CODIFICACIÓN APVQ DE VOZ EN BANDA ANCHA PARA VELOCIDADES ENTRE 16 Y 32 KBPS

Josep M. SALAVEDRA, Enrique MASGRAU**, Marcos FAUNDEZ**

* Department of Signal Theory and Communications. Universitat Politècnica de Catalunya.
Campus Nord UPC, Mòdul D5. Gran Capità s/n , 08034 BARCELONA.
Phone: +34.3.4016440. Telefax: +34.3.4016447. E-mail: mia@gps.tsc.upc.es
** Department of Electrical Engineering and Computers. Universidad de Zaragoza.
María de Luna, 3. 50015-ZARAGOZA.

ABSTRACT

This paper describes a coding scheme for broadband speech (sampling frequency 16KHz). We present a wideband speech encoder called APVQ (Adaptive Predictive Vector Quantization). It combines Subband Coding, Vector Quantization and Adaptive Prediction as it is represented in Fig.1. Speech signal is split in 16 subbands by means of a QMF filter bank and so every subband is 500Hz wide. This APVQ encoder can be seen as a vectorial extension of a conventional ADPCM encoder. In this scheme, signal vector is formed with one sample of the normalized prediction error signal coming from different subbands and then it is vector quantized. Prediction error signal is normalized by its gain and normalized prediction error signal is the input of the VQ and therefore an adaptive Gain-Shape VQ is considered. This APVQ Encoder combines the advantages of Scalar Prediction and those of Vector Quantization. We evaluate wideband speech coding in the range from 1 to 2 bits/sample, that leads to a coding rate from 16 to 32 kbps.

1. INTRODUCCIÓN

La combinación de las técnicas de división en subbandas mediante cuantificación vectorial y predicción adaptativa proporciona muy buenos resultados en codificación de señal de voz de banda estrecha (4KHz) a velocidades medias de 1 bit/muestra (8 Kbps.). Un ejemplo de este tipo de codificadores es el denominado APVQ (Adaptive Predictive Vector Quantization) [1] que consiste, básicamente, en una división de la señal de voz en 8 subbandas de 500 Hz cada una mediante un banco de filtros QMF, seguido de una cuantificación ADPCM "backward" de cada una de las subbandas, con la particularidad de que la cuantificación del error de predicción en cada una de las subbandas se realiza mediante una cuantificación vectorial (VQ), de tal modo que cada uno de estos errores de predicción constituye una de las componentes del vector de entrada al VQ. Es decir, en vez de cuantificar el error de predicción de cada una de las subbandas de forma independiente mediante un cuantificador escalar, se cuantifican en bloque mediante un VQ. Además, este VQ es adaptativo en el sentido de que los errores de predicción son previamente normalizados en ganancia mediante una estimación "backward" de ésta. Es decir, se hace uso de un VQ ganancia-forma adaptativo. Como se detalla en la referencia [1], a esta velocidad de transmisión moderada de 1 bit/muestra, la predicción adaptativa no aporta ninguna ventaja en las bandas 5 a 8 (por encima de 2 KHz), con lo que puede prescindirse de ella en estas bandas. Esto es debido a que el error de cuantificación producido en la representación de la señal en cada una de las subbandas enmascara el potencial de blanqueado en tiempo proporcionado por la predicción, el cual ya está bastante reducido debido a la división en subbandas (blanqueado en frecuencia).

En este trabajo se presenta la extensión de este codificador al caso de voz de ancho de banda de 7KHz, es decir, de calidad conversacional, adecuada para aplicaciones multimedia. En este caso, los requerimientos de calidad obligan a trabajar a velocidades de 1 a 2 bits/muestra (de 16 a 32 Kbps con una frecuencia de muestreo de 16Khz). En este caso, el número de subbandas en que se divide el margen de 0 a 8Khz de la señal es de 16, siendo todas ellas de 500Hz de anchura, y despreciándose las dos subbandas superiores debido a su bajo contenido energético. En este caso, la mayor precisión de representación de las muestras en cada subbanda proporciona un mejor aprovechamiento del potencial de blanqueamiento de la predicción adaptativa, lo que aconseja el uso de ésta solamente en las primeras subbandas, las de mayor contenido energético. Por otro lado, la cuantificación vectorial de las 14 componentes correspondientes a cada una de las subbandas útiles no puede realizarse en bloque, ya que el coste computacional es inabordable: ¡cuantificación VQ de un vector de dimensión 14 mediante un codebook de 2^{14} a 2^{28} palabras codigo! Para soslayar este problema se hace uso de un cuantificador multi-VQ, consistente en la división del vector total en varios subvectores de dimensiones adecuadas para que su cuantificación requiera una complejidad moderada. Estos subvectores, y su consiguiente cuantificación, puede hacerse de dos formas diferentes: 1) dimensiones fijas de los subvectores, con una asignación dinámica de bits entre éstos; 2) dimensiones variables de los subvectores tal que sus energías sean similares y sea posible una asignación uniforme de bits. En ambos casos, la asignación de bits se basa en las

