información cuantizada.

El VQ está formado por el codificador y el decodificador y sus respectivos vocabularios. Específicamente, un analizador de voz produce un espectro de duración corta en cada segmento de análisis (frame). El codificador selecciona el patrón de espectro de su vucabulario más parecido al espectro de entrada de acuerdo a algún criterio de fidelidad, cuyo índice será eventualmente conocido por el decodificador. El decodificador simplemente convierte los índices a los vectores de reproducción correspondientes. Finalmente los espectros cuantizados son usados para reconstruir una señal sintética a través de un sintetizador.

Para diseñar un VQ, hay que especificar tres elementos principales:
1) Una medida cuantitativa de distorsión entre cada par de vector de entrada y vector de reproducción. Es una función no negativa que representa una penalización asociada al error introducido por reemplazar el vector de entrada por el vector de reproducción. Una vez establecida la distorsión d(X,Y), el codificador debe seleccionar el vector de reproducción Y, tal que minimice d(X,Y) para cada vector de entrada X dado. Idealmente la distorsión objetiva debe coincidir con el criterio subjetivo del usuario, en el caso de compresión de voz, una pequeña distorsión debe de corresponder a una alta calidad perceptual mientras gran distorsión implica baja calidad de la voz sintética. Desgraciadamente, todas las distorsiones comúnmente usadas en la

práctica, sólo reflejan parcialmente al criterio subjetivo, debido a la limitación de nuestro conocimiento de la verdadera naturaleza del oído humano - esto es un severo obstáculo que nos impide obtener voz sintética de alta calidad a baja tasa de bit.

2) Una adecuada estructura interna del vocabulario. Una estructura y una distorsión propiamente definidas nos permiten diseñar el procedimiento eficiente de clasificación o búsqueda que mapea cada vector de entrada al mejor vector de reproducción.

3) Un algoritmo de diseño mediante el cual se obtiene un vocabulario óptimo. Los algoritmos prácticos generalmente no requieren conocer las estadísticas de la fuente sino simplemente usan una secuencia de entrenamiento suficientemente rica. Todos los algoritmos existentes son procedimientos iterativos: empezando con un vocabulario arbitrario, en cada iteración genera un nuevo vocabulario de menor distorsión promedio; como la distorsión es no negativa, el procedimiento iterativo producirá un vocabulario óptimo (en general son óptimos locales).

Antes de concluir esta sección, presentamos una descripción formal de un VQ [6]. Sea $X = \{ X_k, k = 1,2... \}$ el cunjunto de todos los vectores de entrada (L-dimencional). Un VQ consiste en dos mapeos. Uno es el codificador:

$$q: X \longrightarrow I$$

donde $I = \{ i, i = 1,...,M \}$, es un conjunto de índices; el otro mapeo es el decodificador:

$$r: I \longrightarrow Y$$

donde $Y = \{ Y_i, i = 1,...,M \}$, es el vocabulario del VQ que contiene M vectores de reproducción (L-dimensional). La tasa del VQ está definida por:

$$R = \log_2 (M) \qquad \text{bits/vector}$$

o $$R = (1/L) \log_2 (M) \quad \text{bits/muestra}$$

Dada una medida de distorsión $d(X,Y) \geq 0$ definida para todo par de vector de entrada X y vector de reproducción Y, un VQ es óptimo si minimiza la distorsión promedio $E\{ d[X,r(q(X))] \}$ ($E\{ \}$ es el operador de esperanza matemática). Un VQ óptimo debe de tener las dos propiedades siguientes:

1) dado un decodificador r, el mejor codificador q es el que asigna $i = q(X)$ tal que $d(X,r(i))$ sea mínima, para todo X, ya que

$$\min_i E\{ d[X,r(i)] \} \geq E\{ \min_i d[X,r(i)] \}$$

El codificador q junto con el vocabulario Y define una partición del espacio de entrada (un conjunto de celdas):

$$P_i = \{ X \mid q(X) = i \}, \quad i = 1,2,\ldots,M$$

entonces,

$$X = \bigcup_{i=1}^{M} P_i \qquad \text{y} \qquad P_i \cap P_j = 0, \quad i \neq j$$

2) dado un codificador q y su partición asociada, el mejor decodificador r es el que asigna a cada índice i el centroide (generalizado) de la celda P_i ; esto es, $Y_i = r(i)$ tal que minimiza $E\{ d(X,Y) \mid X \in P_i \}$.

Estas propiedades forman la base de los algoritmos de diseño de los VQ que presentamos más adelante.

DEFINICION DE LA DISTORSION DE ERROR CUADRATICO PONDERADO

Existen muchas medidas de distorsión para diseñar sistemas de compresión de voz [15], quizá la distorsión más conocida y más sencilla es la de error cuadrático ponderado:

$$d(X,Y) = (1/N) \sum_{K=0}^{N-1} P_k |x_k - y_k|^2 \qquad (1)$$

donde X, Y son los vectores de entrada y de reproducción, respectivamente,

$$X = (x_0, \ldots, x_{N-1})$$

$$Y = (y_0, \ldots, y_{N-1})$$

y $P_k > 0$, $k = 0,1,\ldots N-1$, normalizados de manera que:

$$(1/N) \sum_{k=0}^{N-1} P_k = 1 \qquad (2)$$

son los pesos asociados a cada componente de X debido a su importancia relativa. Note que la distorsión (1) es una suma de varias distorsiones, cada una de ellas involucra sólo subvectores (de dimensión reducida) de X y de Y, esta propiedad será explotada posteriormente cuando estudiemos la estructura de vocabularios.

En general, las componentes de DSTFT son números complejos, es conveniente expresar la distorsión (1) en términos de partes real e imaginaria, o bien, magnitudes y fases:

$$d(X,Y) = (1/N) \sum_{k=0}^{N-1} P_k \left[(Re(x_k) - Re(y_k))^2 + (Im(x_k) - Im(y_k))^2 \right] \qquad (1a)$$

$$d(X,Y) = (1/N) \sum_{k=0}^{N-1} P_k \left[(|x_k| - |y_k|)^2 + 2|x_k||y_k| (1 - \cos(\measuredangle x_k - \measuredangle y_k)) \right] \qquad (1b)$$

también usamos una versión modificada de (1b):

$$d(X,Y) = (1/N) \sum_{k=0}^{N-1} P_k \left[(|x_k| - |y_k|)^2 + 2|x_k|^2 (1 - \cos(\measuredangle x_k - \measuredangle y_k)) \right] \qquad (1c)$$

Note que (1c) es una suma de dos distorsiones que involucra solo magnitudes y fases, respectivamente. Esta separación nos permite diseñar

y accesar los respectivos vocabulàrios de manera independiente, y en consecuencia, obtenemos un ahorro de esfuerzo computacional.

La distorsión (1) está definida directamente en términos de la DSTFT; no obstante, tiene una interpretación interesante en el dominio del tiempo. Supongase que los espectros de las señales x(n) e y(n) son, respectivamente:

$$X_k(m) = \sum_{n=-\infty}^{\infty} h(mN-n)\ x(n)\ e^{-j\frac{2\pi}{N}nk}$$

$$Y_k(m) = \sum_{n=-\infty}^{\infty} h(mN-n)\ y(n)\ e^{-j\frac{2\pi}{N}nk} \quad k = 0,1,\ldots,N-1$$

Defínase la señal de error

$$z(n) = x(n) - y(n) \tag{3}$$

debido a la linealidad de la DSTFT,

$$Z_k(m) = X_k(m) - Y_k(m)$$

donde $$Z_k(m) = \sum_{n=-\infty}^{\infty} h(mN-n)\ z(n)\ e^{-j\frac{2\pi}{N}nk} \quad k = 0,1,\ldots,N-1 \tag{4}$$

La distorsión puede reescribirse en función de la señal de error z(n):

$$d(X,Y) = (1/N) \sum_{k=0}^{N-1} P_k \left|Z_k(m)\right|^2$$

$$= (1/N) \sum_{k=0}^{N-1} P_k \sum_{i=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} h(mN-i)z(i)\ h(mN-l)z(l)\ e^{j\frac{2\pi}{N}(i-l)k}$$

$$= \sum_{i=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} z_m(i)\ p(i-l)\ z_m(l) \tag{5}$$

donde: $$p(i-l) = (1/N) \sum_{k=0}^{N-1} P_k\, e^{j\frac{2\pi}{N}(i-l)k} \tag{6}$$

es periódica con periodo N y positivamente definida; y

$$z_m(n) = h(mN-n)\, z(n) \qquad (7)$$

es la señal de error modificada por la ventana centrada en n=mN, que representa al error de cuantización (de duración corta) alrededor de n=mN. Entonces, minimizar la suma ponderada de las magnitudes de la DSTFT de error de cuantización equivale a minimizar el miembro derecho de (5) que representa (en cierto sentido) la energía de la señal de error alrededor de n = mM. Si además z(n) es ruido blanco:

$$\begin{aligned} E\{z(n)z(m)\} &= E\{|z(n)|^2\}, && n = m; \\ &= 0, && n \neq m \end{aligned} \qquad (8)$$

y suponemos que el filtro de análisis es ideal,

$$\begin{aligned} h(n) &= \mathrm{sinc}(n/N) \\ &= \sin(\pi n/N)/(\pi n/N), && n \neq 0; \\ &= 1, && n = 0 \end{aligned} \qquad (9)$$

entonces,

$$\begin{aligned} E\{d(X,Y)\} &= p(0)\, E\{|z(n)|^2\} \sum_{i=-\infty}^{\infty} |h(mN-i)|^2 \\ &= E\{|z(n)|^2\} \end{aligned} \qquad (10)$$

Aquí hemos usado (2) y la identidad [13]:

$$\sum_{n=-\infty}^{\infty} |\mathrm{sinc}(n/N)|^2 = 1$$

Una manera más eficiente de calcular la distorsión (1) cuando $P_k = 1$, $k = 0,1,\ldots,N-1$ es calcular $\sum_{n=0}^{N-1} |\tilde{x}_n(m) - \tilde{y}_n(m)|^2$, aquí $x_n(m)$ e $y_n(m)$ son las respectivas IDFT de $X_k(m)$ e $Y_k(m)$. Entonces podemos eliminar el cálculo de FFT [para obtener $X_k(m)$]. Desgraciadamente, si queremos dividir todo espectro en varias sub-bandas y cuantizarlo con VQ de estructura de producto cartesiano, no podemos aprovechar esta propiedad.

