On-Line and Off-Line Computational Reduction Techniques Using Backward Filtering in CELP Speech Coders

Mark Johnson and Tomohiko Taniguchi

Abstract—**In this correspondence, we review the backward-filtering algorithm, and give a compact proof of its validity using matrix notation. We will review the relation between backward filtering and off-line perceptual weighting in sparse-codebook CELP, and will show how a combination on-line/off-line parallel weighting algorithm can be used to reduce the search complexity of an overlapped sparse codebook by 30% to 50%.**

I. Introduction

Off-line perceptual weighting, and the related backward-filtering algorithm, were originally proposed in simultaneous papers by Adoul *et al.* [1], who coined the term "backward filtering," and by Davidson and Gersho [2], who proposed both algorithms as an enhancement to their sparse-vector fast search (SVFS). Section II will review the SVFS, and show how it reduces computation by moving all perceptual weighting calculations off-line with respect to the VQ codebook search. Section III will give a brief derivation, in matrix notation, of the backward-filtering algorithm, and will describe its relation to the SVFS. Finally, Section IV will focus on a particular CELP coder: the Department of Defense 4.8 kb/s standard coder, as it was originally proposed [3]. Since the DoD coder uses an overlapped codebook, normal on-line perceptual weighting is much more efficient than off-line weighting. We will therefore describe a combined on-line/off-line algorithm, using backward-filtering, which yields a 30% computational reduction in the DoD search complexity.

II. Off-Line Perceptual Weighting

Fig. 1 shows the structure of a typical CELP speech coder, capable of representing most of the algorithms in current use. As shown, a typical CELP coder processes the input speech signal S before vector quantizing it, first by filtering it to emphasize the perceptually important frequency bands, and second by removing the pitch periodicity using a long-term predictor of some sort, usually a closed-loop "adaptive codebook" predictor [4] as shown in the figure. The pitch-removed residual signal is then used as the "target vector" X for a weighted-squared-error gain-shape vector quantizer based on a stochastic codebook of more or less uncorrelated random codevectors.

The perceptual weighting filter in most CELP coders consists of an LPC analysis filter of order N_P, $A(z)$, in series with a weighted LPC synthesis filter, $1/A'(z)$, where

$$A(z) = 1 - \sum_{i=1}^{N_P} \alpha_i z^{-i}, \qquad A'(z) = 1 - \sum_{i=1}^{N_P} \alpha_i \gamma^i z^{-i}, \qquad \gamma \approx 0.9. \tag{2.1}$$

The codevectors C are designed to model the noise-like characteristics of the unweighted LPC residual signal. Because of this, finding the optimum codevector involves filtering each vector C with the weighted LPC synthesis filter, $1/A'(z)$, to create a weighted code vector.

Passing an N-sample vector C through an IIR filter with an initial state of zero, such as $1/A'(z)$, gives the same result, assuming proper handling of the filter memory at frame boundaries, as multiplying C by the FIR filter matrix

$$H = \begin{bmatrix} h_1 & 0 & \cdots & 0 \\ h_2 & h_1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ h_N & h_{N-1} & \cdots & h_1 \end{bmatrix} \tag{2.2}$$

where h_i is the ith sample of the impulse response of $1/A'(z)$ [2]. Since these two forms are functionally equivalent, we lose nothing by thinking of the filtering operation as a matrix multiplication, and by writing the weighted codevector as a matrix product HC. This gives us the opportunity to manipulate the weighting filter, in our equations, as if it were a triangular Toeplitz matrix, even if the actual implementation of the filter is IIR.

If our coder is capable of quantizing the stochastic codebook gain g at a close approximation to its optimum value, then choosing a code-vector C to minimize the weighted quantization error, $E = X - gHC$, involves maximizing the function $F(X, C)$ according to the block diagram in Fig. 2, where $F(X, C)$ can be written as

$$F(X, C) = \frac{R_{XC}^2}{R_{CC}} \tag{2.3}$$

$$R_{XC} = X^T(HC), \qquad R_{CC} = (HC)^T(HC). \tag{2.4}$$

The correlation term R_{XC} and the energy term R_{CC}, as defined above, both require a vector product involving the perceptually weighted code vector, HC, as shown in Fig. 2.