Este trabajo ha sido subvencionado por la CICYT TIC95-1022-C05-03

estimaciones backward de la energia o ganancia de cada subbanda, información disponible en el codificador y en el decodificador, lo que no requiere uso de información lateral. Además, una mejora de la calidad subjetiva de la voz puede obtenerse mediante un conformado espectral de ruido, el cual se obtiene introduciendo una ponderación de la ganancia en cada una de las subbandas, lo que equivale a una ponderación espectral.

2. ESTRUCTURA BÁSICA DEL CODIFICADOR APVQ

En la Fig.1 se han representado los esquemas correspondientes al Codificador-Decodificador APVQ. Un banco de Filtros QMF de cuatro etapas separa la señal de voz de banda ancha x(n) en 16 subbandas distintas $x_i(n)$ cuyo ancho de banda es 500Hz. Tal como se ha mostrado en trabajos anteriores [3], la capacidad de blanqueado debida a la predicción adaptativa PRED solamente se aprovecha en las primeras 10 subbandas. Para la subbanda i-ésima la estimación de $x_i(n)$ origina por sustracción un valor del error de predicción $e_i(n)$, reduciéndose el márgen dinámico y las redundancias de la señal a presentar al Cuantificador Vectorial. Evidentemente las características de la señal de voz por subbanda $x_i(n)$ varían considerablemente de una subbanda a otra. Por esta razón se ha diseñado un predictor PRED específico para cada una de las 10 primeras subbandas. El predictor seleccionado ha sido un algoritmo GAL en estructura 'backward' para evitar la transmisión de información lateral. Los valores de los parámetros diseñados para cada subbanda, así como las prestaciones obtenidas en términos de Ganancia de Predicción, se presentaron en [3]. En la Fig.1 se puede apreciar la supresión del predictor de voz PRED en las 6 subbandas superiores, puesto que en dichas subbandas el error de cuantificación enmascara la capacidad de blanqueado relacionada con la predicción temporal, originándose valores de Ganancia de Predicción cercanos a 0dB o incluso ganancias negativas. Además, las dos últimas subbandas $x_{15}(n)$ y $x_{16}(n)$ no se procesan debido a su despreciable contenido energético. Esto permite un relativo ahorro operacional y origina una menor dimensión vectorial sin una pérdida apreciable en la calidad subjetiva del codificador APVQ.

Seguidamente la señal error de predicción se cuantifica vectorialmente: el vector de señal se forma a partir de una muestra procedente de cada subbanda. Se considera un cuantificador vectorial adaptativo consistente en una estructura Gain-Shape: la señal a codificar se normaliza previamente a su cuantificación y el error de predicción normalizado $d_i(n)$ es la señal a cuantificar por el VQ. Esta normalización permite reducir el márgen dinámico de la señal a cuantificar y consecuentemente mejora la calidad de la cuantificación y ofrece robustez frente a cambios de nivel en la potencia de la señal entrante. De este modo, se presenta al VQ el error de predicción normalizado $d_i(n)$ pero su factor de normalización debe considerarse durante la etapa de diseño del codebook, pues el error de cuantificación originado en cada componente del vector se magnifica (o se reduce) por este factor de Ganancia. Debe remarcarse que esta normalización se realiza de forma independiente para cada subbanda y ello permite adecuar el cuantificador vectorial a las diferencias relativas del nivel de potencia entre las distintas subbandas. Además, su estructura 'backward' (ver bloque G en la Fig.1) evita la necesidad de una transmisión de información lateral. Como algoritmo de predicción G se ha considerado una simple estimación recursiva de un solo polo. Sus buenas prestaciones y su simplicidad han conducido a descartar el uso de técnicas de predicción más sofisticadas. Aunque las señales correspondientes a cada subbanda presentan

