Vector Predictive Coding of Speech at 16 kbits/s

VLADIMIR CUPERMAN, MEMBER, IEEE, AND ALLEN GERSHO, FELLOW, IEEE

***Abstract*—Vector quantization, in its simplest form, may be regarded as a generalization of PCM (independent quantization of each sample of a waveform) to what might be called "vector PCM," where a block of consecutive samples, a vector, is simultaneously quantized as one unit. In theory, a performance arbitrarily close to the ultimate rate-distortion limit is achievable with waveform vector quantization if the dimension of the vector, k, is large enough.**

The main obstacle in effectively using vector quantization is complexity. A vector quantizer of dimension k operating at a rate of r bits/sample requires a number of computations on the order of $k2^{kr}$ and a memory of the same order. However, a low-dimensional vector quantizer (dimensions 4–8) achieves a remarkable improvement over scalar quantization (PCM). Consequently, using the vector quantizer as a building block and imbedding it with other waveform data compression techniques may lead to the development of a new and powerful class of waveform coding systems.

This paper proposes and analyzes a waveform coding system, adaptive vector predictive coding (AVPC), in which a low-dimensionality vector quantizer is used in an adaptive predictive coding scheme. In the encoding process, a locally generated prediction of the current input vector is subtracted from the current vector, and the resulting error vector is coded by a vector quantizer. Each frame consisting of many vectors is classified into one of m statistical types. This classification determines which one of m fixed predictors and of m vector quantizers will be used for encoding the current frame.

I. Introduction

THERE is currently a great need for a low-complexity speech coder at the rate of 16 kbits/s which achieves essentially "toll" quality, roughly equivalent to that of standard 64 kbit/s log PCM. Adaptive DPCM schemes are able to achieve this quality with low complexity for the new 32 kbit/s CCITT standard. At 16 kbits/s, the quality of ADPCM or adaptive delta modulation schemes is inadequate. More powerful methods such as subband coding or transform coding not only require a much higher level of complexity, but are also speech-specific, so that they are inadequate for voiceband data signals. In the last few years an important new approach to data compression, vector quantization, has emerged, which may provide the needed breakthrough for 16 kbit/s coding. For a comprehensive review of VQ, see Gray [26].

Vector quantization (VQ) has already made a significant impact on very low rate speech coding based on vocoder techniques [1]. Recently, vector quantization has been applied to waveform coding of speech [2]. The initial results show that direct use of VQ offers a significant improvement over scalar quantization, i.e., PCM, but it is not by itself competitive with other highly developed waveform coders.

In this paper, we describe a new approach to waveform coding by using VQ as a building block in a coding system rather than as a complete coder in itself. Specifically, the underlying idea of predictive coding schemes, such as DPCM, is generalized to vector signal processing so that the enhanced efficiency of VQ over scalar quantization can be exploited. In order to utilize the capability of predictors and vector quantizers to be "tuned" to specific waveform statistics, we introduce an adaptive scheme where a classifier is used to identify particular categories of local speech statistics. This scheme also has the potential capability of handling voiceband data signals. The basic ideas of vector predictive coding were first reported in [24]. The approach was also proposed in [27] for use in quantized control systems.

For waveform coding, VQ may be regarded as a generalization of PCM (independent quantization of each sample of a waveform) to what might be called "vector PCM," where a block of consecutive samples, regarded as a vector, is quantized as one unit. Rate-distortion theory guarantees a performance arbitrarily close to optimal for waveform vector quantizers if the dimension of the vector, k, is sufficiently large. The main obstacle in using the capability is implementation complexity. A vector quantizer of dimension k operating at a rate of r bits/component requires a number of computations of the order of $k2^{kr}$ and a memory of the same order.

The first approach we have considered to circumvent the complexity problem is based on suboptimal vector quantization.Various suboptimal vector quantization techniques, such as tree-structured codebooks or multistage quantizers, achieve an important reduction in computational complexity [1], [3], [4]. The resulting quantizers compare favorably with scalar quantization, but the acheived overall performance is not competitive with sophisticated scalar coding systems using compression techniques such as adaptive differential coding methods, transform coding, and subband coding.

A different approach for achieving a viable waveform coding system is to exploit the performance of a low-dimensionality vector quantizer by combining it with other waveform compression techniques. In this approach, the vector quantizer becomes a building block in the waveform coding system. Consequently, the entire system must be designed as a vector processing system, as a result of the presence of the vector quantizer. Thus, just as basic PCM is used as a building block in ADPCM and other more sophisticated coders, vector PCM can be used as a building block in a vector-based predictive coding system. This approach is based on the idea that the basic vector dimension is limited by complexity considerations to values that are much lower than the span of samples that contain significant correlation; hence, to further exploit the correlation between successive low-dimensional blocks (vectors) of speech samples, vector prediction is used.

This paper presents and analyzes a waveform coding system in which a low-dimensionality vector quantizer is used in an adaptive predictive coding scheme. In the encoding process, a locally generated prediction of the current input vector is subtracted from the current vector, and the error vector is coded by a vector quantizer. The vector quantizer observes a vector of error samples and identifies the closest matching error pattern from a codebook of stored patterns. A binary word, defining the address of that pattern in the codebook, is then transmitted to the receiver. The system is made adaptive by classifying each frame of speech, consisting of many

Paper approved by the Editor for Signal Processing and Communication Electronics of the IEEE Communications Society for publication after presentation at IEEE GLOBECOM, Miami, FL, December 1982. Manuscript received August 26, 1983; revised October 12, 1984. This work was supported in part by the Air Force Office of Scientific Research under Grant 82-0008.

V. Cuperman is with Calltalk, Ltd., Tel-Aviv 67 214, Israel.

A. Gersho is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106.

0090-6778/85/0700-0685$01.00 © 1985 IEEE

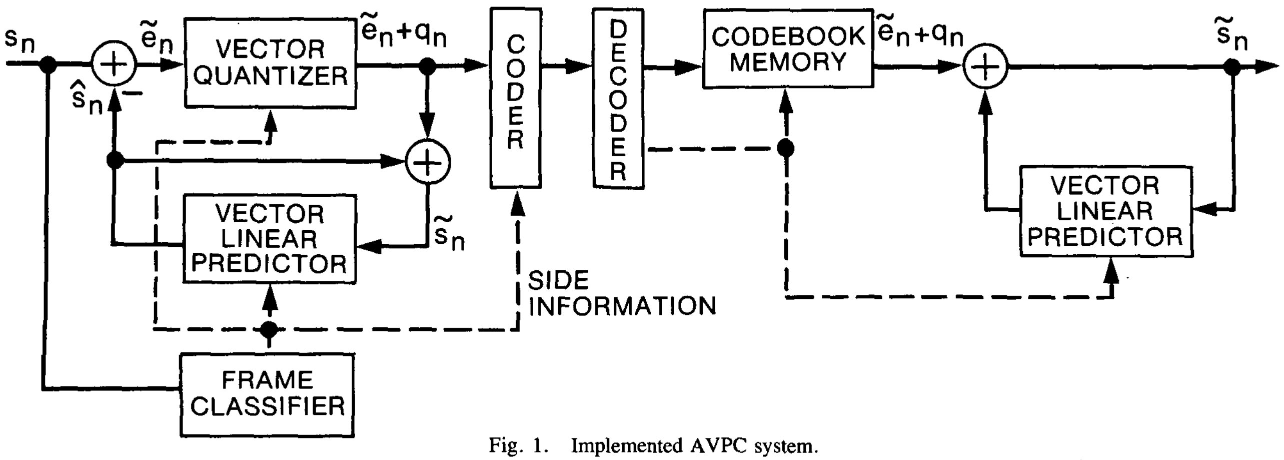


Fig. 1. Implemented AVPC system.

vectors, into one of m statistical types. This classification determines which one of m fixed predictors and of m vector quantizers will be used for coding the current frame. The class of each frame is transmitted as side information. The overall system is called adaptive vector predictive coding (AVPC).

The general configuration proposed for vector predictive coding is shown in Fig. 1. A vector linear predictor (VLP) locally generates a prediction $\hat{s}_n$ of the current vector s_n. The prediction is subtracted from the current input vector giving the prediction error vector $\tilde{e}_n$:

$$\tilde{e}_n = s_n - \hat{s}_n.$$

The prediction error vector is encoded by a vector quantizer, which selects from its codebook a codevector v_n to represent the error pattern. The difference between v_n and $\tilde{e}_n$ is the quantization error vector q_n. The feedback configuration is the same as that used in scalar DPCM to assure that the VLP in the transmitter and the identical VLP in the receiver have the same input $\tilde{s}_n$, which is also the reconstructed vector at the system output.

The coder output shown in Fig. 1 is actually the binary index of the codebook entry chosen to represent the prediction error $\tilde{e}_n$. Unless otherwise specified, the encoding process is assumed to perform a full search through the codebook for finding the closest match to $\tilde{e}_n$ under the chosen fidelity criterion.

The receiver has an "inverse" vector quantizer which simply uses the codebook as a table lookup, addressed by the received index, to retrieve the error pattern (codevector) v_n. The error pattern has the value $\tilde{e}_n + q_n$, which differs from the prediction error $\tilde{e}_n$ by the quantization error q_n. The receiver uses an identical VLP to reconstruct the output vector $\tilde{s}_n$ using the error pattern v_n.