If the weighting filter can be manipulated as a matrix, then a simple expansion of (2.4) will give the following:

$$R_{XC} = (HC)^T X = C^T(H^T X) \equiv C^T Z \tag{2.5}$$

$$R_{CC} = (HC)^T HC = C^T(H^T H)C \equiv C^T \Gamma C. \tag{2.6}$$

In effect, (2.5) and (2.6) show how we can implement the perceptual weighting operation off-line, by "weighting" X and H, instead of the codevectors C. This potential for off-line computation is a property of the correlation operator, and is independent of the structures of H, X, or C. Unfortunately, the weighted autocorrelation $C^T\Gamma C$, which must now be computed on-line, will require twice as much computation, in the general case, as the matrix multiplication HC, simply because H is lower triangular, while Γ is not. This means that even if it is always possible, (2.6) is often not practical.

Off-line filtering is often practical, however, if C is sparse, that is, if many of its samples are zero valued. The complexity of the FIR product HC decreases in direct proportion to the sparsity of C (defined as the percentage of nonzero samples), while the complexity of $C^T\Gamma C$ decreases as the square of the sparsity. The correlation $C^T\Gamma C$ will therefore be more efficient than the product HC for almost any code vector of 50% or lower sparsity (Section IV will discuss an exception). If C is sparse, (2.5) and (2.6) describe the enhanced sparse-vector fast search proposed by Davidson and Gersho [2].

Off-line computation is also valid if the codebook search criterion (2.3) is modified to allow simultaneous optimization of the

Manuscript received March 19, 1990; revised February 29, 1992.

M. Johnson was with the Speech Signal Processing Group, Fujitsu Laboratories, Ltd., Nakahara-ku, Kawasaki 211, Japan. He is now with the Speech Group, Research Laboratory for Electronics, M.I.T., Cambridge, MA 02139.

T. Taniguchi is with the Speech Signal Processing Group, Fujitsu Laboratories Ltd., Nakahara-ku, Kawasaki 211, Japan.

IEEE Log Number 9201058.

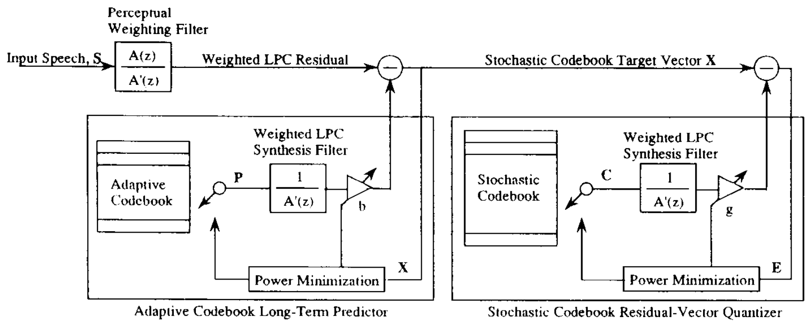

Fig. 1. Standard CELP: Overall structure.

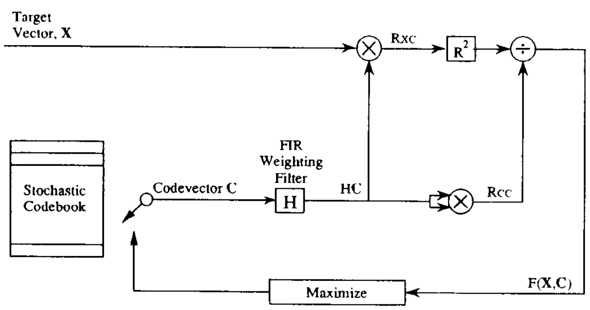

Fig. 2. Standard CELP: Stochastic codebook search.

adaptive and stochastic codebook gains. Equations for this algorithm are given elsewhere ([6], for example), and will not be repeated here, but the most computationally significant new operation is the correlation of each weighted stochastic code vector with a previously chosen weighted pitch. The perceptual weighting involved in this operation can also be moved off-line, exactly as shown in (2.5):

$$R_{PC} = (HC)^T(HP) = C^T(H^THP). \qquad (2.7)$$

III. The Backward-Filtering Algorithm

The backward-filtering algorithm is an interesting technique for efficiently computing the off-line matrix multiplications shown in (2.5)-(2.7). A derivation is given in [1]; we will give a more compact derivation, in matrix notation.