VOCABULARIOS DE ESTRUCTURA DE PRODUCTO CARTESIANO

Un esquema para construir un vocabulario grande a partir de una serie de vocabularios de menor dimensión y de menor tamaño es como sigue.

Sean Y_k, $k = 1,2,\ldots,M$, M vocabularios de dimensión L_K y tamaño N_K. Aquí, la dimensión se refiere al número de componentes de los vectores y el tamaño se refiere al número de vectores en el vocabulario. Definimos un nuevo vocabulario como el siguiente producto cartesiano de M subvocabularios Y_k:

$$Y = Y_1 \times Y_2 \times \ldots \times Y_M \qquad (11)$$

Obviamente, Y es de dimensión

$$L = \sum_{k=1}^{M} L_K \qquad (12)$$

y de tamaño

$$N = \prod_{k=1}^{M} N_K \qquad (13)$$

La utilidad de la descomposición en (11) radica en que si la distorsión puede expresarse como una suma de distorsiones que involucran solo vectores de menor dimensión. Específicamente, sea:

$$X = (X_1, \ldots X_M)$$

un vector de entrada con dimensión L dividido en M subvectores de L_k componentes cada uno e Y un vector de reproduccion en Y. Supongamos además que

$$d(X,Y) = \sum_{k=1}^{M} d_k(X_K, Y_k) \qquad (14)$$

donde Y_k, en Y_k, son vectores de dimensión L, $k = 1,2,\ldots,M$; los vectores de reproducción se forman concatenando los vectores Y_k, en orden apropiado:

$$Y = (Y_1, \ldots, Y_M)$$

Bajo las condiciones (11), (14), la búsqueda del vector de reproducción

Y en $\mathcal{Y}$ que minimiza la distorsión $d(X,Y)$, dado el vector de entrada X, es equivalente a determinar los M vectores Y_k en $\mathcal{Y}_k$ que minimizan $d_k(X_k, Y_k)$, respectivamente.

Para estimar el ahorro de esfuerzo computacional que se puede obtener usando los vocabularios con estructura de producto cartesiano, empleamos una medida de complejidad de un vocabulario $\mathcal{Y}$ de acceso secuencial de dimensión L y de tamaño N, definida como:

$$f(\mathcal{Y}) = L\,N = \left(\sum_{k=0}^{M} L_k\right)\left(\prod_{k=1}^{M} N_k\right) \qquad (15)$$

Esta medida de complejidad es proporcional al número de multiplicaciones y de comparaciones que se requieren para determinar el vector de reproducción óptimo en $\mathcal{Y}$ mediante búsqueda total, también es proporcional al número de localidades de memoria requeridas para su almacenamiento. Si $\mathcal{Y}$ ya está en la forma factorizada en (12), la complejidad es entonces la suma de las complejidades de los M vocabularios componentes $\mathcal{Y}_k$:

$$f(\mathcal{Y}_1 x \ldots x \mathcal{Y}_M) = \sum_{k=1}^{M} f(\mathcal{Y}_k) = \sum_{k=1}^{M} L_k N_k \qquad (16)$$

Evidentemente, el vocabulario con estructura de producto cartesiano de la misma tasa [$R = (1/L)\log_2(N)$ bits/muestra] tiene una complejidad mucho menor:

$$\sum_{k=1}^{M} L_k N_k \ll \left(\sum_{k=1}^{M} L_k\right)\left(\prod_{k=1}^{M} N_k\right), \qquad \text{si} \quad N_k \gg 1,\ k = 1,\ldots,M$$

Por ejemplo, si $M = 8$, $L_k = 8$, $N_k = 256$, $k = 1,2,\ldots,8$; $L = 64$, $N = 65536$. El vocabulario sin estructura interna tiene complejidad igual a 4194304,

mientras que la complejidad del vocabulario de estructura de producto cartesiano es de 16384. El ahorro es impresionante: 256 veces más rápido y 256 veces menos memoria. Por esta razón los vocabularios con estructura interna son de gran utilidad práctica, a pesar de que son inherentemente subóptimos.

Otro esquema interesante de organización de vocabulario es el producto cartesiano de un vocabulario de escalares por un vocabulario de vectores (reales) propiamente normalizados, los vectores de reproducción se forman multiplicando cada vector normalizado (forma) por un número no negativo (ganancia); los vocabularios son llamados vocabularios de ganancia y de forma (shape), respectivamente. Los VQ que usan este tipo de vocabularios son comúnmente conocidos como VQ de ganancia y forma (GS-VQ) [5,6]. Específicamente, sea G un vocabulario de ganancia,

$$G = \{ g_i > 0,\ i = 1,2,\ldots,M \}$$

y $\mathcal{S}$ un vocabulario de forma,

$$\mathcal{S} = \{ S_j = (s_{j0},\ldots s_{jL-1}),\ j = 1,2,\ldots,N \}$$

donde los vectores S_j están normalizados adecuadamente dependiendo de la medida de distorsión escogida. Un vertor de reproducción se forma multiplicando:

$$Y = g \cdot S = (g \cdot s_0,\ldots g \cdot s_{L-1}) \qquad (18)$$

Supóngase que la distorsión es la de error cuadrático ponderado:

$$d(X,Y) = (1/L) \sum_{k=0}^{L-1} P_k\ (x_k - y_k)^2 \qquad (19)$$

donde $P_k > 0$, $k = 0,1,\ldots,L-1$, son los coeficientes de ponderación; entonces, dado un vector de entrada:

$$X = (x_0,\ldots,x_{L-1})$$

se desea determinar una ganancia g^* y una forma S^* tales que $Y^* = g^* S^*$

minimice la distorsión (19).

Con objeto de determinar un procedimiento eficiente de clasificación, reescribimos (19) como:

$$d(X,Y) = d(X, g\cdot S)$$

$$= (1/L) \sum_{k=0}^{L-1} P_k (x_k - g\cdot s_k)^2$$

$$= (1/L) \sum_{k=0}^{L-1} P_k |x_k|^2 - 2g(1/L) \sum_{k=0}^{L-1} P_k s_k x_k + g^2 (1/L) \sum_{k=0}^{L-1} P_k |s_k|^2$$

Esta última expresión nos sugiere la siguiente normalización de los vectores en el vocabulario de forma:

$$(1/L) \sum_{k=0}^{L-1} P_k |s_k|^2 = 1 \qquad (20)$$

Definiendo

$$\tilde{g} = (1/L) \sum_{k=0}^{L-1} P_k s_k x_k \qquad (21)$$

obtenemos la expresión deseada:

$$d(X, g\cdot S) = (1/L) \sum_{k=0}^{L-1} P_k |x_k|^2 - 2 g \tilde{g} + g^2 \qquad (22)$$

Un procedimiento eficiente de clasificación es entonces,

1) seleccionar un S^* en $\mathcal{S}$ tal que maximice $\tilde{g}$;

2) dado $\tilde{g}$ obtenido en 1), seleccionar g^* en $\mathcal{G}$ tal que minimice

$$g^2 - 2 g \tilde{g}$$

Para lograr mayor eficiencia, las ganancias se ordenan de menor a mayor y se incluye un conjunto de umbrales en el vocabulario de ganancia; entonces el paso 2) del procedimiento anterior puede llevarse a cabo eficientemente mediante la comparación de los umbrales. Específicamente,

suponemos que

$$0 < g_1 < \ldots < g_i < g_{i+1} < \ldots < g_M$$

si $u_i \leq \tilde{g} < u_{i+1}$, entonces, $g^* = g_i$; aquí u_k, $k = 1,2,\ldots,M-1$, son los umbrales que satisfacen las ecuaciones siguientes:

$$g_i - 2\, g_i\, u_i = g_{i+1} - 2\, g_{i+1}\, u_i \qquad (23)$$

o bien, $$u_i = (g_i + g_{i+1})/2 \qquad (23a)$$

La complejidad de un GS-VQ es:

$$f(G \times S) = M + L \cdot N \qquad (24)$$

mientras que un vocabulario de búsqueda total Y, de la misma tasa

$$R = (1/L)\ \log_2(MN)$$

tiene la complejidad

$$f(Y) = L \cdot M \cdot N \qquad (25)$$

siendo L, la dimensión de los vectores normalizados, M, el tamaño del vocabulario de ganancia y N, el tamaño del vocabulario de forma. Nuevamente, la comparación favorece al GS-VQ:

$$M + L \cdot N << L \cdot M \cdot N \quad \text{si } M >> 1$$

y el factor de ahorro es proximadamente igual a M. Por ejemplo, si L =16, M = 32, y N = 256, f(GxS) = 4128, y f(Y) = 131072; el factor de ahorro es entonces 31.75.

ALGORITMOS DE DISEÑO DE VQ [5,6]

Todos los algoritmos existentes de diseño utilizan directa o indirectamente el siguiente procedimiento iterativo basado en las propiedades de VQ (óptimos) discutidas anteriormente. Consideremos el siguiente algoritmo básico:

0) Dados una secuencia de entrenamiento (un conjunto finito de vectores de entrada, L-dimensional), $X = \{ X_m,\ m = 1,2,\ldots,M \}$ y un vocabulario inicial arbitrario, $Y = \{ Y_n,\ n = 1,2,\ldots,N \}$, donde $M >> N$.

1) Codificar toda la secuencia de entrenamiento X usando el vocabulario Y y la regla de clasificación:

$$q(X) = i \quad \text{si } d(X,Y_i) = \min_j d(X,Y_j) \text{ para todo } X \text{ en } \mathbf{X}.$$

2) Reemplazar el viejo vocabulario por un nuevo cuyo i-ésimo vector de reproducción es el centroide de la subsecuencia de entrenamiento que fue codificada en i:

$$r(i) = \text{centroide de } \{ X \,|\, q(X) = i \}$$

3) Repetir los pasos 1) y 2) hasta que el decremento de la distorsión promedio:

$$D = E \{ d[X,r(q(X))] \}$$

sea suficientemente pequeño. Donde la esperanza matemática es frecuentemente reemplazada por la media muestral:

$$\bar{D} = (1/M) \sum_{m=1}^{M} d[X_m, r(q(X_m))]$$

el cual corresponde a la distribución de probabilidad $Pr\{X_m\} = 1/M$, $m = 1,\ldots,M$.

Obviamente este procedimiento siempre termina, porque después de cada iteración la distorsión promedio será menor que la de la iteración anterior y convergerá a un valor mínimo (generalmente mínimo local).

