Exhibit O

COMPLEXITY REDUCTION METHODS FOR VECTOR EXCITATION CODING

Grant Davidson and Allen Gersho
Department of Electrical and Computer Engineering

University of California, Santa Barbara, CA 93106

Abstract

Vector Excitation Coding (VXC) is based on a new and general source-filter modeling technique in which the excitation signal for a speech production model is encoded at very low bit-rates using vector quantization. Various speech coder structures which fall into this class have recently been shown to reproduce speech with very high perceptual quality.

The primary drawback of VXC is the large amount of computation required in the process of selecting an optimal excitation signal. We present several schemes in this paper which substantially reduce search computation in VXC coders while retaining their remarkably high reconstructed speech quality.

1. INTRODUCTION

Excitation Coding (XC) is a convenient name for a powerful new approach to speech coding at medium to low bit-rates and with very high perceptual quality. XC is based on a source-filter synthesis model as in LPC but is distinguished by the use of an analysis-by-synthesis technique and a perceptually weighted mean-square error measure for selecting the appropriate excitation (source) for each frame. The basic approach was introduced by Atal [1] and subsequently led to Multi-Pulse LPC (MPLPC) [2].

A particularly effective form of XC uses vector quantization to encode the excitation signal at very low bit-rates (typically 0.25 bits per sample). We refer generically to such coding techniques as Vector Excitation Coding (VXC). Two examples of VXC coders are [3] and [4]. In the latter paper, the term CELP has been used to describe the coding technique which uses Gaussian random variables for the codevector components. In the encoding process for VXC, high-dimensional excitation vectors from a codebook are input to the speech production model to generate a set of synthetic speech vectors. An optimal excitation vector is selected which produces minimal perceptually-weighted error between the synthetic and input vectors. Nearly transparent synthetic speech is achieved at rates in the neighborhood of 5 kbits/s due in part to the spectral noise masking effect of the error weighting mechanism.

The primary disadvantage of VXC is the very high computational complexity associated with the selection of an optimal excitation signal. In CELP, for example, an exhaustive search algorithm with a codebook of size 1024 and dimension 40 requires more than 50,000 multiply/adds per input speech sample. This operation count is much higher than can be reasonably achieved by conventional implementations using today's DSP chips or special-purpose VLSI processors.

We have developed two techniques for reducing the complexity of VXC without sacrificing the perceptual quality of the reconstructed speech signal. Both of these new codebook search schemes streamline the procedure for selecting an optimal excitation codevector, and each results in a factor of approximately ten reduction in overall search computation compared to the CELP algorithm described in [4].

2. VECTOR EXCITATION CODING

As a first step toward the goal of complexity reduction, we identify a VXC structure which is amenable to the incorporation of codebook fast-search methods. As a secondary advantage, this structure requires less computation than the original CELP structure even though it is conceptually equivalent to it.

A block diagram of the VXC encoder we consider in this paper is presented in Figure 1. The original speech input $\mathbf{s}_n$ is a vector with a nominal dimension of $k = 40$ samples. This vector is filtered by a time-varying perceptual weighting filter $W(z)$ and then subtracted from each member of a set of N weighted synthetic speech vectors $\{\hat{\mathbf{s}}_j\}$, $j \in \{1,...,N\}$. The set $\{\hat{\mathbf{s}}_j\}$ is generated by filtering Gaussian-like codevectors $\mathbf{c}_j$ with cascaded long-term and short-term weighted synthesis filters $H_l(z)$ and $\bar{H}_s(z)$. Each codevector is scaled by a gain G_j which is determined by minimizing the mean-squared error between $\hat{\mathbf{s}}_j$ and the weighted input speech vector.

In an exhaustive search VXC coder of this type, an excitation vector $\mathbf{c}_j$ is selected which minimizes the squared Euclidean distance $\| \tilde{\mathbf{e}}_j \|^2$ between preprocessed $\mathbf{s}_n$ vectors and every member of $\{\hat{\mathbf{s}}_j\}$. An index I_n having $\log_2 N$ bits which identifies the optimal $\mathbf{c}_j$ is transmitted for each input vector along with G_j and the synthesis filter parameters asso-

This work was performed in part for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration.