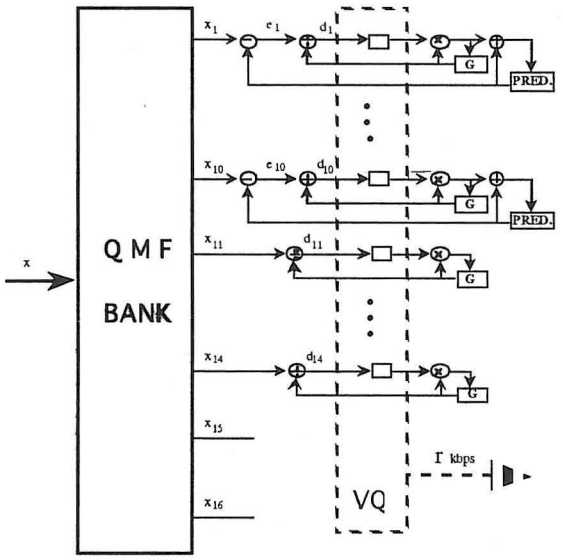

Figura 1.a: Esquema del codificador APVQ.

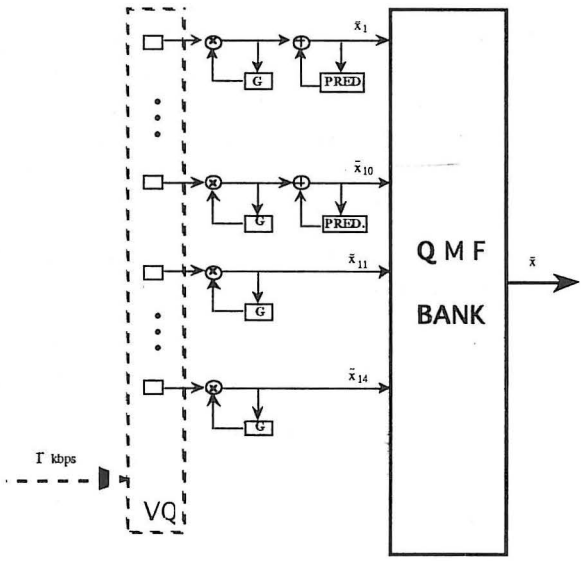

Figure 1.b: Esquema del decodificador APVQ.

características muy diferentes, las señales blanqueadas (error de predicción) presentan características bastante similares, de tal manera que el diseño apropiado para el estimador de ganancia ha resultado el mismo para todas la subbandas. Con un valor óptimo β_i=0.88 se han obtenido valores de Ganancia de Predicción global entre 16.5 y 18.8dB y valores de Ganancia de Predicción segmentada entre 14 y 18dB en las distintas subbandas evaluadas para 16 locutores distintos.

2.1. Diseño de los Codebooks

Todo el proceso descrito previamente presenta claramente una estructura escalar e incluso se podría interpretar como una estructura ADPCM para cada subbanda. Sin embargo el cuantificador presenta una estructura vectorial y a su entrada van apareciendo vectores de señal de dimensión 14 donde cada componente procede de una subbanda diferente:

$$\underline{v}(n) = [\, d_1(n), d_2(n), \ldots, d_{14}(n) \,] \qquad (1)$$

Tal como se ha mencionado anteriormente, el diseño de un codebook de dimensión 14 para un márgen de velocidades comprendidas entre 16 y 32 kbps presenta una complejidad operacional excesiva. En consecuencia, se impone la consideración de una partición del vector de señal $\underline{v}(n)$ en m subvectores distintos $\underline{v}_i(n)$:

$$\underline{v}(n) = [\, \underline{v}_1(n), \underline{v}_2(n), \ldots, \underline{v}_m(n) \,] \qquad (2)$$

y se procede al diseño de un codebook para cada subvector, es decir, se considera una técnica Multi-VQ [4] donde cada subvector se cuantifica de forma independiente. Esta segmentación vectorial equivale a una solución subóptima pero la pérdida de calidad no es apreciable siempre que dicha segmentación y su asociada asignación dinámica de bits sean realizadas adecuadamente.

