A Baseband Residual Vector Quantization Algorithm for Voiceband Data Signals

THAO D. TRAN, V. JOHN MATHEWS, MEMBER, IEEE, AND CRAIG K. RUSHFORTH, SENIOR MEMBER, IEEE

***Abstract*—In this paper, we present a new approach to the digitization and compression of a class of voiceband modem signals. Our approach, which we call baseband residual vector quantization (BRVQ), relies heavily upon the simple structure present in a modem signal. After the signal is converted to baseband, the magnitude sequence and the sequence of residuals obtained when the phase within each baud of the baseband signal is modeled by a straight line are separately vector quantized. In order to carry out these operations, we developed the new carrier-frequency estimation and baud-rate classification schemes described in the paper. Experimental results show that the performance of the BRVQ system at and below 16 kbits/s is better than that of a previously developed vector quantization scheme that has itself been shown to outperform traditional speech-compression techniques such as adaptive predictive coding, adaptive transform coding, and subband coding when these techniques are used to compress modem signals.**

I. Introduction

IT IS frequently necessary to digitize and store a waveform for subsequent analysis or retransmission. When this waveform is a modem signal, it should be encoded with sufficient fidelity that both the information sequence carried by the waveform and the important features of the waveform itself are adequately preserved. At the same time, it is desirable that the encoding be done with as few bits as possible in order to reduce the required memory or transmission capacity. One way to achieve an acceptable tradeoff between these two conflicting objectives is to design an encoder that is carefully matched to the structure of the class of waveforms to be encoded. To a degree, most waveform encoders incorporate information about the waveforms to be encoded and about how the resulting coded signals are to be used. On the other hand, an encoder whose structure is highly dependent upon the characteristics of a particular class of waveforms may not work well when applied to waveforms with significantly different characteristics. The choice between a coder strongly tuned to a particular class of waveforms and one that is more broadly applicable will depend upon the waveforms to be encoded, the required fidelity, the complexities of the alternatives being considered, and many other elements of the specific problem to be addressed.

The objective that motivated the work described in this paper was to develop an efficient waveform coder that would encode with high fidelity any member of a specific set of voiceband data signals employing a wide range of modulation types, carrier frequencies, and bit rates. The technique we developed to meet this objective makes deliberate and extensive use of the simple structure of data signals, and it probably will not work particularly well on other types of signals. In particular, we would not advocate its application to speech signals. It could, however, be incorporated along with an appropriate speech encoder into a larger system designed to handle a mixture of voice and data signals in the telephone network if the demands on performance justified the extra cost.

Compression of voiceband data signals has not attracted much attention until recently, and consequently the literature on this subject is quite limited. O'Neal and Stroh [1] examined the performance of differential pulse code modulation (DPCM) applied to both speech and data signals. They showed that a DPCM system can be built that performs better than a PCM system for speech signals and is as good as PCM for data signals with raised cosine spectra. O'Neal [2] later conducted an analytical study of the performance of delta modulation on various voiceband data signals. Transmission of data signals using companded delta modulation was evaluated by May *et al.* [3]. Their results showed that delta modulation performs at least as well as PCM operating at the same effective channel bit rate.

Petr [4] developed a new adaptive differential PCM (ADPCM) algorithm operating at 32 kbits/s for speech and voiceband data signals. His system, which he calls ADPCM with a dynamic locking quantizer (ADPCM–DLQ), was shown to perform nearly as well as a PCM coder operating at 64 kbits/s.

Recently, Anderton [5], [6] developed a scheme known as adaptive baseband codebook vector quantization (ABCVQ). In this method, a sequence of passband vectors is first computed from a set of baseband codebooks using the estimated carrier frequency. A given bauded signal vector is encoded into that passband code vector which is closest to it in Euclidean distance. Optimal encoding requires the solution of a transcendental equation for each element of the codebook, the computation of several inner products, and the determination of the distance to each code vector.

The solution of the transcendental equation must be obtained numerically and is computationally expensive. A suboptimal solution that requires the computation of some trigonometric functions has been devised by Anderton [5]. Data encoding using this method requires $(2 + 5/k)L$ multiplies, $(2 + 2/k)L$ adds, and $6L/k$ trigonometric function computations per sample where L and k are the size of the codebook and the dimension of the data vector, respectively. Different types of bauded signals are accommodated in the ABCVQ algorithm by using four different codebooks in parallel, each of which is tuned to a specific subclass of bauded signals. Extensive simulations conducted by Anderton have shown that ABCVQ exhibits better performance, as measured by several different criteria, than that of adaptive predictive coding, adaptive transform coding, or subband coding at transmission rates of 16 kbits/s or less.

In this paper, we present an alternative algorithm for compressing voiceband data signals which we call baseband residual vector quantization (BRVQ). This algorithm takes

Paper approved by the Editor for Speech Processing of the IEEE Communications Society. Manuscript received November 15, 1987; revised May 15, 1988. This work was supported in part by the Communication Systems Division of the Unisys Corporation under University of Utah Contract 5-25196. This paper was presented at GLOBECOM'87, Tokyo, Japan, November 1987.

T. D. Tran was with the Department of Electrical Engineering, University of Utah, Salt Lake City, UT 84112. He is now with the IBM Corporation, Manassas, VA 22110.

V. J. Mathews and C. K. Rushforth are with the Department of Electrical Engineering, University of Utah, Salt Lake City, UT 84112.

