

Structured Codebook Design in CELP

W.P. LeBlanc and S.A. Mahmoud
 Carleton University, Ottawa, Canada, K1S-5B6
 Phone: 1-613-788-5740/5753
 FAX: 1-613-788-5727
 Email: wilf@sce.carleton.ca

ABSTRACT*

Codebook Excited Linear Prediction [1] is a popular analysis by synthesis technique for quantizing speech at bit rates from 4 to 16 kbps. Codebook design techniques to date have been largely based on either random (often gaussian) codebooks, or on known binary or ternary codes which efficiently map the space of (assumed white) excitation codevectors. It has been shown that by introducing symmetries into the codebook, good complexity reduction can be realized with only marginal decrease in performance. In this paper we consider codebook design algorithms for a wide range of structured codebooks.

INTRODUCTION

This paper considers CELP codebook design algorithms for a variety of structured codebooks. A structured codebook has certain properties which enable it to be searched faster than unstructured codebooks. The design algorithms are applied to CELP coders, but are sufficiently general to be applied to other distortion measures as well.

Consider the CELP analysis structure shown in Figure 1. The long term (quantized) inverse filter (with $2q + 1$ non-zero taps), $B(z)$, for subframe n is given by:

$$B(z) = 1 - \sum_{k=-q}^q b_k z^{-(M+k)} \quad (1)$$

and the short term (quantized) inverse filter (order p), $A(z)$, for subframe n is given by:

$$A(z) = 1 - \sum_{k=1}^p a_k z^{-k} \quad (2)$$

* This work has been sponsored by the Telecommunications Research Institute of Ontario (TRIO).

The perceptual weighting filter, which attempts to obtain a larger signal to noise ratio in inter-formant regions is given by:

$$W(z) = \frac{A_p(z/\beta)}{A_p(z/\gamma)} \quad (3)$$

where γ and β are optimized based on subjective measures, and $A_p(z)$ is the optimum unquantized inverse filter (for subframe n).

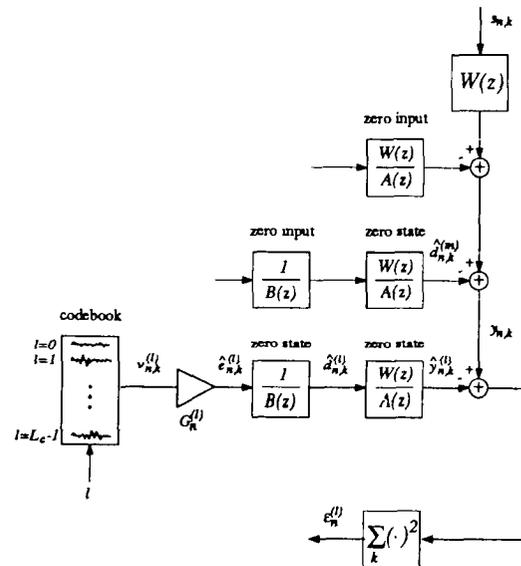


Figure 1: CELP Search Procedure. The codebook dimension, or subframe size is K_c . The index n is over all subframes, and the index k is over all elements of a particular subframe. Thus, $s_{n,k}$ is the k^{th} element of the n^{th} subframe.

Typically, $A_p(z)$ is determined to minimize the open loop residual energy, and $B(z)$ is determined (closed loop) to minimize the noise weighted error before determination of the codebook excitation (the

energy in $y_{n,k}$). The determination of these parameters and complexity reduction techniques based on the structured codebooks is beyond the scope of this paper. The interested reader is referred to [2, 3, 4, 5, 6]. Overlap is often used to reduce block coding edge effects. That is, components of the excitation vector near the end of a subframe have little effect on the current subframe, but may adversely affect future subframes. Overlap considers the influence of these elements by letting the filters ring for K_o samples after the last sample in the excitation vector.

