REAL-TIME DEMONSTRATIONS OF QUANTIZATION AND PREDICTION USING THE 'c31 DSK

J.W. Goode, **IBM Corporation**

S.A. McClellan, **UAB Elec. & Comp. Engr**

Abstract

In the study of data compression, signal processing, and digital communications (among others), the topics of quantization and linear prediction play a central role. For example, the finite wordlengths required for storage of digital filter coefficients as well as the number of bits used in "acquiring" a signal can be of significant concern in many applications.

The effect of quantization in these cases is often quite difficult to explain in a fashion which makes sense to undergraduate students. Since much of the theory surrounding quantizer optimization and optimal linear prediction is well-beyond the undergraduate level, and since a good understanding of these basic topics is increasingly important in the "information age", we have approached the problem from the perspective that "a good demonstration is worth a thousand equations".

To enable our demonstrations, we use an inexpensive TI 'c31-based DSK. The DSK is quite portable, flexible, has excellent analog input/output capabilities, and has more than adequate horsepower for our purposes. Using some assembly language and some C-language constructs, we have implemented a full-featured quantization and prediction platform for real-time acquisition, processing, and reconstruction of speech. We have found that the real-time demonstration of both quantization and prediction in a carefully selected progression is very helpful to undergraduate (as well as graduate) students in grasping these sometimes difficult concepts.

I. INTRODUCTION

Digital communications systems have been used for many years to transmit analog signals because of inherent advantages that digital encoding presents in comparison to analog transmission. Some advantages of digital communication include low susceptibility to transmission noise, applicability of digital encryption, and an integration of various types of data (e.g., computer data, digitized speech and images) in a common transmission format. In the study of digital communications as well as data compression and signal processing, the topics of quantization and linear prediction play a central role. Unfortunately, the effect of quantization is often quite difficult to explain in a fashion which makes sense to undergraduate students. Since much of the theory surrounding quantizer optimization and optimal linear prediction is well-beyond the undergraduate level, we have approached the problem from the perspective that "a good demonstration is worth a thousand equations".

To enable our demonstrations, we focus on the well-understood mechanism of speech production, and we utilize the inexpensive and quite flexible "DSP starter kit" from Texas Instruments. The starter kit is comprised of an evaluation board for the TMS320C31 50 MHz digital signal processor (DSP) which has the necessary analog/digital conversion features as well as an interface for communications with the personal computer. We have developed some software for the DSP which can be configured to illustrate pertinent concepts in quantization and prediction by processing speech in real-time.

Good quality speech coding at low bit rates has a growing number of applications (e.g., mobile telephony, voice-mail systems). Additionally, with the explosive growth of the Internet and World-Wide-Web, the effects of these technologies are increasingly present in routine settings. Consequently, the exposure to "quantized" and "compressed" information is very high whereas exposure to the theoretical underpinnings and a firm understanding of the associated tradeoffs is very low. We begin here with a brief introduction to the theory surrounding both the mechanics of speech production and the mathematical modelling of vocalization, including basic quantization and prediction. The dryness of the mathematical development is then nicely contrasted with the real-time demonstrations of speech coding which rely on a participant's vocalizations.

II. Human Speech and Linear Prediction

The encoding and transmission of human speech is one of the primary areas in which digital communications techniques are applied with broad success. Speech waveforms are acoustic pressure waves formed by the human vocal apparatus.[1] Different speech sounds are distinguishable by the human ear on the basis of their different short-time spectra and how these evolve in time. Hiss-like or fricative sounds, such as the spoken *"s"*, *"f'*, and *"sh"* have spectra which are like those of random noise. For a much larger class of speech sounds, known as *voiced speech*, (for example the spoken vowels, *"a"*, *"i" "e"*, and so on) the vocal cords vibrate approximately regularly, producing a quasi-periodic pulsed excitation of the vocal tract. The fundamental frequency of this excitation is known as the *pitch frequency*. Pitch variation during a spoken sentence is used in most languages to give structure to the sentence, as a kind of audible punctuation.