CH2243-4/86/0000-3055 $1.00

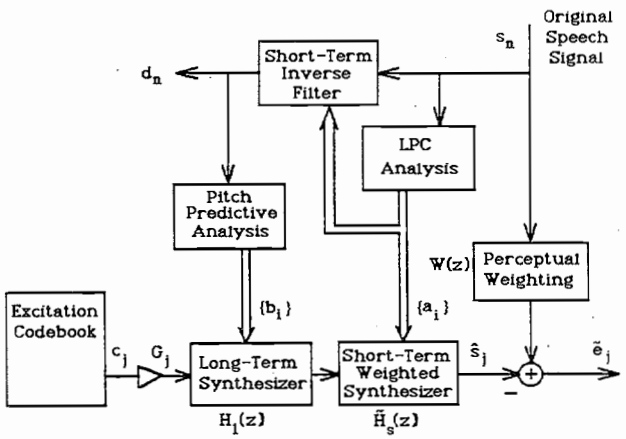

Figure 1. A VXC Structure

ciated with the current input frame.

The transfer functions of the time-varying recursive filters $\tilde{H}_s(z)$, $H_l(z)$, and $W(z)$ are given by

$$\tilde{H}_s(z) = \frac{1}{P(z/\gamma)}$$

$$H_l(z) = \frac{1}{B(z)} \qquad (1)$$

$$W(z) = \frac{P(z)}{P(z/\gamma)},$$

where $P(z) = 1 + \sum_{i=1}^{p} a_i z^{-i}$, $B(z) = 1 + \sum_{i=-J}^{J} b_i z^{-L-i}$, the a_i are LPC predictor coefficients obtained by the stabilized covariance method [1] of order p, the b_i are predictor coefficients of a long-term LPC covariance analysis of order $q = 2J + 1$, and the lag term L can roughly be described as the sample delay corresponding to one pitch period. The parameter γ $(0 < \gamma < 1)$ determines the amount of perceptual weighting applied to the error signal.

In Figure 1, we see that $W(z)$ has been moved from its conventional location (at the output of the error subtraction operation) to both of its input branches. In this case, $\mathbf{s}_n$ will be weighted only *once* by $W(z)$ (prior to the start of a codebook search). Another desirable effect of moving $W(z)$ is that its zeros cancel the poles of the conventional short-term LPC filter $1/P(z)$, producing the *pth* order weighted synthesis filter $\tilde{H}_s(z)$. This representation requires a factor of 3 less computation per codevector than the conventional approach since only $k(p + q)$ multiply/adds are required for filtering a codevector instead of $k(3p + q)$ when the synthesis and weighting filters are separate.

Computation can be further reduced by removing the effect of the memory in $\tilde{H}_s(z)$ and $H_l(z)$ on the selection of an optimal excitation vector for the current frame. This is accomplished using a very low-complexity technique to preprocess the weighted input speech vector once prior to the subsequent codebook search. The result of this procedure is that the initial memory in these filters can be set to zero when synthesizing $\{\hat{\mathbf{s}}_j\}$ without affecting the choice of the optimal codevector. Once the optimal codevector is determined, the filter memory from the previous frame can be updated for use in the subsequent frame. This approach also allows us to efficiently express the speech synthesis operation as a matrix-vector product, as shown in Section 3.

3. SPARSE VECTOR FAST-SEARCH

The Sparse Vector Fast Search method is motivated by MPLPC. In this method, we develop a new formulation of the LPC synthesis filters and show how a suitable algebraic manipulation and an appropriate but modest constraint on the Gaussian-like codevectors leads to an overall reduction in codebook search complexity by a factor of approximately ten. The complexity reduction factor can be increased by varying a parameter of the codebook design process. The result is that the performance versus complexity characteristic exhibits a threshold effect that allows a substantial complexity saving before any perceptual degradation in quality is incurred. A side-benefit of this technique is that memory storage for the excitation vectors is reduced by a factor of 7 or more.