Let ϵ_n be the overall error vector for the coding system, defined by

$$\epsilon_n = s_n - \tilde{s}_n.$$

Using the input-output relations for the two summation points in the transmitter structure, it is easy to see that

$$\epsilon_n = -q_n. \tag{1}$$

This relation proves that the variance of the overall system error, σ_ϵ^2, is equal to the variance of σ_q^2 of the vector quantization error q_n, which is a fundamental property of the predictive coding configuration. This result is a direct generalization of the same well-known feature of scalar DPCM. (See [5] for a comprehensive theoretical review of DPCM with emphasis on adaptive prediction, and [6] for a broad tutorial description of quantization and DPCM.) Note that in the vector case, σ_ϵ^2 and σ_q^2 are energies (variances) of the random vectors ϵ_n and q_n, respectively, defined as the expectations of the sum of squares of the components of the respective vectors.

The signal-to-noise ratio of the vector quantizer, $(\mathrm{SNR})_{\mathrm{VQ}}$, is a function of the statistics of the prediction error $\tilde{e}_n$. The determination of these statistics, given the statistics of the input vector s_n, is an analytically intractable problem because the feedback loop contains a nonlinear element (the vector quantizer). Nevertheless, the signal-to-noise ratio $(\mathrm{SNR})_{\mathrm{VQ}}$ is independent of the amplitude scaling of the prediction error vector, $\tilde{e}_n$, i.e., independent of its variance $\sigma_{\tilde{e}}^2$ [7].

On the basis of the above considerations, the variance of the quantization error may be written as the variance of the quantizer input signal divided by the signal-to-noise ratio:

$$\sigma_q^2 = \frac{\sigma_{\tilde{e}}^2}{(\mathrm{SNR})_{\mathrm{VQ}}}.$$

Then, the overall signal-to-noise ratio for the predictive vector coding system is given by

$$\mathrm{SNR} = \frac{\sigma_s^2}{\sigma_\epsilon^2} = (\mathrm{SNR})_{\mathrm{VQ}}\, G$$

where G, the prediction gain of the vector linear predictor in the given configuration, is defined by

$$G = \frac{\sigma_s^2}{\sigma_{\tilde{e}}^2}.$$

Thus, the signal-to-noise ratio for the predictive vector coding system is equal to the signal-to-noise ratio for the vector quantizer multiplied by the prediction gain. It should be noted that the prediction gain as defined above differs from the prediction gain associated with the unquantized prediction error. The relation between the actual prediction error and the prediction error in the absence of quantization will be discussed in the next section. The difference decreases as the quantization error decreases, i.e., as the number of output vectors (codebook size) in the vector quantizer increases.

II. Design of the Vector Predictive Coding System

The system design first requires the selection of a suitable quantitative measure of performance. Once a tractable measure has been selected, three major components to be designed are the vector predictor, the vector quantizer, and the speech classifier. This section of the paper examines these issues.

A. Vector Distortion Criteria for Coder Design

Vector signal processing conveniently allows more meaningful measures of fidelity to be used in optimizing system performance than does scalar processing. By regarding a vector of speech samples as a short segment of speech, it is possible to quantitatively measure the distance of distortion between the two vectors that to some degree reflects their perceptual dissimilarity.

A useful and general class of vector fidelity criteria can be defined as a weighted mean-square error:

$$e_{\mathrm{WMS}} = E\{(s_n - \hat{s}_n)' W (s_n - \hat{s}_n)\} \tag{2}$$

where s_n is the input vector, $\hat{s}_n$ is the corresponding output vector, and W is a positive definite weighting matrix. If W is the k-dimensional identity matrix I_k, the definition reduces to the sum of the mean-squared errors of the vector components, or in other words, the mean of the squared Euclidean distance between the vectors viewed as points in k-dimensional space.

More generally, the weighting matrix may be selected to reflect perceptually important differences between the vectors. Since speech is only a locally stationary process, the perceptually important features that need to be accurately reproduced must depend on the local or short-term statistics of the speech waveform. Hence, the desirable weighting matrix for the performance measure varies with time, and must be determined by the local (short-term) statistical character of the input signal itself.

Perceptually meaningful distortion criteria have been used in LPC coding by treating an entire frame of a hundred or so samples of speech as a single vector. The Itakura–Saito distortion measure, extensively used in linear predictive coding (LPC), may also be defined in the inner product form with a weighting matrix depending on the input vector s_n (see [8]). For low-dimensional vector coding, we do not have quite so meaningful distortion measures, but we still can do better than the standard mean-square error that is pervasive in studies of scalar coding systems.

For simplicity, in this paper we present only the special case where the weighting matrix is a scalar constant multiplied by the identity matrix. The most relevant local statistical characteristic that we can easily utilize is the short-term energy or power of the speech waveform. Suppose the speech waveform is partitioned into frames (for example, of length 10 ms, or 80 samples). Suppose also that each frame can be considered as a realization of a segment of a stationary process. Different frames may have different statistics, and are viewed as realizations of different random processes. The signal vectors are assumed to have dimensionality lower than the frame length, so that several vectors are contained in each frame. (In particular, $N_v = 10$ vectors per frame corresponds to $k = 8$ dimensions per vector and a frame length of 80 samples.) Then, if $s_n^{(l)}$ is a vector in the frame with the index l, the weighting matrix W_l is defined by

$$W_l = \frac{1}{E\|s_n^{(l)}\|^2} I_k.$$

This expectation is actually empirically estimated by time averaging the vector energies in the lth frame.

This gives a reasonable and simple performance measure for an individual frame of speech, within which we assume local stationarity of the speech waveform. However, we need a single performance measure for assessing the reconstructed quality of a record of speech of much longer duration than a single frame. The general approach, where a measure of performance is assigned to each frame or "segment" and these numbers are somehow combined to obtain an overall numerical indicator of performance, is what might be called a segmental approach. In this section we describe two particular segmental measures for assessing coder performance.

One way to average the frame performance measure over many frames is to use a geometric mean. The geometric mean of the weighted mean-square error measure, taken over K consecutive frames, we call the segmental mean square error, e_{SEG}:

$$e_{\mathrm{SEG}} = \left[\prod_{l=1}^{K} \frac{E\|s_n^{(l)} - \hat{s}_n^{(l)}\|^2}{E\|s_n^{(l)}\|^2} \right]^{\frac{1}{K}} \tag{3}$$

where $\hat{s}_n^{(l)}$ is the output vector corresponding to the input vector $s_n^{(l)}$. The expectation in the numerator can be empirically estimated by time averaging over all error vectors in the lth frame. This performance measure is essentially equivalent to the so-called segmental signal-to-noise ratio, or SEGSNR, previously introduced for evaluating conventional scalar speech coders [9]. This measure was found to give a better correspondence with the subjective quality of speech than the usual MSE measure, where the total mean-square error over the entire record is compared to the total signal energy over the entire record. The SEGSNR is obtained from

$$\mathrm{SEGSNR} = -10 \cdot \log (e_{\mathrm{SEG}})$$

so that an arithmetic average of the frame SNR values in decibels is actually being calculated [9]. In practice, a thresholding operation is also added to limit the maximum or minimum SNR value in each frame [6].

Unfortunately, the use of the SMSE measure for the design of waveform coders leads to intractable optimization problems. An alternate segmental criterion which we call the energy weighted mean square error, e_{EWMS}, is defined by a direct arithmetic (rather than geometric) average of the performance measure for each frame:

$$e_{\mathrm{EWMS}} = \frac{1}{K} \sum_{l=1}^{K} \frac{E\|s_n^{(l)} - \hat{s}_n^{(l)}\|^2}{E\|s_n^{(l)}\|^2}. \tag{4}$$

Since the arithmetic mean is always greater than the geometric mean ($e_{\mathrm{EWMS}} \geqslant e_{\mathrm{SEG}}$), by minimizing e_{EWMS}, an upper bound on e_{SEG} is minimized. The experimental results indicate that an improvement in both the segmental signal-to-noise ratio and the subjective quality of the reconstructed speech is obtained when using this criterion in coder design instead of the standard MSE.

B. Design of the Vector Linear Predictor

Although not so widely known as scalar linear prediction, the basic ideas of optimal linear prediction readily generalize to the case of a vector-valued time series. The optimal linear predictor for a stationary vector input process may be found by solving the Wiener–Hopf equation in matrix form [10], [11]. Recent work has also generalized these results to the degenerate case, where the current vector may be exactly predictable from a linear combination of previous vectors [12]. The prediction coefficients A_i are fixed matrices and are determined from the second-order statistics of the process using a generalization of the Levinson–Durbin algorithm. This is a precomputation that is performed once as part of the system design. In our application, the vector process is generated by "blocking" the original scalar speech process, that is, partitioning the waveform into consecutive blocks of k samples each and defining a vector composed of these k samples, consistently ordered. This particular vector process has some special statistical properties that simplify the computation of predictor coefficient matrices.

Because speech is a "locally stationary" process, better performance can be obtained by using predictor coefficients that are optimized for each frame. The disadvantage of this "frame-adaptive" prediction is that the coefficient matrices A_i must be recomputed in real time for each frame and transmitted to the receiver as side information, greatly increasing the complexity of the coding system.