IEEE Log Number 8929595.

0090-6778/89/0900-0949$01.00

advantage of the structure of bauded signals to improve the efficiency and reduce the complexity of the system as much as possible while still retaining its ability to handle modem signals employing a variety of modulation types and a wide range of parameter values.

After the signal is converted to baseband, the magnitude sequence and the sequence of residuals obtained when the phase within each baud of the baseband signal is modeled by a straight line are separately vector quantized. In order to carry out these operations, we developed the new carrier-frequency estimation and baud-rate classification schemes described later in the paper. Although the phase model does not directly take into account possible pulse shaping of the baseband signals, information about such pulse shaping will be contained in the residual sequence.

There are two reasons for quantizing the residuals rather than the phase itself. First, the residual usually has a smaller dynamic range than that of the phase sequence. As a result, a vector quantizer using a fixed number of bits will generally perform better with the residual than with the phase sequence. Second, since the residual sequence does not have as much structure as the phase sequence, the BRVQ algorithm is robust to variations in modulation types. In fact, we will show that it is possible to construct a single codebook that is adequate for the compression of several different types of modulation.

The BRVQ algorithm has several advantages. First, as noted above, it performs very well even when a single codebook is used for encoding several different types of voiceband data signal. Second, the system has very low sensitivity to errors in carrier-frequency estimation. Third, even when a single codebook is used, the BRVQ algorithm outperforms common speech-compression techniques when applied to voiceband data signals, and at the same time achieves a performance comparable to that of the more complicated ABCVQ method. In essence, the BRVQ scheme described in this paper is a conceptually simple algorithm that is robust to changes in signal type within a broad class of modem signals and to errors in the estimation of several of the parameters involved. The method is somewhat simpler than the ABCVQ algorithm, and it can be implemented to operate in real time using modern VLSI technology.

The rest of the paper is organized as follows. In Section II, we provide a more formal statement of the problem and introduce the baseband residual vector quantization algorithm. Experimental results demonstrating the ability of the BRVQ to perform well at low data rates are presented in Section III. This section also contains a discussion of several aspects of the BRVQ algorithm. Concluding remarks are contained in Section IV.

II. The Baseband Residual Vector Quantization Algorithm

Consider a voiceband data signal of the form

$$s(t)=\mathrm{Re}\ [g(t)e^{-j(2\pi f_c t+\theta)}] \qquad (1)$$

where Re $[\cdot]$ denotes the real part, f_c is the carrier frequency, θ is the initial phase of the carrier, and $g(t) = g_I(t) + jg_Q(t)$ is the equivalent information-bearing baseband signal. $g_I(t)$ and $g_Q(t)$ are referred to as the in-phase and quadrature components, respectively, of $g(t)$. Equation (1) can also be written as

$$s(t)=Am(t)\cos\ [2\pi f_c t+p(t)+\theta] \qquad (2)$$

where A denotes the amplitude of the signal, $m(t)$ represents the pulse shape, and $p(t)$ is the phase (the information-bearing signal) of $s(t)$.

The class of signals to be considered in this paper includes differentially encoded binary, quadrature, and octal phase-shift-keyed signals (DBPSK, DQPSK, DOPSK), coherent binary phase-shift-keyed signals (CBPSK), and continuous-phase frequency-shift-keyed (CFSK) signals, all with information rates of 4800 bits/s or less. The signals were initially sampled at a rate of 8000 samples/s and quantized to 8 bits per sample, resulting in a data rate of 64 kbits/s. Our objective is to design a data compression algorithm that will work well for this class of signals when the transmitted data rate is reduced from 64 kbits/s to 16 kbits/s or less.

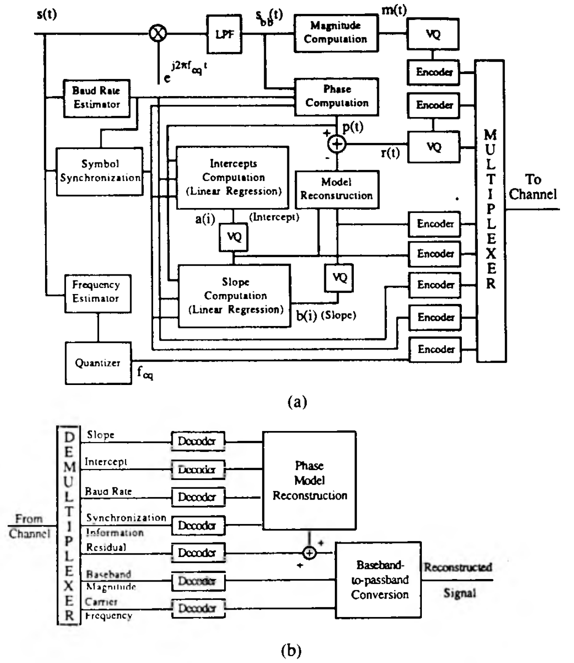


Fig. 1. Block diagram of the baseband residual vector quantization system. (a) Transmitter. (b) Receiver.