In Section 2, we noted that memory terms in the infinite impulse response (IIR) filters $\tilde{H}_s(z)$ and $H_l(z)$ can be set to zero prior to synthesizing $\{\hat{\mathbf{s}}_j\}$. This implies that the output of the IIR filters can be expressed as a convolution of two *finite* sequences of length k:

$$\hat{s}_j(m) = h(m) * c_j(m), \qquad (2)$$

where $\hat{s}_j(m)$ is a sequence of weighted synthetic speech samples, $h(m)$ is the impulse response of the combined short-term and long-term filters, and $c_j(m)$ is a sequence of samples from the *jth* excitation vector.

A matrix representation of the convolution in (2) may be given as:

$$\hat{\mathbf{s}}_j = \mathbf{H}\mathbf{c}_j, \qquad (3)$$

where $\mathbf{H}$ is a $k \times k$ lower triangular matrix whose elements are from $h(m)$:

$$\mathbf{H} = \begin{bmatrix} h(0) & 0 & 0 & \cdots & 0 \\ h(1) & h(0) & 0 & \cdots & 0 \\ h(2) & h(1) & h(0) & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & 0 \\ h(k-1) & h(k-2) & h(k-3) & \cdots & h(0) \end{bmatrix} \qquad (4)$$

Now the weighted distortion from the *jth* codevector can be expressed simply as

$$\| \tilde{\mathbf{e}}_j \|^2 = \| \mathbf{s}_n - \hat{\mathbf{s}}_j \|^2 = \| \mathbf{s}_n - \mathbf{H}\mathbf{c}_j \|^2 \qquad (5)$$

In general, the matrix computation to calculate $\hat{\mathbf{s}}_j$ requires $k(k+1)/2$ multiply/adds, versus $k(p+q)$ for the conventional linear recursive filter realization. For our chosen set of filter parameters ($k = 40$, $p + q = 19$), it would be slightly more expensive for an *arbitrary* $\mathbf{c}_j$ to compute $\| \tilde{\mathbf{e}}_j \|^2$ using the matrix formulation since $(k+1)/2 > p + q$. However, if we suitably choose each $\mathbf{c}_j$ to have only N_p pulses per vector (the other components are zero), then (5) can be computed very efficiently. More specifically, if the matrix-vector product $\mathbf{H}\mathbf{c}_j$ is calculated using:

```
For m = 0 to k − 1
    If c_j(m) = 0, then
        Next m
    else
        For i = m to k − 1
            ŝ_j(i) = ŝ_j(i) + c_j(m) h(i)
    endif
```

then the average computation for $\mathbf{H}\mathbf{c}_j$ is $N_p(k+1)/2$ multiply/adds, which is less than $k(p+q)$ if $N_p < 37$ (for the k, p, and q given previously). A very straightforward codebook design procedure exists which uses an initial set of Gaussian vectors to construct a set of pulse excitation codevectors. The complexity reduction factor of this fast-search technique is adjusted by varying N_p, a parameter of the codebook design process.

Zeroing of selected codevector components is consistent with results obtained in MPLPC, since it has been shown that only about 8 pulses are required per pitch period (one pitch period is typically 5 *ms* for a female speaker) to synthesize natural-sounding speech [5]. Even more encouraging, our simulation results indicate that reconstructed speech quality does not start to deteriorate until the number of pulses per vector drops to 2 or 3 out of 40. Since, with the matrix formulation, computation decreases as the number of zero components increases, significant savings can be realized by using only 4 pulses per vector. In fact, when $N_p = 4$ and $k = 40$, filtering complexity reduction by a factor of ten is achieved.

Figure 2 shows plots of segmental SNR (SNR_{seg}) and overall codebook search complexity versus N_p. Observe that as N_p decreases, SNR_{seg} does not start to drop until N_p reaches 2. In fact, informal listening tests show that the perceptual quality of the reconstructed speech signal actually improves slightly as N_p is reduced from 40 to 4, and at the same time the filtering computation drops significantly.