The vocal tract is a resonant cavity whose first four resonances (for adult speakers) are located at roughly 500Hz, 1500Hz, 2500Hz, and 3500Hz. As the tongue moves around in the mouth and the lips open or close, the shape of the vocal cavity changes causing these resonant frequencies to migrate, grow in energy, or disappear. It is this modulation of the acoustic wave passing out of the speaker's mouth which carries the spoken message. Thus, speech spectra are not arbitrary but are characterized by resonant peaks or *formants*, with each formant being determined by two parameters: center frequency and bandwidth.[12]

The characterization of a single vocal resonance by just two parameters is exploited easily through techniques of time series analysis. For voiced speech, waveform samples which are separated by time intervals equal to the period of the pitch frequency are known to be highly correlated. In addition, each sample is correlated with the immediately preceding samples because the resonances of the vocal tract "ring" for a finite time which equals their reciprocal bandwidth (roughly 10 msec).[2] So, at a sampling rate of 6400Hz, blocks of 64 samples exhibit significant correlation. Because of this interdependence, a sample of digitized speech data, s(n), can be approximated (or, predicted) by a linear combination of the immediately preceding M samples, $s(n-k), k=1,\ldots,M$ plus a random term, e(n):

$$s(n) = a_1 s(n-1) + a_2 s(n-2) + \ldots + a_M s(n-M) + e(n). \tag{1}$$

The weights in *(1)* are the *predictor coefficients,* and the number of coefficients, *M,* is the *order* of the predictor. The part of s(n) which cannot be represented by the weighted sum of M previous samples is *e(n),* the *prediction residual.* The prediction residual resulting from two typical predictions (short term or *formant* and long term or pitch) as in typical speech processing systems, is a highly unpredictable waveform of relatively small power.[3]

Quantization of the prediction residual and transmission of the spectrum of the speech signal through "side information" (such as predictor coefficients, order, and quantizer characteristics) is a method used commonly for the encoding of speech signals. Quantization noise contributes to the overall distortion of the reconstructed waveform. Approximation of the structure of the signal by autoregressive models or "linear prediction" is also a contributing factor.

In any case, the study of quantization and prediction techniques and the noise or *distortion* that they introduce is a very broad and highly important discipline involving sophisticated mathematics, equally sophisticated computer science, and a significant amount of intuition.

III. QUANTIZERS

The basic input-output characteristic of several scalar quantizers are shown in Figure 1, Figure 2, and Figure 3. In the figures, the amplitude of an input sample is shown on the horizontal axis, and the amplitude of the corresponding output sample is shown on the vertical axis. The purpose of a scalar quantizer is to map contiguous regions of the horizontal axis into each of several discrete values on the vertical axis. The horizontal axis is usually considered to be the support set of a continuous random variable, taking values from the real line. The vertical axis, or set of "reconstruction values" is usually a small set of real scalars. There are also *vector* quantizers which consider similar mappings in multidimensional spaces, but we only focus on scalar quantizers here. The reconstruction levels have a one-to-one relationship with a set of N integers so that

$$N = 2^b,$$

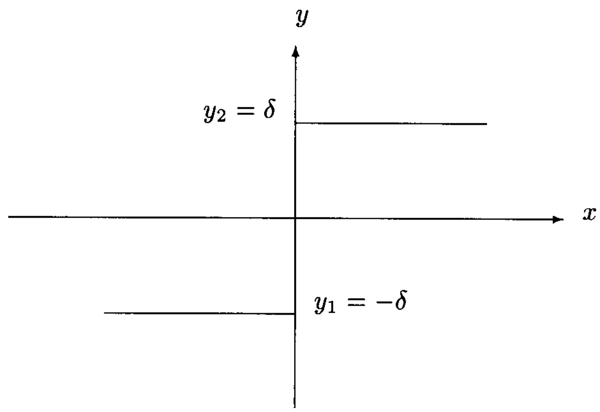

Fig. 1. 2-Level Quantizer.

where b is the number of "bits" required to encode and transmit the quantized sample. So, with a speech waveform sampled at 8000 samples/sec and a quantizer having 256 levels ($b = 8$), we must "spend" 64,000 bits to digitally represent (transmit or store) each second of speech. This rate, which is typical of the "toll quality" coding technique known as "μ-law PCM" is computed as

$$\left(8000\frac{\text{samples}}{\text{sec.}}\right)\left(8\frac{\text{bits}}{\text{sample}}\right) = 64,000\frac{\text{bits}}{\text{sec.}}.$$

When a constant relationship exists between the width of the regions on the horizontal axis and the values chosen for reconstruction levels, the quantizer is called *uniform.* In this case, the relationship is called the quantizer *step size*, and the step size is often denoted by the Greek letters Δ or δ. When no such relationship exists, the quantizer is called *non-uniform.* In either case, an "N-level" or "b-bit" quantizer approximates a continuous input value using the "best" reconstruction level at its disposal. In the process, the continuous-valued input amplitude is compressed in a "lossy" fashion because unrecoverable distortion is incurred.