Fig. 1 is a block diagram of the system we propose as a solution to this problem. The carrier frequency of the received signal $s(t)$ is first estimated using a novel approach to be described later, and then the corresponding baseband signal is obtained using a quantized version of this carrier-frequency estimate. If $s(t)$ is given by (2), with θ taken to be zero for convenience, the corresponding baseband signal $s_{bb}(t)$ will be

$$s_{bb}=0.5Am(t)e^{-j[2\pi(f_c-f_{cq})t+p(t)]} \qquad (3)$$

where f_{cq} is the quantized estimate of the carrier frequency. This complex baseband signal is then sampled to create a sequence of vectors whose magnitudes and phases are vector quantized separately, the magnitudes directly and the phases with the help of a model.

The unwrapped phase within a single baud of the baseband signal can be approximately modeled as a straight line. In the system of Fig. 1, the starting value (the intercept) of the straight-line model is first obtained using linear regression within each baud interval. The intercepts are vector quantized and used together with the unquantized phase to yield the slope of the line. The quantized model parameters are then used to compute the residual signal, the difference between the actual phase and that given by the model. The resulting sequence is then vector quantized. The residual sequence contains information about modeling errors, pulse shaping, and any other preprocessing performed on the baseband signals at the transmitter. Consequently, the reconstructed signal after quantization will retain most of the characteristics of the original pulse-shaped signal.

Since the modeling of the phase is done on a baud-by-baud basis, estimates of the baud rate and the baud boundaries of the input signal are required. In order to simplify the problem, we assume that the BRVQ system will encounter only a finite number of known and reasonably well-separated baud rates. This is a reasonable assumption when working with standard commercial modems, and it reduces the baud-rate estimation problem to a classification problem. We discuss our approaches to this classification problem and to the associated symbol synchronization problem in Section II-B.

The quantized magnitude and phase residual are encoded and sent to the channel along with the estimated carrier frequency, the baud rate estimate, the synchronization information, and the quantized values of the parameters of the straight line model for the phase sequence. At the receiver, the quantized passband signal is obtained from the reconstructed baseband signal as illustrated in Fig. 1.

Our system employs four vector quantizers in parallel, one each for the magnitude, the phase residuals, and the two parameters of the straight-line model for the baseband phase. The four codebooks required can be designed using the Linde-Buzo-Gray (LBG) algorithm [7], a widely used procedure that is a generalization of an algorithm developed by Lloyd [8] for scalar quantization. The training sequences for designing the four codebooks were obtained by computing the magnitudes, the phase residuals, and the straight-line model parameters of the baseband equivalents of a long sequence of modem signals representative of the class of signals that the system will process during normal operation.

A. Carrier Frequency Estimation

The process of converting the passband signal into its baseband equivalent requires knowledge of the carrier frequency. In most practical situations, the carrier frequency is not known *a priori* and must be etimated. The problem of estimating the frequency of a sinusoid embedded in noise has been studied by many researchers [9]–[13]. Because of the time-varying and possible discontinuous nature of the phase, the methods in [9]–[13] cannot be used without modification to estimate the carrier frequency of a bauded signal. In our approach, the carrier frequency is computed as the average of the derivative of the instantaneous phase [14] of the passband signal. Because of possible phase jumps, the estimates of the phase derivative at the baud boundaries are not necessarily related to the carrier frequency. Before the actual frequency estimate is computed, these aberrant estimates of the phase derivative are removed by examining the first differences of the phase derivative estimates. Carrier-frequency estimation is carried out on contiguous nonoverlapping segments of the signal, typically of about 0.25 s duration. The resulting estimates are uniformly scalar quantized, and these quantized values are employed for baseband signal generation and are also transmitted to the receiver.

Experimental results have shown that the estimated frequencies obtained from our method are unbiased and have small variances. Moreover, our procedure has been shown to outperform several competing techniques [15]. Details of the carrier-frequency estimation algorithm may be found in [15], and thus are omitted here.

B. Baud-Rate Estimation and Symbol Synchronization

As mentioned earlier, the unwrapped phase within each baud is modeled as a straight line. This requires knowledge of the baud rate as well as the baud boundaries of the signals being transmitted. Since the type of transmitted signal, and therefore the baud rate, can change from time to time, it is important to estimate the baud rate in an adaptive fashion.

We have developed an accurate baud-rate estimation scheme for the class of bauded signals being studied. As stated earlier, our method assumes that the possible baud rates are known, finite in number, and significantly different from one another. Signals with different baud rates will therefore have significantly different bandwidths. This assumption is valid for a large class of standard modem signals. In particular, this assumption holds for a large class of modem signals satisfying CCITT Recommendations V.22, V.23, V.26, and V.27 [21] and employing raised cosine pulse shaping [22]. We estimate the power density spectrum of the received signal using the Blackman–Tukey algorithm [16], and we take the bandwidth to be the width of the interval over which that spectrum lies above a threshold. We then map the bandwidth estimate into a corresponding baud rate.

Several experiments were conducted to evaluate the performance of the baud-rate estimator. We used data signals of different modulation types (CFSK, CBPSK, DBPSK, DQPSK, and DOPSK), different baud rates, and different noise levels (probability of bit error up to 10^{-2}), each of duration approximately 10 s. For each type of data signal, the baud rate was estimated using contiguous nonoverlapping segments varying from 0.125 to 1.00 s. In every case, the baud-rate estimate was correct.