A different compromise between performance, complexity, and rate is offered by using a small set of fixed predictors and selecting one according to the statistics of the input signal estimated for each speech frame. This method retains the adaptive character, while eliminating the need for computing predictor coefficient matrices "on line" for each frame and transmitting them to the receiver. The side information for this case is the index identifying which predictor from the finite set has been used for the current frame. It is only this index, rather than a full description of the predictor, which must be transmitted for each frame.

Suppose the speech waveform is partitioned into frames, so that each frame contains kN_v samples of N_v vectors, each of k contiguous samples. Each frame will be assigned to one particular statistical class by a classifier which uses some simple statistical characteristics of that frame to make the classification. The objective is to have some degree of statistical similarity for all frames belonging to a particular class. We then need to design a fixed predictor that can be used for all frames belonging to a particular class. Although this predictor will be optimal for the class, it will not be optimal for any specific frame in the class. For this reason, the quantization error will be considered a second-order effect and the design of the optimal predictor for a class will be presented for unquantized speech. Since the actual statistical characters of different frames belonging to the same class are not identical, we require a slight generalization of linear prediction theory to obtain the correct predictor design formulas.

Suppose that a given record of speech is partitioned into frames and all the frames that are identified as belonging to class c are extracted and numbered from 1 to L. Suppose that for each l, the lth frame is a realization of a particular stationary vector process $\boldsymbol{P}_l$. The statistics of the process $\boldsymbol{P}_l$ will be estimated by time averaging within the speech frame with the index l. All the processes $\boldsymbol{P}_l$ for $l = 1, 2, \cdots, L$ belong to one class c, and a vector linear predictor is to be designed which is optimal for this class, by using as data the L observed frames belonging to class c. The following relations, leading to the needed predictor design for class c, will then apply in the same way for each other class.

Let $\{s_i^{(l)}\}$ be a sequence of random vectors chosen from the process $\boldsymbol{P}_l$ and define the covariance matrices:

$$C_{ij}^{(l)} = E\{s_{n-i}^{(l)} s_{n-j}^{(l)\prime}\} \tag{5}$$

where s' denotes the transpose of s. Actually, the matrix $C_{ij}^{(l)}$ depends on $i - j$ only, given the stationarity of $\boldsymbol{P}_l$.

Consider the linear prediction $\hat{s}_n^{(l)}$ of $s_n^{(l)}$ given by

$$\hat{s}_n^{(l)} = \sum_{m=1}^{M} A_m s_{n-m}^{(l)} \tag{6}$$

where A_m are matrices of dimension $k \times k$ and k is the dimension of the vectors $s_i^{(l)}$. This estimate uses the same prediction matrices A_m for all the processes $\boldsymbol{P}_l$, $l = 1, \cdots, L$.

We wish to find optimal predictor matrices A_m, using as fidelity criterion the energy weighted mean square error e_{EWMS} as given by (4), where now the averaging is taken over the L processes $\boldsymbol{P}_l$ belonging to the particular class c. Denote by σ_l^2 the variance of each vector in the process $\boldsymbol{P}_l$:

$$\sigma_l^2 = E\|s_n^{(l)}\|^2 = \mathrm{tr}\,(C_{00}^{(l)}) \tag{7}$$

where tr denotes the trace of the matrix argument.

Then, it can readily be shown that the following generalized orthogonality principle gives the conditions for optimality of the predictor using the e_{EWMS} criterion:

$$\sum_{l=1}^{L} \frac{1}{\sigma_l^2} E\{(s_n^{(l)} - \hat{s}_n^{(l)}) s_{n-m}^{(l)\prime}\} = 0 \tag{8}$$

for $m = 1, 2, \cdots, M$. This form of the orthogonality principle may be used to derive a set of equations in A_j for finding the optimal prediction coefficients:

$$\sum_{j=1}^{M} A_j \tilde{C}_{ji} = \tilde{C}_{0i} \qquad \text{for } i = 1, 2, 3, \cdots, M \tag{9}$$

where the matrices $\tilde{C}_{ji}$ are defined by

$$\tilde{C}_{ji} = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{\sigma_l^2} C_{ji}^{(l)}, \tag{10}$$

i.e., $\tilde{C}_{ji}$ are the weighted averages of the covariance matrices over the processes belonging to a given class.

Equation (9) is a generalized form of the multivariate Wiener–Hopf equation (the generalized error criterion e_{EWMS} was used and the optimization was performed over a class of processes rather than over a given process). This equation will be used to design the vector linear predictors used in adaptive vector predictive coding (AVPC). Note that the equation differs from the usual multivariate Wiener–Hopf equation for a stationary process by replacing the covariance matrices with a weighted average of the covariance matrices for each process $\boldsymbol{P}_l$. The optimum solution under the standard (unweighted) MSE criterion for the nonstationary case may be obtained as a special case by setting $\sigma_l = \sigma =$ const. The relation (10) for the covariance matrices then reduces to

$$\tilde{C}_{ji} = \frac{1}{L} \sum_{l=1}^{L} C_{ji}^{(l)} \tag{11}$$

which is simply an unweighted average of the covariance matrices for the L processes.

To assess the performance achieved by vector linear prediction on typical speech waveforms, a series of experiments was executed. A speech database (see the Appendix) of about 128 000 samples (16 s of speech), containing six sentences spoken by four male talkers and two female talkers, was used for these experiments.

The segmental prediction gain G_{SEG}, used as a criterion for evaluating the predictor performance, is based on the segmental mean-square error and is defined by

$$G_{\mathrm{SEG}} = \left[\prod_{l=1}^{L} \frac{E\|s_n^{(l)}\|^2}{E\|s_n^{(l)} - \hat{s}_n^{(l)}\|^2} \right]^{\frac{1}{L}} \tag{12}$$

where a partition into frames of the speech waveform is assumed, and l is the frame index. For all the following results, a frame length of 7.5 ms was used unless otherwise indicated.

The vector linear prediction configuration used in this experiment is depicted in Fig. 2. The procedure may be regarded as a reversible "compression" by linear prediction: the transmitter compresses the input vector process s_n into the error process $s_n - \hat{s}_n$, and the receiver must reconstruct the waveform from the prediction error sequence. No quantization or other noise is added at this time.

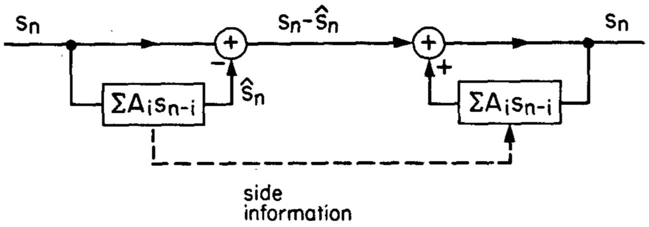


Fig. 2. Vector linear prediction configuration.

The experimental results which will be presented are based on first-order vector prediction ($M = 1$). The reason is that, for the vector dimensions k being used, there is a rapid decrease in the contribution of a prior "data" vector to the prediction gain achieved as the time lag between the data vector and the predicted vector increases. This is due to the exponential decrease of the autocorrelation function versus lag, which is characteristic of speech waveforms. On the other hand, the computational and memory complexity increase linearly with the predictor order.

Fig. 3 compares the segmental prediction gain in decibels versus dimension for two types of adaptive predictors. In the frame-adaptive prediction method, a separate predictor is designed for each frame using the statistical data obtained from that frame. This is a very high-complexity method, corresponding to the processing performed in LPC analysis, and gives us a point of reference for comparison with reduced complexity schemes. The second method uses switched prediction with one of three predictors selected for each frame by a simple three-way frame classification summarized in Table I. (The classifier will be discussed later.) The optimal predictors for each class were designed by separately solving the generalized Wiener–Hopf equations (9) for each class, using the energy-weighted mean-square error criterion. The classifier was used to select those speech frames belonging to each class, so that the weighted average covariance matrices could be computed for each class.

The results presented in Fig. 3 indicate that the predictor with switched coefficients achieves a good compromise between performance and complexity. The three-way "switched" predictors designed in this way will be used in the vector predictive coding system.

C. Design of the Vector Quantizer

A vector quantizer is used to code the prediction error vector generated by the closed-loop vector predictive configuration shown in Fig. 1. Two design approaches were developed for optimizing the quantizer to the statistical character of the error vectors to be coded. Both approaches use a training sequence of speech data and a modified version of the Linde–Buzo–Gray (LBG) codebook design algorithm proposed in [13]. The first approach uses the prediction error vectors generated from the unquantized speech signal as the training sequence for codebook design, and will be called the open-loop design. The second approach uses actual prediction error vectors generated within the predictive coder loop, and will be called the closed-loop design. In this section, some aspects of the iterative codebook design algorithm are presented, followed by a detailed discussion of the two approaches studied.

1) Codebook Design Algorithms: In general, a vector quantizer is fully specified by the codebook of stored patterns or "codevectors" and the distance measure used by the encoding process to find the best match to an input vector. The codebook is designed on the basis of empirical data regarding the statistics of the vectors to be coded. These data consist of a large but finite set of "training" vectors that are chosen to be statistically representative of the signal source. Associated with each codebook is a partition of the training set into "cells" or clusters of training vectors, with each cluster corresponding to a particular codevector.