Una medida para evaluar la complejidad de un codebook puede definirse como:

$$C_i = k_i \cdot 2^{k_i \cdot r_i} \qquad ,, \qquad i=1, \ldots, m \qquad (3)$$

donde k_i representa la dimensión del subvector $\underline{v}_i(n)$ y r_i la velocidad promedio (en bits/muestra) asignada al subvector $\underline{v}_i(n)$. Para el diseño de los distintos codebooks se ha estipulado un valor máximo para dicha complejidad ($C_i \leq 3072$) y se ha considerado la primera técnica de asignación dinámica de bits entre los distintos subvectores debido a su menor complejidad: subvectores de dimensión fija se reparten de forma variable el número total de bits disponibles para cada vector. De este modo el diseño de los codebooks puede resumirse en las dos primeras etapas mientras las etapas 3ª y 4ª permiten la realización de una cuantificación vectorial adecuada:

Etapa 1ª : estimación de la mejor segmentación vectorial a partir de una base de datos de entrenamiento suficientemente grande.

Etapa 2ª : (para cada subvector) diseño de codebooks con tamaños entre 8 y 1024 centroides.

Etapa 3ª : asignación dinámica de bits a cada subvector $\underline{v}_i(n)$, en términos de velocidad r_i (en bits/muestra):

$$r_i = r + \delta_i + \frac{1}{2} \cdot \log_2 \frac{\left(\prod_{j=1}^{k_i} \sigma_{ij}^2 \right)^{\frac{1}{k_i}}}{\left(\prod_{j=1}^{m} \prod_{h=1}^{k_i} \sigma_{jh}^2 \right)^{\frac{1}{k}}} \qquad (4)$$

donde r es la velocidad disponible en bits/muestra, m es el numero de subvectores y σ_{ij}^2 representa la energia promedio de la componente j-ésima correspondiente al subvector i-ésimo [3].

Etapa 4ª : Selección del codebook de tamaño $S_i = 2^{k_i \cdot r_i}$ adecuado para cada subvector.

Nótese que este proceso representa una asignación de bits adaptativa, vector a vector, donde el número total de bits por vector se distribuye de forma entera entre los distintos m subvectores. Además se considera un tamaño mínimo de codebook para evitar pérdidas en el bucle 'backward' del codificador APVQ. El diseño de los codebooks, especificado en la 2ª Etapa, se ha realizado aplicando el algoritmo LBG a partir de un codebook inicial del mismo tamaño. Para la obtención de este codebook inicial se ha usado una técnica presentada en [3] como KUO_U, cuya ventaja más significativa es su bajo coste operacional en comparación a la técnica clásica de inicialización por Splitting, especialmente cuando el tamaño del codebook no es pequeño. Además se ha considerado una ponderación espectral del ruido de cuantificación para mejorar la calidad subjetiva de la voz. Esta ponderación espectral trata de garantizar para cualquier frecuencia que el nivel de ruido nunca supera al nivel de señal [3].

3. RESULTADOS

Durante el diseño de los distintos bloques que corforman este codificador APVQ extendido se ha utilizado una base de datos 'inside' compuesta por frases distintas correspondientes a 8 mujeres y 8 hombres distintos. Para evaluar sus prestaciones, ante señal no presente durante el diseño de los codebooks, se ha considerado también una base de datos 'outside' compuesta por 16 locutores, de los cuales 8 son comunes a ambas bases aunque pronunciando frases distintas. Un detallado estudio acerca de la segmentación vectorial ha conducido a seleccionar, a priori, un reducido grupo de posibles particiones. Tomando como nomenclatura de una partición la dada por las dimensiones de sus subvectores k_1-k_2-..-k_m se han evaluado:

Particion (1) segmentación del vector $\underline{v}(n)$ en m=4 subvectores $\underline{v}_1(n)=[d_1(n), d_2(n)]$, $\underline{v}_2(n)=[d_3(n), d_4(n), d_5(n)]$, $\underline{v}_3(n)=[d_6(n), .. , d_9(n)]$, $\underline{v}_4(n)=[d_{10}(n), .. , d_{14}(n)]$. También denominada partición 2-3-4-5.

Particion (2) segmentación de $\underline{v}(n)$ en m=4 subvectores $\underline{v}_1(n)=[d_1(n), d_2(n)]$, $\underline{v}_2(n)=[d_3(n), d_4(n)]$, $\underline{v}_3(n)=[d_5(n), .. , d_7(n)]$, $\underline{v}_4(n)=[d_8(n), .. , d_{14}(n)]$. También denominada partición 2-2-3-7.