We take advantage of the cyclostationarity property [17] of bauded signals to synchronize the symbols. Under the assumptions that the noise process is zero mean and white, that the information-bearing signal $p(t)$ is zero mean and independent for different bauds, and that the pulse shape is such that its significant frequency components are less than the baud rate, one can easily show that the mean-squared value of the baseband signal $g(t)$ consists of a dc term and a sinusoid of frequency $1/T$ Hz where $1/T$ is the baud rate of the signal. This suggests the following approach for symbol synchronization. First, a timing waveform $w(t)$ is generated by passing the baseband signal through a bandpass filter with passband centered at $1/2T$ Hz, evaluating the magnitude squared value of the filter output, and then extracting the component of this sequence centered around $1/T$ Hz by passing it through another bandpass filter tuned to this frequency. That is,

$$w(t) = \{[h_1(t) * g_I(t)]^2 + [h_1(t) * g_Q(t)]^2\} * h_2(t) \quad (4)$$

where $*$ denotes convolution, and $h_1(t)$ and $h_2(t)$ are the unit impulse responses of the bandpass filters with passbands centered at $1/2T$ and $1/T$, respectively. It can be shown [17] that $w(t)$ is cyclostationary in the wide sense. That is, the mean timing waveforms $E[w(t)]$ and $E[w(t + \tau)w(t)]$ are both periodic functions of t. To be specific, the mean timing waveform $E[w(t)]$ was shown in [17] to be a periodic function of t whose period is T. As a result, the zero crossings of the mean timing waveform occur at a fixed time offset relative to the symbol edges when the signal is noise free. In the presence of noise, this is no longer true, and further processing of the timing waveform is required to yield accurate estimates of the baud boundaries.

The timing waveform $w(t)$ is next passed through a hard limiter to yield a rectangular waveform with the same zero crossings as $w(t)$. This hard-limited signal is then cross correlated with the desired clock signal for $M + 1$ different lags, and the timing phase is chosen to be that lag for which the cross-correlation estimate is maximum. The desired clock signal $c(t)$ is a hard-limited sinusoid with zero initial phase and frequency $1/T$ Hz.

The algorithm is illustrated in Fig. 2. The a_i are the cross-correlation estimates for lags $\tau_i = i\Delta t$, $0 \le i \le M$, Δt is the sampling interval, and $M\Delta t$ is the smallest integer multiple of Δt larger than or equal to the baud interval. Synchronization is performed on a block-by-block basis, with T_0 the block length in seconds.

The system of Fig. 2 was simulated on a digital computer. Experiments were performed on noise-free and noisy signals (bit-error probability $= 10^{-4}$) with different types of modula-

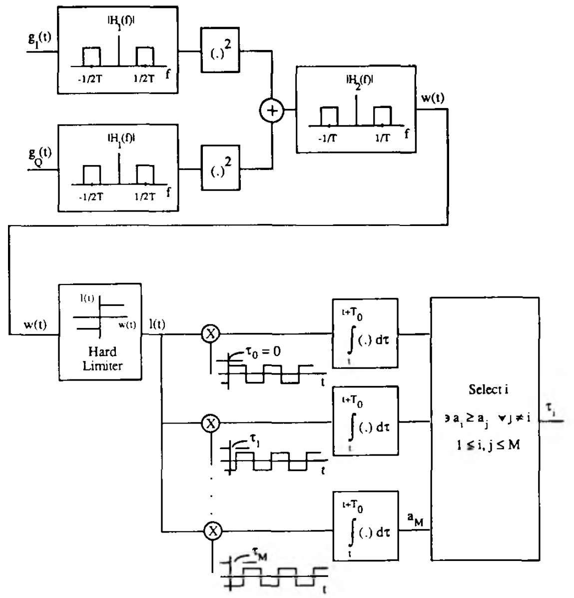


Fig. 2. Block diagram of the scheme used for detection of symbol edges.

tion at baud rates of 300, 600, 1200, and 1600. The parameters used were $T_0 \approx 0.25$ s and $\tau_i = (i-1) * T/(M-1)$ where M is the smallest integer greater than or equal to the number of samples per baud. Tests were conducted on a signal set of total duration 3360 s, and in every case the symbol edges were detected correctly.

C. Phase Modeling

In this section, we show that the phase sequence of the baseband equivalent of the bauded signal can be approximately modeled as a straight line within each baud. Our approach will be to fit the phase sequence with the appropriate straight lines and then to vector quantize the residuals (the modeling errors).

To derive our model, we initially consider a noise-free bauded signal without pulse shaping. Such a signal can be written in the form

$$s(t) = A \cos [2\pi f_c t + p(t)] \tag{5}$$

where f_c is the carrier frequency and $p(t)$ is the phase of $s(t)$. Let the estimated carrier frequency be

$$f_{cq} = f_c + e \tag{6}$$

where e is the estimation error. The baseband signal obtained by using f_{cq} for demodulation is

$$s_{bb}(t) = 0.5A \ \{\cos [2\pi et - p(t)] + j \sin [2\pi et - p(t)]\}. \tag{7}$$

The phase of the baseband signal is easily found to be

$$p_e(t) = p(t) - 2\pi et. \tag{8}$$

Within a single baud interval, $p(t)$ is a constant for PSK signals and is linear with a nonzero slope for FSK signals. In either case, the phase sequence is linear inside the baud interval.

Once pulse shaping and channel noise are introduced into the system, the above results are no longer valid. However, one can assume in this more general case that the deviations from the straight line model are small. Moreover, the phase residual will in this case contain information about pulse shaping and channel noise.