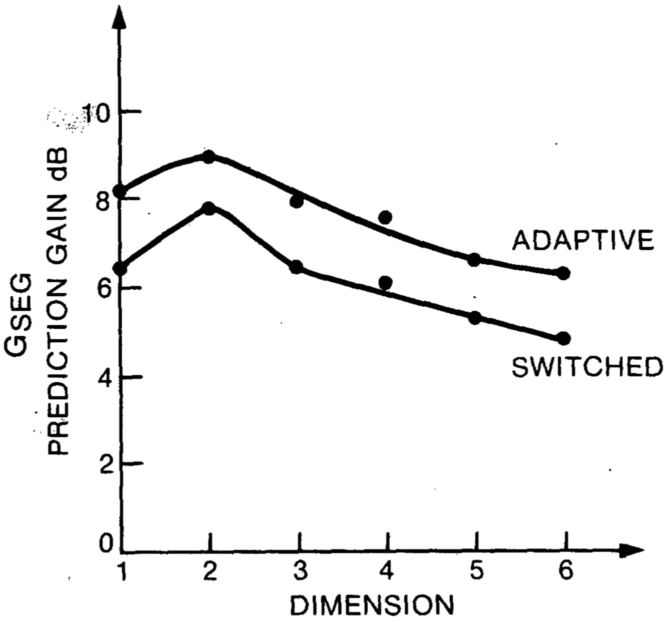


Fig. 3. Vector prediction gain versus vector dimension for frame-adaptive prediction and switched prediction.

TABLE I

Class 1	Class 2	Class 3
$r(1) \geq 0.94$ and $\sigma \geq 0.25\sigma_0$ Highly correlated and high energy	$0.94 > r(1) \geq 0.70$ and $\sigma \geq 0.25\sigma_0$ Moderately correlated and high energy	$r(1) < 0.70$ or $\sigma < 0.25\sigma_0$ Low correlated or low energy

The codebook design begins with the selection of some initial codebook. Then, a simple but computationally demanding iterative algorithm leads to a final codebook design. The goal of the algorithm is to find a codebook whose member codevectors best represent the large set of training vectors used as input to the algorithm. The basic codebook design algorithm consists of iteratively performing the basic operation of generating the clusters for a given set of codevectors, then finding a new set of codevectors for these clusters. This algorithm always terminates in a finite number of steps and achieves a local optimum with respect to the training set for a given distortion measure.

The optimal codevectors in the LBG algorithm for the EWMSE criterion are generalized centroids given by

$$y_j = \left(\sum_{s \in S_j} \frac{1}{\sigma_s^2} \right)^{-1} \sum_{s \in S_j} \frac{s}{\sigma_s^2} \tag{13}$$

where σ_s^2 is the energy of the frame to which the current vector s belongs and S_j is the partition region for which the centroid is computed. This result is a special case of the centroid formula for the error criterion (2) and was first noted in [21]. If the distortion criterion is the MSE, the weighting matrix is the identity matrix and the previous relation becomes

$$y_j = \frac{1}{n_j} \sum_{s \in S_j} s \tag{14}$$

where n_j denotes the number of vectors in the partition S_j.

2) The Empty Cell Problem: One of the problems which frequently occurs when using this algorithm is that a codevector in the final codebook may be assigned to a cluster containing only a few, if any, input training vectors and, hence, may contribute very little to the reduction of overall distortion. The main reason for this "empty cell problem" is that the choice of the initial codebook determines to which of several local optima the algorithm converges. For some choices of the initial codebook, the final codebook, of size N, may contain a subset of N_1 codevectors which happen to be locally optimal for a quantizer of size N_1. The remaining $N - N_1$ codevectors may then be poorly utilized, having in some cases empty or nearly empty cells in spite of the overall codebook being locally optimal. A choice of the initial codebook which would avoid empty cells requires prior knowledge about the final codebook, which is generally unavailable.

One approach that partially alleviates this problem is to begin with a codebook of only two output points (one bit), and to build a codebook of $b + 1$ bits from a codebook of b bits by "splitting" every output point into two new output points. Although this technique gives good experimental results [13], [2], it does not guarantee the absence of empty cells from the resulting codebooks.

An alternate solution to the "empty cell problem" is to use some features of the ISODATA algorithm in pattern recognition [14]. This algorithm allows for splitting and lumping of clusters at every iteration, at the cost of a substantial increase in complexity.

As a compromise solution to the above considerations, a modified LBG type algorithm was used. In this modified version, after the clustering operation, the number of input vectors assigned to every cluster is checked. If a cell is empty, the corresponding codevector is deleted and a new codevector is created by "splitting" the codevector v_* representing the cluster with the highest distortion. The splitting operation retains the highest distortion codevector v_* but replaces the empty cell codevector with a perturbed version, $\tilde{z}_*$, of v_*. The highest distortion cluster is then split into two clusters. The perturbed codevector $\tilde{z}_*$ is obtained by multiplying all the components of the output vector by a constant factor. It can be shown that the result of such an operation can only result in a decrease in the overall distortion, so that the convergence of the modified algorithm is assured. In practice, we have found that convergence is achieved for speech data even when the cell is "nearly empty," containing one or two vectors.

3) Codebook Design for Predictive Coding: The codebook design was based on the training set of speech vectors residing in the speech database described in the Appendix. The classifier was used to select those speech frames belonging to one particular class. The design procedures described in this section were applied separately to obtain separate codebooks for each class.

The covariance matrices were computed by time averaging over each speech frame, using a vector generalization of the "autocorrelation" method known in scalar LPC [15]. Any element in a covariance matrix is equal to the scalar correlation coefficient of a given lag, which is computed by using the method used in scalar LPC analysis. Note that in this approach the covariance matrices are Toeplitz matrices, i.e., C_{ij} depends only on $|j - i|$.

The optimal prediction coefficients were then computed by solving the generalized Wiener-Hopf equations, (9), where the "weighted" covariance matrices $\tilde{C}_{ji}$ are given by (10) under the EWMSE criterion and by the relation (11) under the MSE criterion. These predictor matrices were used in the vector predictive coding systems.

Using the prediction matrices and the same training set of speech vectors, a training set of prediction error vectors for the unquantized input, e_n, was generated by applying the relation

$$e_n = s_n - \hat{s}_n = s_n - \sum_{j=1}^{M} A_j s_{n-j}. \tag{15}$$

In the open-loop approach, the vectors e_n were used as training data to design the codebook by applying the modified LBG iterative algorithm described previously.

The above vector quantizer design algorithm uses the prediction error e_n, generated by the unquantized input signal, rather than the prediction error $\tilde{e}_n$ that arises in the actual encoding process and is generated by the quantized signal $\tilde{s}_n$. To get an insight into this problem, a relation between the two prediction errors is needed.

In Section I, we noted that the overall system error is equal to the quantization error as given by (1). Thus, from (1), we have

$$\tilde{s}_n = s_n + q_n. \tag{16}$$

The actual prediction error $\tilde{e}_n$ is given by

$$\tilde{e}_n = s_n - \sum_{j=1}^{M} A_j \tilde{s}_{n-j}$$

so that, using (16), we get

$$\tilde{e}_n = s_n - \sum_{j=1}^{M} A_j s_{n-j} - \sum_{j=1}^{M} A_j q_{n-j}$$

or

$$\tilde{e}_n = e_n - \sum_{j=1}^{M} A_j q_{n-j}. \tag{17}$$

Hence, the difference between the two prediction errors is equal to the output of the prediction filter excited by the quantization error vectors.

To a first approximation, taking into account the "random" character of the quantization error q_n, for good quality quantization (large number of codewords), the averages computed over the quantities q_{n-j} in the relation (17) may be considered equal to zero. Now, consider that the computation of a centroid is based on the conditional expectation of a data vector, given a region of the quantizer partition. Note that the data vectors for the closed-loop design are the vectors $\tilde{e}_n$ and those for the open-loop design are e_n. Then, the relation (17) suggests that the centroids obtained by a closed-loop design and those computed by an open-loop design are essentially equal and the two design methods should give similar results.

The *closed-loop* design consists of the following main steps.

1) Using the training set of speech vectors, compute the covariance matrices and the predictor matrices.

2) Choose an initial codebook. A uniform codebook in the signal range may be used, an initial codebook may be designed by using the splitting procedure described in [13], or the result of an open-loop design can be used.

3) Keeping the codebook fixed, compute and store the set of prediction error vectors $\tilde{e}_n$, i.e., the relation (17), corresponding to the configuration of Fig. 1.

4) Using the set of vectors obtained in 3) as a training set, obtain a new codebook by a clustering and centroid computation. Replace the initial or the previous codebook with the new codebook.

