Robust Speech Coding for the Indoor Wireless Channel

Kevin W. Gould
Richard V. Cox
Nikil S. Jayant
Melvin J. Melchner

Alternative methods for digitally transcoding speech for radio transmission in an indoor environment have been investigated and compared to the CCITT standard, adaptive differential pulse code modulation (ADPCM).[1] These alternative coders are designed to minimize the effects of transmission errors on the quality of the transcoded speech. The coders compared are CCITT standard G.721 ADPCM, adaptive sub-band coding, and two other non-standard versions of ADPCM. In general, when packets of data are lost, the adaptive sub-band coder performs extremely well in terms of maintaining speech quality, as the sub-band synthesis filters fill out the gaps in speech. However, the sub-band coder requires the greatest levels of complexity and delay. The other ADPCM systems offer lower complexity and delay—at the expense of lower speech quality.

Introduction

One of the many decisions facing the designer of an indoor wireless, time division multiple access (TDMA) communications system is choosing a method for digitally encoding and compressing speech. Because of the limited bandwidth that is typically available, the speech is compressed to maximize the number of users on the system. A balance must be struck between the perceived quality of the speech resulting from this compression and the overall system cost. Other criteria that must be considered include: the end-to-end encoding delay; the algorithmic complexity of the coder; the DC power requirements; compatibility with existing standards; and the quality of the speech in which transmission errors occur.

A radio transmission channel is susceptible to many physical effects that tend to degrade a transmitted signal. These effects are familiar to the users of radio-based consumer electronics, such as FM car stereos or cordless telephones. Most users know the frustration of trying to tune a car stereo as the receiver moves beyond the *range* of the radio station's transmitter. There also may be locations within the range of the radio station's transmitter where our radio's reception is somewhat degraded, caused by *interference* from another transmitter.

When using a cordless phone, occasionally the quality of the voice we are receiving is unacceptable. We instinctively move our head, repositioning the antenna until the quality of the reception improves. This problem occurs because the cordless phone is receiving multiple signals. Although the base is only transmitting one signal, the signal is reflected off of objects in the room. The reflected signals combine at the phone's antenna, corrupting the resultant signal. This condition is called a *fade*. If the situation exists for a relatively long period of time, it is called a *deep fade*. The effects of these three problems are exacerbated when transmitting digital information using a wireless TDMA scheme, particularly in an indoor environment that is cluttered with physical objects, which tend to reflect radio signals.

In a TDMA system, the transmitted bit stream is divided in time into frames, typically on the order of several milliseconds. A six-millisecond frame is illustrated in Figure 1. A frame is further divided into six one-millisecond time slots, with each time slot assigned to a specific telephone call. Each slot contains a header and a packet of user data for the one wireless telephone call assigned to it. Generally, the header contains

Panel 1. Acronyms and Terms Used in This Paper

ADPCM — Adaptive differential pulse code modulation

CCITT — International Consultative Committee for Telephony and Telegraph

CPU — Central processing unit

FE — Frame erasure

Markov model — Used to predict the probaility of a system moving from one state to another.

MNRU — Modulated noise reference unit, an anchor condition to compare studies.

MOS — Mean opinion score

PCM — Pulse code modulation

SBERR — Single-bit error rate

SNR — Signal-to-noise ratio

TDMA — Time division multiple access

synchronization and addressing information for the user data. If the data in the header is sufficiently corrupted, because of one or more of the transmission problems described above, the entire slot will be lost and no more data will be available for that particular wireless telephone call until the next frame. The loss of an entire packet of data is called a *frame erasure*. During a deep fade, several sequential frame erasures may occur. Typically, however, transmission problems will not cause a frame erasure, but instead will cause a few bits in the header and user data to be corrupted. These are called *single-bit errors*. Frame erasures and single-bit errors are two types of *transmission impairments*.

In the past, many algorithms developed for encoding speech assumed a very robust, error-free transmission channel, such as copper wire or fiber optics. One such algorithm, adaptive differential pulse code modulation (ADPCM), has been standardized by the International Consultative Committee for Telephony and Telegraph (CCITT) as the coding method for toll quality voice at a 32 kb/s bit rate. Unfortunately, the perceived quality of compressed ADPCM speech degrades very rapidly as the rate of transmission impairment increases, for two reasons:

- The incorrect output of the coder is caused by the actual corrupted samples.
- The adaptive nature of the ADPCM algorithm decodes a sample, using both the sample itself and a history of previously decoded samples. Therefore, corrupted ADPCM samples are included in the history of the coder and will have a lasting, although slowly decreasing, effect.

Three alternative 32 kb/s coding techniques, recently designed to minimize the effects of transmission impairments on the perceived quality of transcoded speech, are attractive candidates for indoor wireless systems. The alternative coding techniques considered are: interleaved adaptive differential pulse code modulation with adaptive interpolation; post-filtered ADPCM with block-wise re-initialization; and partially-backward adaptive sub-band coding. In order to compare the performance of these alternative coders, they were subjectively evaluated in a mean opinion score (MOS) study. This document describes the transmission model used in evaluating the coders, each of the voice coders studied, the subjective evaluation, and the results of that evaluation.