It can be easily shown that the product F^TG, for any matrices F and G of dimension, say, $n \times m$ and $n \times p$

$$F^TG = \begin{bmatrix} f_{11} & \cdots & f_{n1} \\ \cdots & \cdots & \cdots \\ f_{1m} & \cdots & f_{nm} \end{bmatrix} \begin{bmatrix} g_{11} & \cdots & g_{1p} \\ \cdots & \cdots & \cdots \\ g_{n1} & \cdots & g_{np} \end{bmatrix}$$

$$= \begin{bmatrix} \Sigma f_{i1}g_{i1} & \cdots & \Sigma f_{1i}g_{ip} \\ \cdots & \cdots & \cdots \\ \Sigma f_{im}g_{i1} & \cdots & \Sigma f_{im}g_{ip} \end{bmatrix} \qquad (3.1)$$

can be computed by transposing F about its antidiagonal ($F^{\perp}$), time reversing G (G_{tr}), multiplying the two, and time reversing the result:

$$F^{\perp}G_{\text{tr}} = \begin{bmatrix} f_{nm} & \cdots & f_{1m} \\ \cdots & \cdots & \cdots \\ f_{n1} & \cdots & f_{11} \end{bmatrix} \begin{bmatrix} g_{n1} & \cdots & g_{np} \\ \cdots & \cdots & \cdots \\ g_{11} & \cdots & g_{1p} \end{bmatrix}$$

$$= \begin{bmatrix} \Sigma f_{im}g_{i1} & \cdots & \Sigma f_{im}g_{ip} \\ \cdots & \cdots & \cdots \\ \Sigma f_{i1}g_{i1} & \cdots & \Sigma f_{i1}g_{ip} \end{bmatrix} = (F^TG)_{\text{tr}}. \qquad (3.2)$$

The matrix H defined in (2.2) is symmetric about its antidiagonal, $H^{\perp} = H$, so, using (3.2), we can rewrite (2.5) and (2.6) as

$$Z = H^TX = (HX_{\text{tr}})_{\text{tr}} \qquad (3.3)$$

$$\Gamma = H^TH = (HH_{\text{tr}})_{\text{tr}}. \qquad (3.4)$$

Multiplication by H, however, is functionally equivalent to filtering by $1/A'(z)$, so that we can implement the right side of (3.4) and (3.5) by filtering X_{tr} and the columns of H_{tr} with an IIR filter. In a normal CELP coder, the order of the IIR filter $1/A'(z)$ is about one fourth of the dimension of H; under these circumstances, we can compute the matrix multiplications H^TX and H^TH with 50% fewer scalar multiplications if we use IIR backward filtering. If H^TX and H^TH are computed off-line, even a savings of 50% amounts to only about half a multiplication per code vector, making IIR backward filtering several orders of magnitude less important than the original decision to take perceptual weighting off-line. For convenience, however, the next section will assume that any off-line multiplications of the form H^TX are computed using IIR backward filtering.

IV. Backward Filtering in Overlapped-Codebook CELP

In general, off-line perceptual weighting is profitable for codebooks of 50% or lower sparsity. Some coders, however, use computational tricks to reduce the complexity of the FIR filter HC, and many of these tricks can not be used in the same way with the correlation $C^T\Gamma C$. In such coders, the most computationally efficient codebook search may use a combination of on-line and off-line perceptual weighting.

One such technique uses an overlapped stochastic codebook [5]. In an overlapped codebook, each code vector is a copy of the previous vector, shifted by one or two samples, and with the empty places filled with appropriate random numbers. If this is done correctly, each weighted vector HC can be computed very efficiently by shifting the previous weighted vector, and filtering the new samples. There is no corresponding trick for simplifying the correlation $C^T\Gamma C$, so an overlapped codebook is almost always easiest to search using on-line FIR perceptual weighting.