We use linear regression to fit a straight line through the phase samples within each baud. The intercept $a(i)$ (see Fig. 1) of the straight line model is first computed and vector quantized. The slope of the model is then determined from the phase and the quantized intercept. Finally, the residual sequence is calculated as the difference between the phase sequence and the reconstructed straight line model and is then vector quantized.

III. Experimental Results

A. Description of Results

In this section, we evaluate the BRVQ system using as performance criteria the signal-to-quantization-noise ratio (SQR), the equivalent change in signal-to-noise ratio (ΔSNR), and the critical data rate [18]. These performance measures are defined below.

The SQR is defined as

$$\mathrm{SQR} = 10 \log_{10} \frac{\frac{1}{N}\sum_{t=1}^{N} x^2(t)}{\frac{1}{N}\sum_{t=1}^{N} [x(t) - y(t)]^2} \tag{9}$$

where $\{x(t)\}_{t=1,N}$ and $\{y(t)\}_{t=1,N}$ denote the input and output signal sequences, respectively, of a data compression system. This is a commonly used performance criterion that has the great advantage of simplicity. Unfortunately, it is only loosely related to the bit-error rate achieved using the compressed waveform. Since we are particularly interested in preserving the information-bearing sequence imbedded in the input signal, we seek alternative criteria that measure this aspect of system performance more accurately.

To define the change in SNR, let P_{b1} and P_{b2} be the bit-error probabilities incurred when the input and processed signals, respectively, are demodulated. If we assume that the noise in the signals (including quantization noise) can be modeled as additive white Gaussian noise, we can determine the equivalent signal-to-noise ratios (more precisely, the ratios of energy per bit to noise power spectral density) SNR_1 and SNR_2 that correspond to the bit-error probabilities P_{b1} and P_{b2}, respectively. Then the change in SNR, ΔSNR is defined as [6], [18]

$$\Delta\mathrm{SNR} = \mathrm{SNR}_1 - \mathrm{SNR}_2. \tag{10}$$

Even though the Gaussian model may not be especially accurate, ΔSNR is nevertheless a useful alternative to ΔP_b because of the steepness of the P_b versus SNR curves.

The critical data rate is defined [6], [18] as the data-transmission rate at which a compression algorithm first introduces errors into a noise-free input signal of a given duration. Let N_e be the number of bit errors in the compressed signals, C_L a confidence level, and P_{b0} a given threshold for the bit-error probability. Then the duration of the modem signals is chosen such that when $N_e = 0$, the bit-error probability of the quantized signal is less than P_{b0} with a confidence level of C_L.

A disadvantage of using either ΔSNR or the critical data rate is the tedious process of compressing and demodulating signals of sufficient length to estimate the bit-error probabilities of the signals involved. If the bit-error probability P_{b2} of the compressed signal is very small, the required signal duration will be unrealistically large. As a result, performance evaluation using these two criteria is usually done on signals with high bit-error probabilities. A detailed description of these performance measures can be found in [6], [18].

Several experiments with different quantizer parameters for data rates of approximately 16 kbits/s and below were conducted. The parameters used in all experiments other than those performed to obtain critical data rate are as shown in

TABLE I
BRVQ PARAMETERS USED IN THE EXPERIMENTS

Quantizer	Number of Codewords	Vector Dimension
Magnitude	128	20
Starting Value	512	4
Slope	128	4
Residual	512	10

TABLE II
SQR'S AND BIT RATES FOR DIFFERENT TYPES OF SIGNALS INSIDE AND OUTSIDE THE TRAINING SEQUENCE (UNCOMPRESSED BIT RATE = 64 kbits/s)

	BRVQ			ABCVQ	
Signal	Inside Training Sequence (dB)	Outside Training Sequence (dB)	Bit Rate (kbits/s)	Inside Training Sequence (dB)	Bit Rate (kbits/s)
CFSK (1200 baud)	28.32	27.44	14.9	23.3	16.0
CBPSK (1200 baud)	16.41	16.21	14.9	13.6	16.0
DBPSK (1200 baud)	26.98	26.04	14.9	18.4	16.0
DQPSK (1200 baud)	20.67	20.10	14.9	19.4	16.0
DOPSK (1600 baud)	18.52	17.79	16.5	18.1	16.0

Table I. While no claim is made here that the values in Table I are optimal in any sense, they were selected after a great deal of experimentation. The required carrier-frequency and baud-rate estimates were updated every 1024 and 4096 samples, respectively, and 81920 samples outside the training sequence were used for each experiment. The codebook design and the quantization of the data were performed using the squared-error distortion measure.

The results of the first set of experiments using the SQR measure are shown in Table II. In these experiments, the codebooks were designed for each modulation type (1200 baud CFSK, 1200 baud CBPSK, 1200 baud DBPSK, 1200 baud DQPSK, 1600 baud DOPSK) separately using 81 920 samples of noise-free signal as the training sequence in each case. For the purpose of comparison, the SQR's obtained from the ABCVQ [5] coder are also shown in Table II. In addition, some results obtained by encoding four of these signals using four common speech coders at 16 kbits/s are reproduced from [6] as Table III.