Transmission Channel Model

Each of the voice coders studied was evaluated using a packet-based transmission channel modeled in software. Included in the simulated transmission channel is a simple impairment model, for the generalized indoor radio environment, which can inject both single-bit errors and frame erasures.

The model for injecting single-bit errors is very simple. Single bits in the simulated channel are pseudo-randomly corrupted at a pre-specified probability. Because deep fades, which occur in the indoor wireless channel, can span several sequential frames, a first-order Markov model is used in the simulated transmission channel to determine the occurrence of frame erasures. The Markov model, which provides a tool to predict the probability of a system moving from one state to another, is depicted in Figure 2.

In the Markov model, two states are defined. The simulator will be in either one or the other state, based on the results of the transmission of the previous packet. If the previous packet had been transmitted correctly (no frame erasure), then the simulator will be in the GOOD state. If the previous packet had been lost because of a frame erasure, then the simulator will be in the BAD state. The probability of a frame erasure occurring for the current packet is dependent upon the state of the simulator. In the model, the probability of a frame erasure occurring while in the GOOD state is P_{GB}, and the probability of a frame erasure occurring while in the BAD

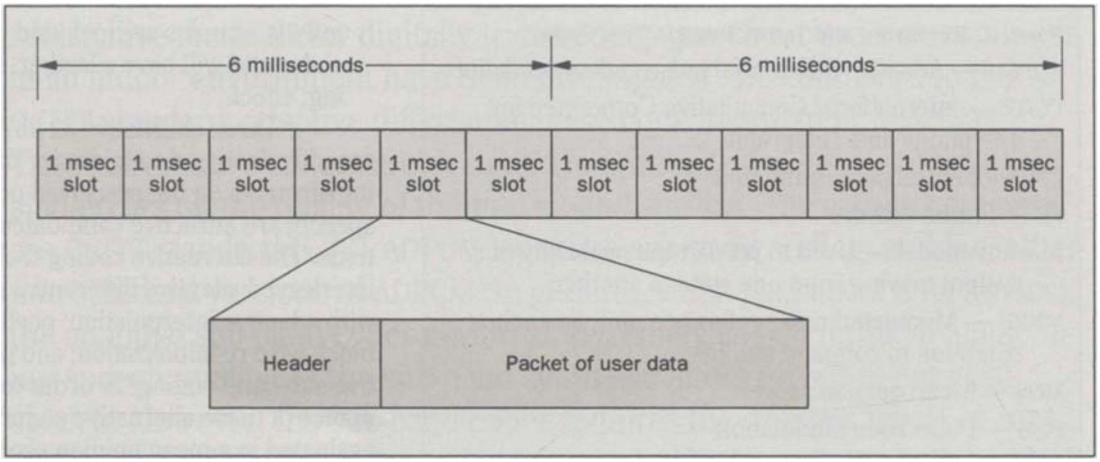


Figure 1. In a time division multiple access (TDMA) system, the bit stream is divided into frames. In this illustration, a 6-millisecond frame is divided into six 1-millisecond time slots. Each time slot consists of a header and a packet of user data.

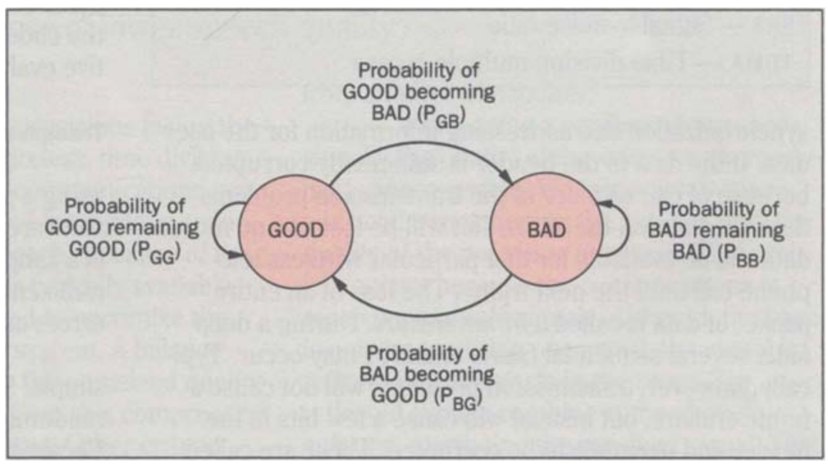


Figure 2. The first-order Markov model was used in a simulated channel to determine the occurrence of frame erasures in each of the four coders tested. If the system is in the GOOD state, P_{GG} is the probability it will remain in that state, and P_{GB} is the probability of a frame erasure. If the system is in the BAD state, P_{BB} is the probability it will remain in that state, and P_{BG} is the probability it will return to the GOOD state. All probabilities in the tests were specified before the simulator runs.