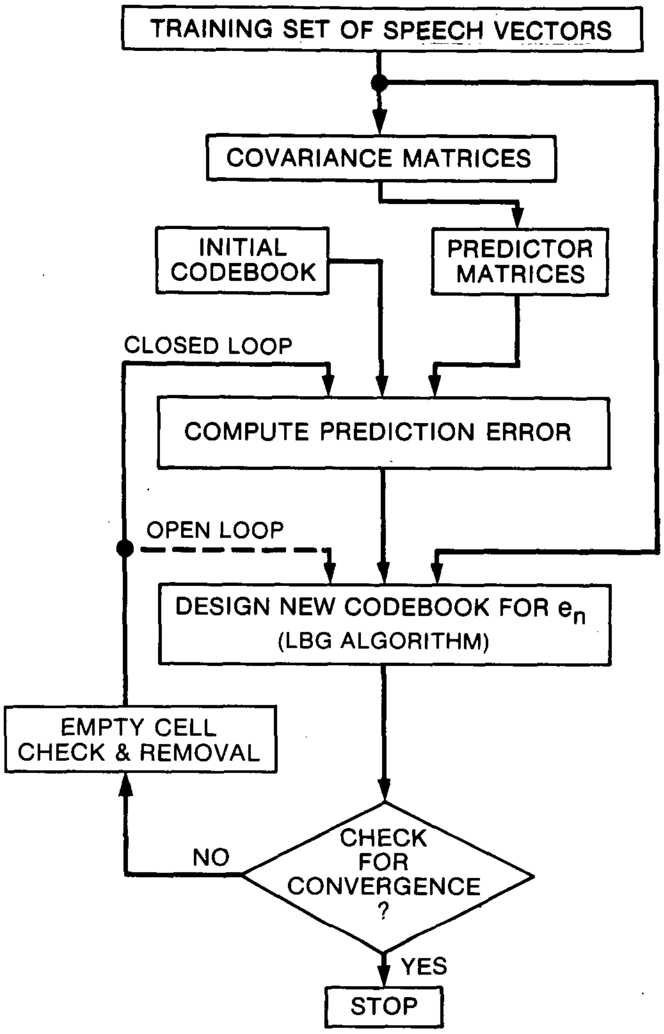


Fig. 4. Flowchart for closed-loop and open-loop codebook design.

5) Iterate steps 3) and 4) until no significant increase in the signal-to-noise ratio is observed.

The complete design algorithm for both approaches (open and closed loop) is depicted in the flowchart of Fig. 4.

The algorithm represented by steps 1)-5) was found experimentally to converge at least as fast as the open-loop algorithm, based on the fixed set of training vectors e_n. Better performance was always achieved by the closed loop design, with a signal-to-noise ratio improvement on the order of 1 or 2 dB. As expected, the closed-loop improvement decreased as the bit rate was increased. The experiments were executed at rates ranging from 1 to 2 bits/sample.

4) The Speech Classifier: An efficient coding system for nonstationary waveforms such as speech needs adaptive procedures, in order to match the coder parameters to the local statistical character of the waveform. Frame-adaptive vector linear prediction achieves good results in vector linear prediction of speech. The disadvantages are high complexity and the considerable overhead rate needed to transmit new prediction matrices for each frame. For example, if each matrix element requires an average of only 3 bits for transmission, 1.25 bits/sample are required only for transmitting prediction matrices for an AVPC system in dimension 5 with a frame length of 60 samples. As a result of complexity and rate considerations, the switched adaptive prediction is preferable to the AVPC system, in spite of a loss of more than 2 dB in performance compared to frame-adaptive prediction. (See the results presented in Fig. 3.)

An adaptive procedure for the vector quantizer is also needed. To track the nonstationarity of the speech waveform, the ideal solution would be to design a new codebook in real time for each frame of speech. This, however, is totally impractical because of the enormous computational complexity that would be needed and the excessive amount of side information needed to transmit a codebook in each frame. Good performance may be expected from a system which simply adapts the codebook to each talker. Such a system would avoid the performance degradation that results when a codebook designed on a given speech database is used to encode speech that is not in the database. (The results "inside" and "outside" the training sequence will be discussed in the next section.) The problem is that such a vector quantizer would require a delay of at least 5-10 s to design the optimal codebook for each talker, before transmission can begin.

The adaptive procedure which was chosen for the vector quantizer in the AVPC system is based on switching the codebooks in each frame according to the classification of the signal. Hence, in order to incorporate adaptive features into the vector predictive coding system, a device which will classify each speech frame into one of a few statistical classes is needed. Then, different codebooks and different prediction matrices will be used for each class. The information about the class will be transmitted to the receiver as side information.

The memory required for the system will increase in proportion to the number of classes (due to separate codebooks and predictors being used for each class). Also, as the number of classes increases, the complexity needed to perform the classification (feature extraction and selection) increases. The increase in the transmission rate due to the side information for m classes will be $\log_2 m$ bits per frame. Taking all these considerations into account, a classifier was designed for only three classes using two simple frame statistics, the variance and the first reflection coefficient. For application to digital telephony, a fourth class that could be added to the system would be used to identify voiceband data; however, the work reported here does not include this feature. This is certainly an "ad hoc" classifier, chosen for its simplicity. A classifier that is in some respects similar was used for switching predictors in scalar ADPCM [18]. Other studies are in progress to design classifiers in a more rigorous way, based on the same rationale that is used in codebook design.

There are also other ways to generalize and expand this classification approach to obtain better performance at the expense of increased complexity. One way would be to classify the frames using the basic acoustical categories of sounds (nasals, plosives, fricatives, etc.). Designing different codebooks and predictors for each such class will significantly improve the performance of the system, but the complexity of such a classifier will probably come close to the complexity of a speech recognition system.

The design of the classifier was based on an examination of the statistical characteristics of speech frames. Histograms of the first normalized correlation coefficient and of the frame variance for 1029 frames of speech sampled at 8 kHz were computed. On the basis of the analysis of these statistics, the classifier presented in Table I was chosen. The three classes were chosen to approximately divide the frames into three equally frequent groups, where each group covers a limited subrange of correlation and energy values.

Here $r(1)$ is the normalized autocorrelation coefficient for unit lag, σ is the frame variance, and σ_0 is the long-term variance computed over the entire speech database.

The first class contains highly correlated, high-energy frames. These frames generally contain voiced speech and are expected to give high prediction gain if an adequate predictor is designed for the class. The second class contains high-energy frames with lower correlation and, hence, lower prediction gain. Voiced speech with a high first formant may be an example of this class. The third class contains low energy and/or low correlation frames, as, for example, unvoiced sounds. It should be noted that this classification is not strictly related to the traditional voiced/unvoiced dichotomy. In fact, the voiced–unvoiced classification is not adequate in this context. Indeed, there are, for example, unvoiced sounds with high energy and correlation (plosives), which have very high predictability and should not be classified together with such unvoiced sounds as fricatives with very low predictability, etc. Typical examples of waveforms from the three classes are presented in Fig. 5.

Although these considerations are related to prediction gain or "predictability," they generally are also true for the vector quantizer. The performance of any given vector quantizer strongly depends on the energy level and autocorrelation of the input signal [7]. Since those frames which have energy and first autocorrelation coefficient in some given restricted range are grouped into the same class, an improvement in the performance achieved by each codebook can be expected.

5) Complexity of the AVPC System: Computational complexity is highly processor-dependent and is difficult to estimate without consideration of particular architectures. Nevertheless, we can make some preliminary assessment by assuming that multiplications are the most demanding basic operation needed. Although additions can often require equal execution times, the count of multiplications still gives a useful order-of-magnitude estimate of complexity. We shall see that the computational complexity of the AVPC system depends primarily on how the vector quantizer is implemented.

The classifier requires only two multiplications per sample to determine the frame energy and first autocorrelation value. Assuming that one-stage prediction is used, the vector linear predictor requires k multiplications per sample to perform the k by k matrix multiplication for input vectors with dimension k. A reduced computational complexity for the vector quantizer can be achieved by designing a codebook with a binary tree structure, allowing a rapid binary search implementation of the encoding operation [2]. This structural constraint does indeed degrade the performance of the quantizer, but the reduction in complexity may justify its use in some situations. In this case, the encoding operation requires only $2kr$ multiplications per sample, where r is the rate in bits per sample. On the other hand, a full-search implementation would require 2^{kr} multiplications per sample. The memory requirement for codebook storage is is $k2^{kr}$ words for a full-search codebook and double that figure for a binary-tree codebook. The predictor memory for a one-stage predictor is k^2 words.

In summary, the AVPC system with a full-search vector quantizer requires a number of multiplications per sample equal to

$$\mathrm{MUL}_{\mathrm{AVPC}} = 2^{kr} + k + 2.$$

If a binary tree is used for the vector quantizer implementation, the number of multiplications per sample becomes

$$\mathrm{MULB}_{\mathrm{AVPC}} = 2kr + k + 2.$$

Let c be the number of classes implemented in the AVPC system. Then, the memory required for the full-search implementation will be

$$\mathrm{MEM}_{\mathrm{AVPC}} = ck2^{kr} + ck^2$$

and for the binary-search implementation

$$\mathrm{MEMB}_{\mathrm{AVPC}} = 2ck2^{kr} + ck^2.$$

As an example, an AVPC system at a rate of 2 bits/sample, dimension $k = 4$, using binary search and three classes requires 22 multiplications per sample and a memory of about 6000 words and, hence, may be characterized as a medium complexity coder.