The output characteristic of a 1-bit (two-level) quantizer is shown in Figure 1. The "midriser" quantizer has two reconstruction levels which are symmetric about zero, and the decision threshold for mapping the continuous amplitudes of discrete-time input sequence $\{x(n)\}$ into the discrete output levels $y_i = \pm\delta$ is chosen to be $x = 0$. That is,

$$\hat{x}(n) = y(n) = \begin{cases} +\delta \text{ if } x(n) \geq 0, \text{ and} \\ -\delta \text{ if } x(n) \leq 0 \end{cases} .$$

Often, quantizers are "optimized" by using some knowledge of the input sequence to minimize the average amount of distortion incurred in the approximation. The characteristics most commonly employed in this optimization are statistical descriptions of the input sequence's amplitudes, and the most common statistical description is the

Normal or Gaussian distribution function. So, a Gaussian-optimized, uniform quantizer is a uniform quantizer which has been optimized for input amplitudes which follow a Gaussian distribution.

The stepsizes, thresholds, and distortion figures for Gaussian-optimized *uniform* and *nonuniform* quantizers are shown in Table I. In Table I, N is the number of quantizer levels, Δ_{opt} is the stepsize between adjacent levels of the optimum quantizer,[4] x_i' is the threshold level for each step, and ϵ_{q*}^2 is the *performance factor* for the optimum quantizer.[3] The performance factor is the variance of the quantization error for a unit-variance input signal. For the nonuniform quantizer, δ_i is the reconstruction value for threshold level x_i'. Figure 2 and Figure 3 show representative 3-level and 5-level "midtread" quantizers.

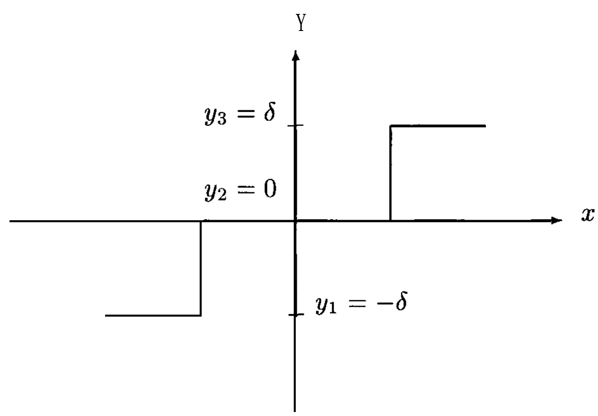

Fig. 2. 3-Level Quantizer.

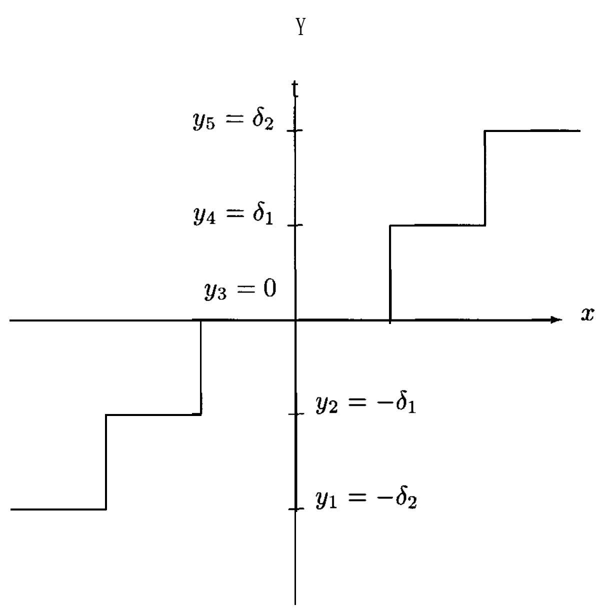

Fig. 3. 5-Level Quantizer.

TABLE I

GAUSSIAN QUANTIZER PARAMETERS.