Examining the results of Table II, we see that the BRVQ algorithm outperforms the ABCVQ system for all signals. For DQPSK and DOPSK signals, the performance of the two coders are comparable, the advantages to BRVQ being 1.27 and 0.42 dB, respectively. For the other signals, the BRVQ algorithm performs from 3 to 8 dB better than the ABCVQ algorithm. Except for the DOPSK signal, the transmission rate of the BRVQ system is slightly less than that of the ABCVQ system. The results in Table III clearly demonstrate that both BRVQ and ABCVQ outperform several common speech coders when the waveforms to be compressed are modem signals.

When the modulation type is unknown or can change from time to time, the BRVQ algorithm can achieve the performance of the individually optimized codebooks by employing more than one codebook in parallel. There would, however, be a corresponding increase in the complexity of the algorithm. To avoid this problem, we designed the BRVQ system to employ just one codebook for each of the four

TABLE III
SQR PERFORMANCE DATA FOR SPEECH CODERS AT 16 kbits/s

	Algorithm			
Signal	APC	ATC	PCM	SBC
CFSK (1200 baud)	15.8	15.3	12.9	16.0
DBPSK (1200 baud)	11.5	15.3	10.5	14.1
DQPSK (1200 baud)	13.5	15.3	10.5	13.9
DOPSK (1600 baud)	13.3	14.7	10.9	13.4

TABLE IV
SINGLE CODEBOOK PERFORMANCE OF THE BRVQ SYSTEM IN TERMS OF SQR

Signal	Baud Rate	Inside Training Sequence (dB)	Outside Training Sequence (dB)	Bit Rate (kbits/s)
CFSK	300		18.79	11.3
CFSK	600		19.39	12.5
CFSK	1200	22.88	21.82	14.9
CBPSK	300		17.83	11.3
CBPSK	600		18.76	12.5
CBPSK	1200	14.84	12.51	14.9
DBPSK	300		17.91	11.3
DBPSK	600		18.87	12.5
DBPSK	1200	21.94	20.56	14.9
DQPSK	1200	17.45	16.94	14.9
DOPSK	1600	15.14	14.71	16.5

quantizers, regardless of signal type. The training sequence used for codebook design was a mixture of the five modulation types employed in the experiments of Table II (81920 samples of each type). The SQR values achieved using these composite codebooks for input signals inside and outside the training sequence are presented in Table IV.

Comparing the results of Tables II and IV, we see that when a single composite codebook is used in the BRVQ system, the performance as measured by SQR suffers by 3–6 dB relative to that achieved by individually optimized codebooks. However, the single-codebook performance of the BRVQ system is still comparable to that of the ABCVQ algorithm employing four parallel codebooks. It is important to point out that the BRVQ system performs very well on the 300 baud and 600 baud signals, even though none of these signals was included in the training sequence. These results demonstrate the robustness of the BRVQ system with respect to the various types of signals with different parameters. Moreover, the bit rates for these lower baud-rate signals are less than 16 kbits/s.

The low SQR obtained for the 1200 baud CBPSK signal is due to the process of low-pass filtering to obtain the complex baseband signal. The loss of fidelity is significant for the CBPSK signal because there was no pulse shaping performed on this signal. However, as we shall see, the increase in bit-error probability due to this degradation is small.

The remainder of the experiments described in this paper were conducted using a single composite codebook for all signal types.

The performance of the BRVQ system in terms of the change in SNR for five types of modem signal is presented in Table V. These results are for input signals with $P_{b1} = 10^{-2}$ and 10^{-4}. The duration of the signals (outside the training sequence) is sufficient to estimate P_{b2} within 5 percent with 90 percent confidence if P_{b1} is 10^{-2}, and within 60 percent with 90 percent confidence if P_{b1} is 10^{-4}. As a point of reference,

TABLE V
PERFORMANCE EVALUATION OF THE BRVQ ALGORITHM IN TERMS OF CHANGE IN SNR

	ΔSNR (dB)			
	BRVQ		ABCVQ	
Signal	$P_{b1} = 10^{-2}$	$P_{b1} = 10^{-4}$	$P_{b1} = 10^{-2}$	$P_{b1} = 10^{-4}$
CFSK (1200 baud)	−0.11	−0.072	0.13	−0.063
CBPSK (1200 baud)	−0.93	−0.93	−1.02	−1.39
DBPSK (1200 baud)	−0.87	−0.64	−0.84	−0.91
DQPSK (1200 baud)	−1.08	−1.05	−0.91	−1.45
DOPSK (1600 baud)	−1.93	−2.99	−1.83	−2.89

results obtained using the ABCVQ coder [5] for the same set of signals are also presented in Table V.

It can be seen from these results that the performance of the BRVQ system with one codebook is similar to that of an ABCVQ system that employs four parallel codebooks despite the fact that the data rate of the BRVQ system is 1.1 kbits/s lower than that of the ABCVQ system in four out of five test signals. Even though bit-error probabilities of 10^{-2} and 10^{-4} are much higher than those encountered in practice, experiments were performed using these values to avoid impractically large simulation times.

Experiments were performed to determine the critical data rates of the BRVQ system for five test signals outside the training sequence. In these experiments, the bits available at a given data rate are distributed among the different quantizers using the same percentages as those of Table I. Each signal was processed by the BRVQ algorithm at an initial data rate of 18 kbits/s, and data rate was then gradually decreased until errors were first found in the demodulated signal. The total number of bits processed for each signal was 46 050. This length is sufficient to determine the critical data rate with 99 percent confidence for a bit-error probability threshold P_{b0} of 10^{-4}. The critical data rates so obtained for each of the five types of signals are shown in Table VI.