state is P_{BB}. Both of these probabilities are specified before the simulator runs. Frame erasures are then injected pseudo-randomly, based on the specified probabilities. By specifying that P_{BB} is greater than P_{GB}, frame erasures will tend to occur in sequential packets, thus simulating deep fades. The overall probability of a frame erasure occurring based on the two specified probabilities can be calculated with the following equation:

$$P_{Overall} = \frac{P_{GB}}{(1 - P_{BB} + P_{GB})}$$

Coders

Each of the coders investigated will inject a certain amount of delay into a wireless system, due to *buffering delay* and *algorithmic delay.*

Buffering delay is incurred because of the packet-based transmission channel. In a TDMA system, a transmitted packet contains exactly enough encoded voice samples to keep the decoder busy until a new packet arrives precisely one frame later. With careful buffer management in the transmitter, the oldest encoded voice sample in each packet will be, at best, no older than the duration of the frame. Received packets must be buffered, and the encoded samples decoded, at a rate that permits the buffer to empty at the moment that a new packet arrives. The end-to-end buffering delay in a TDMA system will be, at best, equal to the duration of the frame.

Algorithmic delay is caused both by the time the encoder requires to process the digitized speech sample into an encoded value, and the time the decoder requires to turn the encoded sample back into digitized speech.

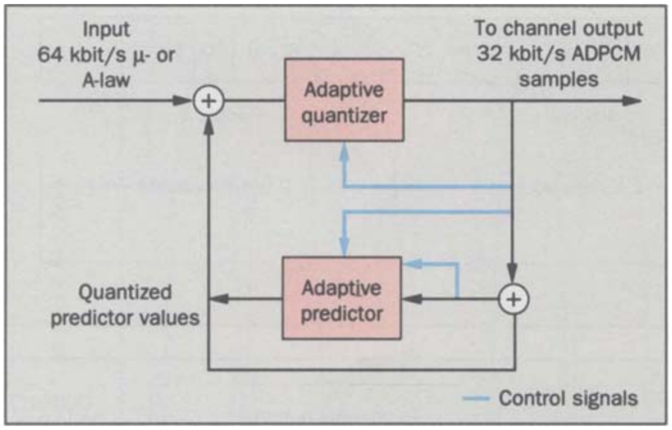


Figure 3. The input to the ADPCM coder is either 64 kb/s μ-law or A-law samples. The output is a 32 kb/s ADPCM sample.

The sum of these encoding and decoding delays is the end-to-end algorithmic delay.

Two key aspects of the computational complexity of a coder are:

- How many central processing unit (CPU) cycles are required to both encode and decode a digitized speech sample. Typically, the computational complexity is directly proportional to the DC power requirements of the coder.
- Memory usage, which can lead to increased chip size and greater power drain. For portable, battery-powered terminals, DC power consumption must be minimized to extend the battery life.

The complexity for each of the alternative coders was then estimated relative to the standard G.721 ADPCM.

Adaptive Differential Pulse Code Modulation. First, the G.721 ADPCM coder was evaluated using the previously described transmission channel simulator with a frame size of 10 milliseconds. A block diagram of the ADPCM coder is given in Figure 3.

Each time that a frame erasure occurs, the decoder has no input samples until the next frame arrives. In this implementation, the decoder will reuse the encoded voice samples from the most recently received packet for its input. Because of the relatively slow changing nature of ADPCM encoded voice, reusing the samples will minimize the lingering effects of the lost data by maintaining the state of the coder somewhat.

For the G.721 ADPCM coder, the buffering delay is 10 milliseconds, the algorithmic delay is 250 microseconds, and the total end-to-end delay is 10.25 milliseconds. Forty bytes of memory are required.

Interleaved ADPCM with Adaptive Interpolation. The next coder studied was a block-wise adaptive, fixed predictor 32 kb/s ADPCM coder. The predictor coefficient was held constant at 0.9, and the block size was five milliseconds. In any encoding scheme, a quantizer measures the signal of an analog sample and assigns it a digital code, based on where the signal sample falls within a specific range of step thresholds. In compressed encoding, an adaptive quantizer is used that varies the step size according to the energy level of the sample. The adaptive quantizer in the encoder is updated at the beginning of each block, and the new quantizer step size must be transmitted to the decoder before each block is decoded. The information to update the adaptive quantizer is quantized, or digitally encoded, into six bits and sent in the header of the slot. A quantized version of the step size is used in the encoder to keep the encoder and decoder quantizers locked.

This coder is then evaluated using the previously discussed channel simulator with a frame size of 2.5 milliseconds. One 5-millisecond block of data is buffered in the encoder, and the samples are interleaved in a *pair* of 2.5-millisecond packets.[2] First, a packet containing all odd-numbered samples from the block is transmitted. Then a packet containing the even-numbered samples is transmitted in the adjacent frame (illustrated in Figure 4). In the decoder, the pair of packets must be received, and the data reassembled into a 5-millisecond buffer, before decoding can begin.