Uniform Quantizer

N	Δ_{opt}/σ_d	x_i'	ϵ^2_{q*}
2	1.596	0.000	0.363
3	1.224	± 0.612	0.190
4	0.996	± 0.996 0.000	0.119
5	0.843	± 0.422 ± 1.265	0.0822

Nonuniform Quantizer

N	Δ_{opt}/σ_d	x_i'	ϵ^2_{q*}
2	1.596	0.000	0.363
3	1.224	± 0.612	0.190
4	$\delta_1 = 0.453$ $\delta_2 = 1.510$	± 0.982 0.0	0.118
5	$\delta_1 = 0.765$ $\delta_2 = 1.724$	± 0.382 ± 1.244	0.0799

IV. DIFFERENTIAL PULSE-CODE MODULATION

Often, *quantization* can be combined with *prediction* to produce quite different compression results, and dramatically better system performance. A block diagram of the classic Differential Pulse-Code Modulation (DPCM) system is shown in Figure 4. The principal components of the DPCM system shown are the quantizer Q, and the linear predictor, $H(z)$. The general DPCM system is primarily concerned with the generation and quantization of the prediction error signal, $d(n)$, from current samples of the input sequence, $x(n)$, and predictions based on past values of the reconstructed sequence, $\hat{x}(n)$. Careful selection of the order of the linear predictor and the corresponding predictor coefficients is important to the problem of minimizing the distortion introduced by the system. Equally crucial to the optimum performance of the DPCM system is the matching of the characteristics of the quantizer to the statistics of the prediction error sequence.

The DPCM system operates as follows. The predicted value at time instant n based on output values through time $n-1$, denoted by $\hat{x}(n|n-1)$, is subtracted from the input sample at time n, designated by $x(n)$, to produce the prediction error signal $d(n)$. The prediction error is then quantized, and the quantized prediction error, $u(n)$, is coded and transmitted to the receiver. Simultaneously with the coding, $u(n)$ is summed with $\hat{x}(n|n-1)$ to yield a reconstructed version of the input sample, $\hat{x}(n)$.[3,5] The basic

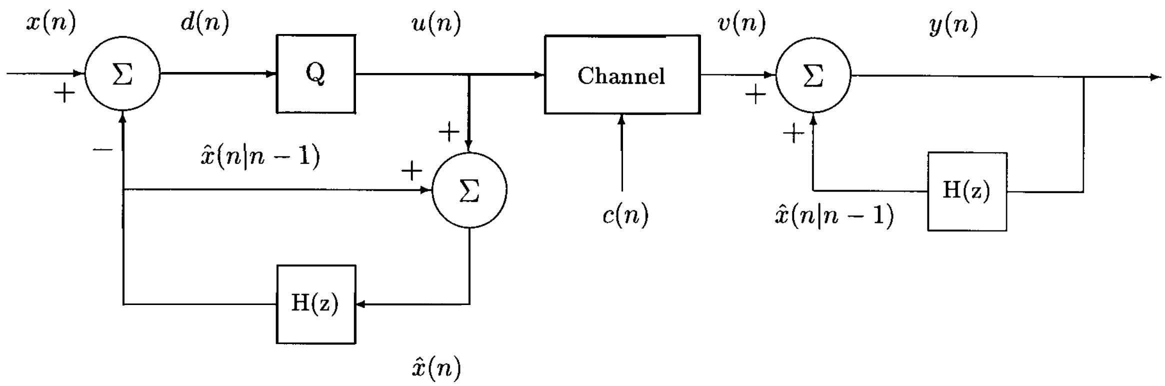

Fig. 4. Classic DPCM System.

equations defining DPCM are, from the preceding and Figure 4,

$$d(n) = x(n) - \hat{x}(n|n-1), \tag{2}$$

$$u(n) = d(n) - q(n),$$

$$y(n) = \hat{x}(n|n-1) + v(n),$$

and

$$r(n) = x(n) - y(n), \tag{3}$$

where $y(n)$ is the DPCM decoder's output approximation to coder input $x(n)$, $d(n)$ is the unquantized prediction error, $q(n)$ is the quantization error, $u(n)$ is the quantized prediction error, $v(n)$ is the receiver's version of $u(n)$, and $r(n)$ is the end-to-end reconstruction error. In the diagram, $c(n)$ is the noise introduced by the channel. Note that with error-free transmission of the quantized prediction error (i.e. $c(n) = 0$), then

$$v(n) = u(n)$$

and

$$r(n) = q(n), \tag{4}$$

or the reconstruction error is due entirely to the distortion introduced by the quantizer. So, the performance of the DPCM system in the absence of channel errors is described completely by the characteristics of the quantizer (i.e. number of levels, step size, thresholds) and by the compatibility of those *expected* characteristics with the *actual* statistics of the quantizer input sequence, *d(n)*.