TABLE VI
PERFORMANCE EVALUATION OF THE BRVQ ALGORITHM IN TERMS OF CRITICAL DATA RATES

Signal	R_c (kbits/s)
CFSK (1200 baud)	6.25
CBPSK (1200 baud)	7
DBPSK (1200 baud)	6
DQPSK (1200 baud)	14
DOPSK (1600 baud)	16.50

These results show that the BRVQ algorithm has the potential to reduce the data rate substantially below 16 kbits/s for most of the modem signals being tested. As one would expect, the more complex the signals require larger codebooks to represent them and therefore have higher critical data rates.

Our experiments have shown that the injection of errors into the signals by the BRVQ system is a rather abrupt process as the data rate is decreased. As an example, CFSK signals exhibited no errors at 6.5 kbits/s, while at 6.25 kbits/s, the bit-error probability was 1.2×10^{-3}.

B. General Discussion

Most of the computational burden in the BRVQ algorithm is due to the four vector quantizers. If $L1$, $L2$, $L3$, and $L4$ denote the number of elements in the codebooks designed for the phase residuals, the magnitude, and the starting values and slopes of the straight line model, respectively, and $N5$ is the number of samples of the modem signal per baud, the four vector quantizers require $L1 + L2 + (L3 + L4)/N5$ multiplications and about twice that many additions per sample when the squared-error distortion measure is used. In addition, a small amount of computation is required for estimating the straight line parameters using linear regression, and also for the passband-to-baseband conversion of the modem signals.

Most of the operations involved in estimating the side information can be performed using fast Fourier transforms, and can be performed separately from the vector quantization operations on a block-by-block basis. The system must also buffer the input data for the duration of the longest block used in the estimation of the system parameters. Despite being more complicated than the speech coders discussed earlier, the BRVQ algorithm can nevertheless be implemented efficiently using current VLSI technology, and can operate in real time on voiceband data signals.

When a single codebook tuned to a specific type of input signal is used, the BRVQ and ABCVQ algorithms have about the same computational complexity, with the BRVQ algorithm faring slightly better. The results of Table II indicate that the BRVQ algorithm performs better than the ABCVQ algorithm when their computational complexities are similar. Furthermore, the performance of the BRVQ algorithm using a single codebook for different types of input signals is comparable to that of an ABCVQ algorithm that uses four parallel codebooks, and in this case, the complexity of the former is substantially lower than that of the latter.

All the vector quantizers employed in the current version of the BRVQ system are of the full-search type. System complexity could be reduced by employing tree-search vector quantizers at some cost in performance as compared to full-search vector quantizers of the same codebook size. Some preliminary experiments conducted using tree-search codebooks have shown that SQR performance degradation is on the order of 1–3 dB.

One drawback of the current version of the BRVQ algorithm is the nonuniform bit rate arising from the fact that baud-by-baud phase modeling requires different number of bits for different baud rates. We are currently investigating ways to encode signals at a uniform bit rate. One method that is being studied is the use of embedded codebooks [19] in which some of the codewords are used only if the bit rate can accommodate them.

Several other experiments were conducted to test the robustness of the BRVQ system against errors in estimating the parameters. These experiments, which are discussed in detail in [20], have indicated that the performance of the BRVQ system is very insensitive to carrier-frequency estimation errors. This property is primarily due to the fact that the straight-line model of the baseband phase sequence compensates for any error in carrier-frequency estimation.

Despite the fact that the baud-rate estimator and the symbol synchronizer described in the paper have been found to be extremely accurate in practice, the effects of errors in baud-rate estimation and symbol synchronization were investigated empirically by deliberately introducing such errors into the system. These experiments have shown that properly designed codebooks can reduce the effects of erroneous baud-rate estimation and symbol synchronization.

IV. Conclusions

In this paper, we have described the baseband residual vector quantization system, a scheme designed to compress a class of voiceband data signals. This class includes signals with different types of modulation and a range of information rates up to 4800 bits/s. Our scheme exploits the common structure in the baseband phase sequences of the different types of bauded signals being studied. In particular, we showed that a straight-line model for the baseband phase works well for signals with different types of modulation, and furthermore, that a single composite codebook suffices for the entire class of bauded signals under consideration. As a byproduct of this study, we have developed a new carrier-frequency estimation scheme for bauded signals and an algorithm that accurately estimates the baud rate of a given signal.

Experimental results have shown that the BRVQ system is successful in compressing a relatively broad class of voiceband data signals at approximately 16 kbits/s. Performance evaluations using SQR, ΔSNR, and critical data rate have demonstrated that the BRVQ system works well with just a single composite codebook for each parameter being quantized. Further, the experiments with critical data rate suggest that the BRVQ algorithm is capable of reducing the data rate significantly below 16 kbits/s for many of the signals that were considered in this paper.

We did not consider the effect of channel errors on the performance of the BRVQ algorithm in this paper. However, it may be noted that the errors introduced in the channel are not propagated in time. The user must be careful to transmit the side information (carrier frequency, baud rate, etc.) with sufficient redundancy since any errors in these parameters will affect a large block of data.

References

[1] J. B. O'Neal, Jr., and R. W. Stroh, "Differential PCM for speech and data signals," *IEEE Trans. Commun.*, vol. COM-20, pp. 900–912, Oct. 1972.