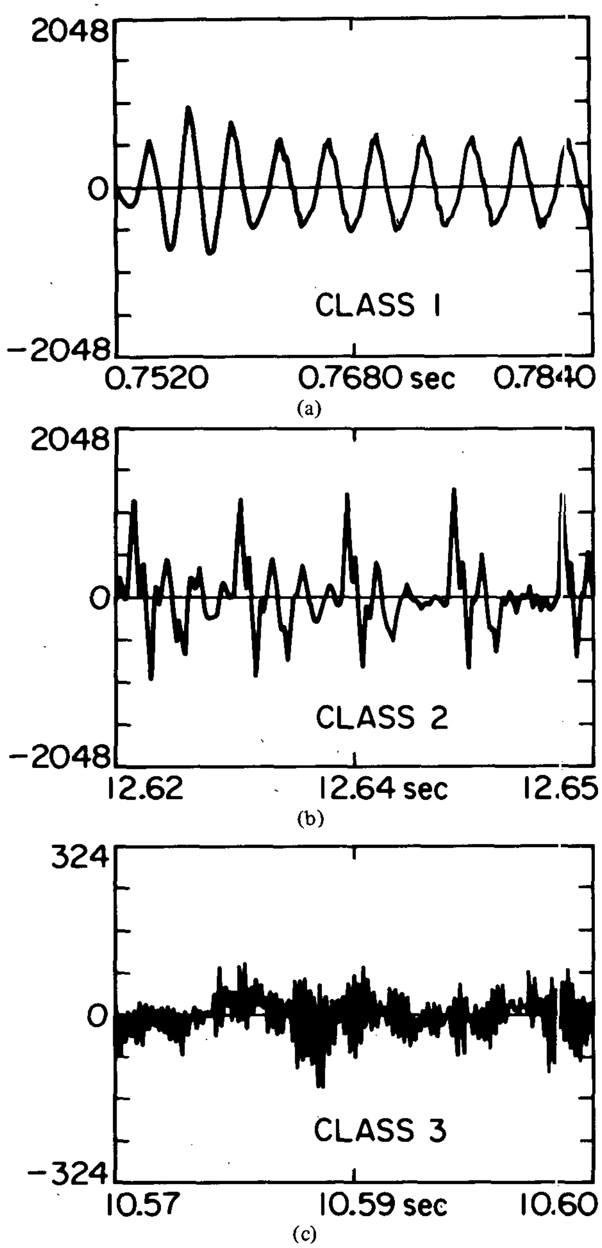


Fig. 5. Typical examples of speech waveforms for the three classes.

III. Experimental Results for the AVPC System

The AVPC system was simulated on both the VAX 11/780 and PDP 11/24 computers and tested using the speech database resident on these computers. Two main versions of the AVPC system were simulated, one in which the predictor and quantizer were optimized under the regular MSE criterion and the other optimized under the EWMSE criterion. In each version, the coding performance was evaluated by measuring the conventionally defined signal-to-noise ratios in both the MSE sense (SNR) and the segmental sense (SEGSNR), as frequently used by other speech coding researchers. Both versions of the AVPC system may be operated in a training mode or in a test mode. In the training mode, a locally optimal codebook is designed for the given input speech data, and the performance of the coder operating on these training data is measured. In the test mode, the coder performance is tested on a different set of speech data from the training set used to design the quantizer. The frame length for all experiments is 7.5 ms (60 samples) unless otherwise indicated.

One of the distinctive features of vector prediction of a blocked scalar process is that each component of the vector to be predicted has a different degree of predictability, depending on the time lag of the component from the observable data samples. This suggests that the AVPC system might introduce a coding noise that varies periodically with a period equal to the vector dimension, thereby introducing an undesirable periodic noise component. In fact, this does not happen, because the vector quantizer is designed to be matched to the statistics of the prediction error vectors.

To gain insight into the operation of the AVPC system, we first examine the variation of the signal-to-noise ratio with the component index inside a vector. Let s_n be an input vector, $\tilde{s}_n$ the reconstructed vector at the output of the system, and s_{ni}, $\tilde{s}_{ni}$ the ith components of these vectors. The vectors are defined from the scalar process, so that increasing the index i corresponds to moving forward in time. Then the MSE signal-to-noise ratio for the ith component may be defined by

$$(\mathrm{SNR})_i = \frac{E[s_{ni}^2]}{E[(\tilde{s}_{ni} - s_{ni})^2]}.$$

This definition is applied to each frame by computing the time average over the frame; then the geometric mean over all frames is found. The result obtained is the per-component segmental signal-to-noise ratio. The same expression defines the per-component prediction gain if the quantizer is eliminated, so that $\tilde{s}_{ni} - s_{ni}$ is simply the prediction error.

Fig. 6 presents the variation of both the per-component prediction gain and the per-component signal-to-noise ratio (SEGSNR) versus the component index i, for the AVPC system in dimension 5 at a rate of 2 bits/sample. Although the prediction gain significantly decreases as the vector component index increases, the input–output signal-to-noise ratio remains practically constant. This remarkable result is due to the fact that the vector quantizer compensates for the nonuniformity introduced by vector prediction. The compensating effect can be understood by examining the shape of the vectors in the codebook. Fig. 7 displays the superimposed codevectors for each of the three codebooks which are used to quantize the prediction error vectors. Each codevector consists of five samples ($k = 5$) which are interpolated in the plot to show a complete waveform segment. As can be seen, the shape of the vectors in each class reflects the statistical properties used to define that class.

Fig. 8 shows the overall performance of the AVPC system, evaluated by MSE and segmental signal-to-noise ratios versus the dimension of the input vectors. The MSE SNR was measured for the AVPC system optimized under the MSE criterion, and the SEGSNR for the system optimized under the EWMSE criterion. For comparison, the performances of some of the best waveform coding systems are noted in the same figure. The performance for standard ADPCM and pitch-adaptive DPCM are taken, respectively, from [16] and [17], the performance for sequential adaptive DPCM from [18], and the performance for tree and trellis coding from [19] and [20]. Fig. 8 shows that the AVPC system is competitive with such high-complexity systems as tree and trellis coding. The performance achieved by the AVPC system is significantly better than the performance of the known scalar versions of ADPCM. One should note that, generally, different researchers use different speech material, and this may produce slight variations in the SNR results (usually less than 1–2 dB). Hence, the comparison with other coding systems gives only an approximate idea of relative performance.

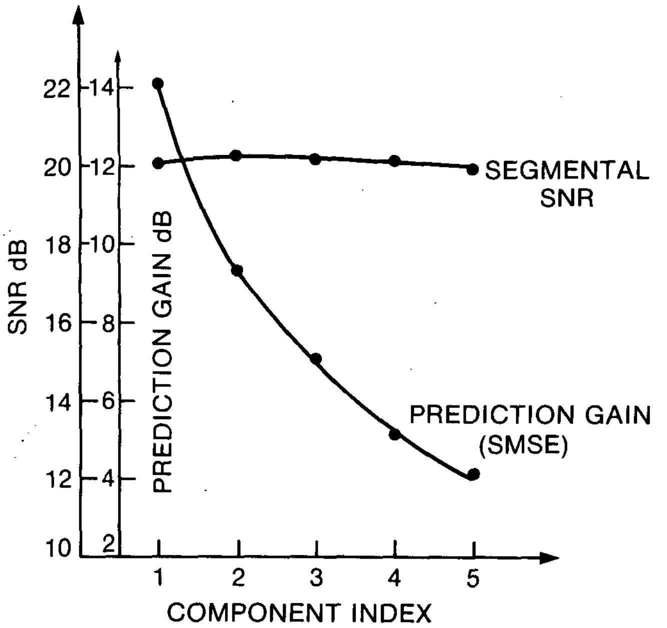


Fig. 6. Variation of segmental SNR with the component index for dimension 5 at 2 bits/sample.

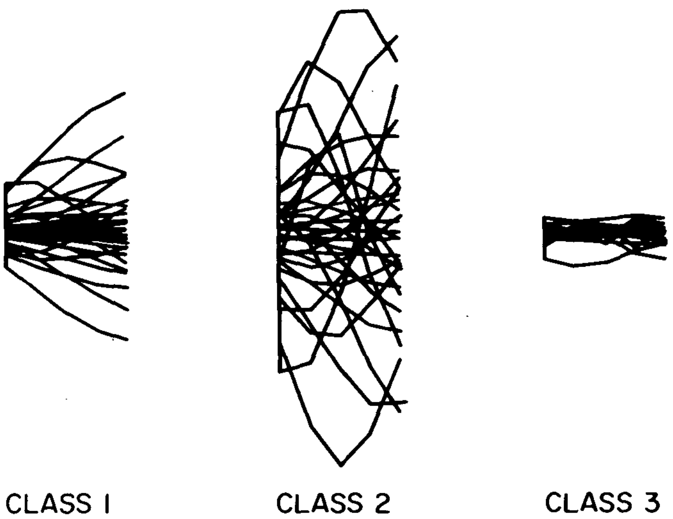


Fig. 7. Codebook for AVPC system for dimension 5 at 1 bit/sample.

It is difficult to give a meaningful complexity comparison between the above-mentioned coding systems, since realistic measures of complexity are dependent on the architecture that is used. However, to give the reader at least a crude idea at this point, we estimate that the AVPC system (using binary search) would require about 22 multiplications/sample, compared to 16 for sequential scalar ADPCM and 110 multiplications/sample for tree-trellis coding. (At low rates, such as 1 bit/sample, tree-trellis coding may have a much lower complexity, using lookup tables for path selection.)