TABLE II

SNR (DECIBELS) FOR DPCM SYSTEMS WITH VARIOUS QUANTIZERS AND MATCHED PREDICTORS OF ORDER 0, 1, AND 10.

N	Predictor Order 0	1	10
2	4.40	10.24	14.64
3	7.21	13.05	17.45
4	9.24	15.09	19.48
5	10.90	16.69	21.09

A. Prediction in *DPCM Systems*

DPCM systems typically use a linear predictor to form the predicted version of the input sequence samples, $\hat{x}(n|n-1)$, based on reconstructed values of previously processed samples. The linear predictor has the form

$$\hat{x}(n|n-1) = \sum_{k=1}^{M} h(k)\hat{x}(n-k), \tag{5}$$

where $\{h(k),\ k = 1,\ldots, M\}$ is a set of predictor coefficients.

The predictor structure is optimized by appropriate choice of coefficients $\{h(k),\ k = 1,\ldots M\}$. Usually, the "appropriate" coefficients are chosen to minimize the mean-square error between the predicted values and the actual values of the sequence. Using vector notation, a compact set of equations can be used to describe these coefficients and resulting prediction error. These equations are known as the *Normal equations* or the *Yule- Waker equations.* With slight modifications, these equations are used in computing optimum predictor coefficients for speech processing, where the technique is known as the *autocorrelation* method.

For reference, Table II shows the analytical Signal-to-Noise Ratio (SNR) for a DPCM system with optimal (matched) predictors and three different correlated sources. In the table, results for DPCM systems using quantizers with N = 2,3,4, and 5 levels are presented.

B. Application of Predictive Coding to Speech

The aim of DPCM is to reduce signal redundancy, producing a difference signal with a dynamic range matched to the characteristics of the quantizer. The periodic *adaptation* of important predictor and quantizer parameters to the time-varying characteristics of the input signal is an improvement to general DPCM which is used to account for the non-stationary nature of human speech. Adaptive DPCM (ADPCM) is the label commonly attached to systems which utilize the quasi-stationary nature of short-time speech signals to update and re-optimize quantizer gain and threshold points and linear predictor coefficients. The periodic readjustment of system parameters ensures that the mean-square error between the predicted and the true value of the signals is minimized over those portions of the signal which have roughly constant periodicity, as well as those which are not highly periodic.[6]

Both quantizer and predictor parameters can be adapted in a variety of ways, and with a variety of results. Adaptation algorithms for predictor or quantizer parameters can be loosely grouped based on the signals which are used as the basis for adaptation. Generally, *forward adaptive* coder elements analyze the input speech (or a filtered version of it) to characterize predictor coefficients, spectral components, or quantizer parameters in a blockwise fashion. *Backward adaptive* coder elements analyze a reconstructed signal, which contains quantization noise, to adjust coder parameters in a sequential or sample-wise fashion. Forward adaptive coder elements can produce a more efficient model of speech signal characteristics, but introduce delay into the coder's operation due to buffering of the signal. Backward adaptive coder elements do not introduce delay, but produce signal models which have lower fidelity with respect to the original speech due to the dependence on the noisy reconstructed signal. Most low-rate coders rely on some form of forward adaptation. This requires moderate to high delay in processing for accuracy of parameter estimation. The allowance of significant delay for many coder architectures has enabled a spectrally-matched pre- or post-processing step to reduce apparent quantization noise and provide significant perceptual improvements.[7]

V. Implementation Details

The effects of quantization and prediction on speech, and the tradeoffs between quantizer levels (rate), predictor order (complexity), quality, and adaptation algorithms can be experienced in real-time through use of the 'c31-based demonstration system hardware and software. The hardware consists of a personal computer (PC) running the demonstration control program, the Texas Instruments DSK evaluation board running the quantization, linear prediction, and adaptation algorithms, a microphone for speech input, and an amplifier and speakers for output. The DSK evaluation board is a part of the TMS320C3x DSP Starter Kit. The evaluation board contains a Texas Instruments TMS320C31-50 digital signal processor, analog input-output circuits and a parallel port interface for communications with the PC.