The weighted mean squared error for a particular codebook index l over a subframe (at subframe index n) of dimension K_c with overlap K_o is given by:

$$\begin{aligned} \epsilon_n^{(l)} &= \left| \mathbf{y}_n - \hat{\mathbf{y}}_n^{(l)} \right|^2 \\ &= \mathbf{y}_n^T \mathbf{y}_n - 2G_n^{(l)} \mathbf{y}_n^T \mathbf{H}_n \mathbf{v}^{(l)} \\ &\quad + G_n^{(l)2} \mathbf{v}^{(l)T} \mathbf{H}_n^T \mathbf{H}_n \mathbf{v}^{(l)} \end{aligned} \quad (4)$$

where the $K_c + K_o$ by K_c dimensional lower triangular Toeplitz matrix \mathbf{H}_n represents the zero state filtering operation (of $W(z)/A(z)$). The l^{th} excitation (column) vector $\mathbf{v}^{(l)}$ is of dimension K_c , and the (column) vectors \mathbf{y}_n and $\hat{\mathbf{y}}_n^{(l)}$ ($= G_n^{(l)} \mathbf{H}_n \mathbf{v}^{(l)}$) are of dimension $K (= K_c + K_o)$.

The codebook design algorithms are all based on the Generalized Lloyd Algorithm (GLA) [7, 8, 9] and require a sufficiently rich training sequence to design the codebook. Due to the long memory in $1/B(z)$, the algorithm is not guaranteed to converge to a local minimum. That is, the set of training vectors $\mathcal{T} = \{\mathbf{y}_n\}$ changes from one iteration to the next. The problem arises because (for simplicity) we assume the training vectors do not depend on the codevectors. Due to the long and short term predictor memory this is not the case. In practice, convergence is similar to the GLA, although the average weighted mean squared error has been observed to increase (slightly) after some iterations.

The optimum codebook is defined as that which minimizes Equation 4 over the whole training sequence. We minimize:

$$\begin{aligned} \bar{\epsilon} &= \sum_{n=0}^{N-1} \epsilon_n^{(l_n)} \\ &= \sum_{n=0}^{N-1} \left(\mathbf{y}_n^T \mathbf{y}_n - 2G_n \mathbf{y}_n^T \mathbf{H}_n \mathbf{v}^{(l_n)} \right. \\ &\quad \left. + G_n^2 \mathbf{v}^{(l_n)T} \mathbf{H}_n^T \mathbf{H}_n \mathbf{v}^{(l_n)} \right) \end{aligned} \quad (5)$$

The index n is over all training vectors (\mathbf{y}_n), l_n is the optimum codebook index for training vector

(or subframe) n , $\mathbf{v}^{(l_n)}$ is the optimum codevector (for subframe n) and G_n is the optimum gain for codevector $\mathbf{v}^{(l_n)}$ ($G_n = G_n^{(l_n)}$). The codebook design techniques are all based on minimization of Equation 5. All design techniques assume training vector \mathbf{y}_n is not a function of the current, or past codevectors.

In Section 2 we consider general codebook design. The codebook is given by L_c distinct K_c dimensional codevectors. This section also considers codebooks in which the codevectors have many zero elements.

GENERAL CODEBOOK DESIGN

We now discuss techniques whereby near optimal codebooks may be design for general, or sparse codebooks. The technique is based on a vector quantizer design algorithm using the noise weighted mean squared error distortion measure. Due to the influence of previous codevectors on future codevectors (via the long term predictor memory), only suboptimal codebooks may be designed, (the error is not guaranteed to decrease continually to a local optimum). In practice, the average distortion usually decreases until a *local optimum* is found, then oscillates slowly in the vicinity of that local optimum.

Unstructured Codebooks

The goal is to minimize Equation 5 over all possible codebooks of size L_c and dimension K_c . Given a training sequence of N speech vectors $\mathcal{S} = \{\mathbf{s}_n\}$, and an initial codebook $\mathcal{C}^{(0)} = \{\mathbf{v}^{(l)}\}$, we analyze the vectors using the CELP structure to obtain the training set $\mathcal{T}^{(0)} = \{\mathbf{y}_n\}$. Essentially, we use the initial codebook to partition the training sequence ($\mathcal{T}^{(0)}$) into L_c cells, or regions $\mathcal{R}^{(j)}$ according to the nearest neighbour search, and compute new centroids (or codevectors) for the regions. Cell j is comprised of those subframes which have $\mathbf{v}^{(l_n)} = \mathbf{v}^{(j)}$ (the optimum codebook index at time n is j). Equation 5 can then be split up into L_c terms, one term for each particular cell:

$$\begin{aligned} \bar{\epsilon} &= \sum_{n \in \mathcal{R}^{(0)}} \left| \mathbf{y}_n - G_n \mathbf{H}_n \mathbf{v}^{(0)} \right|^2 + \\ &\quad \sum_{n \in \mathcal{R}^{(1)}} \left| \mathbf{y}_n - G_n \mathbf{H}_n \mathbf{v}^{(1)} \right|^2 + \dots + \\ &\quad \sum_{n \in \mathcal{R}^{(L_c-1)}} \left| \mathbf{y}_n - G_n \mathbf{H}_n \mathbf{v}^{(L-1)} \right|^2 \end{aligned} \quad (6)$$

where the summation indicates summation over the region in which all codevectors are identical. Minimization of Equation 6 is equivalent to minimizing each term, since a particular codevector only influences the summation in its region. Furthermore, in

each region (j), $v^{(l_n)}$ does not depend on n (since $j = l_n$). Thus we minimize (with respect to $v^{(j)}$):

$$\epsilon^{(j)} = \sum_{n \in \mathcal{R}^{(j)}} \left(\mathbf{y}_n^T \mathbf{y}_n - 2G_n \mathbf{y}_n^T \mathbf{H}_n \mathbf{v}^{(j)} + G_n^2 \mathbf{v}^{(j)T} \mathbf{H}_n^T \mathbf{H}_n \mathbf{v}^{(j)} \right) \quad (7)$$

for each region j , $0 \leq j < L_c$. Since $v^{(l_n)} = v^{(j)}$ is a constant for each region and does not depend on the index n , we may write:

$$\begin{aligned} \epsilon^{(j)} &= \sum_{n \in \mathcal{R}^{(j)}} \mathbf{y}_n^T \mathbf{y}_n - 2 \left(\sum_{n \in \mathcal{R}^{(j)}} G_n \mathbf{y}_n^T \mathbf{H}_n \right) \mathbf{v}^{(j)} \\ &\quad + \mathbf{v}^{(j)T} \left(\sum_{n \in \mathcal{R}^{(j)}} G_n^2 \mathbf{H}_n^T \mathbf{H}_n \right) \mathbf{v}^{(j)} \\ &= \bar{\sigma}_y^{(j)2} - 2\bar{\mathbf{c}}^{(j)T} \mathbf{v}^{(j)} + \mathbf{v}^{(j)T} \bar{\mathbf{R}}^{(j)} \mathbf{v}^{(j)} \end{aligned} \quad (8)$$

where

$$\bar{\sigma}_y^{(j)2} = \sum_{n \in \mathcal{R}^{(j)}} \mathbf{y}_n^T \mathbf{y}_n \quad (9)$$

$$\bar{\mathbf{c}}^{(j)} = \sum_{n \in \mathcal{R}^{(j)}} G_n \mathbf{H}_n^T \mathbf{y}_n \quad (10)$$

and:

$$\bar{\mathbf{R}}^{(j)} = \sum_{n \in \mathcal{R}^{(j)}} G_n^2 \mathbf{H}_n^T \mathbf{H}_n \quad (11)$$

It can easily be shown (differentiate with respect to $\mathbf{v}^{(j)}$), that to minimize equation 8 we choose:

$$\mathbf{v}^{(j)} = \left(\bar{\mathbf{R}}^{(j)} \right)^{-1} \bar{\mathbf{c}}^{(j)} \quad (12)$$

which can be efficiently accomplished by using Choleski decomposition. This is performed over all j , $0 \leq j < L_c$.

We will now have a new codebook ($\mathcal{C}^{(1)}$), which can be used in the CELP analysis structure to obtain the training set $\mathcal{T}^{(1)}$. Unlike typical VQ design techniques, the training set $\mathcal{T}^{(1)}$ will not be the same as $\mathcal{T}^{(0)}$. The above design algorithm is just a simple extension of the GLA for a CELP type distortion based on the above assumptions.

Sparse Codebook Design

To design sparse codebooks, we essentially want to minimize Equation 8 (for each j), given the constraint that there are a large number of zero values in the codevectors. We use the multipulse sequential approach (for complexity reasons), and first compute the optimum pulse location and gain (assuming one non-zero value in the codevector) to minimize Equation 8.

We then iteratively add another pulse location, and so on, until we have the desired number of non-zero pulses in the codevector. After each iteration, the pulse amplitudes are re-optimized.