Existen dos problemas prácticos. Uno es la generación de un vocabulario inicial, una técnica que da buenos resultados es empezar con un vocabulario pequeño, y agregar una versión perturbada de cada vector de reproducción al vocabulario (óptimo) de tamaño N para formar un nuevo vocabulario de tamaño 2N, y usar el algoritmo básico para obtener uno óptimo; así se pueden construir sucesivamente vocabularios de mayor tamaño. El otro problema es el tratamiento de celda vacía, esto es, cuando uno o más conjuntos $P_k = \{ X \,|\, q(X) = k \}$ es vacío, se puede

resolver este problema buscando el índice i correspondiente a una celda de mayor distorsión, $E\{ d(X,Y_i) \mid q(X) = i \} = \max_j E\{ d(X,Y_j) \mid q(X) = j \}$ y asignando una versión ligeramente modificada de Y_i a Y_k; este método garantiza que la distorsión promedio siempre vaya decreciendo.

Por supuesto, el algoritmo básico (y consecuentemente todos los algoritmos de diseño que incorporan este) es muy ineficiente, el tiempo de ejecución de tan solo una iteración es de orden LMN, siendo L, la dimensión de los vectores, M, la longitud de la secuencia de entrenamiento y N el tamaño del vocabulario. Afortunadamente, el VQ de un sistema de compresión especifico sólo se diseña una vez.

Consideremos el siguiente problema de diseño del PC-VQ óptimo. Se desea diseñar un PC-VQ con un vocabulario Y de dimensión L y tamaño N, compuesto de M subvocabularios Y de (menor) dimensión L_k y tamaño N_k por el producto cartesiano (11),(12) y (13). Debido a la restricción de recursos de computación, es necesario que todos los Y_k tengan una complejidad limitada:

$$L_k N_k \leq C, \qquad k = 1,\ldots,M \qquad (25)$$

donde C representa cuantitativamente la limitación del recurso de computación. Además, se desea que el PC-VQ tenga un mínimo número de subvocabularios, es decir, se requiere minimizar M sujeto a la restricción (25), porque ituitivamente, un M mínimo debe corresponder a una distorsión promedio razonablemente menor (en el capítulo 4 presentamos un método particular para determinar estos parámetros "óptimos"). Una vez obtenidos los parámetros L_k , N_K , diseñamos M subvocabularios Y_k , k = 1,...M, independientemente haciendo uso del algoritmo básico.

El problema de diseñar el GS-VQ óptimo es: Dada la restricción de complejidad C, se desea diseñar un GS-VQ óptimo de tasa R

(bits/vector), tal que

$$M + L\,N \leq C, \qquad M\,N = 2^{R} \qquad (26)$$

donde M es el número de ganancias y N es el número de vectores mormalizados (L-dimensional). En principio, podemos determinar primero todos los posibles pares M y N que satisfacen la restrición (26), para cada par diseñamos un GS-VQ óptimo mediante algún algoritmo y después escogemos un GS-VQ que tenga menor distorsión promedio.

Consideremos el algoritmo de optimización conjunta modificado:

0) Inicialización: Dados arbitrariamente un vocabulario de ganancia y un vocabulario de forma, $\mathcal{G}$ y $\mathcal{S}$ respectivamente, y una secuencia de entrenamiento $\mathcal{X}$.

1) Particionar $\mathcal{X}$ usando la regla de clasificación:

$$q_S(X) = i, \qquad \text{si } \tilde{g} = \sum_{K=0}^{N-1} P_K x_K s_{iK} = \min_{j} \sum_{K=0}^{N-1} P_K x_K s_{jK}$$

y $$q_G(\tilde{g}) = j, \qquad \text{si } g_j^2 - 2\tilde{g}g_j = \min_{l}(g_l^2 - 2\tilde{g}g_l)$$

2) Actualizar los vocabularios por

$$\mathcal{G} = \{\, g_j \mid g_j = \text{centroide de } \{\tilde{g} \mid q_G(\tilde{g}) = j\}\,\}$$

$$\mathcal{S} = \{\, S_i \mid S_i = \text{centroide de } \{X \mid q_S(X) = i\}\,\}$$

3) Repetir los pasos 1) y 2) hasta que el decremento de la distorsión promedio

$$D = E\{d[X, r(q_G(\tilde{g})\ q_S(X))]\}$$

sea suficientemente pequeño.

Nota: existen varios algoritmos para el diseño de GS-VQ, por ejemplo, el de optimización conjunta, el de optimización individual, etc [5]. Sin embargo, en esta tesis sólo vamos a usar el algoritmo anterior, el cual es una versión más eficiente del algoritmo de optimización conjunta.

Finalmente consideramos el cálculo de centroides. Sea $\mathcal{X} = \{\, X_i,\ i = 1,\ldots,N \,\}$ un conjunto de vectores, se desea determinar un vector X^* tal que minimice $E\{d(X, X^*) \mid X \in \mathcal{X}\}$. X^* depende de la distorsión

particular, si la distorsión es la de error cuadrático ponderado (1) y la esperanza matemática es interpretada como media muestral, se tiene que

$$X^* = (\bar{x}_0 , \ldots , \bar{x}_K , \ldots , \bar{x}_{L-1}) \qquad (27)$$

donde $\bar{x}_k = (1/N) \sum_{i=1}^{N} x_{iK}$, x_{iK} es el k-ésimo componente de X_i. Si se requiere que X^* esté normalizado de manera que

$$(1/L) \sum_{K=0}^{L-1} P_k \left| x^*_k \right|^2 = 1$$

entonces,

$$X^* = (\bar{x}_0 , \ldots , \bar{x}_{L-1}) \Big/ (1/L) \sum_{K=0}^{L-1} P_k \left| \bar{x}_K \right|^2 \qquad (28)$$

En el caso de la ganancia, el centroide se determina calculando el promedio aritmético de los $\tilde{g}$, para cada conjunto $\{\tilde{g} \mid q_G(\tilde{g}) = j\}$, y los umbrales se determinan mediante (23a).

Resumiendo, en este capítulo hemos considerado en términos muy generales los cuantizadores vectoriales, particularmente los PC-VQ y GS-VQ, describimos dos algoritmos de diseño, el algoritmo básico y el algoritmo de optimización conjunta modificado. También mostramos que los PC-VQ y GS-VQ ofrecen considerable ahorro computacional en comparación con los VQ de búsqueda total. No proporcionamos, sin embargo, ningún método general para especificar los parámetros de los PC-VQ y GS-VQ (tamaño y dimensión de los subvocabularios, etc), sino hasta el próximo capítulo, donde trataremos de "resolver" este problema dificil mediante un método ad hoc, y aplicar todas herramientas descritas al diseño e implementación de un sistema de compresión de voz.

4

SIMULACION Y RESULTADOS EXPERIMENTALES

EQUIPO Y SEÑALES UTILIZADOS PARA LA SIMULACION

Se dispone de una minicomputadora PDP-11/40 y un sistema de adquisición y reproducción de voz. La computadora tiene cerca de 32 K palabras de memoria principal, dos unidades de disco teniendo cada disco una capacidad de 2.2 megabytes y otras facilidades. El sistema de adquisición y reproducción de voz consiste principalmente de una grabadora, un amplificador, filtros analógicos y convertidores A/D, D/A. Hemos preparado dos señales digitalizadas de voz, almacenadas en disco. Ambas consisten de conversaciones en inglés de duración de 40 s cada una, son filtradas con un filtro analógico cuya banda de paso es 200-3200 Hz y digitalizadas a 6400 muestras/s y 12 bits/muestra, la tasa de estas señales es 76.8 Kb/s, y corresponde aproximadamente a una codificación de muy buena calidad telefónica. Una de las dos señales es una conversación entre dos hombres y la usamos como secuencia de entrenamiento; la otra hablada por los dos hombres y una mujer es usada como secuencia de prueba.

IMPLEMENTACION DE UN SISTAMA DE ANALISIS Y SINTESIS

En el capítulo 2 hemos ya estudiado algunos aspectos generales de sistemas de análisis y síntesis basado en la transformada discreta de Fourier de duración corta (DSTFT). Aquí presentamos una implementación

particular con las siguientes características.

1) La resolución del espectro es 64 componentes complejas por espectro (N=128).

2) La razón de decimación es la máxima (M=N), por lo que la representación en frecuencia no contiene redundancia comparada con la señal misma.

3) Los filtros de análisis y de síntesis son diseñados por el método de ventanas, la ventana usada es de Hamming de longitud 1281 (10x128+1) muestras. El sistema tiene entonces un retraso de aproximadamente 200 ms, el cual es relativamente grande comparado con retrasos de otros sistemas de compresión, sin embargo es aceptable en muchas aplicaciones. También disponemos de un programa para diseñar interpoladores de mínimo error cuadrático medio [11], estos interpoladores son más apropiados para implementar filtro de síntesis en el caso M<N; cuando usa decimación crítica (M=N), los filtros diseñados con ventana de Hamming y los interpoladores de mínimo error cuadrático medio dan resultados comparables.

4) El algoritmo FFT es implementado usando un programa para secuencias reales [14]. Si se dispone de un programa para secuencias complejas, es recomendable usarlo eficientemente calculando DFT de dos secuencias reales simultáneamente.

Nuestro sistema es aproximadamente un sistema identidad, es capaz de reproducir voz sintética perceptualmente indistinguible de la señal original cuando no hay modificación de los espectros. Para mayor detalle, presentamos los programas en FORTRAN IV que simulan el sistema de análisis y síntesis de voz, en el apéndice al final de la tesis.

DISEÑO Y REALIZACION DE UN CUANTIZADOR VECTORIAL (VQ)

Usamos dos esquemas para representar y cuantizar los espectro complejos: por partes real e imaginaria o por magnitudes y fases. Bajo la suposición de que la distorsión escogida es la de error cuadrático, el primer esquema tiene varias ventajas sobre el segundo:

1) Como la distorsión de error cuadrático puede expresarse en términos de la suma de dos distorsiones de error cuadrático que sólo involucran las partes real e imaginaria, respectivamente [ver ecuacion (1a) en el capítulo 3], el diseño y búsqueda de los vocabularios correspondientes pueden hacerse independientemente. En el otro caso, esta independencia no existe entre las magnitudes y fases, a menos que sacrifiquemos la optimalidad o modifiquemos apropiadamente la distorsión [ver (1b-c) en el capítulo 3].