A good example of a CELP coder with an overlapped codebook is the U.S. Department of Defense 4.8 kb/s standardized coder, as originally proposed in [3]. The DoD coder uses a stochastic codebook constructed from Gaussian random vectors center clipped

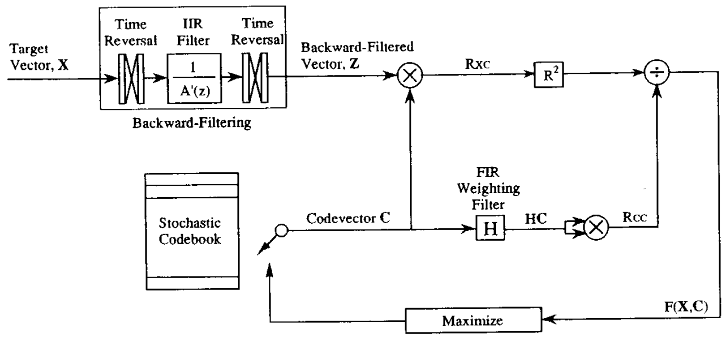

Fig. 3. Backward filtering and the on-line/off-line weighting calculation, in sparse-overlapped-codebook CELP.

TABLE I
MULTIPLICATIONS PER STOCHASTIC CODEVECTOR IN SPARSE-CODEBOOK CELP: COMPUTATIONAL REDUCTION VIA BACKWARD FILTERING OF THE INPUT

Stochastic Codebook Structure	Gain Optimization Technique	Filter	R_{XC}, (R_{PC})	R_{CC}	Total, $N = 40$	ΓC or Filter	R_{XC}, (R_{PC})	R_{CC}	Total, $N = 40$	Total Reduction ($N = 40$)
Sparse Codebooks:		On-Line Weighting					Off-Line Weighting			
1/4 Sparse	Sequential	$N^2/8$	N	N	280	$N^2/16$	$N/4$	$N/4$	120	57%
	Simultaneous	$N^2/8$	$2N$	N	320	$N^2/16$	$N/2$	$N/4$	130	59%
1/5 Sparse	Sequential	$N^2/10$	N	N	240	$N^2/25$	$N/5$	$N/5$	80	67%
	Simultaneous	$N^2/10$	$2N$	N	280	$N^2/25$	$2N/5$	$N/5$	88	69%
Overlapped-Sparse:		On-Line Weighting					Combined On-Line/Off-Line			
1/4 Sparse, Shifted-2	Sequential	$N/2$	N	N	100	$N/2$	$N/4$	N	70	30%
	Simultaneous	$N/2$	$2N$	N	140	$N/2$	$N/2$	N	80	43%
1/5 Sparse, Shifted-2	Sequential	$N/5$	N	N	88	$N/5$	$N/5$	N	56	36%
	Simultaneous	$N/5$	$2N$	N	128	$N/5$	$2N/5$	N	64	50%

at 1.2, making them 1/4 sparse, and with a 2-sample recursive shift between code vectors. The combination of code vector sparsity and overlapping reduces the complexity of filtering an N-sample code vector from $N^2/2$ to only $N/2$ multiplications per code vector.

Regardless of the structure of the stochastic codebook, however, the resulting filtered codevectors $\boldsymbol{HC}$ will be neither sparse nor overlapped, so that finding the correlation and energy terms directly, as shown in (2.4), will require two full vector products, for a full N multiplications each. In the DoD system, this means that each of the two inner products requires twice the computation of the entire weighting filter.

If $\boldsymbol{C}$ is sparse, however, the correlation of $\boldsymbol{C}$ with the backward-filtered vector $\boldsymbol{Z}$, as in (2.5), requires less than N multiplications, in proportion to the sparsity of the code vector. This means that if we backward filter $\boldsymbol{X}$ at the beginning of every frame, off-line, we can use the sparsity of $\boldsymbol{C}$ to cut down the complexity of R_{XC}. Since (2.6) is impractical, we still need to calculate $\boldsymbol{HC}$ on-line, and the inner product of $\boldsymbol{HC}$ with itself to find R_{CC} still requires a full N operations. What we are proposing, however, is a combination on-line/off-line algorithm, in which both the input and the code vector are weighted, and the vectors $\boldsymbol{H}^T\boldsymbol{X}$ and $\boldsymbol{HC}$ are used in parallel to compute the correlations R_{XC} and R_{CC} as efficiently as possible, as shown in Fig. 3.