To minimize:

$$\epsilon^{(j)} = \bar{\sigma}_y^{(j)2} - 2\bar{\mathbf{c}}^{(j)T} \mathbf{v}^{(j)} + \mathbf{v}^{(j)T} \bar{\mathbf{R}}^{(j)} \mathbf{v}^{(j)} \quad (13)$$

for the first pulse position (k_0) and amplitude ($v_{k_0}^{(j)}$) we minimize:

$$\epsilon^{(j,1)} = \bar{\sigma}_y^{(j)2} - 2\bar{c}_{k_0}^{(j)} v_{k_0}^{(j)} + v_{k_0}^{(j)} \bar{R}_{k_0 k_0}^{(j)} v_{k_0}^{(j)} \quad (14)$$

which has solution (for a particular position k_0):

$$v_{k_0}^{(j)} = \frac{\bar{c}_{k_0}^{(j)}}{\bar{R}_{k_0 k_0}^{(j)}} \quad (15)$$

The first position is computed by trying all locations, and choosing that which minimizes Equation 14.

Assuming the first pulse location is fixed, the second location is chosen to minimize:

$$\begin{aligned} \epsilon^{(j,2)} &= \bar{\sigma}_y^{(j)2} - 2\bar{c}_{k_0}^{(j)} v_{k_0}^{(j)} - 2\bar{c}_{k_1}^{(j)} v_{k_1}^{(j)} + v_{k_0}^{(j)} \bar{R}_{k_0 k_0}^{(j)} v_{k_0}^{(j)} \\ &\quad + v_{k_1}^{(j)} \bar{R}_{k_1 k_1}^{(j)} v_{k_1}^{(j)} + 2v_{k_1}^{(j)} \bar{R}_{k_1 k_0}^{(j)} v_{k_0}^{(j)} \end{aligned} \quad (16)$$

If $v_{k_0}^{(j)}$ is not to be modified (as part of the search, for complexity reasons), then:

$$v_{k_1}^{(j)} = \frac{\bar{c}_{k_1}^{(j)} - v_{k_0}^{(j)} \bar{R}_{k_1 k_0}^{(j)}}{\bar{R}_{k_1 k_1}^{(j)}} \quad (17)$$

and:

$$\epsilon^{(j,2)} = \bar{\sigma}_y^{(j)2} - \frac{\left(\bar{c}_{k_1}^{(j)} - v_{k_0}^{(j)} \bar{R}_{k_1 k_0}^{(j)} \right)^2}{\bar{R}_{k_1 k_1}^{(j)}} \quad (18)$$

The mean squared error is minimized by maximizing the square of the second term. At the end of the search for the second pulse position, the amplitudes of the first and second pulse positions can be optimized by minimizing Equation 16 with respect to the unknown amplitudes $v_{k_0}^{(j)}$ and $v_{k_1}^{(j)}$.

In general, the n^{th} pulse position is given by computing the minimum over all pulse locations (k_n) of:

$$\epsilon^{(j,n)} = \bar{\sigma}_y^{(j)2} - \frac{\left(\bar{c}_{k_n}^{(j)} - \sum_{i=0}^{n-1} v_{k_i}^{(j)} \bar{R}_{k_n k_i}^{(j)} \right)^2}{\bar{R}_{k_n k_n}^{(j)}} \quad (19)$$

and the pulse amplitudes are optimized by solving (by Choleski Decomposition):

$$\tilde{\mathbf{v}}^{(j)} = \left(\tilde{\mathbf{R}}^{(j)} \right)^{-1} \tilde{\mathbf{c}}^{(j)} \quad (20)$$

where:

$$\tilde{\mathbf{v}}^{(j)} = \begin{bmatrix} v_{k_0}^{(j)} \\ v_{k_1}^{(j)} \\ \vdots \\ v_{k_n}^{(j)} \end{bmatrix}, \quad \tilde{\mathbf{c}}^{(j)} = \begin{bmatrix} \tilde{c}_{k_0}^{(j)} \\ \tilde{c}_{k_1}^{(j)} \\ \vdots \\ \tilde{c}_{k_n}^{(j)} \end{bmatrix} \quad (21)$$

and:

$$\tilde{\mathbf{R}}^{(j)} = \begin{bmatrix} \bar{R}_{k_0 k_0}^{(j)} & \bar{R}_{k_0 k_1}^{(j)} & \dots & \bar{R}_{k_0 k_n}^{(j)} \\ \bar{R}_{k_1 k_0}^{(j)} & \bar{R}_{k_1 k_1}^{(j)} & \dots & \bar{R}_{k_1 k_n}^{(j)} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{R}_{k_n k_0}^{(j)} & \bar{R}_{k_n k_1}^{(j)} & \dots & \bar{R}_{k_n k_n}^{(j)} \end{bmatrix} \quad (22)$$

The sequential multipulse search procedure is inherently suboptimal since it does not try all combinations of pulse positions. However, a full search technique is prohibitively complex. Rather than keeping a single pulse position from stage to stage however, it is clearly better to keep M_l survivors after each stage (for the sequential optimization described above, $M_l = 1$).