2) Como el vector de partes reales y el vector de partes imaginarias son prácticamente indistinguibles (tienen la misma dimensión y prácticamente las mismas medias y variancias muestreales), es conveniente considerarlos como un solo vector aleatorio y diseñar un solo VQ para cuantizar ambos. Por otro lado es necesario usar dos vocabularios diferentes, uno para la cuantización de magnitudes y el otro para fases.

3) La distorsión expresada en términos de magnitudes y fases involucra cálculo de cosenos, lo cual es una operación costosa: una manera de evitarlo es almacenar los senos y cosenos de las fases, evidentemente, el requerimiento de memoria se duplica.

Por estas razones el esquema de partes real e imaginaria es superior al esquema de magnitudes y fases. Sólo describimos un diseño del VQ para el primer caso, pero el método es igualmente aplicable al otro caso, con algunas modificaciones obvias.

Como primer paso de diseño, especificamos los coeficientes de ponderación que aparecen en la distorsión de error cuadrático.

Idealmente estos coeficientes deben ser variables dependiendo de la característica de cada espectro particular tal que minimice la percepción subjetiva del ruido de cuantización. Por ejemplo, las vocales generalmente tienen espectros con mucha energía (resonancia) en las frecuencias bajas (200-1600 Hz) y fuera de ese rango la energía es relativamente despreciable (de hecho podemos filtrar las componentes espectrales arriba de 2400 Hz de la señal de voz sin afectar mucho la calidad), por eso es más adecuado cuantizar más finamente las componentes de baja frecuencia (pero ésto no quiere decir de ninguna manera que las componentes de alta frecuencia deben de ser sacrificadas por completo; de hecho, el espectro de ruido de cuantización plano no es adecuado perceptualmente). Por desgracia, un estudio detallado de como selecionar los coeficientes de ponderación para minimizar la percepción de ruido de cuantización nos llevará demasiado lejos, por simplicidad, usamos el esquema de ponderación fija y uniforme: $P_k = 1$, para todo k.

El segundo paso de diseño es determinar la división "óptima" del espectro en varios subespectros (sub-bandas) y determinar también los tamaños "óptimos" de los subvocabularios para cada sub-banda. En otras palabras, se desea descomponer el vocabulario del VQ (que cuantiza los vectores de partes real e imaginaria del espectro) en varios subvocabularios de dimensión y tamaño adecuados, tales que

1) cada subvocabulario pueda ser diseñado individualmente sin requerir más recursos computacionales que los disponibles, específicamente, no debe de usar más memoria principal que la disponible;

2) la distribución de potencia de ruido de cuantización en cada sub-banda sea razonable (casi plana);

3) se mantenga un mínimo número de subvocabularios.

Hemos usado el siguiente método ad hoc para solucionar el

problema anterior:

0) Determinar la restricción de memoria principal. En nuestro caso, se dispone de aproximadamente 3740 variables de punto flotante (4 bytes cada una).

1) Calcular la variancia muestral de cada componente del espectro, las partes real e imaginaria de la misma frecuencia se toman como una sola variable aleatoria (vea Tabla 1 y Fig.1).

2) Mediante la siguiente fórmula (función de tasa-distorsión para fuente gaussiana con distorsión de error cuadrático medio) obtener una asignación de bit para cada frecuencia:

$$b_k = \max \{ \log_2 (s_k^2/d_k), 0 \} \qquad \text{(bits/muestra)}$$

$$d_k = \min \{ D^*, s_k^2 \}, \quad k = 1,\ldots,N$$

tal que $\sum_{k=1}^{N} b_k = B$

donde b_k es el número de bit asignado a la k-ésima componente con variancia s_k^2 , d_k es la distorsión de cuantización asociada, B es el número total de bits disponible por vector (N-dimensional) y D^* es un parámetro apropiado para que la última restricción se satisfaga. Para tomar en cuenta los efectos de percepción subjetiva, usamos una versión modificada de esta asignación de bits: substituir $(s_k^2)^{1+r}$ en lugar de s_k^2, con $-1 \leq r \leq 0$ [1]. En nuestro diseño, r=-1/4, N=63, B=64 (6.4 Kb/s) y obtenemos la asignación de bit y distribución de potencia de ruido de cuantización que se muestran en la Tabla 2 y la Fig. 2-3

Los resultados obtenidos del paso 2) pueden servir como una guía conveniente para el diseño, pero la implementación real no necesariamente sigue estrictamente las especificaciones anteriores por razones obvias.

3) Dadas las especificaciones obtenidas anteriormente, determinar una división del espectro adecuada, y los tamaños de los subvocabularios,

por el método de prueba y error. Es aconsejable usar pequeños vectores para las componentes en frecuencias bajas, no sólo para satisfacer las restricciones de complejidad de computación sino también porque los espectros tienen picos de resonancia mucho más angostos en bajas frecuencias. En la tabla 2: tenemos los parámetros del VQ.

Como tercer paso del diseño, usando el algoritmo de optimización conjunta modificado (un programa en FORTRAN IV es disponible en el apéndice) diseñamos los vocabularios de ganancia y de forma para cada sub-banda. Hemos diseñado un PC-VQ como producto cartesiano de cinco GS-VQ que corresponden a cinco sub-bandas de espectro, con tamaños especificados anteriormente. Las distorsiones de diseño y las razones de señal a ruido de cuantización (SQNR) se encuentran en la Tabla 3. Aquí la SQNR está definida por:

$$SQNR = 10 \log_{10}(D_0 / D_*)$$

donde D_0 y D_* son las distorsiones de tasa cero y de tasa deseada [3,5,6].

De manera similar, hemos diseñado también un VQ para cuantizar espectros por magnitudes y fases, para comparar con el esquema de cuantizar partes real e imaginaria. El resultado de la comparación indica que ambos esquemas producen señales de semejante calidad subjetiva, pero cuantizar partes real e imaginaria es mucho más atractivo debido a su simplicidad.

EVALUACION DEL SISTEMA Y ALGUNAS OBSERVACIONES

Una vez diseñado el VQ, aplicamos para codificar la secuencia de prueba usando la misma computadora. Las distorsiones de prueba son sistemáticamente mayores que las de diseño (Tabla 2 y Fig. 5), lo cual indica claramente que la secuencia de entrenamiento no es

suficientemente rica y que el VQ diseñado, basado en ésta, no es suficientemente representativo. A pesar de estas deficiencias del VQ, la calidad subjetiva de la señal reproducida alcanza la calificación subjetiva de "buena calidad de comunicación" según nuestro examen informal. La señal codificada a 6.4 Kb/s (secuencia de prueba) es totalmente inteligible, no tiene prácticamente pérdidas de naturalidad ni de identidad de hablante, pero si se percibe cierto nivel de ruido de cuantización, el cual hace que la señal codificada no suena con suficiente claridad como la original (típica característica de los codificadores de formas de onda a bajas tasas de bit).

Todas las operaciones de análisis, síntesis, codificación y decodificación se realiza fuera de linea (off-line), la operación más pesada es la de codificación, consume varias horas para codificar una señal de tan solo varias decenas de segundos. Los programas han sido diseñados para propósitos de investigación, por lo que no están optimizados.

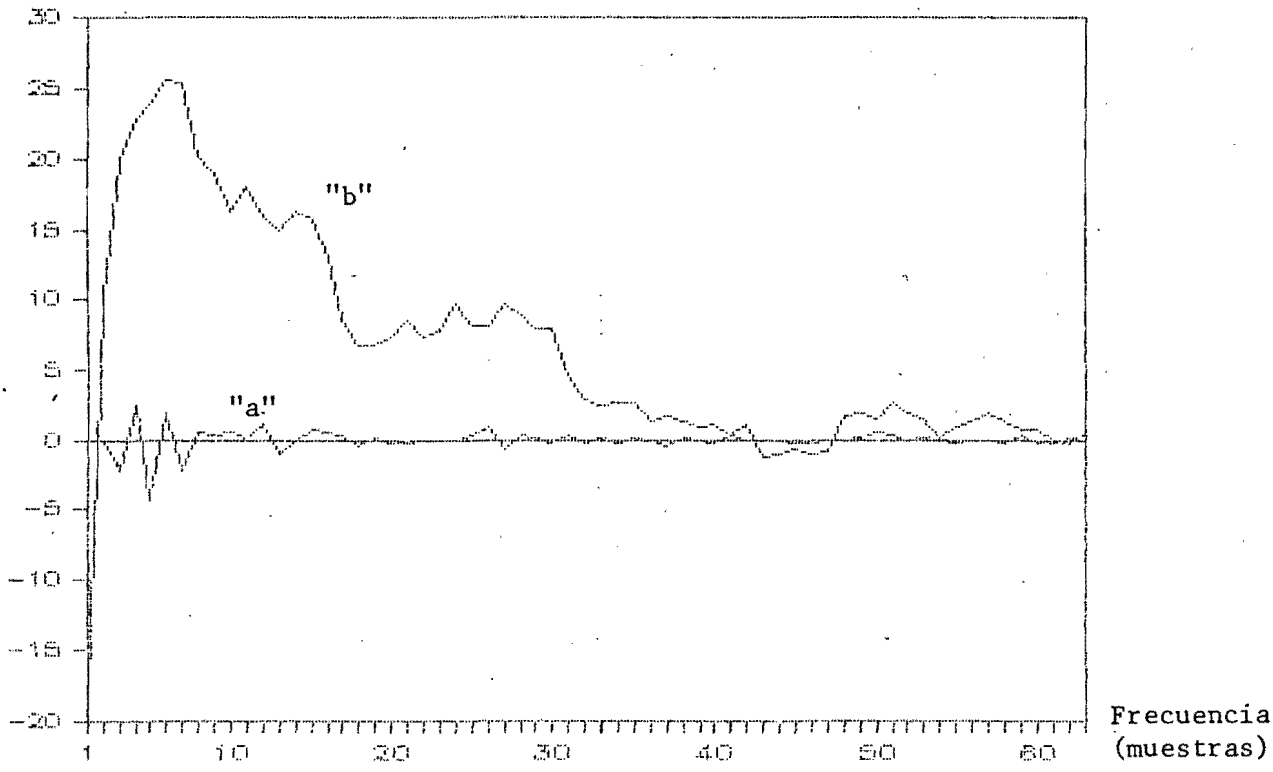


Fig.1 Media y variancia muestral de los espectros de duración corta

"a": media (amplificada por 10)