Table I lists eight CELP coders, four with sparse codebooks, and four with sparse overlapped codebooks. The complexity of searching each overlapped codebook is listed for standard on-line perceptual weighting, and for the combination on-line/off-line approach described in the previous paragraph. For comparison, the complexity of searching each nonoverlapped codebook is listed using fully on-line perceptual weighting, and fully off-line weighting, as described in Section II. Off-line weighting of a nonoverlapped codebook always gives a better percentage improvement than combined weighting of an overlapped codebook, but the percentages provided by combined weighting are still substantial, and overlapped codebooks still provide the lowest total complexities. The DoD standard coder, with 1/4 sparsity, a 2-sample recursive shift, and sequential optimization of the adaptive and stochastic codebook gains, achieved a 30% computational reduction using the combination algorithm, and, as high as this number is, it was the smallest computational improvement achieved by any of the coders listed.

V. CONCLUSIONS

We reviewed the enhanced sparse-vector fast search, and showed how it can be used, with or without IIR backward filtering, to move all of the perceptual weighting computations in a CELP speech coder off-line with respect to the stochastic codebook search. We then showed that fully off-line perceptual weighting is not profitable when searching an overlapped-sparse codebook, simply because an overlapped codebook can be filtered very efficiently on-line. Despite this, we discovered that backward-filtering can be used to implement a sort of combination on-line/off-line weighting algorithm, in which the sparsity of the overlapped codebook is used to reduce the complexity of one part of the codebook search, while the other parts remain just as they would be in a normal search with on-line filtering.

ACKNOWLEDGMENT

The authors would like to thank the reviewer, who suggested the derivation of backward filtering given in Section III.

REFERENCES

[1] J.-P. Adoul, P. Mabilleau, M. Delprat, and S. Morisette, "Fast CELP coding based on algebraic codes," in *Proc. ICASSP*, Apr. 1987, pp. 1957-1960.
[2] G. Davidson and A. Gersho, "Real-time vector excitation coding of speech at 4800 BPS," in *Proc. ICASSP*, Apr. 1987, pp. 2189-2192.
[3] J. P. Campbell, V. C. Welch, and T. E. Tremain, "An expandable error-protected 4800 bps CELP coder (U.S. Federal Standard 4800 bps Voice Coder)," in *Proc. ICASSP*, Apr. 1989, pp. 735-738.
[4] P. Kroon and E. F. Deprettere, "A class of analysis-by-synthesis predictive coders for high quality speech coding at rates between 4.8 and 16 kbits/s," *IEEE J. Select. Areas Commun.*, vol. 6, no. 2, pp. 353-362, Feb. 1988.
[5] D. Lin, "Speech coding using pseudostochastic block codes," in *Proc. ICASSP*, Apr. 1987, pp. 1354-1357.
[6] T. Taniguchi, Y. Tanaka, A. Sasama, and Y. Ohta, "Principal axis extracting vector excitation coding: High quality speech at 8 kb/s," in *Proc. ICASSP*, Apr. 1990, pp. 241-244.

Construction of a Hermitian Toeplitz Matrix from an Arbitrary Set of Eigenvalues

F. Noor and S. D. Morgera

***Abstract*—In this correspondence, we present a solution to the inverse eigenvalue problem for Hermitian Toeplitz matrices. The approach taken is to first construct a real symmetric negacyclic matrix of order $2n$ and to then relate the negacyclic matrix to a Hermitian Toeplitz matrix of order n having the desired eigenspectrum.**