SHIFT SYMMETRIC CODEBOOKS

A shift symmetric codebook is defined as a codebook in which a single codevector has all but t elements in common with the next codevector. The l^{th} codevector can be written $\mathbf{v}^{(l)} = \mathbf{S}^{(l)}\mathbf{C}$ where \mathbf{C} is a $K_c + t(L_c - 1)$ dimensional column vector (the codebook) and $\mathbf{S}^{(l)}$ is a K_c by $K_c + t(L_c - 1)$ dimensional *shifting matrix* with ones on the tl^{th} (upper) diagonal, and zeros elsewhere:

$$\mathbf{S}^{(l)} = [\mathbf{0}_{K_c, tl} | \mathbf{I}_{K_c, K_c} | \mathbf{0}_{K_c, L_c - tl}] \quad (23)$$

where $\mathbf{0}_{K_c, n}$ is a K_c by n matrix of zeros, and \mathbf{I}_{K_c, K_c} is the K_c by K_c identity matrix. With $t = K_c$ we obtain the general codebook discussed above.

Shift symmetric codebooks present a problem since elements from a single codevector are included in possibly many other codevectors. Thus, the design algorithm must reflect this property. A modification to the Vector Quantization (VQ) design algorithm was utilized to account for the shift symmetric codebooks. We have:

$$\bar{\epsilon} = \sum_{n=0}^{N-1} \left(\mathbf{y}_n - G_n \mathbf{H}_n \mathbf{v}^{(l_n)} \right)^2 \quad (24)$$

Again we assume we have an initial codebook, but rather than partitioning the codebook into L_c cells or regions using the nearest neighbour, minimum distortion search criteria, we simply substitute $\mathbf{v}^{(l)} = \mathbf{S}^{(l)}\mathbf{C}$ into Equation 24 which yields:

$$\begin{aligned} \bar{\epsilon} &= \sum_{n=0}^{N-1} \left(\mathbf{y}_n - G_n \mathbf{H}_n \mathbf{S}^{(l_n)} \mathbf{C} \right)^2 \\ &= \bar{\sigma}_y^2 - 2\bar{\mathbf{c}}^T \mathbf{C} + \mathbf{C}^T \bar{\mathbf{R}} \mathbf{C} \end{aligned} \quad (25)$$

where:

$$\bar{\sigma}_y^2 = \sum_{n=0}^{N-1} \mathbf{y}_n^T \mathbf{y}_n \quad (26)$$

$$\bar{\mathbf{c}} = \sum_{n=0}^{N-1} G_n \mathbf{S}^{(l_n)T} \mathbf{H}_n^T \mathbf{y}_n \quad (27)$$

(a $K_c + t(L_c - 1)$ dimensional column vector) and:

$$\bar{\mathbf{R}} = \sum_{n=0}^{N-1} G_n^2 \mathbf{S}^{(l_n)T} \mathbf{H}_n^T \mathbf{H}_n \mathbf{S}^{(l_n)} \quad (28)$$

(a square $K_c + t(L_c - 1)$ dimensional band matrix).

The codebook is thus given by $\mathbf{C} = \bar{\mathbf{R}}^{-1} \bar{\mathbf{c}}$ which, again, can be efficiently computed using Choleski Decomposition. Further storage and computational savings can be realized by using the fact that $\bar{\mathbf{R}}$ is a band matrix. Computation of Equation 28 and 27 can be greatly simplified by exploiting the structure in the shifting matrix.

Sparse shift symmetric codebooks can be designed by applying a multipulse procedure to Equation 25, as was done with general sparse codebooks.