"b": variancia (dB)

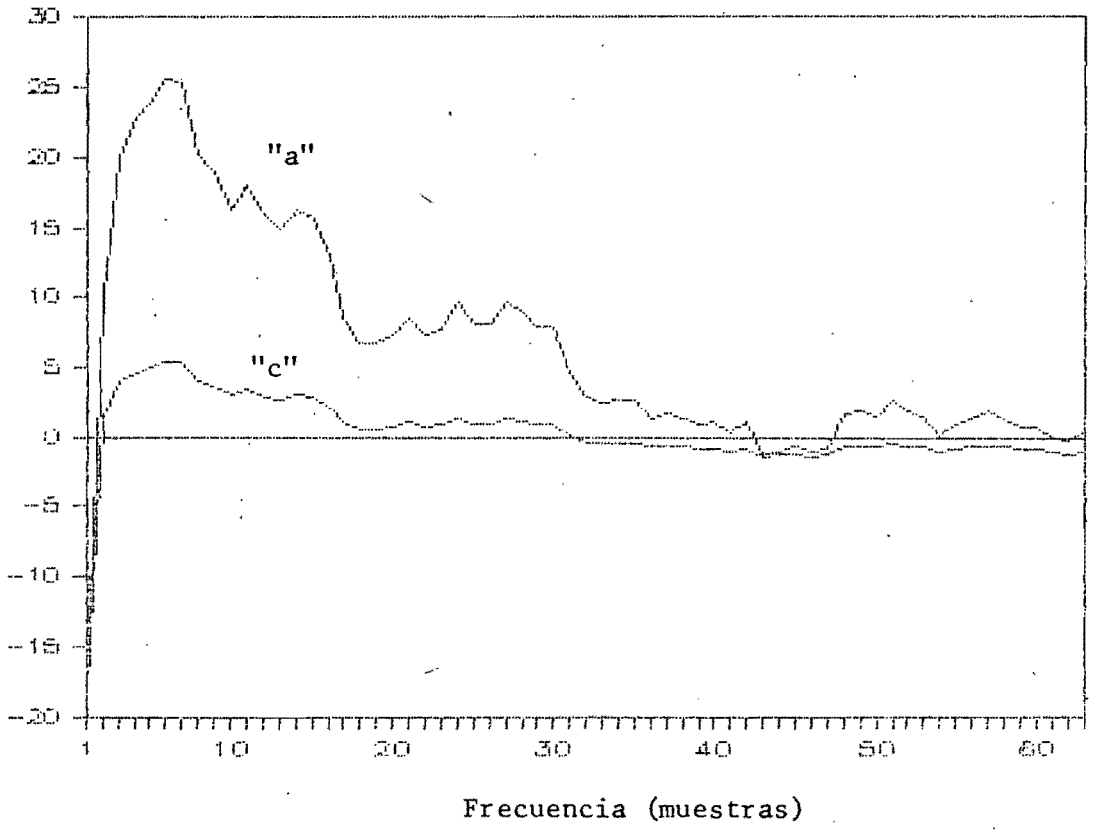


Fig. 2 Espectros de voz y de ruido de cuantización

"c": Espectro de ruido de cuantización de diseño (dB)

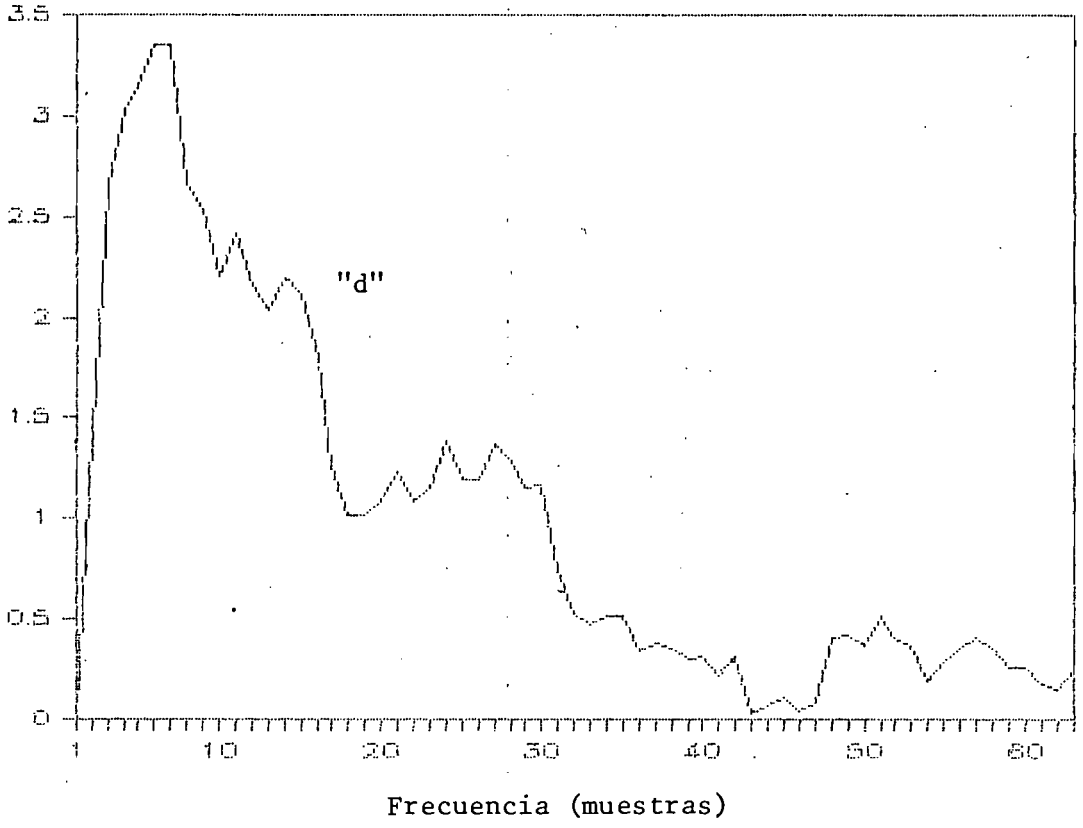


Fig.3 "d": asignación óptima de bit (bits)

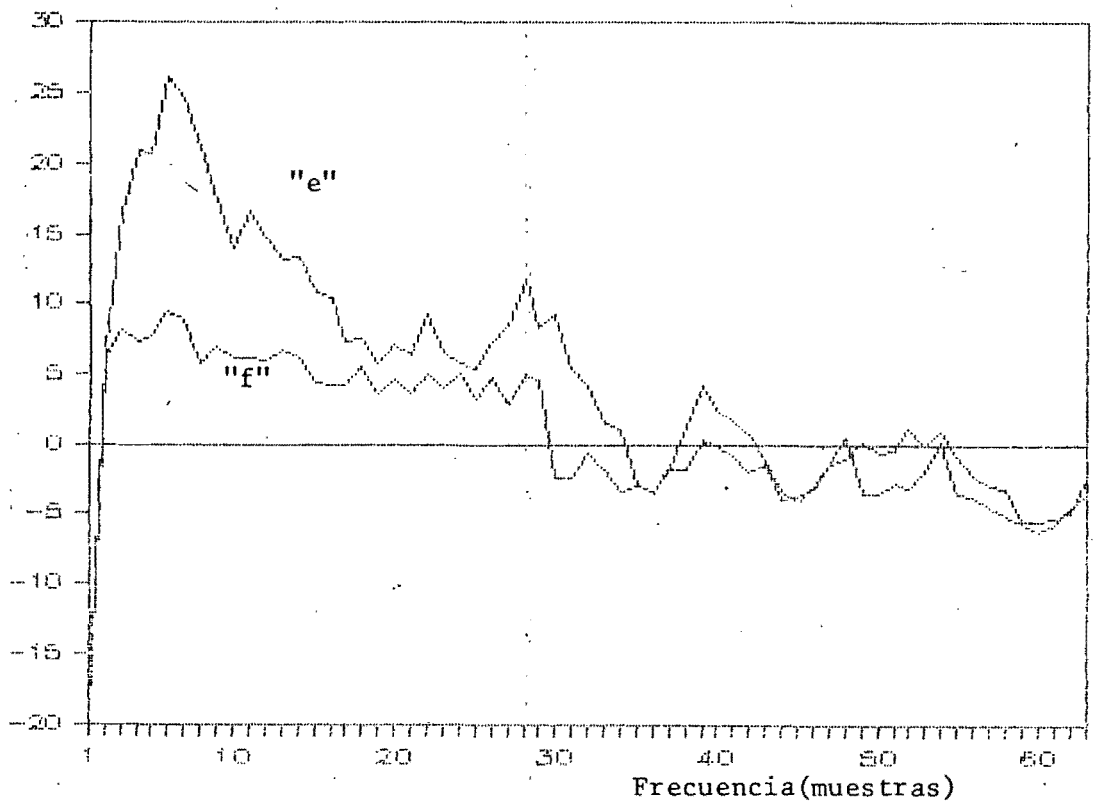


Fig.4 Espectros de voz y de ruido de cuantización (prueba)

"e": Espectro de voz (dB)

"f": Espectro de ruido de cuantización (dB)