If either of the packets in the pair is lost during the transmission, the missing data are reconstructed, using an adaptive interpolation scheme, before decoding begins. The adaptive interpolator is based on the encoder's calculation of the correlation of the data in the block. For example, if Packet 2 in Figure 4 was lost, the adaptive interpolator would reconstruct the samples in the lost time slots 2, 4, 6, 8, etc., using the correlation value of the samples in Packet 1 and Packet 2 of the block. This correlation value is quantized into eight bits at the time of encoding, and transmitted in the header of both packets of the pair. The following calculation is used in

Figure 4. Interleaved ADPCM with adaptive interpolation divides the bit stream into 5-millisecond blocks, and each block consists of two packets. One packet contains all odd-numbered samples of the block, and the other packet contains all even-numbered samples of the block. If either packet is lost in transmission, the adaptive interpolator reconstructs the data using the correlative value of the data in the block.

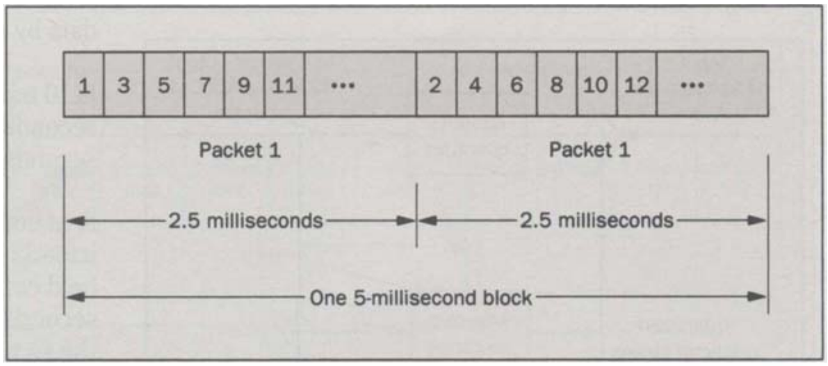
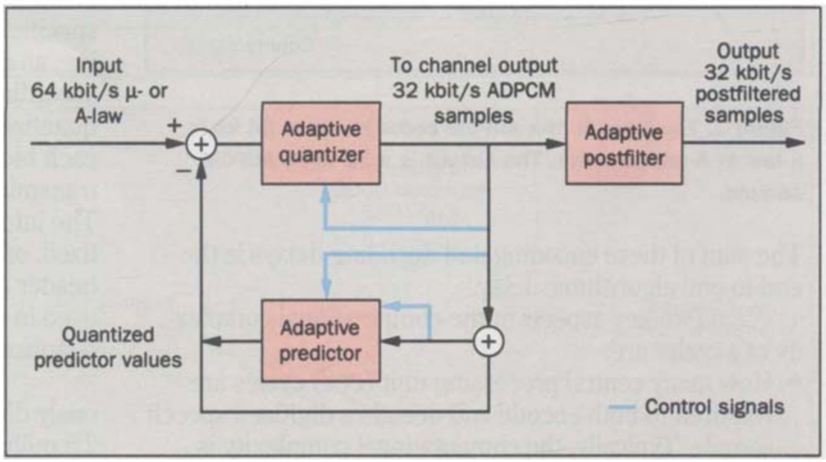


Figure 5. The postfiltered adaptive ADPCM uses an adaptive postfilter to improve the enhanced quality of the transcoded speech.



the decoder to interpolate the value of a lost sample:

$$A = \frac{C}{1 + C^2}$$

$$S_n = (A\ S_{n-1}) + (A\ S_{n+1})$$

where:

S_n = Missing sample

C = Quantized correlation value for current block

Using this interpolation scheme, the highly correlated *voiced* segments of speech, such as "em" sounds, are interpolated much more precisely than those segments that are *un-voiced*, such as "sch" sounds. Once the interpolation has been completed for the block, the decoding of the data proceeds normally. If both packets in the pair are completely lost during the simulated transmission, no interpolation is possible, and the value 0 is used in place of each missing ADPCM sample. In this case, the header information is also unavailable, and the most recently received correct value for the step size is used.

The header information required by this coder increases the overall bit rate above 32 kb/s. There are 14 bits of header information transmitted in every frame, or 28 bits per block, which increases the bit rate by 5.6 kb/s to 37.6 kb/s.

Because of the fixed predictor and block-wise adaptivity, this decoder is somewhat less complex than the G.721 ADPCM coder, even after allowing for some additional complexity to implement the adaptive interpolator. The memory required for this coder is the same as for G.721 ADPCM.