Particion (3) segmentación de $\underline{v}(n)$ en m=3 subvectores $\underline{v}_1(n)=[d_1(n), .. , d_3(n)]$, $\underline{v}_2(n)=[d_4(n), .. , d_6(n)]$, $\underline{v}_3(n)=[d_7(n), .. , d_{14}(n)]$. También referida como partición 3-3-8.

En la Tabla 2 se han representado las prestaciones obtenidas con las particiones más adecuadas para cada velocidad de transmisión r en kbps, comparando la señal de voz original y la voz reconstruida a la salida para ambas bases de datos. A 32 kbps la partición (1) ofrece una mayor calidad subjetiva aunque en algunos vectores las subbandas inferiores reclaman más bits y por ello se ha evaluado una combinación adaptativa de las dos primeras particiones, comprobándose que alrededor de un 15% de los vectores seleccionan la partición (2). La calidad subjetiva es muy buena cuando se consideran las particiones (1) o (1)+(2) para velocidades entre 28 y 32 kbps. Al disminuir la velocidad se reduce el número de bits disponibles y consecuentemente también se reduce el número de subvectores. Así a 20 y 24 kbps la partición (3) supera claramente a la partición (1) mientras a 16 kbps la partición 6-8 ofrece mejores prestaciones en comparación a las segmentaciones 4-10 y 3-3-8. La partición 3-4-7 también ofrece un comportamiento muy similar a (3) para velocidades alrededor de los 20 kbps.

4. CONCLUSIONES

Se ha propuesto una técnica de codificación de voz en banda ancha para el márgen de velocidades comprendidas entre 1 y 2 bits por muestra. Esta codificación APVQ extendida combina las técnicas de Codificación Vectorial en Subbandas y las de Predicción Lineal adaptativa. La cuantificación vectorial se realiza tomando una muestra de cada subbanda, permitiendo una predicción lineal escalar en cada una de las subbandas. Se ha considerado una estrategia Multi-VQ para evitar la enorme complejidad de un único VQ operando sobre todas las subbandas. Se han propuesto particiones vectoriales adecuadas para cada velocidad de transmisión así como las prestaciones alcanzadas por éstas.

5. REFERENCIAS

[1] E.Masgrau, J.B.Mariño. "Subband splitting, adaptive scalar prediction and vector quantization for Speech Encoding". Proc. EUSIPCO, pp.1035-1038. Grenoble, Francia. Septiembre 1988.

[2] I.Katsavounidis, C.C.J.Kuo, Z.Zhang. "A new Initialization Technique for generalized Lloyd Iteration". IEEE Sig. Proc. Letters, Vol. 1, No. 10. Octubre 1994.

[3] J.M.Salavedra, E.Masgrau, A.Cervantes. "Codificación APVQ de voz de Banda Ancha usando Asignación Dinámica de bits". Proc. URSI, pp. 69-72. Valladolid. Septiembre 1995.

[4] T.Moriya, M.Honda. "Transform coding of Speech with Weighted Vector Quantization". Proc. IEEE ICASSP, pp. 1629-1632. Dalias, TX, USA. Abril 1987.

Partición	r	SNR_{ov}	SNR_{seg}	Itakura	Cosh	Cepstrum
(1)	32	20.58	22.41	0.32	3.03	2.86
(2)	32	21.33	22.44	0.54	4.44	4.54
(1) + (2)	32	22.67	22.85	0.31	3.04	2.99
(1)	28	19.88	20.43	0.42	3.41	3.37
(1)	26	18.21	18.92	0.52	3.43	3.45
(1)	24	15.87	17.03	0.65	3.56	3.68

Tabla 2.a: Resultados para la base de datos 'inside'.

Partición	r	SNR_{ov}	SNR_{seg}	Itakura	Cosh	Cepstrum
(1)	32	24.11	24.42	0.32	3.20	3.01
(2)	32	24.37	24.77	0.47	3.86	4.01
(1) + (2)	32	25.00	24.99	0.32	3.23	3.13
(1)	28	22.13	22.78	0.42	3.70	3.61
(1)	26	20.67	21.82	0.48	3.73	3.67
(1)	24	17.69	19.03	0.61	3.78	3.91

Tabla 2.b: Resultados para la base de datos 'outside'.