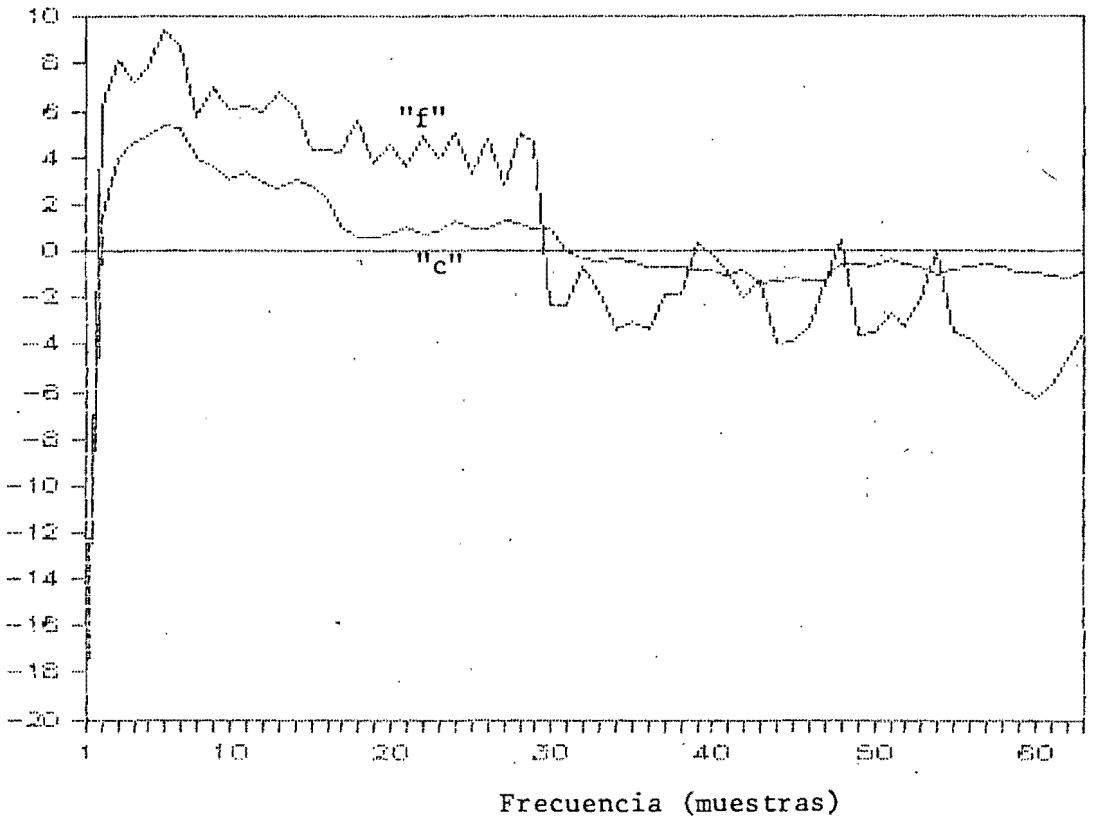


Fig.5 Comparación de espectros de ruido de de cuantización de diseño y de prueba

1	-0.001	0.014	0.014	0	0.01	0.01
2	-0.014	9.935	1.392	1.418	4.175	4.946
3	-0.212	99.199	2.474	2.663	6.596	44.387
4	0.243	192.896	2.921	3.023	5.362	122.051
5	-0.441	243.286	3.095	3.148	5.987	116.727
6	0.197	356.995	3.407	3.356	8.798	411.184
7	-0.219	353.175	3.398	3.35	7.598	292.721
8	0.061	102.36	2.493	2.68	3.783	129.359
9	0.033	78.127	2.33	2.534	4.992	54.955
10	0.049	41.8	1.993	2.195	4.069	24.285
11	0.014	63.358	2.211	2.42	4.206	47.046
12	0.12	39.078	1.96	2.159	3.977	29.795
13	-0.096	31.453	1.856	2.041	4.75	20.717
14	-0.004	42.307	1.999	2.202	4.134	21.895
15	0.082	36.45	1.926	2.121	2.733	12.351
16	0.062	19.588	1.649	1.785	2.687	11.02
17	0.027	7.35	1.291	1.255	2.669	5.339
18	-0.038	4.727	1.156	1.016	3.677	5.611
19	0.023	4.734	1.156	1.017	2.344	3.82
20	-0.023	5.372	1.193	1.085	2.837	5.059
21	-0.016	6.975	1.274	1.227	2.276	4.405
22	-0.01	5.389	1.194	1.087	3.101	8.545
23	-0.013	6.016	1.227	1.147	2.503	4.452
24	-0.006	9.205	1.365	1.377	3.144	3.802
25	0.043	6.537	1.253	1.192	2.102	3.444
26	0.1	6.612	1.257	1.198	3.031	5.496
27	-0.055	9.066	1.36	1.368	1.905	7.179
28	0.04	7.884	1.313	1.293	3.146	14.985
29	0.014	6.072	1.23	1.152	2.896	6.705
30	-0.017	6.218	1.238	1.164	0.582	8.286
31	0.042	2.83	1.017	0.739	0.582	3.464
32	-0.024	1.911	0.921	0.526	0.853	2.632
33	0.022	1.744	0.901	0.477	0.671	1.438
34	-0.03	1.882	0.918	0.518	0.461	1.27
35	0.014	1.846	0.914	0.507	0.501	0.533
36	-0.005	1.337	0.843	0.333	0.467	0.447
37	-0.039	1.451	0.86	0.377	0.66	0.696
38	0.005	1.386	0.85	0.352	0.654	1.376
39	-0.015	1.239	0.827	0.292	1.08	2.629
40	-0.027	1.283	0.834	0.311	0.98	1.764
41	0.027	1.065	0.796	0.21	0.831	1.495
42	-0.013	1.288	0.835	0.313	0.629	1.201
43	-0.006	0.742	0.728	0.015	0.74	0.804
44	-0.01	0.806	0.743	0.059	0.401	0.467
45	-0.02	0.868	0.756	0.099	0.417	0.384
46	-0.019	0.778	0.736	0.04	0.475	0.507
47	-0.012	0.841	0.751	0.082	0.724	0.717
48	-0.008	1.507	0.868	0.398	1.112	0.795
49	0.02	1.571	0.877	0.42	0.434	1.055
50	0.045	1.398	0.852	0.357	0.45	0.853
51	0.03	1.868	0.916	0.514	0.538	0.896
52	-0.011	1.536	0.873	0.408	0.476	1.317
53	0.018	1.393	0.852	0.355	0.62	0.966
54	-0.011	1.019	0.787	0.186	1.016	1.244
55	-0.019	1.227	0.825	0.287	0.446	0.831
56	-0.001	1.365	0.847	0.344	0.424	0.61
57	-0.015	1.512	0.869	0.399	0.36	0.497
58	-0.018	1.374	0.849	0.348	0.315	0.471
59	0.024	1.168	0.815	0.26	0.265	0.283
60	-0.023	1.161	0.814	0.257	0.238	0.283
61	-0.027	0.989	0.782	0.17	0.267	0.291
62	0.008	0.937	0.771	0.141	0.347	0.326
63	-0.006	1.124	0.807	0.239	0.448	0.57

Tabla 1

subvocabulario	1	2	3	4	5
dimensión	6	7	15	15	19
tasa de ganancia	7	6	5	4	2
tasa de forma	9	9	8	8	6
distorsión (diseño)	5.09	2.17	1.34	0.32	0.63
SQNR (dB)	16.06	14.14	9.72	7.43	2.92
distorsión (prueba)	8.36	5.99	3.04	0.80	0.69

Tabla 2

5

CONCLUSION Y RECOMENDACION PARA INVESTIGACION FUTURA

En los capítulos anteriores de la tesis se han descrito el diseño e implementación (simulación por computadora), de un sistema de compresión de voz basado en la codificación de formas de onda en el dominio de la frecuencia con cuantización vectorial. Los resultados experimentales senalan que este sistema de compresión es capaz de alcanzar buena calidad de comunicación a la tasa de 6.4 Kb/s. El sistema requiere gran cantidad de procesamiento para realizar la cuantización vectorial, a pesar de que los vocabularios usados tienen ciertas estructuras internas; esta dificultad sigue siendo el principal obstáculo del mejoramiento del desempeño de los sistemas prácticos.

La transformada discreta de Fourier de duración corta (DSTFT) ha sido una representación muy útil en el dominio de la frecuencia para señales de voz. Los resultados experimentales indican nuevamente que las componentes en la frecuencias bajas (200-1600 Hz aproximadamente) constituye la parte esencial de casi todos los patrones de espectros. Las componentes de alta frecuencia son relativamente menos importantes en la percepción, pero tampoco deben de sacrificarse completamente (repetimos una vez más: el espectro plano de ruido de cuantización no es apropiado); es particularmente adecuado cuantizar estas componentes mediante la cuantización vectorial. Sin embargo la DSTFT no es adecuada

para compresión a muy bajas tasas (menor de 3 Kb/s) sino a tasas bajas y medianas (5-16 Kb/s), porque esta representación no es un modelo especialmente diseñado para representar la señal de voz, no tiene por si misma capacidad de compresión sino es una representación alterna que conserva toda la información necesaria para obtener una reproducción casi perfecta.

Los cuantizadores vectoriales con estructuras de producto cartesiano (PC-VQ) y los de ganancia y forma (GS-VQ), no sólo ha sido demostrado que son adecuados, sino también son necesarios para satisfacer el compromiso de desempeño y complejidad, surgido en la implementación de nuestro codificador de formas de onda a 6.4 kb/s. A pesar del gran requerimiento de procesamiento y de memoria, la lógica o inteligencia del sistema es relativamente sencilla (no requiere algoritmos muy complicados). también hay que notar que los PC-VQ son particularmente adecuados para ser implementados por varios procesadores (más simples) trabajando en paralelo.

Un proyecto de investigación futuro podría ser el diseño e implementación de sistemas de compresión de voz a 4.8 o 9.6 Kb/s (tasas de transmisión comercialmente disponibles), basados también en cuantización vectorial de la DSTFT. Para lograr un posible mejoramiento de calidad y una reducción adicional de la tasa, recomendamos especialmente un estudio detallado con objeto de diseñar una medida de distorsión más adecuada [15]. También recomendamos usar algún esquema de codificación por predicción en la frecuencia ya que entre los espectros sucesivos existe cierta correlación; para ello, la DSTFT con modulación de banda lateral única (single-side band modulation) parece ser más adecuada que con la modulación compleja, ya que la primera produce señales de canal reales [10]. Debido a la estructura peculiar de los

espectros de voz, resulta muy atractivo trabajar con espectros de resolución no uniforme, por ejemplo, el esquema de espaciamiento de octava podría ser una opción interesante [10]. Finalmente, una manera de hacer el sistema más adaptivo y consecuentemente mejorar el desempeño del mismo, es clasificar primero los espectros en diferentes clases dependiendo de su energía o de la frecuencia de resonancia u otros criterios, después, cuantizarlos usando un cuantizador vectorial de estructura de arbol, cada rama del arbol usa un vocabulario diseñado con diferentes medidas de distorsión y/o asignación de bit, que corresponde exclusivamente a una clase de espectros.

REFERENCIAS

[1] Tribolet, J.M., y Crochiere, R.E.,
"Frequency Domain Coding of Speech",
IEEE Trans.on Acoust., Speech, and Signal Process.,
Vol.ASSP-27, No.5, October 1979, pp.512-530.