The algorithmic delay for this coder is very similar to the 250-microsecond algorithmic delay of G.721 ADPCM. The buffering delay, however, will be significantly greater and, for this reason, the short frame time of 2.5 milliseconds was selected. Because one packet contains half of the samples from a block of encoded voice samples in this coder, a packet cannot be

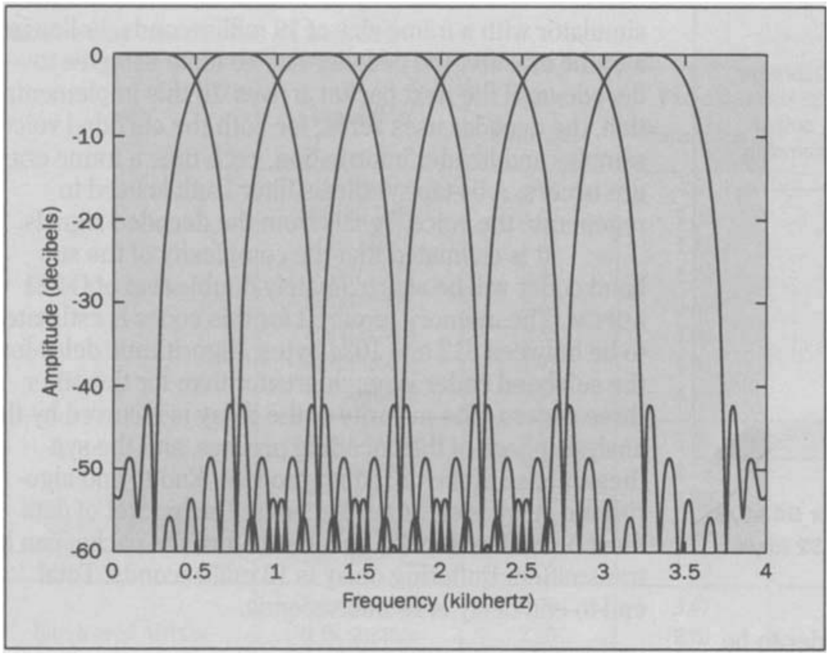


Figure 6. The sub-band coder defines seven 500-Hz sub-bands for the signal sample, which is taken every millisecond. The average energy of each band, measured over the entire 5-millisecond block, as well as the total energy of all five bands measured over the block, are the inputs to the bit-allocation algorithm. The more energy in a band, the more bits it is allocated.

transmitted until the entire block has been encoded. Decoding cannot begin until both packets containing all of the samples for a block have been received. Thus, there is one block's worth of delay added in both the encoder and decoder. The end-to-end buffering delay is 10 milliseconds, and the total end-to-end delay is then 10.25 milliseconds.

Postfiltered ADPCM with Block-wise Re-initialization. A G.721 24 kb/s ADPCM coder was simulated, and an adaptive postfilter was included as the final stage of the decoder, to artificially enhance the perceived quality of the transcoded speech[3] (shown in Figure 5). The control signals shown on this diagram are derived from control signals used to adapt the predictor in both the encoder and decoder. This coder was evaluated using the channel simulator with a frame size of 10 milliseconds.

The goal of this coder is to transmit information in addition to the ADPCM samples, allowing re-initialization of the quantizer and predictor at the beginning of each frame and, thereby, minimizing the lingering effects of frame erasures. Eighty bits were allocated for the transmission of this information, to keep the overall bit rate of this coder to 32 kb/s. In the current simulation of ADPCM, the adaptive properties of the quantizer and predictor are stored in 23 memory locations, or state variables. These 23 state variables are quantized into 67 bits, leaving 13 bits for possible error protection. After receiving a packet, the decoder reloads the 23 state variables before processing any of the ADPCM samples. Quantized versions of the 23 state variables are loaded at the beginning of each block in the encoder, as well. Doing this ensures that the quantizer and predictor in the encoder and decoder will be locked at the beginning of each frame. Use of the quantized state variables introduces a discontinuity at the beginning of each block. This decreases the signal-to-noise ratio (SNR) by less than 0.5 dB in the clear (transmission impairment-free) channel.

The complexity of the 24 kb/s G.721 coder is equivalent to the 32 kb/s G.721 coder. The additional complexity required to implement the quantization and loading of the 23 state variables is relatively small and can be ignored. The adaptive postfilter will increase the complexity by an additional 10 per cent. The memory required for this coder is the same as for G.721 ADPCM.

There are two components to the algorithmic delay of this coder: the end-to-end ADPCM transcoding and the postfiltering in the encoder. The transcoding delay is 250 microseconds, and the postfiltering delay is one millisecond. One packet of data must be buffered in the encoder before transmission. The buffering delay is then 10 milliseconds. The overall end-to-end delay is 11.25 milliseconds.

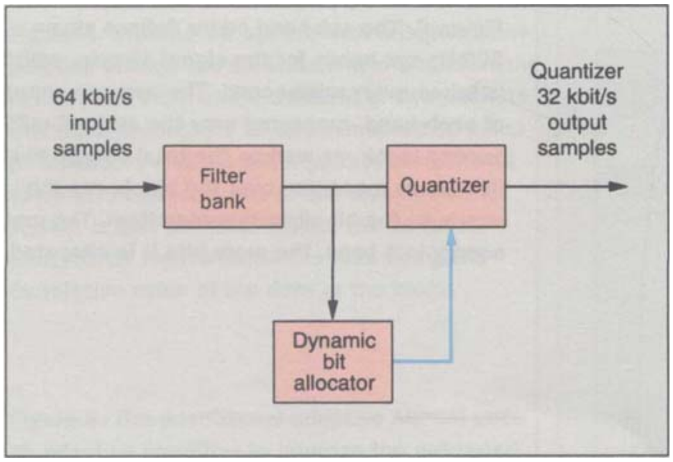


Figure 7. The input to the sub-band coder is either 64 kb/s μ- or A-law samples. The output is the quantized 32 kb/s samples.