I. INTRODUCTION

Inverse eigenvalue problems arise often in applied mathematics; see, for example, [1] for an excellent account of the so-called *additive* and *multiplicative* inverse eigenvalue problems. In this work, we are concerned with the inverse eigenvalue problem within the context of statistical signal processing and Hermitian Toeplitz covariance matrices associated with weakly stationary stochastic processes of complex form. Specifically, we present a method for the construction of a Hermitian Toeplitz matrix from an *arbitrary* set of real eigenvalues. The problem treated is a good example of a situation for which little intuition can be gained from the inverse eigenvalue problem in the *real* weakly stationary stochastic process case for which the covariance is real symmetric Toeplitz. This inverse eigenvalue problem is still unresolved for matrices of order greater than four [2], [3], although numerical procedures do exist [4], [5].

The approach we take is to first construct an even order *negacyclic* real symmetric Toeplitz matrix having the desired eigenspectrum, where each eigenvalue, distinct or not, is repeated twice. The negacyclic matrix of order $2n$ so constructed, is then revealed to be the *real matrix* of a Hermitian Toeplitz matrix of order n which has the desired eigenspectrum. We provide a brief description of negacyclic matrices, describe the approach, and present an example.

Manuscript received May 8, 1990; revised January 28, 1992. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) under Grant A0912 and Québec Fonds pour la formation de chercheurs et l'aide à la recherche (FCAR) under Grant EQ 4112.

The authors are with the Information Networks and Systems Laboratory, Department of Electrical Engineering, McGill University, Montréal H3A 2A7 Québec, Canada.

IEEE Log Number 9201059.

II. NEGACYCLIC MATRICES

Real negacyclic matrices are defined in [7, sec. 3.2.1] as circulant matrices having a change in sign for all elements below the main diagonal. A real *symmetric* negacyclic matrix Q of order m may be specified by the first row of elements, $\boldsymbol{q}^T = [q_0 \;\; q_1 \;\; \cdots \;\; q_{m-1}]$, where $q_{m-k} = -q_k$, $k = 0, 1, \cdots, m-1$, and the index $m - k$ is understood to be modulo m. It is seen, therefore, that real symmetric negacyclic matrices are a subclass of real symmetric Toeplitz matrices.

The eigenspectrum, $\{\lambda_i: i = 0, 1, \cdots, m-1\}$, of a symmetric negacylic matrix has elements which are given by the discrete Fourier transform (DFT) of $\hat{\boldsymbol{q}}^T \doteq [q_0 \;\; q_1\omega \;\; \cdots \;\; q_{m-1}\omega^{m-1}]$, where $\omega = e^{j(\pi/m)}$ [6], [7], i.e.,

$$\lambda_i = \sum_{k=0}^{m-1} q_k e^{j(\pi/m)k} e^{j(2\pi/m)ik}, \qquad i = 0, 1, \cdots, m-1. \tag{1}$$

For a symmetric negacyclic matrix of *even* order $m = 2n$, there are n eigenvalues given by

$$\lambda_i = q_0 + 2\sum_{k=1}^{n-1} q_k \cos\frac{\pi}{m}(2i+1)k, \qquad i = 0, 1, \cdots, n-1 \tag{2}$$

which appear with multiplicity two; specifically, $\lambda_i = \lambda_{m-i-1}$, $i = 0, 1, \cdots, n-1$. Of course, the actual multiplicity may be higher, depending on whether the eigenvalues of (1) are distinct or not.

We now turn the situation around by observing that the vector of elements $\boldsymbol{q}$ of a negacyclic real symmetric Toeplitz matrix of order m may be obtained from a given set of n eigenvalues by use of the inverse DFT, viz.,

$$q_k = \frac{1}{n}\sum_{i=0}^{n-1} \lambda_i \cos\frac{\pi}{m}(2i+1)k, \qquad k = 0, 1, \cdots, n-1. \tag{3}$$

The DFT then becomes a simple vehicle for specifying the elements of Q given a set of eigenvalues $\{\lambda_i: i = 0, 1, \cdots, n-1\}$.

III. RELATION TO HERMITIAN TOEPLITZ MATRICES

The purpose of this section is to reveal the relationship that exists between symmetric negacyclic matrices of order m and Hermitian

1053-587X/92$03.00