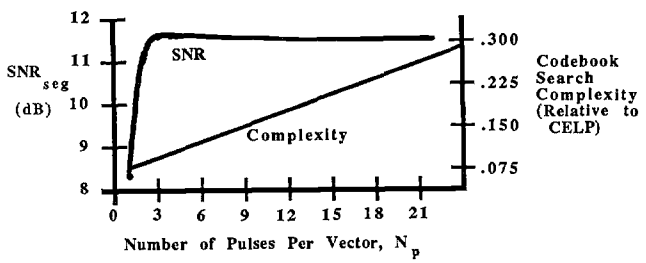

Figure 2. SNR and Codebook Search Complexity vs. N_p

As a final note, observe that the required amount of codebook memory can be greatly reduced by storing only N_p pulse amplitudes and their associated positions instead of k amplitudes (most of which are zero in this scheme anyway). For example, memory storage reduction by a factor of 7.3 is achieved when $k = 40$, $N_p = 4$, and each codevector component is represented by a 16-bit word.

4. SPECTRAL CLASSIFICATION

The second simplification scheme, Spectral Classification, also reduces overall codebook search effort by a factor of approximately ten. It is based on the premise that it is possible to perform a precomputation of simple to moderate complexity using the input speech to eliminate a large percentage of excitation codevectors from consideration before an exhaustive search is performed.

It has been shown [4] that for a given speech frame the number of excitation vectors from a codebook of size 1024 which produce acceptably low distortion is small (approximately 5). Our goal in this fast-search scheme, then, is to use a quick but approximate procedure to find a set of N_c "good" candidate excitation vectors ($N_c \ll N$) for subsequent use in a reduced exhaustive search.

The N_c surviving codevectors are selected as follows. A rough classification of the gain-normalized spectral shape of the current speech frame is made by quantizing its short-term predictor coefficients using a vector quantizer with M spectral shape codevectors (typically $M = 4$ to 8) and which uses the well-known Itakura-Saito distortion measure [6, 7]. The classification operation is a gain-normalized version of the LPCVQ procedure introduced in [8]. In addition, it is very low-complexity (less than .2% of the total codebook search effort). Associated with the *ith* spectral shape class is a precomputed codebook containing vectors generated by shaping the original Gaussian-like excitation codevectors with the *ith* all-pole filter $\tilde{H}_s(z)$ corresponding to that class. By calculating the short-term filtered excitation vectors off-line, this computational expense is saved in the encoder (the short-term filtering and error weighting operation comprises 90% of the total codebook search computation in the original CELP structure). Now the candidate excitation vectors from the original Gaussian-like codebook can be selected simply by filtering the shaped vectors from the selected class with $H_l(z)$, and retaining only those N_c vectors which produce the lowest weighted distortion. The final exhaustive search is conducted using quantized values of the predictor coefficients determined by LPC analysis of the current speech frame.

Computer simulation results show that with $M = 4$, N_c can be as low as 30 with no loss in perceptual quality of the reconstructed speech, and when $N_c = 10$, only a very slight degradation is noticeable. Figure 3 summarizes the results of these simulations by showing how SNR_{seg} and overall codebook search complexity change with N_c. Note that the drop in SNR_{seg} as N_c is reduced does not occur until after the knee of the complexity versus N_c curve is passed.

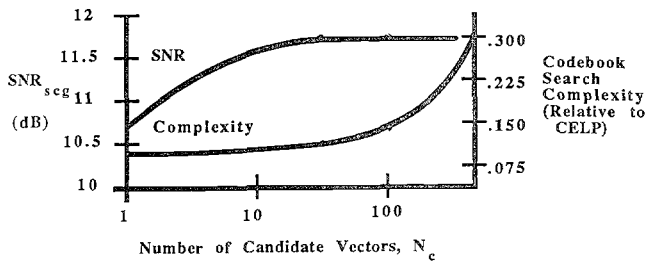

Figure 3. SNR and Codebook Search Complexity vs. N_c

Table I contains a comparative listing of the complete codebook search operation count (in MFlops) for each of the two fast-search methods introduced in this paper. For comparison, the operation count for a CELP structure as described in [4] is also shown. Note that the overall factor of reduction for both fast-search methods is approximately ten.