Adaptive Sub-band Coder. The fourth coder to be studied was a 32 kb/s sub-band coder with a 5-millisecond block size. This coder defines seven 500-Hz sub-bands of the voice signal, equally spaced from DC to 3.5 kHz. The frequency response of the filters in the bank is shown in Figure 6. A block diagram of the coder is shown in Figure 7. In the first stage of the encoder, the average energy for each of the seven bands is measured every millisecond. These energy values are quantized into a number of bits determined by a dynamic bit allocation algorithm and transmitted to the decoder.[4] The average energy in each band, measured over the entire 5-millisecond block, as well as the total energy for all bands measured over the block, are the inputs to the bit-allocation algorithm. The more energy in a band, the more bits it is allocated. A maximum of five bits can be allocated to one band, and a total number of 24 bits is allocated for the samples from each of the seven bands.

The dynamic bit allocation algorithm must also run in the decoder, therefore, these average energies, in addition to the total energy, are transmitted in the header of the slot to the decoder. Each value is quantized into five bits for a total of 40 bits. Quantized versions of the values are used in the encoder so that the bit allocation algorithm produces identical results in both the decoder and encoder.

The coder was evaluated using the channel simulator with a frame size of 10 milliseconds. Following a frame erasure, the decoder has no input samples to decode until the next packet arrives. In this implementation, the decoder uses zeros, for both the encoded voice samples and header information, each time a frame erasure occurs. A 64-tap synthesis filter bank is used to regenerate the voice signals from the decoded signals.

It is estimated that the complexity of the sub-band coder will be approximately double that of G.721 ADPCM. The memory required for this coder is estimated to be between 512 and 1024 bytes. Algorithmic delay for the sub-band coder is much greater than for the other three coders. The majority of the delay is incurred by the analysis phase of the encoding process, and the synthesis phase of the decoding process. End-to-end algorithmic delay is eight milliseconds. One packet of data must be buffered in the encoder before the packet can be transmitted. Buffering delay is 10 milliseconds. Total end-to-end delay is 18 milliseconds.

Subjective Performance

In order to subjectively evaluate the performance of the alternative coders, a mean opinion score (MOS) study was performed. Each of the coders under study was included with several simulated *channel conditions.* These channel conditions were created by selecting different types and rates of impairments. The following were used:

- Clear channel (no errors)
- Pseudo-random single-bit errors injected at a rate of 0.0001
- Frame erasures with Markov probabilities: P_{GB} = 1%, P_{BB} = 10%, $P_{Overall}$ = 1.09%
- Frame erasures with Markov probabilities: P_{GB} = 3%, P_{BB} = 10%, $P_{Overall}$ = 3.22%
- Frame erasures with Markov probabilities: P_{GB} = 10%, P_{BB} = 10%, $P_{Overall}$ = 10%

A total of 19 subjects participated in the study, all of whom were employees of AT&T Bell Laboratories. Thirteen of the subjects were male and six were female. Each of the subjects was given identical instructions and listened to identical utterances. The test began with 10 practice trials. The samples, including the practice trials, were presented to the listeners in four different orders to help eliminate experimental bias.

Eight examples of each coder were presented to

Table I. MOS Results For All Subjects

Coder	Condition	Male Examples	Female Examples	All Examples	Number of Responses				
					Bad	Poor	Fair	Good	Exc
64 kb/s μ-Law	Original	4.61	4.50	4.56	0	0	6	59	95
	MNRU 5dB	1.15	1.25	1.20	132	24	4	0	0
	MNRU 10dB	1.45	1.71	1.58	77	73	10	0	0
	MNRU 15dB	2.31	2.79	2.55	12	65	67	15	1
	MNRU 20dB	3.17	3.69	3.43	1	19	70	50	20
	MNRU 25dB	3.86	4.09	3.97	1	1	37	83	38
	MNRU 30dB	4.41	4.35	4.38	0	0	17	65	78
	MNRU 35dB	4.44	4.39	4.41	0	0	14	66	80
G.721 32 kb/s ADPCM	Clear Channel	3.91	3.89	3.90	1	3	36	91	29
	0.1% SBERRs+	2.03	2.39	2.21	29	79	43	8	1
	1% 10% F.E.*	3.19	3.29	3.24	4	23	75	47	11
	3% 10% F.E.	2.38	2.58	2.47	16	60	76	8	0
	10% 10% F.E.	1.13	1.14	1.13	139	21	0	0	0
Interleaved ADPCM with Adaptive Interpolation	Clear Channel	3.20	3.47	3.34	0	13	89	49	9
	0.1% SBERRs	2.65	2.79	2.72	4	52	90	13	1
	1% 10% F.E.	2.86	3.06	2.96	0	35	99	23	3
	3% 10% F.E.	2.26	2.58	2.42	8	84	61	7	0
	10% 10% F.E.	1.63	1.91	1.77	55	88	16	1	0
Postfiltered ADPCM with Re-initialization	Clear Channel	3.56	3.79	3.67	0	4	66	68	22
	0.1% SBERRs	2.17	2.50	2.34	21	76	51	12	0
	1% 10% F.E.	2.46	3.28	2.87	5	50	68	35	2
	3% 10% F.E.	2.28	2.44	2.36	20	69	65	6	0
	10% 10% F.E.	1.61	1.71	1.66	69	77	13	1	0
Sub-band Coder	Clear Channel	4.40	4.43	4.41	0	0	9	76	75
	0.1% SBERRs	2.06	2.65	2.36	31	62	49	15	3
	1% 10% F.E.	3.97	4.01	3.99	0	2	32	91	35
	3% 10% F.E.	3.31	3.49	3.40	0	12	87	46	15
	10% 10% F.E.	1.85	2.04	1.94	39	92	28	1	0