The AVPC system optimized using the EWMSE criterion achieves better segmental SNR (SEGSNR) than the same system designed using the usual MSE. Table II presents the SEGSNR results for the AVPC system designed using the EWMSE criterion (labeled EWMSE system in the table) and for the AVPC system designed using the MSE criterion (labeled MSE system). The EWMSE system achieves an improvement of 1.1–1.4 dB in SEGSNR. For both systems the rate was 2 bits/sample.

Informal listening tests of the processed speech indicate

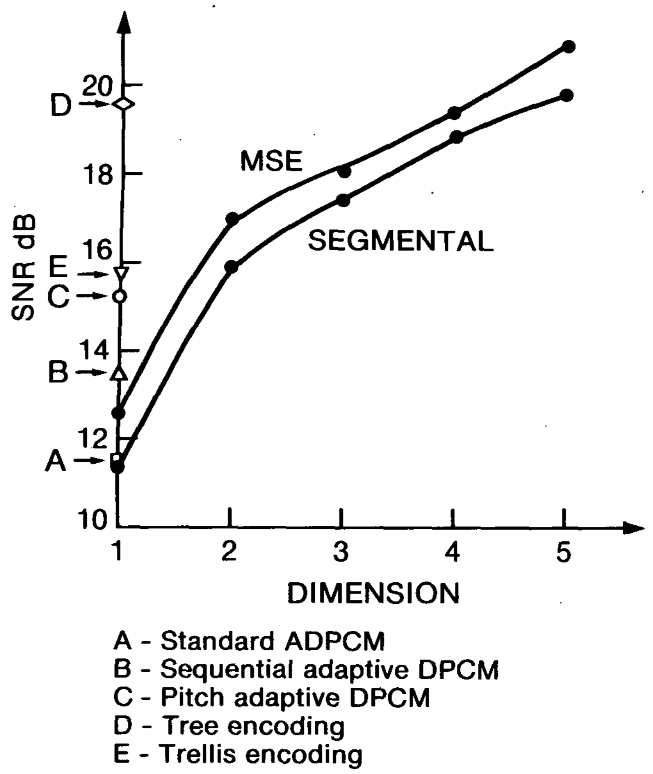


Fig. 8. MSE and segmental SNR versus dimension for the AVPC system.

TABLE II
SEGSNR IN DECIBELS AT 2 BITS/SAMPLE

Dimension	1	2	3	4	5
EWMSE system	11.4	16.0	17.5	18.9	19.9
MSE system	10.3	14.6	16.4	17.8	18.6

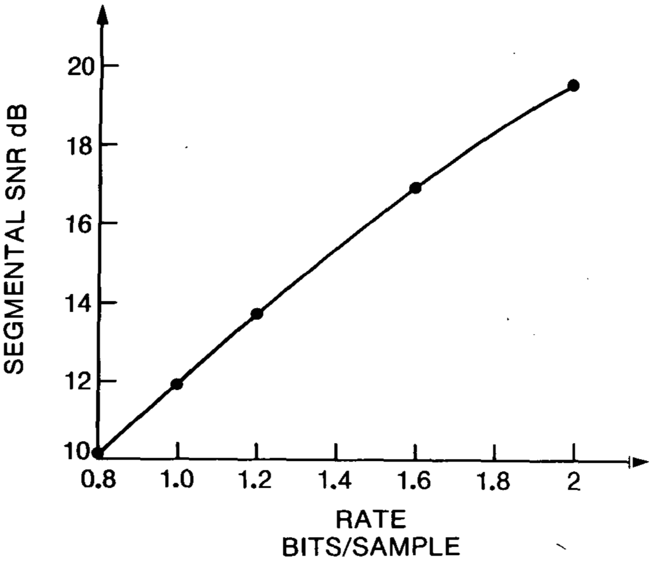


Fig. 9. Segmental SNR versus rate for dimension 5.

good communications quality for rates close to 2 bits/sample and for vector dimensions 4 or 5. The subjective quality is better for the system optimized under the EWMSE criterion, verifying that this criterion is more suitable than the regular MSE for optimizing speech quality.

The results of Fig. 8 were obtained by encoding the same speech database as that used for designing the vector quantizer codebook and the predictor. This database (see the Appendix) contains 16 s of speech from six different talkers, four males and two females. Whenever the design of the codebook and the performance evaluation are done on the same data, the corresponding results are described as "inside the training sequence" or simply "training mode." The results for the tree and trellis coders noted in Fig. 8 were also inside the training sequence (optimum codebooks for the given data were used). When encoding speech which is not the design database, the results are described as "outside the training sequence" or "test mode." It is obvious that, to obtain good performance outside the training sequence, large databases containing a large variety of speech should be used for system design.

In the design procedure, each codevector in the codebook is defined by a centroid computation, averaging over a cluster of training vectors. Hence, the average number of training vectors per codevector, which we call the "training ratio" in vector quantizer design, should have a value acceptable from a statistical point of view. For example, at least 50–100 input vectors per codebook entry is desirable. Since the number of vectors in the codebook increases exponentially with the rate and the dimension, it follows that the size of the training sequence should also increase exponentially. In an example presented in [13], the same performance was obtained in both training and test modes by using a training sequence of 12 000 vectors for a codebook of size 16, i.e., a training ratio of 750. Such a high training ratio was not affordable with currently available resources for the codebook sizes of interest in this work. The limitation is primarily due to the considerable computer time needed for designing the codebook.

A first evaluation of the performance of the AVPC system outside of the training sequence was done by using the system designed on the database to encode a different talker (96 000 samples at 8 kHz sampling rate, male talker; see the Appendix). The SEGSNR in the test mode was found to be lower than the SEGSNR in the training mode by amounts ranging from 1 to 3.5 dB, the gap increasing with the dimension. The increase of this gap with the dimension is due to the exponential decrease of the training ratio with increasing dimension when the size of the training sequence is constant. (Recall that the number of codevectors grows exponentially with dimension for a fixed rate.)

The variation of performance of the AVPC system with the bit rate is presented in Fig. 9 for dimension 5. The signal-to-noise ratio (SEGSNR) in decibels increases smoothly with the rate, the curve being close to a straight line with a slope of about 9.5 dB/bit. This is different from the familiar scalar PCM slope of 6 dB/bit. In fact, the vector quantizer itself has a slope higher than 6 dB/bit in this range of rates. Note that the asymptotic theory of vector quantization gives 6 dB/bit in the high-rate region [7].

Fig. 10 shows the quantization error waveform of the AVPC system (dimension 5, rate 2 bits/sample) versus time for a typical speech waveform (15 frames of 60 samples each at 8 kHz sampling frequency). The original waveform is also shown in Fig. 10 with the same time scale. Fig. 11 presents the spectrum of quantization error compared to the spectrum of the original speech. The spectrum of the quantization error is generally flat, indicating a "white noise" type of distortion. This was confirmed by listening to the reconstructed speech.

It would be of interest to compare AVPC to ADPCM incorporating noise shaping [22], [23], which reportedly has good perceptual quality, but this was not done since SNR is an inadequate performance indicator for the latter technique, and adequate speech material for a subjective comparison was not available. A comprehensive subjective study would be a worthwhile future project.

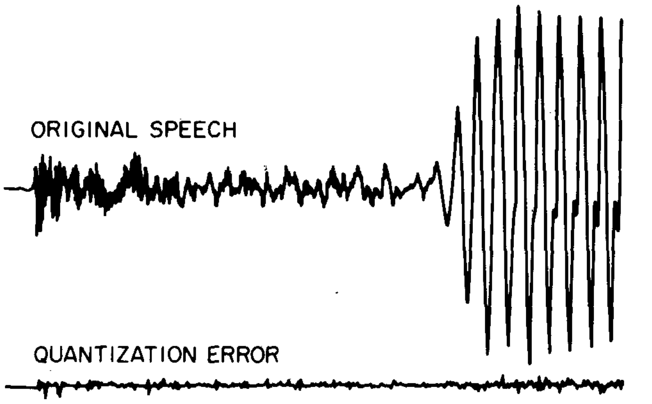


Fig. 10. Original speech and quantization error waveforms for dimension 5 at 2 bits/sample (16 kbits/s).

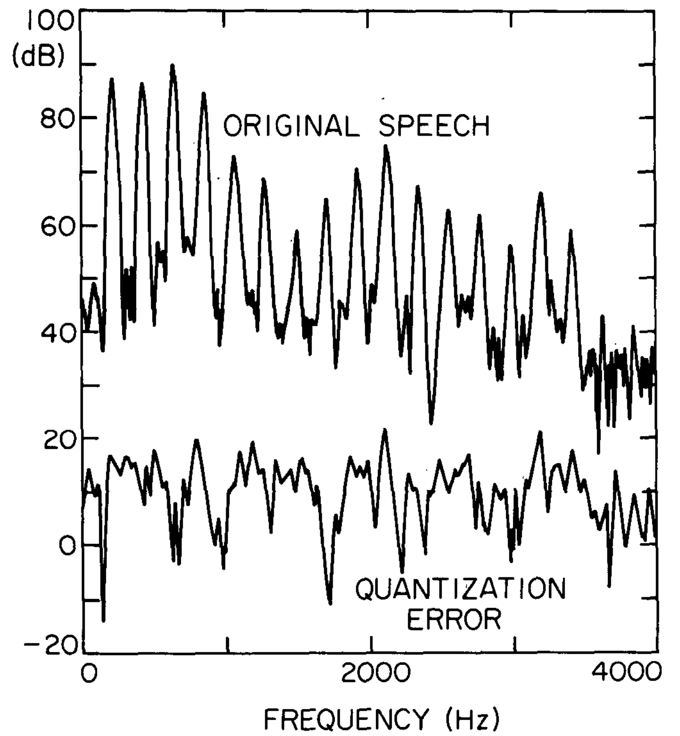


Fig. 11. Spectrum of the original speech and of the quantization error.