VSELP CODEBOOK DESIGN

Let $L_c = 2^M$, where M is the number of bits in the codebook index. The VSELP excitation can be given by $\mathbf{v}^{(l)} = \tilde{\mathbf{C}} \mathbf{b}^{(l)}$ where $\tilde{\mathbf{C}}$ is the VSELP codebook (a K_c by M dimensional matrix), and $\mathbf{b}^{(l)}$ (an M dimensional column vector with elements ± 1) is the l^{th} codeword. Alternatively, yet equivalently, the excitation can be written as $\mathbf{v}^{(l)} = \mathbf{B}^{(l)}\mathbf{C}$ where \mathbf{C} is a $K_c M$ dimensional column vector (containing the stacked columns of $\tilde{\mathbf{C}}$) and $\mathbf{B}^{(l)}$ is a K_c by $K_c M$ dimensional Toeplitz matrix, with the first row having elements $b_k^{(l)}$ in positions $B_{0, k K_c}$.

Over the training sequence, we may write:

$$\bar{\epsilon} = \sum_{n=0}^{N-1} \left(\mathbf{y}_n - G_n \mathbf{H}_n \mathbf{v}^{(l_n)} \right)^2 \quad (29)$$

Substituting $\mathbf{v}^{(l)} = \mathbf{B}^{(l)}\mathbf{C}$ into 29 leads to:

$$\begin{aligned}\bar{\epsilon} &= \sum_{n=0}^{N-1} \left(\mathbf{y}_n - G_n \mathbf{H}_n \mathbf{B}^{(l_n)} \mathbf{C} \right)^2 \\ &= \bar{\sigma}_y^2 - 2\bar{\mathbf{c}}^T \mathbf{C} + \mathbf{C}^T \bar{\mathbf{R}} \mathbf{C}\end{aligned}\quad (30)$$

where:

$$\bar{\sigma}_y^{(j)2} = \sum_{n=0}^{N-1} \mathbf{y}_n^T \mathbf{y}_n \quad (31)$$

$$\bar{\mathbf{c}} = \sum_{n=0}^{N-1} G_n \mathbf{B}^{(l_n)T} \mathbf{H}_n^T \mathbf{y}_n \quad (32)$$

(a $K_c M$ dimensional column vector) and:

$$\bar{\mathbf{R}} = \sum_{n=0}^{N-1} G_n^2 \mathbf{B}^{(l_n)T} \mathbf{H}_n^T \mathbf{H}_n \mathbf{B}^{(l_n)} \quad (33)$$

is a $K_c M$ by $K_c M$ dimensional matrix.

The VSELP (stacked) codebook \mathbf{C} is computed by solving (again by Choleski Decomposition):

$$\mathbf{C} = \bar{\mathbf{R}}^{-1} \bar{\mathbf{c}} \quad (34)$$

Computation of Equation 32 and 33 can be greatly simplified by exploiting the structure of $\mathbf{B}^{(l_n)}$.

RESULTS

In this section we present results of computer simulations conducted on a 10 minute speech database and a 30 second speech database. Codebooks were trained on the large database and the performance was computed on both databases. Objective measures of performance included the segmental signal to noise ratio defined by:

$$\text{SEGSNR} = \frac{1}{N} \sum_n 10 \log_{10} \left(\frac{|\mathbf{s}_n|^2}{|\mathbf{s}_n - \hat{\mathbf{s}}_n|^2} \right) \quad (35)$$

where

$$\hat{\mathbf{s}}_n \quad (36)$$

is the synthesized (20 msec) speech vector and the noise weighted signal to noise ratio defined by:

$$\text{NWSNR} = \frac{\sum_n \mathbf{s}^T \mathbf{s}}{\bar{\epsilon}} \quad (37)$$

In our examples, the CELP coder used sub-frame dimensions of 40 samples, 2 samples of overlap (which was determined to be near optimal), and

frame sizes of 160 samples. The inverse filter ($A(z)$) was determined at the frame rate using the autocorrelation method and quantized using interframe vector linear prediction of the line spectrum pairs followed by scalar quantization of the error [10]. The long term predictor was optimized closed loop to minimize the closed loop weighted mean squared error. The pitch period was constrained to be in the range from 21 to 148 samples. The general codebooks used the autocorrelation method discussed in [2] (which does contain certain approximations). Our experiments with shift symmetric codebooks considered $t = 1$ only, (and no approximations were used). The design of the sparse codebooks used the tree searched multipulse search procedure outlined above, with $M_l = 128$. The sparse shift symmetric codebooks had more than 90% zero samples (52 non-zero samples in a 512 level codebook).