+SBERRs is an abbreviation for pseudo-random single-bit errors.
*F.E. is an abbreviation for *frame erasure*. The two percentages given in the table are P_{GB} and P_{BB} for the Markov Model. For example, the "1% 10% F.E." means there is a 1 per cent chance of the system going from the GOOD state to the BAD (P_{GB}) and a 10 per cent chance of the system remaining in the BAD state (P_{BB}).

the subjects for each of the above channel conditions. Four of the examples were recorded by male speakers and four were recorded by female speakers. Each example consisted of two short sentences separated by silence, thus enabling the listener to judge the quality of coded silence, as well as speech.

Several *anchor conditions* were included in the study. These included 64 kb/s μ-law for all eight examples, as well as software-generated CCITT Modulated Noise Reference Unit (MNRU) examples at 5dB, 10dB, 15dB, 20dB, 25dB, 30dB, and 35dB for all examples. These anchor conditions are typically included in all MOS studies to calibrate the results of one study to another.

In all, there were 224 two-sentence utterances for each subject to evaluate. The utterances were presented to both ears of the subjects using headphones in a double-walled sound booth. Because headphones were used instead of a telephone handset, the inter-coder

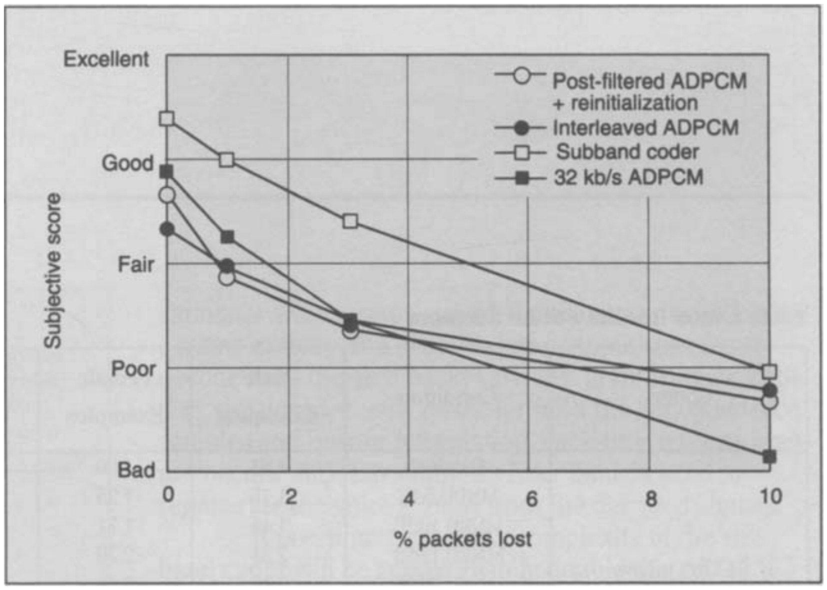


Figure 8. This table shows the perceived rating of the test subjects (from 5=excellent to 1=bad) as a percent of packets lost during transmission. The sub-band coder samples were rated as higher in perceived quality than the samples of the other coders tested.

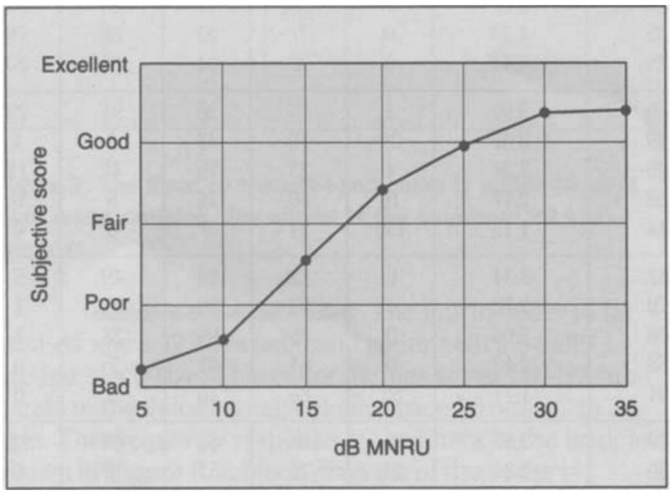


Figure 9. This table shows one of several anchor conditions included in the study. Software-generated CCITT Modulated Noise Reference Unit (MNRU) examples are shown at 5dB, 10dB, 15dB, 20dB, 25dB, 30dB, and 35dB for all 8 examples. These anchor conditions are typically included in all MOS studies to calibrate the results of one study to another.