Table I
Operation Count for VXC with Complexity Reduction

(Million Floating-Point Ops. Per Sec.)

Codebook Search Computation	Complexity Reduction Method		
	SVFS	SC	None (CELP)
Short-Term/ Weight. Filter	20.9	5.4	399
Long-Term Filter		12.4 (.1)	12.3
Gain, Gj	16.4	16.5	16.4
Norm squared of $\tilde{e}_j$	12.3	12.4	12.3
Total	49.6	46.7 (34.4)	440

* SVFS = Sparse Vector Fast-Search, $N_p = 4$
SC = Spectral Classification, $M = 4$, $N_c = 10$

In Table I, the operation count for long-term filtering in the Spectral Classification method is a very conservative estimate. It was derived assuming that the smallest value for the lag term L in $H_l(z)$ ($L = 20$) occurs every speech frame. In most instances, L is considerably greater than 20, so computation will be much less than the 12.4 MFlop value given in Table I. For the case when $L > k$, filtering of the shaped codevectors with $H_l(z)$ is not necessary at all when conducting the codebook search (recall that the filter memory in $H_l(z)$ is set to zero prior to use). For successive frames in which $L > k$, the total operation count for VXC with Spectral Classification is only 34.4 MFlops, as noted in parenthesis in Table I.

5. CONCLUSIONS

The sparse-vector and spectral classification fast codebook search techniques for VXC have each been shown to reduce complexity by an order of magnitude without incurring a loss in subjective quality of the reconstructed speech signal. In the sparse-vector method, a matrix formulation of the LPC synthesis filters is presented which possesses distinct advantages over conventional all-pole recursive filter structures. In spectral classification, we are able to eliminate approximately 97% of the excitation codevectors from the codebook search by using a crude identification of the spectral shape of the current frame. These two methods can be combined together or with other compatible fast-search schemes to achieve even greater reduction, thereby bringing VXC-class coders within the realm of implementation using today's VLSI technology.

Acknowledgement

The authors are grateful to Dr. Ioannis Dologlou for helpful discussions of the Spectral Classification fast-search method presented in this paper.

References

[1] B. S. Atal, "Predictive Coding of Speech at Low Bit Rates," *IEEE Trans. Communications*, Vol. COM-30, No. 4, April 1982.

[2] B. S. Atal and J. R. Remde "A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates," *Proceedings Int'l Conference on Acoustics, Speech, and Signal Processing*, Paris, May 1982.

[3] M. Copperi and D. Sereno, "Vector Quantization and Perceptual Criteria for Low-Rate Coding of Speech," *Proceedings Int'l Conference on Acoustics, Speech, and Signal Processing*, Tampa, March 1985.

[4] M. R. Schroeder and B. S. Atal, "Code-Excited Linear Prediction (CELP): High-Quality Speech at Very Low Bit Rates," *Proceedings Int'l Conference on Acoustics, Speech, and Signal Processing*, Tampa, March 1985.

[5] S. Singhal and B. S. Atal, "Improving Performance of Multi-Pulse LPC Coders at Low Bit Rates," *Proceedings Int'l Conference on Acoustics, Speech, and Signal Processing*, San Diego, March 1984.

[6] F. Itakura and S. Saito, "Analysis Synthesis Telephone Based Upon the Maximum Likelihood Method," *Conf. Record, 6th Int. Congr. Acoust.*, Y. Yonasi, Ed., Tokyo, Japan, 1968.

[7] R. M. Gray et al., "Distortion Measures for Speech Processing," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, Vol. ASSP-28, Aug. 1980.

[8] A. Buzo et al., "Speech Coding Based Upon Vector Quantization," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, Vol. ASSP-28, Oct. 1980.