Table 1 displays the performance of random gaussian codebooks for various codebook sizes (L_c).

Codebook Size (bits)	NWSNR (SEGSNR)
7	14.39 (17.13) dB
8	14.84 (17.75) dB
9	15.24 (18.21) dB
10	15.59 (18.66) dB
11	16.10 (19.26) dB

Table 1: Performance of random gaussian codebooks of various sizes (30 second database). The values are accurate (with 95% confidence) to within 0.1 dB.

By comparison, a 9 bit random gaussian shift symmetric codebook obtained a noise weighted SNR of 15.05 dB (SEGSNR=18.11 dB) and a 9 bit random VSELP codebook obtained a NWSNR of 13.92 dB (SEGSNR=16.83 dB). Again the values are accurate (with 95% confidence) to within 0.1 dB.

Trained 9 bit general codebooks, sparse shift symmetric, and VSELP codebooks (using the design techniques discussed above) obtained performance both inside and outside of the training sequence as shown in Tables 2 and 3.

Outside the training sequence the performance (NWSNR) of sparse shift symmetric codebooks is within 0.2 dB of the general codebooks which is within the 95% confidence intervals. Inside the training sequence the performance of the general codebook is approximately 0.7 dB better than the sparse shift symmetric codebooks. Imposing structure limits the performance inside the training sequence but has little effect outside the training sequence in this

Codebook	NWSNR (SEGSNR)
General	14.67 (19.40) dB
Sparse Shift Symmetric	13.95 (18.84) dB
VSELP	13.43 (18.54) dB

Table 2: Performance of trained general, sparse shift symmetric, and VSELP codebooks inside the training sequence (10 minute database). The values are accurate (with 95% confidence) to within 0.1 dB.

Codebook	NWSNR (SEGSNR)
General	15.75 (18.79) dB
Sparse Shift Symmetric	15.60 (18.48) dB
VSELP	14.72 (17.84) dB

Table 3: Performance of trained general, sparse shift symmetric, and VSELP codebooks outside the training sequence (30 second database). The values are accurate (with 95% confidence) to within 0.1 dB.

instance. The VSELP codebooks appear to have too much structure and performance suffers by more than 0.8 dB both inside and outside the training sequence. However, the performance of VSELP improves by more than 0.8 dB after codebook design (outside the training sequence).

The general and sparse shift symmetric trained 9 bit codebooks have objective performance virtually equivalent to the untrained 10 bit random codebooks. Thus, for equivalent objective performance half the number of levels in the codebook are required, resulting in a lower data rate and a lower complexity.

CONCLUSIONS

This paper considered the CELP codebook design problem for a variety of structured codebooks. It was determined that a savings of one bit per vector could be realized with virtually no decrease in the objective measures while decreasing complexity by a factor of two. For fixed codebook sizes, improvements of more than 0.5 dB were observed with no increase in computational complexity. It was observed that Vector Sum Excited Linear Prediction had too much structure, and performance was noticeably inferior to the general or sparse shift symmetric codebooks.

The structured codebook design techniques are relatively simple, and only require Choleski Decomposition or a relatively straightforward multipulse algorithm. The design algorithms were applied to CELP coders, but are sufficiently general to be applied to other distortion measures as well.

REFERENCES

- [1] M. Schroeder and B. Atal, "Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 1985.
- [2] I. Trancoso and B. Atal, "Efficient Procedures for Finding the Optimum Innovation in Stochastic Coders," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1986.
- [3] G. Davidson and A. Gersho, "Complexity Reduction Methods for Vector Excitation Coding," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1986.
- [4] G. Davidson, M. Yong, and A. Gersho, "Real-Time Vector Excitation Coding of Speech at 4800 bps," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1987.
- [5] W. Kleijn, D. Krasinski, and R. Ketchum, "Improved Speech Quality and Efficient Vector Quantization in SELP," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 155-158, April 1988.
- [6] I. Gerson and M. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP)," *IEEE Speech Coding Workshop*, 1989.
- [7] S. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 129-137, March 1982.
- [8] Y. Linde, A. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84-95, January 1980.
- [9] J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," *Proceedings of the IEEE*, vol. 73, pp. 1551-1588, November 1985.
- [10] M. Yong, G. Davidson, and A. Gersho, "Encoding of LPC Spectral Parameters Using Switched-Adaptive Interframe Vector Prediction," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 402-405, April 1988.