MOS differences were exaggerated in relation to previous MOS studies using telephone handsets. After rating 60 samples, each subject was given 15-second breaks. The test typically took 50 minutes to complete.

The subject rated each example using the following scale: 5=Excellent, 4=Good, 3=Fair, 2=Poor, and 1=Bad.

Results

The MOS score for each coder/condition was determined by averaging all of the responses, for each of the eight examples, for each subject in the population. The results are presented in Table I. Included are the MOS scores for coded male speech, coded female speech, as well as the score for coding both male and female speech. The count of each response (Bad -> Excellent) for each coder/condition also is given in the table. Figure 8 shows the MOS score vs. percent packet loss for the data given in Table I. The data points shown for 1% packet loss are for the Markov probabilities P_{BB} = 1% and P_{GB} = 10%. The data points shown for 3% packet loss are for the Markov probabilities P_{BB} = 3% and P_{GB} = 10%. The data points shown for 10% packet loss are for the Markov probabilities P_{BB} = 10% and P_{GB} = 10%.

Figure 9 shows the MOS Score for the MNRU anchor conditions for all subjects in the study.

Discussion

There is a wide variety of criteria that must be considered when selecting a coder for a digital wireless telecommunications system. It is not possible to make a design recommendation without weighing all system parameters and customer needs. However, some general comparisons are made to aid the system designer in selecting a coder.

Results indicate that the performance of the subband coder is superior to all others studied for both the clear channel and when errors are introduced. The benefits of this coder come at the expense of complexity and delay. The delay, while not perceptible to the human ear, may require the system to include echo control, which can be quite costly, to achieve acceptable voice quality.

The G.721 coder has good performance in the clear channel. However, it has poor performance, with errors, when compared to the three alternative coders. Because G.721 has been an industry standard since 1986, it is well understood and supported by many silicon vendors, thus affording widely available, cost-effective solutions.

Both the interleaved ADPCM and re-initialized ADPCM coders suffer from reduced quality in the clear channel, but each outperforms G.721 ADPCM as the frame erasure rate increases. The re-initialized ADPCM coder performs well in the clear channel, but is more complex due to the postfilter. The interleaved ADPCM solution has the lowest complexity of all four coders, but has relatively high delay.

References

1. CCITT Req. G.721, Bluebook, Geneva 1986.
2. Jayant, N.S. and Christensen, S.W., "Effects of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation Procedure," *IEEE Transactions on Communications*, Vol. Com-29, pp. 101–109, February 1981.
3. Ramamoorthy, V., Jayant, N.S., Cox, R.V., Sondhi, M.M., "Enhancement of ADPCM Algorithms for Postfiltering and Noise Feedback," *IEEE Journal on Selected Areas in Communications*, Vol. 6. No. 2, pp. 364–382, February 1988.
4. Cox, R.V., Gay, S.L., Seshadri, N., Shoham, Y., Quackenbush, S., and Jayant N.S., "New Directions in Sub-band Coding," *IEEE Journal on Selected Areas in Communications*, Vol. 6, pp. 391–409, February 1988.

(Manuscript approved August 1993)

Kevin W. Gould *is a member of technical staff in the Premises Wireless Small Systems Development Department at AT&T Bell Laboratories, Middletown, N.J. He joined AT&T 1981 and is a software engineer developing platforms for premises-based wireless communications products. He has a B.S.E.E. from Monmouth College, West Long Branch, N.J.*

Richard V. Cox *is Head of the Speech Coding Research Department at AT&T Bell Laboratories in Murray Hill, N.J. His organization conducts research in speech transmission and storage. He joined AT&T in 1979, and has a B.S.E.E. from Rutgers University, and a M.S. and Ph.D. in electrical engineering from Princeton University.*

Nikil S. Jayant *is Head of the Signal Processing Research Department at AT&T Bell Laboratories in Murray Hill, N. J. His organization is responsible for research in speech and image processing, with applications in coding, communications, and recognition. He joined AT&T in 1968. He has a Ph.D. in electrical communications engineering from the Indian Institute of Science, Bangalore, India.*

Melvin J. Melchner *is a member of technical staff in the Speech Coding Research Department at AT&T Bell Laboratories in Murray Hill, N. J. He is involved with developing software tools for DSP programming environments and implementing signal processing algorithms to run in real time. Mr. Melchner joined the company in 1971. He has a B.S in engineering science from New York University, an M.S. in physics from the University of Illinois, and an M.S. in computer science from Stevens Institute of Technology.*