IV. Conclusions

Our results indicate that vector quantization may be used as a basis for a new and powerful class of waveform coding devices. These devices encode vectors, rather than discrete samples of the process, and use, together with vector quantization, vector linear prediction and adaptive techniques. An improvement of about 7 dB in SNR over the previous vector quantization studies was obtained using the adaptive vector predictive coder. The experimental results are significantly better than known results for ADPCM, the corresponding scalar coding method. At a rate of 2 bits/sample (16 kbits/s), a signal-to-noise ratio of about 20 dB was obtained, and the corresponding subjective speech quality may be described as good communications quality. These results compete favorably with such methods as tree and trellis coding at the same bit rate. We are optimistic that, with some further improvements and the use of larger training sets for the codebook design, the AVPC system may offer a viable option for a 16 kbit/s single-chip speech codec that is well within current VLSI capabilities.

TABLE III
Average Pitch Values

Talker	F1	F2	M1	M2	M3	M4
Pitch Hz	270	220	140	100	170	120

Appendix

The database contains 16 s of speech from six different talkers, four males and two females. The talkers will be denoted M1–M4 (male talkers) and F1 and F2 (female talkers), respectively. The database contains one utterance for each talker as follows.

F1) "The pipe began to rust while new."
F2) "Add the sum to the product of these three."
M1) "Open the crate but don't break the glass."
M2) "Oak is strong and also gives shade."
M3) "Thieves who rob friends deserve jail."
M4) "Cats and dogs each hate the other."

As can be seen, these utterances are well balanced, including voiced speech, plosives, fricatives, etc. The speech waveform was bandpass filtered at 200–3900 Hz and sampled at 8 kHz. A high-slope elliptical filter was used for low-pass filtering to avoid aliasing. The analog-to-digital conversion was performed using a 12 bit A/D converter. The main statistics of the sampled signal are: peak value 1792, mean value 1.165 (nominally zero), and standard deviation 279.23. Table III gives the average pitch for each talker.

The speech file used to test the performance outside of the training sequence contains 12 s of speech (96 000 samples at 8000 Hz sampling rate) from one male talker. The utterance used is:

M5) "When the sunlight strikes rain drops in the air they act like a prism and form a rainbow. The rainbow is a division of white light into many beautiful colors."

The average pitch for this utterance is about 150 Hz.

References

[1] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-28, pp. 562–574, Oct. 1980.

[2] H. Abut, R. M. Gray, and G. Rebolledo, "Vector quantization of speech and speech-like waveforms," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-30, pp. 423–436, June 1982.

[3] B. H. Juang, "Multiple stage vector quantization for speech coding," in *Proc. IEEE ICASSP,* Paris, France, May 1982, pp. 597–600.

[4] A. Gersho and V. Cuperman, "Speech waveform coding using vector quantization," in *Proc. IEEE GLOBECOM,* San Diego, CA, 1983.

[5] J. D. Gibson, "Adaptive prediction in speech differential encoding systems," *Proc. IEEE,* vol. 68, pp. 488–525, Apr. 1980.

[6] N. S. Jayant and P. Noll, *Digital Coding of Waveforms.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[7] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory,* vol. IT-25, pp. 373–380, July 1979.

[8] R. M. Gray, J. Kieffer, and Y. Linde, "Locally optimal block quantization for sources without a statistical model," Inform. Syst. Lab., Stanford Univ., Stanford, CA, Tech. Rep. L-904-1, May 1979.

[9] P. Noll, "Adaptive quantizing in speech coding systems," in *Proc. Int. Zurich Sem. Digital Commun.,* Zurich, Switzerland, Mar. 1974, pp. B3.1–B3.6.

[10] P. Whittle, "On the fitting of multivariate autoregressions and approximate factorization of a spectral density matrix," *Biometrica,* vol. 50, pp. 129–134, 1963.

[11] R. A. Wiggins and E. A. Robinson, "Recursive solution to the multichannel filtering problem," *J. Geophys. Res.,* vol. 70, pp. 1885–1891, Apr. 1965.

[12] Y. Inouye, "Modeling of multichannel time series and extrapolation of matrix-valued autocorrelation sequences," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-31, pp. 45–55, Feb. 1983.

[13] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.

[14] G. H. Ball and D. J. Hall, "A clustering technique for summarizing multivariate data," *Behavioral Sci.*, vol. 12, pp. 153-155, Mar. 1967.

[15] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech.* New York: Springer-Verlag, 1976.

[16] N. S. Jayant, "Digital coding of waveforms: PCM, DPCM and DM quantizers," *Proc. IEEE*, vol. 62, pp. 611-632, 1974.

[17] ——, "Pitch-adaptive DPCM coding of speech with two-bit quantization and fixed spectrum prediction," *Bell Syst. Tech. J.*, vol. 56, pp. 439-454, Mar. 1977.

[18] C. S. Xydeas, C. C. Evci, and R. Steele, "Sequential adaptive predictors for ADPCM speech encoders," *IEEE Trans. Commun.*, vol. COM-30, pp. 1942-1954, Aug. 1982.

[19] J. B. Anderson and J. B. Bodie, "Tree encoding of speech," *IEEE Trans. Inform Theory*, vol. IT-21, pp. 379-387, July 1975.

[20] L. C. Stewart, R. M. Gray, and Y. Linde, "The design of trellis waveform coders," *IEEE Trans. Commun.*, vol. COM-30, pp. 702-710, Apr. 1982.

[21] R. M. Gray and E. D. Karnin, "Multiple local optima in vector quantizers," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 256-261, Mar. 1982.

[22] M. R. Schroeder and B. S. Atal, "Predictive coding of speech signals and subjective error criteria," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 247-254, June 1979.

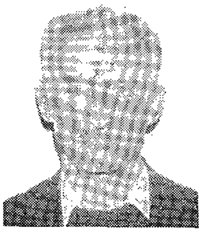
[23] J. Makhoul and M. Berouti, "Adaptive noise spectral shaping and entropy coding in predictive coding of speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 63-73, Feb. 1979.

[24] V. Cuperman and A. Gersho, "Adaptive differential vector coding of speech," in *Conf. Rec. IEEE GLOBECOM*, Dec. 1982, pp. 1092-1096.

[25] J. Adoul, J. Debray, and D. Dalle, "Spectral distance measure applied to the optimum design of DPCM coders with *L* predictors," in *Proc. IEEE ICASSP*, Denver, CO, Apr. 1980, pp. 512-515.

[26] R. M. Gray, "Vector quantization," *ASSP Mag.*, vol. 1, Apr. 1984.

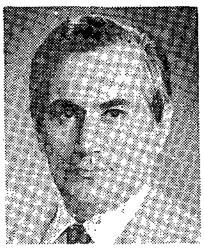
[27] T. R. Fischer, "Quantized control with differential pulse code modulation," in *Proc. 21st Conf. Decision Contr.*, 1983.



Vladimir Cuperman (S'80-M'83) was born in Bucharest, Rumania, in 1938. He received the B.S. degree in electrical engineering (communications) from the Polytechnic of Bucharest in June 1960, the M.A. degree in mathematics from the University Bucharest in June 1972, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara, in March 1981 and June 1983, respectively.

From 1963 to 1974, he worked at the Institute for Automation, Bucharest, on data transmission and error correcting codes. Since 1974, he has worked at Tadiran, Israel, on digital communications, speech, and image coding. From January 1981 to March 1983 he was a Teaching Assistant, Research Assistant, and Research Associate at U.C.S.B. He is currently managing research and development in speech processing at Calltalk, Ltd., Tel-Aviv, Israel, and is also a Visiting Professor at the University of Tel-Aviv.

★



Allen Gersho (S'58-M'64-SM'78-F'82) received the B.S. degree from the Massachusetts Institute of Technology, Cambridge, in 1960, and the Ph.D. degree from Cornell University, Ithaca, NY, in 1963.

He is a Professor of Electrical and Computer Engineering at the University of California, Santa Barabara, where his current research interests are in the area of speech and image processing with a focus on the use of vector quantization techniques. He was at Bell Laboratories from 1963 to 1980, where he was engaged in research in signal processing for communications.

Dr. Gersho has served as Editor of the IEEE COMMUNICATIONS MAGAZINE and Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS. He received the Guillemen-Cauer Prize Paper Award in 1980, the Donald McLellan Award in 1983, and an IEEE Centennial Medal in 1984. He also served on the Board of Governors of the IEEE Communications Society from 1981 to 1984.