Software running in the PC controls the activation and parameter settings of the quantizers. The quantization and prediction software is stored on the PC and downloaded to the DSK for execution. The user can select and configure a quantizer, alter settings, and activate the quantizer from the keyboard in real-time while listening to the processed output. The following encoding methods can be configured: pass-through (no quantization), linear (uniform), forward adaptive, backward adaptive and predictive. The linear, forward and backward adaptive quantizers can be selected within the predictive mode, and a sequence field allows switching between two or more coding methods to compare their effects on the speech signals.

The debugger and loader program supplied with the DSK evaluation board is used to load the encoder programs into the DSK from the host PC. It can also be used to stop the operation of the DSK to allow setting of breakpoints and other debugging functions. The communications kernel supplied with the DSK evaluation board is loaded into the board to support the debugger functions and allow I/O operations between the DSK and the PC.

The control program will (under user command) collect buffers of unquantized and quantized data and save them to files for analysis. MATLAB is used to plot data collected from the DSK, and plots of the unquantized and quantized data in both time and frequency domains can be produced.

The demonstration program running in the PC was developed in the Borland C++ (DOS) environment. The quantizer routines were developed using the Texas Instruments TMS320C3x C Compiler and Assembler. The demonstration control program in the personal computer provides a DOS text based interface and runs under Windows 3.1 in the DOS prompt.

VI. Conclusion

We have developed a demonstration environment based on the TI ‘c31 DSP starter kit which enables the real-time demonstration of highly complicated and quite important quantization and prediction concepts. In this paper, we have also provided a version of the introduction to the theory of quantization and prediction used to guide the real-time demonstrations.

This platform is significant because the use of the extremely inexpensive ‘c31 DSK allows students to become simultaneously familiar with fundamental theory, practice, and tradeoffs involved in real-time programming for an important class of applications. Additionally, the demonstration environment uses basic topics in speech coding to produce a real-time, high-level feedback and quickly illustrate important concepts such as quantization and prediction. This platform will be used by the UAB Electrical & Computer Engineering Dept. in an undergraduate laboratory course on signal processing and assembly language. The speech processing content will be used in conjunction with other demonstrations of DSP technologies and algorithms. The quantization module(s) developed for this platform demonstrate both uniform and non-uniform quantization with user-selectable signal characteristics, as well as several popular approaches to quantizer adaptation (forward, or blockwise and backward, or sequential). The prediction module(s) also demonstrate different adaptation methods for closed-loop linear prediction.

Our demonstration generally begins with a speech signal and a mismatched non-adaptive quantizer, and examines the tradeoffs involved in quantizer levels (bit-rate) and signal fidelity. We then proceed into concepts of forward/backward quantizer adaptation, and we use various linear prediction approaches to further improve system performance. The demonstration system, with complete source code, has also been made available for students to use in a self-paced tutorial.

ACKNOWLEDGEMENT

This work was supported, in part, by IBM Corporation and by an NSF travel grant for the Workshop on Digital Signal Processing at Roger Williams University, August, 1997.

REFERENCES

1. J. Markel and A. G. Jr., Linear Prediction of Speech. New York, NY: Springer-Verlag, 1976.

2. M. Schroeder, "Linear predictive coding of speech: review and current directions," *IEEE* Communications Magazine, vol. 23, pp. 54-61, August 1985.

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

4. J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, **vol.** IT-6, pp. 7-12, March 1960.

5. J. Gibson, "Adaptive prediction for speech encoding," *IEEE ASSP Magazine*, vol. 1, pp. 12-26, July 1984.

6. B. S. Atal and M. R. Schroeder, "Adaptive predictive coding of speech," *Bell System Tech. J.*, vol. 49, pp. 229-242, October 1970.

7. S. A. McClellan and J. D. Gibson, "Speech signal processing: Coding, transmission, and storage," in *The Electrical Engineering Handbook* (R. Dorf, ed.), vol. 2 of *CRC* Handbooks, pp. 339-348, New York, NY: IEEE Press, October 1997.

J.W. GOODE is a software analyst with the IBM Corporation. He is pursuing a Ph.D. in Electrical Engineering at the University of Alabama at Birmingham (UAB). His primary interests are in Protocol Analysis, Signal Processing, Communications, and Coding Theory.

S.A. MCCLELLAN received his Ph.D. in Electrical Engineering from Texas A&M University in 1995. He is currently and Assistant Professor in the Dept. of Electrical & Computer Engineering at UAB. His primary interests are in Signal & Image Processing, Communication & Information Theory, and Stochastic Processes.