[2] J. B. O'Neal, Jr., "Delta modulation of data signals," *IEEE Trans. Commun.*, vol. COM-22, pp. 334–339, Mar. 1974.

[3] P. J. May, C. J. Zarcone, and K. Ozone, "Voiceband data modem performance over companded delta modulation channels," in *Conf. Rec., 1975 IEEE Int. Conf. Commun.*, vol. III, June 1975, pp. 40.16–40.21.

[4] D. W. Petr, "32 kb/s ADPCM-DLQ coding for network applications," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 1982, pp. A.8.3.1–A.8.3.5.

[5] D. O. Anderton, "Vector quantization of voiceband data signals," Ph.D. dissertation, Univ. Utah, Salt Lake City, Dec. 1985.

[6] D. O. Anderton and C. K. Rushforth, "Waveform coding of voiceband data: Performance measures," submitted for publication to *IEEE Trans. Commun.*, 1988.

[7] 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.

[8] S. D. Lloyd, "Least-squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–137, Mar. 1982.

[9] D. C. Rife and R. R. Boorstyn, "Single tone parameter estimation from discrete-time observations," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 591–598, Sept. 1974.

[10] S. A. Tretter, "Estimating the frequency of a noisy sinusoid by linear regression," *IEEE Trans. Inform Theory*, vol. IT-31, pp. 832–835, Nov. 1985.

[11] D. W. Tufts and R. Kumaresan, "Estimation of frequencies of multiple sinusoids: Making linear prediction perform like maximum likelihood," *Proc. IEEE*, vol. 70, pp. 975–989, Sept. 1982.

[12] L. C. Palmer, "Course frequency estimation using the discrete Fourier transform," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 104–109, Jan. 1974.

[13] D. Slepian, "Estimation of signal parameters in presence of noise," *IRE Trans. Inform. Theory*, vol. IT-3, pp. 68–89, Mar. 1957.

[14] A. V. Oppenheim and R. W. Shafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[15] T. D. Tran, V. J. Mathews, and C. K. Rushforth, "A new carrier frequency estimation scheme for bauded signals," in *Proc. ICASSP'88*, New York, NY, Apr. 1988.

[16] R. B. Blackman and J. W. Tukey, *The Measurement of Power Spectra*. New York: Dover, 1958.

[17] L. E. Franks, "Carrier and bit synchronization in data communication—A tutorial review," *IEEE Trans. Commun.*, vol. COM-28, pp. 1107–1121, Aug. 1980.

[18] D. O. Anderton and C. K. Rushforth, "Performance measures for waveform coding of voiceband data signals," in *Proc. IEEE Global Telecommun. Conf.*, New Orleans, LA, Dec. 1985, pp. 753–757.

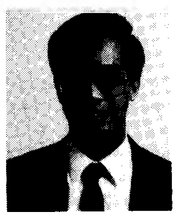
[19] A. Haoui and D. G. Messerschmitt, "Embedded coding of speech: A vector quantization approach," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 1703–1706.

[20] T. D. Tran, "Bauded signals compression using the baseband residual vector quantization algorithm," Ph.D. dissertation, Univ. Utah, Salt Lake City, Dec. 1987.

[21] D. A. Tugal and O. Tugal, *Data Communication: Analysis, Design and Applications*. New York: McGraw-Hill, 1982.

[22] B. Sklar, *Digital Communications: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

★



Thao D. Tran was born in Saigon, South Vietnam, in 1962. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Utah, Salt Lake City, in 1983, 1985, and 1987, respectively.

While at the University of Utah as a graduate student, he was involved in research projects in the area of data compression. He is currently with IBM Corporation, Manassas, VA. His research interests include sonar signal processing, image processing, speech processing, and communication theory.

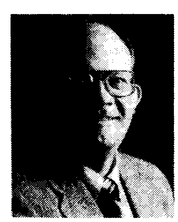
★

V. John Mathews (S'82–M'85) was born in Nedungadappally, Kerala, India, in 1958. He received the B.E. (Honors) degree in electronics and communication engineering from the University of Madras, India, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Iowa, Iowa City, in 1980, 1981, and 1984, respectively.

From 1980 to 1984 he held a Teaching–Research Fellowship at the University of Iowa, where he also worked as a Visiting Assistant Professor with the Department of Electrical and Computer Engineering from 1984 to 1985. He is currently an Assistant Professor with the Department of Electrical Engineering, University of Utah, Salt Lake City. His research interests include adaptive filtering, spectrum estimation, and data compression.

Dr. Mathews is an Associate Editor for the IEEE Transactions on Acoustics, Speech, and Signal Processing.

★



Craig K. Rushforth (S'59–M'63–SM'80) was born in Ogden, UT. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1958, 1960, and 1962, respectively.

He has served on the faculties of Utah State University and Montana State University, and has held research positions at Stanford Research Institute, the Institute for Defense Analyses, and ESL, Inc. Since 1976 he has been with the University of Utah, Salt Lake City, as a Professor of Electrical Engineering. He has served as Chairman of the Department of Electrical Engineering from 1982 to 1985 and from July 1988 to the present. His research interests lie in communication theory and signal processing, with current emphasis on multiple-access communications, VLSI implementation of algorithms, coded modulation, and signal restoration.