[2] Buzo, A., Gray, A.H.Jr., Gray, R.M., y Markel, J.D.,
"Speech Coding Based Upon Vector Quantization",
IEEE Trans.on Acoust., Speech, and Signal Process.,
Vol.ASSP-28, No.5, October 1980, pp.562-574.

[3] Abut, H., Gray, R.M., y Rebolledo, G.,
"Vector Quantization of Speech and Speech-Like Waveforms",
IEEE Trans.on Acoust., Speech, and Signal Process.,
Vol.ASSP-30, No.3, June 1982, pp.423-435.

[4] Cuperman, V. y Gersho, A.,
"Vector Predictive Coding of Speech at 16 Kbits/s",
IEEE Trans.on Communications,
Vol.COM-33, No.7, July 1985, pp.685-696.

[5] Sabin, M.J. y Gray, R.M.,
"Product Code Vector Quantizers for Waveform and Voice Coding",
IEEE Trans.on Acoust., Speech, and Signal process.,
Vol.ASSP-32, No.3, June 1984, pp.474-488.

[6] Gray, R.M.,
"Vector Quantization",
IEEE ASSP Magazine, April 1984, pp.4-29.

[7] Schroeder, M.R.,
"Linear Predictive Coding of Speech: Review and Current Directions",
IEEE Communications Magazine,
Vol.23, No.8, August 1985, pp.54-61.

[8] Crochiere, R.E. y Flanagan, J.L.,
"Current Perspective In Digital Speech",
IEEE Communications Magazine, January 1983, pp.32-40.

[9] Heron, C.D., Crochiere, R.E. y Cox, R.V.,
"A 32-Band Subband/transform Coder Incorporating Vector Quantization for Dynamic Bit Allocation",
Proceedings ICASSP, April 1983, pp.1276-1279.

[10] Crochiere, R.E. y Rabiner, L.R.,
Multirate Digital Signal Processing,
Prentice-Hall Inc., Englewood Cliffs., NJ, 1983, capítulo 7.

[11] Oetken, G., Parks, T.W. y Schussler, H.W.,
"A Computer Program for Digital Interpolator Design",
Programs for Digital Signal Processing,
IEEE ASSP committee, IEEE press, 1979.

[12] Portnoff, M.R.,
"Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform",
IEEE trans.on Acoust.,Speech, and Signal process.,
Vol.ASSP-24, No.3, June 1976, pp.243-248.

[13] Papoulis, A.,
Signal Analysis,
McGraw-Hill Book Company, NY, 1977, p.142.

[14] Bergland, G.D. y Dolan, M.T.,

"Fast Fourier Transform Algoritms",
Programs for Digital Signal Processing,
IEEE ASSP committee, IEEE press, 1978.

[15] Jayant,N.S.,y Noll, P.,
Digital Coding of Waveforms, principles and
applications to speech and video,
Prentice-Hall, Inc., Englewood cliffs., NJ, 1984,
Capítulos 11, 12, Apéndices E y F.

APENDICE

A) PROGRAMA PARA ANALISIS

B) PROGRAMA PARA SINTESIS

C) PROGRAMA PARA IMPLEMENTAR EL ALGORITMO DE OPTIMIZACION CONJUNTA MODIFICADO

```
CALCULO DE LA TRANSFORMADA DISCRETA DE FOURIER DE
DURACION CORTA DE SENAL DE VOZ USANDO SUBRUTINA FAST

PROGRAM AFFTR2

REAL H(2560),X(1280)
REAL B(130)

TYPE 10
FORMAT(2X,'ARCHIVO DE SENAL DE VOZ A ANALIZAR:=',$)
CALL ASSIGN(1,' ',-1)
TYPE 20
FORMAT(2X,'NUMERO DE MUESTRAS POR BLOQUE=N=2**M,M:=',$)
ACCEPT *,M
N=2**M
NV2=N/2
NV2M1=NV2-1
NV2P1=NV2+1
FN=FLOAT(N)
TYPE 30
FORMAT(2X,'LONGITUD DE VENTANA=MUL*N,MUL:=',$)
ACCEPT *,MUL
L=MUL*N
TYPE 35,L
FORMAT(2X,'NUMERO DE MUESTRAS DE LA VENTANA:=',I5)
LV2=L/2
LV2P1=LV2+1
TYPE 40
FORMAT(2X,'NUMERO DE BLOQUES A LEER:=',$)
ACCEPT *,NBL
IRECL1=2*N
TYPE 50
FORMAT(2X,'BLOQUE INICIAL:=',$)
ACCEPT *,INIBL
LSTBL=INIBL+NBL-1
DEFINE FILE 1(LSTBL,IRECL1,U,I1)
I1=INIBL
TYPE 55
FORMAT(2X,'RANGO DE INTERES:=[L1,L2]',$)
ACCEPT *,L1,L2
LR=L2-L1+1
L1=2*L1+1
L2=2*L2+2
TYPE 60
FORMAT(2X,'ARCHIVO PARA ESPECTRO:=',$)
CALL ASSIGN(2,'DK1:AMP.DAT',-1)
IRECL2=4*LR
DEFINE FILE 2(NBL,IRECL2,U,I2)
I2=1

LEER LOS COEFICIENTES DE UNA VENTANA

TYPE 80
FORMAT(2X,'ARCHIVO DE VENTANA DE ANALISIS:=',$)
CALL ASSIGN(4,'DK:FILTER.DAT',-1)
IRECL4=L
DEFINE FILE 4(1,IRECL4,U,I4)
I4=1
READ(4'I4) (H(K),K=1,LV2)
```

```
CALL CLOSE(4)
H(LV2FI)=1.0
DO 90 K=2,LV2
H(L+2-K)=H(K)
DO 100 K=1,L
H(L+K)=H(K)


IB=L-N
DO 120 I=1,IB
X(I)=0.0

CONTINUE
READ(1'I1) (X(IB+I),I=1,N)
IB=IB+N
IF(IB.EQ.L) IB=0
LI=L-IB
DO 300 I=1,N
Z=0.0
K=I
DO 250 J=1,MUL
Z=Z+H(LI+K)*X(K)
K=K+N
B(I)=Z

CALL FAST(B,N)
WRITE(2'I2) (B(I),I=L1,L2,2),(B(J),J=L1+1,L2,2)
IF(I1.LE.LSTBL) GO TO 200

TYPE 500,NBL
FORMAT(2X,'NUMERO DE BLOQUES PROCESADOS:=',I5)
CALL EXIT
END
```

B

```
SINTESIS DE SENAL DE VOZ CON LA TRANSFORMADA DISCRETA
DE FOURIER DE DURACION CORTA USANDO SUBRUTINA FSST

PROGRAM SFFTR2

REAL F(2560),S(1280)
REAL X(128)
REAL B(130)

TYPE 10
FORMAT(2X,'ARCHIVO DE ESPECTRO:=',$)
CALL ASSIGN(1,'DK1:AMP.DAT',-1)
TYPE 30
FORMAT(2X,'NUMERO DE MUESTRAS POR BLOQUE=N=2**M,M:=',$)
ACCEPT *,M
N=2**M
NV2=N/2
NV2M1=NV2-1
NV2P1=NV2+1
FN=FLOAT(N)
TYPE 35
FORMAT(2X,'RANGO DE INTERES[L1,L2]:=',$)
ACCEPT *,L1,L2
L1=2*L1+1
L2=2*L2+2
LR=L2-L1+1
TYPE 40
FORMAT(2X,'LONGITUD DE VENTANA=MUL*N,MUL:=',$)
ACCEPT *,MUL
L=MUL*N
TYPE 45,L
FORMAT(2X,'NUMERO DE MUESTRAS DEL INTERPOLATOR:=',I5)
LV2=L/2
LV2P1=LV2+1
TYPE 50
FORMAT(2X,'NUMERO DE BLOQUES A SINTETIZAR:=',$)
ACCEPT *,NBL
IRECL1=2*LR
DEFINE FILE 1(NBL,IRECL1,U,I1)
I1=1
TYPE 60
FORMAT(2X,'ARCHIVO PARA SENAL SINTETICA:=',$)
CALL ASSIGN(3,'VOZ.SIN',-1)
IRECL3=2*N
DEFINE FILE 3(NBL,IRECL3,U,I3)
I3=1

LEER LOS COEFICIENTES DE UN INTERPOLATOR

TYPE 70
FORMAT(2X,'ARCHIVO DE INTERPOLATOR:=',$)
CALL ASSIGN(4,'DK:FILTER.DAT',-1)
IRECL4=L
DEFINE FILE 4(1,IRECL4,U,I4)
I4=1
READ(4'I4) (F(K),K=1,LV2)
CALL CLOSE(4)
```

```
      F(LV2P1)=1.0
      DO 80 K=2,LV2
      F(L+2-K)=F(K)
      DO 90 K=1,L
      F(L+K)=F(K)


      IB=L-N
      DO 110 I=1,IB
      S(I)=0.0

      CONTINUE
      READ(1'I1) (B(I),I=L1,L2,2),(B(J),J=L1+1,L2,2)
      DO 130 K=1,L1-1
      B(K)=0.0
      DO 140 K=L2+1,N
      B(K)=0.0

      CALL FSST(B,N)
      DO 150 I=1,N
      S(IB+I)=B(I)
      IB=IB+N
      IF(IB.EQ.L) IB=0
      DO 160 I=1,N
      X(I)=0.0
      K1=I
      K2=L-N+I
      DO 160 J=1,MUL
      X(I)=X(I)+F(IB+K1)*S(K2)
      K1=K1+N
      K2=K2-N
      WRITE(3'I3) (X(I),I=1,N)
      IF(I1.LE.NBL) GO TO 120

      TYPE 200,NBL
      FORMAT(2X,'NUMERO DE BLOQUES SINTETIZADOS:=',I5)
      CALL EXIT
      END
```

```
C
PROGRAMA PARA REALIZAR EL ALGORITMO DE
OPTIMIZACION CONJUNTA MODIFICADO

PROGRAM SEGSVQ
REAL A(16),S(128,16)
REAL G(32),U(32)
DATA IR,JR/0,0/
RANGE(IR,JR)=1.0+0.1*(RAN(IR,JR)-0.5)
TYPE 10
FORMAT(2X,'ARCHIVO DE SECUENCIA DE ENTRENAMIENTO:=',$)
CALL ASSIGN(1,' ',-1)
TYPE 20
FORMAT(2X,'NUMERO DE BLOQUES:=',$)
ACCEPT *,NBL
TYPE 30
FORMAT(2X,'TAMANO DE BLOQUE:=',$)
ACCEPT *,N
RFN=1.0/FLOAT(N)
NN=N+N
DEFINE FILE 1(NBL,NN,U,I1)
I1=1
TYPE 40
FORMAT(2X,'TASA INICIAL DE FORMA:=',$)
ACCEPT *,NSI
NSI=2**NSI
IF(NSI.LE.1) GO TO 70
TYPE 50
FORMAT(2X,'ARCHIVO INICIAL DE FORMA:=',$)
CALL ASSIGN(2,' ',-1)
DEFINE FILE 2(NSI,NN,U,I2)
I2=1
DO 60 I=1,NSI
READ(2'I2) (S(I,K),K=1,N)
CALL CLOSE(2)
GO TO 90
CONTINUE
R=0.0
DO 80 K=1,N
S(1,K)=2.0*(RAN(IR,JR)-0.5)
R=R+S(1,K)**2
R=R*RFN
DO 85 K=1,N
S(1,K)=S(1,K)/R
CONTINUE
TYPE 100
FORMAT(2X,'TASA FINAL DE FORMA:=',$)
ACCEPT *,NSF
NSF=2**NSF
TYPE 110
FORMAT(2X,'TASA INICIAL DE GANANCIA:=',$)
ACCEPT *,NGI
NGI=2**NGI
IF(NGI.LE.1) GO TO 130
TYPE 120
FORMAT(2X,'ARCHIVO INICIAL DE GANANCIA:=',$)
CALL ASSIGN(3,' ',-1)
NG2=2*NGI
DEFINE FILE 3(2,NG2,U,I3)
```

```
I3=1
READ(3'I3) (G(K),K=1,NGI)
READ(3'I3) (U(K),K=1,NGI-1),DO
CALL CLOSE(3)
GO TO 140
CONTINUE
G(1)=2.0
U(1)=100.0
DO=300.0
CONTINUE
TYPE 150
FORMAT(2X,'TASA FINAL DE GANANCIA:=',$)
ACCEPT *,NGF
NGF=2**NGF
TYPE 160
FORMAT(2X,'UMBRAL DE ERROR RELATIVO:=',$)
ACCEPT *,EPS
EPSM=4.0*EPS

DFIX=0.0
DO 180 I=1,NBL
READ(1'I1) (A(K),K=1,N)
R=0.0
DO 170 K=1,N
R=R+A(K)**2
DFIX=DFIX+R*RFN
DFIX=DFIX/FLOAT(NBL)

IF(NGI.GT.1.OR.NSI.GT.1) GO TO 1000
TYPE 190,NGI,NSI
FORMAT(2X,'GANANCIA:=',I3,'  FORMA:=',I4)
M=0
M=M+1
CALL MJOASE(N,NSI,NGI,I1,NBL,A,S,G,U,DO,DFIX,E)
TYPE 210,M,DO,E
FORMAT(2X,'NI:=',I2,6X,'D:=',F10.4,6X,'E:=',F9.5)
IF(E.GT.EPS) GO TO 200

IF(NGI.GE.NGF) GO TO 2000
IF(NGI.LE.1) GO TO 1200
DO 1100 I=NGI,2,-1
J=I+I
G(J)=G(I)
G(J-1)=U(I-1)
G(2)=G(1)
G(1)=0.9*G(1)
NGI=NGI+NGI
TYPE 190,NGI,NSI
DO 1300 I=1,NGI-1
U(I)=0.5*(G(I)+G(I+1))
M=0
M=M+1
CALL MJOASE(N,NSI,NGI,I1,NBL,A,S,G,U,DO,DFIX,E)
TYPE 210,M,DO,E
IF((E.GT.EPSM.OR.M.LT.2).AND.M.LT.5) GO TO 1400
CALL ASSIGN(2,'DK1:GAIN.DAT')
NG2=2*NGI
DEFINE FILE 2(2,NG2,U,I2)
```

```
      I2=1
      WRITE(2'I2) (G(K),K=1,NGI)
      WRITE(2'I2) (U(K),K=1,NGI-1),DO
      CALL CLOSE(2)
      CALL ASSIGN(3,'DK1:SHAPE.DAT')
      DEFINE FILE 3(NSI,NN,U,I3)
      I3=1
      DO 1500 I=1,NSI
00    WRITE(3'I3) (S(I,K),K=1,N)
      CALL CLOSE(3)
      GO TO 1000

00    IF(NSI.GE.NSF) GO TO 3000
      DO 2100 I=NSI,1,-1
      J=I+I
      DO 2200 K=1,N
00    S(J,K)=S(I,K)
      JM1=J-1
      R=0.0
      DO 2300 K=1,N
      S(JM1,K)=S(J,K)*RANGE(IR,JR)
00    R=R+S(JM1,K)**2
      R=SQRT(R*RFN)
      DO 2400 K=1,N
00    S(JM1,K)=S(JM1,K)/R
00    CONTINUE
      NSI=NSI+NSI
      TYPE 190,NGI,NSI
      M=0
00    M=M+1
      CALL MJOASE(N,NSI,NGI,I1,NBL,A,S,G,U,DO,DFIX,E)
      TYPE 210,M,DO,E
      IF((E.GT.EPSM.OR.M.LT.2).AND.M.LT.5) GO TO 2500
      CALL ASSIGN(2,'DK1:GAIN.DAT')
      NG2=NGI+NGI
      DEFINE FILE 2(2,NG2,U,I2)
      I2=1
      WRITE(2'I2) (G(K),K=1,NGI)
      WRITE(2'I2) (U(K),K=1,NGI-1),DO
      CALL CLOSE(2)
      CALL ASSIGN(3,'DK1:SHAPE.DAT')
      DEFINE FILE 3(NSI,NN,U,I3)
      I3=1
      DO 2600 I=1,NSI
)0    WRITE(3'I3) (S(I,K),K=1,N)
      CALL CLOSE(3)
      GO TO 2000

)0    TYPE 190,NGF,NSF
      M=0
)0    M=M+1
      CALL MJOASE(N,NSF,NGF,I1,NBL,A,S,G,U,DO,DFIX,E)
      TYPE 210,M,DO,E
      IF(E.GT.EPS.AND.M.LT.5) GO TO 3100
      CALL CLOSE(1)
      CALL ASSIGN(2,'DK1:GAIN.DAT')
      NG2=NGF+NGF
      DEFINE FILE 2(2,NG2,U,I2)
```

```
      I2=1
      WRITE(2'I2) (G(K),K=1,NGF)
      WRITE(2'I2) (U(K),K=1,NGF-1),DO
      CALL CLOSE(2)
      CALL ASSIGN(3,'DK1:SHAPE.DAT')
      DEFINE FILE 3(NSF,NN,U,I3)
      I3=1
      DO 3200 I=1,NSF
00    WRITE(3'I3) (S(I,K),K=1,N)
      CALL CLOSE(3)
      TYPE 3300
00    FORMAT(2X,'ARCHIVO FINAL DE FORMA:=DK1:SHAPE.DAT',/,
     + 2X,'ARCHIVO FINAL DE GANANCIA:=DK1:GAIN.DAT')
      TYPE 3400,DO
00    FORMAT(2X,'DISTORSION FINAL DE DISENO:=',F12.4)
      CALL EXIT
      END


      SUBROUTINE MJOASE(N,NS,NG,I1,NBL,A,S,G,U,DO,DFIX,E)
      REAL A(16)
      REAL S(128,16),SA(128,16)
      REAL G(32),U(32),GA(32)
      INTEGER L(32),M(128)
      DATA IR,JR/0,0/
      RANGE(IR,JR)=1.0+0.1*(RAN(IR,JR)-0.5)
      FN=FLOAT(N)
      DO 10 I=1,NS
      M(I)=0
      DO 20 K=1,N
      SA(I,K)=0.0
      CONTINUE
      DO 30 J=1,NG
      GA(J)=0.0
      L(J)=0
      D=0.0
      I1=1
      DO 40 I=1,NBL
      READ(1'I1) (A(K),K=1,N)
      IMAX=1
      DMAX=-1000.0
      DO 50 J=1,NS
      DT=0.0
      DO 60 K=1,N
      DT=DT+A(K)*S(J,K)
      IF(DT.LE.DMAX) GO TO 50
      DMAX=DT
      IMAX=J
      CONTINUE
      M(IMAX)=M(IMAX)+1
      DMAX=DMAX/FN
      DO 70 J=1,NG-1
      IF(DMAX.LT.U(J)) GO TO 80
      IF(DMAX.LT.0.0) DMAX=0.0
      L(J)=L(J)+1
      GA(J)=GA(J)+DMAX
      DO 90 K=1,N
      SA(IMAX,K)=SA(IMAX,K)+G(J)*A(K)
```

```
D=D-G(J)*(2.0*DMAX-G(J))
CONTINUE

DO 100 I=1,NS
IF(M(I).LT.1) GO TO 100
R=0.0
DO 110 K=1,N
R=R+SA(I,K)**2
R=SQRT(R/FN)
DO 120 K=1,N
S(I,K)=SA(I,K)/R
CONTINUE
DO 130 I=1,NS
IF(M(I).GE.1) GO TO 130
IMAX=1
JMAX=1
DO 140 J=1,NS
IF(M(J).LE.IMAX) GO TO 140
IMAX=M(J)
JMAX=J
CONTINUE
R=0.0
DO 150 K=1,N
S(I,K)=S(JMAX,K)*RANGE(IR,JR)
R=R+S(I,K)**2
CONTINUE
R=SQRT(R/FN)
DO 160 K=1,N
S(I,K)=S(I,K)/R
M(JMAX)=M(JMAX)/2
CONTINUE

DO 170 J=1,NG
IF(L(J).NE.0) G(J)=GA(J)/FLOAT(L(J))
DO 180 J=1,NG
IF(L(J).GT.0) GO TO 180
IF(J.NE.1.AND.J.NE.NG) G(J)=0.5*(G(J-1)+G(J+1))
IF(J.EQ.1) G(J)=0.95*G(J+1)
IF(J.EQ.NG) G(J)=1.05*G(J-1)
CONTINUE
IF(NG.LE.1) GO TO 200
DO 190 J=1,NG-1
U(J)=0.5*(G(J)+G(J+1))
D=D/FLOAT(NBL)+DFIX
E=ABS((DO-D)/D)
DO=D
RETURN
END
```